machines

MDPI

Article

Dual-Adaptive Preview Model Predictive Control for Path Tracking of Differential-Drive Tracked Vehicles

Wei Su [1], Daolei Duan [1], Qinghui Lai [2,*] and Faan Wang [1]

[1] Faculty of Modern Agricultural Engineering, Kunming University of Science and Technology, No. 727 Jingming South Road, Chenggong District, Kunming 650500, China; laisubo@163.com (W.S.); duandaolei26@163.com (D.D.)
[2] School of Energy and Environment Science, Yunnan Normal University, No. 768 Juxian Street, Chenggong District, Kunming 650500, China
* Correspondence: laiginghui007@163.com; Tel.: +86-13529280568

Abstract

This study addresses the challenges of coordinating tracking accuracy and steering control stability in agricultural differential-drive tracked vehicles, particularly their adaptability to varying working conditions. An adaptive preview Model Predictive Control (MPC) path-tracking algorithm is developed using an electronically controlled modified differential-drive tracked vehicle as the test platform. A preview MPC controller is constructed based on kinematic analysis and a vehicle preview tracking error model, with a fuzzy controller optimizing the objective function's weight matrix. A speed- and path-dependent preview-distance selection criterion is formulated, and an improved Sparrow Search Algorithm (SSA) provides an optional numerical implementation of preview-distance selection. The corresponding fixed-state objective admits a closed-form analytical solution. A tracked vehicle model was built based on a Robot Operating System (ROS), and a path tracking controller was built based on Simulink. Simulations and field tests at 0.5, 1.0, and 1.5 m/s evaluate lateral tracking error and steering angular-velocity statistics against separate fixed-preview baselines. Relative to the 3.0 m baseline, mean absolute lateral error decreases by 30.13–56.00% in simulation and 46.05–72.57% in field tests. Relative to the 1.0 m baseline, lateral-error changes depend on speed, while angular-velocity standard deviation decreases by 22.23–40.67% in simulation and 24.10–46.39% in field tests. These results characterize the tested controller configurations and show a condition-dependent trade-off between lateral error and steering-response variability.

Keywords: tracked vehicle; path tracking; fuzzy control; model predictive control

Check for updates

Academic Editor: Zafer Bingul

Received: 26 September 2026
Revised: 27 October 2026
Accepted: 28 October 2026
Published: 4 November 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

In recent years, autonomous navigation technology has been widely applied in agricultural production, enhancing production efficiency and operational quality while reducing labor force [1–3]. The navigation technology for intelligent agricultural equipment primarily involves positioning and attitude measurement, path planning, tracking control, and collaborative transportation [4–6]. One of the core aspects of agricultural machinery navigation technology is path tracking, as the effectiveness of path tracking directly determines the quality of navigation control [7,8]. Therefore, research on path tracking control algorithms for agricultural machinery is particularly important.

Currently, research on agricultural machinery path tracking control primarily focuses on wheeled vehicles [9], with algorithms encompassing MPC, Proportional-Integral-

Derivative (PID) control, fuzzy control, pure pursuit model control, preview tracking model control, and others [10–19]. HE et al. [20] addressed the difficulty of tracking and controlling agricultural machinery during field operations by establishing an MPC method based on the machinery's attitude, using a linear model, objective function, and constraint function. This approach effectively inhibited lateral positional deviations. WANG et al. [21] developed a self-adjusting look-ahead pure pursuit controller for high-clearance sprayers, selecting the preview distance through predicted lateral and heading errors. They evaluated the approach in simulations and field tests against fixed-look-ahead control. JI et al. [22–25] tackled the issues of external disturbances and model parameter uncertainties in path tracking for unmanned agricultural tractors in unstructured environments by using a disturbance observer to estimate the aggregated disturbances in sliding mode dynamics. They established an improved composite nonsmoothed Second-Order Sliding Mode control by combining disturbance feedforward compensation and nonsmoothed feedback, achieving effective path tracking control for agricultural tractors. CHENG et al. [26] designed a tracking system based on heading deviation angle to improve the tracking accuracy of unmanned tractors and developed an MFAPC tracking system. By adding adaptive proportional and derivative terms, they obtained an MFAPC-PID controller. Simulations and experiments demonstrated the algorithm's superior overall performance, adaptability, versatility, and robustness.

Most of the aforementioned path tracking control technologies are based on wheeled agricultural machinery equipment, and there is relatively little research on path tracking algorithms for tracked vehicles. Tracked vehicles exhibit superior passing performance in complex driving environments in hilly and mountainous areas. ZHANG et al. [27] addressed the issue of low control accuracy for tracked vehicles by proposing a state feedback navigation control system based on a virtual Ackermann steering model, which achieved stable path tracking for tracked vehicles in specific orchard environments. HU et al. [28] developed a differential braking model for tracked vehicles under single-side braking conditions and designed an improved pure pursuit path tracking control method, enhancing the path tracking performance of tracked vehicles in agricultural environments. ZHANG et al. [29] proposed a control strategy combining Nonsingular Fast Terminal Sliding Mode (NFTSM) and Finite-Time Disturbance Observer (FDO), which improved the tracking performance of agricultural tracked tractors on straight parallel tracks and curved tracks, considering vehicle slip. HE et al. [30] established a turning model by extracting the turning radius of vehicles on different plots and improved the pure pursuit algorithm through fuzzy control. They designed a two-arc path tracking control method, enabling precise tracking of tracked combine harvesters under various geological conditions. Ruslan et al. reviewed vehicle models and path-tracking strategies for autonomous tracked vehicles and discussed the trade-offs between model complexity and control applicability [31–33]. Focusing on agricultural tracked robots, Zhao et al. developed a slip-aware kinematic model and used an improved sliding-mode observer to estimate slip parameters for future-trajectory prediction [34]. These works indicate that an appropriate model should reflect the intended operating conditions and that unmodeled slip must be acknowledged when a simplified kinematic model is used.

Maneuverability analysis provides a complementary perspective because the commanded speed or torque distribution can affect the actual turning radius and yaw response, and consequently the trajectory followed by a skid-steering platform. Cosenza et al. developed a multibody and multiphysics model of a six-wheeled skid-steering robot to compare alternative wheel-speed allocation strategies and examine how vehicle dynamics influence the planned trajectory [35]. For skid-steering unmanned ground vehicles, Guo et al. proposed a coordinated path-tracking and energy-efficiency strategy in which an upper-level

model-free sliding-mode controller generates the desired yaw moment and a lower-level optimizer distributes drive-motor torque [36]. These studies address the close relationship between maneuverability, steering inputs, and vehicle response, although the former emphasizes multibody analysis and wheel-speed allocation, whereas the latter also considers energy consumption. Wang et al. also established a kinematic state-space model and an MPC-based trajectory-tracking controller for unmanned ground tracked vehicles, with experiments under different driving conditions [37]. Collectively, these studies show that both vehicle modeling and control architecture must be considered when developing a tracking controller for tracked platforms. Zeng et al. proposed an improved nonlinear MPC method for an agricultural tracked vehicle, extending an ICR-based kinematic model to include vehicle roll and pitch, estimating time-varying slip parameters online, and incorporating trajectory-curvature feedforward into the controller [38]. Liu et al. developed a slip-aware MPC approach for tracked agricultural vehicles operating in hilly fields; their method estimates left- and right-track slip ratios using a sliding-mode observer and adjusts the prediction horizon according to vehicle speed and path curvature, while also penalizing velocity variation to promote smoother control [39].

However, the aforementioned algorithms did not comprehensively consider both tracking accuracy and steering control stability for tracked vehicles during field operations. The specific contribution of this study is not the isolated use of fuzzy control, MPC, preview tracking, or the Sparrow Search Algorithm (SSA), each of which has been studied previously. Rather, the controller contains two distinct adaptation mechanisms that act at different levels. First, a fuzzy controller uses the current lateral and heading deviations to adjust the MPC cost weights associated with tracking errors and steering angular velocity. This mechanism changes the trade-off between path-following accuracy and steering behavior without, by itself, changing the preview target. Second, an improved SSA selects the preview distance using vehicle speed, the curvature of the preview region, and current tracking performance. The optional SSA implementation uses a chaotic discoverer update and fuzzy c-means clustering for part of the scavenger population. In the combined controller, the fuzzy module adjusts MPC weights, while preview selection adapts the look-ahead distance. This separation makes the role of each component explicit and distinguishes the proposed integrated controller from methods that adapt only the preview target or use a preview-based MPC formulation without the same dual-adaptation structure.

This study focuses on a preview-error-based MPC controller developed using the kinematic model of a differential-drive tracked vehicle. The fuzzy adaptive-weighting mechanism and the improved-SSA-based preview-distance selection method are incorporated into the controller to coordinate the optimization of tracking errors, steering behavior, and preview distance. The proposed strategy is evaluated through joint ROS/Gazebo–Simulink simulations and real-vehicle experiments at different operating speeds. Its performance is assessed by comparing lateral tracking errors and steering angular velocity with those obtained using fixed-preview-distance MPC controllers. The results are used to examine the effectiveness of the dual-adaptive strategy under different path geometries and operating conditions and to determine whether the coordinated adjustment of cost-function weights and preview distance improves the overall path-tracking performance of differential-drive agricultural tracked vehicles.

2. Materials and Methods

2.1. Kinematics Model of Agricultural Differential-Speed Tracked Vehicles

The agricultural differential-speed tracked vehicle achieves steering control by relying on the speed difference between the two sides of its tracks. Considering the low traveling speed of the vehicle in farmland operating environments, most of the dynamic influencing

factors can be neglected. Based on the following assumptions, a kinematics model for the agricultural differential-speed tracked vehicle is established:

- The centroid of the tracked vehicle coincides with its geometric center.
- The belt speed of the tracks remains constant during steering.
- The nominal kinematic model assumes negligible longitudinal track slip and lateral skidding; this assumption is a modeling approximation and is not guaranteed solely by low vehicle speed.

Model scope and slip assessment. Let vl and vr denote the commanded track speeds and sL and sR the corresponding effective longitudinal slip ratios. A first-order slip-aware comparison uses $v_{L,\text{eff}} = (1 - s_L)v_L$ and $v_{R,\text{eff}} = (1 - s_R)v_R$, with $v_{\text{eff}} = \frac{v_{R,\text{eff}} + v_{L,\text{eff}}}{2}$ and $\omega_{\text{eff}} = \frac{v_{R,\text{eff}} - v_{L,\text{eff}}}{L_{\text{eff}}}$. The nominal model follows by setting $s_L = s_R = 0$ and $L_{\text{eff}} = L$. These expressions do not explicitly model lateral skidding or terrain deformation. The validity of the nominal approximation should therefore be assessed against independent RTK/IMU measurements by reporting translational- and yaw-rate prediction residuals over straight and turning segments.

A kinematics model for the agricultural differential-speed tracked vehicle is established in the ground coordinate system XOY, as shown in Figure 1. The symbols used in the model are defined in Table 1.

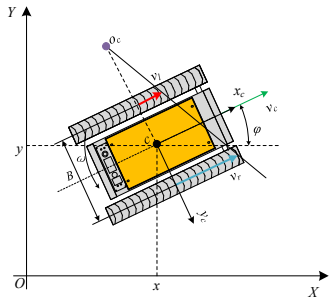


Figure 1. Kinematics model of agricultural differential-speed tracked vehicle.

Table 1. Symbolic meaning.

Symbol	Meaning
c	Geometric center of the vehicle
L	Distance between track centers
ω	Theoretical steering angular velocity
oc	Instantaneous center of turning
vl	Linear velocity of the left track during steering
vr	Linear velocity of the right track during steering
vc	Velocity of the vehicle's center of mass
xc	Horizontal axis of the vehicle coordinate system
yc	Vertical axis of the vehicle coordinate system

In the ground coordinate system, the vehicle state vector is denoted as X (x, y, θ), where x and y represent the position coordinates of the vehicle in the XOY coordinate system, and θ represents the vehicle's heading angle. The kinematic equations for an agricultural differential-drive tracked vehicle are then given as follows

$$\begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{\theta}(t) \end{bmatrix} = \begin{bmatrix} v(t)cos(\theta(t)) \\ v(t)sin(\theta(t)) \\ \omega(t) \end{bmatrix} = \begin{bmatrix} cos(\theta((t)) & 0 \\ sin(\theta(t)) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v(t) \\ \omega(t) \end{bmatrix} \quad (1)$$

The speeds of the left and right drive wheels of the tracked vehicle are as follows [40].

$$v_l(t) = v_c(t) - \frac{\omega(t)L}{2} \tag{2}$$

$$v_r(t) = v_c(t) + \frac{\omega(t)L}{2} \tag{3}$$

2.2. Preview Tracking Error Model

As shown in Figure 2, when a tracked vehicle performs path tracking based on the preview tracking model, it searches the reference path with a certain preview distance Ld to determine an appropriate target point to follow. The perpendicular distance from the vehicle's position to the reference trajectory is defined as the lateral deviation, while the deviation between the vehicle's current heading and the preview direction is defined as the heading deviation. By controlling the differential steering of the motors, the vehicle's position and orientation are continuously adjusted to reduce these deviations, enabling the tracked vehicle to follow the predetermined trajectory strictly.

