

Article

Path Navigation and Precise Deviation Correction Control for Tracked Roadheaders in Confined Roadway Spaces of Underground Coal Mines

Rui Li ^{1,2}, Dongjie Wang ^{1,*}, Weixiong Zheng ³, Tong Li ¹ and Miao Wu ¹

¹ School of Mechanical, Electrical & Information Engineering, China University of Mining & Technology (Beijing), Beijing 100083, China; lirui202504@163.com (R.L.); liyingna024@163.com (T.L.); wum@cumtb.edu.cn (M.W.)

² Xi'an Institute of Mechanical and Electrical Information Technology, Xi'an 710000, China

³ School of Energy and Power, Tsinghua University, Beijing 100083, China; zhengwx@mail.tsinghua.edu.cn

* Correspondence: wdj@student.cumtb.edu.cn

Abstract

Aiming at the complex construction environment and autonomous navigation challenges in underground coal mine roadways, this paper proposes a path navigation and deviation correction control method for tracked roadheaders in confined roadway spaces. First, a two-dimensional planar grid model of the working scenario was constructed, with dimensionality reduction in the roadway model achieved through a heading reference influence degree threshold of the tracked roadheaders. Based on the kinematics theory of tracked roadheaders, kinematic and dynamic models for deviation correction in fully mechanized excavation roadways were established. Subsequently, a path planning and tracking correction algorithm was developed, along with a heading deviation correction control algorithm based on fuzzy neural network PID. Online optimization of the particle swarm algorithm was realized through crossover-mutation operations, enabling optimal strategy solving for construction path planning and precise control of travel deviation correction. Finally, simulation experiments evaluating algorithm performance and comparative simulations of control algorithms validated the feasibility and superiority of the proposed method. This research provides strategic guidance and theoretical foundations for rapid precision deployment and intelligent deviation correction control of tracked engineering vehicles in confined underground coal mine spaces.

Keywords: restricted roadway space; tracked roadheaders; particle swarm algorithm; path planning; deviation correction control

MSC: 93C85



Academic Editor: Davide Valenti

Received: 25 April 2025

Revised: 3 June 2025

Accepted: 18 June 2025

Published: 9 August 2025

Citation: Li, R.; Wang, D.; Zheng, W.; Li, T.; Wu, M. Path Navigation and Precise Deviation Correction Control for Tracked Roadheaders in Confined Roadway Spaces of Underground Coal Mines. *Mathematics* **2025**, *13*, 2557. <https://doi.org/10.3390/math13162557>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The construction environment of fully mechanized excavation roadways in underground coal mines faces severe spatial constraints, complex geological conditions, and multi-process parallel operations [1]. Traditional sequential manual operations exhibit prolonged cycle times (averaging 30% longer than automated systems) and significant safety hazards, with human exposure to roof collapses and gas leaks accounting for 18% of annual mining accidents in China [1]. To address these challenges, tracked roadheaders—including excavation robots and temporary support robots—have been

deployed in recent years. However, their path navigation and deviation correction capabilities in confined roadway spaces remain critically limited. Unconventional road conditions (e.g., undulating floors with height variations exceeding ± 50 mm) induce track slippage rates of 15–25%, causing steering deviations that degrade path-tracking accuracy by 20–40% [2,3], directly compromising construction efficiency and safety.

Existing research predominantly focuses on robot navigation in open environments, neglecting the unique constraints of underground mines. While Niu et al. [4] improved the A* algorithm for global path planning, their model ignored dynamic obstacles like moving support equipment; Wang et al. [5] applied model predictive control (MPC) to trajectory tracking, but its computational complexity (>200 ms per cycle) is infeasible for real-time coal mine operations; Sa-putra et al. [6] leveraged visual SLAM for roadway modeling, yet sensor degradation in low-light, high-dust environments limits localization accuracy to ± 0.5 m—insufficient for centimeter-level deviation correction. Furthermore, traditional PID control struggles with the nonlinear hydraulic dynamics of tracked roadheaders [7,8], requiring manual retuning under varying slip conditions, while fuzzy control lacks self-learning capabilities to handle unmodeled terrain profiles [9,10]. These limitations manifest in three core gaps for confined roadway applications: (1) Over-simplified environmental modeling that fails to distinguish traversability weights for gradient-dependent surfaces; (2) inadequate kinematic accuracy due to neglecting slip-induced shifts in the instantaneous rotation center, causing cumulative pose errors; (3) real-time vs. robustness trade-offs where optimization algorithms converge slowly and stagnate in local minima during dynamic obstacle avoidance.

To overcome these limitations, this paper establishes an integrated framework synergizing scenario modeling, slip-compensated kinematics, and adaptive control. First, a dimensionality-reduced grid model incorporates a heading reference influence degree metric to prioritize critical terrain features (e.g., convex floors with $|z| > 30$ mm). Second, a slip-corrected kinematic model explicitly quantifies track slippage effects on turning radius and angular velocity. Third, an adaptive path planner using crossover-mutation particle swarm optimization (PSO) achieves 89% faster convergence than standard PSO by enriching population diversity through stochastic crossover operations (Section Adaptive Particle Swarm Path Point Planning Algorithm). Finally, a fuzzy neural network PID (FNN-PID) controller dynamically adjusts gains via online learning (Section 5.2), enhancing robustness against unmodeled disturbances. Validated through simulations and field trials in the Datong mining district, our method reduces path-tracking error by 40%, cuts response time by 35%, and maintains stability under 20% slippage—enabling safe, efficient deployment of tracked robots in high-risk confined spaces.

2. Working Condition Scenario Modeling

2.1. Parallel Construction Environment of Excavation-Support-Bolting Operations

2.1.1. Traditional Roadway Construction Process

The conventional roadway construction process in coal mines comprises six sequential stages: undercutting, coal fragling, material haulage, bolt-hole drilling, mesh installation, and permanent anchoring. These discrete technical procedures necessitate manual execution in strict linear sequence to maintain safety protocols, as illustrated in Figure 1. This labor-intensive methodology not only requires substantial on-site personnel deployment but also compromises operational efficiency due to workflow interdependencies and process discontinuities.

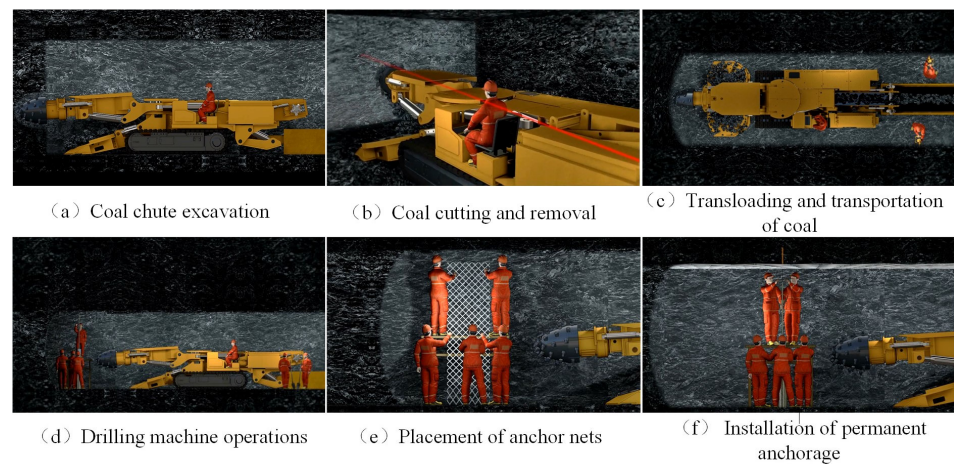


Figure 1. Traditional roadway construction process in coal mines.

2.1.2. Integrated Parallel Construction Technology of Excavation-Support-Bolting Operations

To enhance the construction efficiency of roadway formation and excavation operations, an advanced self-propelled temporary support system is deployed for proactive ground control in fully mechanized excavation faces. Building upon this, we propose the parallel construction framework for robotic excavation equipment clusters [11], which integrates the following coordinated robotic units:

- (1) Excavation Robot: Equipped with autonomous pose detection, cutting trajectory optimization, adaptive coal fragmentation, and self-correcting deviation control capabilities;
- (2) Temporary Support Robot: Features adaptive support force regulation and real-time posture adjustment through hydraulic feedback mechanisms;
- (3) Support Transport Robot: Tasked with relocating retracted temporary supports to the excavation face, incorporating telescopic width adjustment and adaptive support protection functions;
- (4) Drilling–Bolting Robot: Enables automated drilling and synchronized resin anchoring operations;
- (5) Material Carrier Robot: Implements continuous high-efficiency ore haulage from the excavation face via conveyor-interfaced navigation [12].

The parallel construction operation line of excavation, support, and anchoring formed by the robotic excavation machine group and the tracked special robots are shown in Figure 2.

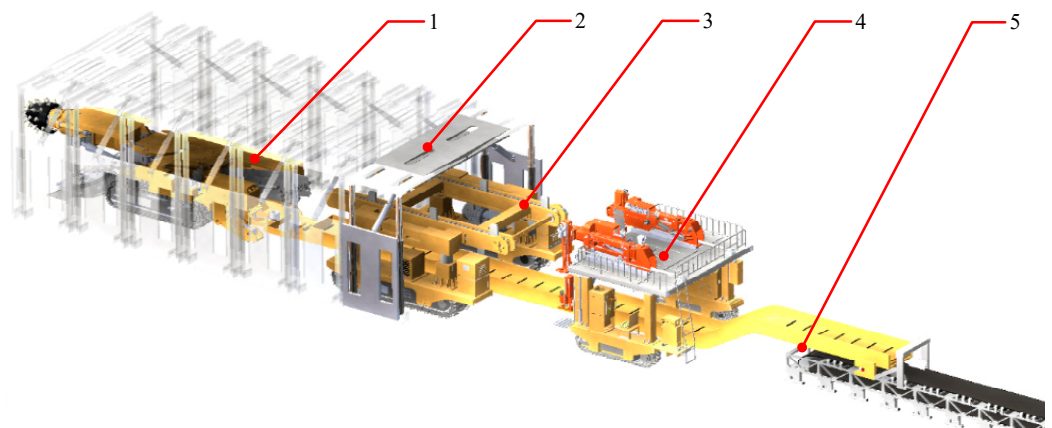


Figure 2. Fast excavation construction operation line with parallel excavation, support, and anchoring (1—excavation robot, 2—temporary support robot, 3—support transport robot, 4—drilling and anchor robot, 5—carrier robot).

of the tracked special robot, the impact of each unconventional road condition on the navigation correction and mobilization of the tracked roadheaders varies. Therefore, this paper proposes the concept of heading reference influence degree, indicating the extent to which unconventional road conditions in the roadway affect the heading correction motion control of the tracked roadheaders.

The calculation method for the heading reference influence degree is as follows:

$$\Omega_i = \left(\zeta_1 \frac{S_n}{S_{\max}} + \zeta_2 \frac{i_n}{1 - i_{\min}} + \zeta_3 \frac{L_{Yn}}{L_{Y\max}} + \zeta_4 \frac{D_{Xn}}{D_{X\max}} + \zeta_5 \left| \frac{Z_n}{Z_{\max}} \right| \right) \quad (1)$$

In Equation (1), Ω_i is the heading reference influence degree; ζ is the weight coefficient for selecting various attributes in the heading reference influence degree; i_n is the slip rate of the n th unconventional road condition area; $i_{\min}$ is the minimum slip rate of unconventional road conditions; $S_{\max}$ is the maximum area of unconventional road conditions, m^2 ; $L_{Y\max}$ is the maximum distance along the Y -axis of the unconventional road condition area, m ; $D_{X\max}$ is the maximum distance along the X -axis of the unconventional road condition area, m ; $Z_{\max}$ is the absolute value of the maximum vertical distance between the convex or concave trench of the unconventional road condition area and the roadway horizontal plane, m .

In the above equation, the weight coefficient ζ is allocated based on the driving performance of tracked engineering vehicles in coal mines and the practical experience at construction sites. The specific allocation and assignment of the weight coefficient ζ are as follows:

$$\zeta_i = \begin{cases} [0.2, 0.35, 0.15, 0.05, 0.25] (Z_n < 0) \\ [0.2, 0.35, 0.05, 0.15, 0.25] (Z_n > 0) \end{cases} \quad (2)$$

When solving the navigation correction strategy for tracked roadheaders in complex road conditions, the difficulty of solving increases with the increase in the dimensionality of environmental factors. In the two-dimensional roadway plane model, the number of grids in unconventional road condition areas is known. To simplify the solving model, road condition areas with a smaller heading reference influence degree can be eliminated, thereby reducing the difficulty of solving the navigation correction strategy. The position interval $[X_{t1}, X_{t2}]$, width interval $[D_{t1}, D_{t2}]$, height interval $[Z_{t1}, Z_{t2}]$, area interval $[S_{t1}, S_{t2}]$, and road surface passability interval $[I_{t1}, I_{t2}]$ are set. The retention and deletion of area grids follow the following principles:

(1) If the position attribute parameter X_n of an unconventional area exceeds the maximum value X_{t2} of the position interval but has a small impact on the driving performance of the tracked roadheaders, the grid of this unconventional road condition area is retained and marked as a passable area;

(2) If the width attribute parameter D_n and height attribute parameter Z_n of an unconventional area exceed the maximum values D_{t2} and Z_{t2} of the width interval, the grid of this unconventional road condition area is retained and marked as an impassable area;

(3) If the road surface passability parameter value I_n exceeds the maximum value I_{t2} of the interval, the grid of this unconventional road condition area is retained and marked as an impassable area;

(4) Except for the above situations, set a fixed threshold N_t for the heading reference influence degree, and retain the grid of unconventional road condition areas where the heading reference influence degree value is greater than the threshold N_t ; otherwise, it will be removed.

3. Motion Analysis and System Identification of Tracked Roadheaders

3.1. Navigation Motion Analysis of Tracked Roadheaders

When analyzing the steering motion state of tracked roadheaders in the construction environment of comprehensive excavation tunnels, it is first necessary to make conditional assumptions to simplify the analysis process:

- (1) Consider the components of the tracked roadheaders as rigid bodies, and the driving wheels and track chains do not deform during motion;
- (2) Under the vertical two-dimensional plane projection, the center of mass and the geometric center of the tracked roadheaders are considered to coincide;
- (3) Ignore the influence of internal resistance during the movement of the tracked roadheaders;
- (4) When the tracked roadheaders are moving, the ground pressure between the track and the ground is uniformly distributed;
- (5) Under ideal conditions, the resistance from the floor surface on the left and right tracks of the tracked roadheaders is equal.

