


Article

# Research on Trajectory Planning and Tracking Algorithm of Crawler Paver

Jian Zhan <sup>1</sup>, Wei Li <sup>2</sup> , Jiongfan Wang <sup>3</sup>, Shusheng Xiong <sup>2,3,4,\*</sup>, Xiaofeng Wu <sup>3</sup> and Wei Shi <sup>3</sup><sup>1</sup> D.R. (Zhejiang) Powertrain Technology Co., Ltd., Jiaxing 314000, China; zhanjian@d-r-power.com<sup>2</sup> Longquan Industrial Innovation Research Institute, Lishui 323700, China; liweisail@126.com<sup>3</sup> College of Energy Engineering, Zhejiang University, Hangzhou 310027, China; 12427127@zju.edu.cn (J.W.); 22160132@zju.edu.cn (X.W.); shiw@zju.edu.cn (W.S.)<sup>4</sup> Jiaxing Research Institute, Zhejiang University, Jiaxing 314050, China

\* Correspondence: xiongss@zju.edu.cn

**Abstract:** The implementation of unmanned intelligent construction on the concrete surfaces of an airport effectively improves construction accuracy and reduces personnel investment. On the basis of three known common tracked vehicle dynamics models, reference trajectory planning and trajectory tracking controller algorithms need to be designed. In this paper, based on the driving characteristics of the tracked vehicle and the requirements of the stepping trajectory, a quartic polynomial trajectory planning algorithm was selected with the stability of the curve as a whole and the end point as the optimization goal, combining the tracked vehicle dynamics model, collision constraints, start-stop constraints and other boundary conditions. The objective function of trajectory planning was designed to effectively plan the reference trajectory of the tracked vehicle's step-by-step travel. In order to realize accurate trajectory tracking control, a nonlinear model predictive controller with transverse-longitudinal integrated control was designed. To improve the real-time performance of the controller, a linear model predictive controller with horizontal and longitudinal decoupling was designed. MATLAB 2023A and CoppeliaSim V4.5.1 were used to co-simulate the two controller models. The experimental results show that the advantages and disadvantages of the tracked vehicle dynamics model and controller design are verified.



**Citation:** Zhan, J.; Li, W.; Wang, J.; Xiong, S.; Wu, X.; Shi, W. Research on Trajectory Planning and Tracking Algorithm of Crawler Paver. *Machines* **2024**, *12*, 650. <https://doi.org/10.3390/machines12090650>

Academic Editor: Raffaele Di Gregorio

Received: 13 August 2024

Revised: 7 September 2024

Accepted: 14 September 2024

Published: 17 September 2024



**Copyright:** © 2024 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/>).

**Keywords:** crawler vehicles; route planning; controller algorithms; trajectory tracking

## 1. Introduction

In this manuscript, the target trajectory planning and trajectory tracking algorithms required for the control of walking are introduced, and different algorithms are compared and further verified by experiments. The airport has become an important hub [1], and as the number of airports continues to increase, concrete slipform paving technology has been gradually applied to the construction of highways, airport runways, taxiways and aprons because of its fully automated paving construction process, high construction efficiency, quality control and other highly favored advantages [2–4]. The driving control system and road surface detection system of the slipform paving [5–8] post-processing intelligent vehicle include four parts. The first is the perception of road surface information [9–11]; the second obtains the current location of the vehicle; the third completes the driving control [12–16] and the final step designs the underlying execution module to implement the above functionalities [17]. Jin et al. have used the concept of symmetry to design trajectory planning for parallel parking of autonomous ground vehicles, propose an improved safe travel corridor (I-STC) and design a smooth parking trajectory [18]. Chen et al. adopted an adaptive time domain model predictive control method based on Gaussian function to improve the trajectory tracking control accuracy of unmanned vehicles. Finally, the proposed control algorithm is verified by Carsim and MATLAB/Simulink co-simulation to solve the problem of poor track tracking performance of unmanned vehicles under

complex working conditions [19]. The lack of existing research on trajectory planning tracking [20,21] mainly includes the limitations of the algorithms, the challenges of complex environments and the task difficulty of a high degree of freedom machines. The innovation of this paper is based on three known tracked vehicle dynamics models on which two different algorithms are compared in terms of their tracking ability in realistic scenarios, and joint simulation with MATLAB 2023A and CoppeliaSim V4.5.1 [22] was used to finally select the most suitable target trajectory planning and tracking algorithm for driving control.

## 2. Travel Control Path Planning and Controller Setup

Three common tracked vehicle dynamics models give the relationship between tracked wheel velocity inputs and tracked vehicle motion states, which is the basis for designing reference trajectory planning and trajectory tracking controller algorithms. A stepping cycle of the tracked vehicle can be roughly broken down into the following: image acquisition and lane line extraction (to obtain the initial position state), stepping to the designated position (driving control), using the leveling mechanism to adjust the vehicle posture, construction, detection of the road surface (to determine whether the flatness is qualified) and landing of the track wheel under the leveling mechanism. In this chapter, the trajectory planning and controller design of “step forward to the specified position” are studied. Due to the inevitable interference in the process of lowering the leveling mechanism, the position and posture of the tracked vehicle at the beginning of the next cycle is deviated from that before leveling, so the position and posture of the vehicle need to be adjusted by correcting the deviation in the next step, so as to ensure that the vehicle is in the correct construction position at the end of the step. The method used in this study was to perform an initial position analysis based on the data acquired from the images, plan a stepping trajectory in combination with the constraints and then control the tracked vehicle to track the trajectory through the control system. There are several constraints in the stepping process of the tracked vehicle, and this strategy can deal with the constraints in stages and improve the control efficiency. Because there is no obstacle avoidance involved in trajectory planning and there are many boundary constraints, and considering the smooth running characteristics of tracked vehicles, quartic polynomial trajectory planning was adopted; the controller also needed to meet constraints such as speed and angular velocity, so it adopted the model prediction algorithm and considered the response time problem, and the control algorithm coupling linear model prediction with the PID controller was investigated on the basis of the nonlinear model prediction. Linear models can fit samples with curves, but the decision boundary for classification is a straight line. The only thing that needs to be known is whether it is a linear model or a nonlinear model. Traditional PID control is a linear combination of error ratios, integral and differential. It has the characteristics of a simple principle, easy parameter adjustment and strong robustness, and it has been widely used in industrial process control. But for some complex systems, especially nonlinear objects, the PID controller cannot obtain satisfactory results. If traditional PID is improved to achieve higher control requirements for some nonlinear objects, it will have great practical significance. Nonlinear PID is improved by introducing nonlinear factors on the basis of traditional PID. The basic elements of the control quantity are not directly derived from the input–output error, but the proportion, integral and differential of the error after nonlinear changes. Because the gain parameters in the nonlinear PID controller can change with the control error, the influence of nonlinear factors can be overcome and weakened, and the robustness and adaptability of the controller can be improved. The two algorithms can also be simulated, compared and analyzed, respectively. On the basis of the established tracked vehicle model, a joint simulation with MATLAB 2023A and CoppeliaSim V4.5.1 was carried out, and the advantages and disadvantages of the two algorithms were compared in the real physical field.

