



HAL
open science

A Reduced LPV Polytopic Look-Ahead Steering Controller for Autonomous Vehicles

Dimitrios Kapsalis, Olivier Sename, Vicente Milanés, John Jairo Martinez Molina

► **To cite this version:**

Dimitrios Kapsalis, Olivier Sename, Vicente Milanés, John Jairo Martinez Molina. A Reduced LPV Polytopic Look-Ahead Steering Controller for Autonomous Vehicles. *Control Engineering Practice*, 2022, 129 (December), pp.105360. 10.1016/j.conengprac.2022.105360 . hal-03812669

HAL Id: hal-03812669

<https://hal.science/hal-03812669>

Submitted on 12 Oct 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial - NoDerivatives 4.0 International License

A Reduced LPV Polytopic Look-Ahead Steering Controller for Autonomous Vehicles

Dimitrios Kapsalis^{1,2}, Olivier Sename², Vicente Milanes³ and John J. Martinez²

Abstract—This paper presents a novel design of a Linear Parameter Varying (LPV) controller based-on the polytopic approach, for the path-following system of an automated vehicle. The implementation of the proposed steering system is based-on the reduction of the initial 3D (3 Dimensional) polytope that, due to the conservatism of the specific approach suffers from over-bounding. The proposed algorithm aims at tightening the polytope by reducing the vertices and the volume of the polytope that includes the parameter variations i.e the longitudinal velocity of the car, its inverse and, the look-ahead distance of the steering algorithm. In this paper, a novel real-time implementation of the reduced LPV/Polytopic controller is proposed as a real-time optimization problem according to the current measured values of parameters. The novel control system, has been validated in high-fidelity simulations, and to a real-life experiment at a test-bed platform (Renault Zoe) in a private test track for low and higher speeds showing encouraging performance and sustaining good tracking and comfort at the same time.

I. INTRODUCTION

Automated cars can provide different services according to the selected Operational Design Domain (ODD) [1]. To achieve the desired performance under some specific conditions where the vehicle is intended to function, several components that compose the overall architecture of the vehicle, must cooperate successfully. The more important parts of such an architecture, according to the existing bibliography may be: (i) Perception, (ii) Navigation (NAV) and (iii) Control Actuation [2]. Similar cognitive architectures have been deployed and demonstrated in [3], [4], [5].

The control part of the car mainly addresses the problems of following (a) a path (Lateral Control), (b) a generated longitudinal speed profile or another vehicle (Longitudinal control). A combination of the two previous can be designed, for some cases e.g. during a lane-change in a highway or an obstacle avoidance (Integrated Lateral/Longitudinal Control) [6].

Lateral control refers to the capability of the vehicle to follow a pre-defined path via steering feedback control [7]. Several control algorithms exist and were implemented for

the steering control of an automated vehicle. [8] presents two different approaches based-on receding horizon control that compute the front steering angle to track a path in slippery road condition, while state constraints are treated. More specifically, a Non-linear Model Predictive Controller (NMPC) has been tuned for the non-linear vehicle model, which is discretized off-line for a given sampling time. The second approach refers to the design of a linear MPC for the successive linearized and discretized vehicle model. In [9], preview control theory has been applied to design a lateral control system that achieves a smooth steering operation by taking into account as preview information the future road curvature. [10] applied robust control techniques and in particular the loop-shaping tuning method. The H_∞ steering controller is robust under vehicle model uncertainties of a tractor semi-trailer. Furthermore, [11] presents an adaptive lateral control scheme where the steering controller is self-tuned, while showing good tracking performance and robustness to disturbances. In addition, a more recent adaptive scheme [12], has been combined within the MPC framework aiming at improving lateral stability and performance under various road conditions. The adaptive scheme aims at improving the prediction accuracy of the vehicle model, via a data-driven online identification procedure updating the model which is utilized by the predictive controller to compute the front steering control input.

Gain-scheduling and more advanced LPV approaches emerged as a way to handle parameter variations and model non-linearities, while performance and stability are guaranteed [13]. The last decades LPV control techniques have been successfully utilized in real-world applications [14], and in particularly for automotive problems [15]. Moreover, LPV lateral control approaches already exist in the bibliography. [16] applied the LPV/Gridding approach for the lateral control on tractor-trailers. The velocity's parameter space is gridded for a selected step and LTI controllers are computed via Linear Matrix Inequalities (LMIs) for every grid point. In former studies [17], the authors have utilized the same approach but the key difference is that the look-ahead time of the steering framework is expressed as a continuous function of the speed. Another interesting approach to treat the longitudinal speed, which is an inherent parameter in the lateral control problem, is the polytopic approach. According to that method, the parameter set is represented by a polytope which is defined by the combinations for the maximal values of the parameters. Due to that fact, the polytope may be over-bounded, meaning that the volume of the geometrical representation is far larger than the “true” parameter tra-

¹Dimitrios Kapsalis is with the Research Department, Renault SAS, 1 Avenue de Golf, 78280 Guyancourt, France dimitrios.kapsalis@renault.com

²Dimitrios Kapsalis, John Jairo Martinez Molina and Olivier Sename are with GIPSA-lab, Control Systems Department, 38000 Grenoble, France dimitrios.kapsalis@grenoble-inp.fr john-jairo.martinez-molina@grenoble-inp.fr olivier.sename@grenoble-inp.fr

³Vicente Milanes is with the Spanish Engineering Division, Renault España S.A, 72 Avenue Madrid, 47008 Valladolid, Spain vicente.milanes@renault.com

*This work was supported by Renault.

jectory, and the associated LPV controller cannot satisfy the closed-loop performance adequately. For these reasons, several attempts have been made to reduce the size of the polytope for the scenario of tracking a reference trajectory, and provide a less conservative control-oriented model. In particular in [18], a 2D (2 Dimensional) polytope including only the speed and its inverse is reduced even though a parameter dependent weighting function is utilized for the tuning of the LPV controller and thus leading to a 3D polytopic model which is not treated completely. More recently [19] presents a complete review of the different methods utilized so far to reduce the size and the number of vertices for the 2D polytopes for the synthesis problem of a LPV/Polytopic controller.

A. Extension & Contributions

This paper stands as an extension and improvement of [20]. More specifically, in the former study [20], the lateral control system is based on minimizing errors computed at a look-ahead distance expressed as a linear function of the velocity (so a constant look-ahead time), in front of the car w.r.t. the reference path. The design of the LPV/Polytopic controller is based-on the reduction of a 2D polytope as done in the literature. In this work, the contributions over the previous work and existing attempts collected in [19] to reduce the conservatism of the polytopic approach to design a lateral control system can be summarized below:

- The look-ahead distance is considered here as an extra varying parameter. Thus, a complete control design is achieved over a larger range of selected look-ahead distance profile and of velocity in combination with the LPV control theory. Consequently, the autonomous vehicle can track sufficiently the trajectory for lower and higher speeds while its tracking performance is adjusted.
- A shrinking algorithm is proposed that reduces the initial over-bounded 3D 8-vertices polytope to the final 4-vertices one, while the convexity property is still satisfied. The synthesis of the reduced LPV/Polytopic controller improves significantly the tracking performance over the primal work presented in [20]. In addition, compared to the grid-based approach developed in [17], where the control implementation needs the interpolation of a far larger number of LTI controllers, the implementation of the proposed reduced polytopic approach only needs to compute on-line a combination of a set of 4 LTI controllers (designed off-line).
- A novel real-time implementation procedure of the discretized and reduced LPV/Polytopic controller is presented. It is based-on the solution of a real-time constrained Least-Squares (LS), which mathematically is expressed as a quadratic programming problem. That implementation is validated in a test-bed platform at a test track, proving its embedded-applicable formulation and validity, as it can be seen in the following video recorded during the experiments: <https://drive.google.com/>

file/d/1-VBEj528LXWbQWn89r9GxYU-N2Vj_KJF/view?usp=sharing .

