symmetry

MDPI

Article

Location Adaptive Model Predictive Controller for Autonomous Vehicle Path Tracking with Location Drifting

Jia Xu [1], Xiang Xu [2], Xiaoyan Huang [2,*], Yuanyuan Wang [3,*], Yue Yu [4] and Nan Zhou [4,5]

[1] Audit Office, Sichuan University of Arts and Science, Dazhou 635000, China; xujia@sasu.edu.cn
[2] School of Automation, Chengdu University of Information Technology, Chengdu 610225, China; karryxu616@outlook.com
[3] Dazhou Radio Monitoring Station, Dazhou 635000, China
[4] School of Electronic Information and Electrical Engineering, Chengdu University, Chengdu 610106, China; yuyue@cdu.edu.cn (Y.Y.); zhounan123@cdu.edu.cn (N.Z.)
[5] Tianfu Jiangxi Laboratory, Chengdu 610041, China
* Correspondence: hxy@cuit.edu.cn (X.H.); wyyxj0711@outlook.com (Y.W.)

Abstract

With the rapid development of autonomous driving, path tracking has emerged as a pivotal research direction. Model predictive control (MPC) has become one of the most prevailing approaches for path tracking, owing to its superior capacity in dealing with multi-constrained control problems and compatibility with the symmetry of vehicle dynamic systems. Nevertheless, conventional MPC suffers from performance degradation in path tracking when vehicle localization drift occurs, referring to the noticeable deviation between sensor-measured position and actual physical position over time, which is mainly induced by sensor noise and outliers. To overcome these limitations and enhance the accuracy and stability of path tracking, this paper presents a location-adaptive model predictive control framework. Specifically, a supervisor is designed to detect localization drift, and a Runge–Kutta-based location estimator is activated to predict the current vehicle state once drift is identified. Furthermore, a linear time-varying MPC is utilized to compute the desired control input for real-time multi-objective optimization. A set of co-simulations based on Simulink and CarSim are conducted to validate the effectiveness of the proposed strategy. Numerical results demonstrate that the presented method outperforms traditional MPC in terms of tracking accuracy and stability under localization drift conditions.

Keywords: autonomous driving; path tracking; model predictive control; position and pose adaption

Check for updates

Academic Editor: Chengxi Zhang

Received: 5 January 2026
Revised: 30 January 2026
Accepted: 3 February 2026
Published: 7 February 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Autonomous driving has increasingly become a research hotspot and promising solution to mitigate urban traffic congestion and reduce traffic accident rates, which is capable of fundamentally improving the efficiency and safety of modern transportation systems. The research framework of autonomous driving is generally divided into three core modules: environmental perception and localization, navigation [1], decision-making and planning, as well as trajectory tracking control [2]. Among these critical components, path tracking control (PTC) serves as the indispensable core link for autonomous vehicles (AVs). It is responsible for generating reliable and effective control sequences to ensure that the vehicle accurately tracks the predefined reference path with satisfactory stability [3].

To improve the tracking accuracy and stability of PTC for AVs, a variety of classical control algorithms have been extensively investigated and applied in existing research, including proportional–integral–derivative (PID) control [4], sliding mode control (SMC) [5,6], linear quadratic regulator (LQR) control [7–9], pure pursuit (PP) control [10,11], and model predictive control (MPC) [12–14].

It is well acknowledged that the practical operation of autonomous vehicles is subject to various inherent physical constraints of the vehicle system and external environmental restrictions. Unfortunately, most of the aforementioned conventional control algorithms suffer from prominent inherent limitations in addressing the multi-constraint PTC problem of AVs. A common unrealistic assumption adopted by these traditional methods is that the control inputs generated by the controller will never reach the saturation limits of vehicle actuators. This oversimplified premise often leads to the production of unreasonable and even infeasible control signals in actual vehicle operation, which directly degrades the path tracking accuracy and dynamic stability of AVs, and even induces potential control risks in complex driving scenarios. In addition, partial algorithms such as PID lack sufficient robustness against external disturbances, SMC is prone to chattering phenomena, and LQR relies heavily on accurate linear vehicle models, all of which further limit their practical application performance in real-world multi-constraint conditions.

Against this background, model predictive control (MPC) has emerged as one of the most effective and practical control approaches to tackle the above-mentioned multi-constraint dilemmas in PTC tasks. The superior performance of MPC stems from its unique three-step implementation mechanism: predictive modeling, rolling horizon optimization, and real-time feedback correction [15], which enables it to explicitly incorporate various physical constraints of the vehicle into the optimization process. Benefiting from this advantage, a wealth of MPC-based improved algorithms have been successively proposed for the PTC problem to further enhance control performance. For instance, Wu et al. designed a linear time-varying MPC (LTV-MPC) that locally linearizes the nonlinear vehicle model at each sampling instant to improve tracking adaptability [16]. Zhang et al. proposed a modified MPC method with a dynamically adjustable prediction horizon for real-time optimization. Dai et al. developed an adaptive MPC approach integrated with preview characteristics and longitudinal speed constraints, which effectively boosted the path tracking precision [17].

Moreover, the MPC framework also exhibits a certain degree of anti-disturbance capability, which can alleviate the adverse impacts of various external disturbances on the vehicle control system [18–20]. However, a critical and unresolved defect still exists in the existing MPC-based PTC methods, which severely restricts their control performance and practical application reliability: they are incapable of handling vehicle position drifting. Vehicle location drifting refers to a phenomenon in autonomous driving systems where the measured location of a vehicle (obtained via on-board sensors like GPS, LiDAR, or cameras) deviates noticeably from the vehicle's actual physical location over time. This unresolved issue brings persistent negative effects on the control system: once positioning drift occurs, the real-time state of the vehicle cannot be accurately captured and timely restored, which not only leads to a significant degradation of path tracking accuracy, but also poses a serious threat to driving safety and may even trigger traffic accidents in extreme cases.

To fill this critical research gap and tackle the positioning drift problem in MPC-based path tracking control, a novel MPC-based control method integrated with adaptive location correction is proposed in this paper, which is termed Location Adaptive MPC (LMPC). The proposed LMPC method consists of two core and complementary functional steps to achieve high-performance path tracking under positioning drift conditions. First, a linear time-varying model predictive controller (LTV-MPC) is constructed as the basic path track-

ing controller to ensure the baseline tracking stability and constraint satisfaction. Second, a dedicated supervisor module is designed to detect the occurrence of vehicle positioning drift by conducting a simple and rough estimation of the real-time pose and position of the vehicle. When the positioning drift is detected, a dynamic location correction strategy based on the fourth-order Runge–Kutta method is immediately activated to perform accurate real-time location estimation and correction. Through the above integrated design, the proposed LMPC method is able to effectively suppress the adverse effects of positioning drift and significantly improve the path tracking performance of autonomous vehicles in the presence of location drift. The flowchart of the proposed method is illustrated in Figure 1.

The structure of this paper is as follows. Section 2 details the autonomous vehicle model employed for path tracking control. Section 3 outlines the proposed model predictive control (MPC) framework with adaptive position and pose correction. Section 4 presents the co-simulation results of the proposed approach. Section 5 provides the study's conclusions.

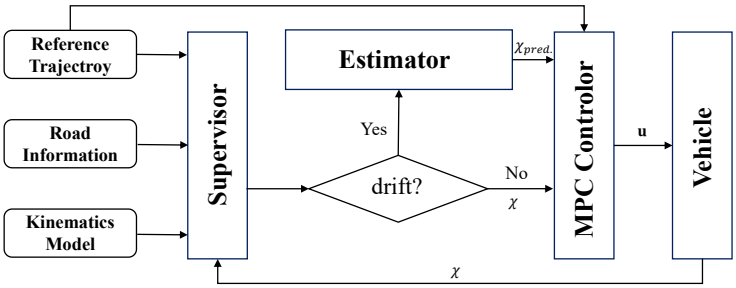


Figure 1. The flowchart of the proposed method.

2. Modeling

An autonomous vehicle is a multi-degree-of-freedom and complex nonlinear time-varying system. Establishing a model that can fully reflect the vehicle's characteristics is difficult. A precise model for autonomous vehicles usually results from high complexity and requires an enormous computational burden. As the dynamic model used in [21], the bicycle kinematics model, which takes "symmetry" of vehicle structure into account, is utilized with the following assumptions:

1. Ignore the vertical movement of the vehicle.
2. The vehicle body and its suspension system are rigid.
3. Consider the front wheel steering of the vehicle only.
4. The single-track model describes vehicle movement by assuming that the steering angles of the left and right front wheels are identical.

