



Article

Biomimetic Adaptive Pure Pursuit Control for Robot Path Tracking Inspired by Natural Motion Constraints

Suna Zhao ¹, Guangxin Zhao ¹, Yan He ¹, Zhihua Diao ¹, Zhendong He ¹ , Yingxue Cui ¹, Liying Jiang ¹, Yongpeng Shen ^{1,*} and Chao Cheng ^{2,*}

¹ College of Electrical and Information Engineering, Zhengzhou University of Light Industry, Zhengzhou 450000, China; snzhao1221@zzuli.edu.cn (S.Z.); 332101050067@email.zzuli.edu.cn (G.Z.); heyhan@zzuli.edu.cn (Y.H.); 2010018@email.zzuli.edu.cn (Z.D.); hezhendong@email.zzuli.edu.cn (Z.H.); 332101030024@email.zzuli.edu.cn (Y.C.); jiangliying@zzuli.edu.cn (L.J.)

² Key Laboratory of Bionic Engineering, Ministry of Education, Jilin University, Changchun 130022, China

* Correspondence: shenyongpeng@zzuli.edu.cn (Y.S.); chengchao@jlu.edu.cn (C.C.)

Abstract: The essence of biomimetics in human–computer interaction (HCI) is the inspiration derived from natural systems to drive innovations in modern-day technologies. With this in mind, this paper introduces a biomimetic adaptive pure pursuit (A-PP) algorithm tailored for the four-wheel differential drive robot (FWDDR). Drawing inspiration from the intricate natural motions subjected to constraints, the FWDDR’s kinematic model mirrors non-holonomic constraints found in biological entities. Recognizing the limitations of traditional pure pursuit (PP) algorithms, which often mimic a static behavioral approach, our proposed A-PP algorithm infuses adaptive techniques observed in nature. Integrated with a quadratic polynomial, this algorithm introduces adaptability in both lateral and longitudinal dimensions. Experimental validations demonstrate that our biomimetically inspired A-PP approach achieves superior path-following accuracy, mirroring the efficiency and fluidity seen in natural organisms.

Keywords: human–computer interaction; FWDDR; path following; bionic motion; kinematic inspiration; adaptive techniques; quadratic polynomial integration



Citation: Zhao, S.; Zhao, G.; He, Y.; Diao, Z.; He, Z.; Cui, Y.; Jiang, L.; Shen, Y.; Cheng, C. Biomimetic Adaptive Pure Pursuit Control for Robot Path Tracking Inspired by Natural Motion Constraints. *Biomimetics* **2024**, *9*, 41. <https://doi.org/10.3390/biomimetics9010041>

Academic Editor: Jinyou Shao

Received: 1 November 2023

Revised: 30 November 2023

Accepted: 28 December 2023

Published: 9 January 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/>).

1. Introduction

In recent decades, wheeled mobile robots have garnered increased attention from researchers. Owing to their exceptional load-bearing capacity, flexibility, and strong stability [1], these robots are extensively used in fields such as modern industry, logistics, transportation, and firefighting. Numerous scholars have conducted relevant research on mobile robots, making significant contributions to automatic navigation [2–10]. Path tracking, a key technology in automatic navigation, critically determines the performance accuracy of autonomous driving. Therefore, it is vital to study path-tracking control algorithms to enhance the autonomous navigation capabilities of robots [11].

Path tracking involves controlling the robot to follow a predetermined path closely, minimize tracking errors, and maintain high robustness against disturbances. The design of the controller is crucial for path following. Consequently, many scholars have conducted corresponding research and proposed various control algorithms. These control algorithms can be mainly categorized into direct feedback control, dynamic model-based control, and geometric tracking control. The direct feedback control primarily adjusts the wheel’s rotation angle based on system feedback without considering the robot’s model, parameters, or other information. For example, methods like the PID (Proportional Integral Derivative) control [12] and the Fuzzy control [13] fall under this category. In [14], a genetic algorithm (GA) was introduced to enhance PID tracking, effectively overcoming external interferences. In [15], fuzzy logic control was integrated with predictive control to improve the robustness and immunity of the controller and to overcome the time delay caused by the sensors.

In addition, in [16,17], the hybrid algorithm of fuzzy logic and Adaptive Neuro-Fuzzy Inference System (ANFIS) also provides some inspiration. The dynamics model-based control primarily analyzes the relationship between the robot's forces and dynamics parameters, like velocity, using the established robot dynamics model. For example, methods like Model Predictive Control (MPC) [18] and Linear Quadratic Regulator (LQR) control [19] fall under this category. In [20], an MPC-based interval trajectory tracking control method was designed to enhance the speed and stability of AGV travel in a feasible area by increasing the limit of the yaw angle. In [21], a fast rolling optimization algorithm was designed based on the LQR lateral controller for a tracked mobile robot, and the weight coefficient matrices Q and R were optimized to improve the tracking accuracy and dynamic performance of the controller. The geometric tracking control method emphasizes analyzing the robot's geometric relationship with the reference path to determine wheel rotation angles. There are many typical representatives of geometric tracking control methods, such as the PP control method [22] and the Stanley control method. In [23], an improved PP control algorithm was proposed, which adaptively modified the pre-sight distance by selecting the temporary pre-sight point, and improved robot tracking accuracy. In [24], an adaptive steering controller based on the Stanley tracking controller was studied. This system integrates a database driven by the particle swarm optimization algorithm and a fuzzy monitoring strategy, enhancing the adaptability of armored vehicles to trajectory alterations.

Among the prevalent control methods, PID and LQR demonstrate reduced robustness against external disturbances and are less stable. MPC, though sophisticated, demands rigorous modeling and intense computational efforts, compromising its real-time efficiency. Meanwhile, the Stanley method necessitates a curvature-continuous path. In contrast, PP control offers robust resilience to external interference without specific path prerequisites. Although the algorithm simulates human driving habits and is progressive, how to determine the best forward-looking distance is still a problem. Addressing this, this paper introduces an adaptive pure pursuit (A-PP) algorithm. This algorithm enhances the quadratic polynomial in both lateral and longitudinal dimensions to adaptively adjust the forward-looking distance. This makes the tracking more accurate.

The main contributions of this paper are as follows: (1) At the lateral level, the lateral error is taken into account. When the error increases, reduce the forward-looking distance in order to eliminate the error as soon as possible and improve the response speed. When the error decreases, increase the forward-looking distance appropriately to avoid the system response overshoot. (2) At the longitudinal level, the path curvature is taken into account and the forward-looking distance is adjusted according to the road conditions of the reference path, featuring online adaption of the control parameters, especially the lateral error during the turning section.

The remaining sections of this paper are organized as follows: In Section 2, the FWDDR kinematic model is established, and the relationship between pose and velocities is elucidated by analyzing the motion characteristics of FWDDR. In Section 3, the principles of the traditional algorithm and A-PP algorithm are introduced. In Section 4, the feasibility of the A-PP algorithm is assessed via simulation. In Section 5, the A-PP algorithm's efficacy is demonstrated in a real-world setting using FWDDR. Section 6 offers a comprehensive summary.