According to Figure 3, when the tracked roadheaders are performing a steering motion, the expressions for the robot's center of mass velocity and the velocities of the two tracks are, respectively, [13,14] as follows:

$$\begin{cases} v = \omega R = \frac{v_1 + v_2}{2} \\ v_1 = (\frac{B}{2} + R)\omega = \omega_1 r \\ v_2 = (\frac{B}{2} - R)\omega = \omega_2 r \end{cases} \quad (3)$$

In the equation, v is the linear velocity of the vehicle's center of mass during steering motion, m/s; v_1 is the linear velocity of the outer track during steering motion, m/s; v_2 is the linear velocity of the inner track during steering motion, m/s; ω is the angular velocity of the vehicle's center of mass during steering motion, rad/s; ω_1 is the angular velocity of the outer track during steering motion, rad/s; ω_2 is the angular velocity of the inner track during steering motion, rad/s; B is the distance between the centers of the two tracks, m; R is the turning radius of the vehicle, m; r is the pitch circle radius of the track driving chain, m.

From Equation (3) above, the steering angular velocity ω and turning radius R of the tracked roadheaders are as follows:

$$\begin{cases} \omega = \frac{v_1 - v_2}{B} \\ R = \frac{B}{2} \cdot \frac{v_1 + v_2}{v_1 - v_2} \end{cases} \quad (4)$$

Based on the analysis of Equation (4) above:

- (1) When $v_1 = v_2$, $\omega = 0$, $R \rightarrow \infty$, the vehicle will travel in a straight line;
- (2) When $v_1 + v_2 = 0$ and both v_1 and v_2 are non-zero, $R = 0$, the instantaneous center of rotation coincides with the center of mass, and the vehicle will perform a pivot turn around its geometric center, with the rotation center called the central turning point;
- (3) When the vector directions of v_1 and v_2 are the same, $R > 0.5B$, the vehicle will perform a turning motion around the turning center outside the vehicle body with a large turning radius;
- (4) When $v_2 = 0$, then $R = 0.5B$, the inner track of the vehicle remains stationary under ideal conditions, satisfying zero linear velocity, and the vehicle performs a turning motion around the geometric center of the inner track, i.e., a single-side braking turning motion.

In the above analysis of the turning motion of the tracked roadheaders, the linear velocities v_1 and v_2 of the tracks on both sides of the tracked roadheaders are theoretical values under ideal conditions without relative slippage. However, during actual tunnel

construction, the vehicle's tracks are often affected by the contact surface, causing slippage, which leads to discrepancies between the theoretical and actual values of the track linear velocities. Therefore, when analyzing the turning motion of the tracked roadheaders, the slippage phenomenon of the tracks must also be comprehensively considered [15,16]. The slip rates of the tracks on both sides in the tunnel environment can be expressed as follows:

$$\begin{cases} i_1 = \frac{v_1 - v'_1}{v_1} \times 100\% \\ i_2 = \frac{v_2 - v'_2}{v_2} \times 100\% \end{cases} \quad (5)$$

In the equation, i_1 and i_2 are the slip rates of the outer and inner tracks, respectively, during the vehicle's turning motion; v'_1 and v'_2 are the actual linear velocities of the outer and inner tracks, respectively, during the vehicle's turning motion, in m/s.

At this point, by combining Equations (4) and (5), the actual turning radius R' and the actual turning angular velocity ω' of the tracked roadheaders during the turning motion can be expressed as follows:

$$\begin{cases} \omega' = \frac{v'_1 - v'_2}{B} = \frac{(1-i_1)v_1 - (1+i_2)v_2}{B} \\ R' = \frac{B}{2} \cdot \frac{v'_1 + v'_2}{v'_1 - v'_2} = \frac{B}{2} \cdot \frac{(1-i_1)v_1 + (1+i_2)v_2}{(1-i_1)v_1 - (1+i_2)v_2} \end{cases} \quad (6)$$

Based on the analysis of Equation (6), since $i_1, i_2 \in (0,1)$, the actual centroid angular velocity $\omega' < \omega$, and the actual turning radius $R' > R$. Therefore, the actual turning radius of the tracked roadheaders during the turning motion will tend to be larger compared to the theoretical turning radius, and the actual turning angular velocity will tend to be smaller compared to the theoretical turning angular velocity. Additionally, the actual instantaneous center of rotation of the tracked roadheaders will be farther away from the instantaneous center of rotation calculated by the theoretical model.

To achieve heading correction motion control for the tracked roadheaders, it is necessary to continue the kinematic model analysis in the two-dimensional Cartesian coordinate system based on the turning motion analysis of the tracked roadheaders. The kinematic schematic of the tracked roadheaders in the tunnel model is shown in Figure 4. When performing the kinematic model analysis in the two-dimensional Cartesian coordinate system, the following assumptions are usually made to simplify the analysis process:

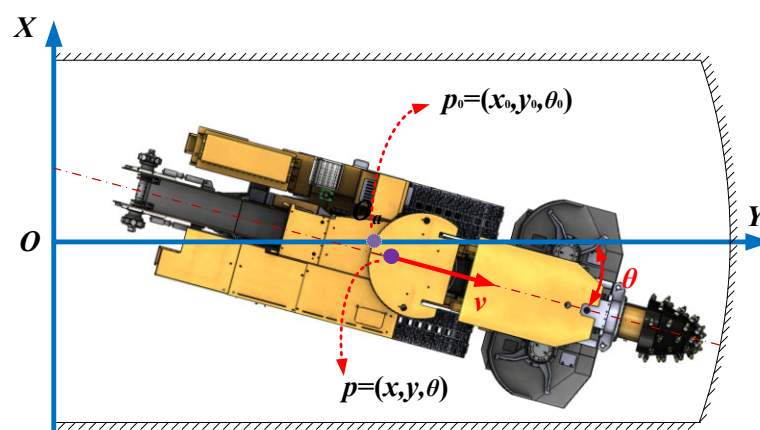


Figure 4. Kinematics principle diagram of crawler support vehicle.

- (1) The resistance coefficient during straight-line motion of the vehicle does not change compared to the resistance coefficient during turning motion;
- (2) During the vehicle's turning motion, only the longitudinal slippage of the tracks on both sides with the floor is considered, while the lateral slippage is ignored;

(3) When the vehicle is in motion, the axis line of the driving wheels of the tracks remains perpendicular to the direction of the vehicle's motion.

As shown in Figure 4, the initial position and real-time position of the tracked roadheaders in the two-dimensional plane tunnel are as follows:

$$\begin{cases} p_0 = [x_0, y_0, \theta_0]^T \\ p = [x, y, \theta]^T \end{cases} \quad (7)$$

Meanwhile, the kinematic model of the tracked roadheaders in the two-dimensional plane tunnel can be expressed as follows:

$$\dot{p} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (8)$$

Substituting Equations (3) and (4) into Equation (8) yields the following:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{r}{2} \cos \theta & \frac{r}{2} \cos \theta \\ \frac{r}{2} \sin \theta & \frac{r}{2} \sin \theta \\ \frac{r}{B} & -\frac{r}{B} \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \end{bmatrix} \quad (9)$$

By performing an integral transformation on Equation (9) and introducing the slip rates i_1 and i_2 of the two tracks, the planar motion equation of the tracked roadheaders considering longitudinal slip between the tracks and the road surface can be obtained

$$\begin{cases} x = x_0 + \frac{r}{2} \int_0^t \cos \theta [(1 - i_1)\omega_1 + (1 + i_2)\omega_2] dt \\ y = y_0 + \frac{r}{2} \int_0^t \sin \theta [(1 - i_1)\omega_1 + (1 + i_2)\omega_2] dt \\ \theta = \theta_0 + \frac{r}{B} \int_0^t [(1 - i_1)\omega_1 + (1 + i_2)\omega_2] dt \end{cases} \quad (10)$$

From Equation (10), it can be seen that the steering angular velocities ω_1 and ω_2 of the two tracks are the theoretical values when the tracked roadheaders are in a pure rolling motion state. At the same time, in tunnel construction sites, the speed and angular velocity of the driving wheels of the tracks on both sides of the vehicle are usually measured in real-time using encoders installed on the motor shafts [17].

3.2. Heading Motion Controller Design

Let the reference position coordinates of the tracked roadheaders at a certain moment in the tunnel be $p_r = [x_r \ y_r \ \theta_r]^T$, and the corresponding reference velocity be $q_r = [v_r \ \omega_r]^T$. By converting the tunnel coordinates to the robot's own coordinate system, the tracking error model of the robot during navigation can be expressed as follows:

$$\begin{bmatrix} \dot{e}_x \\ \dot{e}_y \\ \dot{e}_\theta \end{bmatrix} = \begin{bmatrix} v_r \cos e_\theta \\ v_r \sin e_\theta \\ \omega_r \end{bmatrix} + \begin{bmatrix} e_y \\ -e_x \\ -1 \end{bmatrix} \omega + \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix} v \quad (11)$$

The control objective of the tracked roadheaders when navigating in a confined tunnel space is to find the control inputs $[v \ \omega]$, so that the navigation trajectory tracking error is uniformly bounded, and $\lim_{t \rightarrow \infty} \| [e_x \ e_y \ e_\theta] \| = 0$. For this purpose, construct the Lyapunov function V [18]:

$$V = \frac{1}{2} e_x^2 + \frac{1}{2} e_y^2 + 2 \sin^2 \left(\frac{e_\theta}{2} \right) \quad (12)$$

Differentiating Equation (12) yields

$$\begin{aligned}\dot{V} &= \dot{e}_x e_x + \dot{e}_y e_y + \dot{e}_\theta \sin e_\theta \\ &= e_x v_r \cos e_\theta - e_x v + e_y v_r \sin e_\theta + (\omega_r - \omega) \sin e_\theta \\ &= e_x (v_r \cos e_\theta - v) + (e_y v_r + \omega_r - \omega) \sin e_\theta\end{aligned}\quad (13)$$

Design the control law [19]:

$$\begin{cases} v = v_r \cos e_\theta + K_1 e_x \\ \omega = e_y v_r + \omega_r + K_2 \sin e_\theta \end{cases}\quad (14)$$

In the equation, K_1 and K_2 are control gains, and $K_1 > 0$ and $K_2 > 0$. Substituting into Equation (13) yields:

$$\begin{aligned}\dot{V} &= e_x (v_r \cos e_\theta - v) + (e_y v_r + \omega_r - \omega) \sin e_\theta \\ &= -K_1 e_x^2 - K_2 \sin^2 e_\theta \leq 0\end{aligned}\quad (15)$$

Therefore, using Equation (14) for the motion control of the tracked roadheaders's navigation correction is asymptotically stable, i.e., it satisfies $\lim_{t \rightarrow \infty} \begin{bmatrix} e_x & e_y & e_\theta \end{bmatrix} = 0$

3.3. System Identification Algorithm

To achieve speed control of the tracked roadheaders's navigation system in tunnel space, it is necessary to establish a mathematical model of its drive system [20]. However, traditional mathematical modeling methods involve some degree of model simplification, and the system has delays, so precise identification of the controlled model is required through system identification methods.

3.3.1. Particle Swarm Optimization

Particle swarm optimization is often used to solve global optimization problems of multi-parameter nonlinear systems. Suppose there are n particles in a D -dimensional space, with the position vector of the particles in the population being X_i and the velocity vector being V_i . Substituting X_i into the objective function allows the calculation of the fitness value corresponding to each particle's position. To prevent the individual extremum P_i and the population extremum P_g from falling into local optima in each iteration, a local adaptive mutation operator is added for adjustment, as shown below:

$$\begin{cases} V_i^{n+1} = \omega(t) V_i^n + c_1 r_1 (p_i^n - X_i^n) + c_2 r_2 (S_{Bi}^n - X_i^n) \\ X_i^{n+1} = X_i^n + V_i^{n+1} \end{cases}\quad (16)$$

In the equation, $\omega(t)$ is the inertia weight; n is the evolution generation; i is the particle population size; r is a random number between 0 and 1; c_1 , c_2 are the local and global learning factors; S_B is the optimal solution of the population.

The system identification of the particle swarm optimization algorithm is based on the optimization characteristics of the particle swarm algorithm, which fits the transfer function parameters through the input and output signal responses. Based on the delay in the theoretical model and the actual system, the transfer model can be simplified as [21]

$$G(s) = \frac{K}{(\alpha s^2 + 1)(\beta s + 1)} e^{-Ts}\quad (17)$$

In the equation, $G(s)$ is the transfer function of the controlled object; $X = [\alpha \ \beta \ K \ T]$ is the parameter set to be identified, where α is the second-order term, β is the first-order

term, K is the gain coefficient, and T is the delay coefficient. Due to measurement and execution errors, as well as external disturbances in the actual system, the prediction model cannot completely replace the real model. Therefore, it is necessary to add a residual $e = d_i - f_i$ to the estimation model, where d_i is the output of the i -th test sample, and f_i is the initial fitness value of each particle in the population. To ensure that the identification model is as close as possible to the actual model, the identification error index function is used as the objective function of the particle swarm algorithm. A suitable error index function is selected to update the particle position and velocity, and then the prediction model parameters with the minimum residual absolute value are obtained. The calculation formula for the identification error index function is as follows:

$$J = \sum_{i=1}^N 0.5(d_i - f_i) \quad (18)$$

Let the position vector of the parameters to be identified be X_i , the velocity vector be V_i , and the initial position of each particle be the individual extremum [22]. Based on the above parameters, the initial fitness value $f(X_i)$ of each particle in the population can be calculated. At the same time, the particle velocity and position are updated, and boundary checks are performed on the particle velocity and position. The predicted output of the model is then

$$\begin{cases} f_i[x_n(k+1)] = f_i[x_n(k)e^{-Ts} + x_{n-1}(k)(1 - e^{-Ts})] \\ d_i(k+1) = x_n(k+1 - nQ^{-1}T) \end{cases} \quad (19)$$

In the equation, Q is the simulation step size. The identification error can be calculated using Equation (18) from the f_i and d_i obtained by Equation (19). The smaller the absolute value of the identification error, the closer the prediction model is to the real model. When the identification error index function J gradually approaches 0, the system identification is completed, and the model is closest to the actual model.

