Vehicle Path-Tracking Linear-Time-Varying Model Predictive Control Controller Parameter Selection Considering Central Process Unit Computational Load

Zejiang Wang
Walker Department of Mechanical Engineering,
The University of Texas at Austin,
Austin, TX 78712
e-mail: wangzejiang@utexas.edu

Yunhao Bai
Department of Electrical and
Computer Engineering,
The Ohio State University,
Columbus, OH 43210
e-mail: bai.228@osu.edu

Junmin Wang[1]
Fellow ASME
Walker Department of Mechanical Engineering,
The University of Texas at Austin,
Austin, TX 78712
e-mail: jwang@austin.utexas.edu

Xiaorui Wang
Department of Electrical and
Computer Engineering,
The Ohio State University,
Columbus, OH 43210
e-mail: wang.3596@osu.edu

Model predictive control (MPC) has drawn a considerable amount of attention in automotive applications during the last decade, partially due to its systematic capacity of treating system constraints. Even though having received broad acknowledgements, there still exist two intrinsic shortcomings on this optimization-based control strategy, namely the extensive online calculation burden and the complex tuning process, which hinder MPC from being applied to a wider extent. To tackle these two drawbacks, different methods were proposed. Nevertheless, the majority of these approaches treat these two issues independently. However, parameter tuning in fact has double-sided effects on both the controller performance and the real-time computational burden. Due to the lack of theoretical tools for globally analyzing the complex conflicts among MPC parameter tuning, controller performance optimization, and computational burden easement, a look-up table-based online parameter selection method is proposed in this paper to help a vehicle track its reference path under both the stability and computational capacity constraints. MATLAB-CARSIM conjoint simulations show the effectiveness of the proposed strategy.

[DOI: 10.1115/1.4042196]

1 Introduction

Due to its ability of systematical handling states and inputs constraints, model predictive control (MPC) gained a great attention in automotive applications during the last decade. For instance, in Ref. [1], nonlinear MPC algorithms were designed to restrict estimated ammonia coverage ratios within the target regions in order to maintain the performance of a degraded selective catalytic reduction system. In Ref. [2], MPC-based torque-split strategies for hybrid electric vehicles demonstrated satisfying fuel saving result. In Ref. [3], both holistic and hierarchical MPC controllers were utilized to stabilize a critically unstable vehicle by controlling the independent braking toques of rear wheels. In addition, in Ref. [4], a linear-time-varying (LTV) MPC controller was designed to coordinate active front steering (AFS) system and independent wheel braking/tracking for helping a car follow its reference path.

Even with the prevalence of MPC, two inherent drawbacks of this optimization-based control law, which indeed impede its real-time implementation, remain salient. On the one hand, the receding-horizon characteristic of MPC necessitates in solving a constrained optimization problem within each sampling interval, which leads to an extensive calculation burden. On the other hand, an MPC controller typically includes a considerable number of parameters to be tuned, which can be a laborious task especially when multiple competing objectives exist. To tackle these two aforementioned issues, substantial efforts have been made. To alleviate the huge computational burden of MPC, the explicit MPC [5] solves the optimization problem offline and exploits a look-up table to realize online evaluation. However, explicit MPC can only treat the optimization problem with a relative small dimension: a few states or/and manipulated variables, loose constraints, and short prediction horizon, etc., and cannot handle time-varying systems [6]. Besides, several mechanisms have been proposed to decrease the dimension of the constrained optimization problem and consequently to mitigate the computational burden. For instance, in Ref. [7], a Kreisselmeier–Steinhauser function was employed to replace soft constraints of the optimization problem. In Ref. [8], a combination of the "moving block" and the "constraint set compression" strategies exhibited a favorable computational efficiency. In addition, efficient optimization solvers, such as the Newton–Raphson iteration [9], were also proposed to accelerate problem solving. Apart from the theoretical progress, hardware advances with field-programmable gate array [10] shed some light on the possibility of rapid implementation of MPC on more general embedded systems. Other than the mitigation of computational load, effective parameter tuning also plays a crucial role in a successful MPC implementation as well, since it has a direct influence on the performance of an MPC controller. Roughly speaking, existing MPC tuning methods in the literatures could be divided into three groups: thumb rules, auto-tuning strategies, and analytical approaches. Exhaustive tuning guidelines for MPC controller were introduced in Ref. [11]. However, a common weakness of these general tuning rules lies in the fact that they may become invalid when any system constraints become active. In contrast to the thumb rules, auto-tuning strategies grounded on genetic algorithms [12], particle swarm optimization [13], or fuzzy logic [14] change tuning parameter values in a self-adaptive manner, which could handle constraint violation easily.

[1]Corresponding author.

Contributed by the Dynamic Systems Division of ASME for publication in the JOURNAL OF DYNAMIC SYSTEMS, MEASUREMENT, AND CONTROL. Manuscript received June 5, 2018; final manuscript received November 26, 2018; published online January 14, 2019. Assoc. Editor: Huiping Li.

Nonetheless, the auto-tuning strategies make the computational burden issue of MPC even more pronounced since at each time-step, an extra optimization problem needs to be solved to simultaneously determine the control parameters. Finally, analytical approaches [15] study the effect of parameter values from a control theory viewpoint, which create some new tuning guidelines for MPC.

Even though both the methods that can adequately alleviate MPC computational burden as well as the approaches that lead to effective parameter tunings exist in the literature, there are few techniques treating these two problems within a unified framework. Actually, a group of tuning parameters, which ensures a higher controller performance, generally renders the optimization problem to be solved at each time-step more complicated, and such complicatedness entails a more marked computational burden of the MPC control law. By virtue of the fact that MPC performance and computational burden are closely coupled with each other, they should be handled *dependently* instead of *separately* when parameters are selected. Due to the lack of theoretical tools for analyzing the entangled difficulties among efficient MPC parameter tuning, controller performance optimization, as well as computational burden easement, a look-up table-based online parameter selection method of an LTV-MPC controller for vehicle path tracking is proposed in this paper.

To begin with, a standard LTV-MPC controller was derived. Then various tuning parameters in the controller were divided into two groups separately as the *significant parameters*, i.e., the prediction horizon, the control horizon, and the sampling period, and the *insignificant parameters*, such as the weighting matrices, hard constraint bounds on system input and system input changing rate, as well as soft constraint bounds on vehicle tire sideslip angles for ensuing vehicle stability. Subsequently, methodological approaches for tuning the insignificant parameters were deigned and verified. After that, the performance of the LTV-MPC controller was quantitatively analyzed from three distinct aspects, i.e., path-tracking performance, vehicle stability, and entailed computational load under various combinations of the three significant parameters. Consequently, three performance maps as look-up tables were constructed. Finally, grounded in the three performance maps, a constrained optimal tracking parameter selection algorithm was proposed to achieve the highest attainable path-tracking performance while considering a minimum stability requirement under a limited exploitable central process unit (CPU) computational capacity. The major contribution of this paper is a systematic approach to synchronously adjust the three significant parameters, i.e., the prediction horizon, the control horizon, and the sampling frequency of the LTV-MPC controller in real-time to negotiate the intricate conflicts among path-tracking performance optimization, vehicle stability maintenance, and computational load limitation.

The rest of the paper is organized as follows: To begin with, system modeling and a classical AFS MPC controller are introduced in Sec. 2. Thereafter, insignificant parameters' setting strategies are illustrated and verified in Sec. 3. The definition of "significant" and "insignificant" parameters will also be given in this section. Afterward, controller performance index definitions are given in Sec. 4, followed by the generation of performance maps. Subsequently, based on the generated performance maps, an on-line parameter selection algorithm is proposed and validated in Sec. 5. As a final point, Sec. 6 concludes the paper.

2 Active Front Steering Model Predictive Control Controller Design for Vehicle Path Tracking

Since the focus of this paper is to propose a systematic parameter selection approach for extant MPC controllers rather than designing new controllers, a classical LTV-MPC AFS controller similar in Ref. [4] is utilized here with only minor improvements.

2.1 System Modeling. A three degrees-of-freedom bicycle model for a front steering vehicle is used here to represent the dynamics of the system, as Ref. [4]

$$\begin{cases} \dot{v}_y = -v_x\gamma + \dfrac{\sum F_y}{m}, & \dot{v}_x = v_y\gamma + \dfrac{\sum F_x}{m}, \\ \dot{Y} = v_x\sin(\psi) + v_y\cos(\psi), & \dot{\psi} = \gamma, \\ \dot{X} = v_x\cos(\psi) - v_y\sin(\psi), & \dot{\gamma} = \dfrac{\sum M_z}{I_z} \end{cases} \tag{1}$$

with m being the mass of the vehicle; I_z as the yaw inertia; v_x, v_y, and γ representing vehicle's longitudinal velocity, lateral velocity, and yaw rate at the center of gravity (CG); X, Y, and ψ demonstrating separately the position of vehicle's CG in the inertial coordinate as well as the yaw angle of the vehicle. In addition, $\sum F_y, \sum F_x$, and $\sum M_z$ represent the total lateral tire force, longitudinal tire force, and yaw moment acting on the vehicle, as Ref. [4]

$$\sum F_y = (F_{xfl} + F_{xfr})\sin(\delta_f) + (F_{yfl} + F_{yfr})\cos(\delta_f) + F_{yrl} + F_{yrr} \tag{2}$$

$$\sum F_x = (F_{xfl} + F_{xfr})\cos(\delta_f) - (F_{yfl} + F_{yfr})\sin(\delta_f) + F_{xrl} + F_{xrr} \tag{3}$$

$$\begin{aligned} \sum M_z = {} & l_f[(F_{xfl} + F_{xfr})\sin(\delta_f) + (F_{yfl} + F_{yfr})\cos(\delta_f)] \\ & - l_r[F_{yrl} + F_{yrr}] + l_d[(F_{yfl} - F_{yfr})\sin(\delta_f) \\ & + (-F_{xfl} + F_{xfr})\cos(\delta_f) - F_{xrl} + F_{xrr}] \end{aligned} \tag{4}$$

with l_f, l_r, l_d representing the distances from the CG to the front/rear axle and the half of the vehicle width, δ_f as the front road steering angle of the vehicle, which serves as the unique output of the MPC controller. Finally, $F_{\{x,y\}\{i,j\}}$ ($i \in \{(\mathrm{f})\mathrm{ront}, (\mathrm{r})\mathrm{ear}\}$, $j \in \{(\mathrm{l})\mathrm{eft}, (\mathrm{r})\mathrm{ight}\}$) exhibit the longitudinal/lateral tire forces acting on each wheel.

