applied sciences

MDPI

Article

Target Points Tracking Control for Autonomous Cleaning Vehicle Based on the LSTM Network

Hua Wang [1,2], Xi Chen [1,2] and Zhonghua Miao [1,2,*]

[1] School of Mechatronic Engineering and Automation, Shanghai University, Shanghai 200444, China
[2] Shanghai Key Laboratory of Power Station Automation Technology, Shanghai 200444, China
* Correspondence: zhhmiao@shu.edu.cn; Tel.: +86-138-1647-4089

Received: 27 June 2019; Accepted: 6 September 2019; Published: 11 September 2019

Abstract: In order to efficiently and exactly in tracking the desired path points, autonomous cleaning vehicles have to adapt their own behavior according to the perceived environmental information. This paper proposes a target points tracking control algorithm based on the Long Short-Term Memory network, which can generate the speed and yaw rate to arrive at the target point in real time. The target point is obtained by a parameter named foresight distance that is deduced based on the fuzzy control, whose inputs are the speed and yaw rate of the vehicle at the current point. The effectiveness of the proposed algorithm is illustrated by the simulation and field experiments. Compared with other classical algorithms, this algorithm can track the point sequence on straight path and multiple curvature path more accurately. The field experiment indicates the proposed controller is efficient in following the pre-defined path points, furthermore, it can make the autonomous cleaning vehicle run smoothly in the path which is disturbed by bounded disturbances. The distance errors can meet the actual requirement of the cleaning vehicle during the tracking process.

Keywords: target points tracking; LSTM network; fuzzy control; automatic vehicle

1. Introduction

With the continuous development of advanced technologies such as Information Technology (IT), sensor technology, system engineering, and artificial intelligence, Intelligent Transportation System (ITS) has attracted more and more researchers' attention, especially the design of automatic mobile robots and mobile systems [1]. One of the main hotspots in the field of autonomous mobile robots research is to design a target point tracking controller [2–5], which focuses on reaching the pre-defined point through longitudinal and lateral control of the vehicle [6], thus achieving a whole path points tracking in the given time. To be specific, the longitudinal controller is responsible for adjusting the vehicle's cruise velocity while the lateral controller steers the vehicle's wheels for trajectory tracking [7,8]. Therefore, the resulting control output should minimize the difference between the actual and expected points with respect to the lateral distance and the vehicle heading, and should ensure that the actual paths are smooth. However, for automatic cleaning robot, there is a strong coupling relationship between the longitudinal and lateral dynamics of the vehicle whose dynamic response is very complicated during the cleaning process, because its running involves sprinkling, vacuuming and scrubbing [9]. So the kinematic or dynamic model of the cleaning vehicle is complex [10], and to determine an appropriate target points tracking control algorithm is difficult.

At present, many scholars have studied target points tracking methods based on varied types of models such as geometric, kinematic and dynamic model or combination of them. Compared with the controller based on complex dynamic model, the controllers based on geometry and motion models are easier to implement and more robust to path curvature. In addition, the latter are more suitable for lower velocity driving because they do not consider the internal force of the whole vehicle system.

However, the tracking accuracy of the controller based on dynamic model is usually better at higher speed, because the controller gives priority to the non-linear vehicle dynamic.

Existing tracking methods concretely include PID control [11], fuzzy control [12,13], optimal control [14], pure pursuit control [15], neural network control [16,17], and so on. They improve the goal points tracking performance for automatic vehicles in different aspects. In [11], a classical PID approach is proposed for the path tracking controller, which utilizes a simple linearized model consisting of an integrator and a delay to simulate a mobile robot. This experimental results demonstrate the good tracking performance and robustness of the proposed controller. The advantage of fuzzy control is that it does not require precise kinematics model or dynamic model for robots, reference [12] combines the linear regulator theory with Takagi-Sugeno fuzzy methodology and proposes a method to fulfil path tracking of nonlinear systems. It is worth mentioning that this method is easy to implement due to the use of linear control techniques. Reference [14] requires the vehicle to run at low speed, so the dynamic properties have little influence on the path tracking behavior. In [14], an optimal controller is designed to minimize the orientation error and the offset from a path. The controller structure is defined in terms of the running speed of the vehicle. Thus, it has the superiority of being flexible for speed changes, but the simulation or experiment performances of the controller are not specifically described in the text. Pure pursuit control [15] presents a steering control law for automatic vehicles according to the geometric relation between the ego-vehicle and the desire tracking path points. A parameter named preview distance is usually used to obtain the control variables and it has certain potentially abilities to extend from simple circular arc calculations to much more complex geometric theorems. Although the above methods can achieve good tracking results under specific conditions, some pre-defined path points used are too idealized. Since most of the path points currently used for tracking are obtained by smoothing and filtering from the original path points, which are collected by the map building. The original path usually has some noise points. It is still unknown whether the above methods can maintain a good tracking effect on the path with bounded disturbance, so the robustness of the target point tracking control algorithm remains to be proven. In addition, each algorithm has its own advantages and application occasions. If we can directly extract their good effect data without the tedious analysis process and construct a new control algorithm based on these existing data, it would be a new breakthrough in the field of the automatic control.

According to the universal approximation theorem [18], artificial neural network (ANN) has the potent ability to approximate the nonlinear system and it is considered as an effective approach to deal with complex problems. Related research in neural networks has shown exemplary results in the field of tracking control of wheeled mobile robot [19] and quadrotors [20], which illustrate the power of learning features directly from the data in the area of deep-learning. A unified robust neural network based control scheme is presented for trajectory tracking of a wheeled mobile robot with unknown parameters in reference [19], the proposed controller has a good performance on the steady-state errors and control inputs in both tracking and stabilization problems. Reference [20] proposes a DNN-based reference learning method that can learn the flight data and improve tracking performance in the quadrotor system, because of its superior capability of approximating abstract. Reference [16] proposes a novel path points tracking method based on learning, which combines the Stanley method with a Neural Dynamic Programming (NDP) algorithm perfectly. Although this method demonstrates state-of-the-art performance, the slight vibration is existed during the path tracking process according to the comprehensive simulation experiment. Due to their success in many applications, this paper chooses artificial neural network in the form of a Long Short-Term Memory (LSTM) network as the basis to study the target points tracking control algorithm, for the reason that path points can be treated as time sequence data in tracking. And to the best of our knowledge, the LSTM framework has not been designed for target points tracking control algorithm to predict the vehicles' speed and yaw rate.

