

Article

A Semi-Trailer Path Planning Method Considering the Surrounding Traffic Conditions and Vehicle Roll Stability

Haochuan Zhang ¹, Zhigen Nie ^{1,*} and Yufeng Lian ²

¹ Faculty of Transportation Engineering, Kunming University of Science and Technology, Kunming 650500, China; 20222106016@stu.kust.edu.cn

² The School of Electrical and Electronic Engineering, Changchun University of Technology, Changchun 130012, China; lianyufeng_1982@126.com

* Correspondence: niezhipeng@kust.edu.cn

Abstract: Path planning for intelligent semi-trailers encounters numerous challenges in complex traffic conditions. Serious consequences, such as vehicle rollover, may occur when the traffic conditions change. Therefore, it is vital to consider both the surrounding dynamic traffic conditions and the vehicle's roll stability during the lane-changing process of intelligent semi-trailers. We propose an innovative path-planning method tailored for intelligent semi-trailers. This path-planning method is designed for semi-trailers on straight-road alignments. Firstly, we employ a fuzzy inference system to process information about surrounding traffic, make lane-changing decisions, and determine the starting point. Secondly, the lane-changing path is generated using a B-spline curve. Subsequently, we apply a particle swarm optimization algorithm to enhance the B-spline curve. Thirdly, we utilize a Transformer model to analyze the nonlinear relationships among information about surrounding traffic, vehicle information, and the roll stability of the intelligent semi-trailer. We establish the roll stability boundary for the vehicle. Finally, we design a multi-objective cost function to select the optimal path. The simulation results demonstrate that the proposed method dynamically adapts the planned path to variations in driving parameters, ensuring trackability while reducing the steering angle, lateral acceleration, and yaw rate. This approach meets the roll stability requirements of intelligent semi-trailers, significantly enhances their stability during lane changing, and provides robust support for safe and efficient operation.



Received: 18 January 2025

Revised: 19 February 2025

Accepted: 20 February 2025

Published: 22 February 2025

Citation: Zhang, H.; Nie, Z.; Lian, Y. A Semi-Trailer Path Planning Method Considering the Surrounding Traffic Conditions and Vehicle Roll Stability. *Appl. Sci.* **2025**, *15*, 2353. <https://doi.org/10.3390/app15052353>

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

Keywords: intelligent semi-trailer; path planning; roll stability; fuzzy inference system; B-spline curve; particle swarm optimization; Transformer; adaptability to traffic conditions

1. Introduction

Autonomous driving technology has advanced rapidly in recent years and has expanding applications in commercial vehicles. Among them, intelligent semi-trailers have become the focus of academia and industry because of their broad development prospects [1]. However, path planning for these articulated vehicles remains a significant challenge in dynamic environments. Unlike passenger vehicles, semi-trailers exhibit complex kinematic behaviors and heightened susceptibility to rollover during lane changes. The risk is exacerbated by variable road conditions (e.g., adhesion coefficients), load distributions, and interactions with surrounding traffic [2]. Academic research on path planning has been conducted for a long time. At present, the common path-planning methods in the industry comprise the artificial potential field method (APF), algorithms based on curve interpolation, algo-

rithms based on sampling, algorithms based on graph searching, and algorithms based on intelligent optimization [3].

The goal of APF is to plan a collision-free path from the starting point to the target point for autonomous vehicles in the environment. It abstracts the movement of vehicles as the movement in a virtual potential field, which is superimposed by a gravitational field and a repulsive field. The gravitational field prompts the vehicle to move toward the target point, and the repulsive field makes the vehicle avoid obstacles. The resultant force received by the vehicle in the potential field determines its movement direction. By continuously calculating the resultant force and updating the vehicle position, a movement path is constructed [4]. There are two main problems with the APF. One is that it is prone to getting trapped at a local minimum, and the other is that it has insufficient adaptability to dynamic traffic conditions. Liu et al. successfully enhanced traditional APF to facilitate collision avoidance for both static and moving obstacles, providing a novel strategy for ensuring the safe operation of autonomous vehicles in complicated environments [5]. Chen et al. implemented a strategy involving the discretization of obstacle boundaries to improve safety during obstacle avoidance. Additionally, a stochastic escape force was integrated to inhibit the vehicle from becoming trapped in local minima [6]. Combining the APF with the model predictive control (MPC) also has a good effect on path planning [7,8]. However, the application of the APF approach comes with distinct drawbacks. It performs poorly in dealing with the kinematic constraints of vehicles and environmental dynamics, which reduces the driving safety of vehicles to some extent [9].

Algorithms based on curve interpolation can consider the kinematic and dynamic features of the vehicle, which are more common path-planning methods. Asrofudin et al. used the Sigmoid function to generate the lane-changing path to achieve efficient obstacle avoidance and stable path tracking for tractor-trailers [10]. Yue et al. introduced a robust tube model predictive control method based on fifth-degree polynomial trajectories for solving the automatic lane-changing problem of tractor-trailers [11]. Spline curves, such as B-spline curves and Bézier curves, have the advantages of good smoothness and continuous curvature changes, and they are also quite common in path planning [12–14]. Additionally, the simple curve interpolation method often fails to meet the requirements of vehicles for variable paths. Consequently, in the current stage, the curve interpolation method is customarily integrated with other optimization techniques [15,16]. However, the methods commonly exhibit the problem of elevated computational complexity and demonstrate relative inefficacy when confronted with dynamic obstacles.

Algorithms based on sampling include the rapid search random tree algorithm (RRT), the probabilistic roadmap algorithm (PRM), and their improved algorithms (such as R2-RRT*, APG-RRT, etc.) [17–19]. However, such methods are primarily applied in robotic path planning. When adapted to intelligent semi-trailers, the quality of the resultant planned paths leaves much to be desired, and their practical applicability is notably restricted.

Algorithms based on graph searching include the Dijkstra algorithm, A* algorithm, D* algorithm, and their fusion and improved algorithms [20–23]. However, the limitations of such algorithms are rather prominent when the vehicle velocity is high and in dynamic traffic conditions. They are mainly applicable to the determination of the shortest variable paths in static traffic conditions.

Algorithms based on intelligent optimization are methods that use intelligent optimization algorithms to solve path-planning problems. These algorithms usually simulate biological group behaviors, evolutionary processes, or physical phenomena in nature to find the optimal solution or approximate optimal solution to problems. Common intelligent optimization algorithms include genetic algorithms [24], ant colony algorithms [25], particle swarm optimization algorithms [26], etc. Through continuous iteration and evolu-

tion, they search in the solution space and can effectively handle complex problems that are difficult to solve by traditional optimization algorithms. Relevant research has made improvements targeting the deficiencies of traditional optimization algorithms [27–29]. However, such methods generally have the drawbacks of slow convergence speed and low computational efficiency.

At present, there are relatively few studies specifically focusing on the trajectory planning of intelligent semi-trailers. Hou and Xu designed an Active Trailer Steering (ATS) controller to address the high-speed driving safety issues of semi-trailers. Through theoretical analysis and simulation verification, the lateral stability of semi-trailers and the trajectory tracking performance have been effectively improved [30]. Manav et al. proposed an adaptive path-following control framework for the automatic docking of semi-trailers, effectively solving the path-following problem of semi-trailers during the docking process [31]. Han et al. proposed a new type of lane-keeping path planner and steering wheel angle controller for semi-trailers. Considering the unique dynamic characteristics of semi-trailers, it effectively improves the lane-keeping performance of semi-trailers during driving [32].

Existing path-planning methods face three critical limitations:

1. Roll stability neglect: Most of these methods focus on four-wheeled vehicles, and there is less research on semi-trailers. In comparison, the path planning problem of semi-trailers is more complicated. The intelligent semi-trailers have high requirements for roll stability, which needs to be considered when carrying out path planning.
2. Traffic conditions adaptability: Changes in the surrounding traffic conditions, such as the road surface adhesion coefficient, load capacity, and driving velocity, affect the path planning of intelligent semi-trailers, but they are often ignored.
3. As commercial vehicles for bulk transportation, the impact of load capacity on lane-changing paths cannot be underestimated.

Aiming at the above problems, we propose an intelligent semi-trailer path planning method considering the surrounding traffic conditions and vehicle roll stability. The primary contributions of this article can be outlined as follows:

1. A lane-changing method applicable to intelligent semi-trailers is proposed, which accounts for variations in the surrounding traffic conditions and the characteristics of intelligent semi-trailers, thereby maintaining their roll stability throughout the lane-changing maneuver.
2. A multi-state fusion lane-changing decision-making system based on a fuzzy inference system is constructed. This system comprehensively integrates multiple surrounding traffic conditions, including the road adhesion coefficient, velocity, acceleration, and the distance between the self-vehicle and the surrounding vehicles. Through a hierarchical fuzzy inference architecture, this system accurately makes decisions regarding lane-changing behavior and precisely calculates the optimal starting position for the maneuver.
3. A path generation and optimization scheme that integrates the B-spline curve with a particle swarm optimization algorithm (PSO) is designed. The optimization capabilities of PSO are employed to enhance the B-spline curve, effectively addressing the inherent limitations of traditional path planning algorithms when managing the complex kinematic constraints of intelligent semi-trailers.
4. A Transformer model is established to calculate the roll stability boundary of intelligent semi-trailers. This model utilizes road information, surrounding traffic condition data, and vehicle parameters as input variables to perform regression predictions on the maximum lateral load transfer ratio (LTR) of the intelligent semi-trailer, thereby en-

abling precise determination of the roll stability boundary and enhancing the vehicle's roll stability.

This approach not only enhances safety but also reduces steering effort by 15–25% compared to conventional methods, as validated through MATLAB (2023b)/TruckSim (2019) co-simulations (Section 6). By bridging theoretical advancements with practical deployment ability, our method paves the way for safer and more efficient autonomous freight transportation.

The rest of the article is arranged as follows: Section 2 introduces the multi-state lane-changing decision-making system. Section 3 describes the path generation and optimization scheme. Section 4 introduces the model for calculating the roll stability boundary. Section 5 introduces the optimal path selection module. Section 6 contains the experimental results. Section 7 summarizes the conclusions.

2. Multi-State Lane-Changing Decision-Making System

Hierarchical Fuzzy Inference is an important method in the field of fuzzy inference [33]. It is constructed based on fuzzy logic theory and aims to handle reasoning patterns of uncertain information with complex and multi-level structures. Hierarchical fuzzy inference divides the system into multiple levels and sets fuzzy rules and variables at different levels to deal with complex situations. The method can better deal with systems with complex structures and multiple variables. Therefore, it is feasible to use a hierarchical fuzzy inference system to control the lane-changing decision-making process of an intelligent semi-trailer.

We aim to study decision-making and planning in the automatic driving process. In this study, the intelligent semi-trailer is equipped with the capability to sense and accurately recognize the surrounding traffic condition information essential for its operation. When conducting path planning, the main considerations are straight road sections, flat slopes, and a standard lane width of 3.75 m. The vehicle determines its lane-change behavior by considering various factors, including the coefficient of the road surface attachment, its velocity and acceleration, the velocity and acceleration of a nearby vehicle, and the distance between itself and the surrounding vehicle.

2.1. Design of Fuzzy Control System

The process of the method is illustrated in Figure 1, which depicts the framework of the lane-changing decision-making system. In this system, the lane-changing decision process is modeled as a function:

$$E_{steer} = f(\mu, v_{subj}, v_{sur}, a_{subj}, a_{sur}, d) \quad (1)$$

where E_{steer} quantifies the decision threshold, referred to as the lane-changing decision factor, μ is the adhesion coefficient of the road surface, v_{subj} is the velocity of the self-vehicle, v_{sur} is the velocity of the surrounding vehicle, d is the distance between the self-vehicle and the surrounding vehicle, a_{subj} is the acceleration of the self-vehicle, and a_{sur} is the acceleration of the surrounding vehicle. When the value of E_{steer} exceeds 0.5, the vehicle will execute a lane change; otherwise, it will continue moving straight ahead.

In the first layer of fuzzy reasoning, the collision risk factor E_r is obtained by analyzing v_{subj} and μ . E_r can be expressed as:

$$E_r = FIS_1(v_{subj}, \mu) \quad (2)$$

where E_r is the risk factor. The inference surface of the first layer structure is shown in Figure 2a.

