

Vehicle Obstacle Avoidance Path Planning Method Based on Finite State Machine and Third-order Bézier Curve

ZhongZhi ZHANG¹, XueMei PENG², Yan XU¹

¹ Geely University of China, Chengdu, 641423, China
zzzly310@126.com, 859629225@qq.com

² Chengdu Boe Hospital, No.1 Jingxing Road, Tianfu International Bio-town, Chengdu, 610200, China
46897759@qq.com(*Corresponding author)

Abstract: Currently, vehicle obstacle avoidance path planning methods are facing problems such as a long planning time and a low planning effectiveness. This study optimizes the curve of the finite state machine during state transitions by utilizing third-order Bézier curves that can connect the discrete behaviors of objects and obtain the optimal state transition path. Then, the optimized finite state machine is applied to path planning in order to solve various problems related to vehicle obstacle avoidance path planning. The experimental results showed that for the AgiBot World dataset, the proposed TBC-FSM method obtained a calculation time lower than 90 ms, and an accuracy of curve generation higher than 95%. Moreover, the smoothness score for the planned motion trajectories ranged between 8 and 9, indicating a relatively high smoothness. When analyzing the actual effectiveness of the proposed method, the obstacle avoidance success rate for the planned path reached 98.7%, and the path planning time was only 45.6 ms. To sum up, the proposed optimization method can improve the accuracy of obstacle avoidance for vehicles and reduce the path planning time.

Keywords: Finite state machine; Third-order Bézier curve; Vehicle obstacle avoidance; Path planning; Planning time consumption.

1. Introduction

The development of society has led to the continuous expansion of the use of vehicles and other means of transportation. However, there are various obstacles, such as pedestrians and traffic accidents on urban and rural roads, which seriously affect the safety of vehicle travel (Zhang, Wan & Zhang, 2025). To ensure traffic safety and improve vehicle efficiency, it is necessary to perform the real-time detection of these obstacles and plan Obstacle Avoidance Paths (OAPs) for vehicles based on the detection results (Han & Song, 2024). With the advancement of modern intelligent technology, Artificial Intelligence (AI) technology has been gradually integrated into various industries, and intelligent vehicles are representative of this industry (Simo et al., 2024). How to effectively plan OAPs for intelligent vehicles has become the key to the development of this field (Wang, S. et al., 2023). At present, although some intelligent vehicles use AI technology to plan OAPs, those methods still have problems, including a weak planning stability, poor path planning effectiveness, and a long planning time. It is urgent to improve the current vehicle OAP planning method (Okereke et al., 2023). A Finite State Machine (FSM) includes states, events, and actions, which trigger state transitions and execute the corresponding actions through events to adjust the execution of actions (Yang, Gao & Ren, 2023). The Third-order Bézier Curve (TBC) can generate smooth curves through starting and ending points and two control points, providing theoretical support for path planning (Bungey et al., 2026). This study combines TBC

with FSM and proposes a vehicle OAP planning method based on TBC-FSM to improve the efficiency of vehicle OAP planning and ensure traffic safety. The innovation of the research lies in using FSM to analyze the state of the vehicle's OAP and determine whether there are obstacles in the vehicle's driving path. It then adjusts the control points in the TBC based on the detection results, thereby adjusting the curve shape at the starting and ending points of the path to find the optimal OAP.

The remainder of this paper is structured as follows. Section 2 includes a literature review, which describes the relevant references related to the research object and research methods. Section 3 introduces the research methods, elaborates on the TBC-FSM combination method and explains how it is used for vehicle OAP. Section 4 tests the research methods and analyzes the actual effectiveness of OAPs based on TBC and FSM. Finally, Section 5 includes the conclusion of this paper and it also outlines possible future research directions.

2. Literature Review

With the improvement of transportation convenience, an increasing number of people have conducted research on vehicle OAP planning methods. For example, Mao et al. (2024) proposed a dynamic event-triggered guidance control, utilizing LOS/APF to achieve path tracking and obstacle avoidance for underactuated USVs, and

ensuring accuracy and reducing the communication burden caused by frequent event triggering through sliding mode control and disturbance observation. However, its guidance-control framework designed for underactuated surface vehicles still requires adaptation for vehicle obstacle avoidance path planning. Luo et al. (2024) proposed model-free sliding mode predictive adaptive control, relying solely on input-output data to achieve multi-USV formation and obstacle avoidance; however, predictive control and extended state observer structures introduce a significant online computational burden. In addition, to plan the obstacle avoidance trajectory of an autonomous vehicle during driving, Kim et al. (2023) proposed a vehicle OAP planning method based on a lateral model. The path planning effect was good, but the stability of path planning in complex environments was poor. Li, Z. et al. (2023) introduced an OAP planning method based on a target risk estimator and a multidimensional constraint set to address the multi-track problem in OAP planning for autonomous vehicles. This method could solve the multi-orbit problem in OAP, but it featured a high computational complexity and resource consumption.

FSM is an abstract computational model widely used in various state transition scenarios Xu et al. (2023). For example, Kibria et al. (2023) used FSM to evaluate the security of chip systems. The results indicate that the accuracy of this evaluation method is relatively high. In addition, Pang et al. (2023) also used FSM models to control the state of inverters. The results showed that the power loss was reduced by 78.3%. The TBC is a parameterized curve that is widely used in various fields (Shang et al., (2024). For example, Ghannou et al. (2024) analyzed the annotation framework and regions using a curve-based method during offshore trajectory mining. In addition, Wang, R. et al. (2023) also used TBC to plan the water quality monitoring route for biomimetic underwater vehicles. The results showed that the accuracy of the path planning method could reach 94.3% .

