Publisher version DOI: 10.1109/TCST.2022.3196880

A Non-Optimization-Based Dynamic Path Planning for Autonomous Obstacle Avoidance

Matteo Corno[a], Alex Gimondi[a], Giulio Panzani[a], Federico Roselli[a], Andrea Alessandretti[b] and Sergio M. Savaresi[a]

***Abstract*—This paper presents a non-optimization-based framework for path planning and tracking for evasive maneuvers in autonomous cars. The framework exploits a two-layer approach where a path planner generates a reference trajectory that is then tracked by a path tracking controller. A nested curvature preview controller implements path tracking. In the paper, we show how to describe the closed-loop performance of the controller. The quantification of the closed-loop performance in the frequency domain guides the generation of the evasive path. In this way, the algorithm generates a path that avoids the obstacle (if possible) accounting for both static and dynamic constraints. The proposed framework thus provides a non-optimization-based way to integrate the characteristics of the path tracker in the path planner algorithm, thus avoiding the need to define cost functions and use third party optimizers. The paper validates the proposed evasive maneuver strategy in simulation and on an instrumented vehicle. First, we test the trajectory tracker, showing that it tracks aggressive trajectories (with lateral acceleration close to 1 g) with an error smaller than 30 cm. Subsequently, we integrate the curvature preview with the path generator and show the joint generation-tracking performance in two different scenarios.**

I. INTRODUCTION

Autonomous driving (declined in its several levels [1]) is one of the main drivers of automotive research. The advantages of autonomous driving encompass many aspects of mobility: traffic [2], [3], pollution [4] and safety [5], [6].

From the control point of view, most autonomous driving systems ([7], [8]) are organized around two main control subsystems[1] as shown in Figure 1: path planning and path tracking [9], [10].

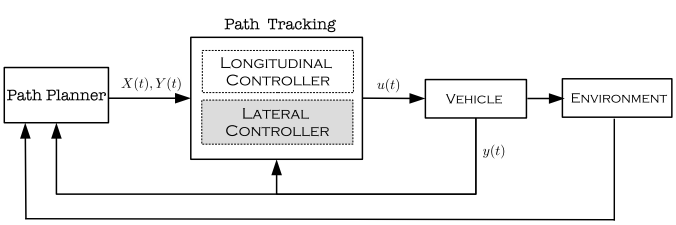

Fig. 1. Functional block diagram of an autonomous vehicle guidance system.

[a] M. Corno, A. Gimondi, F. Roselli and S. Savaresi are with Dipartimento di Elettronica Informazione e Bioingegneria at Politecnico di Milano. [b] A. Alessandretti is with Magneti Marelli. matteo.corno@polimi.it, alex.gimondi@polimi.it, giulio.panzanii@polimi.it federico.roselli@gmail.com, andrea.alessandretti@magnetimarelli.com, federico.roselli@gmail.com, sergio.savaresi@polimi.it.

[1] on top of these control modules, an autonomous driving system requires advanced sensing, localization and communication systems which are out of the scope of this contribution

The outer-most controller, relying on maps, vision and other environment sensors, generates a reference path to safely navigate the environment.

The inner-most controller, the path tracking controller, assumes that a reference trajectory is available and acts on the vehicle actuators (steering, throttle, and braking) to track the reference path. This task clearly falls within the classical closed-loop control framework: sensor fusion algorithms estimate the position of the vehicle with respect to the desired track and the controller computes the control action to track the path. The path tracking task can be further organized in a longitudinal controller (*i.e.* reference speed tracking) and lateral controller (*i.e.* reference path tracking).

The literature on the subject is vast and rapidly growing. The contributions on path-tracking are classifiable along three main axes: driving scenarios, actuation configuration, and control methodology.

Driving Scenarios. Researchers first dealt with low speed driving and highway driving [11] to then move to high speed and aggressive driving [12], [13] as the field matured. The latest trend is now autonomous racing [14], [15]. Autonomous racing may seem fanciful, but the ability to track a path that requires to operate the vehicle close to its handling limits is paramount in every day driving. Even in Level-3 autonomous systems, the vehicle has to be capable of avoiding a sudden obstacle.

Actuation Configuration. Most works consider the two path tracking subsystems of Figure 1 as independent; a steer-by-wire system tracks the desired path, while powertrain control follows the desired speed. Recently, coordinated control has been introduced [16]–[18]. Coordinated control allows the vehicle to reach its physical limits at the cost of needing an accurate model of the vehicle (especially the tyre characteristics).

Control Methodology. The open scientific literature has applied a great variety of control paradigms; the most successful ones include sliding mode control [19], non-linear control [20], [21], potential field control [22]–[24], Model Predictive Control [9], [25], [26], and Optimal Preview Control [27], [28]. While a standard benchmark to compare different methods does not exist, the literature agrees on one important point: accurate path tracking requires a look-ahead (or preview).

The path-tracking subsystem is a closed-loop control system and as such it has limitations. Static, or steady-state, cornering limitations mainly depend on friction conditions. The maximum velocity at which a vehicle can negotiate a corner depends on the road-tyre friction. Dynamic limitations are more difficult to characterize and depend on many factors, for example actuator bandwidth, vehicle yaw moment of inertia, and load transfer dynamics.

Also the path generation subsystem is a rich research field with many important contributions by many scientific communities (artificial intelligence, computer science and systems and control). An effective path generation calls for the solution of two tasks: mapping and obstacle detection, and trajectory generation.

Mapping and Obstacle Detection. Several sensors can provide useful information for the representation of the surrounding space like LiDARs, [29], stereo cameras [30] and sonars. The measured data are then used to generate occupancy grids on a discretized space and calculate collision free waypoints. Other interesting approaches use the Artificial Potential Field method [31] that associates repulsive potential fields to obstacles and attractive ones to the goals.

Trajectory Generation. Once the traversable regions are identified, the desired path needs to be computed. Different algorithms exist. Common methods rely on graph search-based algorithms such as A^* [32] or random sampling-based methods like RRT [33]. While effective in finding the shortest path and guaranteeing a safe distance from known obstacles, graph based solutions cannot account for dynamic constraints. Dynamic optimization techniques can successfully handle complex scenarios taking into account kinematic or dynamic vehicle models as well as other dynamic cost function such as comfort [34]. Among the methods used for optimization, dynamic programming is particularly appreciated because it can deal with non convex problems [35]. This flexibility comes at the cost of a high computational cost that limits the applicability to heavily simplified models. Other approaches lighten the computational burden by limiting the search to curves belonging to specific classes of functions like polynomials [36], splines [37], [38] and clothoids [39] or by using a windowed optimization approach [40].

Many works in autonomous vehicles employ the above cited path generation techniques considering only the static limits of maximum lateral acceleration and kinematic constraints. This is effective and efficient for *normal* driving, but it is risky in emergency situations. A kinematically feasible trajectory (even if it accounts for tyre nonlinearities) may not be dynamically trackable by the path tracker. This may result in a collision, because the path tracking controller is not capable of tracking the desired trajectory. The most common solution to this issue is one-level Model Predictive Control (MPC) [41]; for example [42] implements a single module that takes care of both stabilization and obstacle avoidance on low curvature roads. Funke [43] develops the idea of mediating between stabilization, collision avoidance and path tracking. [44] shows how to compute a dynamically feasible obstacle avoidance maneuver, but are not able to solve the resulting nonlinear optimization problem in real time and [45] propose a lane change maneuver in a highway setting, the approach is not experimentally validated. More recently, thanks to the development of optimization toolboxes, researchers have successfully experimentally validated one-level MPC [46]. Despite these results, there is still an interest in looking into two-layer non-optimization-based techniques as they offer some interesting complementary features; for example, a two-layer algorithm may be easier to integrate in some overall software architectures. Non-optimization-based techniques avoid the issue of cost function weights selection, a challenge often underplayed in MPC literature [47]. Avoiding the use of optimization tools allows the practitioners to more intimately understand the behavior of the system and thus simplifying the functional safety analysis that is required in automotive applications.

The objective of this paper is to generate dynamically safe trajectories with a two-layer non-optimization-based approach. We focus on emergency maneuvers intended as an aggressive action to avoid an unexpected or belatedly detected obstacle. Our approach explicitly incorporates dynamic constraints in the path generation step. We first design a path-tracking controller based on the curvature preview idea and then we design a path generation method that, as the cited method above, restricts the search to a class of functions with the added advantage of considering the bandwidth limitations of the path tracking controller in the search. This feature is not considered in other literature methods. Our focus is on lateral control. We recognize that this choice prevents the full exploitation of the vehicle capability; but it comes with the advantage of not requiring an accurate knowledge of the tyre combined slip characteristic that joint lateral and longitudinal controllers require. In summary, our approach has several contributions:

- It is based on an inner path tracking controller whose tracking performance can be expressed in linear terms as a closed-loop bandwidth. To the best of the authors' knowledge, this path tracking controller is new in the literature.
- The path generation accounts for both kinematic and dynamic limitations, generating paths that are trackable by the path tracking system.
- It does not require any recursive optimization. It thus avoids the lengthy and ofter empirical tuning of cost function weights and it does not rely of third parties numerical optimization which may be difficult to use and may behave unpredictably when convergence issues arise.
- It maintains the standard dual (generation - tracking) architecture and it is therefore more easily integrated in some autonomous vehicle software architectures.

In the paper, Section II illustrates the inner path tracking control, called curvature preview control. Section III discusses the generation of the reference path. Section IV discusses the final extensive experimental validation. Section VI draws the final conclusions.