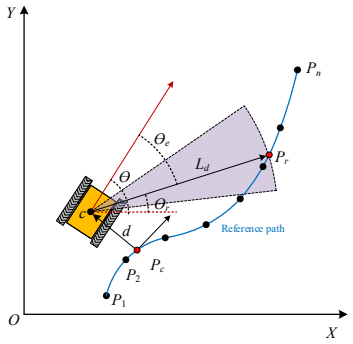


Figure 2. The preview tracking error model for tracked vehicles.

When the tracked vehicle actually performs path tracking, the reference trajectory is typically composed of a series of discrete coordinate points P1, P2, ..., Pn, as shown in Figure 2. For discrete trajectories, due to the absence of target points along the path, it is not feasible to project the vehicle's current position c onto the trajectory for analysis. Therefore, approximate solutions must be sought using neighboring target points. By estimating the distances from the vehicle's current position c to each discrete point on the reference path one by one, the trajectory discrete point that is closest to the preview distance from the projected point Pc is determined as the preview point, denoted as $Pr(x_r\ (t), y_r(t))$. Consequently, the error of the tracked vehicle in the global coordinate system is given by

$$x_e(t) = x_r(t) - x(t) \tag{4}$$

$$y_e(t) = y_r(t) - y(t) \tag{5}$$

$$\theta_e(t) = \theta_r(t) - \theta(t) \tag{6}$$

where the vehicle's reference heading θ_r is

$$\theta_r(t) = atan2(y_r(t) - y_c(t), x_r(t) - x_c(t)) \tag{7}$$

The kinematic model of the tracked vehicle can be simplified as follows.

$$\dot{X} = f(X(t), U(t)) \tag{8}$$

The kinematic model of the tracked vehicle is expanded using a Taylor series in the vicinity of the preview point $P_r(x_r(t), y_r(t))$.

$$\dot{X} = f(X_r(t), U_r(t)) + J_X(X(t) - X_r(t)) + J_U(U(t) - U_r(t)) \tag{9}$$

where $X_r(t) = [x_r(t)\ y_r(t)\ \theta_r(t)]^T$ denotes the preview reference state, $U_r(t) = [v_r(t)\ \omega_r(t)]^T$ denotes a nominal reference input, and Ar and Br are the Jacobians of f with respect to X and U, evaluated at (X_r, U_r). A preview target generated from the moving vehicle pose need not itself follow the nominal kinematics exactly; its reference-dynamics residual is $d_r = \dot{X}_r - f(X_r, U_r)$.

The preview tracking error model for tracked vehicles is as follows.

$$\dot{E}_g(t) = A_r(t)E_g(t) + B_r(t)\tilde{U}(t) + d_r(t) - d_{slip}(t) \tag{10}$$

With the sign convention of Equations (4)–(6), $E_g(t) = X_r(t) - X(t) = [\mathrm{x_e(t)}\ \mathrm{y_e(t)}\ \theta_e(t)]^{\mathrm{T}}$. The heading error is defined as the reference heading minus the vehicle heading and wrapped to the interval $\theta_e = \mathrm{mod}(\theta_r - \theta + \pi, 2\pi) - \pi \in [-\pi, \pi)$, all heading angles and heading errors are expressed in radians, and the input-error vector is $\tilde{\mathrm{U}}(\mathrm{t}) = \mathrm{Ur(t)} - \mathrm{U(t)}$. The wrapped heading error uses $\theta_e = \mathrm{wrap}(\theta_r - \theta)$. The additive term d_{slip} denotes the discrepancy between actual vehicle motion and the nominal no-slip model. Equation (10) reduces to the nominal linear error model only when both d_r and d_{slip} are neglected.

$$A_r = \begin{bmatrix} 0 & 0 & -v_r(t)sin(\theta_r(t)) \\ 0 & 0 & v_r(t)cos(\theta_r(t)) \\ 0 & 0 & 0 \end{bmatrix}, \quad B_r = \begin{bmatrix} cos(\theta_r(t)) & 0 \\ sin(\theta_r(t)) & 0 \\ 0 & 1 \end{bmatrix} \tag{11}$$

Using the approximate discrete method to discretize Equation (10). The forward-Euler discretization of Equation (10) is given below. The displayed MPC prediction model treats d_r and d_{slip} as unmodeled residuals; this is an approximation, not proof that their values vanish.

$$E(k+1) = A_d(k)E_g(k) + B_d(k)\tilde{U}(k) + T_s\left(d_r(k) - d_{\mathrm{slip}}(k)\right) \tag{12}$$

where

$$A_d(k) = I_3 + T_s A_r(k) \tag{13}$$

$$B_d(k) = T_s B_r(k) \tag{14}$$

For the nominal state-error subsystem, $n = 3$, $\mathrm{m} = 2$, I_n is the 3×3 identity matrix and T_s is the sampling period. Accordingly $A_d \in \mathrm{R}^{\{3\times3\}}$ and $B_d \in \mathrm{R}^{\{3\times2\}}$.

A discrete-time configuration uses a supervisory period of $T_s = 0.10\ s$, prediction horizon $N_P = 10$, and control horizon $N_C = 3$, corresponding to prediction and control windows of 1.0 and 0.3 s. In this configuration, the kinematic model is discretized at the stated period, and the local prediction matrices are evaluated at each control update.

3. Controller Design

3.1. Design of a Path Tracking Controller

The objective of path tracking control is to enable the tracked vehicle to follow a reference trajectory quickly and stably. In this section, an MPC path tracking controller is established based on an adaptive preview tracking error model, and adaptive adjustment of the weight matrix is achieved through fuzzy control. The control architecture is shown in Figure 3.

Taking into careful consideration the preview error model and system control inputs, the discrete state-space representation for the tracked vehicle, with control increments as inputs, is given as follows

$$\begin{cases} \xi(k+1) = \tilde{\partial}_k \xi(k) + \tilde{\gamma}_k \triangle U(k) \\ \eta(k) = \tilde{C}\xi(k) \end{cases} \tag{15}$$

where $\xi(k) = \begin{bmatrix} E(k) \\ \tilde{U}(K-1) \end{bmatrix}$, $\tilde{\partial}_k = \begin{bmatrix} \partial(k) & \gamma(k) \\ 0_{m\times n} & \mathbf{I}_m \end{bmatrix}$, $\tilde{\gamma}_k = \begin{bmatrix} \gamma(k) \\ \mathbf{I}_m \end{bmatrix}$, $\tilde{C} = [\mathbf{I}_{n\times n} 0_{n\times m}]$, Im is the m×m identity matrix, 0 m×n is the m×n matrix of all zeros.

In Equation (15), $E(k) \equiv E_g(k)$, $\partial(k) \equiv A_d(k)$, *and* $\gamma(k) \equiv B_d(k)$, where A_d and B_d are defined in Equations (13) and (14). Accordingly, $\xi(k) \in \mathbb{R}^5$, $\tilde{\partial}_k \in \mathbb{R}^{5\times5}$, $\tilde{\gamma}_k \in \mathbb{R}^{5\times2}$, *and* $\tilde{C} \in \mathbb{R}^{3\times5}$. The second block of $\xi(k)$ contains the previous input-error vector $\tilde{U}(k-1)$, rather than the previous physical command. The augmented model is the nominal predictor obtained by neglecting the reference-motion and slip residuals in Equation (12).

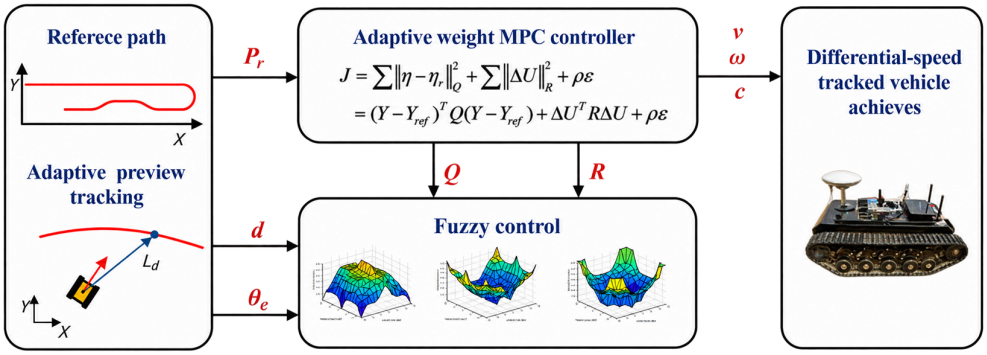


Figure 3. Adaptive preview MPC control architecture.

3.1.1. Predicted Output

The MPC controller utilizes Equation (15) to predict states and outputs, meaning it forecasts the system's state variables and outputs within the prediction horizon N_P. Within the control horizon N_C, which is shorter than the prediction horizon ($N_P > N_C$), it optimizes the increments of the control variables through a rolling optimization process.

For the prediction horizon N_P and control horizon N_C ($1 \le N_C \le N_P$), the stacked predicted-output vector has dimension $3N_P$, and the stacked input-error increment vector has dimension $2N_C$. Beyond the control horizon, increments are set to zero for prediction purposes.

$$Y(k) = [\eta(k+1)\eta(k+2) \ldots \eta(k+N_P)]^{\mathrm{T}} \tag{16}$$

$$\Delta U_{N_C}(k) = [\Delta U(k)\ \Delta U(k+1)\ \ldots\ \Delta U(k+N_C-1)]^T \tag{17}$$

Since $\eta(k) = \tilde{C}\xi(k) = E(k)$, Y(k) stacks predicted tracking errors, not absolute global states. Therefore $Y_{ref}(k) = 0_{3N_P}$. The global reference sequence is used to form $E(k+i) = X_r(k+i) - X(k+i)$ and is not subtracted from the predicted error a second time. To retain the input-error augmented state in Equation (15), $\Delta U(k) = \tilde{U}(k) - \tilde{U}(k-1)$ denotes the input-error increment in Equations (15)–(23), with $\tilde{U} = U_r - U$. The matrices in Equation (18) have sizes $\Psi \in \mathbb{R}^{3N_P\times5}$, $\varpi \in \mathbb{R}^{3N_P\times2N_C}$ and are held constant over each nominal prediction horizon.

$$Y(k) = \Psi_k \xi(k) + \varpi_k \Delta U_{N_C}(k) \tag{18}$$

where

$$\Psi_k = col\left(\tilde{C}\tilde{\partial}_k,\ \tilde{C}\tilde{\partial}_k^{\,2},\ \ldots,\ \tilde{C}\tilde{\partial}_k^{\,N_P}\right)$$

$$\varpi_{k,ij} = \begin{cases} \tilde{C}\tilde{\partial}_k^{\,i-j}\tilde{\gamma}_k & i \ge j \\ 0_3\times_2 & i < j \end{cases},\ i = 1,\ldots,N_P,\ j = 1,\ldots,N_C$$

3.1.2. Objective Function and Adaptive Control of Target Weights

To achieve rapid and stable path tracking for agricultural differentially driven tracked vehicles, the objective function is designed as follows.

$$J = \left(Y - Y_{ref}\right)^T Q\left(Y - Y_{ref}\right) + z^T R z + \rho\varepsilon^2 \tag{19}$$

where $z = \Delta U_{N_C}(k)$, Q weights the predicted tracking errors and R weight the input-error increments. The zero error is $Y_{ref} = 0$. The slack coefficient is ρ; the existing hard-constrained formulation fixes $\varepsilon = 0$.

$$\begin{cases} Q = blkdiag\left(Q_1, \ldots, Q_{N_P}\right) \in \mathbb{R}^{3N_P \times 3N_P} \\ Q_n = diag\left(Q_{xe}, Q_{ye}, Q_{\theta e}\right),\ n = 1, \ldots, N_P \end{cases} \tag{20}$$

$$\begin{cases} R = blkdiag\left(R_0, \ldots, R_{N_C-1}\right) \in \mathbb{R}^{2N_C \times 2N_C} \\ R_i = diag\left(R_{\Delta v}, R_{\Delta\omega}\right),\ i = 0, \ldots, N_C - 1 \end{cases} \tag{21}$$

