Optimal Path Following Simulation of Autonomous Vehicles with Real-time Nonlinear Model Predictive Control

Jun-Ting Li[1], Chih-Keng Chen[1*] and Hongbin Ren[2]

***Abstract*—We designed a controller framework for autonomous racing vehicles that simultaneously coordinates the longitudinal and lateral vehicle dynamics. The system enables a vehicle to follow an curvy reference path at high speed while maintaining stability. First, the Cartesian coordinates of the reference path are transformed to curvilinear coordinates, and the decoupled vehicle heading and lateral error dynamics are established. The nonlinear tire characteristics are then used to construct a third-order vehicle dynamics model that accurately predicts the state of the vehicle. Finally, a ninth-order nonlinear model predictive controller (NMPC) model is developed by combining path tracking dynamics, vehicle dynamics, and actuator slew rates. To formulate the path-following task as an optimization problem, we developed a cost function comprising path tracking errors, control efforts, and constraints, such as the actuator bounds and tire grip limits. The acados solver could solve the NMPC problem in milliseconds. The optimal steering angle and the longitudinal acceleration command were obtained from the NMPC solution, and a lower speed controller implemented the acceleration command as either a rear-wheel driving torque or four-wheel braking torque. The CarSim simulation results revealed that the vehicle can follow a highly dynamic path at an average speed of 85 km/h while maintaining a small tracking error.**

***Index Terms*—Path Following, Nonlinear Model Predictive Controller (NMPC), Embedded Optimization Solver**

I. Introduction

Autonomous driving has become a popular research topic in recent years. It has the potential to revolutionize the way humans travel and greatly improve road safety by eliminating human error as a cause of accidents [1]. Advances in artificial intelligence, machine learning, and sensor technology [2] are also driving the development of autonomous driving technology. As these technologies continue to improve, developing fully autonomous vehicles that are capable of navigating roads safely and efficiently is becoming increasingly feasible. A core task in autonomous driving is path following, which is the ability of an autonomous vehicle to accurately track a planned path that is typically defined by a set of waypoints or trajectory points. A high-performance path-following controller enables an autonomous vehicle to operate safely and efficiently in complex and dynamic environments, such as those in obstacle avoidance and high-speed racing tasks.

[1]Department of Vehicle Engineering, National Taipei University of Technology, Taipei 10608, Taiwan.
[2]School of Mechanical Engineering, Beijing Institute of Technology, Beijing 100081, China (e-mail: renhongbin2106@126.com).
*Correspondence: ckchen@ntut.edu.tw
Digital Object Identifier (DOI): see top of this page.

Driving a vehicle at high speeds while accurately following a curvy road is a challenging task for autonomous systems. Several studies have investigated this problem. Hoffmann *et al.* [3] used the Stanley method, which is based on path geometry, and a vehicle kinematic model to control vehicle steering. They expected that lateral error could be reduced by eliminating the heading angle error; this method was highly successful in the Defense DARPA Grand Challenge. Later, curvilinear coordinates were introduced to decouple the heading angle error and lateral error to formulate a state-space model. Snider *et al.* [4] simulated this model by using LQ feedback combined with feedforward control, achieving a very small tracking error. Kapania *et al.* [5] designed a controller with a nonlinear tire feedforward term and state feedback loop and verified the controller in a real vehicle that precisely followed the reference path near its handling limits. Srinivasan *et al.* [6] proposed a holistic motion planning and controller framework that used NMPC to guide all vehicle motion; the method outperformed a professional racecar driver in a racing experiment.

Embedded optimization [7] plays a critical role in vehicle controllers because the vehicle's control system most process data on vehicle motion, road conditions, traffic, and other factors in real-time to optimize its route and driving behavior. The controller must also be able to provide real-time results when deployed in embedded devices. Several studies have used different controller frameworks to implement NMPC, such as C/GMRES [8], GRAMPC [9], IPOPT [10] and FORCEPRO [11]. All of these solvers have user-frendly Python or MATLAB interfaces for formulating NMPC problems and automatically generating the corresponding C code. To minimize the time required for a solution, a state-of-the-art embedded solver `acados` [12] is used in this article. In one study where `acados` was applied to the ignition engine control problem [13], the authors demonstrated that it completes computations faster than commercial solvers.

This article 1) creates a parametric representation of the reference path and a calculation method of the following error for discrete waypoints, 2) formulates a constrained optimization problem and present a normalized method for the cost function to enable solving the problem efficiently and rapidly, 3) uses state-of-the-art `acados` software to solve the NMPC problem and achieve a computation time in the millisecond range, and 4) validates the controller's performance in a high-speed racing scenario where it provided successful path tracking. Our goal is to provide a design workflow for a path-following controller

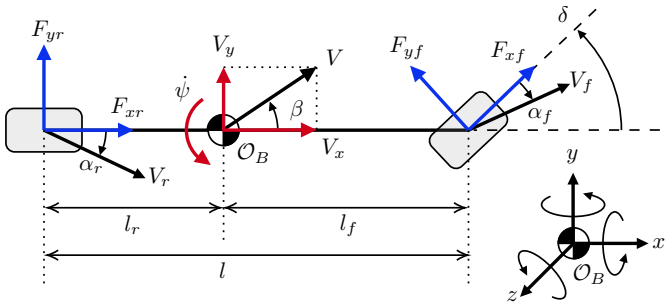

Fig. 1. Diagram of the single-track vehicle model. Red, state variables; blue, tire forces; arrows indicates the positive direction of the corresponding variable.