B. Structure of the paper

The following sections of the paper are organized as follows. Section 2 introduces the lateral control framework, and the complete model of the steering dynamics of the car including the errors and the actuator’s identified model. Section 3 presents the proposed algorithm to tighten the 3-dimensional parameter polytopic set and the associated LPV model. Then, section 4 details the LPV Polytopic Controller synthesis in the H_∞ framework and section 5 describes the off-line and on-line steps of the novel implementation for embedded application. Sections 6 & 7 provide the simulation and experimental results of the novel control scheme, respectively. Section 8 sums up the paper and proposes extensions & future perspectives.

II. PATH-FOLLOWING CONTROL FRAMEWORK

This section presents the modelling of the vehicle’s lateral dynamics, as well as of the look-ahead trajectory tracking model. In particular, the modeling of look-ahead errors is given, including the identified model of the EPS system.

A. Two-Wheeled Dynamic Bicycle Model Formulation

Fig. 1 depicts the bicycle model that captures the lateral motion of the car [21]. The resulted dynamics are expressed in the state space model where the longitudinal velocity v_x is considered as a varying parameter. In that sense, the longitudinal motion of the vehicle can be decoupled from the lateral one and a controller per scenario can be designed independently from the other. As states are considered the lateral speed v_y and the yaw-rate $\dot{\psi}$ of the car. The control input is the steering wheel angle applied to the front wheels of the car δ . α_f , α_r are the tire side-slip angles of the front and rear wheels respectively. β is the side-slip angle of the vehicle body. L_f , L_r are the distances of the front and rear wheel from the center of the gravity of the car. The lateral tire forces are approximated as linear functions of the tire side-slip angles from the well-know empirical model “Magic formula” [22], i.e $F_{yr} = C_r \alpha_r$, $F_{yf} = C_f \alpha_f$. C_f , C_r are the front and rear cornering stiffness respectively.

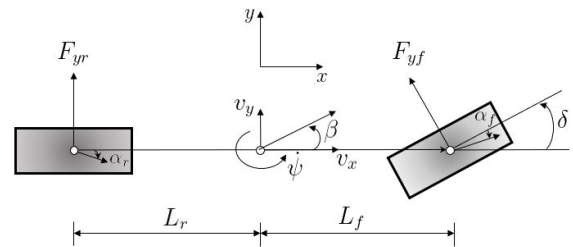


Fig. 1. Two-wheeled bicycle model representing the vehicle lateral dynamics.

$$\begin{aligned}\dot{v}_y &= -\frac{C_f + C_r}{mv_x}v_y + \left(-v_x + \frac{C_r L_r - C_f L_f}{mv_x}\right)\dot{\psi} + \frac{C_f}{m}\delta \\ \dot{\psi} &= \frac{-L_f C_f + L_r C_r}{I_z v_x}v_y - \frac{L_f^2 C_f + L_r^2 C_r}{I_z v_x}\dot{\psi} + \frac{L_f C_f}{I_z}\delta\end{aligned}\quad (1)$$

B. Look-ahead Errors Model

Fig. 2 illustrates the steering framework via the following of a path is achieved. More specifically, the NAV of the vehicle provides a set of control points (blue points in fig. 2) by utilizing polynomials and discretizing it into several points. Thus, at every instant, according to the vehicle's position and its heading the vehicle's pose is computed at the selected look-ahead distance L and it's projected onto the closest segment of the reference trajectory. The closest segment consists of two consecutive control points, and from that segment, the lateral and angular errors at the look-ahead distance are calculated, i.e. y_L , ε_L respectively, [23]. The errors are modeled as follows:

$$\begin{aligned}\dot{y}_L &= -v_y - L\dot{\psi} + v_x \varepsilon_L \\ \dot{\varepsilon}_L &= -\dot{\psi} + v_x \kappa\end{aligned}\quad (2)$$

where κ is the road curvature of the trajectory at the target point.

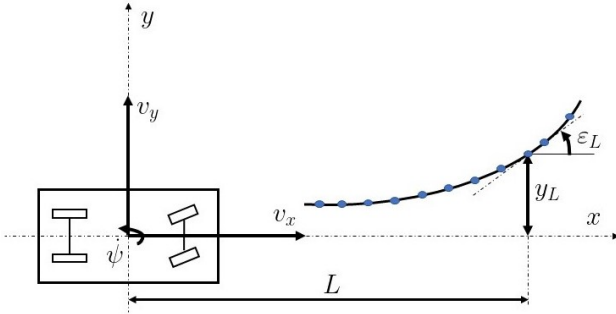


Fig. 2. Look-ahead errors according to the reference trajectory.

The merged model that contains the dynamics from eq. (1) & (2), can be expressed from the concatenated matrices below:

$$\begin{aligned}\dot{x}_v(t) &= A_v x_v(t) + B_{v1} r(t) + B_{v2} u(t) \\ y_v(t) &= C_v x_v(t)\end{aligned}\quad (3)$$

$$A_v = \begin{bmatrix} -\frac{C_f + C_r}{mv_x} & -v_x + \frac{C_r L_r - C_f L_f}{mv_x} & 0 & 0 \\ \frac{-L_f C_f + L_r C_r}{I_z v_x} & \frac{-L_f^2 C_f + L_r^2 C_r}{I_z v_x} & 0 & 0 \\ -1 & -L & 0 & v_x \\ 0 & -1 & 0 & 0 \end{bmatrix},$$

$$x_v(t) = \begin{bmatrix} v_y \\ \dot{\psi} \\ y_L \\ \varepsilon_L \end{bmatrix}, B_{v1} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, B_{v2} = \begin{bmatrix} \frac{C_f}{m} \\ \frac{L_f C_f}{I_z} \\ 0 \\ 0 \end{bmatrix},$$

$$C_v = [0 \ 0 \ 1 \ 0]$$

and $r(t) = \dot{\psi}_{traj} = v_x \kappa$ is an exogenous input.

C. Steering Actuation Model & Augmented Yaw Dynamics

The steering actuation system has been identified as a delayed second-order transfer function. This model is connected in series with the bicycle model so as to design a controller that respects the bandwidth of the actuator. The actuator transfer function is expressed below:

$$G_{act} = \frac{k}{s^2 + 2\zeta\omega_n s + \omega_n^2} e^{-T_d s} \quad (4)$$

where k , ζ , ω_n and T_d are the static gain, the damping, the natural frequency and the time-delay respectively.

A second-order Padé approximation of the time-delay T_d is applied, equation (4) can be re-written as a state space model:

$$\begin{aligned}\dot{x}_{act}(t) &= A_{act} x_{act}(t) + B_{act} \delta(t) \\ \delta_{act}(t) &= C_{act} x_{act}(t)\end{aligned}\quad (5)$$

where $x_{act} \in \mathbb{R}^4$ is the vector expressing the states of the actuator, $A_{act} \in \mathbb{R}^{4 \times 4}$, $B_{act} \in \mathbb{R}^{4 \times 1}$, $C_{act} \in \mathbb{R}^{1 \times 4}$ are the systems matrices and $\delta_{act} \in \mathbb{R}$ is the output.

