Toward a Smart Car: Hybrid Nonlinear Predictive Controller With Adaptive Horizon

Matej Pčolka, Eva Žáčeková, Sergej Čelikovský, *Senior Member, IEEE,* and Michael Šebek, *Senior Member, IEEE*

***Abstract*— This paper focuses on the development of an optimization algorithm for car motion predictive control that addresses both hybrid car dynamics and hybrid minimization criterion. Instead of solving computationally demanding nonlinear mixed-integer programming task or approximating the hybrid dynamics/criterion, the *Hamiltonian-switching hybrid nonlinear predictive control* algorithm developed in this paper incorporates the information about hybridity directly into the optimization routine. To decrease the time complexity, several adaptive prediction horizon approaches are proposed, and for some of them, it is shown that they preserve maneuverability-related properties of the car. All developed alternatives are verified on an example of a motion control of a racing car and compared with the approximation-based nonlinear predictive control and a commercial product. Moreover, a sensitivity analysis examining robustness of the algorithm is included as well.**

***Index Terms*— Autonomous vehicles, hybrid systems, nonlinear model predictive control (MPC), optimization, vehicle control.**

I. Introduction

AUTOMOTIVE industry is one of the most dynamic engineering branches. Recently, huge progress toward an autonomous car has been witnessed [1]–[5], and out of the control methods able to replace a human driver, the model predictive control (MPC) is the most perspective one.

The most frequent variant is the linear MPC [4], [6], [7]. Although computationally simple, simplifications of the nonlinear dynamics/criterion provide only suboptimal performance. Some works present nonlinear MPC, however, they usually focus only on steering control [1], [8], [9]. The nonlinear MPC proposed in this paper manipulates steering, acceleration *and* braking and both satisfies the safety constraints and maximizes the performance indicators.

Complications arise from the strongly nonlinear sideslipping effects. One option is to model them by a steering coefficient being a piecewise continuous function of the forward velocity yielding a hybrid car dynamics model. Its attractiveness consists in replacing one complicated nonlinear function with a series of simpler subfunctions, as exploited in aerospace applications, chemical or electrical engineering [10]–[13].

Usually, mixed-integer programming (MIP) is exploited to handle the hybrid optimal control problems [14]–[16]; however, the nonlinear MIP tasks are NP-hard with exponentially growing time/computational demands [17], [18]. Some works [19], [20] propose alternatives, however, they either rely on restrictive assumptions such as *a priori* knowledge of the subdynamics sequence or perform a possibly time-consuming preprocessing. In this paper, we develop an algorithm requiring no such knowledge that avoids complex preprocessing, exploits *Hamiltonian-switcher* and solves the given optimization task directly as an ordinary nonlinear-programming task.

The computational burden is one of the weaknesses of the optimization-based approaches. Although dividing the "global" control task into smaller pieces and using decentralized approach [21] decreases the computational complexity, the price is the loss of optimality. However, since the complexity of the optimization task depends on the length of the optimization horizon, it can be reduced using adaptive horizon. In this paper, several alternatives are proposed with certain safety guarantees.

This paper is organized as follows. Section II introduces the vehicle behavior description, control requirements and constraints. Section III deals with the mathematical aspects of the problems the newly proposed algorithm focuses at, formulates a novel Hamiltonian-switcher-based algorithm and explains adaptive prediction horizon approaches. In Section IV, the results obtained from the numerical experiments are presented. Section V inspects the robustness of the proposed control algorithm with respect to parameter perturbations. Section VI concludes this paper.

II. Car Motion Modeling, Objectives, and Constraints

In the role of the test-bed system, a racing car with a hybrid steering coefficient was chosen.

A. Car Modeling

Car dynamics modeling is a highly delicate task, since the real car behavior is influenced by many factors, which

Manuscript received May 31, 2017; accepted August 14, 2017. Date of publication October 5, 2017; date of current version October 9, 2018. Manuscript received in final form August 26, 2017. This work was supported in part by the Czech Science Foundation through the Research Grant under Grant 17-04682S and in part by the European Regional Development Fund through the Project Rob4Ind4.0 under Grant CZ.02.1.01/0.0/0.0/15_003/0000470. Recommended by Associate Editor J. Sun. *(Corresponding author: Matej Pčolka.)*

M. Pčolka and E. Žáčeková are with the Department of Control Engineering, Faculty of Electrical Engineering, Czech Technical University in Prague, 121 35 Prague, Czech Republic (e-mail: pcolkmat@fel.cvut.cz; zacekeva@fel.cvut.cz).

S. Čelikovský is with the Institute of Information Theory and Automation, Czech Academy of Sciences, 182 08 Prague, Czech Republic (e-mail: celikovs@utia.cas.cz).

M. Šebek is with the Department of Control Engineering, Faculty of Electrical Engineering , Czech Technical University in Prague, 121 35 Prague, Czech Republic, and also with the Czech Institute of Informatics, Robotics and Cybernetics, Czech Technical University in Prague, 121 35 Prague, Czech Republic (e-mail: sebekm1@fel.cvut.cz; sebekm1@ciirc.cvut.cz).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCST.2017.2747504

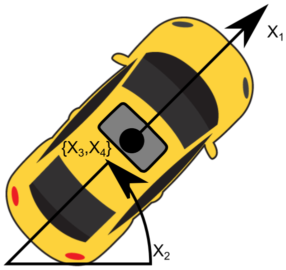

Fig. 1. State variables.

1) are constant (car mass, size and wheelbase), through those that 2) vary slightly/slowly (e.g., road inclination) up to those that 3) are highly nonlinear/stochastic [aerodynamic (im)perfections and their influence, car/road technical conditions and others].

In the literature, two main modeling branches are followed: kinematic (nonholonomic) modeling [22] and dynamic modeling [23]. While the dynamic modeling provides accurate models useful for simulation and analysis, the kinematic (nonholonomic) models are simpler and have low computational requirements, which is attractive for model-based control systems. On the other hand, they do not capture more complicated behavior, such as sideslipping. In this paper, this is overcome by a hybrid coefficient that models sideslipping as a decrease of the steering effectiveness.

The car dynamics is considered as follows:

$$\begin{aligned} x_{1,k+1} &= (p_1 - p_2 B_k)x_{1,k} + p_3 D_k, \\ x_{2,k+1} &= x_{2,k} + p_4 \alpha(x_{1,k}) \tan(S_k) x_{1,k}, \\ x_{3,k+1} &= x_{3,k} + p_5 \cos(x_{2,k}) x_{1,k}, \\ x_{4,k+1} &= x_{4,k} + p_5 \sin(x_{2,k}) x_{1,k}, \end{aligned} \tag{1}$$

where the forward speed $x_1\,(\mathrm{ms}^{-1})$, vehicle orientation x_2 (rad) and its x- and y-position $\{x_3, x_4\}$ (m) represent the state vector $x = [x_1, x_2, x_3, x_4]^{\mathrm{T}}$. For visualization, see Fig. 1. Regarding the manipulated variables $u = [D, B, S]^{\mathrm{T}}$, they correspond to normalized acceleration force $D\,(-)$, normalized braking force $B\,(-)$ and steering angle S (rad).

p_1 expresses how much the car velocity is preserved in the no-gas-no-braking case and meaningful values are those close to 1. Driving a noninclined road with a tarmac surface, p_1 is typically slightly lower than 1 mainly due to ubiquitous friction and air resistance. Lower sub-1 values are caused by driving a rougher terrain (increased friction), uphill driving (effect of gravitational force) or aerodynamic imperfections (increased drag coefficient), while slightly super-1 values indicate downhill driving. In this paper, $p_1 = 0.999$.

p_2 represents the braking effect. Since the braking deceleration can vary from 4.5 up to almost $9.8\,\mathrm{ms}^{-2}$, with the sampling period of 0.1 s and $p_1 = 0.999$, p_2 can range from slightly less than 0.02 to slightly more than 0.04 depending on the velocity, vehicle/road conditions and properties (wet/icy road, bald tires and mass distribution). $p_2 = 0.03$ used in model (1) yields a 100-to-0 kph braking distance of around 85 m, a reasonable value for rather unpaved surfaces.

TABLE I
SYSTEM PARAMETERS

Parameter	Value	Parameter	Value
$p_1\,(-)$	99.9×10^{-2}	$a_1\,(\mathrm{m}^{-1}\mathrm{s})$	-4.5×10^{-2}
$p_2\,(-)$	3×10^{-2}	$a_2\,(-)$	181×10^{-2}
$p_3\,(\mathrm{ms}^{-1})$	35×10^{-2}	$a_3\,(-)$	1913.22×10^{-1}
$p_4\,(\mathrm{rad\,m}^{-1}\mathrm{s})$	36.36×10^{-3}	$a_4\,(\mathrm{m}^{-1}\mathrm{s})$	-19.15×10^{-2}
$p_5\,(\mathrm{s})$	1×10^{-1}	$\{\underline{D}, \overline{D}\}\,(-)$	$\{0, 1\}$
$v_1\,(\mathrm{ms}^{-1})$	18	$\{\underline{B}, \overline{B}\}\,(-)$	$\{0, 1\}$
$v_2\,(\mathrm{ms}^{-1})$	35	$\{\underline{S}, \overline{S}\}\,(\mathrm{rad})$	$\{-\pi/6, \pi/6\}$

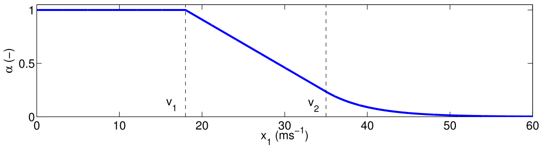

Fig. 2. Hybrid steering coefficient $\alpha(x_1)$.

The acceleration modeled by p_3 is affected by factors similar to those influencing p_1 and p_2. $p_3 = 0.35\,\mathrm{ms}^{-1}$ chosen here corresponds to 0-to-100 kph time around 8.5 s.

p_4 reflecting the influence of the car velocity and steering command on the car orientation is obtained as a product of the sampling period and the reciprocal of the wheelbase of the vehicle. In this paper, $p_4 = 36.36 \times 10^{-3}$ is assumed, which corresponds to a wheelbase of 2.75 m.