2. Kinematic Model

2.1. Kinematic Analysis

Generally, differential drive robots lack enough torque to drive the wheels directly. In order to augment the torque, robot wheels are often connected directly to the reduction motor. The four-wheel differential structure derives its steering power from the left and right motor differential. Power, once generated from the motors, is transmitted to both the left and right front and rear axles via a reducer, eventually reaching the wheels. The FWDDR relies on motors to drive the wheel movement independently, which has the advantages of potent driving force and agile control. During operation, deviations exist between the

robot’s actual and ideal motion states. Considering the non-holonomic constraints of the FWDDR, the following conditions should be assumed [25–27].

- (1) Four wheels of the FWDDR are symmetrically distributed on the same plane;
- (2) None of the FWDDR’s wheels idle;
- (3) The FWDDR does not exhibit longitudinal skidding during steering;
- (4) The FWDDR possesses an ample turning radius;
- (5) The FWDDR’s center of mass is situated on the robot’s x-axis.

The kinematic diagram of the FWDDR is shown in Figure 1. XOY represents the global coordinate system, with xo,y illustrating the local or body coordinate system. The parameters i (1, 2, 3, 4) symbolize the robot’s four wheels. Points (A, B, C, D) indicate where the tire contacts the ground, b denotes the spacing between rear wheels, and O_r and G symbolize the FWDDR’s particle and geometric centers, respectively.

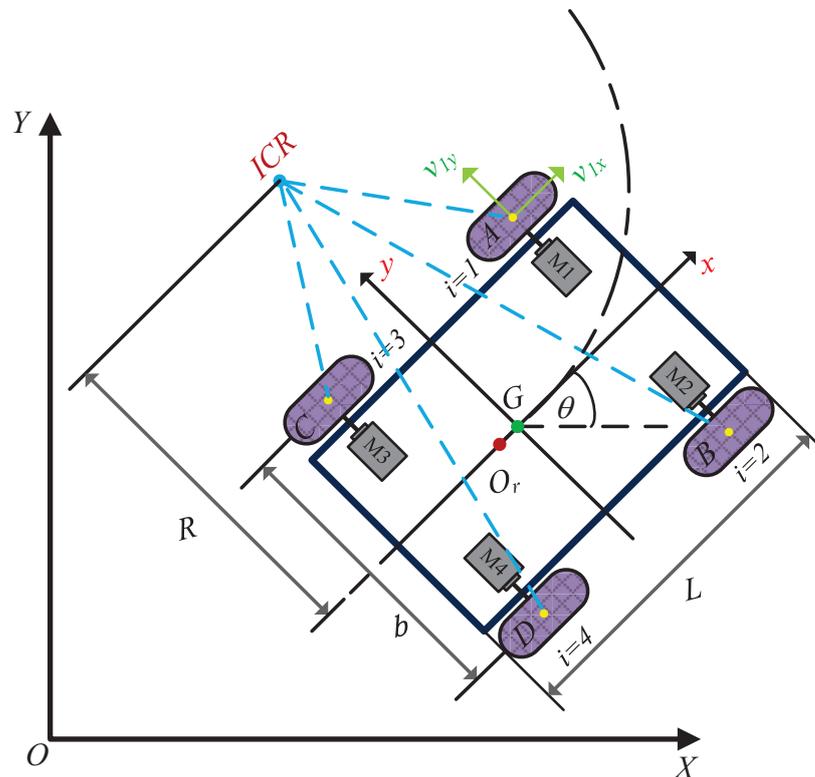


Figure 1. Kinematics of an FWDDR.

The velocity direction of ideal contact points $A, B, C,$ and D between the tire and the ground is perpendicular to the radial direction of the radius of gyration. Furthermore, it can be decomposed into longitudinal component velocity along the rolling direction of the wheel and transverse component velocity along the axial direction of the motor, which can be written as:

$$\begin{cases} v_l = v_{1x} = v_{3x} \\ v_r = v_{2x} = v_{4x} \\ v_f = v_{1y} = v_{2y} \\ v_b = v_{3y} = v_{4y} \end{cases} \quad (1)$$

where v_l and v_r are the longitudinal partial velocities of the left and right wheels, and v_b and v_f are the transverse partial velocities of the front and rear wheels, respectively. From (1), it can be seen that the longitudinal partial velocity of wheels on the same side is equal, and similarly, the transverse partial velocity of wheels on the same end is equal [28].

2.2. Equation of Motion

In Figure 1, the point *ICR* signifies the robot's instantaneous center of rotation, located on the *y*-axis of the center-of-mass coordinate system. Meanwhile, *R* denotes the instantaneous radius of rotation during motion [29,30].

Due to the $v = \omega \cdot R$, if ω is constant, v is proportional to R , the rotation angular velocity of the robot can be expressed as:

$$\omega = \frac{v}{R} = \frac{v_r}{R + b/2} = \frac{v_l}{R - b/2} \quad (2)$$

where v_r and v_l denote the speeds of the right and left wheels, respectively, v signifies the linear velocity, and w is the rotational angular velocity.

According to (2), the following can be obtained:

$$\omega = \frac{v_r - v_l}{b} \quad (3)$$

$$v = \omega R = \frac{v_r + v_l}{2} \quad (4)$$

Combined with Formulas (3) and (4), the FWDDR's instantaneous radius of rotation R can be expressed as:

$$R = \frac{b}{2} \frac{v_r + v_l}{v_r - v_l} \quad (5)$$

Combined with Figure 1, matrix form of the kinematic model can be derived as follows:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} \quad (6)$$

By substituting (3) and (4) into (6), we obtain:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{\cos \theta}{2} & \frac{\cos \theta}{2} \\ \frac{\sin \theta}{2} & \frac{\sin \theta}{2} \\ \frac{1}{b} & -\frac{1}{b} \end{bmatrix} \begin{bmatrix} v_r \\ v_l \end{bmatrix} \quad (7)$$

Equation (7) is the general form of the kinematic model of the FWDDR, establishing the relationship between the robot's position and the speeds of the left and right wheels.

3. Path-Tracking Controller Design

Path tracking involves adjusting a robot's steering to align with a desired trajectory by assessing the deviation between its current position and the trajectory's target point. Figure 2 presents the general block diagram for FWDDR path tracking.

Reference trajectory is obtained through the planning layer. The reference trajectory's desired attitude is relayed to the subsequent layer as the target for trajectory tracking. The subsequent layer consists of a trajectory-tracking controller, designed based on the kinematic model. It utilizes an optimization algorithm to transmit speed instructions to the motors, achieving trajectory tracking.

