

LPV-MP planning for autonomous racing vehicles considering obstacles

Eugenio Alcalá*, Vicenç Puig, Joseba Quevedo

Research Center of Supervision, Safety and Automatic Control (CS2AC), Advanced Control Systems Group, Universitat Politècnica de Catalunya (UPC), Rambla Sant Nebridi, 10, 08222 Terrassa, Spain

ARTICLE INFO

Article history:

Received 15 June 2019

Received in revised form 25 November 2019

Accepted 26 November 2019

Available online 4 December 2019

Keywords:

Autonomous driving

Racing planning

MPC

LPV

Obstacle avoidance

ABSTRACT

In this paper, we present an effective online planning solution for autonomous vehicles that aims at improving the computational load while preserving high levels of performance in racing scenarios. The method follows the structure of the model predictive (MP) optimal strategy where the main objective is to maximize the velocity while smoothing the dynamic behavior and fulfilling varying constraints. We focus on reformulating the non-linear original problem into a pseudo-linear problem by convexifying the objective function and reformulating the non-linear vehicle equations to be expressed in a Linear Parameter Varying (LPV) form. In addition, the ability of avoiding obstacles is introduced in a simple way and with reduced computational cost.

We test and compare the performance of the proposed strategy against its non-linear approach through simulations. We focus on testing the performance of the trajectory planning approach in a racing scenario. First, the case of free obstacles track and afterwards a scenario including static obstacles. Simulation results show the effectiveness of the proposed strategy by reducing the algorithm elapsed time while finding appropriate trajectories under several input/state constraints.

1. Introduction

Autonomous vehicle racing is a variant of the field of autonomous driving that is attracting many researchers in recent years given the challenge that raises [1,2]. Such a problem involves a complex interaction with the environment due to the fast vehicle dynamic variation, i.e. high linear and angular accelerations, what implies directly a short reaction time in certain situations. The main brain of an autonomous vehicle has to carry out several tasks in a short period of time: detection and understanding of the elements in the current scene, corresponding treatment of the data of the sensors mounted on the vehicle, planning of the route depending on the current state and scene and finally the control performance over the actuation systems of the vehicle (steering, braking and acceleration systems).

Trajectory planning is a key point for autonomous guidance through known or unexplored environments. During these last years an important number of contributions can be found where a diversity of planning problems for nonholonomic vehicles are studied and solved due to an increasing interest in autonomous driving. There exist different classifications depending on the strategy used for solving the motion planning problem for autonomous road vehicles. The majority of the existing works may

be broadly classified into four main categories: geometry-based techniques, polynomial-based methods, sampling-based strategies and optimal-based techniques [3–7]. For an interesting survey on motion planning, see [8]. However, although all of them are promising techniques and have recently achieved interesting results in the autonomous driving field, this work focuses particularly on optimal-based strategies.

A classification within the techniques based on optimization can be made. This sorting depends mainly on two variables: time-dependent trajectory and type of vehicle modeling. In this work, we refer to trajectory planning when there exist a temporary dependent position evolution, i.e. the planning strategy computes temporal-based functions (positions and velocities). Otherwise, we will refer to path planning when the route planned is not time dependent, i.e. only lateral position and orientation are considered.

With respect to vehicle modeling, it refers to the wide range of mathematical or data-based models that can be used based on their degree of dynamic representation. Enumerating some of them from lower to greater dynamic complexity description we have the mass point model, the bicycle kinematic model, the bicycle dynamic model with simplified or complex tire dynamics, and the four wheels dynamic model with highly representative tire dynamics that considers roll, pitch and vertical motions.

In [9], the author perform a comparison between the bicycle and the four wheels model showing a hardly any difference

* Corresponding author.

E-mail address: eugenio.alcala@upc.edu (E. Alcalá).

Table 1

Optimal planning works depending on the type of vehicle prediction model and the scenario. *TP* and *PP* refer to trajectory and path planning, respectively.

Scenes\Models	L. Kin.	NL. Kin
Highways		
Highways & obs	Plessen et al. [10] _{TP}	Liu et al. [11] _{PP}
City & obs	Ahmadi Mousavi et al. [12] _{PP}	
Racing	Alrifaae and Maczajewski [13] _{TP}	
	Caporale et al. [14] _{TP}	Verschuere et al. [15] _{TP}
Racing & obs	Plessen et al. [10] _{TP}	
Scenes\Models	L. Dyn.	NL. Dyn.
Highways		
Highways & obs		Hegedus et al. [16] _{TP}
City & obs		
Racing	Liniger et al. [17] _{PP}	Verschuere et al. [18] _{TP}
Racing & obs	Liniger et al. [17] _{PP}	

between both and he states that the main difference is the longitudinal load change which may be also negligible due to the low center of gravity of the vehicle. Furthermore, if the model is going to be used in an optimization process then, a new classification criterion appears according to whether the vehicle model is linear or non-linear.

Particularly, in this work we will focus on the study of trajectory planning for racing using optimization strategies. The way of guiding a vehicle as quickly as possible requires a planning algorithm able to get certain road information (e.g. track curvature in the next meters) from the environment perception layer. The resultant trajectory must not only remain inside the track limits but also fulfill the dynamic constraints such as velocity, acceleration and slip angle limits. While respecting these constraints, the trajectory should also minimize the time to reach the end of the track as well as the difference between front and rear slip angles to avoid understeering and oversteering situations. Additionally, the trajectory should be updated online as the vehicle progresses along the track with the aim of avoiding unexpected static and/or dynamic obstacles.

In [11], the authors propose a path planning strategy based on non-linear Model Predictive Control (MPC) for the case of driving in highways. A unicycle kinematic model is used for predicting future vehicle states and the approach ensures safety by being able to generate free collision trajectories.

In [10], a trajectory planning strategy is presented using linear programming tools. This approach combines kinematic-based modeling with obstacle avoidance to provide a solution to highway planning problem.

In a similar way, a trajectory planning strategy based on solving a non-linear optimization is presented in [16], where a lane change methodology to be employed in highways is proposed. In the literature, we can also find some recent works where the problem of planning in cities or places with a complex environment, i.e. moving obstacles, is solved. This scenario is one of the most challenging because the ability to deal with a variety of mobile obstacles is one of the main skills included in the planner.

In [12], a path planning method is introduced which is based on solving a Linear Time Varying (LTV) MPC using a kinematic formulation of the vehicle. Besides, this method is able to deal with multiple obstacles by means of a convex formulation.

But, nevertheless, one of the most ambitious challenges in this field is on-line racing planning. The resolution of this problem is not trivial at all and there are few studies under this topic, being the main motivation for its study. The objective of this type of planners is to find the optimal trajectory while maximizing the speed or minimizing the lap time. One of the main conditions of racing planning is to accurately consider the dynamics of the vehicle in the algorithm calculations. In this way, one common

strategy is to handle the vehicle acceleration vector as well as the front and rear wheels slip angles under an optimization problem to find their optimal values.

In [14], the problem of trajectory planning for a racing application is solved by means of a two stage approach. In the first stage, a convex optimization that consists in minimizing the length and curvature of the path using a geometric representation is solved. This optimization results in the optimal trajectory. Then obtaining the curvature of such an optimal trajectory and using an equation that relates the curvature, speed and the maximum lateral force tires can sustain, then the optimum velocity profile is obtained.