As mentioned, nonholonomic models describe the vehicle dynamics with sufficient accuracy at lower speeds and considering perfect adherence, however, they do not reflect sideslipping effects. In this paper, the sideslipping is interpreted as a decrease of steering effectiveness and is modeled by a piecewise continuous coefficient $\alpha(x_1)$ that equals 1 at lower speeds $x_1 \leq v_1$, decreases linearly between v_1 and v_2, and decays exponentially at speeds $x_1 > v_2$:

$$\alpha(x_1) = \begin{cases} \alpha_1(x_1) = 1 & 0 \leq x_1 \leq v_1, \\ \alpha_2(x_1) = a_1 x_1 + a_2 & v_1 < x_1 \leq v_2, \\ \alpha_3(x_1) = a_3\,\exp(a_4 x_1) & v_2 < x_1. \end{cases} \tag{2}$$

For graphical interpretation of $\alpha(x_1)$, see Fig. 2. The parameters $\{p_1, p_2, p_3, p_4, p_5\}$, $\{v_{1,2}\}$ and $\{a_1, a_2, a_3, a_4\}$ of (1) and (2) are provided in Table I. Further information on car dynamics and modeling can be found in [22]–[25].

B. Objectives and Constraints

In automobile racing, the lap time is usually minimized. This can be transformed into speed x_1 maximization, which then stands for the performance part of the overall criterion.

The second aspect of the same (if not even greater) importance is the safety, which turns into a requirement that the car stays on the track with a predefined width W. As usual in car racing, some predefined tolerance Δ_r is admitted.

The only technical constraints are those imposed on the manipulated variables D, B and S:

$$\underline{D} \leq D \leq \overline{D}, \quad \underline{B} \leq B \leq \overline{B}, \quad \underline{S} \leq S \leq \overline{S}. \tag{3}$$

The numerical values of $\{\underline{D}, \overline{D}\}$, $\{\underline{B}, \overline{B}\}$ and $\{\underline{S}, \overline{S}\}$ can be found in Table I.

III. Controller Design

In this section, a novel optimization algorithm for hybrid nonlinear predictive control is proposed and its application to the investigated task is explained.

A. Hybrid Nonlinear Predictive Control Algorithm

Let us introduce a general description of a discrete-time system with switched dynamics as follows:

$$x_{k+1} = F(x_k, u_k, \mathfrak{s}_{d,k}) \tag{4}$$

where the *dynamics switcher* $\mathfrak{s}_{d,k} = \mathcal{S}_d(x_k, u_k)$ indicating the current system dynamics is obtained by a mapping $\mathcal{S}_d : \mathbb{R}^{n+m} \mapsto \{1, 2, \dots, N_d\}$. Here, n and m are the dimensions of states x and inputs u, and $N_d \in \mathbb{N}^+$ is the number of switched dynamics. Moreover, let

$$\begin{aligned} F(x_k, u_k, 1) &= f_1(x_k, u_k),\\ F(x_k, u_k, 2) &= f_2(x_k, u_k),\\ &\vdots\\ F(x_k, u_k, N_d) &= f_{N_d}(x_k, u_k), \end{aligned} \tag{5}$$

where $f_{\mathfrak{s}_{d,k}}(x_k, u_k)$ expresses the particular subdynamics.

The hybrid optimization criterion $\mathcal{J}$ minimized at each time k is considered in the following form:

$$\mathcal{J} = \sum_{i=k+1}^{k+P} L(x_i, u_i, \mathfrak{s}_{c,i}) \tag{6}$$

with prediction horizon $P \in \mathbb{N}^+$. Next, assume that the function L can be expressed as

$$\begin{aligned} L(x_k, u_k, 1) &= l_1(x_k, u_k),\\ L(x_k, u_k, 2) &= l_2(x_k, u_k),\\ &\vdots\\ L(x_k, u_k, N_c) &= l_{N_c}(x_k, u_k), \end{aligned} \tag{7}$$

with $l_{\mathfrak{s}_{c,k}}(x_k, u_k)$ being the particular subcriterion term. Here, the *criterion switcher* $\mathfrak{s}_{c,k} = \mathcal{S}_c(x_k, u_k)$ is obtained by a mapping $\mathcal{S}_c : \mathbb{R}^{n+m} \mapsto \{1, 2, \dots, N_c\}$, where $N_c \in \mathbb{N}^+$ is the number of the hybrid parts of the cost criterion term L.

Then, the optimization task is summarized as follows.

For given initial condition x^-, find

$$u^* = \arg\min \mathcal{J}(x, u, \mathfrak{s}_d, \mathfrak{s}_c) \tag{8}$$

with respect to

$$\begin{aligned} x_{k+1} &= F(x_k, u_k, \mathfrak{s}_{d,k}),\\ u_{\min} &\le u \le u_{\max},\\ \mathfrak{s}_{d,k} &= \mathcal{S}_d(x_k, u_k) \in \{1, 2, \dots, N_d\},\\ \mathfrak{s}_{c,k} &= \mathcal{S}_c(x_k, u_k) \in \{1, 2, \dots, N_c\}. \end{aligned} \tag{9}$$

The common implementation [further referred to as *a priori switching MPC* (APS-MPC) algorithm] performed at each sampling instance k is as described by Algorithm 1.

The APS-MPC approach eliminates the hybridity by evaluating $\mathcal{S}_d$ and $\mathcal{S}_c$ prior to solving the optimization task and assuming $\mathfrak{s}_d$ and $\mathfrak{s}_c$ constant over the whole P, which enables use of the standard NLP solvers instead of more demanding MINLP in step 2 of APS-MPC.

```
Algorithm 1 APS-MPC
1) according to the last measured states $x^-$ and the last applied inputs $u^-$, evaluate the dynamics switcher $\mathfrak{s}_d = \mathcal{S}_d(x^-, u^-)$ and the criterion switcher $\mathfrak{s}_c = \mathcal{S}_c(x^-, u^-)$;
2) find $u^*$ minimizing $\mathcal{J} = \sum_{i=k+1}^{k+P} l_{\mathfrak{s}_c}(x_i, u_i)$ such that $x_{k+1} = f_{\mathfrak{s}_d}(x_k, u_k)$ with the state initial conditions $x^-$ and $u_{\min} \le u \le u_{\max}$;
3) apply the first sample of the optimized input $u^*$ into the system, wait for the next measurement, repeat from 1).
```

While for simple tasks, potential issues with the validity of the approximation are not crucial, a suitable alternative needs to be found for less trivial cases. Here, one such alternative—*Hamiltonian-switching hybrid nonlinear predictive control* (HaSH-NPC) algorithm—is derived as an adaptation of the Hamiltonian-based gradient method [26], [27].

The original gradient algorithm makes use of the Hamiltonian $H(x, u, \lambda) = \lambda_{k+1}^{\mathrm{T}} F(x_k, u_k) + L(x_k, u_k)$. In the hybrid case with system dynamics (5) and criterion term (7), it can be derived that

$$\mathcal{H}(x, u, \lambda) = \begin{cases} \lambda_{k+1}^{\mathrm{T}} f_1(x_k, u_k) + l_1(x_k, u_k) & \text{iff } \mathfrak{s}_d = 1,\ \mathfrak{s}_c = 1,\\ \lambda_{k+1}^{\mathrm{T}} f_1(x_k, u_k) + l_2(x_k, u_k) & \text{iff } \mathfrak{s}_d = 1,\ \mathfrak{s}_c = 2,\\ \vdots & \\ \lambda_{k+1}^{\mathrm{T}} f_1(x_k, u_k) + l_{N_c}(x_k, u_k) & \text{iff } \mathfrak{s}_d = 1,\ \mathfrak{s}_c = N_c,\\ \lambda_{k+1}^{\mathrm{T}} f_2(x_k, u_k) + l_1(x_k, u_k) & \text{iff } \mathfrak{s}_d = 2,\ \mathfrak{s}_c = 1,\\ \vdots & \\ \lambda_{k+1}^{\mathrm{T}} f_{N_d}(x_k, u_k) + l_{N_c}(x_k, u_k) & \text{iff } \mathfrak{s}_d = N_d,\ \mathfrak{s}_c = N_c. \end{cases}$$

To make the above description more compact, let us introduce a *Hamiltonian-switcher* $\mathfrak{s}_h = \mathcal{S}_h(\mathfrak{s}_d, \mathfrak{s}_c)$

$$\mathcal{S}_h : \{1, 2, \dots, N_d\} \times \{1, 2, \dots, N_c\} \mapsto \{1, 2, \dots, N_h\}. \tag{10}$$

Here, $N_h \in \mathbb{N}^+$ corresponds to the number of all possible Hamiltonian relations for the hybrid optimization problem. The mapping $\mathcal{S}_h$ can be with advantage chosen as

$$\mathcal{S}_h(\mathfrak{s}_d, \mathfrak{s}_c) = (\mathfrak{s}_d - 1)N_c + \mathfrak{s}_c. \tag{11}$$

Then, the hybrid-problem Hamiltonian can be constructed as

$$\mathcal{H}(x, u, \lambda, \mathfrak{s}_h) = H_{\mathfrak{s}_h}(x, u, \lambda) = \lambda_{k+1}^{\mathrm{T}} f_{\mathfrak{s}_d}(x_k, u_k) + l_{\mathfrak{s}_c}(x_k, u_k). \tag{12}$$

Instead of approximating the hybridity, the HaSH-NPC algorithm handles the problem correctly as described by Algorithm 2.

