

Flatness-based Model Predictive Control for Autonomous Vehicle Trajectory Tracking*

Zejiang Wang, Jingqiang Zha, and Junmin Wang, *Senior Member, IEEE*

Abstract— Model predictive control (MPC) has become more relevant to vehicle dynamics control due to its inherent capacity of treating system constraints. However, online optimization from MPC introduces an extensive computational burden for today's onboard microprocessors. To alleviate MPC computational load, several methods have been proposed. Among them, online successive system linearization and the resulting linear time-varying model predictive controller (LTV MPC) is one of the most popular options. Nevertheless, such online successive linearization commonly approximates the original (nonlinear) system by a linear one, which inevitably introduces extra modeling errors and therefore reduces MPC performance. Actually, if the controlled system possesses the “differential flatness” property, then it can be *exactly* linearized and an equivalent linear model will appear. This linear model maintains all the nonlinear features of the original system and can be utilized to design a flatness-based model predictive controller (FMPC). CarSim-Simulink joint simulations demonstrate that the proposed FMPC substantially outperforms a classical LTV MPC in terms of the path-tracking performance for autonomous vehicles.

I. INTRODUCTION

Model predictive control (MPC), owing to its ability of explicitly treating state and input constraints, has been pervasively adopted for ground vehicle dynamic control [1], [2]. However, because of the inherent online optimization process, MPC, especially nonlinear MPC, entails a heavy computational burden for today's vehicular electronic control unit (ECU). Several approaches [3] have been proposed and verified to alleviate the computational burden of MPC. For instance, efficient optimization solvers based on the Nesterov's gradient method [4] or the real-time iteration algorithm [5] have been designed for reducing both algorithm complexity and memory space requirement. Besides, explicit MPC [3] has attracted attention from both academia and industry. Among all the existing methods, online successive linearization along with the resulting linear-time-varying model predictive controller (LTV MPC) is probably one of the most straightforward and popular options [2], [6]. Nonetheless, local linearization around the current working point introduces extra approximation error, which degrades the control performance of an LTV MPC.

In contrast to the local linearization, a differentially flat system [7] can be *exactly* linearized with the help of the flat output. In fact, the inputs, states, and outputs of a

differentially flat system can be completely parameterized with the flat output and its finite-order derivatives. Hence, a nonlinear flat system can be alternatively expressed as its linear Brunovsky canonical form, which significantly reduces control law design. Despite the benefits, explicitly including constraints into a flat system remains hard. Therefore, combining differential flatness with MPC gives us triple benefits: 1) Systematic constraints handling from the MPC structure; 2) MPC computational load alleviation thanks to the exact linearization; and 3) Control performance enhancement with respect to an LTV MPC by avoiding linearization error.

In fact, there exist several control structures in the literature integrating differential flatness with predictive control. In [8], a flatness predictive controller was illustrated. Here, the predicted flat output was employed to obtain feedforward system inputs and a simple proportional-integral-derivative (PID) controller was added for disturbance rejection. In [9], state references were generated from the desired flat output and a separate MPC was designed for referential states tracking. In [10], [11], and [12], system flat output were firstly parameterized as a combination of basic functions and the original optimization problem was transformed into a nonlinear programming problem (NLP) with respect to the coefficients of the basic functions. However, as indicated in [11], it is difficult to prove that NLP can lead to the global optima. Moreover, the computational time of NLP can be too long to be implemented online [12]. Finally, similar to the philosophy underlies this paper, MPC applied on an equivalent linear flat system can be found in [13] and [14]: The original nonlinear model was firstly converted into a linear-time-invariant (LTI) system. After that, an MPC based on this LTI model was designed to obtain the optimal flat inputs and states, from which the original control inputs can be calculated.

This paper proposes a hierarchical flatness-based model predictive controller (FMPC) for autonomous vehicle trajectory tracking. The high-level predictive controller generates the desired global forces and moment which are then allocated to four wheels' independent steering and torque signals by a simple control-allocation module. CarSim-Simulink joint simulations demonstrate the advantages of the high-level FMPC over an LTV MPC.

The rest of the paper is organized as follows. A novel kinodynamic vehicle model for path tracking is presented in Section II. Its flatness character is proved in Section III. The high-level control for global forces and moment generation from either LTV MPC or FMPC is illustrated in Section IV, where the low-level control allocation algorithm is also mentioned. Simulation result and analysis are given in Section V, and Section VI concludes this paper.

*This work was supported by National Science Foundation award number 1645657.

Zejiang Wang, Jingqiang Zha, and Junmin Wang are with the Walker Department of Mechanical Engineering, University of Texas at Austin, Austin, TX 78712 USA (e-mail: {wangzejiang, zhajingqiang}@utexas.edu, jwang@austin.utexas.edu).

II. SYSTEM MODELING

A. Vehicular Kinodynamic Trajectory Tracking Model

Inspired by [15], [24], the kinodynamic vehicle model for trajectory tracking is demonstrated in Fig. 1.