Considering the output of the actuator as the input of the extended vehicle model (3), the state space of the interconnected system can be written as:

$$\begin{aligned}\dot{x}_v(t) &= A_v x_v(t) + B_{v2} C_{act} x_{act}(t) + B_{v1} r(t) \\ \dot{x}_{act}(t) &= A_{act} x_{act}(t) + B_{act} \delta(t)\end{aligned}\quad (6)$$

The above state space equations can be formulated as the augmented system shown below:

$$\begin{aligned}\dot{x}(t) &= Ax(t) + B_1 r(t) + B_2 \delta(t) \\ y(t) &= Cx(t)\end{aligned}\quad (7)$$

where, $x(t) = \begin{bmatrix} x_v(t) \\ x_{act}(t) \end{bmatrix} \in \mathbb{R}^8$ is the augmented state vector, $A = \begin{bmatrix} A_v & B_{v2} C_{act} \\ 1 - 20 & A_{act} \end{bmatrix} \in \mathbb{R}^{8 \times 8}$, $B_1 = \begin{bmatrix} B_{v1} \\ 0 \end{bmatrix} \in \mathbb{R}^8$, $B_2 = \begin{bmatrix} 0 \\ B_{act} \end{bmatrix} \in \mathbb{R}^8$ and $C = [C_v \ 0] \in \mathbb{R}^{1 \times 8}$ are the augmented system matrices.

III. LPV MODEL & 3D POLYTOPE REDUCTION

This section proposes a 3-dimensional polytope reduction for the augmented system eq. (7) and for the selected operating domain of varying parameters. Note that this is not a generic method able to handle a n -dimensional size polytope.

A. LPV Polytopic Model Construction

The above system representation (7), can be written in an LPV form, by considering the varying parameters in the plant matrices as bounded and real-time measured. In that sense, the system matrices are fixed functions of the parameter vector $\rho(t)$.

More specifically, it is assumed that the vector ρ varies in a polytope Θ of vertices θ_i .

$$\begin{aligned} \rho(t) &\in \Theta \\ \Theta &= \text{Co}\{\theta_1, \theta_2, \dots, \theta_N\} \end{aligned} \quad (8)$$

where Co denotes the convex hull of the finite N vertices θ_i .

The vertices θ_i correspond to the combinations of the extremum values of the parameters ρ_i , i.e. $\underline{\rho}_i \leq \rho_i \leq \bar{\rho}_i$. Respectively, the LPV Polytopic model is constructed by assuming that the LPV plant matrices, which contain the parameters ρ_i , vary in a matrix polytope. That polytope is defined as the convex hull of a number of vertex matrices of the same dimension. These vertex matrices are computed for values of parameters equal to the vertices θ_i . Another assumption, it is the affine dependence of the LPV matrices on the parameter vector ρ [13], i.e

$$A(\rho) = A_{\rho_0} + \sum_{i=1}^n \rho_i A_{\rho_i} \quad (9)$$

where A_{ρ_0}, A_{ρ_i} are LTI state matrices and n is the number of parameters.

The varying parameters chosen in this work, are the speed v_x and the look-ahead distance L . These parameters are explicitly included in the system (7) through the system matrix $A(v_x, L)$. Therefore, by defining the vector of varying parameters $\rho = (\rho_1, \rho_2) = (v_x, L)$ the state matrix can be decomposed as follows:

$$A(\rho) = \rho_1 A_{\rho_1} + \frac{1}{\rho_1} A_{\frac{1}{\rho_1}} + \rho_2 A_{\rho_2} \quad (10)$$

As seen, $A(\rho)$ is then not an affine function of the parameter vector, because the term $1/v_x = 1/\rho_1$ exists in the matrix $A_{\frac{1}{\rho_1}}$ and consequently is not LTI.

To meet this requirement the varying parameter vector ρ is augmented as, $\rho = (\rho_1, \rho_2, \rho_3) = (v_x, 1/v_x, L)$. Thus, the affine property is satisfied as it is shown below and at the right part of the eq. (11) all the matrices are LTI.

$$A(\rho) = \rho_1 A_{\rho_1} + \rho_2 A_{\rho_2} + \rho_3 A_{\rho_3} \quad (11)$$

In this work, the selected look-ahead distance profile per speed is the same as the one considered in [17] for lane-tracking. The range of speed is selected according to the pre-defined ODD of a project launched by Renault that aims at providing robotaxi services around the peri-urban and rural areas of Paris. More specifically, the operating domain of the varying parameters is the following:

$$\begin{aligned} v_x &\in [5, 25] \text{ (m/s)} \\ 1/v_x &\in [0.04, 0.2] \text{ (s/m)} \\ L(v_x) &= av_x e^{bv_x} + cv_x e^{dv_x} \text{ (m)} \end{aligned} \quad (12)$$

where $a = 3.83$, $b = -0.7261$, $c = 1.154$ and $d = -0.01453$.

Then, the LPV matrix $A(\rho)$ can be expressed, according to the vertex property [24], as a convex hull of the vertex matrices created by all the possible combinations of the parameter bounds, as shown below:

$$\begin{aligned} A(\rho) &= \sum_{i=1}^N a_{\theta_i}(t) A_i \\ \sum_{i=1}^N a_{\theta_i}(t) &= 1, \quad a_{\theta_i}(t) \geq 0 \end{aligned} \quad (13)$$

where $a_{\theta}^T(t) = (a_{\theta_1}(t), \dots, a_{\theta_N}(t))^T$, $N = 2^3$ are the scaling variables and the number of vertices accordingly. The variables $a_{\theta}(t)$ are computed according to the real-time position of $\rho(t)$ in the 3D polytope Θ and, subsequently w.r.t. the vertices $\theta_i \in \mathbb{R}^3$, formulated as such:

$$\rho(t) = \sum_{i=1}^N a_{\theta_i}(t) \theta_i \quad (14)$$

The resulting LPV Polytopic model, by replacing the LPV matrix from (13) to the augmented model (7) is the one below:

$$\begin{aligned} \dot{x}(t) &= A(\rho(t))x(t) + B_1 w(t) + B_2 u(t) \\ y(t) &= C(t)x(t) \end{aligned} \quad (15)$$

It has to be remarked that the look-ahead distance is a varying tuning parameter that is selected by the designer per speed [25]. Subsequently, it is clear that the parameter L also depends from v_x . Thus, it seems of high interest to reduce the size of the polytope Θ . The shrinking of the polytope's size decreases the complexity, the conservatism of the approach and it facilitates the implementation of the synthesized controller, as it is detailed in [19] and it has already applied successfully in [18], [26].

B. Reduction of the 3D Polytope

The polytope that describes the LPV system is first created for the upper and lower bounds of the operating domain of parameters (12) and is illustrated in Fig. 3. Due to the

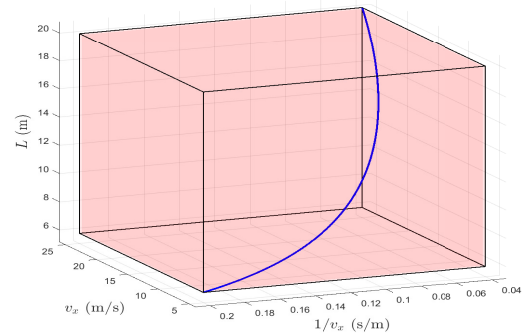


Fig. 3. The initial over-bounded 8-vertices polytope.

facts that (i) the parameter vector ρ is further increased to satisfy the affine-dependence and, (ii) all the parameters ρ_i are all dependent to each other, the inherent conservatism of

the Polytopic approach is enlarged. To treat this limitation of the method and avoid other engineering-based solutions as selecting the look-ahead distance or the time constant, as in [18], [20] accordingly, it is proposed a 3-dimensional polytopic reduction.