### 3. Quartic Polynomial Trajectory Planning

MobileEye's cameras, which have been more successful in the field of perception, use a cubic polynomial to describe lane lines:

$$S = A \times L_1^3 + B \times L_1^2 + C \times L_1 + D \quad (1)$$

$S$  is the lateral offset distance of a point on the lane line ahead relative to the straight line where the camera is located, and  $L_1$  is the longitudinal distance of a point on the lane line from the camera.  $A$ ,  $B$ ,  $C$  and  $D$  are cubic polynomial coefficients. For a vehicle traveling along this polynomial, the trajectory curvature corresponds to the longitudinal distance as follows:

$$Crv(S) = 6A \times L_1 + 2B \quad (2)$$

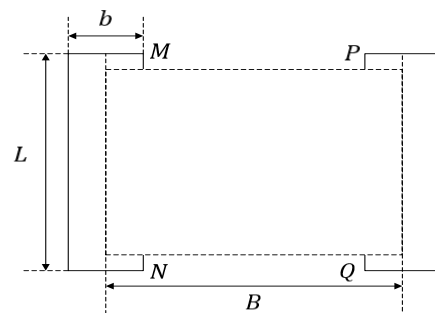
Equation for the rate of change of curvature with respect to longitudinal distance:

$$Crv'(S) = 6A \quad (3)$$

The rate of change of the curvature is constant, which is equivalent to the constant rotational speed of one side track and the constant rate of change of the rotational speed of the other track when the tracked vehicle is traveling. In order to be closer to actual driving conditions, the rotational speed response of one side track is constrained by the mechanical structure and other constraints, and its rate of change is difficult to maintain at a constant, so a quartic polynomial is considered to fit the driving trajectory.

#### 3.1. Trajectory Planning Coordinate Transformation

The stepping trajectory needs to ensure that the tracked vehicle traveling along the trajectory will not collide with the road surface to be constructed, and the grounded part of the tracked vehicle has only two sides of the track, so only the contact between the track and the road surface needs to be considered in the collision analysis. The track is simplified as a rectangle with length  $L$  and width  $b$ . The center distance between the two sides of the track is  $B$ , and the inner vertices of the left and right tracks are  $M$ ,  $N$ ,  $P$ , and  $Q$ , respectively, as shown in Figure 1.



**Figure 1.** Simplification and parameter definition of track.

Point  $M$  and  $N$  are as follows:

$$\begin{cases} x_{M,N} = x - \left(\frac{B}{2} - \frac{b}{2}\right) \sin \varphi \pm \frac{L}{2} \cos \varphi \\ y_{M,N} = y + \left(\frac{B}{2} - \frac{b}{2}\right) \cos \varphi \pm \frac{L}{2} \sin \varphi \end{cases} \quad (4)$$

Point  $P$  and  $Q$  are as follows:

$$\begin{cases} x_{P,Q} = x + \left(\frac{B}{2} - \frac{b}{2}\right) \sin \varphi \pm \frac{L}{2} \cos \varphi \\ y_{P,Q} = y - \left(\frac{B}{2} - \frac{b}{2}\right) \cos \varphi \pm \frac{L}{2} \sin \varphi \end{cases} \quad (5)$$

### 3.2. Collision Constraints for Trajectory Planning

$t$  represents the sampling moment. Since the quartic polynomial curve is continuous and smooth, the interpolation optimization function ensures that the whole curve is close to the optimal curve as long as the sampling frequency is sufficient and each interpolation satisfies the constraints. In order to meet the engineering requirements,  $j$  and  $\rho$  are replaced by interpolation: the coordinates of the geometric center point of the tracked vehicle along the trajectory are  $O(x, y)$ , and the tracked vehicle swings across at an angle  $\varphi$ , as shown in Figure 2. In order to make the tracked vehicle travel without collision with the road surface to be constructed, it is necessary to make restrictions on the position of the apex of the track, setting the width of the road surface to be constructed as  $D$  and the safety distance as  $i$ , as shown in Figure 3.

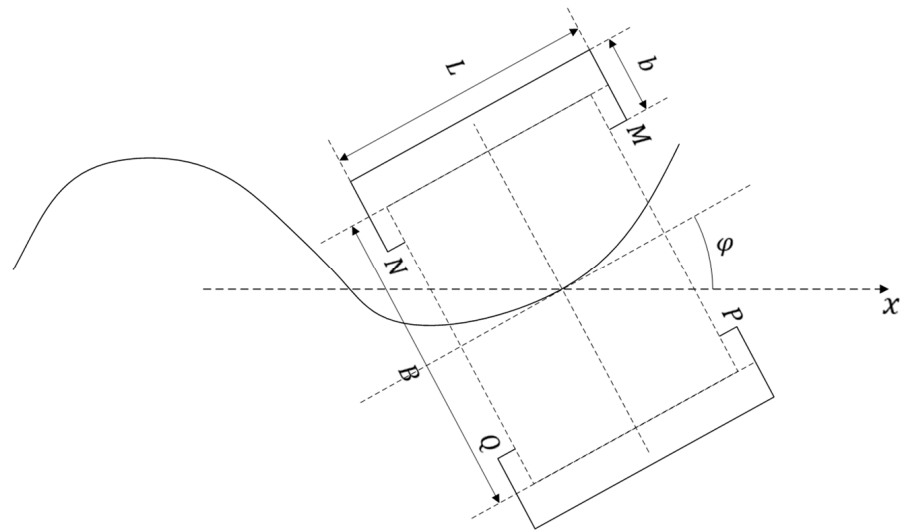


Figure 2. Coordinate transformation of a point on the trajectory.

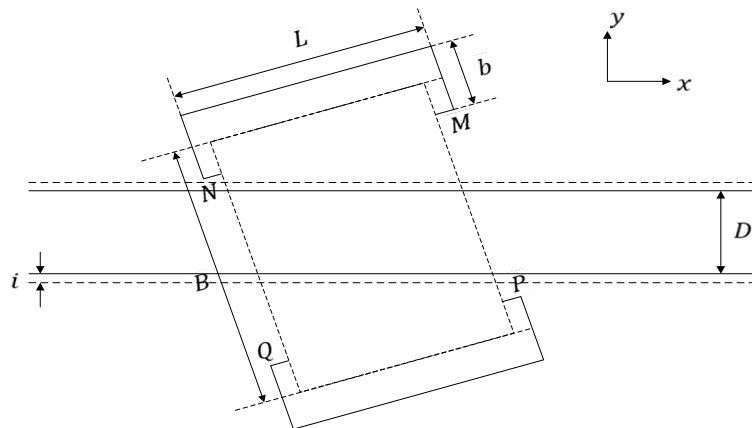


Figure 3. Collision constraints.

