2019 IEEE Intelligent Vehicles Symposium (IV)
Paris, France. June 9-12, 2019

A Novel Robust Lane Change Trajectory Planning Method for Autonomous Vehicle

Dequan Zeng[1,2], Zhuoping Yu[1,2], Lu Xiong[1,2], Junqiao Zhao[3], Peizhi Zhang[1,2], Zhiqiang Li[1,2] , Zhiqiang Fu[1,2], Jie Yao[4], and Yi Zhou[4]

Abstract— **A novel trajectory planning method is proposed in this paper for lane change of autonomous vehicle. Since it is difficult to accurately capture the trajectory of other vehicles, which means the trajectory for autonomous vehicle couldn't always easy to generate quickly. Moreover, the motion planning, as a kind of high-dimensional optimization problem with multiple nonlinear constraints, requires lots of resources to find a right solution. Therefore, we present a trajectory monitoring strategy to keep robust in lane change scenario, which generates the lane change and monitoring trajectory at the same time. If the former does not produce a safe trajectory or is time out, the monitoring trajectory will be taken as the result output. To meet the constraints of vehicle's motion and real-time requirements, B-spline-based method will be employed to plan a continuous curvature path. And RRT-based method works as a supplement for keeping algorithm completeness. Then the monitory trajectory mainly obeys collision-free requirements, which computes deceleration that keeps vehicle stability. The results illustrate that both B-spline-based method and RRT-based could generate curvature continuous and meet the limitation for motion, however, both have the possibility of timeout. Especially, there are challenge to the success rate as environment becomes more complex.**

I. Introduction

As a technology with promising prospects, autonomous vehicles have achieved good development in recent decades, and a large number of technical products have begun to be applied [1-3]. Compared with human beings, intelligent driving system is expected to improve driving safety, ride comfort, traffic efficiency and energy economy due to its rapid operation, perception beyond visual range and accurate decision making [4-6].

In order to safely and fast navigate the intelligent vehicle from the current position to goal point, it is essential to design a robust motion plan method for generating trajectory[7]. As literatures review illustrated [1-2], the motion plan architectures mainly consist of graph search-based planners, sampling-based planners, interpolating curve planners and numerical optimization. Dijkstra [8] and A* [9] are well-kown graph search-based planners, which could find the shortest path by traversing the grid, whether of equal size [10] or variable granularity [11], after discretizing the environment. Although there are many variants of the algorithm used in smart cars such as D* [12], ARA*[13], AD*[14], it is still hard to choose the grid resolution and track the path composed of the connected grids [15]. Different from the discrete approach to the environment, sampling-based planners, such as RRT [16], could produce valid path by extending nodes in a continuous space until the target position is reached. The latest variants of the algorithm applied to the intelligent car are RRT* [17] and CL-RRT [18], however, the result of planning is not optimal and the generated path twists and turns make it difficult for vehicles to track directly due to the randomness of sampling. As the ability to achieve continuity of curvature, interpolating curve planners, such as polynomial curves [19], clothoid curves [20], Bezier curves [21] and B spline [22], always requires a given set of way-points to fit out a path. As the scenario of the autonomous vehicle is dynamic in real time, it is difficult to exhaust all the environments to get the way-points. By constructing obstacle constraints, starting point constraints, goal point constraints, model constraints, and optimization objective functions (e.g., shortest path and most comfortable), numerical optimization [23-25] could plan the optimal smooth trajectory without the way points. However, this method is essentially a kind of high-dimensional optimization problem with multiple nonlinear constraints, which takes lots of resources to find a right solution. Literatures [26-27] shows that in order to generate a parking trajectory satisfying multi-objective constraints, the CPU time is often no less than 300ms, even 400000ms.Moreover, there is also the possibility of planning failure [15].

In order to improve the robustness of the motion planning algorithm avoiding the dilemma of no way out, we propose a trajectory monitoring strategy that generates the lane change trajectory and the monitoring trajectory at the same time. If the former does not produce a safe trajectory or is time out, the monitoring trajectory will be taken as the result output. Otherwise, the output is lane change trajectory. To meet the constraints of vehicle's motion and real-time requirements, B-spline-based method will be employed to plan a continuous curvature path for lane change. And RRT-based method works as a supplement for keeping completeness of lane change plan algorithm. Then the monitory trajectory mainly obeys collision-free requirements, which computes deceleration that keeps vehicle stability. The remainder of the paper is

* This work is supported by the Project supported by the National Key Research and Development Program (Grant no. 2016YFB0100901), the Shanghai Municipal Commission of Economy and Informatization (Grant no. 2018RGZN02050), and the Science and Technology Commission of Shanghai (Grant no. 17DZ1100202; 16DZ1100700).

[1]Dequan Zeng, Zhuoping Yu, Lu Xiong, Peizhi Zhang, Zhiqiang Li and Zhiqiang Fu are with School of Automotive Studies, Tongji University, Shanghai 201804, China (corresponding author to provide phone: 86 18930186089; e-mail: zdq1610849@ 126.com);

[2]Dequan Zeng, Zhuoping Yu, Lu Xiong, Peizhi Zhang, Zhiqiang Li and Zhiqiang Fu are with Clean Energy Automotive Engineering Centre, Tongji University, Shanghai 201804, China;

[3]Junqiao Zhao is with College of Electronics and Information Engineering, Tongji University, Shanghai 201800, China (e-mail: zdq1610849@ 126.com).

[4]Jie Yao and Yi Zhou is with SAIC Motor Corporation Limited, Shanghai 200041, China (e-mail: zdq1610849@ 126.com).