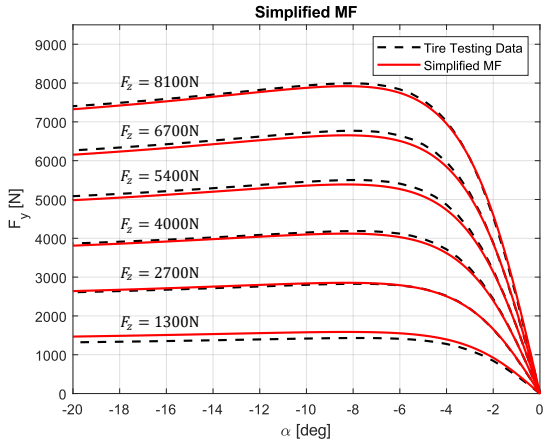

Fig. 2. Simplified Magic Formula tire model.

and eliminate doubts regarding the feasibility of implementing real-time NMPC. The open source sample code for the path-following problem is provided for other researchers and practitioners to easily reproduce this research, set up simulation environments, and explore additional applications of NMPC.

II. Vehicle Modeling

Numerous vehicle models can be selected for the path-following problem [14]. To achieve vehicle control near the handling limits, a vehicle model must capture the dynamic coupling of longitudinal and lateral motion as well as the nonlinearity of tire forces and relevant trigonometric functions.

Solving the path-following problem mainly requires characterizing the planar motion of vehicle; hence, vertical motion such as suspension dynamics can be neglected. To balance model fidelity and computational cost, the three degrees of freedom (3DOF) single-track vehicle model was selected for vehicle modeling.

A. Nonlinar Single-Track Vehicle Model

The layout of the single-track model is shown in Fig. 1. The vehicle body coordinate $\mathcal{O}_B$ is located at the vehicle's center of gravity (CG) position, and the sum of the moment and forces about $\mathcal{O}_B$ are as follows:

$$\sum M_z = l_f \left[F_{yf}\cos(\delta) + F_{xf}\sin(\delta)\right] - l_r F_{yr} \tag{1a}$$

$$\sum F_x = -F_{yf}\sin(\delta) + F_{xf}\cos(\delta) + F_{xr} - F_{\mathrm{d}} \tag{1b}$$

$$\sum F_y = F_{yf}\cos(\delta) + F_{xf}\sin(\delta) + F_{yr}, \tag{1c}$$

where δ is the front wheel steer angle; l is the wheelbase; l_f and l_r are the distance from the CG to the front and rear axles, respectively. For the lumped tire force F_{ij}, the subscript $i = \{x, y\}$ denote the longitudinal or lateral direction and the subscript $j = \{f, r\}$ denote the front or rear wheel. The drag force $F_{\mathrm{d}} = C_{\mathrm{d}} V_x^2$ is the multiplicative product of the compact aerodynamic coefficient and the square of the velocity.

To predict the planar vehicle motion about $\mathcal{O}_B$, we use the angular velocity $\dot{\psi}$ (yaw rate), longitudinal velocity V_x, and lateral velocity V_y to formulate dynamic equations:

$$\ddot{\psi} = \frac{\sum M_z}{I_z} \tag{2a}$$

$$\dot{V}_x = \frac{\sum F_x}{m} + V_y\dot{\psi} = a_x + V_y\dot{\psi} \tag{2b}$$

$$\dot{V}_y = \frac{\sum F_y}{m} - V_x\dot{\psi} = a_y - V_x\dot{\psi}, \tag{2c}$$

where m is the mass of the vehicle, I_z is the moment of inertia, and a_x and a_y are the measurable accelerations on $\mathcal{O}_B$.

B. Lateral Tire Force

The load transfer affects on the characteristics of the lateral tire force and must therefore be included to ensure the accuracy of the model (Fig. 2). For consistency with the single-track model, only longitudinal load transfer was considered. An algebraic loop occurs if the load transfer is calculated from a_x because a_x depends on the lateral tire force in (1b), which in turn depends on the load transfer. Instead, we use the steady-state approach by noting that $a_x = -V_y\dot{\psi}$ from (2b) to calculate the load transfer,

$$F_{zf} = m\left(gl_r + V_y\dot{\psi}h_c\right)/l \tag{3a}$$

$$F_{zr} = m\left(gl_f - V_y\dot{\psi}h_c\right)/l, \tag{3b}$$

where h_c is the height of the CG and g is the gravity constant.

The Simplified Magic Formula [15] with a few parameters is used to capture the nonlinear behavior of lateral tire force. The model is a function of the sideslip angle α, the vertical force F_z of a single wheel, and the road friction coefficient μ as follows:

$$F_{y0}(\alpha, F_z, \mu) = \mu D \sin\left[C\tan^{-1}(B\alpha)\right] \tag{4a}$$

$$D = d_1 F_z + d_2, \tag{4b}$$

where B, C, and D are fitting coefficients. The peak factor D is defined by the first-order function of F_z because it is mainly affected by the vertical tire force and varies linearly

with it. The lumped sideslip angles at the front and rear axles are described as follows:

$$\alpha_f = \tan^{-1}\left[\frac{V_y + l_f\dot{\psi}}{V_x}\right] - \delta \tag{5a}$$

$$\alpha_r = \tan^{-1}\left[\frac{V_y - l_r\dot{\psi}}{V_x}\right]. \tag{5b}$$

The combined tire lateral forces F_{yf} and F_{yr} in (6) can then be substituted into the single-track model (1).

$$F_{yi} = 2F_{y0}\left(\alpha_i, \frac{F_{zi}}{2}, \mu\right), \quad i \in \{f, r\}. \tag{6}$$

C. Longitudinal Tire Force

The longitudinal forces F_{xf} and F_{xr} comprise the rear-wheel traction force F_t and the all-wheel braking force F_b as in the following equations:

$$F_{xf} = k_b F_b \tag{7a}$$

$$F_{xr} = F_t + (1 - k_b)F_b, \tag{7b}$$

where k_b is the braking coefficient, which determines the force distribution between the front axle and rear axle. We set $k_b = F_{zf}/(mg)$ to distribute the braking force in accordance with the axle load ratio.

