

Article

Design and Experiments of Autonomous Path Tracking Based on Dead Reckoning

Songxiao Cao ^{1,*} , Ye Jin ², Toralf Trautmann ³ and Kang Liu ¹ ¹ College of Metrology and Measurement Engineering, China Jiliang University, Hangzhou 310018, China² School of Engineering and Design, Technical University of Munich, Garching bei München, 85748 Munich, Germany³ Mechlab, Dresden University of Applied Sciences, 01065 Dresden, Germany

* Correspondence: caosongxiao@cjlu.edu.cn

Abstract: Path tracking is an important component of autonomous driving and most current path tracking research is based on different positioning sensors, such as GPS, cameras, and LIDAR. However, in certain extreme cases (e.g., in tunnels or indoor parking lots), if these sensors are unavailable, achieving accurate path tracking remains a problem that is worthy of study. This paper addresses this problem by designing a dead reckoning method that is solely reliant on wheel speed for localization. Specifically, a differential drive model is first used for estimating the current relative vehicle position in real time by rear wheel speed, and the deviation between the current path and the reference path is then calculated using the pure pursuit algorithm as a means of obtaining the target steering wheel angle and vehicle speed. The steering wheel and vehicle speed signals are then output by two PID controllers in order to control the vehicle, and the automatic driving path tracking is ultimately realized. Through exhaustive tests and experiments, the stop position error and tracking process error are compared under different conditions, and the effects of vehicle speed, look-ahead distance, starting position angle, and driving mode on tracking accuracy are analyzed. The experimental results show the average error of the end position to be 0.26 m, 0.383 m, and 0.505 m when using BMW-i3 to drive one lap automatically at speeds of 5 km/h, 10 km/h, and 15 km/h in a test area with a perimeter of approximately 200 m.



Citation: Cao, S.; Jin, Y.; Trautmann, T.; Liu, K. Design and Experiments of Autonomous Path Tracking Based on Dead Reckoning. *Appl. Sci.* **2023**, *13*, 317. <https://doi.org/10.3390/app13010317>

Academic Editor: Seokwon Yeom

Received: 8 December 2022

Revised: 22 December 2022

Accepted: 26 December 2022

Published: 27 December 2022



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: autonomous driving; dead reckoning; differential drive kinematics; pure pursuit

1. Introduction

Autonomous driving is a topic of academic research and industry application that has been widely discussed for almost three decades, and it has now been used on a small scale in the manufacturing of vehicles such as Tesla [1,2]. A typical autonomous driving system consists of sensing, perception, decision, and control. Sensing and perception refer to sensing the environment around the vehicle using various sensors and understanding it as a means of helping the vehicle make better decisions. Deep learning and artificial intelligence methods are currently widely used in the field of sensing and perception for achieving target detection and recognition through deep learning of a variety of sensor data types. For example, Beltran et al. used LiDAR in 3D target detection [3], Wu et al. proposed UIU-Net for infrared object detection [4] using remote sensing images for the detection of weak targets [5,6], and the well-known YOLO [7] uses RGB cameras for pedestrian detection. At the same time, the purpose of decision-making and control is to ensure that the vehicle travels on an appropriate path at an appropriate speed based on the sensing results. The focus of this paper is on path tracking, which is part of the control module. The objective of this is the manipulation of the steering wheel and throttle pedal to guide the vehicle to follow the desired path [8].

The optimization control methods that are used in path tracking algorithms can be classified into five categories [9]: (1) Pure pursuit and the Stanley control method.

Wang et al. [10] proposed a novel pure pursuit algorithm based on optimized look-ahead distance as a means of improving tracking accuracy and Elbanhawi et al. [11] proposed a model predictive active yaw control implementation of pure pursuit path tracking for improving tracking performance at high speed. These controllers are based on the geometric concept, using the kinematics model and geometric relation to calculate target steering angles. They are simple and suitable in many low-vehicle-speed applications but may experience problems if the vehicle is involved in a situation with large path curvature. (2) The PID control method: PID control-based algorithms are widely used in autonomous driving. Han et al. [12] established a neural network PID controller for lateral path tracking, while Al-Mayyahi et al. [13] utilized a fractional order PID controller for achieving autonomous path tracking. This method can easily be used in engineering applications, but there is the disadvantage that the algorithms and parameters are application-dependent. (3) Model-free method: This approach is essentially data-driven, such as that proposed by Liu et al. [14] where a novel model-free adaptive control algorithm based on a dual successive projection method used at least five sensor types to generate sufficient data for the algorithm to learn the control parameters, which may present difficulties in situations with insufficient sensor data. (4) The Linear Quadratic Regulator (LQR) method: The LQR controller is a classic method that is used in path tracking. For example, in [15], LQR is adopted as a means of obtaining the optimal active front steering input. LQR controllers require high computing power due to being optimized online. (5) The Model Predictive Control (MPC) method: This [16] has the ability to predict future road curvature and act accordingly. Many researchers have proposed different enhanced MPC controllers for improving performance. Tan et al. [17] combined MPC with particle swarm optimization (PSO) in order to achieve path tracking using four-wheel steering and four-wheel drive (4WS4WD). Tang et al. [18] used a kinematic MPC for dealing with disturbances in road curvature and a vehicle sideslip angle compensator for correcting kinematic model prediction. Although MPC has exhibited good performance in many applications, heavy computation remains a challenge with real-time applications, particularly nonlinear MPC, and this controller type may face optimization failure if the initial values are improper [19].

In an autonomous driving scenario, a vehicle must know where it is currently located and how it is moving in the environment. Autonomous driving uses several localization techniques, such as a global positioning system (GPS)/inertial measurement unit (IMU)-based technique, a camera-based technique, a radar-based technique, a Lidar-based technique, and an ultrasonic-based technique [20]. For example, the work of Mechlab used LiDAR for the detection of lane marks and localization in autonomous driving [21]. Although the majority of these techniques are quite successful in providing accurate positioning, a large amount of data must be processed in real time, which is computationally demanding, and in some cases, sensors may be unable to provide localization information, such as GPS in tunnels or an indoor environment. In addition, in extreme cases when severe occlusions occur and the majority of localization sensor systems fail to work, the vehicle must be able to accurately assess its dynamical motion. Therefore, dead reckoning has become an indispensable technology in autonomous driving, as it can provide the vehicle with a certain degree of accurate positioning and navigation where other sensors cannot. Carlson et al. [22] analyzed the error sources in land vehicle dead reckoning with differential wheel speeds, finding the limiting factors for dead reckoning accuracy during ordinary driving to be primarily unmodeled roll dynamics and surface unevenness. Welte et al. [23] proposed a method able to accurately calibrate the parameters without knowing the ground truth using Rauch–Tung–Striebel smoothing to create an accurate dead reckoning system. Freydin et al. [24] proposed a trained deep neural network for learning car speeds using inertial sensors for dead reckoning navigation. Brossard et al. [25] proposed a novel approach to inertial-only dead reckoning for wheeled vehicles based on deep neural networks for the dynamic adaptation of the parameters of a Kalman filter.

