

Article

A Vision Navigation Method for Agricultural Machines Based on a Combination of an Improved MPC Algorithm and SMC

Yuting Zhai ¹, Dongyan Huang ², Jian Li ¹, Xuehai Wang ¹ and Yanlei Xu ^{2,*} 

¹ School of Information Technology, Jilin Agricultural University, Changchun 130118, China; zhaiyuting@mails.jlau.edu.cn (Y.Z.)

² School of Engineering Technology, Jilin Agricultural University, Changchun 130118, China

* Correspondence: yanleixu@jlau.edu.cn

Abstract

Vision navigation systems provide significant advantages in agricultural scenarios such as pesticide spraying, weeding, and harvesting by interpreting crop row structures in real-time to establish guidance lines. However, the delay introduced by image processing causes the path and pose information relied upon by the controller to lag behind the actual vehicle state. In this study, a hierarchical delay-compensated cooperative control framework (HDC-CC) was designed to synergize Model Predictive Control (MPC) and Sliding Mode Control (SMC), combining predictive optimization with robust stability enforcement for agricultural navigation. An upper-layer MPC module incorporated a novel delay state observer that compensated for visual latency by forward-predicting vehicle states using a 3-DoF dynamics model, generating optimized front-wheel steering angles under actuator constraints. Concurrently, a lower-layer SMC module ensured dynamic stability by computing additional yaw moments via adaptive sliding surfaces, with torque distribution optimized through quadratic programming. Under varying adhesion conditions tests demonstrated error reductions of 74.72% on high-adhesion road and 56.19% on low-adhesion surfaces. In Gazebo simulations of unstructured farmland environments, the proposed framework achieved an average path tracking error of only 0.091 m. The approach effectively overcame vision-controller mismatches through predictive compensation and hierarchical coordination, providing a robust solution for vision autonomous agricultural machinery navigation in various row-crop operations.

Keywords: agricultural tractors; vision autonomous guidance; model predictive control; path tracking



Academic Editor: Maciej Zaborowicz

Received: 20 September 2025

Revised: 14 October 2025

Accepted: 21 October 2025

Published: 22 October 2025

Citation: Zhai, Y.; Huang, D.; Li, J.; Wang, X.; Xu, Y. A Vision Navigation Method for Agricultural Machines Based on a Combination of an Improved MPC Algorithm and SMC. *Agriculture* **2025**, *15*, 2189. <https://doi.org/10.3390/agriculture15212189>

Copyright: © 2025 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 (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In agricultural pest control operations, unmanned agricultural machinery has emerged as a pivotal tool for enhancing production efficiency and reducing labor costs, primarily due to its advanced autonomous navigation technology. It is extensively utilized in various agricultural tasks, including weeding, pesticide application, crop monitoring, and harvesting [1–3]. Currently, mainstream navigation systems predominantly employ a fusion positioning approach that integrates the Global Navigation Satellite System (GNSS) with an Inertial Measurement Unit (IMU) [4]. However, the intricate nature of field environments often leads to weak GNSS signals, while the cumulative error associated with the IMU can substantially increase during prolonged operations. When these two systems fail to coordinate effectively, it may result in trajectory deviations or even operational disruptions [5].

Vision-based navigation technology exhibits superior environmental adaptability through real-time analysis of crop row line features, rendering it particularly suitable for operational scenarios characterized by severe signal obstruction [6,7]. However, the temporal matching issue between dynamic perception and motion control in visual systems can significantly impact path tracking accuracy, as it is susceptible to mechanical motion status and image processing delays. Unlike autonomous passenger vehicles that travel on regular roads, agricultural operations need extremely high precision when conducted between crop rows to avoid damaging precious crops during close-range operations. Agricultural vehicles operate at a lower speed and with significant variations, which means that a perception delay of just a few hundred milliseconds could lead to significant and corrective spatial deviations. Therefore, compensating for the visual delay is a necessary condition for achieving stable and undamaged autonomous agricultural operations. To improve the autonomous operational accuracy of agricultural machinery and achieve dynamic coordination between visual navigation and motion control, it is crucial to develop stable and reliable path tracking algorithms and control strategies.

The core of path tracking lies in the application of control technology, which encompasses a variety of strategies including proportional-integral-derivative (PID) control, adaptive control, SMC, linear quadratic regulator (LQR), MPC and machine learning methods. Among these, the strong robustness of SMC and the optimization capabilities of MPC render their combined architecture promising for addressing uncertainties in complex systems. Zhang et al. designed an adaptive model predictive controller for tracking unmanned aerial vehicles, demonstrating the robust stability and control accuracy of the associated control algorithm [8]. Dong et al. combined sliding mode control with MPC to enhance vehicle trajectory tracking performance on slippery roads [9]. Qi et al. proposed a sliding mode model predictive controller integrated with thrust allocation (TA) and utilized sliding mode control to ensure robust stability [10]. Teng et al. introduced an improved MPC motion stability controller that combines sliding mode variable structure active steering to enhance trajectory tracking stability of vehicles in extreme scenarios [11]. These studies underscore the potential of MPC-SMC fusion architectures in improving robustness and tracking performance. However, in agricultural machinery visual navigation scenarios, existing fusion schemes still fall short in synergistic optimization of significant perception delays and control responses.

Meanwhile, a large number of studies have focused on tracking algorithms using MPC and its nonlinear MPC [12–14]. Mehraban et al. incorporated fuzzy logic into the MPC framework to adaptively generate controller weighting factors [15]. Yin et al. combined extended Kalman filtering with robust tube-based model predictive control to enhance tracking performance [16]. Considering the unique characteristics of agricultural machinery, Soitinaho et al. designed a path tracking and obstacle avoidance method based on nonlinear model predictive control specifically for front-wheel-drive tractors. The speed and steering actuators were simulated as first-order systems without time delay, achieving precise path tracking and obstacle avoidance by optimizing the cost function [17]. However, the increased algorithmic complexity poses real-time bottlenecks for agricultural machinery embedded platforms, leading to two distinct optimization approaches: Zhou et al. proposed an adaptive learning model predictive control strategy that utilizes Bayesian optimization to dynamically adjust control rules and reduce computational burden [18]; Wang et al. employed a particle swarm optimization algorithm to achieve adaptive optimization of the prediction range [19]. Both approaches break through computational efficiency constraints from data-driven and time-domain optimization perspectives, respectively. To further enhance the robustness of path tracking algorithms in various environments, Zhao et al. integrated an adaptive Extended Kalman Filter (EKF) with model mismatch estimation

for state compensation, achieving unbiased path following [20]. Zhu et al. proposed a fault-tolerant MPC method based on a time-varying intermediate estimator to mitigate the effects of actuator faults [21]. In the field of collaborative optimization of visual navigation and path tracking, Chris et al. proposed a learning-based nonlinear model predictive control approach, providing insights for the collaborative optimization of visual perception and control models [22]. While their work provides valuable insights, it does not explicitly tackle the fundamental challenge of synchronizing visual perception latency with the motion response in dynamic operational scenarios.

In summary, although existing vehicle path tracking research has made significant strides in enhancing the accuracy of kinematic models and robust control mechanisms, there remains a notable lack of focus on the development of pure vision-based navigation path tracking algorithms that incorporate robust control mechanisms. This is particularly true in complex farmland environments characterized by considerable perception delays [23].

To address this gap, we propose a predictive control architecture termed HDC-CC. The overall structure framework is illustrated in Figure 1, which integrates visual delay compensation and motion control co-optimization. The core contributions of this work are as follows: (1) To tackle the significant delay issues encountered in agricultural machinery vision systems within farmland environments, we introduce a delay compensation algorithm that based on historical compensation errors. By synergizing a delay state observer with the dynamic characteristics of agricultural machinery, we can real-time predict path drift and integrate a real-time compensation correction module at the forefront of the MPC controller. This approach effectively mitigates path tracking deviations caused by delays in image acquisition and processing; (2) We construct a hierarchical coupling framework that interlinks MPC and SMC, fostering a profound synergy between path tracking and dynamic stability through a bidirectional constraint transmission mechanism. The MPC layer generates precise front wheel steering angle commands based on forward-looking optimization, while the SMC layer continuously monitors vehicle stability boundaries in real time and utilizes the sliding surface to swiftly respond and generate active yaw moments, thereby enhancing the system's robust adaptability to unstructured farmland environments.

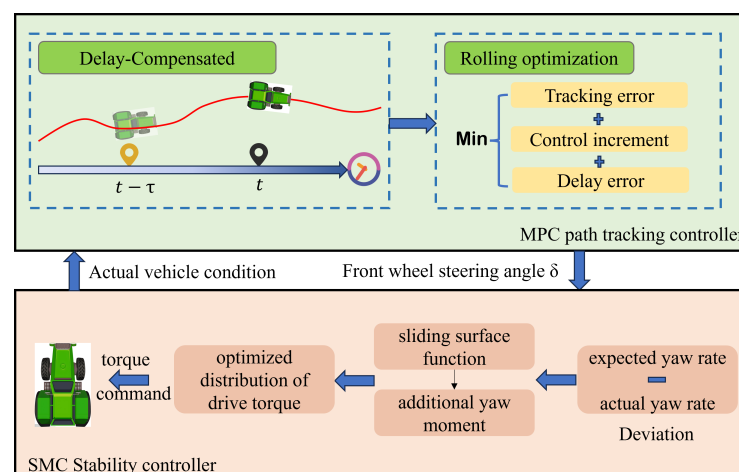


Figure 1. The overall structure framework.

2. Vehicle Model

The typical operating trajectory of an agricultural tractor in the field necessitates traveling in a straight line within a single working line, executing a turn at the end of the row, and subsequently entering the next working line. Throughout this operational process, high-performance straight-line trajectory tracking is critical to the effectiveness of tillage and constitutes the majority of the operation. Concurrently, high-precision curve trajectory track-

ing, particularly at the end of the row, is of paramount importance, as it not only ensures that the tractor smoothly and accurately transitions into the next working line but also significantly mitigates the risk of repeated trampling in the field caused by improper operation.

