Scientia Iranica B (2021) 28(3), 1318–1328

Sharif University of Technology
Scientia Iranica
Transactions B: Mechanical Engineering
http://scientiairanica.sharif.edu

Research Note

A safety navigation method for integrating global path planning and local obstacle avoidance for self-driving cars in a dynamic environment

J.K. Yin[a,b] and W.P. Fu[a,*]

a. *School of Mechanical and Precision Instrument Engineering, Xi'an University of Technology, Xi'an, Shaanxi 710048, China.*
b. *College of Applied Engineering, Henan University of Science and Technology, Sanmenxia, Henan, 472000, China.*

Received 10 December 2018; received in revised form 9 January 2020; accepted 13 July 2020

KEYWORDS
Self-driving car;
Global path planning;
Local obstacle avoidance;
Uncertain dynamic obstacle;
Dynamic environment.

Abstract. This paper proposes a novel method to obtain high-quality paths for self-driving cars in underground parking lots. Self-driving cars require fast and accurate planning of collision less paths. When the self-driving car arrives at a parking lot, it downloads the layout from the corresponding intelligent system and is assigned a parking space; then, the locations of the designated parking space and the car are provided and pinpointed by the intelligent system. A global path is planned by the global algorithm according to the location of both parking space and car. In case of detecting dynamic or unknown obstacles in the process of moving along the global path, the parameters of obstacles can be estimated using the obstacle-detection algorithm. According to these obtained parameters, the local obstacle avoidance path can be planned through the behavior dynamics method. After completing the obstacle avoidance, the car returns to the global path and continues to move toward the target parking space. Finally, the proposed method was simulated by MATLAB, and the results showed that the car could safely park in the target parking space. This method could simultaneously satisfy the smooth and real-time requirements of path planning.

1. Introduction

The concept of self-driving car has recently received extensive attention by many research institutions and it has been applied to military, transportation, and other scientific fields [1]. The performance of path planning determines the intelligence of a self-driving car and is one of the most important core technologies among the related technologies of the self-driving car. The main objective of path planning is to find a continuous and collisionless path from the initial position to the target position so that the path can satisfy the environment, real-time, and kinematic and dynamic constraints associated with self-driving cars.

Given the congested traffic in urban scenarios, a cluttered parking lot would be a challenging place to navigate. Domokos and Gábor [2] presented a global planning method for car-like vehicles, producing paths with continuous curvature. Their proposed method used straight segments, CC_{in}-C-CC_{out} triplets (CC-turns), and elementary paths to generate a feasible and human-like solution, even in narrow environments. However, it was limited to a pre-set environment. Kim et al. [3] adopted an improved Reeds-Shepp curve algorithm for an effective forward and backward auto-

*. *Corresponding author. Tel.: +86 131 8618 6562; Fax: +029 8231 2626*
E-mail addresses: Yjk516@163.com (J.K. Yin); weipingf@xaut.edu.cn (W.P. Fu)

doi: 10.24200/sci.2020.52417.2704

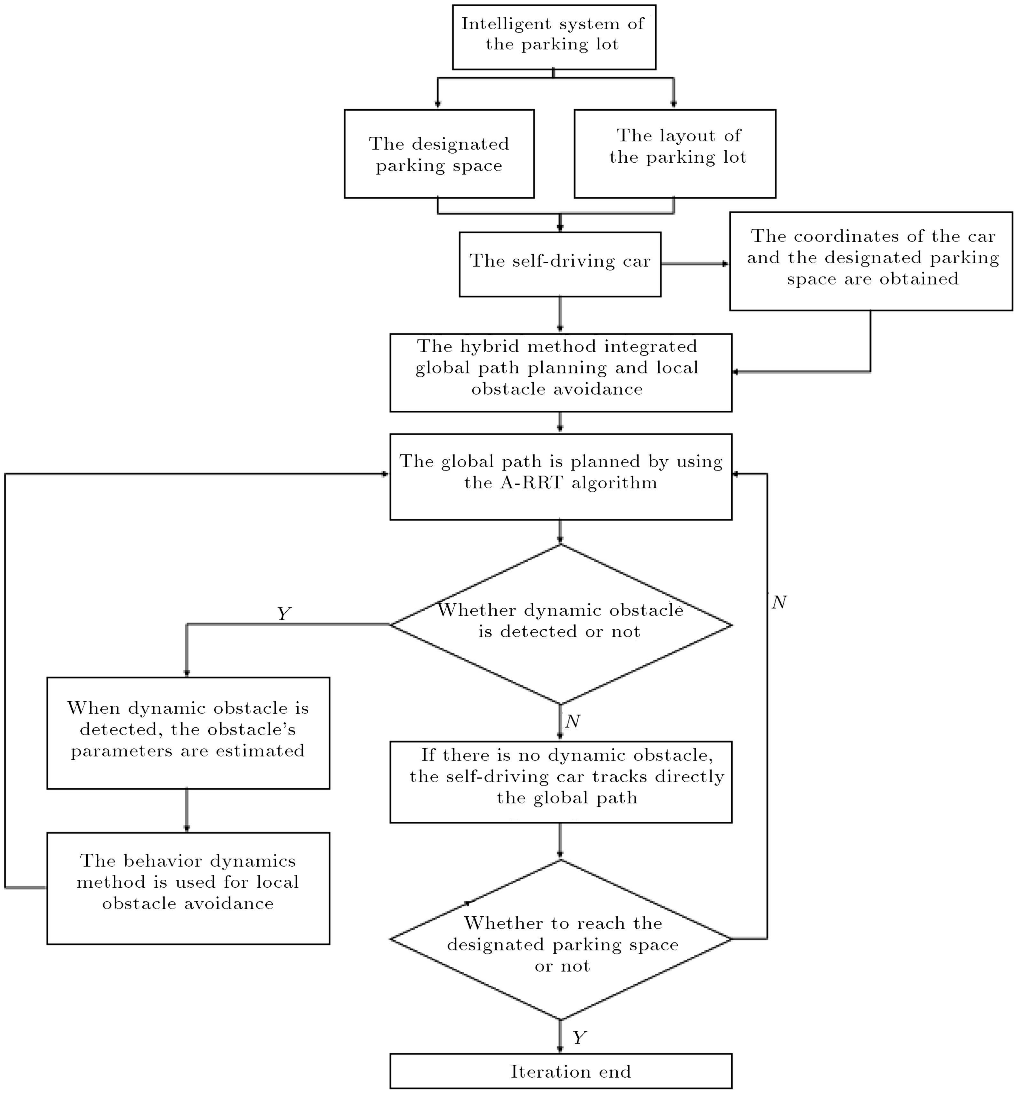


Figure 1. Flow chart of the safety navigation method.

