



# Article Active Obstacle Avoidance Trajectory Planning for Vehicles Based on Obstacle Potential Field and MPC in V2P Scenario

Ruoyu Pan <sup>1</sup>, Lihua Jie <sup>1</sup>, Xinyu Zhao <sup>1</sup>, Honggang Wang <sup>1</sup>, Jingfeng Yang <sup>2</sup> and Jiwei Song <sup>3,\*</sup>

- School of Communications and Information Engineering and School of Artificial Intelligence, Xi'an University of Posts and Telecommunications, Xi'an 710121, China
- <sup>2</sup> Guangzhou Institute of Industrial Intelligence, Guangzhou 511458, China
- <sup>3</sup> China Electronics Standardization Institute, Beijing 100007, China
- \* Correspondence: songjw@cesi.cn

Abstract: V2P (vehicle-to-pedestrian) communication can improve road traffic efficiency, solve traffic congestion, and improve traffic safety. It is an important direction for the development of smart transportation in the future. Existing V2P communication systems are limited to the early warning of vehicles and pedestrians, and do not plan the trajectory of vehicles to achieve active collision avoidance. In order to reduce the adverse effects on vehicle comfort and economy caused by switching the "stop-go" state, this paper uses a PF (particle filter) to preprocess GPS (Global Positioning System) data to solve the problem of poor positioning accuracy. An obstacle avoidance trajectory-planning algorithm that meets the needs of vehicle path planning is proposed, which considers the constraints of the road environment and pedestrian travel. The algorithm improves the obstacle repulsion model of the artificial potential field method, and combines it with the A\* algorithm and model predictive control. At the same time, it controls the input and output based on the artificial potential field method and vehicle motion constraints, so as to obtain the planned trajectory of the vehicle's active obstacle avoidance. The test results show that the vehicle trajectory planned by the algorithm is relatively smooth, and the acceleration and steering angle change ranges are small. Based on ensuring safety, stability, and comfort in vehicle driving, this trajectory can effectively prevent collisions between vehicles and pedestrians and improve traffic efficiency.

Keywords: V2P; artificial potential field; A\*; MPC (model predictive control)

# 1. Introduction

In recent years, autonomous driving technology has undergone considerable improvement. Academic and industrial circles around the world are actively promoting practical commercial applications for this technology, injecting new vitality into the research of autonomous driving technology [1]. Self-driving cars rely on advanced intelligent control vehicle technology to reduce traffic accidents caused by driver errors, and even reduce the incidence of traffic accidents to zero. Autonomous driving technology must ensure accurate avoidance of pedestrians, so V2P is one of the core technologies in autonomous driving [2–4]. It notifies both people and vehicles by analyzing the key information of vehicles and pedestrians; this can prevent collisions between people and vehicles and effectively improve road traffic efficiency, in order to achieve intelligent transportation solutions for vehicle–road coordination.

V2P usually predicts whether there is a risk of collision between people and vehicles, and judges whether an alarm needs to be issued to warn pedestrians and vehicles. A V2X (vehicle-to-everything) collaborative system based on cellular and 802.11p radio is designed in [5], and the SafeNav Android application is proposed, in which the interface sets a determined collision area, and if multiple traffic participants enter the collision area, the color of the traffic participant changes to red and it produces a visual and audible alarm. In [3], a mobile application is proposed by Hussein et al. that supports pedestrians



Citation: Pan, R.; Jie, L.; Zhao, X.; Wang, H.; Yang, J.; Song, J. Active Obstacle Avoidance Trajectory Planning for Vehicles Based on Obstacle Potential Field and MPC in V2P Scenario. *Sensors* **2023**, *23*, 3248. https://doi.org/10.3390/s23063248

Academic Editor: Francisco J. Martinez

Received: 31 January 2023 Revised: 26 February 2023 Accepted: 10 March 2023 Published: 19 March 2023



**Copyright:** © 2023 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/). and vehicles. When the user interacts with their mobile phone, the screen is activated to detect the location coordinates of the mobile phone. The system calculates the potential collision point and judges whether collision is possible. This application, with its user-friendly operation, increases VRU (Vulnerable Road User)s' visual situational awareness of locations near automatic- or manual-control vehicles. The above studies all needed to predict the degree of risk, and the vehicles needed to switch frequently to the "stop-go" state. The vehicles did not actively avoid pedestrians, and the degree of intelligence was not high, which may bring certain safety risks [6]. Trajectory planning can realize active obstacle avoidance for V2P vehicles [7]. In order to safely and quickly move a vehicle from its current location to the target location, it is necessary to design a robust trajectory planning strategy.

At present, classic trajectory planning includes rule-based methods [8,9], graph search methods [10–12], and numerical optimization methods [13]. The rule-based method aims to establish a behavior rule base based on vehicle driving rules and experience information, divide the state according to different environmental information, and determine the vehicle driving trajectory according to the rule logic. In [14], an autonomous vehicle control system based on the FSM (finite state machine) rule-based method was proposed by Sang-Hyeon Bae et al. The proposed vehicle system can handle six major urban environment driving situations. The system actively recognizes the real-time driving environment, formulates an action plan based on the finite state machine, and activates motion model control. The rule-based method can effectively deal with some typical driving scenarios defined in the rule base, but due to the limited number of added driving scenarios, it cannot handle driving scenarios that have not been added to the rule base; thus, it has certain limitations.

