



OPEN

An accurate trajectory tracking method for low-speed unmanned vehicles based on model predictive control

Lifen Wang, Sizhong Chen & Hongbin Ren

Trajectory tracking on a low-speed vehicle using the model predictive control (MPC) algorithm usually assumes a simple road terrain. This assumption does not correspond to the actual road situation, leading to low tracking accuracy. Therefore, a trajectory tracking method considering road curvature based on MPC is proposed in this paper. In this method, the controller can automatically switch between MPC types. Linear model predictive control (LMPC) is selected for small road curvatures, while nonlinear model predictive control (NMPC) is employed for large road curvatures. In addition, the NMPC algorithm in this work considers the effect of road curvature on tracking accuracy, making it suitable for tracking time-varying curvature roads. To verify the feasibility of the algorithm, simulation comparisons with the basic MPC model were carried out at different testing roads and vehicle longitudinal speeds. The results indicate that the method significantly improves trajectory tracking accuracy, all while ensuring real-time calculations. The intelligent switching capability of control models based on road curvature allows its application to track trajectories on arbitrarily complex roads.

With the continuous evolution of people's demands for connected cars that can enhance driving safety and avoid traffic congestion, these vehicles have emerged as a pivotal focus in automotive advancement. As part of the research on intelligent vehicles, trajectory tracking can assist intelligent vehicles in driving stably and accurately along the planned trajectory route^{1,2}. A broad spectrum of research has delved into trajectory tracking, spanning optimal control methods³, sliding mode control methods⁴⁻⁶, adaptive control methods⁷, robust control⁸, and fuzzy control methods⁹.

Analyzing the methods outlined in the previous paragraph allows us to discern their respective strengths and weaknesses. Optimal control, for instance addresses the original problem by breaking it down into multiple subproblems to attain a global optimal solution. However, its efficiency diminishes notably when confronted with high-dimensional or expansive state spaces. Sliding mode control boasts rapid responsiveness but entails a complex controller design. The calculations of adaptive control and robust control are simple. However, their implementation processes are intricate, and they exhibit poor real-time capabilities. Fuzzy control exhibits robustness when addressing nonlinear issues. Yet, the design of fuzzy control is not systematic enough to define the control target.

Compared to these algorithms, model predictive control (MPC) can achieve rolling optimization by predicting future states based on past states^{10,11}. Additionally, it can handle multi-constraint problems by incorporating constraints into the objective function. Furthermore, MPC includes a feedback correction function¹², which endows it with its robustness and anti-interference capabilities^{13,14}, making MPC widely used in trajectory tracking. In the current research on low-speed vehicle trajectory tracking using the MPC algorithm^{15,16}, a simple road terrain is usually assumed, neglecting the road curvature factor. However, due to complex geometric conditions like bending and fluctuation, limited sight lines, and various uncertain factors, the traffic accident rate in curved roads is much higher than that on ordinary roads¹⁷. Consequently, it is imperative to investigate trajectory tracking accuracy under complex road conditions using the MPC algorithm.

Classical MPC methods include the linear model predictive control (LMPC) and nonlinear model predictive control (NMPC) algorithms. LMPC has the advantage of simplicity in calculation and good real-time performance. However, its tracking accuracy is poor, limiting its use to predicting and controlling vehicle movement on small curvature roads or straight lines. In contrast, when unmanned vehicles operate on roads with significant

Beijing Institute of Technology, Beijing 100081, People's Republic of China. email: renhongbin@bit.edu.cn

curvature, NMPC¹⁸ is considered. Abbas has studied the feasibility of using classical NMPC for unmanned vehicle steering, yielding positive results. Nevertheless, the mathematical modeling of NMPC is very complex, and its substantial computing resource requirements can impact real-time performance¹⁹. Rafaila²⁰ has developed a method combining LMPC and NMPC to control vehicle motion, but it lacks the capability of intelligent model selection, making it challenging to apply in the actual driving process.

To address the aforementioned issues, this research examines how road curvature and vehicle speed affect the accuracy of trajectory tracking when LMPC and NMPC are used at low vehicle speeds. The classical LMPC and NMPC methods are then selected to create a new vehicle trajectory tracking method based on road curvature. The new method significantly improves trajectory tracking accuracy, all while ensuring real-time calculations. In addition, the intelligent switching capability of control models based on road curvature allows its application to track trajectories on arbitrarily complex roads in the actual driving process.

The remainder of the article is structured as follows: Section "Trajectory tracking method based on MPC" introduces the trajectory tracking method based on MPC, Section "Simulations and analysis" presents the simulation comparisons between the proposed method and the basic MPC model at different testing roads and vehicle longitudinal speeds, and Section "Real-time performance" gives the results of this paper.

Trajectory tracking method based on MPC Preliminaries of MPC

Many studies have been conducted on trajectory tracking based on MPC^{15,16}. Moreover, some scholars have compared the tracking performance of the MPC algorithm with other algorithms. For instance, Kai Yang²¹ compared MPC with the Robust H-infinity State Feedback Control in Trajectory Tracking, demonstrating that the MPC shows better tracking accuracy and response time. Duoyang Qiu²² designed a controller based on MPC to track parking trajectories, and the results indicate that the designed controller achieves better tracking accuracy compared to traditional PID controllers. In this paper, a trajectory tracking method that considers road curvature based on MPC is proposed.

As illustrated in Fig. 1, MPC is implemented by iterative online optimization across a moving finite prediction horizon. To derive the future system dynamics at the k -th time step, input parameters including state and control variables are defined as the initial parameters. A finite-time and constrained optimal control problem is calculated online in an open-loop condition over the prediction horizon N_p . While a control sequence is optimized over the prediction horizon N_p , only the first component u_k is executed to obtain the system output E_{k+1} , ensuring that the output closely approaches the reference value E_r at $k + 1$.

The optimal process is repeated in the subsequent steps based on the new input until the terminal requirements are satisfied.

Vehicle kinematic model

Figure 2 depicts a typical two-degree-of-freedom (2-DOF) vehicle steering kinematic model, with (x_f, y_f) denoting the front axle center position while (x, y) denotes its rear position, δ represents the front wheel angle, φ represents the yaw angle for the vehicle, v denotes the rear axle center speed, and ℓ represents the vehicle wheelbase.

At the driving center of the rear axle, the speed v can be expressed as:

$$v = \dot{x} \cos \varphi + \dot{y} \sin \varphi \tag{1}$$

The kinematic constraints of the front and rear axles are given by:

$$\dot{x}_f \sin(\varphi + \delta) - \dot{y}_f \cos(\varphi + \delta) = 0 \tag{2}$$

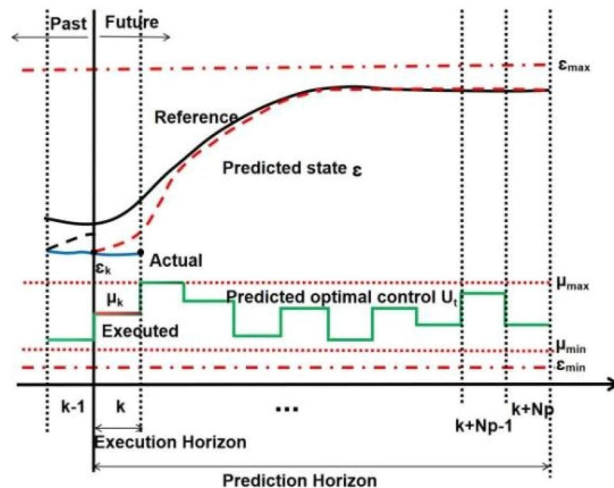


Figure 1. Basic principle of MPC.

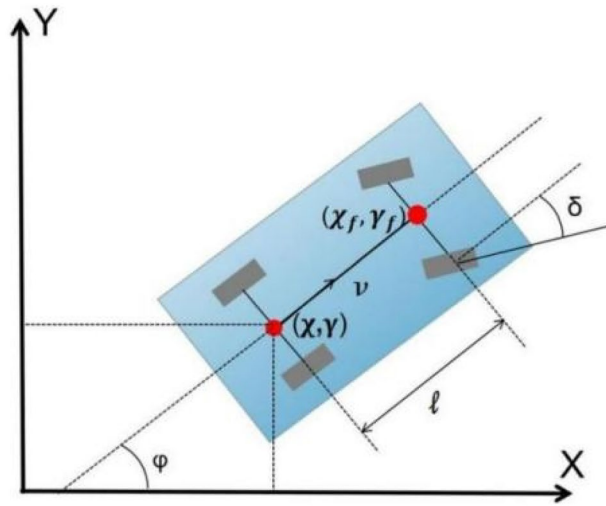


Figure 2. Vehicle kinematics diagram.

$$\dot{x} \sin \varphi + \dot{y} \cos \varphi = 0 \tag{3}$$

According to Eqs. (1–3), we can obtain:

$$\dot{x} = v \cos \varphi \tag{4}$$

$$\dot{y} = v \sin \varphi \tag{5}$$

The following is derived from the geometric relationship between the front and rear wheels:

$$x_f = x + l \cos \varphi \tag{6}$$

$$y_f = y + l \sin \varphi \tag{7}$$

Substituting Eqs. (4–7) into Eqs. (2, 3), we can solve for the yaw velocity as:

$$\omega = \frac{v}{l} \tan \varphi_f \tag{8}$$

Another way to express the yaw velocity:

$$\omega = \dot{\varphi} \tag{9}$$

Thus, the kinematics model for Fig. 2 can be expressed as:

$$\begin{cases} \dot{x} = v_x = v \cos \varphi \\ \dot{y} = v_y = v \sin \varphi \\ \dot{\varphi} = \frac{v \tan \delta}{l} \end{cases} \tag{10}$$

Model predictive control

Linear model predictive control

Because of its simplicity in calculation and good real-time performance, LMPC has been used in trajectory tracking over the past few years^{23,24}. This section outlines the process of linear discretization for the vehicle kinematics model to construct the LMPC problem²⁵.

In the trajectory tracking control process, the control and state quantities can be expressed as Eq. (11) based on Eq. (10):

$$\dot{\varepsilon} = f(\varepsilon, \mu) \tag{11}$$

where $\varepsilon = [x, y, \varphi]^T$ and $\mu = [v, \delta]^T$. Using Taylor expansion at the reference point and keeping only the first order terms, the model can be expressed as:

$$\dot{\varepsilon} = f(\varepsilon_r, \mu_r) + \frac{\partial f}{\partial \varepsilon} \Big|_{\varepsilon = \varepsilon_r, \mu = \mu_r} (\varepsilon - \varepsilon_r) + \frac{\partial f}{\partial \mu} \Big|_{\varepsilon = \varepsilon_r, \mu = \mu_r} (\mu - \mu_r) \tag{12}$$

where $\frac{\partial f}{\partial \varepsilon} \Big|_{\mu = \mu_r} \varepsilon = \varepsilon_r = \begin{bmatrix} \frac{\partial f_1}{\partial \chi_r} & \frac{\partial f_1}{\partial \gamma_r} & \frac{\partial f_1}{\partial \varphi_r} \\ \frac{\partial f_2}{\partial \chi_r} & \frac{\partial f_2}{\partial \gamma_r} & \frac{\partial f_2}{\partial \varphi_r} \\ \frac{\partial f_3}{\partial \chi_r} & \frac{\partial f_3}{\partial \gamma_r} & \frac{\partial f_3}{\partial \varphi_r} \end{bmatrix} = \begin{bmatrix} 0 & 0 & -v_r \sin \varphi_r \\ 0 & 0 & v_r \cos \varphi_r \\ 0 & 0 & 0 \end{bmatrix}$, $\frac{\partial f}{\partial \mu} \Big|_{\mu = \mu_r} \mu = \mu_r = \begin{bmatrix} \frac{\partial f_1}{\partial v_r} & \frac{\partial f_1}{\partial \delta_r} \\ \frac{\partial f_2}{\partial v_r} & \frac{\partial f_2}{\partial \delta_r} \\ \frac{\partial f_3}{\partial v_r} & \frac{\partial f_3}{\partial \delta_r} \end{bmatrix}$

$$= \begin{bmatrix} \cos \varphi_r & 0 \\ \sin \varphi_r & 0 \\ \frac{\tan \varphi_r}{l} & \frac{v_r}{l \cos^2 \delta_r} \end{bmatrix}$$
, and ε_r, μ_r are the referenced state quantity and control quantity, respectively.