In [13], a real-time MPC for racing trajectory planning is presented. The key point of this planning approach resides in reformulating the initial non-convex problem into a linearly constrained convex quadratic optimization problem (QP) that can be solved in real-time. A point mass kinematic model is used for predicting future states while constraining vehicle accelerations.

In [15], the authors present a racing planning strategy based on non-linear model predictive control using a kinematic representation of the vehicle. Years later, the same authors enhance this version by introducing a dynamic vehicle model between other improvements [18].

According to Liniger et al. [17], the authors propose a racing path planning strategy based on linear MPC strategy. This optimization procedure uses a linearized dynamic model of the vehicle which allows the real-time implementation. The paper presents the results with and without obstacles demonstrating in both cases a high performance.

The weaknesses that all these works have in common are the use of models of reduced complexity (kinematic-based models basically) and/or the use of a path-based planner formulation. In this work, we address these deficiencies.

In Table 1, a summary of the different approaches is presented. This table collects all the previously mentioned references and classify them as a function of the modeling type, the scenario and the time-dependency (trajectory planning or path planning).

In this paper, we present a novel approach to solve the optimal trajectory planning task for autonomous racing vehicles considering static obstacles throughout the track. In order to deal with track constraints, we express the road limits as the maximum values, positive and negative, of the allowed vehicle lateral error. In addition, such a way of limiting the road allows a straightforward extension to avoid obstacles. The way we propose to overcome obstacles is to modify the limit of the track at each instant of time. In this way, the limit values of the lateral error interval in the optimal problem vary as a function of the obstacle and vehicle positions. The key idea of this approach is to use an Optimal Quadratic Programming algorithm that maximizes the velocity vector within a certain horizon. This optimal algorithm uses a Linear Parameter Varying (LPV) vehicle model for predicting future behavior. The main idea behind the LPV modeling approach is that the non-linear prediction model can be expressed as a combination of linear models that depend on some scheduling variables without using linearization [19]. Such LPV-scheduling variables are obtained by taking the shifted result of the previous optimal iteration. Then, after discretizing the model, we solve a QP problem using a numerical solver [20]. The computational time is reduced drastically in comparison with its non-linear version. Finally, the resulting trajectory is provided to the controller as the current trajectory to be followed and the process is repeated at the next sampling time.

The paper is structured as follows: Section 2 presents the vehicle and the different models for planning. In Section 3, the optimal control algorithm formulation is presented as a non-linear problem. Section 4 addresses the proposed LPV formulation and the QP optimal problem for trajectory planning. In Section 5, we present the strategy to deal with obstacles. Simulation results are depicted and discussed in Section 6. Finally, last section presents the conclusions of the work.

2. Testing vehicle & modeling

The Driverless UPC Car¹ is a development platform for autonomous racing (see Fig. 1). This has been properly modified to operate autonomously under racing specifications. The vehicle counts with a complete net of sensors for performing localization and environment understanding. A fusion of IMU, encoders and GPS data is made by a Kalman filter in order to estimate a more precise values of the vehicle states.

2.1. Non-linear vehicle model

When a car is running in a race, it is subject to lateral and longitudinal acceleration levels much higher than those produced in normal driving situations. It is expected the vehicle to work in the dynamic limits following an optimal response in terms of traceability and therefore in terms of lap time. However, working at the limits is a hard task if a vehicle model, that faithfully represents all its dynamics, is not used.

Firstly, this subsection presents the non-linear vehicle dynamics. Such a set of behaviors is represented using the bicycle model approach where the lateral forces have been modeled using the simplified Pacejka formula [21]. The model parameters have been adjusted following an exhaustive identification procedure.

Then, denoting the state and control vectors, respectively, as

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

the discrete time non-linear model is presented as

$$x_{k+1} = f(x_k, u_k, \kappa_k), \quad (1b)$$

where variable κ_k represents the curvature at time k . The model (1b) is governed by the following equations

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

where

$$\begin{aligned} F_{yf} &= d_f \sin(c_f \tan^{-1}(b_f \alpha_f)) \\ F_{yr} &= d_r \sin(c_r \tan^{-1}(b_r \alpha_r)) \\ F_{df} &= \mu m g + \frac{1}{2} \rho C_{dA} v_x^2. \end{aligned} \quad (1d)$$

State variables v_x , v_y and ω represent the body frame velocities, i.e. linear in x , linear in y and angular velocities, respectively. States y_e and θ_e represent the body frame lateral error and the



Fig. 1. UPC Driverless vehicle.

Table 2
Dynamic model parameters of the Driverless UPC Car.

Parameter	Value	Parameter	Value
l_f	0.902 m	l_r	0.638 m
m	196 kg	I	93 kg m ²
d_f	8.255	c_f	1.6
b_f	6.1	μ	1.4
d_r	8.255	c_r	1.6
b_r	6.1	ρ	1.225 kg m ³
C_{dA}	1.64	g	9.81 $\frac{m}{s^2}$
d	2.3 m	w	1.45 m

orientation error, respectively (see Fig. 2). The control variables δ 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 (see Fig. 3). Front and rear slip angles are represented as α_f and α_r , respectively. 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. μ , ρ and g are the friction coefficient, the air density and the gravity values, respectively. C_{dA} is the product of drag coefficient and vehicle cross sectional area. All the dynamic vehicle parameters are properly defined in Table 2.

3. MPC as a planner

The main objective of the racing planning algorithm is to maximize the vehicle speed vector and minimize as much as possible an understeer/oversteer situation. On one hand, the velocity vector of the vehicle can be computed by

$$V = \frac{v_x \cos \theta_e - v_y \sin \theta_e}{1 - y_e \kappa} \quad (2)$$

as a relation between curvature and dynamic and error vehicle states.

On the other hand, the understeer and oversteer behaviors depend on the front and rear wheels slip angles and are given as

- The understeer situation is given when the front wheels slip angle is greater than the rear wheels slip angle : $\alpha_f \geq \alpha_r$.
- The oversteer situation is given when the rear wheels slip angle is greater than the front wheels slip angle : $\alpha_r \geq \alpha_f$.

These two situations should be avoided for a smooth performance of the vehicle. However, in racing environments it is allowed to have particular levels of understeer and oversteer.

Both objectives can properly be formulated as a weighted non-linear cost function where it is pretended

¹ <https://driverless.upc.edu/>.