Then, the distance of the vertical coordinates of  $M, N, P$  and  $Q$  from the central axis should all be greater than  $\frac{D}{2} + i$ :

$$\min(|y_M|, |y_N|, |y_P|, |y_Q|) > \frac{D}{2} + i \tag{6}$$

### 3.3. Optimized Function Design for Trajectory Planning

To ensure that the tracked vehicle travels smoothly and stably and stops smoothly at the end of the step, the equation was defined. In summary, the quartic polynomial trajectory planning optimization function is given by the following:

$$J = a\rho^2 + bj^2 + c\sigma^2 \quad (7)$$

$\rho$  is the maximum curvature of the trajectory,  $j$  is the maximum sharpness of the trajectory,  $\sigma$  is the end curvature of the trajectory and  $a$ ,  $b$  and  $c$  are the weighting coefficients.  $\rho$  limits the minimum radius of the curvature of the trajectory to make the overall trajectory smoother, while  $j$  limits the rate of change of lateral acceleration to make the velocity change of the left and right tracks smoother, which reduces the difficulty of trajectory tracking and motor control.

In order to meet the engineering requirements,  $j$  and  $\rho$  are replaced by interpolation:

$$J = \sum_{t=1}^{N_{end}} (a_t \rho_t^2 + b_t j_t^2) + c\sigma^2 \quad (8)$$

$t$  represents the sampling moment. Since the quartic polynomial curve is continuous and smooth, the interpolation optimization function ensures that the whole curve is close to the optimal curve as long as the sampling frequency is sufficient and each interpolation satisfies the constraints.

If we define  $\phi(t)$  as the vehicle motion state at moment  $t$  and  $S$  as the motion state ensemble:

$$\phi(t) = [\rho_t, j_t, y_t, \varphi_t]^T \quad \forall t \in [1, N_{end}] \quad (9)$$

$$S = [\phi(1), \phi(2), \dots, \phi(N_{end})] \quad (10)$$

$lr(t)$ ,  $hr(t)$  denote the motion state constraints of the vehicle at the sampling moment  $t$  and use  $LR$  and  $HR$  to denote the upper and lower boundary conditions, respectively:

$$\begin{cases} lr(t) = [\rho_{min_t}, j_{min_t}, y_{min_t}, \varphi_{min_t}]^T \\ \quad \forall t \in [1, N_{end}] \\ hr(t) = [\rho_{max_t}, j_{max_t}, y_{max_t}, \varphi_{max_t}]^T \\ \quad \forall t \in [1, N_{end}] \end{cases} \quad (11)$$

$$\begin{cases} LR = [lr(1), lr(2), \dots, lr(N_{end})]^T \\ HR = [hr(1), hr(2), \dots, hr(N_{end})]^T \end{cases} \quad (12)$$

In summary, the quartic polynomial trajectory planning optimization function is given as follows.

$$\begin{aligned} \min J &= \sum_{t=1}^{N_{end}} (a_t \rho_t^2 + b_t j_t^2) + c\sigma^2 \\ \text{s.t. } & LR \leq S \leq HR \end{aligned} \quad (13)$$

The optimized quartic polynomial trajectory can be obtained using the `fmincon` function provided by MATLAB 2023A, using the default interior point method or the effective set method.

### 3.4. Verification of Trajectory Planning Simulation

The main parameters in the simulation are as follows: mass of construction machinery  $m = 15,000$  kg, lateral track gauge of bilateral track  $B = 7$  m, grounding length of track  $L = 4$  m, track width  $b = 1$  m, reduction ratio of transmission system  $i = 25$ , transmission efficiency  $\eta = 0.85$ , road width  $D = 3$  m, maximum steering resistance coefficient  $\varphi = 0.68$ , coefficient of rolling friction  $\mu_s = 0.04$ , coefficient of internal resistance of crawler  $f = 0.04$ ,

minimum safety distance between track and road surface to be constructed is 0.2 m, step distance is 6 m, the rotational inertia of engineering machinery  $J = 40,000 \text{ kg}\cdot\text{m}^2$ , the target state of the crawler's position at the end of the road surface is  $[x_{end}, y_{end}, \varphi_{end}] = [6, 0, 0]$  and the longitudinal speed of the crawler is constant at 1 m/s under the ideal state. The starting point inputs for the two typical working conditions are verified separately. The end point transverse swing angle is less than 0.01rad, the absolute value of the end point longitudinal coordinate of the tracked vehicle's geometric center is less than 0.01m and the track does not touch the safety limit throughout the whole process; good trajectory optimization is judged by the curvature at the end point being less than  $0.1 \text{ m}^{-1}$  and the curvature maxima being less than  $0.5 \text{ m}^{-1}$ .

For the first common working condition, the vehicle has a large lateral distance deviation before stepping and traveling. Assuming that the initial position of the vehicle obtained by image processing before stepping and driving is  $[x_0, y_0, \varphi_0] = [0, 0.5, 0]$  and the sampling step size is 0.1 m, the weight coefficients of the optimization function are  $a_t = b_t = 1 (t \in [1, 60])$ ,  $c_t = 100$ , and the optimization algorithm adopts the interior point method. The output result is as follows:

$$y = -0.0012x^4 + 0.0185x^3 - 0.0833x^2 + 0.5 \quad (14)$$

The curvature at the end point is  $1.3 \times 10^{-4}$ , the transverse pendulum angle is  $1.4 \times 10^{-12}$  and the longitudinal coordinate of the geometric center of the tracked vehicle is  $2.12 \times 10^{-12}$ , so it can be seen that the trajectory planning meets the end point motion state requirements.

The trajectory curve, the area swept by the inside of the left and right tracks, the minimum distance of the left and right tracks from the center of the road surface to be constructed and the curvature change curve of the trajectory are shown in Figure 4 as  $a$ ,  $b$ ,  $c$  and  $d$ , respectively. Among them, the blue straight line in Figure 4b,c indicates the boundary of the road to be constructed, and the red dashed line indicates the safety limit, i.e., the point of the inner side of the track that is closest to the road to be constructed (the yellow area) cannot enter into the red dashed line.