The change in the state quantity error can be expressed as:

$$\begin{aligned} \dot{\tilde{\varepsilon}} &= \begin{bmatrix} \dot{\chi} - \dot{\chi}_r \\ \dot{\gamma} - \dot{\gamma}_r \\ \dot{\varphi} - \dot{\varphi}_r \end{bmatrix} = \frac{\partial f}{\partial \varepsilon} \Big|_{(\varepsilon - \varepsilon_r)} + \frac{\partial f}{\partial \mu} \Big|_{(\mu - \mu_r)} \\ &= \begin{bmatrix} 0 & 0 & -v_r \sin \varphi_r \\ 0 & 0 & -v_r \cos \varphi_r \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \chi - \chi_r \\ \gamma - \gamma_r \\ \varphi - \varphi_r \end{bmatrix} + \begin{bmatrix} \cos \varphi_r & 0 \\ \sin \varphi_r & 0 \\ \frac{\tan \varphi_r}{l} & \frac{v_r}{l \cos^2 \delta_r} \end{bmatrix} \begin{bmatrix} v - v_r \\ \delta - \delta_r \end{bmatrix} = A\tilde{\varepsilon} + B\tilde{\mu} \end{aligned} \tag{13}$$

where $A = \begin{bmatrix} 0 & 0 & -v_r \sin \varphi_r \\ 0 & 0 & -v_r \cos \varphi_r \\ 0 & 0 & 0 \end{bmatrix}$, and $B = \begin{bmatrix} \cos \varphi_r & 0 \\ \sin \varphi_r & 0 \\ \frac{\tan \varphi_r}{l} & \frac{v_r}{l \cos^2 \delta_r} \end{bmatrix}$.

The Euler method is consistently used here for model discretization. Given the computational intensity of backward Euler²⁶, we opt for forward Euler to formulate the MPC problem. Upon applying forward Euler discretization to Eq. (13):

$$\dot{\tilde{\varepsilon}} = \frac{\tilde{\varepsilon}(k+1) - \tilde{\varepsilon}(k)}{T} = A\tilde{\varepsilon} + B\tilde{\mu} \tag{14}$$

The transformation of Eq. (14) is:

$$\tilde{\varepsilon}(k+1) = (IA + E)\tilde{\varepsilon}(k) + TB\tilde{\mu}(k) = \tilde{A}\tilde{\varepsilon}(k) + \tilde{B}\tilde{\mu}(k) \tag{15}$$

where I is the identity matrix, $\tilde{A} = \begin{bmatrix} 1 & 0 & -v_r \sin \varphi_r \\ 0 & 1 & v_r \cos \varphi_r \\ 0 & 0 & 1 \end{bmatrix}$, and $\tilde{B} = \begin{bmatrix} T \cos \varphi_r & 0 \\ T \sin \varphi_r & 0 \\ T \frac{\tan \varphi_r}{l} & T \frac{v_r}{l \cos^2 \delta_r} \end{bmatrix}$.

By assigning \tilde{A} to a and \tilde{B} to b , Eq. (15) can be expressed as:

$$\tilde{\varepsilon}(k+1) = a\tilde{\varepsilon}(k) + b\tilde{\mu}(k) \tag{16}$$

where $\tilde{\varepsilon} = \begin{bmatrix} \chi - \chi_r \\ \gamma - \gamma_r \\ \varphi - \varphi_r \end{bmatrix}$ and $\tilde{\mu} = \begin{bmatrix} v - v_r \\ \delta - \delta_r \end{bmatrix}$.

The output equation can be defined as:

$$\gamma(k) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tilde{\varepsilon}(k) = C\tilde{\varepsilon}(k) \tag{17}$$

where $C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$.

The cost function should be able to ensure that the unmanned vehicle can track the desired trajectory quickly and smoothly. Therefore, it is necessary to add the state quantity deviation and the control quantity into the cost function. When designing the trajectory tracking controller, the following cost function is used:

$$J = \sum_{j=1}^N \tilde{\varepsilon}^T(k+j)Q\tilde{\varepsilon}(k+j) + \tilde{\mu}^T(k+j-1)R\tilde{\mu}(k+j-1) \tag{18}$$

where Q and R are weigh matrices.

This objective function cannot limit the control increment in each sampling period, potentially causing sudden changes in control quantity and leading to its discontinuity. To overcome this limitation, a new state quantity is built:

$$\xi(k) = \begin{bmatrix} \tilde{\varepsilon}(k) \\ \tilde{\mu}(k-1) \end{bmatrix} \tag{19}$$

The new state space is expressed as:

$$\begin{aligned} \xi(k+1) &= \begin{bmatrix} \tilde{\varepsilon}(k+1) \\ \tilde{\mu}(k) \end{bmatrix} = \begin{bmatrix} a\tilde{\varepsilon}(k) + b\tilde{\mu}(k) \\ \tilde{\mu}(k-1) + \tilde{\mu}(k) - \tilde{\mu}(k-1) \end{bmatrix} = \begin{bmatrix} a\tilde{\varepsilon}(k) + b\tilde{\mu}(k-1) + b\tilde{\mu}(k) - b\tilde{\mu}(k-1) \\ \tilde{\mu}(k-1) + \tilde{\mu}(k) - \tilde{\mu}(k-1) \end{bmatrix} \\ &= \begin{bmatrix} [a \ b] \\ [o \ I_{Nu}] \end{bmatrix} \begin{bmatrix} \tilde{\varepsilon}(k) \\ \tilde{\mu}(k-1) \\ \tilde{\varepsilon}(k) \\ \tilde{\mu}(k-1) \end{bmatrix} + \begin{bmatrix} b \\ I_{Nu} \end{bmatrix} (\tilde{\mu}(k) - \tilde{\mu}(k-1)) = \begin{bmatrix} a & b \\ o & I_{Nu} \end{bmatrix} \xi(k) + \begin{bmatrix} b \\ I_{Nu} \end{bmatrix} \Delta\tilde{\mu}(k) \\ &= \hat{A}\xi(k) + \hat{B}\Delta\tilde{\mu}(k) \end{aligned} \tag{20}$$

where $\hat{A} = \begin{bmatrix} a & b \\ o & I_{Nu} \end{bmatrix}$ and $\hat{B} = \begin{bmatrix} b \\ I_{Nu} \end{bmatrix}$.
The new output equation can be defined as:

$$\eta(k) = [I_{Nx} \quad o] \begin{bmatrix} \tilde{\varepsilon}(k) \\ \tilde{\mu}(k-1) \end{bmatrix} = C\xi(k) \tag{21}$$

where I_{Nx} is an identity matrix with Nx dimension. From Eq. (20) and Eq. (21), the output quantities can be calculated as follows:

$$Y = \psi\xi(k) + \Theta\Delta\mu \tag{22}$$

$$\begin{aligned} \text{where } Y &= \begin{bmatrix} \eta(k+1) \\ \eta(k+2) \\ \eta(k+3) \\ \vdots \\ \eta(k+Nc) \\ \eta(k+Np) \end{bmatrix}, \psi = \begin{bmatrix} CA \\ CA^2 \\ CA^3 \\ \vdots \\ CA^{Nc} \\ CA^{Np} \end{bmatrix}, \xi(k) = \begin{bmatrix} \tilde{\varepsilon}(k) \\ \tilde{\mu}(k-1) \end{bmatrix}, \Theta = \begin{bmatrix} CB & 0 & 0 & \dots & 0 \\ CAB & CB & 0 & \dots & 0 \\ \dots & \dots & \dots & \ddots & 0 \\ CA^{Nc-1}B & CA^{Nc-2}B & CA^{Nc-3}B & \dots & CA^0B \\ \dots & \dots & \dots & \dots & \dots \\ CA^{Np-1}B & CA^{Np-2}B & CA^{Np-3}B & \dots & CA^{Np-Nc}B \end{bmatrix} \text{ and} \\ \Delta\mu &= \begin{bmatrix} \Delta\tilde{\mu}(k) \\ \Delta\tilde{\mu}(k+1) \\ \Delta\tilde{\mu}(k+2) \\ \vdots \\ \Delta\tilde{\mu}(k+Nc-1) \end{bmatrix} \end{aligned}$$

The optimization objective function can be transformed to:

$$J = \sum_{i=1}^{Np} \|\eta(k+i) - \eta_r(k+i)\|_Q^2 + \sum_{i=1}^{Nc-1} \|\Delta\mu(k+i)\|_R^2 + \rho\lambda^2 \tag{23}$$

Here λ is the relaxation factor.
Define the system output reference values as:

$$Y_r = [\eta_r(k+1) \ \eta_r(k+2) \ \dots \ \eta_r(k+Nc) \ \dots \ \eta_r(k+Np)]^T = [0 \ 0 \ \dots \ 0 \ \dots \ 0]^T \tag{24}$$

Let $E = \psi\xi(k)$, $Q = I_{Np} \otimes Q$, and $R = I_{Np} \otimes R$, then:

$$J = (E + \Theta\Delta\mu)^T Q (E + \Theta\Delta\mu) + \Delta\mu^T R \Delta\mu + \rho\lambda^2 \tag{25}$$

Simplify Eq. (25) as:

$$J = E^T Q E + \Delta\mu^T (\Theta^T Q \Theta + R) \Delta\mu + 2E^T Q \Theta \Delta\mu + \rho\lambda^2 \tag{26}$$

Since $E^T Q E$ is independent of $\Delta\mu$, it can be neglected.
Then the objective function can be transformed into a quadratic form in quadprog:

$$J = [\Delta\mu^T \ \lambda] \begin{bmatrix} \Theta^T Q \Theta + R & 0 \\ 0 & \rho \end{bmatrix} [\Delta\mu^T \ \lambda]^T + [2E^T Q \Theta \ 0] [\Delta\mu^T \ \lambda]^T \tag{27}$$

According to Ye²⁷, under low-speed working conditions, the disparity in tracking accuracy between the kinematic and dynamic models is small. In this paper, we use the kinematic model, which doesn't account for tire slip angle, to construct the MPC controller. Thus, the sideslip angle is not constrained here. The following constraints for control quantity and control increment are introduced to the system to meet the actual operation requirements:

$$\tilde{U}_{\min} \leq \tilde{U} \leq \tilde{U}_{\max} \tag{28}$$

$$\Delta\tilde{\mu}_{\min} \leq \Delta\tilde{\mu} \leq \Delta\tilde{\mu}_{\max} \tag{29}$$

where $\tilde{U} = \begin{bmatrix} \tilde{\mu}(k) \\ \tilde{\mu}(k+1) \\ \dots \\ \tilde{\mu}(k+N_c-1) \end{bmatrix}$, $\Delta\tilde{\mu} = \begin{bmatrix} \Delta\tilde{\mu}(k) \\ \Delta\tilde{\mu}(k+1) \\ \dots \\ \Delta\tilde{\mu}(k+N_c-1) \end{bmatrix}$.
 In this paper, $\tilde{U}_{\min} = [-0.2 \ -0.436]$, $\tilde{U}_{\max} = [0.2 \ 0.436]$, $\Delta\tilde{\mu}_{\min} = [-0.05 \ -0.0082]$, $\Delta\tilde{\mu}_{\max} = [0.05 \ 0.0082]$.
 The parameters in the above formulas are described in Table 1.

