IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 2, 2020, accepted August 16, 2020, date of publication August 19, 2020, date of current version September 2, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3017770

Reinforcement Learning-Based Motion Planning for Automatic Parking System

JIREN ZHANG, (Member, IEEE), HUI CHEN, (Member, IEEE), SHAOYU SONG, AND FENGWEI HU
School of Automotive Studies, Tongji University, Shanghai 201804, China
Corresponding author: Hui Chen (hui-chen@tongji.edu.cn)

ABSTRACT In automatic parking motion planning, multi-objective optimization including safety, comfort, parking efficiency, and final parking performance should be considered. Most of the current research relies on the parking data from expert drivers or prior knowledge of humans. However, it is challenging to obtain a large amount of high-quality expert drivers' data. Furthermore, expert drivers' data or prior knowledge of humans does not guarantee an optimal multi-objective parking performance. In this article, we propose a model-based reinforcement learning method that learns parking policy of the data, by executing the data generation, data evaluation, and training network, iteratively. The trained network is used to guide the data generation cycle in the subsequent iteration. Based on this proposed method, we can get rid of human experience largely and learn parking strategies autonomously and quickly. The learned strategies ensure the multi-objective optimality of above requirements in the parking process. First, an environment model that approximates the actual environment is established, and the learning efficiency is accelerated through the simulated interaction between the agent and the environment model. To make the system independent of expert data or prior knowledge, a data generation algorithm combining Monte Carlo Tree Search (MCTS) and longitudinal and lateral policies is proposed. Then, to meet the multi-objective optimal demands mentioned above, a reward function is constructed to evaluate and filter the parking data. Finally, a neural network is used to learn the parking strategy from the filtered data. From the real vehicle test benchmarked with a mass-produced parking system, the proposed method is found to achieve better parking efficiency and lower requirements for start parking posture, thereby verifying the algorithm's superiority.

INDEX TERMS Automatic parking, motion planning, reinforcement learning, Monte Carlo tree search, neural network.

I. INTRODUCTION

The growth of mobile travel demand and the shortage of road capacity have caused severe traffic and environmental problems due to the continuous increase of car ownership. The application of connected automated vehicles can reduce traffic congestion, gasoline consumption, and transportation emissions significantly [1], [2]. Moreover, the demand for the time-sharing electric vehicle is increasing, as it is eco-friendly, intensive, and efficient. However, drivers still face collision hazards and inefficient road traffic due to tight space and unskilled operations when parking cars. At the same time, the need for wireless charging after shared electric vehicle parking increases the requirements concerning the vehicle's final parking posture. Moreover, comfort is also required during parking, and excessive acceleration and deceleration of the vehicle and high-frequency jitter of a steering wheel should be avoided. Therefore, comprehensive consideration of safety, comfort, parking efficiency, and final parking posture to achieve parking motion planning is of great significance for the future development of shared vehicles.

Generally, a typical automatic parking system, as shown in Fig. 1, includes key technologies such as parking slot detection, motion planning (or path planning and tracking), ego-vehicle's posture estimation, and chassis control. Among them, motion planning is an intermediate module for environment perception and chassis control, which plans vehicle control commands based on real-time vehicle information and parking space information. The motion planning module transmits the information to the vehicle chassis control

The associate editor coordinating the review of this manuscript and approving it for publication was Mostafa Rahimi Azghadi.

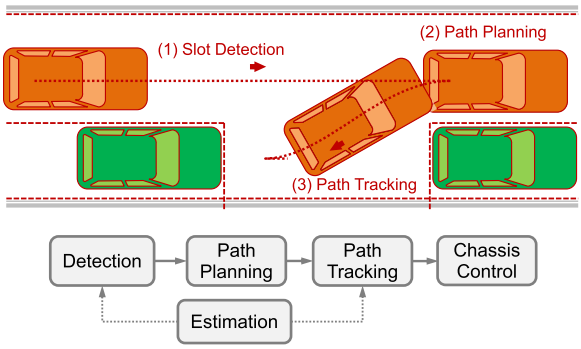


FIGURE 1. A typical automatic parking system includes the following modules: slot detection, path planning, path tracking, ego-vehicle's posture estimation, and chassis control.

module for execution. Current research on parking motion planning can be divided into several methods, such as expert drivers' parking data-based method, human prior knowledge-based method, and reinforcement learning-based method.

A. RELATED WORK

1) EXPERT DRIVERS' DATA-BASED METHOD

Typically, the expert drivers' data-based methods use supervised learning systems, such as neural networks, that are trained to replicate the expert drivers' parking actions. The inputs of the network are environmental perception information, such as visual images, distance, and vehicle status, and the output is the corresponding driver action [3], [4]. Due to the limited generalization ability of the neural network, a large amount of training data is required to cover target working scenarios, as much as possible. Besides, the network performance strongly depends on the quality of parking data. Thus, the data must be of high-quality. However, expert data sets are often expensive and require a lot of labor and time. Moreover, the expert data will impose a ceiling on the performance of systems trained in this manner [5]. Thus, the system can only approach but cannot exceed the expert performance, making it difficult to achieve optimal multi-objective parking performance.

2) PRIOR KNOWLEDGE-BASED METHOD

Prior knowledge-based method refers to abstracting human parking experience into prior knowledge and then using it to guide the planning. It is divided into geometric, heuristic search, and fuzzy logic control methods. The styles of the curves, such as the Reeds-Shepp (RS) curve [6], [7], clothoid curve [8], [9], Bezier curve [10], spline curve [11], and polynomial curve [12] in the geometric method, the heuristic function in search methods such as A* [13], and fuzzy rules in fuzzy logical methods [14], [15] are all prior knowledge abstracted from the human parking experience. The parking experience itself has an obvious bias. Thus, it is more challenging to achieve the optimal multi-objective parking performance due to information loss by abstraction.

3) REINFORCEMENT LEARNING-BASED METHOD

As mentioned above, methods based on expert data and prior knowledge mostly rely on the original or abstraction of human experience, which requires considerable high-quality parking data. Even when expert data sets are available, they will impose a ceiling on the performance of the system. By contrast, reinforcement learning systems are trained from their own experience, in principle allowing them to exceed human capabilities [5].

The reinforcement learning system learns strategies through interaction between the agent and the environment. According to a standard that considers whether it is necessary to model the environment, reinforcement learning is mainly divided into two categories: model-based methods and model-free methods. In the model-based method, a state transition probability matrix and reward function is obtained first, and then a strategy to maximize the cumulative reward is found. Whereas, the model-free method directly estimates the value $Q(s, a)$ of the action a taken in the state s, and then selects the action with the highest estimated return value to be executed in each state.

Regarding the parking planning problems, few studies have been conducted using the reinforcement learning method. Zhang *et al.* [16] used deep deterministic policy gradient (DDPG) [17], a model-free reinforcement learning method, to solve the perpendicular parking problem. They train an agent in the simulation environment first and then transfer the trained agent to the real vehicle to continue the training. The work only focuses on the final parking section of two steps in perpendicular parking, and the speed policy is also simplified to a fixed command. For other scenarios of autonomous driving, such as lane change decisions or lane-keeping assistance (LKA), some scholars used the model-free reinforcement learning methods to study, such as Q-learning [18], deep Q-Network (DQN) [19], [20], Actor-Critic [21]–[23], DDPG [24], [25], and so on.

The model-free method does not require modeling of the environment. However, it can only be learned through the actual interaction between agent and environment that is of low learning efficiency. Kaiser et al. [26] compared the learning efficiency of their proposed model-based method with state-of-the-art model-free methods: Rainbow [27] and proximal policy optimization (PPO) [28], to play video games. Results show that the proposed model-based method outperforms the model-free algorithms in terms of learning speed on nearly all of the games, and in the case of a few games, does so by over an order of magnitude.

At the same time, whether the above research is for the last segment of perpendicular parking or high-speed scenario decisions, such as the lane changes and LKA, the steering wheel angle change range is small. This work presents parallel parking, which is generally considered to be more difficult than perpendicular parking. The steering wheel herein has a wide range of input angles, so the search space is vast, and it is facing an urgent need for rapid learning.

In summary, the model-based method enables an agent to understand how the problem works and predict which actions will produce the desired results without requiring real vehicles to interact with the environment, while improving learning efficiency and safety. MCTS is a representative model-based reinforcement learning method. In the field of games, AlphaGo [29] and AlphaGo Zero [5] surpassed humans in Go using the MCTS combined with neural networks. The success of AlphaGo and AlphaGo Zero demonstrates the effectiveness of the MCTS in large search space problems. Therefore, we believe that the MCTS is applicable to the parking planning problem.

However, there are many differences between Go and automatic parking. In Go, we can directly predict the situation and the final result of the game according to the rules. For parking, however, an environment model needs to be established to estimate the vehicle state, and a reward function needs to be constructed to evaluate the parking performance. Moreover, Go is a two-player game, and parking planning can be regarded as a single-agent task. Finally, the search space for the parking actions is larger than that of Go, and real-time requirements in parking are higher. This article focuses on the new features and studies reinforcement learning to solve the parking plan problem.

B. OBJECTIVES AND CONTRIBUTIONS

The objective of this article is to learn parking strategies autonomously without relying on human experience or prior knowledge and to plan the result meetings the multi-objective optimization of safety, comfort, parking efficiency and final parking posture. The main contributions are summarized as follows:

1) A reinforcement learning method of parking strategy is proposed. The method iteratively executes data generation, data evaluation, and training the network using the selected data. The network is used to guide the next iteration cycle of generating data. In this way, the quality of the generated data is continuously improved and the learned parking strategy is continuously enhanced. Finally, it converges to an optimal state.

2) To construct a vehicle model that approximates the real vehicle for simulation, a method based on the transfer function combined with a kinematic vehicle model is proposed.

3) To generate parking data for training agents, a method based on the longitudinal policy and lateral policy algorithm is proposed. The P-MCTS plays an essential role in the lateral policy, which is a variant of MCTS for parking.

4) Since the multi-objective optimization, including safety, comfort, parking efficiency, and final parking performance should be considered, a reward function to evaluate the performance of parking data is researched and constructed.

C. PAPER OUTLINE

The rest of this article is organized as follows. In Section 2, a reinforcement learning method for parking strategies is proposed. In Section 3, an environment model for reinforcement learning, including the slot model and vehicle model is introduced. In Section 4, we propose a parking data generation method based on the established environment model. A reward function to evaluate the quality of parking data is built in Section 5 and the network is trained using filtered parking data by the reward function. Section 6 demonstrates and discusses the experimental results. Finally, Section 7 concludes the paper.

