Computers, Materials & Continua CMC, vol.65, no.2, pp.1781-1793, 2020

A Hybrid Path Planning Method Based on Articulated Vehicle Model

Zhongping Chen[1], Dong Wang[1,*], Gang Chen[2], Yanxi Ren[3] and Danjie Du[4]

Abstract: Due to the unique steering mechanism and driving characteristics of the articulated vehicle, a hybrid path planning method based on the articulated vehicle model is proposed to meet the demand of obstacle avoidance and searching the path back and forth of the articulated vehicle. First, Support Vector Machine (SVM) theory is used to obtain the two-dimensional optimal zero potential curve and the maximum margin, and then, several key points are selected from the optimal zero potential curves by using Longest Accessible Path (LAP) method. Next, the Cubic Bezier (CB) curve is adopted to connect the curve that satisfies the curvature constraint of the articulated vehicle between every two key points. Finally, Back and Forth Rapidly-exploring Random Tree with Course Correction (BFRRT-CC) is designed to connect paths that do not meet articulated vehicle curvature requirements. Simulation results show that the proposed hybrid path planning method can search a feasible path with a 90-degree turn, which meets the demand for obstacle avoidance and articulated vehicle back-and-forth movement.

Keywords: Path planning, articulated vehicle, back and forth rapidly-exploring random tree, support vector machine, cubic Bezier curve.

1 Introduction

In the past 50 years, path planning technology has developed rapidly, and many planning algorithms have emerged, which include the Genetic Algorithm (GA), Simulated Annealing Algorithm (SAA), Ant Colony Optimization (ACO), and so on. These algorithms have their superiority in solving the problem of the general path planning problems. However, they need to build each obstacle into a specific model, which leads to executable algorithms with high time complexity and does not apply to many obstacles or uneven distribution of a sophisticated environment. Besides, graph search algorithms like A* algorithm, D* algorithm, and Artificial Potential Field (APF) algorithm, although they meet the requirements of optimality and real-time performance in path planning, it

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

[2] School of Mechanical Engineering, Nanjing University of Science and Technology, Nanjing, 210096, China.

[3] 32184 PLA Troops, Beijing, 100093, China.

[4] North Carolina State University, Raleigh, NC, USA.

[*] Corresponding Author: Dong Wang. Email: kingeast16@seu.edu.cn.

Received: 05 April 2020; Accepted: 08 May 2020.

does not meet the vehicle kinematic constraints, which makes it impossible for the vehicle to track the planned path.

Scholars at home and abroad put forward many traditional algorithms for vehicle path planning. Some researchers present vehicle motion planning with RRT-based theory [Karaman and Frazzoli (2011); Karaman, Walter, Perez et al. (2011); Kuwata, Teo, Fiore et al. (2009)], while others come up with algorithms based on one or several curves, such as Bezier curve [Jolly, Kumar, Vijayakumar et al. (2009)], B-spline [Fauser, Chadda, Goergen et al. (2019); Wan, Xu, Ye et al. (2018)] and Dubins curve [Héerissé and Pepy (2013)], which all have good effects on their vehicle models.

In recent years, a novel path planning algorithm using SVM is proposed. As a classic theory in the field of machine learning, this algorithm has great progress in reducing the search space, and sometimes can directly plan an excellent path [Chen, Jiang, Zhao et al. (2017)].

In this paper, we propose a hybrid path planning method using SVM, CB curve and BFRRT-CC. First, SVM theory is used to obtain the two-dimensional optimal zero potential curve and the maximum margin, and then, several key points are selected from the optimal zero potential curve by using LAP. Next, we use CB curve to connect the curve that satisfies the curvature constraint of the articulated vehicle between every two key points. Finally, BFRRT-CC is designed to connect paths that does not meet articulated vehicle curvature requirements [Xia, Hu, Luo et al. (2017)].