To address the demand for high-precision, robust operational trajectory tracking in complex and variable farmland environments, an accurate vehicle model is essential as the foundation for the path tracking control system [24,25]. Vehicle models are primarily categorized into kinematic models and dynamic models: kinematic models describe vehicle pose changes based on geometric relationships, providing the advantage of simplified calculations; dynamic models incorporate parameters such as mass, moment of inertia, and tire mechanics to more accurately characterize a vehicle's dynamic response characteristics on unstructured terrain, particularly demonstrating superior predictive capability for complex conditions such as side slip and load transfer. For agricultural machinery operating at moderate speeds with significant lateral dynamics influences, full vehicle models with higher degrees of freedom offer superior fidelity but introduce prohibitive computational complexity for real-time model predictive control applications. Conversely, purely kinematic models lack the necessary dynamic coupling representation for accurate path tracking under varying soil conditions. The 2-DoF and 3-DoF dynamic models thus represent an optimal compromise, capturing essential lateral-yaw-roll couplings while maintaining computational tractability for embedded implementation. Given the pronounced disturbance characteristics of uneven road surfaces and time-varying soil parameters in agricultural operational environments, the development of a dynamic vehicle model is a prerequisite for achieving reliable path tracking. Among these, two-degree-of-freedom and three-degree-of-freedom vehicle models are commonly selected due to their favorable balance between model accuracy and computational complexity.

2.1. 2-DoF Vehicle Dynamics Model

The two-degree-of-freedom model of the vehicle is predicated on several idealized assumptions:

1. Constant Longitudinal Motion: The vehicle maintains a constant speed, resulting in zero translational acceleration along the x -axis (longitudinal direction);
2. Planar Ground Motion: Disregarding the effects of the suspension system, the vehicle body is constrained to perform only planar motion parallel to the ground, meaning that vertical motion, pitch motion, and roll motion are all restricted to zero;
3. Simplified Steering Input: The dynamics of the steering system are not considered; instead, the steering angle of the front wheels is directly utilized as the model input.

Based on these assumptions, vehicle motion is simplified to a single-track model that incorporates only lateral translational motion (along the y -axis) and yaw motion (rotation about the z -axis). While these simplifications ignore suspension dynamics and terrain-induced roll/pitch effects, which may reduce model fidelity on highly uneven terrain, they provide a sufficient dynamics representation for designing the upper-layer path tracking controller during typical steady-state field operations. Furthermore, the model presumes that the lateral stiffness of the left and right wheels on the same axle is identical, allowing a single wheel to effectively represent the entire axle. The motion and force diagram of the single-track model is illustrated in Figure 2.

Figure 2a illustrates the change in the vehicle's state from time t to $t + \Delta t$. From the geometric relationship, the rate of change of velocity along the x and y axes of the ground coordinate system is described by Equation (1), where ω represents the yaw angular velocity, v_x and v_y represent the longitudinal and lateral velocities respectively.

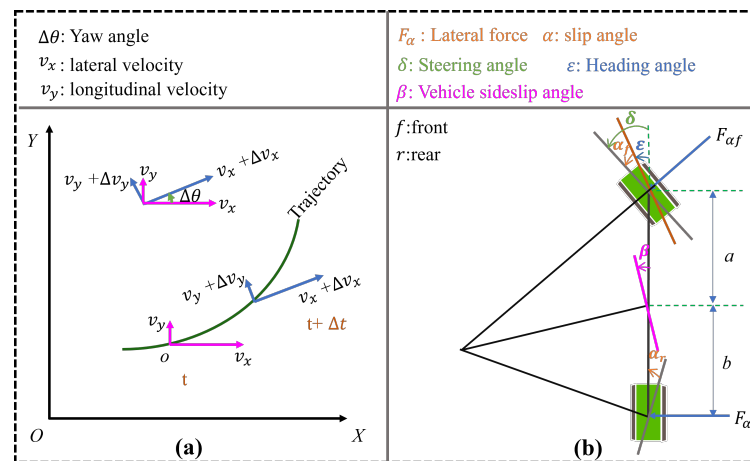


Figure 2. The motion and force diagram of the single-track model. (a) the change in the vehicle's state from time t to $t + \Delta t$. (b) a simplified vehicle subjected solely to lateral forces acting on the front and rear axles.

$$\begin{cases} a_x = \frac{\Delta v_x - v_y \Delta \theta}{\Delta t} = \dot{v}_x - v_y \omega \\ a_y = \frac{\Delta v_y + v_x \Delta \theta}{\Delta t} = \dot{v}_y + v_x \omega \end{cases} \quad (1)$$

Figure 2b depicts a simplified vehicle subjected solely to lateral forces acting on the front and rear axles. When the lateral acceleration of the vehicle is limited to 0.4 g or below, the lateral characteristics of the tires remain within the linear range, indicating that the lateral force F_{α} and lateral stiffness C_{α} are equivalent [26]. Based on Newton's second law and Euler's law of rotation, let the vehicle mass be denoted as m , I_z as the moment of inertia of the vehicle about the z-axis, a as the distance from the front axle to the center of mass, and b as the distance from the rear axle to the center of mass. The differential equations describing the dynamics of this two-degree-of-freedom system are established as shown in Equation (2).

$$\begin{cases} 2C_{\alpha f} \alpha_f \cos \delta + 2C_{\alpha r} \alpha_r = m a_y \\ 2C_{\alpha f} \alpha_f \cos \delta \cdot a - 2C_{\alpha r} \alpha_r \cdot b = I_z \dot{\omega} \end{cases} \quad (2)$$

When θ is small, the calculations for the heading angle ε and the center of mass lateral deviation angle β are presented in Equations (3) and (4).

$$\varepsilon = \tan \varepsilon = \frac{v_y + \omega a}{v_x} \quad (3)$$

$$\beta = \tan \beta = \frac{v_y}{v_x} \quad (4)$$

Thus, let $v_y = v$ and $v_x = u$ (constant), and calculate the lateral deviation angles of the front and rear wheels using Equations (5) and (6).

$$\alpha_f = -(\delta - \varepsilon) = -\left(\delta - \frac{v + \omega a}{u}\right) = -\left(\delta - \frac{u\beta + \omega a}{u}\right) = \beta + \frac{a\omega}{u} - \delta \quad (5)$$

$$\alpha_r = \frac{v - \omega b}{u} = \frac{u\beta - \omega b}{u} = \beta - \frac{\omega b}{u} \quad (6)$$

During agricultural straight-line tracking and gentle turning maneuvers, the vehicle exhibits sufficiently small yaw rates and sideslip angles, which confine the lateral dynamics to the linear region of tire forces. Under these conditions of small angular velocities, $\cos \delta = 1$.

By substituting Equations (5) and (6) into Equation (2), let $k_1 = 2C_{\alpha f}$ and $k_2 = 2C_{\alpha r}$, further linearizing the model as shown in Equation (7).

$$\begin{cases} k_1\left(\beta + \frac{a\omega}{u} - \delta\right) + k_2\left(\beta - \frac{b\omega}{u}\right) = m(\dot{v} + u\omega) \\ k_1\left(\beta + \frac{a\omega}{u} - \delta\right)a - k_2\left(\beta - \frac{b\omega}{u}\right)b = I_z\dot{\omega} \end{cases} \quad (7)$$

The simplification yields:

$$\dot{\beta} = \frac{k_1\left(\beta + \frac{a\omega}{u} - \delta\right) + k_2\left(\beta - \frac{b\omega}{u}\right) - mu\omega}{mu} = \omega\left(\frac{ak_1 - bk_2}{mu^2} - 1\right) + \beta\frac{k_1 + k_2}{mu} - \delta\frac{k_1}{mu} \quad (8)$$

$$\dot{\omega} = \frac{k_1\left(\beta + \frac{a\omega}{u} - \delta\right)a - k_2\left(\beta - \frac{b\omega}{u}\right)b}{I_z} = \omega\frac{a^2k_1 + b^2k_2}{uI_z} + \frac{ak_1 - bk_2}{I_z}\beta - \frac{ak_1}{I_z}\delta \quad (9)$$

2.2. 3-DoF Vehicle Dynamics Model

The three-degree-of-freedom model and the two-degree-of-freedom model differ in their capacity to comprehensively describe the vehicle's motion state. The three-degree-of-freedom model, in addition to accounting for the vehicle's lateral displacement and heading angle, incorporates an additional roll degree of freedom that characterizes the vehicle's rotational motion around the vertical axis. This inclusion allows for a more accurate representation of the lateral dynamic coupling effects observed during the steering maneuvers of agricultural machinery. The model is constructed based on the previously stated assumptions 2 and 3. Under the specified force conditions, it is simplified into a three-degree-of-freedom model encompassing longitudinal, lateral, and yaw motions. The vehicle dynamics equations are presented in Equation (10):

$$\begin{cases} m(\dot{v}_x - v_y\omega) = 2F_{xf}\cos\delta - 2F_{yf}\sin\delta + 2F_{xr} \\ m(\dot{v}_y + v_x\omega) = 2F_{xf}\sin\delta + 2F_{yf}\cos\delta + 2F_{yr} \\ I_z\dot{\omega} = a(2F_{xf}\sin\delta + 2F_{yf}\cos\delta) - 2bF_{yr} + \Delta M \end{cases} \quad (10)$$

where F_{xf} and F_{xr} denote the longitudinal forces acting on the front and rear tires of the vehicle, respectively, while F_{yf} and F_{yr} represent the lateral forces acting on the front and rear tires. By employing a linear tire model, the lateral force Equation (11) and longitudinal force Equation (12) for the tires can be derived as follows:

$$\begin{cases} F_{xf} = C_{lf}S_f \\ F_{xr} = C_{lr}S_r \end{cases} \quad (11)$$

$$\begin{cases} F_{yf} = C_{cf}\alpha_f = C_{cf}\left(\frac{v_y + a\omega}{v_x} - \delta\right) \\ F_{yr} = C_{cr}\alpha_r = C_{cr}\left(\frac{v_y - b\omega}{v_x}\right) \end{cases} \quad (12)$$

where C_{lf} and C_{lr} signify the longitudinal stiffnesses; C_{cf} and C_{cr} indicate the lateral stiffnesses; s_f and s_r denote the slip rates; and α_f and α_r represent the lateral angles. Substituting Equations (11) and (12) into (10) and rearranging the terms yields the following results:

$$\dot{v}_x = v_y\omega + \frac{2}{m}\left[C_{lf}S_f + C_{cf}\left(\frac{v_y + a\omega}{v_x} - \delta\right)\delta + C_{lr}S_r\right] \quad (13)$$

$$\dot{v}_y = -v_x\omega + \frac{2}{m}\left[C_{cf}\left(\frac{v_y + a\omega}{v_x} - \delta\right) + C_{cr}\frac{v_y - b\omega}{v_x}\right] \quad (14)$$

$$\dot{\omega} = \frac{2}{I_z} \left[aC_{cf} \left(\frac{v_y + a\omega}{v_x} - \delta \right) - bC_{cr} \frac{v_y - b\omega}{v_x} \right] + \frac{\Delta M}{I_z} \quad (15)$$