In summary, the current vehicle OAP planning method still has problems such as low obstacle avoidance efficiency, a long OAP planning time, high path calculation complexity, and high resource consumption. For addressing these issues, this study combines FSM with TBC to construct the TBC-FSM method, which improves the efficiency of path planning and reduces the path planning time.

To that, although there are currently vehicle path obstacle avoidance methods based on particle swarm optimization, deep reinforcement learning, lateral models, and target risk constraints, the above analysis shows that these methods still have problems such as low obstacle avoidance efficiency, long path planning time, high path calculation complexity, and high resource consumption. Thus, it is necessary to propose a new method for solving the problems related to the current vehicle OAP planning. This study combines FSM with TBC to construct a TBC-FSM vehicle OAP planning method in order to improve path planning efficiency and reduce path planning time. The contribution of this study lies in the deep integration of FSM and TBC, and the construction of an OAP planning method that fills the theoretical gap between state switching and trajectory smoothness in traditional methods, providing a new paradigm for path planning in complex dynamic environments.

3. Research Methodology

3.1 FSM Optimized Based on TBC

The popularity of road vehicles and smart cars is constantly increasing, but with the expansion of its usage, the problem of road traffic congestion is becoming increasingly prominent (Zhou, Olofsson & Frisk, 2023; Zhai et al., 2024). Therefore, it is urgent to plan the traffic routes of vehicles in order to optimize the distribution of traffic flow, improve road utilization, and reduce traffic congestion (Zhu et al., 2024). Vehicle OAP planning refers to using intelligent algorithms to determine the optimal driving route of the vehicle based on a given map and transportation network to improve the transportation efficiency and safety (Li, H. et al., 2023). However, many current path planning methods still feature a low efficiency and an inaccurate path planning with regard to OAP planning, and further optimization of the vehicle OAP planning method is needed (Yang et al., 2024). FSM is a mathematical or computational model utilized to describe the behavior of a system transitioning between a finite number of states according to certain rules. It achieves the real-time control of complex processes by defining clear states and transition conditions for vehicle obstacle avoidance, thereby achieving the effect of obstacle avoidance. The basic structure of FSM is shown in Figure 1.

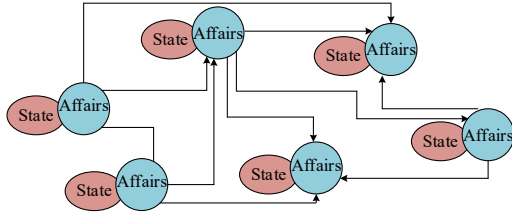


Figure 1. Basic framework of FSM

In Figure 1, the basic principle of FSM refers to the process of mutual influence between different transactions through their states. The term "Affairs" refers to various events, operations, or external factors involved in a system that affect the transition of a state machine between different states. For example, in vehicle OAP planning, it can be defined as the distance between obstacles detected by vehicle sensors, commands input by the driver, and preset conditions. In FSM, "Affairs" is closely related to "State" and "transfer conditions". Each state defines the behavior of the system in a specific situation, and transactions are the key factor triggering state transitions. When a transaction occurs, FSM will transition from the current state to the next state according to pre-defined transition rules and will perform corresponding output operations. As shown in Figure 1, FSM consists of a state set, an input set, a transition function, an initial state, and an output. The state set defines a state machine, which refers to all states that can exist in real-world problems. The input set represents external events or conditions that trigger state transitions, and it can be used to input conditions for users, sensor signals, and pre-set conditions. The transition function is a rule that describes how the current state transitions to the next state after receiving data from the input set. It is expressed by equation (1):

$$f: S_{\text{current}} \times \text{Input} \rightarrow S_{\text{next}} \quad (1)$$

In equation (1), f is the transition function, S_{current} is the current state, Input is the current input, and S_{next} represents the next state. The initial state is the starting point of the state machine's operation and the starting point of state transition. The output set refers to the set of all possible outputs that a state machine can generate, which are the actions or outputs that the state machine performs or generates in a specific state or during a state transition. This output set is generally divided into two types: Moore type (whose output relies only on the current state) and Mealy type (whose output

depends on the current state value and input). The calculation of the Moore type output is given by equation (2):

$$O_{\text{Moore}} = h(S_{\text{current}}) \quad (2)$$

In equation (2), O_{Moore} represents the output of the Moore type, and h is the output function. The formula for the Mealy type output is given by equation (3):

$$O_{\text{Mealy}} = h(S_{\text{current}}, I) \quad (3)$$

In equation (3), O_{Mealy} is the output of the Mealy type, and I is the current input set. FSM describes the motion behavior of an object as a state, thereby controlling the motion of the object based on transition rules, achieving the effect of path obstacle avoidance, and innovatively controlling the relative motion direction of the object. It can achieve the real-time control of complex processes by defining clear states and transition conditions for vehicle obstacle avoidance, thereby achieving the effect of obstacle avoidance (Khan & Wen, 2023). The output of the state machine can be obtained through the above calculations. FSM can process the discrete behavior of objects, but it also requires adding continuous and smooth trajectories to these discrete behaviors, connecting them to form a whole. TBC is a parameterized curve defined by four control points that can generate smooth curves, thereby achieving the goal of dynamically connecting the discrete behaviors of objects. The curve generation process and results for TBC are shown in Figure 2.