2 Nonholonomic constraint articulated vehicle model

The concept of "incomplete" comes from modern analytical mechanics, which first appeared in the scholarly work 'Die Prinzipen der Mechanik' by a German scholar. The non-holonomic constraint refers to the constraint that contains the generalized coordinate derivative of the system and is non-integrable, and expression of the constrained system subject to Eqs. (1) and (2):

$$\mathrm{d}s/\mathrm{d}t = F(s,u,t) \tag{1}$$

$$\mathrm{C}(s,\mathrm{d}s/\mathrm{d}t,t) = 0 \tag{2}$$

where, $s \in \mathbb{R}^n$ is system's state variables, $u \in \mathbb{R}^n$ is system control variables, t is time variable, $\mathrm{C}(s,\mathrm{d}s/\mathrm{d}t,t)$ is the vehicle constraint. The holonomic system can be expressed as $\mathrm{dG}(s,t)/\mathrm{d}t = \mathrm{C}(s,\mathrm{d}s/\mathrm{d}t,t)$, if $\mathrm{G}(s,t)$ exists, the corresponding constraint is holonomic constraint. On the contrary, we call it nonholonomic constraint. Nonholonomic constraint system includes all kinds of vehicles and mobile robots with limited movement. Therefore, researches on path planning under nonholonomic constraint has a broad application and important application value.

In this paper, the articulated vehicle model is simplified as two wheels and an articulated frame, as shown in Fig. 1. Take the forward motion as an example, we view the middle point of the front bridge P_f as the reference point of the roller, and view the velocity direction of this point v_f as the forward direction of the roller , and the coordinate change of the middle point of the front bridge can be expressed as Eqs. (3) and (4):

$$\mathrm{d}x_f/\mathrm{d}t = v_f \cos\theta_f \tag{3}$$

$$\mathrm{d}y_f/\mathrm{d}t = v_f \sin\theta_f \tag{4}$$

where v_f is the speed of the middle point of the front bridge, x_f and y_f are the horizontal and vertical coordinates of the midpoint of the front bridge in the coordinate system separately, θ_f is the course angle of the front bridge, and the gradient of the course angle can be expressed as Eq. (5):

$$\mathrm{d}\theta_f/\mathrm{d}t=(v_f \sin\gamma + l_r\dot{\gamma})/(l_f \cos\gamma + l_r) \tag{5}$$

where l_f is the distance from the front bridge to the hinge point, l_r is the distance from the rear bridge to the hinge point and γ is the articulated steering angle, define the right as γ^+ and the left as γ^- when the roller moves forward, and define the right as γ^- and the left as γ^+ when the roller moves backward.

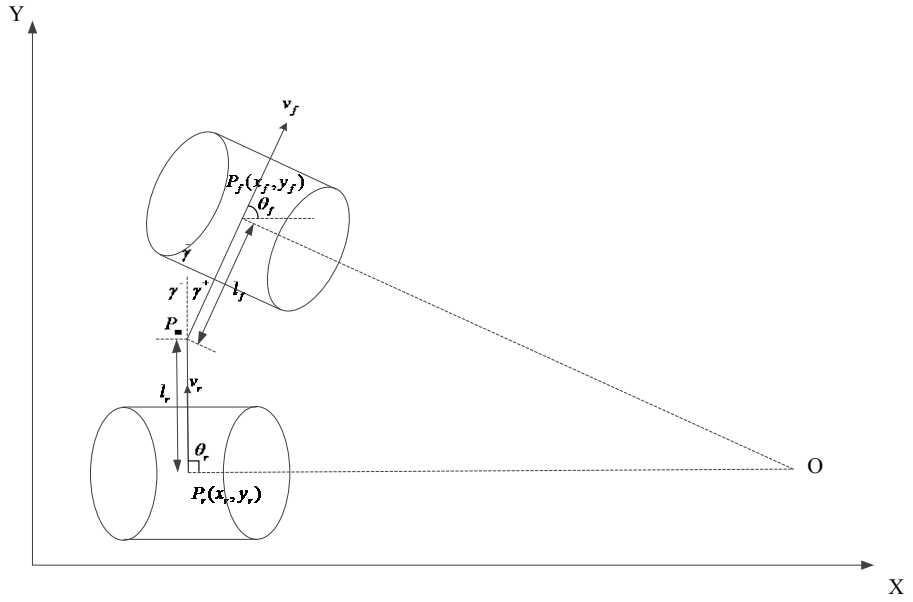

Figure 1: The articulated vehicle model

The articulated vehicle kinematics constraint equation can be obtained by Eqs. (3) to (5):