Nonlinear model predictive control

Due to its high tracking accuracy, NMPC has been applied in various fields for trajectory tracking²⁸. However, this NMPC is based on the vehicle kinematics model, where the impact of curvature is not considered. This can lead to a reduction in tracking accuracy when the curvature is large. To solve this problem, this paper designs a NMPC model based on tracking errors, which can impose constraints on heading angle deviation and distance deviation. At the same time, it can consider the influence of road curvature on the cost function, which is conducive to improving the tracking effect on roads with large curvature. The construction process of NMPC based on tracking errors is outlined in the sequel.

Figure 3 shows the diagram of tracking error model, and P_1 is the projection of the vehicle's rear axle center M on the road center line. It is assumed that the instantaneous turning radius of the vehicle is the same as the curvature radius of the road. Therefore, the reference curvature at P_1 can be denoted as:

$$\kappa_{\text{ref}} = 1/R \tag{30}$$

Parameters	Description	Parameters	Description
ε	State quantity	$\Delta\tilde{\mu}_{\min}$	The allowed minimum $\Delta\tilde{\mu}$
u	Control quantity	\tilde{U}	The control quantity error matrix in the control domain
ε_r	Referenced state quantity	\tilde{U}_{\max}	The allowed maximum \tilde{U}
u_r	Referenced control quantity	\tilde{U}_{\min}	The allowed minimal \tilde{U}
A	Partial derivative of f with respect to x	λ	The relaxation coefficient
B	Partial derivative of f with respect to u	$\xi(k)$	New state quantity
\tilde{A}	Discrete A	$\eta(k)$	Output quantity in the predict domain
\tilde{B}	Discrete B	C	Selection matrix
T	Sample time interval	Y	Output quantities matrix
I	Unit matrix	Y_r	Referenced output matrix
$\tilde{\varepsilon}$	State quantity error	Q, R	Weight matrices
\tilde{u}	Control quantity error	N_p	Predictions horizon
$\Delta\tilde{\mu}$	The increment of \tilde{u}	N_c	Control horizon
$\Delta\tilde{\mu}_{\max}$	The allowed maximum $\Delta\tilde{\mu}$		

Table 1. Employed model parameters.

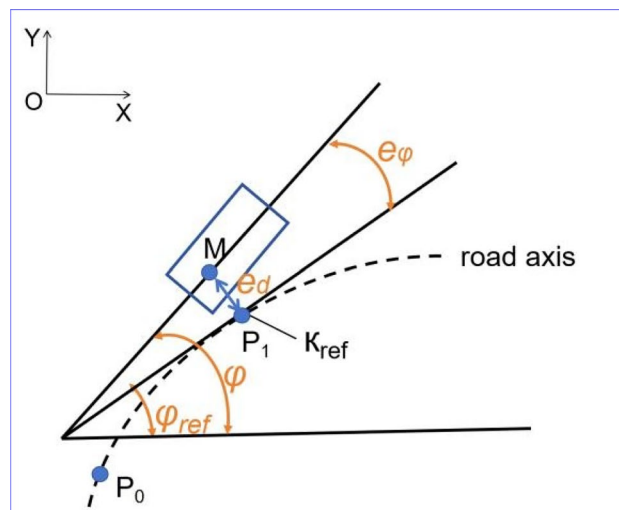


Figure 3. Vehicle tracking error model.

where R is the steering radius of vehicle's rear wheel.

Under the condition that the angular velocity of point M is the same as that of point P_1 , the following relation can be obtained:

$$\omega = \dot{S}/R = [v_\chi \cos(e_\varphi) - v_\gamma \sin(e_\varphi)]/(R - e_d) \tag{31}$$

where S is the arc length between P_1 and P_0 .

Substituting Eq. (30) into Eq. (31), \dot{S} can be expressed as:

$$\dot{S} = \frac{1}{1 - \kappa_{ref}e_d} [v_\chi \cos(e_\varphi) - v_\gamma \sin(e_\varphi)] \tag{32}$$

It can be seen from Fig. 3 that:

$$\begin{cases} e_\varphi = \varphi - \varphi_{road} \\ \dot{e}_d = v_\chi \sin(e_\varphi) + v_\gamma \cos(e_\varphi) \end{cases} \tag{33}$$

Therefore, the vehicle tracking error equations can be expressed as:

$$\begin{cases} \dot{e}_\varphi = \dot{\varphi} - \kappa_{ref}\dot{S} \\ \dot{e}_d = v_\chi \sin(e_\varphi) + v_\gamma \cos(e_\varphi) \end{cases} \tag{34}$$

Assuming that the heading angle deviation is small, then $\sin(e_\varphi) \approx 0$, $\cos(e_\varphi) \approx 1$.

In addition, through the assumption that the tracking error is small, $\kappa_{ref}e_d \approx 0$ can also be obtained.

As a result, Eq. (34) can be simplified as:

$$\begin{cases} \dot{e}_\varphi = \dot{\varphi} - \frac{\kappa_{ref}v_\chi}{1 - \kappa_{ref}e_d} \approx \dot{\varphi} - \kappa_{ref}v_\chi \\ \dot{e}_d = v_\chi e_\varphi + v_\gamma \end{cases} \tag{35}$$

When ignoring the vehicle's lateral speed v_γ , Eq. (36) can be obtained.

$$\dot{\varphi} = \omega = \frac{v_\chi}{R} = \kappa v_\chi \tag{36}$$

Thus, Eq. (35) can be expressed as:

$$\begin{cases} \dot{e}_\varphi = (\kappa - \kappa_{ref})v_\chi \\ \dot{e}_d = v_\chi e_\varphi \end{cases} \tag{37}$$

where $\varepsilon = [e_\varphi, e_d]^T$, $\mu = [v_\chi, \kappa]^T$.

To ensure tracking and control accuracy, we construct the cost function as follows:

$$J = (\varepsilon - \varepsilon_{ref})^T Q (\varepsilon - \varepsilon_{ref}) + (\mu - \mu_{ref})^T P (\mu - \mu_{ref}) + \lambda \rho^2 \tag{38}$$

where $\varepsilon = [e_\varphi, e_d]^T$ is the predicted state quantity under the control quantity $\mu = [v_\chi, \kappa]^T$, ε_{ref} represents the expected state quantity, μ_{ref} represents the expected control quantity which includes vehicle velocity and road curvature, Q is the state quantitative weight and P represents the control quantity weight, $\rho = [\Delta X, \Delta Y]^T$ represents the relaxing factor and λ is the relaxing coefficient.

$$\begin{cases} \Delta X_{k,t} = T * v_\chi * \cos \varphi = T * v_\chi * \cos(e_\varphi + \varphi_{ref}) \\ \Delta Y_{k,t} = T * v_\chi * \sin \varphi = T * v_\chi * \sin(e_\varphi + \varphi_{ref}) \end{cases} \tag{39}$$

$\Delta X_{k,t}$ is the component of the vehicle's longitudinal path increment in the X-axis of the inertial coordinate at the k-th time step. $\Delta Y_{k,t}$ is the component of the vehicle's longitudinal path increment in the Y-axis of the inertial coordinate at k-th time step.

Thus, nonlinear model predictive control can be realized by solving the following nonlinear minimum optimization with constraints in Eq. (41)–(43):

$$\min J(\varepsilon_t, \mu_t)$$

$$s.t. \quad \varepsilon_{k+1,t} = f(\varepsilon_{k,t}, \mu_{k,t}), \quad k = t, \dots, N - 1 \tag{40}$$

$$\varepsilon_{k,t} \in I, \quad k = t, \dots, t + N - 1, \tag{41}$$

$$\mu_{k,t} \in \Gamma, \quad k = t, \dots, t + N - 1, \tag{42}$$

$$E \leq \mu_{k,t} - \mu_{ref} \leq H, \quad k = t, \dots, t + N - 1, \tag{43}$$

Here, Eq. (41) denotes the state constraint, while Eqs. (42)–(43) are the constraint for control quantities.

In this paper, $I = \begin{bmatrix} [-0.24, 0.24] \\ [-0.7, 0.7] \end{bmatrix}$, $\Gamma = \begin{bmatrix} [v_{ref} - 0.4, v_{ref} + 0.4] \\ [-0.4, 0.4] \end{bmatrix}$, $E = \begin{bmatrix} -0.4 \\ -0.1 \end{bmatrix}$, $H = \begin{bmatrix} 0.4 \\ 0.1 \end{bmatrix}$.

The parameters in the aforementioned formulas are described in Table 2.

Parameters	Description	Parameters	Description
ε	State quantity	ρ	The relaxation factor
u	Control quantity	λ	The relaxation coefficient
e_d	The distance between P_1 and M	R	The steering radius of vehicle's rear wheel
e_φ	The heading angle deviation	κ	The steering curvature of vehicle
M	The vehicle's rear axle center	ΔX	The component of the vehicle's longitudinal path increment in the X-axis of inertial coordinate
P_1	The projection of the vehicle's rear axle center M on the road center line	ΔY	The component of the vehicle's longitudinal path increment in the Y-axis of inertial coordinate
φ_{ref}	The angle between the road's tangent line and the X-axis of the inertial coordinate	Q, P	Weight matrices
φ	The yaw angle of vehicle	N	Predictions horizon
κ_{ref}	The road curvature at point P_1	v_x	The longitudinal velocity of the vehicle in the vehicle coordinate system
P_0	A reference point on the center line of the road	v_y	The lateral velocity of the vehicle in the vehicle coordinate system
ω	The yaw angler velocity of vehicle	\dot{s}	The speed of P_1 along the road's center line
S	The arc length between point P_1 and P_0		

Table 2. Employed model parameters.

Trajectory tracking scheme design

Several trajectory tracking methods have been proposed^{29–31}. However, these methods often struggle to ensure both tracking accuracy and real-time performance at the same time. In this section, we introduce the trajectory Tracking Scheme based on MPC. The new control approach can significantly improve trajectory tracking accuracy, all while ensuring real-time calculations.

According to the analysis in Section "Trajectory tracking method based on MPC", the unmanned vehicle's control accuracy is restricted by the vehicle's nonlinear kinematics. When vehicles navigate complex curved roads, significant tracking deviations may occur. To address this issue, a novel control approach for trajectory tracking is proposed in this work. In the new control approach, vehicles choose different model discretization methods based on MPC according to the road curvature. The linear prediction model is used to predict and control the vehicle's trajectory when driving on roads with straight sections or small curvatures. On the other hand, the nonlinear prediction model is used to predict and control the vehicle's trajectory on roads with significant curvatures. This method can combine the advantages of LMPC and NMPC to achieve real-time and accurate trajectory tracking.

Figure 4 depicts the suggested MPC scheme's framework. The vehicle model, system restrictions, and the optimizer make up the three blocks of the MPC module. At step k , the measured state \mathcal{E}_k is denoted as the initial state and the predicted control value u_{k-1} derived from the previous loop is denoted as the initial control value. Accordingly, the vehicle model is developed. Then, driving safety, comfort, and system constraints are designed to balance tracking accuracy, ride comfort, and vehicle maneuverability. Finally, based on the road curvature calculation result, an optimization solver is selected to calculate the control sequence μ_k according to the road curvature. Optimization solver-1, corresponding to LMPC, is selected on highways with straight stretches or mild curves. On the other hand, optimization solver-2, corresponding to NMPC with a higher tracking accuracy,

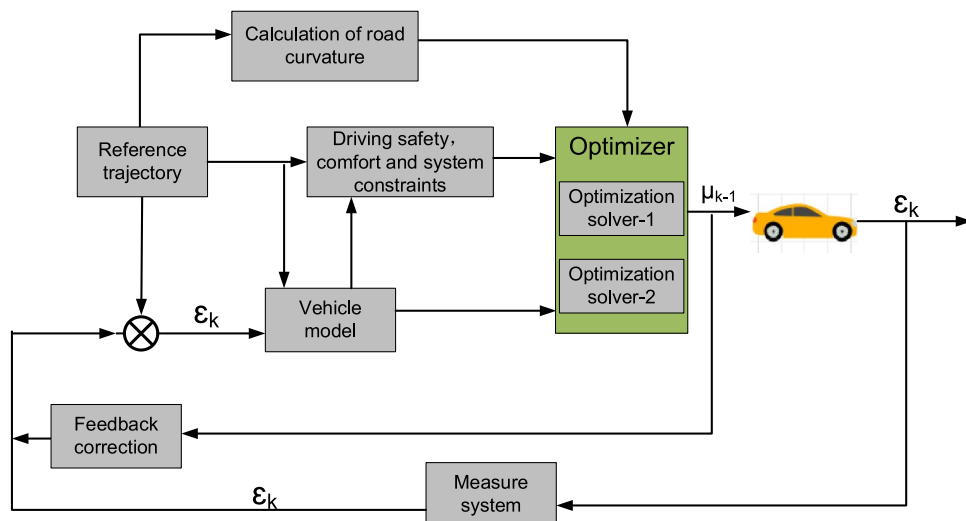


Figure 4. Structure diagram for the MPC scheme.

is selected on largely curved roads. The road curvature is calculated from the coordinates of the test road. This predictive control algorithm improves real-time performance and tracking precision at the same time. Further details regarding the optimization solver selection can be found in Section "Simulations and analysis".