The inputs of the described system are δ, F_t, and F_b. These variables may have large differences in magnitude, which may casue numerical instability for an NMPC solver. Therefore, the normalized acceleration a_t and a_b are used in the model:

$$a_t = F_t/m, \quad a_b = F_b/m. \tag{8}$$

This transformation enables setting the scaling factors and bounds for the optimizer in a more intuitive manner.

After the optimal acceleration commands have been determined by NMPC, a lower controller solves for the desired longitudinal tire forces. The traction and braking torques are distributed proportionally to the relative vertical forces of each wheel as follows:

$$T_{t,rl} = \frac{F_{xr}^+}{2}\left(1 - \frac{\Delta F_{zr}}{F_{zr}}\right),\ T_{t,rr} = \frac{F_{xr}^+}{2}\left(1 + \frac{\Delta F_{zr}}{F_{zr}}\right), \tag{9a}$$

$$T_{b,fl} = \frac{F_{xf}^-}{2}\left(1 - \frac{\Delta F_{zf}}{F_{zf}}\right),\ T_{b,fr} = \frac{F_{xf}^-}{2}\left(1 + \frac{\Delta F_{zf}}{F_{zf}}\right), \tag{9b}$$

$$T_{b,rl} = \frac{F_{xr}^-}{2}\left(1 - \frac{\Delta F_{zr}}{F_{zr}}\right),\ T_{b,rr} = \frac{F_{xr}^-}{2}\left(1 + \frac{\Delta F_{zr}}{F_{zr}}\right). \tag{9c}$$

where the superscripts of $+$ and $-$ denote the positive or negative parts, respectively, of the longitudinal forces. The relative vertical loads are as follows:

$$\Delta F_{zf} = 2m(\frac{l_r}{l}g - \frac{h_c}{l}a_x)\frac{h_c}{Eg}a_y \tag{10a}$$

$$\Delta F_{zr} = 2m(\frac{l_f}{l}g + \frac{h_c}{l}a_x)\frac{h_c}{Eg}a_y. \tag{10b}$$

D. Constraints

During racing, we must ensure that the combined tire force is contained within the friction ellipse to avoid large tire slips and loss of grip. Thus, the following quadratic constraints are applied:

$$\frac{F_{xi}^2 + F_{yi}^2}{(\mu F_{zi})^2} \leq 1, \quad i \in \{f, r\}. \tag{11}$$

The system inputs also have upper bounds and lower bounds due to physical limitations. These bounds are expressed by the following inequality constraints:

$$\delta_{min} \leq \delta \leq \delta_{max} \tag{12a}$$

$$0 \leq a_t \leq \frac{T_{r,\max}}{r_w m} \tag{12b}$$

$$\frac{T_{b,\max}}{r_w m} \leq a_b \leq 0, \tag{12c}$$

where $T_{r,\max}$ is the maximum total traction torque of the rear axle, $T_{b,\max}$ is the maximum braking torque of each axle, and r_w is the effective rolling radius of the wheel.

To achieve optimal actuation efficiency, an equality constraint is imposed as follows:

$$a_t a_b = 0 \tag{13}$$

This equation ensures that the traction and braking commands are orthogonal (i.e., they are not active at the same time).

III. Path Modeling

In this section, the reference path is fitted by a cubic spline function to create a parametric representation. A curvilinear coordinate system is used to describe the relationship between the vehicle position and the reference waypoints. An algorithm is presented to accurately evaluate the path-tracking error for a discrete path.

A. Parametric Path

The discrete path data set $[\mathbf{X}, \mathbf{Y}]$ comprises the path coordinates $\mathbf{X} = [x_0, \ldots, x_n]^\top$ and $\mathbf{Y} = [y_0, \ldots, y_n]^\top$. A spline function $\mathcal{X}(s)$ with n equations is used to fit $\mathbf{X}$ as follows:

$$\mathcal{X}(s) = a_k(s - s_{k-1})^3 + b_k(s - s_{k-1})^2 + c_k(s - s_{k-1}) \quad k = 1, \ldots, n, \quad s \in [s_{k-1}, s_k], \tag{14}$$

where s is the cumulative arc length starting from $s_0 = 0$ as follows:

$$s_k = \sum_{i=1}^{k}\sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2},\ k = 1, \ldots, n. \tag{15}$$

A higher-order derivative of the spline function with respect to the progressive variable s is easily obtained as

$$\mathcal{X}'(s) = 3a_k(s - s_{k-1})^2 + 2b_k(s - s_{k-1}) + c_k \tag{16}$$

$$\mathcal{X}''(s) = 6a_k(s - s_{k-1}) + 2b_k \tag{17}$$

$$k = 1, \ldots, n, \quad s \in [s_{k-1}, s_k].$$

Per this procedure, $\mathcal{Y}(s)$, $\mathcal{Y}'(s)$, and $\mathcal{Y}''(s)$ can also be derived from $\mathbf{Y}$.