Let $z = \Delta U_{N_C}(k)$, $A = \omega_k{}^T Q \omega_k + R$, $c = E_0{}^T Q E_0$. Substitution of Equation (18) gives $J = z^T A z + 2E_0{}^T Q \omega_k z + c + \rho\varepsilon^2$. With $\varepsilon = 0$, dropping the constant c gives the equivalent QP objective $\varphi(z) = z^T H z + f^T z$, where $H = 2A$, $f = 2\omega_k{}^T Q E_0$. Equation (22) writes the same cost with the constant omitted;

$$J_{qp} = \begin{bmatrix} z \\ \varepsilon \end{bmatrix}^T \begin{bmatrix} \omega_k{}^T Q \omega_k + R & 0 \\ 0 & \rho \end{bmatrix} \begin{bmatrix} z \\ \varepsilon \end{bmatrix} + 2\begin{bmatrix} E_0{}^T Q \omega_k & 0 \end{bmatrix} \begin{bmatrix} z \\ \varepsilon \end{bmatrix} \tag{22}$$

where

$$E_0(k) = \Psi_k \xi(k) - Y_{ref} \tag{23}$$

In Equations (22) and (23), E_0 is the predicted output error before applying the future control increments, rather than merely the predicted output. Since Equations (25)–(27) specify hard actuator bounds only, the slack variable mentioned in Equation (19) is fixed to $\varepsilon = 0$ in this hard-constrained formulation. The algorithm implements adaptive MPC weight control based on fuzzy control. When a tracked vehicle performs path tracking using an MPC controller, the main factors affecting its tracking performance include Q_{ye}, $Q_{\theta e}$, and $R_{\Delta\omega}$. The weight adjustment formula is as follows.

$$\begin{cases} Q_{ye} = Q_{yg} \times 6\mu_y \\ Q_{\theta e} = Q_{\theta g} \times 5\mu_\theta \\ R_{\Delta\omega} = R_{\omega g} \times 3\mu_\omega \end{cases} \tag{24}$$

Cost-function parameterization. In the reported tracking experiments, the baseline weight coefficients in Equation (24) were $Q_{yg} = 300$, $Q_{\theta g} = 400$, and $R_{\omega g} = 5$. The fuzzy controller adjusted the weighting factors μ_y, μ_θ, and μ_ω within $[0, 1]$. Accordingly, the adaptive weights were calculated as $Q_{ye} = 1800\mu_y$, $Q_{\theta e} = 2000\mu_\theta$, $R_{\Delta\omega} = 15\mu_\omega$. For the output order $[x_e, y_e, \theta_e]^T$ and increment order $[\Delta v, \Delta\omega]^T$, the stage weighting matrices were $Q_{\text{stage}} = \text{diag}\left(0, 1800\mu_y, 2000\mu_\theta\right)$, $R_{\text{stage}} = \text{diag}(0, 15\mu_\omega)$. The corresponding horizon weighting matrices were constructed as $Q = I_{N_P} \otimes Q_{\text{stage}}$ and $R = I_{N_C} \otimes R_{\text{stage}}$. The fixed-preview baselines and the adaptive-preview controller used the same fuzzy weight-adaptation law and the baseline coefficients specified in Equation (24). Using the same adaptation law does not, by itself, imply identical instantaneous weight histories, because the law depends on tracking errors.

In the fuzzy controller, the lateral deviation and heading deviation of the tracked vehicle are selected as input parameters, while the weight factor coefficients μ_y, μ_θ, and μ_ω serve as output parameters. A dual-input and triple-output fuzzy controller is established, as shown in Figure 4.

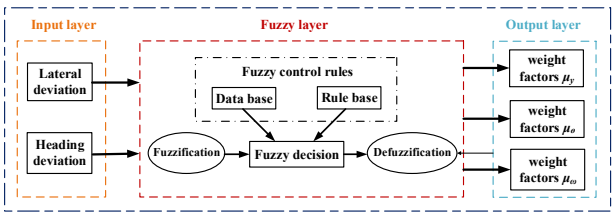


Figure 4. The dual-input and triple-output fuzzy controller.

The dual-input and triple-output fuzzy controller mainly consists of three parts: fuzzification of variables, determination of membership functions, and formulation of fuzzy control rules.

Based on the kinematic state of the differential tracked vehicle and operational experience, the basic universe of discourse for the lateral deviation is determined to be $[-15, 15]$, the basic universe of discourse for the heading deviation is $[-12, 12]$, and the basic universe of discourse for the weighting factor coefficients μ_y, μ_θ, and μ_ω is $[0, 1]$. The fuzzy subsets for the lateral deviation and heading deviation consist of 7 levels: *NB* (Negative Big), *NM* (Negative Medium), *NS* (Negative Small), *ZO* (Zero), *PB* (Positive Big), *PM* (Positive Medium), and *PS* (Positive Small). The fuzzy subsets for the weighting factors μy and μθ consist of 4 levels: *PB* (Positive Big), *PM* (Positive Medium), *PS* (Positive Small), and *ZO* (Zero). The fuzzy subsets for the weighting factor coefficient μ_ω consist of 3 levels: *S* (Small), *M* (Medium), and *B* (Big). The results of the fuzzification process are shown in Table 2.

Table 2. The fuzzification parameters for the input and output variables of the control system.

Relevant Parameters.	Basic Domain	Fuzzy Subset	Quantification Factor
Lateral deviation	$[-15, 15]$	$\{-3, -2, -1, 0, 1, 2, 3\}$	0.2
Heading deviation.	$[-12, 12]$	$\{-3, -2, -1, 0, 1, 2, 3\}$	0.25
Weight factors μ_y	$[0, 1]$	$\{0, 1, 2, 3\}$	3
Weight factors μ_θ	$[0, 1]$	$\{0, 1, 2, 3\}$	3
Weight factors μ_ω	$[0, 1]$	$\{1, 2, 3\}$	3

Determination of Membership Functions. Triangular membership functions are characterized by low computational complexity and high flexibility. Therefore, triangular membership functions are adopted. Membership-function specification for a reference fuzzy inference system. Triangular membership functions are defined by tri(x;a,b,c) = max(0,min((x−a)/(b−a),(c−x)/(c−b))), with left and right shoulder functions used at the domain endpoints. Seven uniformly spaced input terms *NB*, *NM*, *NS*, *ZO*, *PS*, *PM*, and *PB* have centers −15, −10, −5, 0, 5, 10, and 15 for lateral deviation and −12, −8, −4, 0, 4, 8, and 12 for heading deviation; adjacent centers define the corresponding triangle bases. The four output terms *ZO, PS, PM*, and *PB* for μ_y and μ_θ have centers 0, 1/3, 2/3, and 1; the three terms *S, M*, and *B* for μω have centers 0, 0.5, and 1. Mamdani min–max inference and centroid defuzzification provide an explicit reference to the rules in Tables 3–5.

Fuzzy Control Rule TABLE. Establish fuzzy control laws for the differential tracked vehicle's μy, μθ, and μω, as shown in Tables 3–5, respectively.

The input-output fuzzy response surface is shown in Figure 5.

Table 3. Fuzzy control rule TABLE for weighting factor coefficient μ_y.

μ_y		Lateral Deviation						
		NB	**NM**	**NS**	**ZO**	**PS**	**PM**	**PB**
Heading deviation	NB	ZO	PS	PS	PM	PS	PS	ZO
	NM	PS	PS	PM	PM	PM	PS	PS
	NS	PS	PM	PM	PB	PM	PM	PS
	ZO	ZO	PM	PM	PB	PM	PM	ZO
	PS	PS	PM	PM	PB	PM	PM	PS
	PM	PS	PS	PM	PM	PM	PS	PS
	PB	ZO	PS	PM	PM	PM	PS	ZO

Table 4. Fuzzy control rule TABLE for weighting factor coefficient μ_θ.

μ_θ		Lateral Deviation						
		NB	**NM**	**NS**	**ZO**	**PS**	**PM**	**PB**
Heading deviation	NB	PB	PM	PM	PS	PM	PM	PB
	NM	PM	PM	PS	PS	PS	PM	PM
	NS	PM	PS	PS	ZO	PS	PS	PM
	ZO	PB	PS	PS	ZO	PS	PS	PB
	PS	PM	PS	PS	ZO	PS	PS	PM
	PM	PM	PM	PS	PS	PS	PM	PM
	PB	PB	PM	PS	PS	PS	PM	PB

Table 5. Fuzzy control rule table for weighting factor coefficient μ_ω.

μ_ω		Lateral Deviation						
		NB	**NM**	**NS**	**ZO**	**PS**	**PM**	**PB**
Heading deviation	NB	B	B	M	M	M	B	B
	NM	B	M	M	S	M	M	B
	NS	M	M	S	S	S	S	M
	ZO	M	S	S	S	S	S	M
	PS	M	S	S	S	S	M	M
	PM	B	M	M	S	M	M	B
	PB	B	B	M	M	M	B	B

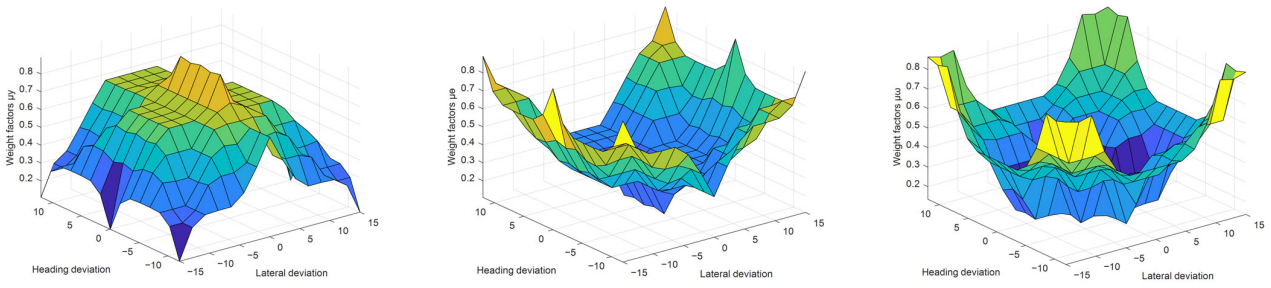


Figure 5. Fuzzy output response surface.

3.1.3. Receding Horizon Optimization and Feedback Control

Considering the steering control capability of the tracked vehicle and the safety of path tracking, constraints are imposed on the control inputs and their increments, as shown in Equation (25).

$$\begin{cases} U_{min} \le U(k) \le U_{max} \\ \Delta U_{min} \le \Delta U(k) \le \Delta U_{max} \end{cases} \tag{25}$$

in Equation (25), U and ΔU are physical commands and their increments. In Equation (27), the bounds are repeated NC times. Let $t_e = 1_{N_C} \otimes \tilde{U}(k-1)$, $S = tril(1_{N_C \times N_C}) \otimes I_2$. Let $U_{r,c}$ stack the reference inputs and $\Delta U_{r,c}$ stack their time increments, including the first increment from k − 1 to k. Then the physical command and increment sequences are $U_{cmd,c} = U_{r,c} - t_e - Sz$ and $\Delta U_{cmd,c} = \Delta U_{r,c} - z$. Only the first move is applied using $U_{cmd}(k) = U_r(k) - \tilde{U}(k-1) - z_1$, where z1 is the first two-component block of z.

The control problem of the preview tracking error model is transformed into a standard quadratic programming problem, as shown in Equation (26).

$$min_z \, \varphi(z) = z^T H z + f^T z \tag{26}$$

$$\begin{cases} U_{min,c} \le U_{r,c} - t_e - Sz \le U_{max,c} \\ \Delta U_{min,c} \le \Delta U_{r,c} - z \le \Delta U_{max,c} \end{cases} \tag{27}$$

A comparison of the mathematical formulation in Equations (10)–(23), together with the command reconstruction and constraints in Equations (25)–(27), with the MATLAB/Simulink implementation used in the simulations and vehicle experiments con-firms their consistency. The reported tracking trajectories and lateral-error and angu-lar-velocity statistics were obtained using this implementation. The reported percentage reductions are calculated directly from the corresponding tracking statistics.

3.2. Adaptive Preview Distance

When the differential-drive tracked vehicle employs an MPC algorithm with preview for path tracking, as illustrated in Figure 6, the preview distance is a core parameter that significantly influences the control performance. As shown in Figure 6b, when the preview distance L_d is too small, the tracked vehicle will track the reference path more precisely, but the path tracking controller may experience tracking oscillation issues, leading to instability in the vehicle's motion. Conversely, a larger preview distance avoids excessive path tracking oscillation, as depicted in Figure 6c. However, when the preview distance exceeds a certain threshold, it increases the vehicle's lateral error and reduces control accuracy, resulting in the cutting-corner phenomenon shown in Figure 6d. Therefore, determining an appropriate preview distance is crucial for the path tracking performance of tracked vehicles.