Using Equations (8), (9), and (13)–(15) to construct the state-space representation, both 2-DOF and 3-DOF single-track models were implemented in MATLAB (version R2021b) and Simulink (version R2021b), with their simulation results compared. With initial values of v_y and ω set to 0 and v_x initialized at 5 m/s, simulations were conducted for the sinusoidal angle input scenario. The front wheel steering input was defined as $\delta = \frac{10\pi}{180} \sin(0.8\pi \cdot \text{time})$, reflecting the typical frequency of human steering wheel rotation, approximately 0.3 to 0.5 Hz. As depicted in Figure 3, the comparison curves illustrate that both models exhibit a high degree of consistency in the trends of lateral velocity and yaw angular velocity, with their waveform characteristics and phase relationships generally aligning. This observation suggests that both models are grounded in the same theoretical framework regarding fundamental dynamic characteristics. Analysis of the tire slip angle response confirms that the 3-DOF model accurately characterizes centrifugal load transfer dynamics. Notably, the front wheel slip angle increases from 3.12° to 6.12° , while the slip angle rises from 1.82° to 3.88° during center-of-gravity shifts, accurately quantifying tire load transfer. This characteristic is critical for predicting unique tire wear patterns and adhesion boundary changes in agricultural field operations. During a 30° sharp turn at 5 m/s, the 3-DOF model predicts a maximum roll angle of 4.86° . By including roll dynamics, the model captures roll-tire coupling phenomena through integrated tire-ground interaction mechanics. Consequently, the 3-DOF model demonstrates high fidelity for typical agricultural scenarios involving low-speed, high-steer maneuvers.

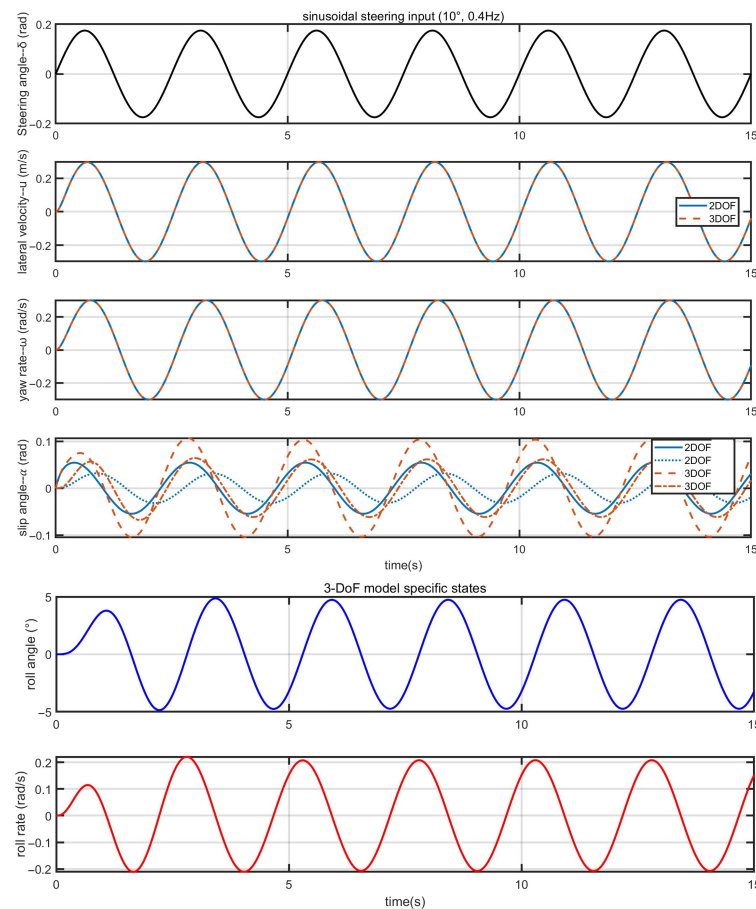


Figure 3. Comparison of dynamic responses between 2-DOF and 3-DOF models.

3. MPC Path Tracking Controller Design

In path tracking control for visual autonomous driving systems, the visual perception module plays a crucial role in environmental perception and the extraction of crop row center lines. However, field visual navigation line extraction algorithms that rely on deep learning (as referenced in [27–30]) require between 100 to 500 ms for image processing, which results in vehicle posture information being significantly delayed relative to the actual state. To mitigate the reduction in control performance caused by the inherent time delay in visual perception, this paper proposes an MPC architecture that incorporates a visual delay compensation strategy. The fundamental concept of this architecture is to leverage the vehicle dynamics model and real-time motion state to construct a state prediction compensator. Specifically, the system first acquires the visual pose observation values, which are subject to inherent delays, while simultaneously recording the corresponding control input sequence within a historical time window. Subsequently, utilizing the vehicle dynamics model, forward integration calculations are performed to dynamically advance the delayed visual observations and adjust them in real-time to the present moment, thereby yielding an accurate estimate of the current system state. The compensated state estimate is employed directly as the initial state of the MPC prediction model, allowing the controller to optimize the future control sequence in each optimization cycle based on real-time state information.

3.1. Delay State Observer Design

The visual system output at time t , $X_{vision}(t)$, corresponds to the vehicle state at time $t - \tau$. To address this latency, a time-delay state observer is designed that estimates the current state by integrating historical control inputs $U_{history}$ over $[t - \tau, t]$ with the delayed vision output, as formulated in Equation (16):

$$\hat{X}(t) = \Psi(X_{vision}(t), U_{history}) \quad (16)$$

where $\hat{X}(t)$ denotes the estimated state at time t , and $\Psi(\cdot)$ represents the forward integration operator based on the vehicle dynamics model. Within $t - \tau$, control inputs are known while state evolution depends on instantaneous states. The visual delay time τ is primarily determined by the image processing pipeline, including image acquisition, feature extraction, and navigation line computation. In practical implementations, τ can be empirically measured as the average processing latency of the vision system. For the purpose of delay compensation, τ is treated as a known system parameter. The proposed delay state observer does not require real-time estimation of τ , but rather focuses on compensating for its effects through model-based forward prediction. For delay compensation, integration initiates from the vision-measured state at $[t - \tau, t]$, propagating forward to current time t via numerical integration. The continuous-time forward integration is mathematically expressed in Equation (17):

$$\hat{X}(t) = X(t - \tau) + \int_{t-\tau}^t f(X(\xi), U(\xi)) d\xi \quad (17)$$

Here, $f(\cdot)$ denotes the vehicle dynamics model. The forward integration is implemented numerically using the explicit Euler method due to its computational efficiency and suitability for real-time control applications. Combining Equations (10)–(15), the simplified vehicle dynamics model in state-space form is expressed as Equation (18):

$$\begin{cases} \dot{\chi} = f(\chi, u) \\ y = C\chi \end{cases} \quad (18)$$

where the state vector $\chi = [v_x, v_y, \omega, \varphi, X, Y]^T$, control input $u = \delta$, and output vector $y = [\varphi, Y]^T$, $c = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$

Given a sampling time T_s , the delay steps are calculated as $k_\tau = \lfloor \tau/T_s \rfloor$. For implementation in discrete-time systems, numerical integration of Equation (17) proceeds as follows:

$$\hat{X}_k = X_{k-k_\tau} + \sum_{i=0}^{k_\tau-1} f(X_{k-k_\tau+i}, U_{k-k_\tau+i})T_s \quad (19)$$

The delayed state X_{k-k_τ} obtained from the vision system at time step $k - k_\tau$, serves as the integration initial condition. Forward integration is then performed from t_{k-k_τ} to t_k using the dynamics model $f(\cdot)$ and recorded control inputs $U_{k-k_\tau+i}$. After executing k_τ integration steps, the current state estimate $\hat{X}_k$ at time t_k is obtained.

3.2. Prediction Model Construction

To perform prediction and control based on the compensated state, the compensated estimate $\hat{X}_k$ is treated as the current state X_k . An extended state vector is constructed by augmenting this state with the previous control input, as shown in Equation (20):

$$\tilde{\xi}_k = \begin{bmatrix} X_k \\ u_{k-1} \end{bmatrix} \quad (20)$$

where $X_k \in R^n$ and $u_{k-1} \in R^m$. Utilizing a 3-DOF agricultural vehicle dynamics model, a prediction model that accounts for the visual delay period is established through linearization and discretization, which describes the evolution law of the extended state under the control increment $\Delta u(k) = u(k) - u(k-1)$. The prediction model is presented in Equations (21) and (22):

$$\tilde{\xi}_{k+1} = \bar{A}_k \tilde{\xi}_k + \bar{B}_k \Delta u_k + \bar{d}_k \quad (21)$$

$$\eta(k) = C \tilde{\xi}(k) = \begin{bmatrix} \varphi(k) & Y(k) \end{bmatrix}^T \quad (22)$$

where $\bar{A}_k = \begin{bmatrix} A_k & B_k \\ 0 & I \end{bmatrix}$, $\bar{B}_k = \begin{bmatrix} B_k \\ I \end{bmatrix}$, $A_k \in R^{n \times n}$ and $B_k \in R^{n \times m}$ are state and control matrices from the linearized-discrete dynamics at $\tilde{\xi}_k$. I is the identity matrix, and $\bar{d}_k \in R^{n+m}$ represents the discretization error and linearization truncation error.

