IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 9 October 2024, accepted 3 December 2024, date of publication 16 December 2024,
date of current version 26 December 2024.

*Digital Object Identifier 10.1109/ACCESS.2024.3515081*

RESEARCH ARTICLE

# Application of Optimal Control Method to Path Tracking Problem of Vehicle

**YINGJIE LIU**[1] **AND ENHAO WANG**[2]
[1]School of Machinery and Automation, Weifang University, Weifang, Shandong 261061, China
[2]Department of Engineering, Durham University, DH1 3LE Durham, U.K.

Corresponding author: Enhao Wang (enhao.wang@durham.ac.uk)

This work was supported by the Open Research Program of Huzhou Key Laboratory of Urban Multidimensional Perception and Intelligent Computing under Grant UMPIC202404.

**ABSTRACT** Path tracking is an essential stage for vehicle safety control. The study proposes an optimal control method for path tracking problem. Firstly, a nonlinear 4-DOF vehicle model is established. Secondly, the path optimization problem of vehicle is transformed into a nonlinear programming problem(NLP) by discretizing both control variables and state variables using the local collocation method. Then, in order to improve the efficiency of solving the NLP, an efficient calculation method for partial derivatives is established. Finally, a real vehicle test is executed to verify the rationality of the proposed model and methodology. The results show that by decomposing the partial derivative of the NLP into the partial derivative of the original path optimization problem the computational complexity of the first order partial derivative of NLP is significantly reduced improving computational efficiency even more significantly.

**INDEX TERMS** Vehicle dynamics, path tracking, optimal control, symplectic pseudospectral method.

## I. INTRODUCTION

The significant increase in vehicle ownership has brought about a series of problems such as traffic congestion and safety accidents, with the main influencing factors coming from drivers, such as physical condition, psychological state, and improper driving. Autonomous vehicle can transfer and share driving responsibilities, greatly reducing the probability of traffic accidents caused by human factors. Path tracking control, as a difficulty and key point in autonomous driving technology, has also become a hot topic of attention and research for scholars. Intelligent vehicles are considered an ideal solution to solve problems such as road safety accidents and traffic congestion. In recent years, they have been a key research object for governments, universities, and related enterprises in various countries. As one of the three core technologies of intelligent vehicles, the main function of path tracking system is to drive along the planned reference path and maintain the accuracy and stability of path tracking by controlling vehicle steering and speed based on the information provided by the environment perception and planning decision-making module. Path tracking control refers to the use of automatic steering control to ensure that the vehicle always follows the desired path while ensuring driving safety and comfort. Due to the nonlinear and strong coupling characteristics of vehicles, their lateral and longitudinal movements are highly coupled, making it difficult for vehicles to ensure both path tracking accuracy and driving stability. Driving stability is an important factor affecting safety and comfort. A series of path tracking control algorithms have been developed for these coupled optimization objectives. The autonomous vehicle puts forward high requirements for the accuracy of path tracking control, that is, it cannot deviate from the target path under all working conditions. Even if the road conditions are wet or the road is winding, the controller must have the ability to accurately track the target path, also known as the accuracy of path tracking, which is one of the important evaluation indicators of the path tracking algorithm [1], [2], [3].

The problem of vehicle path tracking has been widely studied. A brief review is presented in the following.

Gong et al. presented a cooperative vector field approach for synchronized path following of multiple autonomous surface vehicles with limited communication resources [4].The

The associate editor coordinating the review of this manuscript and approving it for publication was Jjun Cheng.

researcher did not study the problem of vehicle state. Jin et al. proposed a novel cooperative path planning scheme of unmanned surface vehicles for rescuing targets in a complex ocean environment [5]. The method reduced the impact of model accuracy on path tracking control. Rong et al. proposed a fractional-order sliding mode control method for an underactuated autonomous underwater vehicle with random disturbances [6]. Yan et al. proposed a whale optimization algorithm based on forward-looking sonar to tackle the three-dimensional path planning of autonomous underwater vehicles [7]. The method was not conducive to real-time path planning. Alejandro et al. presented a path-following and collision avoidance system for autonomous surface vehicles based on nonlinear model predictive control [8]. The researchers focused on nonlinear systems, but they neglected the influence of noise. Li et al. proposed a human-like motion planning strategy based on the probabilistic prediction in a dynamic environment to overcome the challenge of motion planning for autonomous vehicles becoming more challenging when both driver comfort and collision risk being considered [9].The algorithm could consider more tire constraints. To solve the issue of controlling the autonomous vehicle tracking the target path, Wang et al. put forward a control strategy combining MPC and genetic algorithm [10]. Bejarano et al. proposed a nonlinear model predictive control-based guidance strategy for unmanned surface vehicles, focused on path following [11]. Chen et al. proposed a longitudinal-lateral cooperative estimation algorithm based on the adaptive-square-root-cubature-Kalman-filter and partitioned similarity-principle to estimate the vehicle states and the tire-road peak adhesion coefficient sequentially for four-wheel-independent-drive-electric-vehicle [12]. Wang et al. proposed an innovative integrated path planning and trajectory tracking control framework for a quadrotor unmanned aerial vehicle in the presence of environmental and systematic uncertainties to achieve integrated guidance and control. Firstly, in order to perform real-time path planning, a computationally cost-effective planning algorithm was designed to find an optimal and smooth path while avoiding both static and dynamic obstacles [13]. These methods considered the vehicle state constraints, greatly improving the accuracy of path tracking. However, the complex constraint adjustment greatly increased the computational complexity of the controller. Qie et al. proposed a triggered forward optimal rapidly-exploring random tree method to plan a feasible path in a wide range of cross-country environments for the autonomous flying vehicles [14]. Ruslan et al. provided a review of path tracking strategies used in autonomous vehicle control design [15]. Zhou et al. investigated the use of a biquadratic Lyapunov function in formulating a model reference adaptive path-tracking controller which considered more constraints resulting in poor real time performance [16]. Rokonuzzaman et al. proposed a novel model predictive control framework for the effective adoption of vehicle models to achieve a compromise between MPC's performance and computational cost [17]. The algorithmic efficiency was low. Hu et al. put forward an explicit multiparameter optimal robust control law to ensure the uniform boundedness and ultimate uniform boundedness of the closed-loop path tracking dynamical system [18]. The algorithms were complex. Based on optimal control theory, Korayem and Nikoobin proposed novel methods for solving the problem of path planning of maximum payload for redundant manipulator, etc [19], [20], [21].

In response to the problem of low efficiency and time-consuming of traditional methods for directly calculating NLP partial derivatives, this paper researches the sparsity of the first-order partial derivatives(objective function gradient and constrained Jacobian matrix) of NLP obtained by local collocation method discretization, and establishes an efficient calculation method for non-zero elements. The simulation results show that compared with directly calculating the partial derivatives of NLP using traditional methods, the optimization accuracy of this method is higher. Therefore, the improvement the computational efficiency of first-order/second-order partial derivatives of NLP is of great significance for enhancing the real-time path planning and tracking performance of intelligent vehicles.

## II. MATHEMATICAL MODEL OF VEHICLE PATH-TRACKING PROBLEM

### A. MATHEMATICAL MODEL

A nonlinear 4-DOF vehicle model shown in Fig. 1 is used to describe the vehicle path tracking problem.

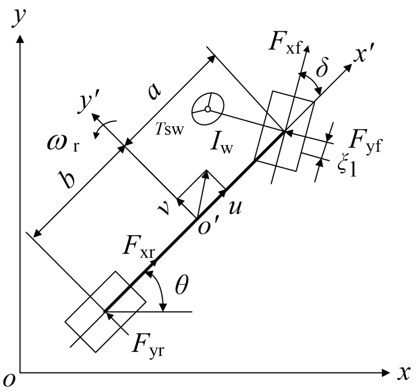


FIGURE 1. Nonlinear 4-DOF vehicle model.

In state space form it is [22]:

$$\begin{cases} \dot{v} = -u\omega + \dfrac{F_{\mathrm{yf}}\cos\delta + F_{\mathrm{yr}} + F_{\mathrm{xf}}\sin\delta}{m} \\ \dot{\omega} = \dfrac{aF_{\mathrm{yf}}\cos\delta - bF_{\mathrm{yr}} + aF_{\mathrm{xf}}\sin\delta}{I_z} \\ \dot{u} = v\omega + \dfrac{F_{\mathrm{xf}}\cos\delta - F_{\mathrm{yf}}\sin\delta + F_{\mathrm{xr}} - F_{\mathrm{f}} - F_{\mathrm{w}}}{m} \\ \dot{\delta} = p \\ \dot{p} = -\dfrac{k_1\xi_1}{I_w u}v - \dfrac{k_1\xi_1 a}{I_w u}\omega + \dfrac{(k_1\xi_1 - k_w)}{I_w}\delta - \dfrac{c_w}{I_w}p + \dfrac{T_{sw}i}{I_w} \end{cases} \tag{1}$$

The parameters and the corresponding definitions are shown in Table 1.