In response to the use of dead reckoning and simple control methods for achieving accurate trajectory tracking, two problems remain with methods that have been proposed in the current literature:

- (1) Among all the dead reckoning methods, the differential drive model is a more common localization method, but it is generally applied to mobile robots, and relatively few papers have applied it and tested it in vehicular situations. For example, Wu et al. [26] used the differential drive model combined with Lidar for indoor trajectory tracking for mobile robots, while Martins et al. [27] proposed a velocity-based dynamic model for differential drive mobile robots, again for indoor robots, and Yasmine et al. [28] also proposed a similar method. The common feature shared by these papers is that the mobile robots are small with only one wheel in the front guide wheel, meaning that their experimental results do not reflect the performance of differential drive methods in real vehicles. Therefore, applying and evaluating this method for its performance in autonomous vehicles is necessary.
- (2) In the available pure pursuit path tracking literature, most studies have used localization sensors such as GPS, demonstrating unsatisfactory tracking accuracy. For example, Wang et al. [29] used a modified pure pursuit for achieving unmanned driving, but the tracking error reached 1.2 m, while Ohta et al. [30] used a pre-built map and LiDAR for localization and pure pursuit for trajectory tracking, but the tracking trajectory still showed a large deviation from the set trajectory in the results. Therefore, whether the adoption of pure pursuit autonomous path tracking without localization sensors can achieve better tracking accuracy than these studies is an issue that is worthy of further study.

This work aims to achieve autonomous path tracking without the use of additional localization sensors, relying solely on the dead reckoning method, which is used for coping with extreme cases where all sensors fail while still allowing effective autonomous driving. To achieve real-time tracking, the simple yet effective pure pursuit control method is adopted in this paper. This paper systematically analyzes several key parameters affecting tracking accuracy in the model based on differential drive kinematics and pure pursuit, including the look-ahead distance, initial position, and driving mode. Furthermore, an RGB camera is used for the accurate calculation of tracking errors at each moment.

The contributions of this paper are as follows:

- By combining two simple but effective methods, pure pursuit and differential drive, it is experimentally demonstrated for the first time on the BMW-I3 that high-precision autonomous path tracking is possible without the aid of positioning sensors. By using this method, autonomous path tracking accuracy can be improved in certain extreme conditions (e.g., inside tunnels or indoors).
- By combining the image processing method, the distance between the vehicle and the lane lines on one side can be used as the tracking accuracy evaluation criterion for each moment in path tracking, which makes it more objective and accurate. With this method, the actual field test of vehicle path tracking is more repeatable and convincing.

This paper is organized as follows. The next section provides an explanation of the details of autonomous driving based on dead reckoning, including differential drive kinematics, pure pursuit control, and error measurement based on image processing. In Section 3, the parameter settings, experimental results, and analysis are provided, and in Section 4, the conclusions and directions of future research are outlined.

2. Autonomous Driving Based on Dead Reckoning

The autonomous path tracking system used in this paper consists of a differential drive kinematics module, a pure pursuit module, a PID control module, and two actuators that control the steering and throttle pedal, as shown in Figure 1. The differential drive kinematics model calculates the current position in relation to the original position based on rear wheel speed, which can be read from the CAN bus. The pure pursuit model then

outputs the target steering angle and target vehicle speed based on the error between the target position in the predefined path and the current position. The target steering angle and vehicle speed are then sent to the PID controller to generate a steering control command and throttle pedal control command as a means of driving the vehicle automatically.

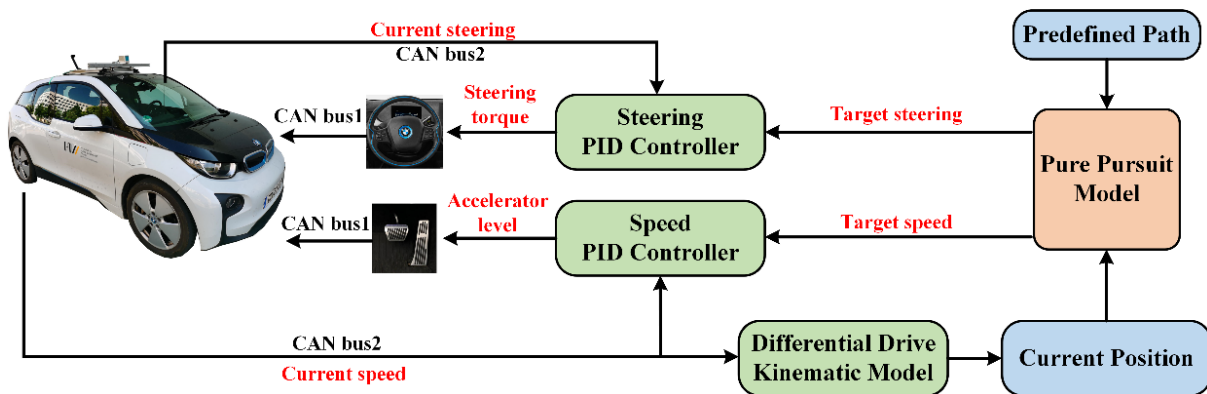


Figure 1. Overview of the path tracking system.

2.1. Dead Reckoning Based on Differential Drive Kinematics

Without positioning sensors, differential drive kinematics can generally be used for calculating the relative position of the vehicle in real time. The basic principle of differential drive kinematics involves using the speed of the left and right wheels and the difference between them for calculating the trajectory of the vehicle.

There are two basic localization methods that are based on wheel speed: Rear wheel position-based feedback and front wheel position-based feedback. According to research [1], the control stability of rear wheel position-based feedback control law is unaffected by the sign of wheel velocity, so it is much more suitable for use in dead reckoning.

In this model, the speed of the rear wheels is used for calculating the real-time position of the vehicle, which is demonstrated schematically in Figure 2.

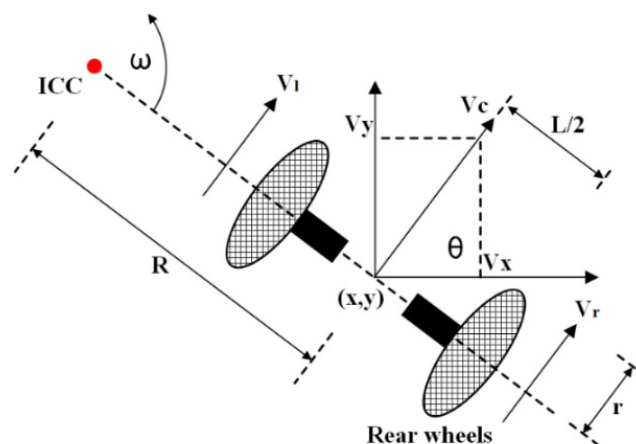


Figure 2. Differential drive kinematics.