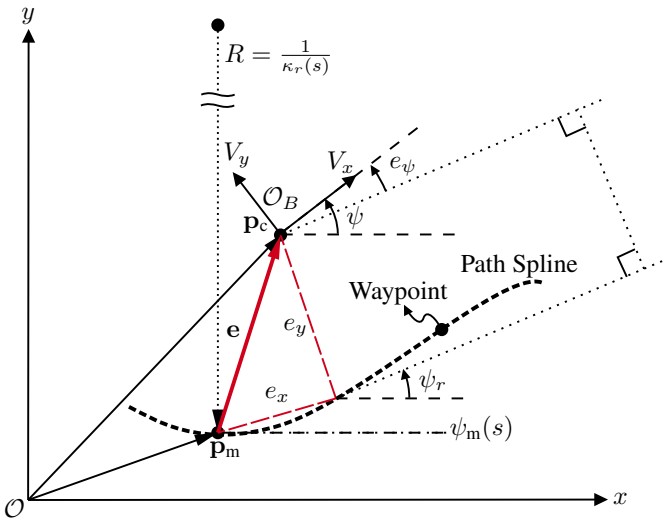

Fig. 3. Single-track vehicle model in curvilinear coordinates.

The reference heading angle $\boldsymbol{\psi}_r$ and the reference curvature $\boldsymbol{\kappa}_r$ are calculated as

$$\boldsymbol{\psi}_r = \arctan 2\left(\mathcal{Y}', \mathcal{X}'\right) \tag{18}$$

$$\boldsymbol{\kappa}_r = \frac{\mathcal{X}'\mathcal{Y}'' - \mathcal{X}''\mathcal{Y}'}{\left[\left(\mathcal{X}'\right)^2 + \left(\mathcal{Y}'\right)^2\right]^{3/2}}, \tag{19}$$

where $\arctan 2$ is the two-argument arctangent function.

B. Tracking Error for a Discrete Path

An overview of path-following model is displayed in Fig. 3. We wish to calculate the exact path-tracking error from the vehicle position $\mathbf{p}$ to its projection point on the reference path; however, only information on waypoints is available. Increasing the waypoint density could facilitate this calculation but would also increase the workload of algorithm. Instead, we use a similar approach to that of [16] to calculate the projection point.

The vehicle CG position is $\mathbf{p}_c = [x_c, y_c]^\top$, and the closest waypoint to $\mathbf{p}_c$ is defined as the "matching point," $\mathbf{p}_m = [x_m, y_m]^\top$, where m is the index of the matching point. The tracking error vector in global coordinates is defined as follows:

$$\mathbf{e} = \mathbf{p}_c - \mathbf{p}_m. \tag{20}$$

To obtain the longitudinal error e_x and the lateral error e_y, the error vector is projected onto curvilinear coordinates by a rotation matrix:

$$\begin{aligned} \begin{bmatrix} e_x \\ e_y \end{bmatrix} &= \mathbf{R}^\top(\psi_r)\,\mathbf{e} \\ \mathbf{R}(\psi_m) &= \begin{bmatrix} \cos(\psi_m) & -\sin(\psi_m) \\ \sin(\psi_m) & \cos(\psi_m) \end{bmatrix}, \end{aligned} \tag{21}$$

where ψ_m is the reference heading angle at $\mathbf{p_m}$. In general, the arc length error e_s can be approximated as the longitudinal error e_x. If the reference curvature of the matching point and projection point are assumed to be the same, a trick for compensating for the reference heading angle can be applied as recommended in [17]:

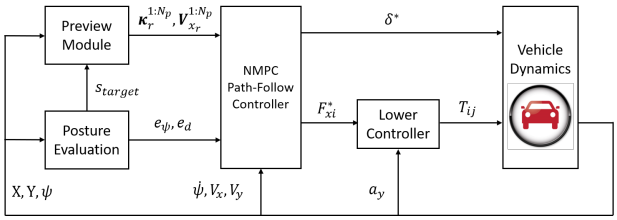

Fig. 4. Control architecture for NMPC path-following.

$$\psi_r = \psi_m + \kappa_r e_s. \tag{22}$$

If the vehicle sideslip angle β is required to be small during racing, the course angle can be approximated as the heading angle. The heading error is then

$$e_\psi = \psi - \psi_r. \tag{23}$$

C. Tracking Error Dynamics

To predict the future path-tracking error, the dynamics of the heading error and lateral error are expressed as follows:

$$\dot{e}_\psi = \dot{\psi} - \dot{\psi}_r = \dot{\psi} - \kappa_r \dot{s}, \tag{24a}$$

$$\dot{e}_y = V_y \cos(e_\psi) + V_x \sin(e_\psi) \tag{24b}$$

$$\dot{s} = \frac{V_x \cos(e_\psi) - V_y \sin(e_\psi)}{1 - \kappa_r e_y}, \tag{24c}$$

where $\dot{s}$ is the instantaneous tangential velocity at the projection point.

IV. Nonlinear Model Predictive Controller

In this section, the path following problem is transformed to a nonlinear programming (NLP) problem. To improve the efficiency and optimize the solution, we further explore the formulation of the NMPC, implementation details, and suitable parameters.

A. Control Architecture

The overall control structure for the path-following problem is presented in Fig. 4, where $i \in \{f, r\}$ and $j \in \{r, l\}$. The posture evaluation module outputs the path tracking errors e_ψ and e_d in accordance with the current position of the vehicle. The preview module looks forward over the preview path and sends the corresponding reference curvature and reference speed to the NMPC. The lower controller distributes the desired longitudinal forces command F_{xi}^* as the rear driving torque or four-wheel braking torque.