It has to be remarked that the reduced parameter polytope has to include the parameter variation as the initial-full 8 vertices (see Fig. 3). This statement arises to retain the convexity property, as it is defined for the Polytopic Linear Differential Inclusions [27].

For the above reasons, the proposed Alg. 1 is proposed. As a first step, the 2-dimensional pairs of parameter variations are illustrated in figure 4(a) and in the second step of figure 4(b), a new set of vertices is selected for which the resulted 3D polytope 4(c) is convex and compact. In step 3 & Fig.4(d), the remaining non-realistic vertices that doesn't affect the inclusion of the 3D parameter trajectory are deleted, i.e θ_4 . In the final step of the algorithm, no-more vertices are deleted, but the coordinates of the remaining vertices are reduced to shrink even more the polytope (see figure 4(e)). In that particular case, the vertical coordinate of vertex θ_2 , $L_{\theta_2} = 20 m$, is reduced giving θ_2^* . Moreover, the vertical coordinate of θ_2^* corresponds to a smaller value of look-ahead distance, $L_{\theta_2^*} = 13 m$, hence permitting the increased bandwidth around that region of parameters for the closed-loop system. It has to be emphasized that θ_3 and θ_5 describe the two realistic cases, and the position of θ_1 is not modified since the inclusion of the parameters trajectory will be affected. Finally, the remaining vertices of the reduced polytope $\tilde{\Theta}$ are the $\theta^* = (\theta_1, \theta_2^*, \theta_3, \theta_5)$.

Algorithm 1: 3D Polytope Reduction Procedure

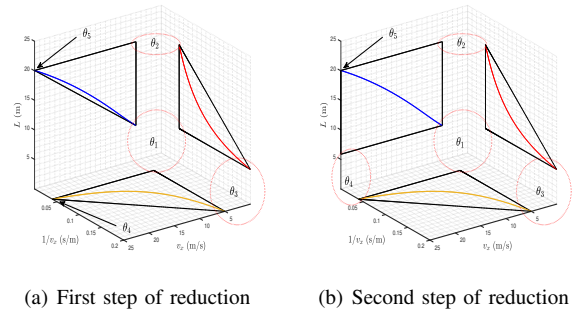
Data: $\Theta = Co\{\theta_1, \theta_2, \dots, \theta_8\}$, Fig. 3.

Result: $\tilde{\Theta} = Co\{\theta_1, \theta_2^*, \theta_3, \theta_5\}$.

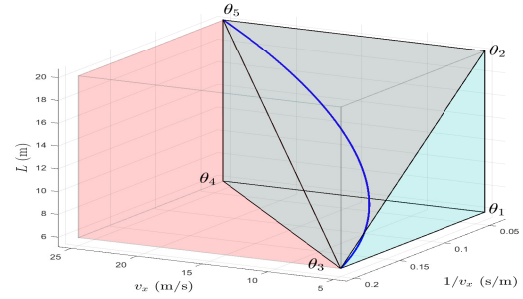
- 1 **Step 1:** Plot the 2D trajectories of the pairs of parameters. Keep the vertices of the triangles that include the 2D parameter variations (Fig. 4(a));
 - 2 **Step 2:** Extend the necessary triangles (Fig. 4(b)) whose vertices are able to merge to a compact 3D convex set (Fig. 4(c));
 - 3 **Step 3:** Reduce the vertices (i.e θ_4 in Fig. 4(d)), for which the 3D parameters trajectory is still included;
 - 4 **Step 4:** Shrink the polytope by reducing the vertical coordinate of θ_2 to θ_2^* , till the final geometry includes the parameters trajectory (Fig. 4(e));
-

It is worth mentioning that the proposed algorithm is not valid for every 3-dimensional polytope but for the specific system, as it is shown in figures 3 and 4.

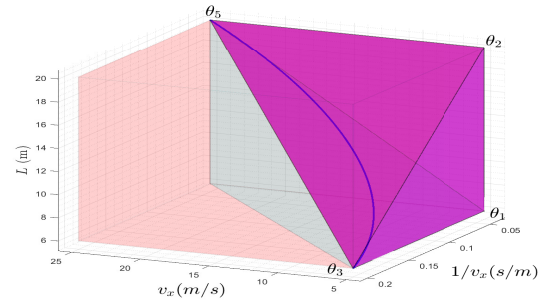
It can also be noted that for the case where the operating domain of parameters changes e.g. the formula of the look-ahead distance, then the 3-dimensional polytope will slightly change as well and a new reduction algorithm should be applied to secure the stability properties of the reduced polytopic system. This is a limitation of the proposed algorithm,



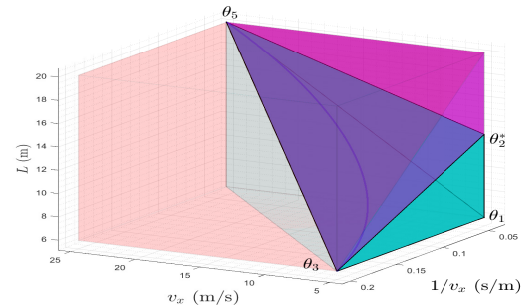
(a) First step of reduction (b) Second step of reduction



(c) Illustration of the 3D polytope after the second step of the algorithm.



(d) Illustration of the 3D polytope after the third step of the algorithm.



(e) Illustration of the 3D polytope after the fourth step of the algorithm.

Fig. 4. Overall comparison from the initial 8-vertices polytope (pink) to the reduced 4-vertices (green).

which is the price to pay for the complexity reduction.

IV. LPV POLYTOPIC CONTROLLER SYNTHESIS

In the H_∞ framework, a LPV Polytopic Controller can be designed off-line, as a solution of a convex optimization. The semi-definite problem subject to a set of LMI, is formulated at every vertex of the reduced polytope (Fig. 4(e)) [24].

A. Tuning & Controller Synthesis

The tuning of the LPV controller is achieved by introducing performance/template weighting functions W_y , $W_u(s)$, W_r & W_n [28] according to the control configuration scheme (Fig. 5). These functions are selected as such:

- The first-order filter $W_u(s)$ is chosen as $W_u(s) = \frac{s + w_{bc}/M}{\varepsilon s + w_{bc}}$ with $w_{bc} = 1 \text{ rad/s}$ to respect the corresponding control objective about the bandwidth of the controller, $M = 2 \text{ (6db)}$ to respect the saturation limits and $\varepsilon = 0.1 \text{ (} 10^{-1} \text{ rad/s)}$ that expresses the frequency where the roll-off starts to achieve better noise attenuation.
- The weighting function W_y is chosen as $W_y = 0.5$. It aims at imposing weight on the lateral error at the target point, as a way to achieve the comfortable tracking of the reference trajectory.
- Finally, constant scaling weights are added on the noise signal and the reference, respectively as: $W_n = 0.5$ and $W_r = 0.3$.

