

DSCC2018-9129

PARAMETER SELECTION OF AN LTV-MPC CONTROLLER FOR VEHICLE PATH TRACKING CONSIDERING CPU COMPUTATIONAL LOAD

Zejiang Wang*

Department of Mechanical Engineering
University of Texas at Austin
Austin, Texas 78712
wangzejiang@utexas.edu

Junmin Wang*

Department of Mechanical Engineering
University of Texas at Austin
Austin, Texas 78712
jwang@austin.utexas.edu

Yunhao Bai

Dept. of Electrical and Computer Engineering
The Ohio State University
Columbus, OH 43210
bai.228@osu.edu

Xiaorui Wang

Dept. of Electrical and Computer Engineering
The Ohio State University
Columbus, OH 43210
wang.3596@osu.edu

ABSTRACT

There exist two intrinsic shortcomings on model predictive control (MPC) strategy, namely the extensive online calculation burden and the complex tuning process, which prevent MPC from being applied to a wider extent. To tackle these two drawbacks, different methods were proposed with majority of them treating these two issues independently. However, parameter tuning in fact has double-sided effects on both controller performance as well as real-time computational burden. Due to the lack of theoretical tools for globally analyzing the complex conflicts among MPC parameter tuning, controller performance optimization as well as 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 simulation shows the effectiveness of the proposed strategy.

1 INTRODUCTION

Due to its ability of systematical handling states and inputs constraints, model predictive control (MPC) gained a great attention in automotive applications recently [1][2][3][4][5][6][7]. Even with the prevalence of MPC, two inherent drawbacks of this optimization-based control law, which indeed impede its real-time implementation, remain. 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 [8] 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 and cannot handle time-varying systems [9]. Besides, several mechanisms have been proposed to decrease the dimension of the constrained optimization problem and consequently to mitigate the computational burden [10][11][12][13]. Roughly speaking, existing MPC tuning methods in the literature could be divided into three groups: thumb rules, auto-tuning strategies and analytical approaches. Exhaustive tuning guidelines for MPC controller were introduced in [14]. 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 (GA) [15], particle swarm optimization (PSO) [16] or fuzzy logic [17] change tuning parameters in a self-adaptive manner, which could handle constraint violation easily. However, 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 tuning parameters. Finally, analytical approaches [18] study the effect of parameter values from a control theory point of view, which create some new tuning guidelines for MPC.

Even though both the methods that can adequately alleviate MPC real-time computational burden and the approaches leading to effective parameter tunings exist in the literature, there are few methodologies treating these two problems within a *unified*

*Previously affiliated with Ohio State University

framework. A group of tuning parameters, which ensures a better 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 tuning parameters are selected. Due to the lack of theoretical tools that could analyze 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 a LTV-MPC controller for vehicle path tracking is proposed in this paper. The selected parameters assure the highest tracking performance, attainable in the look-up table, of the MPC controller while satisfying a specific minimal stability metric, under a given available central process unit (CPU) computational capacity.

The rest of the paper is organized as follow. System modeling and a classical AFS MPC controller are introduced in Section 2. Insignificant parameter setting strategies are illustrated and verified in Section 3. The definition of ‘significant’ and ‘insignificant’ parameters will be given in this section. Definitions of controller performance index are given in Section 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 Section 5. Section 6 concludes this paper.

2 AFS MPC 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 ones, a classical LTV-MPC AFS controller in [4] is utilized here with only minor improvements.

2.1 SYSTEM MODELING

The three degrees-of-freedom bicycle model of a front-steering vehicle is used here to represent the dynamics of the system, as:

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

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

$$\begin{aligned} \sum F_y &= (F_{yfl} + F_{yfr}) \sin(\delta_f) + (F_{yrl} + F_{yrr}) \cos(\delta_f) + F_{yrl} + F_{yrr}, \\ \sum F_x &= (F_{xfl} + F_{xfr}) \cos(\delta_f) - (F_{yfl} + F_{yfr}) \sin(\delta_f) + F_{xrl} + F_{xrr}, \\ \sum M_z &= l_f \left[(F_{xfl} + F_{xfr}) \sin(\delta_f) + (F_{yfl} + F_{yfr}) \cos(\delta_f) \right] \\ &\quad - l_r \left[F_{yrl} + F_{yrr} \right] + l_d \left[\left(\frac{F_{yfl} - F_{yfr}}{F_{xfl} - F_{xfr}} \right) \sin(\delta_f) - F_{xrl} + \right. \\ &\quad \left. \left(\frac{F_{xfr} - F_{xfl}}{F_{yfl} - F_{yfr}} \right) \cos(\delta_f) + F_{xrr} \right], \end{aligned}$$

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, $F_{\{x,y\}\{i,j\}}$, ($i \in \{(f)ront, (r)ear\}$, $j \in \{(l)eft, (r)ight\}$) exhibiting the longitudinal/lateral tire force acting on each wheel.