978-1-7281-0560-4/19/$31.00 ©2019 IEEE

structured as follows: In Sec. II, a description of trajectory monitoring strategy. B-spline-based lane change path planning is described in Sec. III. RRT-based lane change path planning is presented in Sec. VI. Sec. V describes the velocity planning method. Sec. IV implements and analysis the trajectory planning algorithm for typical lane change scenario. Finally, Sec. VII provides the conclusion..

In generally, it is difficult to accurately grasp the trajectory of surrounding vehicles, which means the lane change trajectory for autonomous vehicle couldn't always easy to get quickly. Moreover, the nature of motion planning is a kind of high-dimensional optimization problem with multiple nonlinear constraints, which takes lots of resources to find a right solution. In consequence, the planner doesn't always generate trajectory in the right time. In order to avoid collision, it is necessary to set up a security policy, which is like the functional security of a vehicle's component, that could provide the safest trajectory when necessary. Therefore, we propose a trajectory monitoring strategy, as illustrated in Fig.1, for lane change scenario. If the driving task remains unchanged and the current trajectory is safe, the current trajectory shall be maintained. On the contrary, the lane change trajectory and the monitoring trajectory will start planning at the same time. As shown in Fig.2, the lane change trajectory (light blue curve with No.2) is generated by B-spline-based and RRT-based method, and the monitoring trajectory (yellow line with No.3) is generated at deceleration. If the former does not produce a safe trajectory or is time out, the monitoring trajectory will be taken as the result output. Otherwise, the output is lane change trajectory.

Although the monitoring trajectory has the effect similar to AEB, it is more "intelligent" than AEB. As illustrated in Fig.3, when the vehicle changes lanes, the roadside anti-collision pile enters the vehicle perception area (the shortest distance will be less than a lane width), which has reached the condition of triggering AEB. However, the lane change trajectory executed by the vehicle should be safe and AEB should not intervene.

If the deceleration of monitory trajectory is a_x, for straight driving

$$a_x = \min\left\{ |a_{\min}|, \left| \frac{v_0^2 - v_g^2}{2L_{0g}} \right| \right\} \tag{1}$$

where a_{min} is minimum acceleration of vehicle, v_0 is start speed, v_g is goal speed, L_{0g} is deceleration length.

If the turn radius is R, for straight driving

$$a_x = \min\left\{ |a_{\min}|, \left| \frac{v_0^2 - v_g^2}{2L_{0g}} \right|, \left| \mu g \sqrt{1 - \left(\frac{v_0^4}{0.4\mu g R^2} \right)} \right| \right\} \tag{2}$$

where μ is road adhesion coefficient.

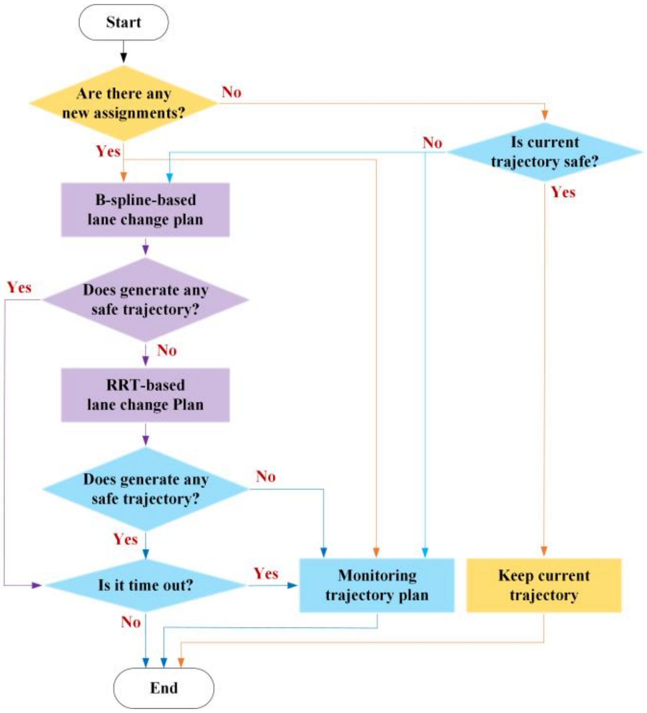

Start
Are there any new assignments?
No
Yes
Is current trajectory safe?
No
Yes
B-spline-based lane change plan
Yes
Does generate any safe trajectory?
No
RRT-based lane change Plan
Does generate any safe trajectory?
No
Yes
Is it time out?
Yes
No
Monitoring trajectory plan
Keep current trajectory
End

Fig.1 Trajectory monitoring strategy.

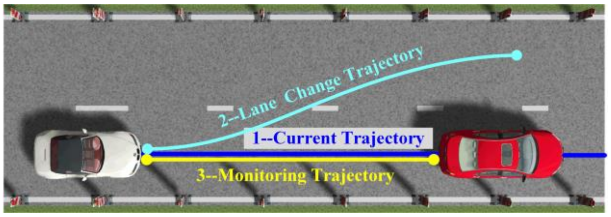


Fig.2 Lane change scenario 1

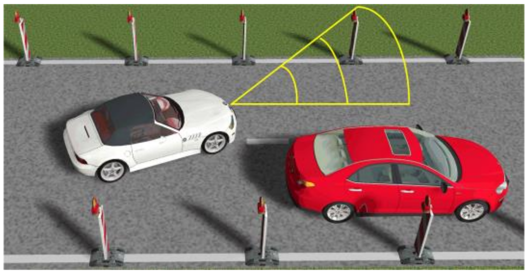


Fig.3 Lane change scenario 2

II. B-SPLINE-BASED LANE CHANGE PATH PLANNING

A. Problem definition

In general, the path planning is typical two-points boundary value problem, which could be defined as

$$\begin{aligned} \min \quad & f(x) \\ s.t. \quad & g_i(x) = 0, i = 1, \ldots, n \\ & h_j(x) \le 0, j = 1, \ldots, m \end{aligned} \tag{3}$$

where $f(x)$ is objective function that is usually using the shortest time or journey to describe the cost, $g_i(x)$ are equality constraints like start and target points, and $h_j(x)$ are inequality constraints such as safe distance.