II. ALGORITHM FRAMEWORK

The algorithm framework proposed in this article is shown in Fig. 2. It is divided into three parts: data generation, data evaluation, and update of the network with the best data selected. Usually, the parallel parking can be split into two steps: entering a parking slot first and then aligning the vehicle with parked vehicles. The result of the first step directly affects the number of shifts in the parking slot and parking time. Also, it even determines whether the automatic parking will be successful, which is more complicated and

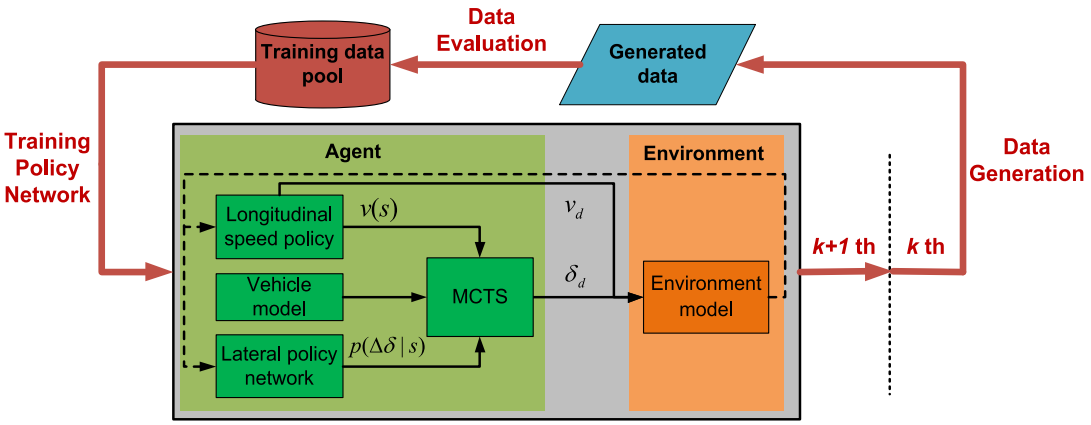


FIGURE 2. The proposed algorithm framework comprising three parts: data generation, data evaluation, and training policy network with the best data selected.

essential. Therefore, this work applies the proposed method to the entering parking slot stage.

A. DATA GENERATION

In this section, the agent in Fig. 2, i.e., the motion planning algorithm, controls the vehicle model and performs simulations under different working scenarios to generate a large amount of parking data.

The motion planning algorithm is composed of longitudinal vehicle speed policy, vehicle model, lateral policy network, and MCTS. The inputs of the algorithm are parking space size and vehicle's real-time state, and the outputs are commands of the steering wheel angle and vehicle speed. The expected speed v_d is directly obtained by the speed policy, which helps to avoid the problem of slow convergence caused by the lateral and longitudinal strategies entirely obtained by the reinforcement learning. The expected steering angle δ_d is obtained from a large number of simulations based on the policy network, speed policy, vehicle model combined with MCTS. The probability of different steering wheel angle changes $p(\Delta\delta|s)$ output by the policy network and the vehicle speed policy $v(s)$ are used as the default lateral and longitudinal policies of the MCTS in the simulation phase. In this way, it can help concentrate computing resources on those branches with higher probability, thus improving algorithm performance. The vehicle model estimates the state of the vehicle after performing steering angle and speed actions. The MCTS generates optimal control commands through numerous simulations. The vehicle model executes the control commands and then generates the parking data.

B. DATA EVALUATION

A reward function is constructed comprehensively considering factors such as safety, comfort, parking efficiency, and final parking posture. It is used to evaluate the quality of parking data. Finally, the data with the best parking quality of each parking scenario are selected.

C. NETWORK TRAINING

The network parameters are updated using the selected data with the best quality. The inputs of the network are vehicle status and slot information, and the vehicle status includes the position and attitude relative to the slot and real-time steering wheel angle. The output of the network is the probability distribution of different steering wheel angles. Note that there is no data at the beginning of the learning process. To prevent the algorithm from being introduced to human experience, a random strategy is used as the default policy of MCTS in the simulation phase to generate initial data for training the network.

This updated network is used in the process of generating data in the next iteration, as a new default policy in the MCTS simulation phase to provide a stronger search guide. In this way, the quality of the generated parking data is improved continuously, and the learned parking strategy is continuously enhanced. Eventually, it converges to an optimal state. The reinforcement learning agent with final learning convergence in Fig. 2 is deployed on the real vehicle as the parking motion planning controller. The above three stages will be described in detail in the subsequent section.

III. ENVIRONMENT MODEL

The model-based reinforcement learning method uses an environment model to predict the possible future states of different actions and the expected reward value from these states, so as to select the optimal action. For automatic parking, the parking planning algorithm can be regarded as the agent, and the ego-vehicle and the parking slot composed of front and rear obstacles can be regarded as the environment. Therefore, this section introduces how to model the environment.

A. PARKING SLOT MODEL

In this article, parallel parking on the right side is taken as the research scenario, which is formed by two parked cars in front and behind. As shown in Fig. 3, the front and rear obstacles are abstracted into four rays and O and R at both ends. The two rays describing the left and rear sides of the front parked vehicle have a common end point O and the two rays describing the left and front sides of the rear parked vehicle have a common end point R. The left rear corner of the front parked car is the coordinate origin. The direction along the left side of the vehicle body toward the front of the vehicle is the positive X-axis X_P, and the parking coordinate system of motion planning is established. The coordinate of the intersection point R is (x_R, y_R). The angles between these two rays and the coordinate axis are $\theta_1 = -180°$ and $\theta_2 = -90°$. The angle between the rays on the rear side of the front parked vehicle and the coordinate axis is $\theta_3 = -90°$. Taking start parking posture as an example, the ego-vehicle is represented by the midpoint S of its rear axle. The coordinate is (x_s, y_s) and the heading angle is indicated by θ_s. The motion planning is performed in the above parking scenario and parking coordinate system.

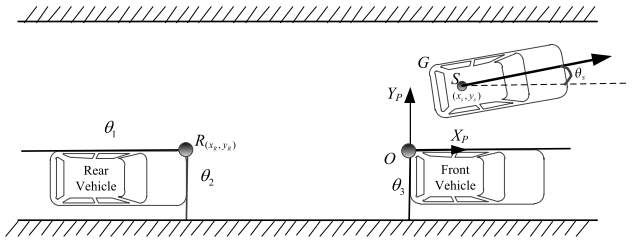


FIGURE 3. Parallel parking coordinate system.

B. VEHICLE MODEL

1) VEHICLE STATE PREDICTION BASED ON KINEMATIC VEHICLE MODEL

As the vehicle's velocity is relatively low during parking, the kinematic vehicle model shown in Fig. 4 is used in motion

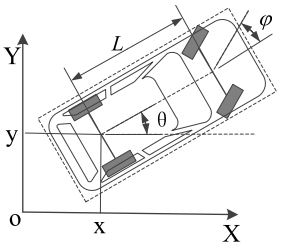


FIGURE 4. Kinematic vehicle model.

planning, as shown in (1),

$$\begin{aligned} \dot{x} &= v\cos(\theta) \\ \dot{y} &= v\sin(\theta) \\ \dot{\theta} &= v\tan(\phi)/L \end{aligned} \tag{1}$$

where (x, y, θ) represents the posture of the vehicle in parking coordinate, ϕ denotes steering angle of the front wheel, v is the velocity at the center of the rear axle. Based on the vehicle kinematic model, the actual steering wheel of the front wheel and actual vehicle speed can be used to predict the vehicle's posture.

2) VEHICLE LATERAL AND LONGITUDINAL CHASSIS CONTROL MODEL BASED ON SYSTEM IDENTIFICATION

The vehicle chassis control process is shown in Fig. 5. Due to the design error of the chassis control algorithm, there is a certain error between the actual steering wheel angle and the expected order. Therefore, it is necessary to model between the expected order and the observation of the action performed using the vehicle chassis control. It ensures that under the same desired steering wheel angle, the steering wheel angle output of the model is close to the actual steering wheel angle. Thus, the vehicle posture estimated by the kinematic vehicle model (Fig. 5b) and the actual posture (Fig. 5a) are close to each other.

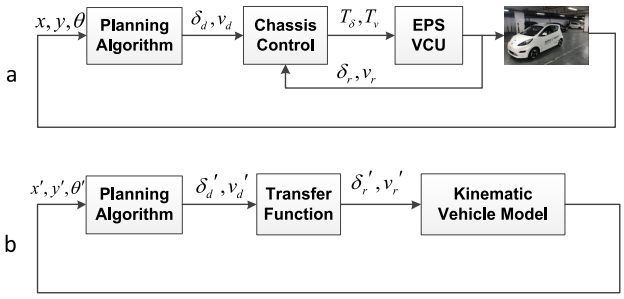


FIGURE 5. Vehicle chassis control process: (a) Real vehicle (b) Vehicle model.

In this article, the transfer function is used to approximate the relationship between the action command and the actual observed action. First, the simulated parking command is used to control the vehicle. The steering angle and vehicle speed order values and corresponding actual observation values are collected. Then, the data are used to identify the lateral and longitudinal transfer functions as follows:

$$\begin{aligned} \delta_r &= G_\delta(s) \cdot \delta_d \\ v_r &= G_v(s) \cdot v_d \end{aligned} \tag{2}$$

where δ_d and v_d are the command values of the steering wheel angle and vehicle speed, respectively, δ_r and v_r are the actual values of the steering wheel and vehicle speed, respectively. The $G_\delta(s)$ and $G_v(s)$ are the identified lateral and longitudinal transfer functions, respectively.

Next, we use the identified transfer functions combined with the kinematic vehicle model as the constructed vehicle model. We also obtain the final parking strategy using the proposed reinforcement learning method based on the constructed vehicle model. Finally, simulations and real vehicle experiments are performed to verify the accuracy of the constructed vehicle model. Further, we select three start postures from each of the five parking slots in the training data, i.e., a total of 15 working scenarios are verified. The coordinates of the three start postures are uniformly distributed in the range of training data. The mean absolute error of the final parking posture between the generated data and the real vehicle data is calculated. For all verification scenarios, the deviations of the abscissa and ordinate are 7 and 8 cm, respectively, and the deviation of the heading angle is 3°. Furthermore, for the standard test slot (i.e., the parking slot of 4.57 m), the deviations of the abscissa and ordinate are 3 and 1 cm, respectively, and the deviation of the heading angle is 1.25°. The above results demonstrate that the generated data is close to actual vehicle data.

