[[1]](#footnote-1)

DL-AMP and DBTO: An Automatic Merge Planning and Trajectory Optimization and its Application in Autonomous Driving

Yuncheng Jiang, Qi Lin

*Abstract*— This paper presents an automatic merging algorithm for autonomous driving vehicle, which decouples the specific motion planning problem into a Dual-Layer Automatic Merge Planning (DL-AMP) and a Descent-Based Trajectory Optimization (DBTO). This work leads to great improvements in finding the best merging opportunity, lateral and longitudinal merging control, trajectory postprocessing and driving comfort. Our algorithm’s efficiency, adaptiveness and robustness have been tested and validated both in simulations and on-road tests.

# INTRODUCTION

In the past several decades, autonomous driving has made a great process extending the ability of autonomous vehicle from Adaptive Cruise Control (ACC) and Lane Keeping Assistance (LKA) to more complex functions such as automatic lane changing, off-road navigation, stopping at traffic light, etc. [1][2]. In merging scenarios, human can predict other vehicles intentions and react intelligently to make the best merge choice. To deal with such scenario, prediction and planning should be well coupled considering driving efficiency, safety, and comfort. Therefore, a robust and intelligent algorithm that can interact with human-operated traffic on freeways is still under research and development. There are mainly two approaches in dealing with automatic merge problem:

One category is prediction and planning coupled method which convert the prediction and planning problem into a convex optimization form, and prediction information is transferred into inequality constraints [3]. Mixed Logical Dynamics (DML) [4][5] is also applied in solving this optimization problem. [6] uses quintic polynomials in merge planning, while polynomial parameters are chosen as optimization targets. This approach usually lacks computational efficiency, and its robustness is not well guaranteed. The other category is prediction and planning decoupled method which first find the best merging opportunity, and then plan a merge maneuver. [7] uses Prediction and Cost Function Based (PCB) method to find the best merge opportunity, while this planner must run continuously throughout the whole merge process, and its acceleration and velocity outputs are discontinuous. [8] uses a rule-based planner to decide when to merge based on its fixed definition of environment safety, but it may fail when faced with complex traffic scenarios.

In terms of motion planning, there are also two major approaches: path/speed decoupled method and path/speed coupled method. In decoupled method, polynomial curvature spiral is generated, and spatial trajectory is solved by using Lagragian method [9][10][11]. Cubic polynomial is also used to generate spatial trajectory [12][13]. In the two methods, piece -wise velocity profile is then generated satisfying some road and trajectory constraints. In merge problem, path/speed decoupled methods do not work well since merging maneuver require that the vehicle reaches some position at some specific time stamp (position/time strictly coupled). In coupled method, quintic polynomials are generated longitudinally and laterally, and they are combined to generate spatial and temporal trajectories [14][15]. Such method, however, does not consider comfort and merge opportunity when deal with merge problem. Therefore, it may lead to uncomfortable lateral acceleration in lane changing, and even lead to danger if unreasonable merge opportunity is selected.

In trajectory postprocessing, gradient descent method is used in [16] to optimize trajectory curvature, but it fails to do collision checking after smoothing. A Dual-Loop Iterative Anchoring Path Smoothing (DL-IAPS) is used to smooth trajectory generated by hybrid A\* algorithm, by converting the smoothing problem into a convex optimization problem. The algorithm has an average running time about 0.18s- 0.21s [17]. However, in merging problem, where vehicles usually move at high speed, optimization-based algorithm may degrade system instantaneity.

In this paper, we propose a novel path/speed coupled method for automatic merge planning. More specifically, we decoupled the method into two hierarchical steps, including, Dual-Layer Automatic Merge Planning (DL-AMP) and a Descent-Based Trajectory Optimization (DBTO). Our method addresses above mentioned issues with the following advantages:

1) **The Best Merging Opportunity**: In our DL-AMP, the first layer is a prediction and cost function-based algorithm which can find the best merging opportunity. The difference between our method and that in [18] are: 1) our method is only used in find the best merging opportunity, when the autonomous vehicle begins to merge, the second layer algorithm will be trigger. Therefore, the output continuity is guaranteed by the second layer algorithm. Meanwhile, since the task is much easier for the first layer algorithm, it has less cost function parameters to tune. In our method, candidate ego vehicle acceleration and time are sampled in some ranges. Based on the sampled acceleration and time, constant acceleration prediction model is used to predict the relative distance and velocity between ego vehicle and other traffics. Cost function that emphasis driving efficiency and comfort is designed, and the maneuver that satisfies safety hard constraints with the minimum cost is chosen.