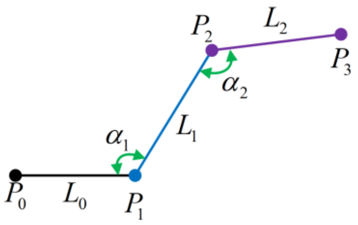


Fig.4 Control segments of cubic B-spline curve

B. Curvature constraints

To meet to restriction of vehicle steering actuators, which means the generated path should be subject to continuous curvature, peak curvature and maximum curvature rate, the parameterized cubic B-spline curve is employed to plan path.

For a 4 control points $\{P_i\}_{i=0}^{3}$ and 7 parametric nodes $T_{4,3}=\{t_i\}_{i=0}^{6},(t_i<t_{i+1})$, an 3 degree B-spline curve can be defined as (4).

$$P(t)=\sum_{i=0}^{3}P_iB_{i,3}(t),\quad t\in[t_2,t_4] \tag{4}$$

where $B_{i,j}$ is B-spline basic function, which could be computed using deBoox-Cox equations like (5).

$$\begin{cases} B_{i,1}(t)=\begin{cases}1, & t\in[t,t_{i+1}) \\ 0, & t\notin[t_i,t_{i+1})\end{cases} \\ B_{i,3}(t)=\dfrac{t-t_i}{t_{i+2}-t_i}B_{i,3}(t)+\dfrac{t_{i+2}-t}{t_{i+3}-t_{i+1}}B_{i+1,2}(t)\end{cases} \tag{5}$$

To meet the constraints of start point and goal point, the endpoints at the top and bottom need to be set to triple nodes, so the parametric nodes' vector would be set as [0,0,0,0,0.5,1,1,1,1], then the expression of the generated path points via cubic B-spline is respectively described by equation (6).

$$\begin{cases} x_i(t)=\left(-t^3+3t^2-3t+1+t^3\cos\alpha_i\right)L_j \\ y_i(t)=t^3L_j\sin\alpha_i\end{cases},i=1,2,j=0,1,2 \tag{6}$$

The curvature expression of a curve is defined as (7).

$$k=\frac{\dot{x}(t)\ddot{y}(t)-\ddot{x}(t)\dot{y}(t)}{\sqrt{\left[(\dot{x}(t))^2+(\dot{y}(t))^2\right]^3}} \tag{7}$$

The control line segments constructed by the control points are as shown in the Fig.4. To satisfied the peak curvature of autonomous vehicle, the length of control line and included angle are subject to [28]

$$\frac{1}{6}\frac{\sin\alpha_i}{L_j}\left(\frac{1-\cos\alpha_i}{8}\right)^{-3/2}\le k_{\max}\quad ,i=1,2,j=0,1,2 \tag{8}$$

C. Safe distance constraints

Based on convex characteristics of b-spline curve, the safe detection of generating path could be represented by the distance between the control segments and the obstacles. A typical scenario is shown in Fig.5.

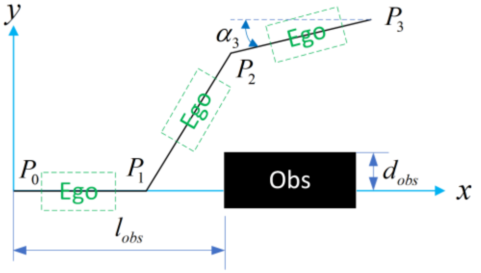


Fig.5 Safe detection

According to the ego vehicle's location, there are three phases occurring collision risk, of which respective security constraints are in violation. It is assumed that the autonomous vehicle's obstacle avoidance is achieved by forced steering rather than changing the speed, which is aiming at making passengers feel comfortable. Therefore, the detail safe detection are as follows:

1) longitudinal safe distance for the first control segments length L_0

$$l_{obs}+v_{obs}t_0-\left(L_0+l_{fba}\right)\ge\Delta_{log}\quad ,t_0=L_0/v_{ego} \tag{9}$$

2) latitudinal safe distance for the second control segments length L_1

$$\begin{aligned} &l_{obs}+v_{obs}t_1-L_0+\left(L_1+l_{fba}\right)\left|\cos\alpha_1\right| \\ &+0.5w\left|\sin\alpha_1\right|\ge\Delta_{lat}\quad ,t_1=L_1/v_{ego}\end{aligned} \tag{10}$$

3) latitudinal safe distance for the third control segments length L_1

$$\begin{aligned} &l_{obs}+v_{obs}t_2-\left(x_2-x_0+\left(L_1+l_{fba}\right)\right)\sin\alpha_3-0.5w \\ &-\left(d_{obs}-\left(y_2-y_1\right)\right)\cos\alpha_3\ge\Delta_{lat}\quad ,t_2=L_2/v_{ego}\end{aligned} \tag{11}$$

where l_{fba} is ego vehicle length from rear axis to head, w is ego vehicle width, Δ_{log} is longitudinal safe distance, Δ_{lat} is lateral safe distance, v_{ego} is ego autonomous vehicle speed.