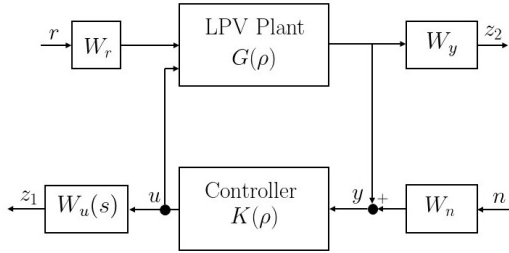


Fig. 5. Path-following feedback control configuration.

The generalized plant (16) that describes the control configuration of the weighted control scheme (depicted in Fig. 5), is expressed below.

$$\begin{bmatrix} \dot{x}_g(t) \\ y(t) \\ z(t) \end{bmatrix} = \begin{bmatrix} \mathcal{A}(\rho) & \mathcal{B}_1 & \mathcal{B}_2 \\ \mathcal{C} & 0 & 0 \\ \mathcal{C}_z & \mathcal{D}_{z_1} & \mathcal{D}_{z_2} \end{bmatrix} \begin{bmatrix} x_g(t) \\ w(t) \\ u(t) \end{bmatrix} \quad (16)$$

where x_g is the augmented state vector that consists of the states of the extended model (7) and the states x_z , which express the dynamics of the weighting functions i.e $x_g(t) = \begin{bmatrix} x(t) \\ x_z(t) \end{bmatrix} \in \mathbb{R}^9$, $z(t) = \begin{bmatrix} z_1(t) \\ z_2(t) \end{bmatrix} \in \mathbb{R}^2$ and $w(t) = \begin{bmatrix} r(t) \\ n(t) \end{bmatrix} \in \mathbb{R}^2$. The generalized plant in (16) is also in polytopic form consisted of the vertices θ^* , of the reduced polytope.

Problem Definition: The LPV control synthesis problem consists in finding an LPV controller $K(\rho)$ so that the closed-loop system represented in Fig. 5 is stable and there exists

a $\gamma > 0$ s.t.

$$\sup_{\|w\| \neq 0} \frac{\|z\|_2}{\|w\|_2} < \gamma, \forall \theta^* \quad (17)$$

B. Solution & Frequency response of the LPV Controller

The polytopic approach is here considered to design the vertex controllers K_i , $i = 1, 2, 3, 4$, by solving off-line the appropriate set of LMIs (for more details see [24] or [29]). The solution of the synthesis of the LPV/Polytopic controller is computed with an attenuation level $\gamma = 2.6$. The solution is the set of the vertex LTI controllers in a state space form:

$$\begin{aligned} \dot{x}_c(t) &= A_i x_c(t) + B_i y(t) \\ u(t) &= C_i x_c(t) + D_i y(t) \end{aligned}, i = 1, 2, 3, 4 \quad (18)$$

where $x_c(t) \in \mathbb{R}^9$ denotes the state space vector of the controller.

Fig. 6(a) depicts the controller sensitivity function u/r over the closed-loop system for every vertex controller. As it is visualized, the desired performance imposed by the template weighting function $W_u(s)$ is satisfied over the frequency domain for every vertex θ_i^* . Whereas, in figure 6(b), it is depicted the closed-loop behavior for the lateral error at the target point to the yaw-rate reference. Fig. 6(c) illustrates the bode plot for the vertex LTI continuous controllers, proving that the smaller magnitudes of vertex controllers is provided by the two non-realistic vertices θ_1 and θ_2^* .

V. REAL-TIME IMPLEMENTATION OF THE LPV POLYTOPIC CONTROLLER

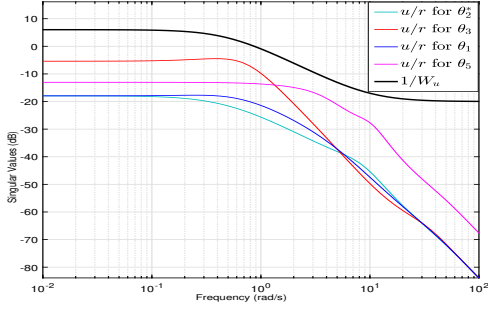
A dynamic output feedback LPV/Polytopic controller is a convex combination of vertex LTI controllers computed in real-time from the scaling variables a_{θ_i} . Different methods can be to compute these interpolation variables. In [30] binary representations are utilized for the case where the polytope Θ consists of a box, cube or hyper cube when the number of parameters is $n = 2, 3$ or > 3 respectively. That approach is accurate for the cases where no reduction is applied to the parameters polytope. The formula that computes the interpolation variables for the i_{th} vertex controller are as follows:

$$a_{\theta_i}(k) = \prod_{j=1}^N \frac{\alpha_{ij} \rho_j(k) + \beta_{ij}}{\bar{\rho}_j - \underline{\rho}_j} \quad (19)$$

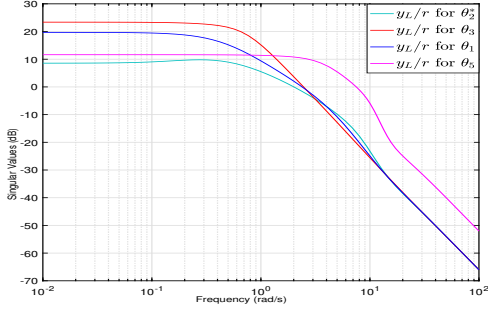
where $\rho_j(k)$ is the real-time measurement of the j parameter and $\bar{\rho}_j$, $\underline{\rho}_j$ are the upper and lower bounds of that vertex respectively. α_{ij} and β_{ij} provide automatically the appropriate sign according to the binary representation that is given to the j vertex.

For a more general case where the polytope is reduced another way can be applied as in the proposition 3.1 of [31]. The interpolation variables are computed as the solution of a linear system that is augmented with the equality $\sum_{i=1}^N a_{\theta_i} = 1$.

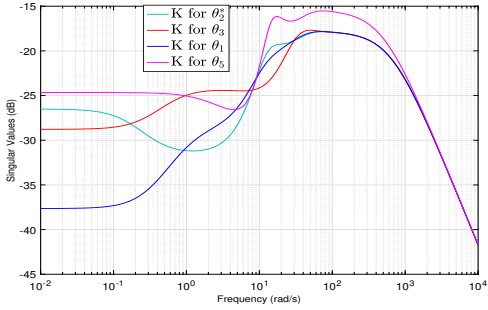
$$\begin{bmatrix} \theta_1 & \theta_2 & \dots & \theta_N \\ 1 & 1 & \dots & 1 \end{bmatrix} a_{\theta} = \begin{bmatrix} \rho(k) \\ 1 \end{bmatrix} \quad (20)$$



(a) Controller sensitivity function of the vertex closed-loop system relatively to the weighting function $1/W_u$.



(b) Frequency response of the y_L/r of the closed-loop vertex systems.



(c) Frequency response of the LTI vertex controllers K_i .

Fig. 6. Bode Plots of the reduced 4-vertices polytopic closed-loop system.

However, that approach may cause numerical problems especially for the cases where the polytope is tightened and since the inequality $a_{\theta_i} \geq 0$ is absent. Subsequently, a novel on-line implementation procedure is proposed for the LPV/Polytopic controller $K(\rho)$.