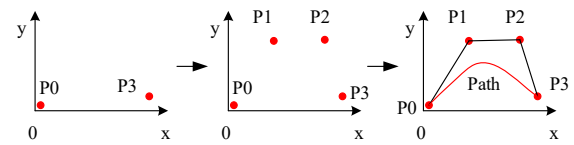


Figure 2. TBC drawing process

In Figure 2, when drawing a curve, it is necessary to first determine the starting point $P0$ and ending point $P3$ of the curve, and then provide any number of control points for the curve. If the control point is 0, it is a linear Bézier curve; if the control point is 1, it is a second-order Bézier curve; if the number of control points is 2, it is TBC. This study uses TBC and represents the control points as $P1$ and $P2$. The parametric equation of TBC is expressed as equation (4):

$$B(t) = (1-t)^3 P0 + 3(1-t)^2 t P1 + 3(1-t) t^2 P2 + t^3 P3, t \in [0,1] \quad (4)$$

In equation (4), $B(t)$ is a parameterized curve function. $(P0, P1, P2, P3)$ represent the starting point, control point 1, control point 2, and the endpoint in the curve. t is a random parameter. After determining the four points, the curve needs to be discretized into straight segments through sampling to draw a continuous curve. In this process, it is necessary to select n equally spaced t values based on accuracy, substitute each t value into the equation, calculate its $B(t)$ coordinates, and obtain a series of points. This series of points is connected by straight lines, and the continuous curve is approximated by straight segments to obtain a complete curve, namely TBC. The FSM improved based on TBC connects the discrete behaviors in the state machine using TBC to obtain continuous and smooth trajectories for object path planning. The basic process of FSM improved based on TBC is shown in Figure 3.

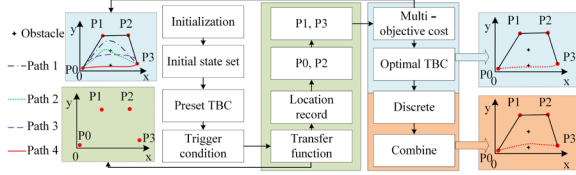


Figure 3. Basic process of TBC-FSM

As shown in Figure 3, TBC-FSM is first initialized, then it defines an initial set of states, and sets a TBC template for each possible state transition. When a specific condition is triggered, the state transfer is started, the state machine calls the transfer function, and the position at the time when the condition is triggered is recorded as the initial position $P0$ of the TBC. Based on the constraints of the actual situation, multiple sets of candidate points $P1$ and $P2$ in the TBC are selected, each corresponding to a TBC. The next step is to calculate the multi-objective cost of each curve to select the optimal TBC, as shown in equation (5):

$$J = w_1 \cdot \text{Length}(B(t)) + w_2 \cdot \text{Curvature}^2 + w_3 \cdot \text{CollisionRisk} + w_4 \cdot \text{Time} \quad (5)$$

In equation (5), J is the multi-objective cost, Length is the curve arc length, and Curvature is the square integral of curvature. CollisionRisk is the reciprocal of the minimum distance to the obstacle, and Time denotes the execution time. w_1 , w_2 , w_3 , and w_4 represent weight coefficients. When J is minimized, this is the optimal transfer trajectory. The optimal transfer trajectory can be regarded as the initial path for solving vehicle OAP planning. Afterwards, the optimal TBC is selected and discretized into multiple points

according to a fixed step size, and the multiple points are combined. When there is a point deviating from the trajectory, the optimal TBC needs to be re-selected. If a new obstacle appears on the trajectory, the current transition of the state machine will be terminated in advance and a new transition will be initiated. Finally, a complete TBC is obtained.

For example, when planning OAP for a vehicle, if the vehicle needs to move from the starting point to the end point and needs to bypass obstacles along the way, FSM can be used to define behaviors such as driving in a straight line, turning left, turning right, and parking. When a vehicle obstacle approaches, the sensors in the vehicle detect the obstacle, triggering an input event, and the FSM will transition to left turn, right turn, or stop state. After completing this state, the vehicle will transition back to straight ahead state. When the vehicle switches from moving in a straight line to other states, a transition curve is generated using the TBC. The position and direction at the end of the straight-line motion state are set as the starting point of the TBC, and the position and direction at the beginning of the next state are set as the endpoint of the TBC. The two control points between the two lines are used for adjusting the curvature and smoothness of the curve, dealing with obstacles encountered in unexpected situations. Similarly, vehicles also use this method for OAP planning when returning to driving in a straight line.

3.2 Vehicle OAP Planning Method Based on FSM and TBC

The TBC-FSM method can detect obstacles on the path and plan an appropriate route, while the current vehicle OAP planning method still needs to be adjusted as regards its planning effectiveness. The paper proposes a vehicle OAP planning method on the basis of TBC-FSM to find the optimal path and reduce traffic congestion (Li et al., 2024; Yang et al., 2025). The process for the TBC-FSM method is displayed in Figure 4.