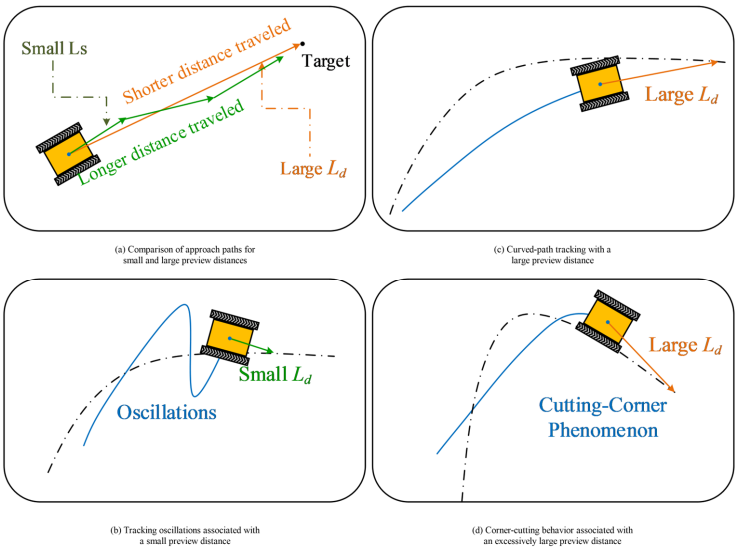


Figure 6. Effects of preview distance on tracked-vehicle path tracking.

3.2.1. Speed-Based Reference Preview Distance

A longer preview distance can be considered at higher vehicle speed, while a shorter distance provides more local path information. The linear speed-based reference below is defined for $v_{min} \leq v \leq v_{max}$, with $v_{max} > v_{min}$. It is a reference value, not an inequality constraint on the selected preview distance.

$$L_{d,ref} = L_{dmin} + \frac{L_{dmax} - L_{dmin}}{v_{max} - v_{min}}(v - v_{min}) \tag{28}$$

Equation (28) is not imposed as $L_d \geq L_{d,ref}$. The admissible preview interval remains $L_{dmin} \leq L_d \leq L_{dmax}$ (1.0–3.0 m). In the implemented controller, the reference from Equation (28) is computed from the current vehicle speed and used when initializing the feasible candidate preview distances in the improved SSA. Subsequent candidate updates and fitness evaluations follow Equations (31)–(39) within the same admissible interval. Equation (39) determines the target from path geometry and absolute lateral deviation; it contains no explicit speed term. Thus, speed enters the search initialization, without changing the fixed-state objective or its analytical optimum. The selected distance can be smaller than the initialization reference.

3.2.2. Degree of Path Curvature in the Preview Area

The curvature of the local path within the preview area influences the selection strategy of the preview distance L_d. As the curvature of the preview path increases, the preview distance should be correspondingly reduced to enhance the accuracy of the vehicle's path tracking control. When the curvature of the preview path is relatively small, a larger preview distance should be chosen to ensure the smooth movement of the vehicle. The starting point of the preview area is selected as the discrete path point that is at a distance greater than Ldmin from the geometric centroid of the tracked vehicle, and the ending point is the discrete path point that is at a distance less than Ldmax from the centroid, as shown in Figure 7. The angle difference ρi between two line segments at discrete path points within the preview area is defined as

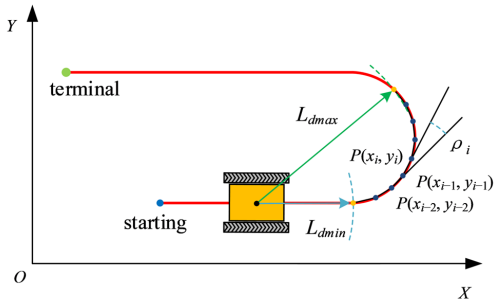


Figure 7. Curvature degree of the preview area.

Let $P_0, \ldots, P_n$ denote the ordered path points within the preview region, and let $\mathbf{t}_i = P_i - P_{i-1}$ denote the corresponding nonzero segment vectors. The local unsigned turning angle between two consecutive segments is defined by Equation (29). Thus, ρ_i characterizes the directional change at vertex P_{i-1}, rather than the accumulated turning over the preview region.

$$\rho_i = arccos\left(\frac{\boldsymbol{t}_{i-1}^T \boldsymbol{t}_i}{\|\boldsymbol{t}_{i-1}\| \, \|\boldsymbol{t}_i\|}\right)\frac{180}{\pi}, \; i = 2, \ldots, n \tag{29}$$

where i represents the ith path point within the vehicle's preview area.

The angular straightness indicator of the preview area is defined as

$$W = 180° - \sum_{i=2}^{n} \rho_i \tag{30}$$

The sum $\sum_{i=2}^{n} \rho_i$ represents the accumulated unsigned turning angle within the preview region. Equation (30) defines W as an aggregate angular straightness indicator, expressed in degrees. A straight preview path gives $W = 180°$, whereas greater accumulated turning produces a smaller W. This indicator describes regional angular geometry and is distinct from differential path curvature.

3.2.3. Sparrow Search Algorithm

The improved SSA [41] was used to select the adaptive preview distance in the reported simulations and field experiments. The analytical solution below provides an exact reference for the same fixed-state objective and was not used to generate the reported tracking results.SSA is therefore optional as a solution method for this objective, but it is the numerical implementation used in the reported tests. The SSA population comprises discoverers and scavengers, with sentinel updates. In a Z-dimensional search space, the position of individual i is $X_i = [x_{i1}, \ldots, x_{iZ}]$; for the scalar preview-distance problem, Z = 1. Candidates are ranked by decreasing fitness, and all updated distances are clipped to the admissible interval before evaluation.

The discoverers occupy the top 10–20% of the energy in the total number of the sparrow flock [42], and its position update formula is

$$x_{i,z}^{g+1} = \begin{cases} x_{\mathrm{i,z}}^{g} \cdot exp\left(\frac{-i}{R_{and} \cdot G}\right), & A_{larm} < S_f \\ x_{\mathrm{i,z}}^{g} + E \cdot M_Z, & \text{Other} \end{cases} \tag{31}$$

where g is the current iteration number of the sparrow flock, G is the maximum iteration number of the sparrow flock, $x_{i,z}^{g}$ is the z-dimensional position of the i-th sparrow in the sparrow flock at iteration number g, R_{and} is a random number distributed between $(0, 1)$, E is a normally distributed random number between $(0, 1)$, M_Z is a matrix of all ones with dimensions $1 \times Z$, A_{larm} is an early warning value within the range $[0.5, 1]$, S_f is the sparrow flock safety threshold within the range $[0.5, 1]$.

The remaining sparrows in the population act as scavengers, and the formula for updating their positions is

$$x_{\mathrm{i,z}}^{g+1} = \begin{cases} E \cdot exp(\frac{x_{\mathrm{bad,z}}^{g} - x_{\mathrm{i,z}}^{g}}{i^2}), & i > N/2 \\ x_{g,z}^{g+1} + \left|x_{\mathrm{i,z}}^{g} - x_{g,z}^{g+1}\right| + {Q_Z}^{+} \cdot M_Z, & \text{Other} \end{cases} \tag{32}$$

where $x_{bad,z}^{g}$ is the position of the sparrow with the lowest energy in the searching sparrow flock, $x_{1g,z}^{g+}$ is the position with the highest energy of the discoverers in the current population searching process, Q_Z is a $1 \times Z$-dimensional matrix, where each element is randomly -1 or 1, and ${Q_Z}^{+} = {Q_Z}^{\top}\left(Q_Z {Q_Z}^{\top}\right)^{-1}$.

Sparrows will be threatened by natural enemies during the food search process. Among them, 10–20% of sparrows will serve as sentinel sparrows. The position update formula for them is

$$x_{\mathrm{i,z}}^{g+1} = \begin{cases} x_{\mathrm{best,z}}^{g} + \rho\left|x_{i,z}^{g} - x_{best,z}^{g}\right|, & f\left(L_d\right)_i < f\left(L_d\right)_g \\ x_{i,z}^{g} + R_S \left|\frac{x_{i,z}^{g} - x_{bad,z}^{g}}{(f(L_d)_i - f(L_d)_z) + S_V}\right|, & f(L_d)_i = f(L_d)_g \end{cases} \tag{33}$$

where $x_{best,z}^{g}$ is the position of the sparrow with the highest energy in the z-dimensional position of the sparrow flock in the previous search of the algorithm, ρ is the step size control parameter during the search process, R_S is a random number between $[-1, 1]$, $f(L_d)_i$, $f(L_d)_z$,

$f(L_d)_g$ are the current, worst, and best fitness values of the population, respectively, and S_V is the safety value.

(a) Introduce chaotic factor

The essential characteristic of chaos theory is its extreme sensitivity to initial conditions, meaning that minor changes can lead to significant alterations in the system's behavior. Moreover, when the control parameter is 4, the chaos factor function transitions into a fully chaotic state [43]. At this point, the system exhibits ergodicity, and the probability density function is independent of the initial values. The update formula for the chaos factor C_f is

$$C_f = \varpi_{n+1} = 4\varpi_n(1 - \varpi_n) \tag{34}$$

where $\varpi_n \in (0, 1)$, and $\varpi_n \neq 0.25, 0.5, 0.75, 1$.

The chaotic factor C_f generated by Equation (34) is incorporated into the discoverer position-update rule. Its deterministic nonlinear evolution governs the update factor in the optional SSA implementation. This mechanism affects the numerical search trajectory, while the analytical maximizer of the fixed-state objective in Equation (38) remains unchanged. The resulting position-update rule is expressed as follows:

$$x_{i,z}^{g+1} = \begin{cases} x_{i,z}^{g} \cdot exp(\frac{-i}{C_f \cdot G}), & A_{larm} < S_f \\ x_{i,z}^{g} + E \cdot M_Z, & \text{Other} \end{cases} \tag{35}$$

(b) Population clustering

Fuzzy c-means clustering determines search centers for a subset of the scavenger population. The procedure minimizes the membership-weighted sum of squared distances in Equation (36), and the resulting centers guide the scavenger position updates in Equation (37). The clustering subproblem concerns population organization, whereas preview-distance selection maximizes the fitness in Equation (38). The effects of clustering on solution accuracy and computational cost require independent numerical evaluation.

Fuzzy c-means clustering determines the class membership of each sample point by obtaining the membership degree of each sample point to the center through optimizing the objective function, thus realizing automatic classification of sample data. Through the fuzzy c-means clustering algorithm, the top 5% of discoverers with moderate values in the sparrow population are set as the initial clustering centers, and the Euclidean distance sum is taken as the objective function. The calculation formula of the objective function J is

$$J = \sum_{i=1}^{\varepsilon} \sum_{k=1}^{\tau} (\mu_{ik})^b \|x_i - v_k\|^2 \tag{36}$$

where ε is the number of clustering samples, τ is the number of clustering centers, b is the fuzzy exponent, with $b > 1$, μ_{ik} is the membership degree of data i belonging to the k-th cluster, x_i is the i-th element, corresponding to the i-th sparrow's position, and v_k is the k-th clustering center.

During each iteration, the membership function is used to calculate the data membership, and the clustering centers v_k and membership degrees μ_{ik} are updated. After clustering is completed, the top 30% of scavengers in the population will search near the clustering centers as targets. Their search formula is

$$x_{i,z}^{g+1} = \begin{cases} E \cdot exp(\frac{x_{bad,z}^{g} - x_{i,z}^{g}}{i^2}), & i > N/2 \\ x_{v_k}^{g+1} + \left| x_{i,z}^{g} - x_{v_k}^{g+1} \right| + Q_Z{}^{+} \cdot M_Z, & \text{Other} \end{cases} \tag{37}$$

3.2.4. Moderation Function

For a given vehicle state and preview-area geometry, KH and the coefficients in Equation (38) are fixed. The fitness is maximized (not minimized). This optimization direction is essential:

for $KA > 0$, the function has a unique unconstrained maximum at $L_d = KH$ and no nonglobal local maxima. Accordingly, the optimum on the admissible interval has the analytical closed form below. The shift KC changes the optimal fitness value but not the maximizing preview distance. The moderation function $f(x)$ is the core of the sparrow search algorithm; an appropriate moderation function can ensure the tracking effect of tracked vehicles. In order to take into account both the path tracking accuracy and the steering control stability of tracked vehicles, the following moderation function is constructed

$$f(L_d) = \frac{1}{exp(KA \cdot (L_d - KH)^2 + KC)} \tag{38}$$

where $K_A > 0$ has units m^{-2} and controls the peak width; K_C is a dimensionless offset. With distance-valued L_d *and* K_H, the exponent is dimensionless. K_C rescales the fitness but does not change its maximizing distance.

$$KH = L_{dmin} + \left(\frac{\lambda \cdot |W - W_{min}|}{W_{max} - W_{min}} + \frac{\xi \cdot d}{d_{max}} - \xi \right) \cdot (L_{dmax} - L_{dmin}) \tag{39}$$

where $\lambda \in (0, 1)$ is the weight associated with the angular straightness indicator, $\xi \in (-1, 0)$ is the lateral-deviation weight, d denotes the current absolute lateral deviation, and d_{max} is its normalization bound. The limits W_{min} and W_{max} are used to normalize W, with $W_{max} > W_{min}$. The coefficients K_A and K_C determine the shape and magnitude of the fitness function in Equation (38), respectively.