The vehicle kinematic model is illustrated in Figure 2. Point Q is both the instantaneous turning center and the vehicle's center of mass. Point N represents the left and right front wheels, while point M represents the rear wheels. Lines OM and ON are perpendicular to the orientations of the rear and front wheels, respectively, intersecting at point O, the geometric reference. All model notations are defined in Table 1.

Table 1. Kinematics model symbol definition.

Notation	Definition	Notation	Definition
δ_f	front wheel steering angle	δ_r	rear wheel steering angle
l_f	front suspension length	l_r	rear suspension length
β	slip angle	φ	heading angle
v	velocity of center of mass	R	turning radius
M	rear wheel center	N	front wheel center
Q	vehicle center of mass	O	turning center

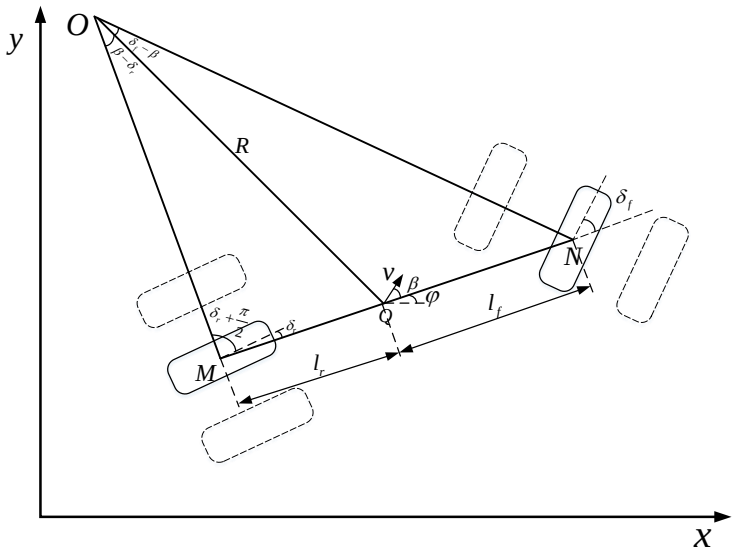


Figure 2. Vehicle kinematics model.

The expression of the kinematics model of the vehicle is

$$\begin{cases} \dot{x} = v\cos(\varphi + \beta) \\ \dot{y} = v\sin(\varphi + \beta) \\ \dot{\varphi} = \frac{v\cos\beta}{l_f + l_r}(\tan\delta_f - \tan\delta_r) \end{cases} \tag{1}$$

Assume the vehicle operates in steady driving conditions, with a small sideslip angle such that $\beta = 0$. The model only accounts for front-wheel steering, so setting $\delta_r = 0$ is entirely reasonable. With these constraints, the preceding model simplifies to

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} \cos\varphi \\ \sin\varphi \\ \frac{\tan\delta}{l} \end{bmatrix} v \tag{2}$$

where $l = l_f + l_r$ represents the vehicle wheelbase, with $\delta = \delta_f$. We define the system state vector and control input vector as $\chi = [x, y, \varphi]^{\mathrm{T}}$ and $\mathbf{u} = [v, \delta_f]^{\mathrm{T}}$, respectively. From these definitions, the system dynamics take its general form:

$$\dot{\chi} = f(\chi, \mathbf{u}) \tag{3}$$

For path tracking efficiency, we linearly approximate the nonlinear model in (3) using a first-order Taylor series expansion about the reference point $(\chi_r, \mathbf{u}_r)$, neglecting higher-order terms to yield

$$\dot{\chi} = f(\chi_r, \mathbf{u}_r) + \frac{\partial f(\chi, \mathbf{u})}{\partial \chi}(\chi - \chi_r) + \frac{\partial f(\chi, \mathbf{u})}{\partial \mathbf{u}}(\mathbf{u} - \mathbf{u}_r) \tag{4}$$

Combining (3) and (4), we can obtain

$$\dot{\tilde{\chi}} = A_t\tilde{\chi} + B_t\tilde{\mathbf{u}} \tag{5}$$

where $\dot{\tilde{\chi}} = \dot{\chi} - \dot{\chi}_r$, $\tilde{\chi} = \chi - \chi_r$, $\tilde{\mathbf{u}} = \mathbf{u} - \mathbf{u}_r$, and A_t, B_t are the Jacobian matrix and can be presented as follows:

$$A_t = \begin{bmatrix} \frac{\partial f_1}{\partial \chi_1} & \frac{\partial f_1}{\partial \chi_2} & \frac{\partial f_1}{\partial \chi_3} \\ \frac{\partial f_2}{\partial \chi_1} & \frac{\partial f_2}{\partial \chi_2} & \frac{\partial f_2}{\partial \chi_3} \\ \frac{\partial f_3}{\partial \chi_1} & \frac{\partial f_3}{\partial \chi_2} & \frac{\partial f_3}{\partial \chi_3} \end{bmatrix} = \begin{bmatrix} 0 & 0 & -v\sin\varphi \\ 0 & 0 & v\cos\varphi \\ 0 & 0 & 0 \end{bmatrix} \tag{6}$$

$$B_t = \begin{bmatrix} \frac{\partial f_1}{\partial u_1} & \frac{\partial f_1}{\partial u_2} \\ \frac{\partial f_2}{\partial u_1} & \frac{\partial f_2}{\partial u_2} \\ \frac{\partial f_3}{\partial u_1} & \frac{\partial f_3}{\partial u_2} \end{bmatrix} = \begin{bmatrix} \cos\varphi & 0 \\ \sin\varphi & 0 \\ \frac{\tan\delta}{L} & \frac{v}{L\cos^2\delta} \end{bmatrix} \tag{7}$$

The continuous-time state equation cannot be directly implemented in a model predictive controller. Using the first-order difference quotient method with sampling time T, the equation in (5) is discretized as:

$$\dot{\tilde{\chi}} = \frac{\tilde{\chi}(k+1) - \tilde{\chi}(k)}{T} = A_t\tilde{\chi}(k) + B_t\tilde{\mathbf{u}}(k) \tag{8}$$

By merging the similar terms, the equation in (8) can be simplified as a linear discrete error model as:

$$\tilde{\chi}(k+1) = A_{k,t}\tilde{\chi}(k) + B_{k,t}\tilde{\mathbf{u}} \tag{9}$$

where $A_{k,t} = A_tT + I_n$, $B_{k,t} = B_tT$, and I_n denotes the identity matrix.

3. Location Adaptive MPC for Path Tracking Framework

The PTC task is a multi-objective and multi-constrained optimization problem. The LTV-MPC method approximates a nonlinear time-varying (NTV) system to a linear time-varying (LTV) system, significantly improving the efficiency of path tracking applications. In the actual PTC, the LTV system can dynamically adjust the vehicle's parameters timely according to the current state. Such a system dramatically enhances real-time performance. However, suppose that the vehicle's current location drifts due to sensor noise or sensor abnormality, the control sequences generated by the LTV system will be wrong. This mistake may cause serious traffic accidents.

A LMPC path tracking framework is proposed to increase PTC task performance. The LMPC generates the correction when the vehicle's location is drifted and integrates the correction with the MPC approach in the PTC task. In this approach, a supervisor is first to predict the vehicle's drifts (drifted: 1; non-drift: 0) according to both vehicle's pose and position. Suppose the current location of the vehicle is drifted. In that case, a location estimation method, based on the fourth-order Runge–Kutta strategy, is applied for dynamic current location correction. The entire LMPC framework is illustrated in Figure 3. The details of this framework are introduced in the following subsections.

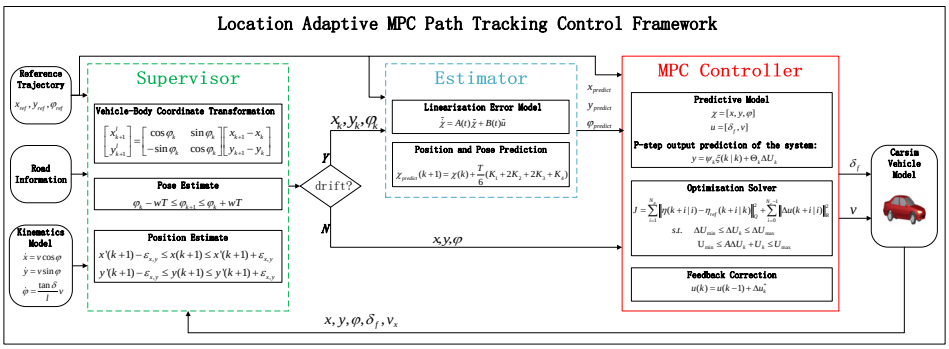


Figure 3. LMPC path tracking control framework.