Simulations and analysis

Simulation description

MATLAB2020 and CarSim2019 were used as the simulation tools, and a B class Hatchback with rear-wheel drive was used as the controlled unmanned vehicle. The test road coordinates including double-shifting road, arbitrary road, and sinusoidal road, are depicted in Fig. 5. The quadprog solver³² is used for linear model predictive control on roads with small curvature. On the other hand, the fmincon solver³³ is used for nonlinear model predictive control on roads with large curvature. The values of MPC controller parameters are shown in Table 3.

Controller selection

To verify the control accuracy and computational efficiency of the proposed predictive control algorithm, we designed three MPC controllers to compare the tracking performance on the test road:

- *Controller I* MPC controller with a linear prediction model.
- *Controller II* MPC controller with nonlinear prediction model.
- *Controller III* MPC controller combining both linear and nonlinear prediction models, as proposed in this work.

The control results from the MPC controllers I and II are analyzed under various speed conditions and road curvatures on the double-shifting road. Based on the analysis, controller III is used to enhance the tracking performance. The simulation results for MPC controller III are presented in Section "Simulations and analysis".

In order to confirm whether the suggested approach can be used to track different trajectories, the tracking results of controllers I, II, and III are compared when tracking arbitrary curves and sinusoidal trajectories. The switching curvature for different solvers of controller III is determined based on the tracking results from the double-shifting road, as detailed in Section "Simulations and analysis".

Performance evaluation when tracking double-shifting line

Performance evaluation using a single algorithm

To examine how road curvature and vehicle speed affect tracking accuracy, we tested the tracking performance of controller I and controller II when the vehicle speed was 1 m/s and 2 m/s. The influence of road curvature on tracking accuracy was also studied.

When the vehicle speed is 1 m/s, the comparative results with controller I and controller II are shown in Figs. 6 and 7. As shown in Fig. 6, the X station, Y station, and yaw angle are close to the corresponding reference value.

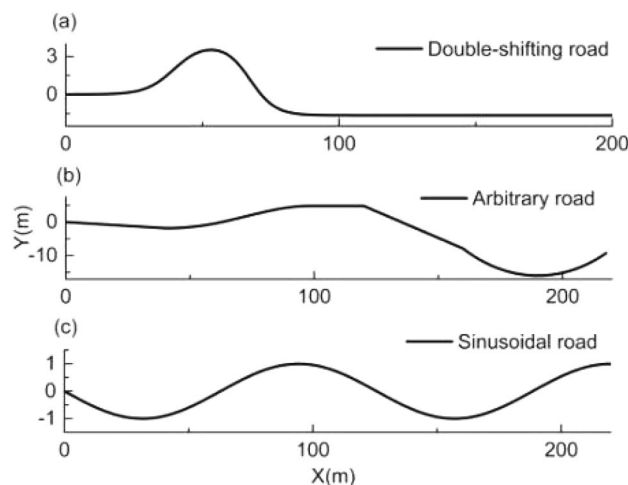


Figure 5. Test road.

Parameters	N _p	N _c	T (s)
Values for LMPC	20	30	0.1
Values for NMPC	10	2	0.03

Table 3. MPC controller parameters values.

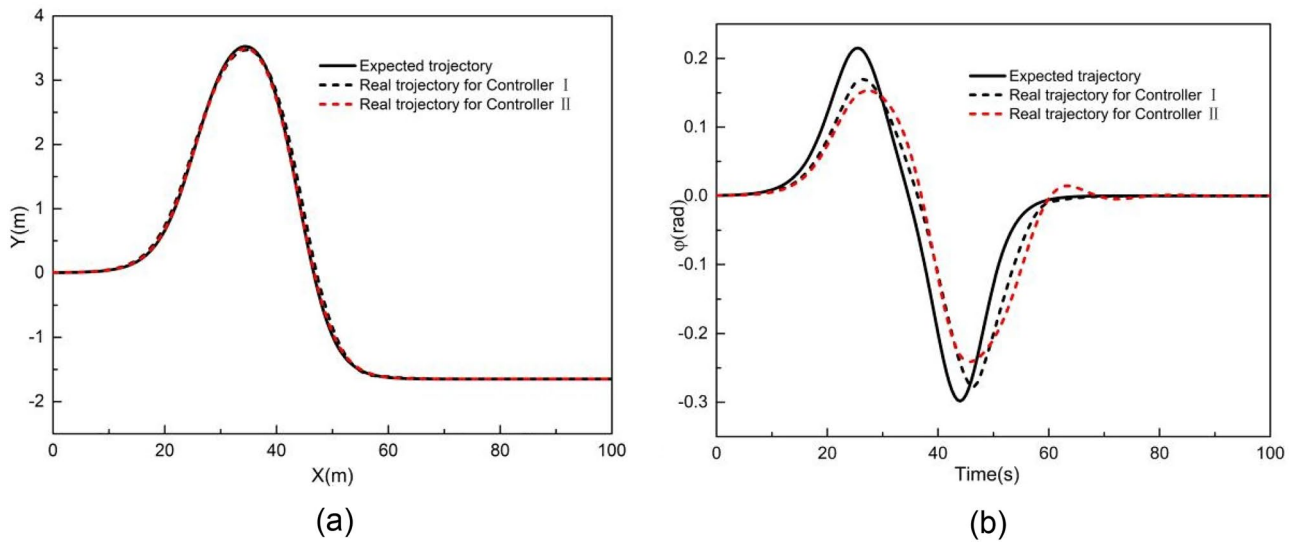


Figure 6. Comparison of (a) XY station and (b) heading angle at 1 m/s.

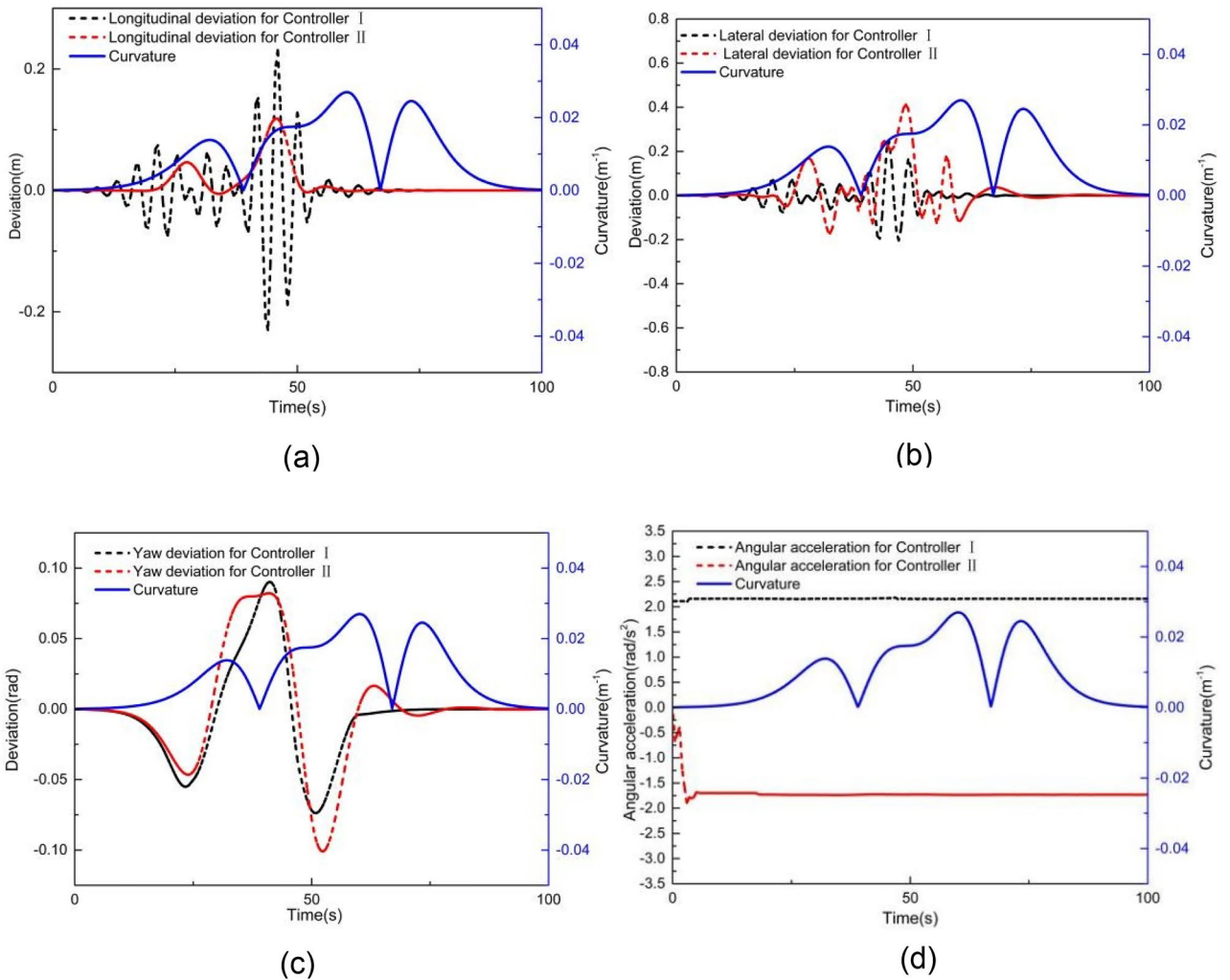


Figure 7. Deviation of (a) longitudinal position, (b) lateral position, (c) heading angle at 1 m/s considering road curvature, and (d) is the angular acceleration at 1 m/s considering road curvature.

Figure 7 presents the tracking deviation of the longitudinal trajectory, lateral trajectory, yaw angle, and angular acceleration, considering the road curvature.

It can be observed that the tracking deviation of controllers I and II is small. When using controller I, the maximum absolute longitudinal deviation is 0.23 m, the maximum absolute lateral deviation is 0.24 m, the yaw angle's maximum absolute deviation is 0.09 rad, and the maximum angular acceleration is 2.16 rad/m^2 . When using controller II, the maximum absolute deviation of longitudinal position, lateral position, and yaw angle are 0.10 m, 0.41 m, and 0.10 rad, respectively. The maximum angular acceleration is 1.90 rad/s^2 . The tracking deviation of controllers I and II is almost the same. In addition, the road curvature has little effect on tracking deviation when the vehicle speed is 1 m/s, to some extent, particularly for the Y station and heading angle.

