Research Article

Path Planning for Autonomous Articulated Vehicle Based on Improved Goal-Directed Rapid-Exploring Random Tree

Tong Xu ¹, Yang Xu,¹ Dong Wang ¹, Siwei Chen,¹ Weigong Zhang ¹
and Lihang Feng ^{1,2}

¹School of Instrument Science and Engineering, Southeast University, Nanjing 210096, China

²College of Electrical Engineering and Control Science, Nanjing Tech University, Nanjing 210096, China

Correspondence should be addressed to Dong Wang; kingeast16@seu.edu.cn and Weigong Zhang; weig_zhang88@126.com

Received 11 February 2020; Accepted 7 April 2020; Published 20 May 2020

Academic Editor: Oscar Reinoso

Copyright © 2020 Tong Xu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The special steering characteristics and task complexity of autonomous articulated vehicle (AAV) make it often require multiple forward and backward movements during autonomous driving. In this paper, we present a simple yet effective method, named head correction with fixed wheel position (HC-FWP), for the demand of multiple forward and backward movements. The goal-directed rapid-exploring random tree (GDRRT) algorithm is first used to search for a feasible path in the obstacle map, and then, the farthest node search (FNS) algorithm is applied to obtain a series of key nodes, on which HC-FWP is used to correct AAV heading angles. Simulation experiments with Dynapac CC6200 articulated road roller parameters show that the proposed improved goal-directed rapid-exploring random tree (IGDRRT), consisting of GDRRT, FNS, and HC-FWP, can search a feasible path on maps that require the AAV to move back and forth.

1. Introduction

Autonomous articulated vehicle (AAV) path planning is usually divided into two stages: global static path planning [1–4] and local dynamic path planning [5–8]. Global static indicates that the map is known and has no moving obstacles, and a feasible path without collisions can be designed at this stage. Although the vehicle may encounter moving obstacles while following the planned path, AAV will recognize moving obstacles through various types of sensors and stop to maintain a safe distance. Therefore, the core of the AAV path plan is to design a global static path that meets vehicle kinematics and dynamic constraints.

Many graph-based search algorithms have been proposed for global static path planning. Artificial potential field, one type of graph-based search algorithms, treats each obstacle as a specific model, leading to high time complexity and poor real-time performance in a multiobstacle environment [9]. A* and D*, the other type of graph-based search algorithms, are computationally expensive because the completeness and optimality are given priority [10]. The two types of algorithms mentioned above fail to address

vehicle motion constraints, so new methods must be applied to AAV path planning.

The random sampling strategy adopted by the rapidly-exploring random tree (RRT) has a wide search range, fast search speed and high computational efficiency, and there is no need to model obstacles during path planning. Therefore, it is often applied to path planning for multiple obstacle maps. However, the algorithm has high path cost and slow convergence. The goal-directed method proposed by Kuffner and LaValle [11] makes RRT branches move towards the target position with a certain probability, which greatly reduces its convergence speed. RRT* algorithm [12] reduces the path cost by searching for newly generated nodes and adjacent nodes. Based on [12], Nasir et al. [13] use triangle inequality for intelligent sampling and optimize the path through beacon nodes. In response to vehicle motion constraints, [14] designs a variety of kinematic constraint models, including cars, flat panel, and tricycle, and uses the proposed improved RRT algorithm for simulation verification. Du et al. [15] improves RRT cost function based on vehicle kinematic constraints and uses the pruning function and B-spline algorithm for subsequent processing. These

FIGURE 1: A possible experimental scenario near the construction area.

documents address vehicle motion constraints but still fail to meet the demand of multiple forward and backward movements.

The motivation for this paper comes from unmanned road construction. The trailer transports and drops the articulated road roller, a kind of AAV, near its construction area and where the roller is placed determines its starting state. To prepare the road construction, the end state of AAV generally locates on the extension of the first predetermined path. Figure 1 shows a possible experimental scenario near the construction area.

Compared to the above-related work, the present paper addresses three main contributions that are given as follows:

- (1) First, farthest node search (FNS) is proposed to reduce the computational complexity of GDRRT and select key nodes for subsequent processes.
- (2) Second, head correction with fixed wheel position (HC-FWP) is proposed for the demand of multiple forward and backward movements.
- (3) Third, IGDRRT, consisting of GDRRT, FNS, and HC-FWP, can search a feasible path on maps that require AAV to move back and forth globally.