To solve the problem of limited driving scenarios, the graph search method is used for trajectory planning. Dijkstra, BFS (Best First Search), and A\* are common graph search planning algorithms [15]. Dijkstra's algorithm starts from the initial point where the object is located, visits the adjacent nodes of the initial point in the graph until it reaches the target point, and finally, outputs the shortest path; however, the algorithm runs for a long time and is not in real time. The BFS algorithm quickly guides the target point by introducing a heuristic function, thereby effectively improving the running speed of the algorithm. The heuristic function evaluates the cost from any node to the target node, and selects the node closest to the target instead of the node closest to the initial point. Therefore, there is no guarantee that the shortest path will be found. The A\* algorithm combines the advantages of the heuristic algorithm, BFS, and the Dijkstra algorithm; not only can it find the shortest path, but in a simple static road environment, the algorithm running time is also as fast as BFS. This, it can meet the real-time requirements of vehicle trajectory planning [16]. The graph search method is suitable for situations where obstacles are stationary, while pedestrians cannot be considered static obstacles, similarly to roads, so they cannot be directly applied to V2P scenarios.

The artificial potential field method has become a classic algorithm in numerical optimization methods due to its simple principle, its small amount of calculation, and its high execution efficiency, and is widely used in robot trajectory planning [17]. The artificial potential field method has a small amount of calculation and fast calculation speed; thus, it meets the high real-time requirements of vehicle trajectory planning, and supports the independent design of the attraction potential field generated by the target position, and the repulsive potential field generated by obstacles, to meet the requirements of different scenarios. In [18], Lu et al. proposed a method to find a path by fitting the curve of the target point and the critical oscillation point, reducing the path jitter, and making the path smoother. Virtual repulsion was added by Azzabi A et al. [19] to improve the repulsion potential field function and solve the local optimum of the artificial potential field. However, since the direction of virtual repulsion is uncertain, it may cause new problems. The application of the above-mentioned artificial potential field method does not consider the vehicle's motion constraints, so it cannot be directly applied to the vehicle trajectory planning problem.

Due to its ability to deal with system multi-constraint problems, the model predictive control (MPC) algorithm in numerical optimization is widely used in the field of automatic driving [20]. MPC establishes a mathematical model with a cost function and constraints. Constrained optimization problems can be solved using linear programming methods according to the cost and the complexity of the constraint function. The advantage is that the optimal trajectory is not constrained by a predefined pattern. Cost and constraint functions adapt to the driving environment with road geometry, boundaries, and obstacles. This adaptability makes the final trajectory highly adaptable to changes in the environment. A solution proposed by Mekala et al. was to use MPC to control speed for obstacle avoidance, so that the vehicle can accelerate and decelerate smoothly, but obstacles are represented as hard constraints in the optimization process; this may lead to the non-existence of feasible solutions in practice [21]. An iterative nonlinear model predictive control path planner using a point-mass vehicle model was proposed by Murillo et al., which considered vehicle dynamic constraints but not road boundaries [22].

This paper proposes an obstacle avoidance trajectory planning algorithm that meets the needs of vehicle path planning. First, pedestrians are ignored and only road boundaries are considered, and the A\* algorithm is used to give the shortest path from the initial point of the vehicle to the target point as a reference path. Secondly, pedestrians are regarded as dynamic obstacles, and a linear prediction model is established on the basis of the vehicle model, which is used to predict the future state of the vehicle. Based on the artificial potential field method and the vehicle motion constraints, the input constraints and output variable constraints are jointly controlled to plan the vehicle's trajectory. Finally, the trajectory planning algorithm proposed in this paper is verified via test verification.

The rest of this paper is organized as follows. Section 2 describes the preprocessing process of GPS data information. Section 3 introduces the proposed trajectory planning algorithm and gives the algorithmic framework to safely and efficiently solve the active obstacle avoidance problem in vehicles. Section 4 outlines the test verification and analyzes the results of the algorithm to verify the validity of the algorithm. Finally, the conclusions are given in Section 5.

### 2. Data Preprocessing

Due to the influence of noise in the transmission process, the GPS position information obtained by the actual system has incompleteness and uncertainty, which affects the accuracy of the position information. For noisy data, filters are usually used for preprocessing. The Kalman filter is only suitable for linear systems, and the actual system had different degrees of nonlinearity, so the nonlinear Kalman filter can be better for application in the filtering process of GPS data. In nonlinear filtering, the model is divided into a state model and a measurement model.

### 2.1. State Model

The basic state vector of the state model includes the GPS position coordinates, the velocity, and the error caused by noise in the propagation process. The first-order Gauss-Markov process was used to model the error as non-white noise. The state vector of the system can be given by expression (1).

$$x_t = [X_t, X_t, Y_t, Y_t, \varepsilon_{1t}, \varepsilon_{2t}, \dots, \varepsilon_{nt}]$$
  

$$x_{t+1} = f(x_t)$$
(1)

where  $(X_t, Y_t)$  is the position coordinate component,  $(X_t, Y_t)$  is the velocity coordinate component, and  $\varepsilon_{it}$  (i = 1.2..., n) is the error caused by the *n*-th noise source in the receiving process, so the state model of the system is obtained as follows:

$$X_{t+1} = X_t + T\dot{X}_t + \frac{T^2}{2}\delta\ddot{X}_t$$
  

$$\dot{X}_{t+1} = \dot{X}_t + T\delta\ddot{X}_t$$
  

$$Y_{t+1} = Y_t + T\dot{Y}_t + \frac{T^2}{2}\delta\ddot{Y}_t$$
  

$$\dot{Y}_{t+1} = \dot{Y}_t + T\delta\ddot{Y}_t$$
(2)

where  $\delta X_t$ ,  $\delta Y_t$  is the interference of other systems, *T* is the sampling period, and the error caused by noise in the propagation process is expressed using the first-order Gauss–Markov process as follows:

$$\begin{aligned}
\varepsilon_{1(t+1)} &= \alpha \varepsilon_{1t} + \beta \sigma_{1t} \\
\varepsilon_{2(t+1)} &= \alpha \varepsilon_{2t} + \beta \sigma_{2t} \\
&\vdots \\
\varepsilon_{n(t+1)} &= \alpha \varepsilon_{nt} + \beta \sigma_{nt}
\end{aligned} \tag{3}$$