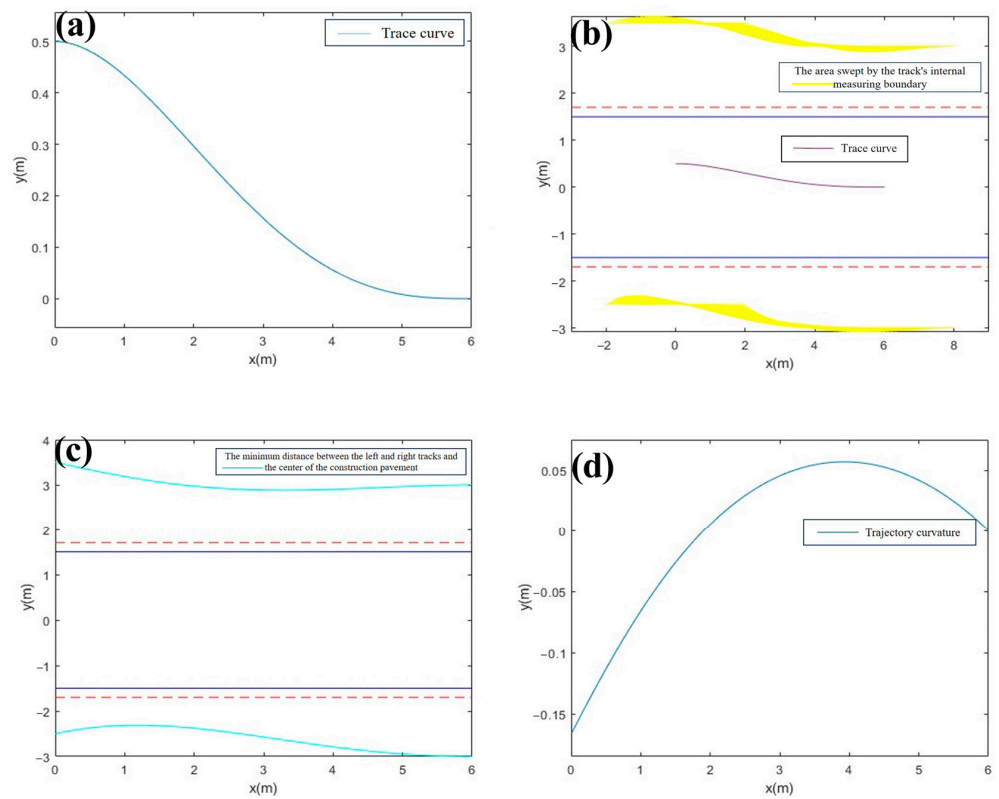
The initial position of the vehicle obtained by image processing is  $[x_0, y_0, \varphi_0] = [0, 0.2, 0.4]$ . The sampling steps of curvature and snap in the optimization function are both 0.1 m. The weight coefficients of the optimization function are as follows:  $a_t = b_t = 1 (t \in [1, 60])$ ,  $c_t = 100$ , and the optimization algorithm adopts the interior point method. The output of quartic polynomial trajectory planning is as follows:

$$y = -0.0014x^4 + 0.0293x^3 - 0.1991x^2 + 0.4x + 0.2 \quad (15)$$

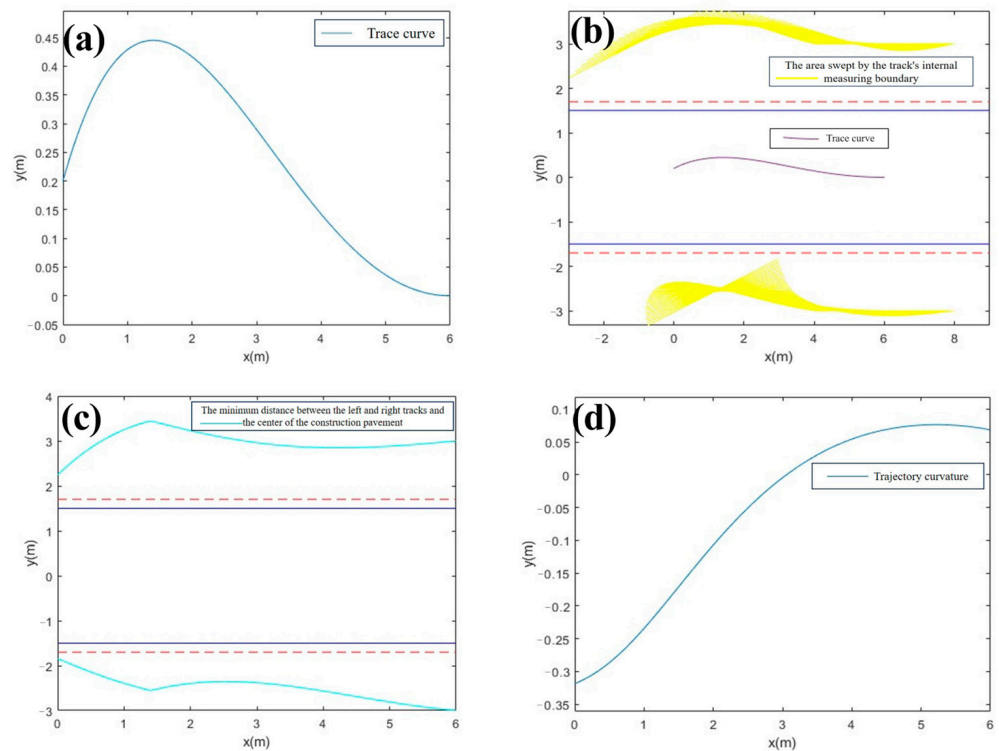
The curvature at the end point is  $6.8 \times 10^{-2}$ , the yaw angle is  $1.69 \times 10^{-5}$ , and the vertical coordinate of the geometric center of the track vehicle is  $4 \times 10^{-5}$ . It can be seen that the trajectory planning meets the constraint and optimization requirements of the motion state of the end point.

The trajectory curve, the area swept by the inside of the left and right tracks, the minimum distance of the left and right tracks from the center of the pavement to be constructed and the curvature change curves of this trajectory are shown in Figure 5a, b, c and d, respectively.

In Figures 4 and 5, the minimum distance between the left and right tracks and the center of the pavement to be constructed does not exceed the safety limit, the trajectory meets the constrained boundary conditions, the absolute value of the curvature of the trajectory is within 0.5 and the trajectory at the end is stable, achieving the optimization objective of the optimization function.



**Figure 4.** Simulation output of working condition 1: (a) trace curve; (b) the area swept by the inside of the left and right tracks; (c) the minimum distance between the left and right tracks and the center of the road surface to be constructed; (d) trajectory curvature.



**Figure 5.** Simulation output of working condition 2: (a) trace curve; (b) the area swept by the inside of the left and right tracks; (c) the minimum distance between the left and right tracks and the center of the road surface to be constructed; (d) trajectory curvature.



### 3.5. Trajectory Tracking Control Based on MPC

#### 3.5.1. Basic Principles of Model Predictive Control

The model predictive control mechanism is shown in Figure 6, which can be roughly summarized as follows: at the current sampling moment, solve an open-loop optimization problem within the prediction step according to the current moment state and the prediction model; next, apply the first element in the optimization output to the controlled object, and at the next sampling moment, use the new actual state quantity as the initial condition and repeat the above steps until the end of the problem. Model predictive control consists of three important components: predictive modeling, rolling optimization and feedback correction.

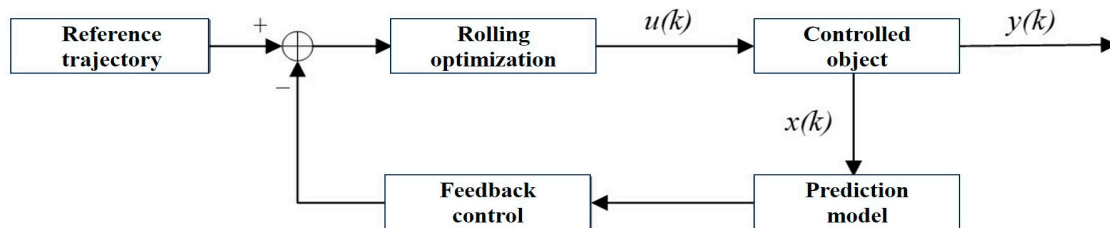


Figure 6. MPC basic principle.