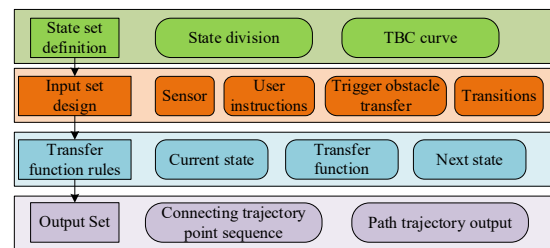


Figure 4. Basic process of TBC-FSM method

As shown in Figure 4, this process mainly consists of state set definition, input set design, transition function rule design, and output set. In the state set definition, the vehicle OAP is divided into different states, each corresponding to a different stage of TBC generation. The vehicle OAP consists of an initial state, a path generation state, an obstacle avoidance state, and a path completion state. In the initial state, the state machine defines the triggering conditions for state transitions while waiting for system instructions. For example, a state transition is triggered when a sensor detects the disappearance of an obstacle or a change in the environment. The path generation state obtains the initial control points of TBC based on the starting point, end point, and initial environment information, and discretizes the curve to generate a series of trajectory points. The discretization formula for trajectory points is shown in equation (6):

$$t_i = \frac{i}{N}, i = 0, 1, \dots, N \quad (6)$$

In equation (6), t_i is the sampling point and N denotes the total number of sampling points. After obtaining the discrete trajectory points, when generating the trajectory point sequence, if the sensor or camera on the trajectory point detects an obstacle, the state transition condition is triggered and it transitions to the obstacle avoidance state. If the trajectory points are connected without obstacles, the vehicle state transitions to the path completion state. When transitioning to the obstacle avoidance state, this stage dynamically adjusts the TBC so that the final TBC can avoid obstacles while maintaining the smoothness of the path trajectory. When an obstacle is detected, its position on the path is converted into an offset based on the initial control points $P1$ and $P2$. The expression for the offset of obstacles relative to the curve is shown in equation (7):

$$\begin{cases} \Delta x = x0 - Bx(t^*) \\ \Delta y = y0 - By(t^*) \end{cases} \quad (7)$$

In equation (7), $B(t^*)$ is the parametric equation of the initial curve at parameter t^* . $Bx(t^*)$ and $By(t^*)$ are the horizontal and vertical coordinates of $B(t^*)$. $(x0, y0)$ are the coordinates of the obstacle. Δx and Δy denote the horizontal and vertical axis offsets. If the obstacle is located on the right side of the initial TBC, the control points $P1$ and $P2$ will shift to the left. For the case when the

obstacle deviates to the right and left sides, the adjustment method for the control point is shown in equation (8):

$$\begin{cases} P1' = P1 + (-\alpha\Delta x, -\beta\Delta y), P2' = P2 + (-\alpha\Delta x, -\beta\Delta y) \\ P1'' = P1 + (\alpha\Delta x, \beta\Delta y), P2'' = P2 + (\alpha\Delta x, \beta\Delta y) \end{cases} \quad (8)$$

In equation (8), α and β are adjustment coefficients that control the offset amplitude. $P1'/P2'$ and $P1''/P2''$ are the offsets of control point 1 and control point 2 when shifted to the left and right. Moreover, the offset process maintains the starting and ending points unchanged, and limits the adjustment range for control points to avoid the situation when the curves would exceed the road boundary. The constraint expression for adjusting the range of control points is given in equation (9):

$$\|Pm' - Pm\| \leq d \max, m = 1, 2 \quad (9)$$

In equation (9), $d_{\max}$ is the maximum allowable offset distance. Pm' is the new control point obtained by adjusting control point Pm . In addition to constraining the path boundary, the smoothness of the curve also needs to be constrained. The constraint formula for the curvature change after adjusting the control point is shown in equation (10):

$$k(t) = \frac{\|B'(t) \times B''(t)\|}{\|B'(t)\|^3} \quad (10)$$

In equation (10), $B'(t)$ and $B''(t)$ denote the 1st and 2nd derivatives of the curve, respectively. $k(t)$ represents curve curvature. By adjusting $k(t)$ to be less than the maximum allowable curvature, the vehicle can avoid sudden turns during driving. Finally, based on the adjusted control points, the trajectory point sequence is recalculated. The above steps are repeated until all obstacles are bypassed. When the vehicle reaches the endpoint, the optimal trajectory point sequence is output, which is the optimal path route. In input set design, sensors are used for detecting obstacles and triggering state transitions, or paths are re-planned based on user instructions to trigger state transitions. The function transition rule describes how the current state transitions to the next state after receiving input, and combines state transition rule with TBC to achieve dynamic path generation. The formula is given by equation (11):

$$\begin{cases} P1_f = P1 + \Delta x \\ P2_f = P2 + \Delta y \end{cases} \quad (11)$$

In equation (11), $P1_f$ and $P2_f$ are the final coordinates of the control points 1 and 2. The output set is meant to output the complete path trajectory when the trajectory sequence can be fully connected, to obtain the vehicle OAP. By following the above steps and continuously calculating, the final vehicle OAP is obtained. For example, when the sensor detects an obstacle directly ahead or when overtaking is required during vehicle operation, the command is transmitted as input to TBC-FSM. After triggering the state transition condition, the path generation status transitions to the obstacle avoidance state, and the actual obstacle offset is calculated to adjust the control points of TBC and generate a smooth trajectory point sequence that bypasses the obstacle.

4. Results and Discussion

4.1 Performance Analysis for TBC-FSM

To analyze the performance of TBC-FSM, this study focused on the curve generation speed, smoothness, stability, and response speed of the TBC-FSM method. Table 1 shows the parameter configuration during the experiment.

Table 1. Experimental Environment Configuration

Environment	Index	Configuration
Hardware	CPU	Intel i9-13900K
	GPU	NVIDIA RTX 4090
	EMS	64GB DDR5
	OS	Windows10
Software environment	C++	VC++6.0
	Python	Python 3.11

AgiBot World was selected as the experimental dataset, which contains 1 million real robot motion trajectories, and used this motion trajectory as the

dataset for testing the performance of TBC-FSM. The initial starting coordinates for FSM were set to (0,0,0), and the minimum distance between the actuator and the target was set to 0.05m. The performance of TBC-FSM was analyzed based on the above experimental environment and dataset. The dataset included 1,000 motion trajectories from home robots (A), industrial robots (B), logistics robots (C), and catering route planning robots (D) in AgiBot World, which were divided into training and testing sets in a 7:3 ratio. Firstly, the calculation time and trajectory smoothness results for the TBC-FSM trajectory generation are illustrated in Figure 5.