II. CURVATURE PREVIEW CONTROLLER

We design a model-based trajectory tracking controller. In this phase, we assume that a reference trajectory is available.

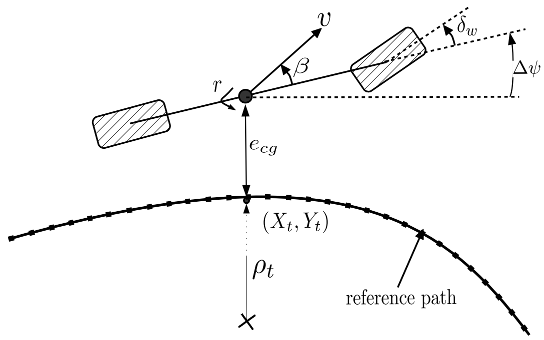

Fig. 2. Lateral error and variable definition.

A. Lateral dynamics model

The vehicle lateral dynamics are described by the classical single-track model represented in Figure 2. Assuming a path radius larger than the wheelbase, slowly-varying longitudinal speed and linear tire characteristics, the model results in the following state-space equations:

$$\begin{aligned}\dot{\beta} &= -\frac{C_f + C_r}{Mv}\beta + \left(\frac{C_r l_r - C_f l_f}{Mv^2} - 1\right) r + \frac{C_f}{Mv}\delta_w \\ \dot{r} &= \frac{C_r l_r - C_f l_f}{J_z}\beta - \frac{C_f l_f^2 + C_r l_r^2}{J_z v} r + \frac{C_f l_f}{J_z}\delta_w\end{aligned} \tag{1}$$

where v is the velocity, r is the yaw rate, β is the side slip angle and δ_w is the steering angle at the wheels, $C_{r,f}$ are the cornering stiffnesses, J_z the yaw inertia, $l_{f,r}$ are the distances between the center of gravity and the axles, M is the mass of the vehicle.

The lateral dynamics model (1) is augmented to describe the evolution of the vehicle position, e_{cg} with respect to the reference path. This is well characterized by introducing the reference path curvature ρ_t. Assuming a small angular difference between the vehicle heading and the track yaw, the rate of change of the lateral error is

$$\dot{e}_{cg} = v(\beta + \Delta\psi) \tag{2}$$

with

$$\Delta\dot{\psi} = r - \rho_t v. \tag{3}$$

Equation (4), in the next page, describes the overall model. In addition to the vehicle lateral dynamics, we also consider the steering actuator dynamics, described by an underdamped second order linear system:

$$G_{steer}(s) = \frac{\omega_{steer}^2}{s^2 + 2\xi_{steer}\omega_{steer}s + \omega_{steer}^2} e^{-T_{act}s} \tag{5}$$

where $\omega_{steer} = 2\pi 4.1$, $\xi_{steer} = 0.203$ and $T_{act} = 0.06$ are experimentally identified [48]. The transfer function describes the dynamics between the desired wheel steering angle δ and the actuated wheel steering angle δ_w.

The system is a Multiple Inputs Multiple Outputs (MIMO) system; the inputs are the controllable steering angle and the known reference curvature (computed at the nearest point on the reference path); while the outputs are the yaw rate and the error at the Center of Gravity (CoG). The vehicle speed v appears as a parameter.

We are referring to a linearized model; we will illustrate that, thanks to the joint closed-loop performance parametrization and feasible reference generation, the controller shows good performance also when performing very aggressive maneuvers, that leads to high side slip angles.

B. Lateral dynamics controller

It is well known that integrating a look ahead (or preview) of the reference path into the controller greatly improves its path tracking accuracy. The tuning of the look-ahead distance is often complex and the best tuning (the tracking error optimizer) depends on the type of trajectory. Starting from this fact, we design a controller based on two loops: a higher bandwidth yaw rate loop and a lower bandwidth position error loop. The preview information is only used in the yaw-rate loop. Figure 3 describes the main constituents of the controller.

Despite being inspired by a nested loop rationale, the two controllers (R_r and $R_{e_{cg}}$) are not decoupled in frequency and cannot be designed independently. An important element of this control strategy is the filter F_ρ that determines the reference yaw rate; its role will be explained below. We propose a sequential design approach, wherein we first focus on the yaw rate controller; and subsequently, based on the closed-loop dynamics, we design the yaw rate reference filter F_ρ and the position control loop.

1) Yaw Rate Control: Let us initially assume $F_\rho = 1$; under this assumption, the reference yaw rate is the product between the path curvature and the longitudinal speed: $r_{ref}(t) = \rho_t(t)v$. The yaw rate controller is designed using standard loop-shaping applied to the linearized Single Input Single Output (SISO) system. In the design, one should consider the following specifications: a) the controller should guarantee a null steady state error in response to a step reference, and b) the closed-loop bandwidth should be as large as possible, considering that the steering actuator limits the achievable bandwidth.

Figure 4 plots the two transfer functions that contribute to the lateral error e_{cg}. The closed-loop transfer function from the reference path curvature ρ_t to the lateral error (considering only $R_r(s)$ in the scheme of Figure 3) and the open-loop transfer function from the reference path curvature ρ_t to the lateral error e_{cg}, through G_ρ: a double integrator with a gain of v^2. In the magnitude plot, the solid lines are the closed-loop transfer functions for different tunings; the dashed line represents the open-loop reference path curvature to error transfer function. Note that as the controller bandwidth increases, the closed-loop transfer functions magnitude get closer to the G_ρ. Calling G_ρ^{CL} the closed-loop transfer function from ρ_t to e_{cg} considering only the $R_r(s)$ loop, the total lateral error dynamics is:

$$e_{cg} = G_\rho^{CL}(s)\rho_t - G_\rho(s)\rho_t. \tag{6}$$

Perfect path tracking is reached when $G_\rho^{CL}(s) = G_\rho(s)$.

$$
\begin{bmatrix} \dot{\beta} \\ \dot{r} \\ \Delta\dot{\psi} \\ \dot{e}_{cg} \end{bmatrix} = \begin{bmatrix} -\frac{C_f+C_r}{Mv} & \frac{C_r l_r - C_f l_f}{Mv^2} - 1 & 0 & 0 \\ \frac{C_r l_r - C_f l_f}{J_z v} & -\frac{C_f l_f^2 + C_r l_r^2}{J_z v} & 0 & 0 \\ 0 & 1 & 0 & 0 \\ v & 0 & v & 0 \end{bmatrix} \begin{bmatrix} \beta \\ r \\ \Delta\psi \\ e_{cg} \end{bmatrix} + \begin{bmatrix} \frac{C_f}{Mv} & 0 \\ \frac{C_f l_f}{J_z} & 0 \\ 0 & -v \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \delta_w \\ \rho_t \end{bmatrix}
$$

$$
\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \beta \\ r \\ \Delta\psi \\ e_{cg} \end{bmatrix} \tag{4}
$$

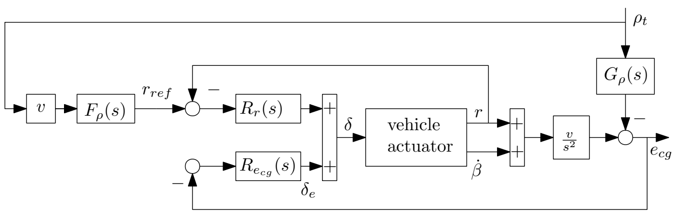

Fig. 3. Curvature Preview Controller architecture.

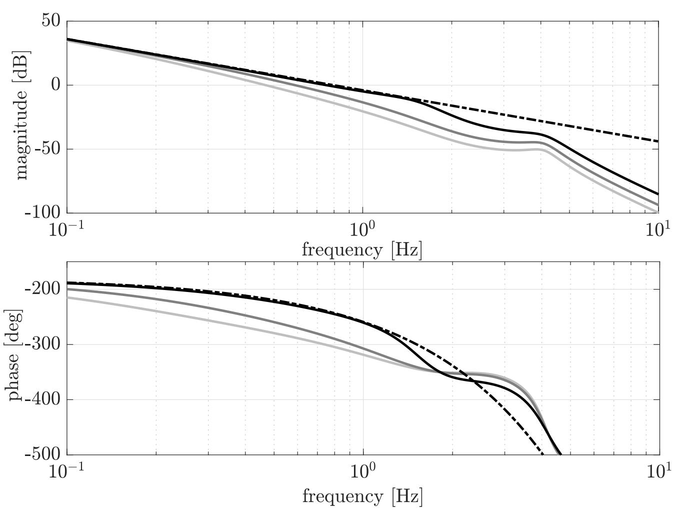

Fig. 4. Closed-loop transfer function from the reference curvature to the lateral error for several yaw rate controller tunings at 90 km/h. The dashed line in the magnitude plot is a double integrator representing the transfer function from the reference curvature to the lateral error. The dotted line in the phase plot represents the phase of a pure delay that approximates the closed-loop phase shift.

If one considers maneuvers that excite the later dynamics up to the frequency where the two transfer functions magnitudes are similar, then one expects to have a perfect tracking of the shape of the reference trajectory but phase shifted. As a matter of fact, note from Figure 4 that, in the same frequencies, the phase of G_ρ^{CL} is well approximated by a pure delay (the dotted curve in the phase diagram). This means that, if the reference trajectory is band limited, the lateral position of the vehicle during the maneuver will be very similar to the reference but time-shifted.