The MPC controller predicts the future output of the system by iterating the system Equations (20)–(22). Over prediction horizon N_p and the control horizon N_c , the vectorized output prediction equation is provided by Equation (23):

$$Y(k) = \underbrace{\Psi(k)\tilde{\xi}(k)}_{\text{Initial State Influence}} + \underbrace{\Theta(k)\Delta U(k)}_{\text{Control Action Effects}} + \underbrace{\Gamma(k)\Omega(k)}_{\text{Model Error Effects}} \quad (23)$$

where,

$$\Psi(k) = \begin{bmatrix} C\bar{A}_k \\ C\bar{A}_k^2 \\ \vdots \\ C\bar{A}_k^{N_p} \end{bmatrix}, \Theta(k) = \begin{bmatrix} C\bar{B}_k & 0 & \cdots \\ C\bar{A}_k\bar{B}_k & C\bar{B}_k & \cdots \\ \vdots & \vdots & \ddots \\ C\bar{A}_k^{N_p-1}\bar{B}_k & C\bar{A}_k^{N_p-2}\bar{B}_k & \cdots \end{bmatrix}, \Gamma(k) = \begin{bmatrix} \frac{1}{\bar{A}_k} & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \bar{A}_k^{N_p-1} & \bar{A}_k^{N_p-2} & \cdots & 1 \end{bmatrix}.$$

$Y(k)$ is the output sequence, $\Delta U(k)$ is the control increment sequence, and $\Omega(k)$ is the error sequence.

3.3. Objective Function

Building upon the prediction model framework, this study presents a refined objective function design for model predictive control. The parameters for the MPC controller as listed in Table 1.

Table 1. MPC Controller Parameters.

Parameter	Symbol	Value
Prediction Horizon	N_p	20
Control Horizon	N_c	10
State Weight Matrix	Q	diag(10, 100)
Control Weight Matrix	R	1
Compensation Error Weight Matrix	S	diag(50, 10)
Compensation History Steps	M	5
Forgetting Factor	λ	0.8
Sampling Time	T_s	0.02 s

The function is designed to minimize the error between the predicted state variables and the desired values in the time domain while constraining the controller output. Traditional MPC formulations exhibit significant limitations in vision-based navigation systems due to inherent visual processing latency, compromising real-time performance and robustness in dynamic environments. To address this, a modified objective function retains conventional tracking error and control penalty terms while incorporating a novel delay compensation error penalty, establishing an MPC path-tracking controller suitable for visual delay scenarios, as shown in Equation (24).

$$J(k) = \sum_{j=1}^{N_p} \left\| \eta_{ref}(k+j) - \eta(k+j) \right\|_Q^2 + \sum_{j=0}^{N_c-1} \left\| \Delta u(k+j) \right\|_R^2 + \left\| e_{comp} \right\|_s^2 + \rho \varepsilon_s^2 \quad (24)$$

Here, ε_s represents a slack variable, and ρ serves as its penalty weight for managing constraint conflict. The error compensation term accounts for the cumulative error generated during the forward integral compensation process. When optimizing at timestep k , the compensation error from the most recent M steps is employed to construct the penalty term. By weighting the historical compensation errors, the controller is compelled to consider past compensation accuracy when optimizing future control actions, thus suppressing the cumulative drift caused by visual delay. The historical compensation error penalty term is defined as illustrated in Equation (25).

$$\left\| e_{comp} \right\|_s^2 = \sum_{m=0}^{M-1} \lambda^m e_{comp}(k-m)^T S e_{comp}(k-m) \quad (25)$$

where, $e_{comp}(k-m)$ denotes the historical compensation error from m steps prior, λ represents the error weight, and S is a positive definite diagonal weight matrix prioritizing positional tracking accuracy. To facilitate the solution process, Equation (24) is reformulated into standard quadratic programming form based on the prediction Equations (21) and (22). The resulting quadratic programming problem in Equation (26) is solved using the quadprog solver from MATLAB's Optimization Toolbox.

$$\min_{\Delta U(k), \varepsilon} J(k) = \frac{1}{2} \Delta U(k)^T \mathbf{H} \Delta U(k) + \mathbf{f}^T \Delta U(k) + \rho \varepsilon_s^2 \quad (26)$$

s.t.

$$\Delta U_{\min} \leq \Delta U(k) \leq \Delta U_{\max}$$

$$U_{\min} \leq \mathbf{C}_p \mathbf{u}(k-1) + \mathbf{D} \Delta U(k) \leq U_{\max}$$

$$\eta_{\min} - \varepsilon_s \mathbf{E} \leq \Psi \xi(k) + \Theta \Delta U(k) \leq \eta_{\max} + \varepsilon_s \mathbf{E}$$

$$\varepsilon_s \geq 0$$

During the optimization at a single time step k , the historical errors $e_{\text{comp}}(k-m)$ for $m = 0, \dots, M-1$ are already known and fixed values. Therefore, the entire term $\|e_{\text{comp}}\|_s^2$ is a scalar constant during the minimization of $J(k)$ with respect to the future control sequence $\Delta U(k)$. Since adding a constant to an objective function does not change the optimal solution $\Delta U^*(k)$, this term is omitted when constructing the standard Quadratic Programming (QP) problem in Equation (26). However, this term plays a vital role across consecutive control steps: the compensation error $e_{\text{comp}}(k)$ is updated at every time step based on the difference between the predicted state and the newly available (but delayed) vision measurement. This creates a feedback mechanism on the quality of the delay compensation, influencing the controller's behavior over time by shifting the state prediction baseline $\hat{X}(k)$.

The objective function incorporates the following constraints: steering angle $-0.52 \text{ rad} \leq \delta \leq 0.52 \text{ rad}$; steering angle change rate $-0.26 \text{ rad} \leq \Delta \delta \leq 0.26 \text{ rad}$. Additionally, path tracking boundary constraints are imposed to ensure that the agricultural machinery does not deviate excessively from the predefined path during operation, with constraints of lateral deviation $e_y \leq 0.1 \text{ m}$ and heading angle deviation $e_\phi \leq 10^\circ$. Given the earlier assumption that the tires operate within the linear region, the tire force calculation employs an approximate expression. To ensure that the lateral force of the tire is linearly related to the lateral deviation angle and that the tire force remains within the adhesion limit, restrictions are applied to the slip angles of the tire and center of mass: $-5^\circ \leq \alpha_r \leq 5^\circ$, $-2.5^\circ \leq \alpha_f \leq 2.5^\circ$, and $-2^\circ \leq \beta \leq 2^\circ$, the controller sampling time is $T_s = 0.02 \text{ s}$.

The controller implements the first element U_0 of the optimized control sequence U as the vehicle's actuation command. This rolling horizon optimization process repeats each control cycle, enabling continuous trajectory tracking. The objective function design achieves precise path following while compensating for vision-induced cumulative errors, enhancing system robustness and control smoothness in dynamic environments.

4. Stability Controller Design

Within the MPC path tracking framework, which incorporates visual delay compensation, the vehicle acquires environmental information via the visual perception system and generates a reference path. The MPC controller computes the front wheel steering angle δ , through a rolling optimization process. However, under typical agricultural conditions, such as navigating rough terrain or experiencing localized variations in adhesion, path tracking may lead to vehicle instability due to uneven ground or interference from crop residues. To address this limitation, a stability controller is implemented as a lower-layer actuator, forming a coordinated control architecture with the predictive controller. Following visual delay compensation, the corrected path information feeds into the MPC module. Its output steering command δ then directly inputs to the stability controller for execution.

4.1. Calculation of Expected Yaw Angular Velocity

The core input parameters for the stability controller include: Yaw rate reference value generated by MPC ω_{mpc} ; Longitudinal vehicle speed v_x (m/s); Front wheel steering angle δ ; Road adhesion coefficient μ . Based on vehicle dynamic stability conditions, the yaw rate

safety boundary under current operating conditions is calculated through Equation (27), g is the acceleration due to gravity (9.81 m/s^2).

$$|\omega_{max}| = \frac{\mu g}{\max(v_x, v_{min})} \quad (27)$$

A safeguard mechanism sets a minimum speed threshold $v_{min} = 0.5 \text{ m/s}$ to prevent boundary overflow during low-speed operation. A dual-layer decision logic generates the final desired value per Equation (28), where $\text{sgn}(\delta)$ retains the steering direction sign from MPC.

$$\omega_d = \begin{cases} \omega_{mpc} & \text{if } |\omega_{mpc}| \leq \omega_{max} \\ \text{sgn}(\delta) \cdot \omega_{max} & \text{otherwise} \end{cases} \quad (28)$$

This architecture prioritizes vehicle dynamic stability. When ω approaches the safety boundary, stability intervention is triggered, prompting the MPC controller to temporarily relax path-tracking accuracy. This achieves a dynamic trade-off between safety and tracking precision.

4.2. Additional Yaw Moment Calculation

The required additional yaw moment ΔM is computed via a sliding mode control algorithm based on the deviation between the desired yaw rate ω_d and the actual yaw rate ω . Defining the yaw rate tracking error as $e_\omega = \omega - \omega_d$, the sliding surface function is designed as Equation (29). When the system state reaches the sliding surface, $s_\omega = 0$.

$$s_\omega = c_\omega e_\omega + \dot{e}_\omega \quad (29)$$

Here, c_ω represents a positive weighting coefficient balancing error magnitude and error change rate. To ensure robustness during the approach phase and reduce chattering, a constant-velocity reaching law is adopted, as shown in Equation (30):

$$\dot{s}_\omega = -\varepsilon \cdot \text{sgn}(s_\omega) - K s_\omega, \quad \varepsilon > 0 \quad (30)$$

To rigorously demonstrate the stability of the designed sliding mode control law and verify that the sliding condition is satisfied, a Lyapunov stability analysis is conducted. Consider the following candidate Lyapunov function:

$$V = \frac{1}{2} s_\omega^2 \quad (31)$$

This function is positive definite, as $V > 0$ for all $s_\omega \neq 0$, and $V = 0$ only when $s_\omega = 0$. The time derivative of V is obtained as:

$$\dot{V} = s_\omega \dot{s}_\omega \quad (32)$$

Substituting the reaching law, $\dot{s}_\omega = -\eta \cdot \text{sgn}(s_\omega)$ from Equations (30)–(32) yields:

$$\dot{V} = s_\omega \cdot (-\eta \cdot \text{sgn}(s_\omega))$$

For $s_\omega \neq 0$, the sign function can be expressed as $\text{sgn}(s_\omega) = s_\omega / |s_\omega|$. Therefore, the derivative simplifies to:

$$\dot{V} = -\eta \cdot s_\omega \cdot \frac{s_\omega}{|s_\omega|} = -\eta \cdot |s_\omega| \quad (33)$$

Given that the parameter η is strictly positive ($\eta > 0$), it follows from Equation (33) that $\dot{V} \leq 0$ for all s_ω , with equality holding only if $s_\omega = 0$. Specifically, $\dot{V} < 0$ for any $s_\omega \neq 0$. This satisfies the reaching condition of sliding mode motion, guaranteeing that the system state will be driven from any initial point in the phase space to the sliding manifold $s_\omega = 0$ within a finite time. Once the system state reaches the manifold, its dynamics are governed by the equation $s_\omega = 0$ defined in Equation (29), ensuring that the tracking error e_ω converges to zero exponentially.