Let us note that the search step length choice is a highly complicated and still open question. While computationally least demanding, constant search steps often provide poor convergence. On the other hand, search steps obtained by a line search usually yield best convergence, however, their calculation might be prohibitively time-consuming. A fair tradeoff

Algorithm 2 HaSH-NPC

```
Obtain the state measurements $x^-$ and initial input profile estimate $U^0$; then iteratively repeat:
  1) use the initial condition $x_0 = x^-$ corresponding to the currently measured states and input profile from the previous iteration $U^{l-1}$ to obtain the state trajectories $X = [x_0, x_1, \dots, x_P]$ according to (4); store the dynamics-switcher profile $S_d = [\mathfrak{s}_{d,1}, \mathfrak{s}_{d,2}, \dots, \mathfrak{s}_{d,P}]$;
  2) evaluate the criterion-switcher mapping $\mathcal{S}_c(X, U^{l-q})$ and obtain the criterion-switcher profile $S_c = [\mathfrak{s}_{c,1}, \mathfrak{s}_{c,2}, \dots, \mathfrak{s}_{c,P}]$;
  3) according to mapping (11), obtain hamiltonian-switcher profile $S_h = [\mathfrak{s}_{h,1}, \mathfrak{s}_{h,2}, \dots, \mathfrak{s}_{h,P}]$;
  4) create the piece-wise continuous Hamiltonian $\mathcal{H}(x, u, \lambda, \mathfrak{s}_h)$ according to (12), calculate its derivatives with respect to $x$ and $u$;
  5) evaluate the co-state backward dynamics
       $$\lambda_k = \frac{\partial \mathcal{H}}{\partial x}(x_k, u_k^{l-1}, \lambda_{k+1}, \mathfrak{s}_{h,k}) = \frac{\partial \mathcal{H}_{\mathfrak{s}_{h,k}}}{\partial x}(x_k, u_k^{l-1}, \lambda_{k+1})$$
     with $\lambda_P = \partial J / \partial x|_P$; obtain the co-state trajectory $\Lambda = [\lambda_0, \lambda_1, \dots, \lambda_P]$;
  6) evaluate the gradient $\partial \mathcal{H} / \partial u$, obtain input profile $U^l$ by performing the gradient step as follows:
       $$U^l = U^{l-1} - \alpha_l \frac{\partial \mathcal{H}}{\partial u}(X, U^{l-1}, \Lambda, S_h), = U^{l-1} - \alpha_l \frac{\partial \mathcal{H}_{S_h}}{\partial u}(X, U^{l-1}, \Lambda); \quad (13)$$
  7) project $U^l$ on the admissible interval $\langle u_{\min}, u_{\max} \rangle$;
  8) if $\|U^l - U^{\{*,l-1\}}\| \leq \epsilon_1 \vee |J(U^l) - J(U^{l-1})| \leq \epsilon_2$, $\epsilon_1 > 0$, $\epsilon_2 > 0$, where
       $U^{\{*,l-1\}} = \arg\min(\min\{J(U^0), J(U^1), \dots, J(U^{l-1})\})$
     then terminate; apply the first sample of $U^{\{*,l\}}$ into the system, wait for the new measurements, go to 1);
     else $l = l + 1$, repeat from 1).
```

is offered by the heuristically chosen cost-function-dependent search steps that 1) are small (and prevent oscillations) if the cost criterion decreases rapidly and 2) increase (and speed up the convergence) if the cost criterion change is small. Moreover, their computational demands are negligible, since they can be expressed analytically. In this paper, the search step length α_l is considered as a function of the cost criterion value decrease $\Delta J_{l-1} = |J(U^{l-1}) - J(U^{l-2})|$ as follows:

$$\alpha_l = \beta \max(\overline{\alpha}, \min(\underline{\alpha}, -\log_{10}(\Delta J_{l-1}))), \tag{14}$$

where $\beta \gg 0$ and $\overline{\alpha} > \underline{\alpha} \gg 0$ shape and constrain the step.

B. Control Design

As indicated, the performance part of the criterion minimized over prediction horizon $P \in \mathbb{N}^+$ is expressed as

$$J_p = \sum_{i=k+1}^{k+P} -x_{1,i}. \tag{15}$$

TABLE II

CASE STUDY PARAMETERS

Parameter	Value	Parameter	Value
$\overline{R}$ (m)	3.5	ω_2 (−)	100
Δ_r (m)	0.5	ω_3 (−)	2
ω_1 (−)	7		

The satisfaction of the safety requirements can be accomplished in several ways. The first option is to track the central line given by $\{x_{\text{cent},k}, y_{\text{cent},k}\}$, which, however, disables speed optimization. Rather than that, keeping the x- and y-position within admissible limits is more advantageous. To handle this, a new state x_5 (m) representing the total driven distance is introduced, and the model (1) is extended as follows:

$$\begin{aligned} x_{\{1,\dots,4\},k+1} &\widehat{=} (1), \\ x_{5,k+1} &= x_{5,k} + p_5 x_{1,k}. \end{aligned} \tag{16}$$

As in [28], [29], feasibility issues are eliminated introducing relaxed safety part of the criterion formulated as follows:

$$J_s = \sum_{i=k+1}^{k+P} L(x_{3,i}, x_{4,i}, \mathbf{C}_X(x_{5,i}), \mathbf{C}_Y(x_{5,i})), \tag{17}$$

where

$$L = \begin{cases} 0 & r_i < \overline{R}, \\ |r_i - \overline{R}| & \overline{R} \leq r_i < \overline{R} + \Delta_r, \\ \omega_3 (r_i - \overline{R})^2 & \overline{R} + \Delta_r \leq r_i. \end{cases} \tag{18}$$

Here,

$$r_i = \sqrt{(x_{3,i} - \mathbf{C}_X(x_{5,i}))^2 + (x_{4,i} - \mathbf{C}_Y(x_{5,i}))^2} \tag{19}$$

represents the distance of the car from the central line $[\mathbf{C}_X, \mathbf{C}_Y]$, $\overline{R} = W/2$ is the half-width of the track, Δ_r is the considered tolerance, and ω_3 is a weighting parameter.

Having specified a set of discrete points $\{x_{\text{cent}}, y_{\text{cent}}\}$ lying on the central line and the corresponding driven distances $\{d_{\text{cent}}\}$ and exploiting spline interpolation techniques [30], functions $\mathbf{C}_X(x_5)$ and $\mathbf{C}_Y(x_5)$ can be obtained as $\mathbf{C}_X \approx x_{\text{cent}}(d_{\text{cent}})$, $\mathbf{C}_Y \approx y_{\text{cent}}(d_{\text{cent}})$, and then directly incorporated into the cost criterion (20).

To avoid simultaneous use of gas and brake, additional minimization term $D_k B_k$ is considered. The overall criterion for the predictive controller is then formulated as

$$\mathcal{J}_k = \omega_1 J_p + \omega_2 J_s + \sum_{i=k}^{P} D_i B_i. \tag{20}$$

J_p and J_s correspond to (15) and (17), respectively, and ω_1 and ω_2 are user-defined weights. The values of $\omega_{\{1,2,3\}}$, $\overline{R}$ and Δ_r are listed in Table II. Last of all, let us note that the solution is required to respect the hybrid dynamics (16) with $N_c = 3$ and $N_d = 3$ and satisfy constraints (3).

C. Adaptive Prediction Horizon

Prediction horizon is one of the key parameters specifying the tradeoff between computational complexity and optimality. The idea of adaptive prediction horizon comes in very naturally in case of car motion control—intuitively, the higher the velocity is, the longer horizon is needed to handle the car satisfactorily and respect the track constraints. In this paper, three adaptive prediction horizon approaches are considered.

1) Linear Adaptive Horizon ($\theta - P$ Approach): The prediction horizon is calculated using $[\theta x_1^-]$ being the nearest integer to a θ-multiple of x_1^-

$$P = \max(1, [\theta x_1^-]), \tag{21}$$

where $\theta > 0$ is a tuning parameter. Despite simple calculation, the choice of the parameter θ is tricky and depends on the current track—combination of long straight parts where the car velocity increases rapidly and short sharp curves demanding intensive braking requires higher θ, while presence of only low-curvature passages might also allow for lower θ. Absence of such information degrades the control performance. This shortcoming is eliminated by the more advanced alternatives for adaptive prediction horizon.

2) Nominal Logarithmic Adaptive Horizon (nom-$\log - P$ *Approach):* In this case, the horizon P is calculated as

$$P = \begin{cases} 1 & x_1^- \le v_1, \\ 1 + \left\lceil \log_{\overline{p}}\left(\frac{v_1}{x_1^-}\right)\right\rceil & v_1 < x_1^-, \end{cases} \tag{22}$$

where $\overline{p} = p_1 - p_2\overline{B} < 1$ represents the velocity dynamics coefficient with maximum braking and minimum acceleration. Here, $\lceil \epsilon \rceil$ denotes the smallest integer not less than ϵ. Now, let us define the *nominal car dynamics* as dynamics (1) with $x_1 \le v_1$, i.e., $\alpha(x_1) = 1$, and let us specify the *preservation of nominal maneuverability* as the physical capability of the car to drive over a trajectory that is realizable by the nominal car dynamics. Then, the following statement can be made.

Theorem 1. *Consider a vehicle with dynamics (16) with $\overline{p} = p_1 - p_2\overline{B} < 1$ and $\underline{D} = 0$. Let us assume that given initial conditions x^-, an optimal controller OC^∞ with prediction horizon $P = \infty$ with respect to criterion (20) and constraints (3) results in $\sup(r_k) \le \overline{R} + \Delta_r$. Then, given the same initial conditions x^-, an optimal controller $OC^\star$ with prediction horizon $P^\star$ calculated according to (22) preserves nominal maneuverability and also leads to $\sup(r_k) \le \overline{R} + \Delta_r$.*

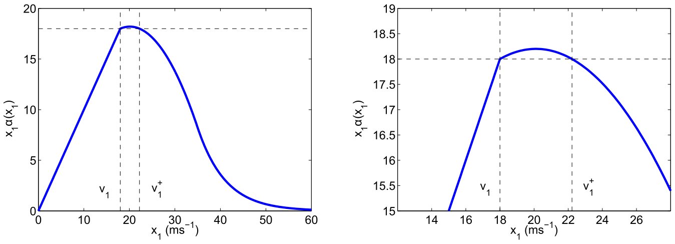

Fig. 3. $p_M(x_1)$ and supernominal maneuverability range $\langle 0, v_1^+\rangle$, detail.