3.3.2. System Identification Simulation Analysis

To verify the feasibility of this identification method, the theoretical model of the support transport robot system parameters is used as the identification object for simulation verification. To address the computational difficulty of traditional third-order system identification with delay, the number of particles is set to 80, and the maximum number of iterations is 200. To ensure a balance between the local search ability and global search ability of the particle population, the maximum particle velocity V_{max} is set to 1.0 m/s. The search ranges for the four parameters in X_i are $[0,1]$, $[0,1]$, $[0,10]$, and $[0,1]$. The learning factors are set to $c_1 = 1.1$ and $c_2 = 1.3$. A linearly decreasing inertia weight is used, with the inertia weight decreasing linearly from 0.9 to 0.1. The reciprocal of the objective function is used as the fitness function of the particle swarm, and the identification error is used as the objective function. The system generates historical data for identification through excitation by a pseudo-random binary sequence signal and adds random disturbances during measurement. It is also compared with several other intelligent identification methods to verify the superiority of the identification method proposed in this paper. The simulation results are shown in Table 1.

Table 1. Identification errors of different identification methods.

Method	Identification Error			
	α	β	K	T
Least Squares Identification Method	7.8×10^{-5}	1.3×10^{-4}	0.45	0.005
Genetic Identification Method	3.2×10^{-5}	6.3×10^{-5}	0.37	0.002
Particle Swarm Identification Method	1.1×10^{-6}	3.15×10^{-6}	0.04	0.000

From Table 1, it can be seen that the average identification error of the four sets of identification parameters of the transfer function under test using the particle swarm identification algorithm is reduced by 96.82% and 95.19% compared to the least squares method and the genetic algorithm, respectively. Additionally, the identification error for system delay is close to 0. This indicates a significant improvement in identification accuracy compared to the other two methods. Therefore, the particle swarm algorithm can achieve high-precision identification for nonlinear dynamic models such as the navigation motion control system of tracked roadheaders.

4. Path Planning for Tracked Roadheaders Based on the Mutated Particle Swarm Algorithm

After completing the establishment of the fully mechanized roadway model, the analysis of the driving performance of the crawler support vehicle, and the modeling of the crawler walking hydraulic system, it is necessary to achieve the path planning and deviation correction motion control of the crawler support vehicle, and to complete targeted driving or steering navigation commands within a certain distance range. First, during the rapid excavation construction cycle with parallel excavation and support, the construction path is autonomously planned based on the external dimensions and driving characteristics of the crawler support vehicle, and the key points of the path that require command execution in the construction line are marked. Then, based on the kinematic characteristics of the crawler support vehicle, the marked key points are sequentially tracked to complete the tracking and deviation correction in the restricted roadway space.

Based on the position information of the established two-dimensional plane roadway model, the crawler support vehicle obtains the initial point heading angle θ_0 and the target point heading angle θ_g through the pose detection process. Then, in a construction cycle, the change in the vehicle's heading angle $\Delta\theta$ and the total path length $[L]$ can be expressed as follows:

$$\Delta\theta = \arctan \left| \frac{\frac{x_{j+1}-x_j}{y_{j+1}-y_j} - \frac{x_j-x_{j-1}}{y_j-y_{j-1}}}{1 + \frac{x_j-x_{j-1}}{y_j-y_{j-1}} \times \frac{x_{j+1}-x_j}{y_{j+1}-y_j}} \right| \quad (20)$$

$$[L] = \sum_{j=0}^n \sqrt{(x_j - x_{j+1})^2 + (y_j - y_{j+1})^2} \quad (21)$$

During the driving process of the crawler support vehicle, non-traversable areas have a significant impact on the construction path. Therefore, the concept of a heading safety index is proposed, which represents the distance between the path point coordinates and the center of the non-traversable area or the roadway boundary. A higher heading safety index indicates better safety of the planned path, while a lower index indicates poorer safety. The expression for this heading safety index is

$$[SI]_i = \frac{\sum_{j=0}^n \sqrt{(x_j - x_{Bi})^2 + (y_j - y_{Bi})^2}}{n[L]} \quad (22)$$

In the formula, $[SI]_i$ is the heading safety index of the crawler support vehicle for the i -th non-traversable area, and x_{Bi} and y_{Bi} are the horizontal and vertical coordinates of the center point of the i -th non-traversable area, respectively.

Adaptive Particle Swarm Path Point Planning Algorithm

As a well-developed swarm intelligence algorithm, the particle swarm algorithm demonstrates strong advantages in solving optimization problems. In the particle swarm algorithm model, a search collective $X_i = [x_{i1}, x_{i2}, \dots, x_{iD}]^T$ ($i = 1, 2, \dots, N$) is formed by N particles in a D -dimensional target search space. The state of each particle is described by a position vector $p_i = [p_{i1}, p_{i2}, \dots, p_{iD}]^T$ ($i = 1, 2, \dots, N$) and a velocity vector $V_i = [v_{i1}, v_{i2}, \dots, v_{iD}]^T$ ($i = 1, 2, \dots, N$). Among these, the velocity of the particle directly affects the search distance of each step in the search space and can be adjusted based on the performance of other particles and its own fitness. The optimal individual extremum found by each particle is recorded as $p_{best} = [p_{bi1}, p_{bi2}, \dots, p_{biD}]^T$ ($i = 1, 2, \dots, N$), and the global optimal extremum found by the entire particle swarm is recorded as $g_{best} = [g_1, g_2, \dots, g_D]^T$. Then, the particle properties are updated using Equation (23):

$$\begin{cases} vx_{ij}(t+1) = \omega \cdot v_{ij}(t) + c_1 r_1 [p_{bij}(t) - x_{ij}(t)] + c_2 r_2 [g_{bj}(t) - x_{ij}(t)] \\ x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1) \\ \omega = \omega_{\max} - \frac{(\omega_{\max} - \omega_{\min}) \cdot t}{T_{\max}} \end{cases} \quad (23)$$

In the equation, i represents the i -th particle in the particle swarm algorithm; j represents the j -th dimension of the individual particle; t represents the current iteration count of the particle swarm algorithm; $v_{ij}(t)$ represents the j -th dimensional flight velocity component of individual particle i at the t -th generation of updates; $x_{ij}(t)$ represents the j -th dimensional position component of individual particle i at the t -th generation of updates; $p_{bij}(t)$ represents the j -th dimensional individual optimal position p_{best} component of individual particle i at the t -th generation of updates; $g_{bj}(t)$ represents the j -th dimensional component of the global optimal position g_{best} of the entire particle swarm at the t -th generation of evolution; c_1, c_2 are acceleration factors; r_1, r_2 are random numbers in the range $[0, 1]$; $T_{\max}$ is the maximum generation count for particle updates; $\omega_{\max}, \omega_{\min}$ are the maximum and minimum inertia weights, respectively [23,24].

Based on the established grid model of the two-dimensional plane roadway, the position components of each particle in the particle swarm algorithm in each dimension are, respectively, associated with the corresponding horizontal and vertical coordinates in the two-dimensional plane model. The position coordinates of individual particles in each dimension in the particle swarm algorithm correspond to the horizontal and vertical coordinates within the two-dimensional plane roadway model, thereby obtaining the optimal solution p_{bi} , achieving path point planning and optimal path point search for the tracked support vehicle within the two-dimensional plane roadway. When using the particle swarm algorithm to solve the optimal path for the tracked support vehicle, it is also necessary to ensure that the planned path curve completely avoids non-traversable areas.

As individual particles in the particle swarm algorithm continuously update and iterate, individual particles are highly likely to fall into local optimal convergence traps, and the diversity within the particle swarm algorithm sharply decreases. Eventually, all individual particles in the particle swarm algorithm may concentrate within a smaller search range for iterative updates. When this smaller search range contains the optimal position point, the accuracy of the algorithm can meet the precision requirements of path planning. However, when this search range does not contain the optimal position point, the particle swarm algorithm will be unable to meet the efficiency and accuracy requirements

of the tracked support vehicle for path planning. Therefore, this paper will improve the standard particle swarm algorithm by introducing crossover operations for particles using adaptive mutation probabilities, fully enriching the diversity of the particle population in the later stages of algorithm iteration, while defining the adaptive probability of the algorithm as follows:

$$P = \mu + \text{Re} \cdot \sigma \quad (24)$$

In the formula, μ and σ are the mutation rate adjustment parameters, and Re is the number of consecutive update iterations of the optimal value.

In the adaptive particle swarm algorithm with mutation operations, if the population in the algorithm continues to update and iterate, no mutation intervention is needed; if the population convergence speed stagnates and no updates can be made for several consecutive generations, the Re value in Equation (24) will continuously accumulate, indicating that the adaptive probability of the algorithm for mutation operations will continuously increase. The measure for mutation intervention is as follows: sequentially extracting all individual particles from the entire population that has stagnated in updates, calculating the Euclidean distance between each extracted individual particle and the global optimal particle, and comparing it with the basic threshold. Individual particles smaller than this basic threshold will undergo crossover operations, then update and iterate to the next step, ensuring a global optimal search.

At the same time, the Euclidean distance O_p and the basic threshold Λ are defined as follows:

$$\begin{cases} O_p = \sqrt{\sum_{i=1}^D [x_1(i) - x_2(i)]^2} \\ \Lambda = (1 - t/T)^\varepsilon \times (ub - lb) \end{cases} \quad (25)$$

In the formula, x_1 and x_2 are two particles in the particle population; t and T are the number of updated iterations and the maximum number of iterations set by the algorithm, respectively; ub and lb are the range boundaries; ε is the adjustment parameter.

In the early stages of algorithm update iterations, the diversity of population particles is relatively rich, and the basic threshold should be set high to avoid mutation intervention in the population. As the algorithm continues to update and iterate into the middle and late stages, the population particles will tend to aggregate in a small range, potentially converging to a local optimum. At this point, the basic threshold should be lowered to allow more population particles to enter the iterative loop of crossover operations, expanding the search range of the particles [21,25].

The crossover operation rules are as follows:

$$\begin{cases} cx_1 = x_1e + x_2(1 - e) \\ cx_2 = x_1(1 - e) + x_2e \end{cases} \quad (26)$$

In the formula, cx_1 and cx_2 are the new particles generated by the crossover operation.

After the crossover operation is executed, if the fitness value of the new particle is better than before the crossover operation, the particle needs to be updated and replaced; if the fitness value is worse than before the crossover operation or does not change at all, the following mutation operation needs to be introduced to verify the optimality of the fitness values of the surrounding individual particles. If there is a particle with a better fitness value after mutation, the particle will still be updated and replaced. The specific mutation operation rules are as follows:

$$\begin{cases} mx_1 = x + (1 - \frac{t}{T})^\delta (ub - x) \\ mx_2 = x - (1 - \frac{t}{T})^\delta (x - lb) \end{cases} \quad (27)$$

In the formula, m_{x1} and m_{x2} are the mutated particles; δ is the mutation weight coefficient.

5. Deviation Correction Motion Control of Tracked Roadheaders

5.1. Deviation Correction Control System Scheme

The above has completed the working condition scene modeling, driving motion analysis and system identification, and path planning algorithm of the tracked roadheaders in the restricted roadway space. However, it is still necessary to construct the navigation and deviation correction control system of the tracked roadheaders, and based on the path navigation information as input, complete the correction of the real-time coordinate parameters of the tracked roadheaders, thereby realizing the navigation and deviation correction motion control of the tracked roadheaders in the restricted roadway space. The navigation and deviation correction control system of the tracked roadheaders is shown in Figure 5.

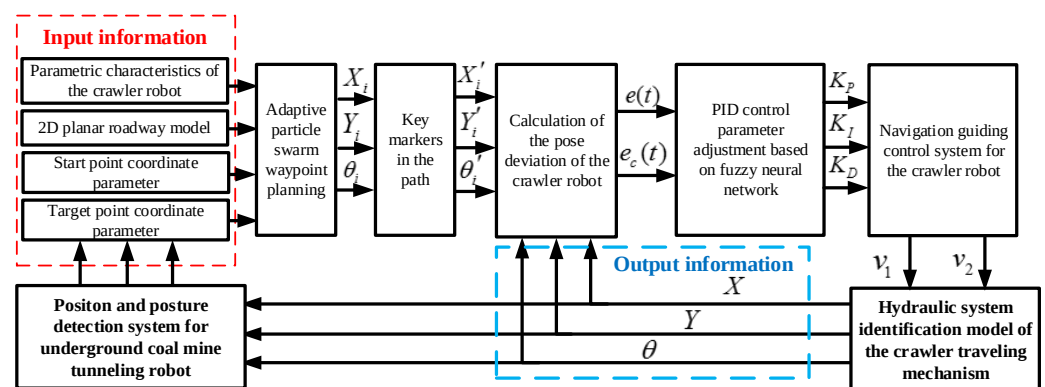


Figure 5. Navigation and deviation correction control system of tracked roadheaders.

(1) Input Information Module

When conducting rapid excavation and support construction technology in parallel in the roadway, it is first necessary to complete the geological exploration of the surrounding rock to confirm the construction conditions of the roadway. The initial calibration and alignment of the tracked roadheaders are performed using an intelligent onboard pose detection system. Then, the coordinates and heading angle parameters of the initial position point and the target position point in the two-dimensional plane roadway model are obtained, forming a complete set of input information parameters that are input into the control system, known as the input information module.

(2) Control Algorithm Module

The control algorithm module includes adaptive particle swarm optimization, tracking and deviation correction algorithms, and FNN PID control algorithms. These algorithms are used for autonomous path planning based on the local grid model of the tracked roadheaders, marking key position points, and controlling navigation and deviation correction movements. Based on the input information, the hydraulic system of the crawler walking mechanism is controlled.

(3) Output Information Module

The output information module is mainly used to receive control algorithm instructions for adjusting the speed of the crawler on both sides of the robot. Then, the real-time measurement of the robot's pose information is performed through the intelligent onboard pose detection system and fed back to the control algorithm module, forming an effective closed-loop control system.