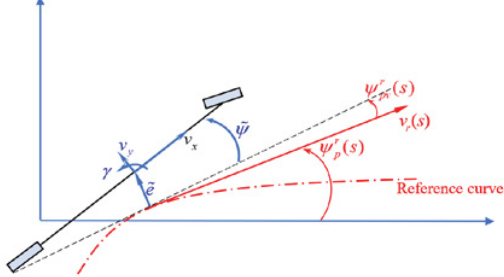


Fig. 1. Kinodynamic trajectory tracking model.

This new system contains five states, as:

$$x = [\tilde{e} \quad \tilde{\psi} \quad \tilde{v} \quad v_y \quad \gamma]^T. \quad (1)$$

In (1), $\tilde{e}$ represents the minimum distance from vehicle's center of gravity (CG) to the reference path. $\tilde{\psi}$ indicates vehicle yaw angle error with respect to the desired yaw $\psi_p^r(s) + \psi_{pv}^r(s)$, with $\psi_p^r(s)$ being the referential road direction and $\psi_{pv}^r(s)$ accounting for the vehicle sideslip amendment [16]. $\tilde{v}$ is the velocity tracking error in regard to the desired tangential speed $v_r(s)$. v_y specifies vehicle lateral speed at CG and γ shows vehicle yaw rate. $\psi_p^r(s)$, $\psi_{pv}^r(s)$, and $v_r(s)$ are external reference signals that are dependent on the station s along the reference path. We assume them available from a planning module. Vehicle longitudinal velocity v_x , despite not included in the model, is regarded as a measured parameter for determining $\tilde{v}$.

Besides, system admits three inputs as:

$$u = [F_x \quad F_y \quad M_z]^T, \quad (2)$$

where $F_{x,y}$ represents respectively the global longitudinal and lateral force and M_z is the global yaw moment around CG.

According to the kinematics relationships, we have:

$$\dot{\tilde{e}} = v_x \sin(\tilde{\psi} + \psi_{pv}^r(s)) + v_y \cos(\tilde{\psi} + \psi_{pv}^r(s)), \quad (3)$$

and

$$\dot{\tilde{\psi}} = \gamma - \left(\frac{\partial \psi_p^r(s)}{\partial s} + \frac{\partial \psi_{pv}^r(s)}{\partial s} \right) (\tilde{v} + v_r(s)), \quad (4)$$

along with

$$\begin{aligned} \dot{\tilde{v}} = & \frac{F_x}{m} \cos(\tilde{\psi} + \psi_{pv}^r(s)) - \frac{F_y}{m} \sin(\tilde{\psi} + \psi_{pv}^r(s)) \\ & + \frac{\partial \psi_p^r(s)}{\partial s} (\tilde{v} + v_r(s)) \dot{\tilde{e}} - \frac{\partial v_r(s)}{\partial s} (\tilde{v} + v_r(s)). \end{aligned} \quad (5)$$

Then, from the vehicle lateral dynamics, we have:

$$\dot{v}_y = F_y/m - v_x \gamma, \quad (6)$$

and

$$\dot{\gamma} = M_z/I_z. \quad (7)$$

Equations (3)-(7) form the complete vehicular kinodynamic trajectory tracking model.

In practice, by assuming $\tilde{e} \ll 1/\kappa_r(s)$, where $\kappa_r(s)$ stands for the referential road curvature, we have:

$$\begin{cases} \tilde{v} \approx (v_x \cos(\tilde{\psi} + \psi_{pv}^r(s)) - v_y \sin(\tilde{\psi} + \psi_{pv}^r(s))) - v_r(s), \\ \tilde{\psi} = \psi - \psi_{pv}^r(s) - \psi_p^r(s). \end{cases} \quad (8)$$

In (8), $\psi_{pv}^r(s)$ can be seen as the steady-state vehicle sideslip angle during turning [17] as:

$$\psi_{pv}^r(s) = -l_r \kappa(s) + \frac{m l_f v_r(s)^2 \kappa(s)}{2 C_y (l_f + l_r)}, \quad (9)$$

where l_f and l_r respectively indicate the distances from vehicle CG to the front and to the rear axles, m is vehicle mass, and C_y represents the cornering stiffness of a single tire.

III. FLATNESS SYSTEM PROOF

We prove here that the vehicle kinodynamic model depicted by (3)-(7) is differentially flat with the flat output:

$$\eta = [\tilde{e} \quad \tilde{\psi} \quad \tilde{v}]. \quad (10)$$

The essence to prove a system being flat is to show that the system states and inputs can be parameterized with the flat output and its finite-order derivatives. Since η already groups three system states, we only need to concentrate on the parameterization of v_y , γ , and F_x , F_y , M_z .