The lateral and longitudinal transfer functions are determined as follows:

$$G_\delta(s) = \frac{8.57s^3 + 26.90s^2 + 78.34s + 248.60}{s^4 + 8.33s^3 + 36.60s^2 + 74.92s + 248.2} \tag{3}$$

$$G_v(s) = \frac{25.75s + 47.85}{s^4 + 4.03s^3 + 27.09s^2 + 46.48s + 52.80} \tag{4}$$

In summary, the vehicle kinematic model and the lateral and longitudinal transfer functions are selected to realize the construction of the vehicle model.

IV. PARKING DATA GENERATION

Based on the parking data generation architecture introduced in Section 2, the longitudinal and lateral motion planning in the automatic parking motion planning controller (the agent in Fig. 2) is introduced in this section.

A. LONGITUDINAL MOTION PLANNING

The requirements of safety, comfort, and parking time are taken into account in the vehicle speed policy. It is divided into acceleration, stable, and deceleration phases. During the acceleration phase, the acceleration simulates the actual parking process, which increases first and then decreases, as shown in Fig. 6. The vehicle speed is negative in the entering parking slot stage. To facilitate understanding, the acceleration in the following refers to its absolute value, which

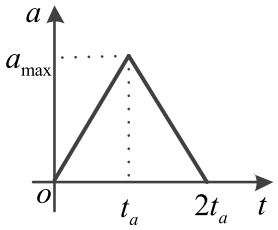


FIGURE 6. Acceleration profile of the parking acceleration section.

is actually a negative value. The acceleration of the vehicle changes continuously during the acceleration phase. The maximum acceleration $a_{\max}$ is 0.3 m/s^2, the acceleration time $2t_a$ is 5 s, and the initial order is $v_0 = -0.2$m/s.

The speed of the parking acceleration section is as follows: when $0 \le t < t_a$,

$$v_{acc}(t) = v_0 - 0.5a_{\max}t^2 \Big/ t_a, \tag{5}$$

and when $t_a \le t < 2t_a$,

$$v_{acc}(t) = v_0 - 0.5a_{\max}t_a - 0.5\left(a_{\max} + \left(a_{\max} - a_{\max}\left(t - t_a\right)\Big/ t_a\right)\right) \times \left(t - t_a\right) \tag{6}$$

The final speed of the acceleration section is the speed of the stable section.

$$v_{stable}(t) = v_0 - a_{\max}t_a \tag{7}$$

The speed of the deceleration section is controlled by defining the shortest distance between the left rear corner G of the ego-vehicle and the rear parked vehicle, as shown in Fig. 3. The vehicle speed observation is inaccurate at low speed. For the vehicle used in the experiment, when the absolute value of vehicle speed is lower than the threshold $v_{threshold}$, the observed value is 0. To control the real vehicle more accurately, we conducted the following experiment. It demonstrates that the driving distance of the vehicle on a horizontally structured road decelerating naturally from the $v_{threshold}$ to 0 is approximately 0.05 m. Accordingly, the speed policy is designed as follows:

$$v_{dec}(t) = \begin{cases} kd + b & 0.25 \le d \le 1.50 \\ 0 & d < 0.25 \end{cases} \tag{8}$$

where d indicates the distance between the left rear corner of the vehicle and the rear parked vehicle. According to experience, when the vehicle is at a distance of 1.5 m, it decelerates from the stable speed. After decelerating to a distance of 0.25 m, the vehicle speed decelerates to the $v_{threshold}$. Then the expected vehicle speed is zero, and the vehicle naturally decelerates for 0.05 m and finally stops at a distance of 0.20 m before the rear parked vehicle. The 0.20 m is the safety distance D_{safe} for vehicle posture alignment in the parking slot based on the characteristics of the ultrasonic sensor equipped in the vehicle. The values k and b can be determined by the above process.

B. LATERAL MOTION PLANNING

A method of combining P-MCTS (see Fig. 7) with neural network and vehicle speed policy is proposed in this article. The P-MCTS is a variant of MCTS for a single-agent decision-making process, such as parking. It further takes into account the prior probability of action during the selection process. Besides, it tracks maximum simulation results at each node, in addition to average results.

The neural network and vehicle speed policy are used as the default lateral policy π_δ and longitudinal policy π_v, respectively, in the simulation phase. So, the performance of the strategy can be significantly improved, and computing resources can be concentrated on a beam of high-probability actions.

Each node of the search tree in the P-MCTS contains edges for all possible actions (steering wheel angle) $a \in A(s)$. A set of statistics

$$\{N(s, a), W(s, a), Q(s, a), P(s, a)\} \tag{9}$$

is stored in each edge, where $N(s, a)$ is the visit count, $W(s, a)$ represents the total action value, $Q(s, a)$ is the mean action value, and $P(s, a)$ depicts the prior probability. The P-MCTS continues executing X iterations, starting at the root node each time, until the search is completed. Each iteration includes the following five steps: selection, expansion, simulation, backup, and search chain storage and final action selection. The steering wheel angle is selected depending on the accumulated statistics in the tree. The vehicle executes the action order to reach a new state and iterates as a new root node. This is repeated until the vehicle reaches the target parking area.

1) SELECTION

The selection step proceeds in the following way. A selection strategy is applied recursively, from the root node, until the most urgent expandable node s_E. An expandable node suggests that it is a non-terminating node and has unvisited child nodes. The selection strategy strikes a balance between exploitation and exploration. On the one hand, the task tends to select the action that achieves the highest reward so far (exploitation). On the other hand, the less promising actions still need to be tried, due to the limited search space so far (exploration) [30]. Here, an action is selected based on the statistics in the search tree

$$a_t = \text{argmax}_a(Q(s_t, a) + u(s_t, a)) \tag{10}$$

so as to maximize action value plus a bonus as follows:

$$u(s, a) = c_{P-MCTS}P(s, a)\frac{\sum_b N(s, b)}{1 + N(s, a)} \tag{11}$$

where c_{P-MCTS} is a parameter determining the level of exploration. The search causes selection strategy initially prefers actions with low visit count and high prior probability, but asymptotically prefers actions with high action value [27].

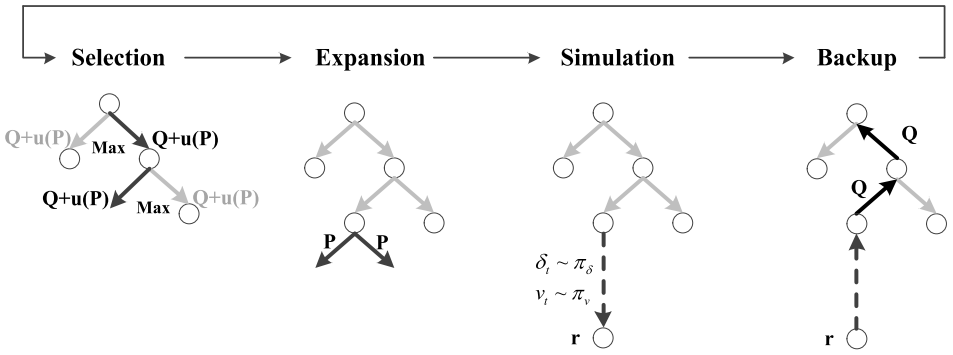


FIGURE 7. One iteration of P-MCTS.

2) EXPANSION

The child leaf node s_L of the node s_E is expanded, and each edge (s_L, a) is initialized to $\{N(s_L, a) = 0, W(s_L, a) = 0, Q(s_L, a) = 0, P(s_L, a) = p_a\}$. p_a is determined by the neural network.

3) SIMULATION

Each time the simulation starts from the leaf node s_L. The steering wheel angle is sampled according to the neural network $a_t \sim p(\cdot|s_t)$. Therefore, the greater the probability of the candidate action, the easier it is to be selected. Meanwhile, it also ensures that the actions with small probability are randomized to play a role of exploration. The vehicle speed is determined based on the vehicle speed policy designed in the previous section. The new state of the vehicle is calculated according to the vehicle model, after performing the sampling actions. The simulation continues until the vehicle meets the stop condition: the vehicle speed command is 0, or the vehicle ordinate is smaller than the target ordinate. The reward obtained is calculated using the reward function $z_t = r(s_T)$. The design of the reward function is introduced in Section 5.

4) BACKUP

After the end of each simulation, the visit counts and action values of all traversed edges are updated. The visit counts increases as follows:

$$N(s_t, a_t) = N(s_t, a_t) + 1 \tag{12}$$

and the action value is updated to the mean value as follows:

$$W(s_t, a_t) = W(s_t, a_t) + z_t \tag{13}$$

$$Q(s_t, a_t) = \frac{W(s_t, a_t)}{N(s_t, a_t)} \tag{14}$$

5) SEARCH CHAIN STORAGE AND FINAL ACTION SELECTION

The traditional MCTS is mainly used in the field of two-player games, where both parties jointly decide the game result. When making node selection, the algorithm needs to simulate the strategies of both parties of the game. Due to random sampling, a strong line of play would probably rely on an unrealistic assumption that the opponent plays weak moves. Therefore, the algorithm needs a lot of sampling to choose a reliable action, so that it is not sensitive to these outliers.

In the motion planning process, the parking controller is the only agent that can change the vehicle state. The averaging simulation results can hide a strong line of play if its siblings are weak, instead of favoring regions where all lines of play are of medium strength [31]. To counter this, the P-MCTS adds a memory mechanism in the action selection and simulation phases. The algorithm tracks the action sequence with a maximum reward after each iteration is completed. Fig. 8 shows the pipeline of the P-MCTS.

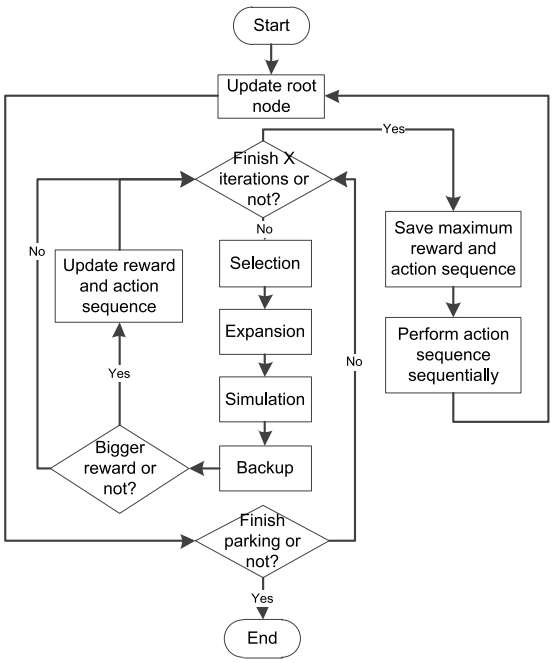


FIGURE 8. P-MCTS running pipeline.