In Figure 2, ICC represents the instantaneous center of curvature, meaning that when the velocity of each wheel changes, the vehicle rotates about this point. (x,y) is the coordinate of the midpoint between both wheels in the world coordinate system. R represents the radius of curvature, r is the wheel radius, L is the distance between the two wheels, V_c is vehicle velocity, V_l is left wheel velocity, and V_r is right wheel velocity. θ is the angle of the forward direction of the vehicle in the world coordinate system, and ω is the rate of

rotation, which is identical for both wheels. Based on the above model and assumptions, the following equations are obtained:

$$\omega \left(R + \frac{L}{2} \right) = Vr \tag{1}$$

$$\omega \left(R - \frac{L}{2} \right) = Vl \tag{2}$$

Based on the above equations, it can be solved for ω and R at any moment:

$$\omega = \frac{Vr - Vl}{L} \tag{3}$$

$$R = \frac{L}{2} \frac{Vr + Vl}{Vr - Vl} \tag{4}$$

If the rotational speeds of the left and right wheels are known, the horizontal and vertical velocities of the vehicle in the coordinate system at a given moment can be obtained:

$$v_x = v_c \cos(\theta) = R\omega \cos(\theta) = \frac{L}{2} \frac{Vr + Vl}{Vr - Vl} \times \frac{Vr - Vl}{L} \cos(\theta) = \frac{Vr + Vl}{2} \cos(\theta) \tag{5}$$

$$v_y = v_c \sin(\theta) = R\omega \sin(\theta) = \frac{L}{2} \frac{Vr + Vl}{Vr - Vl} \times \frac{Vr - Vl}{L} \sin(\theta) = \frac{Vr + Vl}{2} \sin(\theta) \tag{6}$$

Therefore, the position and angle of the vehicle at moment t can be calculated using the following equation:

$$x = \int_0^t v_x dt + x_0 \tag{7}$$

$$y = \int_0^t v_y dt + y_0 \tag{8}$$

$$\theta = \int_0^t \omega dt + \theta_0 \tag{9}$$

As the wheel radius is known and the rotation speed of both wheels can be obtained through the CAN bus, the speed of both wheels at any given moment can be calculated. Therefore, the final vehicle position and angle in real time relate only to the rotation speed of both wheels.

2.2. Pure Pursuit Control with Bicycle Model

Based on the rear wheel position calculated using the above differential drive model, the vehicle can be further simplified as a bicycle model, as shown in Figure 3.

$$\tan(\delta) = \frac{B}{R} \tag{10}$$

$$v_c = R\omega \tag{11}$$

$$\dot{\theta} = \omega = \frac{v_c \tan(\delta)}{B} \tag{12}$$

With the vehicle's real-time position based on the above bicycle model, the pure pursuit model can be used for tracking a given reference path. The pure pursuit establishes a geometric model between the moving vehicle and the reference path, as can be seen in Figure 4. The reference path (green) is a predefined path upon which all the points in the path are known, and the vehicle trajectory (orange) is the predicted trajectory based on the current steering angle and speed. The objective of the pure pursuit is calculating the target steering angle and vehicle speed in real time based on the geometric relationship.

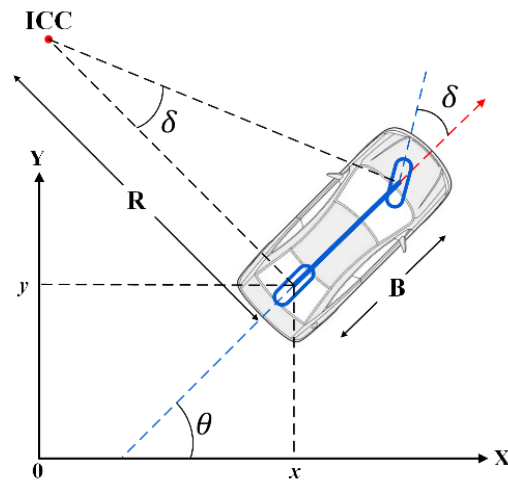


Figure 3. Bicycle model.

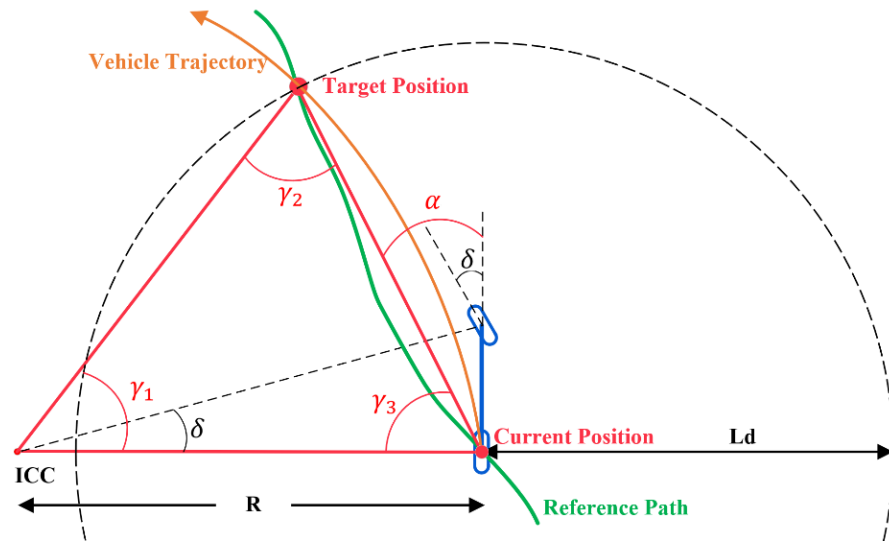


Figure 4. Pure pursuit model.

From Figure 4, the following formula is obtained:

$$180^\circ = \gamma_1 + \gamma_2 + \gamma_3 = \gamma_1 + (90^\circ - \alpha) + (90^\circ - \alpha) \tag{13}$$

which yields $\gamma_1 = 2\alpha$, according to the law of sines:

$$\frac{l_d}{\sin(\gamma_1)} = \frac{R}{\sin(\gamma_2)} \tag{14}$$

Based on the result of Equation (14), the following formulas are obtained step by step:

$$\frac{l_d}{\sin(2\alpha)} = \frac{R}{\sin(90^\circ - \alpha)} \tag{15}$$

$$\frac{l_d}{2 \sin(\alpha) \cos(\alpha)} = \frac{R}{\cos(\alpha)} \tag{16}$$

$$R = \frac{l_d}{2 \sin(\alpha)} \tag{17}$$

$$\delta = \arctan\left(\frac{2B\sin(\alpha)}{l_d}\right) \quad (18)$$

The angle is the target steering angle, which is required in the PID controller.

2.3. Error Measurement Based on Image Processing

To accurately measure the tracking accuracy of each position during the trajectory tracking process, a GoPro recording camera was installed on the side of the vehicle for shooting the lane line on the left side of the vehicle and calculating the distance between the vehicle and the lane line based on the lane line position in the video, as shown in Figure 5. The actual distance is compared with the reference trajectory distance as a means of calculating the tracking error for each position.