In fact, by rearranging (4), we directly obtain:

$$\gamma = \dot{\tilde{\psi}} + \left(\frac{\partial \psi_p^r(s)}{\partial s} + \frac{\partial \psi_{pv}^r(s)}{\partial s} \right) (\tilde{v} + v_r(s)), \quad (11)$$

and by combining (3) and (8), we have:

$$v_y = \dot{\tilde{e}} \cos(\tilde{\psi} + \psi_{pv}^r(s)) - (\tilde{v} + v_r(s)) \sin(\tilde{\psi} + \psi_{pv}^r(s)). \quad (12)$$

Subsequently, by calculating the second-order derivative of $\tilde{e}$ and $\tilde{\psi}$, we have:

$$\begin{aligned} \ddot{\tilde{e}} = & \frac{F_x}{m} \sin(\tilde{\psi} + \psi_{pv}^r(s)) \\ & + \frac{F_y}{m} \cos(\tilde{\psi} + \psi_{pv}^r(s)) - \frac{\partial \psi_p^r(s)}{\partial s} (\tilde{v} + v_r(s))^2, \end{aligned} \quad (13)$$

$$\begin{aligned} \ddot{\tilde{\psi}} = & \frac{M_z}{I_z} - \left(\frac{\partial^2 \psi_p^r(s)}{\partial s^2} + \frac{\partial^2 \psi_{pv}^r(s)}{\partial s^2} \right) (\tilde{v} + v_r(s))^2 \\ & - \left(\frac{\partial \psi_p^r(s)}{\partial s} + \frac{\partial \psi_{pv}^r(s)}{\partial s} \right) \left(\dot{\tilde{v}} + \frac{\partial v_r(s)}{\partial s} (\tilde{v} + v_r(s)) \right). \end{aligned} \quad (14)$$

From (5), (13), and (14), system inputs parameterization can be shown as:

$$\begin{aligned} \frac{F_x}{m} = & \sin(\tilde{\psi} + \psi_{pv}^r(s)) \left(\ddot{e} + \frac{\partial \psi_p^r(s)}{\partial s} (\tilde{v} + v_r(s))^2 \right) \\ & + \cos(\tilde{\psi} + \psi_{pv}^r(s)) \left(\dot{\tilde{v}} - \frac{\partial \psi_p^r(s)}{\partial s} (\tilde{v} + v_r(s)) \dot{e} + \frac{\partial v_r(s)}{\partial s} (\tilde{v} + v_r(s)) \right), \end{aligned} \quad (15)$$

along with

$$\begin{aligned} \frac{F_y}{m} = & \cos(\tilde{\psi} + \psi_{pv}^r(s)) \left(\ddot{e} + \frac{\partial \psi_p^r(s)}{\partial s} (\tilde{v} + v_r(s))^2 \right) \\ & + \sin(\tilde{\psi} + \psi_{pv}^r(s)) \left(-\dot{\tilde{v}} + \frac{\partial \psi_p^r(s)}{\partial s} (\tilde{v} + v_r(s)) \dot{e} - \frac{\partial v_r(s)}{\partial s} (\tilde{v} + v_r(s)) \right), \end{aligned} \quad (16)$$

and finally,

$$\begin{aligned} \frac{M_z}{I_z} = & \ddot{\tilde{\psi}} + \left(\frac{\partial^2 \psi_p^r(s)}{\partial^2 s} + \frac{\partial^2 \psi_{pv}^r(s)}{\partial^2 s} \right) (\tilde{v} + v_r(s))^2 \\ & + \left(\frac{\partial \psi_p^r(s)}{\partial s} + \frac{\partial \psi_{pv}^r(s)}{\partial s} \right) \left(\dot{\tilde{v}} + \frac{\partial v_r(s)}{\partial s} (\tilde{v} + v_r(s)) \right). \end{aligned} \quad (17)$$

As indicated in (11) and (12), the rest two system states γ and v_y can be expressed with the flat output η and its first-order derivatives. Similarly, as shown in (15), (16), and (17), three system inputs: F_x, F_y, M_z can also be expressed with the flat output η and its derivatives up to the second order. Thus, the vehicle kinodynamic model is indeed flat.

IV. MODEL PREDICTIVE CONTROLLER DESIGN

The aim of the high-level MPC is to obtain the optimal global forces and moment u in (2) which minimize the vehicle trajectory tracking error η in (10) and satisfy several system constraints. Both an LTV MPC and an FMPC are designed and compared in this Section. In addition, a typical low-level control-allocation algorithm is also included at the end to transform the optimal global forces and moment into the optimal four-wheel torque and steering signals.

A. Linear Time-Varying Model Predictive Controller

The kinodynamic model in Section II can be compactly written as:

$$\begin{cases} \dot{x} = f(x, u, \psi_{pv}^r(s), \psi_p^r(s), v_r(s)), \\ y = x. \end{cases} \quad (18)$$

with x and u indicated in (1) and (2) and we assume all the states can be measured. Then, the online optimization problem can be summarized as:

$$\begin{aligned} \min_{F_x, F_y, M_z, \xi_{v_y}, \xi_\gamma} & \frac{1}{2} \sum_{i=1}^{H_p} \left(\|\tilde{e}(k+i|k)\|_{Q_e}^2 + \|\tilde{\psi}(k+i|k)\|_{Q_\psi}^2 + \|\tilde{v}(k+i|k)\|_{Q_v}^2 \right) \\ & + \frac{1}{2} \sum_{i=0}^{H-1} \left(\|F_x(k+i|k) - F_x^*(k+i|k)\|_{R_{F_x}}^2 + \|F_y(k+i|k) - F_y^*(k+i|k)\|_{R_{F_y}}^2 \right) \\ & + \frac{1}{2} \sum_{i=0}^{H-1} \left(\|M_z(k+i|k) - M_z^*(k+i|k)\|_{R_{M_z}}^2 \right) + \rho_{v_y} \xi_{v_y} + \rho_\gamma \xi_\gamma, \end{aligned} \quad (19)$$

such that

$$\begin{cases} x(k+1) = A_k x(k) + B_k u(k) + d_{k,t}, \\ y(k) = x(k), \end{cases} \quad (20)$$

and

$$\begin{cases} |v_y(k+i|k)| < 0.02 v_x \mu g + \xi_{v_y}, \\ |\gamma(k+i|k)| < \frac{0.85 \mu g}{v_x} + \xi_\gamma, i = 1 \dots H_p. \end{cases} \quad (21)$$