The rest of this paper is organized as follows. Section 2 defines the AAV model. Section 3 introduces GDRRT and the proposed IGDRRT in detail. The experimental results are given in Section 4, and discussion and conclusion are summarized in Section 5.

2. Autonomous Articulated Vehicle Model

The autonomous articulated vehicle (AAV) model can be simplified with two wheels and an articulated frame, as

FIGURE 2: Autonomous articulated vehicle model.

shown in Figure 2, and kinematic equation is expressed as [16, 17]

$$\begin{bmatrix} \dot{x}_f \\ \dot{y}_f \\ \dot{\theta}_f \\ \dot{v}_f \\ \dot{\gamma} \end{bmatrix} = \begin{bmatrix} v_f \cos \theta_f \\ v_f \sin \theta_f \\ (v_f \sin \gamma + l_r \dot{\gamma}) / (l_f \cos \gamma + l_r) \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} u_0 + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u_1, \quad (1)$$

where x_f and y_f are the horizontal and vertical coordinates of the midpoint of the front axis, θ_f , $\dot{v}_f$, and $\dot{\gamma}$ are the heading angle, AAV speed, and the articulated steering angle of the front axis, respectively, and derivatives represent changes in AAV motion over time. u_0 and u_1 are the acceleration and the rate of the change of the hinged steering angle.

FIGURE 3: (a) Undirected graph network, where V1, V3, V4, V8, V9, and V12 are the preserved nodes determined by the GDRRT algorithm. (b) Adjacency matrix with preserved nodes updated by the Dijkstra algorithm.

FIGURE 4: (a) Searched key nodes from the end node to the starting node by FNS. (b) The process of FNS with ① to ⑬, where ①, ④, ⑦, ⑩, and ⑬ are the process of key node confirmation.

3. Improved Goal-Directed RRT

3.1. Goal-Directed RRT. Goal-directed RRT, presented in [11], is a single query tree structure algorithm that moves towards the target position with a certain probability. The algorithm defines a directed graph $G = \langle V, E \rangle$ containing the vertex set V and edge set E . The vertex set stores the

searched node position, and the edge set stores the index relationship between the vertices. Algorithm 1 shows that q_{rand} moves towards the target position with a certain probability. In function **FindQNear**, Euclidean distance is selected as the cost function to pick the least cost $q_{near} \in Q_{near}$ from V , and in function **FindQNew**, linear interpolation is adopted to detect obstacles between q_{near}

```

Input:  $K \in \mathbb{N}, q_{init} \in C_{free}, q_{goal} \in Q_{goal} \in C_{free}$ 
Output:  $G$ 
 $V \leftarrow \{q_{init}\}, E \leftarrow \{\phi\}, G \leftarrow (V, E)$ 
for  $k \leftarrow 1$  to  $K$  do
  if  $rand < p1$ , then
     $q_{rand} \leftarrow q_{goal}$ 
  else
     $q_{rand} \leftarrow \text{RANDOMSTATE}$ 
     $q_{near} \leftarrow \text{FindQNear}(q_{rand}, V)$ 
     $(q_{new}, \text{success}) \leftarrow \text{FindQNew}(q_{near}, q_{rand})$ 
  if  $\text{success}$ , then
     $V \leftarrow V \cup \{q_{new}\}, E \leftarrow E \cup \{q_{near}, q_{new}\}$ 
    if  $q_{new} == q_{goal}$  or  $q_{goal}$  is on the edge then
      if  $q_{new} \sim q_{goal}$ , then
         $V \leftarrow V \setminus \{q_{new}\}, V \leftarrow V \cup \{q_{goal}\}$ 
       $\text{path} \leftarrow \text{TRUE}$ 
       $G \leftarrow (V, E)$ 
  Return  $G$  if  $\text{path} == \text{TRUE}$ ; else failure