In this paper, we consider the problem of autonomous mobile cleaning robot in target points tracking, where the running speed is relatively lower than other scenarios, but the accuracy of the tracking process is higher because it involves cleaning efficiency. Inspired by the pure pursuit algorithm, we use a key parameter of this algorithm, foresight distance, which simulates the vision of the driver in driving and has the bionic characteristics. The foresight distance can be obtained by the fuzzy control, its inputs are the current velocity and yaw rate of the vehicle, and its output is an appropriate target foresight distance. After setting a suitable foresight distance, a simple geometric method is intended for obtaining the target point which is relative to the current vehicle position. In order to arrive at this target point, two-layer LSTM network controller is proposed to adjust the autonomous vehicle's speed and yaw rate simultaneously and intelligently to reduce the lateral error of the path points tracking. The LSTM model has been trained in proper times based on abundant and good sample data, then the kinematic model of automatic vehicle are established and simulation experiments are carried on two typical paths. The simulation results are compared with the results of three effective control algorithms, and the advantages of the proposed algorithm are analyzed. Beyond that, this paper examines the effectiveness of the algorithm on a cleaning vehicle, studies the variation of velocity and foresight distance in the tracking, and evaluates its performance by the distance deviation of the straight line and the turning section in the field experiment. The real vehicle experiment result shows that the proposed controller is efficient not only in tracking the pre-defined path points smoothly but also in not affecting by the path noise points.

The rest of this article is structured as follows. In Section 2, we briefly review the technical background of the vehicle model, stability condition and the chain structure of LSTM. Section 3 describes the proposed tracking algorithm, which includes the discrimination of the foresight distance, the system description and the detailed network structure. In Section 4, the simulation and field experiments results are presented and Section 5 concludes this paper.

2. Technical Background

In this section, a brief review of the vehicle model and stability condition used in this paper will be analyzed firstly. Then the brief chain structure of LSTM and detail of the process in the information transmission are presented.

2.1. Vehicle Model and Stability Condition Analysis

When deriving the target points tracking control algorithm, the kinematic vehicle model [21] is used. The kinematic vehicle model is based on the Ackermann steering geometry that assumes the line perpendicular to each wheel should intersect at the center of the turning arc of the vehicle with a turning radius of R. Vehicle tires are regarded as rigid wheels and the lateral sliding between the tires and the ground is not considered. This model is usually used to obtain the steering control law as it can approximate the motion of the vehicle reasonably well at moderate steering angles and low velocities. The kinematics model of the two-wheel vehicle is simplified as shown in Figure 1.

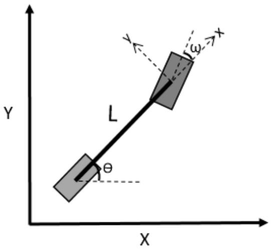


Figure 1. The simplified two-wheel vehicle kinematic model.

The vehicle is described with two sets of Cartesian coordinates, local (x-y) and global (X-Y). Local coordinate is regarded as a moving coordinate with respect to the global coordinate because the vehicle is moving constantly. Supposing the speed of the vehicle (v) in the body frame is purely in x-direction. The relationship between the front wheel steering angle ω and the vehicle's yaw θ is established as below. The L is the wheelbase, the $\theta'(t)$ is the yaw rate, $x'(t)$ is the X-direction speed and $y'(t)$ is the Y-direction speed. The following relations can be obtained by the kinematic analysis.

$$x'(t) = v\sin\theta \tag{1}$$

$$y'(t) = v\cos\theta \tag{2}$$

$$\theta'(t) = \frac{v\tan\omega}{L} \tag{3}$$

The steering wheels of the cleaning vehicle is driven by stepping motors, whose speeds are proportional to the output of the controllers, so the whole steering system of the cleaning vehicle can be regarded as the first order inertia link, its formula is

$$\dot{\omega} = -\frac{1}{T}(\omega - \omega_U). \tag{4}$$

The ω_U is the target angle of front wheel, T is the time constant in the first order inertia link. The relationship between the curvature of turning and front wheel angle ω is as follows

$$\gamma = \frac{\tan\omega}{l}. \tag{5}$$

The γ is the actual turning curvature. When the front wheel angle is small, the above formula can be linearized as $\gamma = \frac{\omega}{l}$, so the changes of the turning curvature for the cleaning vehicle also be regarded as the first order inertia link.

$$\dot{\gamma} = -\frac{1}{T}(\gamma - \gamma_U) \tag{6}$$

The γ_U is the target turning curvature, the $\theta'(t)$ in the kinematics formula can be regard as $\theta'(t) = v\gamma$. According to the method adopted in reference [22], when the model is in equilibrium $y'(t) = 0$ and $\omega = 2/\pi$, the system characteristic polynomial is

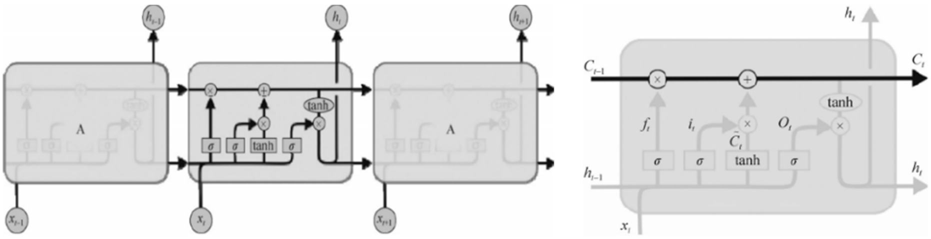
$$f(\lambda) = \lambda^3 - \frac{1}{T}(\varphi_\gamma - 1)\lambda^2 - \frac{v}{T}\varphi_\theta\lambda + \frac{v^2}{T}\varphi_y. \tag{7}$$

According to Routh Criterion and the parameter L_d in pure pursuit algorithm [23], we can get the condition of the system is in a stable state.

$$L_d \geq Tv \tag{8}$$

2.2. LSTM Network

LSTM (Long Short-Term Memory) neural network was proposed by Sepp Hochreiter and Jurgen Schmidhuber in 1997 [24]. It is an optimized RNN (Recurrent Neural Network) that is designed to treat for the time sequence data based on the recurrent framework. With the increase number of times of information transmission, RNN will have problems in learning to convey the information due to the gradient vanishing or exploding issue, but the LSTM network can avoid these difficulties perfectly. The most remarkable point of LSTM is its repeated neural network memory module, also known as the hidden layer which is no longer a simple neuron node; it's a computing node with rich internal structure, which contains a self-connected memory cell and three "gates" that control the flow of information. The gates include input gate, output gate and forget gate. Due to their recurrent structure, even a single layer of LSTM nodes can be considered as a "deep" neural network. The structure of LSTM is shown in Figure 2.