**TABLE 1. Parameter and definition.**

| Parameter | Definition |
|---|---|
| $v$ | lateral speed |
| $u$ | longitudinal speed |
| $\omega$ | yaw rate of the vehicle |
| $m$ | vehicle mass |
| $I_z$ | moment of inertia around the z axis |
| $k_w$ | synthesized cornering stiffness |
| $I_w$ | moment of inertia of the steering system |
| $a$ | distances of front axle from the center of gravity |
| $b$ | distances of rear axle from the center of gravity |
| $k_1$ | synthesized stiffness of front tire |
| $k_2$ | synthesized stiffness of rear tire |
| $i$ | transmission ratio of the steering system |
| $c_w$ | drag coefficient |
| $\xi_1$ | front wheel aligning arm of force |
| $\delta$ | front steering angle |
| $p$ | steering rate |
| $T_{sw}$ | steering torque |

The state variable is $\boldsymbol{x}(t) = [u(t), v(t), \omega(t), x(t), y(t)]$ and the control variable is $z(t) = [\delta]$. Then the Eq. (2) can be obtained by simplifying Eq. (1).

$$\dot{\boldsymbol{x}} = f[\boldsymbol{x}(t), z(t)] \tag{2}$$

### B. OPTIMAL CONTROL OBJECTIVE OF LANE PATH TRACKING PROBLEM

The minimum time required to complete the path tracking process is determined as the control object.

The cost function is:

$$J(z) = \int_{t_0}^{t_f} \mathrm{d}t \tag{3}$$

where $t_0$ is the initial time, $t_f$ is the final time.

### C. CONSTRAINTS

The initial and terminal states are described as:

$$\boldsymbol{x}(t_0) = [u(t_0), 0, 0, 0, 0]^T \tag{4}$$

$$\boldsymbol{x}(t_f) = [u(t_f), 0, 0, x(t_f), y(t_f)]^T \tag{5}$$

When the braking maneuver is applied to decelerate the vehicle, the constraints on $F_{\mathrm{xf}}, F_{\mathrm{xr}}$ can be rewritten in the following manner:

$$\left.\begin{array}{l} F_{\mathrm{xf}} \geq -\frac{\mu m g(b+\mu h_g)}{a+b} \\ F_{\mathrm{xr}} = \frac{a-\mu h_g}{b+\mu h_g} F_{\mathrm{xf}} \end{array}\right\} \tag{6}$$

So the optimal path tracking problem can be described as:

$$\min J;\ s.t.\dot{\boldsymbol{x}} = f[\boldsymbol{x}(t), u(t), t], h \leq 0 \tag{7}$$

where $h$ expresses the inequality constraint.

## III. MATHEMATICAL DESCRIPTION OF VEHICLE PATH TRACKING PROBLEM

The optimal path tracking problem is essentially an optimal control problem. Taking the Bolza type optimal control problem as an example, it can be described as: solving the control variable $\boldsymbol{u}(t) \in R^m$ to minimize the following objective function:

$$J = \psi(\boldsymbol{x}(t_0), t_0, \boldsymbol{x}(t_f), t_f) + \int_{t_0}^{t_f} L(\boldsymbol{x}(t), \boldsymbol{u}(t), t)dt \tag{8}$$

where $\psi : R^n \times R \times R^n \times R \rightarrow R$ is the Mayer cost function; $L : R^n \times R^m \times R \rightarrow R$ is the integrand of the Bolza cost function; $\boldsymbol{x}(t) \in R^n$; $\boldsymbol{u}(t) \in R^m$; $t \in [t_0, t_f] \subseteq R$.

For the sake of convenience, the vehicle minimum time maneuver collision avoidance problem should be transformed into a Bolza problem.

The dynamic constrain is:

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}(t), \boldsymbol{u}(t), t)t \in [t_0, t_f] \tag{9}$$

The boundary constrain is:

$$\Phi(\boldsymbol{x}(t_0), t_0, \boldsymbol{x}(t_f), t_f) = \boldsymbol{0} \tag{10}$$

The path constraint is:

$$C[\boldsymbol{x}(t), \boldsymbol{u}(t), t] \leq \boldsymbol{0}t \in [t_0, t_f] \tag{11}$$

where $f : R^n \times R^m \times R \rightarrow R^n$; $\Phi : R^n \times R \times R^n \times R \rightarrow R^q$; $C : R^n \times R^m \times R \rightarrow R^c$.

The problem described in Eqs. (8)-(11) is called a continuous Bolza type optimal control problem

## IV. DISCRETIZATION OF COLLOCATION METHOD BASED ON RUNGE-KUTTA FORMAT

Although the collocation method has multiple advantages, its specific implementation method has a significant impact on improving optimization efficiency. At present, gradient based NLP solvers represented by Sequential Quadratic Programming (SQP) need to provide first-order partial derivatives or even second-order partial derivatives of the objective function and constraints of NLP. Among them, the first-order partial derivative NLP solver needs to provide the first-order partial derivative, and then use the quasi Newton method to construct an approximate second-order partial derivative. The second-order partial derivative NLP solver not only requires first-order partial derivatives, but also accurate second-order partial derivatives. However, the computational complexity of partial derivatives is usually relatively large, even exceeding the optimization algorithm itself. Therefore, improving the computational efficiency of NLP first-order/second-order partial derivatives is of great significance for enhancing trajectory optimization efficiency.

Betts et al. studied the sparsity characteristics of local collocation methods earlier and obtained sparse forms of partial derivatives with discrete residual constraints for state equations in trapezoidal, Hermite Simpson, and Runge Kutta formats [23]. They also transformed non-zero elements into partial derivatives for optimal control problems, effectively reducing computational complexity. However, there were some shortcomings in the research: Firstly, they only provide the sparse characteristics of partial derivatives with discrete residual constraints on the state equation and the calculation methods for non-zero elements, without providing the sparse types and calculation methods for partial derivatives with objective function, path constraints, and endpoint constraints; Secondly, for the most commonly used Hermite Simpson scheme, the sparse form of the partial derivatives of the discrete residuals of the compact state equation has not been fully explored. Instead, the complete sparse form is obtained by adding discrete format constraints and state variables in the middle of the discrete nodes to transform it into a separated form.

Based on the research work of Betts, Patterson et al. studied the sparsity of the first and second partial derivatives of the Radau pseudospectral method, obtained a complete sparse form, and derived an efficient calculation method for non-zero elements (decomposing NLP partial derivatives into partial derivatives of the optimal control problem) [24].

Therefore, if the sparse form of NLP partial derivatives based on the above research can be fully explored and establishing an efficient calculation method for partial derivatives, it is of great significance for improving the optimization efficiency of local collocation methods.

Firstly, the integral transformation $\tau = (t - t_0)/(t_f - t_0)$ is used to transform the trajectory optimization problem (Eqs. (8)-(11)) into a time interval $\tau \in [0, 1]$. It is assumed that the nodes of $N$ discrete intervals on the unit interval [0,1] are:

$$G = \left\{ \begin{array}{l} \tau_i : \tau_i \in [0, 1], i = 0, 1, \cdots, N; \\ \tau_0 = 0, \tau_N = \tau_f = 1; \\ \tau_i < \tau_{i+1}, i = i = 0, 1, \cdots, N - 1 \end{array} \right\} \tag{12}$$

where $\tau_i$ is called a node or grid point, $\tau_i$ can be uniformly distributed or non-uniformly distributed on [0,1].

It is set that $x_i = x(\tau_i), u_i = u(\tau_i)$. Then the dynamic equation can be transformed as Eq. (13) based on the $q$- oder Runge-Kutta method.

$$x_{i+1} = x_i + (\Delta t)h_i \sum_{j=1}^{q} \beta_j f_{ij}, \quad (i = 0, 1, \cdots, N - 1) \tag{13}$$

where $\Delta t = t_f - t_0$;$h_i = \tau_{i+1} - \tau_i$;$f_{ij} = f(x_{ij}, u_{ij}, \tau_{ij}; t_0, t_f)$;$x_{ij}$, $u_{ij}$ and $\tau_{ij}$ are intermediate variables on $[\tau_i, \tau_{i+1}]$.

And $x_{ij}$ can be described as:

$$x_{ij} = x_i + (\Delta t)h_i \sum_{l=1}^{q} \alpha_{jl} f_{il}, 1 \le j \le q \tag{14}$$

where $\tau_{ij} = \tau_i + h_i \rho_j$;$u_{ij} = u(\tau_{ij})$. $\rho_j$, $\beta_j$ and $\alpha_{jl}$ are known constants and satisfying that $0 \le \rho_1 \le \rho_2 \le \cdots \le \rho_q \le 1$. When $\alpha_{jl} = 0 (l \ge j)$, the discrete format is explicit, otherwise it is implicit. By using a similar method, the objective function can be discretized. Commonly used discrete formats include trapezoidal format ($q = 2$), Hermite-Simpson format ($q = 3$), and classical fourth order Runge-Kutta format ($q = 4$).