Because $\frac{d \ln f}{dL_d} = -2K_A(L_d - K_H)$, log fitness has second derivative $-2K_A < 0$. Its unique unconstrained maximizer is $L_d = K_H$, and the constrained maximizer is $L_{d,ana} = min\{L_{dmax}, max\{L_{dmin}, K_H\}\}$. This requires a fixed number of scalar operations after computing K_H; SSA adds population evaluations and clustering. For the same frozen objective, no feasible SSA candidate can exceed the analytical optimum. This is an analytical result, not a measured timing advantage or a guarantee of identical closed-loop trajectories.

3.2.5. Adaptive Preview Distance Algorithm Process and Parameter Setting

When determining the preview distance, tracked vehicles based on the adaptive preview tracking fuzzy control algorithm input the current lateral deviation of the vehicle and the degree to which the path of the preview area is curved into the improved SSA to search for the most suitable preview distance value. Set the population size N of the improved SSA to 100, the maximum number of iterations G to 200, the maximum value of the weight function KH to 3, the minimum value to 0.8, the range of the preview distance L_d to [1.0 m, 3 m], the number of population discoverers to 20, and the number of scavengers to 80. The analytical illustration in Figure 8 is separate from these population and iteration settings.

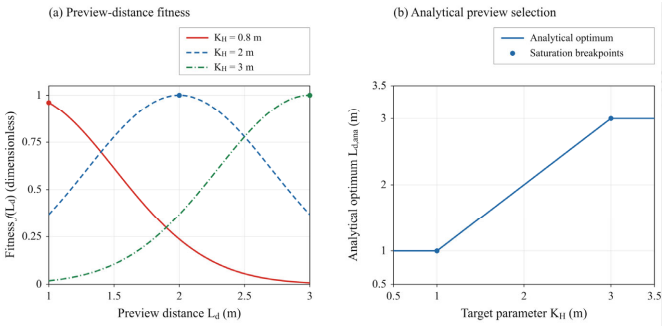


Figure 8. Analytical properties of the preview fitness and its constrained optimum, illustrated with $K_A = 1\ m^{-2}$, $K_C = 0$ and an admissible interval of 1.0–3.0 m. (**a**) Fitness for three fixed target values; dots indicate constrained maxima. (**b**) Optimal preview distance obtained by interval projection.

The preview-selection procedure is summarized in Algorithm 1.

Algorithm 1. Optional SSA implementation of preview selection

Input:
G: Maximum number of iterations for the population
FD: Number of explorers
SD: Number of sentinels
N: Total population size
Output:
L_{dbest}: Optimal preview distance
Freeze the current vehicle state and preview-area geometry; compute KH once, and initialize N feasible candidate distances and the best-so-far record.
Obtain vehicle lateral deviation, curvature of the preview area, and initialize algorithm parameters.
1: **while** $(g < G)$
2: Evaluate feasible candidates using Equation (38), maximizing f; retain the best-so-far individual.
Set the best and worst individuals by descending fitness.
3: **for** $i = 1{:}FD$
4: Introduce chaos factors using equation (34), and update the positions of explorers using equation (35).
5: **end for**
6: Apply fuzzy C-means clustering to categorize the top 30% of the scavenger population;
7: **for** $i = (FD + 1){:}N$
8: Update the positions of scavengers using Equation (37).
9: **end for**
10: **for** i = 1:SD
11: Update the positions of sentinels using Equation (33).
12: **end** for
13: Clip updated distances to the interval and re-evaluate fitness; let Fnew be the current maximum.
14: if Fnew > Fbest
15: Store Fbest = Fnew and the corresponding preview distance.
16: end if
17: $g = g + 1$;
18: **end while**
19: **return** L_{dbest}.

4. Results

The evaluation comprises an analytical assessment of preview selection and a closed-loop tracking study. MATLAB/Simulink and ROS/Gazebo simulations compare adaptive and fixed preview distances at different vehicle speeds, followed by tests on the tracked-vehicle platform. The analytical assessment concerns fixed-state optimization, while the tracking experiments characterize the integrated controller.

4.1. Analytical Assessment of Preview Selection

Figure 8 characterizes the fixed-state fitness and the interval projection that determines the analytical optimum. For a feasible SSA estimate, the preview-distance error is $e_L = |L_{d,SSA} - L_{d,ana}|$, and the logarithmic optimality gap is $\delta_q = K_A[(L_{d,SSA} - K_H)^2 - (L_{d,ana} - K_H)^2] \geq 0$. The corresponding relative fitness loss is $g_f = 1 - exp(-\delta_q)$. These indicators quantify distance deviation and objective suboptimality under identical fixed-state conditions. With best-so-far retention, the maximum fitness is nondecreasing across iterations, while the corresponding minimum cost $q = -ln\, f$ is nonincreasing. The analytical solution provides an exact reference for both indicators. This assessment establishes the optimization properties of the fixed-state objective; differences in execution time and closed-

loop performance between SSA and direct analytical evaluation remain unquantified. The tracking experiments evaluate the integrated controller and do not isolate the effects of the preview solver.

4.2. Analysis of Path Tracking Simulation Tests for Vehicles at Different Speeds

To further verify the effectiveness of the tracking control algorithm proposed in this paper, a joint simulation of vehicle path tracking control is carried out based on ROS/Simulink according to the actual parameters of the motor-driven differential tracked vehicle, as shown in Table 6. A reference path with a total length of 80 m is designed, including straight-line driving, turning and U-turn road conditions, as shown in Figure 9. Set the vehicle speeds to 0.5 m/s, 1.0 m/s and 1.5 m/s, respectively. Based on operational experience, the preview distances of 1.0 m and 3.0 m are selected, respectively, for the tracked vehicle path tracking simulation experiments of the fixed preview distance preview MPC tracking control algorithm. The 1.0 m and 3.0 m fixed-preview baselines used the same fuzzy weight-adaptation law and baseline coefficients as the adaptive-preview controller. Their preview distances were held constant, whereas the adaptive-preview controller selected its distance using the improved SSA. These comparisons examine preview selection within a common fuzzy-weighting architecture; they do not establish that all other run-specific numerical settings were matched.

Table 6. Main parameters of tracked vehicle.

Project	Technical Parameters
Overall machine dimensions/mm × mm × mm	1850 × 815 × 900
Mass/kg	39.86
Operating speed/$(m{\cdot}s)^{-1}$	0~1.5
Track width/mm	80
Distance between track centers L/mm	295
Track ground contact length B/mm	375
Height of center of gravity from ground hg/mm	175
Drive sprocket radius rz/mm	60

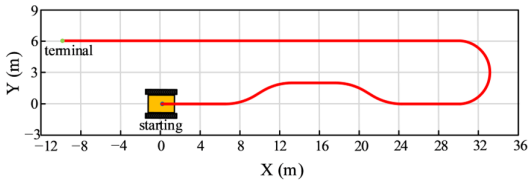


Figure 9. Multi-angle plan path of vehicle.

For a trajectory-level summary of the r-th run, let $d_{r,j}$ denote the signed lateral deviation at the j-th valid sample, and let N_r denote the number of samples retained within the evaluation interval. The following definitions assign equal weight to each retained sample. The maximum absolute lateral error, mean absolute lateral error, and sample standard deviation are defined as $d_{\max,r} = \max_{1\le j\le N_r}\left|d_{r,j}\right|, \mathrm{MAE}_r = \frac{1}{N_r}\sum_{j=1}^{N_r}\left|d_{r,j}\right|, s_{d,r} = \sqrt{\frac{1}{N_r-1}\sum_{j=1}^{N_r}\left(d_{r,j}-\bar{d}_r\right)^2}, \bar{d}_r = \frac{1}{N_r}\sum_{j=1}^{N_r} d_{r,j}$. Where the standard deviation describes the dispersion of the signed lateral deviation within a trajectory.

The definitions above apply to an individual trajectory; they do not specify how summaries from multiple runs are combined. For each speed and controller configuration, five independent runs were performed in both simulation and field experiments. The re-tained observations from the five runs were pooled before calculating the statistics report-ed for

the simulation and field tests. The maximum absolute lateral error, mean absolute lateral error, and standard deviation were evaluated over the pooled observations, rather than obtained by averaging five run-level statistics. The standard deviation therefore characterizes dispersion within the pooled sequence and does not quantify variability between independent runs. No confidence intervals or statistical-significance claims are derived from these summaries.

Mean angular velocity is calculated as the arithmetic mean of the retained signed angular-velocity observations pooled across the five runs for each condition. Absolute values are not taken before averaging. Opposite turning directions can therefore cancel, so a lower signed mean does not establish lower turning magnitude, steering effort, or com-mand smoothness. Percentage reductions in mean angular velocity describe arithmetic changes in this signed statistic and retain the same interpretive limitation.

Figure 10 compares the simulated trajectories of the adaptive-preview controller with those of the fixed-preview controllers at 1.0 and 3.0 m. The 1.0 m baseline has lower mean absolute lateral error than the 3.0 m baseline at all three speeds in Table 7. The angular-velocity responses provide a complementary comparison of steering behavior. The numerical results are assessed separately against each baseline.

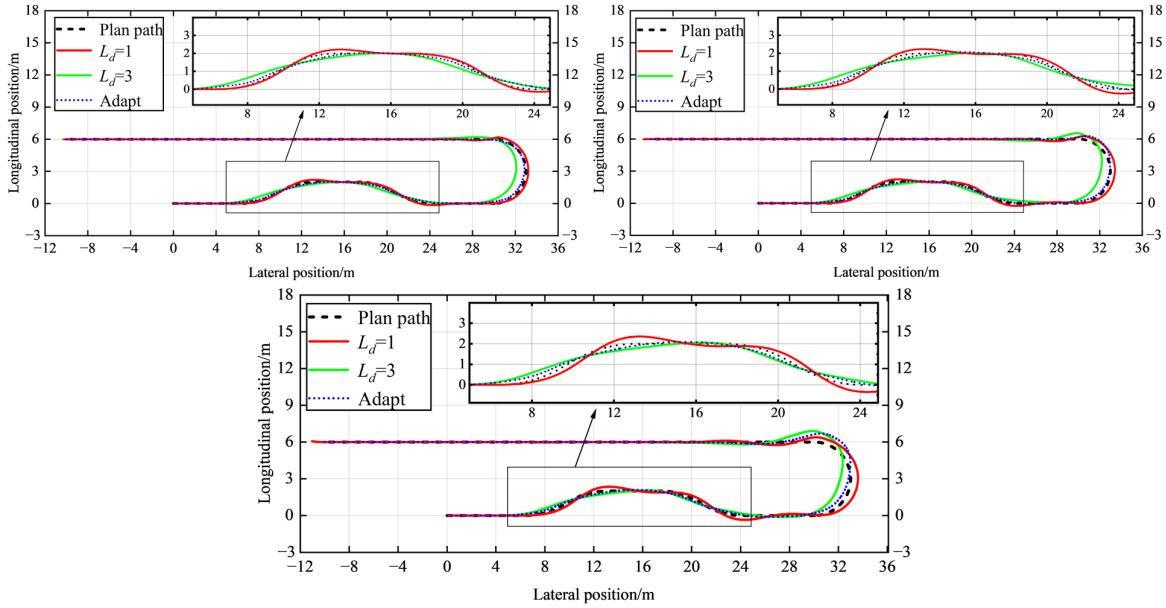


Figure 10. Tracked trajectory simulation with different speeds and multiple curvatures.

Figure 11 presents the adaptive preview distance along the simulated trajectory. The tracking responses are shown in Figure 12 and summarized in Tables 7 and 8. It can be seen from Figure 12a–c that when the vehicle selects preview distances of 1.0 m and 3.0 m for fixed-preview MPC tracking, or uses the proposed adaptive-preview algorithm, the fluctuations in lateral deviation increase with vehicle speed.

Figure 12a–c show the lateral-deviation responses, and Figure 12d–f show the steering angular-velocity responses. Tracking error and angular-velocity dispersion provide different measures of controller behavior. The adaptive controller has lower angular-velocity standard deviation than both fixed-preview baselines at all three simulated speeds, but does not minimize every tracking-error statistic in Table 7. These observations describe the measured response characteristics and do not establish closed-loop stability.

Table 7. Statistics of the lateral deviation results of tracked vehicle path tracking under simulated working conditions.