The biggest advantage of the MPC control method lies in its explicit multi-constraint processing ability, which can easily and effectively represent constraints in quadratic programming or nonlinear optimization problems by adding constraints to control, state and prediction quantities. However, the MPC method itself is online to solve the optimization problem, and if the solution problem is more complex, it is difficult to meet the requirements of the response speed of the control algorithm in practical engineering. According to whether the control system is linear or not, MPC can be categorized into linear MPC and nonlinear MPC.

#### 3.5.2. Nonlinear Model Predictive Control

In the nonlinear MPC control algorithm:

$$J(k) = \sum_{i=1}^{N_p} \left\| \chi(k+i) - \chi_{ref}(k+i) \right\|_Q^2 + \sum_{i=1}^{N_c-1} \left\| \Delta u(k+i) \right\|_R^2 + \rho \varepsilon^2 \quad (16)$$

$$s.t. \quad \begin{bmatrix} u_{min} \\ \Delta u_{min} \\ \chi_{min} \end{bmatrix} \leq \begin{bmatrix} u_k \\ \Delta u_k \\ \chi_k \end{bmatrix} \leq \begin{bmatrix} u_{max} \\ \Delta u_{max} \\ \chi_{max} \end{bmatrix}$$

The first term in this objective function reflects the tracking ability of this control algorithm to the target trajectory, the second term reflects the smoothness of the change in the control quantity in this control algorithm,  $\rho$  is the weight and  $\varepsilon$  is the relaxation factor, which is designed to avoid the situation of no solution to the optimization problem. In the optimization solution process, the control quantity, control increment and state quantity should satisfy the constraints. The above nonlinear optimization problem is solved by taking  $\Delta U = [\Delta u(k), \Delta u(k+1), \dots, \Delta u(k+N_c-1)]$  as a parameter. The nonlinear MPC control can be achieved by applying the first element of the optimized result sequence  $\Delta U$  to the controlled object and repeating the above steps at the next moment.



### 3.5.3. Linear Model Predictive Control

Similarly, the optimization objective function is defined as follows:

$$J(k) = \sum_{i=1}^{N_p} \left\| \eta(k+i) - \eta_{ref}(k+i) \right\|_Q^2 + \sum_{i=1}^{N_c-1} \left\| \Delta u(k+i) \right\|_R^2 + \rho \varepsilon^2 \quad (17)$$

$$s.t. \quad \begin{bmatrix} u_{min} \\ \Delta u_{min} \\ \chi_{min} \end{bmatrix} \leq \begin{bmatrix} u_k \\ \Delta u_k \\ \chi_k \end{bmatrix} \leq \begin{bmatrix} u_{max} \\ \Delta u_{max} \\ \chi_{max} \end{bmatrix}$$

The ideal control effect of the simulation and experiment is that the vehicle running the trajectory and the target trajectory coincide, so  $Y_{ref} = [0, 0, \dots, 0]^T$ . Combined with Equation (16), the optimization function can be rewritten as follows:

$$J(k) = \Delta U^T (\Theta^T Q \Theta + R) \Delta U + 2E^T Q \Theta \Delta U + \rho \varepsilon^2 + E^T Q E \quad (18)$$

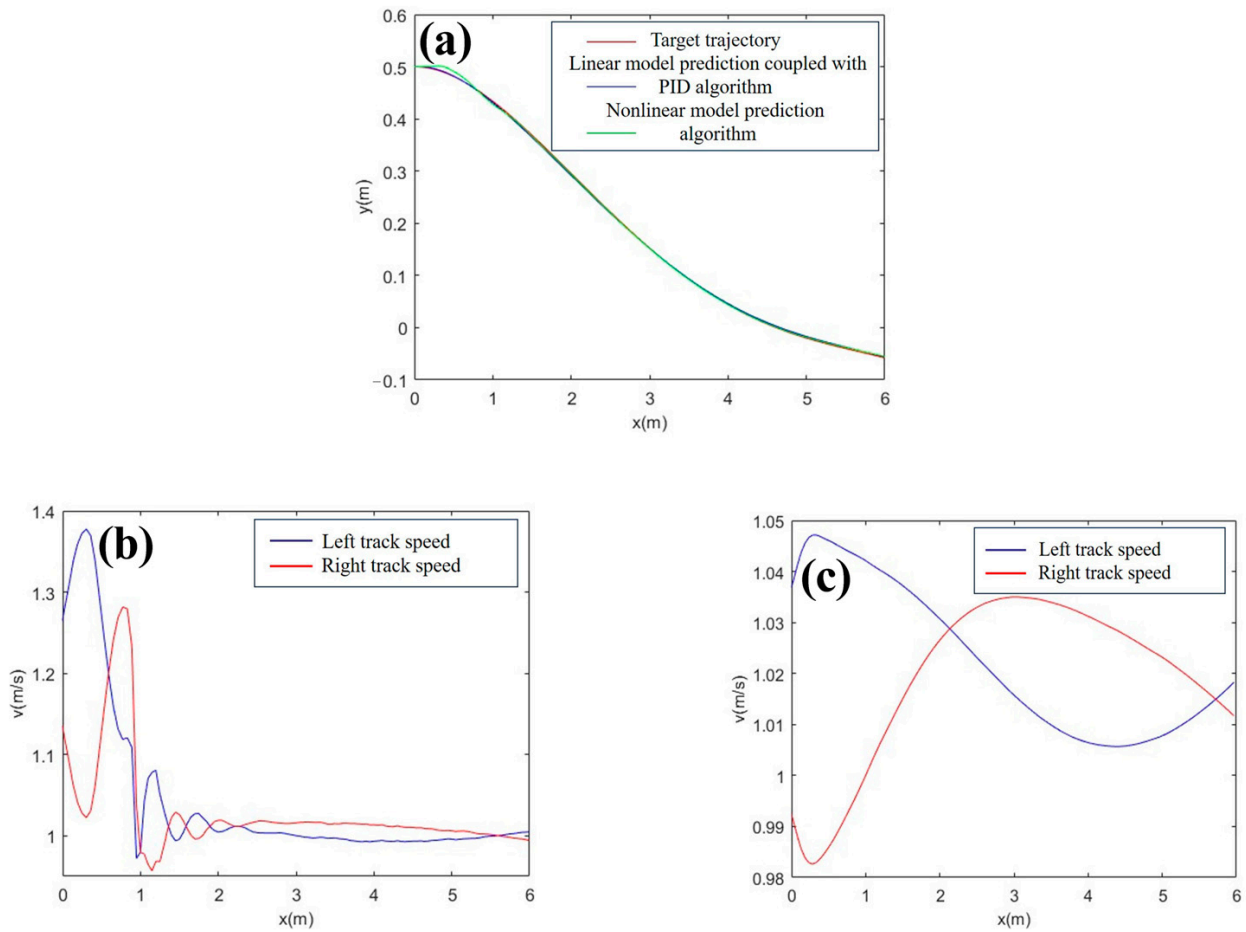
$$s.t. \quad \begin{bmatrix} u_{min} \\ \Delta u_{min} \\ \chi_{min} \end{bmatrix} \leq \begin{bmatrix} u_k \\ \Delta u_k \\ \chi_k \end{bmatrix} \leq \begin{bmatrix} u_{max} \\ \Delta u_{max} \\ \chi_{max} \end{bmatrix}$$

Taking  $\Delta U$  as a parameter, the above objective function is transformed into a quadprog quadratic programming problem, the first element in the optimized sequence of results  $\Delta U$  is applied to the controlled object and the above steps are repeated at the next moment to realize the model predictive control. Since the quadratic programming problem has a more complete and efficient solution method, the model is faster and can be combined with PID auxiliary control in practical applications to realize constant speed in the longitudinal direction as much as possible.