(**a**) (**b**)

Figure 2. (**a**) The chain structure of Long Short-Term Memory. It takes the output of a LSTM cell at previous time step as the input of the LSTM cell at current time step and forms a multilayer LSTM; (**b**) Details of the information transfer process. The memory cell is like a conveyor belt, which directly passing through the whole cell.

At time step t, the calculation formulas of the l-th layer units can be expressed as:

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f) \quad i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \tag{9}$$

$$\widetilde{C}_t = \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \quad C_t = f_t \otimes C_{t-1} + i_t \otimes \widetilde{C}_t \tag{10}$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}C_t + b_o) \quad h_t = o_t \otimes \mathrm{tanh}h(C_t) \tag{11}$$

where the vectors f_t, i_t, o_t, C_t represent the activation of the forget, input, output gates, and memory cells, respectively. W_{ab} terms denote the weight matrix between a and b (e.g., W_{xi} is the weight matrix from the inputs x_t to the input gate i_t), and these four weight matrixes are associated with the inputs x_t. The b_α describes the bias term of α with $\alpha \in \{i, f, C, O\}$, σ is the logistic sigmoid function as $\sigma(x) = 1/(1 + e^{-x})$, and '$\otimes$' denotes the Hadamard product. The multiplicate LSTMs model can be applied to solve many real-life time sequence modeling problems, and multiple LSTMs can be stacked or concatenated flexibly to form more complex structures.

3. Proposed Trajectory Tracking Control Algorithm

In this section, the method to determine the foresight distance is described, the preview point is selected, the target points tracking system is designed, the network structure, and training method of the LSTM algorithm are proposed.

3.1. Determination of the Foresight Distance

In practical application, an autonomous vehicle have to work on different curvature paths, so it needs a very robust tracking controller. Motivated by the pure pursuit algorithm [25–27], a key parameter of this algorithm, foresight distance, are used, which has been widely applied in the field of target points tracking. It simulates the vision of the driver during the driving and has the bionic characteristics.

The preview ability is very significant for target points tracking, but to select the foresight distance as a linear function of vehicle velocity is evidently not a good choice [28]. On the one hand, a shorter foresight distance provides a smaller tracking error for a vehicle, but its stability is affected. On the other hand, a longer foresight distance provides a smoother tracking performance, but it may cause cut corners disaster while executing turns. Therefore, the trade-off between stability and tracking performance is a thorny matter. As a result, a method to improve the tracking performance by preview distance based controller is proposed in this paper.

In order to closely approximate the continuous path, we divide the reference path into uniform points with index numbers, so the denser discretization of the path is realized. In this paper, the method of selecting the preview point is to draw a circle with radius L_d centered on rear axle of the vehicle, as indicated in Figure 3. L_d is the foresight distance obtained by the fuzzy control according to the real-time situation of the vehicle. A double-input and single-output structure is adopted, in which the inputs of the fuzzy controller are the current velocity v and the yaw rate ω of the vehicle, and the output is an appropriate target foresight distance L_d.

To take the fuzzy set of speed grades as {Z, PS, PM, PB, PL}, its domain are [0, 1.0]. The fuzzy set of yaw rate grades is {NB, NM, NS, Z, PS, PM, PB}, its domain are [−90, 90], where B is Big; N is Negative; M is Middle; S is Small; Z is Zero; P is Positive; L is Large. To take the typical triangle function as the membership functions, and the fuzzy control rules will affect the outputs.

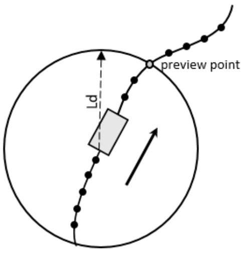


Figure 3. Selection of preview point.

When designing the fuzzy control rules, the following principles are considered: (1) If the absolute value of ω is smaller, the L_d is larger; If the absolute value of ω is larger, the L_d is smaller. (2) Under the condition of satisfying the system stability, if the v of the vehicle is slower, the L_d is smaller. If the v is faster, the L_d is larger. The basic field of foresight distance L_d is [0.1, 1.3] (/m) and the quantization level is {0.1, 0.4, 0.7, 1.0, 1.3} = {S, SM, M, ML, L} of fuzzy control, as shown in Table 1. After that, the output from any input or combination of input fuzzy sets can be determined. Here, the "and" rule is used between two inputs. It is obvious that foresight distance, driving velocity, and the yaw rate are coordinated.

Table 1. Foresight distance fuzzy control lookup table.

ω/(°) v/(m/s)	NB	NM	NS	Z	PS	PM	PB
Z	S	S	SM	SM	SM	S	S
PS	S	SM	M	M	M	SM	S
PM	S	SM	M	ML	M	SM	S
PB	S	M	ML	L	ML	M	S
PL	SM	ML	L	L	L	ML	SM

Based on the above method, the appropriate foresight distance can be determined according to different conditions in the process of goal points tracking.

3.2. Trajectory Tracking Control System Description

The main objective of this study is to design an adaptive controller that can determine the control variables ω and v for the autonomous cleaning vehicle to reach the goal points. LSTM network is robust and efficient for non-linear system control, which can make a series of non-linear transformations on the input data, and solve the problems of bad training approximation and generalization as much as possible. The use of deep learning neural network provides possibilities for multiple types of complex function calculations [29,30]. Since the path points and some process variables during the tracking can be considered as the time sequence data intuitively, the LSTM network can be used for predicting the control variables of the cleaning vehicle. The overall structure of the target points tracking algorithm is shown in Figure 4.

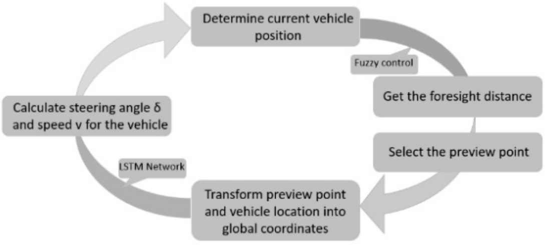

Determine current vehicle position
Fuzzy control
Get the foresight distance
Select the preview point
Transform preview point and vehicle location into global coordinates
LSTM Network
Calculate steering angle δ and speed v for the vehicle

Figure 4. The overall structure of the proposed trajectory tracking algorithm.