V. CONSTRUCTION OF REWARD FUNCTION FOR DATA EVALUATION

In this section, we evaluate the generated parking data by constructing a reward function to select the best performing

data for subsequent network training. Safety, comfort, parking efficiency, and final parking posture are considered in the following reward function:

$$r = r_{safe} + r_{pos} + r_{com} + r_{eff} \tag{15}$$

where r is the total reward, r_{safe}, r_{pos}, r_{com}, and r_{eff} represent the reward of safety, final parking posture, comfort, and parking efficiency, respectively.

A. SAFETY

Safety is a requirement that an automatic parking system must meet. In this article, the collision method is used to construct the safety terms in the reward function. As shown in Fig. 9, in the parallel parking scenario, the ego-vehicle easily collides with the left and rear sides of the front vehicle and the left and front sides of the rear vehicle during the parking process. It has been considered in the longitudinal speed policy to avoid collision with the rear car. To avoid collision with the vehicle in front, a danger zone is set. Within d_{safe} from the left and rear sides of the front vehicle are hazardous areas, as shown by the shaded area in Fig. 9. Once the vehicle enters the above-mentioned area, it is considered that a collision has occurred and a large penalty value is assigned. Otherwise, the reward of safety is 0. Additionally, the safety distance d_{safe} is set to 0.15 m.

$$r_{safe} = \begin{cases} -20000 & if\ collision \\ 0 & else \end{cases} \tag{16}$$

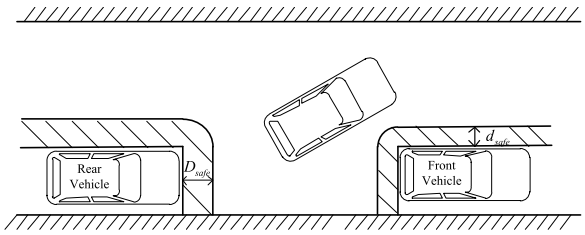


FIGURE 9. Parking collision risk area.

B. FINAL PARKING POSTURE

The reward of final posture in the entering parking slot stage is considered by the following:

$$r_{pos} = h - k\,(s - s_{tar}) \tag{17}$$

where s is the final vehicle posture in the entering parking slot stage, s_{tar} represents the target posture, k is the weight, and h denotes the upper limit value of the reward.

1) THE METRIC OF THE FINAL VEHICLE POSTURE DURING ENTERING PARKING SLOT STAGE

As shown in Fig. 10, F is a final posture during the entering parking slot stage. The ordinate of the midpoint of the rear axle is y_{tar}, and the closet distance of the vehicle to the rear is D_{safe} according to the vehicle speed policy. The positional

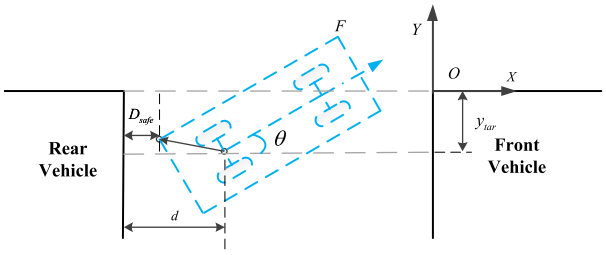


FIGURE 10. Relationship between the abscissa and heading angle of the final vehicle posture in the entering parking slot stage.

relationship between the left rear corner of the vehicle and the midpoint of the rear axle is fixed. When the heading angle F of is further determined, the abscissa of the midpoint of the rear axle is also determined. Therefore, the reward function only considers ordinate and heading angle.

2) DETERMINE THE TARGET VALUE OF POSTURE

If the vehicle eventually aligns with the adjacent parked vehicle, the final target ordinate $y = -0.5 \times W_{vehicle} = -0.78$m, where $W_{vehicle}$ is the width of the vehicle. However, in the case of a small parking space, a posture alignment process within the parking slot is also required so that the vehicle aligns with the parked vehicle. Fig. 11 shows a typical posture alignment process within the slot. In this case, the vehicle needs to shift twice in the parking space. The vehicle first reverses to point B, then drives forward to point C, and finally drives backward to point G. The posture of the vehicle at point B is the final vehicle posture of the entering parking slot stage, and the posture at point G represents the final parking posture.

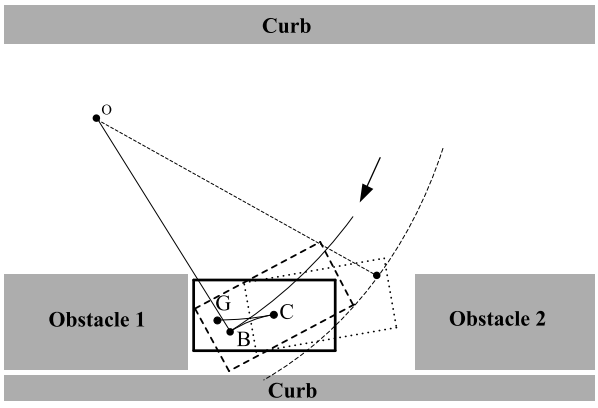


FIGURE 11. Posture alignment process in parallel parking spaces.

It can be seen from Fig. 11 that the ordinate of the midpoint of the rear axle of the vehicle increases from point B to point G. Therefore, it is necessary to determine the ascent distance during the alignment process. Considering the user experience of the automatic parking system and parking efficiency, the goal of the motion planning module in this article is to complete the parking process with no more than 6 shifts

in the parking slot. When the vehicle aligns posture in the parking slot, the minimum turning radius is adopted, and it stops when the distance to the front and rear vehicle is D_{safe}. We conclude that for different number of shifts, the maximum ascent distance is less than 0.07 m. Therefore, the target ordinate y_{tar} is set to -0.85 m. This ensures that the vehicle is still fully parked in the slot after the maximum ascent distance reached by the posture alignment.

The target heading angle θ_{tar} is set to 0°. The algorithm will make the final heading angle during the entering parking slot as small as possible while meeting safety and parking depth (i.e., the absolute value of the ordinate of final parking posture) requirements. Thus, the number of shifts in the parking slot will be reduced to improve parking efficiency.

In summary, the reward of final posture in the entering parking slot stage is as follows:

$$r_{pos} = \left(h_y + k_y \left|y - y_{tar}\right|\right) + \left(h_\theta + k_\theta \left|\theta - \theta_{tar}\right|\right) \quad (18)$$

where h_y and h_θ are the upper limit reward of y and θ, k_y and k_θ are reward weights in y and θ, y_{tar} and θ_{tar} are target values of y and θ.

C. COMFORT

During the parking process, the sudden acceleration and deceleration of the vehicle and the high-frequency swing or fast rotation of the steering wheel affect passenger comfort. The vibration model of the human sitting posture is specified in ISO 2631-1:1997 [32], including 12 axial vibrations from 3 input points. The vibration total value of weighted root mean square (r.m.s.) acceleration, determined from vibration in orthogonal coordinates is calculated as follows:

$$a_v = \sqrt{k_x^2 a_{wx}^2 + k_y^2 a_{wy}^2 + k_z^2 a_{wz}^2} \quad (19)$$

where a_{wx}, a_{wy}, a_{wz} are the weighted r.m.s. accelerations with respect to the orthogonal axels x, y, z, respectively, k_x, k_y, k_z are multiplying factors. The use of the vibration total value, a_v, is recommended for comfort. The values in Table 1 give approximate indications of likely reactions to various magnitudes of overall vibration total values.

TABLE 1. Human reactions to overall vibration total values.

Vibration total value a_v / $(m \cdot s^{-2})$	Reaction
<0.315	not uncomfortable
0.315-0.63	a little uncomfortable
0.5-1.0	fairly uncomfortable
0.8-1.6	uncomfortable
1.25-2.5	very uncomfortable
>2.0	extremely uncomfortable

Considering that the parking experiment is basically performed only on the x-y plane, it is considered that $a_{wz} = 0$. The method used in this article is to learn parking strategies through a lot of simulation and training. During the training process, the reward function is frequently used to evaluate the data. To speed up the efficiency of the algorithm, the calculation method is simplified here. Parking comfort in x is considered in the vehicle speed policy. Parking comfort in y is calculated as follows:

$$r_{com} = k_\delta \sum_{1}^{n-1} \left|\delta_{k+1} - \delta_k\right| \quad (20)$$

where δ_k and δ_{k+1} are the steering wheel angles for time steps k and $k+1$, respectively, k_δ is the weight of comfort. When the steering wheel changes smoothly without frequent shaking, the cost of this item is small.

D. PARKING EFFICIENCY

Parking efficiency measures the time it takes to complete parking meeting the requirements of safety and final parking posture. It is indirectly considered in the posture and comfort items. The larger the reward value of the posture item, the smaller the deviation between the final heading angle and the target heading angle. Thus, the smaller the heading angle that the vehicle needs to adjust in the subsequent posture alignment stage, the higher the parking efficiency. For the comfort item, the smaller the reward of the comfort item, the smoother the steering wheel angle. Therefore, the vehicle's path will also avoid unnecessary swings and improve comfort while reducing the parking time. Hence, it is not explicitly considered.

In summary, the reward function is constructed as follows,

$$r = \left(h_y + k_y \left|y - y_{tar}\right|\right) + \left(h_\theta + k_\theta \left|\theta - \theta_{tar}\right|\right) + k_\delta \sum_{1}^{n-1} \left|\delta_{k+1} - \delta_k\right| + r_{safe} \quad (21)$$

The requirements of the parking system for safety and final posture during the entering parking slot stage are more important than comfort. Therefore, the parameters in the reward function are determined, as shown in Table 2.

TABLE 2. Parameter table in the reward function.

h_y	k_y	y_{tar}	h_θ	k_θ	θ_{tar}	k_δ
20000	-29000	-0.85	20000	-29000	0	-1

VI. TRAINING NEURAL NETWORK

A. NETWORK STRUCTURE

By learning data with high reward selected by the reward function, a better parking performance can be achieved. Here a neural network is trained to learn the selected parking data.

The input feature of the network is the representation of the vehicle of different scenarios, i.e., the vehicle state and environment observation information. The vehicle state observation refers to the posture (x, y, θ) of the vehicle in the parking coordinate system, and the environment observation refers to the length L of the parking slot.