2) **Efficient optimization- and sampling- based method**: although quintic polynomial sampling method has been used in [14], it does not well deal with merge scenario motion planning. EM planner [23] generate optimal trajectory by solve iteratively dynamic and quadratic programming problems (DP & QP) with an average of 259*ms*. In our DL-AMP, the second layer algorithm is an optimization- and sampling- based path/speed quintic polynomial trajectory generation method. We only sample on time, and under each fixed time, we convert the trajectory generation into a constrained quadratic programming (QP) problem and solve it efficiently. After suboptimal trajectory is generated under each sampling time, another cost function is formulated that considers sampling time. The optimal trajectory is the one with the lowest cost function. Our optimization- and sampling- based method does not have DP process and piece-wise polynomial trajectory is simplified into a single trajectory, thus greatly reducing computation effect. Time *t* is selected as a sampling variable to keep the objective function in quadratic form which can be efficiently solved by robust QP solver.

The longitudinal planning is divided into distance planning and velocity planning to adapt to difference merge scenarios. In lateral planning, we divide the scenarios into four categories, and different desired distance constraints are selected under different scenarios. To improve efficiency, the constraints of sampling time, desired lateral offset in lateral planning and desired distance and velocity in longitudinal planning are well selected (Sec-II-B) based on road geometry, vehicle dynamic constraints and human driver statistics.

3) **Driving Comfort and Control Feasibility**: To improve driving comfort and control feasibility. We do trajectory smoothing after optimal trajectory is generated. A gradient based method that considers trajectory smoothness and curvature is us ed to reduce trajectory discontinuity and turning radius which results in reduced lateral acceleration and wheel turning angles in control. Since we use DL-AMP to find the best merge opportunity, free space is ensured at the very beginning, and collision checking is not necessary after smoothing. Unlike [19], we also modify our gradient based method to deal with exploding and vanishing gradient problems.

This paper is organized as follows: DL-AMP and DBTO are illustrated in Section II and III, respectively. The simulation and on-road tests results are shown in Section IV.

# Dual-Layer Automatic Merge Planning

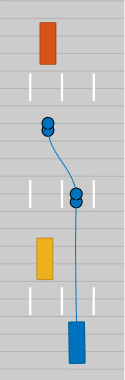
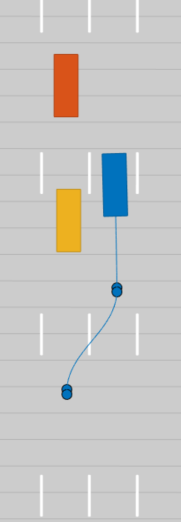
In this section, we introduce the two parts of DL-AMP: In III-A we introduce how to find the best merging opportunity and in III-B we introduce merging trajectory generation. The overall algorithm is shown in 1.

## A. finding the best merging opportunity

Generally, the actual position of the ego vehicle may not be a good one for us to immediately trigger a merging maneuver, so the first layer of AMP is also called adjustment phase, in which the best merging opportunity will be sampled according to the PCB algorithm to guarantee both efficiency, comfort and safety. We first introduce the merging scenarios we are going to handle in this paper, and then build up prediction models for both ego vehicle and other vehicles of target merging lane by different movement hypotheses. The desired acceleration of ego vehicle and the time of adjustment is then sampled, and the performance is evaluated according to several well-designed cost functions to ensure a reasonable initial condition for the merging phase.

*1) Scenario:* in this paper, we mainly discuss 4 typical scenarios depicted in Fig. X which would be frequently met in real world. In the Case 1 in Fig. X, there are two vehicles driving in adjacent lane and the ego vehicle is asked to merge into the position between them. The initial position of ego vehicle is free, it can be either at the rear side of the rear adjacent vehicle, at the front of the front adjacent vehicle or between the adjacent vehicles. The ego vehicle is supposed to first adjust itself in longitudinal direction only to a proper state (including position and velocity), and then trigger the merging maneuver. The proper state may locate near the middle of the adjacent vehicles in longitudinal direction with a suitable relative velocity to ensure a low collision risk during the following merging phase.