The nonlinear programming problem obtained by discretization is to solve variables $\boldsymbol{X}$, $\boldsymbol{U}$, $\bar{\boldsymbol{U}}$, $t_0$ and $t_f$ to minimize the following objective functions:

$$J = \psi(x_0, t_0, x_f, t_f) + \Delta t \sum_{i=0}^{N-1} h_i \sum_{j=1}^{q} \beta_j L_{ij} \tag{15}$$

And satisfying the following constrains:

$$\boldsymbol{\xi}_i = \boldsymbol{x}_{i+1} - \boldsymbol{x}_i - (\Delta t)h_i \sum_{j=1}^{q} \beta_j \boldsymbol{f}_{ij} = 0, (i = 0, 1, \cdots, N - 1) \tag{16}$$

$$\boldsymbol{C}_i = \boldsymbol{C}(\boldsymbol{x}_i, \boldsymbol{u}_i, \tau_i; t_0, t_f) \le 0, (i = 0, 1, \cdots, N) \tag{17}$$

$$\bar{\boldsymbol{C}}_{ij} = \boldsymbol{C}(\boldsymbol{x}_{ij}, \boldsymbol{u}_{ij}, \tau_{ij}; t_0, t_f) \le 0, \tau_{ij} \in \bar{\boldsymbol{G}} \tag{18}$$

$$\boldsymbol{\Phi}(\boldsymbol{x}_0, t_0, \boldsymbol{x}_f, t_f) = 0 \tag{19}$$

where $L_{ij} = L(\boldsymbol{x}_{il}, \boldsymbol{u}_{il}, \boldsymbol{\tau}_{il}), i = 0, 1, \cdots, N_\tau - 1$;$\boldsymbol{X} = \{\boldsymbol{x}_0, \boldsymbol{x}_1, \cdots, \boldsymbol{x}_N\}$;$\boldsymbol{U} = \{\boldsymbol{u}_0, \boldsymbol{u}_1, \cdots, \boldsymbol{u}_N\}$;$\bar{\boldsymbol{G}} = \left\{\tau_{ij} \in [0, 1] : \tau_{ij} \notin \boldsymbol{G}, i = 0, 1, \cdots, N - 1, 1 < j < q\right\}$;$\bar{\boldsymbol{X}} = \left\{\boldsymbol{x}_{ij} : \tau_{ij} \in \bar{\boldsymbol{G}}\right\}$;$\bar{\boldsymbol{U}} = \left\{\boldsymbol{u}_{ij} : \tau_{ij} \in \bar{\boldsymbol{G}}\right\}$.

The commonly used discrete formats include trapezoidal format ($q = 2$), Hermite Simpson format ($q = 3$, abbreviated as HS format), and classical fourth order RungeKutta format ($q = 4$, abbreviated as RK format).

Taking the HS format as an example, this format requires the use of variables and function values at the midpoint of the interval. Therefore, the control variables at the midpoint of the interval are used as optimization variables, and path constraints are added at the midpoint of the interval:

$$\bar{\boldsymbol{G}} = \{\tau_{i+1} = \tau_i + h_i/2, i = 0, 1, \cdots, N - 1\} \tag{20}$$

The optimization variable and the objective function of the NLP obtained in HS format are:

$$(\boldsymbol{x}_0, \boldsymbol{x}_1, \cdots, \boldsymbol{x}_N; \boldsymbol{u}_0, \boldsymbol{u}_1, \cdots, \boldsymbol{u}_N; \bar{\boldsymbol{u}}_1, \bar{\boldsymbol{u}}_2, \cdots \bar{\boldsymbol{u}}_N; t_0, t_f) \tag{21}$$

And the objective function is:

$$J = \psi(x_0, t_0, x_f, t_f) + \Delta t \sum_{i=0}^{N-1} \frac{h_i}{6}(L_i + 4\bar{L}_{i+1} + L_{i+1}) \tag{22}$$

And the constraints are:

$$\boldsymbol{\xi}_i = \boldsymbol{x}_{i+1} - \boldsymbol{x}_i - (\Delta t)\frac{h_i}{6}(f_i + 4\bar{f}_{i+1} + f_{i+1}) = 0 \tag{23}$$

$$\boldsymbol{C}_i = \boldsymbol{C}(\boldsymbol{x}_i, \boldsymbol{u}_i, \tau_i; t_0, t_f) \le 0 \tag{24}$$

$$\bar{\boldsymbol{C}}_{i+1} = \boldsymbol{C}(\bar{\boldsymbol{x}}_{i+1}, \bar{\boldsymbol{u}}_{i+1}, \bar{\boldsymbol{\tau}}_{i+1}; t_0, t_f) \le 0 \tag{25}$$

$$\boldsymbol{\Phi}(\boldsymbol{x}_0, t_0, \boldsymbol{x}_f, t_f) = 0 \tag{26}$$

where $\bar{\boldsymbol{x}}_{i+1} = \frac{(\boldsymbol{x}_i + \boldsymbol{x}_{i+1})}{2} + \frac{h_i}{8}(\boldsymbol{f}_i - \boldsymbol{f}_{i+1})$; $\bar{\boldsymbol{f}}_{i+1} = \boldsymbol{f}(\bar{\boldsymbol{x}}_{i+1}, \bar{\boldsymbol{u}}_{i+1}, \bar{\boldsymbol{\tau}}_{i+1}; t_0, t_f)$; $\bar{L}_{i+1} = L(\bar{\boldsymbol{x}}_{i+1}, \bar{\boldsymbol{u}}_{i+1}, \bar{\boldsymbol{\tau}}_{i+1}; t_0, t_f)$.

In the processing of numerical optimization, in order to make the problem have practical physical significance, it is also necessary to add a time difference constraint:

$$\Delta t = t_f - t_0 > 0 \tag{27}$$

## V. PARTIAL DERIVATIVE CALCULATION METHOD OF THE NLP

### A. DEPENDENCY MATRIX

When deriving the sparse characteristics of the first order partial derivative of NLP, it is necessary to use the dependence of the original trajectory optimization problem on the independent variable.

Due to the fact that the state equation, path constraint, and Lagrange integral terms of the objective function are all defined throughout the entire time domain, this article defines the partial derivatives of these three terms on the independent variable together:

$$\boldsymbol{G}_1 = \begin{bmatrix} \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}^T} & \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}^T} & \frac{\partial \boldsymbol{f}}{\partial t} \\ \frac{\partial \boldsymbol{C}}{\partial \boldsymbol{x}^T} & \frac{\partial \boldsymbol{C}}{\partial \boldsymbol{u}^T} & \frac{\partial \boldsymbol{C}}{\partial t} \\ \frac{\partial \boldsymbol{L}}{\partial \boldsymbol{x}^T} & \frac{\partial \boldsymbol{L}}{\partial \boldsymbol{u}^T} & \frac{\partial \boldsymbol{L}}{\partial t} \end{bmatrix} \tag{28}$$

Each term of $\boldsymbol{G}_1$ is still a matrix. Taking $\frac{\partial f}{\partial \boldsymbol{x}^T}$ as an example:

$$\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}^T} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_3}{\partial x_n} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \end{bmatrix} \tag{29}$$

It is easy to know that $\boldsymbol{G}_1$ is a $(n + c + 1) \times (n + m + 1)$ vector.

Normally, $\boldsymbol{G}_1$ is a sparse matrix. In order to describe the sparsity of $\boldsymbol{G}_1$, the following struct function is defined:

$$struct(x) = \begin{cases} 1 & x \neq 0 \\ 0 & x = 0 \end{cases} \tag{30}$$

It is set that

$$S_1 = struct(G_1) \tag{31}$$

where $S_1 = struct(G_1)$ represents the struct operation on each element of $G_1$. $S_1$ represents the sparse type of $G_1$. In order to obtain $S_1$, it is not necessary to calculate the specific values of each element of $G_1$, only to determine whether it is 0.

Similar to the dependency matrix and sparse form of the Mayer term on independent variables that can define endpoint constraints and objective functions:

$$\boldsymbol{G}_2 = \begin{bmatrix} \frac{\partial \boldsymbol{E}}{\partial \boldsymbol{x}_0^T} & \frac{\partial \boldsymbol{E}}{\partial t_0} & \frac{\partial \boldsymbol{E}}{\partial \boldsymbol{x}_f^T} & \frac{\partial \boldsymbol{E}}{\partial t_f} \\ \frac{\partial \boldsymbol{M}}{\partial \boldsymbol{x}_0^T} & \frac{\partial \boldsymbol{M}}{\partial t_0} & \frac{\partial \boldsymbol{M}}{\partial \boldsymbol{x}_f^T} & \frac{\partial \boldsymbol{M}}{\partial t_f} \end{bmatrix} \tag{32}$$

It is easy to know that $\boldsymbol{G}_2$ is a $N \times 1$ vector.

### B. VARIABLE NOTATION

The aforementioned discrete format records variables or constraints at the same node as a column vector, which is consistent with the form of the numerical integration format, but is not conducive to deriving the sparse characteristics of the partial derivative matrix. Therefore, this article defines a new variable notation that records the values of the same component of a variable or constraint at different nodes as a new vector.

Taking the discrete residual of the state equation as an example, Eq. (33) is defined.

$$\boldsymbol{\xi}_{:,j} = (\xi_{0,j}, \xi_{1,j}, \cdots, \xi_{N-1,j})^T, (j = 1, 2, \cdots, n) \tag{33}$$

where the subscript of $\boldsymbol{\xi}_{i,j}$ represents the $i^{th}$ node, and the second subscript represents the $j^{th}$ component. It is easy to know that $\boldsymbol{\xi}_{:,j}$ is a $N \times 1$ vector.

Similarly, $C_{:,j}$, $\bar{C}_{:,j}$, $x_{:,j}$, $u_{:,j}$, $\bar{u}_{:,j}^T$, $t_:$, $\bar{t}_:$, $\tau$, $L_:$ and $\bar{L}_:$ can be defined.