$$\mathrm{d}s/\mathrm{d}t=\begin{bmatrix} \mathrm{d}x_f/\mathrm{d}t \\ \mathrm{d}y_f/\mathrm{d}t \\ \mathrm{d}\theta_f/\mathrm{d}t \\ \mathrm{d}v_f/\mathrm{d}t \\ \mathrm{d}\gamma/\mathrm{d}t \end{bmatrix} = \begin{bmatrix} v_f\cos\theta_f \\ v_f\sin\theta_f \\ (v_f\sin\gamma + l_r\dot{\gamma})/(l_f\cos\gamma + l_r) \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} u_0 + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u_1 \tag{6}$$

where the control variable u_0 is acceleration and the control variable u_1 is the rate of the change of hinged steering angle. These constrain conditions should be satisfied: $|u_0| \le u_{0\max}$, $|u_1| \le u_{1\max}$ and $|\gamma| \le \gamma_{\max}$. The maximum curvature is expressed as Eq. (7):

$$\kappa^*(\tau)=\frac{1}{\mathrm{r}_{\min}}=\frac{1}{l}\sin\frac{\gamma_{\max}}{2}=0.0661\mathrm{m}^{-1} \tag{7}$$

3 The hybrid path planning method

The improved algorithms are presented in detail in this part, and it mainly divides into four sections. Section 3.1 briefly introduces the principle of SVM and the way to obtain SVM model parameters. Section 3.2 introduces how to search key points with LAP theory. In Section 3.3, a suitable CB curve is selected to connect the curve segment that satisfies the curvature constraint of the articulated vehicle between every two key points. In Section 3.4, BFRRT-CC is designed to connect paths that does not meet articulated vehicle curvature requirements.

3.1 Optimal zero-potential decision boundary and free space extraction based on SVM theory

SVM is a binary model and the idea of the model is to map vectors to a higher dimensional space and establish an optimal classification hyper-plane in this space, which maximizes margin between two classes [Chen, Xiong, Xu et al. (2019)].

Since the state of start and end points including articulated vehicle position and course is given initially, in order to meet the actual needs, we add three positive virtual obstacles and three negative ones on both sides, and extract optimal zero-potential decision boundary and obstacle-free space based on SVM theory [Yuan, Yao, Tan et al. (2018); Tang, Xie, Yang et al. (2019)].

Define (p_i, y_i) as SVM training samples, where $p_i=(X_i, Y_i)$ extracts the coordinates of the data points from the known obstacles, and the equation $f(p)=w^T\phi(p)+b=0$ can be converted to the convex optimization problem:

$$\begin{aligned} &min\frac{1}{2}\|w\|^2+c\sum_{i=1}^{n}\xi_i \\ &\text{s.t.}\quad y_i(w^T\phi(p_i)+b)\geq 1-\xi_i \quad , \quad \xi_i\geq 0, i=1,2,...,n \end{aligned} \tag{8}$$

where $w=(w_1; w_2; ...; w_d)$ is the normal vector, which determines the direction of the hyper-plane, b is displacement, which indicates the distance between the hyper-plane and the origin, n is the number of support vector, ξ_i are the slack variables, c is the error penalty factor, $\phi(p):\Re^2\rightarrow\Re^H$ maps data points from two-dimensional spatial to high-dimensional reproducing kernel Hilbert space, to separate the data set linearly. Through the Lagrange multiplier method, we can get the dual problem in Eq. (9) of the Eq. (8).

$$\begin{aligned} &max\sum_{i=1}^{n}\alpha_i-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j k(p_i,p_j) \\ &\text{s.t.}\quad \sum_{i=1}^{n}\alpha_i y_i=0, 0<\alpha_i<c \quad , \quad i=1,2,...,n \end{aligned} \tag{9}$$

where α_i is the Lagrange multiplier, and the kernel function $k(p_i,p_j)=\phi(p_i)^T\phi(p_j)$ is satisfied with Mercer's theorem. In this paper, RBF kernel is selected as kernel function and the expression is Eq. (10):

$$k(p,p_i)=exp(-\gamma\|p-p_i\|^2) \tag{10}$$

where γ is the kernel parameter of RBF function. Set the decision function to zero, the

calculation formulas of the hyper-plane $f(p)$ and interval distance $D(p_i)$ can be obtained as follows Eqs. (11) and (12):

$$f(p)=\sum_{i=1}^{n}\alpha_i y_i k(p,p_i)+b \tag{11}$$

$$D(p_i)=|w^{\mathrm{T}}\cdot p_i+b|/\|w\| \tag{12}$$

where $w=\sum_{i=1}^{n}\alpha_i y_i \phi(p_i)$.