The cost function (19) minimizes three items: The tracking errors η within the prediction horizon H_p , the discrepancies between the actual high-level inputs u and the desired system inputs u^* within the control horizon H_c , and the soft constraint violation penalty with respect to v_y and γ . In (19), the desired system inputs u^* can be decided by assigning $\tilde{e} = \dot{\tilde{e}} = \ddot{\tilde{e}} = 0$, $\tilde{\psi} = \dot{\tilde{\psi}} = \ddot{\tilde{\psi}} = 0$, and $\tilde{v} = \dot{\tilde{v}} = 0$ in (15), (16), and (17), and the penalty coefficients are fixed as: $\rho_{v_y} = \rho_\gamma = 1e6$.

The constraints demonstrated in (20) represent the online successively linearized system model. Additionally, soft constraints on vehicle lateral speed v_y as well as yaw rate γ [6] are included in (21) to ensure vehicle stability. In (21), μ stands for the tire road friction coefficient, g is Earth gravity, ξ_{v_y} and ξ_γ are the slack variables.

No explicit constraints on system inputs u are accounted for, majorly due to two reasons: 1) The thresholds of global forces and moment are difficult to be estimated in real time [18]; 2) High-level input constraints are partially redundant with the low-level control allocation algorithm.

B. Flatness Model Predictive Controller

Before the FMPC design, the original nonlinear system (18) needs to be converted into an equivalent LTI system.

By assigning $\ddot{\tilde{e}} = v_e$, $\ddot{\tilde{\psi}} = v_\psi$, and $\dot{\tilde{v}} = v_v$, we can *exactly* represent the original nonlinear model (18) as a linear system, with *flat states* as:

$$x_p = [\tilde{e} \quad \dot{\tilde{e}} \quad \tilde{\psi} \quad \dot{\tilde{\psi}} \quad \tilde{v}], \quad (22)$$

which satisfy

$$\begin{bmatrix} \dot{\tilde{e}} \\ \ddot{\tilde{e}} \\ \dot{\tilde{\psi}} \\ \ddot{\tilde{\psi}} \\ \dot{\tilde{v}} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{e} \\ \dot{\tilde{e}} \\ \tilde{\psi} \\ \dot{\tilde{\psi}} \\ \tilde{v} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_e \\ v_\psi \\ v_v \end{bmatrix}, \quad (23)$$

or abstractly written as:

$$\dot{x}_p = A_p x_p + B_p v, \quad (24)$$

where v is called the *flat input*.

Based on the LTI model (24), an online successive optimization problem can be formulated as:

$$\begin{aligned} \min_{v_e, v_\psi, v_v, \xi_e, \xi_\psi, \xi_v} & \frac{1}{2} \sum_{i=1}^{H_p} \left(\|\tilde{e}(k+i|k)\|_{Q_e}^2 + \|\tilde{\psi}(k+i|k)\|_{Q_\psi}^2 + \|\tilde{v}(k+i|k)\|_{Q_v}^2 \right) \\ & + \frac{1}{2} \sum_{i=0}^{H_p-1} \left(\|v_e(k+i|k)\|_{R_{v_e}}^2 + \|v_\psi(k+i|k)\|_{R_{v_\psi}}^2 + \|v_v(k+i|k)\|_{R_{v_v}}^2 \right) \\ & + \rho_\gamma \xi_\gamma + \rho_{\xi_e} \xi_e + \rho_{\xi_\psi} \xi_\psi + \rho_{\xi_v} \xi_v, \end{aligned} \quad (25)$$

such that

$$\begin{cases} x_p(k+1) = A_{pk} x_p(k) + B_{pk} u(k), \\ y_p(k) = x_p(k), \end{cases} \quad (26)$$

and

$$\begin{cases} \left| \dot{\tilde{e}}(k+i|k) \cos(\tilde{\psi}(k+i|k) + \psi_{pv}^r(k+i|k)) \right. \\ \left. - (\tilde{v}(k+i|k) + v_r(k+i|k)) \sin(\tilde{\psi}(k+i|k) + \psi_{pv}^r(k+i|k)) \right| < 0.02 v_x \mu g + \xi_{v_x}, \\ \left| \dot{\tilde{\psi}}(k+i|k) + \left(\frac{\partial \psi_p^r(s)}{\partial s} (k+i|k) + \frac{\partial \psi_{pv}^r(s)}{\partial s} (k+i|k) \right) \right| < \frac{0.85 \mu g}{v_x} + \xi_\gamma, i=1 \dots H_p, \\ \left| (\tilde{v}(k+i|k) + v_r(k+i|k)) \right| < \frac{0.85 \mu g}{v_x} + \xi_\gamma, i=1 \dots H_p, \end{cases} \quad (27)$$