As shown in Figure 5(a), when using the TBC-FSM method to plan robot motion trajectories for datasets A, B, C, and D, the average planning times are 50.2 ms, 81.2 ms, 73.2 ms, and 63.7 ms, respectively. The TBC-FSM method has obtained a relatively low planning time for the trajectories of all four types of robots. After multiple tests, the fluctuation of each planning time for TBC-FSM is relatively low, and the planning stability is high. Trajectory smoothness can indicate the continuity of a motion trajectory and it can be rated by changes in path angle, curvature, and acceleration between paths. The higher the score, the stronger the continuity of the movement path and the better the performance related to movement efficiency. As regards Figure 5(b), the smoothness scores for the motion trajectories planned by TBC-FSM fluctuate between 8 and 9, indicating that the smoothness of the trajectories planned by this method is relatively high and the continuity of the motion path is strong. The stability of its trajectory path and the accuracy of the TBC-FSM curve generation under dynamic obstacles, sensor noise, and complex scene interference conditions are experimentally analyzed, as shown in Figure 6.

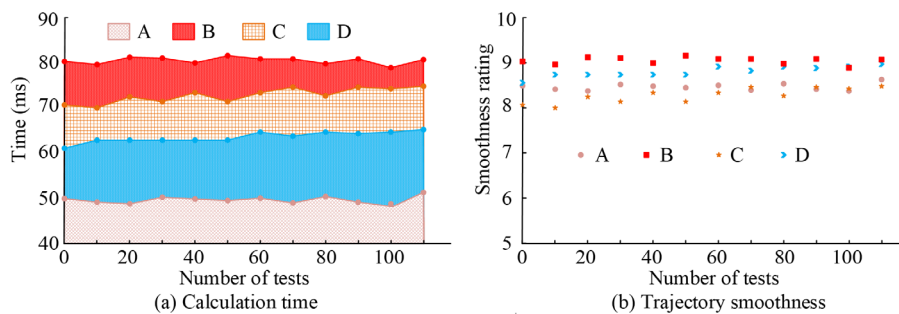


Figure 5. Analysis of calculation time and curve trajectory smoothness for the TBC-FSM Method

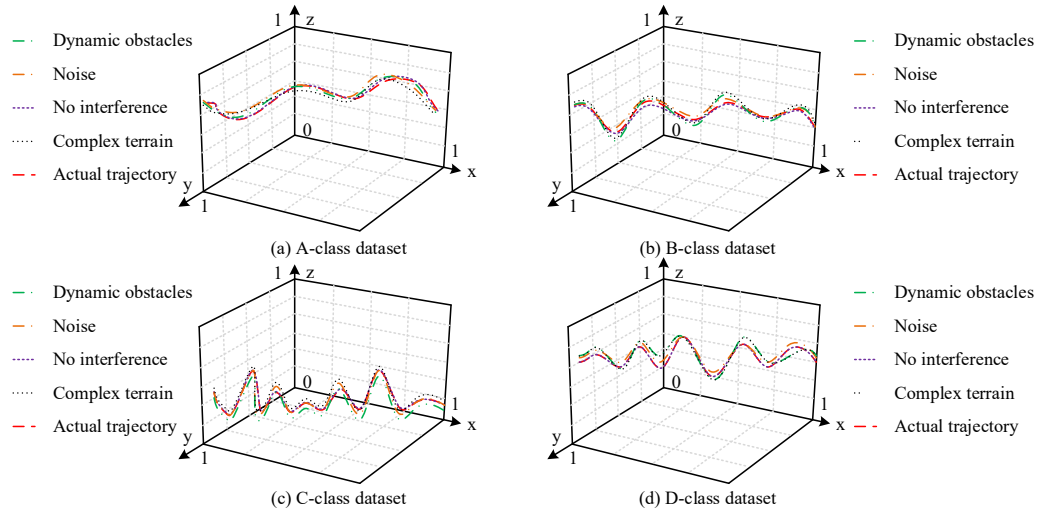


Figure 6. Accuracy of TBC-FSM curve generation

In Figure 6, under the condition of no interference, the motion trajectory planned by TBC-FSM almost coincides with the actual motion trajectory for the four types of datasets. Under dynamic obstacle interference, complex terrain, and noise interference conditions, the difference between the motion trajectory planned by the TBC-FSM method and the actual trajectory is also relatively low. Based on Figures 6(b-d), the accuracy of curve generation for TBC-FSM is higher than 95% for the datasets B, C, and D. Finally, the response time for TBC-FSM under the aforementioned interference is analyzed to verify the stability of the trajectory path. The response speed variation for TBC-FSM is shown in Figure 7.

As shown in Figure 7(a), the TBC-FSM method has an average response time of 6.32 ms under undisturbed conditions, while under noise interference, dynamic obstacle interference, and complex terrain conditions, its average response times are 6.21 ms, 6.34 ms, and 6.52 ms, respectively. Under different interference conditions, TBC-FSM features a strong response time stability and a low fluctuation amplitude

with regard to the response time. With regard to Figures 7(b-d), when TBC-FSM plans three types of motion trajectories for the B-, C-, and D-class dataset, the fluctuation amplitude for the response time under different interference conditions is less than 1 ms, indicating a strong stability. This indicates that the trajectory planning accuracy of the TBC-FSM method is high, the path planning time is short, and its stability is excellent.