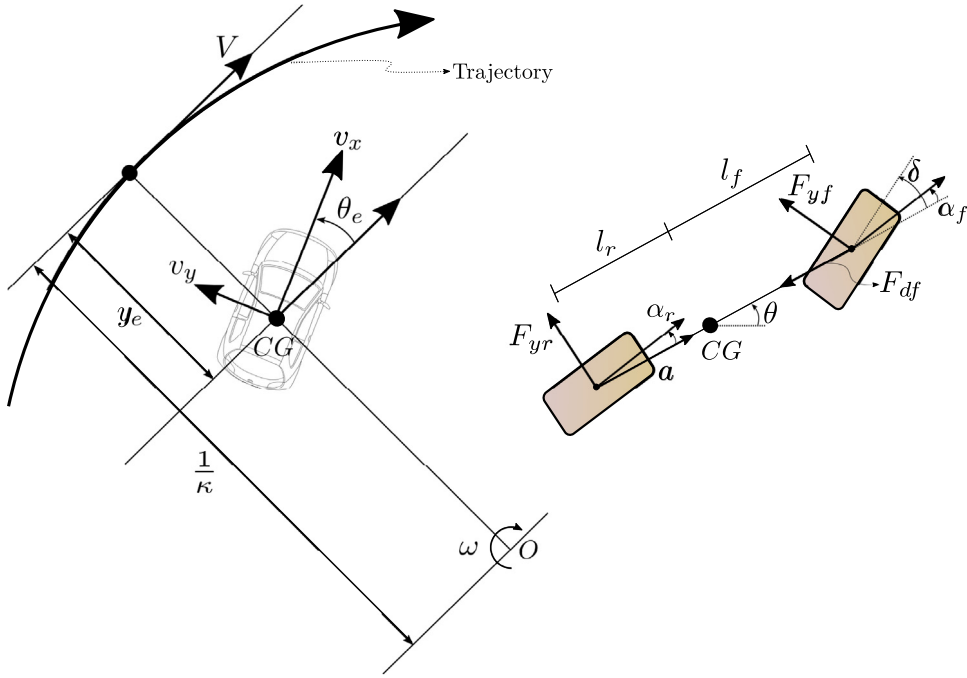


Fig. 2. Variables representation. Left: error variables representation. Right: dynamic model variables.

- to maximize the linear velocity vector which implies to minimize the travel time

$$\text{maximize : } J_V(x_{k+i}, \kappa_{k+i}) = \sum_{i=0}^N \|V_{k+i}\|_Q^2 \quad (3)$$

- to minimize the difference between the front and rear slip angles with the aim of avoiding high levels of understeer or oversteer behaviors

$$\text{minimize : } J_\alpha(x_{k+i}) = \sum_{i=0}^N \|\alpha_{fk+i} - \alpha_{rk+i}\|_R^2 \quad (4)$$

where $Q \in \mathbb{R}^{1 \times 1} \geq 0$, $R \in \mathbb{R}^{1 \times 1} \geq 0$ are the proper weighting scalars and $N \geq 0$ is the prediction horizon.

Finally, the constrained non-linear optimal control (CNLOC) problem in (5) is solved at each discrete time step k to compute an optimal control input sequence. The first control input in such a sequence is applied to the vehicle and the predicted state vector x_k is used as the starting point for the next optimization problem. Combining previous objectives, the following multi-objective optimization can be formulated

$$\min_{\Delta U_k, X_k} \left(-J_V(x_{k+i}, \kappa_{k+i}) + J_\alpha(x_{k+i}) \right)$$

s.t.

$$\begin{aligned} x_{k+i+1} &= f(x_{k+i}, u_{k+i}, \kappa_{k+i}), \quad \forall i = 0, \dots, N \\ u_{k+i} &= u_{k+i-1} + \Delta u_{k+i} \\ u_{k+i} &\in [\bar{u}, \underline{u}] \\ y_{e,k+i} &\in [\bar{y}_e - \sigma_{k+i}, \underline{y}_e + \sigma_{k+i}] \\ \alpha_{rk+i}^f &\in [\bar{\alpha}_r^f, \underline{\alpha}_r^f] \end{aligned} \quad (5)$$

where Δu_{k+i} , also known as slew rate, represents the time variation of u_{k+i} and is used to add an integral action to the system. ΔU_k and X_k represent the optimal control input sequence and state variables sequence, respectively. However, note that, although this CNLOC problem is able to provide a solution to the

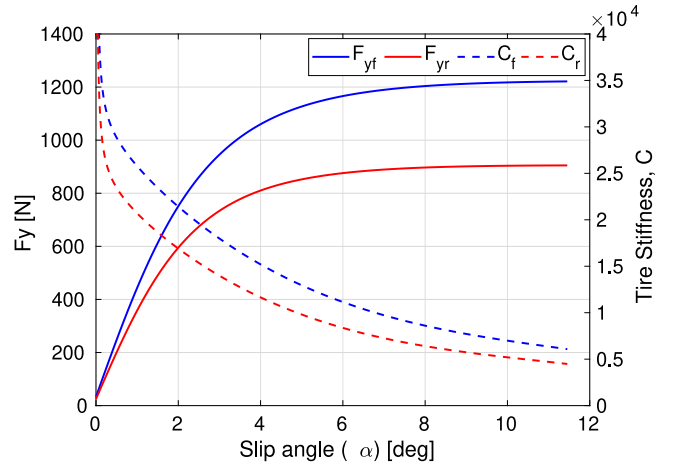


Fig. 3. Tires lateral force as function of the slip angle (from equations in (1d)) with the left y axis. Tire stiffness coefficient profiles in dashed lines with the right y axis (computed with (8)).

planning problem there exists a high computational load when solving it and, hence, becoming a not implementable solution for online planning problem in real embedded systems.

4. LPV-MPP formulation

In this section, we present a novel formulation for the optimization problem (5) using the LPV representation of the non-linear vehicle model. The key idea of this approach relies on the use of an LPV-based modeling which provides the ability to simulate the vehicle dynamics with a low computational cost. This imply that, at every discrete time k , an instantiation of the non-linear vehicle model (1) is required to obtain the LPV representation.

In addition, in order to formulate the optimization problem in the same form as (5), the non-linear cost function (3) has to

Table 3
Polynomial parameters of (8) for the front and rear tires (upper indexes f and r).

Parameter	Value	Parameter	Value
n	4	ϵ	10^{-4}
p_1^f	-2.167×10^6	p_2^f	1.284×10^6
p_3^f	-0.288×10^6	p_4^f	0.029×10^6
p_5^f	15.038		
p_1^r	-2.130×10^6	p_2^r	1.198×10^6
p_3^r	-0.252×10^6	p_4^r	0.024×10^6
p_5^r	14.551		

be represented in a quadratic form to handle the problem as a constrained linear quadratic optimal control (CLQOC) problem.

Following subsections address both problems.

4.1. Formulation of the LPV representation

A LPV system is a combination of linear models that depend on some scheduling variables [19].

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.

Then, the transformation of the non-linear model into a LPV one passes through applying two necessary reformulations.

First, the arctangent function in the slip angles equation is neglected to simplify the formulation. Note that this assumption is coherent since the slip angle value remains under 0.1745 rad where $\tan^{-1}(\alpha) \approx \alpha$.

Second, the Pacejka tire equations for front and rear wheels in (1d) are reformulated in a LPV representation for a proper introduction in the final LPV vehicle model. In this work, a LPV formulation is introduced to improve linearization-based approaches. Hence, a least-squares algorithm is used to find two polynomials as

$$F_y(\alpha) = p_1\alpha^n + p_2\alpha^{n-1} + \dots + p_n\alpha + p_{n+1}, \quad (6)$$

where p constants are the estimated coefficients that define the particular model structure and n represents the order of the corresponding polynomial.

Once the polynomial is adjusted, the embedding approach of the non-linearities inside a varying parameter has to be used in order to obtain its LPV representation. Then, the following formulation is proposed

$$F_y = C(\alpha) \alpha, \quad (7)$$

where

$$C(\alpha) = p_1\alpha^{n-1} + p_2\alpha^{n-2} + \dots + p_n + p_{n+1}/(\alpha + \epsilon) \quad (8)$$

is known as the tire stiffness coefficient and ϵ is a very small constant. In Fig. 3, both, front and rear tire stiffness coefficient curves are presented in dashed lines. Note that, as α becomes close to zero in (8), $C(\alpha)$ grows exponentially. To avoid this behavior, a saturation is added in the small interval $\alpha \in [0, 0.0075]$ such that $C(\alpha)$ value stay at 4×10^4 . Table 3 in the results section shows the coefficients used in (8).