3.1. Supervisor

In a real driving scenario, continuous position drifting is prone to occur due to various factors, e.g., sensor noise and sensor abnormality. Under these circumstances, the traditional MPC has difficulties correcting the vehicle's abnormal position and pose, dramatically degenerating the path tracking accuracy. Corresponding to this gap, we propose a supervisor to predict the drift of the vehicle. The module for location correction will be activated when the vehicle is predicted as being drifted.

The pose and position determine the vehicle's drift. The vehicle's location can be judged as not in drifted condition only when both these two aspects are regarded as usual. The details about the prediction of drifted location are introduced in the following subsubsections.

3.1.1. Judge from the Pose

When the vehicle's location is neither drifted nor considered being reversed, the position at time $k+1$ is in front of that at time k. Therefore, we can first identify the drift from the vehicle's direction difference between time k and $k+1$. The difference is computed by the position coordinates of a vehicle at time $k+1$ under the vehicle's body coordinate system at time k as:

$$\begin{bmatrix} x_{k+1}^l \\ y_{k+1}^l \end{bmatrix} = \begin{bmatrix} \cos\varphi_k & \sin\varphi_k \\ -\sin\varphi_k & \cos\varphi_k \end{bmatrix} \begin{bmatrix} x_{k+1} - x_k \\ y_{k+1} - y_k \end{bmatrix} \tag{10}$$

where x_{k+1}^l and y_{k+1}^l are the longitudinal and lateral coordinates under the time k's vehicle body coordinate system. It can be noted that the vehicle can be judged as not in front of the time k's position at time $k+1$ if

$$x_{k+1}^l < 0. \tag{11}$$

If the condition in (11) is satisfied, the vehicle can be judged as drifted directly; otherwise, more details about the heading direction need to be checked to predict whether the vehicle is drifted or not. It is known that the heading direction change at each time should be under the physical constraints of the vehicle, which can be presented as follows:

$$\varphi_k - (1+\gamma)w_k T \leq \varphi_{k+1} \leq \varphi_k + (1+\gamma)w_k T \tag{12}$$

where w is the angular velocity, which has been defined in (2) and computed as

$$w = \dot{\varphi} = \frac{\tan\delta_f}{l} v, \tag{13}$$

and $\gamma \in [0, 1]$ is a parameter to decide the sensitivity of drift decision of the system.

3.1.2. Judge from the Position

In the control process at time k, the controller generates the control value $\mathbf{u}(k)$ to track the reference path according to the current state $\chi(k)$. The vehicle's movement at each moment is affected by the speed, steering angle and the corresponding constraints of the vehicle. Suppose that the time k's position, speed and steering angle can be obtained. In that case, the position at the next time $k+1$ can be roughly predicted in a certain range before the current position, which is shown in Figure 4. Suppose the vehicle's position is out of that range at time $k+1$. The vehicle's position can be regarded as drift. The details about how to decide the region will be introduced as follows.

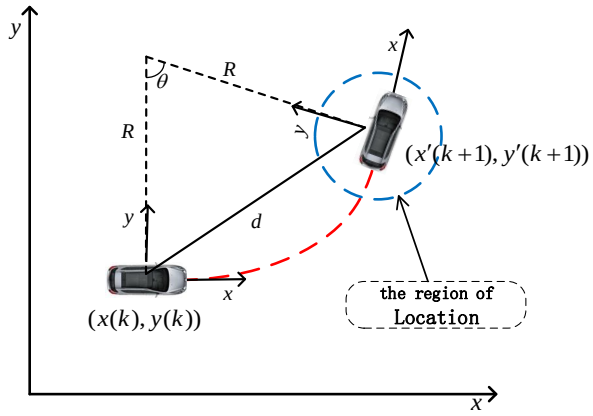


Figure 4. Schematic diagram of vehicle movement between two adjacent times.

Because the sampling interval between the two time points is very short, we can assume that the speed v is a constant. Combining this with the kinematics model of the vehicle in (13) and the front-wheel steering model in (14), the approximate distance of the vehicle between two adjacent times can be calculated. It is shown in Figure 4 that

$$R = \frac{l}{\tan \delta_f} \tag{14}$$

$$\theta = wT \tag{15}$$

According to the sine rule:

$$\frac{d}{\sin \theta} = \frac{R}{\sin(\frac{\pi-\theta}{2})}, \tag{16}$$

where d is the approximate distance of vehicle between two adjacent time points. Considering the equations from (14) to (16), the distance d can be calculated as:

$$d = \frac{l \sin(\frac{vT \tan(\delta_f)}{l})}{\tan(\delta_f) \cos(\frac{vT \tan(\delta_f)}{2l})} \tag{17}$$

By the small-angle approximation of the trigonometric function, the distance d in (17) can be approximated as:

$$d \approx vT. \tag{18}$$

Therefore, the roughly predicted position $(x'(k+1), y'(k+1))$ of the vehicle at time $k+1$ is:

$$\begin{aligned} x'(k+1) &= d * \cos(\varphi_k + wT/2) + x(k) \\ y'(k+1) &= d * \sin(\varphi_k + wT/2) + y(k), \end{aligned} \tag{19}$$

where $x'(k+1)$ and $y'(k+1)$ denote the vehicle's predicted longitudinal and lateral positions in the global inertial frame at time $k+1$, φ_k is the heading angle at time k, and $x(k)$, $y(k)$ are the corresponding longitudinal and lateral coordinates at time k. Without localization drift, the position at time $k+1$ lies within the approximate range calculated in the prior step k, as:

$$\begin{aligned} x'(k+1) - \varepsilon_{x,y} \le x(k+1) \le x'(k+1) + \varepsilon_{x,y} \\ y'(k+1) - \varepsilon_{x,y} \le y(k+1) \le y'(k+1) + \varepsilon_{x,y} \end{aligned} \tag{20}$$

where $\varepsilon_{x,y}$ is the local region parameter that is used to zoom the range of the approximate positioning area, and the specific value depends on the actual scene and application.

Combining conditional expression (11), (12) and (20), whether the location status of the vehicle is drifted or not can be judged at each moment in real time. If the location drifts, a correction strategy is implemented as:

$$\chi_{current} = \chi_{predict} \tag{21}$$

where $\chi_{current}$ is the current location state, and $\chi_{predict}$ is the predicted location state at the time $k+1$, which is obtained by the state at the previous time k.

3.1.3. Prediction of Vehicle Position

Based on the supervisor's state judgments and predictions, we adaptively correct the vehicle's position. To generate a high-precision single-step predicted state χ_{predict}, we use the fourth-order Runge–Kutta method to predict the vehicle's position, which is well-established in industrial applications for its high prediction accuracy [22]. It numerically solves differential equations with robust error suppression and exceptional single-step precision, so from the vehicle model in (5), we rederive the derivatives of state variables as:

$$\dot{\chi} = A\tilde{\chi} + B\tilde{\mathbf{u}} + \dot{\chi}_r \tag{22}$$

Then, Equation (22) is predicted using the fourth-order Runge–Kutta algorithm:

$$\chi_{predict}(k+1) = \chi(k) + \frac{T}{6}(K_1 + 2K_2 + 2K_3 + K_4), \tag{23}$$

where

$$\begin{aligned} K_1 &= A\tilde{\chi}(k) + B\tilde{\mathbf{u}}(k) + \dot{\chi}_r \\ K_2 &= A[\tilde{\chi}(k) + \tfrac{TK_1}{2}] + B\tilde{\mathbf{u}}(k) + \dot{\chi}_r \\ K_3 &= A[\tilde{\chi}(k) + \tfrac{TK_2}{2}] + B\tilde{\mathbf{u}}(k) + \dot{\chi}_r \\ K_4 &= A[\tilde{\chi}(k) + TK_3] + B\tilde{\mathbf{u}}(k) + \dot{\chi}_r \end{aligned}$$

According to equation (23), the state value at time $k+1$ can be precisely predicted by the state at time k. According to the judgment result of the supervisor at current time, the system will decide whether to activate the correction or not. If the location of the vehicle is judged as drifted, the predicted state will be used to replace the current drifted state, then MPC generates control values based on the predicted state values. Otherwise, the current state is directly input into the MPC controller to generate the control values.