The output feature of the network is the probability distribution of different steering wheel angles. The range of

steering wheel angle during parking is very large. To reduce the number of network outputs, simplify the network structure, and improve the algorithm efficiency, the output of the network becomes the probability of different change values of steering wheel angle command at time step t +1 relative to the actual steering wheel angle at time step t. Considering the limitation of the electric power steering motor power during the parking process, the upper limit change of the steering wheel angle per unit step time (50 ms) is b = 20° (i.e., the maximum rotation speed of the steering wheel is 400°/s). Then the change range of the steering wheel angle per unit time step is $[-b, +b]$. The angel of change in this range is equally divided by 2°, and the probability corresponding to each angle change is $[P_{-b}, \ldots, P_{-2}, P_0, P_{+2}, \ldots, P_{+b}]$, $\sum_{i=-b}^{b} P_i = 1$. In summary, Table 3 shows the network input and output features.

TABLE 3. Input and output features for neural network.

Feature	Description
x	The real-time x coordinate of the vehicle
y	The real-time y coordinate of the vehicle
θ	The real-time θ of the vehicle
δ	The real-time actual steering wheel angle
L	The length of the parking slot
$p(\Delta\delta \mid s)$	The probability distribution over change values of the steering wheel angle $\Delta\delta$ in state s

The feed-forward back propagation network shown in Fig. 12 is used. The number of network inputs is 5. For determining the network size, the learning ability, real-time requirement of the algorithm, and prevention of over-fitting must be considered. Owing to a large amount of training data, a large-sized network should be used for training. However, the algorithm could not run in real-time during the real vehicle test. Therefore, the runtime of different network sizes determined via node search was tested. While meeting real-time requirements, we selected the largest network to achieve

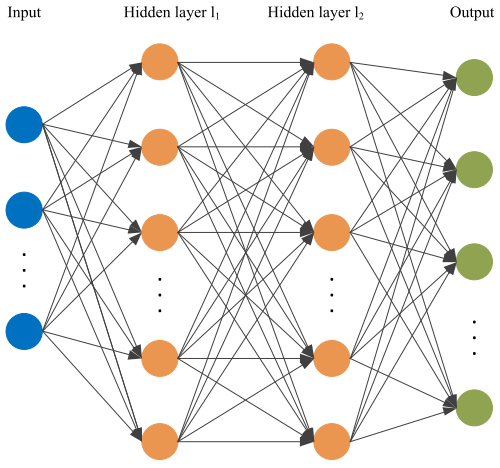


FIGURE 12. Structure of the lateral policy network.

maximum improvement in the network's learning ability. The determined network has two hidden layers, l_1 and l_2, with 25 nodes each.

The activation function of both hidden layers is tansig. Then, the number of nodes for probability corresponding to different steering wheel angle changes is 21. The activation function of the output layer is softmax. The probability of different steering wheel angle command values at the current moment can be determined based on network input and output.

Cross-entropy is used as a performance function to measure the difference between two probability distributions,

$$H(p, q) = -\sum p(x) \log q(x) \tag{22}$$

where $p(x)$ is the true probability distribution, $q(x)$ is the predicted probability distribution of the trained model. The network is trained on state-action pairs $(s, \Delta\delta)$, using scaled conjugate gradient propagation to minimize the cross-entropy.

B. TRAINING SCENARIO SETTING

Due to the limited generalization ability of the neural network, a large amount of training data is required to cover different parking scenarios. The test standard ISO 16787:2017 [33] describes the performance requirements and test procedures of the parking system. The standard mainly determines the size of the test parking slot, and performance requirements during slot search mode and for the end posture. From the performance requirements during the slot search mode, we can deduce the range of the start posture. Based on the performance test requirements and the vehicle parameters, the training scenarios are determined, as shown in Table 4.

TABLE 4. Training scenario settings and performance test requirements.

Parameter	Value
Length of parking slot	4.57m 4.75m 5.00m 5.25m 5.50m
Abscissa range of start posture	[1.00m, 3.00m]
Ordinate range of start posture	[1.03m, 2.28m]
Heading angle of start posture	[−5°,5°]
Heading angle of final posture	[−3°,3°]
Position of final posture	Completely parked in parking space

The standard test parking slot length corresponding to the experimental vehicle is 4.57 m. However, we will encounter different parking slot sizes. When the length of the parking slot is 5.5 m, the vehicle can enter the parking slot without shifts. Therefore, the range of parking slot length is $[4.57m, 5.5m]$. The range of initial heading angles between the ego-vehicle and the connecting line of parked vehicles is $[-5°, 5°]$. To reduce the training data and improve the convergence speed, the initial heading angle is set to 0°. The coordinate of training data is uniformly distributed at certain intervals in the above range of start posture.

VII. EXPERIMENT AND DISCUSSION

To verify the effectiveness of the proposed parking strategy reinforcement learning method, this section first analyzes the performance of the reinforcement learning process. Then, the effectiveness of the motion planning controller obtained by reinforcement learning is verified by real vehicle experiments. Finally, a benchmark test with a mass-produced parking system is performed.

A. VARIATION AND ANALYSIS OF PARKING PERFORMANCE DURING REINFORCEMENT LEARNING

The improvement of the training data reward during the reinforcement learning, illustrates the improvement of the parking strategy. Therefore, the mean sum reward change of the training data of all training scenarios during the reinforcement learning process, as shown in Fig. 13, is used to analyze the performance change of the parking strategy during the iteration.

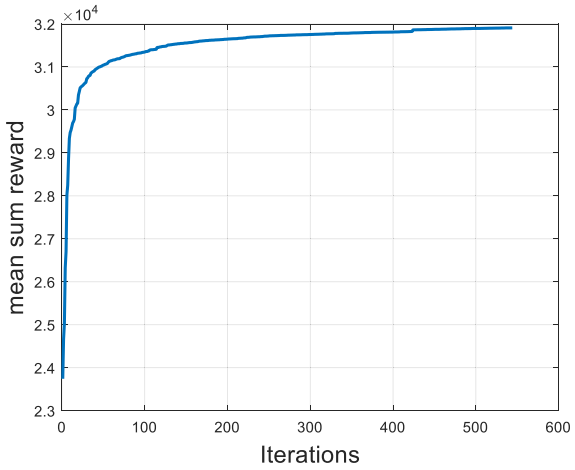


FIGURE 13. Mean sum reward of training data in the reinforcement learning process.

It can be seen from Fig. 13 that the mean sum reward gradually increases, and stabilizes eventually. Therefore, it is considered that the learned parking strategy tends to be optimal, and the learning process is terminated. Throughout the iteration process, the parking performance improved rapidly in the first 50 generations, and then the speed of improvement is reduced. This occurs because the parking strategy is very poor at the beginning of the learning, so the data reward value generated by the simulation is small. Once the simulation produces better parking data, they are stored to update the parameters of the network, making the action probability distribution provided by the neural network more reliable. The MCTS is combined with the guidance of the neural network, and at the same time, due to the exploration of the selection action, the parking performance can be improved steadily. When the training data reward is close to the ceiling of the system, the probability of being able to explore better quality data decreases. Therefore, the improved speed of parking performance reduces. The above process illustrates the effectiveness of the proposed reinforcement learning method.

The improvement of the training data reward for two consecutive generations in Fig. 13 shows that the data produced by the neural network combined with the MCTS (k + 1th) is superior to data previously used for training the network (k th), as shown in Fig. 2. Due to network training errors, the quality of training data can be regarded as the ceiling of the parking performance of the trained network. Therefore, the parking performance of the neural network combined with MCTS is better than the neural network itself. This suggests that the parking strategies based on the reinforcement learning methods are superior to expert data-based methods, such as the neural network.

In the reinforcement learning process, the mean reward of each item in the reward function of the training data is shown in Fig. 14. According to the reward function constructed in Section 5, the ceiling of the reward value is 40000. The final mean sum reward of the training data, as shown in Fig. 13, approaches 32000. It is because the target heading angle is set to 0°, but for a small size parking slot, the heading angle of the vehicle at the end of entering the parking slot stage has the deviation. As shown in Fig. 14(c), the mean reward of heading angle item eventually tends to 14000. Besides, the steering wheel angle will not remain constant throughout the parking process. Therefore, there must be some penalty for the comfort item as shown in Fig. 14(d). Figs. 14(a) and 14(b) illustrate that as the iteration goes on, the parking strategy ensures the safety and parking depth.

It can also be seen from Fig. 14 that the performance of the parking strategy in terms of safety and posture increases rapidly, and then tends to become stable. In contrast, the performance of comfort items improves more slowly. In the constructed reward function, the reward value of safety and pose items account for a large weight in the total reward value. Therefore, in the learning process, the parking strategy will first satisfy the item with a greater weight. This shows the effectiveness of the constructed reward function. On the one hand, the reward function ensures the performance of the parking strategy on the requirements mentioned above. On the other hand, by determining the weighs of different indexes in the reward function, the parking strategy meets the important indexes first.

B. REAL VEHICLE TEST

1) VEHICLE EXPERIMENT PLATFORM AND PROCESS

The experiment platform is refitted from the Roewe E50 electric vehicle (see Fig. 15). A 2D lidar is used for parking slot detection during the slot searching. Six ultrasonic sensors are used to detect the distance to the front and rear parked vehicles during the posture alignment in the parking space. The automatic parking system algorithms run in the MicroAutoBox (dSPACE) controller, with a CPU clock speed of 300MHz.

Fig. 16 shows the real vehicle test scenario and a parking process when the slot length (4.57 m) is the standard

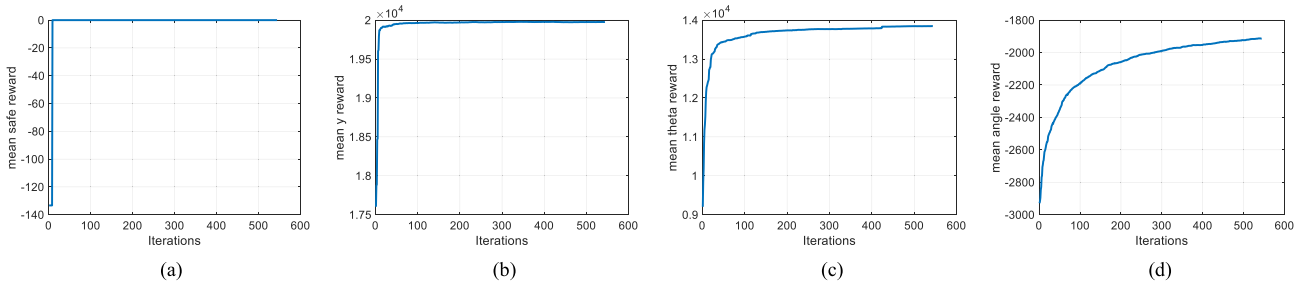


FIGURE 14. Mean reward of each item in the reward function of training data in the reinforcement learning process. (a) Safety. (b) Y in the posture. (c) θ in the posture. (d) Comfort.