stop system. Their proposed auto-stop system enjoyed an advantage, that is, the vehicle control system was equipped with simple command data making the vehicle move forward, backward, and laterally, as well as tracking the travel distance values. However, obstacle detection was neglected in the parking process. Wang et al. [4] modeled a parking lot with a time-varying graph. Their proposed system utilized a time-dependent shortest-path algorithm and dynamically tuned arc transit times based on planned vehicle routes and traffic flow sensor data. Simulation results showed an average travel time reduction of more than 40% for each in the best case. However, in the proposed system, when a road was blocked at time t, for the simplicity of implementation, the transit time of the corresponding arc was chosen so that it could be increased at time t by a constant to two times the speed limit of the parking lot. In some cases, this method may cause an over-tuning problem, i.e., the system tunes the transit time of an arc to an unnecessarily large value so that the time-dependent shortest-path algorithm cannot output the actual optimal route.

In response to the above deficiencies, in this paper, a safety navigation method integrating global path planning and local obstacle avoidance was proposed to make the planned path suitable for the self-driving car. The flow chart of the safety navigation method is shown in Figure 1.

A great number of path-planning algorithms have been taken from the self-driving car to face the challenges of road networks and driving rules. The most relevant path planning algorithms implemented in motion planning for self-driving car are described below.

The A* algorithm is a graph search algorithm that enables a fast node search due to the implementation of heuristics. Its most significant design aspect is heuristic reducing computation time, yet the planning path is not continuous and has many turns [5,6]. The

Probabilistic Road Map (PRM) is a graph-based search method that randomly selects N nodes in a planning space. These nodes are first connected, the connection lines with the obstacles are removed, and a path is obtained [7]. Dijkstra's algorithm finds the shortest path in a series of nodes or grids and is suitable for global planning in both structured and unstructured environments. However, it functions slowly in vast areas due to the significant number of nodes. Since the search is not heuristic and the resulting path is not continuous, it is not suitable for real-time applications [8–10].

Although the above-mentioned path planning algorithms enjoy several advantages in solving general planning problems, they are still required to model the obstacles in a deterministic space and these constructed models are quite complex. Therefore, these algorithms cannot satisfy the needs of self-driving car path planning in a dynamically complex environment.

Rapidly exploring Random Tree (RRT) is a data structure algorithm whose unique advantages make it directly applicable to nonholonomic planning and motion planning. The algorithm takes the given initial point as the root node of the random tree and searches quickly and efficiently for a feasible space according to the current environment. Therefore, RRT algorithm is a randomized algorithm that can explore a large space in a relatively short time, which is fast and efficient for the path planning of the self-driving car [11–14].

This paper is structured as follows: The global path-planning algorithm is introduced in Section 2. The local obstacle avoidance algorithm is discussed in Section 3. Simulation results and discussion are presented in Section 4. Finally, the conclusion and future work are given in Section 5.

2. Global path-planning algorithm

In recent years, RRT algorithm has been widely studied and used in the field of self-driving car path planning. Since the random sampling strategy of the algorithm does not need to preprocess the state space and has fast velocity in the process of searching, it can effectively solve the problem of path planning in a complex environment. However, there are some defects:

1. The global uniform sampling strategy may increase unnecessary cost and decrease the convergence rate;
2. The randomness of the algorithm generates an unsmooth path that may not be directly executed by the non-holonomic constrained self-driving car.

In view of the defects of the classical RRT algorithm, some scholars have improved the algorithm. To improve the search efficiency, Kuffner and LaValle [15] proposed Bidirectional search tree (Bi-RRT) algorithm in which two trees were simultaneously generated at the initial point and target point to accelerate the convergence rate of the algorithm. In view of the unsmooth path generated by the randomness of the classical RRT algorithm, Fraichar and Scheuer [16] employed the method of a convolution curve to smooth the path. However, the convolution curve method does not have a closed-form solution, hence the path of the self-driving car cannot promptly and accurately be planned. Lau et al. [17] planned the path of a self-driving car using a Quintic B'ezier curve, but the curvature continuity of the path and constraints of the self-driving car were neglected. Amiryan and Jamzad [18] employed some randomized sampling methods such as the RRT or its derivatives to plan a prior path, aiming to overcoming the drawbacks of the Artificial Potential Field (APF) method which comprise local minima and oscillation. Qureshi et al. [19,20] introduced APF into the RRT* algorithm to accelerate the rate of convergence and significantly reduce the number of iterations compared with the classical RRT algorithm. Through combining the reachability RRT and resolution-complete RRT, Jaillet et al. [21] improved the success rate of the search, especially the narrow channel, and reduced the number of nodes in searching. However, the constraints of the self-driving car are ignored by these methods, which may lead to the planned path that does not meet the requirements of the car.

2.1. Improved RRT algorithm

To solve the strong randomness of the generating node of the classic RRT, the idea of gravitational force in APF [22] was introduced into the classic RRT algorithm (hereinafter referred to as A-RRT). The gravitational force guides the random trees to grow toward the direction of the target point, as shown in Figure 2.