To describe the longitudinal and lateral tire forces, different tire force models have been proposed [19]. In this paper, the Brush tire model in [20] is applied. The expression of the longitudinal and lateral tire forces $F_{x,y}$ at each wheel reads:

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

$$F = \begin{cases} f - \frac{1}{3\mu F_z} f^2 + \frac{1}{27\mu^2 F_z^2} f^3, & f \leq 3\mu F_z, \\ \mu F_z, & f > 3\mu F_z, \end{cases} \quad (3)$$

$$(F_x, F_y) = \left(C_x \left(\frac{s}{s+1} \right) * F / f, -C_y \left(\frac{\tan \alpha}{s+1} \right) * F / f \right), \quad (4)$$

with $C_{x,y}$ as the longitudinal and cornering tire stiffness of a single tire, μ representing the tire-road friction coefficient (TRFC), F_z the vertical tire force, s and α being separately the tire slip ratio and the tire sideslip angle.

The vertical tire force acting on each tire can be calculated as:

$$\begin{cases} F_{zfl} = m(l_r g - a_x h) / 2(l_f + l_r) - l_r m a_y h / 2l_d(l_f + l_r), \\ F_{zfr} = m(l_r g - a_x h) / 2(l_f + l_r) + l_r m a_y h / 2l_d(l_f + l_r), \\ F_{zrl} = m(l_f g + a_x h) / 2(l_f + l_r) - l_f m a_y h / 2l_d(l_f + l_r), \\ F_{zrr} = m(l_f g + a_x h) / 2(l_f + l_r) + l_f m a_y h / 2l_d(l_f + l_r), \end{cases} \quad (5)$$

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

In equation (2), the tire sideslip angles and tire slip ratios read:

$$\begin{cases} \alpha_{fl} = \tan^{-1} \left(\frac{v_y + l_f \gamma}{v_x - l_d \gamma} \right) - \delta_f, \alpha_{fr} = \tan^{-1} \left(\frac{v_y + l_f \gamma}{v_x + l_d \gamma} \right) - \delta_f, \\ \alpha_{rl} = \tan^{-1} \left(\frac{v_y - l_r \gamma}{v_x - l_d \gamma} \right), \alpha_{rr} = \tan^{-1} \left(\frac{v_y - l_r \gamma}{v_x + l_d \gamma} \right), \end{cases} \quad (6)$$

$$\begin{cases} s_{fl} = \frac{R_w \omega_{fl} - v_{xfl}}{\max(R_w \omega_{fl}, v_{xfl})}, & s_{fr} = \frac{R_w \omega_{fr} - v_{xfr}}{\max(R_w \omega_{fr}, v_{xfr})}, \\ s_{rl} = \frac{R_w \omega_{rl} - v_{xrl}}{\max(R_w \omega_{rl}, v_{xrl})}, & s_{rr} = \frac{R_w \omega_{rr} - v_{xrr}}{\max(R_w \omega_{rr}, v_{xrr})}, \end{cases} \quad (7)$$

with R_w as the effective wheel rolling radius, $\omega_{\{i,j\}}$ being the wheel spinning angular velocity and $v_{x\{i,j\}}$ representing the longitudinal velocity at wheel center, as:

$$\begin{cases} v_{xfl} = (v_y + l_f \gamma) \sin(\delta_f) + (v_x - l_d \gamma) \cos(\delta_f), \\ v_{xfr} = (v_y + l_f \gamma) \sin(\delta_f) + (v_x + l_d \gamma) \cos(\delta_f), \\ v_{xrl} = v_x - l_d \gamma, \\ v_{xrr} = v_x + l_d \gamma. \end{cases} \quad (8)$$

Equations (1)-(8) constitute the whole dynamics of the system, and an equivalent compact form can be represented as:

$$\begin{cases} \dot{\zeta}(t) = f(\zeta(t), u(t)), \\ \eta(t) = h(\zeta(t), u(t)), \end{cases} \quad (9)$$

with $\zeta = [v_y, v_x, \psi, \gamma, Y, X]$, $\eta = [\psi, Y, \alpha_{fl}, \alpha_{fr}, \alpha_{rl}, \alpha_{rr}]$ along with $u = \delta_f$ as separately the state vector, output vector as well as the unique system input. To apply a standard MPC, equation (9) needs to be discretized and successively linearized online to produce an approximated linear time-varying discrete system [4], as:

$$\begin{cases} \zeta(k+1) = A_{k,t} \zeta(k) + B_{k,t} u(k) + d_{k,t}, \\ \eta(k+1) = C_{k,t} \zeta(k) + D_{k,t} u(k) + e_{k,t}, \\ u(k) = u(k-1) + \Delta u(k), \end{cases} \quad (10)$$

with

$$\begin{cases} A_{k,t} = \left. \frac{\partial f}{\partial \zeta} \right|_{\zeta_t, u(k-1)}, & B_{k,t} = \left. \frac{\partial f}{\partial u} \right|_{\zeta_t, u(k-1)}, \\ C_{k,t} = \left. \frac{\partial h}{\partial \zeta} \right|_{\zeta_t, u(k-1)}, & D_{k,t} = \left. \frac{\partial h}{\partial u} \right|_{\zeta_t, u(k-1)}, \end{cases} \quad (11)$$

where $k = t \dots t + N - 1$, and $d_{k,t}$, $e_{k,t}$ correspond to the linearization residual items.