Therefore, the non-linear equations presented in (1c) can be expressed as

$$\begin{aligned} \dot{v}_x &= a_r + \frac{-F_{yf} \sin \delta - F_{df}}{m} + \omega v_y \\ \dot{v}_y &= \frac{F_{yf} \cos \delta + F_{yr}}{m} - \omega v_x \\ \dot{\omega} &= \frac{F_{yf} l_f \cos \delta - F_{yr} l_r}{I} \\ \dot{y}_e &= v_x \sin \theta_e + v_y \cos \theta_e \\ \dot{\theta}_e &= \omega - \frac{v_x \cos \theta_e - v_y \sin \theta_e}{1 - y_e \kappa} \kappa \\ \alpha_f &= \delta - \frac{v_y}{v_x} + \frac{l_f \omega}{v_x} \\ \alpha_r &= -\frac{v_y}{v_x} - \frac{l_r \omega}{v_x} \\ F_{yf} &= C_f(\alpha_f) \alpha_f \\ F_{yr} &= C_r(\alpha_r) \alpha_r \\ F_{df} &= \mu m g + \frac{1}{2} \rho C_{dA} v_x^2. \end{aligned} \quad (9)$$

Then, denoting the state and control vectors as in (1a), the non-linear model (9) is transformed into the discrete LPV representation

$$x_{k+1} = A(\zeta_k) x_k + B(\zeta_k) u_k, \quad (10a)$$

by embedding the non-linearities within varying linear parameters. Note that sub-index k represents the discrete time. Each one of these parameters is function of the vector of scheduling variables defined as

$$\zeta_k := [v_x, v_y, \theta_e, \kappa, y_e, \delta]_k. \quad (10b)$$

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

Then, the LPV matrices, i.e. $A(\zeta_k)$ and $B(\zeta_k)$, are obtained using the non-linear embedding approach [22] as

$$A(\zeta_k) = \begin{bmatrix} A_{11} & A_{12} & A_{13} & 0 & 0 & 0 & 0 \\ 0 & A_{22} & A_{23} & 0 & 0 & 0 & 0 \\ 0 & A_{32} & A_{33} & 0 & 0 & 0 & 0 \\ 0 & A_{42} & 0 & 1 & A_{45} & 0 & 0 \\ -A_{51} & A_{52} & A_{53} & 0 & 1 & 0 & 0 \\ 0 & A_{62} & A_{63} & 0 & 0 & 0 & 0 \\ 0 & A_{72} & A_{73} & 0 & 0 & 0 & 0 \end{bmatrix} \quad (10c)$$

and

$$B(\zeta_k) = \begin{bmatrix} -\frac{1}{m} \sin \delta C_f & 1 \\ \frac{1}{m} \cos \delta C_f & 0 \\ \frac{1}{I} \cos \delta C_f l_f & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{1}{T_s} & 0 \\ 0 & 0 \end{bmatrix}, \quad (10d)$$

being

$$\begin{aligned}
A_{11} &= 1 + \left(\frac{-\mu g}{v_x} - \frac{\rho C_d A v_x}{2m} \right) T_s \\
A_{12} &= \frac{C_f \sin \delta}{m v_x} T_s \\
A_{13} &= \left(\frac{C_f l_f \sin \delta}{m v_x} + v_y \right) T_s \\
A_{22} &= 1 + \left(-\frac{C_r + C_f \cos \delta}{m v_x} \right) T_s \\
A_{23} &= \left(-\frac{C_f l_f \cos \delta - C_r l_r}{m v_x} - v_x \right) T_s \\
A_{32} &= -\frac{C_f l_f \cos \delta - l_r C_r}{l v_x} T_s \\
A_{33} &= 1 + \left(-\frac{C_f l_f^2 \cos \delta + l_r^2 C_r}{l v_x} \right) T_s \\
A_{51} &= \frac{\kappa}{(1 - y_e \kappa)} T_s \\
A_{52} &= \frac{\kappa}{(1 - y_e \kappa)} \sin \theta_e T_s \\
A_{53} &= T_s, \quad A_{42} = \cos \theta_e T_s \\
A_{45} &= v_x T_s, \quad A_{62} = \frac{-1}{v_x T_s}, \quad A_{63} = \frac{-l_f}{v_x T_s} \\
A_{72} &= \frac{-1}{v_x T_s}, \quad A_{73} = \frac{l_r}{v_x T_s}
\end{aligned} \tag{10e}$$

where T_s represents the discretization sampling time. For a easier comprehension $C_i(\alpha_i)$ is denoted by C_i being $i = f, r$.

4.2. Convexifying the objective function

At this point, the cost function of the CNLOC problem (5) consists on a quadratic term (J_α) and a non-linear part (J_V). In this section, the methodology for convexifying the non-linear term J_V is addressed. The objective is to find out a linear-quadratic formulation that approximates the non-linear equation of the vehicle velocity

$$V = \frac{v_x \cos \theta_e - v_y \sin \theta_e}{1 - y_e \kappa}, \tag{11}$$

such as

$$V \approx V^{QP} = s^T Q s + q^T s. \tag{12}$$

Since the original objective function in the optimization problem is to maximize (11), then we look for a concave formulation such that it can be introduced in the CLQOC problem as the minimization of its convex version. Studying the problem, we find that approximating the original function is a difficult task and only making some assumptions we can find a suitable result. In particular, on one hand, we observe that there is not a strong relationship with variable κ , such that $s := \begin{bmatrix} v_x & v_y & \theta_e & y_e \end{bmatrix}^T$. On the other hand, we consider the use of a diagonal Q matrix which simplifies the approximation of (11) using (12).

Least-squares techniques for fitting polynomials are limited for this purpose to provide a quadratic model, with constant and quadratic terms but avoiding the linear term which may not fit the objective.

To solve this problem, we propose the following constrained linear quadratic optimal problem

$$\begin{aligned}
\min_{Q, q} \quad & \sum_{i=0}^N \|V_i - V_i^{QP}\|^2 \\
\text{s.t.} \quad & \\
& V_i^{QP} = s_i^T Q s_i + q^T s_i \\
& \text{diag}(Q) < 0
\end{aligned} \tag{13}$$

$$\begin{aligned}
V_i^{QP} &= s_i^T Q s_i + q^T s_i \\
\text{diag}(Q) &< 0
\end{aligned}$$

where $Q \in \mathbb{R}^{4 \times 4} = Q^T < 0$ and $q \in \mathbb{R}^{4 \times 1}$ are the optimization variables and N denotes the length of the optimization problem.

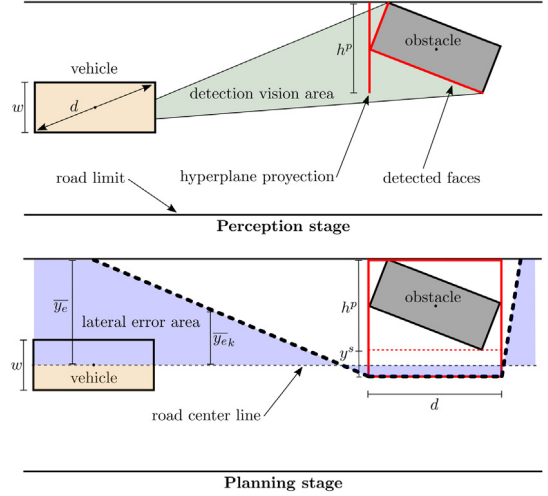


Fig. 4. The above figure depicts how the perception layer detects the vehicle using stereo-based cameras. Below, the planning layer computes the lateral error area based on the obstacle set (red box). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Note that due to the last constraint in (13), the resulting V^{QP} is defined as a strictly concave function.