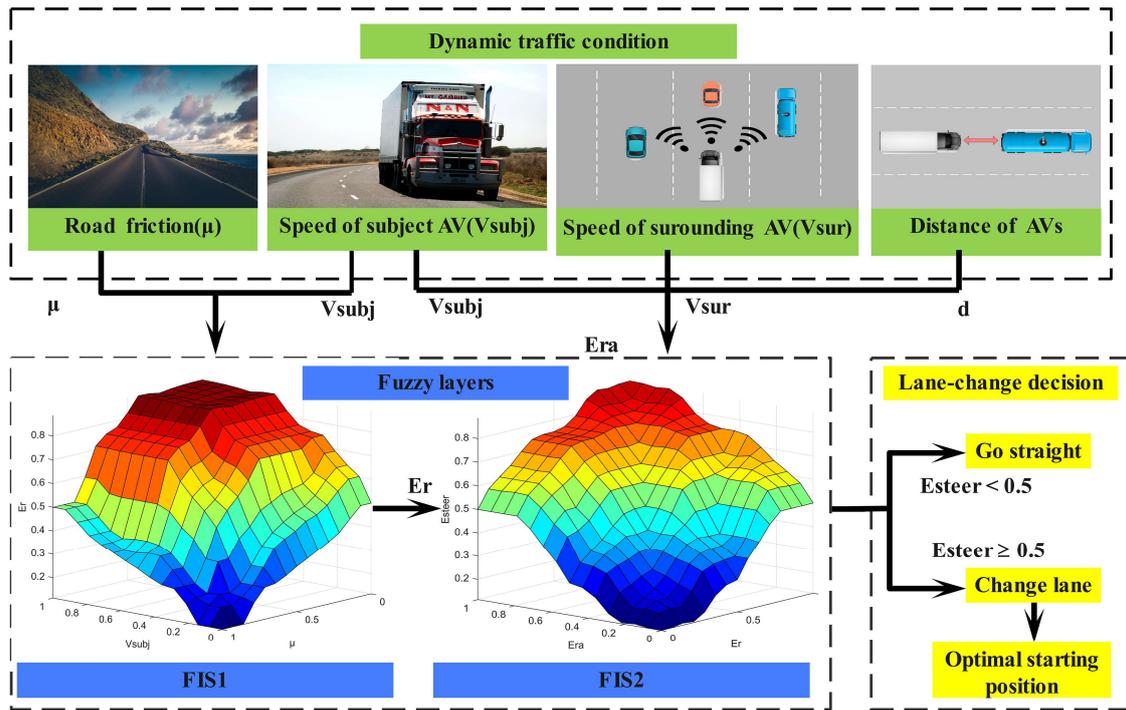


Figure 1. Hierarchical lane-changing decision framework for intelligent semi-trailers.

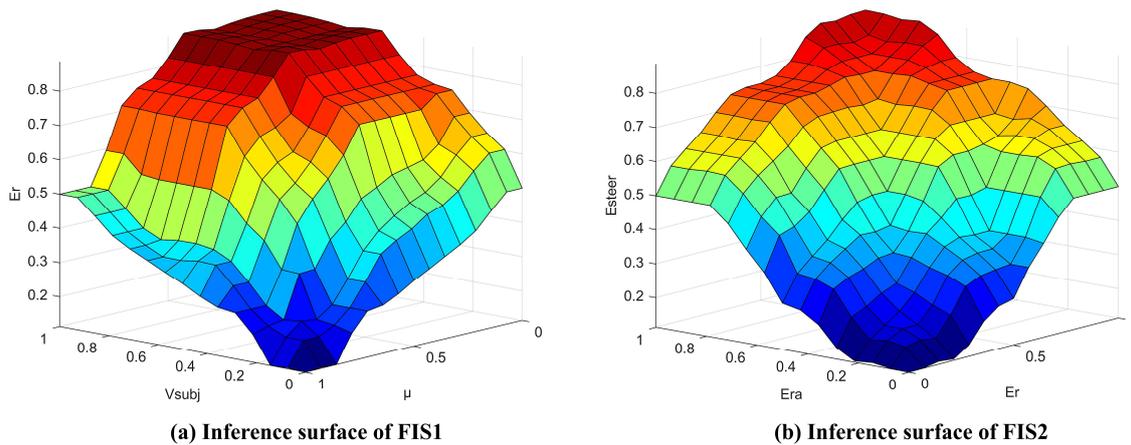


Figure 2. Inference surface.

In the second layer of fuzzy reasoning, the risk avoidance factor E_{ra} can be calculated by the velocity of the self-vehicle, the velocity of the surrounding vehicle, and the distance between the self-vehicle and the surrounding vehicle. Then, E_{ra} and E_r are input together into the second layer of the system to obtain the lane-changing decision factor E_{steer} :

$$E_{steer} = FIS_2(E_r, E_{ra}) \tag{3}$$

The inference surface of the second layer structure is shown in Figure 2b.

2.2. Establish Membership Function

The membership function is an important concept in fuzzy set theory. It is used to describe the degree to which an element belongs to a fuzzy set. It maps explicit input values to a membership degree value between 0 and 1. The closer the degree of membership is to 1, the more the input belongs to the set. Conversely, the closer it is to 0, the less it belongs to the set [34]. In this study, the membership function is used to convert explicit variables such

as vehicle state and environmental information into fuzzy variables for fuzzy reasoning. In this system, the membership function composed of a triangular membership function and an S-shaped membership function is adopted.

In the first layer fuzzy inference system (FIS1), the input variables are μ and v_{subj} , and the output variable is E_r . First, we standardize v_{subj} as follows:

$$\tilde{v}_{subj} = \frac{v_{subj} - v_{min}}{v_{max} - v_{min}} \tag{4}$$

setting the minimum velocity v_{min} to 0 m/s and the maximum velocity v_{max} to 35 m/s. In the first layer fuzzy inference system (FIS1), the explicit variables \tilde{v}_{subj} , μ , and E_r are converted to fuzzy variables through the membership functions. The membership functions of \tilde{v}_{subj} , μ , and E_r are shown in Figure 3a–c.

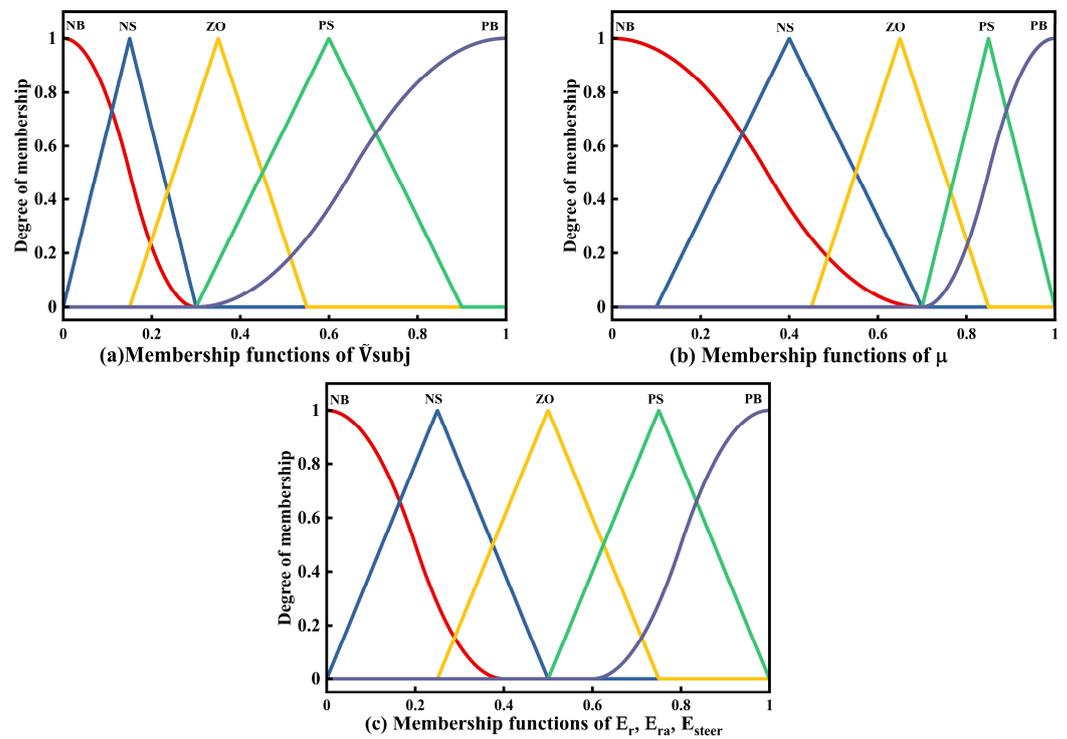


Figure 3. Membership functions.

In the second layer fuzzy inference system (FIS2), the variables for input are E_r and E_{ra} , the output variable is E_{steer} . E_{ra} is calculated from v_{subj} , v_{sur} , and d :

$$E_{ra} = \frac{D_s}{d} \times \frac{v_{subj} - v_{sur}}{v_{subj}} \tag{5}$$

where D_s is the safety distance and defined as:

$$D_s = \frac{v_{subj}^2}{2a_{subj}} - \frac{v_{sur}^2}{2a_{sur}} + v_{sur} \times t_x + D_0 \tag{6}$$

where a_{subj} and a_{sur} are the maximum accelerations of the self-vehicle and surrounding vehicle; $a_{subj} = 8 \text{ m/s}^2$, $a_{sur} = 8 \text{ m/s}^2$. t_x is the delay time of the vehicle control system, which equals 0.2 s. D_0 represents the minimum safe distance maintained between two vehicles, established at 1 m [35].

In the second layer fuzzy inference system (FIS2), the explicit variables E_r , E_{ra} , and E_{steer} are converted to fuzzy variables through the membership functions. The membership functions of E_r , E_{ra} , and E_{steer} are shown in Figure 3c.

2.3. Fuzzy Control Rules and Defuzzification

The control rules in this system mainly include the following two points: (1) As v_{subj} and E_{ra} increase, the lane-changing maneuver should be initiated earlier, i.e., E_{steer} should increase. Also, as μ decreases, the lane changes should be advanced, and E_{steer} should increase. (2) When the input variable v_{subj} is large or μ is small, the output variable E_{steer} is more sensitive to changes. Fuzzy control rules are the core component of the multi-state lane-changing decision-making system. They provide a logical basis for the lane-changing decision-making of intelligent semi-trailers based on the relationship between different input variables. Through careful consideration of various influencing factors, these rules can determine whether and when a vehicle should change lanes. The following is a detailed introduction to the fuzzy control rules formulated according to different input variables. These rules are summarized in Tables 1 and 2, respectively.

Table 1. Fuzzy inference rules for FIS1.

E_r		\tilde{v}_{subj}				
		NB	NS	ZO	PS	PB
μ	NB	ZO	PS	PS	PB	PB
	NS	NS	ZO	PS	PB	PB
	ZO	NS	NS	ZO	PS	PS
	PS	NB	NS	NS	ZO	ZO
	PB	NB	NB	NS	NS	ZO

Table 2. Fuzzy inference rules for FIS2.

E_{steer}		E_{ra}				
		NB	NS	ZO	PS	PB
E_r	NB	NB	NB	NS	ZO	ZO
	NS	NB	NS	ZO	PS	PS
	ZO	NS	ZO	ZO	PS	PS
	PS	ZO	ZO	PS	PS	PB
	PB	ZO	PS	PS	PB	PB

Where ZO represents zero, PS represents positive small, PB represents positive big, NS represents negative small, and NB represents negative big.

Table 1 shows the fuzzy inference rules of the first-level fuzzy inference system (FIS1). In it, rows represent different fuzzy sets of the vehicle’s own speed \tilde{v}_{subj} , and columns represent different fuzzy sets of the road adhesion coefficient μ . The element E_r in the table is the fuzzy set of the collision risk factor inferred from μ and \tilde{v}_{subj} . For example, when μ is NB (Negative Big) and \tilde{v}_{subj} is NB (Negative Big), E_r is ZO (Zero). This indicates that, in this case, the collision risk factor is at a relatively low level.