A. Off-line Step

First, the discrete-time state space vertex controllers, K_{d_i} are computed as indicated for LTI systems in [32], for a

given sampling time T_s :

$$K_{d_i} = \begin{bmatrix} A_{d_i} & B_{d_i} \\ C_{d_i} & D_{d_i} \end{bmatrix}$$

$$x_c(k+1) = A_{d_i}x_c(k) + B_{d_i}y(k)$$

$$u(k) = C_{d_i}x_c(k) + D_{d_i}y(k) \quad (21)$$

$$\begin{bmatrix} A_{d_i} & B_{d_i} \\ 0 & 0 \end{bmatrix} = \exp\left(\begin{bmatrix} A_i & B_i \\ 0 & I \end{bmatrix} T_s\right)$$

$$C_{d_i} = C_i, D_{d_i} = D_i$$

B. On-line Step

The discretized LPV controller is implemented as a convex combination of the vertex controllers K_{d_i} i.e

$$K_d(\rho) = \sum_{i=1}^4 a_{\theta_i}(k) K_{d_i} \quad (22)$$

where the constants $a_{\theta_i}(k)$ are the scaling variables, as they specify the contribution of each vertex controller, according to the real-time measurement of the vector $\rho(k)$ [33].

In this work, the proposition 3.1 from [31] is extended. Indeed the novel computation of $a_{\theta}(k)$, is formulated as a real-time constrained LS problem, as presented below:

$$\begin{aligned} & \text{minimize} && \|\mathcal{M}a_{\theta}(k) - \rho(k)\|_2^2 \\ & \text{subject to} && a_{\theta_i}(k) \geq 0 \\ & && \sum_{i=1}^4 a_{\theta_i}(k) = 1 \end{aligned} \quad (23)$$

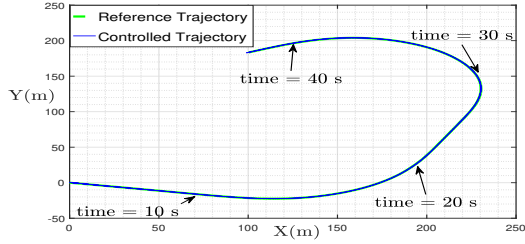
where \mathcal{M} contains the coordinate information of the vertices i.e $\mathcal{M} = [\theta_1, \theta_2^*, \theta_3, \theta_5] \in \mathbb{R}^{3 \times 4}$ and $a_{\theta}^T(k) = (a_{\theta_1}(k), a_{\theta_2^*}(k), a_{\theta_3}(k), a_{\theta_5}(k))^T \in \mathbb{R}^4$.

VI. SIMULATION RESULTS

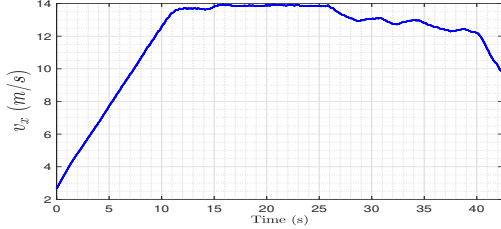
This section presents the simulated results (Fig. 7). The synthesized LPV controller, and consequently the vertex matrices, has been discretized for a sampling time $T_s = 0.01$ s at which the steering actuation system of a Renault Zoe model accepts steering command. The real-time implementation of the constrained LS problem that computes the variables a_{θ_i} is achieved by the custom-generated solver CVXGEN [34].

As a case scenario is utilized a segment of a test track (Fig. 7(a)) that contains two turns, for a varying longitudinal speed profile depicted in figure 7(b). The dataset that contains the information of the trajectory has been collected experimentally from the work of [17].

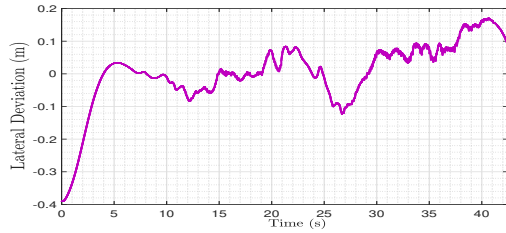
The vehicle at the beginning started with an initial lateral offset at the center of gravity, i.e y_{cg} , of 0.4 m (figure 7(c)) and for that reason the steering, in figure 7(d) was abrupt at the beginning. Apart from that, the steering is smooth enough during the turns while the control system doesn't allow more than 0.2 m of lateral deviations. It has to be remarked that the LS solution, i.e $a_{\theta_i}(k)$, didn't fail when the velocity of the car at the beginning started outside of the polytope's minimum velocity value. On the contrary, the optimization successfully computed that the closest vertex is θ_3 . As the speed increases, the contribution of each controller varies, as it expected.



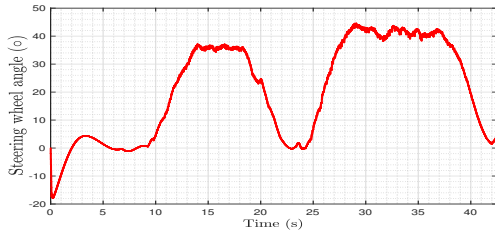
(a) Trajectory followed during the simulated test.



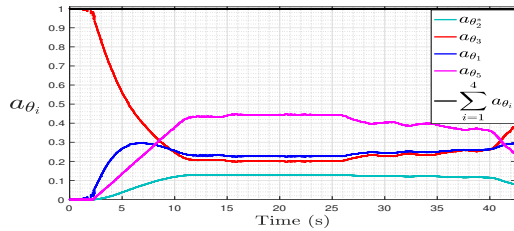
(b) Longitudinal speed profile fed by the NAV to the lateral control.



(c) Lateral deviation at the center of gravity y_{cg} , computed throughout the simulation.



(d) Steering wheel angle command of the reduced LPV/Polytopic controller.



(e) Scaling variables a_{θ_i} computed as an output of the optimization problem (23), w.r.t. the position of the vector $\rho(k)$ to the 3D reduced polytope.

Fig. 7. Simulation results.

VII. EXPERIMENTAL RESULTS

In a private test track at Satory of France, the proposed control architecture has been validated in an automated elec-

tric Renault Zoe (Fig. 8). The control has been implemented in a MicroAutobox that has been installed at the trunk of the car. The real-time deployment is achieved by using dSPACE to upload the built code. The vehicle is equipped with a Real-Time Kinematic Differential Global Positional System (RTK-DGPS) that provides the vehicle's pose at the global frame. Consequently, it is feasible to calculate at every instant the lateral error at the target point, according to the speed provided by the Navigation, and eventually feed the control system and the real-time optimization problem. The LTI vertex controllers are discretized at 0.01 s and thus, the LPV controller provides the steering command that is computed in rad and is converted in degrees that leads as input to the Electric Power Steering (EPS) system. In the two following subsections, two different segments of the test track are utilized for low and high speeds.

A. Path-Tracking at Low Speed Curves

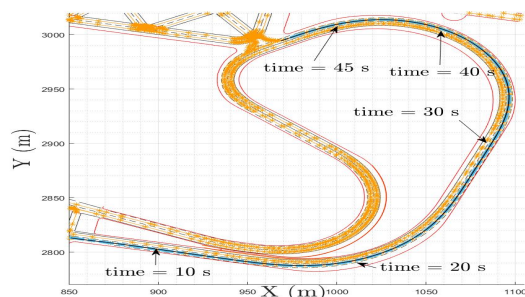
At first, the vehicle is tested under two turns as where the measured experimental result are depicted in figure 9. The vehicle starts in autonomous mode at 8 m/s and an initial lateral error of 0.4 m and for that reason an initial abrupt steering is applied. Then, the vehicle decelerates to 2 m/s and then accelerates again to 13 m/s to enter the first turn and keeps a velocity around $12\text{--}14\text{ m/s}$ (Fig. 9(e)) till then end of the second turn. The plot of the lateral offset of the car (Fig. 9(c)) proves that the proposed controller is able to sustain a good tracking performance since the maximum lateral offset never gets more than 0.2 m . The applied steering command, as it is depicted in figure 9(d), shows that the steering is smooth and comfortable enough throughout the ride. Figure 9(e) show that, even though the measured velocity is outside of the parameter variation (when it gets to $2 < 5\text{ m/s}$, the proposed optimization achieves to compute the closest vertex controller, i.e θ_3 , to treat that case and that is real-time implementable since it respects the computational power for the utilized sampling time of 0.01 s .