The core idea of this improvement is to introduce the target gravitational function $G(n)$ to each node n in the path. Here, n represents the nth point of the x_{new} growing from the initial point x_{init}. Moreover, x_{rand} and x_{goal} denote the random node and target point, respectively, and ρ represents the search step length.

In the gravitational function $G_x = dU_x/d_x = k_p\,|x_{goal} - x_{near}|$, x is the current position vector of the self-driving car, k_p the coefficient of the gravitational

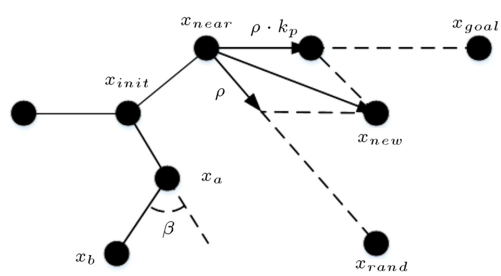


Figure 2. The growing process of nodes.

field, x_{near} the nearest point, and $|x_{goal} - x_{near}|$ the absolute value of the geometric distance between the node x_{near} and the target point x_{goal}.

According to the above-mentioned growing process of nodes, the suitable target gravitational function $G(n)$ for RRT algorithm can be constructed as follows:

$$G(n) = \rho \cdot k_p \cdot \frac{x_{\text{goal}} - x_{\text{near}}}{|x_{\text{goal}} - x_{\text{near}}|}. \tag{1}$$

If a new leaf node is added through the A-RRT algorithm, the target gravitational function will influence the selection of the new node by calculating the gravity from each node to the target and then, the random tree will be guided toward the target to grow [23].

According to Eq. (1), the equation of generating new nodes that introduced the idea of gravitation can be obtained as:

$$x_{\text{new}} = x_{\text{near}} + \rho\left\{\frac{x_{\text{rand}} - x_{\text{near}}}{|x_{\text{rand}} - x_{\text{near}}|} + k_p \times \frac{x_{\text{goal}} - x_{\text{near}}}{|x_{\text{goal}} - x_{\text{near}}|}\right\}. \tag{2}$$

2.2. Constraint condition

To make a planned path effectively applicable to the self-driving car, the path should be tracked so that it would meet the road environment constraints. It is assumed that B_l and B_r are the left and right boundaries of the road and that the nodes of the generated random tree should be within the boundaries. The coordinates of the node position should meet Eqs. (3) and (4):

$$B_r \le t_y \le B_l, \tag{3}$$

$$P_{ini} \le t_x \le P_{tar}, \tag{4}$$

where P_{ini} is the initial point of each extension and P_{tar} is the target point of each extension.

Given that a car has a geometric shape, the width of the car can be expressed as D. Since Eq. (3) is the restriction on the coordinates in the y direction of the nodes, we have:

$$B_r - D/2 \le t_y \le B_l + D/2. \tag{5}$$

Given that the center of the mass of the self-driving car moves along the planned path, the curvature of the planned path cannot change too much to ensure the stability while driving. If the actual front wheel steering angle has a maximal value, $\theta_{\max}$, the connection line between the child node x_b and its parent node x_a is $x_a x_b$; and the connection line between the parent node x_a and its parent node x_{init} is $x_{init}x_a$; further, the angle β between $x_a x_b$ and $x_{init}x_a$ must satisfy $\beta < \theta_{\max}$. Generally, the value of θ is between 30°~40°. Then, the angle constraint can be expressed as follows:

$$\arctan\left((K_1 - K_2)/(1 + K_1 K_2)\right) < \beta \le \theta_{\max}, \tag{6}$$

where K_1 is the curvature of the straight line $x_a x_b$, and K_2 is the curvature of the straight line $x_{init}x_a$.

To ensure that the extended points do not intersect with obstacles, the method for elliptically enveloping obstacle and properly enlarging the safe ellipse was adopted to meet the requirements of obstacle avoidance. If the connection line between the new node and its parent node does not intersect with the safe ellipse, the expanded new point would meet the requirement of the obstacle avoidance. In case the five-equal partition point $P(x, y)$ is placed on the connection line, the constraint equation can be expressed as:

$$(x - x_{obs})^2/(s \cdot a)^2 + (y - y_{obs})^2/(s \cdot b)^2 > 1, \tag{7}$$

where (x_{obs}, y_{obs}) are the coordinates of the obstacle and s is the safe ellipse magnification coefficient. In addition, the half-length and width of the car are $a =$ 2 m and $b = 1$ m, respectively. When s equals $\sqrt{2}$, the safe ellipse just right envelops the rectangular obstacle; therefore, it is necessary to guarantee $s \ge \sqrt{2}$ in terms of security obstacle avoidance.

2.3. The process of smoothing the path

The path planned by the classic RRT algorithm usually has a small range of twists and turns, thus being discontinuous. To make the path meet the stability and safety requirements of the car while moving, it is necessary to smooth the planned path. The B-spline can locally adjust the path without changing the entire shape of the path. According to the feature of the B-spline, the path planned by the classic RRT algorithm can fulfill the purpose of the smooth path using the interpolation technique. The commonly used B-spline is the cubic spline curve.

The proposed A-RRT algorithm does not need an accurate model of the global environment, which can greatly shorten the planning time and improve the real-time performance of the algorithm. Furthermore, the cubic B-spline curve is employed to smooth the path generated by the A-RRT algorithm, which can ensure the curvature continuity of the path and satisfy the constraints of the self-driving car.

3. Local obstacle avoidance using behavioral dynamics

The environmental information can be detected by the sensors of the self-driving car. In case of detecting a dynamic obstacle, the behavioral dynamics method is employed to plan the local obstacle avoidance path for the self-driving car and then, the self-driving car begins to avoid the obstacle.

According to the behavioral dynamics theory [24,25], the target and the obstacle can be represented as an attractor and a repeller, respectively. The attractor can generate virtual attraction between the target and current position of the self-driving car in the process of moving. The repeller can generate a virtual repulsive force between the obstacle and current position of the self-driving car in the process of moving. First, the relationship between virtual attraction and behavioral variables was established by making the target of the typical driving behavior an attractor. Second, the relationship between the virtual repulsive force and behavioral variables was structured by making the obstacle around the self-driving car a repeller. Finally, the behavioral dynamics model of self-driving car path planning was established by combining the attraction and repulsive force model.