Table 2 shows the fuzzy inference rules of the second-level fuzzy inference system (FIS2). In it, rows represent different fuzzy sets of risk avoidance factor E_{ra} , and columns represent different fuzzy sets of risk factor E_r . The element E_{steer} in the table is the fuzzy set of the lane-changing decision factor inferred from E_{ra} and E_r . For example, when E_{ra} is PB (Positive Big) and E_r is PS (Positive Small), E_{steer} is PB (Positive Big). At this time, E_{steer} is much greater than 0.5, and the vehicle will be more inclined to change lanes to avoid potential risks or obtain more favorable driving conditions. In this way, the fuzzy

control rules achieve effective control over the lane-changing decision-making of intelligent semi-trailers by comprehensively processing multiple input variables.

On the other hand, compared with other defuzzification methods, the center of gravity method offers distinct advantages. It considers the entire distribution of the fuzzy set, considering both the membership degrees of all elements within the set and their corresponding output values. For example, in our lane-changing decision-making system, when calculating the lane-changing decision coefficient E_{steer} , the center of gravity method does not just focus on individual high-membership elements. Instead, it weighs all elements according to their membership degrees and the values they contribute to the decision-making process. This careful consideration of the overall distribution of the fuzzy set ensures that the defuzzing result is more representative of the entire fuzzy inference process. Moreover, the center of gravity method has a full mathematical theoretical foundation [36], providing a reliable and consistent approach for obtaining a crisp output from the fuzzy set. Therefore, we choose the center of gravity method. Then, E_{steer} will be obtained. When E_{steer} is greater than 0.5, the intelligent semi-trailer will make a lane change, and the following formula is used to calculate the starting position where the lane change occurs:

$$D_{sp} = E_{steer} \times D_{sd} + D_s \tag{7}$$

where D_{sp} is the distance between the self-vehicle and the surrounding vehicle at the beginning of the lane changing, and D_s is the distance of the lane-changing decision. Based on experimental findings, an optimal value of D_s is determined to be 10 m. Figure 4 illustrates the lane-changing distance.

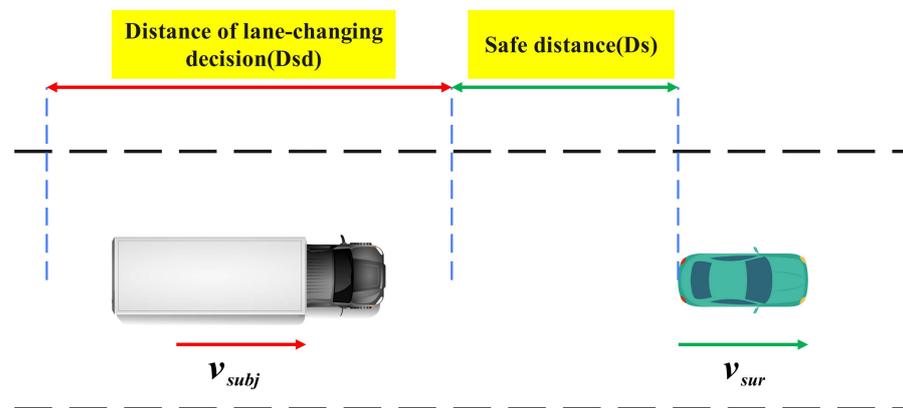


Figure 4. Allowed distance of lane changing.

3. Path Generation and Optimization Scheme

After obtaining the lane-changing position information, we perform path planning starting from this position. Subsequently, we construct the solution space for paths utilizing the B-spline curve. On this basis, a cost function that takes both the path length and the average curvature into consideration is designed. Then, we employ PSO to optimize the path and generate a set of optimal path candidates.

3.1. Creation of Lane Change Solution Space Based on B-Spline Curve

The B-spline curve offers several advantages suitable for vehicle lane-changing paths [37]. Firstly, it has good smoothness, and the curvature of each point on the path changes continuously. This property can satisfy the demands of the vehicle lane changes. Secondly, in the case of the quasi-uniform B-spline curve, both the initial and final points have a curvature of zero. In addition, the B-spline curve has a mature calculation algorithm

and strict mathematical definition, which is more efficient in calculation and analysis. The formula for calculating the B-spline curve is presented below:

$$P(u) = \sum_{i=0}^n P_i \cdot N_{i,k}(u) \tag{8}$$

where $P(u)$ represents the position of a point on the B-spline curve at the parameter u , and $u \in [0, 1]$ traverses the curve from start to end. $P_i (i = 0, 1, \dots, n)$ are the control points that define the shape and boundary of the curve. These points act as “magnets” to guide the curve’s trajectory without necessarily lying on the curve itself. $N_{i,k}(u)$ are the B-spline basis functions of order k . These functions determine the influence of each control point P_i on the curve at parameter u . k is the order of the B-spline, which defines its smoothness. u is a parameter that varies along the curve. $n + 1$ represents the number of control points. For the lane-changing paths of vehicles, the cubic B-spline curve can usually achieve a better balance between smoothness and computational efficiency. Therefore, k is set as four. The basic functions $N_{i,k}(u)$ are computed recursively using the Cox-deBoor formula:

$$N_{i,k}(u) = \begin{cases} \begin{cases} 1, u_i \leq u < u_{i+1} \\ 0, otherwise \end{cases}, k = 1 \\ \frac{u-u_i}{u_{i+k-1}-u_i} N_{i,k-1}(u) + \frac{u_{i+k}-u}{u_{i+k}-u_{i+1}} N_{i+1,k-1}(u), k \geq 2 \end{cases} \tag{9}$$

As illustrated in Figure 5, the B-spline curve is constructed with six control points (A, B, C, D, E, and F). Point A marks the lane change initiation determined by the decision system. More control points are not conducive to subsequent path optimization. To enhance the efficiency and practicality of path generation, the following settings will be set for the control points: (1) The line ABC and the line DEF are parallel to the road centerline. (2) Set $AB = BC = DE = DF$ and O as the midpoint of the path. Then, the degree of freedom of the B-spline reduces to two. The distance between points A and C is denoted as k_1 and the distance between point C and point D is denoted as k_2 .

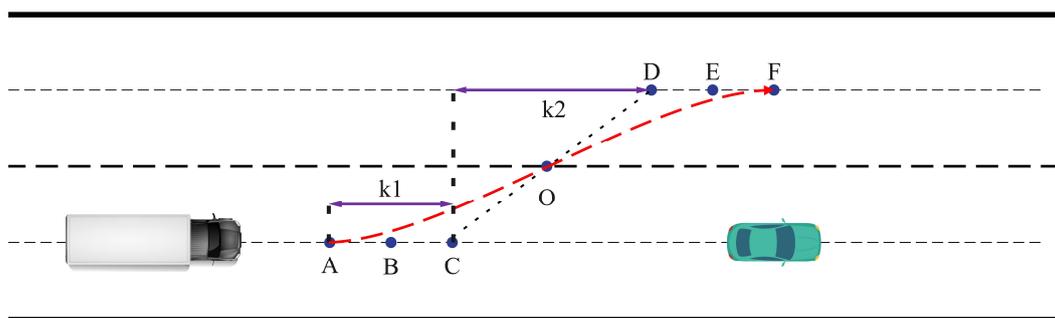


Figure 5. Path generation based on B-spline curve.

The B-spline trajectory is designed to align with the center of the tractor’s rear axle of the semi-trailer. This choice ensures that the planned path accounts for the vehicle’s kinematic constraints and simplifies the tracking control logic. Specifically, each point $P(u)$ on the B-spline curve (Equation (8)) corresponds to the projected position of the rear axle center during lane-changing maneuvers. This alignment allows the control system to directly map the generated path to the vehicle’s steering dynamics while maintaining roll stability.

3.2. Path Optimization Using PSO

PSO is an optimization method inspired by the behavior patterns of biological populations. It effectively identifies optimal solutions or approaches the desired goals by mimicking the feeding strategies of birds [38]. PSO is characterized by its rapid convergence and robust global optimization capabilities, making it well-suited for optimizing local pathways.

To define a population within a two-dimensional space, the positional data and boundary constraints of the i -th particle during the n -th iteration are described below:

$$\begin{cases} X_i(n) = [k_{1,i}(n), k_{2,i}(n)] \\ 0 < k_{1,i} \leq 20 \\ 0 < k_{2,i} \leq 50 \end{cases} \quad (10)$$

The method for updating the speed and position of the particle is illustrated in Figure 6. The speed update formula is as follows:

$$V_i(n + 1) = \omega V_i(n) + c_1 r_1 [p_{best} - X_i(n)] + c_2 r_2 [g_{best} - X_i(n)] \quad (11)$$

$$X_i(n + 1) = X_i(n) + V_i(n+1) \quad (12)$$

where p_{best} (personal best position) refers to the best solution a particle has achieved individually during the optimization process, and g_{best} (global best position) represents the best solution found by the entire particle swarm. These positions guide the particles toward the optimal path by balancing individual exploration and swarm collaboration. c_1 is the personal learning factor and fixed at 1.3, c_2 is the global learning factor and fixed at 1.7. They are utilized to modulate the step length of particle learning, concerning and g_{best} . r_1 and r_2 are two stochastic numbers falling in the interval from 0 to 1. The inertia weight denoted as ω , is essential for balancing the global and local search abilities within the particles. To enhance optimization efficiency, ω ought to progressively diminish as the iteration count rises. It is defined in the following way:

$$\omega = \omega_{start} - (\omega_{start} - \omega_{end}) \times (n/g)^2 \quad (13)$$

where ω_{start} represents the initial inertia weight, assigned a value of 0.9, whereas ω_{end} represents the final inertia weight, assigned a value of 0.4. n refers to the current iteration count, and g denotes the upper limit on the number of iterations.

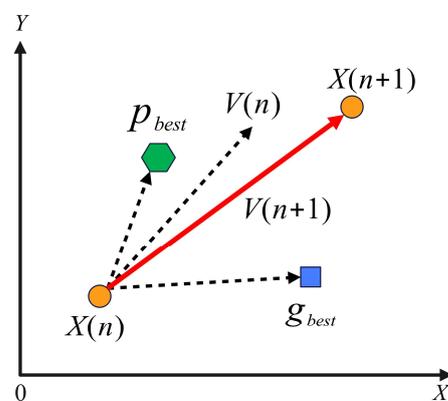


Figure 6. Process of particle updating.

The fitness function of the particles is presented below:

$$f = \omega_1 f_1 + \omega_2 f_2 \quad (14)$$

$$f_1 = \sum \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} \quad (15)$$

$$f_2 = \frac{\sum K_i}{n} \quad (16)$$

where f is the fitness function of the particles, f_1 is the length of the generated path, f_2 is the mean curvature of the path, K_i is the curvature of points on the path, ω_1 is the weight of the path length, and set to 1, and ω_2 is the weight of the mean curvature of the path and set from 100 to 10,000. By changing the ratio of ω_1 and ω_2 , the optimal path with different weight ratios can be obtained. To handle the computational cost, we pre-calculate the optimal paths corresponding to different ratios of ω_1 and ω_2 . Subsequently, during the actual application, we efficiently utilize pre-calculated optimal paths through offline processing. By doing so, we can effectively bypass the computational cost that would otherwise be incurred during real-time optimization. This approach ensures that the PSO-based path optimization can be carried out efficiently without sacrificing the real-time performance of the overall system.

4. Model for Calculating the Roll Stability Boundary

Transformer, introduced by Vaswani et al. in 2017, features a self-attention mechanism originally designed to tackle challenges in natural language processing [39]. Due to its advantages, such as powerful feature extraction ability, efficient parallel computing ability, and the ability to process complex data with high dimensionality and nonlinearity, it also has better performance in regression prediction. In the lane-changing path generation module, we adjust the ratio of ω_1 and ω_2 to generate a series of unconstrained path clusters. Additionally, changes in the road adhesion coefficient, vehicle velocity, load capacity, and other driving conditions will also affect the stability boundary of the vehicle on the same path. The roll stability boundary of the intelligent semi-trailer is determined based on the maximum lateral load transfer ratio (LTR). LTR is a key index for evaluating the rollover risk of a vehicle. We use the Transformer model to predict the maximum LTR under different driving conditions. When the maximum LTR, denoted as LTR_{max} , is less than 0.9, the semi-trailer is considered to maintain a stable state and is within the roll stability boundary. In other words, the value of 0.9 for LTR_{max} serves as a threshold to define the roll stability boundary [40]. This relationship allows us to precisely determine whether a planned path meets the roll stability requirements of the intelligent semi-trailer.