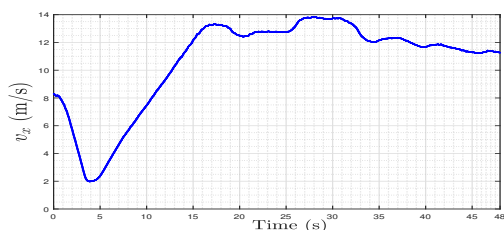
Fig. 8. Autonomous Electric Renault Zoe

B. Higher Velocity Curves

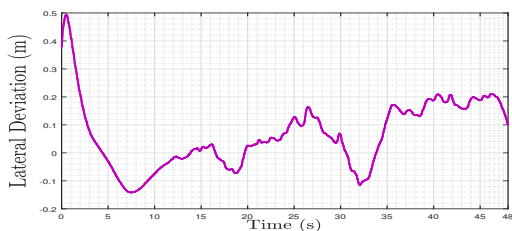
In this experimental part (set of figures 10), the autonomous mode is activated at 12 m/s and accelerates at a maximum value of 18 m/s to track two smooth curves. As



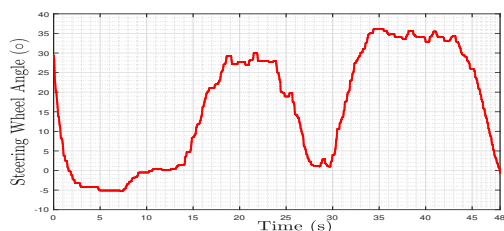
(a) Trajectory followed at the test track.



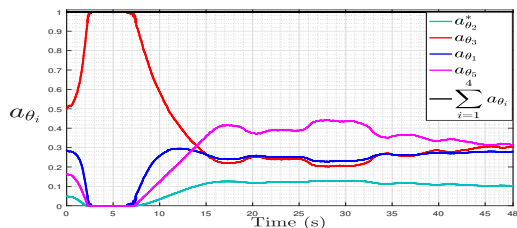
(b) Longitudinal speed profile fed by the NAV to the lateral control.



(c) Lateral deviation at the center of gravity y_{cg} .



(d) Applied steering wheel angle command.



(e) Real-time solution of the LS (23), a_{θ_i} , w.r.t. the position of the vector $\rho(k)$ to the 3D reduced polytope.

Fig. 9. *First case scenario a)*: Experimental results at the test track of Satory.

it can be seen, even though the lateral offset of the vehicle is high at the beginning (Fig. 10(c)), the car is able to apply a smooth steering command and track the curves sufficiently

comfortable without permitting more than 0.2 m of lateral error at the center of gravity. The steering, as it is depicted in the subfigure 10(d) is not sensitive to noise at higher speeds and even though it is increased at 20° during the second curve it was increased in a slow pace. Once again, the solutions of the optimization problem, i.e a_{θ_i} are provided in the subfig. 10(e), where it is illustrated that the scaling variable a_{θ_5} is increased since the closest vertex is θ_5 .

VIII. CONCLUSION

This paper presents the reduction, design and the implementation of a 3D polytope-based LPV controller. An algorithm is proposed to reduce the number of vertices of the initial over-bounded polytope and its real-time implementation is achieved as the solution of a constrained LS problem. The validity and the performance of the proposed path-following control system is assessed and proven at both simulation and experimental results for low and high speeds for an automated electric Renault Zoe.

Future works will include the comparison of the performance for different reductions of the 3D polytope. An interesting perspective could be the merging of the auto-tuning of weighted LS [35] for the implementation of the LPV controller. In that sense, different performance objectives about longitudinal or lateral performance could be taken into account in the optimization or as constraints.

DECLARATION OF COMPETING INTEREST

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

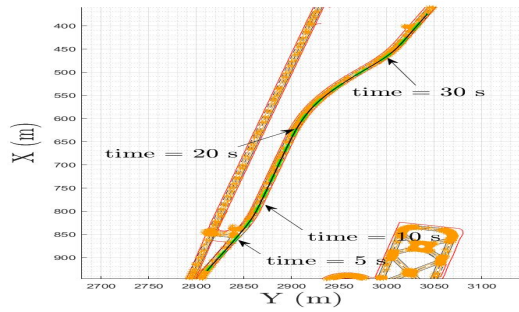
ACKNOWLEDGMENTS

This paper reflects solely the views of the authors and not necessarily the view of the company they belong to.

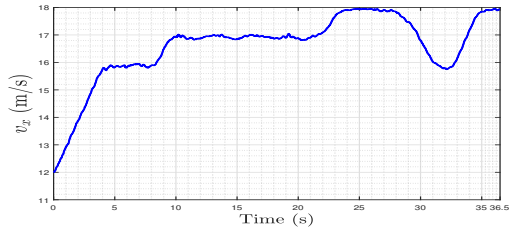
Authors would like to thank Renault R&D department for the support during the conduction of this work and Alexandre Armand for his help during testing in the track facilities.

REFERENCES

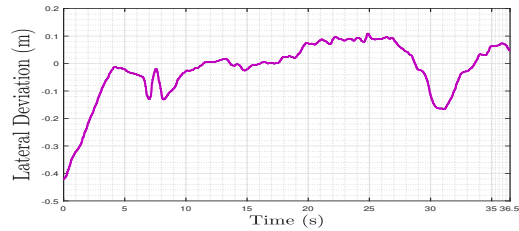
- [1] N. H. T. S. Administration *et al.*, “Automated driving systems 2.0: A vision for safety,” *Washington, DC: US Department of Transportation, DOT HS*, vol. 812, p. 442, 2017.
- [2] D. González, J. Pérez, V. Milanés, and F. Nashashibi, “A review of motion planning techniques for automated vehicles,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 4, pp. 1135–1145, 2015.
- [3] C. Berger and B. Rumpe, “Autonomous driving-5 years after the urban challenge: The anticipatory vehicle as a cyber-physical system,” *arXiv preprint arXiv:1409.0413*, 2014.
- [4] J. Ziegler, P. Bender, M. Schreiber, H. Lategahn, T. Strauss, C. Stiller, T. Dang, U. Franke, N. Appenrodt, C. G. Keller, *et al.*, “Making bertha drive—an autonomous journey on a historic route,” *IEEE Intelligent transportation systems magazine*, vol. 6, no. 2, pp. 8–20, 2014.
- [5] A. Broggi, P. Cerri, S. Debattisti, M. C. Laghi, P. Medici, D. Molinari, M. Panciroli, and A. Prioletti, “Proud—public road urban driverless-car test,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 6, pp. 3508–3519, 2015.
- [6] S. E. Shladover, “Review of the state of development of advanced vehicle control systems (avcs),” *Vehicle System Dynamics*, vol. 24, no. 6-7, pp. 551–595, 1995.