By utilizing this new variable notation, the NLP problem obtained by discretizing the HS format can be rephrased as: solving the optimization variable $z \in R^{(N+1)\cdot(n+m)+Nm+2}$ to minimize the following objective function:

$$J(z) \tag{34}$$

And satisfying the constraint conditions:

$$F_{\min} \leq F(z) \leq F_{\max} \tag{35}$$

where the expression of the objective function $J(z)$ can be found in the equation, and the definitions of the optimization variable z and the constraint function $F(z)$ are as follows:

$$z = \begin{bmatrix} \boldsymbol{x}_{:,1} \\ \vdots \\ \boldsymbol{x}_{:,n} \\ \boldsymbol{u}_{:,1} \\ \vdots \\ \boldsymbol{u}_{:,m} \\ \bar{\boldsymbol{u}}_{:,1} \\ \vdots \\ \bar{\boldsymbol{u}}_{:,m} \\ t_0 \\ t_f \end{bmatrix} \boldsymbol{F} = \begin{bmatrix} \boldsymbol{\xi}_{:,1} \\ \vdots \\ \boldsymbol{\xi}_{:,n} \\ \boldsymbol{C}_{:,1} \\ \vdots \\ \boldsymbol{C}_{:,c} \\ \bar{\boldsymbol{C}}_{:,1} \\ \vdots \\ \bar{\boldsymbol{C}}_{:,c} \\ \boldsymbol{E} \\ \Delta t \end{bmatrix} \tag{36}$$

### C. GRADIENT OF THE OBJECTIVE

The objective function gradient refers to the partial derivative of the objective function over the optimization variable, which is specifically defined as follows:

$$\begin{aligned} \frac{\partial J}{\partial z^T} = [&\frac{\partial J}{\partial x_{:,1}^T}, \cdots, \frac{\partial J}{\partial x_{:,n}^T}, \frac{\partial J}{\partial u_{:,1}^T}, \cdots, \\ &\frac{\partial J}{\partial u_{:,n}^T}, \frac{\partial J}{\partial \bar{u}_{:,1}^T}, \cdots, \frac{\partial J}{\partial \bar{u}_{:,n}^T}, \frac{\partial J}{\partial t_0}, \frac{\partial J}{\partial t_f}] \end{aligned} \tag{37}$$

Eq. (38) can be obtained by writing the objective function in matrix product form:

$$J = \psi(x_0, t_0, x_f, t_f) + D\frac{\Delta t}{6}h(D_2 L_: + 4\bar{L}_:) \tag{38}$$

where $D = [1, 1, \cdots, 1]_{1xN}$; $\bar{x}_{:,i} = \frac{1}{2}D_2 x_{:,i} - \frac{\Delta t}{8}hD_1 f_{:,i}$.

And the diagonal matrix $h = diag(h_0, h_1, \cdots, h_{N-1})$; $h_i (i = 0, 1, \cdots, N - 1)$ is the integral step size.

The matrices $\boldsymbol{D}_1$ and $\boldsymbol{D}_2$ are defined as follows:

$$\boldsymbol{D}_1 = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & \ddots & \ddots & 0 \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix} \tag{39}$$

$$\boldsymbol{D}_2 = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & \ddots & \ddots & 0 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} \tag{40}$$

$\boldsymbol{D}_1$ and $\boldsymbol{D}_2$ are both $N \times (N+1)$ matrix, and blank elements are 0.

By applying the vector chain differentiation rule, the partial derivatives of the objective function for $x_{:,j}^T$, $u_{:,j}^T$, $\bar{u}_{:,j}^T$, $t_0$ and $t_f$ can be derived as:

$$\begin{cases} \dfrac{\partial J}{\partial x_{:,j}^T} = \dfrac{\partial M}{\partial x_{:,j}^T} + D\dfrac{\Delta t}{6}h(D_2\dfrac{\partial L_:}{\partial x_{:,j}^T} + 4\dfrac{\partial \bar{L}_:}{\partial x_{:,j}^T}) \\ \dfrac{\partial J}{\partial u_{:,j}^T} = D\dfrac{\Delta t}{6}h(D_2\dfrac{L_:}{\partial u_{:,j}^T} + 4\dfrac{\partial \bar{L}_:}{\partial u_{:,j}^T}) \\ \dfrac{\partial J}{\partial \bar{u}_{:,j}^T} = D\dfrac{\Delta t}{6}h(4\dfrac{\partial \bar{L}_:}{\partial \bar{u}_{:,j}^T}) \\ \dfrac{\partial J}{\partial t_0} = \dfrac{\partial M}{\partial t_0} - D\dfrac{1}{6}h(D_2L_: + 4\bar{L}_:) + \\ D\dfrac{\Delta t}{6}h(D_2\dfrac{\partial L_:}{\partial t_:^T}(-\tau_: + 1) + 4\dfrac{\partial \bar{L}_:}{\partial t_0}) \\ \dfrac{\partial J}{\partial t_f} = \dfrac{\partial M}{\partial t_f} + D\dfrac{1}{6}h(D_2L_: + 4\bar{L}_:) + \\ D\dfrac{\Delta t}{6}h(D_2\dfrac{\partial L_:}{\partial t_:^T}\tau_: + 4\dfrac{\partial \bar{L}_:}{\partial t_f}) \end{cases} \tag{41}$$

where

$$\begin{aligned} \frac{\partial \bar{L}_:}{\partial x_{:,j}^T} &= \sum_{l=1}^{n} \frac{\partial \bar{L}_:}{\partial \bar{x}_{:,l}^T}(\frac{1}{2}D_2\frac{\partial x_{:,l}}{\partial x_{:,j}^T} - \frac{\Delta t}{8}hD_1\frac{\partial f_{:,l}}{\partial x_{:,j}^T}) \\ \frac{\partial \bar{L}_:}{\partial u_{:,j}^T} &= \sum_{l=1}^{n} \frac{\partial \bar{L}_:}{\partial \bar{x}_{:,j}^T}(-\frac{\Delta t}{8}hD_1\frac{\partial f_{:,l}}{\partial u_{:,j}^T}) \\ \frac{\partial \bar{L}_:}{\partial t_0} &= \sum_{l=1}^{n} \frac{\partial \bar{L}_{:,i}}{\partial \bar{x}_{:,l}^T}(\frac{1}{8}hD_1f_{:,l} - \frac{\Delta t}{8}hD_1\frac{\partial f_{:,l}}{\partial t_:^T}(-\tau_: + 1)) \\ &\quad + \frac{\partial \bar{L}_{:,i}}{\partial \bar{t}_:^T}\frac{1}{2}D_2(-\tau_: + 1) \\ \frac{\partial \bar{L}_:}{\partial t_f} &= \sum_{l=1}^{n} \frac{\partial \bar{L}_{:,i}}{\partial \bar{x}_{:,l}^T}(\frac{1}{8}hD_1f_{:,l} - \frac{\Delta t}{8}hD_1\frac{\partial f_{:,l}}{\partial t_:^T}\tau_:) \\ &\quad + \frac{\partial \bar{L}_{:,i}}{\partial \bar{t}_:^T}\frac{1}{2}D_2\tau_: \end{aligned} \tag{42}$$

It can be seen that the objective function gradient of NLP can be decomposed into the objective function of trajectory optimization problems and the partial derivatives of the state equation.

### D. JACOBIAN MATRIX

The Jacobian matrix of NLP is defined as the partial derivative matrix of constraints of the NLP on optimization variables. For the HS format, the form is as follows:

$$G_F = \frac{\partial F}{\partial z^T} \tag{43}$$

The definitions of vectors F and z can be found in Eq. (36). The expansion form of the Jacobian matrix $G_F$ follows the rules for vector partial derivative operations (Eq. (29)). The mathematical expression for $G_F$ is derived below.

#### 1) PARTIAL DERIVATIVES OF DISCRETE RESIDUAL CONSTRAINTS FOR STATE EQUATIONS

Combining the variable notation defined above, the discrete residual constraint of the state equation can be written in the following form:

$$\xi_{:,i} = D_1x_{:,i} - \frac{\Delta t}{6}hD_2f_{:,i} + 4\bar{f}_{:,i} \tag{44}$$

The definitions of matrices $\boldsymbol{D}_1$ and $\boldsymbol{D}_2$ are the same as those in the equation. Eqs.(45) can be obtained by deriving the partial derivatives of the equations for $x_{:,j}^T$, $u_{:,j}^T$, $\bar{u}_{:,j}^T$, $t_0$ and $t_f$ respectively.

$$\begin{aligned} \frac{\partial \xi_{:,i}}{\partial x_{:,j}^T} &= D_1\frac{\partial x_{:,i}}{\partial x_{:,j}^T} - \frac{\Delta t}{6}h(D_2\frac{\partial f_{:,i}}{\partial x_{:,j}^T} + 4\frac{\partial \bar{f}_{:,i}}{\partial x_{:,j}^T}) \\ \frac{\partial \xi_{:,i}}{\partial u_{:,j}^T} &= -\frac{\Delta t}{6}h(D_2\frac{\partial f_{:,i}}{\partial u_{:,j}^T} + 4\frac{\partial \bar{f}_{:,i}}{\partial u_{:,j}^T}) \\ \frac{\partial \xi_{:,i}}{\partial \bar{u}_{:,j}^T} &= -\frac{\Delta t}{6}h(4\frac{\partial \bar{f}_{:,i}}{\partial \bar{u}_{:,j}^T}) \\ \frac{\partial \xi_{:,i:}}{\partial t_0} &= \frac{1}{6}h(D_2f_{:,i} + 4\bar{f}_{:,i}) \\ &\quad - \frac{\Delta t}{6}h(D_2\frac{\partial f_{:,i}}{\partial t_:^T}(-\tau_: + 1) + 4\frac{\partial \bar{f}_{:,i}}{\partial t_0}) \\ \frac{\partial \xi_{:,i:}}{\partial t_f} &= -\frac{1}{6}h(D_2f_{:,i} + 4\bar{f}_{:,i}) \\ &\quad - \frac{\Delta t}{6}h(D_2\frac{\partial f_{:,i}}{\partial t_:^T}\tau_: + 4\frac{\partial \bar{f}_{:,i}}{\partial t_f}) \end{aligned} \tag{45}$$