To describe the longitudinal and lateral tire forces generated from the friction between tire and road surface, different tire force models have been proposed [16]. In this paper, the Brush tire model in Ref. [17] is applied. The expression of the longitudinal and lateral tire forces $F_{x,y}$ at each wheel reads

$$f = \sqrt{C_x^2\left(\frac{s}{s+1}\right)^2 + C_y^2\left(\frac{\tan\alpha}{s+1}\right)^2} \tag{5}$$

$$F = \begin{cases} f - \dfrac{1}{3\mu F_z}f^2 + \dfrac{1}{27\mu^2F_z^2}f^3, & f \le 3\mu F_z\,, \\ \mu F_z, \quad f > 3\mu F_z \end{cases} \tag{6}$$

$$\begin{pmatrix} F_x, & F_y \end{pmatrix} = \left(C_x\left(\frac{s}{s+1}\right)F\Big/f, \quad -C_y\left(\frac{\tan\alpha}{s+1}\right)F\Big/f \right) \tag{7}$$

where $C_{x,y}$ is the longitudinal and cornering tire stiffness of a single tire, μ represents the tire-road friction coefficient, F_z is the vertical tire force, and s and α are separately the tire slip ratio and the tire slip angle.

The vertical tire force F_{zij} acting on each tire can be calculated as [4]

$$\begin{cases} F_{zfl} = m(l_rg - a_xh)/2(l_f + l_r) - l_rma_yh/2l_d(l_f + l_r), \\ F_{zfr} = m(l_rg - a_xh)/2(l_f + l_r) + l_rma_yh/2l_d(l_f + l_r), \\ F_{zrl} = m(l_fg + a_xh)/2(l_f + l_r) - l_fma_yh/2l_d(l_f + l_r), \\ F_{zrr} = m(l_fg + a_xh)/2(l_f + l_r) + l_fma_yh/2l_d(l_f + l_r) \end{cases} \tag{8}$$

with h as the height of CG and g as the gravity constant.

In Eq. (5), the tire slip angles and tire slip ratios read [4]

$$\begin{cases} \alpha_{fl} = \tan^{-1}\left(\dfrac{v_y + l_f\gamma}{v_x - l_d\gamma}\right) - \delta_f, \alpha_{fr} = \tan^{-1}\left(\dfrac{v_y + l_f\gamma}{v_x + l_d\gamma}\right) - \delta_f, \\ \alpha_{rl} = \tan^{-1}\left(\dfrac{v_y - l_r\gamma}{v_x - l_d\gamma}\right), \alpha_{rr} = \tan^{-1}\left(\dfrac{v_y - l_r\gamma}{v_x + l_d\gamma}\right) \end{cases} \tag{9}$$

$$\begin{cases} s_{fl} = \dfrac{R_w\omega_{fl} - v_{xfl}}{\max\left(R_w\omega_{fl}, \quad v_{xfl}\right)}, \quad s_{fr} = \dfrac{R_w\omega_{fr} - v_{xfr}}{\max\left(R_w\omega_{fr}, \quad v_{xfr}\right)}, \\ s_{rl} = \dfrac{R_w\omega_{rl} - v_{xrl}}{\max\left(R_w\omega_{rl}, \quad v_{xrl}\right)}, \quad s_{rr} = \dfrac{R_w\omega_{rr} - v_{xrr}}{\max\left(R_w\omega_{rr}, \quad v_{xrr}\right)} \end{cases} \tag{10}$$

where R_w is the effective rolling radius of each wheel, $\omega_{\{i,j\}}$ is the wheel spinning angular velocity, and $v_{x\{i,j\}}$ represents the longitudinal velocity at each wheel's center, whose expressions can be shown as [4]

$$\begin{cases} v_{xfl} = (v_y + l_f\gamma)\sin(\delta_f) + (v_x - l_d\gamma)\cos(\delta_f), \\ v_{xfr} = (v_y + l_f\gamma)\sin(\delta_f) + (v_x + l_d\gamma)\cos(\delta_f), \\ v_{xrl} = v_x - l_d\gamma, \\ v_{xrr} = v_x + l_d\gamma \end{cases} \tag{11}$$

Equations (1)–(11) constitute the whole dynamics of the system. An equivalent compact form can be represented as

$$\begin{cases} \dot{\boldsymbol{\zeta}}(t) = f(\boldsymbol{\zeta}(t), u(t)), \\ \boldsymbol{\eta}(t) = h(\boldsymbol{\zeta}(t), u(t)) \end{cases} \tag{12}$$

with $\boldsymbol{\zeta} = [v_y, v_x, \psi, \gamma, Y, X]^{\mathrm{T}}$, $\boldsymbol{\eta} = [\psi, Y, \alpha_{fl}, \alpha_{fr}, \alpha_{rl}, \alpha_{rr}]^{\mathrm{T}}$, $u = \delta_f$ as separately the state vector, output vector as well as the unique system input. In the output vector $\boldsymbol{\eta}$, the first two items, i.e., the vehicle yaw angle ψ and the ordinate of vehicle's CG Y, constitute the variables to be predicted and controlled toward their referential values within the prediction horizon, and the tire sideslip angles of the front-left, front-right, rear-left, and rear-right wheels, which also need to be predicted within the prediction horizon as the constrained variables on which soft constraints were imposed to ensure the stability of the vehicle. Detail on this soft constraint will be revealed in MPC controller design in Sec. 2.2.

To apply a standard MPC, Eq. (12) needs to be discretized and successively linearized online to produce an approximated linear-time-varying discrete system [4], as

$$\begin{cases} \boldsymbol{\zeta}(k+1) = \mathbf{A}_{k,t}\boldsymbol{\zeta}(k) + \mathbf{B}_{k,t}u(k) + \mathbf{d}_{k,t}, \\ \boldsymbol{\eta}(k+1) = \mathbf{C}_{k,t}\boldsymbol{\zeta}(k) + \mathbf{D}_{k,t}u(k) + \mathbf{e}_{k,t}, \\ u(k) = u(k-1) + \Delta u(k) \end{cases} \tag{13}$$

with

$$\begin{cases} \mathbf{A}_{k,t} = \left.\dfrac{\partial f}{\partial \xi}\right|_{\zeta_t, u(k-1)}, \quad \mathbf{B}_{k,t} = \left.\dfrac{\partial f}{\partial u}\right|_{\zeta_t, u(k-1)}, \\ \mathbf{C}_{k,t} = \left.\dfrac{\partial h}{\partial \xi}\right|_{\zeta_t, u(k-1)}, \quad \mathbf{D}_{k,t} = \left.\dfrac{\partial h}{\partial u}\right|_{\zeta_t, u(k-1)} \end{cases} \tag{14}$$

where $k = t \ldots t + N - 1$, and $\mathbf{d}_{k,t}$, $\mathbf{e}_{k,t}$ correspond to the linearization residual items.

2.2 Model Predictive Control Controller Design.

Grounded on the discrete system (13), a constrained optimization problem can be formulated as

$$\min_{\Delta \mathbf{U}_t, \boldsymbol{\varepsilon}} J(\boldsymbol{\zeta}(t), \Delta\mathbf{U}_t, \boldsymbol{\varepsilon}) \tag{15}$$

such that

$$\begin{cases} \boldsymbol{\zeta}(k+1) = \mathbf{A}_{k,t}\boldsymbol{\zeta}(k) + \mathbf{B}_{k,t}(u(k-1) + \Delta u(k)) + \mathbf{d}_{k,t}, \\ \boldsymbol{\eta}(k+1) = \mathbf{C}_{k,t}\boldsymbol{\zeta}(k) + \mathbf{D}_{k,t}(u(k-1) + \Delta u(k)) + \mathbf{e}_{k,t}, \\ u_{\min} \le u \le u_{\max}, \\ \Delta u_{\min} \le \Delta u \le \Delta u_{\max}, \\ \alpha_{i,j\min} - \varepsilon_{i,j} \le \alpha_{i,j} \le \alpha_{i,j\max} + \varepsilon_{i,j} \quad i \in \{f, r\}, j \in \{l, r\} \end{cases} \tag{16}$$

with

$$J(\boldsymbol{\zeta}(t), \Delta\mathbf{U}_t, \boldsymbol{\varepsilon}) = \sum_{i=1}^{H_p} \|\hat{\mathbf{h}}_{t+i,t} - \mathbf{h}^*_{t+i,t}\|_Q^2 + \sum_{j=0}^{H_c-1} \|u_{t+j,t}\|_S^2 + \sum_{j=0}^{H_c-1} \|\Delta u_{t+j,t}\|_R^2 + \sum_{j=\mathrm{r\{ight\}}}^{\mathrm{l\{eft\}}} \sum_{i=\mathrm{f\{ront\}}}^{\mathrm{r\{ear\}}} \|\varepsilon_{i,j}\|_\rho^2 \tag{17}$$

Equation (16) summarizes the constraints on the formulated optimization problem, including the inherent system dynamics described in Eq. (13), the constraint on the road front steering angle u, the limit on the increment of the road front steering angle Δu, and the constraints on the tire sideslip angles of front-left, front-right, rear-left, and rear-right wheels $\alpha_{fl}, \alpha_{fr}, \alpha_{rl}, \alpha_{rl}$, used for ensuring the stability of vehicle during path tracking. In order to improve the feasibility of the constrained optimization problem, the constraints on the tire sideslip angles were made soft by adopting slack variables $\varepsilon_{i,j}$.