Deep learning requires a large set of samples in order to converge to a solution without overfitting and to be well extended to the previously unseen samples. The training samples are collected from our own field and simulation experiments. In the field experiment, the cleaning vehicle should be pushed to create a pre-defined path firstly. Besides, the laser visual sensor and ultrasonic sensors can perceive and record the characteristics of the pre-defined path such as the passing path points (Containing noise points, which are the induction error of sensors), and the surrounding environment such as the location of walls. Then the traditional and improved pure pursuit [28] algorithms are separately used to obtain the sample velocity and sample yaw rate at each actual position of vehicle passed. The real difference between two methods lies in their foresight distance. Traditional pure pursuit algorithm with constant foresight distance performs well in the straight line path points, while the improved algorithm with dynamic foresight distance has good performance at the turning due to its linear function of vehicle velocity. Finally, when the detected wall is within 0.5 m around the vehicle, the vehicle velocity needs to be readjusted based on the aforementioned velocity by the following formula.

$$v_{final} = v_t * \left(\frac{d}{0.5}\right) \tag{12}$$

The v_t is the speed obtained by the experimental tracking controller. The d is the minimum distance from the vehicle nose to the wall, and $\frac{d}{0.5}$ is equivalent to a decay factor that can reduce the speed of the vehicle at the appropriate times to ensure safety. The resulting velocity v_{final} and yaw rate values will be fed into the neural network as the sample data (label). The simulation platform can extract more representative data through simulating cleaning routes in different environments by human intervention. Ultimately, 100 maps are analyzed, each with 500 different pre-defined path, for a total of 50,000 paths and about 100,000,000 path points, including diverse turning and straight path segments. About 20,000,000 groups of velocity and yaw rate sample data are obtained under two tracking algorithm.

At time t, the system supposes that the cleaning vehicle can determine the global coordinates of the target point $(x_{\text{target}}, y_{\text{target}})$ in the pre-defined path detected by the sensors (e.g., laser sensor and ultrasonic sensors). Besides, it can get its owner accurate position (x_c, y_c) based on the merge of gyroscopes, odometers, and amcl algorithm. According to the coordinate transformation relation, the angle α_t and distance d_t between the current position and the target point can be acquired as Equations (13) and (14). It is stipulated that the positive value α representing the target point is on the left side of the vehicle, while the negative value α representing the target point is on the right side of the vehicle.

$$\alpha_t = \tan^{-1}\left(\frac{y_{\text{target}} - y_c}{y_{\text{target}} - x_c}\right) \tag{13}$$

$$d_t = \sqrt{(x_{\text{target}} - x_c)^2 + (y_{\text{target}} - y_c)^2} \tag{14}$$

The set of the observations used for reaching the target point on the reference path at the time step t is given by: $O_t = \{\alpha_t, d_t, k_{1t}, k_{2t}, r_t\}$, The features of α_t as well as d_t are chosen to replicate the driving habits of the human driver, because drivers tend to visually measure the distance and the angle to the target point. Then the speed and yaw rate are determined according to above three variables. k_{1t}, k_{2t} represent the minimum distances from the obstructions to the two edges in front of the rectangular-shaped cleaning vehicle equipped with ultrasonic sensors. r_t is a value detected by the laser, which stands for the distance from the obstructions to the vehicle nose. Considering the length and width of the vehicle and above five values comprehensively, the vehicle can control the speed and yaw rate more intelligently when turning or approaching obstacles. For example, if the vehicle's length is too long to turn around a curve or the obstacle is too close to the vehicle's body, the control variables need to be fine-tuned the control parameters. So the input of the LSTM network is the observations data O_t, whose features are scaled to remain in an acceptable range with respect to the activation function, and the output is the predicted data M: $M = \left\{v_{final}, \omega\right\}$, where, v_{final} is the speed and ω is the yaw rate to reach the target point.

Figure 5 depicts the detail architecture of the proposed LSTM model, a two-layer deep LSTM network is used to build a structure for the input sequences in our network.

Appling the proposed LSTM network to the target points tracking algorithm can provide the cleaning vehicle with a robust controller, and at the same time promote the progress of tracking control algorithm greatly.

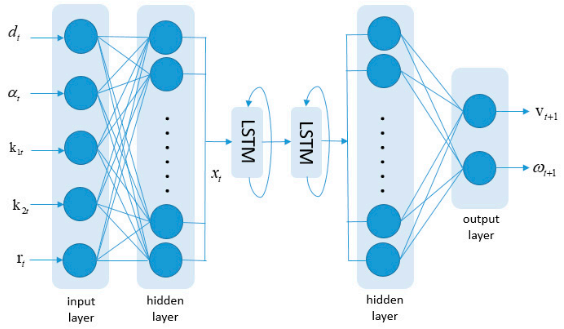


Figure 5. Architecture of the proposed LSTM model.

3.3. Network Structure

In this paper, deep-learning neural network can learn autonomously on the premise that the model knows the inputs and outputs, and it transforms the samples into some rules. After many iterations, the ability of neural network to recognize features increases gradually. The focus of the entire network learning process is to continuously optimize the learned laws based on the minimum condition of the loss function. Based on those rules, the model can not only predict what has been learned ahead, but also makes more reasonable predictions for the content that is not learned before.

After several tests, 256 features are extracted from the input layer, so the length of x_t vector is 256. Then a two-layer LSTM architecture is used, since the large and complex trajectory training data need to be modeled to find more potential relationships of the temporal events in sequences. The two-layer is best for our task, which increasing the generalization capability of our model and expanding the network capacity as much as possible to capture the trajectory data. Finally, the hidden layer is used to transfer the 256-dimension of data into 2-dimension being consistent with the output layer.

The proposed network model is built using Python on Keras with a Tensorflow backend. Because there is still no known appropriate method for selecting correct hyper parameters for LSTM, so a variety of experiments are carried out and finally some parameters are selected with better performances. The training dataset is divided into batches of sequences of size n = 256. We use the popular zero-mean and unit variance to normalize the input data. During the training, the loss function is optimized as follows:

$$loss = \lambda_1 \frac{\sum_{i=1}^{n} \left(v_{final\_i} - v_i{}'\right)^2}{n} + \lambda_2 \sum\nolimits_{i=1}^{n} \frac{\left(\omega_i - \omega_i{}'\right)^2}{n}. \tag{15}$$

where, v_i and ω_i are the labels of the speed and yaw rate data in a batch, $v_i{}'$ and $\omega_i{}'$ are the predictive value obtained by the neural network. We set the weights $\lambda_1 = 0.4$ and $\lambda_2 = 0.6$ to reflect the importance of the yaw rate, which can influence the accuracy of path tracking, while the speed affects the efficiency of cleaning. For the minimization of the loss function, the Adam optimization is applied in the training process, with the dynamic learning rate start with 0.004 and decay with a rate of 0.0005, which can reduce the neural network training time without losing the prediction accuracy. The dropout rate is 0.8. In order to regularize the network, the L2 regularization is used. Combining all parameters and methods, the whole system model is optimized to improve the network performance.