5.2. Navigation and Deviation Correction Motion Control Algorithm

The classical PID controller generates the final control output $\Delta u(k)$ to correct trajectory deviations. Its proportional (K_P), integral (K_I), and derivative (K_D) gains are dynamically tuned by the fuzzy-neural network to handle hydraulic nonlinearities and slippage disturbances. This enables precise velocity adjustments for both tracks, ensuring stable path tracking in confined roadways.

The hydraulic system of the tracked roadheaders' walking mechanism is highly nonlinear and the road conditions are complex and variable, making it difficult to meet the high-precision dynamic response requirements using traditional PID control algorithms. Additionally, fuzzy control can be applied to nonlinear systems, but this has the disadvantage of poor adaptability, while neural networks possess nonlinear mapping and self-learning capabilities. The combination of the two forms a fuzzy neural network with good adaptability [26,27]. The control block diagram of the tracked roadheaders' walking mechanism based on FNN PID control is shown in Figure 6.

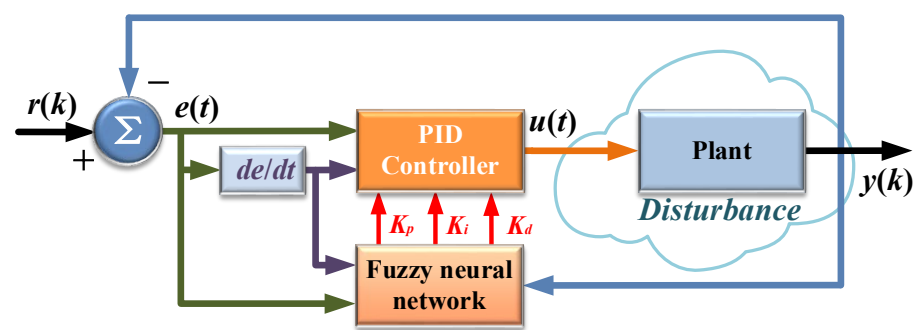


Figure 6. Block diagram of tracked roadheaders' walking mechanism based on FNN PID control.

(1) PID Control Parameter Tuning

Based on the transfer function of the tracked roadheaders' walking mechanism hydraulic system determined above, the initial values of the PID control parameters are set to 1.52, 0.12, and 0.15 using the Ziegler–Nichols tuning method.

(2) Fuzzy Logic Design

In the fuzzy controller, the fuzzy set domains for the position deviation $e(t)$ and the rate of change $e_c(t)$ of the tracked roadheaders are $\{-3, -2, -1, 0, 1, 2, 3\}$, divided into seven linguistic variables: {NB, NM, NS, ZO, PS, PM, PB}, corresponding to seven fuzzy subsets: {negative large, negative medium, negative small, zero, positive small, positive medium, positive large}. The fuzzy set domains are $\{-3, -2, -1, 0, 1, 2, 3\}$, $\{-1.5, -1, -0.5, 0, 0.5, 1, 1.5\}$, and $\{-0.3, -0.2, -0.1, 0, 0.1, 0.2, 0.3\}$, respectively. The fuzzy inference process is set as follows: the fuzzy controller adopts the Mamdani algorithm, each with two inputs and three outputs using trimf triangular and z-shaped membership functions. The fuzzy control rules are established, and the membership functions of the input and output parameters are shown in Figure 7.

(3) Fuzzy Neural Network Algorithm

Figure 8 shows the structure of the fuzzy neural network. The two inputs of information are the position deviation of the robot and the rate of change in the deviation, and the three outputs of information correspond to the control parameters of the PID. The algorithm contains a total of three hidden layers and one output layer. The first hidden layer is the fuzzification layer, which contains seven neurons corresponding to the seven fuzzy subsets of the controller input. The second hidden layer is the rule calculation layer, which contains 49 neurons corresponding to the 49 control rules of the controller. The third hidden layer is the defuzzification inference layer, which contains seven neurons corresponding to the seven fuzzy subsets of the controller output.

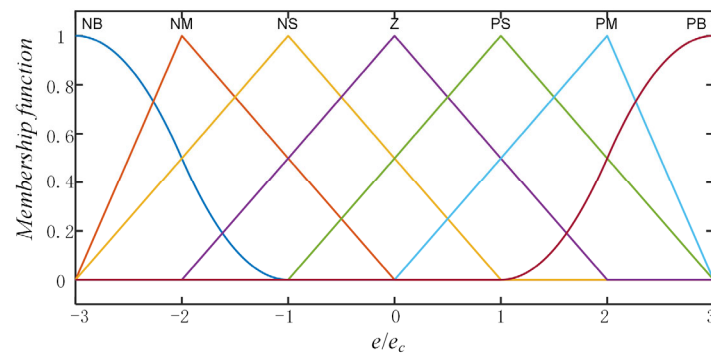


Figure 7. Membership functions of input and output variables.

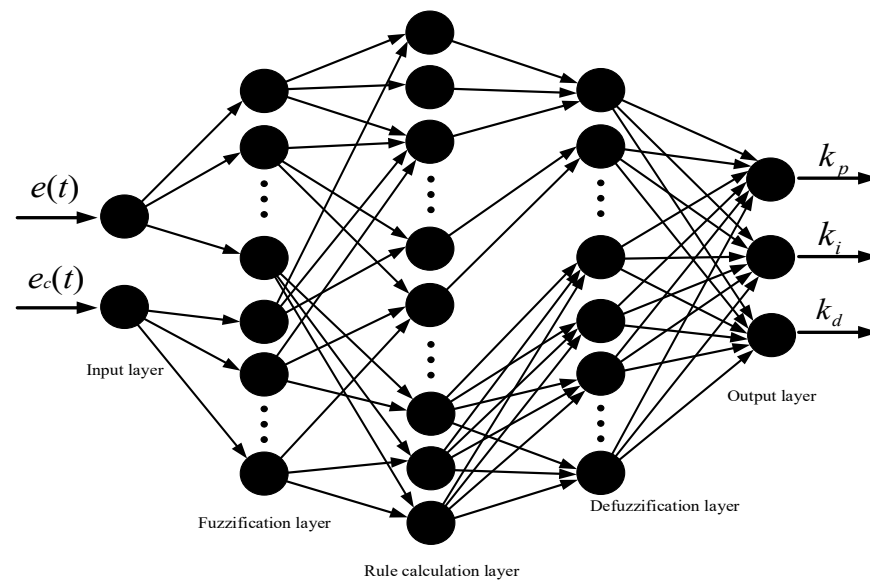


Figure 8. Architecture diagram of fuzzy neural network system.

According to the basic structure of input and output set by the fuzzy neural network in Figure 8, the relevant analysis is carried out:

(1) The input layer of the fuzzy neural network has two pieces of information: $e(k)$ and $e_c(k)$, where $k \in [0, t]$; the corresponding outputs after input are: $O_1(1,j) = e(k)$ and $O_1(2,j) = e_c(k)$, where $j = 1, 2, \dots, 7$.

(2) When the parameters are passed from the input layer to the fuzzification layer, the relationship between input and output can be expressed as follows:

$$\begin{cases} I_2(i,j) = \frac{[O_1(i,j) - b_{ij}]^2}{c_{ij}^2} \\ O_2(i,j) = \exp[I_2(i,j)] \end{cases} \quad (28)$$

In the formula, b_{ij} is the center value of the fuzzy membership function; c_{ij} is the width of the fuzzy membership function.

(3) The input and output of the fuzzy rule calculation layer and the fuzzy inference layer are:

$$\begin{cases} I_3(m) = O_2(1,k_1) \times O_1(2,k_2) (k_1 = 1, 2, \dots, 7; k_2 = 1, 2, \dots, 7) \\ O_3(m) = I_3(m) (m = 1, 2, \dots, 49) \end{cases} \quad (29)$$

(4) The input and output of the output layer are as follows:

$$\begin{cases} I_4(n) = \sum_{m=1}^{49} O_3(m) \cdot \lambda_{mn} (m = 1, 2, \dots, 49) \\ O_4(n) = I_4(n) (n = 1, 2, 3) \end{cases} \quad (30)$$

In the formula, λ_{mn} is the weight coefficient of the participating terms, each of which corresponds to the proportional, derivative, and integral control parameters of the output PID. Meanwhile, the PID control adopts an incremental PID control algorithm, as shown in Equation (31):

$$\begin{cases} \Delta u(k) = [e(k) - e(k-1)]k_p + e(k)k_I + [e(k) - 2e(k-1) + e(k-2)]k_D \\ u(k) = u(k-1) + \Delta u(k) \\ k_x = k_{x0} + \Delta k_x \end{cases} \quad (31)$$

In the formula, k_x is the output value of k_p , k_I , and k_D ; k_{x0} are the initial values of k_p , k_I , and k_D , respectively.

Define the performance index function during the system error feedback process as $E(k) = 0.5[r(k) - y(k)]^2$, where $r(k)$ is the target value of the system at time k , and $y(k)$ is the actual output value of the system at time k ; when $y(k)$ infinitely approaches $r(k)$, the performance index function will also infinitely approach zero. Using a neural network, the real-time calibration of the center value b_{ij} , width c_{ij} , and weight coefficient λ_{mn} of the fuzzy membership function can be completed, with the corresponding adjustment rules as follows [28]:

$$\begin{cases} \lambda_{mn}(k+1) = \lambda_{mn}(k) + \xi(k) \frac{\partial E}{\partial \lambda} \\ b_{ij}(k+1) = b_{ij}(k) + \xi(k) \frac{\partial E}{\partial b} \\ c_{ij}(k+1) = c_{ij}(k) + \xi(k) \frac{\partial E}{\partial c} \end{cases} \quad (32)$$

In the formula, $\xi(k)$ is the learning rate of the neural network.

At the same time, to further ensure that the learning rate of the neural network and the established control system can mutually adapt and match, ensuring the training intensity and control performance of the algorithm during the self-learning phase, the learning rate adaptive rules are set in segments:

$$\xi(k) = \begin{cases} \xi(k-1) \times 0.99 & E(k) > 1.04 \times E(k-1) \\ \xi(k-1) \times 1.01 & E(k) \leq E(k-1) \\ \xi(k-1) & E(k-1) \leq E(k) \leq 1.04 \times E(k-1) \end{cases} \quad (33)$$

The partial derivatives of $E(k)$ in equation $\frac{\partial E}{\partial b}$, $\frac{\partial E}{\partial \lambda}$ and $\frac{\partial E}{\partial c}$ can be solved using the error backpropagation method, and the related calculation methods are as follows:

$$\begin{cases} \frac{\partial E}{\partial \lambda} = \frac{\partial E(k)}{\partial y(k)} \times \frac{\partial y(k)}{\partial \Delta u(k)} \times \frac{\partial \Delta u(k)}{\partial O_4} \times \frac{\partial O_{ij}}{\partial \lambda_{mn}} = M \cdot \lambda \\ \frac{\partial E}{\partial b} = \frac{\partial E(k)}{\partial y(k)} \times \frac{\partial y(k)}{\partial \Delta u(k)} \times \frac{\partial \Delta u(k)}{\partial O_4} \times \frac{\partial O_4}{\partial O_3} \times \frac{\partial O_3}{\partial O_2} \times \frac{\partial O_2}{\partial b_{ij}} = M \cdot \frac{2[I_1(i) - b_{ij}]}{c_{ij}^2} \\ \frac{\partial E}{\partial c} = \frac{\partial E(k)}{\partial y(k)} \times \frac{\partial y(k)}{\partial \Delta u(k)} \times \frac{\partial \Delta u(k)}{\partial O_4} \times \frac{\partial O_4}{\partial O_3} \times \frac{\partial O_3}{\partial O_2} \times \frac{\partial O_2}{\partial c_{ij}} = M \cdot \frac{2[I_1(i) - b_{ij}]}{c_{ij}^2} \end{cases} \quad (34)$$

In equation $\frac{\partial y(k)}{\partial \Delta u(k)}$, it can be approximately replaced by its variable symbolic function for calculation, thus, $\frac{\partial y(k)}{\partial \Delta u(k)} \approx \text{sgn}[\frac{y(k) - y(k-1)}{\Delta u(k) - \Delta u(k-1)}]$ let,

$$M = \text{sgn}[\frac{y(k) - y(k-1)}{\Delta u(k) - \Delta u(k-1)}] - [r(k) - y(k)] \cdot \frac{\partial \Delta u(k)}{\partial O_4} \cdot O_3$$

6. Control Experiment Research

6.1. Constraints and Parameter Settings

6.1.1. Heading Correction Control Constraints

In the actual construction scenarios of coal mine roadways, the hydraulic system of the crawler walking mechanism often encounters issues where the starting acceleration is too small to overcome the dead zone characteristics. Conversely, if the starting or braking acceleration is too large, it can cause the tracked roadheaders to experience sudden surges or abrupt stops [29]. When performing navigation and correction movements of the tracked roadheaders in the confined space of a comprehensive excavation roadway, such abrupt driving states should be fully avoided to prevent potential safety hazards at the construction site. Therefore, the maximum linear and angular accelerations of the crawler driving wheel are set as $a_{\max}$ and $\alpha_{\max}$, respectively, and the minimum values are $a_{\min}$ and $\alpha_{\min}$. The following will analyze and set the range and conditions of the system input quantities.