where

$$\begin{aligned} \frac{\partial \bar{f}_{:,i}}{\partial x_{:,j}^T} &= \sum_{l=1}^{n} \frac{\partial \bar{f}_{:,i}}{\partial \bar{x}_{:,l}^T}(\frac{1}{2}D_2\frac{\partial x_{:,l}}{\partial x_{:,j}^T} - \frac{\Delta t}{8}hD_1\frac{\partial f_{:,l}}{\partial x_{:,j}^T}) \\ \frac{\partial \bar{f}_{:,i}}{\partial u_{:,j}^T} &= \sum_{l=1}^{n} \frac{\partial \bar{f}_{:,i}}{\partial \bar{x}_{:,l}^T}(-\frac{\Delta t}{8}hD_1\frac{\partial f_{:,l}}{\partial u_{:,j}^T}) \\ \frac{\partial \bar{f}_{:,i}}{\partial t_0} &= \sum_{l=1}^{n} \frac{\partial \bar{f}_{:,i}}{\partial \bar{x}_{:,l}^T}(\frac{1}{8}hD_1f_{:,l} - \frac{\Delta t}{8}hD_1\frac{\partial f_{:,l}}{\partial t_:^T}(-\tau_: + 1)) \\ &\quad + \frac{\partial \bar{f}_{:,i}}{\partial \bar{t}_:^T}\frac{1}{2}D_2(-\tau_: + 1) \end{aligned}$$

$$\frac{\partial \bar{f}_{:,i}}{\partial t_f} = \sum_{l=1}^{n} \frac{\partial \bar{f}_{:,i}}{\partial \bar{x}_{:,l}^T}(-\frac{1}{8}hD_1 f_{:,l} - \frac{\Delta t}{8}hD_1 \frac{\partial f_{:,l}}{\partial t_{:}^T}\tau_{:}) + \frac{\partial \bar{f}_{:,i}}{\partial \bar{t}_{:}^T}\frac{1}{2}D_2\tau_{:} \tag{46}$$

### 2) PARTIAL DERIVATIVES OF PATH CONSTRAINTS

The partial derivatives of path constraints of the node can be obtained by applying vector chain differentiation. $\frac{\partial C_{:,i}}{\partial x_{:,j}^T}$ and $\frac{\partial C_{:,i}}{\partial u_{:,j}^T}$ are already partial derivatives of trajectory optimization problems and $\frac{\partial C_{:,i}}{\partial \bar{u}_{:,j}^T} = 0$.

The remaining items are:

$$\begin{cases} \dfrac{\partial C_{:,i}}{\partial t_0} = \dfrac{\partial C_{:,i}}{\partial t_{:}^T}(-\tau_{:} + 1) \\ \dfrac{\partial C_{:,i}}{\partial t_f} = \dfrac{\partial C_{:,i}}{\partial t_{:}^T}\tau_{:} \end{cases} \tag{47}$$

Similarly, the partial derivative of the path constraint at the midpoint of the node on the optimization variable $z$ can be derived.

where $\frac{\partial \bar{C}_{:,i}}{\partial \bar{u}_{:,j}^T}$ can be directly extracted from the partial derivative of the trajectory optimization problem.

$$\begin{aligned} \frac{\partial \bar{C}_{:,i}}{\partial x_{:,j}^T} &= \sum_{l=1}^{n} \frac{\partial C_{:,i}}{\partial \bar{x}_{:,l}^T}(\frac{1}{2}D_2\frac{\partial x_{:,l}}{\partial x_{:,j}^T} - \frac{\Delta t}{8}hD_1\frac{\partial f_{:,l}}{\partial x_{:,j}^T}) \\ \frac{\partial \bar{C}_{:,i}}{\partial u_{:,j}^T} &= \sum_{l=1}^{n} \frac{\partial C_{:,i}}{\partial \bar{x}_{:,l}^T}(-\frac{\Delta t}{8}hD_1\frac{\partial f_{:,l}}{\partial u_{:,j}^T}) \\ \frac{\partial \bar{C}_{:,i}}{\partial t_0} &= \sum_{l=1}^{n} \frac{\partial \bar{C}_{:,i}}{\partial \bar{x}_{:,l}^T}(\frac{1}{8}hD_1 f_{:,l} - \frac{\Delta t}{8}hD_1\frac{\partial f_{:,l}}{\partial t_{:}^T}(-\tau_{:} + 1)) \\ &\quad + \frac{\partial \bar{C}_{:,i}}{\partial \bar{t}_{:}^T}\frac{1}{2}D_2(-\tau_{:} + 1) \\ \frac{\partial \bar{C}_{:,i}}{\partial t_f} &= \sum_{l=1}^{n} \frac{\partial \bar{C}_{:,i}}{\partial \bar{x}_{:,l}^T}(-\frac{1}{8}hD_1 f_{:,l} - \frac{\Delta t}{8}hD_1\frac{\partial f_{:,l}}{\partial t_{:}^T}\tau_{:}) \\ &\quad + \frac{\partial \bar{C}_{:,i}}{\partial \bar{t}_{:}^T}\frac{1}{2}D_2\tau_{:} \end{aligned} \tag{48}$$

### 3) PARTIAL DERIVATIVES OF THE ENDPOINT CONSTRAINTS

The corresponding expression can be obtained by taking the partial derivative of the endpoint constraint on the variable $z$.

where $\frac{\partial E}{\partial t_0}$ and $\frac{\partial E}{\partial t_f}$ are already the partial derivatives of trajectory optimization problems, and $\frac{\partial E}{\partial u_{:,j}^T} = 0$, $\frac{\partial E}{\partial \bar{u}_{:,j}^T} = 0$.

The remaining items are:

$$\frac{\partial E}{\partial x_{:,j}^T} = [\frac{\partial E}{\partial x_{0,j}}, 0, \cdots, 0, \frac{\partial E}{\partial x_{N,j}}] \tag{49}$$

### 4) PARTIAL DERIVATIVE OF TIME DIFFERENCE CONSTRAINT

The time difference constraint is linear and Eq. (50) are easily to be obtained:

$$\begin{cases} \dfrac{\partial \Delta t}{\partial x_{:,j}^T} = 0 \\ \dfrac{\partial \Delta t}{\partial u_{:,j}^T} = 0 \\ \dfrac{\partial \Delta t}{\partial \bar{u}_{:,j}^T} = 0 \end{cases} \tag{50}$$

The remaining items are:

$$\begin{cases} \dfrac{\partial \Delta t}{\partial t_0} = -1 \\ \dfrac{\partial \Delta t}{\partial t_f} = 1 \end{cases} \tag{51}$$

It can be seen that the Jacobian matrix of the NLP can be decomposed into partial derivatives of the state equations, path constraints, endpoint constraints, and time constraints for trajectory optimization problems. The Jacobian matrix is assembled using the method described in this section after calculating the partial derivatives of these constraints at discrete nodes and intervals (for $f$ and $C$), as well as at endpoints (for $E$ and $\Delta t$).

For the elements that may not be zero, their specific values are related to the dependency matrix. Taking $\frac{\partial \xi_{:,i}}{\partial x_{:,j}^T}$ as an example:

$$struct(\frac{\partial \xi_{:,i}}{\partial x_{:,j}^T}) = \begin{cases} D_2 & (S_1)_{i,l} = 1 or (S_1)_{i,l}(S_1)_{l,j} \geq 1 \\ 0 & other \end{cases} \tag{52}$$

## VI. NUMERICAL SIMULATIONS AND EXPERIMENTAL VERIFICATION

### A. NUMERICAL SIMULATIONS

#### 1) DOUBLE LANE CHANGING PATH UNDER u = 30km/h

Fig. 2(a) indicates that the actual path can follow the set reference trajectory for double lane changing, with a maximum error of only 0.1 m and a smooth overall path tracking trajectory, meeting the needs of vehicles in practical use. But there are prone to minor deviations at the intersection of straight and curved roads. The reason for the analysis error is that the transition distance between the double transfer lines is too short and the vehicle speed is high. Fig. 2(b) indicates that the front wheel angle of the vehicle changes smoothly. From the graph, it can be seen that the front wheel angle has already occurred at zero time. The reason for the analysis that the straight path at the beginning of the target path is relatively short, and due to the high simulation speed setting, in order to ensure the accuracy of path tracking and the smoothness of automatic steering, the controller has already calculated the front wheel angle at the beginning. The trend of changes in Figs. 2(c)-2(d) is almost consistent, and it can be seen from the figure that both the yaw rate and the