According to the basic theory of SVM, the error penalty factor and the kernel parameter play a crucial role. This paper adopts 10-fold cross-validation to find the optimal parameters.

3.2 Key points searching with longest accessible path theory

After drawing the zero-potential decision boundary on the two dimensional map, we need to find several key points and three key points are selected, as shown in Fig. 2. We define the connection between two points as the path, and when the path does not go through any obstacles, we call it an accessible path, otherwise it is termed as an illegal path. Then try to connect the latest point with the starting point and so on, and find the longest accessible path, and that point is set to a key point and so on until the end point is connected.

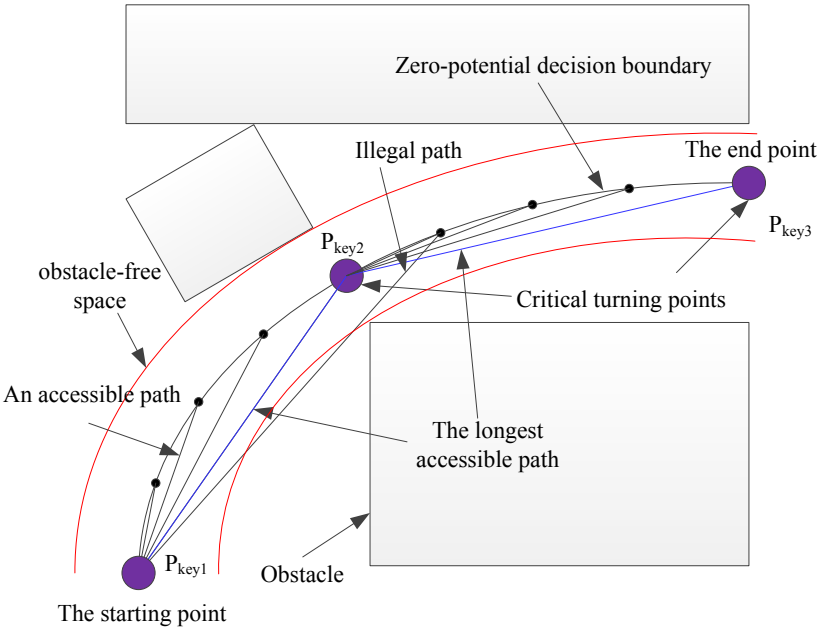

Figure 2: Key points set on the zero-potential decision boundary

3.3 Cubic bezier curves

CB curve is determined by four vertices, where the first vertex and the last vertex determine the start and the end point of the curve, and the remaining two vertices control

the course of the curve at the start and end points, as well as the shape of the curve [Chen, Qin, Xu et al. (2019); Yoon, Lee, Jung et al. (2018)]. Define four vertices in two-dimensional space, CB curve can be expressed as Eq. (13):

$$B(t) = (1-t)^3 b_0 + 3(1-t)^2 t b_1 + 3(1-t)t^2 b_2 + t^3 b_3 \quad , \quad t \in [0,1] \tag{13}$$

The course and position of the start and the end points are shown in the Eq. (14), and the position of the other two vertices are determined by the Eq. (15):

$$\begin{cases} B(0) = b_0, \quad B(1) = b_3 \\ \theta_{start} = atan2(b_{1y} - b_{0y}, b_{1x} - b_{0x}) \\ \theta_{end} = atan2(b_{3y} - b_{2y}, b_{3x} - b_{2x}) \end{cases} \tag{14}$$

$$\begin{cases} b_{1x} = b_{0x} + step_1 * cos(\theta_{start}) \\ b_{1y} = b_{0y} + step_1 * sin(\theta_{start}) \\ b_{2x} = b_{3x} - step_2 * cos(\theta_{end}) \\ b_{2y} = b_{3y} - step_2 * sin(\theta_{end}) \end{cases} \tag{15}$$

where $step_1$ and $step_2$ are the given constant value, and we use the linear interpolation to test whether the generated curve can avoid obstacles. Curvature at any point on the curve is calculated as Eq. (16):

$$\kappa(t) = \frac{p_x'(t) p_y''(t) - p_x''(t) p_y'(t)}{(p_x'(t)^2 + p_y'(t)^2)^{\frac{3}{2}}} \tag{16}$$