Finally, the path planning is typical two-points boundary value problem, which could be defined as

$$\begin{aligned} \min\quad & \sum_{j=0}^{2}L_j \\ s.t.\quad & X_0=\left(x_0,y_0,\varphi_0,v_0\right) \\ & X_g=\left(x_g,y_g,\varphi_g,v_g\right) \\ & \alpha_{\min}\le\alpha_i\le\alpha_{\max}\quad ,i=1,2 \\ & \frac{1}{6}\frac{\sin\alpha_i}{L_j}\left(\frac{1-\cos\alpha_i}{8}\right)^{-3/2}-k_{\max}\le 0 \\ & \Delta_{log}-h\left(l_{obs},v_{obs},L_0,l_{fba},v_{ego}\right)\le 0 \\ & \Delta_{lat}-h\left(l_{obs},v_{obs},L_0,L_1,w,\alpha_1,l_{fba},v_{ego}\right)\le 0 \\ & \Delta_{lat}-h\left(l_{obs},v_{obs},L_0,L_1,L_2,w,\alpha_1,\alpha_3,l_{fba},v_{ego}\right)\le 0 \end{aligned} \tag{12}$$

where $X_0=(x_0,y_0,\varphi_0,v_0)$ is start point and $X_g=(x_g,y_g,\varphi_g,v_g)$ is goal point.

Equation (1) and (10) illustrates that the path planning of intelligent vehicle is a linear programming problem with multiple constraints. However, the constraints are mostly nonlinear. There are three main method to deal with this kind of problem: offline data set method, convex optimization and Bionic intelligent optimization algorithm. Due to the environment of the intelligent vehicle is difficult to be exhaustive and many off-line data storage, extraction and matching require a large amount of computing resources, the application of the offline data set method is limited. As convex optimization method requires that the domain is convex and all obstacles in the environment must be expressed by formal generalized affine set, it aggravates the processing requirements of environment map and makes it difficult to be applied in high-speed dynamic environment. Compared with the above solution methods, the bionic intelligent optimization algorithm is very suitable for solving high-dimensional nonlinear optimization problems due to its low requirements on the definition domain, function composition and environment description of the optimization function, and has been widely used in motion planning [29]. After continuous attempts, we finally chose ant colony algorithm to solve path planning problem.

III. RRT-BASED LANE CHANGE PATH PLANNING

Due to the optimization problem established by the equation (10) is strict with the curvature limitation and the ant colony solution algorithm's local minimum value problem makes the algorithm lose its completeness. Therefore, an RRT-based programming link is added to find the path as far as possible when the B-spline-based algorithm has no solution but the path exists. Of course, in order to speed up the solving speed of the algorithm, some measures are introduced to reduce the blindness of sampling, such as gaussian sampling, adaptive sampling space and nearest neighbor search strategy. At the same time, the smoothing post-processing approach of the spline curve is employed to enable the planning results to be executed by the intelligent vehicle, since the result of original RRT programming contains a large number of twists and turns leading to path curvature discontinuity.

A. Gaussian sampling strategy

Random sampling function is traditional sampling method for the basic RRT, which means the extended probability is equal for every point in the configuration space. However, considering the limitation of car's minimum turning radius, the probability of expansion in front of the locomotive should be higher than that in the rear. Therefore, the gaussian sampling will be more consistent with actual driving behavior. As shown in Fig.6, the sampling node locates in the shaded area of a fan.

$$\begin{cases} S_x = x_0 + r\cos\theta \\ S_y = y_0 + r\sin\theta \end{cases} \tag{13}$$

where (S_x, S_y) is the sampled node, (x_0, y_0) is reference node and (r, θ) is gaussian parameters, which is defined by (14)

$$\begin{cases} r = \sigma_r r_{rand} + r_0 \\ \theta = \sigma_\theta \theta_{rand} + \theta_0 \end{cases} \tag{14}$$

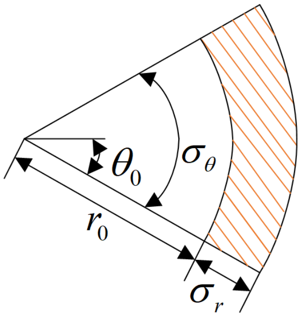


Fig.6 Gaussian sampling distribution.

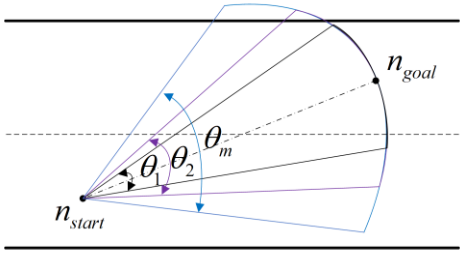


Fig.7 Adaptive sampling space

where (r_0, θ_0) is offset parameter and (σ_r, σ_θ) is gaussian standard deviation, and (r_{rand}, θ_{rand}) is random parameters.

B. Adaptive sampling space

The smaller the sampling space, the faster the expansion to the target point will be. However, this may make it impossible to extend to the target point in a complex environment. In order to improve the adaptability of the algorithm to the environment, the success factor ε (ε >0) is introduced in the offset angle θ as defined in Equation (15).

$$\theta_i \approx 0.5\varepsilon\theta_0 \quad ,\varepsilon \ge 1 \tag{15}$$

The success factor ε could be adaptive adjustment not only according to the complexity of the environment that the perception system is telling about, but also based on the success rate of the previous extension and the number of extensions. Equation (15) employs the approximately equal to denote generalized equals, as the maximum configuration space is a rectangle map instead of a 360-degree fan.