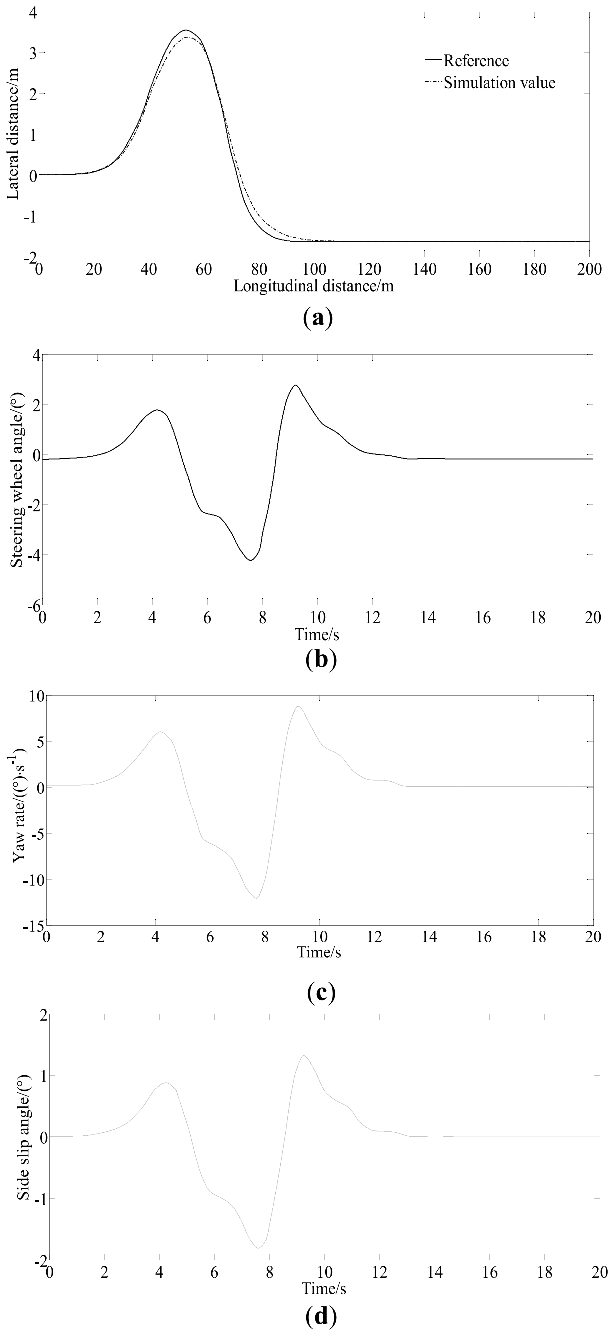


**FIGURE 2.** Simulation results under the condition of double lane changing path with u = 30km/h. (a) Lateral distance, (b) Steering wheel angle, (c) Yaw rate, (d) Side slip angle.

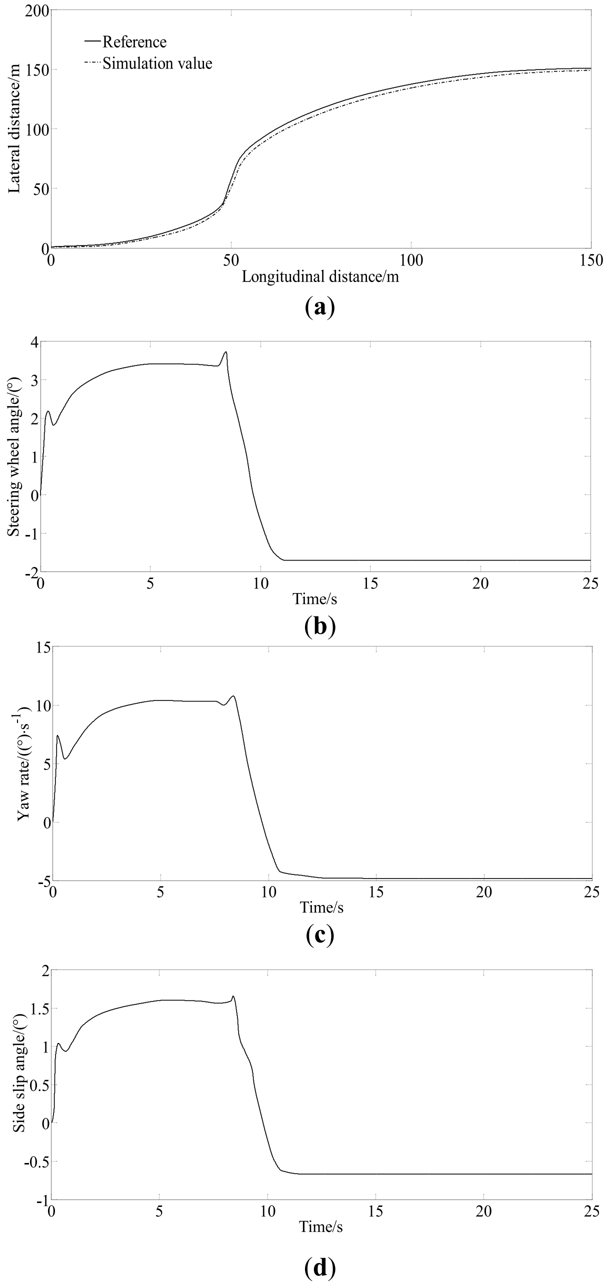


**FIGURE 3.** Simulation results under the condition of varied road curvature path with u = 30km/h. (a) Lateral distance, (b) Steering wheel angle, (c) Yaw rate, (d) Side slip angle.

sideslip angle of the vehicle have a relatively small range of overall smooth changes. It is found that the maximum value repeatedly appears at several vertices of the double lane changing path. The reason for this analysis is that the path is highly curved and the vehicle speed is too high. It can be seen that the vehicle controlled by the method can maintain a good stable state and achieve good tracking accuracy during operation at medium to high speeds. And also, both the yaw rate and the sideslip angle of the vehicle can enter a stable state. The simulation results show that the proposed control strategy effectively improves the path tracking ability of the vehicle and ensures the stability of the vehicle.

### 2) VARIED-ROAD-CURVATURE PATH UNDER u = 30km/h

Fig. 3(a) illustrates the validity of the proposed controller on different curvature paths. As the vehicle is performing path tracking under the double lane changing conditions, the fluctuation range has increased. The simulation results of the double lane changing condition show that this control method can control the vehicle to drive on double lane change roads with significant changes in road curvature, and can track the target path better. In addition, at the exit of the double lane changing reference path, the vehicle no longer drives along the reference path, resulting in instability. The reason is that the curvature of these two positions changes rapidly, so the vehicle fails to respond in time. Fig. 3(b) indicates that the front wheel angle of the vehicle changes smoothly. From the figure, it can be seen that the front steering angle has a slight jitter under changing curvature path road conditions. From Figs. 3(c)-3(d) it can be seen that the sideslip angle and yaw rate of the vehicle can be significantly reduced under the control of the proposed algorithm, ensuring the accuracy of vehicle path tracking while further improving the lateral stability of the vehicle during driving.

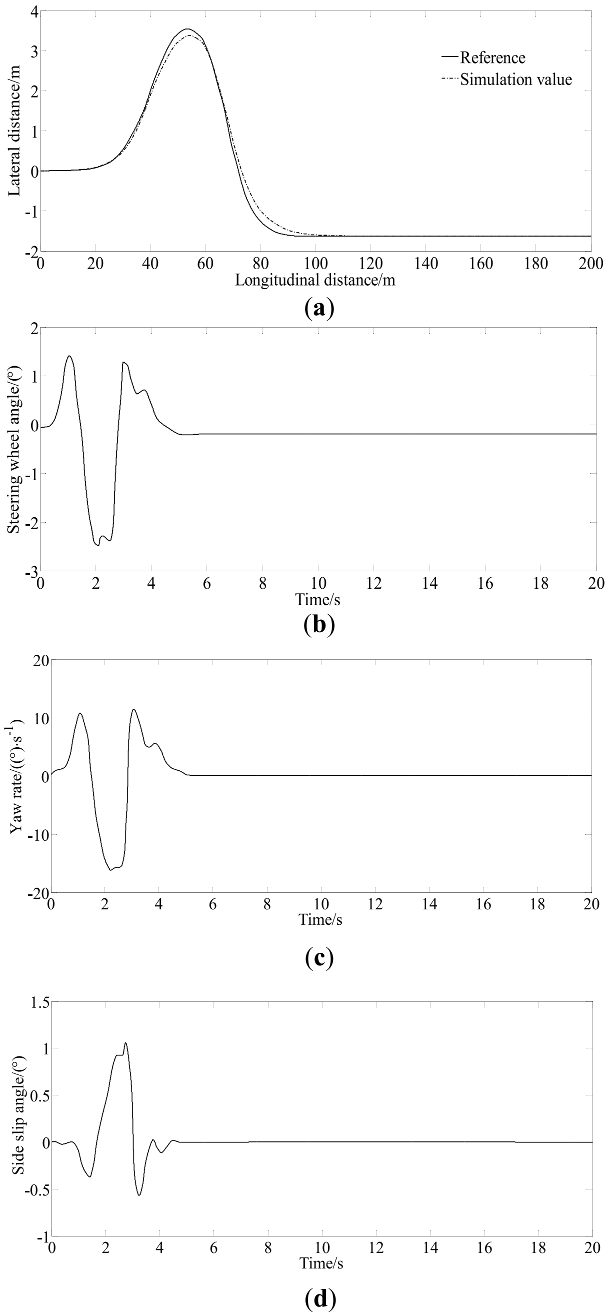
### 3) DOUBLE LANE CHANGING PATH UNDER u = 90km/h

From Fig. 4(a), it can be seen that the maximum error usually occurs in the two bends which are in the longitudinal position of 60m and 80m. From Fig. 4(b), it can be seen that the proposed controller makes the front-wheel angle change rapidly to ensure better tracking accuracy when the curvature of the reference path changes. At the same time, from Figs. 4(c)-(d), it can be seen that the yaw rate and sideslip angle change significantly at 1s to 5s, indicating the tracking accuracy decreasing a little at a high speed.

### 4) VARIED-ROAD-CURVATURE PATH UNDER u = 90km/h

Fig. 5(a) indicates that it has high tracking accuracy at a low speed, while the tracking accuracy decreases a little at a high speed. And also, from fig. 5(b) it can be seen that the steering wheel angle change rapidly to ensure better tracking performance when the curvature of the given path changes. In additional, from fig. 5(c)-(d) it can be seen that yaw rate and sideslip angle fluctuate greatly at the time 1s to 5s. This is because the curvature of the varied road curvature path fluctuates significantly. It illustrates the validity of the proposed controller on different curvature paths.