Speed/ (m·s^{-1})	Preview Distance/m	Max Absolute Error/m	Mean Absolute Error/m	Lateral-Error SD (m)
0.5	1	0.2563	0.0447	0.0670
	3.0	0.9735	0.1200	0.1975
	Adapt	0.2224	0.0528	0.0660
1.0	1	0.4471	0.0738	0.1109
	3.0	0.9774	0.1321	0.1945
	Adapt	0.4805	0.0650	0.0902
1.5	1	0.6091	0.1070	0.1503
	3.0	0.9397	0.1417	0.2186
	Adapt	0.7841	0.0990	0.1538

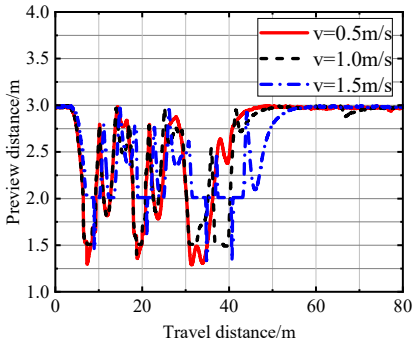
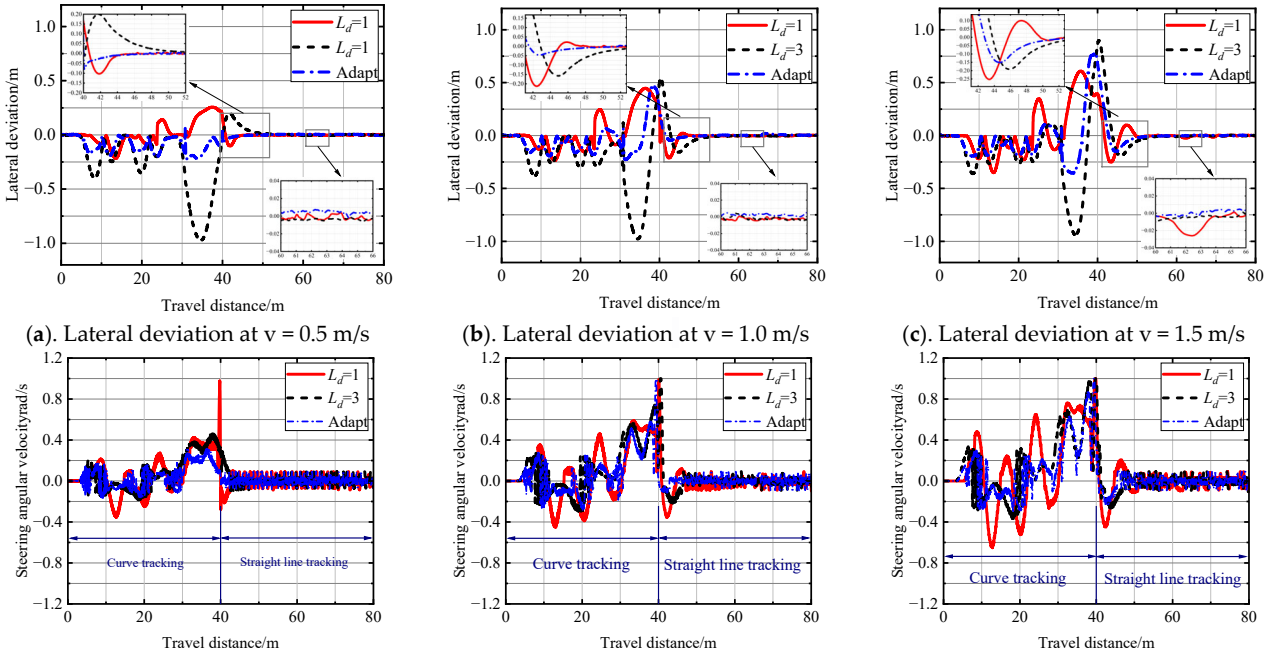


Figure 11. The simulation environment adaptively previews the forward view distance of MPC tracked vehicle path-following control algorithm.



(**a**). Lateral deviation at v = 0.5 m/s (**b**). Lateral deviation at v = 1.0 m/s (**c**). Lateral deviation at v = 1.5 m/s

(**d**). Steering angular velocity at v = 0.5 m/s (**e**). Steering angular velocity at v = 1.0 m/s (**f**). Steering angular velocity at v = 1.5 m/s

Figure 12. Multi-angle path following simulation evaluation diagram of tracked vehicle.

Table 8. Statistics of the steering angular velocity results of tracked vehicles under simulated working conditions.

Speed/ ($m \cdot s^{-1}$)	Preview Distance/m	Max Angular Velocity/($rad \cdot s^{-1}$)	Mean Angular Velocity/($rad \cdot s^{-1}$)	Angular-Velocity SD (rad/s)
0.5	1	1	0.1212	0.1217
	3.0	0.5454	0.0959	0.1007
	Adapt	0.3927	0.0870	0.0722
1.0	1	1	0.1562	0.1714
	3.0	1	0.1365	0.1647
	Adapt	1	0.1186	0.1333
1.5	1	1	0.2073	0.2306
	3.0	1	0.1729	0.2001
	Adapt	1	0.1500	0.1696

Tables 7 and 8 give adaptive-controller mean absolute lateral errors of 0.0528, 0.0650, and 0.0990 m and mean angular velocities of 0.0870, 0.1186, and 0.1500 rad/s at 0.5, 1.0, and 1.5 m/s, respectively. Relative to the 1.0 m baseline, the lateral-error reductions are −18.12%, 11.92%, and 7.48%; relative to the 3.0 m baseline, they are 56.00%, 50.79%, and 30.13%. A negative reduction denotes an increase in error. The corresponding angular-velocity standard-deviation reductions are 40.67%, 22.23%, and 26.45% relative to 1.0 m, and 28.30%, 19.06%, and 15.24% relative to 3.0 m. The two fixed-preview baselines are evaluated independently rather than combined into a pooled comparator.

Figure 11 shows the adaptive preview distance along the simulated trajectory at the three speeds. Together with Tables 7 and 8, these results describe the lateral-error and angular velocity responses obtained with the adaptive-preview controller under the tested conditions.

Figure 11 shows the variation in the selected preview distance with travel distance at vehicle speeds of 0.5, 1.0, and 1.5 m/s. At the beginning of the trajectory, the preview distances for all three speeds are close to 3.0 m. Over approximately 5–45 m of travel, the curves exhibit repeated reductions and recoveries, with the selected distances varying approximately between 1.3 and 3.0 m. This behavior is consistent with the changing path geometry and tracking state along the reference trajectory shown in Figure 9. According to Equation (39), the preview selection depends on both the preview-area geometric indicator and the current lateral deviation. Consequently, its variation cannot be attributed to vehicle speed or path geometry alone.

Over approximately 45–55 m of travel, the selected distances recover toward 3.0 m and subsequently remain close to this value for most of the remaining trajectory. Shorter preview distances provide more local path information, whereas longer distances extend the look-ahead range. The differences among the three curves indicate that the adjustment process varies with operating speed and the associated tracking state. A 3.0 m reference at the upper reference speed therefore does not require the selected distances in Figure 11 to remain at 3.0 m. The angular-velocity responses in Figure 12 provide a complementary comparison of steering behavior. The numerical results are assessed separately against each baseline.

Tables 7 and 8 summarize lateral error and angular velocity from the same simulation runs. For each controller configuration, each of the five runs retained 1601, 801, and 534 observations at 0.5, 1.0, and 1.5 m/s, respectively. The corresponding pooled sample counts were 8005, 4005, and 2670. The three speeds and three controller configurations therefore involved 45 simulation runs in total. Tables 9 and 10 were also calculated from five pooled field runs per condition.

Table 9. Statistics of lateral deviation of tracked vehicle path tracking under real vehicle conditions.

Speed/ (m·s^{-1})	Preview Distance/m	Max Absolute Error/m	Mean Absolute Error/m	Lateral-Error SD (m)
0.5	1	0.1870	0.0434	0.0388
	3.0	1.2146	0.19041	0.2617
	Adapt	0.5075	0.0896	0.0767
1.0	1	0.4497	0.0945	0.0903
	3.0	1.3694	0.2443	0.2610
	Adapt	0.4168	0.0670	0.0751
1.5	1	0.4355	0.1107	0.0985
	3.0	1.1689	0.2089	0.2104
	Adapt	0.4315	0.1127	0.0934

Table 10. Statistics of angular velocity of tracked vehicles under real vehicle conditions.

Speed/ (m·s^{-1})	Preview Distance/m	Max Angular Velocity/(rad·s^{-1})	Mean Angular Velocity/(rad·s^{-1})	Angular-Velocity SD (rad/s)
0.5	1	0.7283	0.1296	0.1151
	3.0	0.3259	0.0676	0.06316
	Adapt	0.3574	0.0768	0.0617
1.0	1	1	0.2284	0.2026
	3.0	0.5484	0.1031	0.0961
	Adapt	0.9253	0.1230	0.1219
1.5	1	1	0.3200	0.2112
	3.0	0.7805	0.1464	0.1537
	Adapt	0.9781	0.1512	0.1603

5. Field Vehicle Test and Result Analysis

5.1. Test Platform Construction

To further verify the effectiveness of the algorithm proposed in this paper, an unmanned tracked vehicle retrofitted with electronic navigation control is used as the experimental platform, as shown in Figure 13 (tracked vehicle). The left and right tracks are driven by independent traction motors with high-precision speed regulation, ensuring stable steering and traveling performance under low-speed operating conditions. The sensor detection system uses a ten-axis IMU (Inertial Measurement Unit) inertial navigation module to calculate the vehicle's heading information, with a heading accuracy of 0.1°. An RTK differential positioning module composed of a 4G communication module and a G70 positioning module is adopted to calculate the vehicle position information. The horizontal positioning accuracy is 1.0 cm + 1 ppm, and the vertical positioning accuracy is 1.5 cm + 1 ppm. In addition, a 1000-line incremental optical encoder is integrated into the drive motor to realize real-time monitoring of the vehicle speed. The Jetson Nano development board and the Asus Chosen Computer together form the vehicle's onboard computer. The two devices exchange data through a local area network. The sensor detection system communicates with the Jetson Nano via the serial communication protocol and is encapsulated into ROS nodes through the corresponding ROS driver packages to achieve distributed communication. The Asus Chosen Computer's Matlab/Simulink R2022b software is also encapsulated into ROS nodes and integrates sensor information to complete algorithm processing. The lower-level STM32F04 microcontroller subscribes to the control topics of the onboard computer to achieve vehicle path tracking control. In an illustrative 0.10 s supervisory-cycle policy, IMU,

RTK, and encoder samples carry acquisition timestamps. At each supervisory instant, the newest valid samples not later than that instant are selected; interpolation uses available timestamped samples where appropriate, and a stale required measurement triggers rejection or retention of the previous feasible command. The controller calculates $vr_{cmd} = vc_{md} + L\omega_{cmd}/2$ and $vl_{cmd} = v_{cmd} - L\omega_{cmd}/2$, applies the velocity and rate limits, and passes separate commands to the two motor-drive channels through the STM32. The embedded control layer uses encoder feedback to regulate track speed and enforces command timeout and emergency-stop handling as implementation safeguards. In an illustrative 0.10 s supervisory-cycle policy, IMU, RTK, and encoder samples carry acquisition timestamps. At each supervisory instant, the newest valid samples not later than that instant are selected; interpolation uses available timestamped samples where appropriate, and a stale required measurement triggers rejection or retention of the previous feasible command. This example describes possible timestamp handling and stale-data responses.

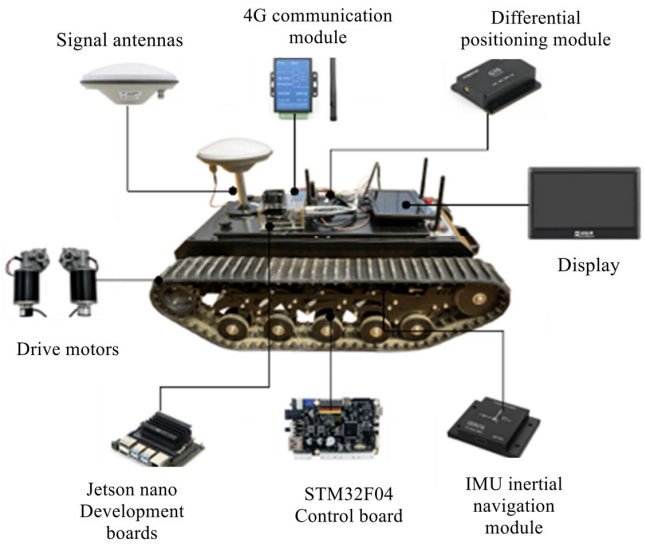


Figure 13. Test platform for unmanned tracked vehicles.