Proof: The only difference between the nominal car dynamics and the dynamics of the real car is caused by the fact that $\alpha(x_{1,k}) < 1 \Leftrightarrow x_{1,k} > v_1$. Assuming an optimal controller, it can be expected that given enough information about the upcoming trajectory (represented by infinite prediction horizon $P = \infty$), the controller decreases the velocity $x_{1,k}$, such that $\alpha(x_{1,k}) = 1$ when necessary. Here, it should be noted that infinite prediction horizon, in fact, collapses to a horizon of such a finite length that the whole track is covered. A controller with shorter P (not covering the whole track) is able to ensure such a decrease only in case that P is large enough to bring x_1 from $x_{1,0} = x_1^-$ to $x_{1,P} < v_1$ with $D_k \equiv \underline{D}$, $B_k \equiv \overline{B}$. Directly substituting $D_k \equiv \underline{D}$, $B_k \equiv \overline{B}$ and $P^\star$ calculated according to (22) into the dynamics of the car (1), the nonlinear velocity dynamics turns into a linear one,

$$\overline{x}_{1,k+1} = \overline{p}\overline{x}_{1,k}, \tag{23}$$

with $\overline{p} = p_1 - p_2\overline{B} < 1$. Then, $\overline{x}_{1,P^\star} = \overline{p}^{P^\star}x_1^-$. Obviously, $x_1^- \le v_1$ yields $P^\star = 1$, $\overline{x}_{1,P^\star-1} \le v_1$, and $\alpha(\overline{x}_{1,P^\star-1}) = 1$. For $x_1^- > v_1$, the substitution leads to

$$\overline{x}_{1,P^\star} = \left(\overline{p}^{\left(1+\left\lceil \log_{\overline{p}}\left(\frac{v_1}{x_1^-}\right)\right\rceil\right)}\right) x_1^- \le \overline{p}\,\frac{v_1}{x_1^-}x_1^- < v_1. \tag{24}$$

Furthermore,

$$\overline{x}_{1,P^\star-1} \le \frac{v_1}{x_1^-}x_1^- \le v_1 \Leftrightarrow \alpha(\overline{x}_{1,P^\star-1}) = 1, \tag{25}$$

which means that $x_{2,P^\star}$ evolves from $x_{2,P^\star-1}$ according to the nominal car dynamics. Therefore, $P^\star$ calculated according to (22) provides enough information to preserve nominal maneuverability. □

3) Supernominal Logarithmic Adaptive Horizon (S-nom-$\log - P$ *Approach):* Inspecting the dynamics (1), it can be seen that the function $\tan(S)$ is multiplied not only by $\alpha(x_1)$ but by the *maneuverability product* $p_M(x_1) = x_1\alpha(x_1)$ that specifies the resulting efficiency of the steering S. Looking at Fig. 3, it is obvious that the value of p_M can be even higher than $v_1\alpha(v_1)$. Let us call such values *supernominal values of* p_M, and let us define the *supernominal maneuverability range* as

$$\langle 0, v_1^+\rangle = \{x_1 |\, p_M(x_1) \ge v_1\alpha(v_1) \;\vee\; \alpha(x_1) = 1\}.$$

The value of v_1^+ can be found as the higher solution of the equation $p_M(x_1) = v_1\alpha(v_1)$. Let us also specify the *preservation of supernominal maneuverability* as the capability of achieving that $x_{2,P-1}$ evolves to $x_{2,P}$ with either supernominal p_M or nominal $\alpha = 1$.

Now, let us have P calculated based on x_1^- as follows:

$$P = \begin{cases} 1 & x_1^- \le v_1^+, \\ 1 + \left\lceil \log_{\overline{p}}\left(\frac{v_1^+}{x_1^-}\right)\right\rceil & v_1^+ < x_1^-, \end{cases} \tag{26}$$

and formulate the following statement.

Theorem 2. *Consider a vehicle with dynamics (16) with $\overline{p} = p_1 - p_2\overline{B} < 1$ and $\underline{D} = 0$. Let us assume that given initial conditions x^-, an optimal controller OC^∞ with prediction horizon $P = \infty$ with respect to criterion (20) and constraints (3) results in $\sup(r_k) \le \overline{R} + \Delta_r$. Then, given the same initial conditions x^-, an optimal controller*

$OC^{s\star}$ with prediction horizon $P^{s\star}$ calculated according to (26) preserves supernominal maneuverability and also leads to $\sup(r_k) \le \overline{R} + \Delta_r$. *Furthermore, the following holds for the average value of prediction horizon exploited by controllers $OC^{s\star}$ and $OC^{\star}$:*

$$\mathrm{mean}(P^{s\star}|OC^{s\star}) \le \mathrm{mean}(P^{\star}|OC^{\star}).$$

Proof: The first part of the proof is similar to the previous case and consists in direct substitution of $D_k \equiv \underline{D}$, $B_k \equiv \overline{B}$, and $P^{s\star}$ calculated by (26) into (1). Having accomplished this, it can be shown that with $x_1^- > v_1^+$ and $P^{s\star}$,

$$\overline{x}_{1,P^{s\star}-1} = \left(\overline{p}^{\left\lceil \log_{\overline{p}}\left(\frac{v_1^+}{x_1^-}\right)\right\rceil}\right) x_1^- \le \frac{v_1^+}{x_1^-} x_1^- \le v_1^+, \qquad (27)$$

i.e., the velocity can be decreased from $x_1^- > v_1^+$ such that the supernominal maneuverability range is reached at $k = P^{s\star} - 1$ and $x_{2,P^{s\star}}$ can evolve from $x_{2,P^{s\star}-1}$ with supernominal p_M.

The second part of the proof comes from the comparison of expressions (22) and (26)—since $v_1 \le v_1^+$, $P^{s\star}(x_1^-)$ calculated according to (26) is not higher than $P^{\star}(x_1^-)$ calculated by (22) for any value of x_1^-. □

Remark. Considering r as an additional system output and $\mathbb{R}_{\mathrm{adm}} = \langle 0, \overline{R} + \Delta_r \rangle$ as admissible set for r, a controller can be found stabilizable if it ensures that $r_k \in \mathbb{R}_{\mathrm{adm}}\ \forall k \ge 0$ iff $r_0 \in \mathbb{R}_{\mathrm{adm}}$ or $\lim_{k\to\infty} r_k = r_a \in \mathbb{R}_{\mathrm{adm}}$ iff $r_0 \notin \mathbb{R}_{\mathrm{adm}}$. Then, it can be deduced that starting from initial conditions $r_0 \in \mathbb{R}_{\mathrm{adm}}$, a suitably tuned optimal controller with the proposed adaptive prediction horizon is able to keep r within the admissible bounds given that this is achievable by the nominal car, i.e. $\mathbb{R}_{\mathrm{adm}} = \langle 0, \overline{R} + \Delta_r \rangle$ is forward invariant with the proposed predictive controller and the adaptive predictive horizon. This covers the first part of the stability requirements. Their second part is covered by incorporating the track violation into the criterion (20). Since the controller makes control moves in the direction of negative gradient of the cost function, choosing suitable weights makes the nonzero safety part of the criterion decrease gradually from time $k-1$ to k, i.e., $J_{s,k} \le J_{s,k-1}$. Therefore, if $r_0 \notin \mathbb{R}_{\mathrm{adm}}$, the controller produces a series of control moves u_k such that $\lim_{k\to\infty} r_k = r_a \in \mathbb{R}_{\mathrm{adm}}$, which covers the second part of the stability requirements. As such, Theorems I and II and their proofs guarantee the recursive feasibility when using the nom-log-P and S-nom-log-P prediction horizons.

IV. Results

Several numerical experiments were performed on different tracks to examine the performance of all presented alternatives. Their results are presented in this section.

A. APS-MPC Versus HaSH-NPC Comparison

At first, the hybrid predictive control algorithms were tested on Track 1 with the nominal prediction horizon $P = 25$ samples. Fig. 4 shows the behavior of the car on the track.

Looking at Fig. 4, it seems that both algorithms respect the safety constraints satisfactorily. However, more details are provided by the topmost subfigure of Fig. 5, where the distance of the car from the central line is shown. The yellow line $r = \overline{R}$ indicates the inner zero-penalized part of the track, while the green line $r = \overline{R} + \Delta_r$ indicates the transition between the linear penalization and quadratic penalization.

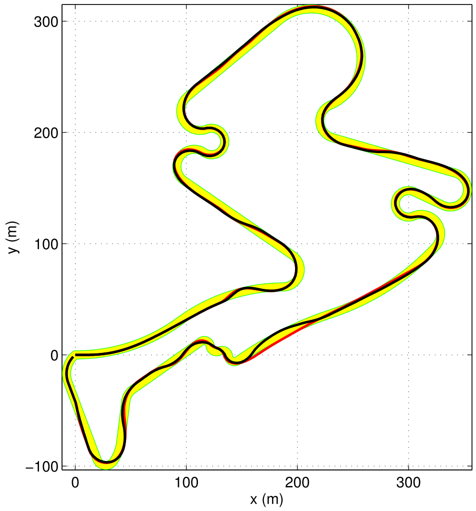

Fig. 4. Track 1. Black line: HaSH-NPC. Red line: APS-MPC.

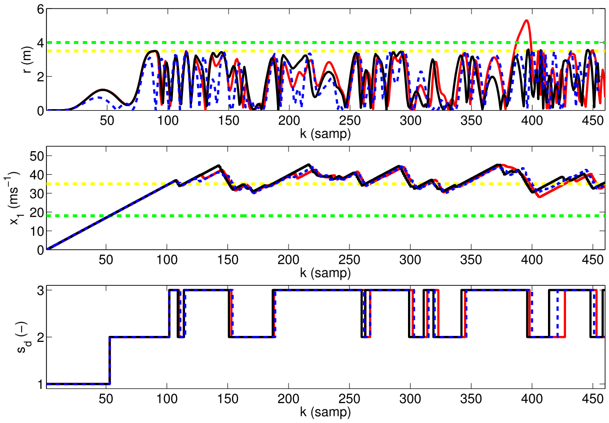

Fig. 5. Track 1—r, x_1, and $\mathfrak{s}_d$. Black line: HaSH-NPC with $P = 25$. Red line: APS-NPC with $P = 25$. Blue dashed line: APS-NPC with $P = 30$.