3.2. MPC Basic Framework

In order to consider the comfort of the passengers, the equation in (22) needs to be transformed into the form of an incremental formulation by defining the state variable as

$$\xi(k|k) = \begin{bmatrix} \chi(k|k) \\ \mathbf{u}(k-1|k) \end{bmatrix}. \tag{24}$$

Let $\Delta\mathbf{u}(k)$ as the control increment, the expression of the state quantity and system output quantity can be reformulated as

$$\begin{aligned} \xi(k+1|k) &= \tilde{A}_k\xi(k|k) + \tilde{B}_k\Delta\mathbf{u}(k|k) \\ \eta &= \tilde{C}_k\xi(k|k), \end{aligned} \tag{25}$$

where $\tilde{A}_k = \begin{bmatrix} A_{k,t} & B_{k,t} \\ 0_n & I_n \end{bmatrix}$, $\tilde{B}_k = \begin{bmatrix} B_{k,t} \\ I_n \end{bmatrix}$, $\tilde{C}_k = \begin{bmatrix} C_{k,t} & 0_n \end{bmatrix}$.

Let N_p denote the prediction horizon and N_c the control horizon, with $N_c < N_p$ typically holds. We predict the states at each sampling step over the prediction horizon as:

$$\begin{aligned}
\xi(k+1|k) &= \tilde{A}_k\xi(k|k) + \tilde{B}_k\Delta\mathbf{u}(k|k)\\
\xi(k+2|k) &= \tilde{A}_k^2\xi(k|k) + \tilde{A}_k\tilde{B}_k\Delta\mathbf{u}(k|k) + \tilde{B}_k\Delta\mathbf{u}(k+1|k)\\
&\vdots\\
\xi(k+N_p|k) &= \tilde{A}_k^{N_p}\xi(k|k) + \tilde{A}_k^{N_p-1}\tilde{B}_k\Delta\mathbf{u}(k|k) + \cdots\\
&\quad + \tilde{A}_k^{N_p-N_c-1}\tilde{B}_k\Delta\mathbf{u}(k+N_c-1|k).
\end{aligned} \tag{26}$$

The system output at each sampling time in the prediction horizon can be can be predicted as

$$\begin{aligned}
\eta(k+1|k) &= \tilde{C}_k\tilde{A}_k\xi(k|k) + \tilde{C}_k\tilde{B}_k\Delta\mathbf{u}(k|k)\\
\eta(k+2|k) &= \tilde{C}_k\tilde{A}_k^2\xi(k|k) + \tilde{C}_k\tilde{A}_k\tilde{B}_k\Delta\mathbf{u}(k|k)\\
&\quad + \tilde{C}_k\tilde{B}_k\Delta\mathbf{u}(k+1|k)\\
&\vdots\\
\eta(k+N_p|k) &= \tilde{C}_k\tilde{A}_k^{N_p}\xi(k|k) + \tilde{C}_k\tilde{A}_k^{N_p-1}\tilde{B}_k\Delta\mathbf{u}(k|k) + \cdots\\
&\quad + \tilde{C}_k\tilde{A}_k^{N_p-N_c-1}\tilde{B}_k\Delta\mathbf{u}(k+N_c-1|k).
\end{aligned} \tag{27}$$

Let

$$\mathbf{y} = \begin{bmatrix} \eta(k+1|k)\\ \eta(k+2|k)\\ \vdots\\ \eta(k+N_p|k) \end{bmatrix} \tag{28}$$

$$\psi_k = \begin{bmatrix} \tilde{C}_k\tilde{A}_k^1\\ \tilde{C}_k\tilde{A}_k^2\\ \vdots\\ \tilde{C}_k\tilde{A}_k^{N_p} \end{bmatrix} \tag{29}$$

$$\Theta_k = \begin{bmatrix} \tilde{C}_k\tilde{B}_k & 0 & \cdots & 0\\ \tilde{C}_k\tilde{A}_k\tilde{B}_k & \tilde{C}_k\tilde{B}_k & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ \tilde{C}_k\tilde{A}_k^{N_c-1}\tilde{B}_k & \tilde{C}_k\tilde{A}_k^{N_c-2}\tilde{B}_k & \cdots & \tilde{C}_k\tilde{B}_k\\ \tilde{C}_k\tilde{A}_k^{N_c}\tilde{B}_k & \tilde{C}_k\tilde{A}_k^{N_c-1}\tilde{B}_k & \cdots & \tilde{C}_k\tilde{A}_k\tilde{B}_k\\ \vdots & \vdots & \ddots & \vdots\\ \tilde{C}_k\tilde{A}_k^{N_p-1}\tilde{B}_k & \tilde{C}_k\tilde{A}_k^{N_p-2}\tilde{B}_k & \cdots & \tilde{C}_k\tilde{A}_k^{N_p-N_c}\tilde{B}_k \end{bmatrix} \tag{30}$$

$$\Delta U_k = \begin{bmatrix} \Delta\mathbf{u}(k|k)\\ \Delta\mathbf{u}(k+1|k)\\ \vdots\\ \Delta\mathbf{u}(k+N_c-1|k) \end{bmatrix}. \tag{31}$$

The system outputs across the entire prediction time domain can be expressed in a more compact form as:

$$\mathbf{y} = \psi_k\xi(k|k) + \Theta_k\Delta U_k. \tag{32}$$

https://doi.org/10.3390/sym18020307

3.3. Objective Function

The reference path of the entire prediction horizon is

$$\mathbf{y}_{ref} = \begin{bmatrix} \eta_{ref}(k+1|k) \\ \eta_{ref}(k+2|k) \\ \vdots \\ \eta_{ref}(k+N_p|k) \end{bmatrix}. \tag{33}$$

MPC aims to generate the vehicle's control sequence over the control horizon for high-precision reference path tracking, expressed as the following objective function:

$$\begin{aligned} J = &\sum_{i=1}^{N_p} \left\| \eta(k+i|i) - \eta_{ref}(k+i|i) \right\|_Q^2 \\ &+ \sum_{i=0}^{N_c-1} \|\Delta\mathbf{u}(k+i|i)\|_R^2 + \rho\varepsilon^2, \end{aligned} \tag{34}$$

where ε is the relaxation factor ensuring a feasible solution [23], ρ is a relaxation parameter, Q is the state error weight matrix, and R is the control increment weight matrix. Specifically, the first term in (34) characterizes path tracking performance: the smaller its value, the better the tracking effect. The second term reflects the smoothness of vehicle control, with a smaller value indicating more stable tracking. The third term guarantees an optimal feasible solution. To facilitate computation using existing algorithms, the objective function in (34) is always converted into a standard quadratic form as:

$$\begin{aligned} J(\xi(k), u(k-1), \Delta U(k)) = &[\Delta U(k); \varepsilon]^T H [\Delta U(k); \varepsilon] \\ &+ f[\Delta U(k); \varepsilon], \end{aligned} \tag{35}$$

where $H = \begin{bmatrix} \Theta_k^T Q \Theta_k + R & 0 \\ 0 & \rho \end{bmatrix}$, $f = \begin{bmatrix} 2E^T Q \Theta_k \\ 0 \end{bmatrix}$, and $E = \psi_k \xi(k|k) - \mathbf{y}_{ref}(k)$ represents the deviation of the output in the prediction time domain.

3.4. Constraints of the Problem

In practical control implementation, an autonomous vehicle must satisfy certain operational constraints, which imposes corresponding limits on the system's control variables. In this work, only box constraints on the control variables and their increments are considered, given as:

$$\begin{aligned} \mathbf{u}_{\min} \leq &\mathbf{u}(k+i|k) \leq \mathbf{u}_{\max} \ \forall i = 0, \ldots, N_c - 1. \\ \Delta\mathbf{u}_{\min} \leq &\Delta\mathbf{u}(k+i|k) \leq \Delta\mathbf{u}_{\max} \ \forall i = 0, \ldots, N_c - 1 \end{aligned} \tag{36}$$

Since the objective function in (35) only corresponds to the control increments, the constraints on the control variables need to be reformulated in terms of the control increments, based on the fact that

$$\mathbf{u}(k+i|k) = \mathbf{u}(k+i-1|k) + \Delta\mathbf{u}(k+i|k) \tag{37}$$

Let

$$U_k = 1_{N_c} \otimes \mathbf{u}(k-1), \tag{38}$$

$$A = \underbrace{\begin{bmatrix} 1 & 0 & \cdots & \cdots & 0 \\ 1 & 1 & 0 & \cdots & 0 \\ 1 & 1 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 1 & 1 & \cdots & 1 & 1 \end{bmatrix}}_{N_c \times N_c} \otimes I_m, \tag{39}$$

where $\otimes$ is the Kronecker product, and U_k is the control variable in the whole control horizon. Combining with equations from (37) to (39), the box constraints on control variables can be reformulated by control increments as

$$U_{\min} \leq A\Delta U_k + U_k \leq U_{\max}, \tag{40}$$

where $U_{\min}$ and $U_{\max}$ represent the lower and upper bounds of the control variable over the control horizon.