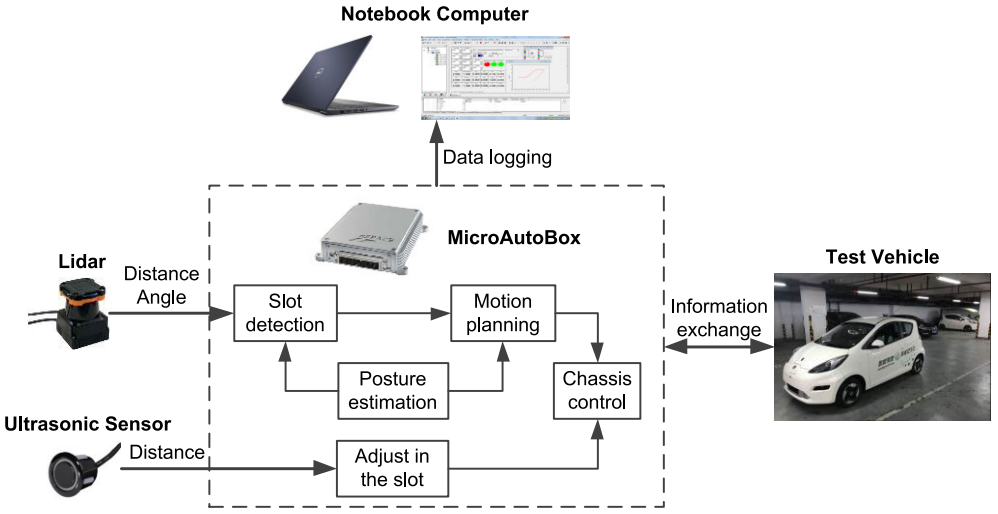


FIGURE 15. Experimental platform.

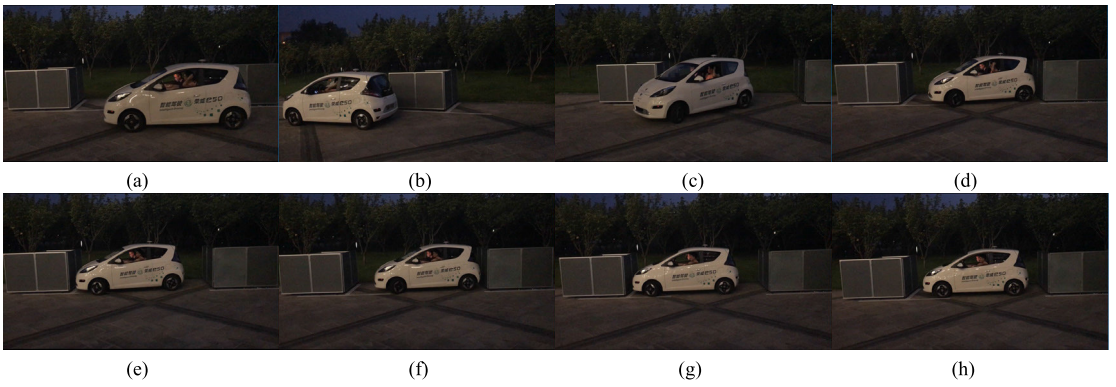


FIGURE 16. Real vehicle parking process. (a) Slot detection. (b) The parking slot is detected, and the vehicle starts to reverse. (c) The entering parking slot stage. (d) Finish the entering parking slot stage. (e)-(h) Adjusting the heading angle of the vehicle in the parking slot.

test scenario corresponding to the experimental vehicle. The parking slot is formed by front and rear obstacles shown in Fig. 16. The automatic parking system uses a 2D lidar to detect the parking slot first [34]. Then, the motion planning module, based on the detected parking slot information and the real-time vehicle posture estimated by the dead reckoning module, plans the action command in real-time and sends it to the chassis control module for execution. After the vehicle enters into the parking slot, the system determines whether the heading angle of the vehicle is smaller than the threshold. If it is true, the parking is completed. Otherwise, based on the distance to the front and rear parked cars detected by the ultrasonic radar, a method similar to the mass-produced parking system adjusting posture in the

parallel parking slot [35] is used to align the ego-vehicle with obstacles.

2) REAL VEHICLE EXPERIMENT

The real vehicle experiment is to verify the performance of the algorithm's generalization ability, real-time performance, and parking performance on different parking scenarios, where there are errors in the detected parking slot and the established vehicle model. We choose three typical parking scenarios with different lengths of 4.57, 5, and 5.5 m.

The length of the parking slot shown in Fig. 17(a) is 4.57 m, and the length detected by the parking slot module is 4.45 m. The start posture in the parking coordinate system is $(1.75m, 1.92m, -0.65°)$, and the final parking posture is $(-3.76m, -0.83m, -0.76°)$. The green line in the figure is the path of the midpoint of the rear axle of the vehicle during the entire parking process. It can be seen from the path that the vehicle completed the parking with six shifts after the entering parking slot stage, which meets the parking efficiency requirements. The blue rectangles represent the area scanned by the vehicle outline during the entering parking slot stage. There is no collision with the obstacles during the parking process, which meets the safety requirements. From the final vehicle posture, it can be seen that the parking depth of the vehicle is greater than 0.78 m, and the final heading angle is $-0.76°$, suggesting that the vehicle is completely parked in the slot and align with the obstacles, which satisfies the final vehicle posture requirements.

Figs. 17(b) and 17(c) show desired and actual steering wheel angle and vehicle speed during the entering parking slot stage. The commands of steering angle and vehicle speed are smooth. The weighted acceleration during the parking is 0.40m/s^2. The human reaction is a little uncomfortable with this vibration total value. It can be seen from the figure that there is a deviation between the action command and the actual value, further illustrating the necessity of the vehicle model to consider the chassis control deviation. The algorithm calculates the control command with a cycle of 50 ms for one planning loop, which satisfies the real-time requirements.

The experimental results of 5 and 5.5 m parking lengths are shown in Figs. 18 and 19, respectively. The lengths detected by the parking slot module are 4.99 and 5.44 m, respectively.

In Fig. 18, the vehicle's start posture is $(1.81m, 2.05m, -1.66°)$, and the final parking posture is $(-4.18m, -0.79m, -0.8°)$. It can be seen from the path that the vehicle successfully completed the parking with two shifts after the entering parking slot stage without collision, which meets the requirements of parking efficiency and safety. In Fig. 19, the vehicle's start posture is $(1.70m, 2.28m, 0.92°)$, and the final parking posture is $(-4.73m, -0.78m, -0.69°)$.The vehicle completed the

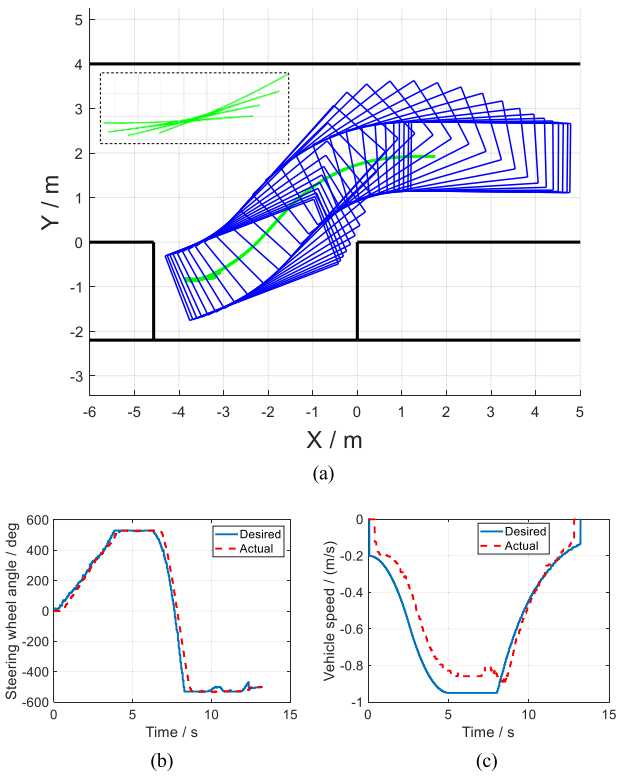


FIGURE 17. Experimental results of 4.57 m parking slot. (a) Parking path. (b) Desired and actual steering wheel angle. (c) Desired and actual vehicle speed.

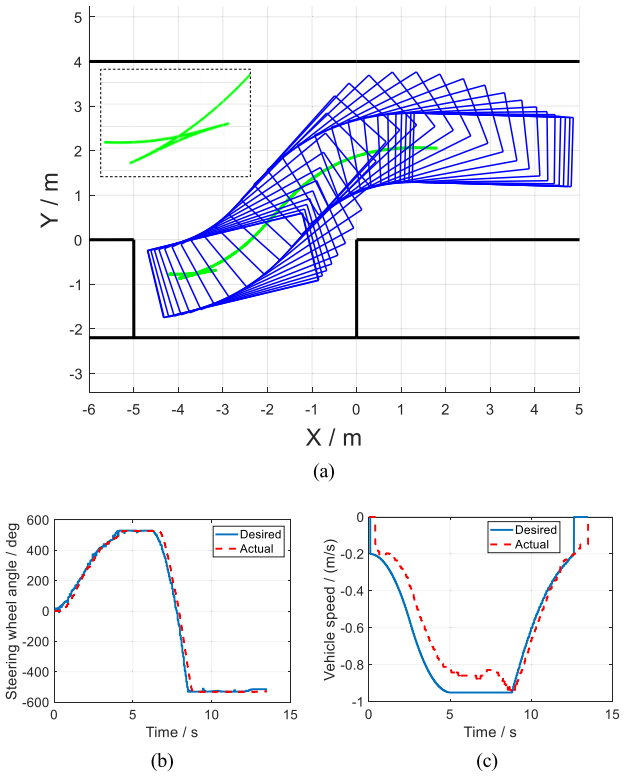


FIGURE 18. Experimental results of 5 m parking slot. (a) Parking path. (b) Desired and actual steering wheel angle. (c) Desired and actual vehicle speed.