2.2 MPC CONTROLLER DESIGN

Grounded on the discrete system (10), a constrained optimization problem can be formulated as:

$$\min_{\Delta U_t, \varepsilon} J(\zeta(t), \Delta U_t), \quad (12)$$

such that,

$$\begin{aligned} \zeta(k+1) &= A_{k,t} \zeta(k) + B_{k,t} (u(k-1) + \Delta u(k)) + d_{k,t}, \\ \eta(k+1) &= C_{k,t} \zeta(k) + D_{k,t} (u(k-1) + \Delta u(k)) + e_{k,t}, \\ u_{\min} &\leq u \leq u_{\max}, \\ \Delta u_{\min} &\leq \Delta u \leq \Delta u_{\max}, \\ \alpha_{i,j \min} - \varepsilon_{i,j} &\leq \alpha_{i,j} \leq \alpha_{i,j \max} + \varepsilon_{i,j}, \end{aligned} \quad (13)$$

with

$$\begin{aligned} J(\zeta(t), \Delta U_t) &= \sum_{i=1}^{H_p} \|\hat{\mathbf{h}}_{t+i,t} - \mathbf{h}_{t+i,t}^*\|_Q^2 + \\ &\sum_{j=0}^{H_c-1} \|u_{t+j,t}\|_S^2 + \sum_{j=0}^{H_c-1} \|\Delta u_{t+j,t}\|_R^2 + \sum_{i=\text{right}}^{\text{left}} \sum_{j=\text{front}}^{\text{rear}} \|\varepsilon_{i,j}\|_\rho^2, \end{aligned} \quad (14)$$

where $\Delta U_t = [\Delta u_{t,t}, \dots, \Delta u_{t+H_c-1,t}]$ acts as the major manipulated variable vector along the control horizon H_c at the time instant t , $\hat{\mathbf{h}}_{t+i,t} = [\hat{\psi}_{t+i,t} \quad \hat{Y}_{t+i,t}]^T$ representing a part of the predictive system output vector η , which shall be converged to their corresponding reference vector $\mathbf{h}_{t+i,t}^*$ along the prediction horizon H_p . Then, $\varepsilon_{i,j}$ represents the slack variable of the soft constraints on the tire sideslip angle $\alpha_{i,j}$, which constitutes the rest part of the manipulated variables. As a final point, Q, S, R, ρ serve as positive-definite weighting matrix. In addition, two implicit constraints include $H_p \geq H_c$ and $\Delta u_{t+j,t} = 0, \forall j \geq H_c$. After solving the constrained optimization problem described in (12)-(14), the first element of ΔU_t will be used to construct the controller output at the time instant t as:

$$u(t) = u(t-1) + \Delta u_{t,t}. \quad (15)$$

At the next time step $t+1$, the same constrained optimization problem will be solved again with updated system measurements ζ_{t+1} .

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 (10), the weighting matrix Q, S, R, ρ , the upper- and lower-bounds of control/control changing rate magnitude $u_{\max}, u_{\min}, \Delta u_{\max}, \Delta u_{\min}$, 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 which directly affect *both* controller performance and computational load. As indicated in [21], [22], 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 as well as the number of constraints boost, which substantially increases the necessary 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 [23]. Consequently, the matrix Q is fixed as:

$$Q = \begin{bmatrix} 1/\psi_{\max} & \\ & 1/Y_{\max} \end{bmatrix}, \quad (16)$$

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

3.2 INPUT/INPUT RATE WEIGHTING MATRIX

Similarly, to normalize control effort, the input weighting matrix as well as the input changing-rate weighting matrix are separately settled as:

$$R = 1/(\max(\dot{\delta}_f)T_s), \quad S = 1/\max(\delta_f), \quad (17)$$

where $\max(\delta_f)$ and $\max(\dot{\delta}_f)$ are individually the mechanical range limit of the front road steering angle and the maximal angular velocity of δ_f . T_s is the sampling period whose selection method will be shown later. Further, equation (17) indeed suggests that the upper- and lower-bounds of the control/control changing-rate shall be accordingly fixed as:

$$\begin{cases} u_{\max} = \max(\delta_f), & \Delta u_{\max} = \max(\dot{\delta}_f)T_s, \\ u_{\min} = -\max(\delta_f), & \Delta u_{\min} = -\max(\dot{\delta}_f)T_s. \end{cases} \quad (18)$$

3.3 BOUNDS ON THE SOFT CONSTRAINTS

Soft constraints on four wheel's sideslip angles are crucial to establish the stability of the vehicle. In [4], the upper- and lower-bounds of $\alpha_{i,j}$ were constantly fixed as ± 3 degrees. Virtually, the allowable limits of tire sideslip angles are radically determined by vehicle states, tire's proper characteristics as well as road condition. Therefore, adaptive constraint bounds on tire sideslip angles inspired by [24] will be adopted here. According to the Brush tire model described in (2)-(4), the lateral tire force will saturate if the absolute value of tire sideslip angle becomes excessive. Hence, the bounds of tire sideslip angle $\alpha_{i,j \max}, \alpha_{i,j \min}$ must ensure that the lateral force of each tire are restricted within a region where its extreme value is not totally saturated in near

future. 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 upper bound 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 of the tire sideslip angle within prediction horizon, and the reciprocal upper-bound of the tire sideslip angle is simply the opposite number of the lower-bound.