3.5. Optimization Problem

Combining the objective function in (35) with the constraints mentioned above, the optimization problem of MPC can be formulated as

$$\begin{aligned} &\min_{\Delta u_k, \varepsilon} = [\Delta U(k); \varepsilon]^T H[\Delta U(k); \varepsilon] + f[\Delta U(k); \varepsilon] \\ &\text{s.t. } \Delta U_{\min} \leq \Delta U_k \leq \Delta U_{\max} \\ &\quad U_{\min} \leq A\Delta U_k + U_k \leq U_{\max}. \end{aligned} \tag{41}$$

The problem in (41) is a standard quadratic programming (QP) problem, which can be solved efficiently via the interior point method (IPM). In each control cycle, we thus obtain a set of optimal control increments denoted ΔU_k^* as:

$$\Delta U_k^* = \left[\Delta \mathbf{u}_k^*, \cdots, \Delta \mathbf{u}_{k+N_c-1}^*\right]^T. \tag{42}$$

Finally, at each time, only the first item in ΔU_k^* is taken to generate the actual control input

$$\mathbf{u}^*(k) = \mathbf{u}(k-1) + \Delta \mathbf{u}_k^*, \tag{43}$$

until the new time is re-predicted [24].

4. Simulation Experiment Results and Analysis

We validated the effectiveness of the proposed LMPC approach through comparative experiments against a standard MPC controller (used as the baseline [23]). We built a co-simulation platform with CarSim 9.0.3 (vehicle simulation software) and MATLAB/Simulink R2021b, where a speed closed-loop controller built into CarSim maintains constant longitudinal vehicle speed. To comprehensively evaluate performance, we tested the two controllers under two distinct scenarios: a double-shifting curve (representing a standard operating condition) and a sine curve (representing a more challenging condition). Both the conventional MPC baseline and the proposed LMPC are implemented strictly in accordance with the formulations detailed in Sections 3.2–3.4, with their simulation parameters consistent with those listed in Table 2. Constraints on the heading angle (rad), velocity variation (m/s) and heading angle variation (rad) are set to the intervals $\left[-\frac{5}{36}\pi, \frac{5}{36}\pi\right]$, $[-0.05, 0.05]$ and $\left[-\frac{6.47}{180}\pi, \frac{6.47}{180}\pi\right]$, respectively.

Table 2. Parameters used in the simulation.

Symbol	Interpretation	Value (Unit)
N_p	Predictive horizon	30
N_c	Control horizon	15
T	Model discretized step	0.1 (s)
Q	State weight matrix	diag([10, 10, 10])
R	Control weight matrix	diag([1, 1])
ρ	Relaxation penalty	0.01
$\varepsilon_{x,y}$	Distance relaxation factor	0.35 (m)
φ	Heading angle	$-\frac{25}{180}\pi \sim \frac{25}{180}\pi$ (rad)
$\triangle v$	Velocity variation	$-0.05 \sim 0.05$ (m/s)
$\triangle \varphi$	Heading angle variation	$-\frac{6.47}{180}\pi \sim \frac{6.47}{180}\pi$ (rad)

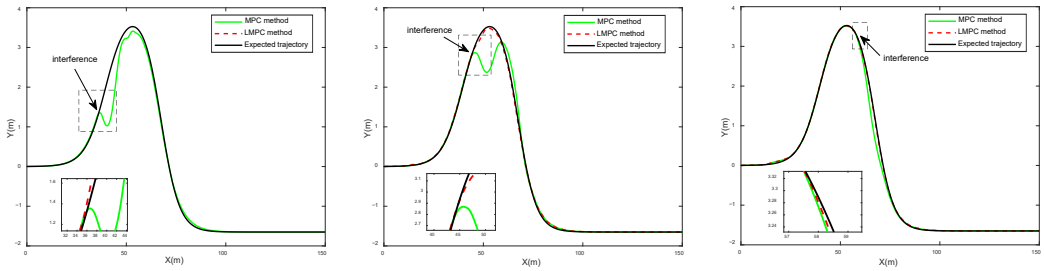
The selection of MPC weights (Q, R) and relaxation penalty (ρ) balances path tracking performance, control smoothness, and optimization feasibility. Specifically, $Q = \text{diag}([10, 10, 10])$ assigns equal high weights to longitudinal/lateral positions (x, y) and heading angle (φ), as these states are equally critical for accurate path following, with excessive deviation in any degrading precision or stability. $R = \text{diag}([1, 1])$ uses small weights to balance tracking accuracy and control smoothness: it suppresses abrupt changes in speed (v) and front wheel steering angle (δ_f) to avoid actuator saturation and ensure comfort, without overly restricting inputs. ρ enhances quadratic programming (QP) feasibility: under location drift or external disturbances, strict constraints may yield no feasible solution, and the relaxation factor ε (weighted by ρ) provides a flexible margin to resolve conflicts, with a small ρ minimizing compromise to tracking accuracy. Collectively, Q prioritizes precision, R guarantees smoothness, and ρ ensures feasibility, which is critical for stable performance under targeted location drift conditions.

4.1. Simulation on Double-Shifting Curve

During the vehicle control process, a significant deviation arises between the observed location and the ground truth when the vehicle experiences location drift. Specifically, to verify the path tracking performance of the proposed LMPC method under location drift scenarios, we introduced a constant disturbance to the vehicle state from 5 s to 6 s in the simulation, where $\chi(k) = \chi(k) + 10$ (consistent offset). Notably, there is a discrepancy between the constant disturbance adopted in the simulation and practical scenarios—disturbances in real-world settings are often time-varying or random. Nevertheless, this discrepancy does not compromise the validity of the proposed method, as its core advantage lies in addressing large-magnitude position deviations (the essence of location drift) rather than being restricted to specific disturbance forms. Additionally, due to the inherent limitation of the kinematic model (which cannot accurately capture complex nonlinear dynamic characteristics at high speeds), the simulations were conducted under different medium and low velocities, i.e., 30 km/h, 40 km/h, and 50 km/h.

The path tracking simulation results of the two controllers under different velocities are shown in Figure 5. The position where interference is added is marked in the figure. It can be noted that a significant tracking error is generated with the involvement of the interference using the conventional MPC. Due to the addition of the interference started at the point marked on the figure, the vehicle's current location is drifted and incorrect. The conventional MPC could not detect this and generated a wrong control sequence for the vehicle. Alternatively, the proposed LMPC method alleviates the adverse effects of the interference. It generates more precious path tracking results than the conventional MPC method. This phenomenon may be because the proposed supervisor can efficiently detect the drifted condition of the vehicle, or the location correction strategy based on the

Runge–Kutta method may generate precise location prediction. In this case, the MPC still outputs the control sequences with a high path tracking accuracy. From Figure 5, it can also be noted that when the location condition of the vehicle is back to normal, both methods have good performance in tracking the reference path.



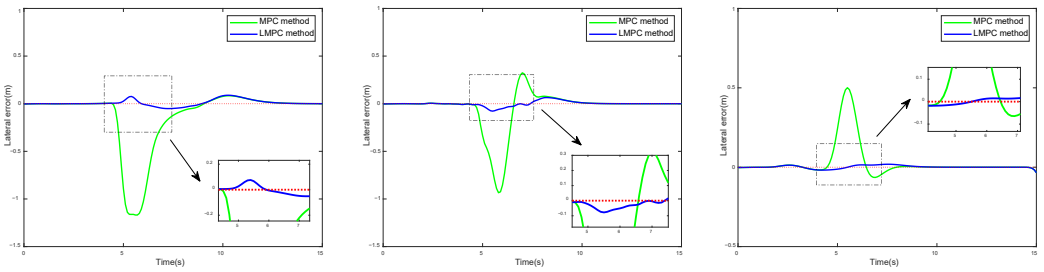
(a) 30 km/h (b) 40 km/h (c) 50 km/h

Figure 5. The path tracking performance in double-shifting curve with location drifted scenario under the different velocities.