```

ALGORITHM 1: Goal-directed RRT (GDRRT).

```

Input:  $\theta_H, \theta_T, \theta_{maxerror}, G$ 
Output:  $G$ 
 $V \leftarrow G.V, E \leftarrow G.E$ 
 $\Delta\theta \leftarrow \theta_H - \theta_T$ 
if  $|\Delta\theta| \leq \theta_{th}$  and  $|\Delta\theta| > \theta_{maxerror}$ , then
   $\text{Times2} \leftarrow \text{DeltatoTimes1}(\Delta\theta)$ 
   $V \leftarrow \text{FowardBackward}(V, \Delta\theta, \text{Times2})$ 
   $E \leftarrow E \cup \{q_{end}, q_{new}\}$ 
else if  $|\Delta\theta| \leq \pi/2$  and  $|\Delta\theta| > \theta_{th}$ , then
  do  $V \leftarrow \text{FowardBackward}(V, \Delta\theta, \text{Times1})$  and
   $E \leftarrow E \cup \{q_{end}, q_{new}\}$  until  $|\Delta\theta| \leq \theta_{th}$ ,
  then  $\text{Times2} \leftarrow \text{DeltatoTimes1}(\Delta\theta)$ 
   $V \leftarrow \text{FowardBackward}(V, \Delta\theta, \text{Times2})$ 
   $E \leftarrow E \cup \{q_{end}, q_{new}\}$ 
else if  $|\Delta\theta| \leq \pi - \theta_{th}$  and  $|\Delta\theta| > \pi/2$ , then
  do  $V \leftarrow \text{BackwardFoward}(V, \Delta\theta, \text{Times1})$  and
   $E \leftarrow E \cup \{q_{end}, q_{new}\}$  until  $|\Delta\theta| \leq \theta_{th}$ ,
  then  $\text{Times2} \leftarrow \text{DeltatoTimes2}(\Delta\theta)$ 
   $V \leftarrow \text{BackwardFoward}(V, \Delta\theta, \text{Times2})$ 
   $E \leftarrow E \cup \{q_{end}, q_{new}\}$ 
else if  $|\Delta\theta| \leq \pi - \theta_{maxerror}$  and  $|\Delta\theta| > \pi - \theta_{th}$ , then
   $\text{Times2} \leftarrow \text{DeltatoTimes2}(\Delta\theta)$ 
   $V \leftarrow \text{BackwardFoward}(V, \Delta\theta, \text{Times2})$ 
   $E \leftarrow E \cup \{q_{end}, q_{new}\}$ 
 $G \leftarrow (V, E)$ 
Return  $G$ 

```

ALGORITHM 2: Head correction with fixed wheel position.

and q_{rand} . If the new node q_{new} is found, $\{q_{new}\}$ and $\{q_{near}, q_{new}\}$ are added to V and E , respectively, and q_{new} is added as a subset of Q_{near} in the next cycle.

3.2. Improved GDRRT

3.2.1. Farthest Node Search. As shown in Figure 3(a), the search path is $V1 \rightarrow V3 \rightarrow V4 \rightarrow V8 \rightarrow V9 \rightarrow V12$

with the GDRRT algorithm. The numerical values next to the line segment represents GDRRT function cost, 3(a). Figure 3(b) shows the adjacency matrix calculated by the Dijkstra algorithm, and its time complexity is $T(n) = O(n^2)$.

To reduce the computational complexity, farthest node search (FNS) is proposed in this paper. Unlike the Dijkstra algorithm, FNS first connects the end node to postorder nodes in turn until the cost reaches infinity, and then the preorder

FIGURE 5: AAV position changes from (a) state 1 to 2, (b) state 2 to 3, (c) state 3 to 4, and (d) display of HC-FFWP.

node of the node with infinite cost, we call it the farthest node, is set as the key node; next, we view the key node as the end node and find the next key node until the starting node is reached, as shown in Figure 4(a), and FNS time complexity is $T(n) = O(n)$. Figure 4(b) details the process of FNS with ① to ⑬, where ①, ④, ⑦, ⑪, and ⑬ are the process of key nodes confirmation.

3.2.2. Head Correction with Fixed Wheel Position. In general, the key nodes cause the designed path to fail to reach C^2 continuity, which makes it impossible for AAV to directly follow the planned path by the algorithm mentioned in Section 3.2.1. In view of this problem, this paper proposes an algorithm called head correction with fixed wheel position (HC-FWP), which includes head

FIGURE 6: AAV position changes from (a) state 1 to 2, (b) state 2 to 3, (c) state 3 to 4, and (d) display of HC-FRWP.

FIGURE 7: Experimental scenario near the construction area.

TABLE 1: Dynapac CC6200 articulated road roller parameter settings.