The solution method of roll stability boundary takes the road adhesion coefficient, vehicle driving velocity, trailer load weight, and ratio of ω_1 and ω_2 as input features and takes the LTR_{max} as the output feature to conduct regression prediction. After training, the LTR_{max} under different driving conditions can be obtained, thus determining the stability boundary. LTR is a widely used index for vehicle rollover risk assessment [41] and is defined as:

$$LTR = \frac{|F_{Z,L5} - F_{Z,R5}|}{F_{Z,L5} + F_{Z,R5}} \quad (17)$$

where $F_{Z,L5}$ represents the load on the left wheels of the five-axle semi-trailer, $F_{Z,R5}$ represents the load on the right wheels of the five-axle semi-trailer. This ratio reflects the distribution of the vehicle's load between the left and right sides. When a vehicle is in a dynamic state, such as during lane-changing or cornering, the load distribution between the left and right wheels changes. A higher LTR indicates a greater imbalance in the load distribution, which increases the risk of the vehicle rolling over. For example, if the load on

the right wheels is much larger than that on the left wheels, the LTR value will be closer to 1, and the vehicle is at a higher risk of rollover.

We selected the LTR as an index to evaluate the roll stability of intelligent semi-trailers because it directly reflects the load-distribution situation related to vehicle rollover. Compared with other potential indexes, LTR is a well-established and widely used metric in the automotive industry for rollover risk assessment. It has a clear physical meaning and can effectively quantify the rollover risk under different driving conditions. By using LTR, we can accurately analyze the roll stability of the vehicle during the lane-changing process and ensure that the planned path can maintain the vehicle's stability, which is crucial for the safe operation of intelligent semi-trailers.

4.1. Transformer Model Construction

The Transformer model has demonstrated powerful performance in many fields. Its main architecture consists of two major components: the encoder and the decoder.

The encoder undertakes a crucial task. First, it performs position encoding on the input data. Since the Transformer model adopts the self-attention mechanism, there is no natural sequential relationship among data elements. Thus, position encoding becomes particularly crucial. Through a specific position encoding method, each data element is given unique position information, enabling the model to effectively distinguish data at different positions. After completing the position encoding, the encoder starts the feature recognition work. It extracts key features from the input data and integrates and transforms these features through a series of complex operations and processing. In this process, multiple components within the encoder work together. For example, the Multi-Head Attention mechanism can focus on different parts of the input data from different perspectives, thus more comprehensively capturing the relationships between data features. The Feed-Forward Neural Network further performs non-linear transformations on the features, enhancing the model's ability to understand and represent data. After these operations, the encoder converts the processed information into the hidden layer. As an intermediate link of the model, the hidden layer stores and integrates the important feature information extracted by the encoder. This information contains the deep-level features and patterns of the data, providing crucial support for the subsequent decoding of work.

At the same time, the decoder plays an indispensable role. It receives data from the hidden layer and, based on this encoded information, converts them into the final output data through its own structure and operation logic. The decoder also contains multiple functional modules. For example, the Masked Multi-Head Attention layer can prevent the decoder from peeking into future information, ensuring that the output depends only on the generated data. The Multi-Head Encoder-Decoder Attention layer can effectively integrate the information of the encoder and the decoder, enabling the model to make better use of the global information of the input data when generating the output. The Feed-Forward Neural Network layer performs another non-linear transformation on the processed features to further optimize the output result. Through the coordinated operation of these components, the decoder finally outputs results that meet the requirements of the task.

The framework of the Transformer is depicted in Figure 7. The specific operation and calculation process of the Transformer will be introduced later in the text.

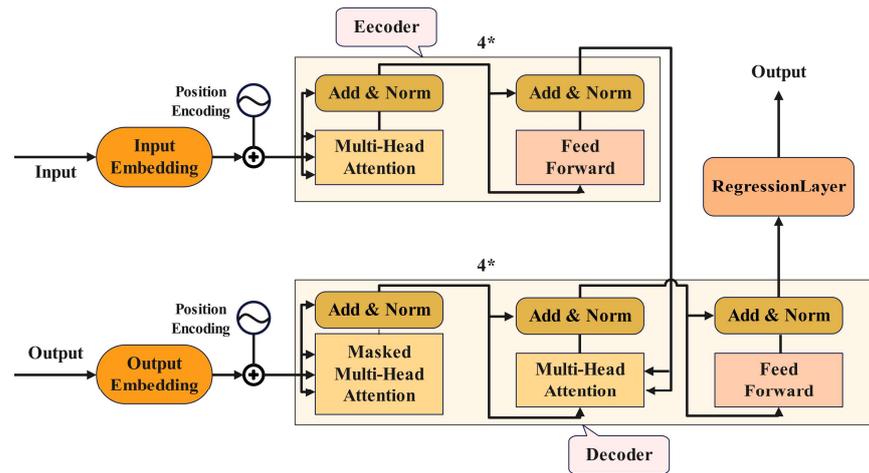


Figure 7. Structure of Transformer.

4.1.1. Input Data Processing

We obtained the input dataset through MATLAB/TruckSim co-simulation. Each piece of data contain five features: roadway adhesion coefficient, velocity of the self-vehicle, load, ω_1 , and ω_2 , with LTR_{max} as the output. We import 1472 such data points into MATLAB to obtain a 1472×6 matrix D. The matrix is processed as follows:

1. Data partitioning: D is divided into four matrices, which are the training set feature matrix, training set output matrix, test set feature matrix, and test set output matrix. And the ratio of training data to test data is 7:3.
2. Data normalization: To speed up training and improve training stability, we normalized the input data and processed the input data matrix by mapping the row maximum and minimum values to [0, 1]:

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{18}$$

where x_{norm} is the normalized data, x is the original data, x_{max} indicates the maximum value within the dataset, and x_{min} indicates the minimum value within the dataset.

3. Position encoding: The Transformer model utilizes a self-attention mechanism. Unlike some traditional neural network architectures, the self-attention mechanism in the Transformer does not inherently capture the order of elements in the input data. As a result, the elements within the data lack an implicit order for the model to distinguish them. Therefore, position coding is necessary. Position coding assigns unique position information to each element in the input data sequence. This enables the Transformer to differentiate between different elements, which is crucial for accurately processing sequential data. The formula for calculating position coding is as follows:

$$\begin{cases} PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{\frac{2i}{d_{model}}}}\right) \\ PE_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{\frac{2i}{d_{model}}}}\right) \end{cases} \tag{19}$$

where $PE_{(pos,2i)}$ is the position encoding, pos is the position, i is the index of the dimension, and d_{model} is the dimension of the input and output vectors. The use of sine and cosine functions in position encoding offers several advantages. First, these functions can represent the position information continuously and periodically.

The sine and cosine waves have unique periodicity properties, which can capture the relative positions of elements in the sequence effectively. For example, as the position increases, the values of sine and cosine functions change periodically, and this periodic change can be used to distinguish different positions. Secondly, the combination of sine and cosine functions allows the model to learn both the absolute and relative positions of elements. The sine and cosine of the same position value provide two different dimensions of information, which enriches the position-related features available for the model. This is crucial for the Transformer to understand the order of input data, especially in tasks like path planning for intelligent semi-trailers, where the order of data related to different driving conditions and vehicle states is significant. Thirdly, the use of these functions in position encoding is computationally efficient. They can be easily integrated into the neural network architecture without introducing excessive computational complexity, which is beneficial for the overall performance of the Transformer model during training and prediction.

4.1.2. Encoder

The encoding component comprises four identical encoders arranged in a stack, with each encoder consisting of two sub-layers known as the Multi-Head Attention layer and the Feed-Forward Neural Network layer.

The process of the calculation is illustrated in Figure 8. The processed data first flow into the self-attention mechanism layer, allowing the model to focus on other pertinent information during the data processing. The pertinent information includes data that are related to the currently processed elements in terms of time or space and is valuable for determining the roll stability boundary of intelligent semi-trailers. Specifically, it encompasses input features such as the road adhesion coefficient, vehicle driving speed, load weight, and the variation of these features at different times or positions. This information is of utmost importance for accurately assessing the vehicle’s driving state and predicting the rollover risk.

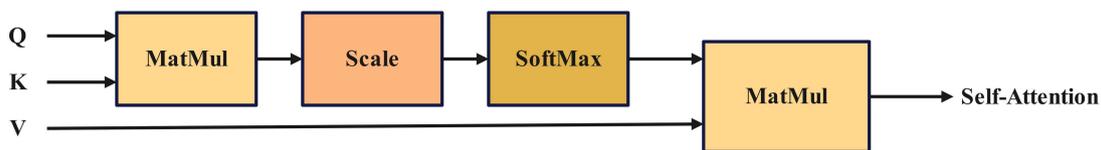


Figure 8. Computational procedure of self-attention.

The self-attention mechanism layer dynamically assigns attention weights to input features by computing pairwise dependencies. Specifically, given the input sequence, the self-attention mechanism calculates query (Q), key (K), and value (V) matrices through linear transformations:

$$\begin{cases} Q = XW_Q \\ K = XW_K \\ V = XW_V \end{cases} \quad (20)$$

where W_Q , W_K , and W_V are learnable weight matrices. The attention scores are then computed as scaled dot-products between Q and K:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_K}}\right)V \quad (21)$$

where d_K represents the dimension of K, and the calculation of QK^T measures the correlation between each element in the query matrix Q and the key matrix K. Specifically, Q is multiplied by K^T in matrix operation, and each element in the resulting matrix represents

the similarity between the query vector and the corresponding key vector. $\sqrt{d_k}$ is used to scale the result. This is because when d_k is large, the value of QK^T may become very large, causing the gradient of the softmax function to become extremely small, resulting in the problem of gradient vanishing. By dividing by $\sqrt{d_k}$, this situation can be effectively avoided, making the model training more stable. The softmax function then transforms the scaled result into a probability distribution. These probability values are the attention scores, which represent the degree of attention the model pays to each other element when processing the current element. The higher the attention score, the stronger the model believes the correlation between the element and the currently processed element is, and the greater the weight assigned to this element in subsequent calculations.

In semi-trailer path planning, this mechanism prioritizes interactions among dynamic parameters, such as road adhesion coefficient, vehicle velocity, and load mass. For instance, when predicting roll stability boundaries, higher attention weights may be assigned to velocity-load correlations under low-friction conditions, ensuring adaptive path adjustments.

By connecting different $head_i$, the multi-head attention can be obtained, and the calculation process is illustrated in Figure 9. The formula for the calculation is presented as follows:

$$MultiHead(Q, K, V) = Concat(head_1, head_2, \dots, head_n)W^O \tag{22}$$

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \tag{23}$$

where $head_i$ is the attention output of a single head, the *Concat* function represents concatenating all $head_i$, and W^O is the learnable weight matrix of the linear transformation.

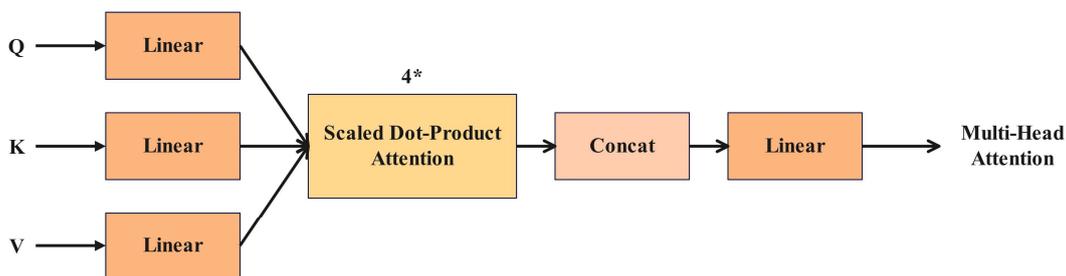


Figure 9. Computational procedure of multi-head attention.

After being processed by the multi-head attention layer, the input data are fed into the feedforward neural network. This network can make further non-linear changes to the processed input data, aiding the model in understanding the relationships among the input data across various dimensions. This step can be represented as:

$$FFN(x) = \max(0, xW_1 + b_1)W_2 + b_2 \tag{24}$$

where $\max(0, xW_1 + b_1)W_2$ is a Rectified Linear Unit (ReLU), x is the input data, W_1 and W_2 denote two matrices of weights, and b_1 and b_2 denote two vectors of bias.