B. Prediction Model

With vehicle dynamics (2) and path-following dynamics (24) considered, the prediction model is formulated as

$$\dot{\mathbf{x}} = \begin{bmatrix} \ddot{\psi} \\ \dot{V}_x \\ \dot{V}_y \\ \dot{e}_\psi \\ \dot{e}_y \\ \dot{s} \\ \dot{\delta} \\ \dot{a}_t \\ \dot{a}_b \end{bmatrix} = \begin{bmatrix} \frac{\sum M_z}{I_z} \\ \frac{\sum F_x}{m} + V_y \dot{\psi} \\ \frac{\sum F_y}{m} - V_x \dot{\psi} \\ \dot{\psi} - \kappa_r \dot{s} \\ V_y \cos(e_\psi) + V_x \sin(e_\psi) \\ \frac{V_x \cos(e_\psi) - V_y \sin(e_\psi)}{1 - \kappa_r e_y} \\ \Delta\delta \\ \Delta a_t \\ \Delta a_b \end{bmatrix} \coloneqq \mathbf{f}(\mathbf{x}, \mathbf{u}, \mathbf{p}). \tag{25}$$

Given a prediction horizon N_p, $\mathbf{p} = \left[\kappa_r^1, \ldots, \kappa_r^{Np}\right]$ is the parameterized trajectory composed by the reference curvature at each prediction stage. The first six equations in (25) predict future vehicle states, path tracking errors, and driving distance. In the last three equations, we assign the slew rate of system inputs as extended states in a similar manner to [6] and [18]. This approach reduces drastic input changes, ensuring smooth command signals. Then, the manipulated variables of the NMPC are

$$\mathbf{u} = \begin{bmatrix} \Delta\delta & \Delta a_t & \Delta a_b \end{bmatrix}^\top. \tag{26}$$

C. Discretization

A processor can only perform discrete operations; therefore, the model must be discretized. Variable-step methods, such as fourth-order Runge-Kutta, may introduce noise into the numerical calculations. Instead, the multistep Euler method was used to discretize the model in a simple but precise manner:

$$\begin{cases} k_1 = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k, \mathbf{p}) \\ k_2 = \mathbf{f}(\mathbf{x}_k + hk_1, \mathbf{u}_k, \mathbf{p}) \\ \vdots \\ k_n = \mathbf{f}(\mathbf{x}_k + hk_{n-1}, \mathbf{u}_k, \mathbf{p}) \end{cases} \tag{27}$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + h \sum_{i=1}^{n} \mathbf{f}(\mathbf{x}_k) \tag{28}$$

$$\coloneqq \mathbf{f}_\mathrm{d}(\mathbf{x}_k, \mathbf{u}_k, \mathbf{p}), \quad k = 0, \ldots, N_p - 1, \tag{29}$$

where t_s is the NMPC sample time or, equivalently, the time interval for solving each optimization problem in (33). $h = t_s/n$ is the step size of Euler method. Smaller steps can be used to increase the model prediction accuracy. In this study, $t_s = 0.1$ s and $h = 0.025$ s.

D. Reference Outputs

The outputs of NMPC are $\mathbf{y} = [V_y, V_x, e_y]$, and the reference output trajectory is given in each stage of the prediction horizon N,

$$\mathbf{V}_y^\mathrm{ref} = \mathbf{0}^{1\times N_p} \tag{30a}$$

$$\mathbf{V}_x^\mathrm{ref} = \left[V_x^\mathrm{ref}(s_1^\mathrm{pv}), \ldots, V_x^\mathrm{ref}(s_{N_p}^\mathrm{pv})\right] \tag{30b}$$

$$\mathbf{e}_y^\mathrm{ref} = \mathbf{0}^{1\times N_p}. \tag{30c}$$

The reference lateral speed is set to the zero vector to improve lateral stability, and $\mathbf{V}_x^\mathrm{ref}$ is the preview reference speed over the prediction distance. For the path tracking error, heading error must be regulated. Lateral error must also be regulated because the lateral error dynamics (24b) are driven by lateral error.

E. Cost Function

Through the composition of the tracking and control costs over the prediction horizon, the cost function $\mathcal{J}$ to be minimized is defined as follows:

$$\mathcal{J} = \sum_{k=1}^{N_p} \frac{1}{2} \left\| \mathbf{S}_\mathbf{y}^{-1} \left(\mathbf{y}_k - \mathbf{y}_k^\mathrm{ref}\right) \right\|_\mathbf{Q}^2 + \sum_{k=0}^{N_p-1} \frac{1}{2} \left\| \mathbf{S}_\mathbf{u}^{-1} \mathbf{u}_k \right\|_\mathbf{R}^2, \tag{31}$$

where $\mathbf{S_y}$, and $\mathbf{S_u}$ are square matrices with diagonal scaling factors. These factors exist to normalize each variable; a simple method of selecting these factors is in terms of the maximum acceptable deviation from the nominal value. $\mathbf{Q}$ is a positive weighting matrix to penalize the difference between reference states and actual system states, $\mathbf{R}$ similarly penalizes control actions to ensure a smooth input trajectory. Their values in the method as implemented are as follows:

$$\mathbf{S_y} = \mathrm{diag}\,[2,\ 30,\ 0.3] \tag{32a}$$

$$\mathbf{S_u} = \mathrm{diag}\left[20\frac{\pi}{180},\ 20,\ 20\right] \tag{32b}$$

$$\mathbf{Q} = \mathrm{diag}\,[0.1,\ 10,\ 5] \tag{32c}$$

$$\mathbf{R} = \mathrm{diag}\,[5,\ 0.5,\ 0.5]. \tag{32d}$$

Finally, we collect cost function, prediction model, and constraints to formulate the path-following optimization problem:

$$\min_{\substack{\mathbf{x}_0,\ldots,\mathbf{x}_{N_p} \\ \mathbf{u}_0,\ldots,\mathbf{u}_{N_p-1}}} \quad \mathcal{J} \tag{33a}$$

$$\text{s.t.} \quad \mathbf{x}_0 = \hat{\mathbf{x}}(t) \tag{33b}$$

$$\mathbf{x}_{k+1} = \mathbf{f}_\mathrm{d}(\mathbf{x}_k, \mathbf{u}_k, \mathbf{p}) \tag{33c}$$

$$\text{constraints (11), (12), (13),} \tag{33d}$$

where $\hat{\mathbf{x}}(t)$ is the estimated or measured state at the current time. The problem (33) was solved using the software package `acados`, which can generate C code for an embedded NMPC for real-time applications. The `HPIPM` was selected as the sequential QP (SQP) solver. All programs were run on a desktop computer and given access to a single thread of an Intel Core i5-12500 processor. In each step, the problem can be solved within milliseconds.

V. Numerical Results

The commercial software CarSim is widely used to simulate complete vehicle dynamics. Its cosimulation interface with simulink enables easily implementing a closed-loop racing scenario to validate the NMPC performance.

A. Reference Path and Velocity Profile

The optimal reference raceline was generated by the minimum curvature algorithm, and the corresponding reference speed was generated by the quasi-steady-state lap time simulation tool. All of these programs are open source and available in [19]. The reference curvature and heading angle of the raceline were obtained by following the process in Section III-A.

B. Vehicle Configuration

The B-class sports car included in CarSim was used as the test vehicle in this study; this vehicle is a neutral-steering vehicle. The vehicle parameters are listed in Table I.

TABLE I
VEHICLE PARAMETERS.

m	1209 [kg]
I_z	1020 [kg-m^2]
l_f, l_r, h_c	1.165, 1.165, 0.35 [m]

C. Path Tracking Performance

Fig. 5 presents the overall path tracking response. The vehicle accelerated from the starting point and drove safely through the U-turn in Section ①. Section ② was a sharp corner that required a large yaw rate. In Section ③, the vehicle entered the L-turn at nearly 100 km/h. Its speed was maintained at over 100 km/h while accelerating through the straight line in Section ④. Finally, the vehicle smoothly drove through another U-turn in Section ⑤. The average speed of the vehicle was 85.23 km/h, and the lap time was 94.18 s.

Fig. 6 reveals that the lateral error in the all path sections was less than 0.1 m, and the heading error was below 4°; hence, the vehicle accurately followed the curved reference path.

The recorded vehicle state variables are presented in Fig. 7. The vehicle properly decelerated before turns and accelerated during straight paths. During the turn, the yaw rate was used to track the curvature profile while lateral stability was maintained by suppressing the sideslip angle to be below 2°. The speed profile shows that the vehicle tracked the reference speed at various curvatures.

Fig. 8 reveals that the NMPC output an appropriate steering angle at both low and high speeds to precisely control the lateral dynamics of the vehicle. Virtual acceleration commands controlled the vehicle to follow the reference speed profile and also met the orthogonal constraint.

Fig. 9 presents the traction and braking torque of the four wheels. The controller adjusted the torque command in accordance with the load transfer on each wheel. The smoothness

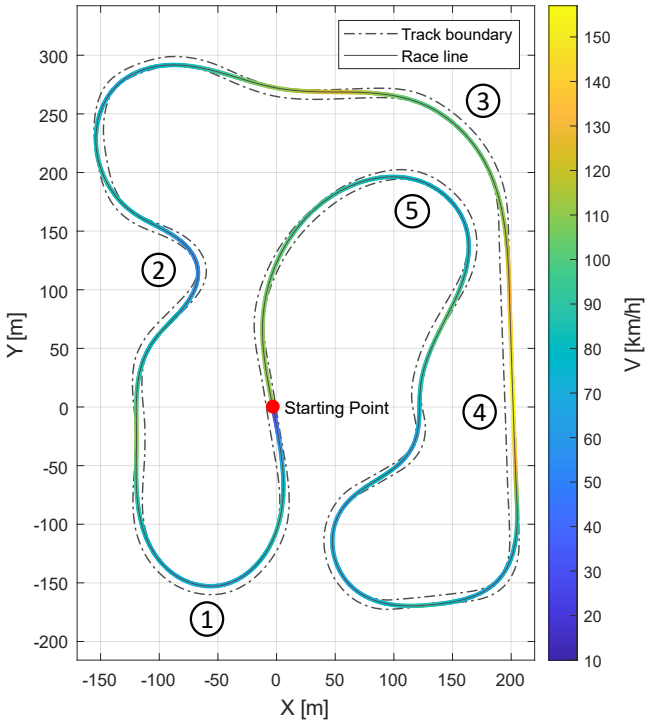

Fig. 5. Path-tracking results in Cartesian coordinates

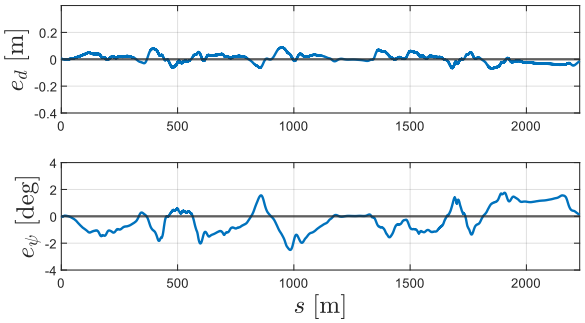

Fig. 6. Path tracking error response

of the signal helped the lower controller implement these commands.