4. Experiment and Analysis

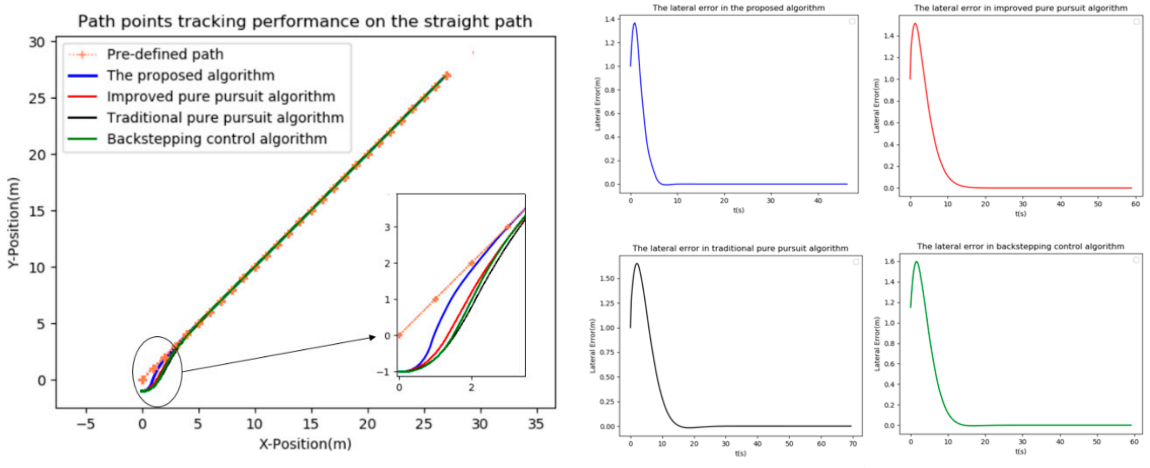
In order to verify the performance of the proposed algorithm, the simulation platform and field experiment are used to test the target points tracking control capability of the cleaning vehicle in this section.

4.1. The Analysis of Simulation Performance

The kinematics model of a two-wheel vehicle is taken as the control object to validate the control performance of this algorithm in the simulation platform. The wheelbase of the vehicle is 1 m, the initial velocity is 0 m/s, the vehicle's nose direction is X-axis positive direction and the initial position is (0,−1), so the initial position deviation is 1 m. The simulation experiments are carried out under the conditions of the straight path and the multiple curvature path.

When tracking the straight path, the simulation performance is shown in Figure 6a. Four methods, namely the proposed method, the traditional pure pursuit method, the improved pure pursuit method and backstepping control method [31], are adopted to evaluate the tracking performances and are compared as well. The foresight distance and speed are set as constant values 0.7 m and 0.5 m/s in the traditional pure pursuit method. In the improved pure pursuit [25] method, the foresight distance is changed from constant to dynamic adjustment, which is a linear function of the vehicle velocity. In the backstepping control method, the desired configuration is the origin of the state space, while the initial configuration at t = 0 is x(0) = 0, y(0) = −1, θ(0) = 0, φ(0) = 0. We can intuitively see the difference among the above four controllers in Figure 6a, the proposed algorithm takes precedence over the other algorithms in tracking the pre-defined path points as it possesses the rapidity in the initial phase. The lateral error on the straight path in Figure 6b also shows that the proposed intelligent controller has lower lateral error when tracking some straight path points, followed by the improved pure pursuit method, while the traditional pure pursuit method and backstepping control method have the biggest lateral error. However, all four methods can perform well tracking abilities after reaching steady state.

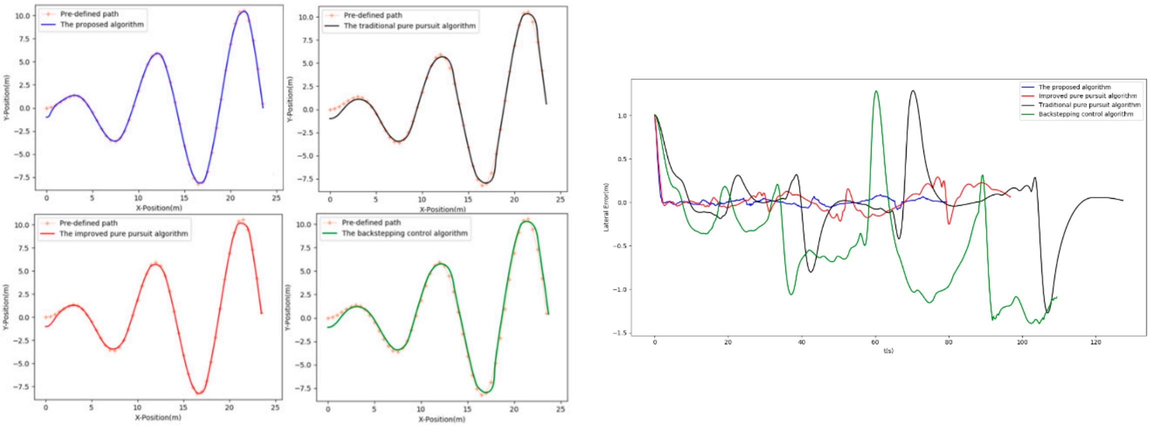
When tracking the path with multiple curvature, the simulation performance is shown in Figure 7a and lateral error is shown in Figure 7b. The chief reason for choosing this path is that the controller's navigation ability can be tested in various turning operations. As can be seen from Figure 7a, the traditional pure pursuit algorithm can track the expected path after a period of oscillations in the initial stage, but the algorithm presented in this paper can quickly stabilize and show significant advantages. The errors of the traditional pure pursuit algorithm are large on the corners and the vehicle needs to be adjusted over and over again to track the expected path after turning. Although the improved pure pursuit algorithm expresses less deviation than the traditional pure pursuit algorithm at the turning points, it remains worse than the proposed algorithm. However, due to the existence of path curvature, the backstepping algorithm has the largest error in the tracking process. At corners, the proposed algorithm can adapt to the change of curvature quickly, and shows flexible adaptive ability to trace the expected path with tiny errors. In addition, the larger change rate of path curvature, the greater tracking errors of the four algorithms, but the proposed algorithm always performs a superior performance over other algorithms, and its maximum lateral error is 0.08 m after stabilization.



(**a**) (**b**)

Figure 6. (**a**) Target points tracking performance on the straight path. (**b**) Tracking lateral error of straight path points under four different control schemes.

The above simulation results explain that the proposed method has better control performances and effects in ideal status without noise points. However, in order to verify the performance of this method, it is not enough without real vehicle experiments. Therefore, the field experimental result on a real cleaning vehicle are given below.



(**a**) (**b**)

Figure 7. (**a**) Target points tracking performance on the path with multiple curvature. (**b**) Tracking lateral error under the path with multiple curvature.