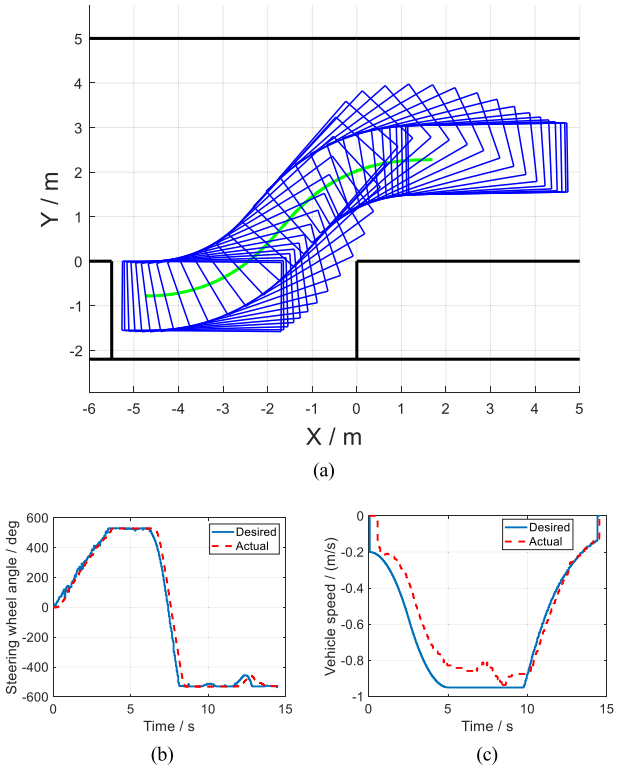


FIGURE 19. Experimental results of 5.5 m parking slot. (a) Parking path. (b) Desired and actual steering wheel angle. (c) Desired and actual vehicle speed.

parking after reversing into the parking slot without collision, which satisfies the requirements of parking efficiency and safety. In both cases, the deviations of the vehicle's final heading angle are less than 1° and the parking depths are greater than 0.78 m, suggesting that the vehicle is completely parked in the slot and align with the obstacles, which meets the final vehicle posture requirements.

The commands of steering angle and vehicle speed in both cases are smooth. The weighted accelerations during the two parking scenarios are 0.31m/s^2 and 0.12m/s^2. The human reactions are not uncomfortable with the two vibration total values. The algorithm calculates the control command with a cycle of 50 ms for one planning loop, which meets the real-time requirements.

3) BENCHMARK TEST WITH MASS-PRODUCED PARKING SYSTEM

In this section a benchmark test with a mass-produced parking system is performed. Among mass-produced parking system suppliers, Bosch occupies a large market share and is most technically representative. The parking planning method it uses is the geometric method of the prior knowledge-based method [36], which is also the mainstream method of mass-produced parking system suppliers [37], [38]. Therefore, we selected the latest generation of the parking assist system of Bosch for the benchmark. The system is deployed on the Skoda Kodiaq. Then, due to the difference in the size of the test vehicle, the size of the corresponding test parking slot is determined based on the test standard of the parking assist system ISO 16787:2017 and the test vehicle parameters, as shown in Table 5.

TABLE 5. Test vehicle parameters and corresponding slot length.

Test vehicle	Length/m	Width/m	Wheelbase/m	slot length/m
Roewe	3.569	1.551	2.305	4.57
Kodiaq	4.698	1.883	2.791	5.87

Most of the mass-produced parking system is a parking assist system. The steering wheel is automatically controlled by the parking system, while the driver needs to control the throttle, brake, and gear. To ensure the fairness of the test, during the mass-produced system test, the driver does not control the accelerator pedal at all. The car runs at idling speed (i.e., 4km/h). When the vehicle approaches front and rear obstacles, the driver controls the brake pedal to slow down. At the same time, the driver pays attention to the dashboard and immediately takes action as prompted.

The difference in configuration (start and end postures) may affect the ease or complexity of searching the final solution for nonholonomic systems.

The end posture depends on vehicle parameters. To ensure fairness during the test, the requirements for the end posture of the two systems are based on the test standard ISO 16787:2017, as shown in Table 4.

Due to the nonholonomic constraints, the vehicle cannot move laterally or change the heading angle without moving. Therefore, y and θ in the start posture have a considerable impact on the complexity of finding the solution. The test standard specifies performance requirements during slot search mode. According to the test standard, the supported lateral clearing distance to parked vehicles shall be in the range of $[0.5m, 1.5m]$, and the angle between the ego-vehicle and connecting line of the parked vehicles shall be in the range of $[-5°, 5°]$. The above performance requirements determine the range of y and θ in the start posture. Thus, the requirements for y and θ in the start posture of the two systems are also based on the test standard.

However, x in the start posture, is not specified in the test standard. Thus, it is controlled and determined by the parking systems. For the mass-produced parking system, as mentioned above, the driver diligently follows the system prompts and does not actively intervene.

In summary, factors that significantly influence the nonholonomic restraint system, such as y and θ in the start posture, and the end posture, are determined based on the test standard. For x in the start posture, as there is no regulation provided in the test standard, it is controlled and determined by the respective parking system. This ensures the fairness of the benchmark test as much as possible without active human intervention in the parking system.

The data recorded in the parking process are shown in Tables 6(a) and 6(b), including the start parking posture,

TABLE 6. Experimental results of benchmark tests. (a) Mass-produced parking system. (b) Reinforcement learning parking system.

(a)

Num	Start posture			Final posture			Number of shifts	Parking time/s
	x / m	y / m	θ / deg	x / m	y / m	θ / deg		
1	8.39	2.31	4.21	-4.59	-1.10	0.51	8	68
2	6.07	2.65	1.15	-4.44	-1.16	0.69	14	87
3	5.25	1.81	1.63	-4.07	-1.02	-1.05	9	57
4	4.09	1.73	1.30	-4.71	-1.01	0.60	6	48
5	5.27	1.61	2.93	-4.61	-1.08	1.45	7	50
Mean	5.81	2.02	----	----	-1.07	0.44	8.8	62
Std deviation	----	----	----	----	0.06	0.91	3.11	16.02

(b)

Num	Start posture			Final posture			Number of shifts	Parking time/s
	x / m	y / m	θ / deg	x / m	y / m	θ / deg		
1	1.75	1.92	-0.65	-3.76	-0.83	-0.76	6	32
2	1.74	1.95	1.08	-3.68	-0.79	3.40	4	27
3	1.70	1.77	0.81	-3.67	-0.90	-0.50	6	30
4	1.78	1.81	-0.67	-3.88	-0.80	-2.28	5	29
5	1.76	1.94	0.65	-3.20	-0.82	-2.54	5	31
Mean	1.75	1.88	----	----	-0.83	-0.54	5.2	29.8
Std deviation	----	----	----	----	0.04	2.38	0.84	1.92

final parking posture, total parking time (time from vehicle reverse into the parking slot to the completion of parking), and the number of shifts.

The results of the start posture demonstrate that the average lateral distance of the mass-produced parking system to parked vehicles is $d_1 = 2.02 - 1.88/2 = 1.08$m, whereas the distance of our proposed system is $d_2 = 1.88 - 1.55/2 = 1.1$m. d_1 and d_2 are basically equal and within the range of [0.5m, 1.5m]. The start heading angles of the two parking systems are also within the range of $[-5°, 5°]$, as specified by the test standard.

During the experiment, neither of the two parking systems collided with the parked obstacle, which confirms the safety of the parking systems.

For the final parking posture, the target final heading angle is 0°. The target final parking depth depends on the width of the vehicle. It can be seen from the parking coordinate system shown in Fig. 3 that when the parking depth is greater than half the vehicle width, it indicates that the vehicle is completely parked in the slot. From the vehicle parameters shown in Table 5, the parking depth of the mass-produced parking system and our parking system should be greater than 0.99 and 0.78 m, respectively. The average parking depths of the two parking systems, as shown in Tables 6(a) and 6(b), are 1.07 and 0.83 m, which meets the parking depth requirements mentioned above. For the heading angle of the final parking posture, the mean deviation of our automatic parking system (0.54°) is basically equal to that of the mass-produced parking system (0.44°). However, the standard deviation of the final heading angle of the mass-produced parking system is smaller, which suggests that its performance is more stable. The heading angle of the vehicle in the parking slot is determined according to the distances returned by the ultrasonic sensors to the front and rear obstacles. Due to the low refresh rate (5Hz) of the ultrasonic sensor used in our parking system, the vehicle cannot respond in time during the posture alignment process in the parking slot, which affects the stability of the final parking posture.

Regarding the number of shifts and parking time, our parking system performs better than the mass-produced one. The average number of shifts in our proposed system is 3.6 less, and the average parking time is half of the mass produced system. The standard deviation of the above two items of our proposed system is also much smaller.

There are two main reasons for the short parking time of our parking system through the analysis of the experimental results. First, the mean start posture of our parking system is closer to the parking slot. It can be seen from Table 6(a) and Table 6(b) that the abscissas of our parking system are much smaller than the mass-produced parking system. Therefore, the parking path is also shorter, making parking more efficient. After the mass-produced system detects the parking slot, the driver will be prompted to continue driving for a period before starting to park, which makes the vehicle far away from the parking slot. This is probably because the vehicle needs to reach a certain area before parking, i.e., the start posture range where it can park is limited. However, our parking system has large data coverage during training. This allows the motion planning module to adapt to a variety of parking scenarios, reducing the start parking posture requirements, and thus reducing parking space external to the parking slot. Second, the average number of shifts in our parking system is significantly smaller. It can be seen from Tables 6(a) and 6(b) that the number of shifts has a great influence on the parking time. At the same time, the final vehicle posture of the entering parking slot stage has a decisive influence on the number of shifts during the subsequent posture alignment stage. Therefore, it can be

concluded that our parking system has a better posture at the end of entering the parking slot stage.

In summary, on the basis of ensuring safety and the final parking posture performance, our system can effectively reduce the number of shifts and parking time, improving parking efficiency and driver experience significantly. Meanwhile, our system has a lower requirement for the start parking posture. The results demonstrate the superiority of the proposed reinforcement learning algorithm.

VIII. CONCLUSION

In this article, we innovatively propose a model-based reinforcement learning method for automatic parking motion planning. The method removes human experience to a great extent and learns parking quickly and autonomously. The learned strategies ensure multi-objective optimization, including safety, comfort, paring efficiency and final parking performance. First, a framework of reinforcement learning methods for the parking strategies is proposed, which iteratively executes data generation, data evaluation, and training network to learn data knowledge. The updated network is used for the subsequent iteration cycle of the data generation. To achieve a fast learning, a vehicle model based on the transfer function combined with a kinematic vehicle model is constructed to simulate the interaction between vehicle and environment. Then, based on the model, a large amount of data is obtained using a data generation algorithm. Besides, to select the parking data with the best performance, a reward function is constructed, and the above performance requirements are considered comprehensively. Finally, a neural network is trained to learn the knowledge of selected parking data.

In future research, other influencing factors in the environment, such as environmental perception and road environment can be considered during modeling, to further improve the accuracy of the model, and thus improve the performance of the parking system. Furthermore, the reinforcement learning algorithms can be extended to the longitudinal policy to further reduce the introduction of human experience.