It can be seen that while the HaSH-NPC algorithm (represented by black solid line) almost never allows the car to leave the inner zero-penalized part of the track $r < \overline{R}$ and very safely satisfies the condition $r < \overline{R} + \Delta_r$, APS-MPC (represented by red solid line) working with the approximated description of the optimization problem happens to violate even the additional tolerance on the distance from the central line. Such a significant track violation can eventually bring the car to a point at which it is not able to return back to track and continue racing any more. This negative effect can be eliminated considering APS-MPC with increased prediction horizon $P = 30$ (represented by the blue dashed line). Although the use of longer prediction horizon complies with the expectations and helps to keep the car on the track, increase of the computational time can also be expected.

Fig. 5 also shows the x_1 profiles for the three above-mentioned variants (middle subfigure). Since the velocity determines the current system subdynamics, the velocities v_1 and v_2 of expression (2) are indicated by the green and yellow

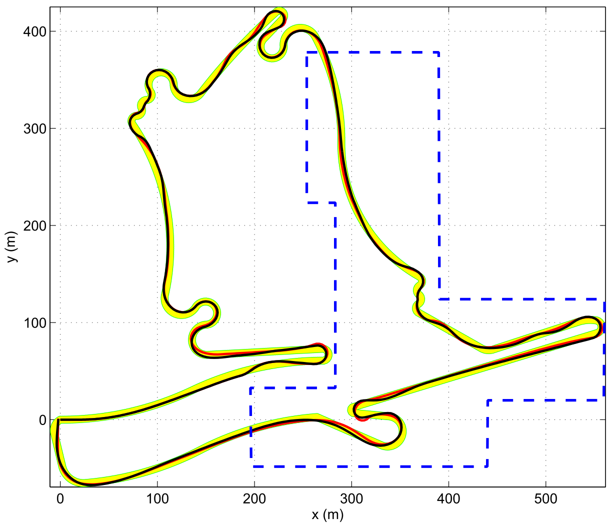

Fig. 6. Track 2. Black line: HaSH-NPC. Red line: APS-MPC.

dashed lines, and also the dynamics-switcher profiles $\mathfrak{s}_c$ are provided (see the bottommost subfigure).

It is obvious that the assumption on *a priori* known sequence of the system subdynamics cannot hold in this case, and therefore, the approaches relying on it could not be used. Given that the velocity (and thus the dynamics switcher) profiles are quite similar for the three depicted alternatives, while the track satisfaction differs significantly for HaSH-NPC versus APS-MPC with $P = 25$, it can be concluded that with equal prediction horizon, the HaSH-NPC handles the switching dynamics in a more appropriate way.

To obtain a more reliable comparison, another set of numerical experiments with the longer and more complicated Track 2 was performed. Track 2 and the behavior of the car with the two hybrid predictive control algorithms are presented in Fig. 6. Also in this case, nominal $P = 25$ was used.

In this case, the difference between the performances of the two algorithms is more significant. While the HaSH-NPC handles the complex track as well as the simpler one, certain problems in keeping the car on the track can be observed in case of APS-MPC. This is demonstrated by Fig. 7, where several details of the track are provided. Especially when driving at limit speed and cornering, the APS-MPC with nominal prediction horizon sometimes happens to get out of the track. Fig. 8 shows the velocity profiles and distances from the central line in one such situation in more detail. In the first subfigure, black and red lines represent the distance r reached by HaSH-NPC and APS-MPC at particular distance driven from the start d. Dashed lines mark $r = \overline{R}$ (yellow line) and $r = \overline{R} + \Delta_r$ (green line). In the second subfigure, black and red lines show velocity reached by HaSH-NPC/APS-MPC, and green and yellow dashed lines mark v_1 and v_2, respectively.

From Fig. 8, it can be seen that due to the approximation, APS-MPC does not decrease the speed sufficiently when cornering. This is not the case of HaSH-NPC, which acts appropriately and successfully satisfies the track tolerance. By increasing prediction horizon to $P = 45$ samples in case of APS-MPC (blue dashed line), even the algorithm working with approximation achieves satisfactory performance.

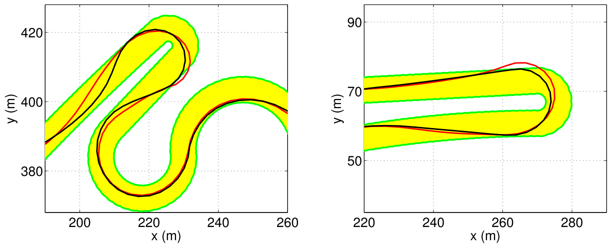

Fig. 7. Track 2. Black line: HaSH-NPC. Red line: APS-MPC.

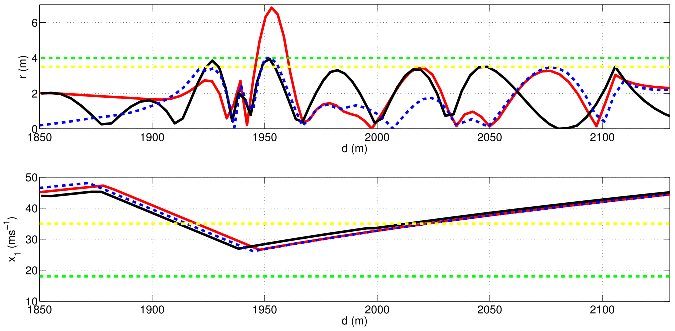

Fig. 8. Track 2—r and x_1. Black line: HaSH-NPC with $P = 25$. Red line: APS-NPC with $P = 25$. Blue dashed line: APS-NPC with $P = 30$.

TABLE III

COMPARISON OF HYBRID PREDICTIVE CONTROL ALGORITHMS

		$\overline{x}_1$ (ms^{-1})	$\overline{TV}$ (m)	$d_{\Delta_r V}$ (m)
T1	HaSH-NPC	33.4	0.1	0
	APS-MPC$_{25}$	33.0	1.8	34
	APS-MPC$_{30}$	33.2	0.1	0
T2	HaSH-NPC	34.7	0.4	0
	APS-MPC$_{25}$	34.2	3.3	42
	APS-MPC$_{35}$	34.4	0.9	10
	APS-MPC$_{45}$	34.6	0.5	0

The results of all experiments with constant prediction horizon are summarized in Table III, where T1 and T2 indicate the particular track. Several numerical evaluators were chosen as follows to provide a comprehensive comparison. As the first evaluator, the average velocity $\overline{x}_1$ was considered. The second evaluator $\overline{TV} = \max\{TV_k\}$ corresponds to the maximal track violation $TV_k = \max(0, r_k - \overline{R})$, where r_k is defined by (19). Let us remind the tolerance for the track violation $\Delta_r = 0.5$ m. The last evaluator $d_{\Delta_r V}$ represents the distance driven by the car when violating even the track tolerance ($r_k > \Delta_r$). For HaSH-NPC, prediction horizon $P = 25$ samples was considered, while in case of the other algorithm, the prediction horizon is indicated by the subscript (e.g., APS-MPC$_{35}$ means APS-MPC with horizon $P = 35$ samples).

Inspecting Table III, it can be seen that all algorithms achieve similar average velocities on particular track with slight superiority of HaSH-NPC results and (as expected), increase of prediction horizon results in increase of $\overline{x}_1$ in case of APS-NPC algorithm. However, a big difference can be noticed comparing $\overline{TV}$ and $d_{\Delta_r V}$. On Track 1, the HaSH-NPC algorithm never violates the track tolerance (see $\overline{TV} = 0.1 <$

TABLE IV
COMPARISON OF ADAPTIVE PREDICTIVE HORIZON APPROACHES

		$\overline{x}_1$ (ms^{-1})	*SP*	$\overline{P}$ (−)	max(P) (−)	E (−)
θ-P	$\theta = 0.2$	30.7	✗	6.6	11	–
	$\theta = 0.4$	32.7	✗	13.6	20	–
	$\theta = 0.6$	34.6	✓	20.6	31	1.68
	$\theta = 0.8$	34.7	✓	27.3	38	1.27
	$\theta = 1.0$	34.9	✓	33.3	49	1.05
nom-log-P		34.9	✓	22.4	36	1.56
S-nom-log-P		34.7	✓	17.8	29	1.95
c-HaSH-NPC		34.7	✓	25	25	1.39

Δ_r and $d_{\Delta_r V} = 0$). On the other hand, APS-MPC with $P = 25$ drives 34 m violating the track by more than 0.5 m with a maximal violation of 1.8 m. To make APS-MPC achieve acceptable performance the prediction horizon needs to be increased to $P = 30$ samples.

The situation is even more significant in case of the more complicated Track 2. Although $\overline{TV}$ of HaSH-NPC rises to 0.4 m, it stays within the defined track tolerance Δ_r with P as low as 25 samples. The APS-MPC algorithm, however, is not able to achieve desirable track-satisfaction performance even with $P = 35$ for which it still violates the track by up to 0.9 m. The satisfaction of the track tolerance is achieved with as long predictions as $P = 45$ samples. The poorer behavior of the APS-MPC algorithm is caused mainly by the approximation of the hybrid dynamics/cost criterion. MPC is a model-based controller, and therefore, neglecting/approximating the system dynamics in a significant way comes hand in hand with performance degradation. On the other hand, increasing P can remedy these negative effects, since more time is provided to take the corrective action. This explains why APS-MPC is outperformed by HaSH-NPC with equal prediction horizons and why also APS-MPC can satisfy the safety requirements with increased P.

B. Adaptive Prediction Horizon Approaches Comparison

To inspect the performance of the adaptive prediction horizon approaches, only HaSH-NPC algorithm was evaluated. Track 2 was considered because of its more complicated shape and a need for more aggressive car handling and maneuvering.