For a positive sideslip angle, the procedure is similar.

Fig. 1 illustrates this idea, with parameters fixed as: $s = 0$, $\mu = 0.8$, $F_z = 4000N$, $C_y = 62700N/rad$.

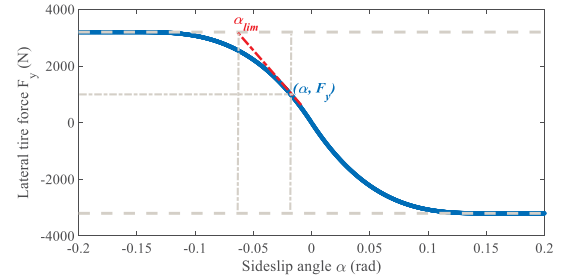


Fig. 1. Bounds on tire slip angle.

3.4 SLACK VARIABLE WEIGHTING MATRIX

The slack variables $\varepsilon_{i,j}$ were introduced in the soft-constraints on the tire sideslip angles to ensure the feasibility of the constrained optimization problem (12)-(14). It is preferable if the magnitude of $\varepsilon_{i,j}$ could remain as small as possible to rend these constraints, which are pivotal to ensure the stability of the vehicle, still valid. In this paper, $\rho_{i,j}$ is defined as:

$$\rho_{i,j} = \frac{1}{\left| \partial F_{yi,j} / \partial \alpha_{i,j} \right|_{\alpha_{i,j} = \alpha_{i,j \lim}}}. \quad (19)$$

The function of such an adaptive penalty is twofold. Firstly, if the sideslip angle α 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 α approaches the critical value leading to a saturated lateral tire force, then the weight on the slack variable drastically rises into infinity, which prevents the sideslip angle further enlarging and consequently maintains the stability of the vehicle.

3.5 VERIFICATION

In Sections 3.1-3.4, all the insignificant parameters are 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 show the online selection strategy with respect to these three significant parameters, it is necessary to verify the proposed tuning methods of these insignificant parameters. Thus, simulations of a typical double lane changing (DLC) scenario on

the Matlab-Carsim conjoint simulation platform were conducted. The vehicle configurations and other constant parameters used during the simulation were fixed as: $l_f = 1.232, l_r = 1.468, l_d = 0.77, h = 0.54, m = 1723, R_w = 0.3, I_z = 1960, C_x = 66900N, C_y = 62700N/rad, \psi_{\max} = 0.1489, \mu = 0.8, Y_{\max} = 2.8921, \max \dot{\delta}_f = 17.5 \text{ deg/s}, \max \delta_f = 20 \text{ deg}$. Unless indicated, all parameters are in SI units. And the three significant parameters were arbitrarily fixed as: $H_p = 30, H_c = 10, T_s = 0.05$. The vehicle longitudinal velocity remained 108km/h during the simulation.

Fig. 2 shows the path-tracking result. Fig. 3 presents the front road steering angle and Fig. 4 demonstrates the sideslip angle of each tire.

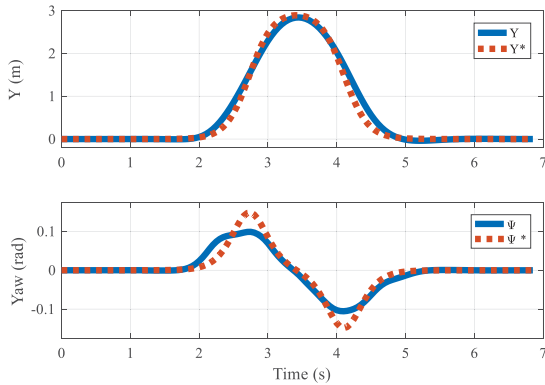


Fig. 2. Path tracking result.

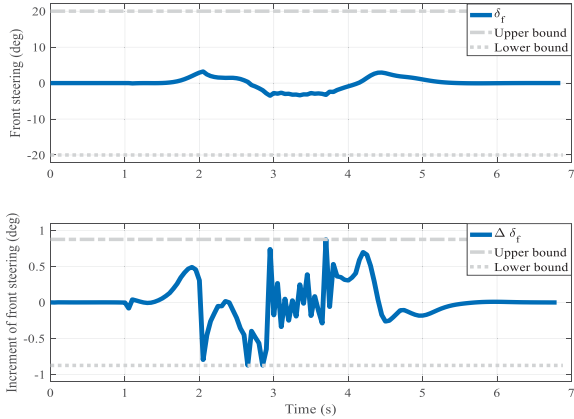


Fig. 3. Front steering angle.

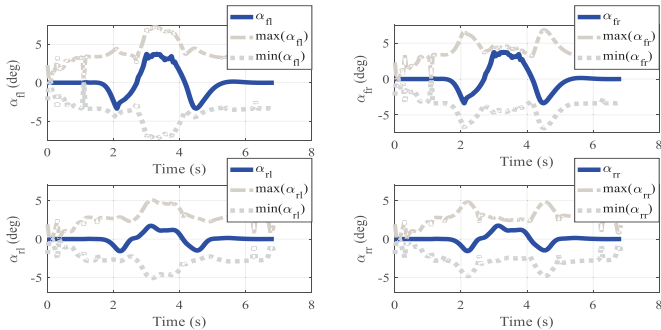


Fig. 4. Four tire sideslip angles.