 $\sigma_{nt}$  is the error brought by the *n*-th noise source in the propagation process. This article assumes that n = 2, and that these noises are non-correlated random white noise with a mean value of 0; the parameters  $\alpha$ ,  $\beta$  are, respectively,

$$\alpha = \frac{2\tau - 1}{2\tau + 1}, \beta = \frac{2\tau}{2\tau + 1}$$

where  $\tau = 100$ .

### 2.2. Measurement Model

Using Z(t) to represent the observed value of GPS position information at the time t, the measurement model is expressed as:

$$Z_t = Hx_t + V_t \tag{4}$$

$$Z_{1t} = X_t + \varepsilon_{1t} + V(t)$$
  

$$Z_{2t} = Y_t + \varepsilon_{2t} + V(t)$$
(5)

In the formula, it is assumed that V(t) is Gaussian white noise with variance  $\sigma_v^2$  and a mean value of 0.

The system is discretized, and the obtained state model and measurement model are as follows:

$$x_{k+1} = f(x_k)$$

$$Z_k = Hx_k + V_k$$
(6)

where the subscript *k* represents the time.

# 3. Trajectory Planning

Trajectory planning was divided into path planning and speed planning. First, under the environment in which obstacles are ignored, the A\* algorithm was used to construct the drivable area, and initial global planning of the path from the initial point to the target point was carried out to produce a reference path. Secondly, the artificial potential field method and MPC were combined for trajectory planning.

# 3.1. A\* Path Planning

The Dijkstra, Best First Search (BFS), and A\* algorithms are three typical path planning algorithms. Dijkstra's algorithm is guaranteed to find the shortest path from the initial point to the goal point, but it runs slowly [23,24]. BFS runs faster than Dijkstra's algorithm, but it cannot be guaranteed to find the shortest path [25]. The A\* algorithm is a heuristic search algorithm that can perform global path planning in a static environment according to a defined evaluation function [17]. It responds quickly to the environment and searches the path directly, so it is widely used in path planning research. This algorithm can greatly

reduce search time and improve path search efficiency while keeping the path as short as possible. Since the static road environment where the vehicle is driving is relatively simple, in simple cases, the A\* algorithm runs as fast as BFS. Therefore, this paper uses the A\* algorithm to plan the global path and generate a vehicle reference path.

The heuristic function of the A\* algorithm contains information on the initial point and the target point at the same time, and the priority of each node is calculated using Formula (7):

$$f(n) = g(n) + h(n) \tag{7}$$

where f(n) is the comprehensive priority of the node n. When selecting the next node to traverse, the node with the smallest f(n) value and the highest priority will be selected. g(n) is the cost of the node n from the initial point. h(n) is the heuristic function of the A\* algorithm, and is the estimated cost of the node distance from the target point. This paper allowed the vehicle to move in eight directions (namely: front, rear, left, right, right front, right rear, left rear, and left front, corresponding to g(n) values of 1, 1, 1, 1, 1.5, 1.5, 1.5, and 1.5), and the Euclidean distance was chosen as the heuristic function.

# 3.2. Vehicle and Pedestrian Models

When using MPC for local trajectory planning, the motion states of vehicles and pedestrians are required. Therefore, the monorail vehicle kinematics model and the pedestrian (dynamic obstacle) model were established first, and were used to describe the state of the vehicle and pedestrian, respectively.

### 3.2.1. Vehicle Model

This paper mainly considers the plane motion of the vehicle, and used a single-track model to describe the vehicle, taking the rear wheel of the vehicle as a reference point, as shown in Figure 1.  $\theta$  represents the direction of the speed of the vehicle, *l* represents the wheelbase, *v* represents the vehicle's current speed, and  $\delta$  represents the angle between the two speed directions at the front and rear of the vehicle.



Figure 1. Schematic diagram of vehicle monorail model.

The updated formula of each state quantity in the model is as follows:

$$x(t+dt) = x(t) + v(t)\cos(\theta)dt$$
  

$$y(t+dt) = y(t) + v(t)\sin(\theta)dt$$
  

$$\theta(t+dt) = \theta(t) + \frac{v}{l}\tan(\delta)dt$$
  

$$v(t+dt) = v(t) + adt$$
  
(8)

where *x* and *y* are the longitudinal and lateral positions of the vehicle, and *a* is the acceleration of the vehicle.

# 6 of 15

### 3.2.2. Pedestrian Model

The basic aim of the traditional artificial potential field method is to control the robot to find a collision-free path by constructing an artificial force field based on the attraction field of the target point and the repulsive force fields of the obstacles, and using the falling direction of the search potential function. The path planning of the robot does not need to consider the boundary and road environment, but for the vehicle, it needs to consider the constraints of the road, so the model construction is more complicated.