According to the traffic rules, the solid line cannot be crossed by the vehicle and the dotted line can be driven over for a short time, but the car cannot ride on the line for a long time. The solid line is defined as a strong constraint, and the dotted line is defined as a weak constraint. The strong constraint must be avoided while driving the car. The weak constraint should not be avoided if the car needs to change lanes or overtake another car while driving; only when the car is driven in the mode of lane-keeping, for driving safety, must a safe distance be kept between the car and the lane.

3.1. Attraction model

According to the typical driving behavior of the self-driving car, the relationship between the attraction of tending to the target and the behavioral variables including heading angle and velocity can be established [26].

The behavioral variables of the self-driving car are illustrated in Figure 3.

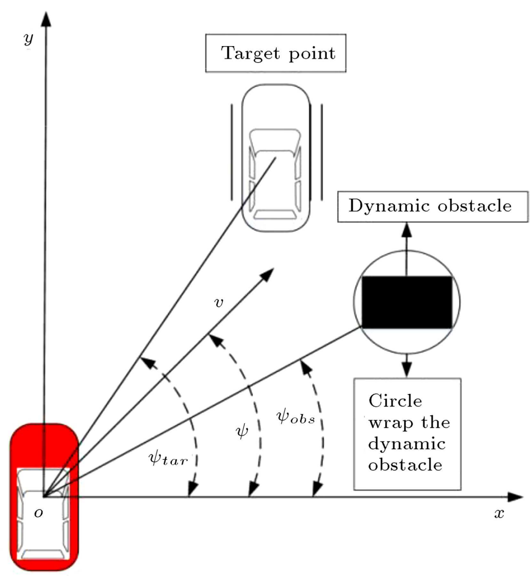


Figure 3. The behavioral variables of the self-driving car.

3.1.1. Heading angle attraction model

The heading angle attraction model is established by the typical driving behavior of the self-driving car. The heading angle ψ of the self-driving car must finally be consistent with the direction of the target. By assuming that the direction of the target is the attractor, it can be concluded that the heading angle of the self-driving car must satisfy $\psi_{tar} \in [-\pi/2, \pi/2]$. The heading angle attraction equation of tending to the target can be established according to Fu et al. [27], which is expressed as follows:

$$f_{tar} = f_{tar,\psi}(\psi) = -\lambda_{tar,\psi} \tan(\psi - \psi_{tar}), \quad (8)$$

where $\lambda_{tar,\psi}$ is the heading angle attraction strength factor, and the magnitude of the virtual attraction is changed by adjusting $\lambda_{tar,\psi}$. Moreover, ψ_{tar} is the heading angle between the target and the self-driving car in the world coordinate system:

$$\psi_{tar} = \arctan((P_{tar,y} - P_{veh,y})/(P_{tar,x} - P_{veh,x})).$$

In addition, P_{tar} ($P_{tar,x}$, $P_{tar,y}$) and P_{veh} ($P_{veh,x}$, $P_{veh,y}$) are the positions of both target point and the self-driving car in the world coordinate system, respectively.

Figure 4 shows the local coordinate system on the self-driving car. The x'-axis direction is the moving direction along the axis of the car, and the y'-axis direction is perpendicular to the axis direction of the car. Further, $\phi_{tar}{}'$ is the angle between the direction of vehicle velocity and x' coordinate, which is a behavioral variable.

To ensure the final velocity direction in line with the axis of the car and parking space, the specified parking space centerline is set as the target pose attractor.

The pose behavioral dynamics model [28] of the

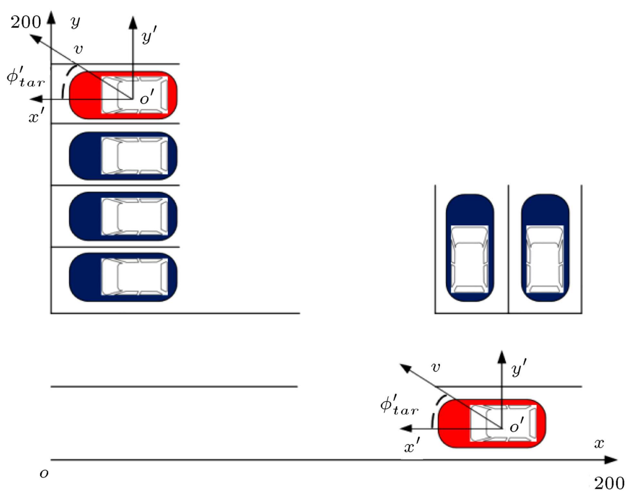


Figure 4. The coordinate system of the car.

self-driving car can be established as follows:

$$\dot{\phi} = -\lambda \left|\phi' - \phi'_{tar}\right|, \tag{9}$$

where λ is the factor of pose attraction strength.

3.1.2. Velocity attraction model

In the path planning process of the self-driving car, it is necessary to consider the safety and dynamic characteristics of the car while establishing the velocity attraction model. The velocity of the car must have an upper limit. The contact time can be expressed as $T = d_{tar}/v$, where d_{tar} is the distance between the self-driving car and the target point. The contact time T cannot be too long nor too short and must ensure the safe distance of the car. Here, $T_{\max}$ is the maximum contact time:

$$T_{\max} = (d_{tar} - D_s/v),$$

where D_s is the safe following distance. Furthermore, the acceleration of the self-driving car $\dot{v}$ must be lower than that of the target car $\dot{v}_{tar}$; otherwise, there will be a collision.