Case2 and Case3 are similar scenarios. In Case2, the free space of adjacent lane locates at the rear of the traffic. Ego vehicle should move and merge to the end of adjacent traffic. In Case3, ego vehicle are supposed to move and merge to the front of adjacent traffic.

Case1 Case2 Case3

*2) Prediction:* the prediction of ego vehicle is designed as a simple constant acceleration model to mimic human’s constant pedal behavior in adjustment phase.

We denote dEgo(t) and vEgo(t) as the distance and velocity of ego vehicle with predicted time t under a certain constant acceleration a from vEgoInit as its initial velocity, which can be written as

(1)

(2)

As to other vehicles in target lane, we assume that their acceleration would change exponentially from current value toward zero and then keep constant velocity. This hypothesis is derived by the assumption that vehicles in highway are intended to drive with a constant speed in long term even if they have relatively large absolute acceleration value in a short time period.

3) Cost Functions: the cost functions are used to evaluate both efficiency, comfort and safety situation at sampled time t. Cost functions are implemented from four dimensions (distance to other vehicles, relative velocity to other vehicles, total time and acceleration value) to guarantee that ego vehicle would move to a suitable initial position and velocity in a proper time with a comfortable drivability. We take Case1 as an example to demonstrate the implementation of cost functions shown in Figure X.

CDistance is used to evaluate the longitudinal distance gap to the adjacent vehicles in target lane at prediction time t. We use distance percentage to calculate the cost because the distance gap between two adjacent vehicles is uncertain, so a fixed distance threshold is not suitable. Distance percentage of 0% means ego vehicle is at the same longitudinal position of the rear adjacent vehicle, and 100% means ego vehicle is at the same longitudinal position of the front adjacent vehicle, and is bounded to [0,1]. The cost of distance reaches its minimum when the percentage lies between 40% and 60%, which means the ego vehicle is close to the middle of the two adjacent vehicles. It increases to its maximum when the final adj position is too close to any of the vehicles. The left upper figure of Fig. X shows the relationship between CDistance and the distance percentage.

CTTC evaluates how safe the initial position is to trigger a merging action at prediction time t by means of calculating TTC (time to collision). The safety evaluation is coupled with both relative velocity and distance, so we use TTC instead of relative velocity to design this cost function. The cost is calculated as shown in right upper figure in Fig. X for both adjacent vehicles in Case1 and the total cost is calculated as their mean value:

CTime describes the total adjustment time, since we always prefer to get the initial merging position as soon as possible.The left lower figure in Fig. X shows CTime with the increase of total adjustment time.

CAccel is determined by the constant absolute acceleration value during adjustment phase. For comfort purpose, a small absolute acceleration value is preferred. As shown in the right lower figure in Fig. X, The absolute acceleration value in the area of [0,1] is the most preferred, and its cost increases rapidly in the area of [1,2].

The final cost for adjustment is calculated as:



Fig. X

PLEASE FILL IN THE PLANNER DESIGN

## B. merging trajectory generation

We use optimization- and sampling- based method to generate optimal quintic polynomials longitudinally and laterally by solve a quadratic programming problem with its standard form:

We evaluate the candidate trajectories using objective function considering driving efficiency, comfort, and safety. Equality and inequality constraints are applied to guarantee fixed initial and terminal states, vehicle dynamics and driving comfort. To make merge maneuver more like human behaviors, we calculate in prior a reasonable sampling time range that satisfies comfortable lateral acceleration and curvature limits which is shown if Fig.1. It affects the constraints of terminal desired distance in longitudinal distance planning and the constraints of terminal velocity in longitudinal velocity planning. We divide merging scenarios into four categories, which is shown in Fig.1. under different scenarios, desired longitudinal distance is calculated differently.