### 3.5.4. CoppeliaSim and MATLAB-Based Model Prediction Co-Simulation

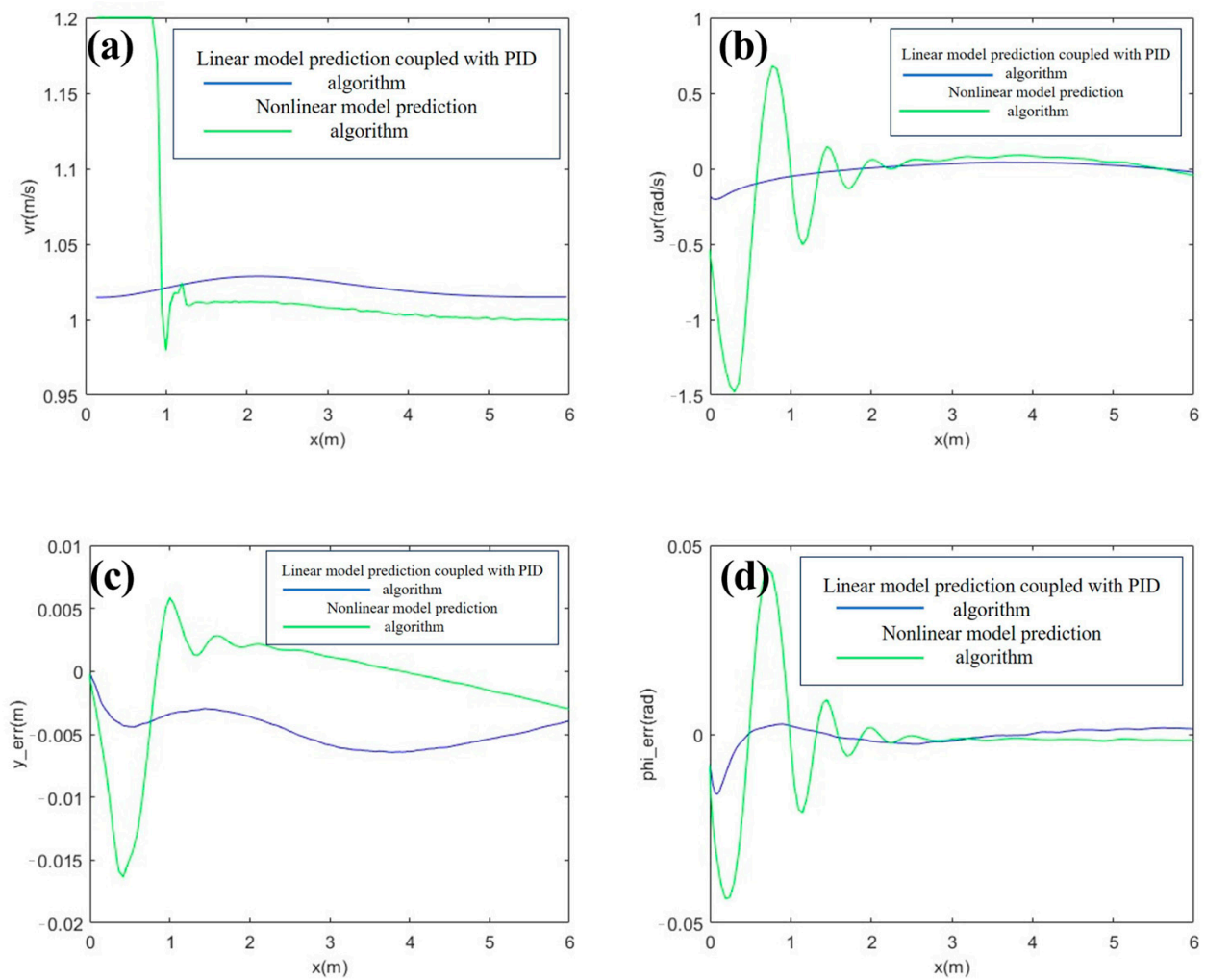
Previously, in the MATLAB-based simulation, the integral of the velocity was used instead of the actual vehicle position, but in real working conditions, due to the interference of mechanical structure, road conditions, etc., there will be a deviation between the actual position of the tracked vehicle and the integral position of the velocity; therefore, the model prediction based on MATLAB 2023A and CoppeliaSim V4.5.1 is considered a joint simulation to validate the algorithm's performance of the target in real physical scenarios.

The model of the tracked vehicle in CoppeliaSim V4.5.1 gives the main parameters: four grounded wheels on the left and right and four joints on the left and right, corresponding to four grounded wheels on the left and right of the tracked vehicle, with the rest of the tracked vehicle weighing 200 kg. The rest are driven wheels, and the weight of the left and right tracks is 200 kg. The model is symmetrical and only considers the mechanical relationship between the body and the tires, ignoring the suspension and other tracked vehicle transmission structures. The ground friction factor is 0.68, and the ground is horizontal and isotropic.  $T = 0.05$  s is used as the synchronous simulation step, and the CoppeliaSim V4.5.1 interface is used in MATLAB 2023A to control the model and read the model motion data in real time. The read position and velocity data are used as the actual state quantities, and the reference points of each step in the nonlinear model prediction are set as follows: control step  $N_c = 3$ , prediction step  $N_p = 30$  and sampling step  $T = 0.06$  s. The boundary values are set as velocity  $v(-2, 2$  m/s), angular velocity  $\omega(-0.5, 0.5$  rad/s), acceleration  $a(-0.2, 0.2$  m/s<sup>2</sup>), angular acceleration  $\delta(-0.1, 0.1$  rad/s<sup>2</sup>) and both left and right wheel velocities  $(-2, 2$  m/s). The reference point for each step in the linear model prediction and PID coupled control is the point on the reference trajectory that is closest to the actual position. The typical working condition I in Section 3.4 is simulated and verified, and the output of the joint simulation is shown in Figure 7.



**Figure 7.** Co-simulation output results: (a) trace curve; (b) linear models predict left and right track speeds; (c) nonlinear model predicts PID coupled left and right track speed.