During the driving process of the vehicle, the most common dynamic obstacle is the pedestrian. In this paper, the pedestrian is abstracted as an obstacle point. Since the pedestrian is a dynamic obstacle, the position and velocity of the pedestrian should also be taken into account in the obstacle potential field. According to the space dynamics equation and the Lagrangian equation, the resultant force of the vehicle is the vector superposition of the repulsive force and the attraction force, and the direction of the resultant force is the moving direction of the vehicle. Under the action of the resultant force, the vehicle can bypass obstacles and reach the end point. The resultant force *F* can be expressed as:

$$F = F_a + F_r \tag{9}$$

where  $F_a$  is the attraction force generated by the target point on the controlled object, and  $F_r$  is the repulsive force generated by the obstacle on the controlled object. Assuming there are *n* obstacle points, the attraction function  $F_a$  and repulsion function  $F_r$  are redefined as:

ŀ

$$F_a = \frac{1}{2} \times K_a \times \sqrt{(x - g_x)^2 + (y - g_y)^2}$$
(10)

$$F_r = \begin{cases} \frac{1}{2} \times K_r \times e^{-\sum_{i=1}^n \sqrt{(x - O_{x_i})^2 + (y - O_{y_i})^2}}, 0 < d < d_r \\ 0, d > d_r \end{cases}, 0 < d < d_r \end{cases}$$
(11)

where (x, y) are the position coordinates of the vehicle at any time,  $(g_x, g_y)$  are the coordinates of the end position of the vehicle,  $K_a$  is the gravitational potential energy gain coefficient, and d is the Euclidean distance between the vehicle and the nearest obstacle.  $d_r$  is the radius of the obstacle repulsion field. When the distance between the vehicle and the obstacle is less than  $d_r$ , the vehicle will be affected by the repulsion.  $K_r$  is the repulsive potential energy gain coefficient,  $(O_{x_i}, O_{y_i})$  are the obstacle coordinates, and  $O_{x_i} = [x_i, v_{x_i}]$ ,  $O_{y_i} = [x_i, v_{y_i}]$  is the combination of position and velocity.

### 3.3. MPC Trajectory Planning

The MPC method is a numerical optimization method. According to the model and the current state quantity output, the deviation between the predicted trajectory and the expected trajectory is calculated, and control quantity input and output constraints are imposed to ensure that the vehicle can meet the corresponding motion constraints and avoid collisions. In the trajectory planning process, the future behavior of the vehicle needs to be predicted within the specified forecast horizon, and the control input at the next moment is calculated by minimizing the error between the predictor and the reference point under various constraints. On the basis of Formula (8), the linear prediction model of the current sampling moment is established as follows:

$$\dot{x}(t+dt) = v\cos(\phi) - v\theta\sin(\phi)$$

$$\dot{y}(t+dt) = v\theta\cos(\phi) + v\sin(\phi)$$

$$\dot{\phi}(t+dt) = \frac{v\tan(\delta)}{l} + \frac{v\delta\tan(\delta^{2}+1)}{l}$$

$$\phi = \theta + \delta$$
(12)

Since the controlled system in MPC usually adopts a discrete state space model, it is necessary to establish a state space expression under a linear discrete time system, and expand the state space variable  $x_s(k)$  and the control variable u(k) into a new state variable  $x_s(k+1)$ . The new state space expression is as follows:

$$x_s(k+1) = Ax_s(k) + Bu(k) + C$$
  

$$y_o(k) = Dx(k)$$
(13)

where  $x_s(k) \in \mathbb{R}^4$ ,  $u(k) \in \mathbb{R}^2$ , and  $y_o(k) \in \mathbb{R}^2$  are output variables.  $x_s(k), y_o(k)$  are all related to the four identical variables, and Formula (14) lists the expression for  $x_s(k)$ .

$$x_{s}(k) = [x(k), y(k), \phi(k), v(k)]'$$

$$u(k) = [a(k), \delta(k)]'$$

$$\dot{v}(k) = a(k)$$
(14)

$$A = \begin{bmatrix} 1 & 0 & -v\sin(\theta) & \cos(\theta) \\ 0 & 1 & v\cos(\theta) & \sin(\theta) \\ 0 & 0 & 1 & \frac{\tan(\delta)}{l} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & \frac{v}{l\cos^{2}(\delta)} \\ 1 & 0 \end{bmatrix}$$

$$C = \begin{bmatrix} v\theta \sin(\theta) & -v\theta\cos(\theta) & -v\frac{\delta}{l\cos^{2}(\delta)} & 0 \end{bmatrix}$$

$$D = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
(15)

Based on the state space model for predicting the future dynamics of the system, Formula (13) is rewritten as an incremental model as follows:

$$\Delta x_s(k+1) = A \Delta x_s(k) + B \Delta u(k)$$
  

$$\Delta y_o(k) = C \Delta x(k)$$
(16)

where

$$\Delta x_s(k) = x_s(k) - x_s(k-1)$$
  

$$\Delta u(k) = u(k) - u(k-1)$$
(17)

The prediction range of the MPC-based upper decision controller is set to  $N_p$  and the control range is set to  $N_c$ . In addition,  $N_p < N_c$ . Assuming that the current moment is k, k > 0. Using the current state information to define the input vector and predicted output vector of the system in the future  $N_c$ , the steps are as follows:

$$U(k) \stackrel{def}{=} \begin{bmatrix} u(k) \\ u(k+1) \\ \dots \\ u(k+N_c-1) \end{bmatrix}$$
(18)

$$Y_{N_{p}}(k+1 \mid k) \stackrel{def}{=} \begin{bmatrix} y_{o}(k+1 \mid k) \\ y_{o}(k+2 \mid k) \\ \dots \\ y_{o}(k+N_{p} \mid k) \end{bmatrix}$$
(19)