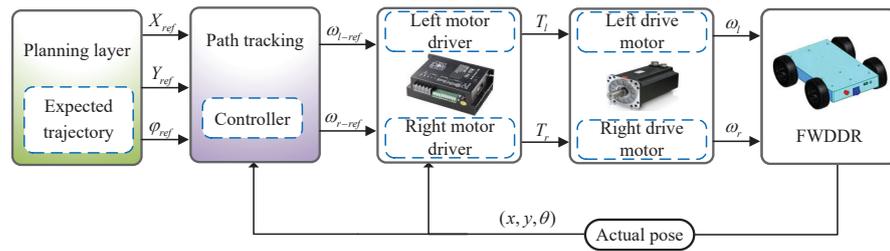


Figure 2. General block diagram of FWDDR path tracking.

3.1. Traditional PP Control

The PP algorithm’s primary characteristic is its robustness against external interference, with modest path requirements. It is apt for both continuous and discontinuous path scenarios, boasting a broad application spectrum. This algorithm is a typical lateral control method. The main idea involves identifying a preview point from the rear wheel’s central position towards the desired trajectory. Then, it is assumed that the vehicle can travel along an arc passing through the preview point, and the vehicle angle is determined according to the geometric relationship among the preview distance, rotation radius, and rotation angle [31]. Thus, the mobile robot achieves precise tracking of the desired trajectory. As shown in Figure 3, point A represents the FWDDR’s rear wheel center, whereas point B denotes the reference trajectory’s preview point, distanced l_d from point A. The robot takes point O as the circle center and R as the rotation radius. Following a circular motion with an angle of 2α , the robot reaches the preview point B to achieve the desired trajectory tracking.

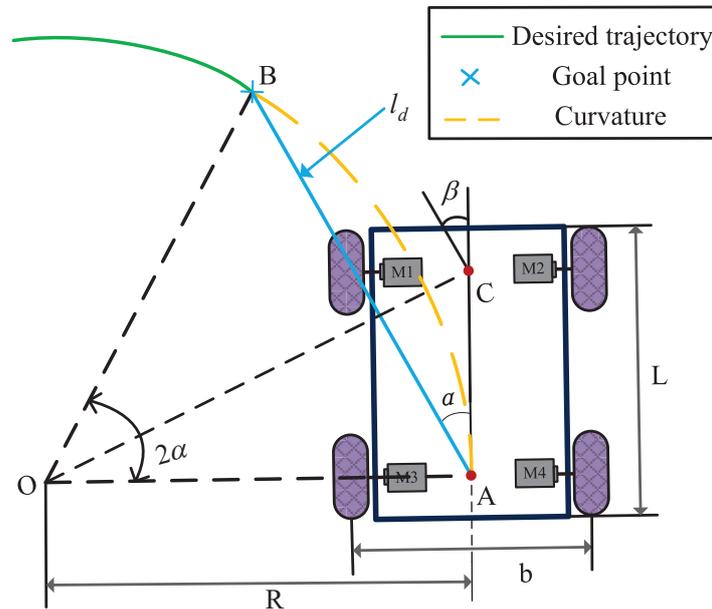


Figure 3. Schematic diagram of PP algorithm of the FWDDR.

To ensure that the rear wheel’s center point (point A) of the robot follows the dotted arc to reach point B, the following conditions must be satisfied:

$$\angle AOB = \pi - 2\angle CAB = \pi - 2\left(\frac{\pi}{2} - \alpha\right) = 2\alpha \tag{8}$$

Note that in $\triangle AOB$, the law of sine is satisfied and is given as:

$$\frac{AB}{\sin(2\alpha)} = \frac{AO}{\sin\left(\frac{\pi}{2} - \alpha\right)} \tag{9}$$

Which can then be written as:

$$\frac{l_d}{\sin(2\alpha)} = \frac{R}{\sin(\frac{\pi}{2} - \alpha)} \tag{10}$$

$$\frac{l_d}{2 \sin \alpha \cos \alpha} = \frac{R}{\cos \alpha} \tag{11}$$

The formula of robot radius of rotation can be simplified as:

$$R = \frac{1}{2} \frac{l_d}{\sin \alpha} \tag{12}$$

Substituting (5) into (12), it can be obtained:

$$\frac{l_d}{\sin \alpha} = \frac{b(v_r + v_l)}{v_r - v_l} \tag{13}$$

By solving simultaneous Equations (4) and (13), the left and right wheel velocities of the FWDDR under PP algorithm can be obtained:

$$\begin{cases} \frac{v_r}{v} = 1 + \frac{L \sin \alpha}{l_d} \\ \frac{v_l}{v} = 1 - \frac{L \sin \alpha}{l_d} \end{cases} \tag{14}$$

The lateral error between the mobile robot's current pose and the target point, denoted as e_r , can be formulated as:

$$e_r = l_d \sin \alpha \tag{15}$$

The lateral error e_r can also be expressed as:

$$e_r = \frac{l_d^2}{2L} \beta \tag{16}$$

where L is the long wheelbase of the FWDDR, β is the steering angle.

The equations highlight that the PP algorithm essentially acts as a P controller for lateral cornering, with the preview distance l_d being crucial for effective tracking. In general, a linear relationship is established between the preview distance and the current vehicle speed, as defined in [32]:

$$l_d = kv + l_{d0} \tag{17}$$

where k is an adjustable parameter, l_{d0} is the preset preview distance, and v is the current vehicle speed.

Substituting (17) into (14), it can be obtained:

$$\begin{cases} \frac{v_r}{v} = 1 + \frac{L \sin \alpha}{kv + l_{d0}} \\ \frac{v_l}{v} = 1 - \frac{L \sin \alpha}{kv + l_{d0}} \end{cases} \tag{18}$$

The aforementioned formulas elucidate the relationship between k and the speeds of the left and right wheels. In fact, the preview distance profoundly influences control effects. When the look-ahead distance is longer, the control effect will be smoother, but it will lead to premature steering. Conversely, when the look-ahead distance is shorter, the control effect will be more accurate, but it will bring some oscillation and lead to an unstable tracking state, as shown in Figure 4.

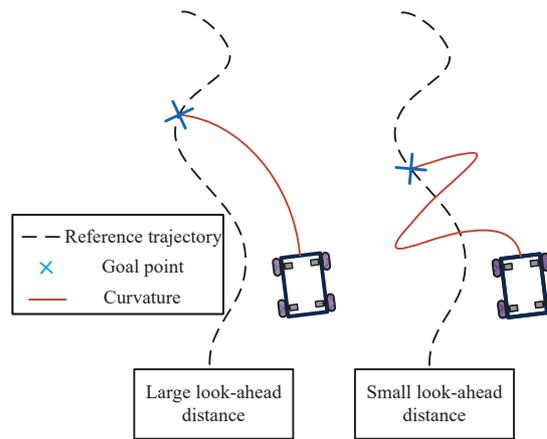


Figure 4. Renderings at different look-ahead distances.

3.2. Quadratic Polynomial-Based A-PP Control

As analyzed in the previous section, the look-ahead distance not only has a significant impact on the tracking effect but also affects the operational stability of the mobile robot to a certain extent. However, in a traditional PP algorithm, the look-ahead distance remains a constant value, which lacks the ability of randomization. Based on human driving habits, the look-ahead distance should correlate with parameters such as speed and path conditions.