Finally, after formulating the LPV model and the convex objective function, we present the following constrained linear-quadratic optimization control (CLQOC) problem that is solved at each time k to determine the next sequence of states considering that the values of x_k and u_{k-1} are known

$$\begin{aligned}
\min_{\Delta U_k, X_k} \quad & \left(-\sum_{i=0}^N V^{QP}(x_{k+i}) + \sum_{i=0}^N \|\alpha_{f_{k+i}} - \alpha_{r_{k+i}}\|_R^2 \right. \\
& \left. + \sum_{i=0}^N \|\sigma_{k+i}\|_P^2 \right) \\
\text{s.t.} \quad & \\
& x_{k+i+1} = A(\zeta_{k+i})x_{k+i} + B(\zeta_{k+i})u_{k+i}, \quad \forall i = 0, \dots, N \\
& u_{k+i} = u_{k+i-1} + \Delta u_{k+i} \\
& u_{k+i} \in [\bar{u}, \underline{u}] \\
& y_{e_{k+i}} \in [\bar{y}_{ek} + \sigma_{k+i}, \underline{y}_{ek} - \sigma_{k+i}] \\
& \alpha_{f_{k+i}}^f \in [\bar{\alpha}_r^f, \underline{\alpha}_r^f]
\end{aligned} \tag{14}$$

where $R \in \mathbb{R}^{1 \times 1} \geq 0$ and $P \in \mathbb{R}^{1 \times 1} \geq 0$ are weighting scalars and $N \geq 0$ is the prediction horizon. Note that, the third component in the objective function aims to minimize the slack variable σ introduced over the lateral error state. This choice is made to provide the optimal problem some flexibility.

5. Introducing static obstacles

This section addresses the static obstacle avoidance problem during the planning task. The procedure is mainly based on two steps. First, the computation of a safety polytope that contains the obstacle is done based on the information provided by the perception layer (not presented in this paper). This polytope is chosen to be a rectangle in this work. Second, the computation of the new lateral bounds of the road taking into account the obstacle polytope is addressed.

The proper detection and position computation of the particular obstacle are simulated as if they were done by a higher

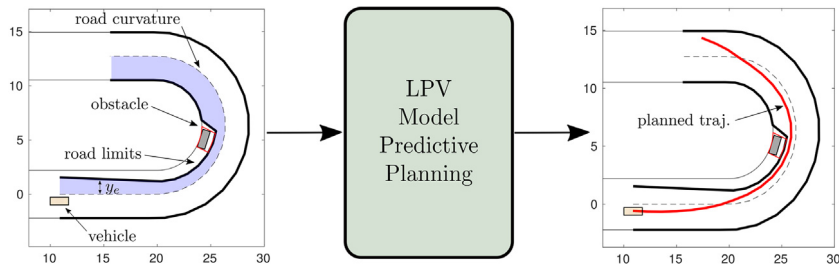


Fig. 5. Simplified view of the planning strategy. The left part represents the set of inputs, i.e. vehicle position, road limits and track curvature. The right side shows the planned trajectory for a particular discrete time k .

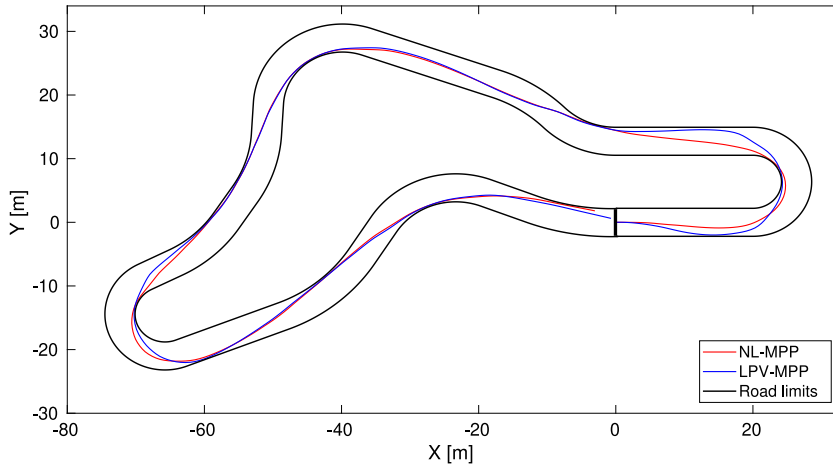


Fig. 6. Resulting two laps trajectories for the proposed LPV approach and the NL approach on the free obstacle scenario.

perception layer using stereo-based cameras. This hardware provide a cloud of points on the obstacle's edge with their respective RGB data and distance to the camera. Then, projecting these cloud points to the hyperplane h^p , we are able to compute the frontal face of the polytope containing the obstacle (see Fig. 4). Note that, such a hyperplane is always orthogonal to the road orientation.

At this point, the planning stage computes the polytope that contains the obstacle. First, the hyperplane h^p is extended using a safety distance y^s . Then, the rectangle is closed by using the vehicle diagonal length, denoted by d . Note that, the obstacle is assumed to have the same width and diagonal length than the UPC Driverless vehicle (see Table 2). In addition, the election of y^s is made taking into account the half width of the vehicle plus an extra distance for safety reasons. Once the polytope is obtained, the new lateral bounds can be computed. To do so, an incremental variation of the limit of the lateral error ($\bar{y}_{e,k}$) is determined using Algorithm 1 computing then the lateral error vector as

$$\bar{Y}_e = [\bar{y}_{e0}, \dots, \bar{y}_{eN}]. \quad (15)$$

This vector is computed at every discrete time k and introduced as an input to the CLQOC problem. Note that this approach may be conservative but very efficient computationally.

6. Results

In this section, the performance of the proposed racing LPV MPP approach (14) is evaluated. To do so, a comparison against its non-linear version (5) is presented as well as the performance in different scenarios. However, before entering into the details, it is important to emphasize that in racing scenarios it is necessary to

Algorithm 1: Road limits algorithm (see Fig. 4)

Input : s_{obs} , h^p , last predicted vehicle states (1a)

Output: Free road limits (\bar{Y}_e)

- 1 integrate (11) to obtain \hat{s} using the last predicted states and the reference κ
 - 2 given the predicted vector \hat{s} , obtain the vector index (i_{obs}) at which $\hat{s} \approx s_{obs}$
 - 3 compute $\Delta y_e = \frac{h^p + y^s}{N}$, where N is the prediction horizon
 - 4 **for** $i = 1$; $i < N + 1$; $i = i + 1$ **do**
 - 5 $\sum_{E_y} += \Delta y_e$;
 - 6 **if** $\hat{s}(i_{obs} - i) > 0$ **then**
 - 7 $\bar{Y}_e(i) = \sum_{E_y}$
 - 8 **end**
 - 9 **end**
-

have a minimum of knowledge about the evolution of the track. Thus, in this work, we consider the curvature of the track as a known variable to perform the racing trajectory planning.