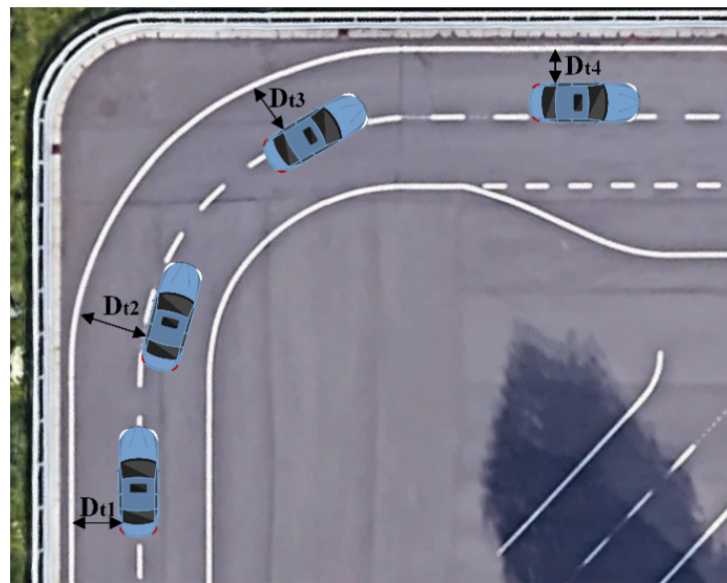


Figure 5. Error measurement based on the distance between the vehicle and side lane.

According to the differential drive model that was introduced earlier, the vehicle position is calculated based on the position of the center of the rear wheel, so the GoPro was installed directly above this position.

2.3.1. Distance Calibration

As the measurement target is the distance between the center of the rear wheel and the lane line, only calculating the longitudinal height of the detected lane line in the image is necessary. For the accurate calculation of the actual distance that corresponds to each vertical height in the image, the GoPro must be calibrated for distance.

The distance calibration method can be seen in Figure 6. Firstly, the scale marks representing distance were drawn on the ground, and each scale line was spaced 20 cm apart. The scale line was at the horizontal center of the GoPro image. Secondly, the vertical coordinates corresponding to each scale line in the image were found manually. Finally, the distance between each of the two scale lines was linearly interpolated, and the actual distance corresponding to each vertical pixel coordinate in the image was then calibrated.

2.3.2. Image Processing and Lane Detection

Following each test, the distance between the vehicle and the lane line in each frame was calculated by analyzing the saved video, frame by frame. In this paper, a simple and effective image processing algorithm was used for the automatic detection of the lane lines, and the distance was then calculated. The lane line detection and calculation method can be seen in Figure 7.

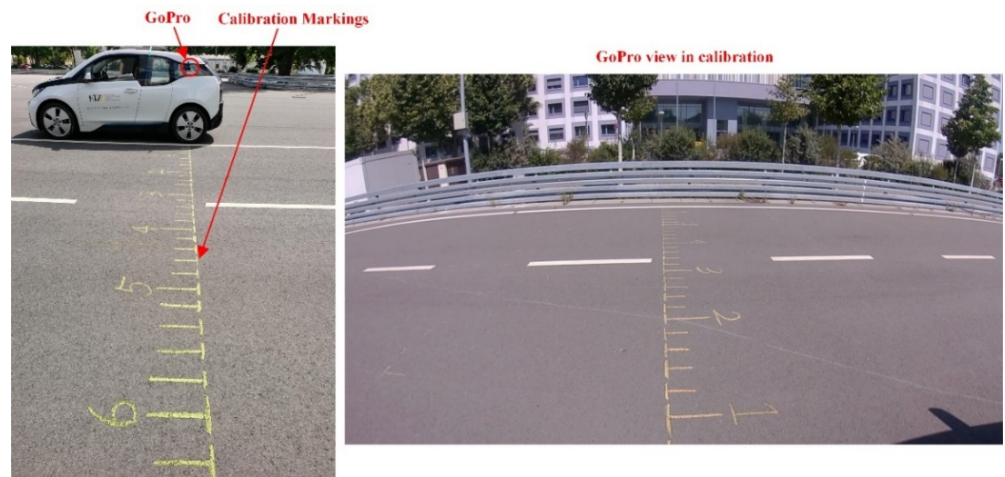


Figure 6. GoPro distance calibration.

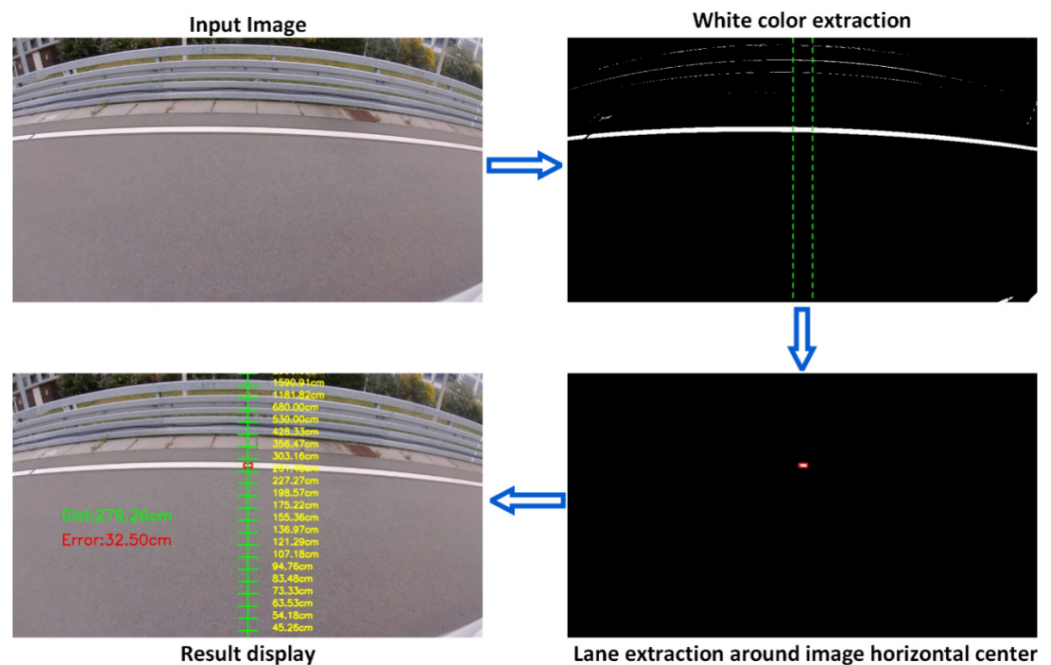


Figure 7. Lane line detection and calculation.

The image was first converted to HSV color space for each input frame before the white area was segmented out into HSV space using the color extraction method. Morphological operations such as erosion and dilation were then performed on the segmentation results as a means of filtering out the noise, and candidate targets were ultimately extracted at intervals of 20 pixels on the left and right with the horizontal center of the image as the central axis before that with the largest area was chosen as the final lane line position.