Similar to (19), cost function (25) minimizes the tracking errors within the prediction horizon H_p , the flat input v within the control horizon H_c , and the soft constraints violation penalty. Constraint (26) is the discrete expression of (24) and the soft constraints on the lateral velocity and yaw rate are maintained in (27). Clearly, system parameterizations in (11) and (12) convert the simple box constraints (21) into the state-dependent inequalities. Moreover, the constraint on v_y becomes nonlinear. To treat this newly introduced nonlinear constraint, we use the method in [19] to online successively linearize this constraint. Unlike LTV MPC which linearizes the whole state vector x in (1), only lateral velocity is linearized for constructing a quadratic programming problem for online optimization.

Using the first-order Taylor expansion, the parameterized v_y in (27) within the prediction horizon can be expressed as:

$$\begin{aligned} v_y(k+i|k) &= g(\tilde{e}(k+i|k), \tilde{\psi}(k+i|k), \tilde{v}(k+i|k), \psi_{pv}^r(k+i|k), v_r(k+i|k)) \\ &\approx g(\tilde{e}^*(k+i|k-1), \tilde{\psi}^*(k+i|k-1), \tilde{v}^*(k+i|k-1), \psi_{pv}^r(k+i|k), v_r(k+i|k)) \\ &+ \frac{\partial g}{\partial \tilde{e}} \bigg|_{\substack{\tilde{e}^*(k+i|k-1) \\ \tilde{\psi}^*(k+i|k-1) \\ \tilde{v}^*(k+i|k-1)}} [\tilde{e}(k+i|k) - \tilde{e}^*(k+i|k-1)] + \frac{\partial g}{\partial \tilde{\psi}} \bigg|_{\substack{\tilde{e}^*(k+i|k-1) \\ \tilde{\psi}^*(k+i|k-1) \\ \tilde{v}^*(k+i|k-1)}} [\tilde{\psi}(k+i|k) - \tilde{\psi}^*(k+i|k-1)] \\ &+ \frac{\partial g}{\partial \tilde{v}} \bigg|_{\substack{\tilde{e}^*(k+i|k-1) \\ \tilde{\psi}^*(k+i|k-1) \\ \tilde{v}^*(k+i|k-1)}} [\tilde{v}(k+i|k) - \tilde{v}^*(k+i|k-1)], i=1 \dots H_p, \end{aligned}$$

where

$$\begin{cases} \frac{\partial g}{\partial \tilde{e}} = \cos(\tilde{\psi} + \psi_{pv}^r), \frac{\partial g}{\partial \tilde{v}} = -\sin(\tilde{\psi} + \psi_{pv}^r), \\ \frac{\partial g}{\partial \tilde{\psi}} = -\tilde{e} \sin(\tilde{\psi} + \psi_{pv}^r) - (\tilde{v} + v_r) \cos(\tilde{\psi} + \psi_{pv}^r). \end{cases} \quad (29)$$

and $\tilde{e}^*(k+i|k-1), \tilde{\psi}^*(k+i|k-1), \tilde{v}^*(k+i|k-1)$ are the optimal predicted flat states within the prediction horizon obtained at the last step.

After solving the optimization problem (25)-(27), the optimal flat inputs $v^* = [v_e^* \ v_\psi^* \ v_v^*] = [\tilde{e}^* \ \tilde{\psi}^* \ \tilde{v}^*]$ need to be substituted back into (15), (16), and (17) to give us the real optimal control $u^* = [F_x^* \ F_y^* \ M_z^*]$.

C. Control Allocation

The optimal global forces and moment from either the LTV MPC or the FMPC must be allocated to each wheel's spinning torque and steering. We assume the autonomous vehicle under control enjoys four-wheel independent steering and actuation (4WIS-4WIA) capacity. For such an over-actuated system, the pseudo-inverse method in [20][25] is utilized to obtain the optimal tire force vector:

$$F_{tire}^* = [F_{xi}^* \ F_{yi}^*]^T, i = \{fl \ rl \ fr \ rr\}, \quad (30)$$

where $F_{x,yi}^*$ represents the desired longitudinal and lateral tire force of each wheel.

As indicated in [20], the final expression of F_{tire}^* can be formulated as:

$$F_{tire}^* = (B^T Q B + W)^{-1} B^T Q u^*. \quad (31)$$

In (31), B represents the system matrix which associates four wheels' tire forces with the global forces and moment. Q is the allocation error matrix. W is responsible for restraining excessively demanded tire force, and u^* stands for the optimal control from the high-level MPC. A minor modification was adopted in (31): Instead of using the grip margin in [20], the dialog elements in W are changed as the following one to avoid tire force saturation. In (32), s_{ij} is tire slip ratio and α_{ij} is tire sideslip angle. C_x, C_y constitute tire longitudinal and cornering stiffness, and $\hat{F}_{xij}, \hat{F}_{yij}$ represent the modelled longitudinal and lateral tire force of each wheel. There exist various tire models in the literature, such as the brush tire model in [20], the parameter varying model in [21], the magic formula in [22] to estimate tire forces. To be consistent, the brush tire model in [20] was utilized to calculate $\hat{F}_{xij}, \hat{F}_{yij}$.