Reference [33] considers the mathematical relationship between the look-ahead distance and the speed, and after several debug sessions, a formula was obtained for the relationship between the look-ahead distance and the speed:

$$l_d = \frac{1}{6}v_c^2 + \frac{1}{5}v_c + 5 \tag{19}$$

where v_c is the speed of motion of the car.

In (19), only the vehicle’s speed influence on the look-ahead distance is considered, neglecting continuous speed changes and road curvature effects. Therefore, reference [34] introduced the consideration of path curvature and proposed a new look-ahead distance calculation method:

$$l_d = k_1v + k_2\Omega + l_{d0} \tag{20}$$

where k_1 is the speed coefficient with positive value, k_2 is the road curvature coefficient with negative value, Ω is the road curvature, and l_{d0} is the initial value of forward-looking distance.

To determine the optimal values, it is necessary to adjust the parameters k_1 and k_2 . Equation (20) is usually applied to path tracking under constant speed conditions and may not be suitable for scenarios involving continuous speed changes. Therefore, to ensure stability during mobile robot navigation on curved roads and to maintain consistent speed changes, we propose a new A-PP control rooted in quadratic polynomials. When the mobile robot deviates from the desired path in terms of lateral position, the forward-looking distance is promptly adjusted and optimized to minimize lateral position errors. This enhances the effectiveness of path-tracking control and enables adaptive control, as outlined below:

$$l_d = k_1v^2 + k_2\varphi + k_3e_r + l_{d0} \tag{21}$$

where v is the linear velocity of the FWDDR and φ is the governing factor, k_1 can be expressed as:

$$k_1 = \frac{1}{2a_{\max}} \tag{22}$$

Taking into account the robot’s nonholonomic constraints, the following parameters are relevant: a_{max} represents the maximum acceleration of the FWDDR, k_2 is a real-time path curvature adjustment factor, k_3 is a real-time lateral error adjustment factor, and l_{d0} is the initial forward-looking distance. Optimal values for k_2 and k_3 need to be determined through meticulous debugging. The flowchart of the A-PP algorithm is shown in Figure 5, and the specific steps are as follows:

Step 1: Retrieve reference path information, including horizontal and vertical coordinates (X_{ref}, Y_{ref}) and heading angle φ_{ref} .

Step 2: Calculate the path curvature and lateral error at each position, and use them as inputs in the forward-looking distance calculation formula. This allows for real-time determination of the appropriate forward-looking distance at the current position.

Step 3: Analyze the reference path by taking into account the look-ahead distance and FWDDR’s current position to identify the optimal preview point.

Step 4: Calculate the turning radius of the FWDDR using the PP algorithm’s principles. Convert the coordinates of the lookahead point to the robot’s coordinate system. This is achieved by translating the lookahead point’s coordinates to a system with the vehicle’s coordinates as the origin and then rotating this translated system by the heading angle of the differential drive robot to establish the final FWDDR coordinate system.

Step 5: Determine the FWDDR’s current position using its linear and angular velocities. Assess if it is the path’s endpoint. If so, conclude the program; otherwise, recalculate the preview distance.

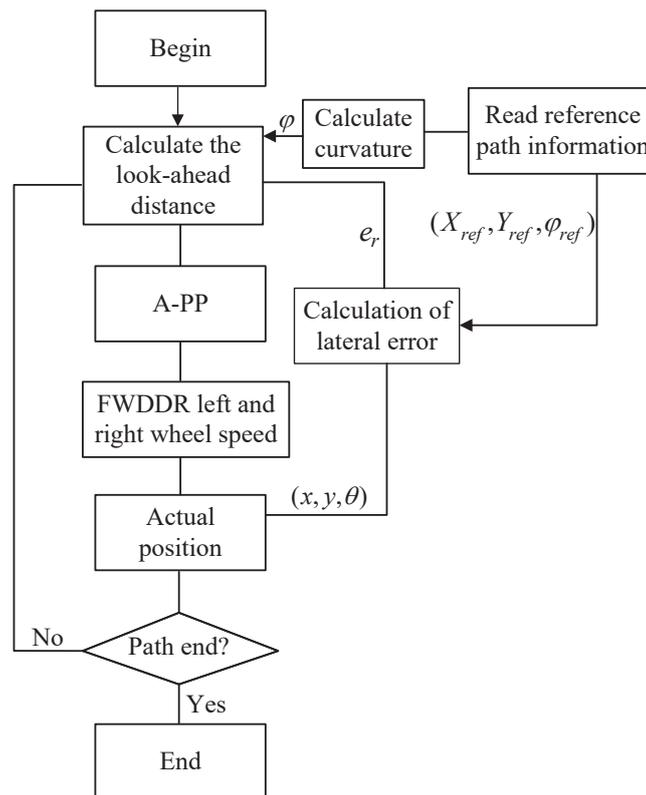


Figure 5. A-PP algorithm path-tracking flowchart.

In the lateral control of the A-PP algorithm, the error is multiplied with the adjustment factor and fed back to (21). When the error increases or decreases, the forward-looking distance is adjusted quickly and adaptively to eliminate the error as soon as possible and improve the response speed. In the longitudinal control, the reference path curvature and the adjustment factor are multiplied and fed back to (21) to adaptively adjust the forward-looking distance online, especially in the turning road condition.

3.3. Road Curvature Calculation Method

Given three consecutive discrete points (A, B, C), the parametric curve equation of three adjacent discrete points is as follows:

$$\begin{cases} x(t) = a_0 + a_1t + a_2t^2 \\ y(t) = b_0 + b_1t + b_2t^2 \end{cases} \tag{23}$$

where a_0, a_1, a_2 and b_0, b_1, b_2 are the undetermined coefficients.

The three-point parametric equation diagram is shown in Figure 6. Additionally, the curvature calculation formula for a parameter curve is as follows:

$$k = \frac{x''y' - x'y''}{(x'^2 + y'^2)^{3/2}} = \frac{a_2b_1 - a_1b_2}{(a_1^2 + b_1^2)^{3/2}} \tag{24}$$

Obviously, to accurately calculate the curvature of the parameter curve in (24), it is essential to first determine the six undetermined coefficients in Equation (23). The lengths of the two vectors derived from the three discrete points can be denoted as:

$$\begin{cases} t_a = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \\ t_b = \sqrt{(x_3 - x_2)^2 + (y_3 - y_2)^2} \end{cases} \tag{25}$$

where t_a is the distance between discrete points A and B, and t_b is the distance between discrete points B and C.