The output vector prediction equation for the next  $N_p$  steps is as follows:

$$Y_{N_n}(k+1|k) = S_x x_s(k) + S_u U(k)$$
(20)

where

$$S_x = [CA, CA^2 \dots CA^{N_p}] \tag{21}$$

$$S_{u} = \begin{bmatrix} CB & 0 & \cdots & 0 \\ CAB & CB & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ CA^{N_{p}-1}B & CA^{N_{p}-2}B & \cdots & \sum_{i=1}^{N_{p}-N_{c}+1}CA^{i}B \end{bmatrix}$$
(22)

In order to ensure that the predicted output variable is as close as possible to the reference trajectory, that is, that the vehicle uses the shortest path as much as possible, the objective function for constructing the optimization solution is as follows:

$$\min J = K \sum_{k=0}^{N_p} \| Y_{N_p}(k+1 \mid k) - R(k+1) \|_Q^2 + K_u \| \Delta U(k) \|_R^2 + \sum_{k=0}^{N_p} F_r(k+1|k) + K_h h^2$$
(23)

In the formula,  $K_h$  is the weight coefficient, h is the relaxation factor, Q, R are the weighted matrices of the output error and control input, respectively, and R(k + 1) is the reference path, which is the path obtained by the A algorithm. The first item of the objective function represents the planning trajectory, and the reference trajectory should be as close as possible to ensure the shortest planned trajectory. The second item aims to control the incremental size to ensure that there will be no drastic changes in speed, heading angle, or acceleration during the operation of the vehicle, and to ensure the stability and comfort of the vehicle to a certain extent. The third item represents the potential energy value of the vehicle in the obstacle potential field, which ensures that the vehicle can effectively avoid dynamic obstacles. The fourth term is the relaxation factor, which can enhance the solution of the feasible domain, so as to ensure that there is an optimal solution to the programming problem.

To ensure the safety of vehicle obstacle avoidance and reduce the influence of the maneuvering process on the comfort of the vehicle, the control process needs to take into account the corresponding constraints. Through these constraints, the vehicle can avoid accelerating or decelerating too quickly, thereby ensuring driving safety and comfort, and the control input must meet the following constraints [26,27]:

$$\begin{cases} -8 \text{ m/s}^2 \le a \le 3 \text{ m/s}^2, \text{ the pedestrian was not detected} \\ -2 \text{ m/s}^2 \le a \le 1 \text{ m/s}^2, \text{ the pedestrian was detected} \\ -\pi/4 \le \delta \le \pi/4 \\ 0 \text{ m/s} \le v \le 16.7 \text{ m/s} \\ v_{ref} = 12 \text{ m/s} \end{cases}$$
(24)

where  $v_{ref}$  is the best reference speed. Speeds of 12 m/s and 16 m/s were chosen because they correspond to typical speeds of 40 m/h and 60 m/h.

### 3.4. Algorithm Framework Description

Figure 2 describes the proposed algorithm framework. The algorithm takes the road environment, pedestrian trajectory, and vehicle running state as inputs, and combines the

A\* algorithm, the artificial potential field method, and MPC trajectory planning to finally output the vehicle control state and guide the vehicle to avoid obstacles. First, according to the initial point and target point of the vehicle, combined with the road environment, the A\* algorithm is used to plan a global path, which will function as the reference path. Secondly, based on the global path, PF processes the position information of pedestrians and vehicles, and uses the artificial potential field method and MPC algorithm to plan a trajectory that satisfies the obstacle model and vehicle motion constraints.



Figure 2. The framework of the trajectory planning algorithm system proposed in this paper.

# 4. Test Verification and Analysis of Results

# 4.1. Test Environment

The test required pedestrians and vehicles to be equipped with GPS to determine their position, speed, and direction. The pedestrian terminal included GPS and LoRa modules, and the GPS working frequency was 1 HZ. MCU (Microcontroller Unit) adopted an STM32 chip, and the radio frequency module was the LoRa RFM98 chip.

The test used the LoRa wireless transmission system, which consisted of gateways and terminals. The gateway was responsible for receiving and processing messages sent by the terminal. Figure 3 presents a hardware frame diagram and hardware physical diagram of the pedestrian terminal, in which the GPS module and the MCU perform one-way communication through the USART (Universal Synchronous/Asynchronous Receiver/Transmitter), and the LoRa module performs two-way communication with the MCU through the SPI (Serial Peripheral Interface) bus. Figure 4 is a hardware frame diagram of the gateway and its corresponding physical diagram. MT7620A is the main control chip of the gateway. The chip can meet the design requirements of the smart gateway in terms of performance, and its price is lower than that of similar chips; thus, it conforms to the low cost requirements of the LoRa network. It consists of four groups of the same modules, which can support simultaneous communication with different frequency bandwidths. Among them, LoRa communicates with the MCU through the SPI bus in a two-way manner. The gateway software runs on the OpenWRT operating system. The LoRa radio frequency module and its MCU in the gateway are the same as those of the terminal. The gateway uses a POE power supply, and an RGB indicator light indicates its working status. It possesses a single debugging serial port, an Ethernet interface LAN/WAN, and 4G. The module interface has dial-up Internet access.



**Figure 3.** Pedestrian terminal (the hardware frame diagram is on the **left**, and the physical hardware diagram is on the **right**).



**Figure 4.** Gateway (the hardware frame diagram is on the **left** and the physical hardware diagram is on the **right**).

The gateway was located on the roof of the No. 3 experimental building (15 m in height) of Xi'an University of Posts and Telecommunications. Pedestrians wearing V2P pedestrian terminals walked at a constant speed outside the teaching building of the Chang'an Campus of Xi'an University of Posts and Telecommunications.