C. Nearest neighbor search strategy

The new node puts into RRT tree should be linked to the nearest node in RRT path tree after sampling. The nearest node is searched by the metric algorithm. It is classical to employ Euclidean distance as the metric. However, due to the constraint of vehicle's maximum steering angle, the steering angle between two nodes should also be considered as a smaller angle meaning a smoother path. As illustrated in Fig.8, the node $\boldsymbol{n}_1$ is nearest node of new sampling node $\boldsymbol{n}$ according to Euclidean distance. However, the steering angle from node $\boldsymbol{n}_{new}$ to node $\boldsymbol{n}_1$ is larger leading to the path $\boldsymbol{s}_1$ connected $\boldsymbol{n}_{new}$ and $\boldsymbol{n}_1$ is sharper. To make the path more executable for vehicle, the traditional metric function defined by (16) is modified as (17).

$$d_{Eu}(X_1, X_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \tag{16}$$

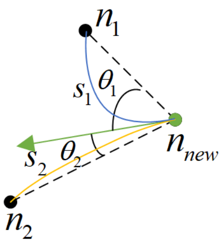


Fig.8 Schematic diagram of metric.

$$d_{Eu}(X_1,X_2)=\sqrt{(x_1-x_2)^2+(y_1-y_2)^2}+\theta \tag{17}$$

IV. Velocity Planning

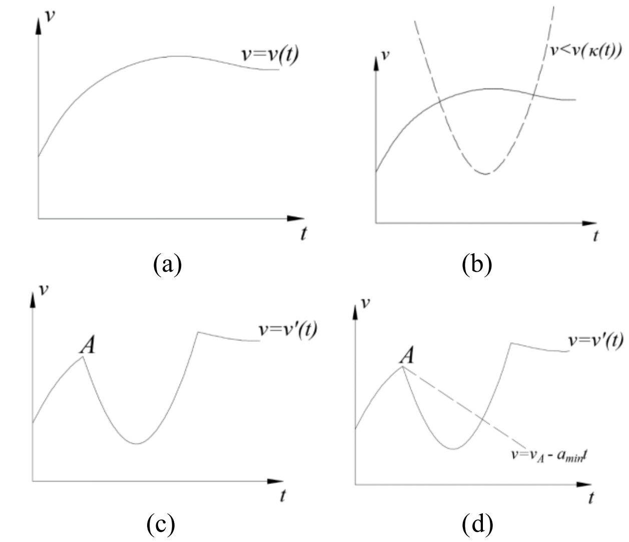
After path generated, the velocity profile should be computed obeying to boundary constraint (such as start speed and goal speed), actuator saturations (like maximum speed and maximum acceleration) and stability constraint (maximum latitudinal acceleration), which are listed as follows:

$$\begin{cases} v(t_0)=v_0 \\ v(t_g)=v_g \\ v_{\min}\le v(t)\le v_{\max} \\ a_{\min}\le \dot{v}(t)=a(t)\le a_{\max} \\ |a_y(t)|=v(t)^2|k|\le a_{y\max}=0.4\mu g \end{cases} \tag{18}$$

Cubic polynomials describe velocities as follow:

$$v(t)=at^3+bt^2+ct+d \tag{19}$$

Firstly, based on start speed and target speed, the velocity profile as shown in Fig.9 (a). Secondly, according to maximum latitudinal acceleration limitation (dash line) as shown in Fig.9 (b), the velocity curve can be obtained as shown in Fig.9 (c). Finally, combined with the actuator acceleration limit, the final velocity curve can be obtained as shown in Fig.9(d).



(a)

(b)

(c)

(d)

Fig.9 Velocity profile generation.

TABLE I. Vehicle Parameters

Parameter	***Value***	***Parameter***	Value
Bus Length	10.496 m	Bus Width	2.49 m
Max. Curvature	0.08 (1/m)	Max. Speed	19 m/s
Max. Acc	2.0 m/s^2	Min. Acc	-4.9 m/s^2

V. Experimental Results and Analysis

Simulations were conducted in the C++ 11/Linux and executed on a Jetson TX2 that runs at HMP Dual Denver 2/2 MB L2 + Quad ARM® A57/2 MB L2. The Vehicle parameters are listed as Table I.

A. B-spline-based lane change trajectory planning

The scenario 1 of Lane change is as shown in Fig.8, the start point is (0,0,0,15), the goal point is (25,3.5,0,10). The static obstacle locates in (20, 0.7) with length of 4.5m and width of 1.4m. The curvature for trajectory is continuous, as shown in Fig.9, which has maximum of 0.05729/m (less than limitation of 0.08/m), and the curvature rate is less than 0.02/m^2 as shown in Fig.10.