*1) lateral planning:* according to human driving statistics, we limit sampling time in a reasonable range to improve sampling efficiency. Lateral acceleration is the most critical factor that affect driver comfort, and trajectory curvature is also an important factor that affect the performance of lateral controller. Based on the feature of quintic polynomial, we calculate a reasonable time range that satisfy both which is shown in Fig. x. the sampling time range is constrained between 3.5s and 6.0s.

Chart, surface chart

描述已自动生成Chart, surface chart

描述已自动生成

Fig. x:

We calculate the maximum lateral acceleration a quintic polynomial can generate with a constant lateral offset (in China, standard road width 3.75m) and longitudinal velocity varying from 36km/h to 108 km/h. We find that the lateral acceleration profile is like a saddle with its peak values (time fixed) ranging between 68km/h and 76km/h. According to driver statistics, we find that lateral acceleration should be less than 0.15g, the minimum time is therefore limited at 4.0s. The curvature graph is then used to check if the maximum curvature is too aggressive with time equaling 4.0s. To ensure merging efficiency, we limit maximum time at 6.0s which is like human drivers’ behavior.

In lateral direction, the objective function is denoted by:

where , and is penalty weights on different parameters. is lateral jerk of the trajectory, and is lateral offset of the trajectory w.r.t the reference line. Unlike cost function in [6] which contains more than ten parameters, we want to select cost function terms as less as possible, as heavy parameter calibration is always time consuming and does not appeal to engineering application. In terms of comfort, we find that jerk is the most direct factor that affect driver comfort; thus, we only keep jerk term, neglecting lateral velocity and acceleration terms. The curvature-related terms are also neglected since we not only calculate in priori a reasonable sampling time range that ensures proper trajectory curvature, but also do trajectory smoothing in DBTO to optimize curvature. The consecutivity term is neglected in merge scenarios where there exists no symmetry. Less cost function terms can improve computation efficiency and reduce parameter calibration burden. For the sake of application, the objective function is converted into a discretization form:

we define the initial state constraint = [,, ,] where is ego vehicle’s lateral offset with respect to reference line. is ego vehicle’s lateral velocity, is ego vehicle’s lateral acceleration, and is current time which is always assumed to be zero. The terminal constraints are defined as = [,, , ] where is terminal lateral offset and is terminal time. We let  *= = 0*, as we always want the terminal lateral velocity and lateral acceleration equal to zero. And it becomes equality constraints. is an inequality constraint which is limited between the boundary of target road denoted by *RW*. Middle points constraint checking is performed on these discretized points: . The constraints are as follows:

The details of the constrained QP problem are shown in [22].

*2) longitudinal planning:* longitudinal planning is divided into distance planning and velocity planning. Similar with lateral planning, we use quintic polynomial in distance planning and quantic polynomial in velocity planning. The optimal trajectory is achieved by solve similar constrained QP problem.

The objective functions for distance planning and velocity planning are denoted, respectively, by:

where and .

And their discretization form are as follows:

where and .

, and are respective penalty weights on different parameters. is longitudinal jerk at of the trajectory, is longitudinal offset between desired distance and terminal distance, and is longitudinal velocity offset between set speed and terminal speed.

In distance planning, we define the initial state constraints = [,, ,] where , ,and are respectively ego vehicle’s longitudinal distance with respect to reference line, longitudinal velocity, and longitudinal acceleration. is current time which is always assumed to be zero. The terminal state constraints are defined as = [,, ,] where ,, are terminal longitudinal distance, velocity, and acceleration, respectively. is the same as that in lateral planning which is fixed in each QP problem. In velocity planning, the terminal states constraints are a little bit different: = [, ,] = [, ,], we leave as unconstrainted. Middle points constraint checking is performed on these discretized points: .

Distance planning is further divided into four categories shown in Fig. X. In different categories, the desired longitudinal distance is calculated differently, we use Case I as an example for our next contents. We use an exponentially decreasing acceleration model to predict the vehicle kinematics:

By solve boundary value problems, the position, velocity and acceleration of both vehicle a and b (,,,,,) can be easily achieved. We omit the calculation for brevity and the details are shown in [22].

In real traffic, human drivers are more likely to move closer to the front vehicle when merging, therefore we define:

.