Simulation parameters	Value
Compaction width of the steel wheel B/m	2.13
Distance from hinge point to front axle midpoint l_f/m	1.845
Distance from hinge point to rear axle midpoint l_r/m	1.845
Maximum speed $v_f/(m \cdot s^{-2})$	2
Sampling time $\Delta t/s$	0.1
Maximum steering angle $\gamma_{\max}/(^{\circ})$	14
Maximum acceleration $u_0/(m \cdot s^{-2})$	1
The gradient of articulated steering angle $u_1/(^{\circ} \cdot s^{-1})$	10

TABLE 2: IGDRRT parameter settings.

Simulation parameters	Value
Search step $\Delta q/m$	5
AAV maximum curvature/ m^{-1}	0.0661
Stepsize1	20
$\theta_{th}/(^{\circ})$	15
$\theta_{\max error}/(^{\circ})$	1

IGDRRT (Improved Goal Directed Rapidly-Exploring Random Trees)

FIGURE 8: GDRRT with farthest node search.

FIGURE 9: One-time HC-FFWP implementation.

TABLE 3: **DeltatoTimes1** mapping matrix.

$ \Delta\theta /(^{\circ})$	1	2	3	4	5	6	7	8
Times	1	3	4	5	7	8	9	11
$ \Delta\theta /(^{\circ})$	9	10	11	12	13	14	15	
Times	12	14	15	16	17	18	20	

TABLE 4: **DeltatoTimes2** mapping matrix.

$ \Delta\theta /(^{\circ})$	179	178	177	176	175
Times	1	3	4	5	7
$ \Delta\theta /(^{\circ})$	174	173	172	171	170
Times	8	9	11	12	14
$ \Delta\theta /(^{\circ})$	169	168	167	166	165
Times	15	16	17	18	20

FIGURE 10: Continued.

FIGURE 10: Continued.

FIGURE 10: IGDRRT with goal-directed probability of 0.01(a) (b), 0.1(c) (d), 0.2(e) (f), and 0.3(g) (h), respectively.

TABLE 5: Algorithm processing time with 95% confidence interval under 20 different goal-directed probabilities.

Goal-directed probability	Number of case (a)/(c)/(e)/(g)	95% confidence interval for case (a)/(c)/(e)/(g)	Number of case (b)/(d)/(f)/(h)	95% confidence interval for case (b)/(d)/(f)/(h)
0.01 (a) (b)	10	(19.0572s, 23.8877s)	10	(19.8248s, 28.1721s)
0.1 (c) (d)	12	(21.0483s, 26.0185s)	8	(17.0006s, 26.0764s)
0.2 (e) (f)	16	(18.3258s, 24.2589s)	4	(24.5175s, 32.4981s)
0.3 (g) (h)	16	(19.6846s, 26.7702s)	4	(17.9538s, 23.5815s)

correction with fixed front wheel position (HC-FFWP) and head correction with fixed rear wheel position (HC-FRWP). Take HC-FFWP as an example, when the front wheel arrives at a certain key node, we make the head angle gradually approach the target angle at each key node. Figures 5 and 6 display the process of HC-FFWP and HC-FRWP, expressed by **ForwardBackward** and **BackwardForward** function in simplified pseudocode, respectively, and the pseudocode is shown in Algorithm 2.

4. Experiments

4.1. The Experimental Scenarios and Parameter Settings. The $100\text{ m} \times 100\text{ m}$ experimental scenario near the construction area is used to verify the effectiveness of the proposed algorithm, as shown in Figures 7(a)–7(d), where gray areas indicate obstacles, blue stars and circles indicate AAV front and rear wheels at the starting position, and the black star and circle indicate AAV front and rear wheels at the end position, respectively.

The Dynapac CC6200 articulated road roller, a type of AAV, is selected for simulation experiment, and its parameters are set in Table 1. Table 2 shows IGDRRT parameter settings.

4.2. Experimental Results

4.2.1. Farthest Node Search. GDRRT regards AAV as a single point for path search. As shown in Figure 8, this algorithm searches for the path in the fixed-size step from the starting point marked by the dark blue star, the path and single-point position are marked with blue line segments and cyan stars, respectively, and the final path is shown in red. The path optimized by FNS is drawn as a black line. Compared with the original path generated by the traditional GDRRT, the path length is shortened and the number of key nodes is also reduced, which reduces the operation of subsequent algorithms.

4.2.2. Head Correction with Fixed Front Wheel Position. Figure 9 shows one-time head correction with fixed front wheel position implementation, where the dark blue and red rhombus indicate AAV front and rear wheel position, respectively, the black curve and the cyan curve are AAV trajectories. From AB to EF, the coordinates of AAV front wheel hardly change, but the heading changes to some extent.