Fig. 10 is the g-g-v diagram presenting the vehicle accelerations and velocity. The maximum lateral and longitudinal accelerations were both approximately $0.8\,g$; hence, the tire grip limit was almost reached, but a safety margin was preserved. The tire force did not exceed the limit of the tire ellipse during racing.

Adding preview information greatly improves the overall system damping of high-frequency oscillations at high speeds. The NMPC performed well in this study at 10 Hz.

D. Execution Performance

Fig. 11 presents the computation time of the NMPC solver; the mean and maximum solving times were 0.0018 and 0.0052

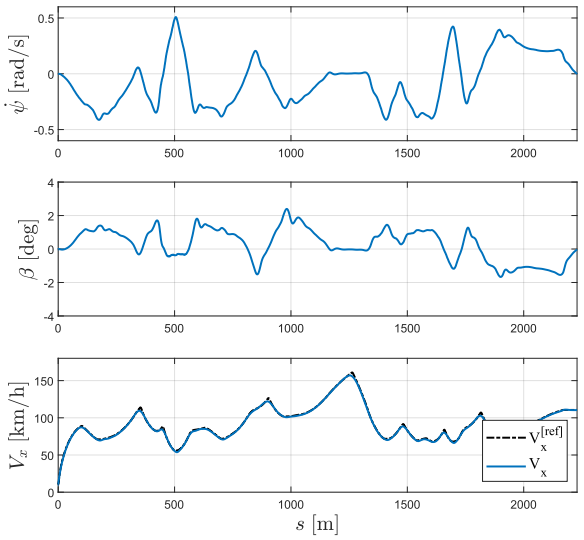

Fig. 7. Vehicle state response

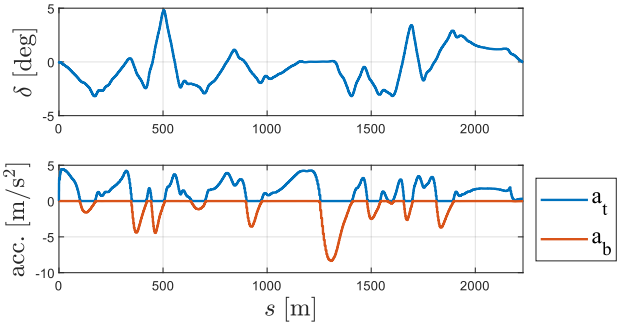

Fig. 8. Optimal NMPC commands

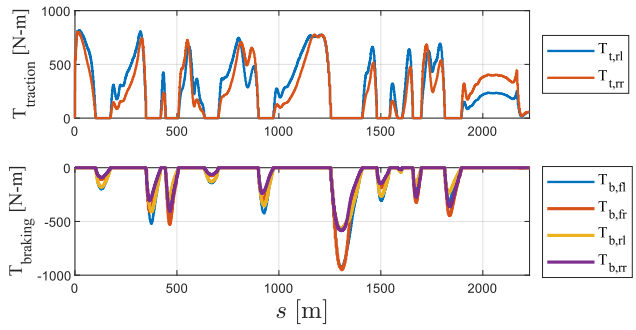

Fig. 9. Torque distribution of the low-level controller

s, respectively. The computation time was proportional to the number of SQP iterations, and the number of iterations increased when the vehicle entered a nonlinear region during high-speed cornering. However, the computation time was acceptable overall. This solver has great potential for real-time applications on embedded hardware.

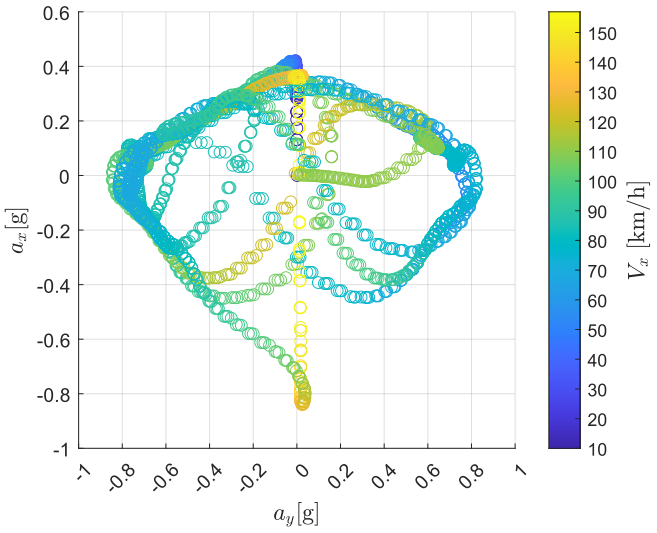

Fig. 10. g-g-v diagram for data recorded every 0.1 s

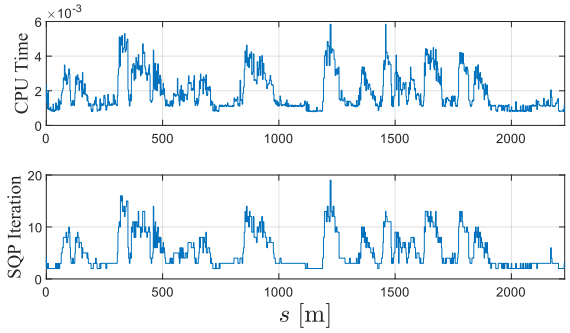

Fig. 11. Execution time and SQP iterations of the NMPC solver

VI. Conclusions

The proposed controller scheme coordinates the vehicle's lateral and longitudinal dynamics to follow the racing path at high speed. By combining path dynamics and vehicle dynamics to build a prediction model for NMPC, excellent path-following performance and driving stability were achieved. The CarSim simulation results demonstrates that the vehicle can follow a highly dynamic path at an average speed of 85 km/h while maintaining a small tracking error. The NMPC implemented by `acados` also can solve the path-following problem in real-time at a reasonable computational load.