In the cost function Eq. (17), $\Delta\mathbf{U}_t = [\Delta u_{t,t} \ldots\ldots \Delta u_{t+H_c-1,t}]^{\mathrm{T}}$ represents the optimal front road steering angle increment vector along the control horizon H_c at the time instant t. Then, $\boldsymbol{\varepsilon} = [\varepsilon_{fl}, \varepsilon_{fr}, \varepsilon_{rl}, \varepsilon_{rr}]^{\mathrm{T}}$ represents the slack variable vector of the soft constraint on each tire sideslip angle $\alpha_{i,j}$. $\Delta\mathbf{U}_t = [\Delta u_{t,t} \ldots\ldots \Delta u_{t+H_c-1,t}]^{\mathrm{T}}$ and $\boldsymbol{\varepsilon} = [\varepsilon_{fl}, \varepsilon_{fr}, \varepsilon_{rl}, \varepsilon_{rr}]^{\mathrm{T}}$ constitute together the manipulated variables to be calculated at each time-step. Besides, $\hat{\mathbf{h}}_{t+i,t} = [\hat{\psi}_{t+i,t} \quad \hat{Y}_{t+i,t}]^{\mathrm{T}}$ represents the predictive vehicle yaw angle and CG's ordinate along the prediction horizon H_p, which shall be veered toward the corresponding reference vector $\mathbf{h}^*_{t+i,t}$. Finally, positive-definite weighting matrices Q, S, R, ρ regulate the relative importance of different control objectives.

Hence, the meaning of each item in the cost function Eq. (17) becomes clear: The first term shows the accumulated predictive tracking. The second term and the third term represent the control effort and its changing rate along the control horizon. The last term demonstrates the cost of violating the soft constraints. In addition, there exist two implicit constraints, as $H_p \ge H_c$ and $\Delta u_{t+j,t} = 0, \forall j \ge H_c$.

After solving the constrained optimization problem described in Eqs. (15)–(17), the first element of $\Delta\mathbf{U}_t$ will be used to construct the sole controller output at the time instant t, as

$$u(t) = u(t-1) + \Delta u_{t,t} \tag{18}$$

At the next time-step $t+1$, the same constrained optimization problem will be solved again with updated state measurements $\boldsymbol{\zeta}(t+1)$.

Remark. Compared with the original MPC controller in Ref. [4], the number of slack variables $\varepsilon_{i,j}$ is augmented from one to four to better reflect the discrepancy among sideslip angles of each tire.

3 Insignificant Parameter Settings

There are a bunch of parameters to be tuned in the MPC controller, including the prediction horizon H_p, the control horizon H_c, the sampling period T_s that is implicitly used for the system discretization in Eq. (13), the weighting matrix Q, S, R, ρ, the upper and lower bounds of control/control changing-rate $U_{\max}, U_{\min}, \Delta U_{\max}, \Delta U_{\min}$, as well as the upper and lower bounds on the soft constraints $\alpha_{i,j\max}, \alpha_{i,j\min}$.

In this paper, more attention will be paid to the parameters directly affecting both the controller performance and the computational load. As indicated in Refs. [18] and [19], the performance and computational load of a digitally implemented MPC controller are essentially influenced by the prediction horizon H_p, the control horizon H_c and the sampling period T_s. Actually, H_p and H_c govern the complexity of the constrained optimization problem to be solved at each time instant. As H_p and H_c become longer, both the number of manipulated variables and the number of constraints boost, which substantially increases the CPU execution time to find the optimal solution, and this execution time is rigorously bounded by the sampling period T_s. Accordingly, H_p, H_c, and T_s are regarded as the *significant parameters* in this paper due to their conspicuous double-sided effects on both controller performance and computational burden. All other parameters are regarded as *insignificant parameters*.

To make the conclusion of this paper more universally valid, instead of arbitrarily fixing these insignificant parameters as constants, a group of methodical parameter-tuning strategies will be used to assign them with reasonable values.

3.1 Output Weighting Matrix. The output-weighting matrix Q indicates the relative importance of different tracking objectives. A larger weight on a specific tracked variable implies that more control effort will be conducted to minimize the corresponding tracking error. A popular approach to fix Q is to take variable scaling and normalization into account [20]. Consequently, the matrix Q is fixed as

$$Q = \begin{bmatrix} 1/\psi_{\max} & \\ & 1/Y_{\max} \end{bmatrix} \tag{19}$$

with $\psi_{\max}$ and $Y_{\max}$ corresponding to the maximal heading angle and maximal ordinate of the tracked reference path in the inertial coordinate system.

3.2 Input/Input Rate Weighting Matrix. Input weighting matrix S reflects the importance of the actuator output. A higher S will decrease actuator's control effort. Meanwhile, input rate weighting matrix R influences the fluctuation of actuator output and a higher R renders the actuator output smoother. Similarly, to normalize control effort, the input weighting matrix as well as the input changing-rate weighting matrix is separately fixed as

$$R = 1/(\max(\dot{\delta}_f)T_s), \quad S = 1/\max(\delta_f) \tag{20}$$

where $\max(\delta_f)$ and $\max(\dot{\delta}_f)$ are individually the mechanical limit of the front road steering angle and the maximal steering angular velocity of δ_f, with T_s as the sampling period. Further, Eq. (20) indeed implies that the upper and lower bounds of the control/control changing rate shall be consequently chosen as

$$\begin{cases} u_{\max} = \max(\delta_f), & \Delta u_{\max} = \max(\dot{\delta}_f)T_s, \\ u_{\min} = -\max(\delta_f), & \Delta u_{\min} = -\max(\dot{\delta}_f)T_s \end{cases} \tag{21}$$

3.3 Bounds on the Soft Constraints. Soft constraints on each wheel's sideslip angle are crucial to establish the stability of the vehicle. In Ref. [4], the upper and lower bounds of $\alpha_{i,j}$ were constantly fixed as ± 3 deg. Virtually, the allowable limits of tire slip angles shall be determined by vehicle states, tire's proper characteristics, as well as road condition. Therefore, adaptive constraints of tire sideslip angles inspired by Katriniok and Abel [21] will be adopted in this paper. According to the Brush tire model described in Eqs. (5)–(7), the lateral tire force will saturate if the absolute value of slip angle reaches above a threshold. Hence, the

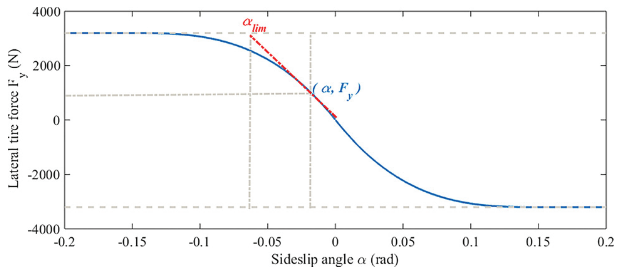

Fig. 1 Bounds on tire slip angle

bounds of tire slip angle $\alpha_{i,j\max}, \alpha_{i,j\min}$ must ensure that the lateral tire force of each tire is restricted within a region where its extreme value is not totally saturated in the prediction horizon. Consequently, the procedure to find $\alpha_{i,j\max}, \alpha_{i,j\min}$ can be summarized as:

(1) Linearize the tire force curve around the current tire slip angle.
(2) Then, find the intersection between the tangent and the maximum of the lateral tire force if the current sideslip angle is negative.
(3) The abscissa of this intersection point, named $\alpha_{\lim}$, represents the lower-bound $\alpha_{\min}$ of the tire sideslip angle within the prediction horizon, and the upper-bound $\alpha_{\max}$ of the tire sideslip angle is simply the opposite number of the lower bound.

On the contrary, if the current sideslip angle is positive, then just find the intersection between the tangent and the minimum of the lateral tire force, whose abscissa will become the upper bound $\alpha_{\max}$ of the tire sideslip angle within the prediction horizon, and the opposite number of this value will become the lower bound $\alpha_{\min}$ of the tire sideslip angle. Figure 1 illustrates the lower bound of the tire sideslip angle $\alpha_{\lim} = \alpha_{\min}$ with parameters fixed as: $s = 0$, $\mu = 0.8$, $F_z = 4000\,\text{N}$, $C_y = 62,700\,\text{N/rad}$.

Remark. δ_f serves as the unique output of the MPC controller in this paper. Therefore, the tire slip ratio of each wheel $s_{i,j}$ remains zero.

3.4 Slack Variable Weighting Matrix. As indicated in Eq. (16), the slack variables $\varepsilon_{i,j}$ were introduced in the soft constraints on each tire slip angle to ensure the feasibility of the constrained optimization problem. In general, it is preferable that $\varepsilon_{i,j}$ could remain as small as possible to render these constraints, which are pivotal to ensure the stability of the vehicle, still effective. Hence, the slack variable weighting factor $\rho_{i,j}$ of each tire is defined as

$$\rho_{i,j} = \left| \frac{\partial F_{yi,j}}{\partial \alpha_{i,j}} \right|^{-1}_{\alpha_{i,j} = \alpha_{i,j\lim}} \tag{22}$$

which is the absolute value of the reciprocal of the partial derivative of the lateral tire force $F_{yi,j}$ with respect to the tire sideslip angle $\alpha_{i,j}$, evaluated at the adaptive bound $\alpha_{i,j\lim}$ of the current tire sideslip angle $\alpha_{i,j}$. As demonstrated in Sec. 3.3, if the current sideslip angle $\alpha_{i,j} > 0$, the adaptive bound $\alpha_{i,j\lim}$ will be the upper-bound $\alpha_{i,j\max} > 0$. Instead, if the current sideslip angle $\alpha_{i,j} < 0$, the adaptive bound $\alpha_{i,j\lim}$ will be the lower-bound $\alpha_{i,j\lim} = \alpha_{i,j\min} < 0$.

The relationship between $\rho_{i,j}$ and $\alpha_{i,j}$ is shown in Fig. 2.