Given the proximity of points A, B, and C, the central angle of the corresponding curvature radius remains relatively consistent. Consequently, we can approximate the arc AB length to the straight-line distance between A and B. For Equation (23), this implies that the variation in the independent variable t adheres to the approximation:

$$\begin{cases} (x, y)|_{t=-t_a} = (x_1, y_1) \\ (x, y)|_{t=0} = (x_2, y_2) \\ (x, y)|_{t=t_b} = (x_3, y_3) \end{cases} \tag{26}$$

By substituting (26) into (25), the following can be obtained:

$$\begin{cases} x_1 = a_0 - a_1t_a + a_2t_a^2 \\ x_2 = a_0 \\ x_3 = a_0 + a_1t_b + a_2t_b^2 \end{cases}, \begin{cases} y_1 = b_0 - b_1t_a + b_2t_a^2 \\ y_2 = b_0 \\ y_3 = b_0 + b_1t_b + b_2t_b^2 \end{cases} \tag{27}$$

The preceding formula can be reformulated in matrix representation:

$$\begin{cases} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 & -t_a & t_a^2 \\ 1 & 0 & 0 \\ 1 & t_b & t_b^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} \\ \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 1 & -t_a & t_a^2 \\ 1 & 0 & 0 \\ 1 & t_b & t_b^2 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \end{bmatrix} \end{cases} \tag{28}$$

Note:

$$\begin{aligned}
 X &= \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, Y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} \\
 A &= \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix}, B = \begin{bmatrix} b_0 \\ b_1 \\ b_2 \end{bmatrix} \\
 M &= \begin{bmatrix} 1 & -t_a & t_a^2 \\ 1 & 0 & 0 \\ 1 & t_b & t_b^2 \end{bmatrix}
 \end{aligned} \tag{29}$$

Subsequently, the matrix of undetermined coefficients can be written as:

$$\begin{cases} A = M^{-1}X \\ B = M^{-1}Y \end{cases} \tag{30}$$

where M^{-1} is the inverse matrix of M .

Finally, by substituting (30) into (24), the curvature can be obtained.

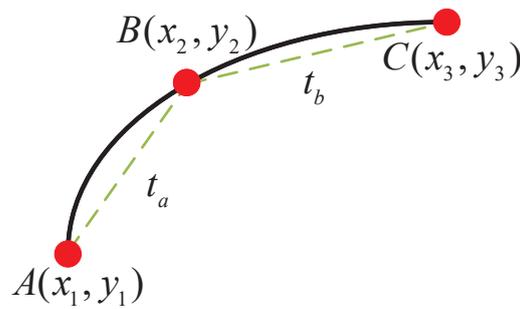


Figure 6. Three-point parametric equation.

4. Path Planning and Tracking

In this design, the stability of the mobile robot when tracking both straight and curved paths will be verified. The coordinates of the starting and ending points are (0.8, 0.2) and (0.2, 0.6), respectively. The reference path is shown in the blue curve in Figure 7. In this comparative simulation, the feasibility of the A-PP algorithm is assessed. The mobile robot’s speed is fixed at 0.2 m/s, whereas the look-ahead distance varies (0.1 m in yellow, 0.2 m in green, and 0.3 m in red) for comparison with the proposed A-PP algorithm. At the same time, compared with the current mainstream MPC algorithm, the adaptability of A-PP algorithm is evaluated.

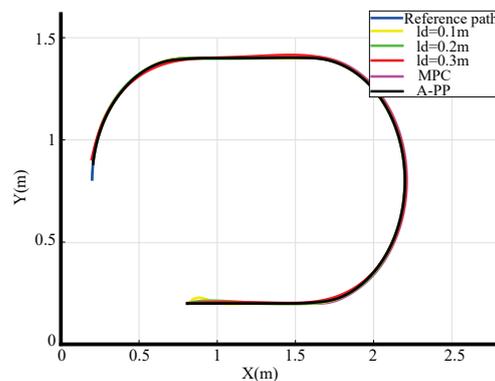


Figure 7. Simulation path-tracking effect.

Using the three-point parametric equation method (as per (24)), we derive the road curvature depicted in Figure 8. It can be seen that the curved part of the reference path exhibits a maximum curvature value of 1.66679. Conversely, the straight part holds a minimum curvature value close to 0.

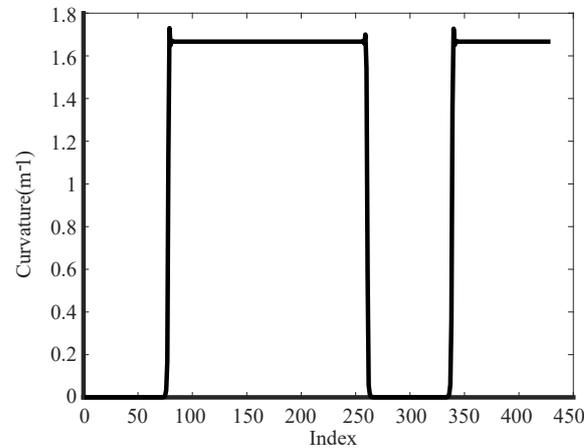


Figure 8. Simulation road curvature.

The simulation tracking results are shown in Figure 7. With a look-ahead distance of 0.1 m (yellow), significant control oscillations are observed, especially during the initial phase, resulting in considerable tracking errors and poor alignment with the reference path. In contrast, with the A-PP algorithm, the robot starts without pronounced oscillations and maintains stability throughout. The simulation of partial curve enlargement is shown in Figure 9. It is evident that a significant curvature change occurs when entering the curve. At this time, the A-PP algorithm can rapidly make necessary adjustments, enabling it to closely follow the desired path and enhancing the stability and reliability of the mobile robot. However, with forward-looking distances of 0.2 m (green) and 0.3 m (red), significant lateral errors occur on curved roads. This inconsistency arises because traditional algorithms use a fixed forward-looking distance that remains unaltered, irrespective of road conditions. MPC has improved stability compared to the traditional PP algorithm, but it is not as effective as the A-PP algorithm in terms of tracking effectiveness, especially in road conditions with large curvature changes. In Figure 10, it is evident that as the mobile robot travels in a straight line, the curvature decreases, whereas the forward distance increases. Conversely, as the robot reaches a turning position, the curvature increases, and the forward distance decreases. Forward distance adjusts according to the reference path's curvature, ranging between 0.08 m and 0.2 m.

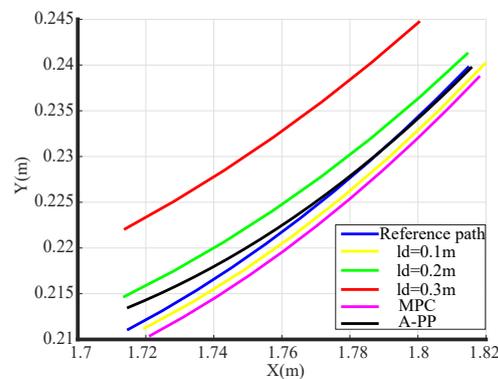


Figure 9. Simulation of partial curve enlargement.

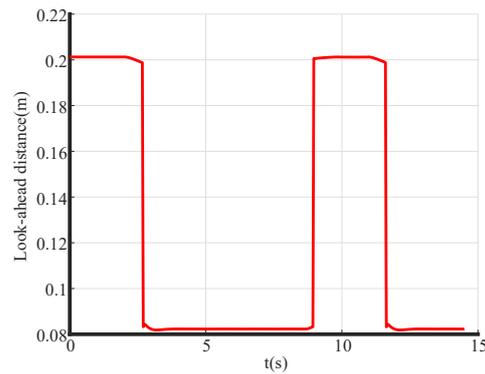


Figure 10. Simulation forward view distance.