The designs proposed in Figure 4 show that, with the considered setup, it is possible to obtain a $G_\rho^{CL}(s)$ with a magnitude very close to G_ρ up to around 1 Hz, while the phase starts to decrease at frequencies lower than 1 Hz. In the following, we use road preview to improve the path tracking performance.

2) Yaw Rate Reference Filtering: The filter $F_\rho(s)$ exploits the availability of a preview of the trajectory to compensate the phase distortion introduced by the yaw-rate closed-loop control. In principle, by inverting the closed-loop dynamics, one could achieve a perfect tracking of the reference. This solution is not implementable because the inversion yields an improper transfer function. In practice, we are interested in having a good match only in the frequency range that the maneuvers excite. If the closed-loop bandwidth of the yaw rate controller is sufficiently large, the complete closed-loop transfer function from reference curvature to lateral error can be approximated by:

$$G_\rho^{CL}(s) \approx \frac{v^2}{s^2} e^{-sT_r} F_\rho(s). \tag{7}$$

By choosing

$$F_\rho(s) = e^{sT_r}, \tag{8}$$

the tracking error becomes 0 in the excited frequency range (see (6)). This filter is a simple time shift: the reference yaw rate is computed from the future curvature. Figure 5 shows the comparison between the proposed approximation ($T_r = 0.25s$) and the actual transfer function from reference curvature to tracking error for several longitudinal speeds. From the figure, the following remarks are due:

- The delay approximation yields a good fitting up to approximately 0.8 Hz, which provides a quantitative indication for which kind of maneuvers can be executed.
- The choice of not introducing a velocity scheduling in the yaw rate controller does not have a large effect on the closed loop behavior.

3) Lateral Error Controller: The yaw rate controller is designed to accurately track high dynamics maneuvers; however being based on yaw rate, it is subject to slow trajectory drifts. The presence of uncertainties in the model and the approximations made in the construction of the filter $F_\rho(s)$ cause the vehicle to drift with respect to the reference path. From the block diagram, one can write the residual error after the yaw rate controller is closed as:

$$E_{cg}(s) = (F_\rho(s) G_\rho^{CL}(s) - G_\rho(s)) P_t(s), \tag{9}$$

where E_{cg} and P_t are the Laplace transforms of e_{cg} and ρ_t. The second loop, the lateral error controller, corrects for these drifts. The lateral error controller computes δ_e based on the

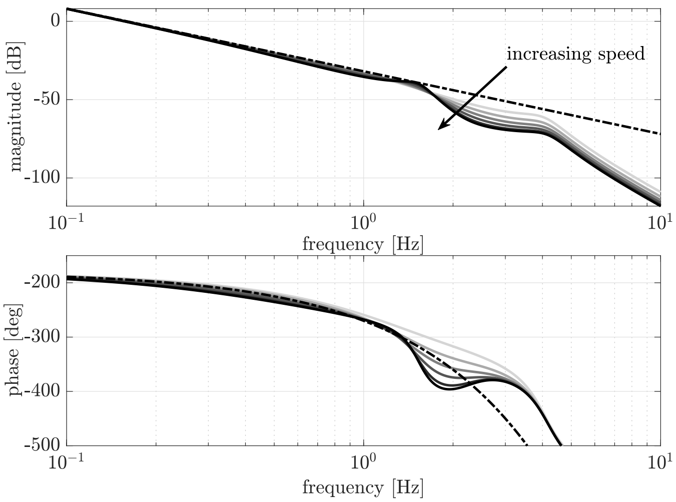

Fig. 5. Longitudinal speed sensitivity ($v \in [50, 130]$ km/h) of the transfer function between the reference curvature and the lateral error. Note that the transfer functions are normalized by v^2 to allow for an easier comparison. The dashed line in the magnitude plot is a double integrator representing the transfer function from the reference curvature to the lateral error. The dotted line in the phase plot represents the phase of a pure delay that approximates the closed-loop phase shift.

lateral error and adds it to the steering angle generated by the yaw rate controller. We design the lateral error controller based on the closed-loop lateral error dynamics shown in (9) obtaining

$$R_{e_{cg}}(s) = \frac{K_e}{s} \frac{(s + 2\pi f_z)^2}{(s + 2\pi f_p)^2}, \tag{10}$$

We tuned the controller (f_z and f_p) around the nominal speed of 90 km/h to obtain a crossover frequency of 0.3 Hz and a phase margin of 60 deg. Starting from this nominal design, we introduced the following velocity dependent gain scheduling

$$K_e(v) = K_{e,90} \frac{90/3.6}{v} \tag{11}$$

where $K_{e,90}$ is the gain at 90 km/h.

III. DYNAMIC PATH GENERATION

The curvature preview controller has several advantages: it provides a quantitative way of tuning the preview (or look ahead parameter) and a way of quantifying the path tracking performance in linear terms. In particular, when used in the context of tracking a pre-computed path, the ability of the vehicle to track a given path is expressed in terms of closed-loop bandwidth. This latter property is at the basis of the obstacle avoidance trajectory generation. Here, we will assume to have adequate sensors capable of detecting the position, dimension and velocity of the obstacles. For clarity's sake, we will first explain the method for obstacle's driving at constant speed on a straight road. Once we outline the method, we will briefly explain how to remove these hypotheses.

The detection of an obstacle triggers a three-step procedure:

1) Collision check phase. It checks whether the current path crosses the collision area related to an obstacle.

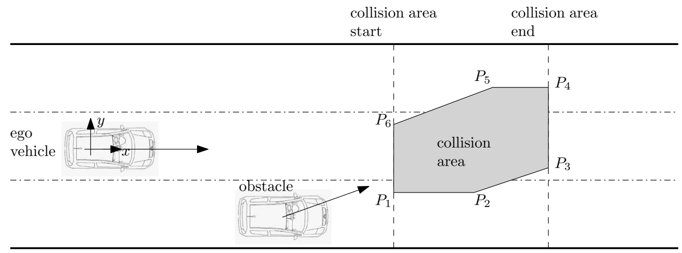

Fig. 6. Collision area computed exploiting the obstacle position and speed at detection.

2) Avoidance feasibility check phase. It determines whether it is possible to avoid the obstacle. Two trajectories are calculated, maximum shift to the left and to the right according to the roadway limits and lateral dynamics.
3) Maneuver selection and adjustment. The most aggressive maneuvers computed at the second step is not usually necessary; this phase aims at reducing the maneuver dynamic excitation to avoid discomfort, if possible.

A. Collision Check Phase

The obstacle and ego vehicle information are used to compute a region, called collision area, where a crash is possible. Figure 6 shows the collision area in case of another vehicle merging from the right. The collision area is computed when the longitudinal velocity of the ego car is higher than that of the obstacle. The area starts at the longitudinal distance from the ego vehicle that corresponds to the point where the front of the ego vehicle and the rear of the obstacle will have the same position, should the velocities not change. This value is

$$s_{c,start} = v_{x,ego} \frac{s_{obs} - L_{obs}}{v_{x,ego} - v_{obs}} \tag{12}$$

where s_{obs} is the distance between the front of the obstacle and the front of the ego vehicle and L_{obs} is the length of the osbstacle. $v_{x,ego}$ and $v_{x,obs}$ are the longitudinal speeds of the ego vehicle and the obstacle respectively. The end of the collision area considers the intersection between the rear of the controlled vehicle and the front of the obstacle:

$$s_{c,end} = v_{x,ego} \frac{s_{obs} + L_{ego}}{v_{x,ego} - v_{x,obs}} - L_{ego}. \tag{13}$$

where L_{ego} is the length of the ego vehicle. The collision area is the space that will be occupied by the obstacle from $s_{c,start}$ to $s_{c,end}$. If the obstacle velocity points to the left (with respect to the ego vehicle), the collision area polygon has the following vertexes:

$$
\begin{aligned}
P_1 &= \left(s_{c,start}\ ,\ y_{obs,start} - \frac{W_{obs}}{2}\right)\\
P_2 &= \left(s_{c,start} + L_{obs}\ ,\ y_{obs,start} - \frac{W_{obs}}{2}\right)\\
P_3 &= \left(s_{c,end}\ ,\ y_{obs,end} - \frac{W_{obs}}{2}\right)\\
P_4 &= \left(s_{c,end}\ ,\ y_{obs,end} + \frac{W_{obs}}{2}\right)\\
P_5 &= \left(s_{c,end} - L_{obs}\ ,\ y_{obs,end} + \frac{W_{obs}}{2}\right)\\
P_6 &= \left(s_{c,start}\ ,\ y_{obs,start} + \frac{W_{obs}}{2}\right).
\end{aligned}
\tag{14}
$$

A similar set of equations arises when the obstacle's velocity points to its right. In the previous equations, W_{obs} denotes the width of the obstacle, while $y_{obs,start}$ and $y_{obs,end}$ are the lateral position of the obstacle at the collision region start and end, computed by plugging the obstacle lateral speed $v_{y,obs}$, the ego vehicle lateral position and the widths of the ego and obstacle in (12) and (13).

In order to keep the computation efficient, we assume that the heading of the obstacle does not change its bounding box. Therefore, in the definition of W_{obs}, one may want to consider the effect of the relative heading between the two objects. The collision check phase computes the collision region every time an obstacle is detected, and continuously updates it while the obstacle is tracked by the onboard sensors. [2]

B. Avoidance Feasibility Check and Extreme Trajectories Computation

In this phase, we compute the trajectories that guarantee the most rapid lateral movement to the street limits while considering the limitations due to the path tracking controller. The outputs are two feasible trajectories (left and right) that bring the vehicle to the limit of the roadway. We will refer to trajectories that, after the obstacle is avoided, return the vehicle to the same heading the vehicle had before commencing the maneuver.