According to Fig. 2, the function of using such a dynamic penalty $\rho_{i,j}$ is twofold. First, if the magnitude of a tire sideslip angle $\alpha_{i,j}$ is small, the weight on the slack variable is negligible, which in turn encourages more control effort to achieve a better path-tracking result. Instead, if the tire sideslip angle approaches to the critical value corresponding to a saturated lateral tire force, then

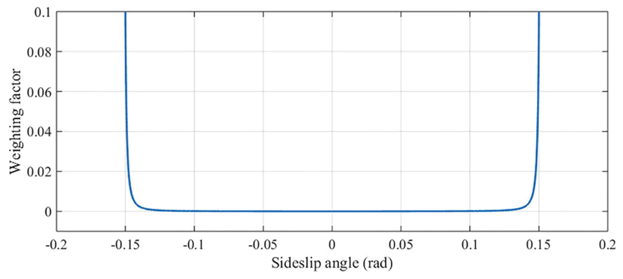

Fig. 2 Weight on slack variable

Table 1 Simulation configurations

Symbol	Value	Unit
l_f	1.232	m
l_r	1.468	m
l_d	0.77	m
h	0.54	m
m	1723	kg
R_w	0.3	m
I_z	1960	kg/m^2
C_x	66,900	N
C_y	62,700	N/rad
$\psi_{\max}$	0.1489	rad
$Y_{\max}$	2.8921	m
$\max\dot{\delta}_f$	17.5	deg/s
$\max\delta_f$	20	deg
μ	0.8	\

the weight on the slack variable drastically rises into infinity, which prevents the sideslip angle $\alpha_{i,j}$ further enlarging and consequently improves the stability of the vehicle.

3.5 Verification. In Secs. 3.1–3.4, all the insignificant parameters are methodically fixed and only three significant parameters, namely, the prediction horizon H_p, the control horizon H_c, and the sampling period T_s are left for tuning. Before entering into the next phase to illustrate the online selection strategy with respect to these three significant parameters, it is necessary to verify the efficiency of the proposed tuning methods for these insignificant parameters. Thus, a path-tracking simulation under a typical double lane-change (DLC) scenario on the MATLAB-CARSIM conjoint simulation platform was conducted.

CARSIM is a vehicle dynamics simulation software, widely used for vehicular system analysis, controller design, and performance assessment. To effectuate the verification, vehicle configurations, e.g., physical size, mass, tire characteristics, yaw inertial, etc., were first set up. Then, the test maneuver of the default DLC scenario in CARSIM was chosen, with a constant tire-road friction coefficient as 0.8. The longitudinal velocity remained as 30 m/s during the simulation. After the configurations on vehicle and test maneuver, the CARSIM model was imported into Simulink for the conjoint simulation.

The vehicle configurations and other constant parameters used for the simulation are listed in Table 1.

The three significant parameters were arbitrarily fixed as: $H_p = 30$, $H_c = 10$, $T_s = 0.05$. The vehicle longitudinal velocity remained 108 km/h during the simulation. Figure 3 shows the path-tracking result, Fig. 4 presenting the front road steering angle as well as its increment value, and Fig. 5 demonstrating the sideslip angle of each tire.

According to Figs. 3–5, conclusion can be drawn that the proposed systematic insignificant parameter tuning strategies can successfully realize the path-tracking task without violating neither the hard constraints on actuators nor the soft constraints on sideslip angles.

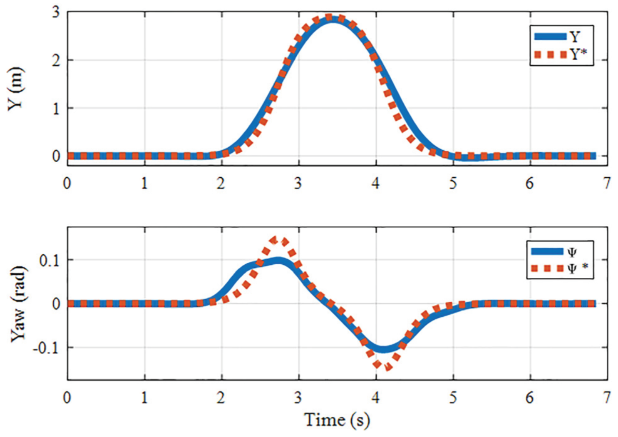

Fig. 3 Path-tracking result

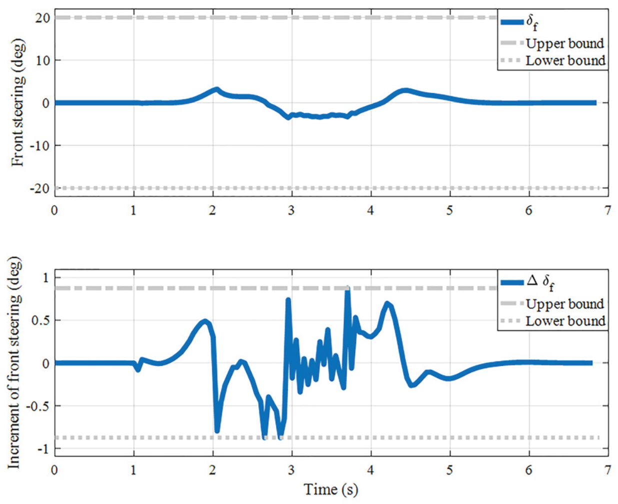

Fig. 4 Front steering angle

4 Performance Map

Fixing all the insignificant parameters adequately decreases the degrees-of-freedom for MPC tuning. In this section, a look-up table-based online parameter selection method will be illustrated to set the rest three significant parameters. As mentioned in Sec. 1, the selected significant parameters should lead to the highest attainable tracking performance while considering the minimal stability requirement under a given CPU computational capacity. Hence, all three aspects, namely, tracking performance, vehicle stability, and computational load, need to be defined and quantified to show the overlapping effects of the significant parameters.

4.1 Performance Index Definition. *Tracking index*, *stability index*, and *computational load index* form the three fundamental performance indices.

To begin with, *tracking index* reflects the capacity of a vehicle to maintain itself along the centerline of the reference path in order to stay away from road boundaries. In other words, the larger the minimal distance between vehicle body and road boundaries, the higher the tracking performance shall be. Therefore, the concept of the safe driving envelop in Ref. [22] is utilized here to define the tracking performance. Instead of treating vehicle as a mass point at its CG, vehicle's physical dimension is taken into consideration when the minimal distance between vehicle body and road boundaries is determined. Precisely speaking, the ordinates of four wheels in an inertial coordinate can be calculated as

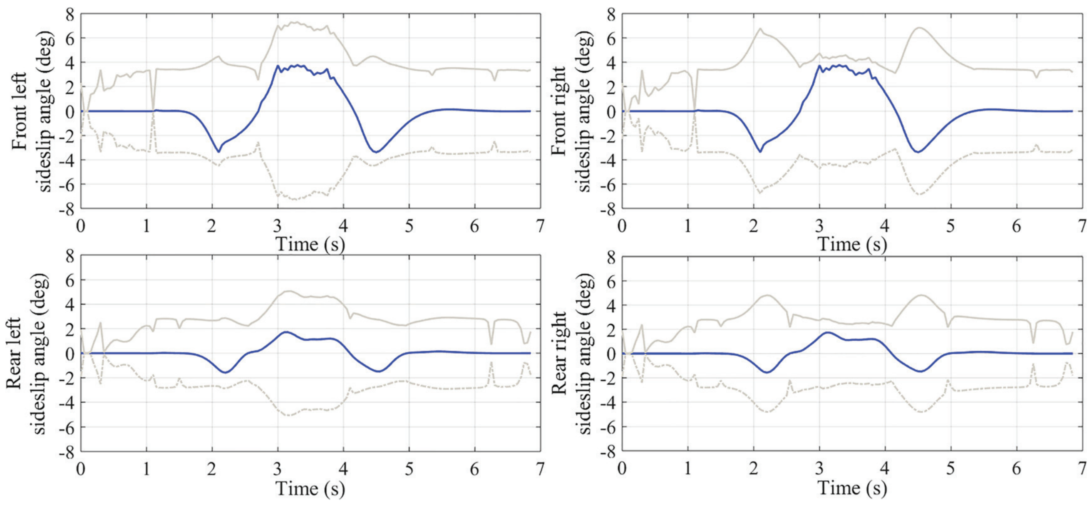

Fig. 5 Four tire sideslip angles

$$\begin{cases} Y_{fl}(t) = Y_{cg}(t) + \sqrt{l_f^2 + l_d^2}\sin(\Delta\psi(t) + \tan^{-1}(l_d/l_f)), \\ Y_{fr}(t) = Y_{cg}(t) - \sqrt{l_f^2 + l_d^2}\sin(-\Delta\psi(t) + \tan^{-1}(l_d/l_f)), \\ Y_{rl}(t) = Y_{cg}(t) + \sqrt{l_r^2 + l_d^2}\sin(-\Delta\psi(t) + \tan^{-1}(l_d/l_r)), \\ Y_{rr}(t) = Y_{cg}(t) - \sqrt{l_r^2 + l_d^2}\sin(\Delta\psi(t) + \tan^{-1}(l_d/l_r)) \end{cases} \quad (23)$$

where $Y_{cg}(t) = Y(t)$ indicates the ordinate of vehicle's CG and $\Delta\psi(t)$ shows vehicle's yaw error with respect to the reference path. Further, the ordinate of upper- and lower-road boundaries can be formulated as

$$Y_{\text{upper}}(t) = Y^*(t) + \frac{W}{2\cos(\psi^*(t))}, \quad Y_{\text{lower}}(t) = Y^*(t) - \frac{W}{2\cos(\psi^*(t))} \quad (24)$$

with $Y^*(t)$ and $\psi^*(t)$ representing the reference centerline's ordinate and heading angle and W as the width of the reference lane, fixed as 3.6 m [22]. Consequently, by combing Eqs. (23) and (24), the minimum margin between vehicle body and path boundaries at each time instant (noted as $\text{MT}(t)$) can be calculated as

$$\text{MT}(t) = \min(Y_{\text{upper}}(t) - \max(Y_{fl}(t), Y_{fr}(t), Y_{rl}(t), Y_{rr}(t)), \min(Y_{fl}(t), Y_{fr}(t), Y_{rl}(t), Y_{rr}(t)) - Y_{\text{lower}}(t)) \quad (25)$$

A necessary and sufficient condition to avoid collision between vehicle and road boundaries is: $\text{MT}(t) > 0, \forall t \geq 0$ and the maximal possible margin: $\text{MT}_{\max} = W/2 - l_d$ can be obtained only when vehicle's CG lays exactly on the centerline of the reference path along with a zero yaw error. Clearly, $\text{MT}_{\max}$ can have different value according to vehicle width l_d. To normalize $\text{MT}(t)$, a hyperbolic tangent function was utilized to limit the value of $\text{MT}(t)$ within [0, 1], which can be seen in Fig. 6.