The DeltatoTimes mapping matrix described in Algorithm 2 is designed depending on multiple experiments at a fixed-size sampling time and AAV speed, the mapping

FIGURE 11: The comparison between (a) HC-FFWP and (b) HC-FWP.

matrices, DeltatoTimes1 and DeltatoTimes2 , describe the relationship between the articulated steering angle and times, depicted in Tables 3 and 4, respectively.

4.2.3. IGDRRT with HC-FFWP under Different Goal-Directed Probabilities. In the existing documents, goal-directed probability is generally set between 0.01 and 0.3. In order to explore the impact of probability values on the proposed algorithm, we choose four different goal-directed probability values of 0.01, 0.1, 0.2, and 0.3, respectively. Figure 10 shows the experiment results intuitively, and Table 5 indicates that the change in goal-directed probability parameter does not have a great impact on algorithm processing time.

4.2.4. Optimization of Head Correction. Although the designed algorithm has acceptable processing time, the head correction strategy of some key nodes can still be optimized, as shown in Figure 10(h); the body of AAV turns almost 180° in the starting node, which is not in line with general driving behavior. Since AAV uses the same steering strategy when moving forward and backward, it may be a better way to use

both the front and the rear wheel to correct the course. Figure 11(a) shows the heading correction for fixed front wheel only, and Figure 11(b) shows the heading correction using both the front and the rear wheel, and experiments intuitively show that the method used in the latter experiments can be significantly improved in some cases.

4.2.5. IGDRRT with HC-FWP Using the Experimental Scenario near the Construction Area. The experimental scenario near the construction area, shown in Figure 7, is designed for simulation. As shown in Figure 12, the UAV final state of the following four cases is all set to $[10, 90]$ with 90° heading angle, and the AAV initial state is $[25, 20]$ with 135° heading angle (a), $[50, 20]$ with 90° heading angle (b), $[80, 20]$ with 45° heading angle (c), and $[80, 80]$ with 180° heading angle (d), respectively.

Table 6 shows processing time of the four algorithms with 95% confidence interval under 20 experiments. Due to the single-point model of the traditional GDRRT algorithm, AAV cannot search for a feasible path. The time complexity of GDRRT with HC-FWP using the Dijkstra algorithm consumes relatively much time. Although the algorithm mentioned in [16] and the algorithm proposed in this paper

FIGURE 12: (a) AAV final state set to $[10, 90]$ with 90° heading angle and AAV initial state set to $[25, 20]$ with 135° heading angle, (b) $[50, 20]$ with 90° heading angle, (c) $[80, 20]$ with 45° heading angle, and (d) $[80, 80]$ with 180° heading angle, respectively.

TABLE 6: 95% confidence interval for four algorithms.

Algorithms	Initial states in Figure 7	Algorithm processing time
Traditional GDRRT algorithm	(a)–(d)	Unreachable
GDRRT with HC-FWP using Dijkstra algorithm	(a)	(4.5762s, 5.8423s)
	(b)	(6.3485s, 7.4384s)
	(c)	(6.9432s, 8.1357s)
	(d)	(4.9132s, 6.0157s)
Document [16]	(a)	(1.4571s, 1.5831s)
	(b)	(2.3578s, 2.5439s)
	(c)	(2.7936s, 2.9128s)
	(d)	(1.6257s, 1.6527s)
The proposed algorithm	(a)	(1.2559s, 1.7180s)
	(b)	(2.1972s, 3.0762s)
	(c)	(2.4013s, 3.1109s)
	(d)	(1.5985s, 1.7127s)

take almost the same time, the algorithm presented in [16] requires a few minutes of preprocessing under each different initial state to determine the model parameters. Therefore, the algorithm proposed in this paper is significantly less time consuming than the other three algorithms using 95% confidence interval.

5. Discussion and Conclusion

Most traditional path planning algorithms do not consider C^2 continuity, so AAV cannot track the path planned by these algorithms. In recent years, some path planning algorithms for the car model are proposed, but the special steering characteristics and task complexity of AAV make it often require multiple forward and backward movements, and the algorithm designed for the car model fails to meet the demand. In regard to this, we present an algorithm called IGDRRT, which introduces the FNS optimization algorithm with $T(n) = O(n)$ time complexity on the basis of GDRRT, changes the single-point model of GDRRT to the two-point model, and incorporates the HC-FWP algorithm to meet the needs of forward and backward movements. Simulation results show that the proposed IGDRRT can search a feasible path on maps that require AAV to move back and forth.