4.1.3. Decoder

The structure of the decoder resembles that of the encoder and is composed of three sublayers:

1. Masked Multi-Head Attention layer: This layer adds a mask relative to the multi-head self-attention layer in the encoder, which makes it impossible for the decoder to peek into future information, ensuring that the output of the current position depends only on the generated information.

2. Multi-Head Encoder-Decoder Attention layer: In this layer, the input data is derived from two sources: the query matrix originating from the masked multi-head self-attention layer and the key matrix alongside the value matrix taken from the encoder’s output, which allows the model to learn and train the relevant information more accurately.
3. Feedforward neural network layer: With the same structure as the feedforward neural network in the encoder, the processed features are further non-linearly transformed, which enhances the learning and training capabilities of the model.

4.1.4. Output

After the data are learned and trained by the encoder and decoder, it is then linearly transformed and finally passed into the output layer, and the result of regression prediction is obtained. The output layer is created using the regression Layer command in the MATLAB software toolbox.

4.2. Analysis of Model Training Results

After setting up the training network, the parameters for training are configured as follows: the upper limit for training epochs is 5000, the strategy for adjusting the learning rate is piecewise, beginning with an initial learning rate of 0.01, a decay factor of 0.5, and a decay interval of 80. The parameter settings were obtained through multiple experimental comparisons. After training, the relationship between the predicted values and the actual values is illustrated in Figure 10.

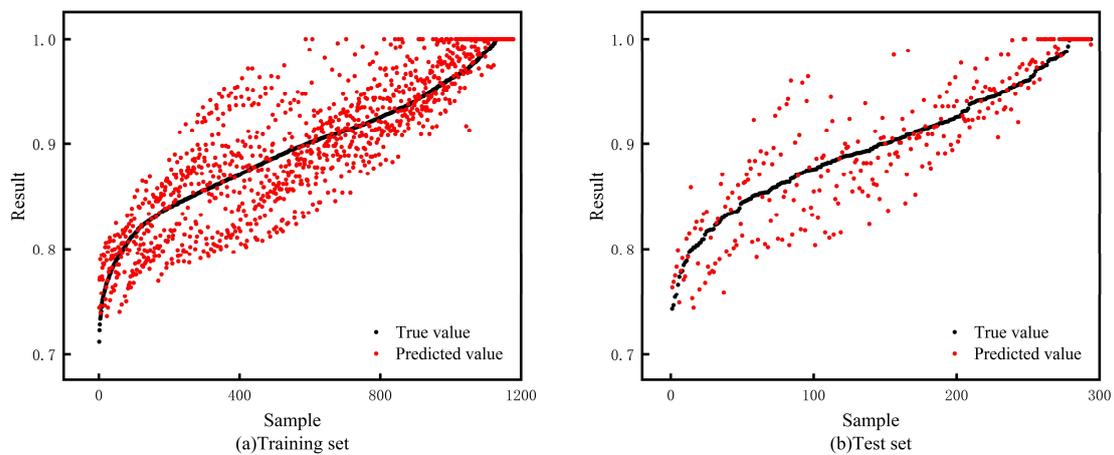


Figure 10. Comparison of predicted vs. true LTR values in Transformer training.

To confirm the precision of the model, various metrics are used to assess its performance, including the following:

4.2.1. Mean Squared Error (MSE):

MSE calculates the average sum of the squares of the differences between the predicted values and the true values. It is highly sensitive to the magnitude of the errors. The larger the errors, the larger the value of MSE will be. In this article, MSE is used to measure the overall error degree between the predicted values and the actual values when the Transformer model predicts the maximum LTR of intelligent semi-trailers. Its calculation formula is as follows:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \tag{25}$$

where n represents the number of data samples used to evaluate the Transformer model, y_i represents the actual value of the maximum LTR, and \hat{y}_i represents the maximum LTR value predicted by the Transformer model for the corresponding sample. Their meanings are the same in Equations (25)–(28).

4.2.2. Root Mean Square Error (RMSE):

RMSE is the square root of MSE. It is also used to measure the error between the predicted values and the true values. Compared with MSE, RMSE is more sensitive to the magnitude of the errors because it considers the square of the errors, amplifying the impact of larger errors. When evaluating the Transformer model, RMSE can intuitively reflect the average error magnitude between the model's predicted values and the true values. A smaller RMSE value indicates that the maximum LTR predicted by the model is closer to the actual situation, the prediction accuracy of the model is higher, and it has higher reliability in determining the vehicle roll-stability boundary. Its calculation formula is as follows:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (26)$$

4.2.3. Mean Absolute Error (MAE):

MAE calculates the average of the absolute errors between the predicted values and the true values. It can directly reflect the average degree of difference between the predicted values and the true values without considering the direction of the errors. In the research of intelligent semi-trailers, MAE can intuitively show the average deviation between the maximum LTR predicted by the model and the actual value. Different from MSE and RMSE, MAE is relatively less sensitive to outliers and can better reflect the average level of the model's prediction errors, helping researchers understand the overall prediction accuracy of the model. Its calculation formula is as follows:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (27)$$

4.2.4. Mean Absolute Percentage Error (MAPE):

MAPE measures the prediction error as the percentage of the absolute error to the true value, which reflects the relative magnitude of the error between the predicted value and the true value. The advantage of this indicator is that it is not affected by the dimension of the data, allowing for comparisons between data of different magnitudes. When evaluating the Transformer model, MAPE can clearly show the relative error of the maximum LTR predicted by the model under different driving conditions. A lower MAPE value indicates that the deviation between the predicted values and the true values of the model under different driving conditions is relatively small, and the prediction results of the model are relatively reliable in various working conditions, which can provide more accurate information for vehicle stability analysis. Its calculation formula is as follows:

$$MAPE = \frac{100\%}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \quad (28)$$

All evaluation indexes are presented in Table 3. These indicators show that the Transformer can effectively predict the maximum LTR of intelligent semi-trailers under different driving conditions, thus fulfilling the goal of determining the stability boundary of the intelligent semi-trailers.

Table 3. Evaluation indexes of Transformer.

Evaluation Index	MSE	RMSE	MAE	MAPE
Training set	0.00104	0.03225	0.02493	0.02827
Test set	0.00105	0.03246	0.02509	0.02848

5. Optimal Path Selection

After determining the stability boundary, the paths that do not meet the boundary conditions can be eliminated to obtain the feasible path cluster. To choose the optimal path from this viable cluster, the path evaluation cost functions should be designed. The path evaluation cost functions include the comfort cost function, the velocity maintenance function, the lateral offset cost function, and the lane-changing efficiency cost function.

5.1. Comfort Cost Function

During the lane-changing process of intelligent semi-trailers, a high jerk can reduce driving comfort. We define the comfort cost function based on the lateral jerk and longitudinal jerk during lane changes. The comfort cost function is defined as:

$$J_c = k_1 \int_{t_0}^{t_e} \ddot{y}_t(t)dt + k_2 \int_{t_0}^{t_e} \ddot{x}_t(t)dt \tag{29}$$

where k_1 and k_2 are the weighting parameters, t_0 represents the initiation time of the lane change, and t_e represents the final time of the lane change.

5.2. Velocity Maintenance Function

During the process of changing lanes in intelligent semi-trailers, minimizing the disparity between the real driving velocity and the intended velocity is essential. We regard the deviation between the real velocity and the intended velocity as the cost function for maintaining velocity. This cost function for velocity maintenance is defined as:

$$J_e = k_3 \int_{t_0}^{t_e} \left(v_r - \sqrt{\dot{x}_t^2(t) + \dot{y}_t^2(t)} \right) dt \tag{30}$$

where k_3 is the weight parameter and v_r is the intended velocity.

5.3. Lateral Offset Cost Function

During the lane-changing process of intelligent semi-trailers, it is advisable to be as close to the centerline of the roadway as possible. We use the square of the lateral offset as the offset cost function. The offset cost function is defined as:

$$J_r = k_4 \int_{t_0}^{t_e} (y_t(t) - y_r)^2 dt \tag{31}$$

where k_4 is the weight parameter and y_r is the lateral coordinate of the centerline.

5.4. Lane-Changing Efficiency Cost Function

During the lane-changing process of intelligent semi-trailers, efforts should be made to shorten the lane-changing time as much as possible to improve lane-changing efficiency. We use the time difference between the initiation and completion of the lane change to represent the lane-changing efficiency cost. The lane-changing efficiency cost function is defined as:

$$J_t = k_5(t_e - t_0) \tag{32}$$

where k_5 is the weight parameter.

Consequently, the overall cost function can be represented as:

$$J = J_c + J_r + J_e + J_t \tag{33}$$

Each path within the feasible path cluster is analyzed, and the path with the lowest total cost function is chosen as the optimal option.

6. Simulation Validation

To assess how effective the proposed planning method is for intelligent semi-trailers across various driving conditions, simulation experiments are carried out. Firstly, a simulation experiment platform should be built. Then, the experiment settings should be made, and the experiments should be completed. Finally, the experimental results will be described and analyzed.

6.1. Simulation Experiment Design

6.1.1. Simulation Experiment Platform

MATLAB/TruckSim is utilized to conduct co-simulation experiments. These simulation experiments can establish real-time communication between the vehicle model and the planning method, ensuring the reliability of the experimental results. Figure 11 shows the illustration of the co-simulation experiments.

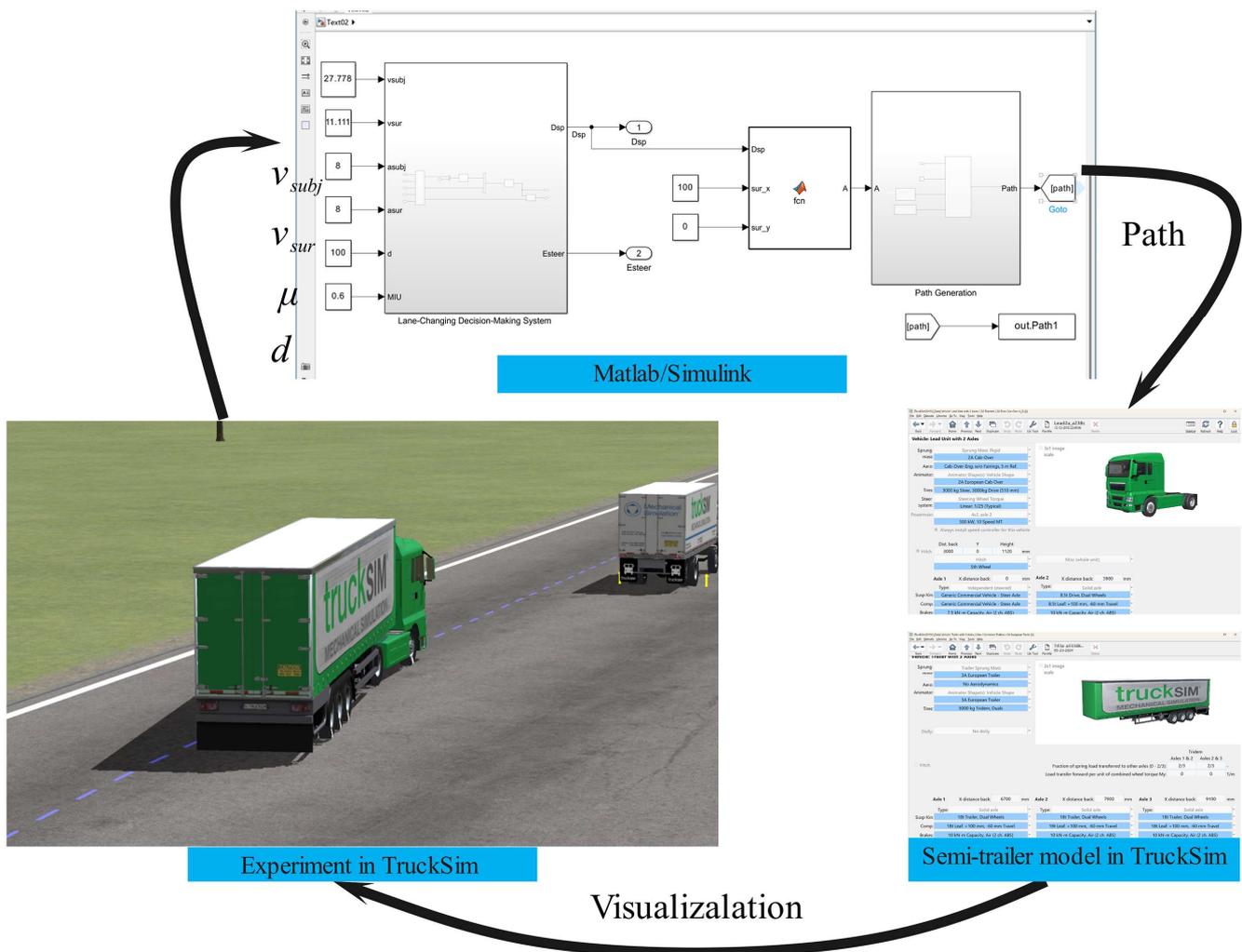


Figure 11. Illustration of the co-simulation experiments.