As can be seen in Figure 7, both models can track the trajectory better, and the nonlinear model prediction control has a smaller amount of change in speed and no obvious oscillation due to the addition of PID coupling control. The analysis of the joint simulation results and their deviations are shown in Figure 8. On the whole, the speed curve of linear model prediction and PID coupling control is more stable, but the angular speed difference between the two control methods at the end point is not much, and both meet the expectation of smooth stopping. The lateral position deviation in linear model prediction and PID coupling control is within 0.01 m, and that in nonlinear model prediction control is within 0.02 m, both of which are much smaller than the safety limit of 0.2 m, so both controls satisfy the collision limit requirements. The amplitude of the transverse swing angle deviation in the nonlinear model prediction control is larger, which is because the vehicle position deviation is larger and the linear velocity is smaller in the starting stage, which easily has small overshoots and oscillation. The linear model prediction algorithm joins the PID speed controller to reduce the instability in the starting stage effectively. Both the linear model prediction and PID coupling algorithm and the nonlinear model prediction algorithm can excellently accomplish the trajectory tracking task, but the former is better than the latter in terms of speed tracking, smoothness and tracking performance in the start phase.



**Figure 8.** Analysis of co-simulation output results: (a) linear velocity curve; (b) angular velocity curve; (c) lateral position deviation; (d) yaw angle deviation.

#### 4. Experimental Details and Validation

There are eight joints in Figure 9, four on each side, corresponding to four grounding wheels on the left and right side of the track car, and the weight of the left and right track was 200 kg. CoppeliaSim V4.5.1, formerly known as V-rep, is a professional robot modeling and simulation software which can edit the geometry and mechanical properties of the model and has a physical field close to the real one. The model built in this software can be used to verify and analyze the theoretical model, and the software can be co-simulated with MATLAB 2023A. It can also be used to verify the control algorithm in the subsequent controller design. The frame length of the track vehicle was 1.08 m, and the width was 0.28 m. The motor power supply, camera, controller and other parts were placed on the frame. The motor adopts DJI RoboMaster M3508 and DJI RoboMaster C620 were selected for electrical modulation; the motor controller adopted DJI RoboMaster. The field test ground was asphalt ground, and the test distance was 3 m. The zoom ratio between the test vehicle and the actual track vehicle was about 1:7, so the imaginary road width was 0.43 m, the imaginary safety distance was 0.03 m and the imaginary longitudinal linear speed was 0.25 m/s during trajectory planning.

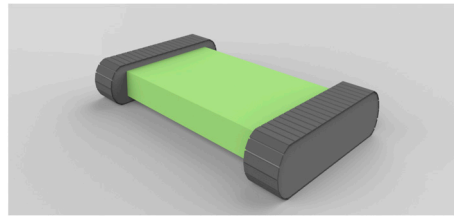


Figure 9. Coppeliasim model diagram.

To fully verify the actual control effect of the driving controller, the test is divided into two groups, based on two different initial state inputs, to verify the effectiveness of the trajectory planning and the accuracy of trajectory tracking. In test 1, the initial state quantity of the tracked vehicle was  $[x_0, y_0, k_0] = [0, -0.088, -0.35]$ . The test results are shown in Figure 10. In test 2, the initial state quantity of the tracked vehicle was  $[x_0, y_0, k_0] = [0, 0.132, 0.23]$ , and the test results are shown in Figure 11. The deviation in test 2 is smaller than that of test 1. Due to the effect of centrifugal force, the load of the right track in test 1 is larger, and the load of the left track in test 2 is larger. Both experiments meet the boundary constraints and errors.

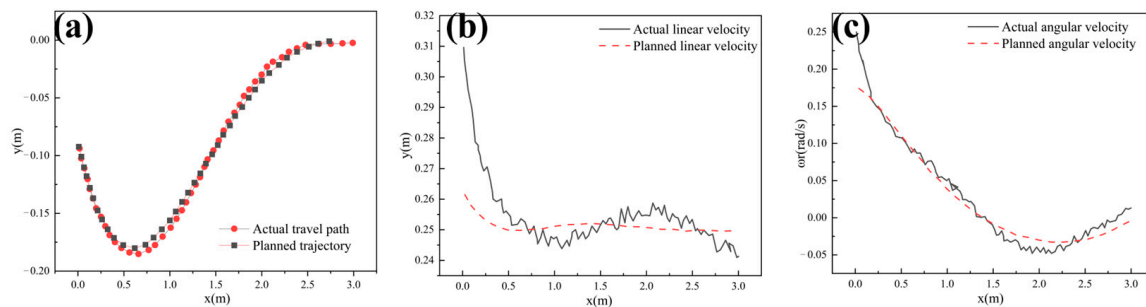


Figure 10. Test 1 result. (a) Trajectory tracking effect; (b) linear speed tracking effect; (c) angular velocity tracking effect.

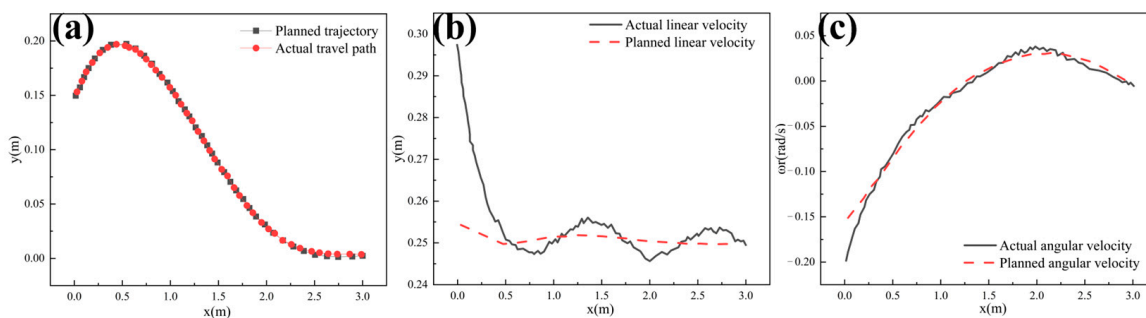


Figure 11. Test 2 result. (a) Trajectory tracking effect; (b) linear speed tracking effect; (c) angular velocity tracking effect.

## 5. Conclusions

This paper took the crawler vehicle traveling control system as the research object and introduced the target trajectory planning and trajectory tracking algorithms required for step traveling control in detail. In order to fit the actual motion law of the crawler, quartic polynomial trajectory planning was adopted to explore the coordinate conversion of the collision corner point of the crawler, the collision limit and the design of the optimization function of trajectory planning, and the trajectory planning simulation was carried out to verify the two typical conditions, which proves that the quartic polynomial trajectory planning is able to excellently complete the optimization design tasks for the curvature of the trajectory, the degree of trajectory urgency, the end point vicinity and the smoothness of the

trajectory while meeting the condition of the collision limit constraints. The optimization of trajectory curvature, trajectory rapidity and smoothness near the end point is demonstrated. The model prediction algorithm was used to track the target trajectory delineated by the quartic polynomial trajectory, the linear model prediction coupled with PID control algorithm was explored on the basis of the nonlinear model prediction algorithm and the tracking ability of the two algorithms in a real physical scenario was explored by using the joint simulation of CoppeliaSim V4.5.1 and MATLAB 2023A, which verifies the theoretical validity of the used tracked vehicle dynamics model and controller design.