According to Fig. 2-4, 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 output nor the soft constraints on sideslip angles.

4 PERFORMANCE MAP

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

4.1 PERFORMANCE INDEX DEFINITION

Three fundamental performance indices include *tracking index*, *stability index*, as well as *computational load index*. 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 word, the larger the minimal distance between vehicle body and road boundaries, the higher the tracking performance shall be. The concept of the safe driving envelop in [25] is utilized here to quantify the tracking performance. Instead of treating vehicle as a mass point at its CG, vehicle's physical dimension is taken into consideration to calculate the minimal distance between vehicle body and road boundaries. The ordinates of vehicle's four wheels in an inertial coordinate can be calculated as:

$$\begin{cases} Y_{fl} = Y_{cg} + \sqrt{l_f^2 + l_d^2} \sin(\Delta\psi + \tan^{-1}(l_d/l_f)), \\ Y_{fr} = Y_{cg} - \sqrt{l_f^2 + l_d^2} \sin(-\Delta\psi + \tan^{-1}(l_d/l_f)), \\ Y_{rl} = Y_{cg} + \sqrt{l_r^2 + l_d^2} \sin(-\Delta\psi + \tan^{-1}(l_d/l_r)), \\ Y_{rr} = Y_{cg} - \sqrt{l_r^2 + l_d^2} \sin(\Delta\psi + \tan^{-1}(l_d/l_r)), \end{cases} \quad (20)$$

where Y_{cg} indicates the ordinate of vehicle's CG, and $\Delta\psi$ shows vehicle's yaw error with respect to the reference path. Further, the ordinate of road boundaries can be formulated as:

$$Y_{upper} = Y^* + \frac{W}{2 \cos(\psi^*)}, \quad Y_{lower} = Y^* - \frac{W}{2 \cos(\psi^*)}, \quad (21)$$

with Y^* and ψ^* representing the reference ordinate and heading angle of a point on path centerline whose abscissa is the same as vehicle CG's. In addition, W is the width of the reference lane, fixed as 3.6m [25]. Consequently, the minimum margin between vehicle body and path boundaries can be calculated as:

$$MT = \min \left(\begin{aligned} & \left(Y_{upper} - \max(Y_{fl}, Y_{fr}, Y_{rl}, Y_{rr}) \right), \\ & \left(\min(Y_{fl}, Y_{fr}, Y_{rl}, Y_{rr}) - Y_{lower} \right) \end{aligned} \right). \quad (22)$$

Clearly, a path tracking that avoids collision between vehicle and road boundaries always needs $MT > 0$ and the maximal possible value $MT_{\max} = W/2 - l_d$ can be attainable only when the vehicle's CG lays exactly on the centerline of the reference path with zero yaw error.

Afterward, *stability index* reflects the margin between vehicle's current state from the critical ones under which the vehicle may spin, drift or roll over. To remain stable, both the body sideslip angle β as well as the yaw rate γ of vehicle need to be constrained within reasonable ranges. According to [26], empirical thresholds about sideslip angle and yaw rate which lead to a critical instability are: $|\beta^*| = \tan^{-1}(0.02\mu g)$ and $|\gamma^*| = 0.85\mu g/v_x$.

Hence, the stability margin can be defined as:

$$MS = \min \left(\min \left(1 \pm \frac{\beta}{\beta^*} \right), \min \left(1 \pm \frac{\gamma}{\gamma^*} \right) \right). \quad (23)$$

Naturally, $MS(t) > 0$ implies that both the sideslip angle and the yaw rate are under their respective critical threshold. In addition, the most stable case occurs when both β and γ are zero, which leads to $MS_{\max} = 1$. In general, $MT_{\max} \neq MS_{\max}$. Hence, after obtaining MT and MS , two smooth hyperbolic tangent functions will be used to normalize separately MT and MS at each time instant, which lead to the normalized minimum margin between vehicle body and path boundaries MT_n and the normalized stability margin as MS_n , both within the range $[0,1]$. Eventually, the overall tracking index and stability index for a complete simulation are designated as:

$$TI = \frac{\int_0^{T_f} MT_n dt}{T_f}, \quad SI = \frac{\int_0^{T_f} MS_n dt}{T_f}, \quad (24)$$

with T_f being the end of simulation time.

Finally, the *computational load index* of the MPC controller is defined as [22]:

$$CI = T_c/T_s, \quad (25)$$

where T_c is the total execution time to find the optimal solution of problem (12)-(14). Intuitively, a higher CI implies a higher CPU load entailed from the MPC controller and the upper-threshold of CI is one.

4.2 PERFORMANCE MAP GENERATION

After defining all three fundamental performance indices, extensive simulations with various combination of H_p , H_c and T_s can be effected to generate three performance maps with respect to the three performance indices TI , SI and CI . Precisely speaking, for a given triplet $\{H_p, H_c, T_s\}$, a double lane change (DLC) scenario identical to the one in Section 3.5 was used to evaluate 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 given H_p and H_c started from 0.01s and ended at 0.05s with an increment of 0.005s. A total number of 3321 simulations were realized. Based on the three generated performance maps, reverse look-ups can be utilized to find the optimal triplets of $\{H_p^*, H_c^*, T_s^*\}$ among the total 3321 possible combinations according to the proposed parameter selection algorithm illustrated in Section 5.