The first order and second order derivatives of are as follows:

In distance planning the constraints are as follows:

In longitudinal velocity planning, the constraints are as follow:

where

The complete formulation of the optimization problem is as follow:

subject to lateral constraints and longitudinal distance constraints or velocity constraints according to different longitudinal scenarios. After optimal trajectory is generated at each sampling time T, total cost function is formulated and the trajectory with minimum cost is selected as global optima:

Although we can formulate the total cost function in another form that contains *t*:

The first two term is in quadratic form, while t become an optimization variable, the objective function therefore become a nonlinear form, and we can only use gradient based method or nonconvex optimization methods to deal with the problem. The computation time is will greatly increase, and we will lose algorithm efficiency in engineering application.

# Descent based trajectory optimization

In this section, to further reduce the curvature and heading angle of the optimal quintic polynomial, we introduce our descent-based trajectory optimization, and the overall algorithm is shown in 1. Our method is a post-processing of trajectory generated by DL-AMP. It is based on gradient descent which is an iterative optimization algorithm. The coordinate of each point is updated iteratively in direction of the negative derivative of objective function. In our method the objective function is weighted sum of three terms: curvature, smoothness, and straightness which is given by:

, , and are three penalty weights on different terms. The first term is penalty on curvature. The second term is penalty on straightness, and the third term is penalty on smoothness where is defined as . Post-processing of trajectory generated by Hybrid A\* is introduced in [20][21], but quintic polynomial has less noise than trajectory generated by Hybrid A\* algorithm, therefore, in real tests, we have not seen expected gradient descent from the smoothness term. Therefore, we introduce another straightness term to deal with vanishing gradient problem. We also introduced a buffer band which is close to original quintic polynomial to avoid bad effect of exploding gradient. To prevent trajectory points’ heading angles that are close to initial and terminal locations from moving considerably, variable penalty weights which have Gaussian distribution are introduced. To guarantee the robustness of the trajectory optimization algorithm, we set three stopping criteria: 1). Maximum iteration number; 2). Buffer band; 3) Minimum curvature. The maximum iteration number ensures that the optimization algorithm is forced to stop at fixed maximum allowed running period. The buffer band ensures that the optimized trajectory is near the original trajectory. It can not only check the explode gradient problem, but also keep the validity of collision free status. Minimum curvature criteria stop the algorithm once the trajectory is smooth enough for controller.

## Vanishing Gradient Problem

In objective function, we introduce smooth term. While in merge scenarios, longitudinal distance change is much longer than lateral distance change, making gradient descent in smooth term very small and the smoothing algorithm very inefficient. Even though we can use bigger descent step, but it has to adapt to different merge scenarios, and sometimes bad step may cause explode descent problem. To ensure obvious gradient descent in trajectory smoothing, we introduce another term . Although it means stretch the trajectory to a straight line, in merge scenarios, it has the same meaning as smoothness. Human driver intends to turn steering wheel as slowly and less as possible, which indicating that the merge trajectory becomes smoother and straighter. We use a simple merge scenario to show the effect of straightness term. In this case, DL-AMP generates an optimal trajectory with maximum lateral offset 3.5m and longitudinal offset 6.5m. if we only use smoothness term for trajectory optimization with descent step size 0.15 and allow 400 iterations. The result is shown in Fig. x and Fig. x. the gradient with term is at least twenty times bigger than that without term. In this case test, the iteration is much more efficient and stops at the 191st iteration saving at least a half computational time.