To obtain an illustrative and reliable comparison of different approaches, several evaluators were inspected. The first one was the average achieved velocity $\overline{x}_1$ representing the performance part of the criterion, while the safety part *SP* of the criterion, $\max(r_k - \overline{R}) \leq \Delta_r$, was evaluated binarily (✓—passed and ✗—failed). As the computational complexity and efficiency markers, average prediction horizon $\overline{P}$ considered by particular controller and "efficiency ratio" $E = \overline{x}_1/\overline{P}$ were evaluated as well. The results are summarized in Table IV. For the sake of completeness, results of algorithm with constant prediction horizon denoted as c-HaSH-NPC are provided as well.

At first, θ-P approach was tested with $\theta \in \{0.2, 0.4, 0.6, 0.8, 1\}$. It can be seen that while all θ-P variants with $\theta \geq 0.6$ passed the safety requirements, those with $\theta \geq 0.8$ might not be regarded as competitive due to their excessive computational complexity demonstrated by $\overline{P} \geq 27.3$. This comes hand-in-hand with a decrease of the efficiency ratio E, which degrades from $E = 1.68$ (for $\theta = 0.6$) to as low as $E = 1.05$ (for $\theta = 1.0$). The efficiency ratio for variants that did not pass the safety requirements was not evaluated.

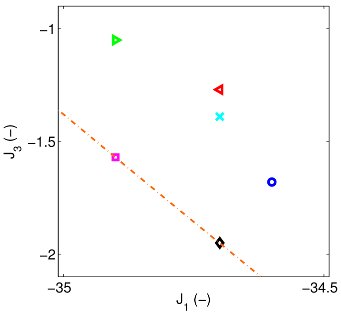

Fig. 9. Pareto optimality (○—0.6-P, ◁—0.8-P, ▷—1.0-P, ×—c-HaSH-NPC, □—nom-log-P, ◇—S-nom-log-P, and -·-—Pareto frontier).

Unlike the θ-P variants, nom-log-P and S-nom-log-P approaches provide both safety constraints satisfaction and attractive performance with high computational efficiency. While nom-log-P approach achieves the highest $\overline{x}_1$, the S-nom-log-P approach is clearly the most computationally efficient with $E = 1.95$.

Considering multiple evaluative criteria J_i, $i \in \{1, \ldots, n_i\}$ and a set of solutions $\mathbf{X}$, solution $\tilde{x} \in \mathbf{X}$ is said to *dominate* solution $\check{x} \in \mathbf{X}$ iff $J_i(\tilde{x}) \leq J_i(\check{x})$ for all i and at least for one $j \in \{1, \ldots, n_i\}$, $J_j(\tilde{x}) < J_j(\check{x})$. A solution $\hat{x} \in \mathbf{X}$ is said to be *Pareto optimal* iff it is not dominated by any other solution $\check{x} \in \mathbf{X}$. Further details on Pareto optimality might be found in [31]–[33] and references therein.

With three evaluative criteria $J_1 = -\overline{x}_1$, $J_2 = \overline{P}$ and $J_3 = -E$, it can be shown that out of all alternatives that passed the safety requirements, only nom-log-P and S-nom-log-P approaches are not dominated by any other solution and thus are *Pareto optimal*. This is demonstrated in Fig. 9—the *Pareto frontier* [31] comprising the Pareto optimal solutions consists exclusively of logarithm-based (nom-log-P and S-nom-log-P) approaches.

Fig. 10 shows the tradeoff between the efficiency E and the safety-requirements satisfaction by depicting P_k (horizon at time k) as a function of $x_{1,k}$. The S-nom-log-P approach splits the approaches into two groups—those lying completely above the S-nom-log-P-profile are safety-acceptable yet efficiently suboptimal while those that "undercrawl" it significantly are in turn more efficient but might be safety-unacceptable.

The overview is completed by a comparison with a commercially available MINLP solver provided in Table V. In this role, *ga* function implementing genetic algorithm being part of MATLAB Global Optimization Toolbox was employed with three different settings denoted as ga_1, ga_2, and ga_3. *CTR* (−) expressing the ratio between the average computational time of *ga* and HaSH-NPC was evaluated as well.

This comparison shows the main advantage of the HaSH-NPC algorithm against the MINLP solvers—elimination of high computational complexity. While with the computation-

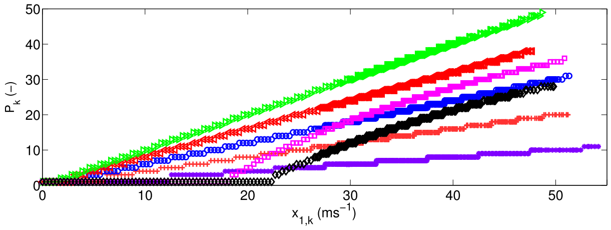

Fig. 10. P_k as a function of $x_{1,k}$ (∗—0.2-P, +—0.4-P, ○—0.6-P, ◁—0.8-P, ▷—1.0-P, □—nom-log-P, and ◇—S-nom-log-P).

TABLE V

MINLP Solver Results

	nom-log-P			S-nom-log-P		
	ga_1	ga_2	ga_3	ga_1	ga_2	ga_3
$\overline{x}_1$ (ms^{-1})	33.0	33.8	34.4	32.9	33.5	34.3
$\overline{\text{TV}}$ (m)	0.3	0.2	0.2	0.3	0.2	0.2
CTR (−)	42.5	150.0	1063.8	41.7	152.1	1069.9

TABLE VI

p_1 Sensitivity Analysis

p_1	nom-log-P		S-nom-log-P	
	$\overline{x}_1$ (ms^{-1})	$\overline{\text{TV}}$ (m)	$\overline{x}_1$ (ms^{-1})	$\overline{\text{TV}}$ (m)
0.99	38.1	0.2	37.6	0.2
0.999	**38.4**	**0.2**	**38.0**	**0.2**
0.99999	38.3	0.2	37.9	0.2
1.001	38.2	0.2	37.8	0.2
1.002	38.0	0.2	37.6	0.3
1.003	37.8	0.2	37.3	0.3
1.004	37.7	0.3	37.2	0.4

ally least demanding settings (which still consumes about 40× more time), the *ga* velocity performance is about 5% worse, the best achieved *ga* solution comparable with the HaSH-NPC one requires more than 1000× longer computations.

V. Sensitivity Analysis

To evaluate the robustness of the proposed control algorithms, a detailed scenario-based sensitivity analysis was performed exploiting a subsection of the second track encircled in Fig. 6 by the blue dashed line. In each of the analyzed cases, *nom*-log-*P* and *S-nom*-log-*P* approaches were tested and $\overline{x}_1$ and $\overline{\text{TV}}$ were evaluated. Out of the model parameters, p_5 was excluded from the sensitivity analysis. The other "p" parameters (i.e., p_{1-4}) were perturbed separately, while the α-formula (i.e., a_{1-4} and $v_{1,2}$ parameters) was changed as a whole. Let us note that the MPC model parameters correspond to Table I unless otherwise stated, and for completeness, the unperturbed cases are presented in the tables in blue.

A. p_1 Sensitivity Analysis

Six values ranging from 0.99 to 1.004 were chosen, which can be interpreted as uphill/rough terrain driving, noninclined road driving, and downhill driving. The results are presented in Table VI.

From the $\overline{\text{TV}}$ results it can be seen that none of the tested values causes safety problems, which means that the algorithm is sufficiently robust against the p_1 mismatch.

TABLE VII

p_2 Sensitivity Analysis

p_2	nom-log-P		S-nom-log-P	
	$\overline{x}_1$ (ms^{-1})	$\overline{\text{TV}}$ (m)	$\overline{x}_1$ (ms^{-1})	$\overline{\text{TV}}$ (m)
0.04	37.9	0.1	37.6	0.1
0.03	**38.4**	**0.2**	**38.0**	**0.2**
0.027	37.5	0.2	36.6	0.2
0.025	36.0	0.3	34.6	0.3
0.024	33.7	0.3	32.0	0.4
0.02	35.3	0.4	33.3	0.4
0.015	33.2	0.5	31.4	0.5
0.01	29.3	0.5	27.6	0.5
0.008	29.9	0.1	29.3	0.1
0.007	29.8	0.1	29.2	0.1
0.006	29.9	0.2	29.2	0.2
0.008	33.1	0.2	32.7	0.2
0.007	31.7	0.2	31.2	0.2
0.006	30.2	0.2	29.8	0.2

B. p_2 Sensitivity Analysis

Robustness against p_2 perturbation was tested on a set of ten scenarios with p_2 ranging from 0.04 to 0.006 to cover both the situations where the braking effectiveness is underestimated and those where the braking effect decreases (which can happen due to rain or snow) and MPC overestimates it. The results are listed in Table VII.

Inspecting the results obtained for $p_2 = 0.04$ to 0.024 (the first subtable), it can be seen that both variants are successful without any adaptations. Decreasing p_2 to 0.02, S-nom-log-P violated Δ_r, and therefore, the following workaround was proposed. Calculation of prediction horizon was performed considering "worst case guess" $p_{2,\text{wg}} = 0.01$, while for the MPC model itself, the original $p_2 = 0.03$ was used. Basically, only the prediction horizon was increased, while the dynamics remained the same. This was successfully tested for $p_2 = 0.02$ to 0.01 (see the second subtable). For $p_2 = 0.008$, Δ_r was again violated, and another workaround consisting in use of "worst case guess" for both the prediction horizon calculation and the MPC model was implemented with $p_{2,\text{wg}} = 0.005$. The usefulness of this adaptation is demonstrated by the results presented in the third subtable. Last of all, a p_2 estimator was designed according to (1) using x_1, D, and B measurements, and as the parameter for the optimizer, moving average calculated from p_2 estimates over the last ten samples was used. These results presented in the last subtable show that while already the original "no-estimator" algorithm had satisfied the safety requirements, the optimality in the sense of x_1 improved with the estimator.

C. p_3 Sensitivity Analysis

Regarding the p_3 parameter, eight values ranging from 0.2 to 0.7 corresponding to 0–100 kph acceleration times of 4–15 s were used. The results can be found in Table VIII.