Remark: The simulations were implemented on a standard Hewlett-Packard (HP) desktop with an Intel i7-4790 processor whose clock rate is 3.60GHz. The optimization solver was the default Matlab QP solver with an 'active-set' method.

5 ONLINE PARAMETER SELECTION

To help a vehicle achieve the highest attainable tracking performance while consider the minimal requirement of stability index $SI_{\min}$ under a given upper threshold of computational load index $CI_{\max}$, a straightforward parameter selection algorithm is proposed as:

Algorithm : Constrained optimal tracking parameter selection

Input : $SI_{\min}$, $CI_{\max}$

Output : $\{H_p^*, H_c^*, T_s^*\}$

- 1: $\Lambda = \text{find}_{T_s, H_p, H_c} (CI \leq CI_{\max})$
 - 2: **if** $\Lambda = \emptyset$ **then**
 - 3: $T_s^* \leftarrow \max(T_s), H_p^* \leftarrow \min(H_p), H_c^* \leftarrow \min(H_c)$
 - 4: **endif**
 - 5: $\Omega = \text{find}_{T_s, H_p, H_c} (SI \geq SI_{\min})$
 - 6: **if** $\Omega = \emptyset \parallel \Omega \cap \Lambda = \emptyset$ **then**
 - 7: $(T_s^*, H_p^*, H_c^*) \leftarrow \text{find}_{(T_s, H_p, H_c) \in \Lambda} (SI = \max(SI))$
 - 8: **else**
 - 9: $(T_s^*, H_p^*, H_c^*) \leftarrow \text{find}_{(T_s, H_p, H_c) \in \Lambda \cap \Omega} (TI = \max(TI))$
 - 10: **end**
-

At the beginning, find all the possible combinations of H_c, H_p, T_s satisfying the computational load constraint $CI \leq CI_{\max}$. However, if the given available computational index $CI_{\max}$ 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 $SI \geq SI_{\min}$. If such combinations cannot be found due to an extremely high stability index threshold $SI_{\min}$ (under the given computational load constraint), then just find the combination of H_c, H_p, T_s which rends the stability index as high as possible. Nonetheless, in this

case, the path-tracking objective is totally ignored. Finally, if both the given stability index constraint as well as 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 demonstrate persuasively the improvement introduced by the algorithm, simulations with a double lane change (DLC) scenario identical to the one in Section 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 Section 3.5. Both the dynamic MPC as well as the static MPC selected their significant parameters according to the proposed constrained optimal-tracking parameter selection algorithm. Nonetheless, the difference was that the available $CI_{\max}^d$ for the dynamic MPC, denoted as $CI_{\max}^d$, changed along the simulation, while the $CI_{\max}^s$ for the static MPC, denoted as $CI_{\max}^s$ remained as the minimum value of $CI_{\max}^d$. The minimum stability index $SI_{\min}$ for both controllers was fixed as 0.4.

Fig. 5 demonstrates both $CI_{\max}^d$ and $CI_{\max}^s$ 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 operate under their individual available upper threshold of computational load. Fig. 6 shows the three significant parameters of the dynamic MPC as well as the static MPC. Clearly, the proposed algorithm can synchronously adjust the three significant parameters according to the given maximum available computational index. For instance, as $CI_{\max}^d$ decreased during 2-4s, 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 could be found within the sampling period. Finally, Fig. 7 shows the comparison of the path tracking result, where the green-dashed lines correspond to the reference path; the blue-dotted lines represent the tracking result of the static MPC while the red-solid lines exhibit the tracking result of the dynamic MPC.

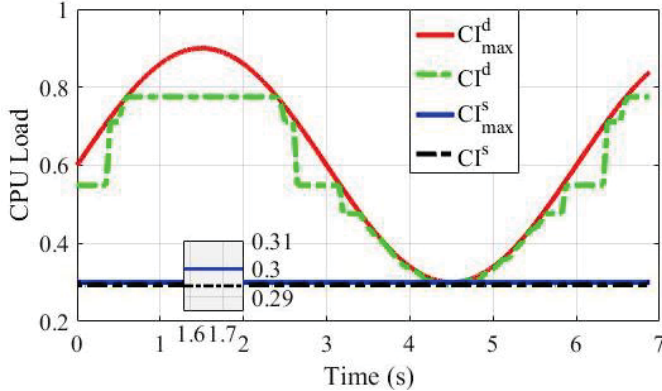


Fig. 5 Computational load index.

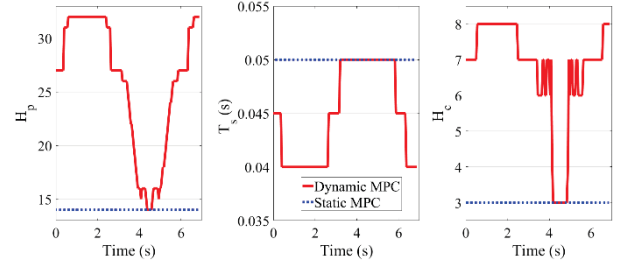


Fig. 6 Constant and dynamic parameter settings.