Variations in the lateral tracking error of two controllers under different velocities are shown in Figure 6. The figure intuitively shows the effectiveness of the proposed LMPC approach. Under different velocities, both MPC and LMPC obtain acceptable lateral tracking errors before the interference is added. However, in the scenario of the drifted location by adding interference, the MPC generates a more significant lateral error, reaching 1.2 m deviations when velocity is 30 km/h. On the contrary, although LMPC is also affected by the drift, the maximum lateral error does not exceed 0.1 m, which is still within an acceptable range. The average lateral tracking errors between 5 s and 6 s of two controllers under different velocities are illustrated in Table 3. The table shows that the average errors generated by the LMPC method are much lower than the conventional MPC method in the double-shifting curve. The total average of the lateral tracking error of the conventional MPC method reaches 1.2944 m, while the LMPC method only generates 0.0429 m of lateral tracking error.

Table 3. The average lateral path tracking error (m) of double-shifting curve between 5–6 s under different velocities.

Methods	30 km/h	40 km/h	50 km/h	Total Average
MPC method	1.3283	0.5808	1.9742	1.2944
LMPC mehtod	0.0303	0.0825	0.0159	0.0429

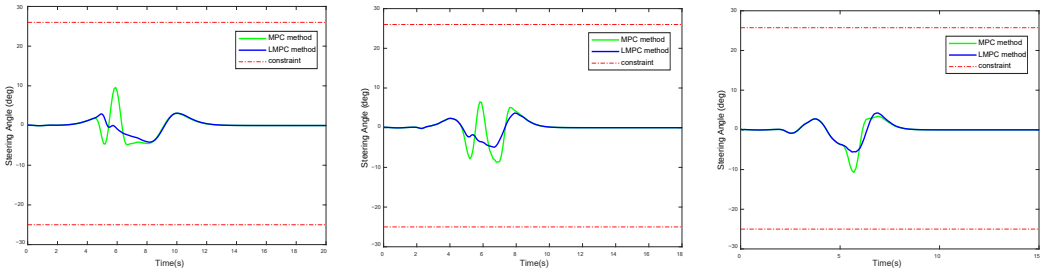


(a) 30 km/h (b) 40 km/h (c) 50 km/h

Figure 6. The lateral tracking errors in double-shifting curve with location drifted scenario under different velocities. The red dashed line denotes the zero-error value.

To verify that the proposed path control method satisfies steering angle constraints, the steering angle variations over time at different velocities are presented in Figure 7. The results show that both controllers satisfy the steering angle constraints irrespective of vehicle drift. Under drift conditions, the steering angle generated by the conventional MPC

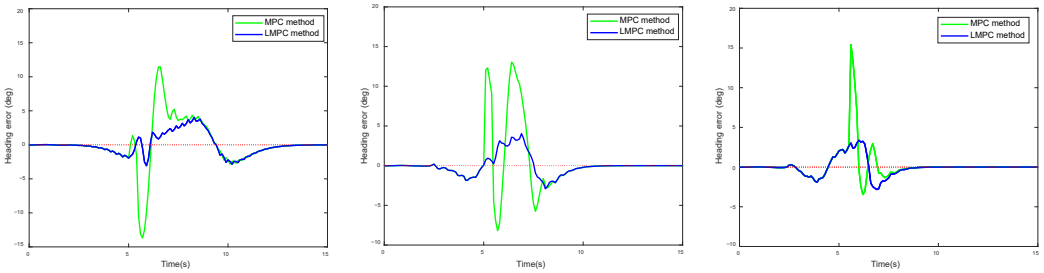
controller exhibits significant fluctuations; in contrast, that of the LMPC controller shows no apparent fluctuations, thus ensuring the stability and smoothness of path tracking.



(a) 30 km/h (b) 40 km/h (c) 50 km/h

Figure 7. The variations in steering angle at different velocities under the double-shifting curve.

Figure 8 presents the heading tracking error variations in conventional MPC and LMPC over time at 30 km/h, 40 km/h, and 50 km/h. Notably, the heading tracking errors of both controllers are identical when no localization drift occurs. Under drift conditions, the errors generated by conventional MPC increase dramatically: their maximum exceeds 13 degrees across all three velocities, and even reaches 15 degrees at 50 km/h. In contrast, the proposed LMPC controller mitigates the adverse effects of drift, producing significantly smaller heading tracking errors than conventional MPC. Specifically, the maximum heading tracking error of LMPC does not exceed 4 degrees under these three velocities, demonstrating that it achieves more stable path tracking performance.



(a) 30 km/h (b) 40 km/h (c) 50 km/h

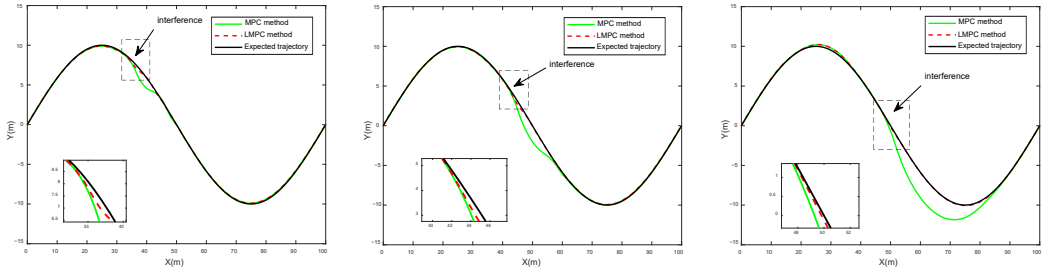
Figure 8. The heading tracking errors in double-shifting curve with location drifted scenario under different velocities. The red dashed line denotes the zero-error value.

4.2. Simulation on Sine Curve

The sine curve is a more complex path tracking scenario than the double shifting curve. A controller with a greater path tracking ability is required to maintain a good path tracking performance in the sine curve. The simulation parameters used in the sine curve are consistent with those used in the double shifting. Figure 9 shows the path tracking results of the MPC and LMPC under different velocities of 30 km/h, 40 km/h and 50 km/h. Figure 9 demonstrates that the path tracking the performance of MPC degrades significantly once the location drifting occurs. The higher velocity that the vehicle has, the MPC generates the worse tracking results. Conversely, the proposed LMPC method still generates acceptable tracking performance after adding interference.

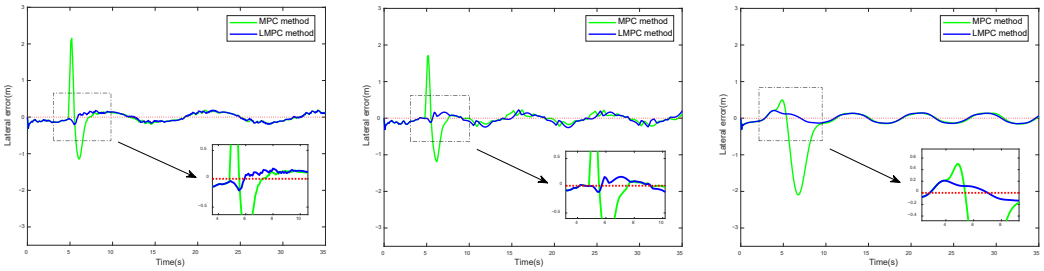
The lateral errors under different velocities in the sine curve path tracking simulation are shown in Figure 10. The figure shows that the path tracking performance of the MPC method degrades significantly with interference involvement. After adding the interference, a 2 m lateral error even occurs under both 30 km/h and 50 km/h conditions. On the contrary, the proposed LMPC method decreases the negative effects of interference and generates acceptable lateral tracking error. The average lateral tracking errors between 5 s and 6 s of two controllers under different velocities are illustrated in Table 3. The table

shows that the average errors generated by the LMPC method are much lower than the conventional MPC method in sine curve. The total average of the lateral tracking error of the conventional MPC method reaches 1.9378 m, while the LMPC method only generates 0.1132 m of lateral tracking error.