### 4.2. Analysis of Results

Using the state model and measurement model in Section 2.1, pedestrian terminals and Superstar GPS OEM GPS receivers were used to collect pedestrian position information at the same time. The sampling interval was 1 s. The experiment collected 1100 sets of data, which were divided into eleven groups. UKF and PFs were used to filter The GPS position information, and the number of particles selected by the PFs was 100. A group was randomly selected, and the trajectory and filtering errors obtained are shown in Figure 5; the experimental results of the other ten groups are consistent with this group.

EKF, UKF, and PF are commonly used filters in nonlinear systems. EKF revolves around its current state through Taylor expansion, discarding high-order terms, linearizing the system under study, and solving it as a linearized system; the result has large deviation. The UKF algorithm uses UT transformation to obtain a Sigma point set, and uses a small number of points to approximate the state distribution, which better describes the nonlinear system, so the filter converges to the correct solution more quickly. However, UKF is only applicable to the standard Kalman filter system under linear assumption (under the linear and Gaussian assumption of the system). When the linearity of the system is not high, the effect of UKF filtering is not good. PF is a parameterless filter that does not have a specific function that enables it to obtain posterior results, and the Monte Carlo method generates a large number of random particles to approximate the posterior distribution of the state and achieve state estimation. In the process of prediction and updating, the PF algorithm only updates the particles in it, and then, obtains a state estimation via weighting and

summing. It does not need to calculate the posterior covariance of the state, so it can approximate any system's state distribution. In this paper, UKF and PF were selected to filter the GPS position information. It can be seen from Figure 5 that the trajectory filtered by PF is closer to the real trajectory. For stability, therefore, the PF filter was used to preprocess the data and improve the positioning accuracy.



**Figure 5.** Analysis of filtering error of PF and UKF algorithms (the **left** picture shows the trajectory generated via filtering after collecting GPS position information, and the track generated by the actual position information. The **right** picture shows the error between the data processed by the filter and the real data).

To verify the effectiveness of the proposed trajectory planning algorithm, the algorithm was tested and verified in this paper. Two pedestrians wearing pedestrian terminals kept moving forward at a constant speed, and the pedestrian speed was 1 m/s. After collecting the GPS position information of pedestrians, this paper modeled the pedestrian driving environment, assuming that the vehicle length was 4.5 m, the width was 2 m, the tire diameter was 0.6 m, the width was 0.25 m, and the front and rear wheelbases, and the rear wheelbases, were 1.5 m. As shown in Figure 6, the vehicle was modeled to a given size. The initial point was (10,2), the target point was (35,47), the best reference speed was given as 12 m/s, the pedestrian was located outside the line of sight of the driver, and the trajectory was uncertain. (0,10). The initial points of the two pedestrians were (0,20). After preprocessing the GPS data using PF, the positions of the vehicle and the pedestrian were determined and are shown in Figure 6a. The two black rectangles in the figure jointly denote the road boundary, which is regarded as a static obstacle, and pedestrians are dynamic obstacles relative to vehicles.

According to the initial point and target point of the given vehicle, we first used the A\* algorithm to plan an optimal global reference path. The output trajectory of the A\* algorithm is given by the discrete points in the predicted time domain, which are discrete trajectory points. Due to the kinematic constraints of the vehicle, if the vehicle position is required to be continuous, the yaw angle continuity requires the curve to be first-order continuous, and the acceleration constraint requires the curve to be second-order continuous; then, the planned path should satisfy the second-order continuity of the curve. Upon integrating the requirements of path planning and calculation cost, the cubic spline curve was selected to smooth the path. Figure 6b shows the fitting results of the planned path using the cubic spline curve.



**Figure 6.** The trajectory generated by the proposed algorithm (the unit in the figure is m). (**a**) The initial state of the test scene, where the green mark at (35,47) is the target point, and the vehicle is at (10,2) as the initial coordinate point. (0,10) and (0,20) are the initial points of the two pedestrians. (**b**) The reference global path planned using the A\* algorithm. (**c**) The path diagram planned using the proposed algorithm.

After testing and verification, the vehicle could avoid pedestrians, and the trajectory planned by the algorithm is shown in Figure 6c. The blue part is the vehicle trajectory, and the black part is the trajectory of the two pedestrians. The obstacle detection range was set to the front wheel of the vehicle as the reference point, within a circle with a radius of 5 m. The speed, acceleration, and heading angle of the vehicle during obstacle avoidance are shown in Figure 7. The vehicle accelerates as closely as possible to the recommended speed of 12 m/s. It stops accelerating to avoid the first pedestrian at the sixth second, and then, decelerates to avoid the second pedestrian, with complete collision avoidance, at the tenth second; finally, it decelerates within the speed limit to reach the target point. The vehicle heading angle changes smoothly within (-pi/4, pi/8).



**Figure 7.** Changes in vehicle speed, acceleration, and heading angle during the trajectory planning process of the proposed algorithm. (**a**) The variation of velocity. (**b**) The variation of acceleration. (**c**) The variation of heading angle.