For clarity, the *normalized* minimum margin between vehicle body and path boundaries is denoted as $\text{MT}_n(t)$. Therefore, a larger $\text{MT}_n(t)$ implies a better path-tracking performance. Finally, the overall tracking index (TI) of a complete simulation maneuver can be defined as

$$\text{TI} = \frac{\int_0^{T_f} \text{MT}_n(t)dt}{T_f} \quad (26)$$

with T_f as elapse of simulation time.

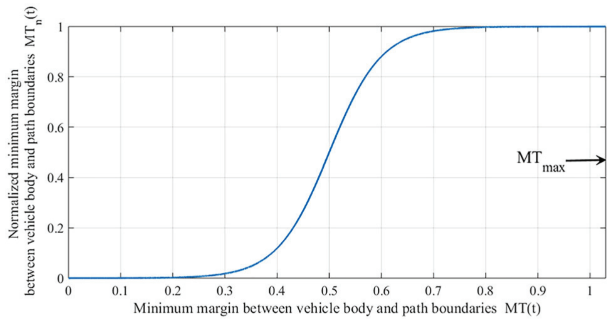

Fig. 6 Hyperbolic tangent function

Afterward, *stability index* reflects the margin between vehicle's current state from the critical states under which the vehicle may spin, drift, or roll over. The stability of a vehicle can be quantitatively assessed through various approaches, such as using the tire grip margin [23]. Nonetheless, stability can be more straightforwardly indicated through observing both the vehicle body sideslip angle $\beta = \tan^{-1}(v_y/v_x)$ as well as the yaw rate γ. Empirical thresholds about body sideslip angle and yaw rate, which may lead to a critical instability, are [24]

$$\begin{cases} \beta^* = \tan^{-1}(0.02\mu g), \\ \gamma^* = \dfrac{0.85\mu g}{v_x} \end{cases} \quad (27)$$

Consequently, the stability margin at each time instant, noted as $\text{MS}(t)$, can be defined as

$$\text{MS}(t) = \min\left(\min\left(1 \pm \frac{\beta(t)}{\beta^*}\right), \min\left(1 \pm \frac{\gamma(t)}{\gamma^*}\right)\right) \quad (28)$$

Naturally, $\text{MS}(t) > 0$ implies that the absolute values of both the body sideslip angle and the yaw rate are under their respective critical thresholds. In addition, the most stable case occurs when both $\beta(t)$ and $\gamma(t)$ remain zero, which gives birth to the maximal stability margin $\text{MS}_{\max} = 1$. To be coherent with $\text{MT}(t)$ in Eq. (25), a similar hyperbolic tangent function was also applied on $\text{MS}(t)$ to produce the normalized stability margin $\text{MS}_n(t)$. Still, a larger $\text{MS}_n(t)$ suggests a more stable condition. As a final point, the overall stability index (SI) for a complete simulation maneuver can be formulated as

$$\mathrm{SI} = \frac{\int_0^{T_f} \mathrm{MS}_n(t)dt}{T_f} \tag{29}$$

with T_f being the end of simulation time.

Remark. Compared with $\mathrm{MT}_n(t)$ and $\mathrm{MS}_n(t)$, which concentrate on the vehicle behavior at each time instant, both TI and SI are defined from an overall average point of view. For example, $\mathrm{SI} \geq 0.5$ does not imply that $|\beta(t)| \leq (\beta^*/2), \forall t \in [0, T_f]$, neither $|\gamma(t)| \leq (\gamma^*/2), \forall t \in [0, T_f]$. Instead, it is certainly possible that at some moments, when the vehicle runs straightforward for example, the normalized stability margin $\mathrm{MS}_n(t)$ of the vehicle can be high enough to reach its maximum. However, when negotiating a sharp curve, $\mathrm{MS}_n(t)$ can be quite low. But all in all, the integrated mean of the time-varying variable $\mathrm{MS}_n(t)$ over the complete time interval $[0, T_f]$ satisfies $\mathrm{SI} \geq 0.5$.

Eventually, the computational load index (CI) of the MPC controller is defined as [19]

$$\mathrm{CI} = T_c / T_s \tag{30}$$

where T_c is the total CPU execution time to find the optimal solution of the constrained optimization problem in Eqs. (15)–(17). Intuitively, a higher CI implies a higher CPU load entailed from the MPC controller and the upper threshold of CI is one. To at least partially overcome the execution time fluctuation issue due to the time-sharing nature of Windows® operating system, execution times T_c were measured fifty times and the minimal one was used to calculate the final computational load index in Eq. (30).

4.2 Performance Map Generation. After defining all three fundamental performance indices, extensive simulations with various combinations of H_p, H_c, and T_s were effected to generate three performance maps with respect to the three performance indices: the tracking index TI, the stability index SI, and the computational load index CI. Precisely speaking, for a given triplet $\{H_p, H_c, T_s\}$, a DLC scenario identical to the one in Sec. 3.5 was used to generate the three fundamental performance indices. The range of H_c used for simulations expanded from one to nine with an increment as one, and the range of H_p satisfied $H_c \leq H_p \leq 45$ with an identical increment as one. Finally, the range of T_s for a given combination of H_p and H_c started from 0.01 s and ended at 0.05 s with an increment of 0.005 s. A total number of 3321 simulations were executed on a standard desktop with an Intel i7-4790 processor whose clock rate was 3.60 GHz. The optimization solver was the default MATLAB Quadratic Programming solver with an "active-set" method.

Remark. As indicted in Eq. (16), the proposed MPC controller did not explicitly impose constraints on Y, ψ. The reason to remove these two state constraints can be summarized into two points: First, as mentioned in the introduction, instead of designing new MPC controllers, this paper emphasizes on a look-up table-based MPC parameter-tuning approach, which explicitly taking the CPU load constraint into account. Thus, the widely recognized LTV-MPC controller in Ref. [4] was directly used here with minor improvements. In fact, the original controller in Ref. [4] did not impose constraints on the tracked outputs Y, ψ neither, since the tracking errors with respect to the CG's ordinate Y and the yaw angle ψ were contained in the cost function (17). Second, the proposed MPC parameter-tuning method, which will be shown in Sec. 5, is able to negotiate the intricate conflicts among optimizing the path-tracking performance, maintaining the vehicle stability and obeying the computational burden constraint. To do so, three performance maps were generated with respect to the path-tracking result, the vehicle stability, and the computational burden. If both Y and ψ were explicitly constrained during MPC formulation, the performance map of the path-tracking result will become trivial. As a consequence, the influence of different combinations of the three significant parameters $\{H_p, H_c, T_s\}$ on the tracking performance as well as the vehicle stability would not be fully demonstrated anymore, which will not only hide the inherent tradeoff between pursuing a higher path-tracking performance and maintaining a better vehicle stability, but also undermine the value of the proposed parameter-tuning approach. Nonetheless, without direct constraints on ψ or Y, under some specific combinations of H_p, H_c, and T_s, collision between vehicle and road boundaries may occur during simulations. In order to maintain the completeness of the tracking index map, instead of directly disregarding these parameter settings, the minimum margin between vehicle body and path boundaries $\mathrm{MT}(t)$ was truncated to lie within $[0, W/2 - l_d]$ before applying the