Figures 8 and 9 present the simulation comparison results for MPC controllers I and II at 2 m/s. As shown in Fig. 8, both controllers deviate from the expected trajectory. Figure 9 illustrates the tracking deviation of the longitudinal position, lateral position, and heading angle considering the road curvature. The angular acceleration considering the road curvature is also shown in Fig. 9d. It is evident that when using controller I, the maximum absolute deviation of longitudinal position, lateral position and yaw angle are 1.41 m, 1.46 m and 0.16 rad, respectively. The maximum angular acceleration for controller I is 0.70 rad/m^2 . When using controller II, the maximum absolute deviations of longitudinal position, lateral position, and yaw angle are 0.49 m, 0.53 m and 0.29 rad, respectively. The maximum angular acceleration for controller II is 0.028 rad/m^2 .

These results clearly indicate that controller II's tracking accuracy is significantly greater than that of controller I. Moreover, the yaw acceleration of controller II is notably less than that of controller I, indicating a substantial improvement in the ride comfort for unmanned vehicles when using controller II. Additionally, it can be observed from Fig. 9 that the tracking errors of controllers I and II are small at the beginning of the simulation where the road curvature is small. However, as the simulation progresses, the longitudinal, lateral, and yaw angle deviations for controller I significantly increase with the increase in road curvature. In contrast, although the longitudinal, lateral, and yaw angle deviations for controller II also change with the increase in road curvatures, their increments are comparatively smaller. Notably, the longitudinal and lateral deviations of controller I exceed the maximum absolute error of controller II at 29 s when the road curvature is 0.017 m^{-1} . Furthermore, when the simulation time exceeds 60 s and the road curvature returns to 0, the longitudinal, lateral, and yaw angle deviations of controller II are all very close to 0, whereas the lateral deviation of controller I is still larger than that of controller II. Therefore, the tracking accuracy of controller I is better for smaller road curvature, while the tracking accuracy of controller II is better for larger road curvature.

Performance evaluation using multiple algorithms

Based on the previous analysis, we have designed controller III, which can switch algorithms according to the road curvature. Figure 10 shows the logic diagram. From Fig. 3, if the road curve is expressed by Eq. (44), the referenced road curve can be calculated through Eq. (45).

$$\begin{cases} X = f_1(t) \\ Y = f_2(t) \end{cases} \quad (44)$$

$$\begin{cases} \varphi_{\text{ref}} = \arctan(\dot{f}_2(t)/\dot{f}_1(t)) \\ \kappa_{\text{ref}} = \dot{\varphi}_{\text{ref}}/v_{\text{ref}} \end{cases} \quad (45)$$

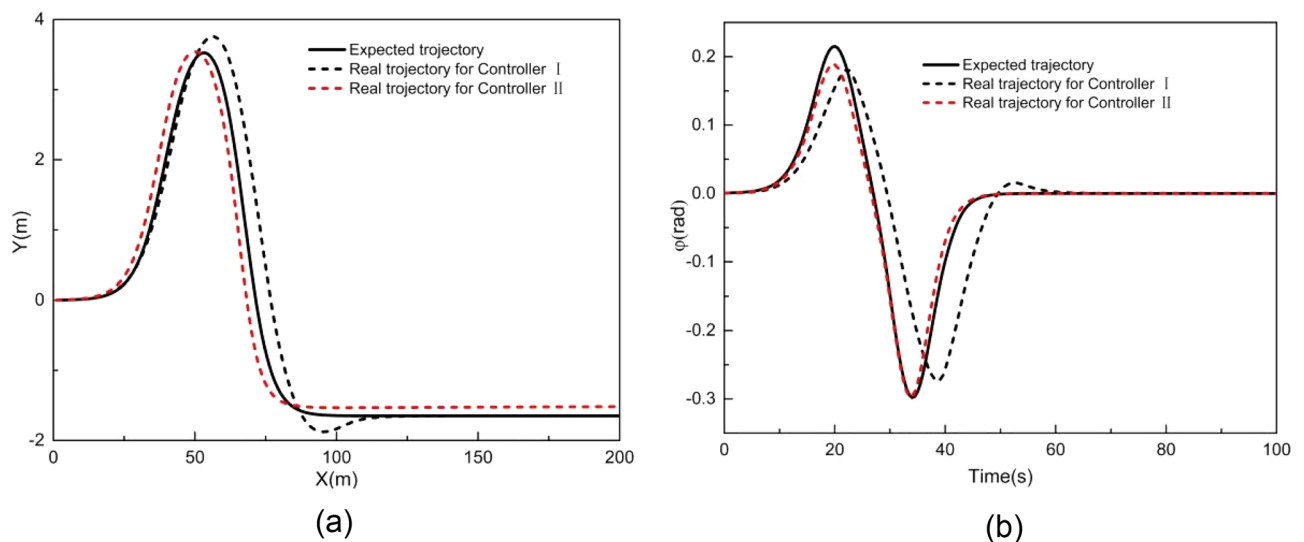


Figure 8. Comparison of (a) XY station and (b) heading angle at 2 m/s.

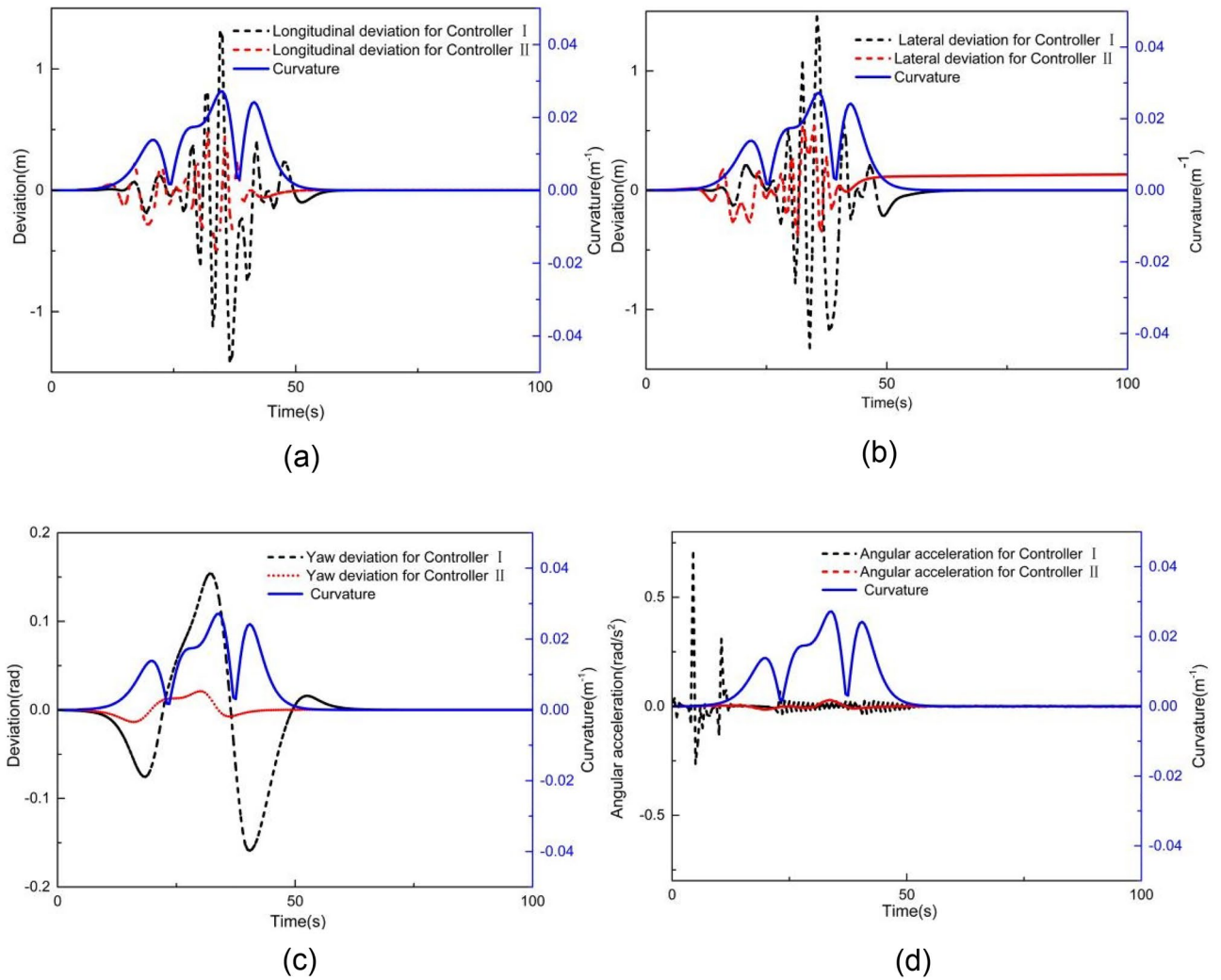


Figure 9. Deviation of (a) longitudinal position, (b) lateral position, (c) heading angle at 2 m/s considering road curvature, and (d) is the angular acceleration at 2 m/s considering road curvature.

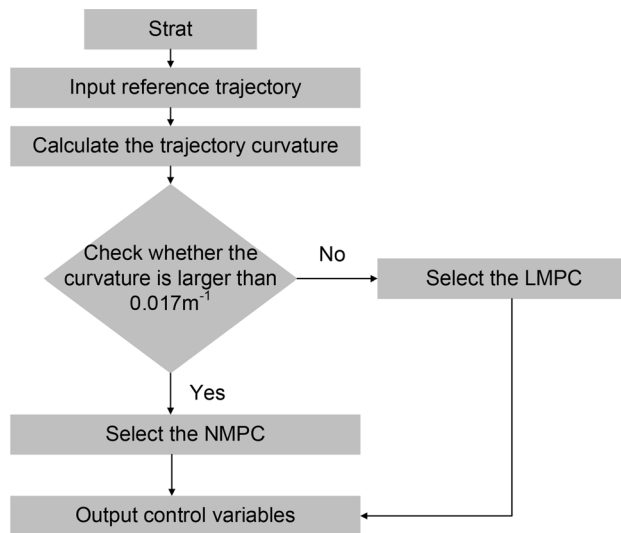


Figure 10. Logic diagram of controller III.

Building upon the double-shifting performance in Fig. 9, the longitudinal and lateral deviations of controller I exceed the maximum absolute error of controller II at 29 s when the road curvature is 0.017 m^{-1} . Thus, controller III adopts a selection mechanism: if the road curvature is less than 0.017 m^{-1} , LMPC is chosen; otherwise, NMPC is employed.

Figures 11 and 12 illustrate the tracking results with MPC controller III when the vehicle speed is 2 m/s. As shown in Fig. 11, the Y station and the yaw angle are in close alignment with the desired trajectory. Figure 12 displays the tracking deviation of longitudinal trajectory, lateral trajectory, yaw angle, and angular acceleration for controller III.

It can be observed that the maximum absolute deviation of longitudinal position, lateral position, and yaw angle are 0.68 m, 0.57 m, and 0.07 rad, respectively. These maximum absolute deviations are smaller than those of controller I. Besides, throughout the entire simulation time, the tracking deviation remains within a small range. The tracking deviation for controller III combines the advantages of controllers I and II throughout the simulation time, resulting in improved overall tracking performance.

Figures 13 and 14 illustrate the tracking results with MPC controller III at longitudinal speeds of 8 m/s and 10 m/s. As shown in Fig. 13, when the longitudinal speed are 8 m/s and 10 m/s, both Y and PHI closely follow the expected values at longitudinal speeds of 8 m/s and 10 m/s. Figure 14 presents the tracking deviations of longitudinal trajectory, lateral trajectory, and yaw angle for controller III at 8 m/s and 10 m/s. It can be seen that as the longitudinal velocity increases, the deviations also increase. The maximum absolute deviations of longitudinal position, lateral position, and yaw angle at 8 m/s are 0.07 m, 0.89 m, and 0.09 rad, respectively. The maximum absolute deviations of longitudinal position, lateral position, and yaw angle at 10 m/s are 0.19 m, 1.29 m, and 0.15 rad, respectively. This may result from unmodeled uncertainties in the kinematic model.