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

**Funding:** This research was funded by Zhejiang Province Spearhead and Leading Goose Research and Development Key Program, grant number 2023C01239.

**Data Availability Statement:** The original contributions presented in the study are included in the article; further inquiries can be directed to the corresponding author.

**Conflicts of Interest:** Author J.Z. was employed by D.R. (Zhejiang) Powertrain Technology Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest. The funder had no role in the design of the study; in the collection, analyses or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

## References

1. Li, Z.; Zhang, J.; Guan, H. Passenger spatiotemporal distribution prediction in airport terminals based on physics-guided spatio-temporal graph convolutional network and its effect on indoor environment prediction. *Sustain. Cities Soc.* **2024**, *106*, 105375. [\[CrossRef\]](#)
2. Yan, X.; Fascetti, A.; Vandenbossche, J.M.; Darnell, M. Computer Vision-Based Estimation of The Effects of Vibration in Slipform Paving. *Transp. Res. Rec.* **2024**. [\[CrossRef\]](#)
3. Zhan, Y.; Zhang, Y.; Nie, Z.; Luo, Z.; Qiu, S.; Wang, J.; Zhang, A.A.; Ai, C.; Tang, X.; Tan, C. Intelligent paving and compaction technologies for asphalt pavement. *Autom. Constr.* **2023**, *156*, 105081. [\[CrossRef\]](#)
4. Khirid, T.; Kangude, P.; Kedari, S. Construction of Rigid Pavement by Slip form Paver. *Int. J. Res. Appl. Sci. Eng. Technol.* **2023**, *11*, 1182–1184. [\[CrossRef\]](#)
5. Pickard, A.; Frappe, H.; Gentile, P.A.; Komoncharoensiri, T.; Tregger, N. Replacing cement through data analytics and process control. *Cem. Concr. Res.* **2023**, *169*, 107163. [\[CrossRef\]](#)
6. Ram, P.V.; Van Dam, T.J.; Sutter, L.L.; Anzalone, G.; Smith, K.D. Field Study of Air Content Stability in the Slipform Paving Process. *Transp. Res. Rec.* **2014**, *2408*, 55–85. [\[CrossRef\]](#)
7. Kawashima, S.; Wang, K.; Ferron, R.D.; Kim, J.; Tregger, N.; Shah, S. A review of the effect of nanoclays on the fresh and hardened properties of cement-based materials. *Cem. Concr. Res.* **2021**, *147*, 106502. [\[CrossRef\]](#)
8. Rasmussen, R.O.; Karamihas, S.M.; Cape, W.R.; Chang, K.G.; Guntert, R.M., Jr. Stringline effects on concrete pavement construction. *Transp. Res. Record.* **2004**, *1900*, 3–11. [\[CrossRef\]](#)
9. Zhang, Z.; Liu, H.; Rai, L.; Zhang, S. Vehicle Trajectory Prediction Method Based on License Plate Information Obtained from Video-Imaging Detectors in Urban Road Environment. *Sensors* **2020**, *20*, 1258. [\[CrossRef\]](#)
10. Jones, J.A.; Hopper, J.E.; Bolas, M.T.; Krum, D.M. Orientation Perception in Real and Virtual Environments. *IEEE Trans. Vis. Comput. Graph.* **2019**, *25*, 2050–2060. [\[CrossRef\]](#)
11. Choi, M.; Kim, M.; Kim, G.; Kim, S.; Park, S.C.; Lee, S. 3D Scanning Technique for Obtaining Road Surface and Its Applications. *Int. J. Precis. Eng. Manuf.* **2017**, *18*, 367–373. [\[CrossRef\]](#)
12. Yang, X.; Ni, J. A cloud-edge combined control system with MPC parameter optimization for path tracking of unmanned ground vehicle. *Proc. Inst. Mech. Eng. Part D* **2023**, *237*, 48–60. [\[CrossRef\]](#)
13. Paliotta, C.; Lefeber, E.; Pettersen, K.Y.; Pinto, J.; Costa, M. Trajectory Tracking and Path Following for Underactuated Marine Vehicles. *IEEE Trans. Control. Syst. Technol.* **2019**, *27*, 1423–1437. [\[CrossRef\]](#)
14. Chen, Y.; Peng, H.; Grizzle, J.W. Fast Trajectory Planning and Robust Trajectory Tracking for Pedestrian Avoidance. *IEEE Access* **2017**, *5*, 9304–9317. [\[CrossRef\]](#)
15. Li, J.; Chen, C.K.; Ren, H. Time-Optimal Trajectory Planning and Tracking for Autonomous Vehicles. *Sensors* **2024**, *24*, 3281. [\[CrossRef\]](#)

16. Xu, Z.; Li, J.; Xiao, F.; Zhang, X.; Wang, D.; Qi, C.; Peng, S.; Song, S.; Wang, J. Energy-Saving Model Predictive Cruise Control Combined with Vehicle Driving Cycles. *Int. J. Automot. Technol.* **2022**, *23*, 439–450. [[CrossRef](#)]
17. Fu, Z.; Chu, S.C.; Watada, J.; Hu, C.C.; Pan, J.S. Software and hardware co-design and implementation of intelligent optimization algorithms. *Appl. Soft Comput.* **2022**, *129*, 109639. [[CrossRef](#)]
18. Jin, X.; Tao, Y.; Opinat Ikiela, N.V. Trajectory Planning Design for Parallel Parking of Autonomous Ground Vehicles with Improved Safe Travel Corridor. *Symmetry* **2024**, *16*, 1129. [[CrossRef](#)]
19. Chen, W.; Liu, F.; Zhao, H. Trajectory-Tracking Control of Unmanned Vehicles Based on Adaptive Variable Parameter MPC. *Appl. Sci.* **2024**, *14*, 7285. [[CrossRef](#)]
20. Qiao, L.; Luo, X.; Luo, Q. Control of Trajectory Tracking for Mobile Manipulator Robot with Kinematic Limitations and Self-Collision Avoidance. *Machines* **2022**, *10*, 1232. [[CrossRef](#)]
21. Zhao, C.; Zhu, Y.; Du, Y.; Liao, F.; Chan, C.Y. A Novel Direct Trajectory Planning Approach Based on Generative Adversarial Networks and Rapidly-Exploring Random Tree. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 17910–17921. [[CrossRef](#)]
22. Ali, S.M. Advancements in Precision Trajectory Generation: A Synergistic Approach to Design and Implementation for the 5-DOF Alpha-II Robot's Digital and Physical Twins. *Int. J. Sci. Res. Eng. Manag.* **2024**, *8*, 27754. [[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.