If only the maximal lateral acceleration constraint (which mainly depends on the tyre-road grip limit) is considered, the longitudinal constant velocity path that minimizes the longitudinal distance needed to move the vehicle of a given distance laterally is composed by two arcs of circumference of minimum cornering radius (*i.e.* maximum curvature - maximum lateral acceleration). This maneuver is not feasible in practice. The controller is not capable of tracking the resulting squared curvature. To guarantee safety, the trajectory needs to account for the dynamic limits of the vehicle, actuators and path tracking controller. The closed-loop characterization of the curvature preview controller is a convenient way to quantify these constraints.

We need to consider two features of the curvature preview controller: it has a finite closed-loop bandwidth and it needs a preview. We consider the closed-loop bandwidth limitation by generating trajectories built with curvature sinusoids at a frequency lower than the controller's bandwidth. Figure 7 shows two examples of path planning using a sinusoidal curvature. The plots in subfigure (a) represent a case of pure sinusoidal trajectory with maximum curvature amplitude; the sinusoid period is set at 2 s in this case, meaning that most excitation will be below 0.5 Hz. With this configuration, the obtained lateral shift is about 3.7 m. If the road limit is

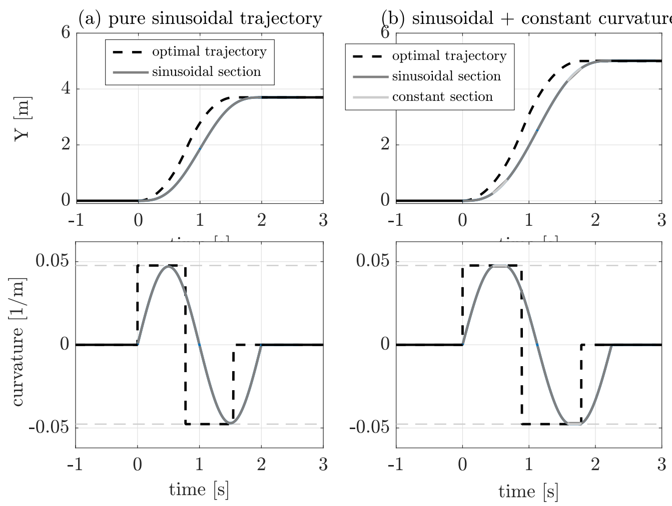

Fig. 7. (a) Pure sinusoidal trajectory with amplitude corresponding to the maximum curvature. (b) Sinusoidal trajectory with constant curvature sections. The dashed curves represent the optimal trajectory that considers only static limits.

closer than 3.7 m, the sinusoid amplitude must be lowered. In principle, the sinusoid period could also be reduced, but this is not advisable because it would increase the dynamic excitation. If instead the road limit is farther than 3.7 m, constant curvature sections at maximum curvature are added as shown in Figure 7 (b). The figure also shows the trajectory arising using only the static constraints (*i.e.* the piecewise constant curvature). The constant curvature segments introduce higher frequencies in the spectrum of the reference curvature; however most of the power of the signal will still be in the target bandwidth. Figure 8 quantitatively shows the frequency spectra for different reference bandwidths.

Based on the above exemplifying cases, we define the generic structure of the path as in Figure 9. The curvature of the

[2] We note that the collision check assumes a perfect knowledge of the obstacle position, and the continuous collision computation may cause chattering in the trajectory generation; if, for a given implementation of the perception module, this were to happen, one could define a larger, conservative, safety area around the obstacle that could be better filtered and stabilized.

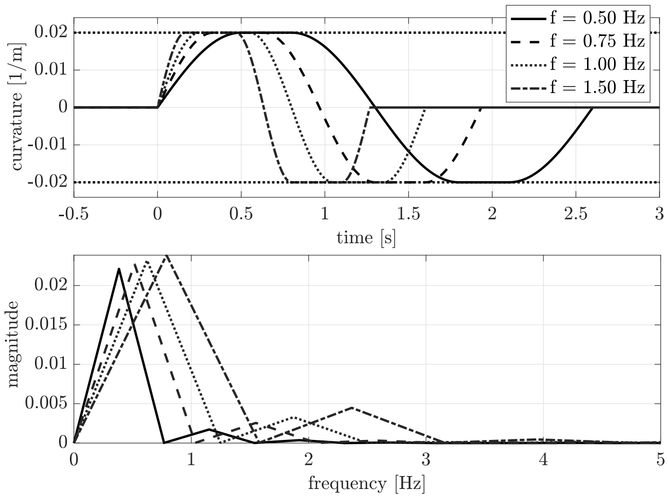

Fig. 8. Example trajectory curvatures and their frequency spectra for different target bandwidths.

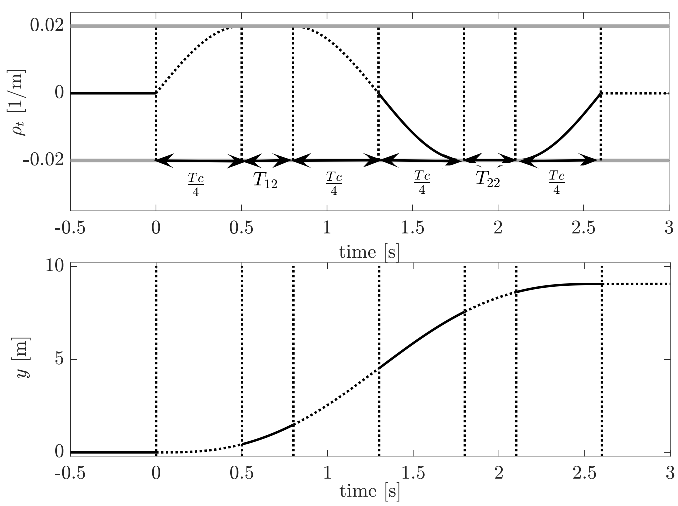

Fig. 9. Generic path structure.

maneuver is described by the following expression:

$$\rho_t(t) = \begin{cases} \rho_1 \sin\left(\frac{2\pi}{T_c} t\right) & \text{if } t \in T_1 \\ \rho_1 & \text{if } t \in T_2 \\ \rho_1 \cos\left(\frac{2\pi}{T_c}\left(t - \frac{T_c}{4} - T_{12}\right)\right) & \text{if } t \in T_3 \\ -\rho_2 \sin\left(\frac{2\pi}{T_c}\left(t - \frac{T_c}{2} - T_{12}\right)\right) & \text{if } t \in T_4 \\ \rho_2 & \text{if } t \in T_5 \\ -\rho_2 \cos\left(\frac{2\pi}{T_c}\left(t - \frac{3}{4}T_c - T_{12} - T_{22}\right)\right) & \text{if } t \in T_6. \end{cases} \tag{15}$$

In the expression, T_c is the sinusoid period (representing the bandwidth of the lateral controller), and

$$\begin{aligned} T_1 &= \left[0, \frac{T_c}{4}\right] \\ T_2 &= \left[\frac{T_c}{4}, \frac{T_c}{4} + T_{12}\right] \\ T_3 &= \left[\frac{T_c}{4} + T_{12}, \frac{T_c}{2} + T_{12}\right] \\ T_4 &= \left[\frac{T_c}{2} + T_{12}, \frac{3T_c}{4} + T_{12}\right] \\ T_5 &= \left[\frac{3T_c}{4} + T_{12}, \frac{3}{4}T_c + T_{12} + T_{22}\right] \\ T_6 &= \left[\frac{3T_c}{4} + T_{12} + T_{22}, T_c + T_{12} + T_{22}\right] \end{aligned}$$

with T_{12} and T_{22} the duration of the constant curvature sections, ρ_1 is the amplitude of curvature of the first wave, while ρ_2 is the one of the second wave.

The final heading and the lateral shift result from integrating the previous expression:

$$\begin{aligned} \psi(t_{end}) &= \int_0^{t_{end}} v\rho(t)dt \\ y(t_{end}) &= \int_0^{t_{end}} v\sin(\psi(t))dt \approx \int_0^{t_{end}} v\psi(t)dt \end{aligned} \tag{16}$$

where $t_{end} = T_c + T_{12} + T_{22}$ is the duration of the maneuver. Equation (15) presents 4 unknowns: ρ_1, ρ_2, T_{12} and T_{22}. Their values are constrained by the fact that, at the end of the maneuver, the heading is equal to the initial one and the lateral shift is equal to the available road:

$$\begin{aligned} \psi(t_{end}) &= v\left(\rho_1\frac{T_c}{\pi} + \rho_1 T_{12} - \rho_2\frac{T_c}{\pi} - \rho_2 T_{22}\right) = 0 \\ y(t_{end}) &= v^2\left(\frac{T_c^2}{4\pi}(\rho_1 + \rho_2) + \frac{1}{2}(\rho_1 T_{12}^2 + \rho_2 T_{22}^2)\right) + \\ &\quad + v^2\left(T_c\left(\frac{1}{4} + \frac{1}{2\pi}\right)(\rho_1 T_{12} + \rho_2 T_{22})\right) = y_{lim} \end{aligned}$$

In solving the above system, one should consider three cases:

1) Both constant curvature sections are needed to achieve the desired lateral shift: $T_{12} > 0$ and $T_{22} > 0$. In this case, the amplitudes assume their maximum values: $\rho_1 = \rho_{1,max}$ and $\rho_2 = \rho_{2,max}$, and only T_{12} and T_{22} are unknown.
2) Only one half wave reaches the maximum value, the other half wave is sinusoidal. This case is possible only when the left and right curvature limits are different, *i.e.* when the maneuver commences while cornering. If $\rho_{1,max} > \rho_{2,max}$ then $T_{12} = 0$ and $\rho_2 = \rho_{2,max}$, therefore only ρ_1 and T_{22} are unknown. On the other hand, if $\rho_{2,max} > \rho_{1,max}$, $T_{22} = 0$ and $\rho_1 = \rho_{1,max}$ and ρ_2 and T_{12} are unknown.
3) Both half waves do not reach the maximum curvature. Since the limits are not reached, the constant curvature sections are not needed. The equation on the heading implies that the two waves have the same amplitude

($\rho_1 = \rho_2 = \bar{\rho}$), while the equation on the lateral shift is used to determine its value.