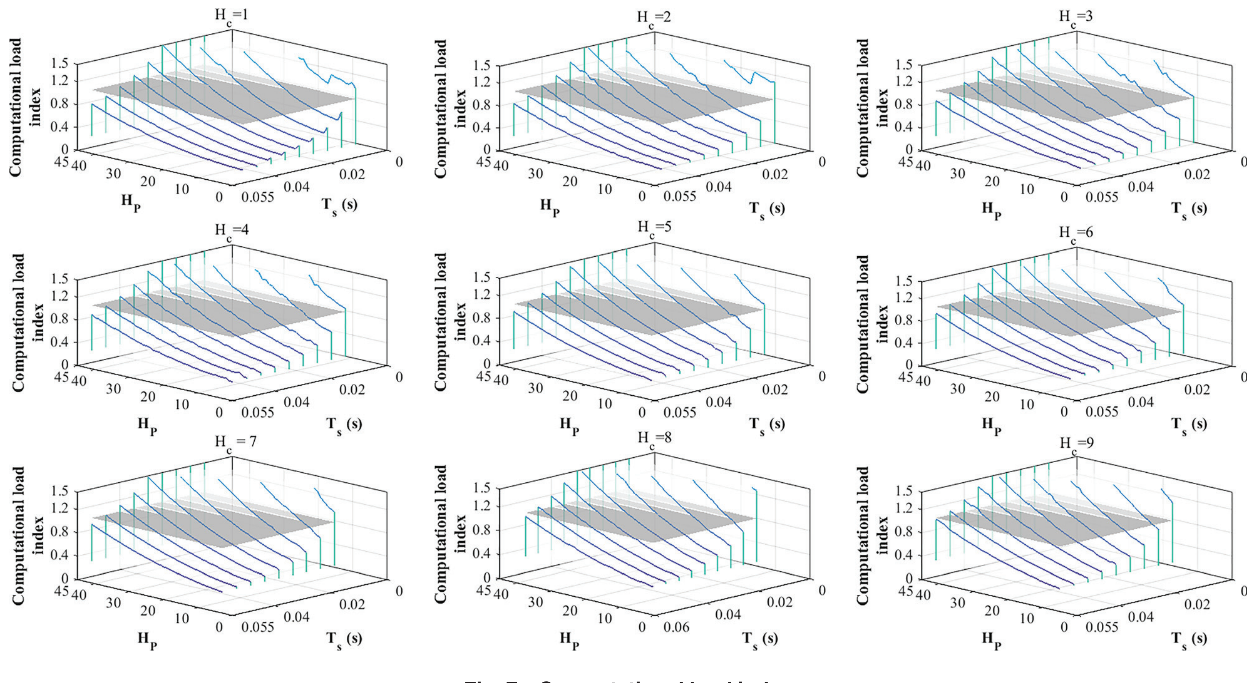

Fig. 7 Computational load index

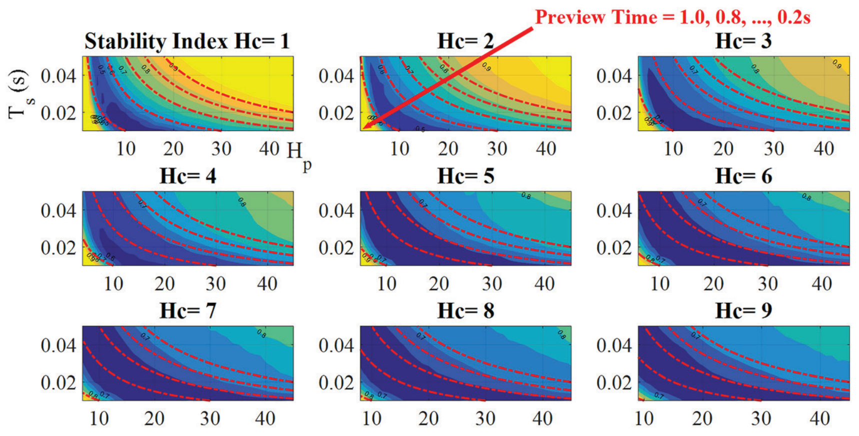

Fig. 8 Stability index

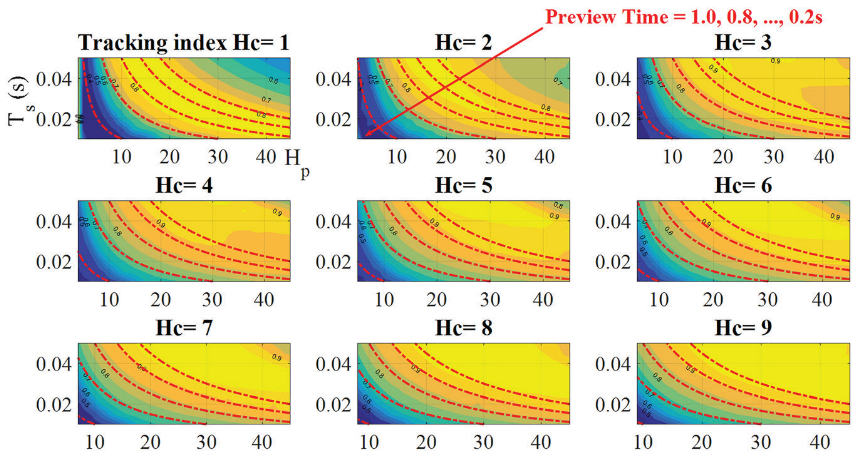

Fig. 9 Tracking index

hyperbolic tangent function in Fig. 6 to produce the *normalized* minimum margin between vehicle body and path boundaries $\text{MT}_n(t)$. Likewise, the stability margin $\text{MS}(t)$ in Eq. (28) was also truncated to lie within $[0, 1]$ before the normalization.

Figures 7, 8, and 9 sequentially demonstrate the computational load index CI, the stability index SI, and the tracking index TI under various combinations of the three significant parameters $\{H_p, H_c, T_s\}$.

To begin with, Fig. 7 shows the simulation result of the computational load index, with the dashed gray plane corresponding to the upper threshold of the computational load index, as one.

Clearly, as prediction horizon H_p enlarges, the computational load rises accordingly. While as the sampling time T_s increases, the computational load decreases. If the computational load index CI is beyond one, real-time control becomes infeasible since the optimal action cannot be found within the sampling period. For a given control horizon H_c, the percentage of feasible combinations of H_p and T_s, which ensure the computational load index less or equal to one, is defined as the *computational feasibility rate*. Figure 10 shows the trend of this rate as H_c increases.

Figure 10 indeed suggests that the control horizon H_c has a fundamental impact on the computational load index, because an obvious drop of computational feasibility rate appears as H_c increases.

Figure 8 shows the simulation result of the stability index (SI). Instead of using the same three-dimensional waterfall-style figure as the case in Fig. 7, a contour plot was exploited to show the influences of the three significant parameters on the stability index. In Fig. 8, the brighter part of each subplot indicates a higher stability index and a darker part reflects a lower stability index. Further, the dashed lines added on each subplot represent the different range of preview time $T_{\text{pre}} = H_p T_s$.

Some general conclusions can be drawn from Fig. 8: As the preview time T_{pre} increases, the stability index improves accordingly, regardless of what the control horizon H_c is. This phenomenon is consistent with the classical MPC theory that a longer preview time will reduce the oscillation of the closed-loop system. On the contrary, as H_c enlarges, the maximal attainable stability index drops. This is because a longer control horizon will make the controller more aggressive, which in turn affects the stability index.

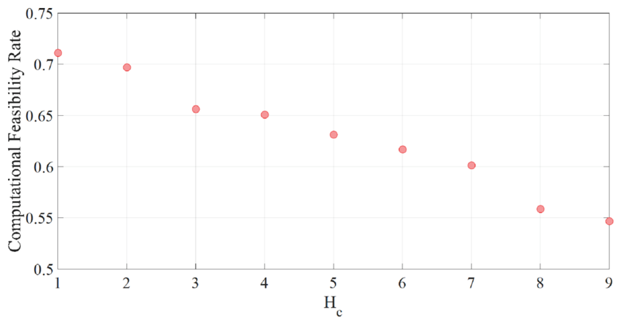

Fig. 10 Computational feasibility rate

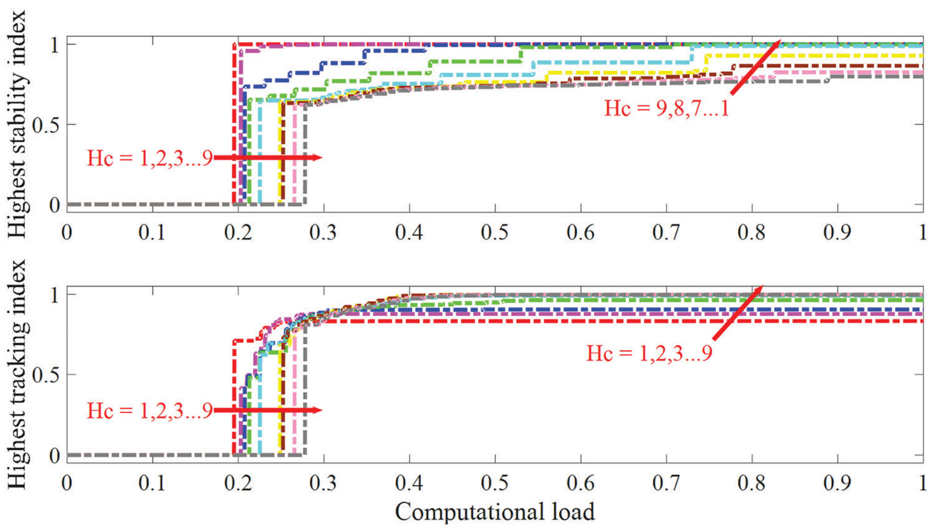

Fig. 11 Confliction between stability and tracking indices

Figure 9 demonstrates the simulation result of the tracking index (TI).

In contrast to Fig. 8, as the preview time T_{pre} increases, the tracking index first goes through an improved phase, followed by a degraded phase if the preview time becomes too large. The improvement at the beginning comes from the fact that the preview time should be larger than the system settling time T_{set} in order to take all the dynamics of the system into consideration before the controller makes the decision.

To obtain the settling time of a dynamic system, the characteristic equation of the transfer function needs to be found a priori. Even though the transfer function from the unique system input δ_f to the tracking index TI in Eq. (26) cannot be obtained, the transfer function from δ_f to yaw rate γ can act as an alternative since it demonstrates the inherent handling characteristics of the vehicle.

A two degrees-of-freedom linear bicycle model [25] shows

$$\begin{cases} \dot{\beta} = \dfrac{2C_y(\delta_f - \beta - l_f\gamma/v_x) + 2C_y(-\beta + l_r\gamma/v_x)}{mv_x} - \gamma, \\ \dot{\gamma} = \dfrac{2C_y(\delta_f - \beta - l_f\gamma/v_x)l_f - 2C_y(-\beta + l_r\gamma/v_x)l_r}{I_z} \end{cases} \tag{31}$$

Based on Eq. (31), the transfer function from front road steering angle δ_f to vehicle yaw rate γ in the *s-domain* can be calculated as

$$\frac{\gamma(s)}{\delta_f(s)} = \frac{a_1 s + a_0}{s^2 + b_1 s + b_0} \tag{32}$$

with

$$\begin{cases} a_1 = 2C_y l_f / I_z, \\ a_0 = 4C_y^2(l_f + l_r)/mv_x I_z, \\ b_1 = 2C_y(l_f^2 + l_r^2)/v_x I_z + 4C_y/mv_x, \\ b_0 = 4C_y^2(l_f + l_r)^2 - 2mv_x^2 C_y(l_f - l_r)/mv_x^2 I_z \end{cases} \tag{33}$$

As a result, the settling time for which the response remains within ±2% of the final steady-state value can be approximately calculated as

$$T_{\mathrm{set}} \approx \frac{4}{0.5 * b_1} = \frac{8}{b_1} \tag{34}$$

By substituting the parameter values in Table 1 into the expression of b_1 along with a constant $v_x = 30$ m/s, Eq. (34) gives $T_{\mathrm{set}} \approx 0.63$ s. From Fig. 9, the lowest threshold of the preview time, resulting in an overall tracking index TI > 0.8, can roughly be observed as 0.6 s. Thus, the simulation result corresponded well with the theoretical deduction.

On the contrary, if the preview time T_{pre} continues to extend, then two factors will negatively affect the tracking index. First, for a very long prediction horizon, the linearization error caused by the mismatch between the real plant and the inherent model of MPC controller will accumulate considerably. Second, as indicated in Fig. 8, a long preview time will reduce the swiftness of the closed-loop system, leading to a sluggish behavior, which impairs the capacity of the vehicle to follow the fast-changing reference path precisely. However, this negative side effect of a long preview time can be moderately neutralized by a longer control horizon H_c, which can be witnessed by comparing the first ($H_c = 1$) and the last subplot ($H_c = 9$) in Fig. 9.