4.2. Field Experiment of Cleaning Vehicle

In order to test the actual working effect of this algorithm, the cleaning vehicle shown in Figure 8 is used in the field test. This cleaning vehicle is equipped with the laser radar and ultrasonic sensors. Its body length is 1.5 m and the wheelbase of two driving wheels is 1 m. There is a cleaning device in front of the vehicle and two driving wheels in the rear. Based on the practical experience and security factors, this machine is designed to have a speed of no more than 1 m/s.

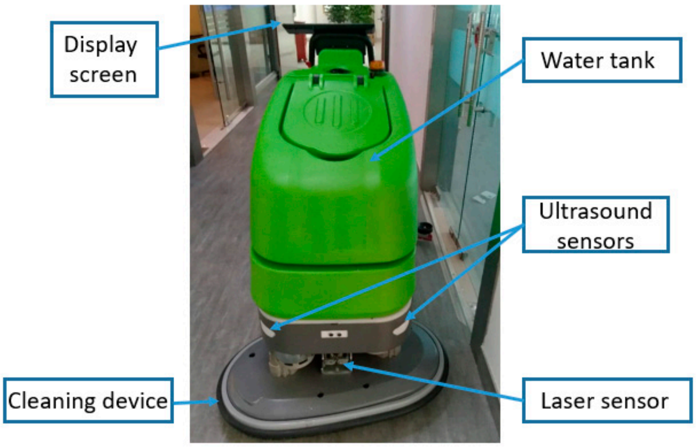


Figure 8. Front view of the cleaning vehicle.

The field experiment is carried out in the subway station hall with the length of 30 m and the width of 3.8 m. It is necessary to manually specify the running route (pre-defined path) before the experiment by pushing the cleaning vehicle. The total length of running route is 197.3 m, the initial position of the vehicle is (0, 0), the initial yaw angle is 0°, and the initial speed is 0 m/s. The dark blue curve in Figure 9 is the specified running route.

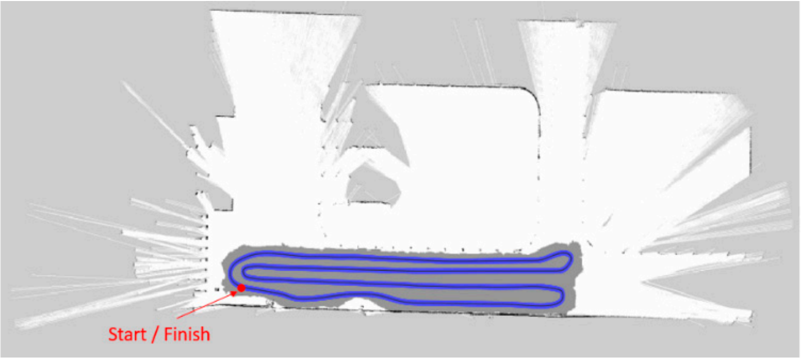


Figure 9. Experimental background and pre-defined path, top down view.

In practice, an autonomous cleaning vehicle has to work on the paths with various shapes, so it needs a very robust tracking controller. Rapid changing path points require a re-computation of the speed and the yaw rate. Because the effect of traditional pure pursuit algorithm and backstepping algorithm control are often poor, these two methods are not tested in the field experiment.

The target points tracking diagram is displayed in Figure 10, it can be seen that the proposed algorithm has more advantages than the improved pure pursuit algorithm, especially in the turning path and encountering the noise points. The proposed controller can track the reference path well in straight lines as well as corners, freedom from the impacts of noise points. However, the improved pure pursuit controller has large deviation at turning and it is easily disturbed by noise points. There may be a phenomenon that the cleaning vehicle does not follow the reference path when the noise point is too close to the real path. Besides, the cleaning vehicle can run or stop smoothly near the start point and the end point by adjusting the control parameters in Figure 10a. But in Figure 10b, noise points near the starting point affect the tracking process in the initial stage. In the final stage, the vehicle tracking ends ahead of schedule, because the preview distance and speed cannot be changed adaptively. Therefore, the field experimental results indicate that the proposed controller is robust and effective for the autonomous cleaning robot to track the pre-defined path points.

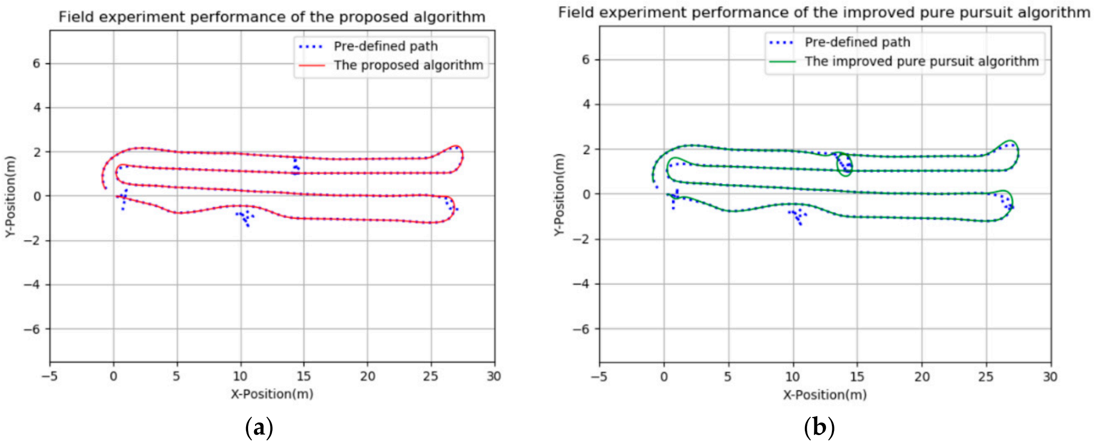


Figure 10. (**a**) Field experiment performance of the proposed algorithm. (**b**) Field experiment performance of the improved pure pursuit algorithm.

The Figures 11 and 12 show the velocity and foresight distance variations of the two algorithms in tracking. The proposed controller is capable of adjusting the cleaning vehicle's speed and the foresight distance intelligently. Throughout this process, these two values dropped observably on the corners. It is important to note that the LSTM network has learned the characteristics of the sample data. And the speed, as well as foresight distance when the vehicle walking against the wall (Saying that in approximately 0–50 s), are significantly less than the speed as well as foresight distance when not walking against the wall, except in the case of turning. However, in the improved pure pursuit algorithm, the speed and foresight distance increase slowly from 0 to the set values, but because the speed released by the controller may not be the same as the actual running speed of the vehicle, and vehicle needs to adjust the foresight distance according to the actual speed, so the speed and foresight distance have some vibration when the vehicle is in steady state. Unexpected speed and forward-looking distance are the reasons why the vehicle tracking ends early in the final phase.