A few considerations on the above method are worth pointing out. First and foremost, the solution of the above equations exists in closed form. Distinguishing between the three cases is a matter of checking the signs of the solution. First the algorithm solves the equations with unknown (T_{12}, T_{22}) and setting $\rho_1 = \rho_2 = \rho_{max}$. If these conditions do not yield positive solutions, the algorithm moves to option 2) using either (ρ_1, T_{22}) as unknowns or (ρ_2, T_{12}). If also in this case unfeasible solutions are found, the algorithm considers case 3) solving the equation with unknown $\rho_1 = \rho_2 = \rho$ and setting $T_{12} = T_{22} = 0$. The approach does not require the iterative solution of a nonlinear optimization problem.

Although we illustrate the method considering a lane change on a straight road, the method can be also extended to cornering situations by including the current initial conditions in the equation and correcting the final conditions according to the centerline curvature. This means that the method deals also with obstacles that change their velocity or course while the evasive maneuver is being executed: a change of obstacle velocity (longitudinal or lateral) triggers the re-computation of the trajectory around the new initial curvature.

The second feature of the curvature preview controller is that it needs a preview. The algorithm curvature reference of ρ_t is entirely computed at the moment when the obstacle is detected; to avoid discontinuities in the curvature reference, only trajectories that start vT_r meters ahead are considered.

This phase generates two possible maneuvers: quickest dynamically feasible left and right shifts that exploit the maximum available space (the roadway limits).

C. Maneuver selection and adjustment

This phase chooses the best type of maneuver among the two computed in the previous phase and adjusts the maneuver to avoid unwarranted excitations. The phase considers two steps. Firstly, we reduce the lateral shift, then, if possible, we increase the sinusoid period as much as possible. Algorithm 1 summarizes the approach:

Algorithm 1: Maneuver adjustment algorithm.

Result: ρ_t - reference avoidance trajectory
$y_{lim} = y_{roadway}$; $T_c = T_{min}$; $n_{iter} = 0$;
ρ_t = null;
while $n_{iter} < n_{max}$ **do**
  ρ_{tmp} = compute_trajectory(y_{lim},T_c);
  $n_{iter} + +$;
  if ρ_t *is collision free* **then**
    $\rho_t = \rho_{tmp}$;
    if $y_{lim} - y_{step} > y_{sh,right}$ **then**
      $y_{lim} = y_{lim} - y_{step}$;
    else
      if $T_c + T_{step} < \frac{s_{c,start}}{v_{ego}}$ **then**
        $T_c = T_c + T_{step}$
      else
        break
      end
    end
  else
    break
  end
end
if ρ_t = *null* **then**
  obstacle cannot be avoided. Brake to reduce impact.
else
  return ρ_t
end

For the first step, the coordinates of the vertexes of the collision area are needed. Consider an obstacle traversing the road from right to left (as shown in Figure 6). The ego vehicle can avoid the collision region by either steering to the right or to the left. In the case of a rightward evasive maneuver, the needed lateral shift depends on the y-coordinate of P_1 and is:

$$y_{sh,right} = y_{obs,start} - W_{obs}/2 - W_{ego}/2 - y_{safe} \quad (17)$$

where y_{safe} represents a safety margin. In the same way, if the ego vehicle steers left, the most critical point is the y-coordinate of P_5:

$$y_{sh,left} = y_{obs,end} + W_{obs}/2 + W_{ego}/2 + y_{safe}. \quad (18)$$

If the obstacle is moving to the right, the vertexes P_2 and P_6 are the most critical ones.

If it possible to reach $y_{sh,right}$ or $y_{sh,left}$, the algorithm tries to increase the sinusoid period to further limit the excitation. This second search stops when the longest period ($\frac{s_{c,start}}{v_{ego}}$) is reached. In the algorithm, n_{max} is the maximum number of allowed attempts, T_{min} is the minimum period compatible with the bandwidth of the lateral error controller. y_{step} and T_{step} are the search algorithm step sizes. When both the right and left handed maneuvers are available, the algorithm chooses the one with the lowest peak curvature.

Note that the algorithm that reduces the maneuver excitation is iterative. This is the only iterative step of the algorithm, and it does not pose any convergence issue. This is a refinement step that can be halted at any time returning a collision-free maneuver.

D. Additional Remarks

The above strategy has been described under a few simplifying hypotheses, that we now better discuss.

1) Road friction: The trajectory that a vehicle can track (especially in emergency evasive maneuvers) heavily depends on the available road friction. The algorithm summarizes the road friction information in two parameters T_{min} and ρ_{max}.

The effect of the friction on ρ_{max} is rather trivial. As discussed in Section III-B, the maximum curvature is statically related to the maximum acceleration (and thus friction) and longitudinal speed. This implies that a friction estimation is needed in order to plan a trajectory. A number of approaches are available in the literature [49].

The effect of the friction on T_{min} is subtler. In Section II, we designed a model based lateral error controller. Figure 10 shows a sensitivity analysis of the closed loop transfer function reference yaw rate to e_{cg} to the friction parameter μ. As the friction decreases, the effective bandwidth of the lateral

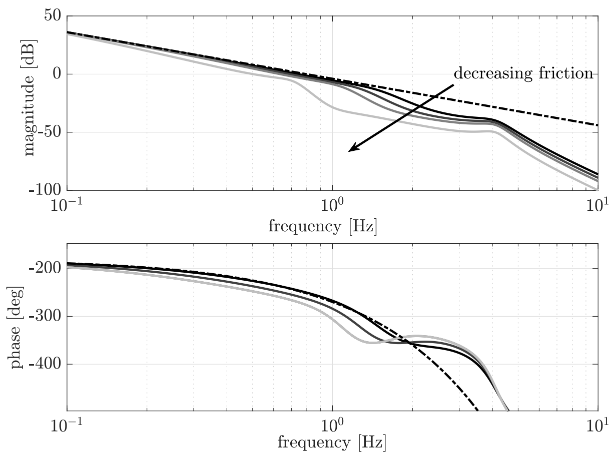

Fig. 10. Sensitivity of the closed loop yaw rate dynamics to $\mu \in [0.2, 1]$.

controller decreases. Given this information, one can consider two paths: the first idea is to use a fixed-structure controller and robustly choose T_{min}. Alternatively one could design a gain-scheduled lateral error controller. In the remainder of the paper, we opted to consider a robust T_{min}.

2) Ego vehicle constant velocity: The algorithm computes the trajectory based on a constant velocity assumption. Depending on the velocity, distance of the obstacle and available friction another strategies could include braking and then steering or braking while steering.

Braking while steering is not advisable. While the literature shows that one could design controllers capable of managing these challenging conditions (see for example [50]), these controllers require an accurate knowledge of the tyre characteristics and of the vehicle state. Alternatively, one could consider to brake and then execute the obstacle avoidance maneuver. Our analysis, omitted for brevity's sake, shows that sequential braking and shifting is never preferred. Direct shift is always preferable if the initial speed is higher than a given speed (friction, obstacle position and T_{min} dependent), otherwise a full stop is recommended.

IV. VALIDATION

Figure 11 depicts our experimental platform. The vehicle and obstacle localization tasks are outside the scope of the paper. The vehicle, a production Dodge Dart, is retrofitted with steer-by-wire, brake-by-wire and integrated RTK-INS accurate localization and state-estimation.

The controller validation considers two aspects: the curvature preview and the dynamic path generation. The first validation will be entirely based on experimental results on prerecorded paths; whereas to better underline some aspects of the approach, simulations will also be used in the second validation.

Fig. 11. The experimental setup.

A. Experimental Curvature Preview Validation

The following experiments compare the Curvature Preview Controller (CPC) against a state-of-the art LPV controller [51] with a look ahead distance of 12 m. The validation is based on the following tests:

- Wide bend at 70, 80, 85 and 90 km/h
- Double Lane Change at 50, 60, 70, 80 and 90 km/h. In this phase on the analysis, the lane change reference path has been manually pre-recorded by a driver performing the most aggressive maneuver he could manage.