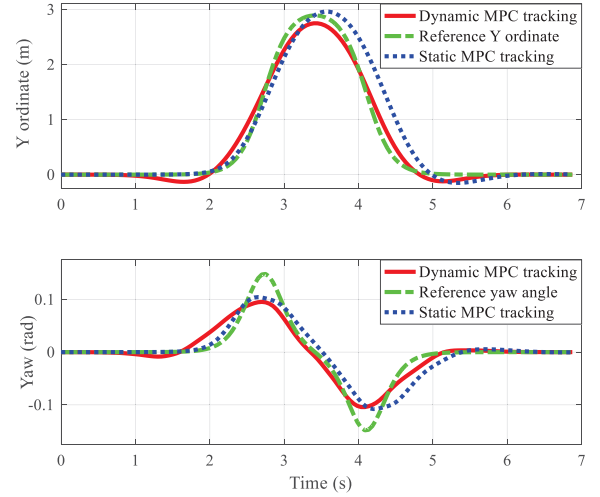


Fig. 7 Path tracking result.

Hence, compared with the static MPC, dynamic MPC induced an overall better path-tracking result and this improvement of path tracking is further expressed in Fig. 8 where the normalized minimum margin between vehicle body and path boundaries (MT_n defined in Section 4.1) is represented.

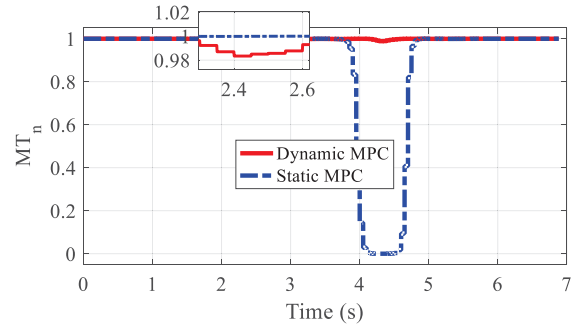


Fig. 8 Normalized margin between vehicle and boundaries.

Clearly, between 4-5s, the static MPC led MT_n equal to zero, which implies that the vehicle collided with the upper boundary of the reference path. In contrast, the dynamic MPC insured the minimal value of MT_n as high as 0.98.

Remark: After the online parameter selection methods integrated with the original MPC controller, the overall computational burden shall increase accordingly. However, simulations showed that the average extra computational burden introduced by this online parameter selection strategy was less 1%. Hence, this extra computational load was disregarded.

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 had a direct influence on both the control performance and the entailed computational load. Then, to realize online tuning of the three significant parameters, extensive simulations were carried out to generate three performance maps with respect to the tracking index, stability index, and the computational load index. Based on the three performance maps, a straightforward constrained optimal path-tracking parameter selection algorithm was generated. The proposed parameter selection algorithm induced the highest attainable tracking performance while avoided violating the minimal stability index constraint under a given available central process unit (CPU) computational capacity. Matlab-CarSim conjoint simulations demonstrated the effectiveness of the proposed online parameter selection method.

ACKNOWLEDGMENT

The authors would like to thank the financial support provided by the NSF, with the project number 1645657.