$$\begin{cases} W(F_{xij}) = \tan\left(\frac{\pi}{2}\left(1 - \frac{\partial \hat{F}_{xij}}{\partial s_{ij}} / C_x\right)\right), \\ W(F_{yij}) = \tan\left(\frac{\pi}{2}\left(1 - \frac{\partial \hat{F}_{yij}}{\partial \alpha_{ij}} / C_y\right)\right), \end{cases} \quad (32)$$

As a final point, the optimal wheel torque T_{ij}^* and steering δ_{ij}^* of each wheel are determined with:

$$\begin{cases} T_{ij}^* = R_{ew} F_{xij}^*, \\ \delta_{ij}^* = K_1 \int (F_{yij}^* - \hat{F}_{yij}) dt + K_2 \int \int (F_{yij}^* - \hat{F}_{yij}) dt dt, \end{cases} \quad (33)$$

where R_{ew} is the effective wheel radius, $\hat{F}_{yij}$ is estimated lateral tire force, and $K_{1,2}$ are control gains.

V. SIMULATION RESULTS AND ANALYSIS

A path composed of a straight line and a curve with time-varying reference speed is designed to compare the tracking performance between the two MPCs. All simulations were conducted on Simulink-CarSim joint platform.

To fairly tune the weighting factors in (19) and (25), a pre-simulation with a simple Brunovsky control law [7] was conducted. Following a classical tuning rule [6], we have:

$$Q_x = 1/\max(|x|), R_y = 1/\max(|y|), \quad (34)$$

where x stands for $\tilde{e}$, $\tilde{\psi}$, $\tilde{v}$ and y represents either F_x, F_y, M_z for the LTMPC or v_e, v_ψ, v_v for the FMPC. The final tuning results of the LTMPC are: $R_{F_x} = 3.41e-4$, $R_{F_y} = 2.13e-4$, $R_{M_z} = 4.5e-3$, $Q_e = 10225$, $Q_\psi = 28846$, and $Q_v = 12220$. As for the FMPC, we have: $R_{v_e} = 1.46$, $R_{v_\psi} = 9.13$, $R_{v_v} = 4.07$, $Q_e = 34.08$, $Q_\psi = 96.15$, and $Q_v = 40.73$. The tracking error weights Q_e, Q_ψ, Q_v of the LTMPC were *intentionally* augmented 300 times which ensured that no obvious improvement of the LTMPC in terms of the tracking results can be witnessed even the tracking error weights continue augmenting. Finally, for both LTMPC and FMPC controllers, we fixed $H_p = 10$, $H_c = 1$, $T_s = 0.05$, where T_s is the sampling period.

Both the LTMPC and the FMPC were implemented by using CVXGEN [23] and the optimal solution was always found within 0.001 second with a 1.8GHz CPU. The low-level control allocation algorithm in Section IV was utilized to generate wheel torque T_{ij}^* and steering δ_{ij}^* for both MPCs.

The global path tracking results can be found in Fig. 2.

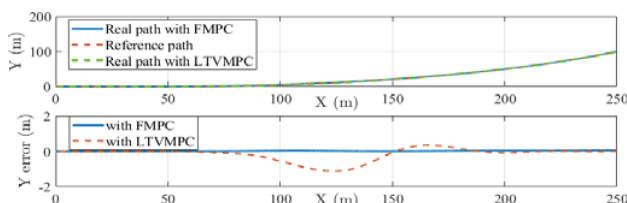


Fig. 2. Path tracking results with LTMPC and FMPC.

As indicated in Fig. 2, both MPC controllers can make the autonomous vehicle follow the given path but the tracking error of the FMPC is much smaller.

Then, the minimum distance between vehicle's center of gravity and the reference path ($\tilde{e}$) is depicted in Fig. 3. Still, FMPC obviously outperformed LTMPC.

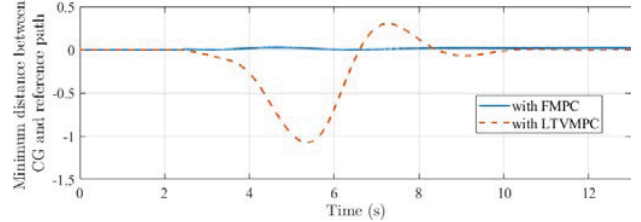


Fig. 3. Minimum distance between CG and referential path.

After that, the yaw tracking results can be found in Fig. 4.

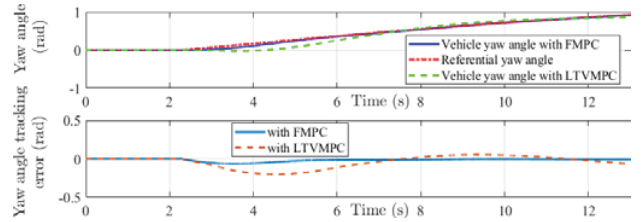


Fig. 4. Road yaw tracking results with LTMPC and FMPC.