As shown in Figure 14, the structural block diagram of the control system of each part of the tracked vehicle platform is constructed for this paper, and the equipment link relationship of each working part module is illustrated in the figure.

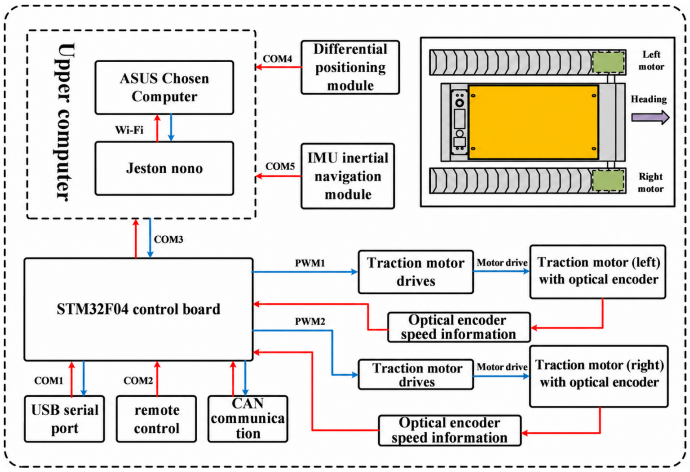


Figure 14. Tracked vehicle control system structure diagram.

5.2. Analysis of Test Results

The path tracking experiment was conducted at the driving center located next to the Panax notoginseng Plantation Experimental Field of the Yunnan Chinese Medicinal Ma-terials Mechanization Engineering Research Center. A section of the experimental field measuring 60 m in length and 8 m in width was selected for the experiment. During the experiments, the vehicle speeds were selected as 0.5 m/s, 1.0 m/s and 1.5 m/s, respectively, to conduct path tracking experiments with a fixed preview distance of 1.0 m and 3.0 m, as well as the path tracking control algorithm of the adaptive preview MPC proposed in this paper. The tracking trajectories are shown in Figure 15. Table 9 shows that the 1.0 m base-line has lower mean absolute lateral error than the 3.0 m baseline at each tested speed. Relative to the 1.0 m baseline, the adaptive controller has lower mean absolute lateral error at 1.0 m/s and higher error at 0.5 and 1.5 m/s. The field comparisons therefore show a speed-dependent tracking-error trade-off. The field baselines used the same fuzzy weight-adaptation law and baseline coefficients as the adaptive-preview controller. The 1.0 m and 3.0 m baselines retained fixed preview distances, whereas the adaptive config-uration used the improved SSA. These results compare preview strategies within a com-mon fuzzy-weighting architecture, rather than a complete factorial ablation of weight ad-aptation and preview adaptation. Figure 16 presents the corresponding lateral-deviation responses in panels (a)–(c) and steering angular-velocity responses in panels (d)–(f), at 0.5, 1.0, and 1.5 m/s, respectively.

Tables 9 and 10 give adaptive-controller mean absolute lateral errors of 0.0896, 0.0670, and 0.1127 m and mean angular velocities of 0.0768, 0.1230, and 0.1512 rad/s at 0.5, 1.0, and 1.5 m/s, respectively. At 1.0 m/s, the lateral-error reductions are $\frac{0.0945-0.0670}{0.0945} \times 100\% = 29.10\%$ relative to the 1.0 m baseline and $\frac{0.2443-0.0670}{0.2443} \times 100\% = 72.57\%$ relative to the 3.0 m baseline. The corresponding reductions at 0.5 and 1.5 m/s are −106.45% and −1.81% relative to 1.0 m, and 52.94% and 46.05% relative to 3.0 m. The comparisons thus show lower lateral error than the 3.0 m baseline at all tested speeds, but lower error than the 1.0 m baseline only at 1.0 m/s.

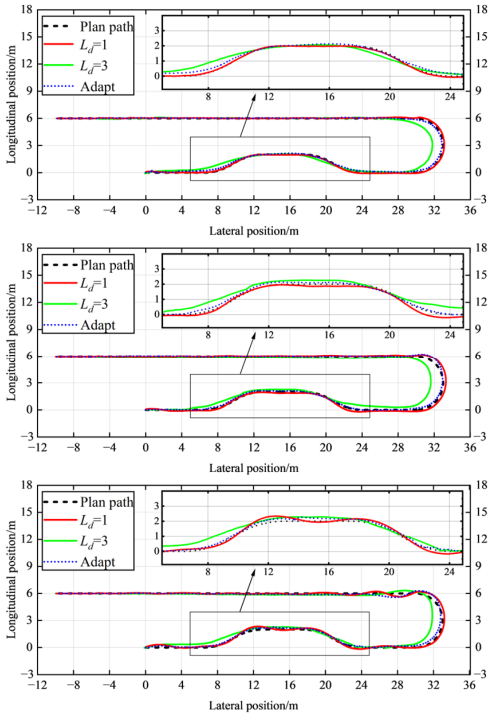


Figure 15. *Cont.*

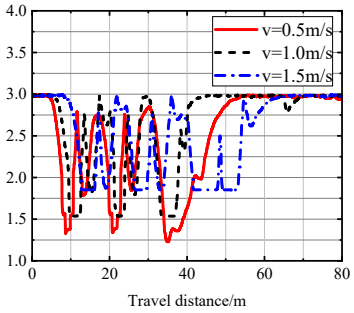
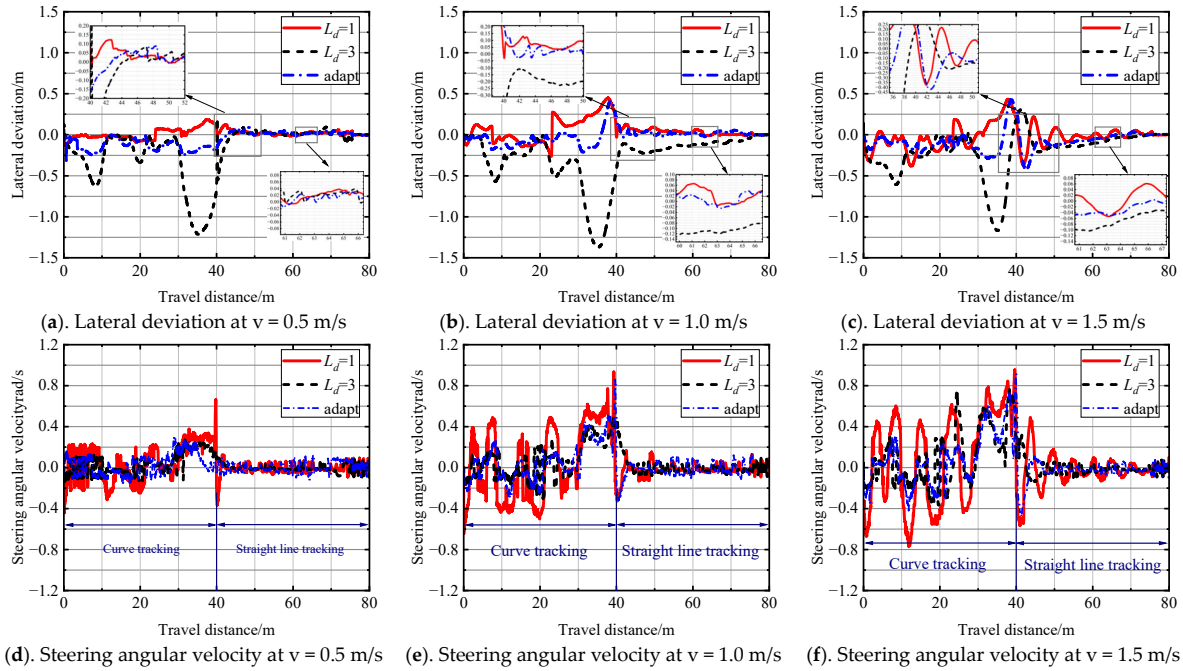


Figure 15. Tracked vehicles with different speeds and multiple curvatures follow the trajectory of real vehicle paths.



(**a**). Lateral deviation at v = 0.5 m/s (**b**). Lateral deviation at v = 1.0 m/s (**c**). Lateral deviation at v = 1.5 m/s

(**d**). Steering angular velocity at v = 0.5 m/s (**e**). Steering angular velocity at v = 1.0 m/s (**f**). Steering angular velocity at v = 1.5 m/s

Figure 16. Tracking results of real vehicle paths with different speeds and multiple curvatures of tracked vehicles.

In the field tests, angular-velocity standard deviation decreases by 46.39%, 39.83%, and 24.10% relative to the 1.0 m baseline at 0.5, 1.0, and 1.5 m/s, respectively. Relative to the 3.0 m baseline, it decreases by 2.31% at 0.5 m/s but increases by 26.85% and 4.29% at 1.0 and 1.5 m/s. Mean angular velocity is lower than that of the 1.0 m baseline at every tested speed, but higher than that of the 3.0 m baseline.

Table 11 evaluates each fixed-preview baseline separately. For a statistic M, the percentage reduction is $r_M = 100\left(M_{fixed} - M_{adaptive}\right)/M_{fixed}$, where positive and negative values denote a decrease and an increase, respectively. Relative to the 3.0 m baseline, mean absolute lateral error decreases by 30.13–56.00% in simulation and 46.05–72.57% in field tests. Relative to 1.0 m, it decreases at 1.0 and 1.5 m/s in simulation and at 1.0 m/s in the field; it increases at the remaining tested conditions. Angular-velocity standard deviation decreases relative to 1.0 m in all cases, but its changes relative to 3.0 m depend on the test condition. The comparisons examine fixed and adaptive preview selection under a shared fuzzy weight-adaptation policy. Because no fixed-weight control group is included, they do not separately quantify the benefit of weight adaptation or its interaction with preview adaptation. The unavailable retained field-sample counts and incomplete evaluation-boundary records limit independent reproduction of the tabulated statistics. Pooling the five runs does not quantify run-to-run variability. The reported mean angular-velocity reductions describe

changes in the signed mean and are not evidence of lower steering effort. No confidence intervals, significance tests, or control-command variation metrics are provided. The effect of weight adaptation, run-to-run repeatability, and command smoothness consequently remain unquantified. Closed-loop stability cannot be inferred from the tracking-error and angular-velocity statistics alone.

Table 11. Percentage reductions in tracking performance of the adaptive-preview controller relative to fixed-preview baselines.

Setting	Speed/ (m/s)	Fixed Preview (m)	Mean Absolute Lateral-Error Reduction (%)	Mean Angular-Velocity Reduction (%)	Angular-Velocity SD Reduction (%)
Simulation	0.5	1.0	−18.12	+28.22	+40.67
		3.0	+56.00	+9.28	+28.30
	1.0	1.0	+11.92	+24.07	+22.23
		3.0	+50.79	+13.11	+19.06
	1.5	1.0	+7.48	+27.64	+26.45
		3.0	+30.13	+13.24	+15.24
Field test	0.5	1.0	−106.45	+40.74	+46.39
		3.0	+52.94	−13.61	+2.31
	1.0	1.0	+29.10	+46.15	+39.83
		3.0	+72.57	−19.30	−26.85
	1.5	1.0	−1.81	+52.75	+24.10
		3.0	+46.05	−3.28	−4.29

6. Conclusions

This study develops a path-tracking MPC controller for differential-drive tracked vehicles by combining fuzzy cost-weight adaptation with adaptive preview selection. The implemented improved SSA uses the speed-based reference in Equation (28) during candidate initialization, while the preview objective depends on the angular straightness indicator and lateral deviation. The fixed-state objective also admits an analytical solution; the present evidence does not establish a numerical-accuracy or runtime advantage of SSA over that solution. Simulations and field tests at 0.5, 1.0, and 1.5 m/s show lower mean absolute lateral error than the 3.0 m fixed-preview baseline. Relative to the 1.0 m baseline, lateral-error changes depend on speed, while angular-velocity SD is lower under all tested conditions. These results describe a condition-dependent trade-off between tracking error and steering-response dispersion for the evaluated configurations.

Author Contributions: Conceptualization, W.S. and D.D.; methodology, D.D.; software, D.D.; validation, D.D.; formal analysis, D.D.; investigation, D.D.; resources, D.D.; data curation, D.D.; writing—original draft preparation, D.D.; writing—review and editing, D.D. and Q.L.; visualization, D.D.; supervision, W.S.; project administration, W.S. and F.W.; funding acquisition, W.S. and F.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Young Scientists Fund of the National Natural Science Foundation of China grant number 52402497.

Data Availability Statement: The raw experimental data generated in this study are not publicly available due to laboratory management restrictions.The datasets supporting the findings of this study are available from the corresponding author upon reasonable request.