From Equations (1) and (2), it is known that $2v + B\omega = 2v_1$. Taking the derivative of both sides: $2\dot{v} + B\dot{\omega} = 2\dot{v}_1$. At this point, the crawler acceleration needs to satisfy the constraint condition: $r\alpha_{\min} \leq a + 0.5B\alpha \leq r\alpha_{\max}$. Assuming that the extreme values of the angular acceleration and linear acceleration of the crawler driving wheel have the same proportional weight in the constraint conditions, the constraint conditions for the input linear and angular accelerations of the tracked roadheaders in a horizontal roadway environment are as shown in Equation (35):

$$\begin{cases} 0.5ra_{\min} \leq a \leq 0.5ra_{\max} \\ \frac{r}{B}\alpha_{\min} \leq \alpha \leq \frac{r}{B}\alpha_{\max} \end{cases} \quad (35)$$

When the tracked roadheaders are operating in an inclined roadway, the driving or braking force required for the vehicle to complete the uphill or downhill construction tasks will be directly related to the roadway inclination angle γ . At this time, the crawler acceleration needs to satisfy the constraint conditions, which will also consider the influence of the roadway inclination angle γ . Therefore, the constraint conditions for the input linear and angular accelerations of the tracked roadheaders in an inclined roadway environment are as shown in Equation (36):

$$\begin{cases} 0.5ra_{\min}(1 + \sin|\gamma|) \leq a \leq 0.5ra_{\max}(1 + \sin|\gamma|) \\ \frac{r}{B}\alpha_{\min}(1 + \sin|\gamma|) \leq \alpha \leq \frac{r}{B}\alpha_{\max}(1 + \sin|\gamma|) \end{cases} \quad (36)$$

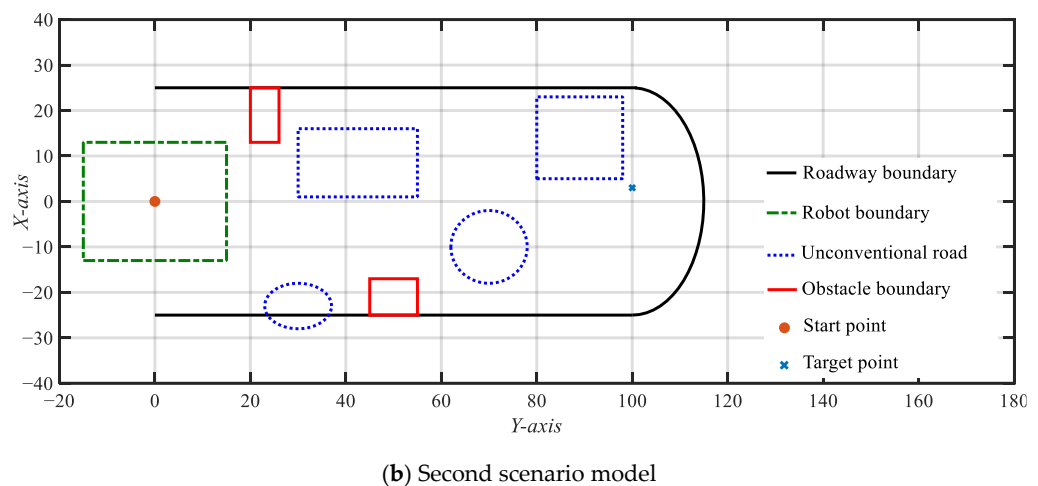
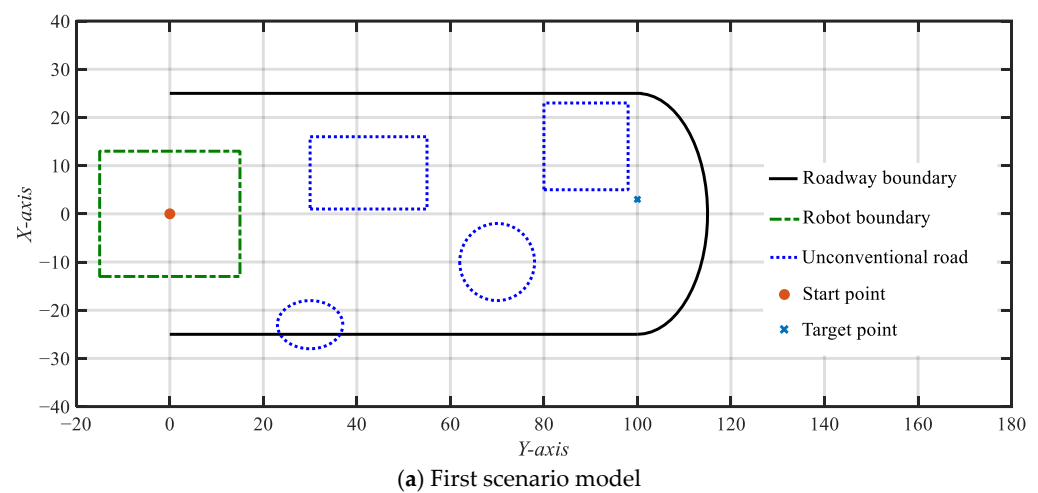
6.1.2. System Simulation Parameter Settings

Based on the current experimental conditions that can be met: Matlab-R2017b is used as the programming and simulation platform, the solver is set to ode45 mode, the CPU main frequency specification is 3.30 GHz, and the computer system is installed with Win7 for compatibility considerations. At the same time, the main parameter settings of the tracked roadheaders simulation model are shown in Table 2.

The scene simulation is conducted according to the fully mechanized roadway in Figure 9, setting up two simulation models of road conditions in restricted roadway spaces, as shown in Figure 9. The first scenario model includes the starting point, target point, vehicle boundary, unconventional road conditions, and roadway boundary of the tracked roadheaders; the second scenario model takes into account the actual working conditions where surrounding rock protrusions often occur on both sides of the roadway, so two non-traversable areas are added to simulate the restricted environment.

Table 2. Main parameters of the tracked roadheaders.

Parameter Name	Variable	Unit	Value
Total Weight	m	t	15
Track Ground Contact Length	L_h	mm	2520
Track Width	l_w	mm	320
Track Center Distance	B	mm	3820
Track Ground Pressure	P_t	MPa	0.12
Vehicle Body Dimensions	-	mm ³	$5790 \times 4160 \times 2900$
Maximum Travel Speed	-	m/s	1
Drive Sprocket Pitch Radius	r	mm	300
Moment of Inertia	J	kg·m ²	15,000
Transmission Ratio	i	-	10

**Figure 9.** Simulation model diagram of construction environment of fully mechanized roadway.

The areas of unconventional road conditions in the first and second scenario models are 1.099 m^2 , 3.75 m^2 , 2.01 m^2 , and 3.24 m^2 , respectively; the two non-traversable areas added in the second scenario model have areas of 0.8 m^2 and 0.72 m^2 , respectively. Additionally, based on the actual soil geological conditions in the roadway, the simulation is conducted with a mixture ratio of sandy loam, powdery coal-bearing soil, and gravelly coal-bearing soil of 0.45, 0.4, and 0.15, respectively. The corresponding heading reference influence degrees for different areas are 0.331, 0.561, 0.511, and 0.667.

Secondly, in order to conduct simulation experiments on path point planning and tracking correction algorithms for tracked roadheaders, it is necessary to complete the basic parameter settings of the adaptive particle swarm optimization algorithm.

In the adaptive particle swarm optimization algorithm, the particle swarm size is set to $N = 100$, the acceleration factors are set to $c_1 = c_2 = 1.49$, the linear decreasing interval of the inertia weight ω is set to $[0.85, 0.41]$, the maximum velocity is set to $v_{max} = 1$, the upper limit of the number of iterations is set to 2000; the mutation weight coefficient is set to $\delta = 2$; at the same time, the probability coefficients of the adaptive particle swarm are tested, and the 30-dimensional Rosenbrock function is selected as the test function, with $\{\mu = [0.1, 0.01, 0.001, 0.0001], \sigma = [0.5, 0.05, 0.005, 0.0005], \delta = [0, 1, 2, 3]\}$ substituted into the test function, respectively, and each set of data is tested 10 times to take the average as the test result. The average fitness value, average number of steps to reach the optimum, and average time to reach the optimum are statistically analyzed, and the test results are shown in Table 3.

Table 3. Algorithm simulation test results.

Parameter	Value	Minimum	Average Fitness Value	Average Time to Reach Optimal/s	Average Steps to Reach Optimal
$\mu(\sigma)$	0.1 (0.5)	0	4.942×10^{-8}	4.624	1996
	0.01 (0.05)	0	2.944×10^{-9}	1.836	1986
	0.001 (0.005)	0	5.512×10^{-15}	1.221	1995
	0.0001 (0.0005)	8.042×10^{-15}	1.462×10^{-9}	1.047	2000
	0	2.421×10^{-19}	3.098×10^{-5}	1.021	2000
δ	1	2.946×10^{-18}	1.388×10^{-9}	1.018	2000
	2	0	1.482×10^{-11}	1.025	1890
	3	2.388×10^{-19}	4.303×10^{-9}	1.039	2000

As can be seen from the test results in Table 3, during the testing process, a probability coefficient that is too large will cause continuous changes in the population individuals, slow down the convergence speed, and increase the algorithm's time consumption. A probability coefficient that is too small will cause the algorithm to stagnate too quickly, falling into local optima. Based on the comprehensive test results, the adaptive probability coefficient is set to $\mu = 0.001, \sigma = 0.005$; for the mutation weight coefficient, if its value is too large, the adjustment of the mutation amount is small, and it obviously cannot play an effective role in the later stages of the algorithm.

6.2. Path Navigation Experiment

6.2.1. Algorithm Performance Analysis

The standard particle swarm algorithm was compared with the adaptive particle swarm algorithm after mutation operation. In addition to the Rosenbrock test function, the Shubert test function was added to compare the convergence performance, further demonstrating the reliability of the adaptive particle swarm algorithm. The test functions are shown in Figure 10; in the simulation experiment, the maximum number of iterations was 100, and each group of tests was run 10 times. The fitness curves of the algorithm were plotted, and the average and optimal values were recorded. The simulation results are shown in Figure 11 and Table 4.

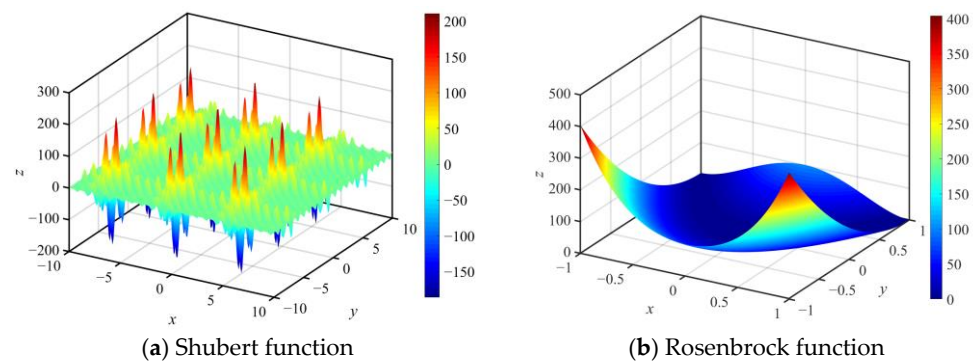


Figure 10. Three-dimensional view of the Rosenbrock and Shubert test functions.

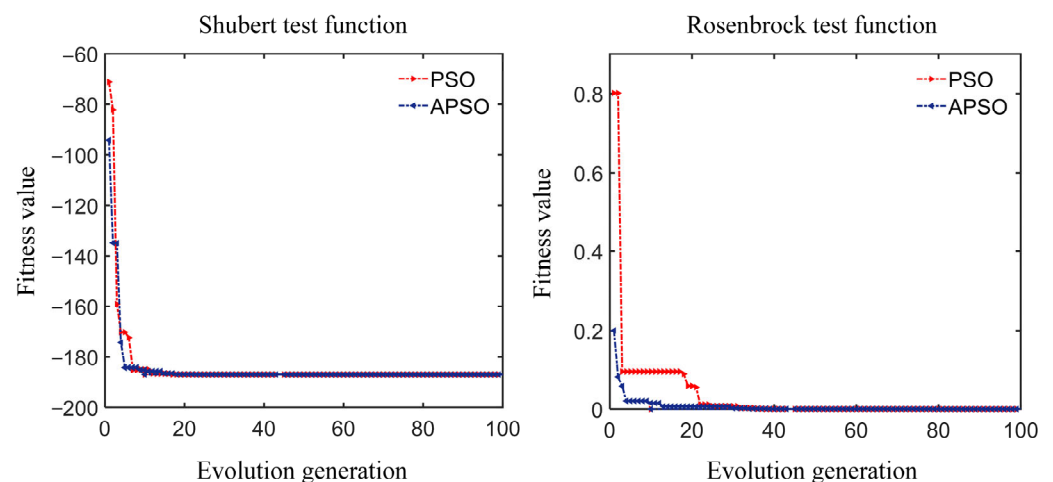


Figure 11. Fitness change curves of adaptive particle swarm optimization and standard particle swarm optimization.

Table 4. Comparison of simulation test results of APSO and PSO.

Test Function	Algorithm	Average	Optimal	Standard Deviation
Shubert	PSO	−185.2376	−186.7309	2.8810
	APSO	−185.1315	−186.7305	2.3547
Rosenbrock	PSO	0.0089	8.0458×10^{-5}	0.9924
	APSO	0.0037	8.8146×10^{-7}	0.0421

As shown in Figure 11 and Table 4, under the test of the Shubert function, both the adaptive particle swarm optimization and the standard particle swarm optimization can converge near the optimal value of the function and can also search for the minimum fitness value with fewer iterations and higher precision. Under the test of the Rosenbrock function, the adaptive particle swarm optimization can still quickly converge near the optimal value of the function, and more stably and reliably perform precise searches for the optimal solution. In this case, the adaptive particle swarm optimization after mutation operation is obviously superior to the standard particle swarm optimization.

6.2.2. Simulation Analysis of Waypoint Planning

Based on the established two-dimensional plane roadway model and path planning and tracking algorithm, two particle swarm optimization algorithms were used to conduct simulation tests on the tracked roadheaders under different roadway conditions. The planning results are shown in Figure 12.