3.4 Back and forth RRT with course correction

Path planning problem can be regarded as the search problem of bounded space $C \in \mathbb{R}^n$. Define bounded space $C \subseteq \mathbb{R}^3$, obstacle space $C_{obs} \subset C$, free space C_{free}=$C \setminus C_{obs}$, and target area $X_{\text{goal}} \in C_{\text{free}}$. Path planning problem in C space can be described as: calculate a feasible continuous path τ:$[0,1] \to C_{free}$, and meet requirements of $\tau(0)$=q_{init} and $\tau(1)$=q_{goal}.

3.4.1 Back and forth RRT

Vehicle motion planning with RRT-based theory mentioned in the latest documents does not consider the demand of articulated vehicle back-and-forth movement, this paper puts forward BFRRT algorithm and its cost function can be expressed as Eq. (17):

$$Cost(q_i, q_{rand}) = \omega_1 \bullet \tilde{E}(q_i, q_{rand}) + \omega_2 \bullet \tilde{C}(q_i, q_{rand}) \tag{17}$$

where $\tilde{E}$ is normalized euclidean distance, $\tilde{C}$ is normalized course metric, and the expressions are Eqs. (18) and (19):

$$\tilde{E}(q_i, q_{rand}) = \varsigma_1(d) \| q_i - q_{rand} \| \tag{18}$$

$$\tilde{C}(q_i, q_{rand}) = \varsigma_2(\theta) | Angle(\overrightarrow{q_i q_{rand}} \bullet \overrightarrow{q_{rand} q_{goal}}) | \tag{19}$$

The Min-Max Normalization is used to realize the normalization, which is expressed as

Eqs. (20) and (21):

$\varsigma_1(d)=(d - d_{min})/(d_{max} - d_{min})$ (20)

$\varsigma_2(\theta)=(|\theta| - \theta_{min})/(\theta_{max} - \theta_{min})$ (21)

The algorithm proposed is shown in Algorithm 1 in detail.

Algorithm 1: Back and Forth RRT Path Planner (BERRT)

Input: $K \in \mathrm{N}$, $q_{init} \in C_{free}$, $q_{goal} \in Q_{goal} \in C_{free}$

Output: G

$V \leftarrow \{q_{init}\}$, $E \leftarrow \{\varnothing\}$, $G \leftarrow \{V, E\}$

for $k \leftarrow 1$ **to** K **do**

  if $rand() < p1$ **then**

    $q_{rand} \leftarrow q_{goal}$

  else

    $q_{rand} \leftarrow$ RANDOMSTATE

  $(q_{near}, q_{neardir}) \leftarrow$ BFFindQNear (q_{rand}, V)

  $(u_{best}, q_{new}, success) \leftarrow$ BFFindQNew $(q_{near}, q_{neardir}, q_{goal})$

  if *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 $q_{new} \cong q_{goal}$ **then**

        $V \leftarrow V \setminus \{q_{new}\}$, $V \leftarrow V \cup \{q_{goal}\}$

      path $\leftarrow$ **TURE;**

      $G \leftarrow \{V, E\}$

Return G **if** path==**TURE; else FAILURE**

3.4.2 Course correction

In general, a feasible path cannot be searched only by using BFRRT theory, in which case CC with with fixed position of the front wheel is presented to deal with it [Wang, Willianms, Angley et al. (2019)].

CC algorithm is shown in Algorithm 2 in detail. Calculate the absolute value of the difference between the target angle θ_{target} and the roller's course angle θ_{real} at the position to be corrected, and judge the relationship between that angular deviation $|\Delta\theta|$ and preset

threshold θ_{th}, if the angular deviation $|\Delta\theta|$ is greater than preset threshold θ_{th}, use the maximum step times 'steptimes1' in the step matrix to adjust the roller's course.