The trajectory tracking problem is transformed into an NLP problem using local collocation method, and SNOPT 7 is used to solve the NLP problem. Two calculation schemes for the first-order partial derivatives of NLP were studied. One solution is to not provide the first-order partial derivatives of NLP, in which case SNOPT uses the finite difference method internally to directly calculate the partial derivatives of NLP. Another approach is the method proposed in this article, which uses the finite difference method to calculate the first-order partial derivatives of the original trajectory optimization problem. The method described in Section III is then used to assemble the first-order partial derivative matrix of NLP and provide it to SNOPT. Using Matlab 2009a programming language, Table 2 shows the comparison of optimization efficiency between two schemes with 32 and 128 uniformly discrete nodes. The software environment for optimizing the calculation process is: Windows 10 Professional 64 bit operating system and MATLAB R2020b. The hardware environment is an Intel (R) Core (TM) i5-8400 processor with 12GB of memory.



(a)

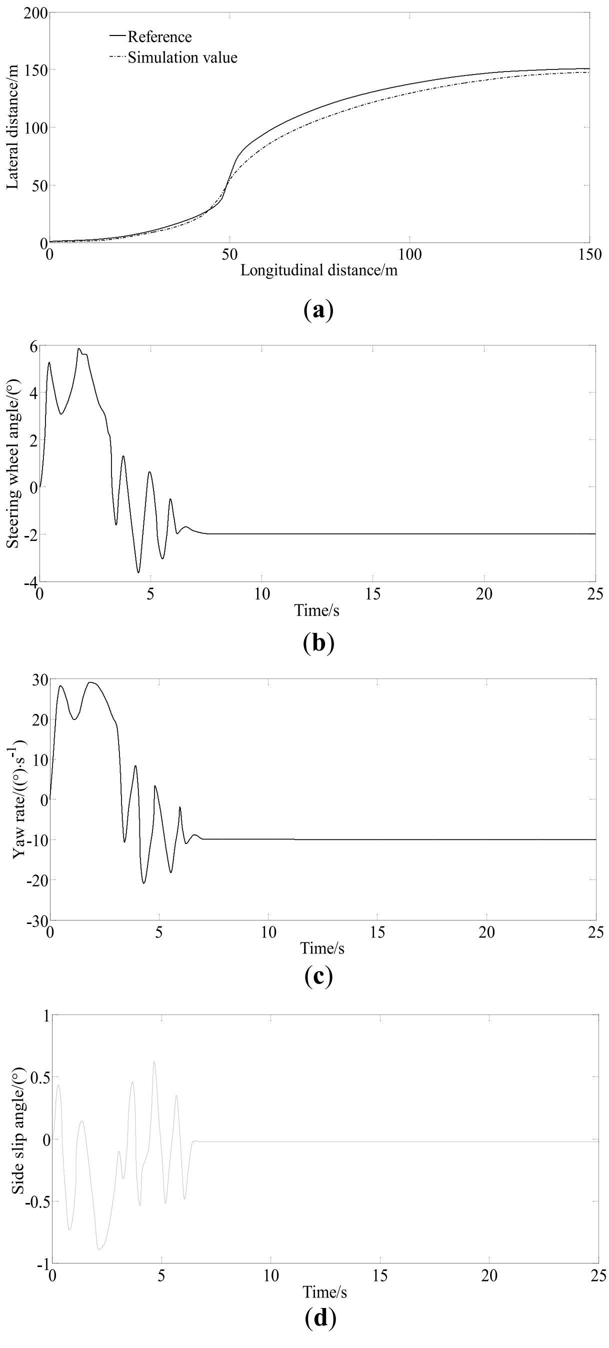
(b)

(c)

(d)

**FIGURE 4. Simulation results under the condition of double lane changing path with u = 90km/h. (a) Lateral distance, (b) Steering wheel angle, (c) Yaw rate, (d) Side slip angle.**

From Table 2, it can be seen that compared with directly calculating the partial derivatives of NLP using the finite difference method, the proposed method can reduce the computation time, and the optimization efficiency of the proposed method is significantly improved with the increase of the number of discrete nodes. This is because as the number of discrete nodes increases, the first order partial derivative matrix of NLP has greater sparsity. And also, the proposed



(a)

(b)

(c)

(d)

**FIGURE 5.** **Simulation results under the condition of varied road curvature path with u = 90km/h. (a) Lateral distance, (b) Steering wheel angle, (c) Yaw rate, (d) Side slip angle.**

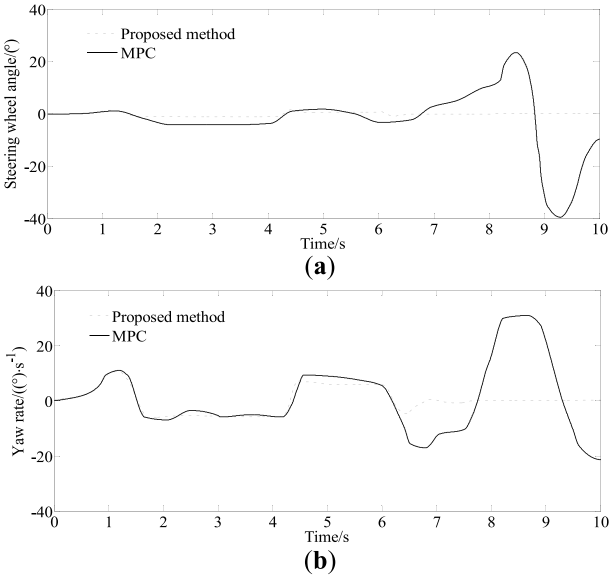
method effectively reduces the scale of NLP and reduces initial sensitivity while improving convergence speed.

## B. EFFICIENCY VERIFICATION

In order to verify the control performance of the proposed method compared with traditional method (MPC), simulation of a road with changing friction coefficient conditions is established. The comparison result is given in Fig. 6.

**TABLE 2.** **Efficiency comparison of different methods.**

| Calculation Method | Number of nodes | Sparsity degree | Calculation time(s) |
|---|---|---|---|
| Direct Difference | 32 | | 89.5 |
| | 128 | | 1270.7 |
| The proposed method | 32 | 92.6% | 3.5 |
| | 128 | 97.2& | 25.1 |



(a)

(b)

**FIGURE 6.** **Evaluation of the calculation accuracy and calculation efficiency. (a) Steering wheel angle, (b) Yaw rate.**

From Fig.6, it can be concluded that the maximum values of both the steering wheel angle and the yaw rate based on the proposed in the article are smaller than that of the MPC method. This is because that when the road friction coefficient changes, the MPC controller still works the same as before leading to vehicle state exceeding the road adhesion limit. The comparison result indicate the advantage of the proposed method for solving the problem of path tracking of vehicle.

## C. EXPERIMENTAL VERIFICATION

The slalom test is performed in accordance with ISO/TR3888-2004, wherein the vehicle traveled through the test section at a constant speed of 80 km/h (± 3 km/h), and the time history curve of each measured variable is recorded to obtain test data such as the yaw rate and steering wheel angle. The real experiment vehicle is shown in Fig. 7. The measurement equipment are shown in Fig. 8. The steering torque/angle tester is used to measure the steering torque or the steering angle. The angular rate gyroscope is used to measure the yaw rate. The AM-2800 vehicle comprehensive performance test system is used as the corresponding data collector. The slalom test road is shown in Fig. 9. A block diagram of the test system is shown in Fig. 10.

**FIGURE 7.** Real test vehicle.

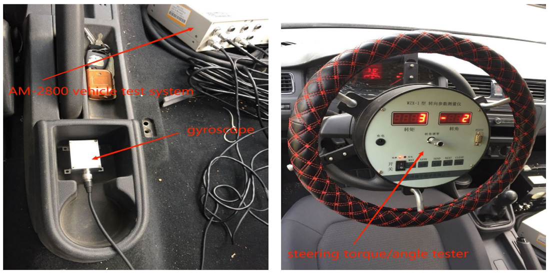


**FIGURE 8.** Measurement equipments.

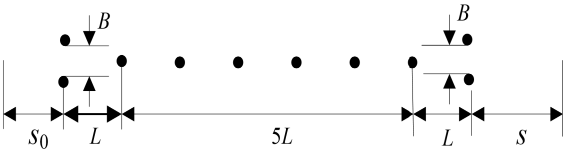


**FIGURE 9.** Pylon course slalom test road(● stands for stake).

where $s_0 = L = 2u; s = 3u$, $u$ is the longitudinal speed. $B$ is the rotary distance, $B = 2.46$m. $L$ is the distance between the stakes.

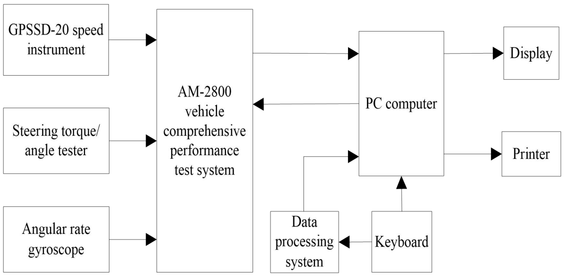

GPSSD-20 speed instrument
Steering torque/ angle tester
Angular rate gyroscope
AM-2800 vehicle comprehensive performance test system
PC computer
Display
Printer
Data processing system
Keyboard

**FIGURE 10.** Block diagram of the test system.

The comparison of simulation and test value is shown in Fig. 11. From the figure it can be seen that there are bigger steering wheel angles at 130m, 180m, 240m, 280m. This will reduce the comfort of passengers and also reduce the safety of the vehicle. After the lateral stability of the vehicle deteriorates, the stability of the vehicle is adjusted by significantly changing the front wheel angle. However, this will further cause severe fluctuations in the lateral stability performance of the vehicle, which will also cause the vehicle to deviate from its trajectory. However, the proposed method can keep the yaw rate of the vehicle within a stable boundary to ensure the safety of the vehicle. At the same time, the changing trends of the curve of the simulation values of both the steering wheel angle and the yaw rate are consistent with the test values indicating the effectiveness of the proposed method.