Clearly, the FMPC produced a swifter response toward abrupt road direction changes and achieved a globally smaller tracking error.

Subsequently, the tangential velocity tracking results are demonstrated in Fig. 5.

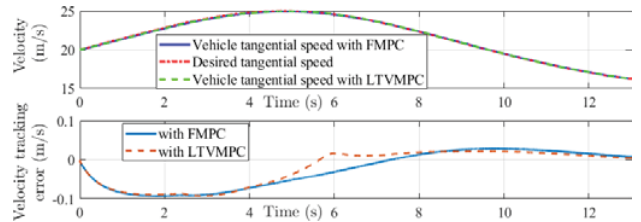


Fig. 5. Tangential speed tracing results with LTMPC and FMPC.

This time, the LTMPC gave us a slightly better tracking result and the underlying reason will be revealed in Fig. 8.

Next, v_y , γ and their constraints are included in Fig. 6.

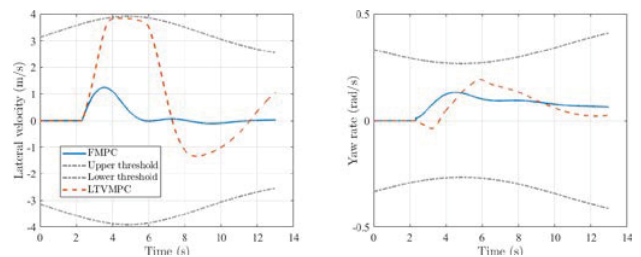


Fig. 6. Lateral velocity and yaw rate constraints satisfaction.

As shown in Fig. 6, both the LTMPC and the FMPC guaranteed that the constraints on v_y and γ were observed. However, FMPC gave us smaller lateral velocity and yaw rate

whereas LTMPC pushed the lateral velocity toward the limit. In other words, FMPC introduced a higher stability index.

According to Figs. 2-6, we can make the conclusion that the FMPC outperformed the LTMPC even though the weights on tracking errors in LTMPC were 300 times higher than FMPC. Two fundamental reasons account for this phenomenon: 1) LTMPC uses local linearization, which inevitably induces approximation error; 2) FMPC exploits the open-loop control terms, which cogently lessens the feedback control efforts.

Finally, four-wheel steering and torque signals from the LTMPC and FMPC are separately depicted in Fig. 7 and Fig. 8.

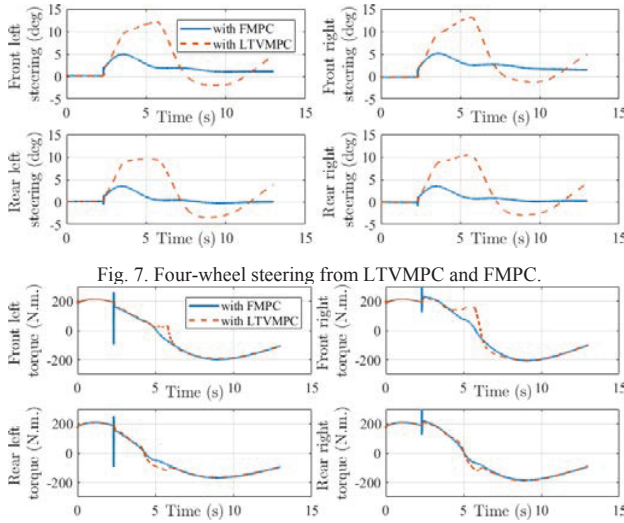


Fig. 7. Four-wheel steering from LTMPC and FMPC.

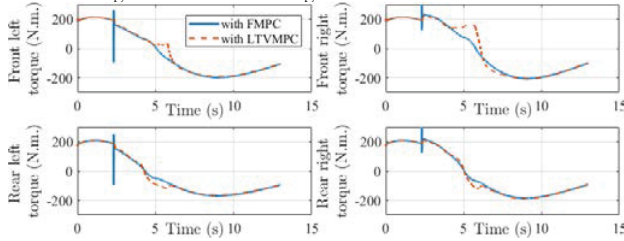


Fig. 8. Four-wheel torques from LTMPC and FMPC.

It is clearly indicated in Fig. 7 that LTMPC entailed unnecessarily drastic four-wheel steering, which strongly affected vehicle stability as reflected in Fig. 6. In Fig. 8, the four wheel torques from FMPC suffered from a spike around 2.2s, because the referential curvature suddenly changed from 0 to a fixed positive number at this moment. This discontinuity substantially impacted the performance of the flatness controller, which eventually resulted in a slightly reduced speed tracking performance as indicated in Fig. 5.

VI. CONCLUSION

This paper proposed an FMPC for autonomous vehicle trajectory tracking control, which outperformed an LTMPC. Future work will concentrate on the hardware-in-the-Loop (HIL) implementation and assessment of this FMPC algorithm.

REFERENCES

- [1] G.V. Raffo et al., "A Predictive Controller for Autonomous Vehicle Path Tracking," in *IEEE Trans. Intell. Transp. Syst.*, vol. 10, no. 1, pp. 92-102, 2009.
- [2] P. Falcone et al., "Predictive Active Steering Control for Autonomous Vehicle Systems," in *IEEE Trans. Control Syst. Technol.*, vol. 15, no. 3, pp. 566-580, 2007.