REFERENCES

[1] Z. Yao, B. Zhao, T. Yuan, H. Jiang, and Y. Jiang, "Reducing gasoline consumption in mixed connected automated vehicles environment: A joint optimization framework for traffic signals and vehicle trajectory," *J. Cleaner Prod.*, vol. 265, Aug. 2020, Art. no. 121836.

[2] Z. Yao, Y. Jiang, B. Zhao, X. Luo, and B. Peng, "A dynamic optimization method for adaptive signal control in a connected vehicle environment," *J. Intell. Transp. Syst.*, vol. 24, no. 2, pp. 184–200, Mar. 2020.

[3] W. Liu, Z. Li, L. Li, and F.-Y. Wang, "Parking like a human: A direct trajectory planning solution," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 12, pp. 3388–3397, Dec. 2017.

[4] Y. Lin, L. Li, X. Dai, N. Zheng, and F. Wang, "Master general parking skill via deep learning," in *Proc. IEEE Int. Vehicles Symp. (IV)*, Jun. 2017, pp. 941–946.

[5] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre, G. van den Driessche, T. Graepel, and D. Hassabis, "Mastering the game of go without human knowledge," *Nature*, vol. 550, no. 7676, pp. 354–359, Oct. 2017.

[6] P. Petrov, F. Nashashibi, and M. Marouf, "Path planning and steering control for an automatic perpendicular parking assist system," in *Proc. 7th Workshop PPNIV*, vol. 15, 2015, pp. 143–148.

[7] J. M. Kim, K. I. Lim, and J. H. Kim, "Auto parking path planning system using modified Reeds-Shepp curve algorithm," in *Proc. 11th Int. Conf. Ubiquitous Robots Ambient Intell. (URAI)*, Kuala Lumpur, Malaysia, Nov. 2014, pp. 311–315.

[8] H. Vorobieva, S. Glaser, N. Minoiu-Enache, and S. Mammar, "Automatic parallel parking with geometric continuous-curvature path planning," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2014, pp. 465–471.

[9] H. Vorobieva, S. Glaser, N. Minoiu-Enache, and S. Mammar, "Automatic parallel parking in tiny spots: Path planning and control," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 1, pp. 396–410, Feb. 2015.

[10] Z. Liang, G. Zheng, and J. Li, "Automatic parking path optimization based on Bezier curve fitting," in *Proc. IEEE Int. Conf. Autom. Logistics*, Zhengzhou, China, Aug. 2012, pp. 583–587.

[11] F. Gómez-Bravo, F. Cuesta, A. Ollero, and A. Viguria, "Continuous curvature path generation based on β-spline curves for parking Manoeuvres," *Robot. Auto. Syst.*, vol. 56, no. 4, pp. 360–372, Apr. 2008.

[12] S. Zhang, M. Simkani, and M. H. Zadeh, "Automatic vehicle parallel parking design using fifth degree polynomial path planning," in *Proc. IEEE Veh. Technol. Conf. (VTC Fall)*, Sep. 2011, pp. 1–4.

[13] L. Cheng, C. Liu, and B. Yan, "Improved hierarchical A-star algorithm for optimal parking path planning of the large parking lot," in *Proc. IEEE Int. Conf. Inf. Autom. (ICIA)*, Jul. 2014, pp. 695–698.

[14] C. M. Sánchez, M. S. Peñas, and L. G. Salvador, "A fuzzy decision system for an autonomous car parking," in *Handbook on Decision Making*. Berlin, Germany: Springer, 2012, pp. 237–258.

[15] Z.-L. Wang, C.-H. Yang, and T.-Y. Guo, "The design of an autonomous parallel parking neuro-fuzzy controller for a car-like mobile robot," in *Proc. SICE Annu. Conf.*, Aug. 2010, pp. 2593–2599.

[16] P. Zhang, L. Xiong, Z. Yu, P. Fang, S. Yan, J. Yao, and Y. Zhou, "Reinforcement learning-based end-to-end parking for automatic parking system," *Sensors*, vol. 19, no. 18, p. 3996, Sep. 2019.

[17] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," 2015, *arXiv:1509.02971*. [Online]. Available: https://arxiv.org/abs/1509.02971

[18] C. You, J. Lu, D. Filev, and P. Tsiotras, "Highway traffic modeling and decision making for autonomous vehicle using reinforcement learning," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2018, pp. 1227–1232.

[19] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015.

[20] D. Isele, A. Cosgun, and K. Fujimura, "Analyzing knowledge transfer in deep Q-networks for autonomously handling multiple intersections," 2017, *arXiv:1705.01197*. [Online]. Available: http://arxiv.org/abs/1705.01197

[21] N. Xu, B. Tan, and B. Kong, "Autonomous driving in reality with reinforcement learning and image translation," 2018, *arXiv:1801.05299*. [Online]. Available: http://arxiv.org/abs/1801.05299

[22] X. Pan, Y. You, Z. Wang, and C. Lu, "Virtual to real reinforcement learning for autonomous driving," 2017, *arXiv:1704.03952*. [Online]. Available: http://arxiv.org/abs/1704.03952

[23] A. El Sallab, M. Abdou, E. Perot, and S. Yogamani, "End-to-end deep reinforcement learning for lane keeping assist," 2016, *arXiv:1612.04340*. [Online]. Available: http://arxiv.org/abs/1612.04340

[24] A. Kendall, J. Hawke, D. Janz, P. Mazur, D. Reda, J.-M. Allen, V.-D. Lam, A. Bewley, and A. Shah, "Learning to drive in a day," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 8248–8254.

[25] L. Yu, X. Shao, Y. Wei, and K. Zhou, "Intelligent land-vehicle model transfer trajectory planning method based on deep reinforcement learning," *Sensors*, vol. 18, no. 9, p. 2905, 2018.

[26] L. Kaiser, M. Babaeizadeh, P. Milos, B. Osinski, R. H. Campbell, K. Czechowski, D. Erhan, C. Finn, S. Levine, A. Mohiuddin, R. Sepassi, G. Tucker, P. Kozakowski, and H. Michalewski, "Model-based reinforcement learning for Atari," 2019, *arXiv:1903.00374*. [Online]. Available: http://arxiv.org/abs/1903.00374

[27] M. Hessel, J. Modayil, H. von Hasselt, T. Schaul, G. Ostrovski, W. Dabney, D. Horgan, B. Piot, M. Azar, and D. Silver, "Rainbow: Combining improvements in deep reinforcement learning," in *Proc. 32th AAAI Conf. Artif. Intell.*, Apr. 2018, pp. 1–14.
[28] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017, *arXiv:1707.06347*. [Online]. Available: http://arxiv.org/abs/1707.06347
[29] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, G. van den Driessche, and D. Hassabis, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, Jan. 2016.
[30] G. Chaslot, "Monte-Carlo tree search," Ph.D. dissertation, Dept. Data Sci. Knowl. Eng., Maastricht Univ., Maastricht, The Netherlands, 2010.
[31] C. B. Browne, E. Powley, D. Whitehouse, S. M. Lucas, P. I. Cowling, P. Rohlfshagen, S. Tavener, D. Perez, S. Samothrakis, and S. Colton, "A survey of Monte Carlo tree search methods," *IEEE Trans. Comput. Intell. AI in Games*, vol. 4, no. 1, pp. 1–43, Mar. 2012.
[32] *Mechanical Vibration and Shock-Evaluation of Human Exposure to Whole-Body Vibration—Part 1: General Requirements*, ISO Standard 2631-1, 1997.
[33] *Intelligent Transport Systems-Assisted Parking System (APS)-Performance Requirements and Test Procedures*, ISO Standard 16787, 2017.
[34] Q. Yang, H. Chen, J. Su, and J. Li, "Towards high accuracy parking slot detection for automated valet parking system," SAE Tech. Paper 2019-01-5061, Nov. 2019.
[35] J. Bammert, N. Jecker, S. Kossmann, V. Vovkuschevsky, and K. Hoffsommer, "Method and device for assisting a driver of a vehicle in exiting from a parking space," U.S. Patent 8 560 175, Oct. 15, 2013.
[36] C. Danz, J. Egelhaaf, W. C. Lee, and B. Steiner, "Method and device for the assisted parking of a motor vehicle," U.S. Patent 8 655 551, Feb. 18, 2014.
[37] H. Barth and N. Jecker, "Method and device for planning a path when parking a vehicle," U.S. Patent 8 497 782, Jul. 30, 2013.
[38] Y. Tanaka, "Parking assist system," U.S. Patent 8 374 749, Feb. 12, 2013.

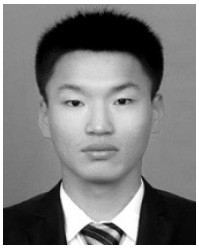
JIREN ZHANG (Member, IEEE) received the B.S. degree in mechanical and electronic engineering from Northwest A&F University, Shaanxi, China, in 2013. He is currently pursuing the Ph.D. degree with the School of Automotive Studies, Tongji University, China, under the supervision of Prof. H. Chen. His research interests include automatic parking systems, motion planning, artificial neural networks, and reinforcement learning.

HUI CHEN (Member, IEEE) received the B.S. and M.S. degrees in automation from Donghua University, Shanghai, China, in 1985 and 1988, respectively, and the Ph.D. degree in electronics and information engineering from Yokohama National University, Yokohama, Japan, in 1996.

From 1988 to 1993, he was a Research Assistant with the Department of Automation, Donghua University. From 1996 to 2002, he was the Assistant Manager of the Research and Development Center, Electric Power Steering System (EPS) Division, NSK. Since 2002, he has been a Professor with Tongji University and served as the Director of the Chassis Electronic Control System Research Center. In 2010, he founded the School of Automotive Studies, Tongji University–JTEKT Automotive Active Safety Technology Joint Laboratory and served as the Laboratory Manager since then. His research interests include automotive chassis electronic control systems and intelligent vehicle technology.

Dr. Chen is a member of SAE and JSAE. He serves as the Director of SAE, China, the Secretary General of the Automotive Intelligent Transportation Branch, and a member of the Steering System Technology Branch.

SHAOYU SONG received the B.S. and M.S. degrees in vehicle engineering from Chongqing University, Chongqing, China, in 2015 and 2018, respectively. He is currently pursuing the Ph.D. degree with the School of Automotive Studies, Tongji University, China, under the supervision of Prof. H. Chen. His research interests include autonomous driving, motion planning, artificial neural networks, and reinforcement learning.



FENGWEI HU received the B.S. degree in vehicle engineering from Jilin University, Jilin, China, in 2017. He is currently pursuing the M.S. degree with the School of Automotive Studies, Tongji University, China, under the supervision of Prof. H. Chen. His research interests include automatic parking systems, motion planning, and Monte Carlo methods.