3. Experiments and Results

All experiments in this paper were conducted at the test field of HTW Dresden (Dresden University of Applied Sciences) using a modified BMW-i3 that allowed the control of its steering wheel and throttle through the electronic control unit (ECU) and CAN bus. The control algorithms in the experiments are MATLAB Simulink programs that were run on a laptop, as can be seen in Figure 8.

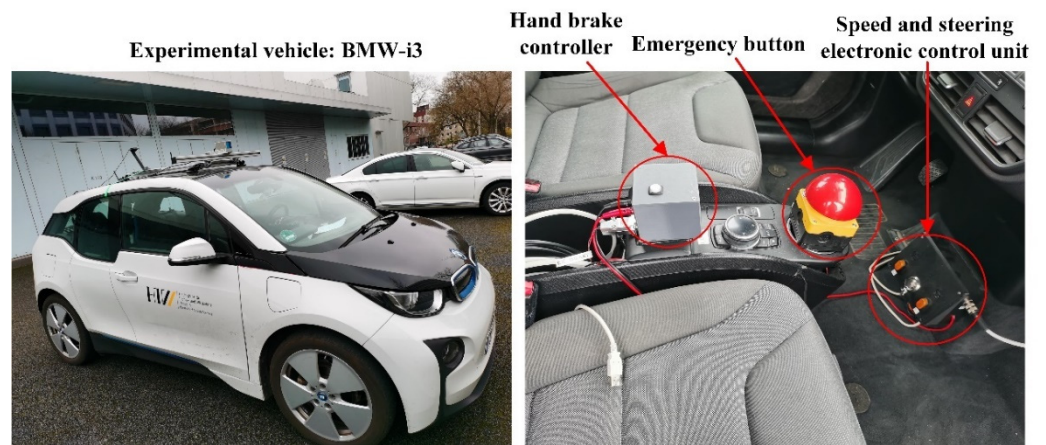


Figure 8. Vehicle used in the experiments.

The demonstration video of the experiment is available with the link provided in the “Supplementary Materials” part of the paper.

3.1. Steering and Speed Control PID Parameter Selection

Steering and speed control play essential roles in autonomous driving, and good trajectory tracking can only be achieved if the steering and speed of the vehicle can be accurately controlled. The entire vehicle is a complex control system and obtaining the steering wheel and speed control parameters by purely theoretical analysis is incredibly difficult, so this paper used repeated practical tests to select the appropriate PID control parameters.

Figure 9 shows the control performance of different PID parameters when the steering wheel and speed (5 km/h) are given a step signal and from the experimental curve, it can be seen that the PID parameters represented by the light blue curve are better regardless of the steering wheel or speed. Following repeated tests, a set of steering wheel PID parameters was obtained that we believe to be optimal, and three sets of speed PID parameters at 5 km/h, 10 km/h, and 15 km/h were also obtained, as can be seen in Table 1.

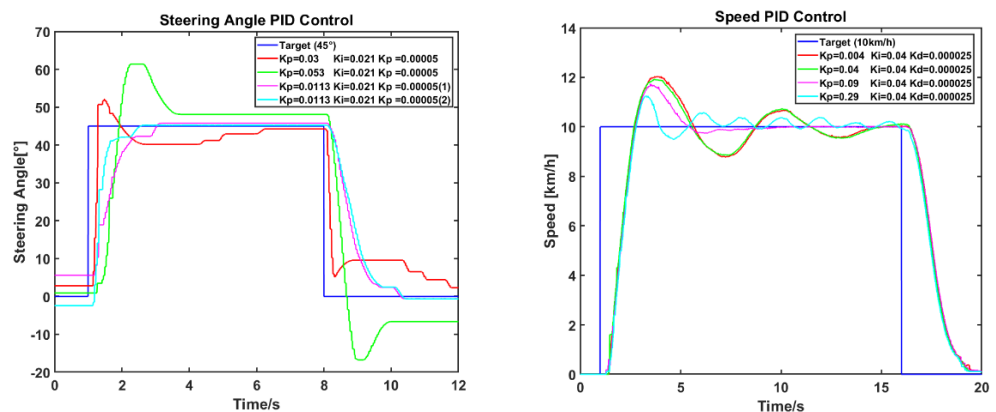


Figure 9. PID control experiments of steering and speed.

Table 1. PID parameters for steering and speed.

		Kp	Ki	Kd
Steering		0.0113	0.021	0.00005
	5 km/h	0.094	0.038	0.000022
Speed	10 km/h	0.09	0.04	0.000025
	15 km/h	0.043	0.013	0.000065

3.2. Stop Position Accuracy Measurement

The purpose of path tracking is to follow a reference trajectory to a specified destination, so the accuracy of the position of the vehicle when it stops is of equal importance. In this experiment, the vehicle starts from the same starting point every time, drives one lap of the test field automatically, and returns to the starting point, as is shown in Figure 10.



Figure 10. Test field and start position.

As no position sensor was used in this experiment, the accuracy of the stop position was obtained manually using a ruler, as is shown in Figure 11.



Figure 11. Stop position accuracy measurement.

The stop position error is displayed in Table 2. Three different speeds were tested in this experiment, and three sets of experiments were conducted for each speed.

3.3. Tracking Accuracy Analysis in Different Conditions

Several factors affect tracking accuracy during path tracking, and they include vehicle speed, look-ahead distance, starting position and angle, and forward or backward driving modes. They are all analyzed in the following sections.

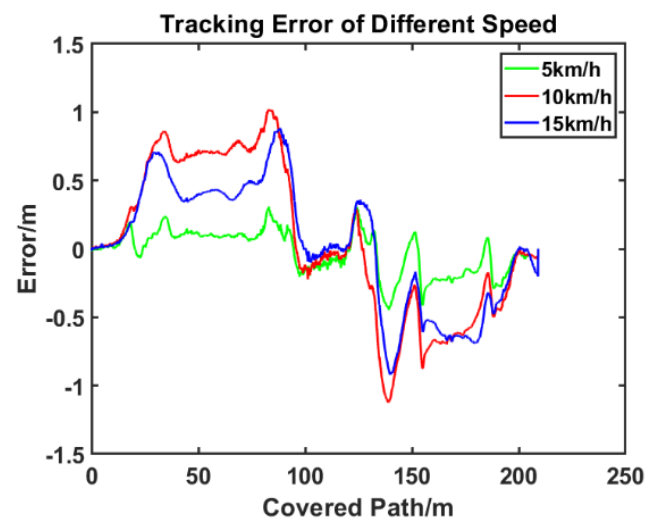
Table 2. Stop position error.