Other application scenarios of the presented algorithm including articulated shovel trucks for earthquake rescue and articulated mining trucks for transport.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was funded by the Natural Science Foundation of Jiangsu Province, China, under grant numbers BK20170681 and BK20180701 and China Postdoctoral Science Foundation under grant number 2018M632292.

References

- [1] N. Wan, D. Xu, and H. Ye, "Improved cubic b-spline curve method for path optimization of manipulator obstacle avoidance," in *Proceedings of the 2018 Chinese Automation Congress*, pp. 1471–1476, Xi'an, China, November 2018.
- [2] K. G. Jolly, R. Sreerama Kumar, and R. Vijayakumar, "A Bezier curve based path planning in a multi-agent robot soccer system without violating the acceleration limits," *Robotics and Autonomous Systems*, vol. 57, no. 1, pp. 23–33, 2009.
- [3] S. Karaman and E. Frazzoli, "Incremental sampling-based algorithms for optimal motion planning," *Robotics: Science and Systems*, vol. 6, pp. 267–274, 2011.
- [4] H. Wang, J. Chen, H. Y. K. Lau, and H. Ren, "Motion planning based on learning from demonstration for multiple-segment flexible soft robots actuated by electroactive polymers," *IEEE Robotics and Automation Letters*, vol. 1, no. 1, pp. 391–398, 2016.
- [5] J. D. Gammell, S. S. Srinivasa, and T. D. Barfoot, "Informed RRT*: optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2997–3004, Chicago, IL, USA, September 2014.
- [6] H. Vorobieva, S. Glaser, N. Minoisu-Enache, and S. Mammar, "Automatic parallel parking in tiny spots: path planning and control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 1, pp. 396–410, 2015.
- [7] J.-G. Wang, C.-J. Lin, and S.-M. Chen, "Applying fuzzy method to vision-based lane detection and departure warning system," *Expert Systems with Applications*, vol. 37, no. 1, pp. 113–126, 2010.
- [8] T. Xu, S. Chen, D. Wang, and W. Zhang, "Bidirectional sliding window for boundary recognition of pavement construction area using GPS-RTK," *Applied Sciences*, vol. 10, no. 4, p. 1277, 2020.
- [9] S. Hassan and J. Yoon, "Haptic assisted aircraft optimal assembly path planning scheme based on swarming and artificial potential field approach," *Advances in Engineering Software*, vol. 69, pp. 18–25, 2014.
- [10] I. Chabini and S. Shan Lan, "Adaptations of the a^* algorithm for the computation of fastest paths in deterministic discrete-time dynamic networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 3, no. 1, pp. 60–74, 2002.
- [11] J. J. Kuffner Jr. and S. M. LaValle, "RRT-connect: an efficient approach to single-query path planning," in *Proceedings of the IEEE International Conference On Robotics And Automation Symposia Proceedings (Cat. No.00CH37065)*, vol. 2, pp. 995–1001, San Francisco, CA, USA, April 2000.
- [12] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 846–894, 2011.
- [13] J. Nasir, F. Islam, U. Malik et al., "RRT*-SMART: a Rapid convergence implementation of RRT*," *International Journal of Advanced Robotic Systems*, vol. 10, 2013.
- [14] P. Zips, M. Böck, and A. Kugi, "Optimisation based path planning for car parking in narrow environments," *Robotics and Autonomous Systems*, vol. 79, pp. 1–11, 2016.

- [15] M. Du, T. Mei, H. Liang, J. Chen, R. Huang, and P. Zhao, "Drivers' visual behavior-guided RRT motion planner for autonomous on-road driving," *Sensors (Switzerland)*, vol. 16, no. 1, pp. 1–19, 2016.
- [16] X. Tong, C. Siwei, W. Dong, W. Ti, X. Yang, and Z. Weigong, "A novel path planning method for articulated road roller using support vector machine and longest accessible path with course correction," *IEEE Access*, vol. 7, no. 1, pp. 182784–182795, 2019.
- [17] F. M. Barbosa, L. B. Marcos, M. M. da Silva, M. H. Terra, and V. Grassi, "Robust path-following control for articulated heavy-duty vehicles," *Control Engineering Practice*, vol. 85, pp. 246–256, 2019.