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

References

1. Bai, Y.H.; Zhang, B.H.; Xu, N.M.; Zhou, J.; Shi, J.; Diao, Z. Vision-based navigation and guidance for agricultural autonomous vehicles and robots: A review. *Comput. Electron. Agric.* **2022**, *205*, 107584. [CrossRef]
2. Han, S.; He, Y.; Fang, H. Recent development in automatic guidance and autonomous vehicle for agriculture: A Review. *J. Zhejiang Univ. (Agric. Life Sci.)* **2018**, *44*, 381–391.
3. Luo, X.W.; Liao, J.; Hu, L.; Zang, Y.; Zhou, Z. Improving agricultural mechanization level to promote agricultural sustainable development. *Trans. Chin. Soc. Agric. Eng.* **2016**, *32*, 1–11.
4. Hu, L.; Hou, K.; He, J.; Man, Z.; Xie, J.; Wang, P.; Tu, T.; Gao, R.; Zi, L.; Huang, Y.; et al. Multi-constraint improved RS path planning method for unmanned rice direct seeding machine. *Comput. Electron. Agric.* **2024**, *224*, 109236. [CrossRef]
5. Li, S.; Xu, H.; Ji, Y.; Cao, R.; Zhang, M.; Li, H. Development of a following agricultural machinery automatic navigation system. *Comput. Electron. Agric.* **2019**, *158*, 335–344. [CrossRef]
6. Badue, C.; Guidolini, R.; Carneiro, R.V.; Azevedo, P.; Cardoso, V.B.; Forechi, A.; Jesus, L.; Berriel, R.; Paixão, T.M.; Mutz, F.; et al. Self-driving cars: A survey. *Expert Syst. Appl.* **2021**, *165*, 113816. [CrossRef]
7. Xie, B.; Jin, Y.; Faheem, M.; Gao, W.; Liu, J.; Jiang, H.; Cai, L.; Li, Y. Research progress of autonomous navigation technology for multi-agricultural scenes. *Comput. Electron. Agric.* **2023**, *211*, 107963. [CrossRef]
8. Gao, X.; Li, J.; Fan, L.; Zhou, Q.; Yin, K.; Wang, J.; Song, C.; Huang, L.; Wang, Z. Review of wheeled mobile robots' navigation problems and application prospects in agriculture. *IEEE Access* **2018**, *6*, 49248–49268. [CrossRef]
9. Yang, Y.; Li, Y.; Wen, X.; Zhang, G.; Ma, Q.; Cheng, S.; Qi, J.; Xu, L.; Chen, L. An optimal goal point determination algorithm for automatic navigation of agricultural machinery: Improving the tracking accuracy of the Pure Pursuit algorithm. *Comput. Electron. Agric.* **2022**, *194*, 106760. [CrossRef]
10. Zhang, W.; Hu, L.; Ding, F.; Luo, X.; Zhang, Z.; Hu, L.; Huang, P.; Peng, M. Parking precise alignment control and cotransporter system for rice harvester and transporter. *Comput. Electron. Agric.* **2023**, *215*, 108443. [CrossRef]
11. Zhan, J.Y.; Ma, Z.B.; Zhang, L.G. Data-driven modeling and distributed predictive control of mixed vehicle platoons. *IEEE Trans. Intell. Veh.* **2023**, *8*, 572–582. [CrossRef]
12. Xu, S.; Peng, H.; Song, Z.; Chen, K.; Tang, Y. Design and test of speed tracking control for the Self-Driving lincoln MKZ platform. *IEEE Trans. Intell. Veh.* **2020**, *5*, 324–334. [CrossRef]
13. Wang, F.A.; Wang, B.Y.; Zhang, Z.G.; Xie, K.; Li, A.; Ni, C.; Fang, H. Adaptive preview tracking fuzzy control algorithm for tracked vehicles. *Trans. Chin. Soc. Agric. Eng.* **2024**, *40*, 32–43.
14. Sun, J.; Li, Q.; Ding, S.; Xing, G.; Chen, L. Fixed-time generalized super-twisting control for path tracking of autonomous agricultural vehicles considering wheel slipping. *Comput. Electron. Agric.* **2023**, *213*, 108231. [CrossRef]
15. Kraus, T.; Ferreau, H.; Kayacan, E.; Ramon, H.; De Baerdemaeker, J.; Diehl, M.; Saeys, W. Moving horizon estimation and nonlinear model predictive control for autonomous agricultural vehicles. *Comput. Electron. Agric.* **2013**, *98*, 25–33. [CrossRef]
16. Faulwasser, T.; Weber, T.; Zometa, P.; Findeisen, R. Implementation of nonlinear model predictive path-following control for an industrial robot. *IEEE Trans. Control Syst. Technol.* **2017**, *25*, 1505–1511. [CrossRef]
17. Ahn, J.; Shin, S.; Kim, M.; Park, J. Accurate path tracking by adjusting preview point in pure pursuit method. *Int. J. Automot. Technol.* **2021**, *22*, 119–129. [CrossRef]
18. Liu, W.L.; Guo, R.; Zhao, J.Y. Predictive control method for the path tracking of agricultural machinery based on preview model. *Trans. Chin. Soc. Agric. Eng.* **2023**, *39*, 39–50.
19. Raikwar, S.; Fehrmann, J.; Herlitzius, T. Navigation and control development for a four-wheel-steered mobile orchard robot using model-based design. *Comput. Electron. Agric.* **2022**, *202*, 107410. [CrossRef]
20. He, J.; Hu, L.; Wang, P.; Liu, Y.; Man, Z.; Tu, T.; Yang, L.; Li, Y.; Yi, Y.; Li, W.; et al. Path tracking control method and performance test based on agricultural machinery pose correction. *Comput. Electron. Agric.* **2022**, *200*, 107185. [CrossRef]
21. Wang, X.; Zhang, B.; Du, X.; Chen, H.; Zhu, T.; Wu, C. Self-Adjusting Look-Ahead Distance of Precision Path Tracking for High-Clearance Sprayers in Field Navigation. *Agronomy* **2025**, *15*, 1433. [CrossRef]
22. Ji, X.; Ding, S.; Wei, X.; Mei, K.; Cui, B.; Sun, J. Path tracking control of unmanned agricultural tractors via modified supertwisting sliding mode and disturbance observer. *IEEE-Asme Trans. Mechatron.* **2024**, *29*, 4051–4062. [CrossRef]
23. Wang, F.; Wang, J.; Ye, L.; Lu, Y.; Liang, J.; Chen, C.-M.; Kumari, S.; Agarwal, K. Robust NLOS Error Mitigation in Hybrid RD-AOA-Based CAVs Localization Using Convex Relaxation in Harsh Urban Forest. *IEEE Trans. Consum. Electron.* **2026**, *72*, 1596–1606. [CrossRef]
24. Wang, F.; Ye, L.; Lin, S.; Wang, J.; Lu, Y.; Liang, J.; Chen, C.-M.; Kumari, S.; Agarwal, K.; Zhang, F. Vision-based integrated perception–planning framework for autonomous navigation on unstructured roads. *IEEE Trans. Consum. Electron.* **2026**. [CrossRef]
25. Jin, X.; Tao, Y.; Zhang, Y.; Wu, H.; Lu, J.; Valerienne, N.; Ikiela, O. A Review of Parking Trajectory Planning and Modeling Techniques for Autonomous Vehicles. *CMES-Comput. Model. Eng. Sci.* **2026**, *148*, 2. [CrossRef]

26. Cheng, J.; Zhang, B.; Zhang, C.; Zhang, Y.; Shen, G. A model-free adaptive predictive path-tracking controller with PID terms for tractors. *Biosyst. Eng.* **2024**, *242*, 38–49. [CrossRef]
27. Zhang, L.N.; Zhang, R.R.; Li, L.L.; Ding, C.; Zhang, D.; Chen, L. Research on virtual ackerman steering model based navigation system for tracked vehicles. *Comput. Electron. Agric.* **2022**, *192*, 106615. [CrossRef]
28. Hu, C.; Ru, Y.; Li, X.; Fang, S.; Zhou, H.; Yan, X.; Liu, M.; Xie, R. Path tracking control for brake-steering tracked vehicles based on an improved pure pursuit algorithm. *Biosyst. Eng.* **2024**, *242*, 1–15. [CrossRef]
29. Zhang, T.; Jiao, X.H.; Lin, Z.M. Finite time trajectory tracking control of autonomous agricultural tractor integrated nonsingular fast terminal sliding mode and disturbance observer. *Biosyst. Eng.* **2022**, *219*, 153–164. [CrossRef]
30. He, Y.; Zhou, J.; Sun, J.; Jia, H.; Liang, Z.; Awuah, E. An adaptive control system for path tracking of crawler combine harvester based on paddy ground conditions identification. *Comput. Electron. Agric.* **2023**, *210*, 107948. [CrossRef]
31. Ruslan, N.A.I.; Amer, N.H.; Hudha, K.; Abd Kadir, Z.; Mohamed Ishak, S.A.F.; Dardin, S.M.F.S. Modelling and Control Strategies in Path Tracking Control for Autonomous Tracked Vehicles: A Review of the State of the Art and Challenges. *J. Terramechanics* **2023**, *105*, 67–79. [CrossRef]
32. Wang, F.; Xu, X.; Zhong, W.; Dong, H.; Lu, Y.; Gao, B.; Zhang, Z. Cloud-Based Eco-Driving Predictive Cruise Control for Connected and Automated Electric Vehicles. *IEEE Trans. Transp. Electrif.* **2026**, *12*, 4044–4056. [CrossRef]
33. Wang, F.; Nie, A.; Lu, Y.; Jiang, S.; Kumari, S.; Kumar, R.; Chen, C.-M.; Li, Z.; Liang, J. A Bio-Inspired Spatio-Temporal Residual Framework for Secure GNSS Localization on Resource-Constrained Consumer Devices. *IEEE Trans. Consum. Electron.* **2026**, *72*, 5041–5054. [CrossRef]
34. Zhao, X.; Lu, E.; Tang, Z.; Luo, C.; Xu, L.; Wang, H. Trajectory Prediction Method for Agricultural Tracked Robots Based on Slip Parameter Estimation. *Comput. Electron. Agric.* **2024**, *222*, 109057. [CrossRef]
35. Cosenza, C.; Malfi, P.; Nicolella, A.; Niola, V.; Savino, S. Multibody Modeling for the Analysis of the Maneuverability of a Skid Steering 6-Wheeled Robot. In *Advances in Italian Mechanism Science; Mechanisms and Machine Science*; Springer: Cham, Switzerland, 2024; Volume 164, pp. 272–280.
36. Guo, S.; Ning, Z.; Lv, M. Path Tracking and Energy Efficiency Coordination Control Strategy for Skid-Steering Unmanned Ground Vehicle. *Control Eng. Pract.* **2024**, *152*, 106048. [CrossRef]
37. Wang, S.; Guo, J.; Mao, Y.; Wang, H.; Fan, J. Research on the Model Predictive Trajectory Tracking Control of Unmanned Ground Tracked Vehicles. *Drones* **2023**, *7*, 496. [CrossRef]
38. Zeng, H.; Lyu, S.; Gao, P.; Cheng, S.; Gao, S.; Chen, J.; Li, Z.; Wei, Z.; Li, Z. Trajectory Tracking Control of an Agricultural Tracked Vehicle Based on Nonlinear Model Predictive Control. *Agriculture* **2026**, *16*, 816. [CrossRef]
39. Liu, H.; Han, Z.; Han, M.; Qi, R.; Liu, X.; Mou, J.; Tan, H.; Zhao, L. Slip-Aware Model Predictive Path Tracking Control for Tracked Agricultural Vehicles in Hilly Wheat Fields. *Smart Agric. Technol.* **2026**, *14*, 102292. [CrossRef]
40. Wu, G.; Wang, G.; Bi, Q.; Wang, Y.; Fang, Y.; Guo, G.; Qu, W. Research on unmanned electric shovel autonomous driving path tracking control based on improved pure tracking and fuzzy control. *J. Field. Rob.* **2023**, *40*, 1739–1753. [CrossRef]
41. Xue, J.; Shen, B. A novel swarm intelligence optimization approach: Sparrow search algorithm. *Syst. Sci. Control Eng.* **2020**, *8*, 22–34. [CrossRef]
42. Sun, L.; Si, S.; Ding, W.; Wang, X.; Xu, J. Multiobjective sparrow search feature selection with sparrow ranking and preference information and its applications for high-dimensional data. *Appl. Soft Comput.* **2023**, *147*, 110837. [CrossRef]
43. Zhang, X.; Xu, Y.; Yu, C.; Heidari, A.A.; Li, S.; Chen, H.; Li, C. Gaussian mutational chaotic fruit fly-built optimization and feature selection. *Expert Syst. Appl.* **2020**, *141*, 112976. [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.