The simulation’s lateral error diagram is shown in Figure 11. It is evident that, compared to traditional PP algorithms, the A-PP algorithm exhibits negligible lateral errors on straight paths. Upon entering a curved road, lateral errors increase; however, the A-PP algorithm promptly adjusts to maintain the mobile robot’s stable operation with minimal error. The MPC algorithm has a similar tracking effect compared to A-PP in straight-line road conditions, but when encountering road conditions with large changes in curvature, the tracking error of MPC increases, whereas the A-PP has a smaller tracking error and is able to respond quickly according to the road conditions. As illustrated in Table 1, while tracking the desired path, the A-PP algorithm achieves an average lateral error of 0.00694 m, a variance of 0.004663 m, and a maximum lateral error of 0.012837 m. The stability and accuracy are notably improved when compared to the traditional PP and MPC algorithms.

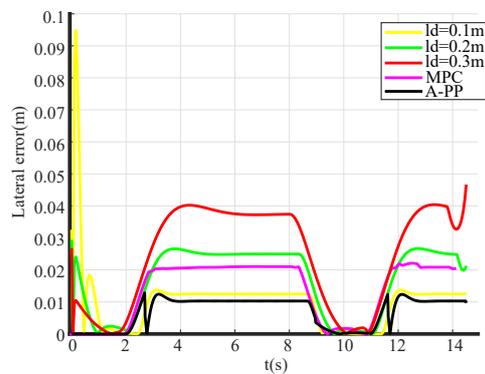


Figure 11. Simulation lateral error of FWDDR.

Table 1. Simulation error comparison table.

Method	Average Value of Lateral Error (m)	Variance (m)	Maximum Lateral Error (m)
$L_d = 0.1 \text{ m}$	0.01007	0.010075	0.094842
$L_d = 0.2 \text{ m}$	0.01688	0.010333	0.029417
$L_d = 0.3 \text{ m}$	0.02462	0.015404	0.046663
MPC	0.01467	0.010247	0.021854
A-PP	0.00694	0.004663	0.012837

5. Experiment

To further validate the practicality of the A-PP algorithm, experimental verification is conducted using a physical FWDDR as the platform.

5.1. Experimental Setup

The overall design specifications of the FWDDR are shown in Table 2:

Table 2. Overall design specifications.

Parameters	Value	Units
Length	770	mm
Width	658	mm
Wheelbase	470	mm
Wheel tread	573	mm
Minimum turning	1015	mm
Tire size (diameter)	260	mm
Maximum motor speed	3600	rpm
Maximum driving torque	47.5	N·m
Maximum load	≥50	kg

The following describes the workflow of the four-wheel differential electric drive system:

- (1) Control commands are sent through the PC Matlab/Simulink control interface via the RS485 bus;
- (2) On the PC side, E32-DTU converts commands transmitted via the RS485 bus into LoRa RF signals;
- (3) On the FWDDR side, E32-DTU converts LoRa signals into RS485 signals;
- (4) The motion controller interprets RS485 signals based on the four-wheel differential model and generates corresponding servo drive commands. These commands are then transmitted to the four servo drives through the CAN bus;
- (5) The motion controller transmits motor status, battery voltage, and other parameters back to the upper computer through a reverse path.

5.2. Experiment Process

The experimental site is shown in Figure 12, where location is obtained through a UWB device based on the DW1000 protocol. The maximum distance measurement range is 150 m, with positioning accuracy reaching the centimeter level. A rectangular positioning area is established using four base stations, with the reference trajectory positioned within this area. One of the base stations is connected to the upper computer via a cable. Furthermore, a UWB tag is positioned at the rear wheel's center on the FWDDR. As the FWDDR moves, the tag continuously updates its coordinate position to record the FWDDR's motion trajectory.

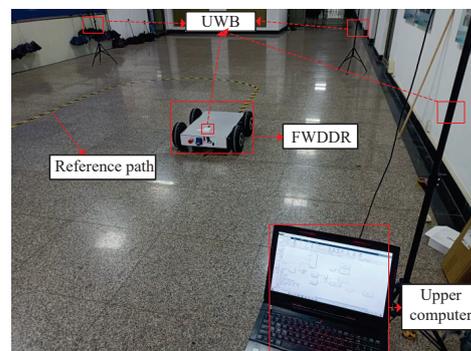


Figure 12. Experimental site.

In this experiment, the A-PP algorithm was compared and analyzed with traditional PP and MPC algorithms to validate A-PP's superiority. We set the speed at 0.1 m/s and

used a forward-looking distance of 0.2 m for the experiment. After multiple rounds of parameter-tuning tests, the tracking effect is best when $k_1 = \frac{1}{4}$, $k_2 = -0.07$, $k_3 = -0.2$.

5.3. Analysis of Experimental Results

The experimental tracking results, along with an enlarged view of a partial bend in the experiment, are depicted in Figures 13 and 14. An observation from the figures reveals that the traditional PP and MPC algorithms track the desired path well under straight road conditions. However, it exhibits significant tracking errors when encountering curved-road conditions, failing to meet the design requirements. In contrast, the A-PP algorithm achieves adaptive adjustments by controlling in both the lateral and longitudinal directions. This enables it to better track both straight-line and curved-road conditions to meet the design requirements. As evident from the experimental lateral error in Figure 15, the traditional PP algorithm exhibits slight oscillations at startup, whereas the A-PP algorithm operates stably with high tracking accuracy on straight sections. When the road surface curvature undergoes significant changes upon entering a curve, A-PP can swiftly make necessary adjustments. In contrast, the traditional pp and MPC algorithms have large errors on the curve and cannot achieve accurate tracking. The experiment error comparison table is presented in Table 3. When the A-PP algorithm tracks the desired path, the average lateral error measures 0.01070 m, with a variance of 0.006663 m and a maximum lateral error of 0.019443 m. These values represent a significant enhancement in stability and accuracy compared to the traditional PP and MPC algorithms.

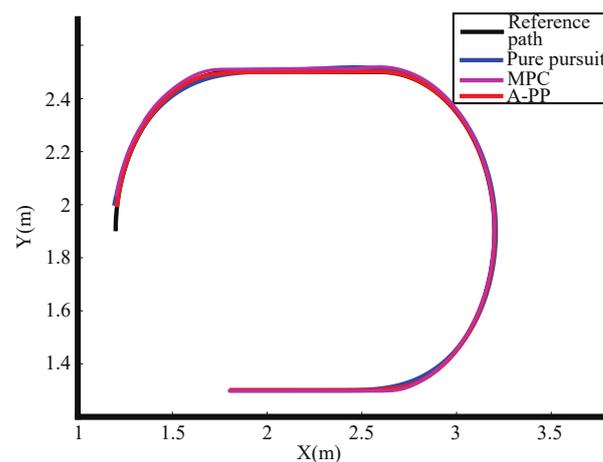


Figure 13. Experimental tracking effect.