REFERENCES

- [1] Chen, Ping and Wang, Junmin. "Estimation and Adaptive Nonlinear Model Predictive Control of Selective Catalytic Reduction Systems in Automotive Applications." *Journal of Process Control* Vol. 40 (2016): pp. 78-92.
- [2] Yan, Fengjun, Wang, Junmin and Huang, Kaisheng. "Hybrid Electric Vehicle Model Predictive Control Torque-Split Strategy Incorporating Engine Transient Characteristics." *IEEE Transactions on Vehicular Technology* Vol. 61 No.6 (2012): pp. 2458-2467.
- [3] Siampis, Efstathios, Velenis, Efstathios and Longo Stefano. "Rear Wheel Torque Vectoring Model Predictive Control with Velocity Regulation for Electric Vehicles." *Vehicle System Dynamics* Vol. 53 No.11 (2015): pp. 1555-1579.
- [4] Falcone, Paolo, Tufo, Manuela, Borrelli, Francesco, Asgari, Jahan and Tseng, H. Eric. "A Linear Time Varying Model Predictive Control Approach to the Integrated Vehicle Dynamics Control Problem in Autonomous Systems." *Proceedings of the 46th IEEE Conference on Decision and Control (CDC)*. ThPI25.3: pp. 2980-2985. New Orleans, LA, USA, Dec. 12-14, 2007
- [5] Ji, Jie, Khajepour, Amir, Melek, Wael William and Huang, Yanjun. "Path Planning and Tracking for Vehicle Collision Avoidance Based on Model Predictive Control with Multiconstraints." *IEEE Transactions on Vehicular Technology* Vol. 66 No.2 (2017): pp. 952-964.
- [6] Liniger, Alexander, Domahidi, Alexander and Morari, Manfred. "Optimization-based Autonomous Racing of 1: 43 Scale RC Cars." *Optimal Control Applications and Methods* Vol. 36 No.5 (2015): pp. 628-647.
- [7] Liu, Jiechao, Jayakumar, Paramsothy, Stein, Jeffrey L. and Ersal, Tulga. "A Multi-Stage Optimization Formulation for MPC-Based Obstacle Avoidance in Autonomous Vehicles Using a LIDAR Sensor." *Proceedings of the ASME 2014 Dynamic Systems and Control Conference (DSCC)*. DSCC2014-6269: pp. V002T30A006-V002T30A006. San Antonio, TX, USA, Oct. 22-24, 2014
- [8] Tøndel, Petter, Johansen, Tor-Arne and Alberto Bemporad. "An Algorithm for Multi-Parametric Quadratic Programming and Explicit MPC Solutions." *Automatica* Vol. 39 No.3 (2003): pp. 489-497.
- [9] Wang, Yang and Boyd, Stephen. "Fast Model Predictive Control Using Online Optimization." *IEEE Transactions on Control Systems Technology* Vol. 18 No.2 (2010): pp. 267-278.
- [10] Richards, Arthur. "Fast Model Predictive Control with Soft Constraints." *European Journal of Control* Vol. 25 (2015): pp. 51-59.
- [11] Li, Shengbo Eben, Jia, Zhenzhong and Li, Keqiang. "Fast Online Computation of a Model Predictive Controller and Its Application to Fuel Economy-Oriented Adaptive Cruise Control." *IEEE Transactions on Intelligent Transportation Systems* Vol. 16 No. 3 (2015): pp. 1199-1209.
- [12] Kouvaritakis, Basil, Cannon, Mark and Rossiter, J. Anthony. "Who Needs QP for Linear MPC Anyway?" *Automatica* Vol. 38 No. 5 (2002): pp. 879-884.
- [13] Ling, Keck-Voon, Wu, Bing Fang and Jan, Maciejowski. "Embedded Model Predictive Control (MPC) Using a FPGA." *Proceedings of the 17th World Congress, The International Federation of Automatic Control*. Seoul, Korea, July 6-11, 2008
- [14] Garriga, Jorge L. and Soroush, Masoud. "Model Predictive Control Tuning Methods: A Review." *Industrial and Engineering Chemistry Research* Vol. 49 No.8 (2010): pp. 3505-3515.
- [15] Van der Lee, JH, Svrcek, WY and Young, B.R. "A Tuning Algorithm for Model Predictive Controllers Based on Genetic Algorithms and Fuzzy Decision Making." *ISA Transactions* Vol. 47 No.1 (2008): pp. 53-59.
- [16] Júnior, Gesner A. Nery, Martins, Márcio AF and Kalid, Ricardo. "A PSO-Based Optimal Tuning Strategy for Constrained Multivariable Predictive Controllers with Model Uncertainty." *ISA Transactions* Vol. 53 No.2 (2014): pp. 560-567.
- [17] Ali, Emad. "Heuristic On-Line Tuning for Nonlinear Model Predictive Controllers Using Fuzzy Logic." *Journal of Process Control* Vol. 13 No.5 (2003): pp. 383-396.
- [18] Garriga, Jorge L. and Soroush, Masoud. "Model Predictive Controller Tuning Via Eigenvalue Placement." *American Control Conference (ACC)*. WeA13.1: pp. 429-434. Westin Seattle Hotel, Seattle, Washington, USA, June 11-13, 2008
- [19] Li, Li, Wang, Feiyue and Zhou, Qunzhi. "Integrated Longitudinal and Lateral Tire/Road Friction Modeling and Monitoring for Vehicle Motion Control." *IEEE Transactions on Intelligent Transportation Systems* Vol. 7 No.1 (2006): pp. 1-19.
- [20] Park, Hyungchai and Gerdes, J.Christian. "Optimal Tire Force Allocation for Trajectory Tracking With an Over-Actuated Vehicle." *IEEE Intelligent Vehicles Symposium (IV)*. pp. 1032-1037. COEX, Seoul, Korea, June 28 - July 1, 2015.
- [21] Bachtar, Vincent, Kerrigan, Eric C, Moase, William H. and Manzie, Chris. "Continuity and Monotonicity of the MPC Value Function with Respect to Sampling Time and Prediction Horizon." *Automatica* Vol. 63 (2016): pp. 330-337.
- [22] Bachtar, Vincent, Manzie, Chris, Moase, William H. and Kerrigan, Eric C. "Analytical Results for the Multi-Objective Design of Model-Predictive Control." *Control Engineering Practice* Vol. 56 (2016): pp. 1-12.
- [23] Yamashita, André Shigueo, Alexandre, Paulo Martin, Zanin, Antonio Carlos and Odloak, Darci. "Reference Trajectory Tuning of Model Predictive Control." *Control Engineering Practice* Vol. 50 (2016): pp. 1-11.
- [24] Katriniok, Alexander and Abel, Dirk. "LTV-MPC Approach for Lateral Vehicle Guidance by Front Steering at the Limits of Vehicle Dynamics." *Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*. pp. 6828-6833. Orlando, FL, USA, December 12-15, 2011
- [25] Stephen M. Erlien. "Shared Vehicle Control Using Safe Driving Envelopes for Obstacle Avoidance and Stability." PhD Thesis. Stanford University, Stanford, CA. 2015.
- [26] Rajesh Rajamani. "Electronic Stability Control" *Vehicle Dynamics and Control*. Springer Science & Business Media (2011): pp. 233-235.