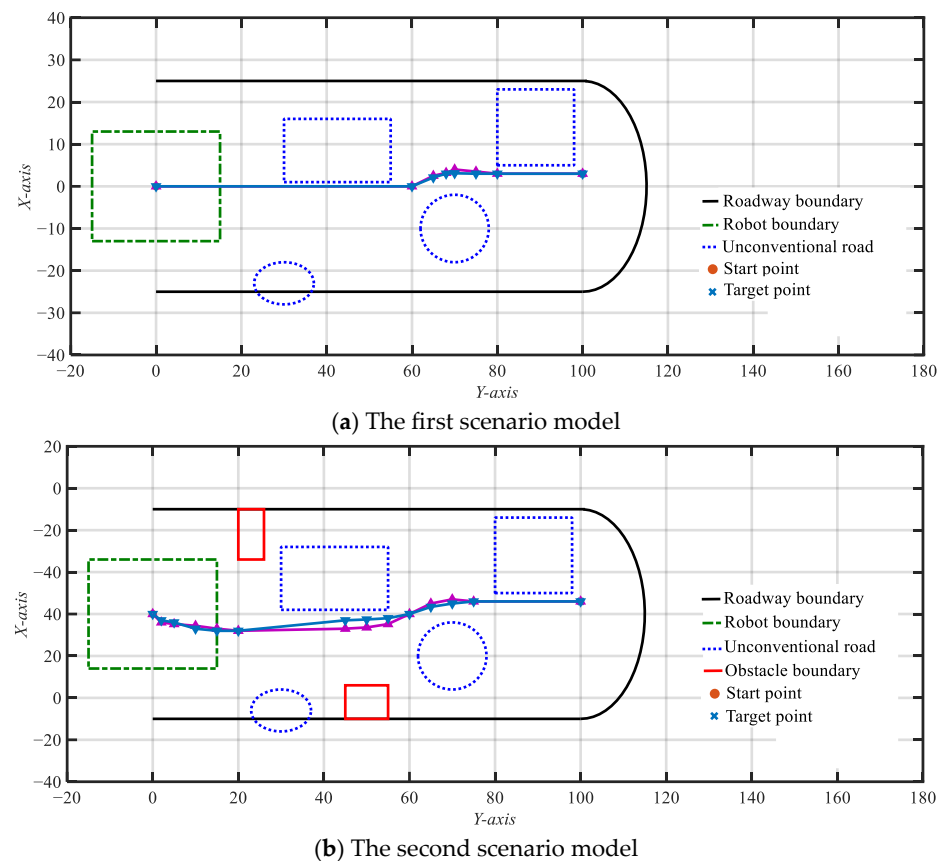


Figure 12. Simulation of construction path point planning for the tracked roadheaders.

(1) The first roadway scenario model

In Figure 12a, the black contour lines at the top and bottom represent the inner sides of the left and right walls of the roadway, respectively; the black rectangle at the far left represents the vertical projection of the tracked roadheaders on the roadway floor at the initial position; the dark blue outlined elliptical and rectangular areas represent the passable regions with different area attributes within the roadway that have been surveyed; the purple upward triangle-marked curve and the light blue downward triangle-marked curve represent the optimal construction paths solved by the standard particle swarm optimization and the adaptive particle swarm optimization, respectively.

To compare the path point planning effects of the standard particle swarm optimization and the adaptive particle swarm optimization, the total centroid displacement from the starting point to the target point, the total heading angle, and the heading safety index were calculated for comprehensive comparison. The specific simulation results are shown in Table 5.

Table 5. Driving performance parameters of path planning for the tracked roadheaders (first scenario).

Driving Performance Parameters/Unit	Adaptive Particle Swarm Optimization	Standard Particle Swarm Optimization
Total Steering Angle/ $^{\circ}$	51.238	81.905
Total Displacement of Centroid/mm	100.559	100.875
Heading Safety Index	0.585	0.581

(2) Second roadway scenario model

The starting point, target point, and calibrated colors of the passable unconventional road condition areas in the second roadway scenario model are the same as those in the

first roadway scenario model. Additionally, two red non-traversable area contours are added within the roadway, as shown in Figure 12b. The two particle swarm optimization algorithms, respectively, solve the optimal construction path for the vehicle based on the known roadway condition information. The specific simulation results are shown in Table 6.

Table 6. Driving performance parameters of path planning for the tracked roadheaders (second scenario).

Driving Performance Parameters/Unit	Adaptive Particle Swarm Optimization	Standard Particle Swarm Optimization
Total Steering Angle/ $^{\circ}$	149.262	161.344
Total Displacement of Centroid/mm	101.393	102.283
Heading Safety Index	0.859	0.857

From the simulation results of vehicle driving performance parameters, it can be seen that the adaptive particle swarm optimization algorithm after mutation operation has a certain degree of improvement in the performance parameter of the total turning angle of the vehicle, resulting in a small reduction in the total centroid displacement of the vehicle. At the same time, the adaptive particle swarm optimization algorithm shows a certain degree of excellence in the heading safety index.

6.2.3. Simulation Analysis of Key Point Tracking and Rectification

When performing rectification control on the key points of the tracked roadheaders' path, it is first necessary to calibrate the key points within the range of path points formed by the adaptive particle swarm optimization algorithm, thereby achieving rectification control for the key points.

Due to the limited space in the comprehensive excavation roadway and the large size of the tracked roadheaders, the optimal construction path points planned by the adaptive particle swarm optimization algorithm are relatively few. Three key points are selected from the path points for tracking and rectification. At the same time, rectification is carried out according to the marking of key points, thereby calculating the total turning angle of the vehicle's heading, the total centroid displacement, and the heading safety index. The simulation test parameters of the path points and the marked key points are compared as shown in Table 7.

As can be seen from the data in Table 7, after marking key points in the path points planned by the adaptive particle swarm algorithm and completing tracking correction, the heading safety index of the vehicle during driving can be effectively improved, and the performance parameters of the total steering angle and total centroid displacement of the vehicle can be enhanced. At the same time, it avoids frequent correction adjustments during the construction process, ensuring the efficiency of the vehicle in navigation correction motion control.

Table 7. Performance parameters of key points for crawler support vehicle in tracking rectification.

Driving Performance Parameters/Unit	Waypoint Tracking		Key Point Tracking	
	Scene Model 1	Scene Model 2	Scene Model 1	Scene Model 2
Total Steering Angle/ $^{\circ}$	51.238	149.262	34.020	51.581
Total Displacement of Centroid/mm	100.559	101.393	100.470	100.887
Heading Safety Index	0.585	0.859	0.793	1.387

6.3. Navigation Correction Control Experiment

First, based on the established fuzzy neural network structure, the real-time adjustment of the proportional, integral, and differential control parameters of the PID is completed. The tuning curves of the relevant control parameters are shown in Figure 13. The simulation results demonstrate that the fuzzy neural network has a good online tuning capability for PID control parameters.

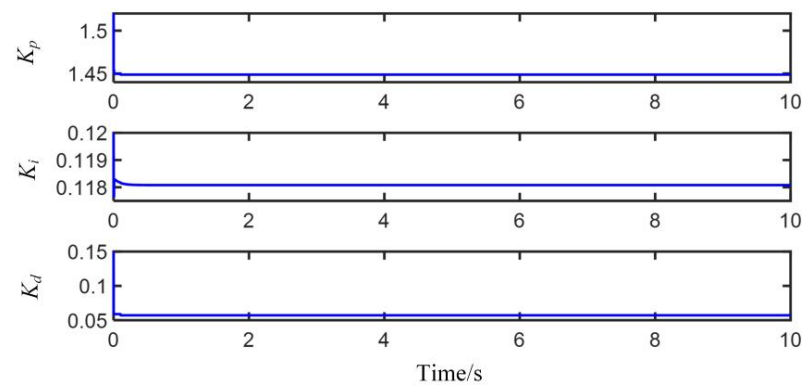


Figure 13. The changing curve of PID control parameters based on fuzzy neural network.

To test the basic performance of the control algorithm, a unit step input signal was applied to the system for tracking performance simulation experiments. Figure 14a shows the step response curve of the tracked roadheaders' deviation correction motion control. Compared with traditional PID control, the system response time of the FNN PID control algorithm is reduced by 55.03%, and the overshoot is reduced by 95.93%. The established FNN PID control algorithm demonstrates better dynamic performance, with specific simulation data shown in Table 8.

Table 8. The simulation results of heading rectification movement.

Control Algorithm	Input Signal/Parameter	Response Time/s	Error/Overshoot
PID	Step Signal	1.29	0.1407 mm
	Sinusoidal Signal	1.25	0.0847 mm
	Sawtooth Wave Signal	1.32	0.5991 mm
	Square Wave Signal	1.24	0.2736 mm
	Centroid Velocity	0.14	0.0175 m/s
	Angular Velocity	0.75	6.399 rad/s
	Track Speed	0.40	0.1768 m/s
	Centroid Coordinates	1.37	0.0013 mm
	Heading Angle	1.83	0.297°
FNN PID	Step Signal	0.58	0.071 mm
	Sinusoidal Signal	0.12	0.0645 mm
	Sawtooth Wave Signal	0.83	0.1443 mm
	Square Wave Signal	0.59	0.0164 mm
	Centroid Velocity	0.36	0.002 m/s
	Angular Velocity	0.66	2.752 rad/s
	Track Speed	0.41	0.0015 m/s
	Centroid Coordinates	0.24	0.001 mm
	Heading Angle	0.78	0.237°

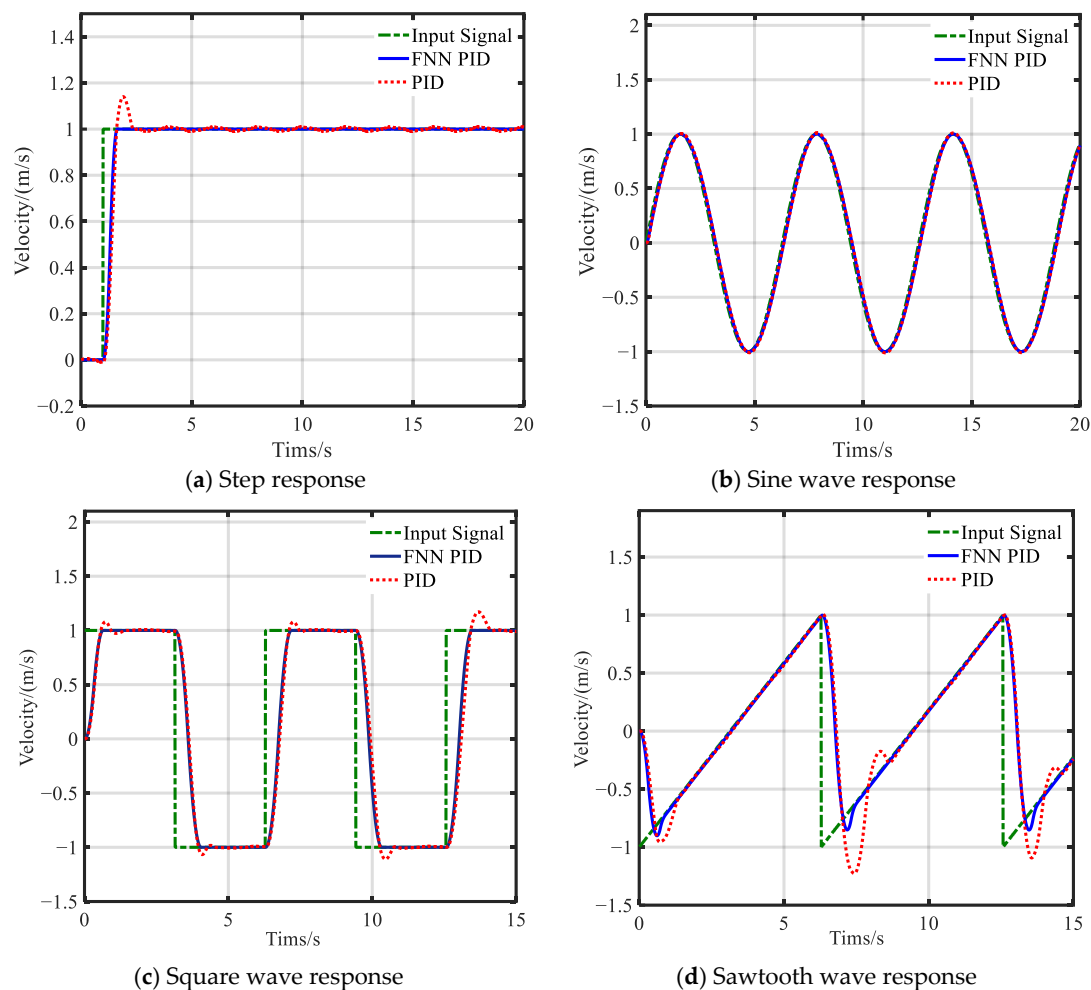


Figure 14. Input signal simulation and error curve.

At the same time, tracked roadheaders often need to perform repeated position and orientation detection and tracking deviation correction control during construction operations. Therefore, sine wave, sawtooth wave, and square wave signals were used as input signals for dynamic tracking deviation correction simulation experiments. The simulation results are shown in Figure 14b–d. The system using the FNN PID control algorithm reduced the response time by 90.4%, 37.1%, and 52.42% under the input of sine wave, sawtooth wave, and square wave signals, respectively. The dynamic errors caused by external disturbances were reduced by 23.85%, 75.91%, and 94.01%. It can be seen that the FNN PID control algorithm can maintain good stability in complex working conditions. The specific simulation data is shown in Table 8.

Figure 15 shows the variation curves of the linear velocity error and angular velocity error of the vehicle's center of mass during the path navigation and deviation correction motion of the tracked roadheaders. In Figure 15, the traditional PID control system has a faster response speed compared to the FNN PID, with more obvious oscillation amplitude and larger overshoot. The specific simulation data is shown in Table 8.

Figure 16 shows the variation curves of the rotational speed of the crawler tracks on both sides during the path navigation motion of the tracked roadheaders. In Figure 16, the FNN PID also has a slightly slower response speed compared to the traditional PID control system, but the oscillation amplitude generated by the traditional PID control system is particularly noticeable. The repeated oscillation of speed errors may lead to chattering phenomena during the vehicle's heading correction. The specific simulation data are shown in Table 8.