- [3] Y. Wang and S. Boyd, "Fast Model Predictive Control Using Online Optimization," in *IEEE Trans. Control Syst. Technol.*, vol. 18, no. 2, pp. 267-278, 2010.
- [4] M. Kögel, and R. Findeisen, "Fast Predictive Control of Linear Systems Combining Nesterov's Gradient Method and The Method of Multipliers," in *IEEE CDC and ECC*, Dec. 2011, pp. 3519-3522.
- [5] S. Gros et al., "From Linear to Nonlinear MPC: Bridging the Gap via The Real-Time Iteration," in *Int. J. Control.*, pp. 1-19, 2016.
- [6] Z. Wang et al., "Vehicle Path Tracking LTV-MPC Controller Parameter Selection Considering CPU Computational Load," in *J. Dyn. Syst.-T. ASME*, vol. 141, pp. 051004-1-12, 2019.
- [7] M. Fliess et al., "On Differentially Flat Nonlinear Systems," in *Nonlinear Control Systems Design*, pp.159-163, 1993.
- [8] M. Fliess, and R. Marquez, "Continuous-Time Linear Predictive Control and Flatness: A Module-Theoretic Setting with Examples," in *Int. J. Control*, vol. 73, no. 7, pp. 606-623, 2000.
- [9] I. Prodan et al., "Receding Horizon Flight Control for Trajectory Tracking of Autonomous Aerial Vehicles," in *Control. Eng. Pract.*, vol. 21, no. 10, pp. 1334-1349, 2013.
- [10] R. Mahadevan et al., "Differential Flatness Based Nonlinear Predictive Control of Fed-Batch Bioreactors," in *Control. Eng. Pract.*, vol. 9, no. 8, pp. 889-899, 2001.
- [11] J.A. De Doná et al., "A Flatness-Based Iterative Method for Reference Trajectory Generation in Constrained NMPC," in *Nonlinear Model Predictive Control*, pp. 325-333, 2009.
- [12] M. Joševski, and D. Abel, "Flatness-Based Model Predictive Control for the Fuel Optimization of Hybrid Electric Vehicles," in *IFAC-PapersOnLine*, vol. 48, no. 23, pp. 464-470, 2015.
- [13] G. Allibert et al., "A Flat Model Predictive Controller for Trajectory Tracking in Image Based Visual Servoing," in *IFAC Proc. Volumes*, vol. 40, no. 12, pp. 993-998, 2007.
- [14] M. Greeff, and A.P. Schoellig, "Flatness-Based Model Predictive Control for Quadrotor Trajectory Tracking," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2018, pp. 6740-6745.
- [15] H. Park, and J.C. Gerdes, "Optimal Tire Force Allocation for Trajectory Tracking with an Over-Actuated Vehicle," in *IEEE Intell. Veh. Symposium (IV'15)*, June. 2015, pp. 1032-1037.
- [16] C. Hu et al., "Should the Desired Heading in Path Following of Autonomous Vehicles be The Tangent Direction of The Desired Path?," in *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 6, pp. 3084-3094, 2015.
- [17] R. Rajamani., *Vehicle Dynamics and Control*. Springer Science & Business Media, 2011, pp. 53.
- [18] M. Jonasson et al., "Global Force Potential of Over-Actuated Vehicles," in *Vehicle System Dynamics*, vol. 48, no. 9, pp. 983-998, 2010.
- [19] W.R. Van Soest et al., "Combined Feedback Linearization and Constrained Model Predictive Control for Entry Flight," in *J. Guid. Control, Dynam.*, vol. 29, no. 2, pp. 427-434, 2006.
- [20] R. Wang et al., "Integrated Optimal Dynamics Control of 4WD4WS Electric Ground Vehicle with Tire-Road Frictional Coefficient Estimation," in *Mechanical Systems and Signal Processing*, vol. 60, pp. 727-741, 2015.
- [21] M. Kissai et al., "A New Linear Tire Model with Varying Parameters," in *2017 2nd IEEE International Conference on Intelligent Transportation Engineering (ICITE)*, Sep. 2017, pp. 108-115.
- [22] R. Wang and J. Wang., "Actuator-Redundancy-Based Fault Diagnosis for Four-Wheel Independently Actuated Electric Vehicles," in *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 1, pp. 239-249, 2013.
- [23] J. Mattingley, and S. Boyd, "CVXGEN: A Code Generator for Embedded Convex Optimization," in *Optim. Eng.*, vol. 13, no. 1, pp. 1-27, 2012.
- [24] X. Huang, H. Zhang, G. Zhang, and J. Wang, "Robust Weighted Gain-Scheduling H_∞ Vehicle Lateral Motion Control With Considerations of Steering System Backlash-Type Hysteresis," *IEEE Trans. On Control Systems Technology*, vol. 22, no. 5, pp. 1740 - 1753, 2014.
- [25] J. Wang and R. G. Longoria, "Coordinated and Reconfigurable Vehicle Dynamics Control," *IEEE Transactions on Control Systems Technology*, Vol. 17, No. 3, pp. 723 - 732, 2009.