The details of the experimental set up and simulated scenarios are presented in the following.

6.1. Simulation/experimental set up

For evaluating the proposed architecture, we perform simulations using the UPC Driverless vehicle model which is described by the equations in (1c)–(1d) with parameters defined in Table 2.

Note that, at the first CLQOC iteration the time evolution of the scheduling vector is not known. At this point, we solve

Algorithm 2: LPV-MP Planning algorithm (see Fig. 5)

Input : Current vehicle states (x_k), past control input (u_{k-1}), Free road limits (\bar{Y}_e), Track curvature (κ)
Output: Vehicle states defining the predicted trajectory (x_{k+i}), $i := 1, \dots, N$

```

1 if  $k = 1$  then
2   | initialize the evolution of scheduling vector  $\zeta$  (10b)
3 else
4   | instantiate the scheduling vector  $\zeta$  (10b) using previous predicted data
5 end
6 compute the set of  $N$  LPV models (10), i.e.  $A_{k+i}$  and  $B_{k+i}$ ,  $i := 0, \dots, N$ 
7  $x_{k+i} \leftarrow \text{CLQOC}(A(\zeta), B(\zeta), x_k, u_{k-1}, \bar{Y}_e, \kappa)$ ,  $i := 1, \dots, N$ 
8 go to step 1
9
10 Function CLQOC( $A(\zeta), B(\zeta), x_k, u_{k-1}, \bar{Y}_e, \kappa$ ):
11   | solve QP (14) using GUROBI,  $x_{k+i} = \text{CLQOC}(A(\zeta), B(\zeta), x_k, u_{k-1}, \bar{Y}_e, \kappa)$ 
12   | interpolate the solution at the control sampling time
13   return

```

this problem by generating such evolution based on previous knowledge on how the states of the system evolve. Once this initialization is done, the predictions are used to instantiate the set of next N models of the LPV model. The sampling time used is $T_s = 300$ ms and such a prediction horizon N is set to 15 steps, this implies 4.5 s of future behavior prediction.

Matrices $A(\zeta)$ and $B(\zeta)$ in (14) are instantiated online before the optimization starts which implies a set of N LPV models entering the optimal problem. Note that, tire stiffness coefficients in (10e), i.e. C_f and C_r , are also properly instantiated online using (8) and Table 3 as a function of α_f and α_r .

In addition, since the model is curvature dependent, we need to know the nominal curvature of the track as well as the limits of the road. These limits will vary depending on possible static obstacles throughout the track.

The solutions of problem (13) are obtained considering

$$\begin{aligned}
 Q &= \text{diag} \left[-1.2 \times 10^{-4} \quad -9.704 \quad 3.5 \times 10^{-5} \quad -0.154 \right] \\
 q &= \left[1.007 \quad 0.187 \quad 6.1 \times 10^{-7} \quad -0.032 \right].
 \end{aligned} \tag{16}$$

Note that, since the vector state in (14) is the one defined in (1a) the obtained matrices Q and q are restructured properly to be

$$\begin{aligned}
 Q &= \text{diag} \\
 &\left[-1.2 \times 10^{-4} \quad -9.704 \quad 0 \quad 3.5 \times 10^{-5} \quad -0.154 \quad 0 \quad 0 \right] \\
 q &= \left[1.007 \quad 0.187 \quad 0 \quad 6.1 \times 10^{-7} \quad -0.032 \quad 0 \quad 0 \right].
 \end{aligned} \tag{17}$$

The control inputs bounds are set to $\bar{\delta} = -\underline{\delta} = 0.3$ rad and $\bar{a}_r = -a_r = 12 \frac{\text{m}}{\text{s}^2}$. Front and rear wheels slip angles are limited to $\bar{\alpha}_r^f = -\alpha_r^f = 0.16$ rad. Lateral error limits are set in next sections since they will vary with the obstacles.

Both the LPV-MPP and NL algorithms are implemented in Matlab R2017a on a Dell Inspiron 15 5000 Series using a Intel core i7-8550U CPU @ 1.80 GHzx8. For solving the non-linear optimization problem (5) used as baseline solution, IPOPT [23] solver is used while solving the QP problems the proposed approach GUROBI [20] is used, both through YALMIP [24] framework.

The pseudo-code for the implementation of LPV-MP Planning is shown in Algorithm 2.

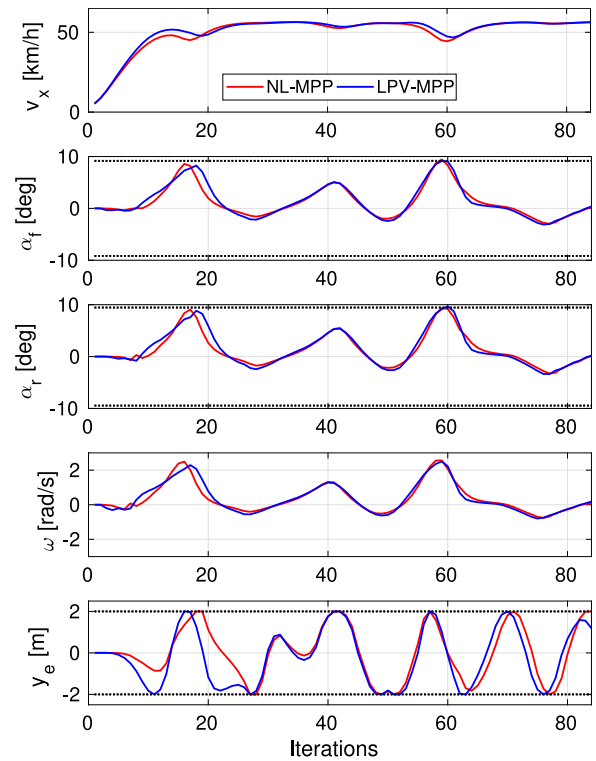


Fig. 7. Velocities, slip angles and lateral error throughout the simulation: dotted lines represent the state limits considered in the optimal problem.

6.2. Free track racing planning

In this experiment, we compare the LPV-MPP strategy against its non-linear version (NL-MPP) in a free obstacles track. Algorithm 2 is used to obtain the LPV-MPP results. To solve the non-linear version for comparison, we solve problem (5).

The simulated trajectories are depicted in Fig. 6 for one lap. It can be seen how the non-linear version performs a bit smoother than the proposed LPV approach, even though at a higher computational cost, around 50 times slower (see Fig. 8).

The simulated track has a width of 4 m, hence, the lateral error bounds in (5) and (14) are set to be symmetric such that $\bar{y}_e = -y_e = 2$ m.

The velocities, slip angles and lateral errors for the whole simulation are displayed in Fig. 7. It can be seen that LPV-MPP

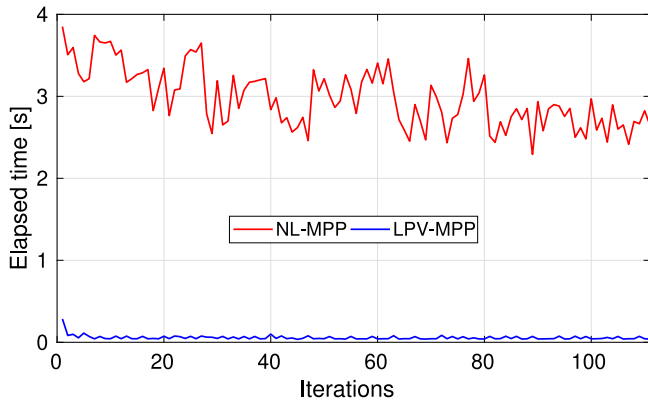


Fig. 8. Computational time cost for both compared strategies: NL-MPP mean elapsed time is 2.961 s and LPV-MPP mean elapsed time is 0.0567 s.