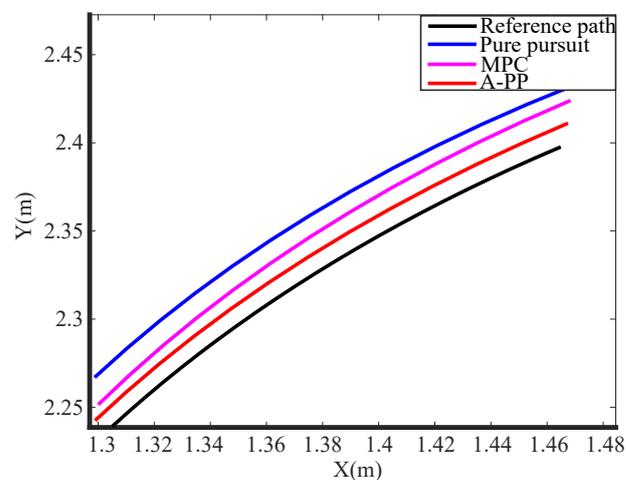


Figure 14. Enlarged view of the experimental partial bend.

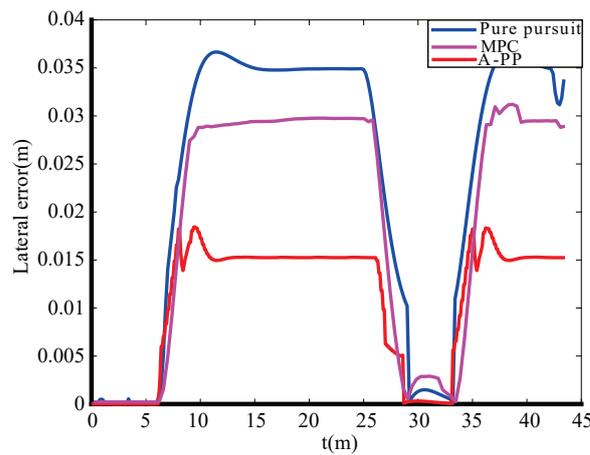


Figure 15. Experimental lateral error.

Table 3. Experiment error comparison table.

Method	Average Value of Lateral Error (m)	Variance (m)	Maximum Lateral Error (m)
Pure pursuit	0.02372	0.016588	0.036705
MPC	0.02069	0.014416	0.031628
A-PP	0.01070	0.006663	0.019443

6. Conclusions

In the field of mobile robot path tracking, achieving a precise trajectory following is of utmost importance. However, traditional methods often lack real-time adjustment capability for the forward-looking distance. In this study, we proposed and verified the A-PP algorithm, which incorporates a quadratic polynomial in both lateral and longitudinal dimensions. The main findings of the paper are as follows:

- (1) The quadratic polynomial is enhanced in both lateral and longitudinal dimensions to facilitate adaptive dynamic adjustment of the forward-looking distance. This enhancement reduces the lateral deviation of the FWDDR during path tracking and enhances both tracking accuracy and operational stability.
- (2) The A-PP algorithm is simulated and verified by Matlab/Simulink, and the results indicate that the A-PP algorithm achieves a mean lateral error of 0.00694 m, a variance of 0.004663 m, and a maximum lateral error of 0.012837 m during path tracking, which represent a significant enhancement in stability and accuracy when compared to the traditional PP and MPC algorithms.
- (3) Experimental tests further validated the A-PP algorithm. The results showed a mean lateral error of 0.01070 m, a variance of 0.006663 m, and a maximum lateral error of 0.019443 m. In comparison with the PP algorithm, the A-PP algorithm achieved faster convergence of deviations, enhanced lateral errors in turning sections, and heightened driving smoothness, all while maintaining low computational time.

This research shows that the A-PP algorithm has the advantages of adaptive control, accurate tracking, fast computation speed, and excellent robustness. However, there are some limitations, such as inaccurate tracking under high-speed motion and large localization deviation of UWB, etc. In future work, further improvement and optimization will be conducted to address these limitations. As road conditions become increasingly complex, the exploration and further refinement of such algorithms will be instrumental in ensuring safe and precise vehicle operations in the future.

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

Funding: This work was supported by the National Natural Science Foundation of China under Grant 62003312, 62073299, 62103076, 62273313; Science and Technology Development Project of Henan Province (232102221032); Young Backbone Teacher Training Program of Henan Province (2021GGJS089).

Institutional Review Board Statement: Not applicable.

Data Availability Statement: The data generated and/or analyzed as well as the source code used in the current study are not publicly available due to their use in an ongoing project, but may be available from the corresponding author on reasonable request.