(a) 30 km/h (b) 40 km/h (c) 50 km/h

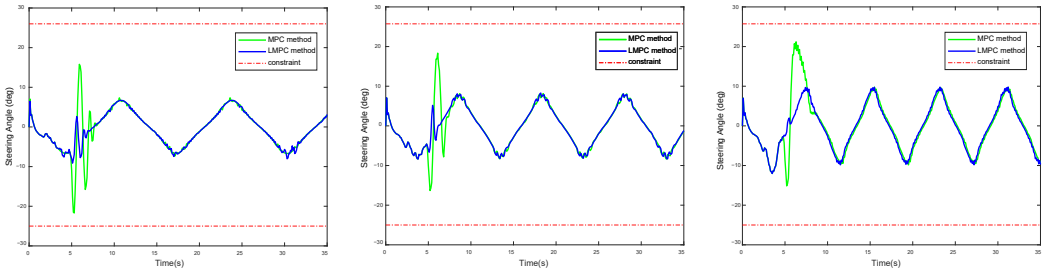
Figure 9. The path tracking performance in sine curve condition with location drifted scenario under the different velocities.



(a) 30 km/h (b) 40 km/h (c) 50 km/h

Figure 10. The lateral tracking errors in sine curve with location drifted scenario under different velocities. The red dashed line denotes the zero-error value.

In Figure 11, we observe that the steering angles of both two MPC-based controllers remain within the constraint range in both paths. Furthermore, both controllers are affected by the drift of the vehicle. However, it can be noted that the steering angle generated by the conventional MPC controller fluctuated more frequently with larger magnitudes than the LMPC controller when the vehicle is drifted. The reason may be that in the LMPC framework, the supervisor can detect the drift of the vehicle accurately, and the estimator can generate the current location prediction precisely.

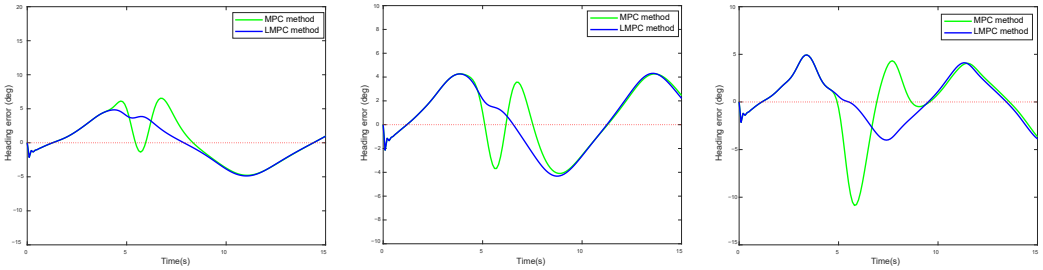


(a) 30 km/h (b) 40 km/h (c) 50 km/h

Figure 11. The variations in steering angle at different velocities.

Figure 12 shows the variations in the heading tracking error of two controllers under different velocities. From the figures, it is shown that the proposed LMPC controller generates smaller heading tracking errors than the conventional MPC controller even the vehicle is drifted. The maximum of the heading tracking error of LMPC under different velocities does not exceed 5 degrees. On the contrary, the conventional MPC generates larger heading errors, which exceeds 10 degrees when velocity is 50 km/h after the vehicle is drifted. Mean-

while, compared with the conventional MPC controller, the heading error curve generated by the LMPC is smoother, which ensures a more stable path tracking performance.



(a) 30 km/h (b) 40 km/h (c) 50 km/h

Figure 12. The heading tracking errors in sine curve with location drifted scenario under different velocities. The red dashed line denotes the zero-error value.

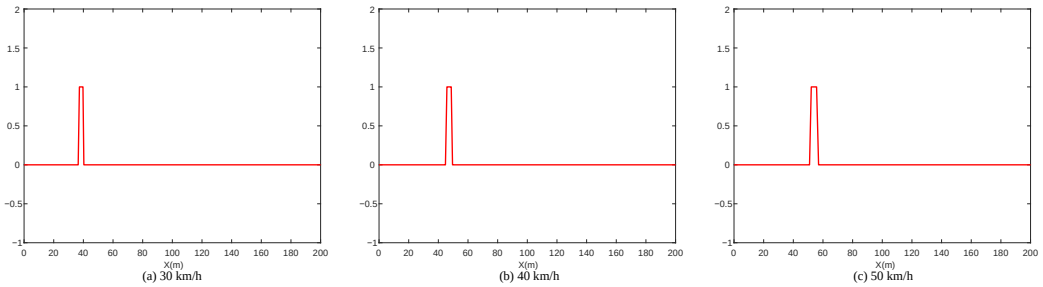
Tables 3 and 4 present that across all velocities and both path scenarios, the MPC method shows notably higher average lateral errors: its total average errors reach 1.2944 m (Table 3) and 1.9378 m (Table 4). In sharp contrast, the LMPC method achieves much smaller errors, with total average values of only 0.0429 m (Table 3) and 0.1132 m (Table 4). These results clearly indicate that LMPC outperforms the conventional MPC method in path tracking accuracy under position drift conditions.

Table 4. The average lateral path tracking error (m) of sine curve between 5–6 s under different velocities.

Methods	30 km/h	40 km/h	50 km/h	Total Average
MPC method	2.3486	1.7342	1.7095	1.9378
LMPC mehtod	0.0557	0.0938	0.1903	0.1132

4.3. Location Drifting Detection

In order to show the effectiveness of the proposed location drifting supervisor, some simulations are conducted to illustrate the supervisor's location drifting detection results. The supervisor output the value one when the location drifting is detected. Otherwise, output the value of zero. The detection results for all path tracking scenarios are illustrated in Figures 13 and 14. Figure 13 shows the sine curve's location-shifting detection results, and Figure 14 shows the results in the double-shifting curve. Combining with the results in Figures 5 and 9, the location shifting detection results show that the supervisor can preciously detect the location shifting under different velocities when it occurs.



(a) 30 km/h (b) 40 km/h (c) 50 km/h

Figure 13. The location shifting detection results in sine curve under different velocities.

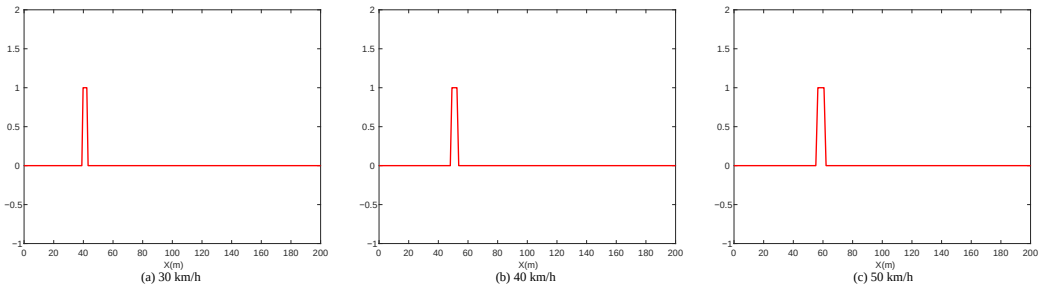


Figure 14. The location shifting detection results in double-shifting curve under different velocities.

5. Conclusions and Future Work

This paper proposes a novel location-adaptive MPC (LMPC) path tracking approach to address autonomous vehicle location drift caused by sensor noise or failure, thus improving path tracking performance. A supervisor based on vehicle pose and position is designed to detect drift conditions, and a fourth-order Runge–Kutta-based state estimator generates current state predictions under drift. Validated via CarSim/Matlab/Simulink co-simulations, the proposed LMPC is compared with conventional MPC under various velocities and scenarios, and the results demonstrate its superior tracking accuracy under drift: in the double-shifting curve scenario, conventional MPC's total lateral tracking error is 1.2944 m versus 0.0429 m for LMPC, while in the sine curve scenario, the errors are 1.9378 m and 0.1132 m, respectively.

The adopted kinematic model (assuming $\beta = 0$ and neglecting vertical movements) is applicable to medium-low speed scenarios (30–50 km/h), where lateral tire slip and dynamic forces (e.g., tire cornering stiffness nonlinearity) are negligible, but it has inherent limitations at speeds > 60 km/h due to significant slip and non-negligible dynamic effects. Future work will integrate a dynamic model (e.g., Pacejka tire model) to enable speed-based adaptive switching between kinematic and dynamic models; additionally, correntropy, derived from information-theoretic learning [25] and effective in addressing non-Gaussian noise [26,27], will be leveraged as the maximum correntropy criterion (MCC) loss function to mitigate location drift, along with high-level feature fusion strategies [28–30] to enhance trajectory tracking performance.

Author Contributions: J.X.: writing—original draft, resources; X.X.: writing—original draft, methodology, visualization, investigation; X.H.: conceptualization, methodology, visualization, writing—review and editing; Y.W.: supervision, writing—review and editing; N.Z. and Y.Y.: writing—review and editing, supervision. J.X. and X.X. contributed equally to this work. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Sichuan Science and Technology Program under Grant No. 2025ZNSFSC1483.