Keen readers may have already found that the global trend of the stability index (SI) and the tracking index (TI) counteracts with each other as H_c increases. This competing behavior is even better demonstrated in Fig. 11, where the highest attainable stability index/tracking index with respect to the upper threshold of the computational load index is shown.

Various interesting phenomena can be found in Fig. 11. For instance, as the exploitable computational load augments, both the highest tracking index and the highest stability index increase until saturated. Further, H_c has a direct influence on the saturated extremes. A longer control horizon results in an improved apex of the tracking index while a degraded highest stability index. Hence, it is in fact impossible to obtain simultaneously both very high stability index and tracking index even when the computational load index reaches one.

The data underlying Figs. 7, 8, and 9 will serve as the basis of the look-up table for the online significant parameter selection, which will be presented in Sec. 5.

5 Online Parameter Selections

To help a vehicle achieve the highest attainable tracking performance while considering the minimal requirement of stability index $\mathrm{SI}_{\min}$ under a given upper threshold of computational load index $\mathrm{CI}_{\max}$, a straightforward parameter selection algorithm is proposed.

Algorithm Constrained optimal tracking parameter selection

```
Input: $\mathrm{SI}_{\min}, \mathrm{CI}_{\max}$
Output: $H_c, H_p, T_s$
1: $\mathbf{\Lambda} = \mathrm{find}_{T_s,H_p,H_c}(\mathrm{CI} \le \mathrm{CI}_{\max})$
2: if $\mathbf{\Lambda} == \mathrm{O}$ then
3:     $T_s \leftarrow \max(Ts), H_p \leftarrow \min(H_p), H_c \leftarrow \min(H_c)$
4: end if
5: $\mathbf{\Omega} = \mathrm{find}_{T_s,H_p,H_c}(\mathrm{SI} \ge \mathrm{SI}_{\min})$
6: if $\mathbf{\Omega} == \mathrm{O} || \mathbf{\Omega} \cap \mathbf{\Lambda} == \mathrm{O}$ then
7:     $(T_s, \ H_p, \ H_c) \leftarrow \underset{(T_s,H_p,H_c)\in\mathbf{\Lambda}}{\mathrm{find}}(\mathrm{SI} == \max(\mathrm{SI}))$
8: else
9:     $(T_s, \ H_p, \ H_c) \leftarrow \underset{(T_s,H_p,H_c)\in\mathbf{\Lambda}\cap\mathbf{\Omega}}{\mathrm{find}}(\mathrm{TI} == \max(\mathrm{TI}))$
10: end if
```

At the beginning, find all the possible combinations of H_c, H_p, T_s satisfying the computational load constraint. However, if the given available computational index is too low, the configuration of H_c, H_p, T_s is designed to reduce as much the computational load as possible. Instead, find the combinations of H_c, H_p, T_s, which further satisfy the minimum stability index constraint. If such combinations cannot be found due to an unreasonably high stability index requirement, then just find the combination of H_c, H_p, T_s, which renders the stability index as high as possible under the given computational load constraint. Nonetheless, in this case, the path-tracking objective will be totally ignored. Finally, if both the stability index constraint and the computational load index constraint can be met, then find the optimal combination of H_c, H_p, T_s, which leads to the highest tracking index.

To verify the improvement introduced by the algorithm, simulations with a DLC scenario identical to the one in Sec. 3.5 were conducted. An MPC controller with a dynamic parameter setting, named the dynamic MPC, along with another constant parameter setting MPC controller, named the static MPC, was equipped individually, on two identical simulation vehicles with the same configurations in Table 1. Both the dynamic MPC and the static MPC selected their significant parameters according to the proposed constrained optimal-tracking parameter selection algorithm. However, the difference lies in the fact that the available upper threshold of computational load index $\mathrm{CI}_{\max}$ for the dynamic MPC, denoted as $\mathrm{CI}^d_{\max}$, changed along the simulation, while the $\mathrm{CI}_{\max}$ for the static MPC, denoted as $\mathrm{CI}^s_{\max}$, remained as the minimum value of $\mathrm{CI}^d_{\max}$. The minimum stability index $\mathrm{SI}_{\min}$ for both controllers was fixed as 0.4.

Figure 12 demonstrates both $\mathrm{CI}^d_{\max}$ and $\mathrm{CI}^s_{\max}$ as well as the *actual* computational load index of the dynamic MPC (CI^d) and the static MPC (CI^s). Clearly, both the dynamic and the static MPC controller work under the available upper threshold of computational load.

Figure 13 shows the parameter selection results of the dynamic MPC as well as the static MPC.

Clearly, the proposed algorithm can synchronously adapt the three significant parameters according to the available computational load. For instance, as $\mathrm{CI}^d_{\max}$ decreased during 2–4 s, both H_c and H_p of the dynamic MPC shrunk to reduce the computational burden of the constrained optimization problem, while T_s extended to ensure that an optimal solution can be found within the sampling period.

As for the stability constraint, the integrated mean of the normalized stability margin $\mathrm{MS}_n(t)$ as time goes by can be calculated as

$$\mathrm{SI}_{\mathrm{time}}(t) = \frac{\int_0^t \mathrm{MS}_n(\tau)d\tau}{t} \tag{35}$$

According to Eq. (29), the stability index (SI) of a complete double lane change maneuver satisfies

$$\mathrm{SI} = \mathrm{SI}_{\mathrm{time}}(t)|_{t=T_f} \tag{36}$$

with T_f as the end of simulation time.

Figure 14 exhibits $\mathrm{SI}_{\mathrm{time}}(t)$ of both the dynamic MPC controller and the static MPC controller.

As we can see, $\mathrm{SI}_{\mathrm{time}}(t)$ of both MPC controllers remained as one at the outset, and then underwent a degradation phase and finally rose again at the end. This behavior corresponded well with the double lane change maneuver: At the beginning, the vehicle ran straightforwardly on the entry lane, suggesting that both the sideslip angle β and the yaw rate γ were negligible. Afterward, the vehicle entered into the left turn, followed by the side lane and the ensuing right turn. During this stage, both the sideslip angle and the yaw rate went through drastic variations, which certainly reduced the integrated mean of the normalized stability margin.

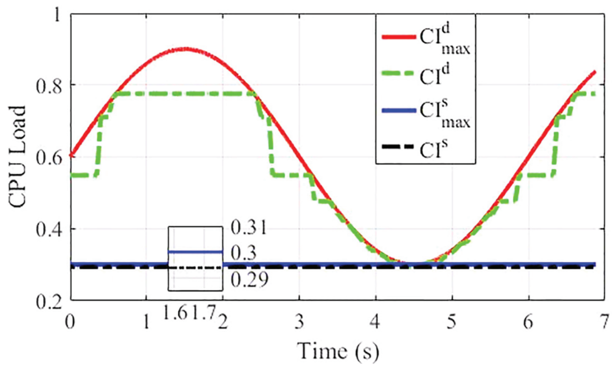

Fig. 12 Computational load index

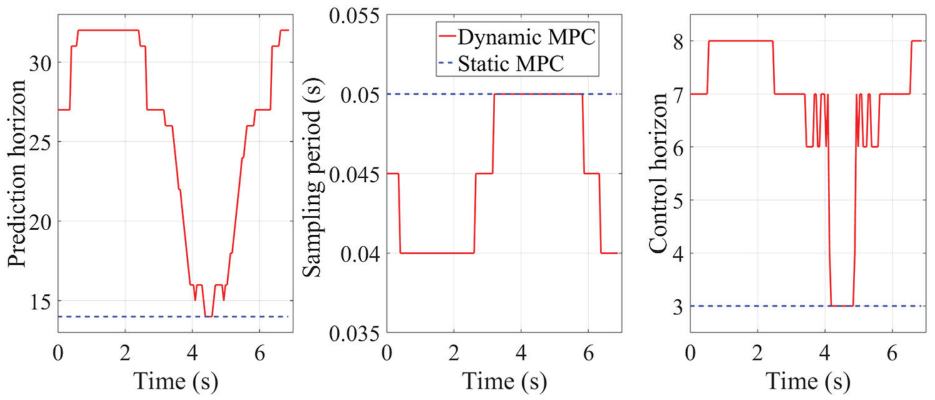

Fig. 13 Constant and dynamic parameter setting

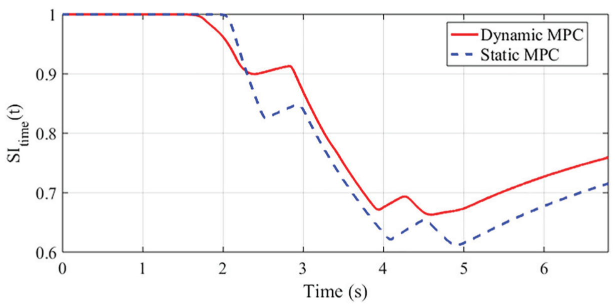

Fig. 14 Integrated mean of the normalized stability margin

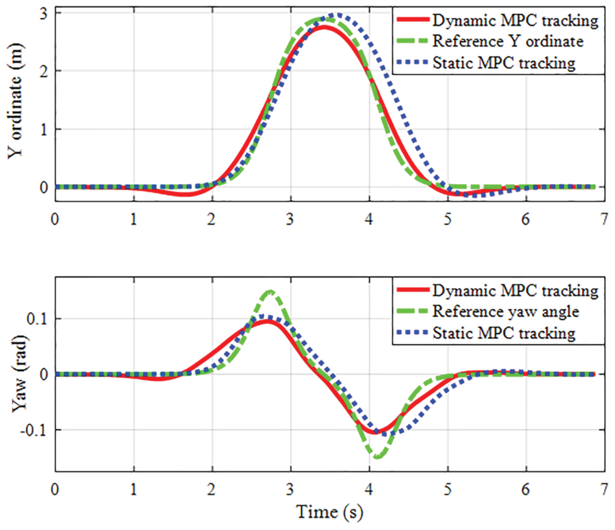

Fig. 15 Comparisons of path-tracking results

Finally, when the vehicle entered into the straight exit lane, both the sideslip angle β and the yaw rate γ returned back around zero and the normalized stability margin $\mathrm{MS}_n(t)$ improved accordingly. Moreover, both MPC controllers had $\mathrm{SI}_{\mathrm{time}}(T_f) > \mathrm{SI}_{\min} = 0.4$ at the end of the simulation, indicating that the stability requirement was satisfied.