Table 4
Mean values of longitudinal velocity and acceleration, elapsed time and slip angles difference.

	\bar{v}_x	\bar{a}_x	\bar{t}_e	$\bar{\alpha}_r - \bar{\alpha}_l$
NL-MPP	50.1	0.615	2.961	$-4.8 \cdot 10^{-4}$
LPV-MPP	50.6	0.616	0.057	$-6.5 \cdot 10^{-4}$

delivers a similar solution to the NL-MPP one for the compared vehicle states. The lateral error with respect to the center line of the road is the most important state for solving this problem. We allow the vehicle to have up to 2 m of lateral error in order to find the best path. By looking into the iterations interval [60–80], it can be appreciated that the LPV-MPP y_e achieves higher values implying then the vehicle approaches more to the road limits in comparison to the NL-MPP response. However, this is still a good solution for the racing trajectory planning that allows the real time implementation.

In Table 4, a comparison is made in terms of mean values. The great difference is not in the solution but in the elapsed time at each iteration what makes the LPV-MPP strategy a much faster approach and therefore a suitable option to be implemented on-line in real-time on embedded systems.

6.3. Static obstacles racing planning

In this section, the validation is made considering static obstacles within the track. Three obstacles are introduced in strategic points of the circuit. The objective of the planner is to maximize the lap velocity while avoiding the three obstacles introduced along the circuit.

The obstacle detection is assumed to be done outside this planning procedure as it was explained in Section 5. Then, as it is depicted in Fig. 5, the new limits of the road are updated at every iteration taking into account the obstacle. Therefore, these limits are introduced as a new bounds for lateral error state ($\bar{y}_e, \underline{y}_e$) in the optimal problem (14).

As in Section 6.2, we compare the performance of the NL-MPP and LPV-MPP planning approaches. The planned trajectories are shown in Fig. 9. At a first glance, it can be seen that the NL-MPP approach provides a smoother trajectory.

The velocities, slip angles and lateral errors for the whole simulation are presented in Fig. 10. In this graphical comparison, it is observed a greater difference between their responses. Both slip angles, which are important states, remain inside the allowed region. However, the LPV-MPP approach perform sharper solutions.

Table 5
Mean values of longitudinal velocity and acceleration, elapsed time and slip angles difference.

	\bar{v}_x	\bar{a}_x	\bar{t}_e	$\bar{\alpha}_r - \bar{\alpha}_l$
NL-MPP	49.9	0.613	3.032	$-6.4 \cdot 10^{-4}$
LPV-MPP	50.6	0.607	0.062	$-8.7 \cdot 10^{-4}$

Table 5 illustrates the resulting comparison in terms of mean values. It can be observed that the non-linear approach is able to minimize more the difference between slip angles reducing then over and understeering behaviors. However, the LPV-MPP performs an acceptable solution and its computational time is much lower.

To conclude this section, we measure the elapsed time of both performances. However, we do not observe large differences comparing to the case without obstacles so we refer to the same Fig. 8.

7. Conclusion

In this paper, we propose an effective online planning solution for autonomous vehicles where we focus on improving the computational load while preserving high levels of performance in racing scenarios. While most of the strategies in the literature overcome the planning problem using linearized models, we aim to reformulate the non-linear vehicle equations to be expressed in a Linear Parameter Varying (LPV) form. To formulate the model predictive planning (MPP) problem, we first convexify the non-linear objective function in a linear-quadratic form. Then, we solve it using the LPV vehicle model for predicting the trajectory in a certain horizon. In addition, the algorithm has the ability of avoiding obstacles in a very simple way. The limits of the lateral error vary inside the model predictive problem to take into account static obstacles.

We test and compare the performance of the proposed strategy against its non-linear approach through simulations. We focus on the performance of our planning approach in a racing track. First, in a free track scenario and next in a scenario with static obstacles allow to show that the proposed method reduces the planning computation elapsed time while finding a suitable trajectory under the proposed constraints.

In future steps of this study, we will focus on the experimental implementation and validation of the proposed planning strategy in the real test vehicle. We will also investigate the viability of representing the 4-wheel dynamic model into a LPV formulation and will test it in the proposed planning environment.

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.

Acknowledgment

Thanks to the support of AEI and FEDER through the project SCAV (ref. MINECO DPI2017-88403-R). The author is supported by a FI AGAUR grant (ref 2017 FI B00433). Thanks to the UPC Driverless Team for supporting this study.

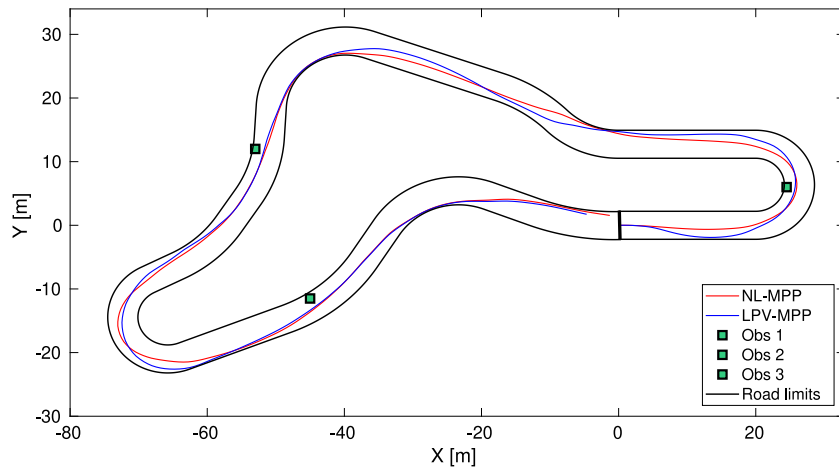


Fig. 9. Resulting one lap trajectories for the proposed LPV approach and the NL approach on the obstacle scenario.

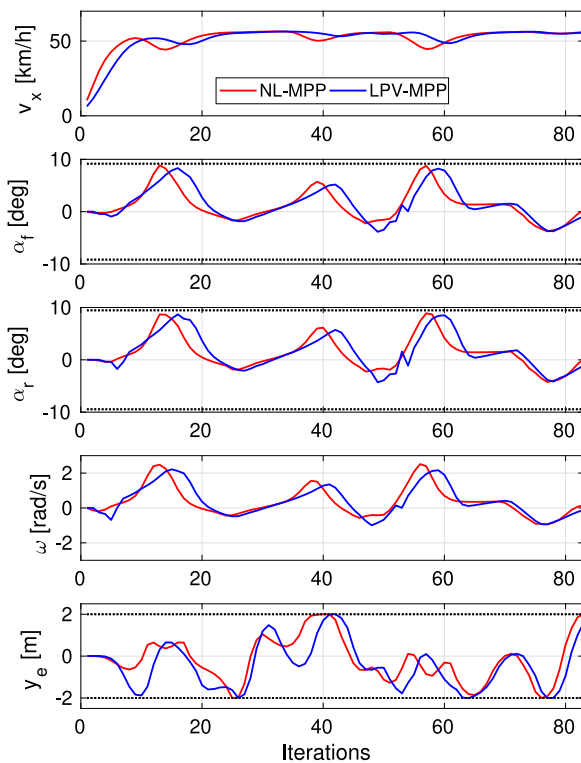


Fig. 10. Velocities, slip angles and lateral error throughout the simulation: dotted lines represent the state limits considered in the optimal problem.