According to [29], the velocity attraction equation of tending to the target can be established as follows:

$$f_{tar,v}(v) = -\lambda_{tar,v}(v - v_{tar}) \exp\left[\frac{(v - v_{tar})^2}{2\sigma_v^2}\right], \tag{10}$$

where $\lambda_{tar,v}$ is the velocity attraction strength factor, and the size of the virtual attraction of the target point to the self-driving car can be changed by adjusting $\lambda_{tar,v}$. In addition, v_{tar} and D_{veh} are the expected velocity and width of the self-driving car, respectively, and σ_v is the range of the attractor, which can be expressed as:

$$\sigma_v = \arcsin\left(\frac{D_s + D_{veh}}{D_{veh}}\right), \tag{11}$$

when the target is ahead of the moving car, v_{tar} is the velocity of the target car. Otherwise, the self-driving car will drive at a constant velocity. The current velocity of the self-driving car v_c can be achieved according to the inertial navigation system. The distance S between the self-driving car and the target car can be measured by the millimeter wave radar, and the initial velocity v_o can be set.

The velocity of the target car v_{tar} can be calculated as follows:

$$v_{tar} = \sqrt{2aS + v_c^2}, \tag{12}$$

$$a = \frac{v_c - v_o}{T_{\max}}. \tag{13}$$

3.2. Repulsive force model

According to the surrounding environment of the self-driving car, the relationship between the repulsive force and behavioral variables including the heading angle and the velocity can be established.

3.2.1. Heading angle repulsive force model

If a static or moving obstacle is detected while moving toward the target point, the self-driving car must be able to safely avoid the obstacle and safely reach the target point. Here, ψ_{obs} represents the repeller as an unstable point that turns the influence of an obstacle to zero in the behavioral dynamics method.

According to [30], the repulsive force equation of the heading angle can be established as follows:

$$f_{obs,i}(\psi_{obs,i}) = -\lambda_{obs,i}(\psi - \psi_{obs}) \times \exp(-Cd_{obs,i})\exp\left(-\frac{(\psi - \psi_{obs,i})^2}{2\sigma_{obs,i}^2}\right), \tag{14}$$

$$\psi_{obs,i} = \arctan\left(\frac{P_{obs,i}y - P_{veh}y}{P_{obs,i}x - P_{veh}x}\right), \tag{15}$$

where $\lambda_{obs,i}$ represents the heading angle repulsive force strength factor, and the repulsive force can be changed by adjusting $\lambda_{obs,i}$; C is the repulsive coefficient of attenuation with increasing distance, $d_{obs,i}$ the distance between the obstacle and the self-driving car; and $P_{obs,i}x$ and $P_{obs,i}y$ are the coordinates of the obstacle in the world coordinate system. Further, $\sigma_{obs,i}$ represents the range of a repeller, which can be expressed as follows:

$$\sigma_{obs,i} = \arcsin\left(\frac{D_s + D_{veh}}{d_{obs,i} + D_{veh}}\right). \tag{16}$$

The heading angle repulsive force equation of multiple obstacles can be written as follows:

$$F_{obs} = \sum_i f_{obs,i}(\psi_{obs,i}). \tag{17}$$

3.2.2. Velocity repulsive force model

In the path planning process, the velocity of the self-driving car is not only related to the distance d_{obs} between the current position of the self-driving car and the obstacle but to the safe distance D_s. Under the premise of guaranteeing the minimal safe distance, the linear velocity of the self-driving car decreases by decreasing d_{obs}.

According to Han et al. [31], the linear velocity repulsive force equation can be established as:

$$f_{obs,i}(v) = -\lambda_{obs,v}(v - v_{obs,i}) \exp\left[\frac{(v - v_{obs,i})^2}{2\sigma_v^2}\right]. \tag{18}$$

3.3. Behavioral dynamics model

According to the above established attraction model and repulsive force model of the heading angle and velocity, the behavioral dynamics model of the self-driving car can be established by the weighting of each attraction and repulsive force. In practical applications, each behavior needs to be coordinated and then, be used for controlling the vehicle behavior.

Synthesized behavioral dynamics model that includes velocity and heading angle can be established as:

$$\dot{v} = f_v = w_{obs} f_{obs,i}(v) + w_{tar} f_{tar}(v), \tag{19}$$

$$\dot{\psi} = f_\psi = \gamma_{obs} F_{obs} + \gamma_{tar} f_{tar}, \tag{20}$$

where w_{obs}, w_{tar}, γ_{obs}, and γ_{tar} are the weight coefficients of the behavioral dynamics model. According to the weight of each behavior in the actual model, the force of each behavioral variable can be changed by altering the weight coefficient. The interference should be eliminated so that the target behavior and obstacle avoidance behavior can simultaneously occur; to this end, the four weight coefficients should differ depending on the situation in the simulation process.

4. Simulation and discussion

To illustrate the effect of the A-RRT algorithm, Figure 5 contrasts the planning results of the A* algorithm with those of the A-RRT algorithm. The initial parameters of the self-driving car are CarPos = [165, 20, 90, 5, 8, 90, 1] (initial x coordinate, initial y coordinate, heading angle, velocity, perceptive distance, perceptive angle, size of car), and the target position parameters are TargetPos = [25, 175, 90] (x and y coordinates of the target parking space). The obstacle position parameters are ObstaclePos = [90, 95]. The horizontal

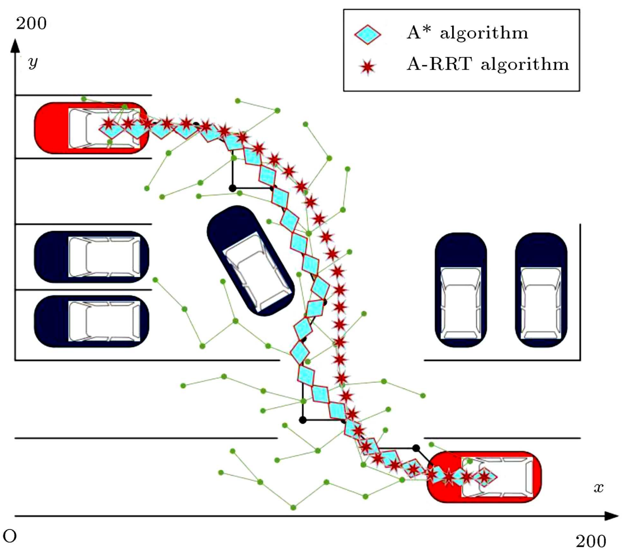


Figure 5. The chart of algorithm effect comparison.