6.1.2. Case Study Design

Four case studies are designed to assess the effectiveness of the planning method under different common driving scenarios. In case study 1, μ is set to 0.6, the semi-trailer load mass is set to 40 t, and v_{subj} is set to 100 km/h. In case study 2, μ is set to 0.9, the semi-trailer load mass is set to 40 t, and v_{subj} is set to 100 km/h. In case study 3, μ is set to 0.9, the semi-trailer load mass is set to 40 t, and v_{subj} is set to 70 km/h. In case study 4, μ is set to 0.6, the semi-trailer load mass is set to 0 t, and v_{subj} is set to 100 km/h. The parameter settings for each case study are presented in Table 4.

Table 4. Simulation condition setting.

	μ	<i>mass</i>	v_{subj}	Result
Case study 1	0.6	40	100	Figure 12
Case study 2	0.9	40	100	Figure 13
Case study 3	0.9	40	70	Figure 14
Case study 4	0.6	0	100	Figure 15

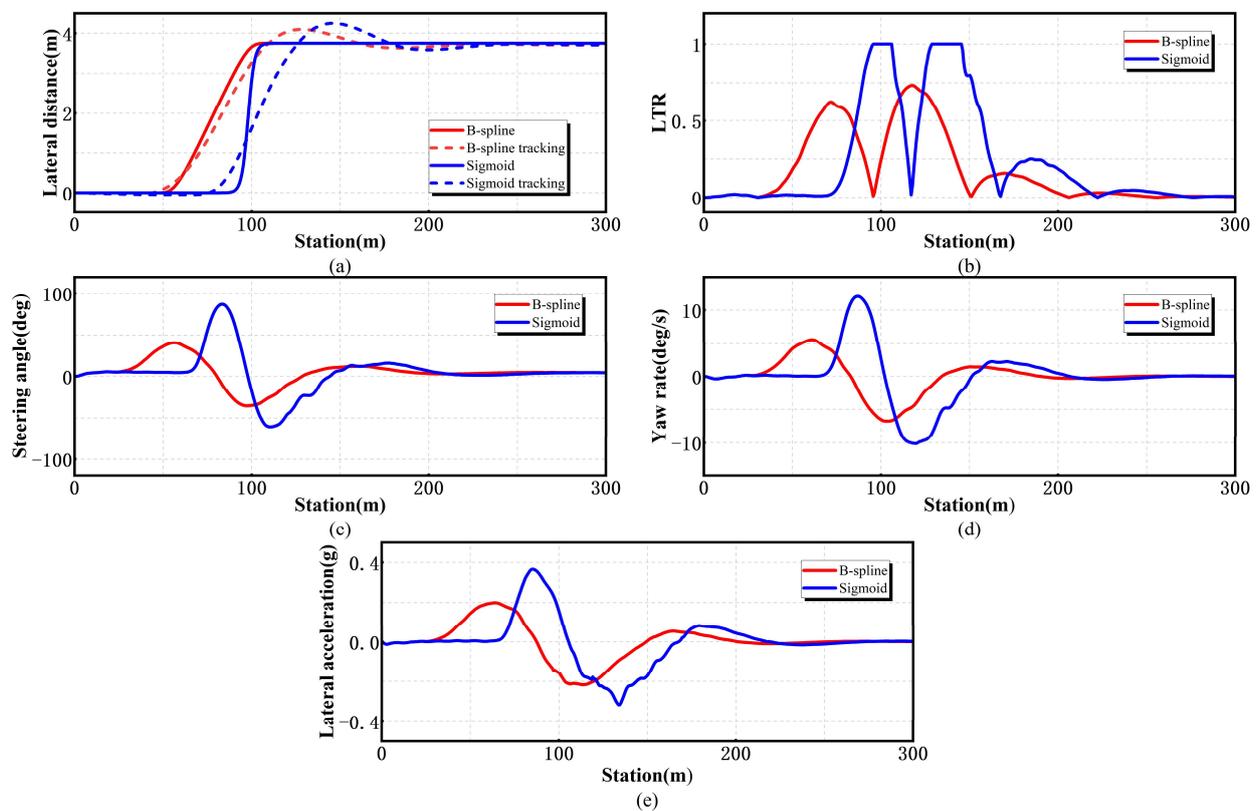


Figure 12. Comparison of the results of case study 1: (a) path tracking, (b) LTR, (c) steering angle, (d) yaw rate, and (e) lateral acceleration.

6.1.3. Comparative Verification

To demonstrate the benefits of our method, we will compare it with the optimization method based on the Sigmoid function [10], which is also utilized for the path planning of intelligent semi-trailers. Moreover, the planned paths will be tracked and verified by using the tracking controller that comes with TruckSim.

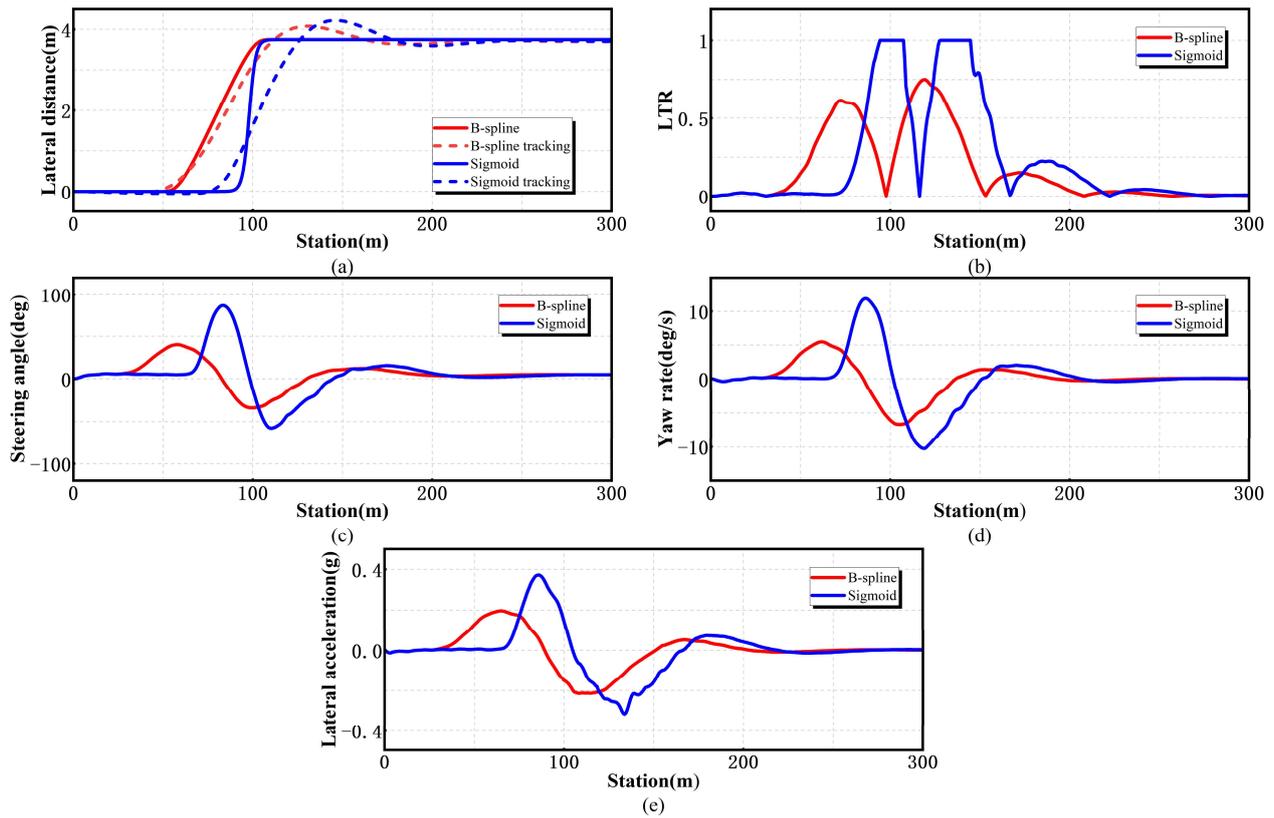


Figure 13. Comparison of the results of case study 2: (a) path tracking, (b) LTR, (c) steering angle, (d) yaw rate, and (e) lateral acceleration.

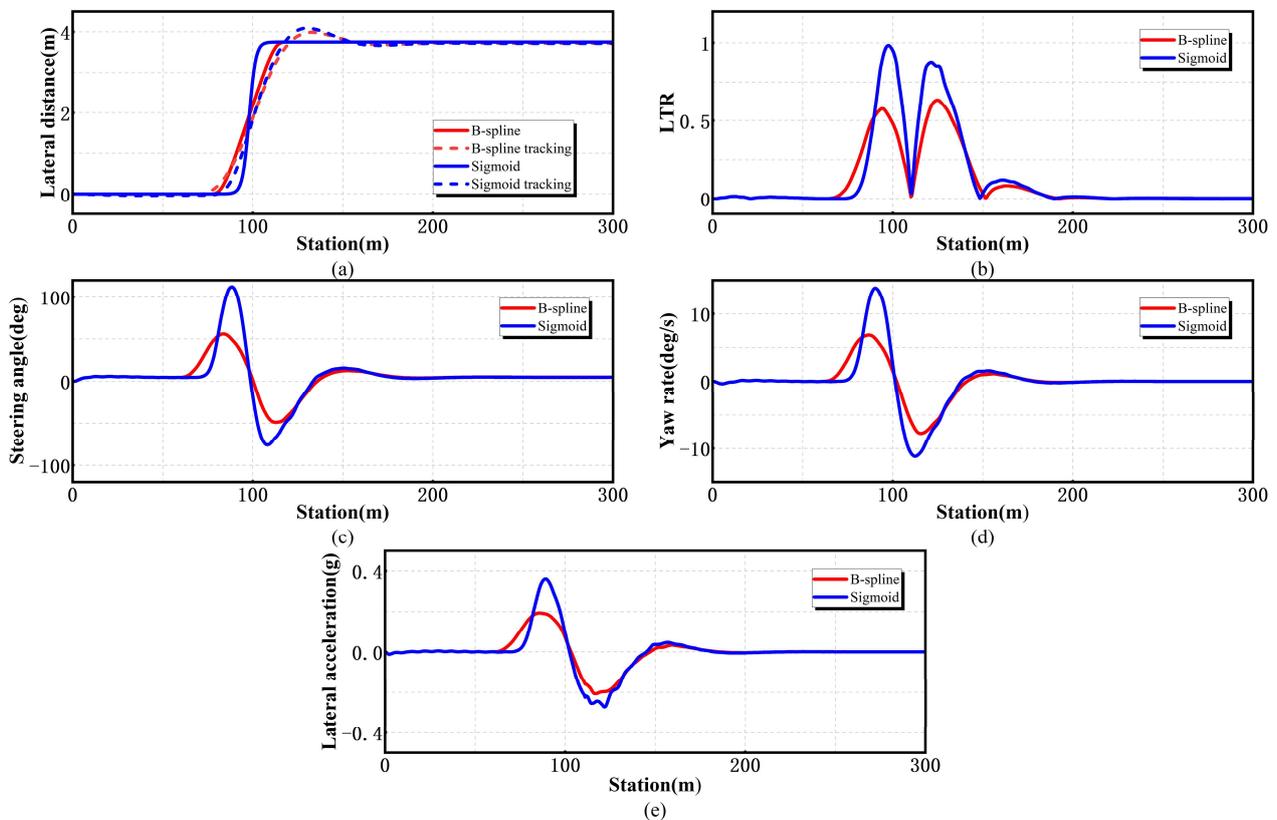


Figure 14. Comparison of the results of case study 3: (a) path tracking, (b) LTR, (c) steering angle, (d) yaw rate, and (e) lateral acceleration.