Speed (km/h)	X Error (m)	Y Error (m)	Total Error (m)	Mean Error (m)
5	−0.32	0.060	0.326	0.260
	0.145	−0.049	0.153	
	0.30	−0.027	0.301	
10	0.15	−0.104	0.183	0.383
	−0.24	−0.088	0.256	
	−0.71	−0.043	0.711	
15	0.70	−0.373	0.793	0.505
	−0.14	−0.229	0.269	
	0.45	−0.040	0.452	

3.3.1. Vehicle Speed

Stop position error. Table 2 shows that the error of the stop position increases as vehicle speed increases, and it is also found that there is a certain randomness of error even at the same speed, which relates to the stability of the speed control during each experiment. When there is a small speed fluctuation, this will have an impact on the positioning progress of the differential drive model.

Tracking position error. Tracking position errors at different speeds can be seen in Figure 12, proving that the error is similar to that in the stop position, with the smallest error at 5 km. However, there is little difference between the error at 10 km and 15 km, with a larger error occurring at 10 km in some positions and a larger error at 15 km in others.

**Figure 12.** Tracking position error at different speeds.

3.3.2. Look-Ahead Distance

From the process of pure pursuit model derivation, it can be seen that the look-ahead distance has a relatively large impact on trajectory tracking accuracy. For a more accurate evaluation of its impact, comparison experiments were conducted for look-ahead distances of 3 m, 4.5 m, and 6 m, the results of which can be seen in Figures 13 and 14. Figure 13 shows that the larger the look-ahead distance, the larger the error between the vehicle turning and the reference path, while a smaller look-ahead distance results in a smaller error. Figure 14 shows the magnitude of the error of the distance, including the maximum error of approximately 0.7 m at a look-ahead distance of 6 m, and the maximum error of just 0.3 m at a look-ahead distance of 3 m.

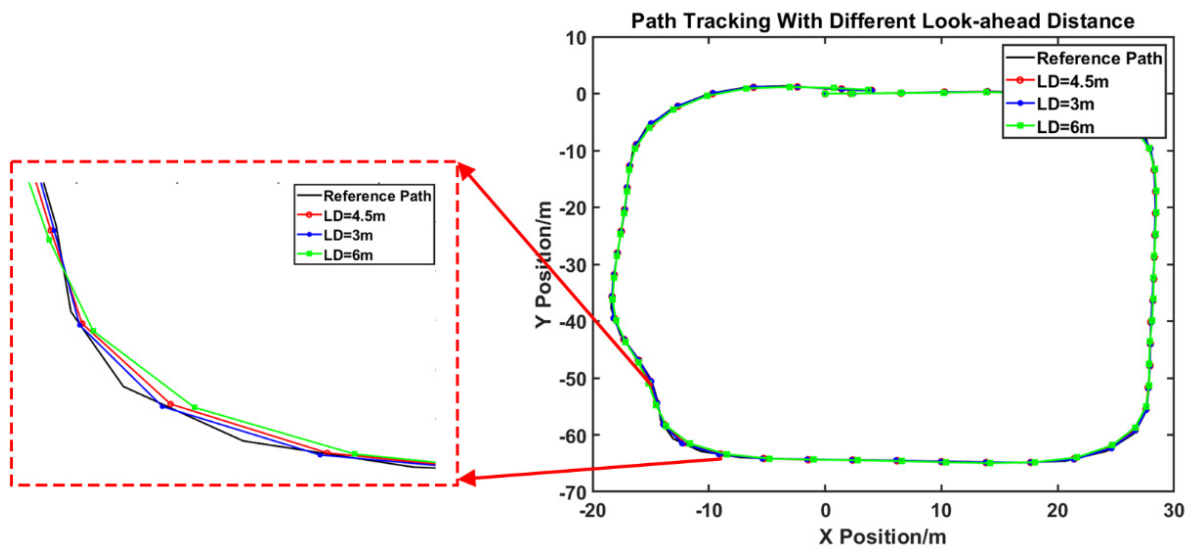


Figure 13. Path tracking with different look-ahead distances.

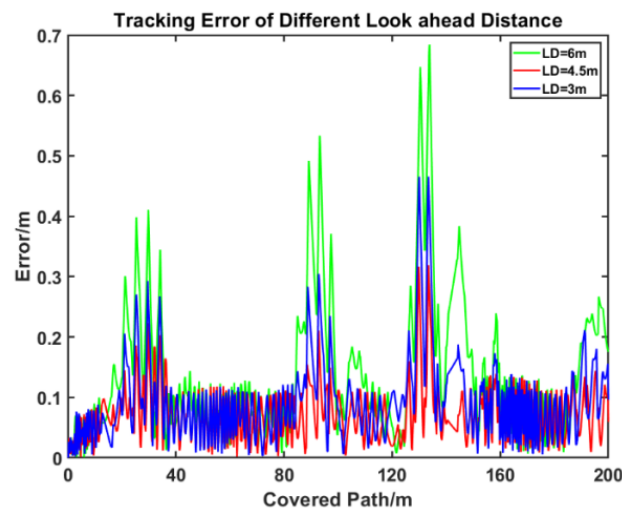


Figure 14. Tracking error with different look-ahead distances.

However, a smaller look-ahead distance is not necessarily better, and when the look-ahead distance is small to a certain value, vehicle control will appear unstable, with the specific performance of the phenomenon being that the steering wheel will quickly swing left and right until the system is out of control. With the experimental vehicle used in this study, the minimum look-ahead distance without the loss of control was found to be 2.45 m, which was obtained after repeated testing.

3.3.3. Start Position and Angle

As the method in this paper uses the dead reckoning method for localization, the initial position and angle have a degree of influence on path tracking accuracy. The position has a small influence on the entire trajectory, but if there is a deviation, the angle will lead to the rotation of the whole path, resulting in a larger deviation, as is shown in Figure 15. Therefore, a comparison experiment was conducted, with one group of data having a certain starting angle deviation, with the other two groups having a very small angle deviation. The results of this are shown in Figure 15.

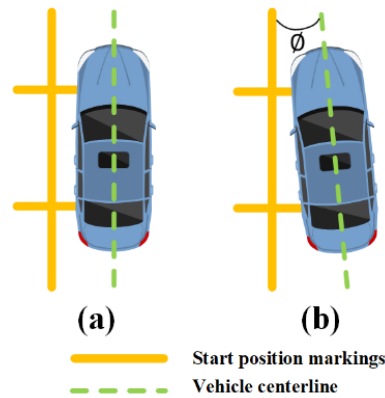


Figure 15. Different start position angles. (a) The initial position without angle deviation; (b) The initial position with angle deviation.