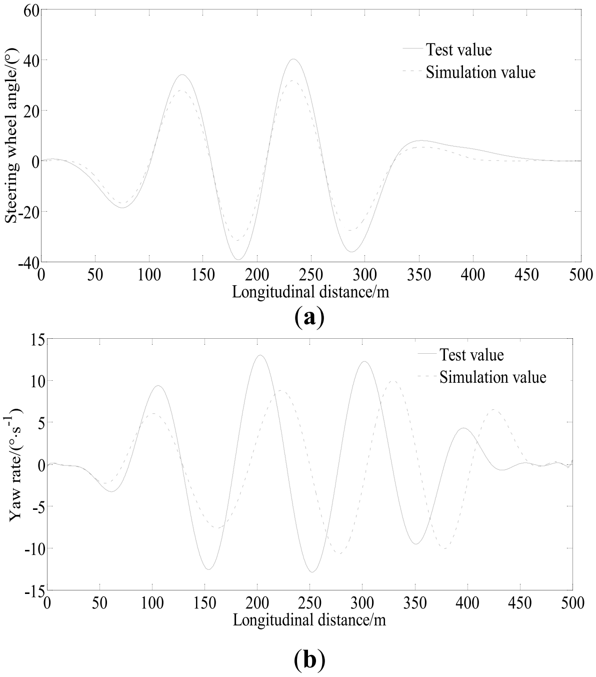


**FIGURE 11.** Comparison of simulation and test value. (a) Steering wheel angle, (b) Yaw rate.

## VII. CONCLUSION

In this paper, the vehicle optimum path tracking problem is analyzed. Firstly, a new variable notation is introduced to write the NLP problem in vector form. And then the vector chain derivative rule is applied to convert the partial derivative of NLP into the partial derivative of the original trajectory optimization problem. Due to the significant reduction in the number of constraints and independent variables in trajectory optimization problems compared to NLP, this approach can significantly reduce the computational complexity of the first order partial derivative of NLP. Simulation results under different conditions including the double lane changing path and the varied-road-curvature path at different speeds have verified the effectiveness of the proposed method. And also, the proposed method can be easily extended to other discrete formats of the local collocation method. The research work in this article is based on the HS format. And also, poor initial value guessing can lead to slow convergence in the solution process. In the future, the research direction is to extend the proposed method to other discrete formats of local collocation method, such as trapezoidal format, classical fourth-order RK format, etc. And it is necessary to study methods for quickly obtaining near optimal initial solutions.

## REFERENCES

[1] Y. Liu, D. Cui, and W. Peng, "Optimum control for path tracking problem of vehicle handling inverse dynamics," *Sensors*, vol. 23, no. 15, p. 6673, Jul. 2023.

[2] Y. Liu and D. Cui, "Estimation algorithm for vehicle state estimation using ant lion optimization algorithm," *Adv. Mech. Eng.*, vol. 14, no. 3, pp. 1–11, Mar. 2022.

[3] Y. Liu, D. Cui, and W. Peng, "Optimal lane changing problem of vehicle handling inverse dynamics based on mesh refinement method," *IEEE Access*, vol. 11, pp. 115617–115626, 2023.

[4] X. Gong, L. Liu, Z. Peng, D. Wang, and W. Zhang, "Resource-aware synchronized path following of multiple unmanned surface vehicles with experiments: A cooperative vector field approach," *Control Eng. Pract.*, vol. 124, Jul. 2022, Art. no. 105184.

[5] X. Jin and M. J. Er, "Cooperative path planning with priority target assignment and collision avoidance guidance for rescue unmanned surface vehicles in a complex ocean environment," *Adv. Eng. Informat.*, vol. 52, Apr. 2022, Art. no. 101517.

[6] S. Rong, H. Wang, H. Li, W. Sun, Q. Gu, and J. Lei, "Performance-guaranteed fractional-order sliding mode control for underactuated autonomous underwater vehicle trajectory tracking with a disturbance observer," *Ocean Eng.*, vol. 263, Nov. 2022, Art. no. 112330.

[7] Z. Yan, J. Zhang, J. Zeng, and J. Tang, "Three-dimensional path planning for autonomous underwater vehicles based on a whale optimization algorithm," *Ocean Eng.*, vol. 250, Apr. 2022, Art. no. 111070.

[8] A. Gonzalez-Garcia, I. Collado-Gonzalez, R. Cuan-Urquizo, C. Sotelo, D. Sotelo, and H. Castañeda, "Path-following and LiDAR-based obstacle avoidance via NMPC for an autonomous surface vehicle," *Ocean Eng.*, vol. 266, Dec. 2022, Art. no. 112900.

[9] P. Li, X. Pei, Z. Chen, X. Zhou, and J. Xu, "Human-like motion planning of autonomous vehicle based on probabilistic trajectory prediction," *Appl. Soft Comput.*, vol. 118, Mar. 2022, Art. no. 108499.

[10] M. Wang, J. Chen, H. Yang, X. Wu, and L. Ye, "Path tracking method based on model predictive control and genetic algorithm for autonomous vehicle," *Math. Problems Eng.*, vol. 2022, pp. 1–16, Sep. 2022.

[11] G. Bejarano, J. M. Manzano, J. R. Salvador, and D. Limon, "Nonlinear model predictive control-based guidance law for path following of unmanned surface vehicles," *Ocean Eng.*, vol. 258, Aug. 2022, Art. no. 111764.

[12] X. Chen, S. H. Li, and L. Li, "Dynamics states based on adaptive-square-root-cubature-Kalman-filter and similarity-principle," *Mech. Syst. Signal Process.*, vol. 176, pp. 109162–109183, Jun. 2022.

[13] B. Wang, Y. Zhang, and W. Zhang, "Integrated path planning and trajectory tracking control for quadrotor UAVs with obstacle avoidance in the presence of environmental and systematic uncertainties: Theory and experiment," *Aerosp. Sci. Technol.*, vol. 120, Jan. 2022, Art. no. 107277.

[14] T. Qie, W. Wang, C. Yang, Y. Li, W. Liu, and C. Xiang, "A path planning algorithm for autonomous flying vehicles in cross-country environments with a novel TF-RRT method," *Green Energy Intell. Transp.*, vol. 1, no. 3, Sep. 2022, Art. no. 100026.

[15] N. A. I. Ruslan, N. H. Amer, K. Hudha, Z. A. Kadir, S. A. F. M. Ishak, and S. M. F. S. Dardin, "Modelling and control strategies in path tracking control for autonomous tracked vehicles: A review of state of the art and challenges," *J. Terramechanics*, vol. 105, pp. 67–79, Feb. 2023.

[16] X. Zhou, Z. Wang, H. Shen, and J. Wang, "A Biquadratic-Lyapunov-function-based adaptive control methodology with application to automated ground vehicle path tracking," in *Proc. Amer. Control Conf. (ACC)*, Jun. 2022, pp. 1703–1708.

[17] M. Rokonuzzaman, N. Mohajer, and S. Nahavandi, "Effective adoption of vehicle models for autonomous vehicle path tracking: A switched MPC approach," *Vehicle Syst. Dyn.*, vol. 61, no. 5, pp. 1236–1259, May 2023.

[18] Z. Hu, J. Huang, Z. Yang, and Z. Zhong, "Cooperative-game-theoretic optimal robust path tracking control for autonomous vehicles," *J. Vibrat. Control*, vol. 28, nos. 5–6, pp. 520–535, Mar. 2022.

[19] M. H. Korayem and A. Nikoobin, "Maximum payload for flexible joint manipulators in point-to-point task using optimal control approach," *Int. J. Adv. Manuf. Technol.*, vol. 38, nos. 9–10, pp. 1045–1060, Sep. 2008.

[20] M. H. Korayem and A. Nikoobin, "Maximum payload path planning for redundant manipulator using indirect solution of optimal control problem," *Int. J. Adv. Manuf. Technol.*, vol. 44, nos. 7–8, pp. 725–736, Oct. 2009.

[21] M. H. Korayem, M. Haghpanahi, and H. N. Rahimi, *Finite Element Method and Optimal Control Theory for Path Planning of Elastic Manipulators*, vol. 199. Cham, Switzerland: Springer, 2009, pp. 117–126.

[22] Y. Liu and J. Jiang, "Optimum path-tracking control for inverse problem of vehicle handling dynamics," *J. Mech. Sci. Technol.*, vol. 30, no. 8, pp. 3433–3440, Aug. 2016.

[23] J. T. Betts and W. P. Huffman, "Exploiting sparsity in the direct transcription method for optimal control," *Computational Optimization Appl.*, vol. 14, no. 2, pp. 179–201, 1999.

[24] M. A. Patterson and A. Rao, "Exploiting sparsity in direct collocation pseudospectral methods for solving optimal control problems," *J. Spacecraft Rockets*, vol. 49, no. 2, pp. 354–377, Mar. 2012.

**YINGJIE LIU** received the Ph.D. degree from the College of Energy and Power Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2014. He is currently with the School of Machinery and Automation, Weifang University, Weifang, China. His current research interests include vehicle system dynamics and control theory to ground vehicles.

**ENHAO WANG** is currently pursuing the Ph.D. degree in communications and THz node with the Department of Engineering, Durham University, U.K. His current research interests include integrated sensing and communication and vehicle to everything.