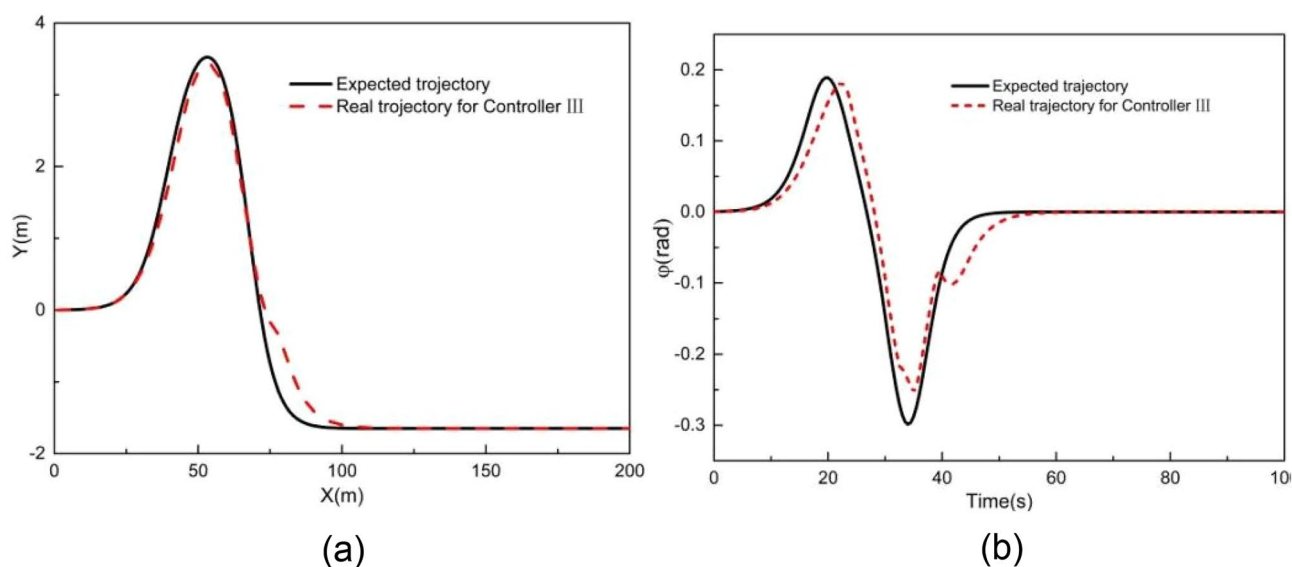


Figure 11. Tracking result of (a) XY station and (b) heading angle at 2 m/s using controller III.

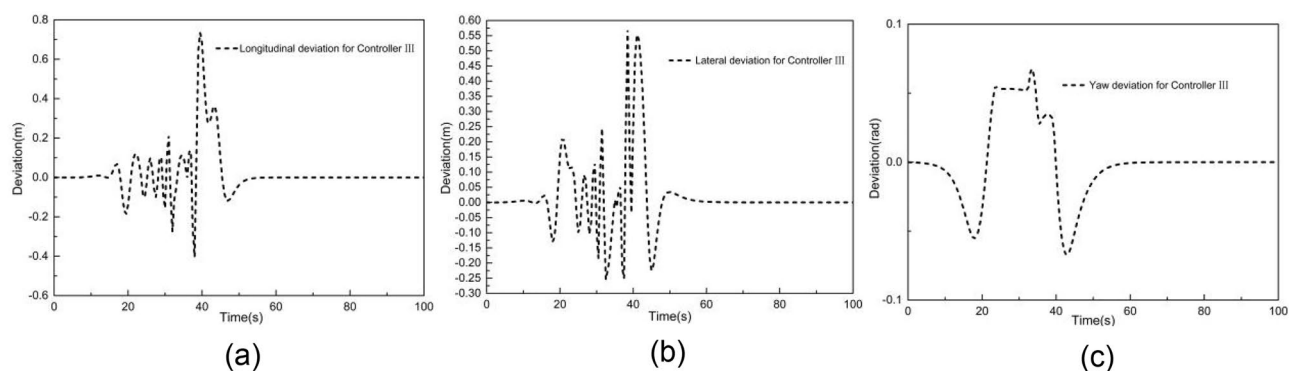


Figure 12. Deviation of (a) longitudinal position, (b) lateral position, (c) heading angle at 2 m/s using controller III.

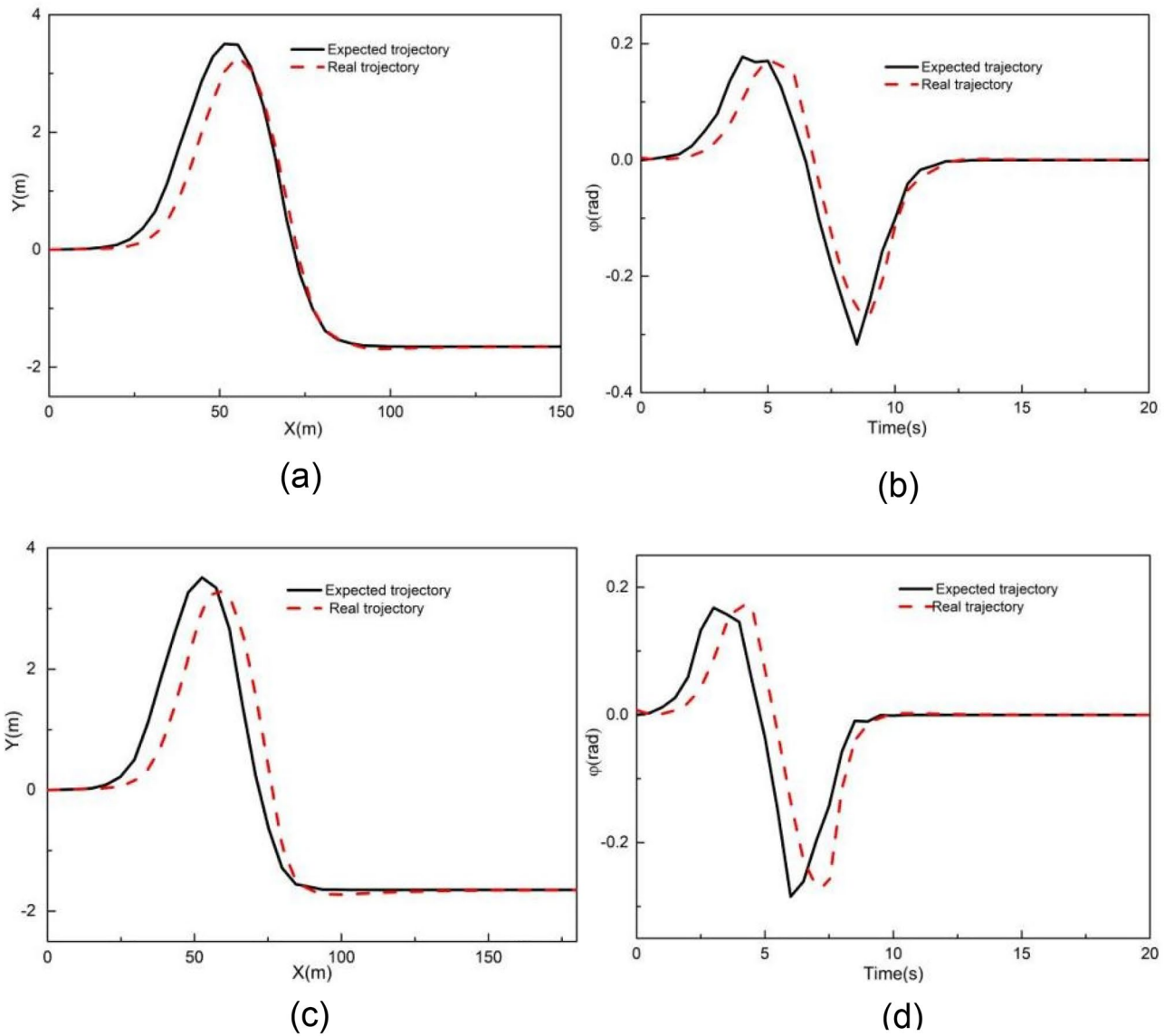


Figure 13. Tracking result of (a) XY station at 8 m/s, (b) heading angle at 8 m/s, (c) XY station at 10 m/s and (d) heading angle at 10 m/s using controller III.

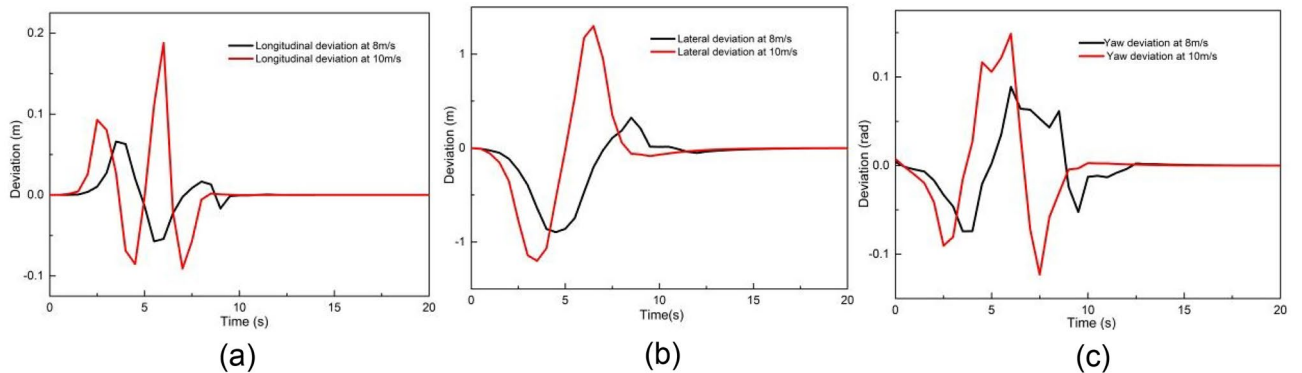


Figure 14. Deviation of (a) longitudinal position, (b) lateral position, (c) heading angle at 8 m/s and 10 m/s using controller III.

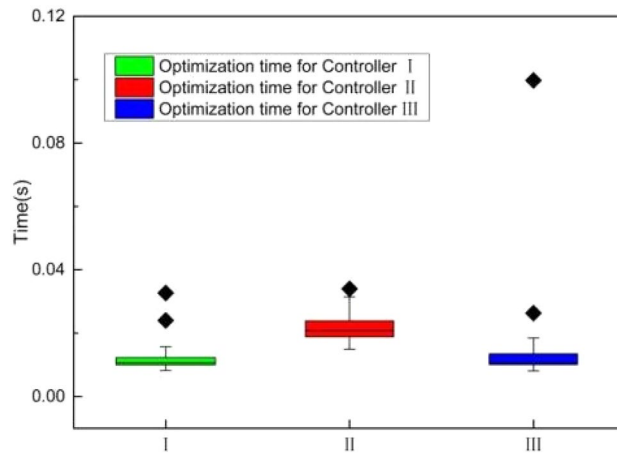


Figure 15. Comparison of the optimization time.

Real-time performance

The three designed controllers' mean optimization time and their standard deviation, at 2 m/s, are depicted in Fig. 15. Controller III's average optimization time is 0.013 s, which is considerably less than controller II's and somewhat greater than controller I's. This suggests that for controller III, the optimization time is comparatively constant. As a result, the proposed MPC scheme is well-suited for real-time applications.