Combining Equations (15), (29) and (30), the additional yaw moment derivation proceeds as follows:

$$f(\omega) = \frac{2}{I_z} \left[ac_{cf} \left(\frac{v_y + a\omega}{v_x} - \delta \right) - bc_{cr} \frac{v_y - b\omega}{v_x} \right]$$

$$\dot{\omega} = \frac{\partial f}{\partial t} + \frac{\Delta M}{I_z} \approx \frac{\Delta M}{I_z}$$

$$\Delta M = I_z [-\varepsilon \operatorname{sgn}(s_\omega) - Ks_\omega - c_\omega(\dot{\omega} - \dot{\omega}_d) + \dot{\omega}_d] - I_z f(\omega)$$

$$\Delta M = I_z (-\varepsilon \cdot \operatorname{sgn}(s) - Ks - c_\omega \dot{e}_\omega + \dot{\omega}_d) - \left[2ac_{cf} \left(\delta - \frac{v_y + a\omega}{v_x} \right) - 2bc_{cr} \frac{v_y - b\omega}{v_x} \right]$$

This dynamically compensates for unstable moments generated by front-wheel steering inputs, ensuring rapid tracking of ω_d by the actual yaw rate.

4.3. Torque Distribution Optimization

After obtaining the additional yaw torque ΔM , it is essential to convert it into execution commands for the four-wheel hub motors through the optimized distribution of drive torque. Considering the uneven vertical load distribution F_{zij} on each wheel and the sudden changes in local road adhesion coefficient μ_{ij} in agricultural environments, an optimization problem is formulated to minimize the tire force utilization ratio. Since tire lateral forces are coupled with driving forces and are relatively small compared to driving forces, only driving force distribution is considered. The objective function is formulated as shown in Equation (34).

$$\min J = \min \sum_{ij=fl,fr,rl,rr} \frac{F_{xij}^2}{(\mu_{ij}F_{zij})^2} \quad (34)$$

This objective function minimizes the sum of squared ratios of each tire's longitudinal force F_{xij} to its adhesion limit $\mu_{ij}F_{zij}$.

The torque allocation must strictly satisfy vehicle-level dynamic requirements from the upper-layer controller, enforced through two key equality constraints (35) and (36). Longitudinal Force Equilibrium: The sum of longitudinal tire force projections along the vehicle coordinates system's x-axis must equal the force required to achieve the target longitudinal acceleration a_x .

$$(F_{xfl} + F_{xfr}) \cos \delta_f + F_{xrl} + F_{xrr} = ma_x \quad (35)$$

Yaw Moment Equilibrium: The moment generated by the differential longitudinal forces between left and right tires must equal the additional yaw moment ΔM computed by the upper controller.

$$\frac{B_f}{2} (F_{xfr} - F_{xfl}) \cos \delta_f + \frac{B_r}{2} (F_{xrr} - F_{xrl}) = \Delta M \quad (36)$$

Here, B_f and B_r denote the front and rear axle track widths, respectively. This constraint produces the required ΔM through driven wheel torque differentiation. Each hub motor's torque capacity is bounded, directly constraining tire longitudinal forces per Equation (37).

$$|F_{xij}| \leq \frac{T_{max}}{R_{eff}} \quad (37)$$

T_{max} represents the maximum brake torque deliverable by each hub motor, while R_{eff} is the effective wheel radius. This constitutes a quadratic programming problem with linear constraints. The MATLAB fmincon solver computes optimal longitudinal forces F_{xij} for all four tires. The torque command for each hub motor is derived as $T_{ij} = F_{xij}R_{eff}$. The motor controller executes precise torque tracking based on these commands. The presented torque distribution strategy is agnostic to the actuator implementation. For tractors with standard differential braking systems, the optimized yaw moment ΔM is naturally executed as a differential braking force between the left and right wheels.

5. Experimental Results and Analysis

5.1. Effectiveness of Delay Compensation

The study validates the delay compensation module's efficacy using a visual delay time τ to 0.4 s. Tracking performance of the original MPC was compared with that of the MPC enhanced by the visual delay compensation module across three typical trajectories: circular, S-curve, and straight, as illustrated in Figure 4. In the trajectory fitting space matching comparison shown in Figure 3a, the original MPC path without delay compensation exhibits significant fluctuations, with noticeable jittering occurring in regions of curvature change along the circular trajectory and at the turning points of the S-curve trajectory. This phenomenon is primarily attributed to the delay-dynamic coupling effect. When the controller generates control commands based on historical state information, the delay hinders real-time responsiveness to curvature changes, resulting in a mismatch between the system state at the execution moment and the predicted state, thereby causing the accumulation of tracking errors. Although the fluctuation amplitude in the straight-line trajectory is minimal, the noise induced by the delay is amplified under integral action, leading to slight oscillations in both speed and attitude angle. As depicted in Figure 3b, the improved MPC algorithm significantly reduces the average tracking error. The tracking error for the circular path decreased from 0.46 m to 0.111 m (75.87% reduction), for the S-curve path from 0.564 m to 0.136 m (75.89% reduction), and for the straight path from 0.236 m to 0.051 m (95.58% reduction). After the introduction of visual delay compensation, the MPC incorporates the delay time into the system dynamics equations, constructs an extended state space model, and enables the controller to pre-compensate for state drift within the τ time frame, effectively mitigating the adverse effects of visual delay on the control system.

To verify the effectiveness of the proposed delay-aware method, it was compared with the baseline MPC and the simple forward prediction (SFP) method. The SFP method uses the same vehicle dynamics model for forward prediction but without the historical-error regularization. The results in Table 2 demonstrate that the proposed delay state observer with historical-error regularization significantly outperforms both the baseline MPC and the SFP method across all tested trajectories. It attains a minimal average tracking error of merely 0.099 m, representing a substantial 42.1% enhancement over the SFP method at 0.171 m. This advantage is consistently demonstrated on each specific path, with errors as low as 0.111 m for the circular path, 0.136 m for the S-shaped path, and 0.051 m for the straight-line path. This pronounced contrast underscores the critical role of

the historical-error regularization mechanism, as it actively compensates for the estimation errors accumulated over time.

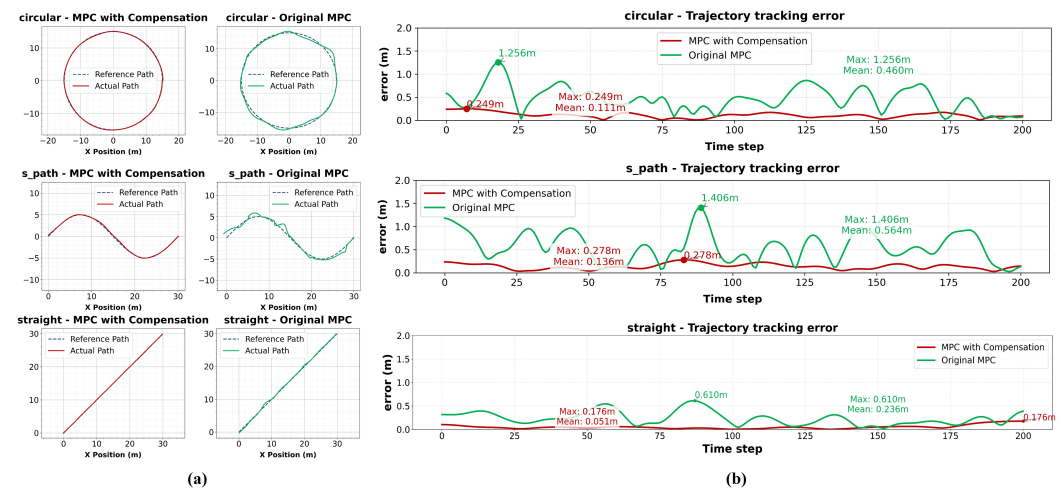


Figure 4. Tracking performance verification of the visual delay compensation module under typical trajectories (a) Path fitting results (b) Tracking error.

Table 2. Performance comparison of different delay compensation methods.

Controller	Circular Path Error (m)	S-Curve Path Error (m)	Straight Path Error (m)	Average Error (m)
Baseline MPC (no compensation)	0.460	0.564	0.236	0.420
Simple Forward Prediction (SFP)	0.185	0.231	0.098	0.171
Proposed Delay State Observer	0.111	0.136	0.051	0.099

To further systematically verify the robustness of the delay compensation module, tests were conducted on the improved algorithm under a delay scenario ranging from 0–1 s. Three typical trajectories circular, S-curve, and straight were selected, and the average tracking error was calculated at 50 random delay time points. The results were compared with the performance of the original MPC, as illustrated in Figure 5.

Figure 5a depicts the trend of tracking error as a function of delay time. As shown in the figure, the tracking error of the original MPC increases non-linearly with increasing delay time, indicating that the predicted trajectory based on the delayed state is spatially mismatched with the actual vehicle orientation. Consequently, the front wheel steering angle command consistently lags behind the actual required direction. This issue is particularly pronounced in the curvature transition regions of the circular and S-curve trajectories, where the misjudgment of steering timing caused by delayed observations significantly amplifies the amplitude of yaw angle oscillations, leading to increased trajectory tracking deviations. The MPC with the integrated delay compensation module effectively addresses the spatial mismatch issue through a real-time state synchronization mechanism. The core of this approach lies in dynamic predictive compensation, which ensures that control commands closely align with the vehicle's current actual pose, thereby maintaining near-zero delay error levels within the range of $\tau \leq 0.8$ s. When τ exceeds 0.8 s, the improved MPC begins to exhibit performance degradation; however, the magnitude of this degradation is significantly smaller than that of the original MPC. This degradation phenomenon highlights the physical limits of the compensation mechanism: when the delay time is excessively long, the accumulated uncertainty of the prediction model substantially diminishes the effectiveness of compensation.