4.2 Analysis of Path Planning Effectiveness

After verifying the performance of TBC-FSM curve generation, the actual effect of the vehicle OAP planning method based on TBC-FSM is analyzed. For the analysis, the transportation route for a certain city is selected as the experimental scenario, the starting and ending points of the area are determined, and multiple obstacles are set up. Vehicles with and without TBC-FSM models are simultaneously placed in the test scenario, keeping the vehicle model and other conditions the same as previously. The obstacle avoidance effects for the two types of vehicles are compared

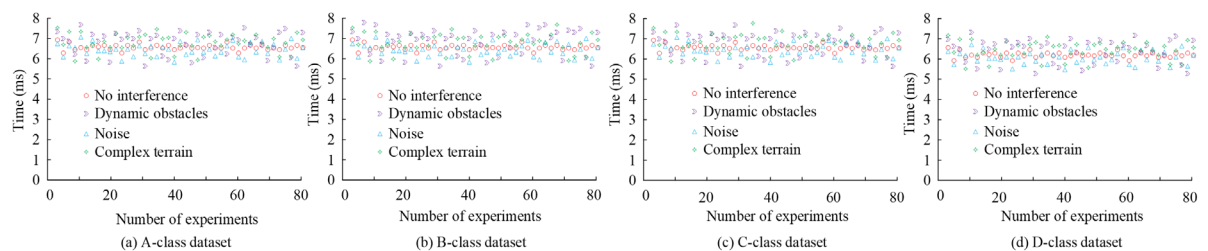


Figure 7. Response time of TBC-FSM method

through multiple experiments to determine the performance of the TBC-FSM method. Figure 8 analyzes the actual OASR of vehicles for two different scenarios.

Figure 8(a) shows the starting point, ending point, and obstacle distribution for the scenario selected for this experiment, along with the scenario involving obstacle avoidance and path planning. As shown in Figure 8(b), the vehicles with TBC-FSM can completely avoid obstacles in the experimental scene when planning OAP, while the OAP planned by vehicles without using the TBC-FSM method contains some obstacles, indicating that if the TBC-FSM method is not employed, there are certain obstacles which cannot be avoided. This result indicates that TBC-FSM can accurately plan obstacle avoidance for vehicles in practical situations, enabling them to avoid obstacles and improve the efficiency of path planning. Figure 9 compares the time consumption and computer memory resource consumption for path planning.

As shown in Figure 9(a), when TBC-FSM plans the vehicle OAP, the average path planning time is 45.6 ms, which is relatively short. The average planning time without utilizing the TBC-FSM method reaches 65.3 ms, which is relatively long. With regard to Figure 9(b), the average memory resource utilization rate when using the TBC-FSM method is 43.7%, while when not using TBC-FSM, the memory resource

utilization rate increases by 21.1%, reaching 64.8%. This indicates that the TBC-FSM method can significantly improve the OASR of vehicle OAP and reduce the planning time and computer memory usage. Finally, the performance of the TBC-FSM method is compared with that of the widely used Ant Colony Optimization - Dynamic Window Approach (ACO-DWA) (Song, Tang & Li, 2024), Adaptive Gaussian Sampling-based Rapidly-exploring Random Tree (AGS-RRT) (Liu et al., 2025), and K-Nearest Neighbor - RRT (KNN-RRT) algorithms (Min et al., 2025). The comparative OASR and path planning time results are shown in Table 2.

Table 2. Analysis of obstacle avoidance effect and path planning time

OAP method	Obstacle avoidance effect		Path planning time	
	OASR	p	Time	p
TBC-FSM	98.7%	**	45.6ms	**
ACO-DWA	93.2%	**	67.4ms	*
AGS-RRT	89.8%	*	79.3ms	**
KNN-RRT	85.2%	*	80.3ms	*

Note: * indicates $p < 0.05$, ** indicates $p < 0.01$

As shown in Table 2, the OASR of the TBC-FSM method can reach 98.7%, and its path planning time only amounts to 45.6 ms. When the ACO-DWA, AGS-RRT, and KNN-RRT methods are