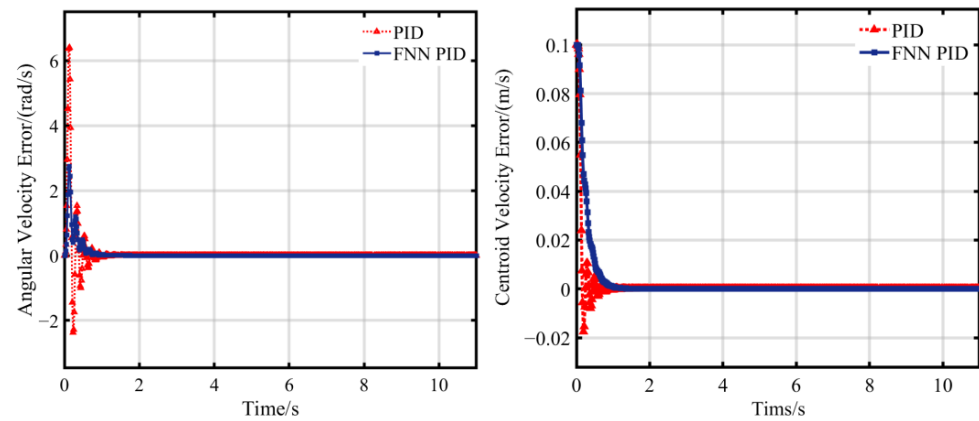


Figure 15. The changing curve of the center-of-mass velocity and angular velocity error.

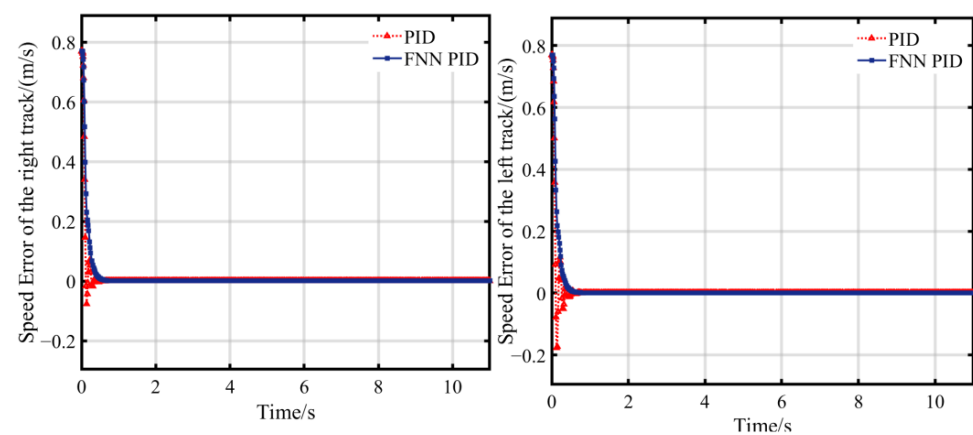


Figure 16. The variation curves of the speed error of the crawler tracks on both sides.

Figure 17 shows the variation curves of the X-axis and Y-axis coordinate deviations of the vehicle during the path point tracking and deviation correction control of the tracked roadheaders. In Figure 17, when the tracked roadheaders are performing navigation and deviation correction motion, the response time of the FNN PID control is reduced by 82.48% compared to the traditional PID control system, and the overshoot is reduced by 23.07%. The specific simulation data is shown in Table 8.

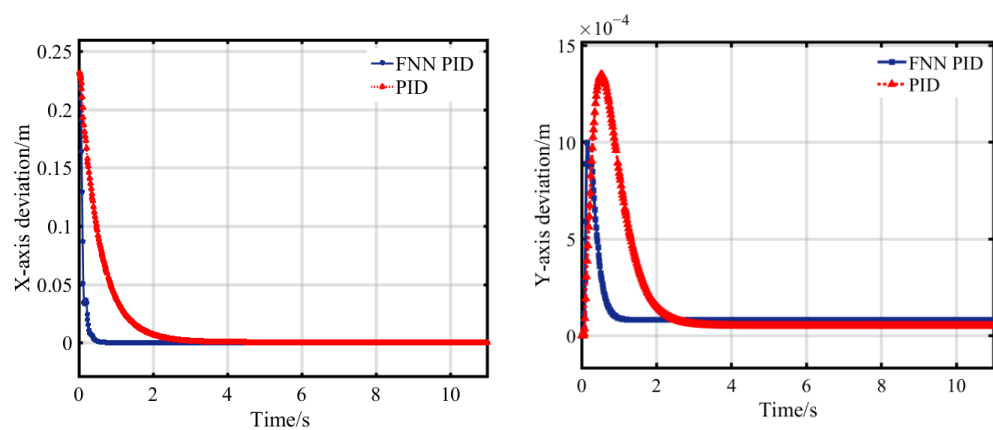


Figure 17. The changing curve of coordinate deviation of crawler support vehicle.

7. Field Experiment

7.1. Intelligent Onboard Controller and Pose Detection System

The EBZ-55 tracked roadheaders have a compact structure and adopt a hybrid transmission method. Its main parameters are listed in Table 9. Under the control of the remote intelligent control platform, the EBZ-55 tracked roadheaders can operate smoothly, and the hydraulic motor is externally assembled for convenient engineering experiments.

Table 9. Main parameters of EBZ-55 tracked roadheaders.

Main Parameter Name	Unit	Value
Dimensions	m ³	$5.79 \times 4.16 \times 2.56$
Track Width	m	0.33
Ground Pressure	MPa	0.113
Travel Speed	m/s	0.05
Gradeability	°	10
Supply Voltage	V	380
Motor Power	KW	45

The EBZ-55 tracked roadheaders require an intelligent onboard controller for local connection. This experiment uses the B&R X20 series 7CP1583 controller, which is equipped with a system adapter module, digital/analog input and output modules, and communication modules. During the vehicle heading correction control experiment, the information to be collected includes heading angle, position coordinates, as well as hydraulic motor speed and flow rate.

The EBZ-55 tracked roadheaders are equipped with a YHJ001 onboard laser rangefinder and a CAR-N28 ultrasonic sensor as auxiliary measurement components, mainly for measuring the vehicle's travel distance, the distance between the vehicle body and the obstacles, and the distance to the simulated tunnel. The laser rangefinder has a measurement accuracy of 0.01 mm, and the ultrasonic sensor has a measurement accuracy of 1 cm. The FOSN fiber optic strapdown inertial navigation system can measure the vehicle's heading angle and position coordinates in real-time during travel. The related pose detection system is shown in Figure 18.

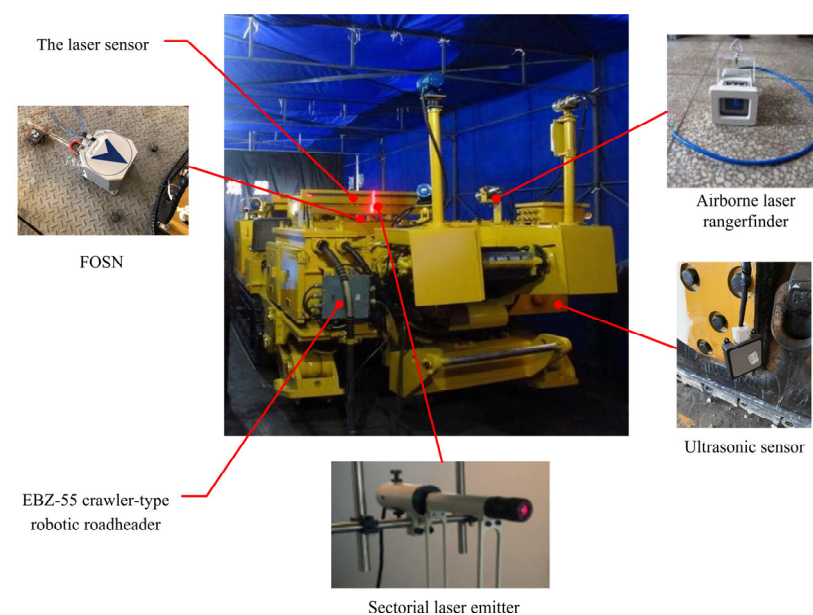


Figure 18. The pose detection system of the EBZ-55 tracked roadheaders.

7.2. Work Parameter Acquisition System

During engineering experiments, the pose detection system of the EBZ-55 tracked roadheaders can measure the main parameters of the machine body. At the same time, it is essential to collect the hydraulic and electrical system parameters of related equipment. Different types of sensors can be used to collect other relevant data in the control system and verify the accuracy of the main parameters in engineering experiments. The specific information of the installed sensors is shown in Table 10.

Table 10. Information table of sensors installed in hydraulic and electrical systems.

Installed Sensor Name	Selected Model	Measurement Range/Accuracy	Main Measurement Object
Pressure Sensor	MB300	0–16 MPa/0.1 Pa	System Pressure
Flow Sensor	PS604	2500 mL/min/5%	System Flow
Torque Sensor	HBM701	0.01 N·m	Hydraulic Motor Torque
Current Sensor	HKA-3YSD	0–50 A/0.1 A	Motor Current
Voltage Sensor	HV100	0–500 V/0.1 V	Motor Voltage
Motor Angle Sensor	ECN413	0.01°	Motor Angle
Photoelectric Sensor	QS18	–20–100 mm/0.1 mm/s	Track Drive Wheel Speed

The main purpose of the navigation and deviation correction control method for tracked engineering vehicles is to complete the vehicle's construction path planning and heading deviation correction motion control in a known tunnel environment. The path planning part requires calculating the optimal set of path points for travel, without the need for engineering experimental verification. The experimental part of the heading deviation correction motion control needs to use the EBZ-55 tracked roadheaders as the experimental subject for engineering control experiments.

The path point tracking and deviation correction control of the EBZ-55 tracked roadheaders requires the use of a remote intelligent monitoring system, with the APC2100 intelligent industrial computer as its core. First, the deviation correction control algorithm is written and burned into the intelligent industrial computer. Then, the intelligent industrial computer issues control commands, which are transmitted via fiber optic communication using the IPX/SPX protocol to the OPC protocol server, converted and transmitted to the onboard programmable controller. The CAN field control bus with an RS485 electrical interface is used to control the movement of the valve core, adjust the hydraulic flow to control the movement of the hydraulic motor, and the NI6032 acquisition card is used for data collection.

7.3. Motion Control Experiment

According to Section 5, the heading deviation correction motion control of the EBZ-55 tracked roadheaders can be divided into coordinate-based straight-line tracking motion and in-place turning correction motion. During the experiment, the vehicle's maximum speed is about 1 m/s, and the experiment is conducted on a hard cement floor.

7.3.1. Straight-Line Tracking and Deviation Correction Motion Control Experiment

The initial and target coordinates of the EBZ-55 tracked roadheaders in the two-dimensional XOY Cartesian coordinate system of the simulated tunnel are set to (0.8, –0.8) and (0, 0), respectively. The straight-line tracking and deviation correction motion experiment lasts for 10 s, and the experimental results of the straight-line tracking and deviation correction motion in the simulated tunnel corridor are shown in Figures 19 and 20.

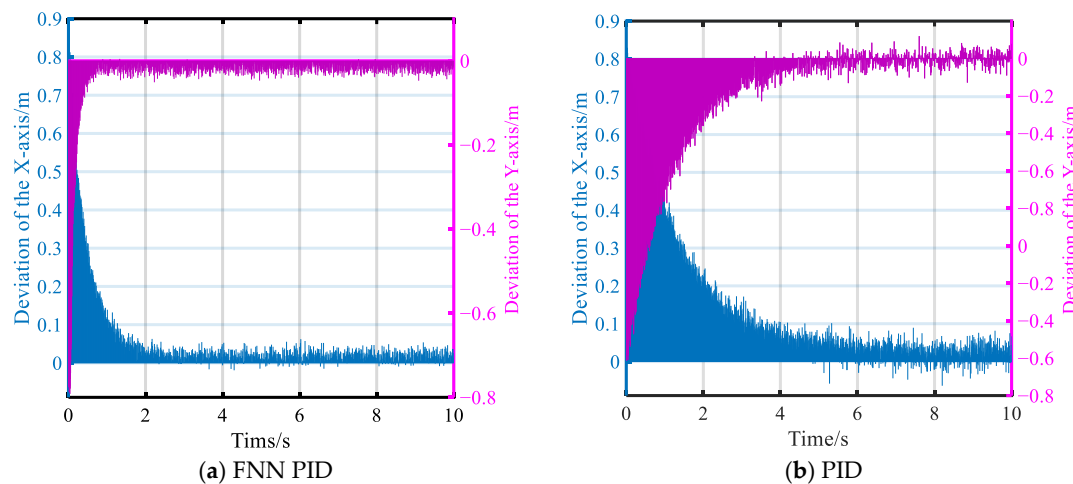


Figure 19. The changing curve of positional deviation in X and Y axial in the process of rectification.

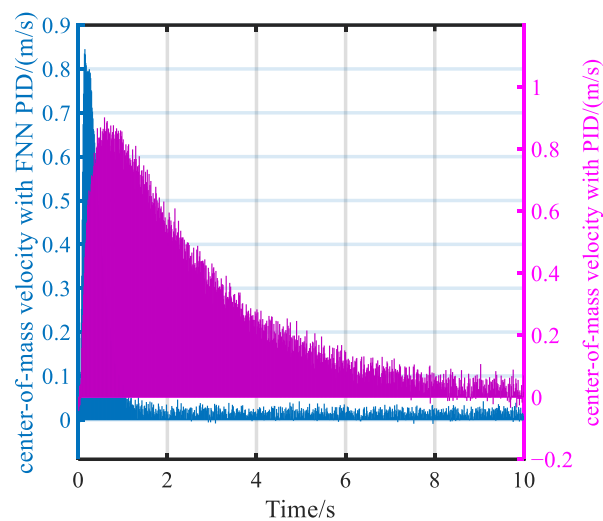


Figure 20. The changing curve of center-of-mass velocity in the process of rectification.

Figure 19 shows the coordinate deviation change curves of the EBZ-55 tracked roadheaders on the X-axis and Y-axis in the simulated tunnel, respectively. After the EBZ-55 tracked roadheaders starts the deviation correction motion, the deviations in the horizontal and vertical coordinates of the position detection system continuously decrease, and the deviations on the X-axis and Y-axis stabilize after about 1.5 s. The control signals of the EBZ-55-tracked roadheaders's deviation correction motion are affected by external disturbances, the vehicle's body inertia jitter, and the measurement system's uncertain errors, resulting in some overshoot and amplitude fluctuations in the deviation correction motion.

Figure 20 shows the linear velocity change curve of the vehicle's center of mass during the straight-line deviation correction motion of the EBZ-55 tracked roadheaders. The linear velocity of the vehicle's center of mass rapidly increases after the deviation correction motion starts, reaches the maximum speed, and then gradually decelerates to zero. Due to the influence of inertial jitter of large tracked roadheaders, the measured curve of the position detection system always exhibits amplitude fluctuations. However, compared to the traditional PID control system, the FNN PID has significant advantages in response time and error fluctuations. The experimental data statistics for the straight-line tracking and deviation correction motion are shown in Table 11.