The influence of the p_3 parameter perturbation on the performance is in some sense proportional to the p_3 perturbation—p_3 decrease/increase results in decrease/increase of both $\overline{x}_1$ and $\overline{\text{TV}}$, nevertheless,

TABLE VIII

p_3 SENSITIVITY ANALYSIS

p_3	nom-log-P		S-nom-log-P	
	$\overline{x}_1$ (ms^{-1})	$\overline{\text{TV}}$ (m)	$\overline{x}_1$ (ms^{-1})	$\overline{\text{TV}}$ (m)
0.20	36.0	0.1	35.7	0.1
0.25	36.9	0.1	36.7	0.1
0.30	37.6	0.1	37.2	0.2
0.325	38.2	0.2	37.7	0.2
0.35	**38.4**	**0.2**	**38.0**	**0.2**
0.40	38.6	0.2	38.3	0.3
0.50	39.3	0.3	38.7	0.3
0.60	39.7	0.3	39.2	0.4
0.70	40.0	0.3	39.6	0.4

TABLE IX

p_4 SENSITIVITY ANALYSIS

$p_4 \times 10^3$	nom-log-P		S-nom-log-P	
	$\overline{x}_1$ (ms^{-1})	$\overline{\text{TV}}$ (m)	$\overline{x}_1$ (ms^{-1})	$\overline{\text{TV}}$ (m)
33.33	36.5	0.4	35.8/35.6	0.6/0.4
34.48	37.5	0.4	37.0	0.5
35.71	38.2	0.3	37.7	0.4
36.36	**38.4**	**0.2**	**38.0**	**0.2**
37.04	38.3	0.2	37.9	0.2
38.46	38.1	0.1	37.7	0.1
40.00	37.9	0.1	37.3	0.1
41.67	37.7	0.0	37.1	0.1

the satisfaction of safety requirements remains unharmed for the whole inspected range.

D. p_4 Sensitivity Analysis

The p_4-perturbation robustness of the control algorithm was verified on a series of seven numerical experiments, where p_4 varied from 33.33×10^{-3} to 41.67×10^{-3}. Such values can be interpreted as wheelbase ranging from 2.4 to 3 m, which covers the vast majority of the race cars. The obtained results are shown in Table IX.

While the *nom*-log-P approach handles all evaluated p_4 values without violating the track constraints, the *S*-*nom*-log-P approach encounters difficulties with the lowest p_4 values representing a car with 3-m wheelbase. The difference between the two approaches is in their prediction horizon—although both algorithms optimize considering the same dynamics, slightly longer prediction horizon of *nom*-log-P approach provides it with enough time to take the corrective action. This weakness of the *S*-*nom*-log-P approach can be remedied by increasing the safety penalty ω_2 from 200 to 500. The results of such a configuration are presented in Table IX in the corresponding row after the slash mark. As can be seen, the unacceptable track violation was successfully eliminated.

Inspecting $\overline{x}_1$, it can be concluded that both understeering and oversteering lead to $\overline{x}_1$ decrease.

E. α Sensitivity Analysis

The α expression was perturbed as a whole to preserve monotonicity of the coefficient. These perturbations mean that multiple parameters were changed at time. Therefore, the perturbed coefficients are plotted instead of exact numerical perturbations of the particular parameters.

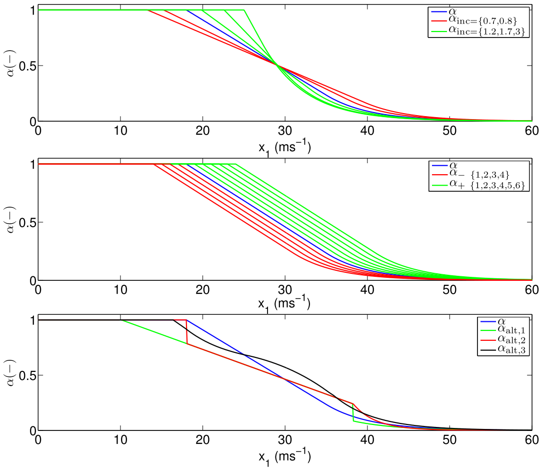

Fig. 11. Perturbations of α.

TABLE X

α SENSITIVITY ANALYSIS

α	nom-log-P		S-nom-log-P	
	$\overline{x}_1$ (ms^{-1})	$\overline{\text{TV}}$ (m)	$\overline{x}_1$ (ms^{-1})	$\overline{\text{TV}}$ (m)
$\alpha_{\text{inc}=0.7}$	31.9	0.4	30.0	0.4
$\alpha_{\text{inc}=0.8}$	34.1	0.3	32.1	0.3
$\boldsymbol{\alpha_{\text{inc}=1.0}}$	**38.4**	**0.2**	**38.0**	**0.2**
$\alpha_{\text{inc}=1.2}$	35.2	0.2	33.5	0.2
$\alpha_{\text{inc}=1.7}$	34.4	0.3	33.1	0.3
$\alpha_{\text{inc}=3.0}$	34.1	0.3	32.4	0.4
α_{+6}	32.2	0.3	31.3	0.4
α_{+5}	32.3	0.2	31.3	0.3
α_{+4}	32.9	0.2	32.0	0.3
α_{+3}	33.3	0.1	32.3	0.2
α_{+2}	33.5	0.0	33.4	0.2
α_{+1}	34.5	0.1	34.2	0.1
$\boldsymbol{\alpha_0}$	**38.4**	**0.2**	**38.0**	**0.2**
α_{-1}	35.8	0.3	34.9	0.3
α_{-2}	31.5	0.4	30.5	0.4
α_{-3}	30.7	0.4	29.5	0.5
α_{-4}	30.1	0.5	28.5	0.5
$\alpha_{\text{alt},1}$	34.3	0.4	32.6	0.4
$\alpha_{\text{alt},2}$	35.2	0.3	34.2	0.4
$\alpha_{\text{alt},3}$	36.3	0.2	35.4	0.3

At first, the inclination of α varied from 0.7 up to 3. These perturbations are in Fig. 11 denoted as $\alpha_{\text{inc}=i}$ (i stands for the inclination). Next, α was "shifted" by -4 up to $+6\,\text{ms}^{-1}$ as shown in Fig. 11, where the corresponding profiles are denoted as α_s (s is the velocity shift). Following the results from Section V-D, $\omega_2 = 500$ was used for *S*-*nom*-log-P with $\alpha_{\{-2,-3,-4\}}$. Three cases were added (see $\alpha_{\text{alt},1}$, $\alpha_{\text{alt},2}$ and $\alpha_{\text{alt},3}$ in Fig. 11). The results can be found in Table X.

Table X demonstrates that although inaccurate α expression degrades $\overline{x}_1$, no significant interventions are needed for the algorithms to keep the car on the track with $\overline{\text{TV}} \leq \Delta_r$. It should be noted that this also holds for the inspected cases where not only the parameters of the α-expression were perturbed, but even completely different mathematical functions (higher powers of x_1, their reciprocals and logarithms) were used, which is represented by $\alpha_{\text{alt},1}$, $\alpha_{\text{alt},2}$ and $\alpha_{\text{alt},3}$.

TABLE XI
PERFORMANCE ENHANCEMENT WITH PARAMETER ESTIMATORS

	nom-log-P		S-nom-log-P	
	$\overline{x}_1$ (ms^{-1})	$\overline{\mathrm{TV}}$ (m)	$\overline{x}_1$ (ms^{-1})	$\overline{\mathrm{TV}}$ (m)
no estimator	30.3	1.2	29.2	1.5
estimator	33.1	0.2	32.8	0.2
ideal case	33.3	0.2	33.0	0.2

F. Performance Enhancement

It can be expected that the performance might further improve with an estimator providing regular parameters corrections. To verify this, $p_{2,\text{real}} = 0.01$ and $\alpha_{\text{real}} = \alpha_{\text{alt},1}$ were used in the real system dynamics. The other parameters were kept at their original values, since either their influence was insignificant or they are not expected to be misestimated. The x_1 and x_2 measurements were corrupted by white noises with the variances of $\sigma_1 = 0.5$ and $\sigma_2 = 35 \times 10^{-3}$. Estimates of p_2 were obtained as described in Section V-B. Regarding α coefficient, its current value was regularly estimated as well, and the obtained $\{x_1, \alpha(x_1)\}$ pairs were used for recursive approximation of $\alpha(x_1)$ expression. These estimates were used for both *nom*-log-P and *S-nom*-log-P algorithms. Table XI presents the results achieved without and with parameter estimator and those obtained with perfect knowledge of the system parameters. The results demonstrate that even though the original MPC parameters might be inaccurate, their continuous estimation can change the performance from *unsatisfactory* to almost equivalent to the *ideal case*.

VI. CONCLUSION

In this paper, the HaSH-NPC being a new hybrid nonlinear MPC algorithm for vehicular control was designed. Unlike the commonly used solution approximating the optimization problem (APS-MPC), the HaSH-NPC handles the hybridity in the system dynamics/cost criterion directly exploiting an auxiliary variable—the *Hamiltonian-switcher*. The HaSH-NPC algorithm was verified on an example of a race car with hybrid dynamics considering hybrid cost criterion. The results show a very attractive performance of the HaSH-NPC, which even with short prediction horizon outperforms the APS-MPC algorithm.

The second part of this paper focused on adaptive prediction horizons. Linear and logarithm-based prediction horizon approaches were proposed. Their results show that while also linear prediction horizons improve the computational burden when compared with the constant prediction horizon, they might not be able to provide acceptable safety-requirements satisfaction. This is overcome by the logarithm-based approaches, which are also shown to be *Pareto optimal* with respect to multiple evaluative criteria. Additional comparison with a commercially available MINLP solver provided the same prediction horizons demonstrates that the HaSH-NPC requires only a fraction of MINLP solver computational time with comparable performance.

In the last part, the results of a detailed sensitivity analysis were presented demonstrating the robustness of the HaSH-NPC with respect to various system parameters perturbations. Several performance enhancements that can further improve the robustness and the overall functionality of the algorithm were also proposed.

The results encourage practical use of the algorithms that provide a "recipe" for computationally effective nonlinear MPC for the automotive area.

REFERENCES

[1] P. Falcone, F. Borrelli, J. Asgari, H. E. Tseng, and D. Hrovat, "Predictive active steering control for autonomous vehicle systems," *IEEE Trans. Control Syst. Technol.*, vol. 15, no. 3, pp. 566–580, May 2007.