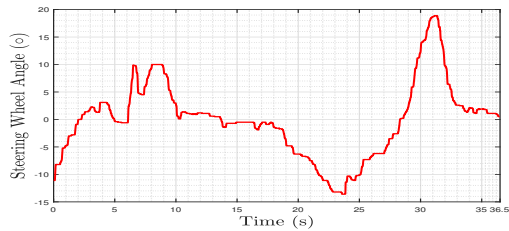
(a) Trajectory followed.



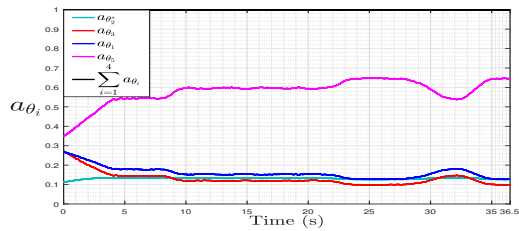
(b) Longitudinal speed profile fed by the NAV to the lateral control.



(c) Lateral deviation at the center of gravity y_{cg} .



(d) Applied steering wheel angle command.



(e) Real-time solution of the LS (23), a_{θ_i} , w.r.t. the position of the vector $\rho(k)$ to the 3D reduced polytope.

Fig. 10. Second case scenario b): Experimental results at the test track of Satory.

[7] B. Paden, M. Čáp, S. Z. Yong, D. Yershov, and E. Frazzoli, “A survey of motion planning and control techniques for self-driving urban vehicles,” *IEEE Transactions on intelligent vehicles*, vol. 1, no. 1, pp. 33–55, 2016.

[8] P. Falcone, F. Borrelli, J. Asgari, H. E. Tseng, and D. Hrovat, “Predictive active steering control for autonomous vehicle systems,” *IEEE Transactions on control systems technology*, vol. 15, no. 3, pp. 566–580, 2007.

[9] S. Xu, H. Peng, P. Lu, M. Zhu, and Y. Tang, “Design and experiments of safeguard protected preview lane keeping control for autonomous vehicles,” *IEEE Access*, vol. 8, pp. 29944–29953, 2020.

[10] J.-Y. Wang and M. Tomizuka, “Robust h_∞ lateral control of heavy-duty vehicles in automated highway system,” in *Proceedings of the 1999 American Control Conference (Cat. No. 99CH36251)*, vol. 5. IEEE, 1999, pp. 3671–3675.

[11] M. S. Netto, S. Chaib, and S. Mammari, “Lateral adaptive control for vehicle lane keeping,” in *Proceedings of the 2004 American Control Conference*, vol. 3. IEEE, 2004, pp. 2693–2698.

[12] Z. Ercan, M. Gokasan, and F. Borrelli, “An adaptive and predictive controller design for lateral control of an autonomous vehicle,” in *2017 IEEE international conference on vehicular electronics and safety (ICVES)*. IEEE, 2017, pp. 13–18.

[13] J. Mohammadpour and C. W. Scherer, *Control of linear parameter varying systems with applications*. Springer Science & Business Media, 2012.

[14] C. Hoffmann and H. Werner, “A survey of linear parameter-varying control applications validated by experiments or high-fidelity simulations,” *IEEE Transactions on Control Systems Technology*, vol. 23, no. 2, pp. 416–433, 2014.

[15] O. Sename, P. Gaspar, and J. Bokor, *Robust control and linear parameter varying approaches: application to vehicle dynamics*. Springer, 2013, vol. 437.

[16] P. Hingwe, H.-S. Tan, A. K. Packard, and M. Tomizuka, “Linear parameter varying controller for automated lane guidance: experimental study on tractor-trailers,” *IEEE Transactions on control systems technology*, vol. 10, no. 6, pp. 793–806, 2002.

[17] D. Kapsalis, O. Sename, V. Milanés, and J. J. Martinez, “Gain-scheduled steering control for autonomous vehicles,” *IET Control Theory & Applications*, vol. 14, no. 20, pp. 3451–3460, 2020.

[18] M. Corno, G. Panzani, F. Roselli, M. Giorelli, D. Azzolini, and S. M. Savaresi, “An lpv approach to autonomous vehicle path tracking in the presence of steering actuation nonlinearities,” *IEEE Transactions on Control Systems Technology*, vol. 29, no. 4, pp. 1766–1774, 2021.

[19] P. Li, A.-T. Nguyen, H. Du, Y. Wang, and H. Zhang, “Polytopic lpv approaches for intelligent automotive systems: State of the art and future challenges,” *Mechanical Systems and Signal Processing*, vol. 161, p. 107931, 2021.

[20] D. Kapsalis, O. Sename, V. Milanés, and J. J. M. Molina, “Design and experimental validation of an lpv pure pursuit automatic steering controller,” in *16th IFAC Symposium on Control in Transportation Systems*, 2021.

[21] R. Rajamani, *Vehicle dynamics and control*. Springer Science & Business Media, 2011.

[22] H. B. Pacejka and E. Bakker, “The magic formula tyre model,” *Vehicle system dynamics*, vol. 21, no. S1, pp. 1–18, 1992.

[23] C. J. Taylor, J. Koščeká, R. Blasi, and J. Malik, “A comparative study of vision-based lateral control strategies for autonomous highway driving,” *Int. J. Rob. Res.*, vol. 18, no. 5, pp. 442–453, 1999.

[24] P. Apkarian, P. Gahinet, and G. Becker, “Self-scheduled h_∞ control of linear parameter-varying systems: a design example,” *Automatica*, vol. 31, no. 9, pp. 1251–1261, 1995.

[25] J. Kosecka, “Vision-based lateral control of vehicles: Look-ahead and delay issues,” 1996.

[26] D. Robert, O. Sename, and D. Simon, “An h_∞ lpv design for sampling varying controllers: Experimentation with a t-inverted pendulum,” *IEEE Transactions on Control Systems Technology*, vol. 18, no. 3, pp. 741–749, 2009.

[27] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear matrix inequalities in system and control theory*. SIAM, 1994.

[28] S. Skogestad, *Multivariable feedback control: analysis and design*. Citeseer, 2007, vol. 2.

[29] C. Poussot-Vassal, O. Sename, L. Dugard, P. Gaspar, Z. Szabo, and J. Bokor, “Attitude and handling improvements through gain-scheduled suspensions and brakes control,” *Control Engineering Practice*, vol. 19, no. 3, pp. 252–263, 2011.

[30] G. I. Bara, J. Daafouz, F. Kratz, and J. Ragot, “Parameter-dependent state observer design for affine lpv systems,” *International journal of control*, vol. 74, no. 16, pp. 1601–1611, 2001.

- [31] J. Martinez, N. Loukkas, and N. Meslem, "H-infinity set-membership observer design for discrete-time lpv systems," *International journal of control*, vol. 93, no. 10, pp. 2314–2325, 2020.
- [32] K. J. Astrom and B. Wittenmark, *Computer controlled systems: theory and design*. Prentice Hall Professional Technical Reference, 1984.
- [33] C. Poussot-Vassal, "Commande robuste lpv multivariable de chassis automobile," Ph.D. dissertation, Grenoble INPG, 2008.
- [34] J. Mattingley and S. Boyd, "Cvxgen: A code generator for embedded convex optimization," *Optimization and Engineering*, vol. 13, no. 1, pp. 1–27, 2012.
- [35] S. T. Barratt and S. P. Boyd, "Least squares auto-tuning," *Engineering Optimization*, vol. 53, no. 5, pp. 789–810, 2021.