In the future, we could further study vehicle dynamics and control under extreme handling situations, such as high-speed obstacle avoidance or drift control. However, further investigations of nonlinear systems are also required to determine the uncertainty of the tire model and road conditions.

Acknowledgments

The project is supported by the NTUT-BIT Joint Research Program under Grant No. NTUT-BIT-111-02 and the Taiwan National Science and Technology Council under Grant No. NSTC 111-2221-E-027-088.

REFERENCES

[1] A. S. Mueller, J. B. Cicchino, and D. S. Zuby, "What humanlike errors do autonomous vehicles need to avoid to maximize safety?" *Journal of safety research*, vol. 75, pp. 310–318, 2020.

[2] J. Kocić, N. Jovičić, and V. Drndarević, "Sensors and sensor fusion in autonomous vehicles," in *2018 26th Telecommunications Forum (TELFOR)*, 2018, pp. 420–425.

[3] G. M. Hoffmann, C. J. Tomlin, M. Montemerlo, and S. Thrun, "Autonomous automobile trajectory tracking for off-road driving: Controller design, experimental validation and racing," in *2007 American control conference*. IEEE, 2007, pp. 2296–2301.

[4] J. M. Snider *et al.*, "Automatic steering methods for autonomous automobile path tracking," *Robotics Institute, Pittsburgh, PA, Tech. Rep. CMU-RITR-09-08*, 2009.

[5] N. R. Kapania and J. C. Gerdes, "Design of a feedback-feedforward steering controller for accurate path tracking and stability at the limits of handling," *Vehicle System Dynamics*, vol. 53, no. 12, pp. 1687–1704, 2015.

[6] S. Srinivasan, S. N. Giles, and A. Liniger, "A holistic motion planning and control solution to challenge a professional racecar driver," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7854–7860, 2021.

[7] H. Ferreau, S. Almér, R. Verschueren, M. Diehl, D. Frick, A. Domahidi, J. Jerez, G. Stathopoulos, and C. Jones, "Embedded optimization methods for industrial automatic control**support by the eu via erc-highwind (259 166), itn-tempo (607 957), and itn-awesco (642 682) and by the dfg within reseach unit for 2401 is gratefully acknowledged." *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 13 194–13 209, 2017, 20th IFAC World Congress. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405896317325764

[8] T. Fu, H. Zhou, and Z. Liu, "Nmpc-based path tracking control strategy for autonomous vehicles with stable limit handling," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 12, pp. 12 499–12 510, 2022.

[9] H. Zhou, F. Jia, H. Jing, Z. Liu, and L. Güvenç, "Coordinated longitudinal and lateral motion control for four wheel independent motor-drive electric vehicle," *IEEE transactions on Vehicular Technology*, vol. 67, no. 5, pp. 3782–3790, 2018.

[10] M. Brunner, U. Rosolia, J. Gonzales, and F. Borrelli, "Repetitive learning model predictive control: An autonomous racing example," in *2017 IEEE 56th annual conference on decision and control (CDC)*. IEEE, 2017, pp. 2545–2550.

[11] J. Suh, H. Chae, and K. Yi, "Stochastic model-predictive control for lane change decision of automated driving vehicles," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 6, pp. 4771–4782, 2018.

[12] R. Verschueren, G. Frison, D. Kouzoupis, J. Frey, N. v. Duijkeren, A. Zanelli, B. Novoselnik, T. Albin, R. Quirynen, and M. Diehl, "acados—a modular open-source framework for fast embedded optimal control," *Mathematical Programming Computation*, vol. 14, no. 1, pp. 147–183, 2022.

[13] A. Norouzi, S. Shahpouri, D. Gordon, A. Winkler, E. Nuss, D. Abel, J. Andert, M. Shahbakhti, and C. R. Koch, "Deep learning based model predictive control for compression ignition engines," *Control Engineering Practice*, vol. 127, p. 105299, 2022.

[14] P. Stano, U. Montanaro, D. Tavernini, M. Tufo, G. Fiengo, L. Novella, and A. Sorniotti, "Model predictive path tracking control for automated road vehicles: A review," *Annual Reviews in Control*, 2022.

[15] M. Metzler, D. Tavernini, P. Gruber, and A. Sorniotti, "On prediction model fidelity in explicit nonlinear model predictive vehicle stability control," *IEEE transactions on control systems technology*, vol. 29, no. 5, pp. 1964–1980, 2020.

[16] Z. Wang, K. Sun, S. Ma, L. Sun, W. Gao, and Z. Dong, "Improved linear quadratic regulator lateral path tracking approach based on a real-time updated algorithm with fuzzy control and cosine similarity for autonomous vehicles," *Electronics*, vol. 11, no. 22, p. 3703, 2022.

[17] V. Wong, "Lecture note in automated-driving-control," *Github Repository*, [Online]. Available: https://github.com/VincentWong3/automated-driving-control.

[18] R. Lot and F. Biral, "A curvilinear abscissa approach for the lap time optimization of racing vehicles," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 7559–7565, 2014, 19th IFAC World Congress. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1474667016428041

[19] A. Heilmeier, "Quasi-steady-state lap time simulation," *Github Repository*, [Online]. Available: https://github.com/TUMFTM/laptime-simulation.