Figure 5b quantitatively compares error distributions between baseline MPC and compensated MPC across three trajectory types using box plots. Overall, complex trajec-

tories, such as circular and S-curve paths, demonstrate more pronounced compensation effects compared to simple straight-line trajectories. Straight-line trajectories exhibit a significantly smaller median error reduction due to their inherent kinematic smoothness, which minimizes baseline control errors and thus constrains compensation potential. In the case of circular trajectories, 82% of the difference data points exceed the median value, confirming that the compensation module maintains stable gain characteristics during continuous curved motion. The median difference value for S-curve trajectories is 0.513, which surpasses the circular trajectory's value of 0.491 and the straight line's value of 0.349. Furthermore, S-curve trajectories exhibit the most concentrated distribution characteristics, with an interquartile range (IQR) of 0.301, which is lower than the circular trajectory's IQR of 0.405 and the straight line's IQR of 0.412. These results indicate that the MPC algorithm, enhanced with the fusion delay compensation module, retains robust performance across the entire delay range (0–1 s) and under various motion conditions.

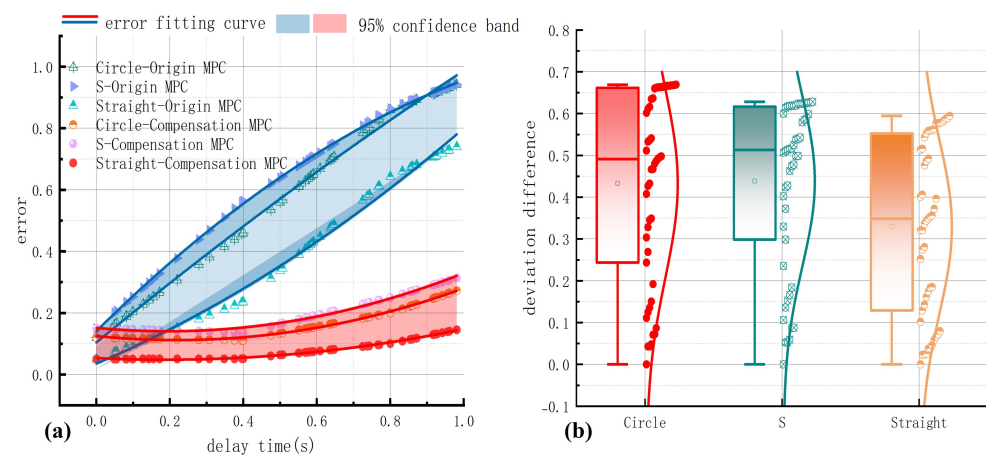


Figure 5. Performance analysis of the visual delay compensation module under three typical trajectories (a) Trend of tracking error with delay time (b) Quantitative analysis of error compensation effect.

5.2. Robustness Analysis of the Cooperative Controller

To verify the effectiveness of the proposed coordinated path-tracking and stability control strategy, a co-simulation platform was developed using CarSim (version 2019.1) and Simulink. Vehicle path tracking simulation analysis was conducted, with CarSim providing the vehicle dynamics model and the designed controller along with the tire force distribution method implemented in Simulink. Figure 6 illustrates the double-line shift scenario and the simulation framework in Simulink. The experiment aimed to track a reference path composed of straight and curved segments. The controlled vehicle was a medium-sized agricultural tractor, configured with key parameters including a mass $m = 3190 \text{ kg}$, yaw inertia $I_z = 11,903 \text{ kg} \cdot \text{m}^2$, wheelbase $L = 2.15 \text{ m}$, a front wheel tread of 1.505 m , a rear wheel tread of 1.635 m , front and rear cornering stiffnesses $C_{\alpha f} = 73 \text{ kN/rad}$ and $C_{\alpha r} = 152 \text{ kN/rad}$, and tread widths of 1.505 m (front) and 1.635 m (rear), as also detailed in Figure 6. It operated at a speed of 5 km/h on road surfaces with a friction coefficient of 0.7 for dry hard soil and 0.4 for wet soft soil.

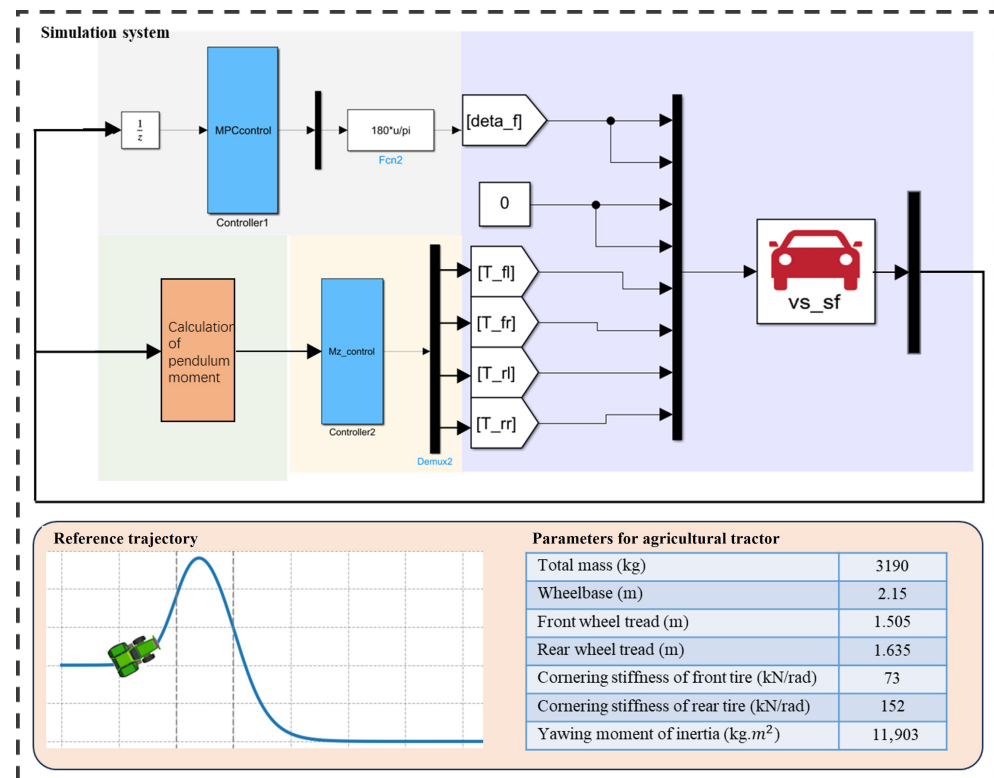


Figure 6. Double-line shift conditions and simulation framework in Simulink.

The simulation results presented in Figure 7 ($\mu = 0.7$) and Figure 8 ($\mu = 0.4$) illustrate effective trajectory tracking performance under varying road surface adhesion conditions. In high adhesion scenarios ($\mu = 0.7$), the average path tracking error decreased from 44.7 mm to 11.3 mm. When the vehicle enters a high curvature zone, the real-time stability compensation mechanism of the SMC layer effectively mitigates path deviation. Compared to the original MPC, the cooperative controller exhibits a significant enhancement in yaw control performance, with the average yaw angle error reduced by 73.24% (from 0.1566° to 0.0419°), alongside a reduction in the amplitude of yaw angular velocity fluctuations. This suggests that the controller continuously monitors variations in the vehicle's yaw rate and, upon nearing the safety boundary, activates sliding mode compensation control to generate an active yaw torque. This torque is distributed through differential braking of the drive wheels to counteract yaw dynamics nonlinear errors. Concurrently, the vehicle's center of gravity lateral deviation angle and front wheel steering angle remain within safety limits, thereby validating the dual-layer collaborative mechanism of the MPC and SMC layers in maintaining state boundary stability.

For low-friction road surfaces ($\mu = 0.4$), the cooperative controller achieved an average error reduction of 56.19% (from 81.2 mm to 35.6 mm), demonstrating superior adaptability to environmental conditions. In contrast to high-friction scenarios, low-friction surfaces generate smaller lateral forces, which result in longer convergence times for yaw rate and minor fluctuations. Nevertheless, the architecture maintains the error magnitude within a strict boundary of 36 mm, representing a 56% improvement over the baseline MPC system, thereby ensuring robust control under low-friction conditions. Regarding stability control, the reduction in the center of gravity lateral deviation angle reached 76.19% (from 0.0147° to 0.0035°), reflecting an enhancement of 1.19 percentage points over high-adhesion road surfaces (75.00%). This indicates that the SMC layer actively reinforces stability control weights through dynamic optimization of domain constraints when adhesion is insufficient. Although front wheel steering angle changes are constrained by the physical limitations of the road surface, direct yaw torque control based on drive wheel differential braking

can effectively compensate for degradation in steering performance, thereby validating the robustness of the sliding mode compensation mechanism under low μ conditions.