The distance errors of target points tracking are obtained by comparing the reference path data with the tracking path data in automatic navigation. The average tracking error is 0.05 m, and the maximum tracking error is 0.13 m, which appears in the turning process of the cleaning vehicle. The average tracking error on the linear path is 0.02 m and the maximum tracking error is 0.04 m. These errors are within the allowable error range of the cleaning vehicle operation, which indicates that this algorithm has good navigation effect. During the cleaning process, the cleaning vehicle's tracking route is accurate and reliable. Compared with the unintelligent speed vehicles, the intelligent changing speeds give the cleaning vehicle higher working efficiency.

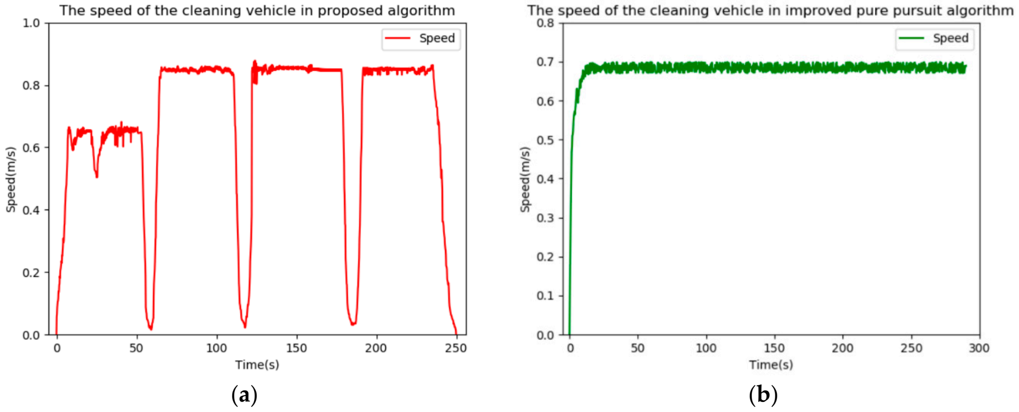


Figure 11. (**a**) The speed of the cleaning vehicle under the proposed algorithm. (**b**) The speed of the cleaning vehicle under the improved pure pursuit algorithm.

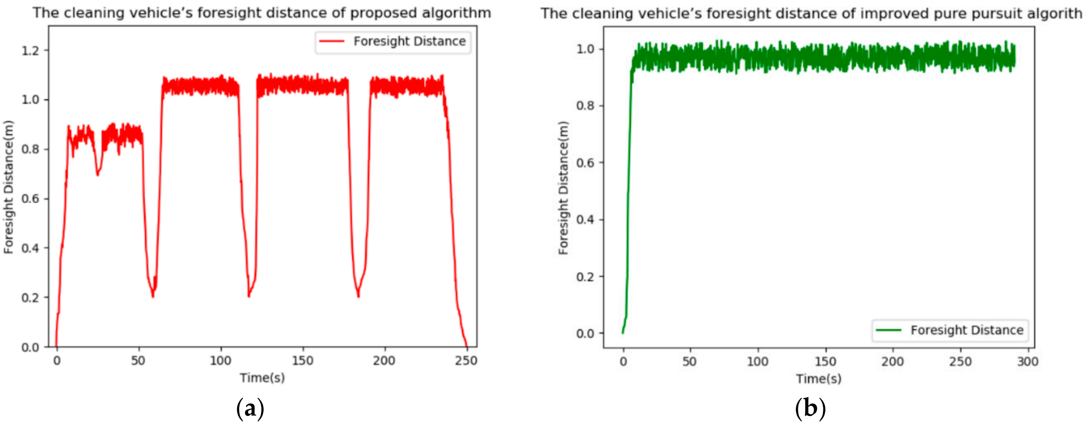


Figure 12. (**a**) The cleaning vehicle's foresight distance under the proposed algorithm. (**b**) The cleaning vehicle's foresight distance under the improved pure pursuit algorithm.

5. Conclusions

Aiming at the influence of the foresight distance, driving speed and yaw rate for autonomous vehicles on the target point tracking control, a tracking controller based on the LSTM network is proposed in this paper. The foresight distance is obtained by the fuzzy control algorithm, whose inputs are the present velocity, yaw rate of the vehicle and output is an appropriate foresight distance. This distance is an important value with reference to the key parameter in pure pursuit algorithm. The two-layer LSTM network controller's function is to regulate the autonomous vehicle's speed and yaw rate simultaneously and intelligently to arrive at the target point, which is gained by a geometrical calculation based on the foresight distance. It is obvious that the foresight distance, driving velocity and yaw rate are interrelated. This method avoids the complicated process of traditional calculation and reduces the tracking error of point sequence tracking significantly.

The simulation experiments are carried out under the conditions of straight path as well as multiple curvature path, its results show that the proposed algorithm can adaptively adjust vehicle's control parameters on the corners or straight segments to track different reference paths quickly and effectively. In order to verify the actual effectiveness of this algorithm, it is also applied to the tracking control of the cleaning vehicle in a subway station. The cleaning vehicle has a precise route and the swinging amplitude of the wheels is moderate during the running process in field experiment. Besides, the proposed controller is freedom from the impacts of path noise points, this phenomenon means that the proposed method can generate better speed and yaw rate for the path tracking in a broader field. Therefore, the proposed algorithm can not only reduce the labor costs, but also greatly improve the work efficiency and quality of the cleaning vehicle. It provides a reference for the research of target points tracking control of autonomous vehicles.

Author Contributions: Conceptualization and Methodology, X.C. and H.W.; Validation, X.C.; Experiments manipulation, X.C. and Z.M.; Writing—Original Draft Preparation, X.C.; Writing—review and editing, X.C., H.W. and Z.M.

Funding: This work was supported by the National Natural Science Foundation of China (No. 51375293, No. 31570998).

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

References

1. Huang, W.; Wei, Y.; Guo, J.H.; Cao, J.D. Next-generation innovation and development of intelligent transportation system in China. *Sci. China Inf. Sci.* **2017**, *60*, 66–76. [CrossRef]
2. Lee, S.; Cho, Y.; Myung, H.; You, B.; Oh, S. A Stable Target-Tracking Control for Unicycle Mobile Robots. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots & Systems, Takamatsu, Japan, 31 October–5 November 2010; pp. 1822–1827.