[2] J. Levinson *et al.*, "Towards fully autonomous driving: Systems and algorithms," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2011, pp. 163–168.

[3] J. Funke *et al.*, "Up to the limits: Autonomous Audi TTS," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2012, pp. 541–547.

[4] C. E. Beal and J. C. Gerdes, "Model predictive control for vehicle stabilization at the limits of handling," *IEEE Trans. Control Syst. Technol.*, vol. 21, no. 4, pp. 1258–1269, Jul. 2013.

[5] H. Martínez-Barberá and D. Herrero-Pérez, "Multilayer distributed intelligent control of an autonomous car," *Transp. Res. C, Emerg. Technol.*, vol. 39, pp. 94–112, Feb. 2014.

[6] P. Falcone, M. Tufo, F. Borrelli, J. Asgari, and H. E. Tseng, "A linear time varying model predictive control approach to the integrated vehicle dynamics control problem in autonomous systems," in *Proc. 46th IEEE Conf. Decision Control*, Dec. 2007, pp. 2980–2985.

[7] P. Falcone, F. Borrelli, H. E. Tseng, J. Asgari, and D. Hrovat, "Linear time-varying model predictive control and its application to active steering systems: Stability analysis and experimental validation," *Int. J. Robust Nonlinear Control*, vol. 18, no. 8, pp. 862–875, 2008.

[8] T. Keviczky, P. Falcone, F. Borrelli, J. Asgari, and D. Hrovat, "Predictive control approach to autonomous vehicle steering," in *Proc. IEEE Amer. Control Conf.*, Jun. 2006, pp. 1–6.

[9] P. Falcone, F. Borrelli, H. E. Tseng, J. Asgari, and D. Hrovat, "A hierarchical model predictive control framework for autonomous ground vehicles," in *Proc. IEEE Amer. Control Conf.*, Jun. 2008, pp. 3719–3724.

[10] M. H. L. Hounjet and J. J. Meijer, "Evaluation of elastomechanical and aerodynamic data transfer methods for non-planar configurations in computational aeroelastic analysis," Nat. Aerosp. Lab., Amsterdam, The Netherlands, Tech. Rep. NLR-TP-95690 U, 1995.

[11] T. Geyer, M. Larsson, and M. Morari, "Hybrid emergency voltage control in power systems," in *Proc. Eur. Control Conf.*, 2003, pp. 1893–1898.

[12] Q. A. Acton, *Issues in Aerospace and Defense Research and Application: 2013 Edition*. Atlanta, GA, USA: ScholarlyEditions, 2012. [Online]. Available: https://books.google.cz/books?id=B4O0hukUBt8C

[13] L. Zhang, S. Wang, H. R. Karimi, and A. Jasra, "Robust finite-time control of switched linear systems and application to a class of servomechanism systems," *IEEE/ASME Trans. Mechatronics*, vol. 20, no. 5, pp. 2476–2485, Oct. 2015.

[14] O. Stursberg and S. Engell, "Optimal control of switched continuous systems using mixed-integer programming," in *Proc. 15th IFAC World Congr. Autom. Control*, Barcelona, Spain, 2002, pp. 433–438.

[15] M. Morari, M. Baotic, and F. Borrelli, "Hybrid systems modeling and control," *Eur. J. Control*, vol. 9, nos. 2–3, pp. 177–189, 2003.

[16] T. Schouwenaars, B. De Moor, E. Feron, and J. How, "Mixed integer programming for multi-vehicle path planning," in *Proc. Eur. Control Conf.*, vol. 1. 2001, pp. 2603–2608.

[17] H. W. Lenstra, Jr., "Integer programming with a fixed number of variables," *Math. Oper. Res.*, vol. 8, no. 4, pp. 538–548, 1983.

[18] L. A. Wolsey, "Mixed integer programming," in *Wiley Encyclopedia of Computer Science and Engineering*. Hoboken, NJ, USA: Wiley, 2008.

[19] X. Xu and P. J. Antsaklis, "Results and perspectives on computational methods for optimal control of switched systems," in *Hybrid Systems: Computation and Control*. New York, NY, USA: Springer, 2003, pp. 540–555.

[20] S. Xu, S. E. Li, K. Deng, S. Li, and B. Cheng, "A unified pseudospectral computational framework for optimal control of road vehicles," *IEEE/ASME Trans. Mechatronics*, vol. 20, no. 4, pp. 1499–1510, Aug. 2015.

[21] E. Kayacan, E. Kayacan, H. Ramon, and W. Saeys, "Robust tube-based decentralized nonlinear model predictive control of an autonomous tractor-trailer system," *IEEE/ASME Trans. Mechatronics*, vol. 20, no. 1, pp. 447–456, Feb. 2015.

[22] R. Pepy, A. Lambert, and H. Mounier, "Path planning using a dynamic vehicle model," in *Proc. 2nd IEEE Inf. Commun. Technol. (ICTTA)*, vol. 1. Apr. 2006, pp. 781–786.

[23] R. N. Jazar, *Vehicle Dynamics: Theory and Application*. New York, NY, USA: Springer, 2008.

[24] H. Pacejka, *Tire and Vehicle Dynamics*. Amsterdam, The Netherlands: Elsevier, 2005.

[25] W. F. Milliken and D. L. Milliken, *Race Car Vehicle Dynamics*, vol. 400. Warrendale, PA, USA: SAE International, 1995.

[26] A. E. Bryson, *Applied Optimal Control: Optimization, Estimation, and Control*. New York, NY, USA: Taylor & Francis, 1975.

[27] K. Zhou, J. C. Doyle, and K. Glover, *Robust and Optimal Control*, vol. 40. Englewood Cliffs, NJ, USA: Prentice-Hall, 1996.

[28] C. R. He, H. Maurer, and G. Orosz, "Fuel consumption optimization of heavy-duty vehicles with grade, wind, and traffic information," *J. Comput. Nonlinear Dyn.*, vol. 11, no. 6, p. 061011, 2016.

[29] H. Maurer and S. Pickenhain, "Second-order sufficient conditions for control problems with mixed control-state constraints," *J. Optim. Theory Appl.*, vol. 86, no. 3, pp. 649–667, 1995.

[30] C. de Boor, *A Practical Guide to Splines*. New York, NY, USA: Springer-Verlag, 1978.

[31] Y. Censor, "Pareto optimality in multiobjective problems," *Appl. Math. Optim.*, vol. 4, no. 1, pp. 41–59, 1977.

[32] P. M. Pardalos, A. Migdalas, and L. Pitsoulis, Eds., *Pareto Optimality, Game Theory and Equilibria*, vol. 17. New York, NY, USA: Springer, 2008.

[33] W. Stadler, Ed., *Multicriteria Optimization in Engineering and in the Sciences*, vol. 37. New York, NY, USA: Springer, 2013.

Matej Pčolka received the bachelor's degree in cybernetics and measurement and the master's degree (Hons.) in aircraft instrument and control systems from Czech Technical University in Prague, Prague, Czech Republic, in 2009 and 2011, respectively, where he is currently pursuing the Ph.D. degree.

He was a Visiting Scholar with Katholieke Universiteit Leuven, Leuven Belgium, in 2012, and Michigan Technological University, Houghton, MI, USA, in 2013. His current research interests include numerical optimization, constrained optimal control, and nonlinear model predictive control with application in the area of bioprocesses, building climate control, and autonomous vehicle control.

Eva Žáčeková received the bachelor's degree in cybernetics and measurement and the master's degree (with honors) in systems and control from Czech Technical University in Prague, Prague, Czech Republic, in 2009 and 2011, respectively, where she is currently pursuing the Ph.D. degree.

She was a Visiting Scholar with Katholieke Universiteit Leuven, Leuven, Belgium, in 2012, and University of California, Berkeley, CA, USA, in 2013. Her current research interests include system identification, control relevant identification, persistent excitation, and dual model predictive control with application in the area of building climate control and autonomous vehicle control.

Sergej Čelikovský (SM'03) received the M.Sc. degree in applied mathematics from the Optimal Control Department, Moscow State University, Moscow, Russia, in 1984, and the Ph.D. degree in technical cybernetics from the Czechoslovak Academy of Sciences, Prague, Czech Republic, in 1988.

He was a Visiting Researcher with the University of Twente, Enschede, The Netherlands, in 1996, the Department of Mechanical and Automation Engineering, Chinese University of Hong Kong, Hong Kong, in 1998, and CINVESTAV, Instituto Politécnico Nacional, Mexico City, Mexico, from 1998 to 2000. He is currently a Research Fellow and the Head of the Department of Control Theory, Institute of Information Theory and Automation, Czech Academy of Sciences, and a Full Professor with the Department of Control Engineering, Czech Technical University in Prague, Prague. He is the coauthor of one book and two book chapters, 57 papers in the journals reported in WoS, and over 100 papers in the international conference proceeding. His current research interests include nonlinear systems, exact feedback linearization, stabilization, chaotic systems, observers, and output feedback control, underactuated mechanical systems, and walking robots.

Michael Šebek (SM'91) received the M.Sc. (Hons.) in electrical engineering from Czech Technical University in Prague, Prague, Czech Republic, and the Ph.D. and D.Sc. degrees in control theory from the Czech Academy of Sciences, Prague.

He is currently the Head of the Control Engineering Department, Czech Technical University in Prague, and the Head of the Cyber-Physical Systems Department, Czech Institute for Informatics, Cybernetics, and Robotics in Prague. He is also the CEO of PolyX Ltd., and a Producer of the Polynomial Toolbox for MATLAB. He held several visiting positions abroad, including the University of Twente, Enschede, The Netherlands, and ETH Zurich, Zürich, Switzerland. He has published well over 200 research papers, developed several commercial software packages, and has led numerous research projects. His current research interests include linear systems and control, networked systems, and vehicle formations.

Dr. Šebek was a Council Member, International Federation of Automatic Control. In 2016, he received the Werner von Siemens Best Educator Prize. He was the General Chair of the 16th World Congress.