Finally, Fig. 15 shows the comparison of the path tracking result, where the dashed lines correspond to the reference path, the dotted lines representing the tracking result of the static MPC, while the solid lines exhibit the tracking result of the dynamic MPC.

Clearly, dynamic MPC induced an overall better path-tracking result and this improvement of path tracking is further expressed in Fig. 16 where the normalized minimum margin between vehicle body and path boundaries ($\mathrm{MT}_n(t)$) is represented.

In Fig. 16, the static MPC led $\mathrm{MT}_n(t)$ equal to zero during 4–5 s, which implies that the vehicle collided with the upper boundary of the reference path. In contrast, the dynamic MPC ensured the minimal value of $\mathrm{MT}_n(t)$ as high as 0.98.

6 Conclusions

To negotiate the inherent conflict between the controller performance optimization and the striking computational burden, a systematic online parameter selection approach for a classical LTV MPC controller for vehicle path tracking was proposed. Various tuning parameters were divided into two groups, namely the insignificant parameters and the significant parameters, according to whether they have a direct influence on both the control performance and the ensued computational load. For insignificant

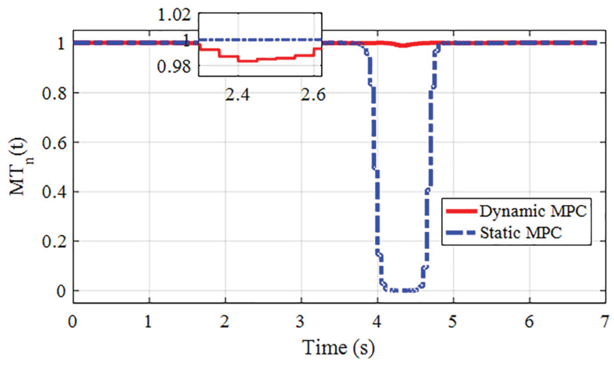

Fig. 16 Normalized minimum margin between vehicle body and path boundaries

parameters, methodological approaches, including the scaled input/output weighting matrix, the adaptive upper/lower bound on soft constraints as well as the dynamic penalty on the slack variables was proposed and justified. Then, to realize online tuning of the remaining three significant parameters, namely, the prediction horizon H_p, the control horizon H_c as well as the sampling period T_s, extensive simulations were carried out to generate three performance maps with respect to the tracking index, the stability index, and the computational load index. Based on the three performance maps, a straightforward constrained optimal path-tracking parameter selection algorithm was designed. Taking the concern of vehicle stability into consideration, the proposed parameter selection algorithm led to the highest attainable tracking performance under a given available CPU computational capacity. MATLAB-CARSIM conjoint simulations demonstrated the effectiveness of the proposed online parameter selection method.

The future improvements of the current work can be summarized into two directions: First, as indicated in Sec. 4.1, the computational load of the MPC controller was determined by measuring the execution time of a default MATLAB solver on a desktop running a Windows® operating system. However, the execution time can vary intensely between consecutive runs due to the time-sharing nature of the operating system. Thus, more robust computational effort indicators, such as the number of floating point operations in Ref. [26], shall be adopted in future work to make the computational load index more accurate. Second, the proposed method needs extensive offline simulations to generate the three fundamental performance maps. In fact, more agile online parameter-selection strategies considering the time-changing geometry of the reference path, such as the road curvature in Ref. [27], shall be adopted in the future work to further improve the overall tracking performance while decreasing the computational load.

Funding Data

- National Science Foundation (Grant No. 1645657, Funder ID. 10.13039/100000001).

References

[1] Chen, P., and Wang, J., 2016, "Estimation and Adaptive Nonlinear Model Predictive Control of Selective Catalytic Reduction Systems in Automotive Applications," J. Process Control, **40**, pp. 78–92.
[2] Yan, F., Wang, J., and Huang, K., 2012, "Hybrid Electric Vehicle Model Predictive Control Torque-Split Strategy Incorporating Engine Transient Characteristics," IEEE Trans. Veh. Technol., **61**(6), pp. 2458–2467.
[3] Siampis, E., Velenis, E., and Longo, S., 2015, "Rear Wheel Torque Vectoring Model Predictive Control With Velocity Regulation for Electric Vehicles," Veh. Syst. Dyn., **53**(11), pp. 1555–1579.
[4] Falcone, P., Tufo, M., Borrelli, F., Asgari, J., and Tseng, H. E., 2007, "A Linear Time Varying Model Predictive Control Approach to the Integrated Vehicle Dynamics Control Problem in Autonomous Systems," 46th IEEE Conference on Decision and Control (CDC), New Orleans, LA, Dec. 12–14, pp. 2980–2985.

[5] TøNdel, P., Johansen, T. A., and Bemporad, A., 2003, "An Algorithm for Multi-Parametric Quadratic Programming and Explicit MPC Solutions," Automatica, **39**(3), pp. 489–497.
[6] Wang, Y., and Boyd, S., 2010, "Fast Model Predictive Control Using Online Optimization," IEEE Trans. Control Syst. Technol., **18**(2), pp. 267–278.
[7] Richards, A., 2015, "Fast Model Predictive Control With Soft Constraints," Eur. J. Control, **25**, pp. 51–59.
[8] Li, S. E., Jia, Z., Li, K., and Cheng, B., 2015, "Fast Online Computation of a Model Predictive Controller and Its Application to Fuel Economy–Oriented Adaptive Cruise Control," IEEE Trans. Intell. Transp. Syst., **16**(3), pp. 1199–1209.
[9] Kouvaritakis, B., Cannon, M., and Rossiter, J. A., 2002, "Who Needs QP for Linear MPC Anyway?," Automatica, **38**(5), pp. 879–884.
[10] Ling, K. V., Wu, B. F., and Maciejowski, J. M., 2008, "Embedded Model Predictive Control (MPC) Using a FPGA," IFAC Proc. Vol., **41**(2), pp. 15250–15255.
[11] Garriga, J. L., and Soroush, M., 2010, "Model Predictive Control Tuning Methods: A Review," Ind. Eng. Chem. Res., **49**(8), pp. 3505–3515.
[12] Van der Lee, J. H., Svrcek, W. Y., and Young, B. R., 2008, "A Tuning Algorithm for Model Predictive Controllers Based on Genetic Algorithms and Fuzzy Decision Making," ISA Trans., **47**(1), pp. 53–59.
[13] Júnior, G. A. N., Martins, M. A., and Kalid, R., 2014, "A PSO-Based Optimal Tuning Strategy for Constrained Multivariable Predictive Controllers With Model Uncertainty," ISA Trans., **53**(2), pp. 560–567.
[14] Ali, E., 2003, "Heuristic On-Line Tuning for Nonlinear Model Predictive Controllers Using Fuzzy Logic," J. Process Control, **13**(5), pp. 383–396.
[15] Garriga, J. L., and Soroush, M., 2008, "Model Predictive Controller Tuning Via Eigenvalue Placement," American Control Conference (ACC), Seattle, WA, June 11–13, pp. 429–434.
[16] Li, L., Wang, F. Y., and Zhou, Q., 2006, "Integrated Longitudinal and Lateral Tire/Road Friction Modeling and Monitoring for Vehicle Motion Control," IEEE Trans. Intell. Transp. Syst., **7**(1), pp. 1–19.
[17] Park, H., and Gerdes, J. C., 2015, "Optimal Tire Force Allocation for Trajectory Tracking With an Over-Actuated Vehicle," IEEE Intelligent Vehicles Symposium (IV), Seoul, South Korea, June 28–July 1, pp. 1032–1037.
[18] Bachtiar, V., Kerrigan, E. C., Moase, W. H., and Manzie, C., 2016, "Continuity and Monotonicity of the MPC Value Function With Respect to Sampling Time and Prediction Horizon," Automatica, **63**, pp. 330–337.
[19] Bachtiar, V., Manzie, C., Moase, W. H., and Kerrigan, E. C., 2016, "Analytical Results for the Multi-Objective Design of Model-Predictive Control," Control Eng. Pract., **56**, pp. 1–12.
[20] Yamashita, A. S., Alexandre, P. M., Zanin, A. C., and Odloak, D., 2016, "Reference Trajectory Tuning of Model Predictive Control," Control Eng. Pract., **50**, pp. 1–11.
[21] Katriniok, A., and Abel, D., 2011, "LTV-MPC Approach for Lateral Vehicle Guidance by Front Steering at the Limits of Vehicle Dynamics," 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC), Orlando, FL, Dec. 12–15, pp. 6828–6833.
[22] Erlien, S. M., 2015, "Shared Vehicle Control Using Safe Driving Envelopes for Obstacle Avoidance and Stability," Ph.D. thesis, Stanford University, Stanford, CA.
[23] Ono, E., Hattori, Y., Muragishi, Y., and Koibuchi, K., 2006, "Vehicle Dynamics Integrated Control for Four-Wheel-Distributed Steering and Four-Wheel-Distributed Traction/Braking Systems," Veh. Syst. Dyn., **44**(2), pp. 139–151.
[24] Rajamani, R., 2011, *Vehicle Dynamics and Control*, Springer Science & Business Media Press, New York, Chap. 8.
[25] Zhang, H., and Wang, J., 2017, "Active Steering Actuator Fault Detection for an Automatically-Steered Electric Ground Vehicle," IEEE Trans. Veh. Technol., **66**(5), pp. 3685–3702.
[26] Wang, J., Solis, J. M., and Longoria, R. J., 2007, "On the Control Allocation for Coordinated Ground Vehicle Dynamics Control Systems," American Control Conference (ACC), New York, July 11–13, pp. 5724–5729.
[27] Guo, H., Liu, J., Cao, D., Chen, H., Yu, R., and Lv, C., 2018, "Dual-Envelop-Oriented Moving Horizon Path Tracking Control for Fully Automated Vehicles," Mechatronics, **50**, pp. 422–433.