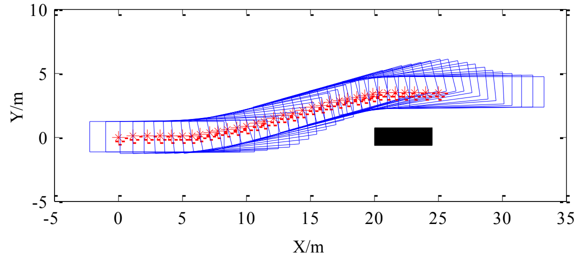


Fig.8 Lane change scenario 1

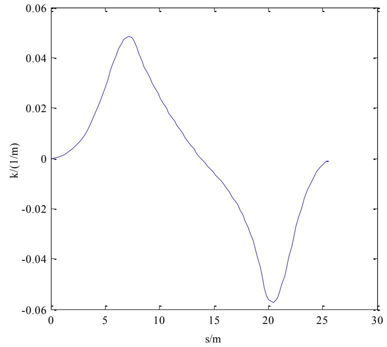


Fig.9 Curvature of path for lane change scenario 1

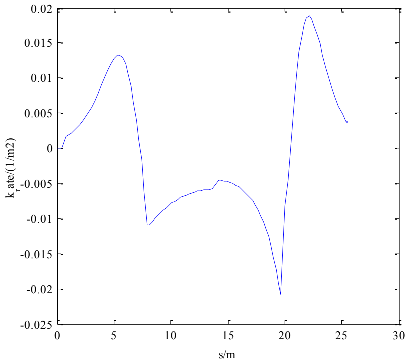


Fig.10 Curvature rate of path for lane change scenario 1

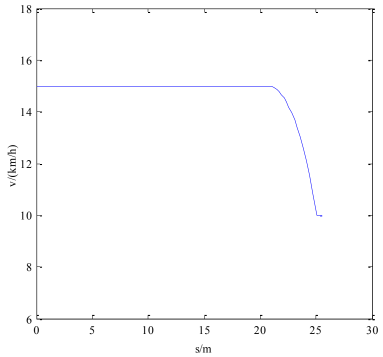


Fig.11 Velocity profile for lane change scenario 1

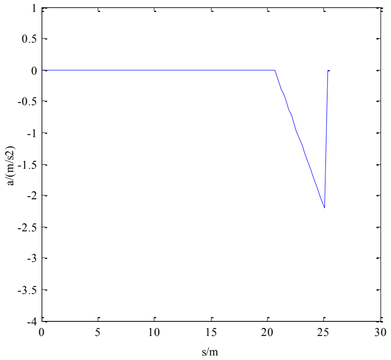


Fig.12 Acceleration profile for lane change scenario 1

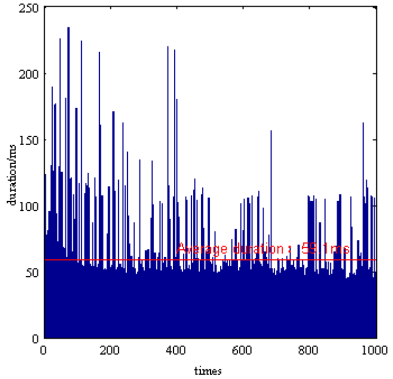


Fig.13 Duration for lane change scenario 1

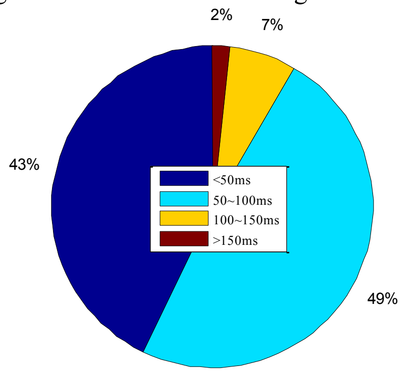


Fig.14 Duration distribution for lane change scenario 1

As shown in Fig.11, the velocity profile is continuous with minimum acceleration less than 2.20 m/s^2 (less than limitation of 4.9 m/s^2). In order to better analyze the planned time performance, we made a total of 1000 rolling computations. Although the mean duration for trajectory generating is 59.1ms (less than our goal of 100ms), however, there are still plenty of times over 100ms, even up to 250ms as shown in Fig.13. And as illustrated in Fig.14, the generating duration has the probability of 9% to exceed 100ms. But all the profile for lane change means B-spline-based method could generate curvature continuous and meet the limitation for motion.

B. RRT-based lane change trajectory planning

The scenario 2 of Lane change is as shown in Fig.15, the start point is (0,0,0,15), the goal point is (25,3.5,0,10). The stop criterion for RRT is $|\Delta x| \leq 1.5m, |\Delta y| < 0.01m$ and maximum iterations are 1500. The static obstacle locates in (17, 0.7) with length of 4.5m and width of 1.4m.