Figure 12 shows the results of the wide bend. One can note

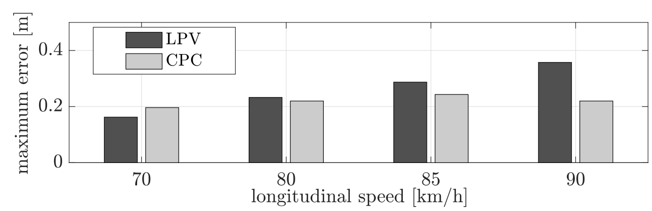

Fig. 12. Maximum tracking error on a wide bend.

that the CPC yields a maximum error smaller than 0.25 m in all conditions, matching the best performance of the LPV. The time domain analysis of Figure 13 better exemplifies some features of the controller. The two controllers approach the corner in different ways: the LPV controller has always a small error directed to the outside of the corner. The CPC yields an almost zero error when the corner curvature increases (between 250 and 400 m). This proves the quality of this control scheme in handling variation of curvature, even at steady state lateral accelerations as high as 0.9 g.

Figure 13 also plots the results of the simulated vehicle during the same maneuver with the CPC. The behavior of the simulator is rather similar to that of the experiments, thus validating the simulation setup that will be employed in in the path validation section. The oscillations in the experiments are due to steering non-idealities, particularly backlash.

The double lane change scenario is more demanding from the dynamic point of view. In this scenario, the curvature varies rapidly. We estimated that the double lane change excites

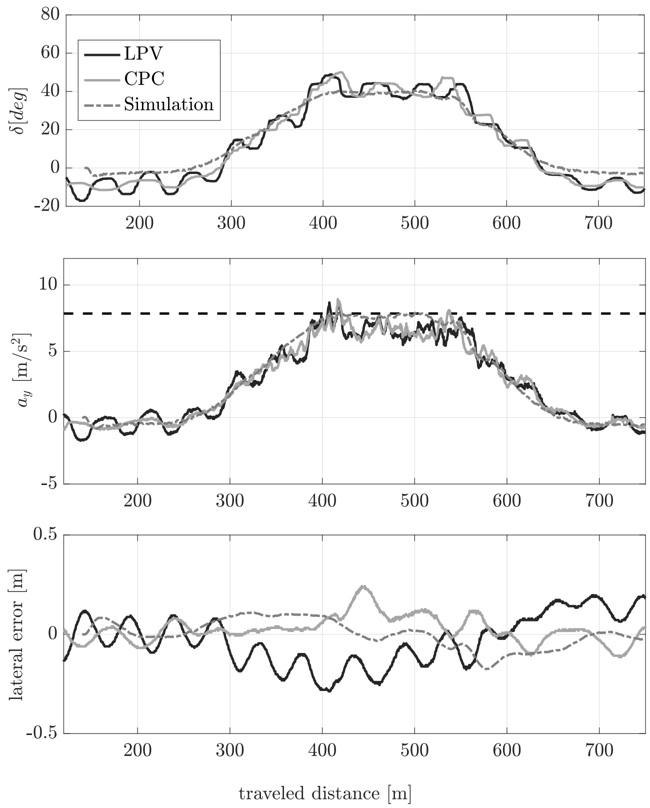

Fig. 13. Steering angle, lateral acceleration and tracking error on a wide bend at 85 km/h.

frequencies up to 1.2 Hz. Figure 14 shows the maximum error between 50 and 90 km/h. The CPC controller outperforms the

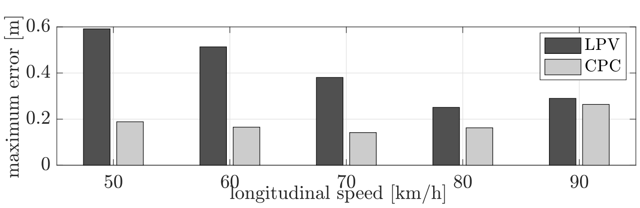

Fig. 14. Maximum error during the DLC maneuver.

LPV at all speeds but at 90 km/h, where the LPV controller and the CPC perform similarly. This is due to the fact that the LPV controller employs a constant look ahead distance and consequently, the error shows a dependency on the speed. The optimal look ahead distance depends on the actual maneuvers and tuning it requires a compromise between different scenarios. Nevertheless, the CPC matches the best LPV performance without the need of introducing an additional scheduling. Figure 15 better visualizes the behavior of the controller at 90 km/h and validates the simulator showing the simulation results with the CPC. One should note that

- the LPV and CPC controllers yield a maximum error lower than 0.25 m.
- At 90 km/h, the lateral acceleration reaches 0.8 g, while the side slip angle has a peak of 2.5°. In these conditions, the vehicle behavior is nonlinear. Despite being designed on a linear model, the CPC yields excellent tracking also when the vehicle is outside its linearity limits.

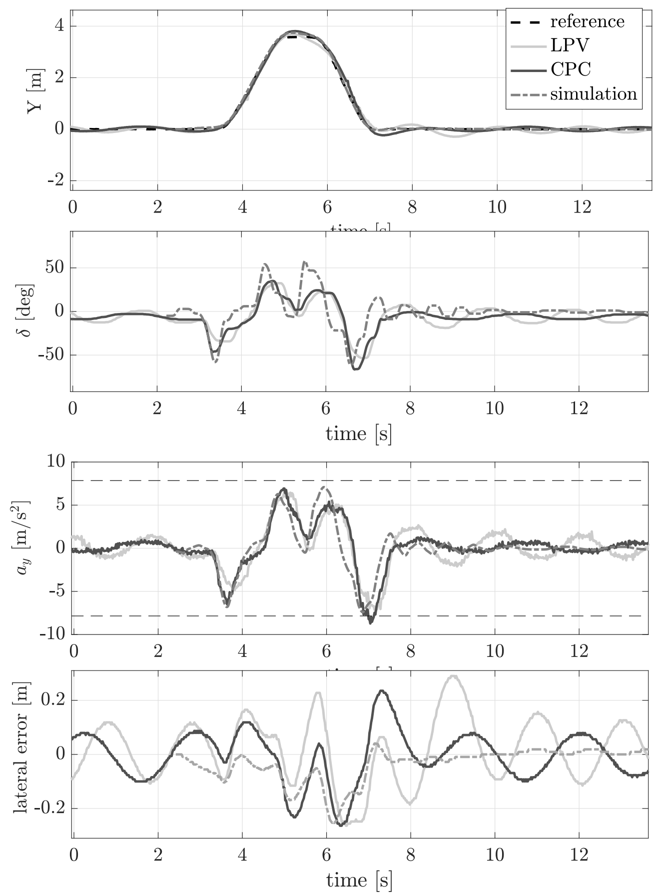

Fig. 15. Time histories for the DLC maneuver at 90 km/h.

The experiments show good tracking performance on both low dynamics and high dynamics paths. The CPC exhibits the same performance of the LPV controller with a simpler control architecture and without the effort of tuning the look ahead distance by trial-and-error; while providing an intuitive way of parametrizing its closed-loop performance. These properties are an ideal match for our path generation algorithm.

B. Trajectory Generation Validation

In validating the trajectory generation algorithm, we will focus on mainly two aspects: showing the computed trajectories in two scenarios and underlining the importance of generating a trajectory that is dynamically feasible for the controller.

We performed these analysis on a multi-body (run in VI-Grade Car Real Time) model in order to more easily test the approach at the limit of safety and stability. Refer to Figures 13 and 15 for a comparison between the behavior of the real system against the simulated vehicle.

We initially consider two scenarios of evasive maneuvers with the nominal vehicle. In the following scenarios, the controlled vehicle is initially driving straight at 90 km/h, the friction coefficient is 0.9, the safety lateral distance is 0.3 m and the minimum sinusoid period is set to 1.25 s equivalent to a bandwidth of 0.8 Hz.

The first scenario considers a static obstacle detected at 25 m. Figure 16 plots a stylized representation of the ego vehicle (black and gray shades) along with its trajectory and the obstacle (red). In this case, the ego vehicle needs to shift to the left limit of the roadway and the sinusoid period is at its minimum: a smaller shift would not have been feasible for obstacle avoidance. The curvature of the planned trajectory is not purely sinusoidal, but constant curvature sections are added. Thanks to the curvature preview controller and the generation of a dynamically feasible trajectory, the maximum path tracking error is 12 cm. Figure 17 shows the time domain behavior of the yaw rate (along with its previewed reference). As expected, the yaw rate tracks the previewed reference with a delay, thus effectively tracking the desired path. The peak lateral acceleration during the maneuver is 0.85 g signifying that the trajectory takes advantage of the full grip potential.

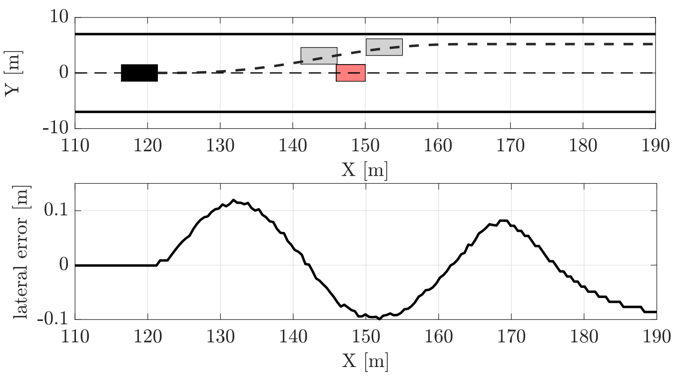

Fig. 16. Scenario 1: static obstacle appearing in front of the vehicle at a very short distance.