The red curve in Figure 16 is the result of a certain starting angle deviation, and the error is obviously larger than the other two groups, demonstrating a certain regularity. The specific deviation angle is estimated by taking two sets of data at the trajectory of the linear segment, such as points A and B in the figure, and the estimated error angle can be calculated using the following formula:

$$\varnothing = \arctan\left(\frac{y_A - y_B}{x_A - x_B}\right) \tag{19}$$

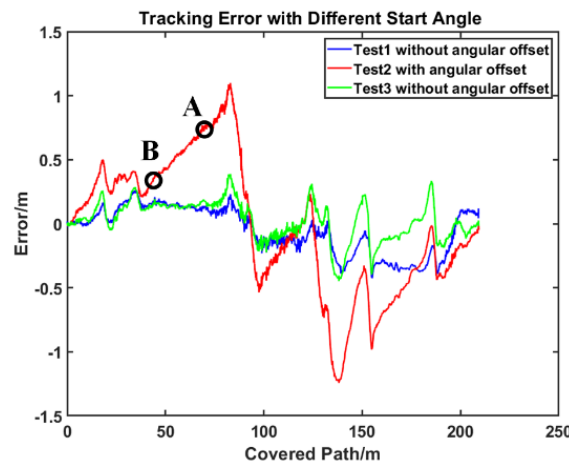


Figure 16. Tracking error with different start angles.

Based on the coordinate values of points A and B on the curve, the calculated angle estimate was 0.931° . In order to verify whether the estimated value was correct, the original trajectory coordinate data from Test2 was rotated by angles of 0.690° , 0.931° , and 1.146° , and the respective tracking errors were then calculated, as is shown in Figure 17. The results show that the curves after rotating 0.690° and 0.931° were closer to the original Test1 and Test3, while rotating 1.146° produced a larger error in the opposite direction. Therefore, the estimated angles can be assumed to be roughly correct.

3.3.4. Driving Mode

The rear wheel speed is used in this paper as an input in the differential drive model for the calculation of real-time vehicle localization, so it is applicable for both forward driving and backward driving modes. To verify the effect the two different driving modes have on path tracking accuracy, this experiment also performed backward autonomous

driving tests, and three sets of comparison experiments were conducted in forward and backward modes, the results of which can be seen in Figure 18. The experimental curves and the average error show the average error in backward mode to be slightly larger than in forward driving, but the difference between them is not considered significant.

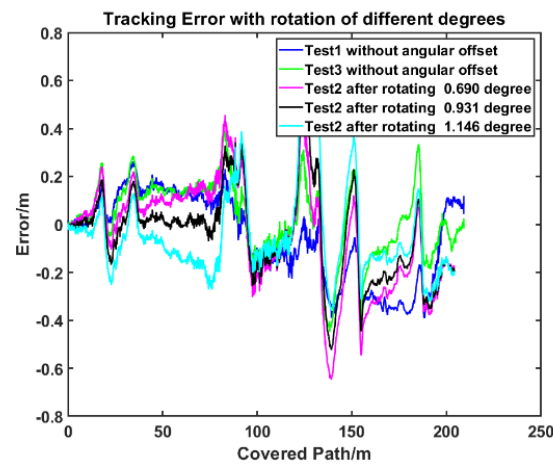


Figure 17. Tracking error with rotation of different angles.

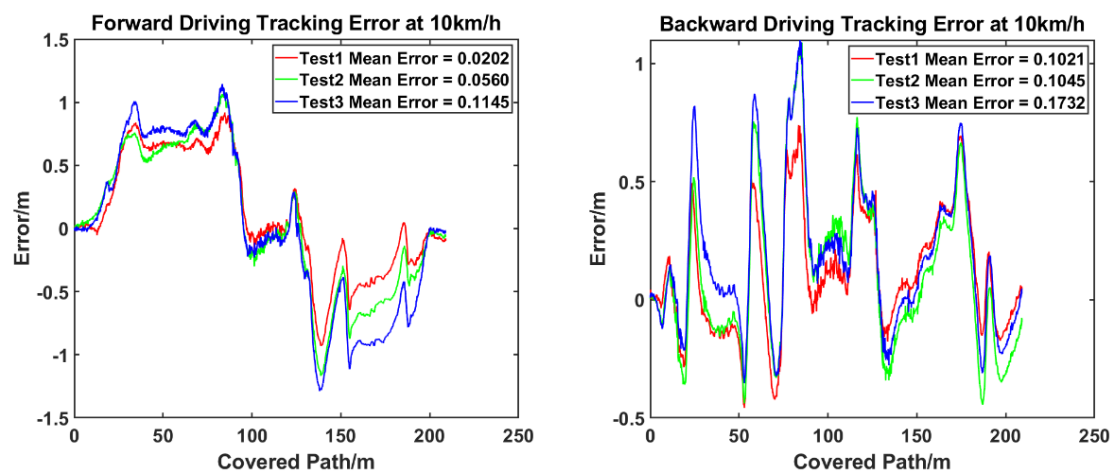


Figure 18. Tracking error of different driving modes at a vehicle speed of 10 km/h.

4. Conclusions

This paper analyzed the principle and details of driverless technology based on the dead reckoning method, including the estimation of the current vehicle position using the differential drive model, the principle of the pure pursuit algorithm, and using a GoPro for calculating the lateral error at each moment of the autopilot process.

The experimental results show that the method can achieve better tracking results at low speeds and the average error increases as speed increases. The results also verify that the look-ahead distance significantly influences the error in the tracking process, particularly when the radius of curvature is large. The greater the look-ahead distance, the smoother the tracking and the larger the error at the turn, but if the look-ahead distance is too small, this may result in control system instability, so choosing a suitable look-ahead distance is very important. As dead reckoning is a relative positioning method, the starting position and angle have a large impact on the tracking result, particularly if the starting angle has a large deviation, which will lead to a global rotation of the entire tracking trajectory. This means that the consistency of the starting position is another key to ensuring the tracking accuracy of this method.

Based on the above conclusions, future research should be conducted on achieving accurate path tracking that can be preserved even when the vehicle travels at high speed and how this method can be combined with path planning as a means of achieving automatic parking in indoor parking lots.

Supplementary Materials: The following supporting information can be downloaded at: <https://www.mdpi.com/article/10.3390/app13010317/s1>, Video S1: Demonstration.

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

Funding: This research was funded by the China Scholarship Council, number 202108330169.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: This work was conducted at the mechatronics laboratory “Mechlab” at the Hochschule für Technik und Wirtschaft Dresden (HTW Dresden, also known as Dresden University of Applied Science) with the support and help of colleagues there, particularly Engert Dirk and Franziskus Mendt.