Table 1. Comparison of simulation experimental data.

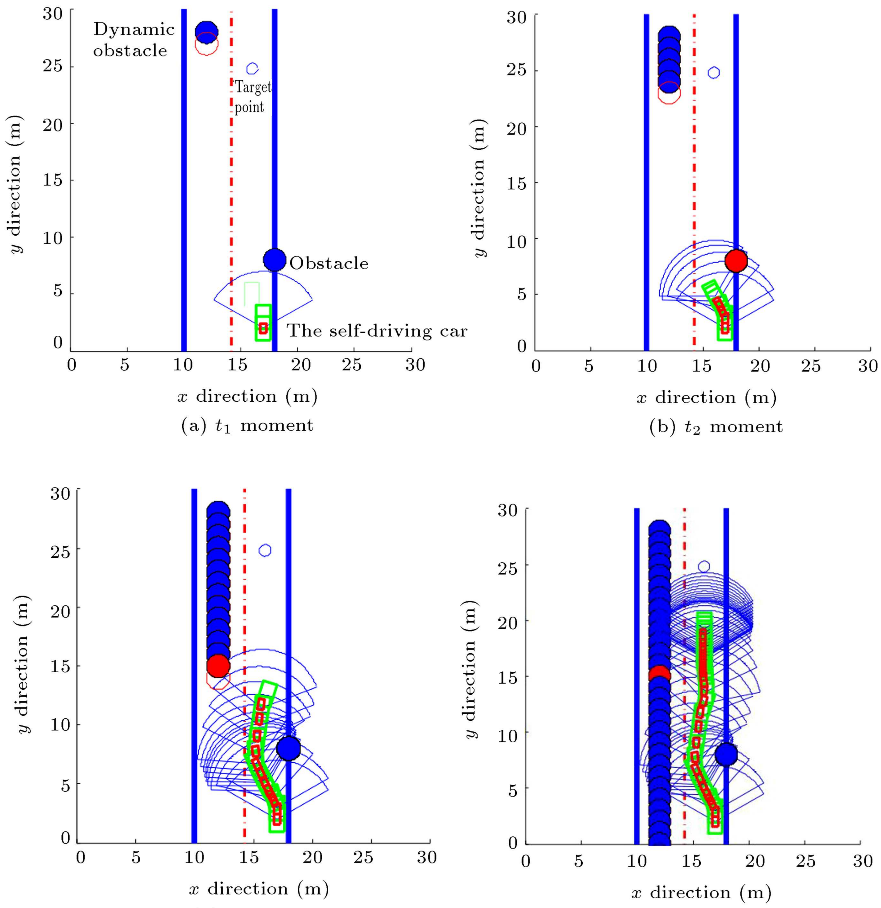
30 Experiments	A*	A-RRT
Planning time (s)	0.166	0.181
	0.189	0.237
	0.262	0.284
	0.381	0.464
	0.155	0.152
The length (m)	130.922	132.423
	131.246	132.680
	129.854	130.262
	130.573	133.025
	130.248	130.883
Successful times	30	30
Mean square deviation of the path curvature	0.138	0.094

length is 200 m. In addition, MATLAB 2014a is used hybrid for simulating the algorithm. A comparison of the results is presented in Figure 5 and Table 1.

The path of the A-RRT algorithm was smoother that of the A* algorithm, and there were no frequent considerable curvature changes. To objectively evaluate the performance of the algorithm, two algorithms were planned 30 times in the same experimental scene mainly because of the randomness of A-RRT algorithm. Table 1 presents the partial results of the 30 experiments, including five planning times, five path lengths, times of success, and mean square deviation of path curvature.

According to the results of the simulation experiments in Figure 5 and the data in Table 1, A^* algorithm outperforms A-RRT algorithm in planning time and path length; however, A-RRT algorithm enjoys some significant advantages in terms of path smoothing and meeting car constraints. Therefore, the A-RRT algorithm is more suitable for the self-driving car.

Figure 6 illustrates the process of obstacle avoidance. The self-driving car and the obstacles are both located in an initial position at t_1 moment. A millimeter wave radar and two ultrasonic radars are arranged on the head of the self-driving car. The detection range is expressed as a sector, and the angle of the sector is 120°. Both car and dynamic obstacle keep their own velocity vector before t_3 moment, and the car moves toward the target point at a speed of 5 km/h. When the dynamic obstacle is detected at t_3 moment, a red mark appears and then, the path of obstacle avoidance is planned using the behavioral dynamics method. The average time of all 20 times of obstacle avoidance path planning is 0.072 s, and



(a) t_1 moment

(b) t_2 moment

(c) t_3 moment

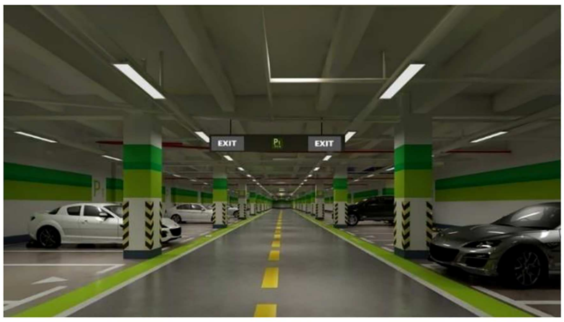
(d) t_4 moment

Figure 6. Obstacle avoidance using the behavioral dynamics method.