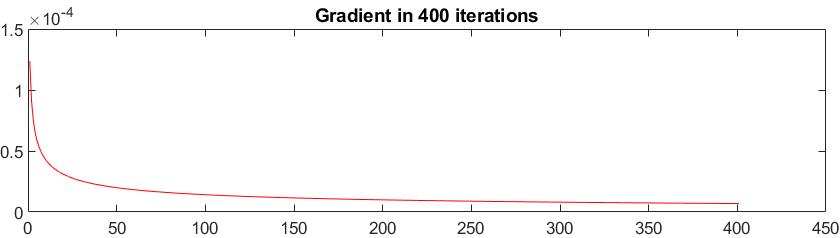


Fig.x Gradient descent without straightness term

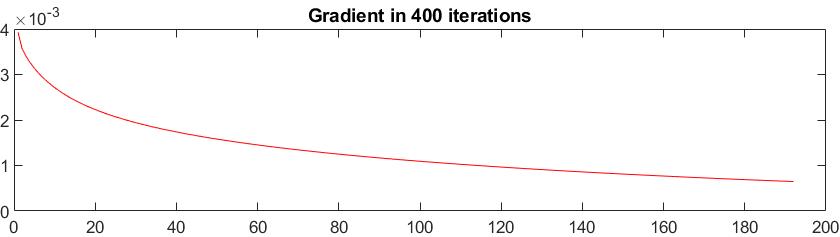
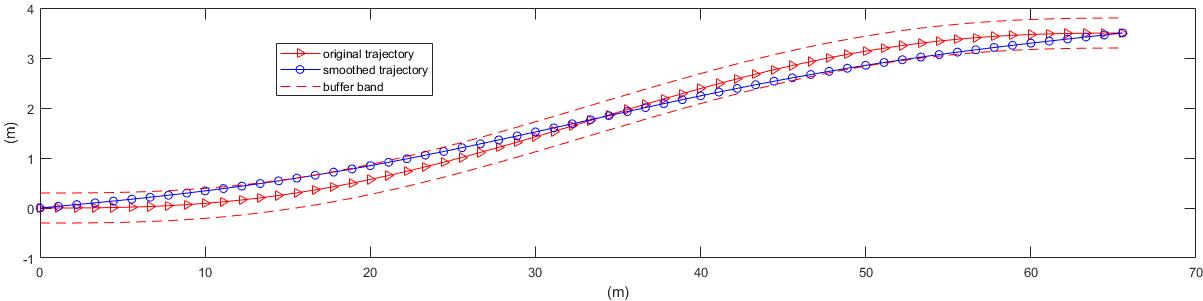


Fig.x Gradient descent with straightness term

## B. Buffer Band

Buffer band is introduced to avoid explode gradient problem and keep the validity of collision free status. As is shown in Fig. x, gradient of smoothness term explodes if a big step size is selected, thus the optimized trajectory becomes unreasonable. We use buffer band to avoid gradient explode. As is shown in Fig. x, algorithm is stopped if the points of optimized trajectory move out of the range of buffer band.



Chart, line chart, scatter chart

描述已自动生成

Fig.x: Optimized trajectory with and without constraints from buffer bands.

## C. Variable Penalty Weights

To overcome the problem of abrupt different of heading angle close to the initial and terminal points, we prevent points close to the initial and terminal locations of the trajectory from moving considerably. The gradient descent step size is modified as a variable parameter with Gaussian distribution. This variable parameter controls the step size with respect to the point index. As a result, points near the initial and terminal points have smaller gradient descent step size, thus moving less compared to those points far from the initial and terminal points. The Gaussian distribution is shown in Fig. x.

A picture containing text, fishing, archery

描述已自动生成

Fig. x: Variable weights with Gaussian distribution

## D. Gradient Descent

The gradient of each term is as follows:

1). **The derivative of smoothness term**

2). **The derivative of straightness term**

**3). The derivative of curvature term**

The partial derivative of with respect to is

Where is the angle between and

The detail of derivation of can be find in [22].

|  |  |
| --- | --- |
| **Algorithm 1**: Gradient Descent Algorithm | |
| **Variable:** | |
| id: trajectory index.  : gradient of trajectory points.  : the maximum absolute value of trajectory curvature  : the maximum absolute value of offset between original trajectory and smoothed trajectory | |
| **Parameters:** | |
| : gradient descent step size of smoothness term.  : gradient descent step size of straightness term.  : gradient descent step size of curvature term.  : the standard deviation of Gaussian distribution.  *L*: the length of trajectory points.  *P*: original trajectory points [].  MAX\_ITER: maximum iteration number  *B*: buffer band.  : satisfactory curvature. | |
| 1 | *COND 1* |
| 2 | **While** *COND = 1* **do** |
| 3 | **for** all **do** |
| 4 |  |
| 5 | *GaussianDistribution (id, , L)* |
| 6 |  |
| 7 |  |
| 8 |  |
| 9 |  |
| 10 |  |
| 11 | **end for** |
| 12 |  |
| 13 | **if**  *MAX\_ITER* **then** |
| 14 | *COND 2* |
| 15 | **break** |
| 16 | **if** **then** |
| 17 | *COND 3* |
| 18 | **break** |
| 19 | **if** *B* **then** |
| 20 | *COND 4* |
| 21 | **break** |
| 22 | **end while** |