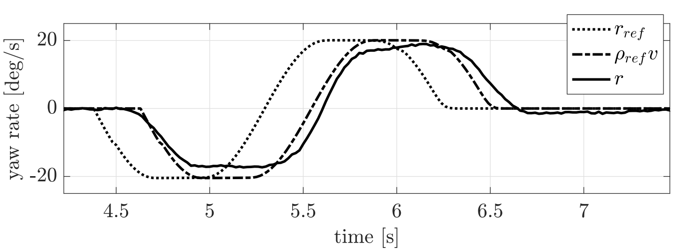

Fig. 17. Scenario 1: reference (preview) yaw rate, actual yaw rate and needed yaw rate to perform the desired obstacle avoidance $\rho_{ref}v$.

In the second scenario, the obstacle is a car merging from the right at 50 km/h that is detected at 40 m; when the obstacle reaches the 53 m mark, it changes its lateral velocity. Initially, the trajectory planner, using the instantaneous information on the obstacle trajectory, decides that the best course of action is to steer left (first panel of Figure 18). Later in the maneuver, the obstacle steers more to the left making the original trajectory for the ego vehicle unfeasible. At that point, the trajectory re-computation is triggered, and a new evasive trajectory is generated (second panel of Figure 18). This re-computation approach requires a well-working object tracking module. Discontinuities in the obstacle tracking could lead to a *flickering* reference path. This issue is better addressed at the perception layer level. Figure 18 also shows the collision patches, the complete generated reference trajectory and the tracking error. Also in this case, the tracking error is below 12 cm. Figure 19 shows the yaw rate traces. Note that, thanks to the F_ρ a causal filter, the actual yaw rate tracks the expected and desired yaw rate.

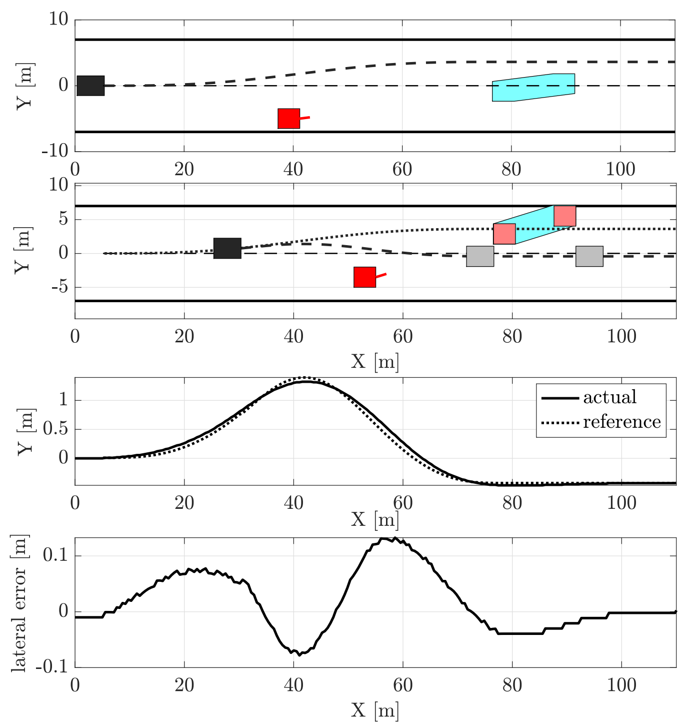

Fig. 18. Scenario 2: merging of an external vehicle from the right with a change of trajectory.

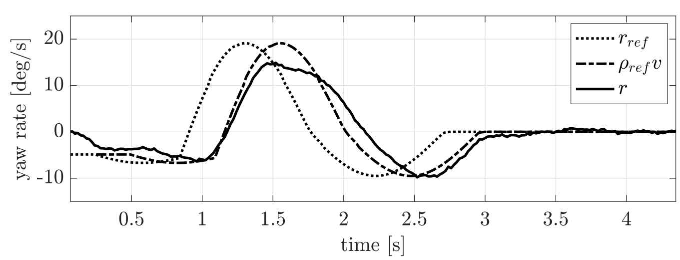

Fig. 19. Scenario 2: merging of an external vehicle from the right with a change of trajectory.

Finally, in order to better visualize the advantages of considering the dynamic constraints, we consider a vehicle equipped with a lower grade actuator. We consider an actuation delay $T_{act} = 0.160$ s; under this limitation, the maximum achievable closed-loop bandwidth of the yaw-rate controller is 0.2 Hz and the trajectory planner considers that. Figure 20 shows two obstacle avoidance maneuvers: in one case, the reference trajectory is computed using the dynamic constraint; in the other case, a piece-wise constant reference is used that achieves the same lateral displacement. Both cases use the same calibration of the curvature preview controller. Although

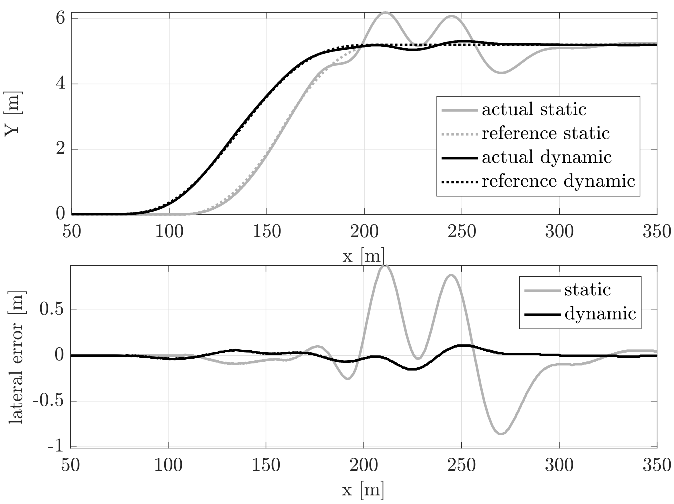

Fig. 20. Comparison between a curvature reference generated considering only the static constraints (maximum curvature) and the curvature reference generated considering dynamic constrains according to the proposed algorithm.

the piece-wise constant curvature reference seems to allow one to achieve the required lateral shift in a shorter distance; when excited at such high frequency, the lateral error controller cannot accurately track the reference. The proposed algorithm, on the other hand, generates a trajectory that is compliant with the dynamic limitations of the controller. In other words, generating a trajectory that does not consider the dynamic constraint could lead to a crash even if the computed trajectory is feasible at computation time.

V. CONCLUSIONS

This paper presents a non-optimization-based path planning and tracking for evasive maneuvers. We adopted the classical two-layer approach where a path planner generates a reference trajectory that is then tracked by a path tracking controller.

At the path tracking level, the paper introduces the curvature preview controller, which is a model-based controller that implements a faster yaw rate controller and a slower trajectory controller. The model-based design allows for a quantitative description of the closed-loop dynamics properties of the controlled vehicle. These closed-loop features offer a quantitative method to design the curvature preview.

The curvature preview controller is the basis to design an evasive maneuver path generation. The path generator considers both the static constraints and the dynamic constraints that depend on the performance of the lower level controller. In this way, it is possible to integrate the specifications of the lower level controller without the need of a one-level architecture.

The two layers interacts bidirectionally: the low level closed-loop performance parametrization informs the generation of the path. The feasible path improves the robustness of the low-level controller; which, despite being designed on a linearized model, performs satisfactorily because the reference path does not exited the nonlinearities in the system.

The proposed framework offers an effective way of dealing with evasive maneuvers without breaking the two-layer classical architecture of autonomous car systems.

VI. ACKNOWLEDGMENTS

The authors are grateful to Magneti Marelli S.p.A. and Michele Giorelli for providing the vehicle employed in the experimental campaign.

REFERENCES

[1] S. Standard, "J3016," *Taxonomy and Definitions for Terms Related to On-Road Motor Vehicle Automated Driving Systems*, vol. 4, pp. 593–598, 2014.

[2] P. Varaiya, "Smart cars on smart roads: problems of control," *Automatic Control, IEEE Transactions on*, vol. 38, no. 2, pp. 195–207, 1993.

[3] J. Monteil, M. Bouroche, and D. J. Leith, "L2 and linf stability analysis of heterogeneous traffic with application to parameter optimization for the control of automated vehicles," *IEEE Transactions on Control Systems Technology*, 2018.

[4] K. Huang, X. Yang, Y. Lu, C. C. Mi, and P. Kondlapudi, "Ecological driving system for connected/automated vehicles using a two-stage control hierarchy," *IEEE Transactions on Intelligent Transportation Systems*, 2018.

[5] J. Han, A. Sciarretta, L. L. Ojeda, G. De Nunzio, and L. Thibault, "Safe- and eco-driving control for connected and automated electric vehicles using analytical state-constrained optimal solution," *IEEE Transactions on Intelligent Vehicles*, vol. 3, no. 2, pp. 163–172, 2018.

[6] K. L. Talvala, K. Kritayakirana, and J. C. Gerdes, "Pushing the limits: From lanekeeping to autonomous racing," *Annual Reviews in Control*, vol. 35, no. 1, pp. 137–148, 2011.

[7] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann *et al.*, "Stanley: The robot that won the darpa grand challenge," *Journal of field Robotics*, vol. 23, no. 9, pp. 661–692, 2006.

[8] J. Levinson, J. Askeland, J. Becker, J. Dolson, D. Held, S. Kammel, J. Z. Kolter, D. Langer, O. Pink, V. Pratt *et al.*, "Towards fully autonomous driving: Systems and algorithms," in *Intelligent Vehicles Symposium (IV), 2011 IEEE*. IEEE, 2011, pp. 163–168.

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

[10] 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, 2016.