In the case of the vehicle's avoidance of pedestrians, the hazard distance between the vehicle and the pedestrian was set to 1.5 m [28]. In this paper, the closest distance between the vehicle and the pedestrian was used as the performance index to evaluate the proposed algorithm. In the process of algorithm execution, the distance between the vehicle and the two pedestrians was counted, and the result is shown in Figure 8; the ordinate is the distance between the vehicle and the pedestrian, the unit is m, and the abscissa is the number of sequences. Among them, the distance between the vehicle and the two pedestrians in the algorithm proposed in this paper was two solid lines. "Distance–1" in the legend indicates the distance between the vehicle and the first pedestrian during the operation of the proposed algorithm; "Distance–2" in the legend indicates the distance between the vehicle and the first pedestrian during the operation of the proposed algorithm; and "Ref. [29]–distance–1" in the legend represents the distance between the vehicle and the first pedestrian during the operation of the proposed algorithm; and "Ref. [29]–distance–1" in the legend represents the distance between the vehicle and the first pedestrian during the operation of the proposed algorithm; and "Ref. [29]–distance–1" in the legend represents the distance between the vehicle and the first pedestrian during the operation of the proposed algorithm; and "Ref. [29]–distance–1" in the legend represents the distance between the vehicle and the first pedestrian during the operation of the proposed algorithm; and "Ref. [29]–distance–1" in the legend represents the distance between the vehicle and the first pedestrian during the operation of the proposed algorithm; and "Ref. [29]–distance–1" in the legend represents the distance between the vehicle and the first pedestrian during the operation of the proposed algorithm; and "Ref. [29]–distance–1" in the legend represents the distance between the vehicle and the first pedestrian during the oper

pedestrian during the operation of the algorithm proposed by Ref. [29]. It can be seen from Figure 8 that the algorithm proposed in this paper has good obstacle avoidance performance, and the closest distance from pedestrians is 2.7 m, while the algorithm proposed by Ref. [29] is within the danger-range distance of pedestrians. Therefore, the algorithm in this paper has better performance and higher safety, and thus, has reference value.



**Figure 8.** The distance between the vehicle and the pedestrian changes during obstacle avoidance. (the ordinate is the distance between the vehicle and the pedestrian, the unit is m, and the abscissa is the number of sequences).

### 5. Conclusions

In order to achieve the goals of vehicles actively avoiding pedestrians, and safely and reliably improving traffic efficiency, this paper proposes a trajectory planning algorithm that combines the A\* algorithm, the artificial potential field method, and MPC; we added the dynamic obstacle potential field into the objective function of the MPC controller, so as to guide the vehicle to avoid obstacles and avoid of human–vehicle collision. The test results show that the planning algorithm proposed in this paper can plan the trajectory of vehicles while keeping a safe distance from pedestrians, to actively avoid collisions and improve traffic safety. The trajectory planned by this algorithm is relatively smooth and conforms to the motion constraints of the vehicle. During vehicle driving, the acceleration change is within the range of  $(-1.6m/s^2, 1.2m/s^2)$ , acceleration during vehicle obstacle avoidance is within the range of  $(-1.6m/s^2, 0.4m/s^2)$ , the heading angle change is in the range of (-pi/4, pi/8), and the vehicle acceleration and steering angle change ranges are small, ensuring steering stability and riding comfort.

**Author Contributions:** Conceptualization, R.P. and L.J.; methodology, H.W. and X.Z.; software, L.J.; validation, L.J.; formal analysis, J.Y. and J.S.; investigation, J.Y. and J.S.; writing—original draft preparation, L.J. and R.P.; writing—review and editing, R.P. and L.J. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the Key Industry Innovation Chain Project of Shaanxi Province (No. 2021ZDLGY07-10, No. 2021ZDLNY03-08), the Science and Technology Plan Project of Shaanxi Province (No. 2022GY-045), the Key Research and Development Plan of Shaanxi Province (No. 2018ZDXM-GY-041), the Scientific Research Program funded by the Shaanxi Provincial Education Department (Program No. 21JC030), the Science and Technology Plan Project of Xi'an (No. 22GXFW0124, No.2019GXYD17.3), the National Innovation and Entrepreneurship Training Program for College Students (No. 202211664023), and the Guangzhou Nansha District Innovation Team Project (No. 2021020TD001).

Data Availability Statement: No applicable.