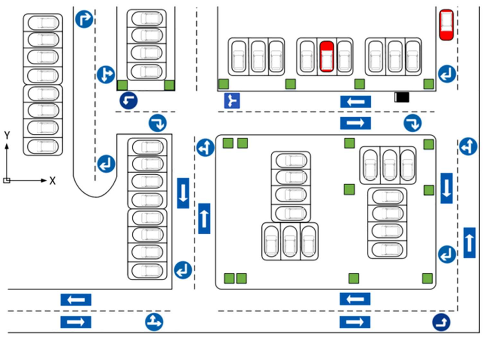
the planning time decreases while improving the computer configuration. Furthermore, the time it takes for a human from discovery to brain judgment to manipulation of the hands and feet is called reaction time, which is about 0.38 s. Hence, the behavioral dynamics obstacle avoidance meets the requirement of real time.

Monte Carlo method is a stochastic simulation method that uses random numbers or some kind of probability phenomenon to simulate the real-world problems. Since the self-driving car may randomly stop during parking to the designated parking space, this method can be employed to simulate the path between the random initial state and target state. As the path length is proportional to the navigation error of the goal point, the navigation path planning problem can be reduced to the shortest path problem from the initial state to the target state under the premise of the maximum allowable error. The process of safety navigation is shown in Figure 7.

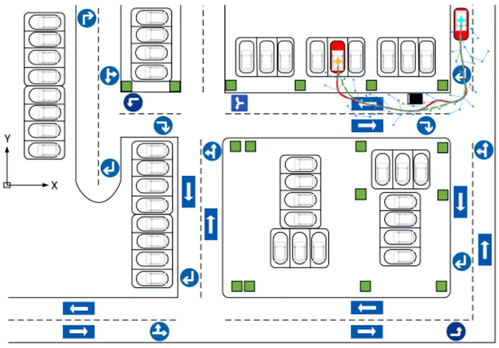
Figure 7(a) shows the environment of the underground parking lot of the Wal-Mart International Shopping Center. The area of the underground parking lot is 26,000 m^2 with 800 parking spaces. The size of the parking space is 5.5×2.4 m. Moreover, the width of the parking space line and the lane line are 9 cm, and the lane width is 6 m. Figure 7(b) shows the layout of the parking lot. Figure 7(c) presents the global and partial enlarged drawing of the planning path. The entrance coordinate of the parking lot is [885, 55], expressed in green, and can be used as the initial point. The designated parking space coordinate is [658, 110], expressed in yellow, and can be used as the target point. The global path can be planned using the A-RRT algorithm and then, the proposed obstacle detection algorithm is used to estimate the obstacle parameters when the dynamic obstacle is detected. The black rectangle represents the dynamic obstacle and the larger rectangle that wraps the dynamic obstacle represents the area of potential collision. According to the initial point coordinates and foregone values, the parameters for the dynamic obstacle can be calculated. The possible radius and velocity of the dynamic obstacle are in the range of 0.5–0.8 m and 3.5–4 m/s, respectively. The estimated parameters for the dynamic obstacle are transmitted to behavioral dynamics for obstacle avoidance, and the process of obstacle avoidance for the self-driving car



(a) The underground parking lot



(b) The layout of the underground parking lot



(c) The global and the partial enlarged diagram

Figure 7. The process of safety navigation

can be clearly observed in Figure 7(c); then, the self-driving car returns to the global path. Finally, it safely reaches the designated parking space.

5. Conclusion

For the typical self-driving scene of the parking lot, an efficient trajectory planning framework for the self-driving car was presented in this study. In this respect, APF was introduced into the classical RRT algorithm to accelerate the convergence speed and obtain an optimal solution. The convergence rate enhanced by almost four times, the smoothness and curvature continuity of the path were greatly improved, as shown in Table 1, and the improved algorithm only required 68% of the original iterations to find a solution. Moreover, constraints of the road and self-driving car were considered during the expansion of the nodes, making the planning path meet the self-driving car requirements. The behavioral dynamics method was employed to plan an obstacle avoidance path based on the dynamic obstacle parameters and the average time of replanning the path was 0.072 s, thus meeting the real-time requirement. The experimental results from MATLAB indicated that the hybrid path planning model showed a good real-time performance and reliability, and the self-driving car could safely bypass the obstacle along the desired path and reach the target point. Future related works should apply the proposed method to a real self-driving car to perform real-life tests and further performance measurement.

Acknowledgments

This work is supported by Project (10872160) of the National Natural Science Foundation of China, Shaanxi Province Important Disciplines of the Vehicle Engineering Construction of China, Scientific Research Projects of Shaanxi Province Education Office Key Laboratory (13JS070), and Scientific Research Project in Shaanxi Province Department of Education (14jk1796).

References

1. Li, A.J., Li, S.M., and Li, D.R. "On the trajectory planning's key technologies for intelligent vehicle", *Mechanical Science and Technology for Aerospace Engineering*, **32**(7), pp. 1022–1026 (2013).

2. Domokos, K. and Gábor, T. "Autonomous path planning for road vehicles in narrow environments: An efficient continuous curvature approach", *Journal of Advanced Transportation*, **2017**(2), pp. 1–28 (2017).

3. Kim, J.M., Lim, K.I., and Kim, J.H. "Auto parking path planning system using modified Reeds-Shepp curve algorithm", *2014 11th International Conference on Ubiquitous Robots and Ambient Intelligence*, pp. 311–315 (2014).

4. Wang, G.Q., Tsuneo, N., and Akira, F. "Time-varying shortest path algorithm with transit time tuning for parking lot navigation", *TENCON 2015-2015 IEEE Region 10 Conference*, pp. 1–6 (2015).

5. Du, M.B., Mei, T., and Chen, J.J. "RRT-based motion planning algorithm for intelligent vehicle in complex environments", *ROBOT*, **37**(4), pp. 443–450 (2015).

6. Dolgov, D., Thrun, S., Montemerlo, M., et al. "Path planning for autonomous vehicles in unknown semi-structured environments", *The International Journal of Robotics Research*, **29**(5), pp. 485–501 (2010).