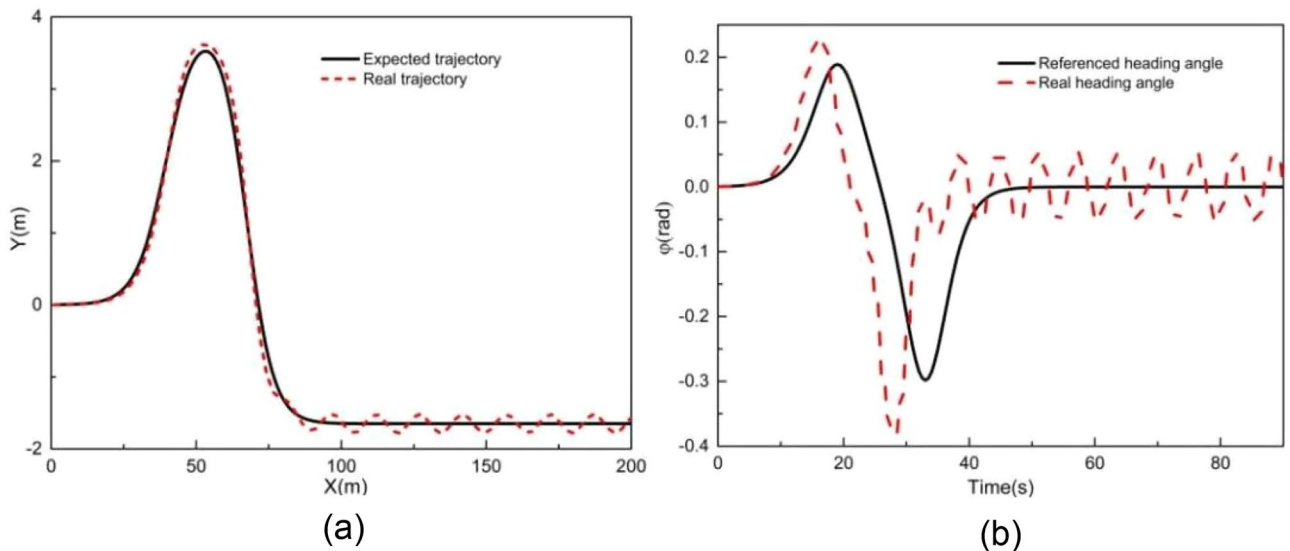


Figure 16. Tracking result of (a) XY station and (b) heading angle at 2 m/s using Stanley algorithm.

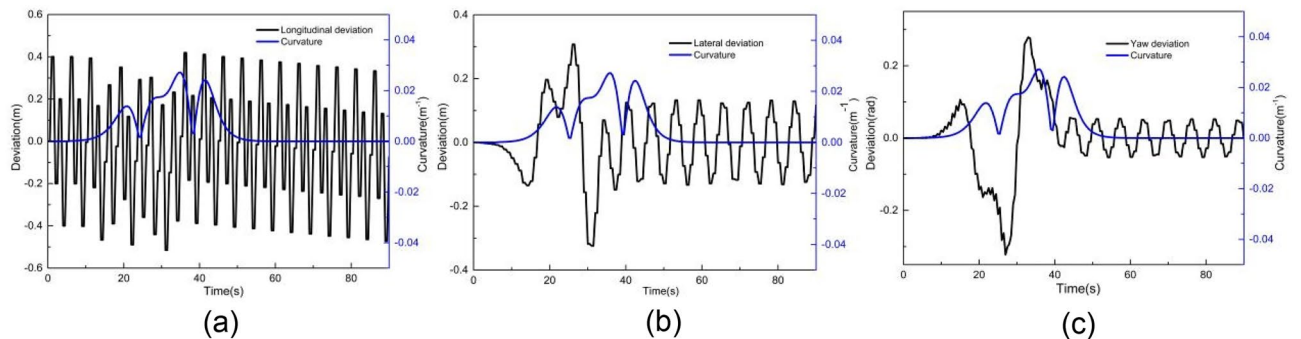


Figure 17. Deviation of (a) longitudinal position, (b) lateral position, (c) heading angle at 2 m/s using Stanley algorithm.

Performance evaluation using Stanley algorithm

To demonstrate the capability of the proposed scheme to address both tracking accuracy and real-time performance, we use Stanley algorithm to track the double-shifting line at a target longitudinal speed of 2 m/s. Figures 16 and 17 illustrate the tracking results with Stanley algorithm at a vehicle speed of 2 m/s. As illustrated in Fig. 16, there are certain deviations between the Y station and the yaw angle with respect to the desired trajectories. Figure 17 presents the tracking deviations of longitudinal trajectory, lateral trajectory, and yaw angle for Stanley algorithm. It can be observed that the maximum absolute deviations of longitudinal position, lateral position, and yaw angle are 0.51 m, 0.32 m, and 0.32 rad, respectively. These maximum absolute deviations are larger than those of controller III. In addition, the trajectory jitters occur after the tracking time of 30 s. The average computation time for the Stanley algorithm is $6E-4$ s, which is slightly shorter than that of Controller III's. These results indicate that the proposed method significantly improves trajectory tracking accuracy, all while ensuring real-time calculations.

Performance evaluation when tracking an arbitrary curve

Figures 18 and 19 illustrate the tracking results of different controllers when the vehicle speed is 2 m/s on an arbitrary road. Building upon the double-shifting performance, controller III adopts a selection mechanism: if the road curvature is less than 0.017 m^{-1} , LMPC is chosen; otherwise, NMPC is employed.

As shown in Fig. 18, both controllers I and II exhibit some deviation in Y station and heading angle from the expected trajectory, whereas controller III achieves better tracking accuracy, with the Y station and yaw angle closely aligned with the desired trajectory.

Figure 19 provides a comparison of tracking deviations in longitudinal trajectory, lateral trajectory, and yaw angle. When using controller I, the maximum absolute deviation of longitudinal position, lateral position, and yaw angle are 0.19 m, 2.27 m, and 0.28 rad, respectively. When using controller II, these values improve to 0.09 m, 1.64 m, and 0.14 rad, respectively. However, when using controller III, the maximum absolute deviations of longitudinal position, lateral position, and yaw angle further reduced to 0.07 m, 1.30 m, and 0.17 rad, respectively.

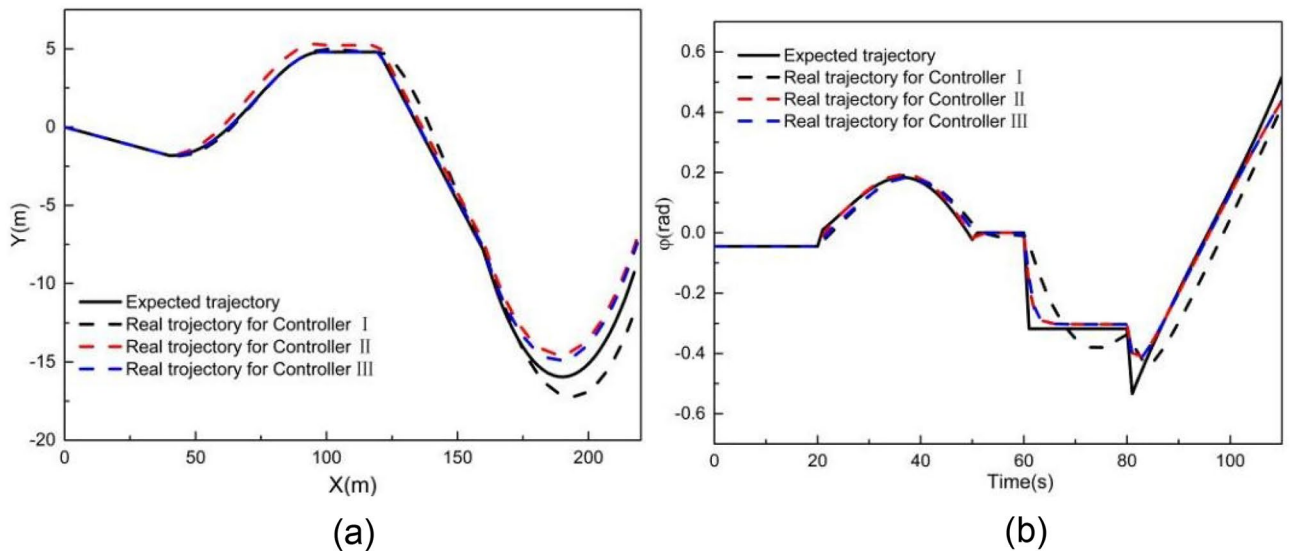


Figure 18. Comparison result when tracking an arbitrary curve at 2 m/s (a) XY station and (b) heading angle.

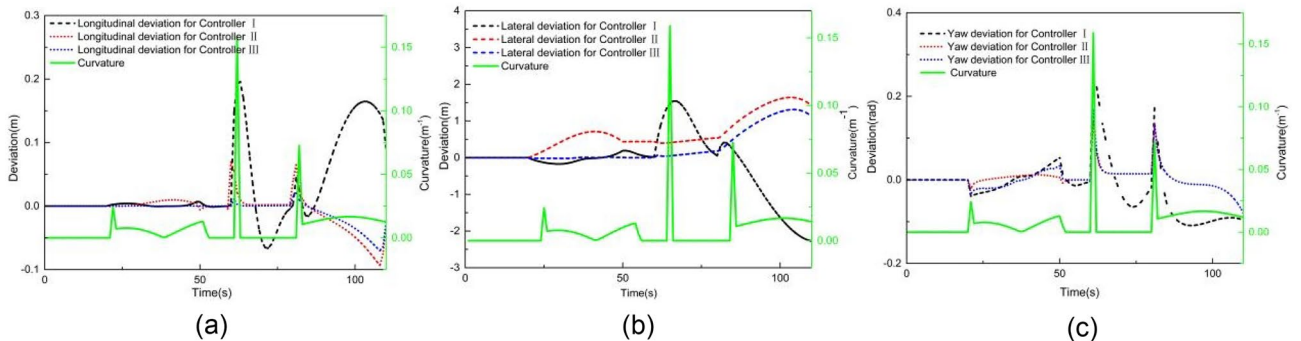


Figure 19. Deviation of (a) longitudinal position, (b) lateral position, (c) heading angle at 2 m/s when tracking an arbitrary curve.

These results demonstrate that controller III outperforms both controllers I and II in terms of tracking accuracy on arbitrary-curve roads.

Performance evaluation when tracking a sinusoidal curve

Figures 20 and 21 show the tracking results, at 2 m/s, on a sinusoidal road. As shown in Fig. 20, controller II exhibits some deviation in Y station and heading angle from the expected trajectory. However, controllers I and III achieve better tracking accuracy, with the Y station and yaw angle closely following the desired trajectory.

Figure 21 provides a comparison of tracking deviations in longitudinal trajectory, lateral trajectory, and yaw angle. As the curvature of the entire sinusoidal road is smaller than 0.017 m^{-1} , controllers I and III exhibit the same tracking deviations. When using these controllers, the maximum absolute deviations of longitudinal position, lateral position, and yaw angle are $8.68\text{E} - 5 \text{ m}$, 0.014 m , and 0.006 rad , respectively. When using controller II, the maximum absolute deviation of longitudinal position, lateral position, and yaw angle are $2.6\text{E} - 4 \text{ m}$, 0.26 m , and 0.00026 rad , respectively. The result of Fig. 21 shows that controller III can still guarantee the tracking accuracy when the road curvature is small. Therefore, in this case, the tracking accuracy can also be guaranteed through intelligent selection of MPC algorithm.

Conclusion

In this paper, we analyzed the effects of road curvature and vehicle speed on tracking accuracy using both LMPC and NMPC algorithms under the double line shifting condition. Based on the analysis results, we introduced a new trajectory tracking method to improve tracking accuracy. The feasibility of the new method was verified on different testing roads.

The key conclusions drawn from this study are as follows:

- (1) At a vehicle speed of 1 m/s, both the LMPC and NMPC demonstrate good tracking performance with very small tracking deviation. Additionally, the road curvature does not significantly affect the tracking accuracy at this speed.