After turning back and forth, judge the relationship between that angular deviation $|\Delta\theta|$ and preset threshold θ_{th} again, until $|\Delta\theta| <= \theta_{th}$. If $|\Delta\theta| <= \theta_{maxerror}$, the vertex set V and the edge set E are delivered to the directed graph G directly, otherwise we must look up the matrix with the relationship between $|\Delta\theta|$ and the step times, and in this case, Fowardbackward function corrects course.

Algorithm 2: CourseCorrection(CC)

Input: θ_{real}, θ_{target}, $\theta_{\max error}$, G

Output: G

$V \leftarrow G.V$, $E \leftarrow G.E$

Set the stepimes1 parameter to a positive integer obtained by experiment and the flag parameter to 1

$\Delta\theta \leftarrow \theta_{real} - \theta_{target}$

$dir \leftarrow \text{sign}(\Delta\theta)$

while $|\Delta\theta| > \theta_{th}$ **do**

- $V \leftarrow \text{Forwardbackward}(V, dir, flag, steptimes1)$
- $E \leftarrow E \cup \{q_{end}, q_{new}\}$
- $V \leftarrow \text{Forwardbackward}(V, -dir, -flag, steptimes1)$
- $E \leftarrow E \cup \{q_{end}, q_{new}\}$
- $\Delta\theta \leftarrow \theta_{real} - \theta_{target}$

if $|\Delta\theta| \le \theta_{th}$ **and** $|\Delta\theta| > \theta_{\max error}$ **then**

- stepimes2 $\leftarrow$ DeltatoStepTime$(\Delta\theta)$
- $V \leftarrow \text{Forwardbackward}(V, dir, flag, steptimes2)$
- $E \leftarrow E \cup \{q_{end}, q_{new}\}$
- $V \leftarrow \text{Forwardbackward}(V, -dir, -flag, steptimes2)$
- $E \leftarrow E \cup \{q_{end}, q_{new}\}$

$G \leftarrow \{V, E\}$

Return G

4 Experimental results

4.1 The experimental scene and parameters of the articulated vehicle

As shown in Fig. 3, the start and end point of the articulated vehicle front wheel is [20, 85] with 0° and [85, 20] with -90° respectively, and the shaded part indicates the obstacle. Parameters of the articulated vehicle are used in this experiment, shown in Tab. 1 (i.e., Dynapac CC6200 vibratory roller).

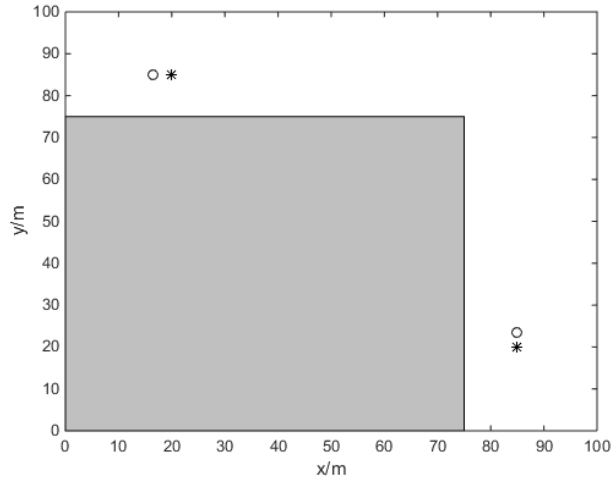

Figure 3: The experimental scene

Table 1: Parameters of the articulated vehicle (i.e., Dynamic CC6200)

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

4.2 Experimental results of improved BFRRT-CC

We first determine the SVM model and draw the free space using the experimental scene shown in Fig. 4, and the best kernel parameter and the best error penalty factor c are 0.0059208 and 36.7583 respectively, thus, the zero-potential decision boundary is drawn and three key points are searched by LAP. Then we use CB curve to connect every two key points in turn and divide the fitted curve into 1000 segments to calculate the curvature at each point. As shown in Fig. 5, the generated CB curve is not suitable for the

articulated vehicle to follow from about the point 900 to the point 1100, and in this case, BFRRT-CC is used to connect this segment. Fig. 6 shows CB curve segment with curvature meeting vehicle constraints and in Fig. 7, the hybrid path planning method based on articulated vehicle model is shown and the number from 1 to 8 indicate the best BFRRT-CC searching path.