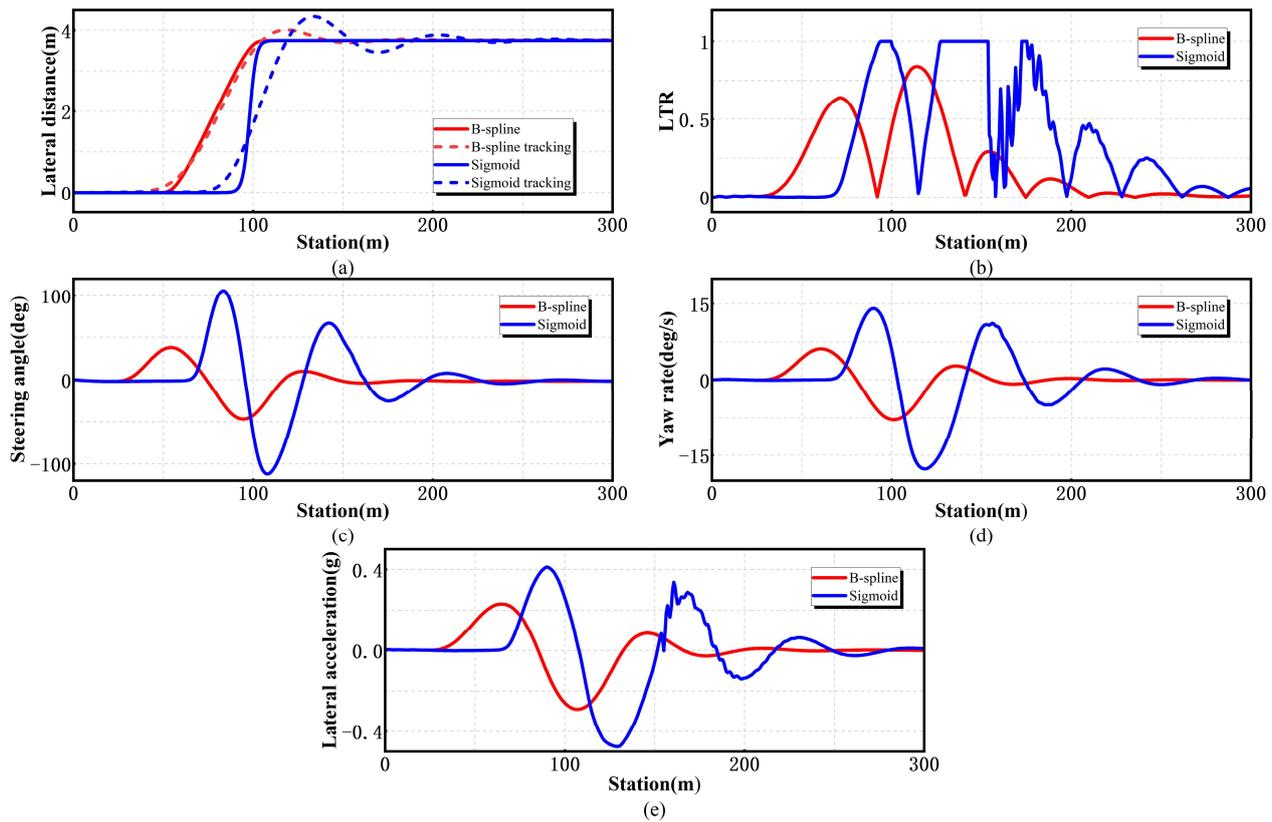


Figure 15. Comparison of the results of case study 4: (a) path tracking, (b) LTR, (c) steering angle, (d) yaw rate, and (e) lateral acceleration.

6.2. Experimental Results

6.2.1. Case Study 1

Figure 12 presents the results of case study 1. The case study is set with a road adhesion coefficient μ of 0.6, a semi-trailer load of 40 t, and a vehicle speed of 100 km/h. This figure compares the method in this paper with the Sigmoid function optimization method. In terms of path tracking, the trajectory of the method in this paper is better; in terms of LTR, the proposed method achieves a maximum LTR of 0.73249, significantly lower than the 1.0 threshold observed in the comparison method, indicating a lower rollover risk; the average steering angle, average yaw rate, and average lateral acceleration are also smaller than those of the comparison method, indicating that the method in this paper performs better in driving comfort and lateral dynamic stability. The comparison of the data from case study 1 can be found in Table 5.

Table 5. Comparison of case study 1 data.

Case Study 1	Our Method	Sigmoid Method
Average curvature of the path	0.00179	0.01775
Maximum of LTR	0.73249	1
Average steering angle	9.76623	12.39901
Average yaw rate	1.09961	1.60889
Average lateral acceleration	0.04277	0.05407

6.2.2. Case Study 2

Figure 13 shows the data from case study 2. The conditions are $\mu = 0.9$, semi-trailer load of 40 t, and vehicle speed of 100 km/h. As shown in the figure, the method in this paper has obvious advantages in path tracking; the maximum LTR value is 0.74934, which

is less than 1 of the comparison method, and the rollover stability is better; the average steering angle, average yaw rate, and average lateral acceleration are all smaller, providing a more comfortable driving experience and better lateral stability of the vehicle. The comparison of the data from case study 2 can be found in Table 6.

Table 6. Comparison of case study 2 data.

Case Study 2	Our Method	Sigmoid Method
Average curvature of the path	0.00175	0.00402
Maximum of LTR	0.74934	1
Average steering angle	7.70237	9.05435
Average yaw rate	0.62028	0.88300
Average lateral acceleration	0.02438	0.03051

6.2.3. Case Study 3

Figure 14 presents the results of case study 3. During the case study, $\mu = 0.9$, the semi-trailer load is 40 t, and the vehicle speed is 70 km/h. Comparing the two methods, the path tracking effect of the method in this paper is good; the maximum LTR value is 0.63285, which is lower than 0.98316 of the comparison method, and the rollover risk is low; the average steering angle, average yaw rate, and average lateral acceleration are all lower than those of the comparison method, and the driving comfort and lateral dynamic stability are better. The comparison of the data from case study 3 can be found in Table 7.

Table 7. Comparison of case study 3 data.

Case Study 3	Our Method	Sigmoid Method
Average curvature of the path	0.00351	0.00801
Maximum of LTR	0.63285	0.98316
Average steering angle	10.26115	12.53339
Average yaw rate	0.88337	1.20238
Average lateral acceleration	0.02678	0.03348

6.2.4. Case Study 4

Figure 15 shows the results of case study 4, where $\mu = 0.6$, the semi-trailer is unloaded, and the vehicle speed is 100 km/h. In terms of path tracking, the method proposed in this paper performs remarkably. The maximum LTR value of 0.83893 is significantly lower than the 1.0 threshold observed in the comparison method, indicating a certain advantage in rollover stability. Moreover, the average steering angle, average yaw rate, and average lateral acceleration are all smaller than those of the comparison method, suggesting better driving comfort and lateral dynamic stability. Under the specific conditions of case study 4, the LTR index in Figure 15b shows sharp fluctuations between 160–190 m. From the perspective of vehicle dynamics, when the load mass of the semi-trailer is 0 t, the position of the vehicle's center of gravity changes. Under high-speed (100 km/h) and low road adhesion coefficient ($\mu = 0.6$) conditions, the vehicle is more sensitive to steering operations. A small path adjustment may cause a large LTR change. The abrupt change in LTR poses significant risks to semi-trailer stability:

1. Increased lateral load transfer, leading to rollover propensity.
2. Sudden weight redistribution, compromising tire-road adhesion.
3. Dynamic instability during transient maneuvers due to coupled yaw-roll effects.

The comparison of the data from case study 4 can be found in Table 8.

Table 8. Comparison of case study 4 data.

Case Study 4	Our Method	Sigmoid Method
Average curvature of the path	0.00175	0.00395
Maximum of LTR	0.83893	1
Average steering angle	7.01638	16.21332
Average yaw rate	1.14704	2.80779
Average lateral acceleration	0.04652	0.08669

6.3. Analysis of Experimental Results

Figures 12–15 show the comparison results of the path planning method proposed in this paper and the optimization method based on the Sigmoid function under different experimental conditions. These figures intuitively present the performance of the two methods in the lane-changing process of intelligent semi-trailers from multiple aspects, such as path tracking, LTR, steering angle, yaw rate, and lateral acceleration, which is helpful for in-depth analysis of the advantages and characteristics of this method and the influence of different driving conditions on the path planning effect. We will analyze the proposed path planning method in terms of the adaptability to traffic conditions, the vehicle roll stability, the driving comfort, and the lateral dynamic stability.

6.3.1. Adaptability to Traffic Conditions

When facing different traffic conditions, the proposed planning method can be flexibly adjusted according to changes in driving parameters, planning lane-changing paths that meet the current traffic conditions.

6.3.2. Vehicle Roll Stability

As can be seen from the comparison chart of LTR, the proposed planning method can make the planned path meet the roll stability requirements by setting the roll stability boundary. Compared with the Sigmoid function optimization method, our method can significantly reduce the LTR during the driving process of the intelligent semi-trailer and ensure that it is below 0.9. This effectively supports the roll stability of the intelligent semi-trailer throughout its operation.

6.3.3. Driving Comfort

As can be seen from the comparison chart of steering angles, compared with the Sigmoid function optimization method, our planning method significantly reduces the average steering angle and has better driving comfort.

6.3.4. Lateral Dynamic Stability

As can be seen from the comparison charts of yaw rate and lateral acceleration, although the lane-changing distance of our planning method is longer than that of the Sigmoid function optimization method, it substantially decreases the average curvature of the path, thereby reducing the steering burden of the intelligent semi-trailer and improving the tracking performance of the intelligent semi-trailer.

7. Conclusions and Future Work

In this study, we proposed a path-planning method for intelligent semi-trailers that consider the surrounding traffic conditions and vehicle roll stability, aiming to address the issue of potential rollover during lane-changing in complex traffic. The experimental results comprehensively demonstrate the effectiveness of our proposed method across multiple aspects.

In terms of path tracking performance, as shown in the comparison with the Sigmoid function-based optimization method in different case studies (case study 1–4), our method exhibits a better-fitting trajectory. This indicates that our method can generate a smoother path, facilitating more accurate vehicle control during lane-changing maneuvers.

Regarding vehicle roll stability, a crucial aspect for semi-trailers, our method has achieved remarkable results. By setting the roll stability boundary based on the maximum LTR, we ensure that the planned paths meet the roll stability requirements. In all case studies, the maximum LTR values of our method are well below 0.9 (e.g., 0.73249 in case study 1, 0.74934 in case study 2, 0.63285 in case study 3, and 0.83893 in case study 4), while those of the comparison method are close to or reach 1. This significant reduction in LTR effectively supports the roll stability of intelligent semi-trailers throughout the driving process, greatly reducing the risk of rollover accidents.

Driving comfort is also improved by our path-planning method. The average steering angle, which is an important indicator of driving comfort, is significantly reduced compared to the Sigmoid-based method. A smaller steering angle not only reduces the driver's physical burden but also provides a more stable and comfortable driving experience for passengers or cargo.

In addition, the lateral dynamic stability of the vehicle is enhanced. Although the lane-changing distance of our method may be slightly longer in some cases, the substantial decrease in the average curvature of the path reduces the steering burden of the intelligent semi-trailer. As a result, the tracking performance of the vehicle is improved, as evidenced by the lower average yaw rate and lateral acceleration values in all case studies. This improvement in lateral dynamic stability is crucial for maintaining vehicle control and safety during lane changing.

Our method also shows a good adaptability to different traffic conditions. It can flexibly adjust the lane-changing paths according to various driving parameters such as the road adhesion coefficient, vehicle velocity, and load capacity. This adaptability ensures that the path-planning results are always suitable for the current traffic environment, further enhancing the practicality and reliability of the method.

However, it should be noted that this study has certain limitations. Although the current framework is mainly applicable to straight-road lane-changing scenarios with constant vehicle speeds, real-world driving conditions are much more complex, including curved roads, variable speeds, and dynamic interactions with other traffic participants. Moreover, simulation experiments rely on predefined environmental parameters, which may not fully reflect real-world unpredictability. Future work will extend the proposed method to address more complex scenarios, such as:

1. Curved road adaptation: Incorporating path planning for curved roads and dynamic speed adjustments.
2. Dynamic obstacle interaction: Enhancing the decision-making system to handle real-time interactions with moving obstacles and multi-vehicle coordination.
3. Multi-objective optimization: Integrating additional stability metrics (e.g., pitch stability) and energy efficiency considerations into the cost function.
4. Real-world validation: Conduct field tests with physical semi-trailers to validate the method's robustness under practical constraints.