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

References

1. Paden, B.; Cap, M.; Yong, S.Z.; Yershov, D.; Frazzoli, E. A Survey of Motion Planning and Control Techniques for Self-Driving Urban Vehicles. *IEEE Trans. Intell. Veh.* **2016**, *1*, 33–55. [[CrossRef](#)]
2. Yurtsever, E.; Lambert, J.; Carballo, A.; Takeda, K. A Survey of Autonomous Driving: Common Practices and Emerging Technologies. *IEEE Access* **2020**, *8*, 58443–58469. [[CrossRef](#)]
3. Belrán, J.; Guindel, C.; Moreno, F.M.; Cruzado, D.; Garcia, F.; De La Escalera, A. Birdnet: A 3d object detection framework from lidar information. In Proceedings of the 2018 21st International Conference on Intelligent Transportation Systems (ITSC), Maui, HI, USA, 4–7 November 2018; pp. 3517–3523.
4. Wu, X.; Hong, D.; Chanussot, J. UIU-Net: U-Net in U-Net for Infrared Small Object Detection. *arXiv* **2022**, arXiv:2212.00968. [[CrossRef](#)]
5. Wu, X.; Hong, D.; Chanussot, J.; Xu, Y.; Tao, R.; Wang, Y. Fourier-based rotation-invariant feature boosting: An efficient framework for geospatial object detection. *IEEE Geosci. Remote Sens. Lett.* **2019**, *17*, 302–306. [[CrossRef](#)]
6. Wu, X.; Hong, D.; Tian, J.; Chanussot, J.; Li, W.; Tao, R. ORSim detector: A novel object detection framework in optical remote sensing imagery using spatial-frequency channel features. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 5146–5158. [[CrossRef](#)]
7. Lan, W.; Dang, J.; Wang, Y.; Wang, S. Pedestrian detection based on YOLO network model. In Proceedings of the 2018 IEEE International Conference on Mechatronics and Automation (ICMA), Changchun, China, 5–8 August 2018; pp. 1547–1551.
8. Xu, S.; Peng, H. Design, Analysis, and Experiments of Preview Path Tracking Control for Autonomous Vehicles. *IEEE Trans. Intell. Transp. Syst.* **2020**, *21*, 48–58. [[CrossRef](#)]
9. Yao, Q.; Tian, Y.; Wang, Q.; Wang, S. Control Strategies on Path Tracking for Autonomous Vehicle: State of the Art and Future Challenges. *IEEE Access* **2020**, *8*, 161211–161222. [[CrossRef](#)]
10. Wang, R.; Li, Y.; Fan, J.; Wang, T.; Chen, X. A novel pure pursuit algorithm for autonomous vehicles based on salp swarm algorithm and velocity controller. *IEEE Access* **2020**, *8*, 166525–166540. [[CrossRef](#)]
11. Elbanhawi, M.; Simic, M.; Jazar, R. Receding horizon lateral vehicle control for pure pursuit path tracking. *J. Vib. Control* **2018**, *24*, 619–642. [[CrossRef](#)]
12. Han, G.; Fu, W.; Wang, W.; Wu, Z. The Lateral Tracking Control for the Intelligent Vehicle Based on Adaptive PID Neural Network. *Sensors* **2017**, *17*, 1244. [[CrossRef](#)]
13. Al-Mayyahi, A.; Wang, W.; Birch, P. Path tracking of autonomous ground vehicle based on fractional order PID controller optimized by PSO. In Proceedings of the 2015 IEEE 13th International Symposium on Applied Machine Intelligence and Informatics (SAMII), Herl’any, Slovakia, 22–24 January 2015; pp. 109–114.
14. Liu, S.; Hou, Z.; Tian, T.; Deng, Z.; Li, Z. A Novel Dual Successive Projection-Based Model-Free Adaptive Control Method and Application to an Autonomous Car. *IEEE Trans. Neural. Netw. Learn. Syst.* **2019**, *30*, 3444–3457. [[CrossRef](#)] [[PubMed](#)]
15. Hu, C.; Wang, R.; Yan, F.; Chen, N. Should the Desired Heading in Path Following of Autonomous Vehicles be the Tangent Direction of the Desired Path? *IEEE Trans. Intell. Transp. Syst.* **2015**, *16*, 3084–3094. [[CrossRef](#)]

16. Falcone, P.; Borrelli, F.; Asgari, J.; Tseng, H.E.; Hrovat, D. Predictive active steering control for autonomous vehicle systems. *IEEE Trans. Control. Syst. Technol.* **2007**, *15*, 566–580. [[CrossRef](#)]
17. Tan, Q.; Dai, P.; Zhang, Z.; Katupitiya, J. MPC and PSO Based Control Methodology for Path Tracking of 4WS4WD Vehicles. *Appl. Sci.* **2018**, *8*, 1000. [[CrossRef](#)]
18. Tang, L.; Yan, F.; Zou, B.; Wang, K.; Lv, C. An Improved Kinematic Model Predictive Control for High-Speed Path Tracking of Autonomous Vehicles. *IEEE Access* **2020**, *8*, 51400–51413. [[CrossRef](#)]
19. Mayne, D.Q. Model predictive control: Recent developments and future promise. *Automatica* **2014**, *50*, 2967–2986. [[CrossRef](#)]
20. Kuutti, S.; Fallah, S.; Katsaros, K.; Dianati, M.; McCullough, F.; Mouzakitis, A. A Survey of the State-of-the-Art Localization Techniques and Their Potentials for Autonomous Vehicle Applications. *IEEE Internet Things J.* **2018**, *5*, 829–846. [[CrossRef](#)]
21. Ahmed, A.N.; Eckelmann, S.; Anwar, A.; Trautmann, T.; Hellinckx, P. Lane marking detection using LiDAR sensor. In Proceedings of the International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, Yonago, Japan, 28–30 October 2020; pp. 301–310.
22. Carlson, C.R.; Gerdes, J.C.; Powell, J.D. Error Sources When Land Vehicle Dead Reckoning with Differential Wheelspeeds. *Navigation* **2004**, *51*, 13–27. [[CrossRef](#)]
23. Welte, A.; Xu, P.; Bonnifait, P. Four-wheeled dead-reckoning model calibration using RTS smoothing. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 312–318.
24. Freydin, M.; Or, B. Learning Car Speed Using Inertial Sensors for Dead Reckoning Navigation. *IEEE Sens. Lett.* **2022**, *6*, 1–4. [[CrossRef](#)]
25. Brossard, M.; Barrau, A.; Bonnabel, S. AI-IMU Dead-Reckoning. *IEEE Trans. Intell. Veh.* **2020**, *5*, 585–595. [[CrossRef](#)]
26. Wu, H.-M.; Zaman, M.Q. LiDAR Based Trajectory-Tracking of an Autonomous Differential Drive Mobile Robot Using Fuzzy Sliding Mode Controller. *IEEE Access* **2022**, *10*, 33713–33722. [[CrossRef](#)]
27. Martins, F.N.; Sarcinelli-Filho, M.; Carelli, R. A Velocity-Based Dynamic Model and Its Properties for Differential Drive Mobile Robots. *J. Intell. Robot. Syst.* **2016**, *85*, 277–292. [[CrossRef](#)]
28. Koubaa, Y.; Boukattaya, M.; Dammak, T. Adaptive sliding-mode dynamic control for path tracking of nonholonomic wheeled mobile robot. *J. Autom. Syst. Eng.* **2015**, *9*, 119–131.
29. 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.
30. Ohta, H.; Akai, N.; Takeuchi, E.; Kato, S.; Edahiro, M. Pure Pursuit Revisited: Field Testing of Autonomous Vehicles in Urban Areas. In Proceedings of the 2016 IEEE 4th International Conference on Cyber-Physical Systems, Networks, and Applications (CPSNA), Nagoya, Japan, 6–7 October 2016; pp. 7–12.

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.