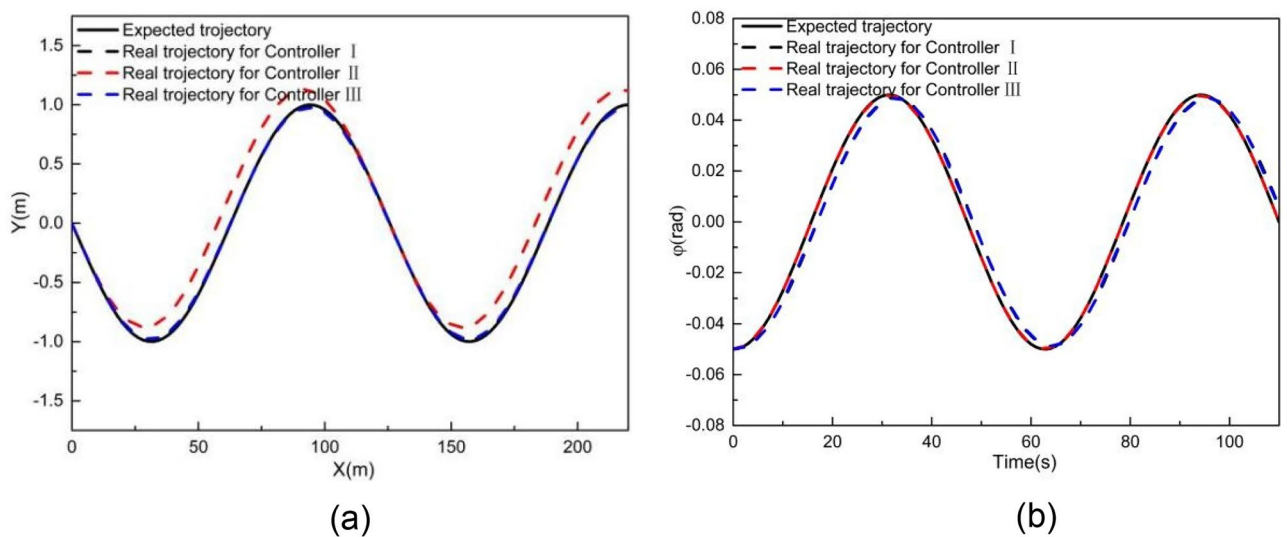


Figure 20. Comparison result when tracking a sinusoidal curve at 2 m/s (a) XY station and (b) heading angle.

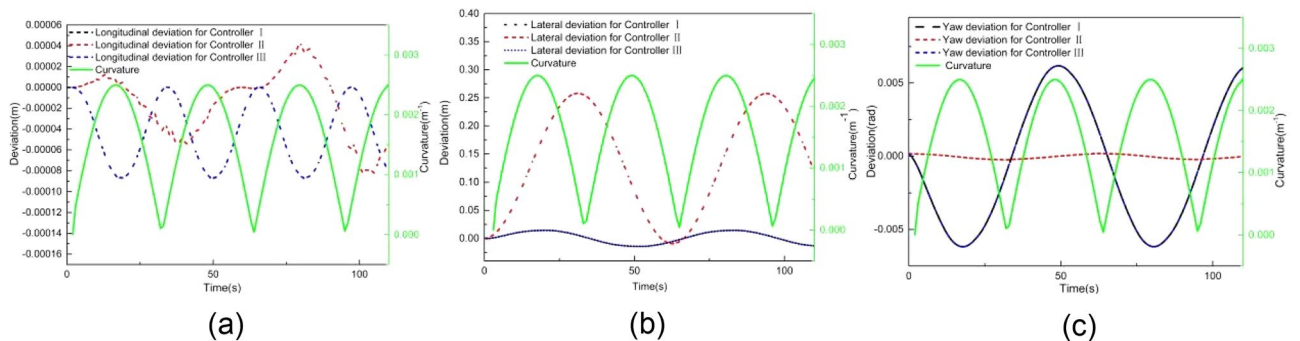


Figure 21. Deviation of (a) longitudinal position, (b) lateral position, (c) heading angle at 2 m/s when tracking a sinusoidal curve.

- (2) At a vehicle speed of 2 m/s, LMPC shows better tracking accuracy when the road curvature is small, while NMPC performs better when the road curvature is large.
- (3) The proposed trajectory tracking method significantly improves tracking accuracy at a vehicle speed of 2 m/s, while maintaining a comparable calculation time to LMPC. This finding suggests that the proposed method achieves a desirable balance between tracking accuracy and computational efficiency.
- (4) The NMPC algorithm based on tracking errors in this work considers the effect of road curvature and vehicle curvature on tracking accuracy, making it suitable for tracking roads with large curvature. This algorithm greatly simplifies the calculation formula for tracking curved roads compared to the vehicle dynamic model.
- (5) The proposed method intelligently switches between LMPC and NMPC based on the road curvature, making it applicable to a wide range of complex trajectories.

It should be noted that this work is based on the 2-DOF vehicle model. In the future, we will consider the applicability of this model to the 3-DOF vehicle model and conduct real-vehicle experiments to further verify the effectiveness of the proposed control scheme.

Data availability

The data supporting the results reported in the article are available from the corresponding author on reasonable request.

Received: 15 January 2024; Accepted: 21 April 2024

Published online: 10 May 2024

References

1. Zhu, D., Yang, S. X. & Biglarbegian, M. A fuzzy logic-based cascade control without actuator saturation for the unmanned underwater vehicle trajectory tracking. *J. Intell. Robot. Syst.* **106**(2), 39 (2022).
2. Li, B. Y., Zhang, B. J., Du, H. P., Wu, Y. & Chen, S. Z. Trajectory planning, dynamics modelling and trajectory tracking method for off-road autonomous vehicles considering the road topography information. *Int. J. Veh. Des.* **87**(1–4), 170–198 (2021).
3. Hu, C., Wang, R. R., Yan, F. J. & Chen, N. Output constraint control on path following of four-wheel independently actuated autonomous ground vehicles. *IEEE Trans. Veh. Technol.* **65**(6), 4033–4043 (2016).
4. Wang, J. E., Chen, W. W., Wang, T. B., Wang, M. L. & Xiao, L. Z. Vision guided intelligent vehicle lateral control based on desired yaw rate. *J. Mech. Eng.* **48**(4), 108–115 (2012).
5. Tang, Y. F. Research on trajectory tracking control method of autonomous navigation vehicle. *PhD Thesis, Dalian University of Technology, DaLian, China.* (2012).
6. Guo, J. H., Li, K. Q. & Luo, Y. G. Review on the research of motion control for intelligent vehicles. *J. Automot. Saf. Energy.* **7**(2), 151–159 (2016).
7. Kayacan, E., Ramon, H. & Saeys, W. Robust trajectory tracking error model-based predictive control for unmanned ground vehicles. *IEEEASME Trans. Mechatron.* **21**(2), 806–814 (2016).
8. Suryanarayanan, S. Fault-tolerant control and its application to lane-keeping control of automated vehicles. *California, USA: University of California, Berkeley.* 55–60 (2002).
9. Chen, T. & Chen, D. Lateral control of intelligent vehicle based on neural networks sliding mode. *Transducer Microsyst. Technol.* **36**(5), 63–67 (2017).
10. Wang, H. *et al.* Crash mitigation in motion planning for autonomous vehicles. *IEEE Trans. Intell. Transp. Syst.* **20**(9), 3313–3323 (2019).
11. Bruschetta, M., Maran, F. & Beghi, A. A fast implementation of MPC-based motion cueing algorithms for mid-size road vehicle motion simulators. *Veh. Syst. Dyn.* **55**(6), 802–826 (2017).
12. Prach, A. & Kayacan, E. An MPC-based position controller for a tilt-rotor tricopter VTOL UAV. *Optim. Control Appl. Methods.* **39**(1), 343–356 (2018).
13. Xi, Y. G. *Predictive Control* (National Defense Industry Press, 2013).
14. Han, J. Q. *In Active Disturbance Rejection Control Technique—A Control Technique that Estimates and Compensates for Uncertainties* (National Defense Industry Press, 2013).
15. Li, S. H., Yang, Z. K. & Wang, X. W. Trajectory tracking control of an intelligent vehicle based on T-S fuzzy variable weight MPC. *J. Mech. Eng.* **59**(4), 199–212 (2023).
16. Cui, Z. Z., Lu, Y. & Zhao, Z. Q. Driverless vehicle trajectory tracking algorithm based on MPC. *Automob. Technol.* **47**(21), 43–46 (2022).
17. Elvik, R., Vadeby, A., Hels, T. & Schagen, I. V. Updated estimates of the relationship between speed and road safety at the aggregate and individual levels. *Accid. Anal. Prev.* **123**, 114–122 (2019).
18. Abbas, M. A., Eklund, J. M. & Milman, R. Real-time analysis for nonlinear model predictive control of autonomous vehicles. *in 2012 25th IEEE Canadian Conference on Electrical and Computer Engineering (CCECE), Montreal, QC, Canada: IEEE.* 1–4 (2012).
19. Wang, K., Li, Q., Wang, Z. Y. & Yang, J. F. Trajectory tracking control for automated vehicle based on NMPC considering vehicle rolling motion. *Control Decis.* **37**(10), 2535–2542 (2022).
20. Rafaila, R. C. & Livint, G. Predictive control of autonomous steering for ground vehicles. *in 2015 9th International Symposium on Advanced Topics in Electrical Engineering (ATEE), Bucharest, Romania: IEEE.* 543–547 (2015).
21. Yang, K. *et al.* Comparative study of trajectory tracking control for automated vehicles via model predictive control and robust H-infinity state feedback control. *J. Chin. J. Mech. Eng.* **34**, 74 (2021).
22. Qiu, D. Y., Qiu, D. L., Wu, B., Gu, M. & Zhu, M. F. Hierarchical control of trajectory planning and trajectory tracking for autonomous parallel parking. *J. IEEE Access.* **9**, 94845–94861 (2021).
23. Lei, Z. L., & Tang, X.M. Trajectory tracking for unmanned vehicle based on model predictive control using linear matrix inequalities. *Chinese Automation Congress (CAC).* (2020).
24. Mogens, M. G. P. & Alberto, B. Reference trajectory planning under constraints and path tracking using linear time-varying model predictive control for agricultural machines. *Biosyst. Eng.* **153**, 28–41 (2017).
25. Zhou, L., Wang, G. Q., Sun, K. K. & Li, X. Trajectory tracking study of track vehicles based on model predictive control. *J. Mech. Eng.* **65**, 329–342 (2019).
26. Huang, Z. J. *et al.* A new trajectory tracking algorithm for autonomous vehicles based on model predictive control. *Sensors* **21**, 7165 (2021).

27. Ye, B. L., Niu, S. F., Li, L. X. & Wu, W. M. A comparison study of kinematic and dynamic models for trajectory tracking of autonomous vehicles using model predictive control. *Int. J. Control Autom. Syst.* <https://doi.org/10.1007/s12555-022-0337-8> (2023).
28. Mohamed, E. *et al.* Non-Linear model predictive control using casadi package for trajectory tracking of quadrotor. *Energies* **16**, 2143 (2023).
29. Zhao, K. G. *et al.* Research for nonlinear model predictive controls to laterally control unmanned vehicle trajectory tracking. *Appl. Sci.* **10**, 6034 (2020).
30. Hu, K. L. & Cheng, K. Trajectory planning for an articulated tracked vehicle and tracking the trajectory via an adaptive model predictive control. *Electronics* **12**, 1988 (2023).
31. Sebastian, T. & Mike, M. S. The robot that won the darpa grand challenge. *J. Field Robot.* **23**(9), 661–692 (2006).
32. Torrente, M. & Pierpaolo, U. A rescaling technique to improve numerical stability of portfolio optimization problems. *Soft Comput.* **27**, 12831–12842 (2023).
33. Gang, Z. & Ye, D. Optimal short-range rendezvous using on-off constant thrust. *Aerosp. Sci. Technol.* **1**, 1–9 (2017).

Author contributions

L.F. wrote the first draft; S.Z. supervised the manuscript and H.B. reviewed and edited the manuscript.

Funding

This work was funded by the State Key Laboratory of Mechanical Transmission for Advanced Equipment, Chongqing University (SKLMT-MSKFKT-202221) and the National Nature Science Foundation of China under Grant 52002025.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to H.R.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2024