References

- [1] Formula Student Germany, URL: <https://www.formulastudent.de/>.
- [2] Roborace, URL: <https://roborace.com/>.
- [3] C. Katrakazas, M. Qudus, W.H. Chen, L. Deka, Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions, *Transp. Res. C* 60 (2015) 416–442.
- [4] B. Paden, M. Čáp, S.Z. Yong, D. Yershov, E. Frazzoli, A survey of motion planning and control techniques for self-driving urban vehicles, *IEEE Trans. Intell. Veh.* 1 (1) (2016) 33–55.
- [5] C. Goerzen, Z. Kong, B. Mettler, A survey of motion planning algorithms from the perspective of autonomous UAV guidance, *J. Intell. Robot. Syst.* 57 (1–4) (2010) 65.
- [6] H. Zhang, S. Yang, Smooth path and velocity planning under 3D path constraints for car-like vehicles, *Robot. Auton. Syst.* 107 (2018) 87–99.
- [7] J.P. Talamino, A. Sanfeliu, Anticipatory kinodynamic motion planner for computing the best path and velocity trajectory in autonomous driving, *Robot. Auton. Syst.* 114 (2019) 93–105.

- [8] D. González, J. Pérez, V. Milanés, F. Nashashibi, A review of motion planning techniques for automated vehicles, *IEEE Trans. Intell. Transp. Syst.* 17 (4) (2015) 1135–1145.
- [9] A. Liniger, *Path Planning and Control for Autonomous Racing*, ETH Zurich, 2018.
- [10] M.G. Plessen, P.F. Lima, J. Martensson, A. Bemporad, B. Wahlberg, Trajectory planning under vehicle dimension constraints using sequential linear programming, in: 2017 IEEE 20th International Conference on Intelligent Transportation Systems, ITSC, IEEE, 2017, pp. 1–6.
- [11] C. Liu, S. Lee, S. Varnhagen, H.E. Tseng, Path planning for autonomous vehicles using model predictive control, in: *Intelligent Vehicles Symposium (IV)*, IEEE, 2017, pp. 174–179.
- [12] M. Ahmadi Mousavi, B. Moshiri, Z. Heshmati, Linear time varying MPC based path planning of an autonomous vehicle via convex optimization, *J. Iran. Assoc. Electr. Electron. Eng.* 14 (4) (2018) 79–88.
- [13] B. Alrifae, J. Maczajewski, Real-time trajectory optimization for autonomous vehicle racing using sequential linearization, in: 2018 IEEE Intelligent Vehicles Symposium (IV), IEEE, 2018, pp. 476–483.
- [14] D. Caporale, A. Fagiolini, L. Pallottino, A. Settimi, A. Biondo, F. Amerotti, et al., A planning and control system for self-driving racing vehicles, in: 2018 IEEE 4th International Forum on Research and Technology for Society and Industry, RTSI, IEEE, 2018, pp. 1–6.
- [15] R. Verschuere, S. De Bruyne, M. Zanon, J.V. Frasch, M. Diehl, Towards time-optimal race car driving using nonlinear MPC in real-time, in: *Decision and Control (CDC)*, 2014 IEEE 53rd Annual Conference on, IEEE, 2014, pp. 2505–2510.
- [16] F. Hegedus, T. Bécsi, S. Aradi, P. Gápár, Model based trajectory planning for highly automated road vehicles, *IFAC-Pap. Online* 50 (1) (2017) 6958–6964.
- [17] A. Liniger, A. Domahidi, M. Morari, Optimization-based autonomous racing of 1: 43 scale RC cars, *Optim. Control Appl. Methods* 36 (5) (2015) 628–647.
- [18] R. Verschuere, M. Zanon, R. Quirynen, M. Diehl, Time-optimal race car driving using an online exact hessian based nonlinear MPC algorithm, in: *Control Conference (ECC)*, 2016 European, IEEE, 2016, pp. 141–147.
- [19] O. Sename, P. Gaspar, J. Bokor (Eds.), *Robust Control and Linear Parameter Varying Approaches: Application to Vehicle Dynamics*, Vol. 437, Springer, 2013.
- [20] Gurobi Optimization, *Gurobi Optimizer Reference Manual*, Google Scholar, 2014.
- [21] Hans Pacejka, *Tire and Vehicle Dynamics*, Elsevier, 2005.
- [22] A. Kwiatkowski, M.T. Boll, H. Werner, Automated generation and assessment of affine LPV models, in: *Decision and Control*, 45th IEEE Conference, 2006, pp. 6690–6695.
- [23] Andreas Wächter, Lorenz T. Biegler, On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming, *106 (1)* (2006) 25–57.
- [24] Johan Lofberg, YALMIP: A toolbox for modeling and optimization in MATLAB, in: *Computer Aided Control Systems Design*, 2004 IEEE International Symposium on, IEEE, 2004, pp. 284–289.

Further reading

- [1] M. Kissai, B. Monsuez, A. Tapus, D. Martinez, Control allocation of active rear steering and vehicle dynamics control using a new tire model, *Int. J. Mech. Eng. Robot. Res.* (2018).



Eugenio Alcalá was born in Spain in 1992. He received the University degree in electronics and automation in 2014 from Universidad de Zaragoza (UNIZAR), Teruel, Spain; and the robotics and control master degree in 2016 from Universitat Politècnica de Catalunya (UPC), Barcelona, Spain. He is currently developing his Ph.D. thesis within the automatic control field applied on autonomous vehicles. He is on the Advanced Control Systems (SAC) research group of the Research Center for Supervision, Safety and Automatic Control (CS2AC) at UPC. His main research interests are linear parameter varying (LPV) based, optimal and robust control applied to solve new autonomous driving problems.



Vicenç Puig was born Spain in 1969. He received the Ph.D. degree in control engineering in 1999 and the telecommunications engineering degree in 1993, both from Universitat Politècnica de Catalunya (UPC), Barcelona, Spain. He is currently a full professor of automatic control and the leader of the Advanced Control Systems (SAC) research group of the Research Center for Supervision, Safety and Automatic Control (CS2AC) at UPC. His main research interests are in fault detection, isolation and fault-tolerant control of dynamic systems. He has been involved in several

European projects and networks, and has published many papers in international conference proceedings and scientific journals.



Joseba Quevedo is Professor at the Polytechnic University of Catalonia (UPC) in automatic control domain, where he has been full professor since 1991.

Joseba Quevedo has published more than 400 articles in scientific journals and congresses in the areas of advanced control, identification and estimation of parameters, detection and diagnosis of faults, fault-tolerant control and their applications in large-scale systems (Systems water distribution and sewage networks) and industrial processes. He has participated in many Spanish and European research projects in the

field of advanced control and supervision and its application in complex Systems.

Since 2014 he is the director of the CS2AC UPC Research Center (Supervision Center and Security and Automatic Control), with more than 30 researchers, and from the year 2014 to 2018 he was the president of the CEA (Spanish Committee of Automatic Control), with more than 300 Automation and Robotics specialists from Spain.