3. Kim, T.U.; Lee, J.W.; Yang, S.Y. Study on development of autonomous vehicle using embedded control board. In Proceedings of the 11th International Forum on Strategic Technology (IFOST), Novosibirsk, Russia, 1–3 June 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 599–603.
4. Li, X.; Sun, Z.; Cao, D.; Liu, D.; He, H. Development of a new integrated local trajectory planning and tracking control framework for autonomous ground vehicles. *Mech. Syst. Signal Process.* **2017**, *87*, 118–137. [CrossRef]
5. Schwarting, W.; Alonso-Mora, J.; Rus, D. Planning and Decision-Making for Autonomous Vehicles. *Annu. Rev. Control Robot. Auton. Syst.* **2018**, *1*, 187–210. [CrossRef]
6. Amer, N.H.; Zamzuri, H.; Hudha, K.; Kadir, Z.A. Modelling and Control Strategies in Path Tracking Control for Autonomous Ground Vehicles: A Review of State of the Art and Challenges. *J. Intell. Robot. Syst.* **2017**, *86*, 225–254. [CrossRef]
7. Filho, C.M.; Wolf, D.F.; Grassi, V.; Osório, F.S. Longitudinal and lateral control for autonomous ground vehicles. In Proceedings of the 2014 IEEE Intelligent Vehicles Symposium (IV), Dearborn, MI, USA, 8–11 June 2014; pp. 588–593.
8. Hu, C.; Wang, R.; Yan, F.; Chadli, M. Composite Nonlinear Feedback Control for Path Following of Four-Wheel Independently Actuated Autonomous Ground Vehicles. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 2063–2074.
9. Prassler, E.; Ritter, A.; Schaeffer, C.; Fiorini, P. A Short History of Cleaning Robots. *Auton. Robot.* **2000**, *9*, 211–226. [CrossRef]
10. Attia, R.; Orjuela, R.; Basset, M. Combined longitudinal and lateral control for automated vehicle guidance. *Veh. Syst. Dyn.* **2014**, *52*, 261–279. [CrossRef]
11. Normey-Rico, J.E.; Ismael, A.; Gómez-Ortega, J.; Camacho, E.F. Mobile robot path tracking using a robust PID controller. *Control Eng. Pract.* **2001**, *9*, 1209–1214. [CrossRef]
12. Begovich, O.; Sanchez, E.N.; Maldonado, M. Takagi-Sugeno fuzzy scheme for real-time trajectory tracking of an under actuated robot. *IEEE Trans. Control Syst. Technol.* **2002**, *10*, 14–20. [CrossRef]
13. Liang, Y.; Xu, L.; Wei, R.; Hu, H. Adaptive Fuzzy Control for Trajectory Tracking of Mobile Robot. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots & Systems, Taipei, Taiwan, 18–22 October 2010; pp. 4755–4760.
14. Hemami, A.; Mehrabi, M.G.; Cheng, R.M. Optimal kinematic path tracking control of mobile robots with front steering. *Robotica* **1994**, *12*, 563–568. [CrossRef]
15. Morales, J.; Martínez, J.L.; Martníez, M.A.; Mandow, A. Pure-Pursuit Reactive Path Tracking for Nonholonomic Mobile Robots with a 2D Laser Scanner. *Eurasip J. Adv. Signal Process.* **2009**, *1*, 1–10. [CrossRef]
16. Zhu, Q.; Huang, Z.; Liu, D.; Dai, B. An adaptive path tracking method for autonomous land vehicle based on neural dynamic programming. In Proceedings of the International Conference on Mechatronics and Automation, Harbin, China, 7–10 August 2016; pp. 1429–1434.
17. Yang, H.; Liu, J. An adaptive RBF neural network control method for a class of nonlinear systems. *IEEE/CAA J. Autom. Sin.* **2018**, *5*, 457–462. [CrossRef]
18. Hornic, K. Multilayer feed forward networks are universal approximators. *Neural Netw.* **1989**, *2*, 359–366. [CrossRef]
19. Huang, H.; Zhou, J.; Di, Q.; Zhou, J.; Li, J. Robust neural network-based tracking control and stabilization of a wheeled mobile robot with input saturation. *Int. J. Robust Nonlinear Control* **2019**, *29*, 375–392. [CrossRef]
20. Li, Q.; Qian, J.; Zhu, Z.; Bao, X.; Helwa, M.K.; Schoellig, A.P. Deep Neural Networks for Improved, Impromptu Trajectory Trackingof Quadrotors. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 5183–5189.
21. Ailon, A.; Berman, N.; Arogeti, S. On controllability and trajectory tracking of a kinematic vehicle model. *Automatica* **2005**, *41*, 889–896. [CrossRef]
22. Ollero, A.; Heredia, G. Stability analysis of mobile robot path tracking. In Proceedings of the International Conference on Intelligent Robots and Systems, Pittsburgh, PA, USA, 5–9 August 1995; IEEE: Piscataway, NJ, USA, 2002; Volume 3, pp. 461–466.
23. Tang, X.; Tao, J.; Li, Z. Stability Improvement of Pure Pursuit Model Based on Fuzzy Control. *Trans. Chin. Soc. Agric. Mach.* **2017**, *12*, 1–8.
24. Hochreiter, S.; Schmidhuber, L.J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]

25. Zhao, J.; Zhang, X.; Shi, P.; Liu, Y. Automatic Driving Control Method Based on Time Delay Dynamic Prediction. *J. Tsinghua Univ. Sci. Technol.* **2016**, *58*, 432–437.
26. Park, M.W.; Lee, S.W.; Han, W.Y. Development of lateral control system for autonomous vehicle based on adaptive pure pursuit algorithm. In Proceedings of the International Conference on Control, Automation and Systems, Jeju, Korea, 18–21 October 2017; pp. 1443–1447.
27. Zhang, W.Y.; Ding, Y.C.; Wang, X.L.; Zhang, X.; Cai, X.; Liao, Q.X. Pure Pursuit Control Method Based on SVR Inverse-model for Tractor Navigation. *Trans. Chin. Soc. Agric.* **2016**, *47*, 29–36.
28. Duan, J.; Yang, C.; Shi, H. Path tracking based on pure pursuit algorithm for intelligent vehicles. *J. Beijing Univ. Technol.* **2016**, *42*, 1301–1306.
29. Noda, K.; Arie, H.; Suga, Y.; Ogata, T. Multimodal integration learning of robot behavior using deep neural networks. *Robot. Auton. Syst.* **2014**, *62*, 721–736. [CrossRef]
30. Xiang, G.; Tao, Z. Unsupervised learning to detect loops using deep neural networks for visual SLAM system. *Auton. Robot.* **2017**, *41*, 1–18.
31. Luca, A.; Samson, C. Feedback Control of a Nonholonomic Car-Like Robot. *Lect. Notes Control Inf. Sci.* **1998**, *229*, 171–253.