In conclusion, this work lays a solid foundation for the lane-changing path planning of intelligent semi-trailers. By continuously improving and expanding the research, we can better meet the requirements of autonomous driving in commercial transportation and contribute to the development of a safer and more efficient transportation system.

Author Contributions: Conceptualization, Z.N.; Methodology, H.Z.; Validation, H.Z.; Writing—original draft preparation, H.Z.; Writing—review and editing, H.Z., Z.N., and Y.L.; Funding acquisition, Z.N. and Y.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (grant number 52262053), HORIZON EUROPE MSCA (grant number 101146785), and Xingdian Talent Support Planning Project.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

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

References

1. Tsai, C.-Y. Trajectory Planning of Semi-Trailer Truck Vehicle Based on Algebraic General Trajectory Formula. *IEEE Trans. Intell. Transport. Syst.* **2022**, *24*, 1495–1501. [[CrossRef](#)]
2. Wang, J.; Yuan, X. Adaptive Dynamic Path Planning Method for Autonomous Vehicle Under Various Road Friction and Speeds. *IEEE Trans. Intell. Transport. Syst.* **2023**, *24*, 10977–10987. [[CrossRef](#)]
3. Alarabi, S.; Santora, M. Review: Path Planning Techniques for Automated Guided Vehicles (AGVs). In Proceedings of the 2024 9th Asia-Pacific Conference on Intelligent Robot Systems (ACIRS), Dalian, China, 18–20 July 2024; pp. 33–38.
4. Yuan, X. Research on the Limitations of UAV Path Planning Based on Artificial Potential Field Method. In Proceedings of the 2022 9th International Forum on Electrical Engineering and Automation (IFEEA), Zhuhai, China, 4–6 November 2022; pp. 619–622.
5. Liu, C.; Zhai, L.; Zhang, X. Research on Local Real-Time Obstacle Avoidance Path Planning of Unmanned Vehicle Based on Improved Artificial Potential Field Method. In Proceedings of the 2022 6th CAA International Conference on Vehicular Control and Intelligence (CVCI), Nanjing, China, 28–30 October 2022; pp. 1–6.
6. Chen, Z.; Gao, Q.; Wang, X. Local Path Planning of Intelligent Vehicle Based on Improved Artificial Potential Field. In Proceedings of the 2020 IEEE 3rd International Conference on Electronic Information and Communication Technology (ICEICT), Shenzhen, China, 13–15 November 2020; pp. 110–116.
7. Xu, M.; Xu, C.; Qi, G. Dynamic Obstacle Avoidance Method for Road Vehicles Via Improved Artificial Potential Field. In Proceedings of the 2023 7th CAA International Conference on Vehicular Control and Intelligence (CVCI), Changsha, China, 27–29 October 2023; pp. 1–6.
8. Han, J.; Cui, M.; Lv, Y. Moving Horizon Path Planning for Intelligent Vehicle Oriented to Dynamic Obstacle Avoidance. In Proceedings of the 2022 6th CAA International Conference on Vehicular Control and Intelligence (CVCI), Nanjing, China, 28 October 2022; pp. 1–6.
9. Glaser, S.; Vanholme, B. Maneuver-Based Trajectory Planning for Highly Autonomous Vehicles on Real Road with Traffic and Driver Interaction. *IEEE Trans. Intell. Transport. Syst.* **2010**, *11*, 589–606. [[CrossRef](#)]
10. Asrofudin, B.; Widyotriatmo, A.; Siregar, P.I. Sigmoid Function Optimization for Path Following Control with Obstacle Avoidance of an Autonomous Truck-Trailer. In Proceedings of the 2021 International Conference on Instrumentation, Control, and Automation (ICA), Bandung, Indonesia, 25–27 August 2021; pp. 180–185.
11. Yue, M.; Hou, X. Robust Tube-Based Model Predictive Control for Lane Change Maneuver of Tractor-Trailer Vehicles Based on a Polynomial Trajectory. *IEEE Trans. Syst. Man Cybern. Syst.* **2020**, *50*, 5180–5188. [[CrossRef](#)]
12. Hoek, R.; Ploeg, J. Cooperative Driving of Automated Vehicles Using B-Splines for Trajectory Planning. *IEEE Trans. Intell. Veh.* **2021**, *6*, 594–604.
13. Li, H.; Luo, Y. Collision-Free Path Planning for Intelligent Vehicles Based on Bézier Curve. *IEEE Access* **2019**, *7*, 123334–123340. [[CrossRef](#)]
14. Suzuki, T.; Usami, R. Automatic Two-Lane Path Generation for Autonomous Vehicles Using Quartic B-Spline Curves. *IEEE Trans. Intell. Veh.* **2018**, *3*, 547–558. [[CrossRef](#)]
15. Zhao, W.; Guo, H.; Zhao, X. Intelligent Vehicle Path Planning Based on Q-Learning Algorithm with Consideration of Smoothness. In Proceedings of the 2020 Chinese Automation Congress (CAC), Shanghai, China, 6–8 November 2020; pp. 4192–4197.
16. Zhang, D.; Chen, B. Path Planning and Predictive Control of Autonomous Vehicles for Obstacle Avoidance. In Proceedings of the 2022 18th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA), Taipei, Taiwan, 28–30 November 2022; pp. 1–6.

17. Jiang, C.; Hu, Z. R2-RRT*: Reliability-Based Robust Mission Planning of Off-Road Autonomous Ground Vehicle Under Uncertain Terrain Environment. *IEEE Trans. Automat. Sci. Eng.* **2022**, *19*, 1030–1046. [[CrossRef](#)]
18. Fu, S. Robot Path Planning Optimization Based on RRT and APF Fusion Algorithm. In Proceedings of the 2024 8th International Conference on Robotics and Automation Sciences (ICRAS), Tokyo, Japan, 21–23 June 2024; pp. 32–36.
19. Wang, Z.; Li, P. APG-RRT: Sampling-Based Path Planning Method for Small Autonomous Vehicle in Closed Scenarios. *IEEE Access* **2024**, *12*, 25731–25739. [[CrossRef](#)]
20. Ju, C.; Luo, Q.; Yan, X. Path Planning Using Artificial Potential Field Method And A-Star Fusion Algorithm. In Proceedings of the 2020 Global Reliability and Prognostics and Health Management (PHM-Shanghai), Shanghai, China, 16–18 October 2020; pp. 1–7.
21. Qing, G.; Zheng, Z.; Yue, X. Path-Planning of Automated Guided Vehicle Based on Improved Dijkstra Algorithm. In Proceedings of the 2017 29th Chinese Control and Decision Conference (CCDC), Chongqing, China, 28–30 May 2017; pp. 7138–7143.
22. Chen, Y.; Xiao, H. Global Dynamic Path Planning Based on Fusion of Improved A* Algorithm and Morphing Algorithm. In Proceedings of the 2019 Chinese Control and Decision Conference (CCDC), Nanchang, China, 3–5 June 2019; pp. 6191–6196.
23. Huang, H.; Huang, P.; Zhong, S. Dynamic Path Planning Based on Improved D* Algorithms of Gaode Map. In Proceedings of the 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), Chengdu, China, 15–17 March 2019; pp. 1121–1124.
24. Hu, J.; Zhu, Q. Multi-Objective Mobile Robot Path Planning Based on Improved Genetic Algorithm. In Proceedings of the 2010 International Conference on Intelligent Computation Technology and Automation, Changsha, China, 11–12 May 2010; Volume 2, pp. 752–756.
25. Sari, D.W.; Dwijayanti, S.; Suprpto, B.Y. Path Planning for an Autonomous Vehicle Based on the Ant Colony Algorithm: A Review. In Proceedings of the 2023 International Workshop on Artificial Intelligence and Image Processing (IWAIPP), Yogyakarta, Indonesia, 1–2 December 2023; pp. 57–62.
26. Yang, M.; Li, C. Path Planning and Tracking for Multi-Robot System Based on Improved PSO Algorithm. In Proceedings of the 2011 International Conference on Mechatronic Science, Electric Engineering and Computer (MEC), Jilin, China, 19–22 August 2011; pp. 1667–1670.
27. Ning, X.; Li, Y.; Liu, Z. Improved Genetic Algorithm-Based Obstacle Avoidance Path Planning Method for Inspection Robots. In Proceedings of the 2023 2nd International Symposium on Control Engineering and Robotics (ISCER), Hangzhou, China, 17–19 February 2023; pp. 346–350.
28. Shao, X.; Lv, Z.; Zhao, X. Research on Robot Path Planning Based on Improved Adaptive Ant Colony Algorithm. In Proceedings of the 2019 Chinese Control and Decision Conference (CCDC), Nanchang, China, 3–5 June 2019; pp. 506–510.
29. Gong, Y.; Luo, M.; Wang, C. A Path Planning Method Based on Improved Particle Swarm Optimization Algorithm. In Proceedings of the 2020 7th International Conference on Information Science and Control Engineering (ICISCE), Changsha, China, 18–20 December 2020; pp. 106–110.
30. Hou, Y.; Xu, X. High-Speed Lateral Stability and Trajectory Tracking Performance for a Tractor-Semitrailer with Active Trailer Steering. *PLoS ONE* **2022**, *17*, e0277358. [[CrossRef](#)] [[PubMed](#)]
31. Manav, A.C.; Lazoglu, I.; Aydemir, E. Adaptive Path-Following Control for Autonomous Semi-Trailer Docking. *IEEE Trans. Veh. Technol.* **2022**, *71*, 69–85. [[CrossRef](#)]
32. Han, S.; Park, G.; Ahn, Y. Estimated State-Based Optimal Path Planning and Control System for Lane-Keeping of Semi-Trailer Trucks. In Proceedings of the 2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC) IEEE, Bilbao, Spain, 24 September 2023; pp. 918–924.
33. Feng, G. A Survey on Analysis and Design of Model-Based Fuzzy Control Systems. *IEEE Trans. Fuzzy Syst.* **2006**, *14*, 676–697. [[CrossRef](#)]
34. Bhattacharyya, R.; Mukherjee, S. Fuzzy Membership Function Evaluation by Non-Linear Regression: An Algorithmic Approach. *Fuzzy Inf. Eng.* **2020**, *12*, 412–434. [[CrossRef](#)]
35. Zuo, Z.; Yang, X. MPC-Based Cooperative Control Strategy of Path Planning and Trajectory Tracking for Intelligent Vehicles. *IEEE Trans. Intell. Veh.* **2021**, *6*, 513–522. [[CrossRef](#)]
36. Oussalah, M.; Nguyen, H.T.; Kreinovich, V. A New Derivation of Centroid Defuzzification. In Proceedings of the 10th IEEE International Conference on Fuzzy Systems (Cat. No.01CH37297), Melbourne, VIC, Australia, 2–5 December 2001; Volume 2, pp. 884–887.
37. Wan, N.; Xu, D.; Ye, H. Improved Cubic B-Spline Curve Method for Path Optimization of Manipulator Obstacle Avoidance. In Proceedings of the 2018 Chinese Automation Congress (CAC), Xi'an, China, 30 November–2 December 2018; pp. 1471–1476.
38. Haris, M.; Nam, H. Path Planning Optimization of Smart Vehicle with Fast Converging Distance-Dependent PSO Algorithm. *IEEE Open J. Intell. Transp. Syst.* **2024**, *5*, 726–739. [[CrossRef](#)]
39. Vaswani, A.; Shazeer, N.; Parmar, N. Attention Is All You Need. In Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 6000–6010.

40. Chen, B.-C.; Peng, H. Differential-Braking-Based Rollover Prevention for Sport Utility Vehicles with Human-in-the-Loop Evaluations. *Veh. Syst. Dyn.* **2001**, *36*, 359–389. [[CrossRef](#)]
41. Li, H.; Li, J.; Su, Z.; Wang, X.; Luo, J. Research on Active Obstacle Avoidance Control Strategy for Intelligent Vehicle Based on Active Safety Collaborative Control. *IEEE Access* **2020**, *8*, 183736–183748. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.