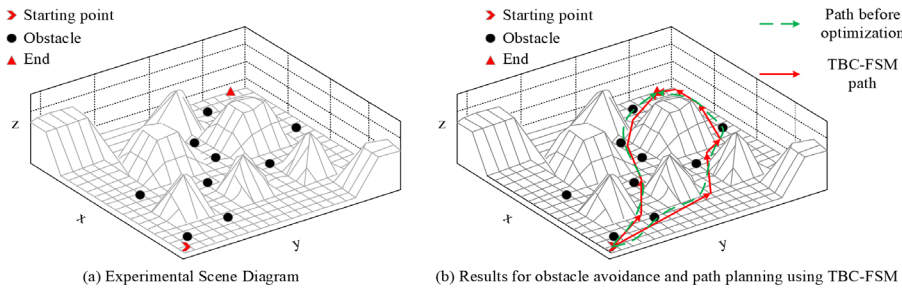


Figure 8. Analysis of obstacle avoidance success rate

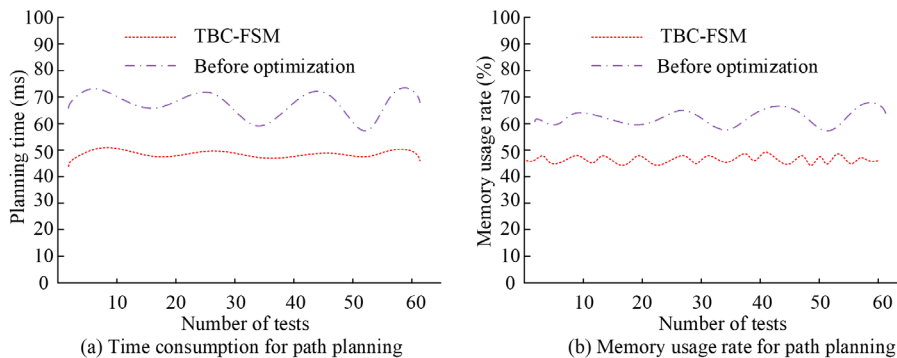


Figure 9. Path planning time and Computer Memory Utilization

employed for planning the OAP of vehicles, their OASR are 93.2%, 89.8%, and 85.2%, respectively, which are lower than the value obtained by the TBC-FSM method. Moreover, the path planning times for the ACO-DWA, AGS-RRT, and KNN-RRT methods are all higher than the value obtained by TBC-FSM. Also, the results of the statistical analysis for all four methods are statistically significant ($p < 0.05$). To sum up, the obtained results show that in comparison with other vehicle OAP planning methods, the TBC-FSM method has obtained the highest obstacle avoidance success rate and the shortest planning time for path planning.

5. Conclusion

Given the problems of poor path planning performance and long planning time specific to the current vehicle OAP planning methods, this study combined TBC and FSM to obtain the mixed TBC-FSM method, aiming to improve the path planning efficiency and reduce the path planning

time. This paper analyzed the effectiveness of TBC-FSM and found that the trajectory generation accuracy was higher than 95%, and the response time of this method for different datasets was lower than 8 ms, which is a relatively short response time. Analyzing the actual effectiveness of the TBC-FSM method, it was found that for the path planned by this method an OASR of 98.7% and a path planning time of only 45.6 ms were obtained. The value of OASR was higher than that obtained by ACO-DWA (93.2%), AGS-RRT (89.8%), and KNN-RRT (85.2%). This paper proved that the proposed TBC-FSM method can accurately plan vehicle OAP, but TBC relies heavily on control points, and the improper selection of control points can reduce the effectiveness of path planning. In this sense, future work can improve the effectiveness of the proposed algorithm by converting constraints such as the maximum curvature and acceleration of the vehicle into control point selection conditions so as to select the optimal control points.

REFERENCES

- Bungey, G., Moller, J., Saunders, J. et al. (2026) Deriving an analytical solution to inversion of royston/parmar restricted cubic spline parametric survival models for discrete event simulation. *PharmacoEconomics*. 44(4), 451-459. <https://doi.org/10.1007/s40273-025-01569-x>.
- Ghannou, O., Thuillier, E. & Boucelma, O. (2024) Maritime Trajectory Mining: An Automatic Zones of Interests Discovery and Annotation Framework. *Computer Science and Information Systems*. 21(4), 1963-1978. <https://doi.org/10.2298/CSIS240301063G>.
- Han, Y. & Song, T. (2024) Path Planning for Shop-Floor Material Transfer Robots Incorporating Particle Swarm and Ant Colony Algorithms. *Studies in Informatics and Control*. 33(3), 49-59. <https://doi.org/10.24846/v33i3y202405>.
- Khan, S. S. & Wen, H. (2023) A fast and low-cost open-circuit fault detection and isolation technique for three-phase dual-active-bridge converters based on finite state machines. *IEEE Transactions on Power Electronics*. 39(2), 2751-2766. <https://doi.org/10.1109/TPEL.2023.3337097>.
- Kibria, R., Farahmandi, F. & Tehranipoor, M. (2023) FSMx-Ultra: Finite state machine extraction from Gate-Level NetList for security assessment. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*. 42(11), 3613-3127. <https://doi.org/10.1109/TCAD.2023.3266368>.
- Kim, C., Yoon, Y., Kim, S. et al. (2023) Trajectory planning and control of autonomous vehicles for static vehicle avoidance in dynamic traffic environments. *IEEE Access*. 11(6), 5772-5788. <https://doi.org/10.1109/access.2023.3236816>.
- Li, H., Han, S., Wu, X. et al. (2023) A Novel Task of Loading and Computing Resource Scheduling Strategy in Internet of Vehicles Based on Dynamic Greedy Algorithm. *Tehnicki vjesnik [Technical Gazette]*. 30(4), 1298-1307. <https://doi.org/10.17559/TV-20221207032927>.
- Li, J., Li, J., Fang, H. et al. (2024) Dynamic Energy-Efficient Path Planning for Electric Vehicles Using an Enhanced Ant Colony Algorithm. *Tehnicki vjesnik [Technical Gazette]*. 31(2), 434-441. <https://doi.org/10.17559/TV-20230510000618>.
- Li, Z., Li, J. & Li, Y. (2023) An integrated path planning framework for multi-obstacle avoidance of the multi-axle autonomous vehicle with enhanced safety and stability. *IEEE Transactions on Vehicular Technology*. 73(5), 6368-6382. <https://doi.org/10.1109/TVT.2023.3346450>.
- Liu, Z., Cui, J., Meng, F. et al. (2025) Research on intelligent ship route planning based on the adaptive step size informed-RRT* algorithm. *Journal of Marine Science and Application*. 24(4), 829-839. <https://doi.org/10.1007/s11804-024-00433-2>.