Table 11. The experimental data of linear tracking motion of EBZ-55 tracked roadheaders in the process of rectification.

Related Parameters/Units	FNN PID	Traditional PID
Maximum X-axis Position Error/mm	36.14	64.02
Maximum Y-axis Position Error/mm	51.06	64.38
Maximum Centroid Velocity/(m/s)	0.8452	0.9024

As shown in Table 11, when the EBZ-55 tracked roadheaders prototype performs straight-line tracking and deviation correction motion control in the simulated tunnel corridor, the FNN PID control algorithm reduces the position fluctuation error on the X-axis by approximately 43.55%, the position fluctuation error on the Y-axis by approximately 20.69%, and the speed fluctuation error by approximately 6.34%, verifying the effectiveness of the FNN PID control algorithm in straight-line tracking motion control.

7.3.2. In Situ Steering Rectification Motion Control Experiment

First, the pose detection system was used to measure the initial heading angle of the EBZ-55 tracked roadheaders at the starting point in the simulated tunnel corridor as 5.5° . The target heading angle was set to 0° , and the total duration of the in situ steering rectification motion control experiment was 5 s. The vehicle achieved heading angle correction through the rotation of both tracks.

From Figure 21, it can be seen that when the EBZ-55 tracked roadheaders performs steering motion correction, the FNN PID control algorithm stabilizes the heading angle deviation near zero after about 0.3 s, while the traditional PID control takes about 0.7 s to complete. Meanwhile, it can be observed that when the vehicle performs steering correction, the angular velocity first increases and then decelerates to zero. There are many interference factors during the correction control process, and the heading angle deviation often exhibits overshoot and amplitude fluctuations. However, compared to the traditional PID control system, the FNN PID control algorithm has a significant advantage in response time. The experimental data statistics of the correction motion control are shown in Table 12.

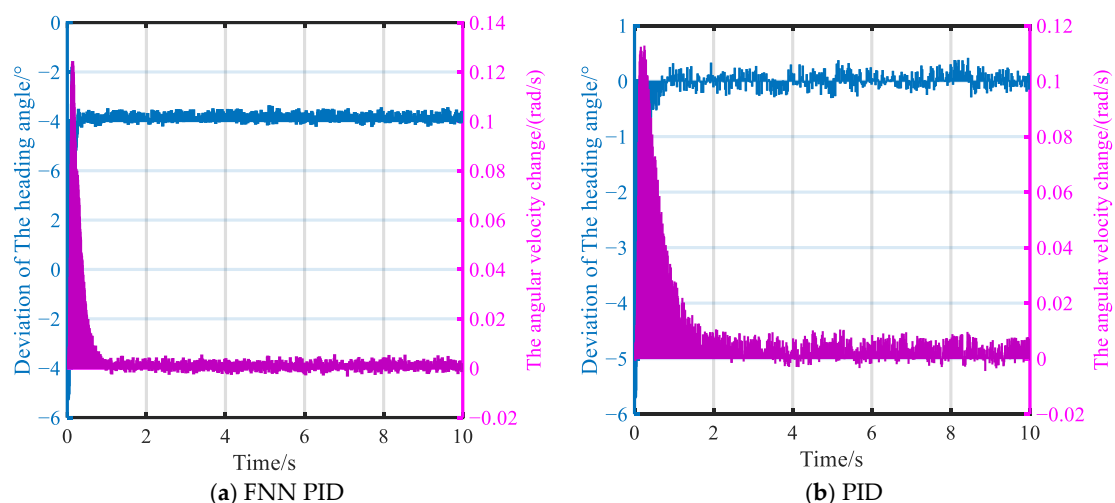
**Figure 21.** The heading angle change curve and angular velocity change curve during the steering correction motion.

Table 12. Experimental data of the steering rectification movement of the EBZ-55 tracked roadheaders during the rectification process.

Related Parameters/Unit	FNN PID	Traditional PID
Maximum Heading Angle Error/ $^{\circ}$	0.31	0.43
Maximum Angular Velocity Error/(rad/s)	0.0076	0.0105

As shown in Table 12, when the EBZ-55 tracked roadheaders performs steering correction on a hard cement floor, the FNN PID control algorithm reduces the fluctuation error of the heading angle by 26.19% compared to PID control. Although the difference in angular velocity fluctuation error is minimal, it to some extent verifies the accuracy of the FNN PID control algorithm in the simulation environment for the heading correction motion control of tracked roadheaders.

8. Conclusions

This study proposes an integrated solution for path navigation and deviation correction control of tracked roadheaders in confined underground coal mine spaces, combining scenario modeling, path planning, and adaptive control. The effectiveness is verified through simulations and experiments, with key contributions summarized as follows:

(1) Working Condition Modeling and Simplification

Through 2D grid modeling and heading reference influence degree threshold analysis, the complex roadway environment is simplified into a low-dimensional representation. This method retains essential road condition characteristics while reducing path planning complexity.

(2) Kinematic Analysis and Controller Design

A kinematic model considering slip influence is established based on the steering dynamics of tracked roadheaders. An adaptive PID control algorithm integrating Lyapunov stability theory and fuzzy neural networks is designed, enhancing the accuracy and robustness of heading correction.

(3) Optimization of Intelligent Algorithms

By introducing crossover and mutation operations into the particle swarm algorithm, the improved method overcomes local optima limitations. Experimental results demonstrate its superiority over standard algorithms in both path planning efficiency and safety performance.

(4) Experimental Validation and Application

Field tests using an EBZ-55 roadheader confirm the advantages of the proposed FNN-PID control in straight-line tracking and steering correction, showing significant improvements in response time and overshoot reduction.

Author Contributions: Conceptualization, R.L. and D.W.; methodology, D.W.; software, W.Z.; validation, D.W. and M.W.; formal analysis, R.L. and T.L.; resources, M.W.; funding acquisition, M.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by Found of the National Natural Science Foundation of China grant number 52474187.

Data Availability Statement: The data used to support the findings of this study are included within the article.

Conflicts of Interest: The authors declare that they have no conflict of interest.

References

- Li, R.; Jiang, W.; Wang, P.; Cheng, J.; Wu, M. Asynchronous coupling approach for leveling control of self-shifting temporary support. *J. China Coal Soc.* **2020**, *45*, 3625–3635.
- Wang, D.; Cheng, J.; Li, R.; Qu, Y.; Jiang, H.; Shen, Y.; Zhao, S.; Wu, M. Research on the Twin-Link Stepping Temporary Support Mechanism and the Repeatable Supportability of the Roof in the Underground Excavation Roadway of Coal Mine. *Appl. Sci.* **2024**, *14*, 6659. [\[CrossRef\]](#)
- Yang, J.; Zhang, Q.; Wang, C.; Chang, B.; Wang, X.L.; Ge, S.R.; Wu, M. Status quo and development of robotization research of coal mine China Coal. *J. China Coal Soc.* **2020**, *45*, 2995–3005.
- Niu, C.; Li, A.; Huang, X.; Li, W.; Xu, C. Research on global dynamic path planning method based on improved A* algorithm. *Math. Probl. Eng.* **2021**, *2021*, 4977041. [\[CrossRef\]](#)
- Wang, D.; Wei, W.; Yeboah, Y.; Li, Y.; Gao, Y. A robust model predictive control strategy for trajectory tracking of omni-directional mobile robots. *J. Intell. Robot. Syst.* **2020**, *98*, 439–453. [\[CrossRef\]](#)
- Saputra, M.R.U.; Lu, C.X.; de Gusmao, P.P.B.; Wang, B.; Markham, A.; Trigoni, N. Graph-based thermal-inertial SLAM with probabilistic neural networks. *IEEE Trans. Robot.* **2021**, *38*, 1875–1893. [\[CrossRef\]](#)
- Wang, N.; Zhou, J.; Zhong, K.; Zhang, X.; Chen, W. Force tracking impedance control based on contour following algorithm. In *International Conference on Intelligent Robotics and Applications*; Springer International Publishing: Cham, Switzerland, 2022; pp. 698–709.
- Lin, J.; Zheng, R.; Zhang, Y.; Feng, J.; Li, W.; Luo, K. CFHBA-PID algorithm: Dual-loop PID balancing robot attitude control algorithm based on complementary factor and honey badger algorithm. *Sensors* **2022**, *22*, 4492. [\[CrossRef\]](#)
- Nguyen, A.T.; Taniguchi, T.; Eciolaza, L.; Campos, V.; Palhares, R.; Sugeno, M. Fuzzy control systems: Past, present and future. *IEEE Comput. Intell. Mag.* **2019**, *14*, 56–68. [\[CrossRef\]](#)
- Driankov, D.; Hellendoorn, H.; Reinfrank, M. *An Introduction to Fuzzy Control*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2013.
- Li, R. Research on the Key Technology of Self-Moving Over-Supporting in Integrated Roadway Excavation. Ph.D. Thesis, Mechanical Electronic and Information Engineering, China University of Mining & Technology (Beijing), Beijing, China, 2021.
- Ding, S.; Bai, J.; Han, J.; Wang, H.; Ma, F. Mechanical Characteristics Analysis and Structural Optimization of Key Component of Self-Moving Temporary Support. *Appl. Sci.* **2022**, *12*, 10745. [\[CrossRef\]](#)
- Ducros, C.; Hauser, G.; Mahjoubi, N.; Girones, P.; Boisset, L.; Sorin, A.; Jonquet, E.; Falciola, J.M.; Benhamou, A. RICA: A tracked robot for sampling and radiological characterization in the nuclear field. *J. Field Robot.* **2017**, *34*, 583–599. [\[CrossRef\]](#)
- Ugenti, A.; Galati, R.; Mantriota, G.; Reina, G. Analysis of an all-terrain tracked robot with innovative suspension system. *Mech. Mach. Theory* **2023**, *182*, 105237. [\[CrossRef\]](#)
- Chen, K.; Kamezaki, M.; Katano, T.; Kaneko, T.; Azuma, K.; Ishida, T.; Ishida, T.; Seki, M.; Ichiryu, K. Compound locomotion control system combining crawling and walking for multi-crawler multi-arm robot to adapt unstructured and unknown terrain. *ROBOMECH J.* **2018**, *5*, 1–17. [\[CrossRef\]](#)
- Dissanayake, M.; Sattar, T.P.; Howlader, O.; Pinson, I.; Gan, T.H. Tracked-wheel crawler robot for vertically aligned mooring chain climbing design, simulation and validation of a climbing robot for mooring chains. In *Proceedings of the 2017 IEEE International Conference on Industrial and Information Systems (ICIIS)*, Peradeniya, Sri Lanka, 15–16 December 2017; IEEE: Piscataway, NJ, USA, 2017.
- Grazioso, A.; Ugenti, A.; Galati, R.; Mantriota, G.; Reina, G. Modeling and validation of a novel tracked robot via multibody dynamics. *Robotica* **2023**, *41*, 3211–3232. [\[CrossRef\]](#)
- Zhou, P.; Hu, X.; Zhu, Z.; Ma, J. What is the most suitable Lyapunov function? *Chaos Solitons Fractals* **2021**, *150*, 111154. [\[CrossRef\]](#)
- Anand, A.; Seel, K.; Gjerum, V.; Håkansson, A.; Robinson, H.; Saad, A. Safe learning for control using control lyapunov functions and control barrier functions: A review. *Procedia Comput. Sci.* **2021**, *192*, 3987–3997. [\[CrossRef\]](#)
- Ziemann, I.; Tsiamis, A.; Lee, B.; Jedra, Y.; Matni, N.; Pappas, G.J. A tutorial on the non-asymptotic theory of system identification. In *Proceedings of the 2023 62nd IEEE Conference on Decision and Control (CDC)*, Singapore, 13–15 December 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 8921–8939.
- Gad, A.G. Particle swarm optimization algorithm and its applications: A systematic review. *Arch. Comput. Methods Eng.* **2022**, *29*, 2531–2561. [\[CrossRef\]](#)
- Li, J.; Zong, T.; Lu, G. Parameter identification of Hammerstein–Wiener nonlinear systems with unknown time delay based on the linear variable weight particle swarm optimization. *ISA Trans.* **2022**, *120*, 89–98. [\[CrossRef\]](#)
- Zacharakis, I.; Giagopoulos, D. Vibration-based damage detection using finite element modeling and the metaheuristic particle swarm optimization algorithm. *Sensors* **2022**, *22*, 5079. [\[CrossRef\]](#)
- Ahandani, M.A.; Abbasfam, J.; Kharrati, H. Parameter identification of permanent magnet synchronous motors using quasi-opposition-based particle swarm optimization and hybrid chaotic particle swarm optimization algorithms. *Appl. Intell.* **2022**, *52*, 13082–13096. [\[CrossRef\]](#)

25. Wang, D.; Tan, D.; Liu, L. Particle swarm optimization algorithm: An overview. *Soft Comput.* **2018**, *22*, 387–408. [[CrossRef](#)]
26. Alateeq, M.; Pedrycz, W. Logic-oriented fuzzy neural networks: A survey. *Expert Syst. Appl.* **2024**, *257*, 125120. [[CrossRef](#)]
27. Zhang, B.; Gong, X.; Wang, J.; Tang, F.; Zhang, K.; Wu, W. Nonstationary fuzzy neural network based on FCMnet clustering and a modified CG method with Armijo-type rule. *Inf. Sci.* **2022**, *608*, 313–338. [[CrossRef](#)]
28. Al Sumarmad, K.A.; Sulaiman, N.; Wahab, N.I.A.; Hizam, H. Energy management and voltage control in microgrids using artificial neural networks, PID, and fuzzy logic controllers. *Energies* **2022**, *15*, 303. [[CrossRef](#)]
29. Chadalavada, R.T.; Andreasson, H.; Schindler, M.; Palm, R.; Lilienthal, A.J. Bi-directional navigation intent communication using spatial augmented reality and eye-tracking glasses for improved safety in human–robot interaction. *Robot. Comput.-Integr. Manuf.* **2020**, *61*, 101830. [[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.