Data Availability Statement: The data supporting the findings of this study are available from the corresponding author upon reasonable request via email.

Conflicts of Interest: The authors declare there are no conflicts of interest.

References

1. Skakauskas, P.; Karpenko, M.; Prentkovskis, O. Design and Implementation of a Hybrid Path Planning Approach for Autonomous Lane Change Manoeuvre. *Int. J. Automot. Technol.* **2024**, *25*, 83–95. [CrossRef]
2. Zhang, C.; Hu, J.; Qiu, J.; Yang, W.; Sun, H.; Chen, Q. A novel fuzzy observer-based steering control approach for path tracking in autonomous vehicles. *IEEE Trans. Fuzzy Syst.* **2018**, *27*, 278–290. [CrossRef]
3. Choi, M.; Choi, S.B. MPC for vehicle lateral stability via differential braking and active front steering considering practical aspects. *Proc. Inst. Mech. Eng. Part D J. Automob. Eng.* **2016**, *230*, 459–469. [CrossRef]

4. Wang, S.; Liu, S.; Wang, G.; Ma, C.; Guo, J.; Zhao, H. Research on Steering Path Tracking Performance of Articulated Quad-Tracked Vehicle Based on Fuzzy PID Control. *J. Field Robot.* **2025**, 42, 3561–3581 . [CrossRef]
5. Yin, Y.; Xu, Z.; Wei, Y.; Jiang, W.; Yao, D.; Dong, Z. Model-free sliding mode control for path following in autonomous vehicle. *J. Frankl. Inst.* **2025**, 362, 107806. [CrossRef]
6. Guo, J.; Luo, Y.; Li, K. An adaptive hierarchical trajectory following control approach of autonomous four-wheel independent drive electric vehicles. *IEEE Trans. Intell. Transp. Syst.* **2017**, *19*, 2482–2492. [CrossRef]
7. Rahim, A.; Rokonuzzaman, M.; Alqumsan, A.A.; Arogbonlo, A. Polytopic LPV-LQR Lateral Controller for Varying Speeds AGVs Exploiting Uncertainty. In Proceedings of the 2025 IEEE International Conference on Mechatronics (ICM), Wollongong, Australia, 28 February–2 March 2025; pp. 1–7.
8. Soudbakhsh, D.; Eskandarian, A. *A Collision Avoidance Steering Controller Using Linear Quadratic Regulator*; Technical Report; SAE Technical Paper: Warrendale, PA, USA, 2010.
9. Morales, S.; Magallanes, J.; Delgado, C.; Canahuire, R. LQR trajectory tracking control of an omnidirectional wheeled mobile robot. In Proceedings of the 2018 IEEE 2nd Colombian Conference on Robotics and Automation (CCRA), Barranquilla, Colombia, 1–3 November 2018; pp. 1–5.
10. Lal, D.S.; Vivek, A.; Selvaraj, G. Lateral control of an autonomous vehicle based on Pure Pursuit algorithm. In Proceedings of the 2017 International Conference on Technological Advancements in Power and Energy (TAP Energy), Kollam, India, 21–23 December 2017; pp. 1–8.
11. Han, Z.; Chen, P.; Zhou, B.; Yu, G. Hybrid path tracking control for autonomous trucks: Integrating pure pursuit and deep reinforcement learning with adaptive look-ahead mechanism. *IEEE Trans. Intell. Transp. Syst.* **2025**, *26*, 7098–7112. [CrossRef]
12. Zhao, H.; Ren, B.; Chen, H.; Deng, W. Model predictive control allocation for stability improvement of four-wheel drive electric vehicles in critical driving condition. *IET Control Theory Appl.* **2015**, *9*, 2688–2696. [CrossRef]
13. Wang, X.; Hu, J.; Ni, J. MPC-based trajectory tracking via active front steering and external yaw moment. In Proceedings of the Proc. ICVMEE, Taiyuan, China, 24–25 July 2017; pp. 1–9.
14. Velhal, S.; Thomas, S. Improved LTVMPC design for steering control of autonomous vehicle. *Proc. J. Phys. Conf. Ser.* **2017**, *783*, 012028. [CrossRef]
15. Bai, G.; Liu, L.; Meng, Y.; Luo, W.; Gu, Q.; Ma, B. Path tracking of mining vehicles based on nonlinear model predictive control. *Appl. Sci.* **2019**, *9*, 1372. [CrossRef]
16. Wu, H.; Si, Z.; Li, Z. Trajectory tracking control for four-wheel independent drive intelligent vehicle based on model predictive control. *IEEE Access* **2020**, *8*, 73071–73081. [CrossRef]
17. Dai, C.; Zong, C.; Chen, G. Path tracking control based on model predictive control with adaptive preview characteristics and speed-assisted constraint. *IEEE Access* **2020**, *8*, 184697–184709. [CrossRef]
18. Zhao, Z.; Xu, J.; Lei, Y.; Liu, C.; Shi, X.; Lai, L.L. Robust dynamic dispatch strategy for multi-uncertainties integrated energy microgrids based on enhanced hierarchical model predictive control. *Appl. Energy* **2025**, *381*, 125141. [CrossRef]
19. Xiang, C.; Peng, H.; Wang, W.; Li, L.; An, Q.; Cheng, S. Path tracking coordinated control strategy for autonomous four in-wheel-motor independent-drive vehicles with consideration of lateral stability. *Proc. Inst. Mech. Eng. Part D J. Automob. Eng.* **2021**, *235*, 1023–1036. [CrossRef]
20. Nikou, A.; Verginis, C.K.; Heshmati-alamdari, S.; Dimarogonas, D.V. A robust non-linear MPC framework for control of underwater vehicle manipulator systems under high-level tasks. *IET Control Theory Appl.* **2021**, *15*, 323–337. [CrossRef]
21. Zhang, C.; Chu, D.; Liu, S.; Deng, Z.; Wu, C.; Su, X. Trajectory planning and tracking for autonomous vehicle based on state lattice and model predictive control. *IEEE Intell. Transp. Syst. Mag.* **2019**, *11*, 29–40. [CrossRef]
22. Bates, D.J.; Hauenstein, J.D.; Sommese, A.J. Efficient path tracking methods. *Numer. Algorithms* **2011**, *58*, 451–459. [CrossRef]
23. Jian-Wei, G.; Wei, X.; Yan, J.; Kai, L.; Hong-fen, G.; Yin-jian, S. Multi-constrained model predictive control for autonomous ground vehicle trajectory tracking. *J. Beijing Inst. Technol.* **2015**, *24*, 441–448.
24. Camacho, E.F.; Alba, C.B. *Model Predictive Control*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2013.
25. Tang, H.; Li, H. Information Theoretic Learning: Reny's Entropy and Kernel Perspectives (Principe, J.; 2010) [Book Review]. *IEEE Comput. Intell. Mag.* **2011**, *6*, 60–62. [CrossRef]
26. Zhou, N.; Deng, Q.; Luo, W.; Huang, X.; Du, Y.; Chen, B.; Pedrycz, W. Correntropy meets cross-entropy: A robust loss against noisy labels. *Eng. Appl. Artif. Intell.* **2026**, *167*, 113830. [CrossRef]
27. Zhou, N.; Luo, W.; Wu, Z.; Du, Y.; Shi, K.; Chen, B. Bipartite graph regularized robust low-rank matrix factorization for fast semi-supervised image clustering. *Appl. Intell.* **2026**, *56*, 22. [CrossRef]
28. Xie, Y.; Ou, J.; Wen, B.; Yu, Z.; Tian, W. A joint learning method for low-light facial expression recognition. *Complex Intell. Syst.* **2025**, *11*, 139. [CrossRef]

29. Zhou, N.; Jika, L.; Wang, W.; Xie, Y.; Du, Y.; Soh, Y.C. BERN: A Novel Framework for Enhanced Emotion Recognition Through the Integration of EEG and Eye Movement Features. *IEEE Trans. Cogn. Dev. Syst.* **2025**, *18*, 263–275. [CrossRef]
30. Yu, Y.; She, K.; Shi, K.; Cai, X.; Kwon, O.M.; Soh, Y. Analysis of medical images super-resolution via a wavelet pyramid recursive neural network constrained by wavelet energy entropy. *Neural Netw.* **2024**, *178*, 106460. [CrossRef] [PubMed]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.