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

References

1. Yan, Y.; Geng, K.; Liu, S.; Ren, Y. A New Path Tracking Algorithm for Four-Wheel Differential Steering Vehicle. In Proceedings of the 2019 Chinese Control Furthermore, Decision Conference (CCDC), Nanchang, China, 3–5 June 2019; pp. 1527–1532. [\[CrossRef\]](#)
2. Shojaei, K.; Shahri, A.M.; Tarakameh, A.; Tabibian, B. Adaptive trajectory tracking control of a differential drive wheeled mobile robot. *Robotica* **2011**, *29*, 391–402. [\[CrossRef\]](#)
3. Gao, H.; Song, X.; Ding, L.; Xia, K.; Li, N.; Deng, Z. Adaptive motion control of wheeled mobile robot with unknown slippage. *Int. J. Control* **2014**, *87*, 1513–1522. [\[CrossRef\]](#)
4. Deng, M.; Inoue, A.; Sekiguchi, K.; Jiang, L. Two-wheeled mobile robot motion control in dynamic environments. *Robot. Comput.-Integr. Manuf.* **2010**, *26*, 268–272. [\[CrossRef\]](#)
5. Coelho, P.; Nunes, U. Path-following control of mobile robots in presence of uncertainties. *IEEE Trans. Robot.* **2005**, *21*, 252–261. [\[CrossRef\]](#)
6. Samson, C.; Morin, P.; Lenain, R. Modeling and control of wheeled mobile robots. In *Springer Handbook of Robotics*; Springer: Berlin, Germany, 2016; pp. 1235–1266. [\[CrossRef\]](#)
7. Cui, M. Observer-based adaptive tracking control of wheeled mobile robots with unknown slipping parameters. *IEEE Access* **2019**, *7*, 169646–169655. [\[CrossRef\]](#)
8. Zhang, Z.; Zhang, B. Development and key technologies of dynamic wireless power transfer system for mobile load. *Electr. Power Eng. Technol.* **2020**, *39*, 21–30.
9. Xie, L.; Jiang, C.; Sun, Q.; Wang, H.; Song, Q.; Guan, G. The global map's creating and positioning of substation inspection robot based on adaptive Monte Carlo particle filter algorithm. *Electr. Power Eng. Technol.* **2019**, *38*, 16–23.
10. Dong, L.; Geng, Z.; Liu, H.; Li, H. High accuracy and high efficiency model of non-standard ratio transformer. *Electr. Power Eng. Technol.* **2021**, *40*, 60–65.
11. Moudoud, B.; Aissaoui, H.; Diany, M. Robust trajectory tracking control based on sliding mode of Differential Driving Four-Wheeled Mobile Robot. In Proceedings of the 2020 IEEE 6th International Conference on Optimization and Applications (ICOA), Beni Mellal, Morocco, 20–21 April 2020; pp. 1–5. [\[CrossRef\]](#)
12. Wang, S.; Yin, X.; Li, P.; Zhang, M.; Wang, X. Trajectory tracking control for mobile robots using reinforcement learning and PID. *Iran. J. Sci. Technol. Trans. Electr. Eng.* **2020**, *44*, 1059–1068. [\[CrossRef\]](#)
13. 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. [\[CrossRef\]](#)
14. Alouache, A.; Wu, Q. Genetic algorithms for trajectory tracking of mobile robot based on PID controller. In Proceedings of the 2018 IEEE 14th International Conference on Intelligent Computer Communication and Processing (ICCP), Cluj-Napoca, Romania, 6–8 September 2018; pp. 237–241. [\[CrossRef\]](#)
15. Jiang, X.; Motai, Y.; Zhu, X. Predictive fuzzy logic controller for trajectory tracking of a mobile robot. In Proceedings of the 2005 IEEE Midnight-Summer Workshop on Soft Computing in Industrial Applications, Espoo, Finland, 28–30 June 2005; pp. 29–32. [\[CrossRef\]](#)
16. Nguyen, T.V.; Huynh, N.T.; Vu, N.C.; Kieu, V.N.; Huang, S.C. Optimizing compliant gripper mechanism design by employing an effective bi-algorithm: Fuzzy logic and ANFIS. *Microsyst. Technol.* **2021**, *27*, 1–24. [\[CrossRef\]](#)
17. Huynh, N.T.; Nguyen, T.V.; Tam, N.T.; Nguyen, Q.M. Optimizing magnification ratio for the flexible hinge displacement amplifier mechanism design. In *Proceedings of the 2nd Annual International Conference on Material, Machines and Methods for Sustainable Development (MMMS2020)*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 769–778. [\[CrossRef\]](#)
18. Cheng, S.; Li, L.; Guo, H.Q.; Chen, Z.G.; Song, P. Longitudinal collision avoidance and lateral stability adaptive control system based on MPC of autonomous vehicles. *IEEE Trans. Intell. Transp. Syst.* **2019**, *21*, 2376–2385. [\[CrossRef\]](#)

19. Xu, C.; Li, M.; Pan, F. The system design and LQR control of a two-wheels self-balancing mobile robot. In Proceedings of the 2011 International Conference on Electrical and Control Engineering, Nanchang, China, 20–22 June 2011; pp. 2786–2789. [\[CrossRef\]](#)
20. Xu, H.; Zhu, J. Interval trajectory tracking for AGV based on MPC. In Proceedings of the 2019 Chinese Control Conference (CCC), Guangzhou, China, 27–30 July 2019; pp. 2835–2839. [\[CrossRef\]](#)
21. Ni, J.; Wang, Y.; Li, H.; Du, H. Path Tracking Motion Control Method Of Tracked Robot Based On Improved LQR Control. In Proceedings of the 2022 41st Chinese Control Conference (CCC), Hefei, China, 25–27 July 2022; pp. 2888–2893. [\[CrossRef\]](#)
22. Wang, W.J.; Hsu, T.M.; Wu, T.S. The improved pure pursuit algorithm for autonomous driving advanced system. In Proceedings of the 2017 IEEE 10th International Workshop on Computational Intelligence and Applications (IWCIA), Hiroshima, Japan, 11–12 November 2017; pp. 33–38. [\[CrossRef\]](#)
23. Sun, Q.; Wang, Z.; Li, M.; Liu, B.; Cheng, J.; Tai, J. Path tracking control of wheeled mobile robot based on improved pure pursuit algorithm. In Proceedings of the 2019 Chinese Automation Congress (CAC), Hangzhou, China, 22–24 November 2019; pp. 4239–4244. [\[CrossRef\]](#)
24. Amer, N.H.; Hudha, K.; Zamzuri, H.; Aparow, V.R.; Abidin, A.F.Z.; Abd Kadir, Z.; Murrad, M. Adaptive modified Stanley controller with fuzzy supervisory system for trajectory tracking of an autonomous armoured vehicle. *Robot. Auton. Syst.* **2018**, *105*, 94–111. [\[CrossRef\]](#)
25. Ding, H.; Liu, H.; Zhuang, Y.; Kan, M.; Xia, D.; Ding, S. Multi-robot path planning based on four-wheel differential speed model. *Control Eng. China* **2023**, *30*, 730–738. [\[CrossRef\]](#)
26. Zhang, K.; Yen, G.G.; He, Z. Evolutionary algorithm for knee-based multiple criteria decision making. *IEEE Trans. Cybern.* **2019**, *51*, 722–735. [\[CrossRef\]](#)
27. Hu, Y.; Chen, G.; Li, Z.; Knoll, A. Robot policy improvement with natural evolution strategies for stable nonlinear dynamical system. *IEEE Trans. Cybern.* **2022**, *2022*, 4002–4014. [\[CrossRef\]](#)
28. Kozłowski, K.; Pazderski, D. Modeling and control of a 4-wheel skid-steering mobile robot. *Int. J. Appl. Math. Comput. Sci.* **2004**, *14*, 477–496.
29. Li, P.; Wang, H.; Zhu, M.; Liu, J. Maneuver control of a four-wheel differentially driven robot based on instantaneous center of rotation. In Proceedings of the 2018 Chinese Control Furthermore, Decision Conference (CCDC), Shenyang, China, 9–11 June 2018; pp. 4917–4922. [\[CrossRef\]](#)
30. Liu, Y.; Jiang, Y. Motion planning of differential driven robot based on tracking. *Control Decis.* **2023**, *38*, 2529–2536. [\[CrossRef\]](#)
31. Coulter, R.C. *Implementation of the Pure Pursuit Path Tracking Algorithm*; Carnegie Mellon University, The Robotics Institute: Pittsburgh, PA, USA, 1992.
32. Wu, X.; Angeles, J.; Zou, T.; Sun, C.; Sun, Q.; Wang, L. Receding-horizon vision guidance with smooth trajectory blending in the field of view of mobile robots. *Appl. Sci.* **2020**, *10*, 676. [\[CrossRef\]](#)
33. Duan, J.; Yang, C.; Shi, H. Path tracking based on pure pursuit algorithm for intelligent vehicles. *J. Beijing Univ. Technol.* **2016**, *42*, 1301–1306.
34. Li, G.; Wang, Y.; Guo, L.; Tong, J. Improved pure pursuit algorithm for rice transplanter path tracking. *Nongye Jixie Xuebao/Trans. Chin. Soc. Agric. Mach.* **2018**, *49*, 21–26.

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.