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

# References

- Ma, Y.; Zhu, X.; Zhang, S.; Yang, R.; Wang, W.; Manocha, D. TrafficPredict: Trajectory Prediction for Heterogeneous Traffic-Agents. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 1 February 2019.
- Rasouli, A.; Tsotsos, J.K. Autonomous Vehicles That Interact with Pedestrians: A Survey of Theory and Practice. In Proceedings
  of the IEEE Transactions on Intelligent Transportation Systems, Yokohama, Japan, 26–29 May 2019.
- Hussein, A.; García, F.; Armingol, J.M.; Olaverri-Monreal, C. P2V and V2P communication for Pedestrian warning on the basis of Autonomous Vehicles. In Proceedings of the IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), Rio de Janeiro, Brazil, 1–4 November 2016.
- 4. Cunningham, W. Honda Tech Warns Drivers of Pedestrian Presence. Available online: https://www.cnet.com/roadshow/news/ honda-tech-warns-drivers-of-pedestrian-presence/ (accessed on 27 January 2023).
- Magdum, S.S.; Franklin, A.; Tamma, B.R.; Pawar, D.S. SafeNav: A Cooperative V2X System using Cellular and 802.11p based Radios opportunistically for Safe Navigation. In Proceedings of the 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC), Rhodes, Greece, 20–23 September 2020.
- 6. Ren, Y.; Zhao, Z.; Zhang, C.; Yang, Q.; Hong, K.-S. Adaptive Neural-Network Boundary Control for a Flexible Manipulator With Input Constraints and Model Uncertainties. *IEEE Trans. Cybern.* **2021**, *51*, 4796–4807.
- Zuo, Z.; Yang, X.; Li, Z.; Wang, Y.; Luo, X. Mpc-based cooperative control strategy of path planning and trajectory tracking for intelligent vehicles. *IEEE Trans. Intell. Veh.* 2021, 6, 513–522. [CrossRef]
- Jaswanth, M.; Narayana, N.K.L.; Rahul, S.; Supriya, M. Autonomous Car Controller using Behaviour Planning based on Finite State Machine. In Proceedings of the 2022 6th International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, 28–30 April 2022.
- 9. Malviya, V.; Reddy, A.K.; Kala, R. Autonomous Social Robot Navigation using a Behavioral Finite State Social Machine. *Robotica* **2020**, *38*, 2266–2289. [CrossRef]
- Shariski, F.H.; Priandana, K.; Wahjuni, S. Performance Analysis of Self-Organizing Map Method for Wheeled Robot Control System. In Proceedings of the 2020 International Conference on Smart Technology and Applications (ICoSTA), Surabaya, Indonesia, 20 February 2020.
- Tian, Z.; Guo, C.; Liu, Y.; Chen, J. An Improved RRT Robot Autonomous Exploration and SLAM Construction Method. In Proceedings of the 2020 5th International Conference on Automation, Control and Robotics Engineering (CACRE), Dalian, China, 18–20 September 2020.
- 12. Hao, B.; Du, H.; Dai, X.; Liang, H.Y. Automatic recharging path planning for cleaning robots. Mob. Inf. Syst. 2021, 2021, 1–19. [CrossRef]
- 13. Elbanhawi, M.; Simic, M. Sampling-based robot motion planning: A review. IEEE Access 2014, 2, 56–77. [CrossRef]
- Bae, S.H.; Joo, S.H.; Pyo, J.W.; Yoon, J.S.; Kuc, T.Y. Finite State Machine based Vehicle System for Autonomous Driving in Urban Environments. In Proceedings of the 2020 20th International Conference on Control, Automation and Systems (ICCAS), Busan, Republic of Korea, 13–16 October 2020.
- 15. González, D.; Pérez, J.; Milanés, V.; Nashashibi, F. A Review of Motion Planning Techniques for Automated Vehicles. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 1135–1145. [CrossRef]
- 16. Tang, G.; Tang, C.; Claramunt, C.; Hu, X.; Zhou, P. Geometric A-Star Algorithm: An Improved A-Star Algorithm for AGV Path Planning in a Port Environment. *IEEE Access* **2021**, *9*, 59196–59210. [CrossRef]
- He, N.; Su, Y.; Guo, J.; Fan, X.; Liu, Z.; Wang, B. Dynamic path planning of mobile robot based on artificial potential field. In Proceedings of the 2020 International Conference on Intelligent Computing and Human-Computer Interaction (ICHCI), Sanya, China, 4–6 December 2020.
- 18. Lu, S.X.; Li, E.; Guo, R. An Obstacles Avoidance Algorithm Based on Improved Artificial Potential Field. In Proceedings of the 2020 IEEE International Conference on Mechatronics and Automation (ICMA), Beijing, China, 13–16 October 2020.
- 19. Azzabi, A.; Nouri, K. An advanced potential field method proposed for mobile robot path planning. *Trans. Inst. Meas. Control* **2019**, *41*, 1–13. [CrossRef]
- Gutjahr, B.; Gröll, L.; Werling, M. Lateral vehicle trajectory optimization using constrained linear time-varying MPC. *IEEE Trans. Intell. Transp. Syst.* 2017, 18, 1586–1595. [CrossRef]
- 21. Mekala, G.K.; Sarugari, N.R.; Chavan, A. Speed Control in Longitudinal Plane of Autonomous Vehicle Using MPC. In Proceedings of the 2020 IEEE International Conference for Innovation in Technology (INOCON), Bangalore, India, 6–8 November 2020.
- 22. Murillo, M.; Sánchez, G.; Genzelis, L.; Giovanini, L. A real-time path-planning algorithm based on receding horizon techniques. *J. Intell. Robot. Syst.* **2018**, *91*, 445–457. [CrossRef]
- 23. Murota, K.; Shioura, A. Dijkstra's algorithm and L-concave function maximization. Math. Program. 2014, 145, 163–177. [CrossRef]
- 24. Dijkstra, E. A note two problems in connection with graphs. J. Numer. Math. 1959, 1, 269–271. [CrossRef]
- Rachmawati, D.; Sihombing, P.; Halim, B. Implementation of Best First Search Algorithm in Determining Best Route Based on Traffic Jam Level in Medan City. In Proceedings of the 2020 International Conference on Data Science, Artificial Intelligence, and Business Analytics (DATABIA), Medan, Indonesia, 16–17 July 2020; pp. 5–12.

- 26. Bokare, P.; Maurya, A. Acceleration-Deceleration Behaviour of Various Vehicle Types. Transp. Res. Procedia 2017, 25, 4733–4749. [CrossRef]
- Bae, I.; Moon, J.; Seo, J. Toward a Comfortable Driving Experience for a Self-Driving Shuttle Bus. *Electronics* 2019, *8*, 943. [CrossRef]
   Nie, B.; Li, Q.; Gan, S.; Xing, B.; Huang, Y.; Li, S.E. Safety envelope of pedestrians upon motor vehicle conflicts identified via
- active avoidance behaviour. *Sci. Rep.* **2021**, *11*, 3996. [CrossRef] 29. Zhao, Y.B.; Han, Z.Z.; Su, K.; Guo, L.; Yang, W.H. Anti-collision Trajectory Planning and Tracking Control based on MPC and
- Znao, Y.B.; Han, Z.Z.; Su, K.; Guo, L.; Yang, W.H. Anti-collision Trajectory Planning and Tracking Control based on MPC and Fuzzy PID Algorithm. In Proceedings of the 2020 4th CAA International Conference on Vehicular Control and Intelligence (CVCI), Hangzhou, China, 18–20 December 2020; pp. 613–618.

**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.