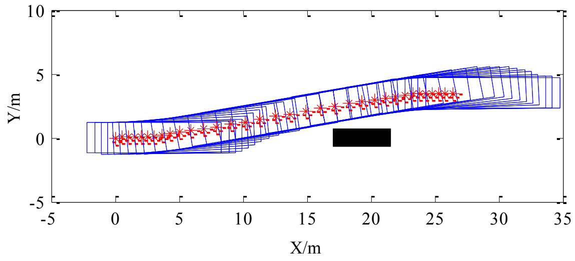


Fig.15 Lane change scenario 2

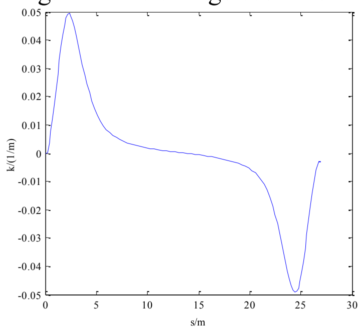


Fig.16 Curvature of path for lane change scenario 2

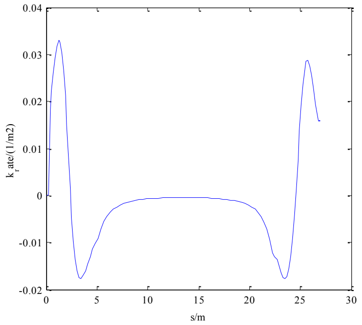


Fig.17 Curvature rate of path for lane change scenario 2

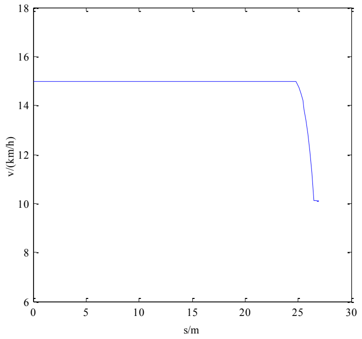


Fig.18 Speed for lane change scenario 2

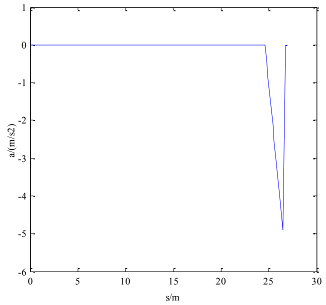


Fig.19 Acceleration for lane change scenario 2

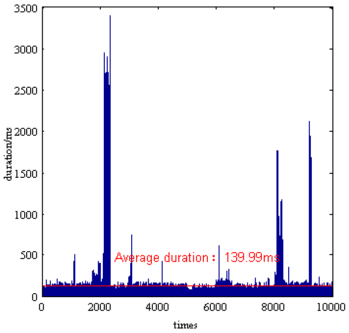


Fig.20 All duration for lane change scenario 2

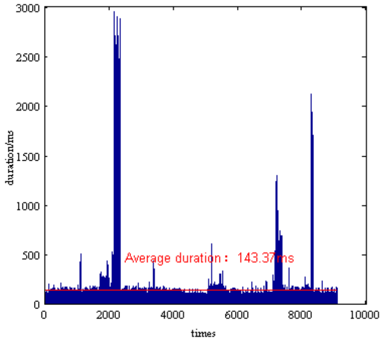


Fig.21 Success duration for lane change scenario 2

The curvature for trajectory is continuous, as shown in Fig.16, which has maximum of 0.050/m (less than limitation of 0.08/m), and the curvature rate is less than 0.033/m2 as shown in Fig.17. As shown in Fig.18, the velocity profile is continuous with minimum acceleration very close to the limitation of 4.9 m/s2. As shown in Fig.20, the mean duration for all trajectory generating is 139.99ms (more than our goal of 100ms), and there are plenty of times over 100ms, even up to 3500ms. As shown in Fig.21, for success planning, the mean duration for all trajectory generating is 143.37ms (more than our goal of 100ms), and there are wide variety of times over 100ms, even up to 3000ms.

In order to better analyze the planned time performance, we made a total of 10000 rolling computations. And as illustrated in Fig.22, the generating duration has only 21% probability to meet the requirement of less than 100ms. There is 92% probability to generate trajectory less than 200ms. However, according to success, the generating duration less than 100ms reducing to 15%, as shown in Fig.23, despite the probability less than 200ms maintaining 92%. Moreover, the success rate could only be achieved at 91%. But all the profile for lane change means RRT-based method could generate curvature continuous and meet the limitation for motion.

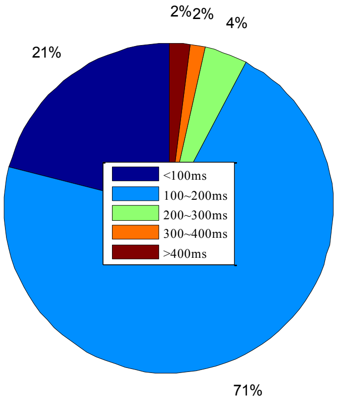


Fig.22 All duration distribution for lane change scenario 2

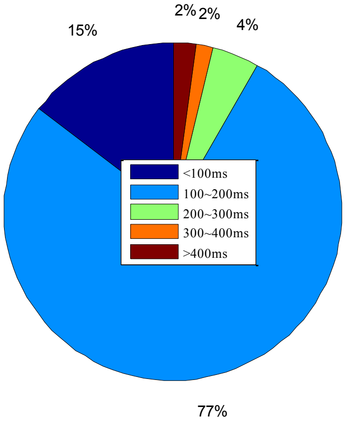


Fig.23 Success duration distribution for lane change scenario 2

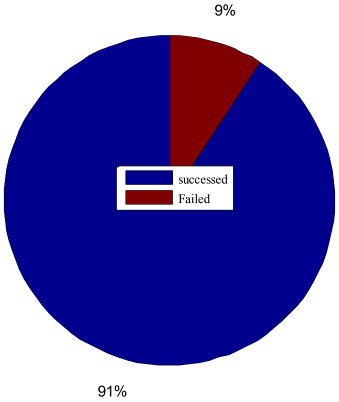


Fig.24 Success rate for lane change scenario 2

VI. Conclusion

In order to improve the robustness of the motion planning algorithm avoiding the dilemma of no way out, a novel trajectory planning method is proposed in this paper for lane change of autonomous vehicle. We present a trajectory monitoring strategy to keep robust in lane change scenario, which generates the lane change trajectory and the monitoring trajectory at the same time. If the former does not produce a safe trajectory or is time out, the monitoring trajectory will be taken as the result output. Otherwise, the output is lane change trajectory. To meet the constraints of vehicle's motion and real-time requirements, B-spline-based method will be employed to plan a continuous curvature path for lane change. And RRT-based method works as a supplement for keeping completeness of lane change plan algorithm. Then the monitory trajectory mainly obeys collision-free requirements, which computes deceleration that keeps vehicle stability. The results illustrate that both B-spline-based method and RRT-based could generate curvature continuous and meet the limitation for motion, however, both have the possibility of timeout. Especially, there are challenge to the success rate as environment becomes more complex. Future work will be focus on real vehicle implementation and testing. And harvest may not be small, if we put efforts on the study of more robust functional safety strategy.

Acknowledgment

This work is supported by the Project supported by the National Key Research and Development Program (Grant no. 2016YFB0100901), the Shanghai Municipal Commission of Economy and Informatization (Grant no. 2018RGZN02050), and the Science and Technology Commission of Shanghai (Grant no. 17DZ1100202; 16DZ1100700). The authors also thank the assistance from other people of the school of Automotive Studies, Tongji University, the College of Electronics and Information Engineering, Tongji University and the SAIC Motor Corporation Limited.

References

[1] B. Paden, M. Cap, S. Z. Yong, D. Yershov, et al. "A Survey of Motion Planning and Control Techniques for Self-driving Urban Vehicles." IEEE Transactions on Intelligent Vehicles 1.1(2016):33-55.

[2] D. Gonzalez, J. Perez, V. Milanes, and F. Nashashibi. "A Review of Motion Planning Techniques for Automated Vehicles." IEEE Transactions on Intelligent Transportation Systems 17.4(2015):1-11.

[3] S. M. Veres, L. Monar, N. K. Lincoln, and C. P. Morice. "Autonomous vehicle control systems—a review of decision making." Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering 225.2 (2011): 155-195.