[11] S. E. Shladover, C. A. Desoer, J. K. Hedrick, M. Tomizuka, J. Walrand, W.-B. Zhang, D. H. McMahon, H. Peng, S. Sheikholeslam, and N. McKeown, "Automated vehicle control developments in the path program," *IEEE Transactions on vehicular technology*, vol. 40, no. 1, pp. 114–130, 1991.

[12] K. Kritayakirana and J. C. Gerdes, "Autonomous vehicle control at the limits of handling," *International Journal of Vehicle Autonomous Systems*, vol. 10, no. 4, pp. 271–296, 2012.

[13] G. Williams, P. Drews, B. Goldfain, J. M. Rehg, and E. A. Theodorou, "Aggressive driving with model predictive path integral control," in *2016 IEEE International Conference on Robotics and Automation (ICRA),*. IEEE, 2016, pp. 1433–1440.

[14] T. Stahl, A. Wischnewski, J. Betz, and M. Lienkamp, "Multilayer graph-based trajectory planning for race vehicles in dynamic scenarios," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019, pp. 3149–3154.

[15] A. Heilmeier, A. Wischnewski, L. Hermansdorfer, J. Betz, M. Lienkamp, and B. Lohmann, "Minimum curvature trajectory planning and control for an autonomous race car," *Vehicle System Dynamics*, 2019.

[16] J. Liu, P. Jayakumar, J. L. Stein, and T. Ersal, "Combined speed and steering control in high-speed autonomous ground vehicles for obstacle avoidance using model predictive control," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 10, pp. 8746–8763, Oct 2017.

[17] V. A. Laurense, J. Y. Goh, and J. C. Gerdes, "Path-tracking for autonomous vehicles at the limit of friction," in *American Control Conference (ACC), 2017*. IEEE, 2017, pp. 5586–5591.

[18] J. Y. Goh, T. Goel, and J. Christian Gerdes, "Toward automated vehicle control beyond the stability limits: Drifting along a general path," *Journal of Dynamic Systems, Measurement, and Control*, vol. 142, no. 2, 2020.

[19] J. Guldner, V. Utkin, J. Ackermann *et al.*, "A sliding mode control approach to automatic car steering," in *American Control Conference, 1994*, vol. 2. IEEE, 1994, pp. 1969–1973.

[20] M. Schorn, U. Stählin, A. Khanafer, and R. Isermann, "Nonlinear trajectory following control for automatic steering of a collision avoiding vehicle," in *American Control Conference, 2006*. IEEE, 2006, pp. 6–pp.

[21] S. C. Peters, E. Frazzoli, and K. Iagnemma, "Differential flatness of a front-steered vehicle with tire force control," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. IEEE, 2011, pp. 298–304.

[22] J. C. Gerdes and E. J. Rossetter, "A unified approach to driver assistance systems based on artificial potential fields," *Journal of Dynamic Systems, Measurement, and Control*, vol. 123, no. 3, pp. 431–438, 2001.

[23] E. J. Rossetter, J. P. Switkes, and J. C. Gerdes, "Experimental validation of the potential field lanekeeping system," *International journal of automotive technology*, vol. 5, no. 2, pp. 95–108, 2004.

[24] Kritayakirana, "Autonomous vehicle control at the limits of handling," Ph.D. dissertation, Stanford, 2012.

[25] F. Borrelli, P. Falcone, T. Keviczky, J. Asgari, and D. Hrovat, "Mpc-based approach to active steering for autonomous vehicle systems," *International Journal of Vehicle Autonomous Systems*, vol. 3, no. 2-4, pp. 265–291, 2005.

[26] C. Qiao, S. Pollmeyer, Y. Wang, and M. Fleps-Dezasse, "Model predictive trajectory planning of autonomous vehicles considering dynamic driving constraints," in *2020 4th CAA International Conference on Vehicular Control and Intelligence (CVCI)*. IEEE, 2020, pp. 136–141.

[27] R. Sharp, D. Casanova, and P. Symonds, "A mathematical model for driver steering control, with design, tuning and performance results," *Vehicle System Dynamics*, vol. 33, no. 5, pp. 289–326, 2000.

[28] M. Thommyppillai, S. Evangelou, and R. Sharp, "Car driving at the limit by adaptive linear optimal preview control," *Vehicle system dynamics*, vol. 47, no. 12, pp. 1535–1550, 2009.

[29] J. Levinson and S. Thrun, "Robust vehicle localization in urban environments using probabilistic maps," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. IEEE, 2010, pp. 4372–4378.

[30] M. Bertozzi and A. Broggi, "Gold: A parallel real-time stereo vision system for generic obstacle and lane detection," *IEEE transactions on image processing*, vol. 7, no. 1, pp. 62–81, 1998.

[31] A. K. Pamosoaji and K.-S. Hong, "A path-planning algorithm using vector potential functions in triangular regions," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 43, no. 4, pp. 832–842, 2013.

[32] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.

[33] S. M. LaValle and J. J. Kuffner Jr, "Randomized kinodynamic planning," *The International Journal of Robotics Research*, vol. 20, no. 5, pp. 378–400, 2001.

[34] C. Alia, T. Gilles, T. Reine, and C. Ali, "Local trajectory planning and tracking of autonomous vehicles, using clothoid tentacles method," in *Intelligent Vehicles Symposium (IV), 2015 IEEE*. IEEE, 2015, pp. 674–679.

[35] T. Gu, J. Snider, J. M. Dolan, and J.-w. Lee, "Focused trajectory planning for autonomous on-road driving," in *Intelligent Vehicles Symposium (IV), 2013 IEEE*. IEEE, 2013, pp. 547–552.

[36] X. Li, Z. Sun, Q. Zhu, and D. Liu, "A unified approach to local trajectory planning and control for autonomous driving along a reference path," in *Mechatronics and Automation (ICMA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 1716–1721.

[37] B. Vandewal, J. Gillis, E. Rademakers, G. Pipeleers, and J. Swevers, "Obstacle avoidance in path following using local spline relaxation," in *2020 IEEE 16th International Workshop on Advanced Motion Control (AMC)*. IEEE, 2020, pp. 85–90.

[38] T. Mercy, R. Van Parys, and G. Pipeleers, "Spline-based motion planning for autonomous guided vehicles in a dynamic environment," *IEEE Transactions on Control Systems Technology*, 2017.

[39] A. Scheuer and T. Fraichard, "Collision-free and continuous-curvature path planning for car-like robots," in *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, vol. 1. IEEE, 1997, pp. 867–873.

[40] O. Brock and O. Khatib, "High-speed navigation using the global dynamic window approach," in *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, vol. 1. IEEE, 1999, pp. 341–346.

[41] S. J. Anderson, S. C. Peters, T. E. Pilutti, and K. Iagnemma, "An optimal-control-based framework for trajectory planning, threat assessment, and semi-autonomous control of passenger vehicles in hazard avoidance scenarios," *International Journal of Vehicle Autonomous Systems*, vol. 8, no. 2-4, pp. 190–216, 2010.

[42] Y. Gao, A. Gray, A. Carvalho, H. E. Tseng, and F. Borrelli, "Robust nonlinear predictive control for semiautonomous ground vehicles," in *American Control Conference (ACC), 2014*. IEEE, 2014, pp. 4913–4918.

[43] J. Funke, M. Brown, S. M. Erlien, and J. C. Gerdes, "Collision avoidance and stabilization for autonomous vehicles in emergency scenarios," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 4, pp. 1204–1216, 2017.

[44] J. Liu, P. Jayakumar, J. L. Stein, and T. Ersal, "A nonlinear model predictive control formulation for obstacle avoidance in high-speed autonomous ground vehicles in unstructured environments," *Vehicle System Dynamics*, vol. 56, no. 6, pp. 853–882, 2018.

[45] J. Wurts, J. L. Stein, and T. Ersal, "Collision imminent steering at high speed using nonlinear model predictive control," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 8, pp. 8278–8289, 2020.

[46] J. Kabzan, M. I. Valls, V. J. Reijgwart, H. F. Hendrikx, C. Ehmke, M. Prajapat, A. Bühler, N. Gosala, M. Gupta, R. Sivanesan *et al.*, "Amz driverless: The full autonomous racing system," *Journal of Field Robotics*, vol. 37, no. 7, pp. 1267–1294, 2020.

[47] A. Lucchini, S. Formentin, M. Corno, D. Piga, and S. M. Savaresi, "Torque vectoring for high-performance electric vehicles: an efficient mpc calibration," *IEEE Control Systems Letters*, vol. 4, no. 3, pp. 725–730, 2020.

[48] F. Roselli, M. Corno, S. Savaresi, M. Giorelli, D. Azzolini, A. Irilli, and G. Panzani, "H∞ control with look-ahead for lane keeping in autonomous vehicles," in *IEEE Conference on Control Technology and Applications (CCTA)*. IEEE, 2017, pp. 2220–2225.

[49] S. Khaleghian, A. Emami, and S. Taheri, "A technical survey on tire-road friction estimation," *Friction*, vol. 5, no. 2, pp. 123–146, 2017.

[50] J. hwan Jeon, R. V. Cowlagi, S. C. Peters, S. Karaman, E. Frazzoli, P. Tsiotras, and K. Iagnemma, "Optimal motion planning with the half-car dynamical model for autonomous high-speed driving," in *American Control Conference (ACC), 2013*. IEEE, 2013, pp. 188–193.

[51] 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*, 2020.