7. Elbanhawi, M. and Simic, M. "Sampling-based robot motion planning: A review", *IEEE Access*, **2**(1), pp. 56–77 (2014).

8. Likhachev, M. and Ferguson, D. "Planning long dynamically-feasible maneuvers for autonomous vehicles", *International Journal of Robotics Research (IJRR)*, **28**(8), pp. 933–945 (2009).

9. Kushleyev, A. and Likhachev, M. "Time-bounded lattice for efficient planning in dynamic environments", *IEEE International Conference on Robotics and Automation*, pp. 1662–1668 (2009).

10. Guo, Q., Zhang, Z., and Xu, Y. "Path-planning of automated guided vehicle based on improved Dijkstra algorithm", *29th Chinese Control and Decision Conference (CCDC)* (2017).

11. Dong, Y. and Camci, E. "Faster RRT-based nonholonomic path planning in 2D building environments using skeleton-constrained path biasing", *Journal of intelligent and Robotic Systems*, **89**(3-4), pp. 387–401 (2018).

12. Otte, M. and Frazzoli, E. "RRTX: Asymptotically optimal single-query sampling-based motion planning with quick re-planning", *International Journal of Robotics Research*, **35**(7), pp. 1–35 (2015).

13. Song, X.L., Zhou, N., Huang, Z.Y., et al. "An improved RRT algorithm of local path planning for vehicle collision avoidance", *Journal of Hunan University (Natural Science)*, **44**(4), pp. 30–37 (2017).

14. Jeon, J.H., Cowlagi, R.V., Peters, S.C., et al. "Optimal motion planning with the half-car dynamical model for autonomous high-speed driving", *American Control Conference*, pp. 188–193 (2013).

15. Kuffner, J.J. and LaValle, S.M. "RRT-connect: An efficient approach to single-query path planning", *IEEE International Conference on Robotics & Automation*, pp. 995–1001 (2002).

16. Fraichard, T. and Scheuer, A. "From Reeds and Shepp's to continuous curvature paths", *IEEE Transactions on Robotics*, **20**(6), pp. 1025–1035 (2004).

17. Lau, B., Sprunk, C., and Burgard, W. "Kino dynamic motion planning for mobile robots using splines", *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2427–2433 (2009).

18. Amiryan, J. and Jamzad, M. "Adaptive motion planning with artificial potential fields using a prior path", *The 3rd RSI International Conference on Robotics and Mechatronics*, pp. 731–736 (2015).

19. Qureshi, A.H., Iqbal, K.F., Qamar, S.M., et al. "Potential guided directional-RRT* for accelerated motion planning in cluttered environments", *IEEE International Conference on Mechatronics and Automation*, pp. 519–524 (2013).

20. Qureshi, A.H., Mumtaz, S., and Iqbal, K.F. "Adaptive potential guided directional-RRT*", *The IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 1887–1892 (2013).

21. Jaillet, L., Hoffman, J., and Berg, J.V.D. "EG-RRT: Environment-guided random trees for kinodynamic motion planning with uncertainty and obstacles", *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2646–2652 (2011).

22. Sun, L.B., Liu, Y., Sun, J.Z., et al. "Path planning model based on mixed perception information", *Computer Engineering*, **36**(10), pp. 32–35 (2010).

23. Lv, W.X., Zhao, L.J., Wang, K., et al. "Efficient exploration of unknown environments with RRT-boundary constraint", *Hua Zhong Univ. of Sci. and Tech. (Natura Science Edition)*, **39**, pp. 366–369 (2011).

24. Y, S.Q., Fu, W.P., and Li, D.X. "Application of dynamic system theory to mobile robot navigation", *Mechanical Science and Technology for Aerospace Engineering*, **29**, pp. 100–104 (2010).

25. Y, S.Q., Fu, W.P., Li, D.X., et al. "Research on application of genetic algorithm for intelligent mobile robot navigation based on dynamic approach", *IEEE International Conference on Automation and Logistics*, pp. 898–902 (2007).

26. Bicho, E., Mallet, P., and Schoner, G. "Using attractor dynamics to control autonomous vehicle motion", *Proceedings of the 24th Annual Conference of the IEEE Industrial Electronics Societh*, pp. 1176–1181 (1998).

27. Fu, W.P., Zhang, P.F., and Yang, S.Q. "Behavioral dynamics of mobile robot and rolling windows algorithm to path planning", *Computer Engineering and Applications*, **45**, pp. 212–214 (2009).

28. Wang, W.Y., Fu, W.P., Wei, M.M., et al. "Behavior dynamics method for the motion planning of the end-effector of autonomous manipulator", *Journal of Xi'an University of Technology*, **32**(4), pp. 468–474 (2016).

29. Fu, W.P., Hao, D.P., Yang, S.Q., et al. "Study on the navigation method of behavior dynamics in mobile robot", *Mechanical Science and Technology for Aerospace Engineering*, **32**(10), pp. 1488–1491 (2013).

30. Han, G.N., Fu, W.P., Hao, D.P., et al. "Study on the motion planning method of intelligent vehicle based on the behavior dynamics", *Mechanical Science and Technology for Aerospace Engineering*, **34**(2), pp. 301–306 (2015).

31. Han, G.N., Fu, W.P., and Wang, W. "The study of intelligent vehicle navigation path based on behavior coordination of particle swarm", *Computational Intelligence and Neuroscience*, **2016**, pp. 1–10 (2016).

Biographies

Jin-Kai Yin is a PhD student in Mechanical and Electronic Engineering at XI'AN University of Technology. He holds his MSc degree in Vehicle Engineering from HENAN University of Science and Technology. His research interests include path planning and decision making of intelligent vehicles.

Wei-Ping Fu is a Professor and a PhD Supervisor at XI'AN University of Technology. He received his BSc and MSc degrees from SHAANXI Institute of Machinery and PhD degree in the same field from XI'AN JIAOTONG UNIVERSITY. His research interests include intelligent vehicle and intelligent robot.