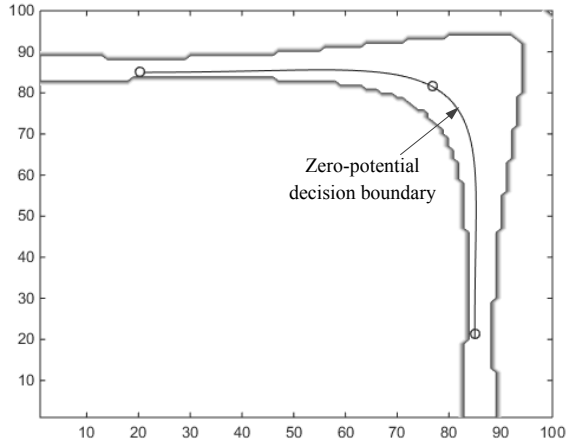

Figure 4: Zero-potential decision boundary fitted by CB curve

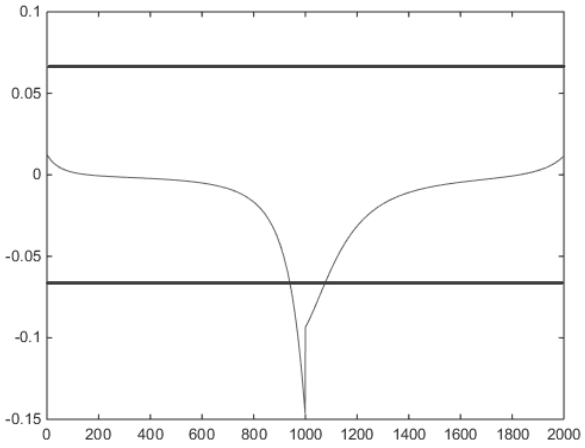

Figure 5: The curvature of each point on CB curve

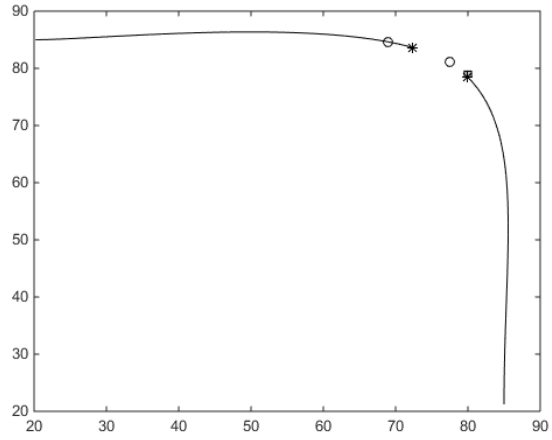

Figure 6: CB curve segment with curvature meeting vehicle constraints

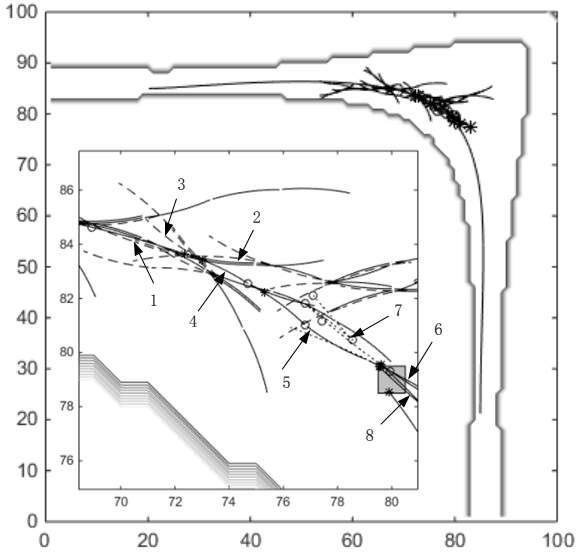

Figure 7: The hybrid path planning method based on articulated vehicle model

5 Conclusion

In this paper, a hybrid path planning method based on articulated vehicle model is proposed to meet the demand of obstacle avoidance and searching the path back and forth. Simulation experiments show that the proposed algorithm can search a feasible path with a 90 degree turn.

In future research, we will use the articulated vehicle to follow the planned path and solve the real-time dynamic obstacle avoidance problem on the planned path. When the

articulated vehicle detects that a person or a vehicle is moving on the planned path, it can also stop in time and wait for the person or the vehicle to leave the path.