[4] D. Lenz, T. Kessler, and A. Knoll. "Stochastic model predictive controller with chance constraints for comfortable and safe driving behavior of autonomous vehicles." Intelligent Vehicles Symposium 2015.

[5] J. Kim, K. Jo, W. Lim, and M. Sunwoo. "A probabilistic optimization approach for motion planning of autonomous vehicles." Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering 232.5 (2018): 632-650.

[6] C. Hubmann, M. Backer, D. Althoff, D. Lenz, et al. "Decision making for autonomous driving considering interaction and uncertain prediction of surrounding vehicles." 2017 IEEE Intelligent Vehicles Symposium (IV) IEEE, 2017.

[7] W. Lim, S. Lee, M. Sunwoo, K. Jo. "Hierarchical Trajectory Planning of an Autonomous Car Based on the Integration of a Sampling and an Optimization Method." IEEE Transactions on Intelligent Transportation Systems 19.2(2018):1-14.

[8] F. M. Marchese. "Multiple mobile robots path-planning with MCA." null. IEEE, 2006.

[9] D. Ferguson, M. H. Thomas, and L. Maxim. "Motion planning in urban environments." Journal of Field Robotics 25.11-12 (2008): 939-960.

[10] D. Ferguson, and S. Anthony. "Using interpolation to improve path planning: The Field D* algorithm." Journal of Field Robotics 23.2 (2006): 79-101.

[11] R. V. Cowlagi, and T. Panagiotis. "Multi-resolution path planning: theoretical analysis, efficient implementation, and extensions to dynamic environments." Decision and Control (CDC), 2010 49th IEEE Conference on. IEEE, 2010.

[12] J. Ziegler, M. Werling, and J. Schroder. "Navigating car-like robots in unstructured environments using an obstacle sensitive cost function." 2008 IEEE Intelligent Vehicles Symposium. IEEE, 2008.

[13] M. Likhachev, D. Ferguson, D. Gordon, and A. Stenz. "Anytime search in dynamic graphs." Artificial Intelligence 172.14 (2008): 1613-1643.

[14] A. Stentz. "Optimal and efficient path planning for partially-known environments." ICRA. Vol. 94. 1994.

[15] M. Liang, J. Xue, K. Kawabata, J. Zhu, et al. "Efficient sampling-based motion planning for on-road autonomous driving." IEEE Transactions on Intelligent Transportation Systems 16.4 (2015): 1961-1976.

[16] J. H. Ryu, D. Ogay, S. Bulavintsev, H. Kim, et al. "Development and experiences of an autonomous vehicle for high-speed navigation and obstacle avoidance." Frontiers of Intelligent Autonomous Systems. Springer, Berlin, Heidelberg, 2013. 105-116.

[17] S. Karaman, M. R. Walter, A. Perez, E. Frazzoli, and et al. "Anytime motion planning using the RRT*." Institute of Electrical and Electronics Engineers, 2011.

[18] V. R. Desaraju, and J. P. How. "Decentralized path planning for multi-agent teams with complex constraints." Autonomous Robots 32.4 (2012): 385-403.

[19] S. Glaser, B. Vanholme, S. Mammar, D. Gruyer, et al. "Maneuver-based trajectory planning for highly autonomous vehicles on real road with traffic and driver interaction." IEEE Transactions on Intelligent Transportation Systems 11.3 (2010): 589-606.

[20] A. Broggi, P. Medici, P. Zani, A. Coati, et al. "Autonomous vehicles control in the VisLab intercontinental autonomous challenge." Annual Reviews in Control 36.1 (2012): 161-171.

[21] J. W. Choi, R. Curry, and G. Elkaim. "Path planning based on bézier curve for autonomous ground vehicles." World Congress on Engineering and Computer Science 2008, WCECS'08. Advances in Electrical and Electronics Engineering-IAENG Special Edition of the. IEEE, 2008.

[22] T. Berglund, A. Brodnik, H. Jonsson, M. Staffanson, et al. "Planning smooth and obstacle-avoiding B-spline paths for autonomous mining vehicles." IEEE Transactions on Automation Science and Engineering7.1 (2010): 167-172.

[23] D. Dolgov, S. Thrun, M. Montemerlo, and J. Diebel. "Path planning for autonomous vehicles in unknown semi-structured environments." The International Journal of Robotics Research 29.5 (2010): 485-501.

[24] T. Gu, J. Snider, J. M. Dolan, and J. W. Lee. "Focused trajectory planning for autonomous on-road driving." Intelligent Vehicles Symposium (IV), 2013 IEEE. IEEE, 2013.

[25] J. Ziegler, P. Bender, T. Dang, and C. Stiller. "Trajectory planning for Bertha—A local, continuous method." Intelligent Vehicles Symposium Proceedings, 2014 IEEE. IEEE, 2014.

[26] B. Li, K. Wang, and Z. Shao. "Time-optimal maneuver planning in automatic parallel parking using a simultaneous dynamic optimization approach." IEEE Transactions on Intelligent Transportation Systems 17.11 (2016): 3263-3274.

[27] B. Li, Bai, Y. Zhang, and Z. Shao. "Spatio-temporal decomposition: a knowledge-based initialization strategy for parallel parking motion optimization." Knowledge-Based Systems 107 (2016): 179-196.

[28] M. Elbanhawi, M. Simic, and R. N. Jazar. "Continuous path smoothing for car-like robots using B-spline curves." Journal of Intelligent & Robotic Systems 80.1 (2015): 23-56.

[29] M. K. Gao, Y. M. Chen, Q. Liu, C. Huang, et al. "Three-dimensional path planning and guidance of leg vascular based on improved ant colony algorithm in augmented reality." Journal of medical systems39.11 (2015): 133.