- Luo, Q., Wang, H., Li, N. et al. (2024) Multi-unmanned surface vehicle model-free sliding mode predictive adaptive formation control and obstacle avoidance in complex marine environment via model-free extended state observer. *Ocean Engineering*. 293(1), Art. No. 116773. <https://doi.org/10.1016/j.oceaneng.2024.118035>.
- Mao, M, Wu, D & You, Z. (2024) Dynamic event-triggered path-following control for underactuated unmanned surface vehicle with obstacle avoidance. *Ocean Engineering*. 312(2), Art. No. 119161. <https://doi.org/10.1016/j.oceaneng.2024.119161>.
- Min, L., Chen, J., Fan, X. et al. (2025) Altruistic nurturing algorithm for multi-objective autonomous underwater vehicles path planning problems. *China Communications*. 22(5), 350-371. <https://doi.org/10.23919/JCC.fa.2025-0010.202505>.
- Okereke, C. E., Mohamad, M. M., Wahab, N. H. et al. (2023) An overview of machine learning techniques in local path planning for autonomous underwater vehicles. *IEEE Access*. 11(3), 24894-24907. <https://doi.org/10.1109/ACCESS.2023.3249966>.
- Pang, Y., Bai, Z., Xiang, Z. et al. (2023) A data-driven finite-state machine-based control for hybrid parallel multiconverters: Fusion topology for high-power applications. *IEEE Transactions on Industrial Electronics*. 70(12), 11853-11864. <https://doi.org/10.1109/TIE.2023.323412>
- Shang, T., Li, X., Suo, Y. et al. (2024) Research on the Support Parameters Optimization of Large Section Roadway in Shallow Coal Seam. *Tehnicki vjesnik [Technical Gazette]*. 31(6), 2028-2035. <https://doi.org/10.17559/TV-20240312001394>.
- Simo, A, Dzitac S, Ferestyan, L et al. (2024) Optimizing Electric Vehicle Performance: Advances in Battery Management Systems for Enhanced Efficiency and Longevity. *International Journal of Computers Communications & Control*. 19(5), 6794-6808. <https://doi.org/10.15837/ijccc.2024.5.6794>.
- Song, B., Tang, S. & Li, Y. (2024) A new path planning strategy integrating improved ACO and DWA algorithms for mobile robots in dynamic environments. *Mathematical Biosciences and Engineering*. 21(2), 2189-2211. <https://doi.org/10.3934/mbe.2024096>.
- Wang, R, Jiang, T., Bai, G. et al. (2023) Stepwise cooperative trajectory planning for multiple BUVs based on temporal-spatial Bezier curves. *IEEE Transactions on Instrumentation and Measurement*. 72(12), Art. No. 8503414. <https://doi.org/10.1109/TIM.2023.3275617>.
- Wang, S, Li Z, Wang, B, et al. (2023) Collision avoidance motion planning for connected and automated vehicle platoon merging and splitting with a hybrid automaton architecture. *IEEE Transactions on Intelligent Transportation Systems*. 25(2), 1445-1464. <https://doi.org/10.1109/TITS.2023.3315063>.
- Xu, H., Xie, H., Chen, Y. et al. (2023) A new cyborg rat auto navigation system based on finite state machine. *IEEE Sensors Journal*. 23(19), 23456-23466. <https://doi.org/10.1109/JSEN.2023.3291870>.
- Yang, M.-C., Chen, W.-N., Wei, F.-F. et al. (2025) Gene expression programming-based ride insert policy for online electric vehicle ride-hailing optimization. *IEEE Transactions on Intelligent Transportation Systems*. 26(12), 22136-22150. <https://doi.org/10.1109/TITS.2025.3618704>.
- Yang, S., Gao, X. & Ren, J. (2023) Sequential circuits synthesis for rapid single flux quantum logic based on finite state machine decomposition. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*. 42(10), 3315-3326. <https://doi.org/10.1109/TCAD.2023.3245542>.
- Yang, X., Xiang, K., Yuan, S. et al. (2024) Vehicle Driving Behavior Recognition and Optimization Strategies Based on Cloud Computing and SSA-BP Algorithm. *Studies in Informatics and Control*. 33(3), 17-28. <https://doi.org/10.24846/v33i3y202402>.
- Zhai, L., Liu, C., Zhang, X. et al. (2024) Local trajectory planning for obstacle avoidance of unmanned tracked vehicles based on artificial potential field method. *IEEE Access*. 12(4), 19665-19681. <https://doi.org/10.1109/ACCESS.2024.3355952>.
- Zhang, X., Wan, G. & Zhang, H. (2025) Transfer learning for cross-city traffic prediction to solve data scarcity. *Transportation Research Record*. 2679(3), 697-706. <https://doi.org/10.1177/03611981241283013>.
- Zhou, J., Olofsson, B. & Frisk, E. (2023) Interaction-aware motion planning for autonomous vehicles with multi-modal obstacle uncertainty predictions. *IEEE Transactions on Intelligent Vehicles*. 9(1), 1305-1319. <https://doi.org/10.1109/TIV.2023.3314709>.
- Zhu, G., Liu, Y. & Wang, J. (2024) Multi-frame Network Feature Fusion Model and Self-attention Mechanism for Vehicle Lane Line Detection. *Computer Science and Information Systems*. 21(4), 1699-1723. <https://doi.org/10.2298/CSIS240314054Z>.



This is an open access article distributed under the terms and conditions of the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.