Funding Statement: This work was supported by the Jiangsu Natural Science Foundation Project BK20170681, and National Natural Science Foundation of China 51675281.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

Chen, J.; Jiang, W.; Zhao, P.; Hu, J. (2017): A path planning method of anti-jamming ability improvement for autonomous vehicle navigating in off-road environments. *Industrial Robot: An International Journal*, vol. 44, no. 4, pp. 406-415.

Chen, L.; Qin, D.; Xu, X.; Cai, Y.; Xie, J. (2019): A path and velocity planning method for lane changing collision avoidance of intelligent vehicle based on cubic 3-D Bezier curve. *Advances in Engineering Software*, vol. 132, pp. 65-73.

Chen, Y.; Xiong, J.; Xu, W.; Zuo, J. (2019): A novel online incremental and decremental learning algorithm based on variable support vector machine. *Cluster Computing*, vol. 22, no. 3, pp. 7435-7445.

Fauser, J.; Chadda, R.; Goergen, Y.; Hessinger, M.; Motzki, P. et al. (2019): Planning for flexible surgical robots via Bézier spline translation. *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3270-3277.

Hérissé, B.; Pepy, R. (2013): Shortest paths for the Dubins' vehicle in heterogeneous environments. *52nd IEEE Conference on Decision and Control*, pp. 4504-4509.

Huy, Q.; Mita, S.; Yoneda, K. (2013): A practical and optimal path planning for autonomous parking using fast marching algorithm and support vector machine. *IEICE TRANSACTIONS on Information and Systems*, vol. 96, no. 12, pp. 2795-2804.

Jolly, K. G.; Kumar, R. S.; Vijayakumar, R. (2009): 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.

Karaman, S.; Frazzoli, E. (2011): Sampling-based algorithms for optimal motion planning. *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 846-894.

Karaman, S.; Walter, M. R.; Perez, A.; Frazzoli, E.; Teller, S. (2011): Anytime motion planning using the RRT. *IEEE International Conference on Robotics and Automation*, pp. 1478-1483.

Kuwata, Y.; Teo, J.; Fiore, G.; Karaman, S.; Frazzoli, E. et al. (2009): Real-time motion planning with applications to autonomous urban driving. *IEEE Transactions on Control Systems Technology*, vol. 17, no. 5, pp. 1105-1118.

Tang, Q.; Xie, M.; Yang, K.; Luo, Y.; Zhou, D. et al. (2019): A decision function based smart charging and discharging strategy for electric vehicle in smart grid. *Mobile Networks and Applications*, vol. 24, no. 5, pp. 1722-1731.

Wan, N.; Xu, D.; Ye, H. (2018): Improved cubic B-spline curve method for path optimization of manipulator obstacle avoidance. *Chinese Automation Congress*, pp. 1471-1476.

Wang, B.; Kong, W.; Li, W.; Xiong, N. N. (2019): A dual-chaining watermark scheme for data integrity protection in Internet of Things. *Computers, Materials & Continua*, vol. 58, no. 3, pp. 679-695.

Wang, B.; Kong, W.; Guan, H.; Xiong, N. N. (2019): Air quality forecasting based on gated recurrent long short term memory model in Internet of things. *IEEE Access*, vol. 7, pp. 69524-69534.

Wang, X.; Williams, S.; Angley, D.; Gilliam, C.; Jackson, T. et al. (2019): RRT* trajectory scheduling using angles-only measurements for AUV recovery. *IEEE 22th International Conference on Information Fusion*, pp. 1-6.

Xia, Z.; Hu, Z.; Luo, J. (2017): UPTP vehicle trajectory prediction based on user preference under complexity environment. *Wireless Personal Communications*, vol. 97, no. 3, pp. 4651-4665.

Yoon, S.; Lee, D.; Jung, J.; Shim, D. H. (2018): Spline-based RRT∗ using piecewise continuous collision-checking algorithm for car-like vehicles. *Journal of Intelligent & Robotic Systems*, vol. 90, no. 3-4, pp. 537-549.

Yuan, L.; Yao, E.; Tan, G. (2018): Automated and precise event detection method for big data in biomedical imaging with support vector machine. *Computer Systems Science and Engineering*, vol. 33, no. 2, pp. 105-114.