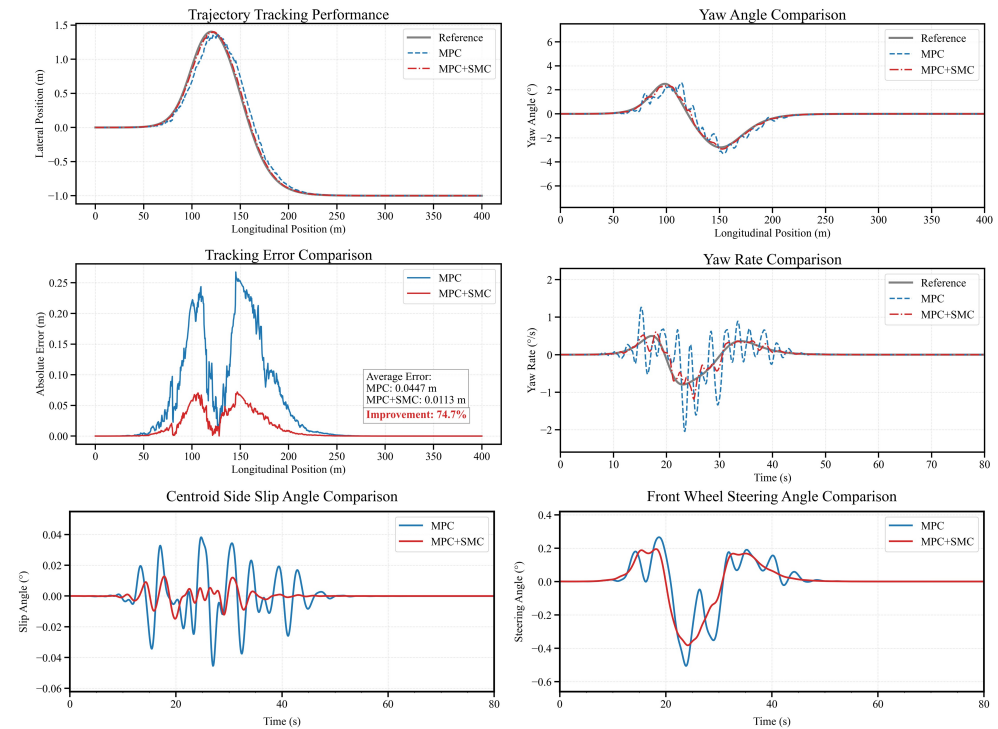


Figure 7. Comparison of path tracking performance under cooperative control on high-adhesion road surfaces ($\mu = 0.7$).

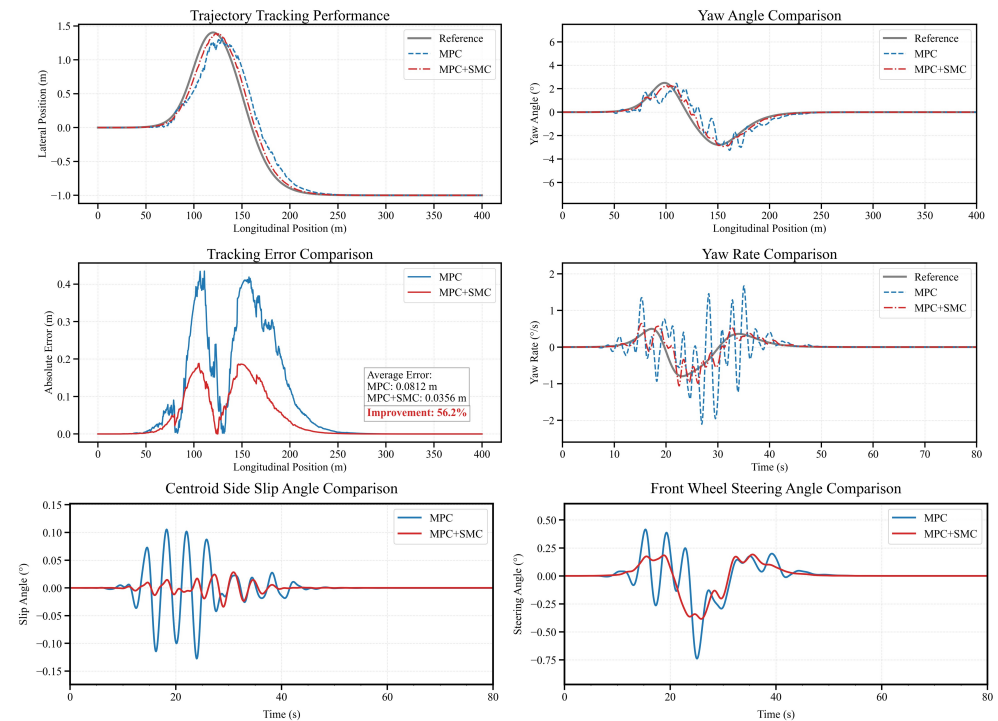


Figure 8. Comparison of path tracking performance under low adhesion road surface conditions ($\mu = 0.4$) with cooperative control.

5.3. Co-Simulation Validation

To verify the path tracking performance of the proposed algorithm for autonomous navigation in agricultural fields, a virtual farm environment was constructed using the Gazebo simulation platform (Figure 9a). This environment simulates essential farm features, including structured crop rows and soil texture. The test platform is equipped with a forward-facing virtual RGB camera, which captures image data in real time via the perception node implemented in Python (version 3.8). In the perception pipeline, the system first employs the ExG segmentation algorithm to accurately extract seedling row regions, followed by robust least squares fitting to generate the desired reference heading. To comprehensively evaluate the real-time performance of the perception system, precise timing measurements were conducted for each processing stage: the ExG vegetation segmentation stage requires an average processing time of 45 ms, while the subsequent robust least squares fitting stage averages 18 ms, resulting in a total processing cycle of 63 ms for the entire perception pipeline. The perception node employs the ExG segmentation algorithm to extract the seedling row region and generates the desired reference heading through robust least squares fitting. The control layer compiles the HDC-CC core algorithm, implemented in Simulink, into an independent ROS node utilizing the ROS Toolbox. This controller node interacts with both the vehicle dynamics model and the perception node in Gazebo in real time through predefined ROS Topic interfaces.

Tracking tests were conducted along a typical serpentine working path, as illustrated in Figure 9b. In an unstructured farmland environment, the overall average tracking error was measured at only 0.091 m, which demonstrates the exceptional robustness and adaptability of the HDC-CC architecture. The maximum error recorded was 0.297 m, occurring in the initial turning region, while subsequent turning errors exhibited a significant decreasing trend (0.231 m in the second turning region and 0.215 m in the third turning region), thereby validating the effectiveness of the delayed state observer. The proposed mechanism effectively mitigates the accumulation of historical errors by specifically compensating for temporal misalignment. The system optimizes control strategies by continuously accumulating environmental information in real-time. Notably, the delay visual error penalty term within the objective function effectively suppresses the cumulative impact of historical errors, allowing the system to persistently learn and enhance accuracy under continuous turning conditions. The test results indicate that the predictive control architecture, which integrates delay compensation and motion control co-optimization, effectively addresses the path tracking challenges encountered by visually autonomous navigation agricultural machinery in unstructured farmland environments, thereby providing a reliable technical foundation for high-precision autonomous operations.

To evaluate the practical deployment potential of the HDC-CC framework, a detailed computational resource profiling was conducted on Gazebo environment. The resource consumption of each core module is meticulously detailed in Table 3. The Model Predictive Control module, as the core algorithmic component, demonstrates the most significant computational load with a single-cycle execution time of 33.3 ms and 25.7% CPU utilization, which aligns with the computational characteristics of solving optimization problems online. The Delay State Observer and Sliding Mode Stability Controller exhibit relatively low computational demands, with execution times of 5.8 ms and 3.5 ms respectively, indicating their suitability for implementation on embedded controllers with limited computational resources. Regarding memory usage, the MPC module requires 108 MB, while the total memory footprint of the framework is 221 MB, which falls within acceptable limits for modern agricultural embedded processors. Comprehensive analysis demonstrates the HDC-CC framework's real-time implementation potential on mainstream agricultural

embedded platforms, providing a critical baseline for evaluating computational demands prior to deployment.

Table 3. Computational resource profile of the HDC-CC modules.

Component	CPU Utilization (%)	Time (ms)	Memory (MB)
Delay State Observer	4.1	5.8	38
MPC Path Tracking	25.7	33.3	108
SMC Stability Controller	3.0	3.5	15
Quadratic Programming	2.8	3.2	12
ROS Communication Overhead	6.5	7.9	48
Total (HDC-CC Framework)	42.1	53.7	221

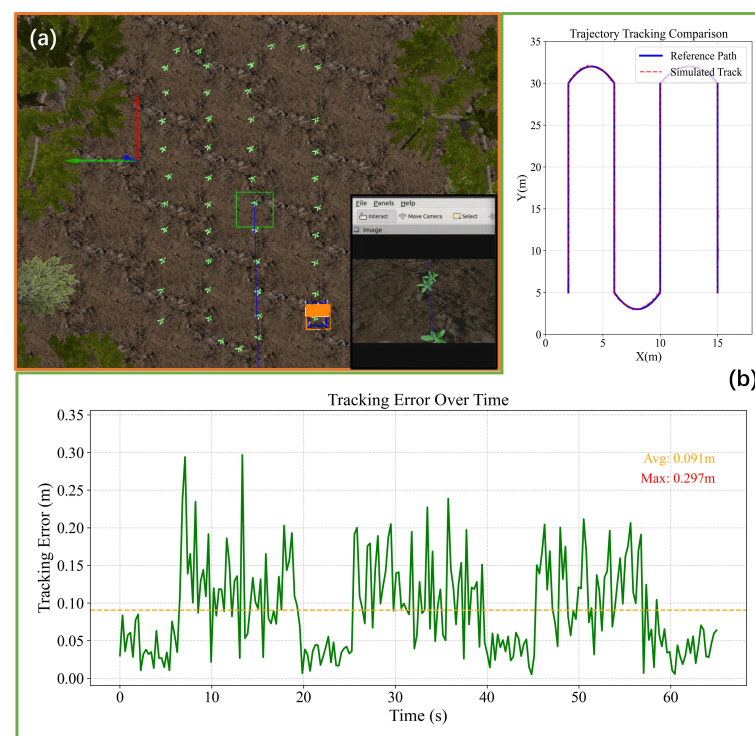


Figure 9. Virtual Farm Simulation Environment and Serpentine Path Tracking Test. (a) Gazebo-based virtual farmland environment (b) Serpentine path trajectory tracking performance.

6. Conclusions

This study presents a hierarchical delay-compensated cooperative control (HDC-CC) method for vision-based navigation of agricultural machinery, addressing the critical need for precision in field operations such as targeted spraying and weeding. This method constructs a real-time pose prediction compensator based on the forward integral of vehicle dynamics, effectively overcoming inherent delays in image processing. Experimental results demonstrate that under a visual delay of 0.4 s, the proposed method reduces the S-curve path tracking error from 0.564 m to 0.136 m, while maintaining near-zero error levels within a 0.8 s delay range. In dual-line shifting tests, the system achieves a tracking accuracy of 0.0113 m on high-friction surfaces and sustains a control error of 0.0356 m on low-friction surfaces, improving robustness by 56% over MPC. By incorporating historical-error regularization, the average tracking error is reduced by 42.1% compared to the SFP method, confirming the effectiveness of accumulated error compensation. In the Gazebo farmland simulation, an average tracking error of 0.091 m for the snake-shaped path verifies that the proposed method mitigates control mismatch caused by visual latency through

state prediction and dynamic synergy optimization. This provides a reliable technical foundation for enhancing operational accuracy and cost efficiency in precision agriculture.

Despite these promising outcomes, it is important to note that the current validation is based on simulation experiments. Future work will include real-world implementation on a dedicated agricultural robotic platform to evaluate performance under unpredictable field conditions. Additionally, optimizing the computational load of the MPC and SMC modules for deployment on embedded systems will be a critical step toward practical application.

Author Contributions: Conceptualization, Y.Z.; Methodology, Y.Z.; Formal analysis and investigation, J.L. and X.W.; Writing—original draft preparation, Y.Z.; Writing—review and editing, J.L. and Y.X.; Funding acquisition, D.H.; Resources, D.H.; Supervision, Y.X. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Key Technologies Research and Development Program of FUNDER grant number 2023YFD1500404.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

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

References

1. Liang, C.; Pan, K.; Zhao, M.; Lu, M. Multi-node path planning of electric tractor based on improved whale optimization algorithm and ant colony algorithm. *Agriculture* **2023**, *13*, 586. [\[CrossRef\]](#)
2. Wei, C.; Li, H.; Shi, J.; Zhao, G.; Feng, H.; Quan, L. Row anchor selection classification method for early-stage crop row-following. *Comput. Electron. Agric.* **2022**, *192*, 106577. [\[CrossRef\]](#)
3. Zhu, L.; Sun, W.; Zhang, Q.; Lu, E.; Xue, J.; Sha, G. Tractor Path Tracking Control Method Based on Prescribed Performance and Sliding Mode Control. *Agriculture* **2025**, *15*, 1663. [\[CrossRef\]](#)
4. Warneke, B.W.; Zhu, H.; Pscheidt, J.W.; Nackley, L.L. Canopy spray application technology in specialty crops: A slowly evolving landscape. *Pest Manag. Sci.* **2021**, *77*, 2157–2164. [\[CrossRef\]](#)
5. Cui, L.; Le, F.; Xue, X.; Sun, T.; Jiao, Y. Design and experiment of an agricultural field management robot and its navigation control system. *Agronomy* **2024**, *14*, 654. [\[CrossRef\]](#)
6. Chen, J.; Qiang, H.; Wu, J.; Xu, G.; Wang, Z. Navigation path extraction for greenhouse cucumber-picking robots using the prediction-point Hough transform. *Comput. Electron. Agric.* **2021**, *180*, 105911. [\[CrossRef\]](#)
7. Zhang, S.; Liu, Y.; Xiong, K.; Tian, Y.; Du, Y.; Zhu, Z.; Du, M.; Zhai, Z. A review of vision-based crop row detection method: Focusing on field ground autonomous navigation operations. *Comput. Electron. Agric.* **2024**, *222*, 109086. [\[CrossRef\]](#)
8. Zhang, S.; Shen, A.; Luo, X.; Tang, Q.; Li, Z. The path following of intelligent unmanned vehicle scheme based on adaptive sliding mode-model predictive control. *J. Frankl. Inst.* **2023**, *360*, 5658–5688. [\[CrossRef\]](#)
9. Dong, Z.; Xu, X.; He, S.; Wu, Z.; Xie, J.; Chen, T. Tire slip angle estimation based lateral stability control strategy for trajectory tracking scenarios of DDAEV. *Control Eng. Pract.* **2025**, *161*, 106343. [\[CrossRef\]](#)
10. Qi, J.T.; Liang, W.C.; Xie, W.B.; Peng, Y. Robust sliding mode model predictive control and thrust allocation methods for autonomous berthing of water-jet propulsion unmanned surface vehicles. *Ocean. Eng.* **2025**, *331*, 121280. [\[CrossRef\]](#)
11. Teng, F.; Wang, J.; Jin, L.; Wang, Z.; Zhou, Z.; Liu, Z. Cooperative control strategy of trajectory tracking and driving stability for distributed-drive vehicles under extreme conditions. *Control Eng. Pract.* **2024**, *147*, 105905. [\[CrossRef\]](#)
12. Nan, J.; Ge, Z.; Ye, X.; Burke, A.F.; Zhao, J. Model predictive control for autonomous vehicle path tracking through optimized kinematics. *Results Eng.* **2024**, *24*, 103123. [\[CrossRef\]](#)
13. Chen, S.; Chen, H. MPC-based path tracking with PID speed control for autonomous vehicles. In *Proceedings of the IOP Conference Series: Materials Science and Engineering*; IOP Publishing: Bristol, UK, 2020; Volume 892, p. 012034. [\[CrossRef\]](#)
14. Yang, J.; Chu, D.; Yin, J.; Pi, D.; Wang, J.; Lu, L. Distributed model predictive control for heterogeneous platoon with leading human-driven vehicle acceleration prediction. *IEEE Trans. Intell. Transp. Syst.* **2023**, *25*, 3944–3959. [\[CrossRef\]](#)
15. Mehraban, Z.; Zadeh, A.Y.; Khayyam, H.; Mallipeddi, R.; Jamali, A. Fuzzy adaptive cruise control with model predictive control responding to dynamic traffic conditions for automated driving. *Eng. Appl. Artif. Intell.* **2024**, *136*, 109008. [\[CrossRef\]](#)
16. Yin, F.; Dong, C.; Li, Y.; Chen, Y.; Wang, H. An anti-disturbance lane-changing trajectory tracking control method combining extended Kalman filter and robust tube-based model predictive control. *J. Intell. Transp. Syst.* **2025**, *29*, 319–334. [\[CrossRef\]](#)
17. Soitinaho, R.; Oksanen, T. Local navigation and obstacle avoidance for an agricultural tractor with nonlinear model predictive control. *IEEE Trans. Control Syst. Technol.* **2023**, *31*, 2043–2054. [\[CrossRef\]](#)

18. Zhou, B.; Hu, C.; Zeng, J.; Li, Z.; Betz, J.; Xie, L.; Su, H. Adaptive learning-based model predictive control strategy for drift vehicles. *Robot. Auton. Syst.* **2025**, *188*, 104941. [\[CrossRef\]](#)
19. Wang, H.; Wang, Q.; Chen, W.; Zhao, L.; Tan, D. Path tracking based on model predictive control with variable predictive horizon. *Trans. Inst. Meas. Control* **2021**, *43*, 2676–2688. [\[CrossRef\]](#)
20. Zhao, W.; Wei, H.; Ai, Q.; Zheng, N.; Lin, C.; Zhang, Y. Real-time model predictive control of path-following for autonomous vehicles towards model mismatch and uncertainty. *Control Eng. Pract.* **2024**, *153*, 106126. [\[CrossRef\]](#)
21. Zhu, J.W.; Ying, L.H.; Wang, X. TVIE-based fault-tolerant model predictive control for trajectory tracking of mobile robot. *J. Frankl. Inst.* **2024**, *361*, 107277. [\[CrossRef\]](#)
22. Ostafew, C.J.; Schoellig, A.P.; Barfoot, T.D.; Collier, J. Learning-based nonlinear model predictive control to improve vision-based mobile robot path tracking. *J. Field Robot.* **2016**, *33*, 133–152. [\[CrossRef\]](#)
23. Chu, D.; Li, H.; Zhao, C.; Zhou, T. Trajectory tracking of autonomous vehicle based on model predictive control with PID feedback. *IEEE Trans. Intell. Transp. Syst.* **2022**, *24*, 2239–2250. [\[CrossRef\]](#)
24. Liu, W.; Guo, R.; Zhao, J. Adaptive Sliding Mode Predictive Control for Path Tracking of Wheeled Agricultural Vehicles. *Machines* **2025**, *13*, 157. [\[CrossRef\]](#)
25. Zhou, B.; Su, X.; Yu, H.; Guo, W.; Zhang, Q. Research on path tracking of articulated steering tractor based on modified model predictive control. *Agriculture* **2023**, *13*, 871. [\[CrossRef\]](#)
26. Zhang, T.; Sun, Y.; Wang, Y.; Li, B.; Tian, Y.; Wang, F.Y. A survey of vehicle dynamics modeling methods for autonomous racing: Theoretical models, physical/virtual platforms, and perspectives. *IEEE Trans. Intell. Veh.* **2024**, *9*, 4312–4334. [\[CrossRef\]](#)
27. Zheng, K.; Zhao, X.; Han, C.; He, Y.; Zhai, C.; Zhao, C. Design and experiment of an automatic row-oriented spraying system based on machine vision for early-stage maize crops. *Agriculture* **2023**, *13*, 691. [\[CrossRef\]](#)
28. Quan, L.; Guo, Z.; Huang, L.; Xue, Y.; Sun, D.; Chen, T.; Geng, T.; Shi, J.; Hou, P.; He, J.; et al. Efficient extraction of corn rows in diverse scenarios: A grid-based selection method for intelligent classification. *Comput. Electron. Agric.* **2024**, *218*, 108759. [\[CrossRef\]](#)
29. Yu, J.; Zhang, J.; Shu, A.; Chen, Y.; Chen, J.; Yang, Y.; Tang, W.; Zhang, Y. Study of convolutional neural network-based semantic segmentation methods on edge intelligence devices for field agricultural robot navigation line extraction. *Comput. Electron. Agric.* **2023**, *209*, 107811. [\[CrossRef\]](#)
30. Wang, S.; Su, D.; Jiang, Y.; Tan, Y.; Qiao, Y.; Yang, S.; Feng, Y.; Hu, N. Fusing vegetation index and ridge segmentation for robust vision based autonomous navigation of agricultural robots in vegetable farms. *Comput. Electron. Agric.* **2023**, *213*, 108235. [\[CrossRef\]](#)

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.