## E. Smoothing results

We use the same test parameters as section A to see the performance of DBTO. In Fig. x, the optimized trajectory is well improved in terms of heading angle and curvature. The maximum heading angle is reduced from 5.618 *deg* to 4.194 *deg*, and the maximum curvature is reduced from 0.004528 to 0.002 . The derivative of curvature is also checked to ensure its continuity is well reserved.

Diagram, engineering drawing

描述已自动生成

Fig. x: Trajectory features before and after smoothing. Gradient based smoothing improves the characteristic of trajectory heading angle and curvature. Variable weight of step length with Gaussian distribution ensures the continuity of optimized heading angle close to initial and terminal positions of original trajectory. The continuity of change rate of curvature is still guaranteed after trajectory smoothing.

# experimental results

After the text edit has been completed, the paper is ready for the template. Duplicate the template file by using the Save As command, and use the naming convention prescribed by your conference for the name of your paper. In this newly created file, highlight all the contents and import your prepared text file. You are now ready to style your paper; use the scroll down window on the left of the MS Word Formatting toolbar.

A picture containing text

描述已自动生成A picture containing text, clipart

描述已自动生成A picture containing text

描述已自动生成

## A. Authors and Affiliations

The template is designed so that author affiliations are not repeated each time for multiple authors of the same affiliation.

1. Table Type Styles

| Table Head | Table Column Head | | |
| --- | --- | --- | --- |
| Table column subhead | Subhead | Subhead |
| copy | More table copya |  |  |

a. Sample of a Table footnote. (Table footnote)

1. Example of a figure caption. *(figure caption)*

# Conclusion

A conclusion section is not required. Although a conclusion may review the main points of the paper, do not replicate the abstract as the conclusion. A conclusion might elaborate on the importance of the work or suggest applications and extensions.

References

1. Paden, B., Čáp, M., Yong, S. Z., Yershov, D., & Frazzoli, E. (2016). A survey of motion planning and control techniques for self-driving urban vehicles. *IEEE Transactions on intelligent vehicles*, *1*(1), 33-55.
2. Schwarting, W., Alonso-Mora, J., & Rus, D. (2018). Planning and decision-making for autonomous vehicles. *Annual Review of Control, Robotics, and Autonomous Systems*.
3. Shamir, T. (2004). How should an autonomous vehicle overtake a slower moving vehicle: Design and analysis of an optimal trajectory. *IEEE Transactions on Automatic Control*, *49*(4), 607-610.
4. Du, Y., Wang, Y., & Chan, C. Y. (2014, October). Autonomous lane-change controller via mixed logical dynamical. In *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)* (pp. 1154-1159). IEEE.
5. Nilsson, J., & Sjöberg, J. (2013, June). Strategic decision making for automated driving on two-lane, one way roads using model predictive control. In *2013 IEEE Intelligent Vehicles Symposium (IV)* (pp. 1253-1258). IEEE.
6. Xu, W., Wei, J., Dolan, J. M., Zhao, H., & Zha, H. (2012, May). A real-time motion planner with trajectory optimization for autonomous vehicles. In *2012 IEEE International Conference on Robotics and Automation* (pp. 2061-2067). IEEE.
7. Wei, J., Dolan, J. M., & Litkouhi, B. (2010, June). A prediction-and cost function-based algorithm for robust autonomous freeway driving. In *2010 IEEE Intelligent Vehicles Symposium* (pp. 512-517). IEEE.
8. Cong, Y., Sawodny, O., Chen, H., Zimmermann, J., & Lutz, A. (2010, September). Motion planning for an autonomous vehicle driving on motorways by using flatness properties. In *2010 IEEE international conference on control applications* (pp. 908-913). IEEE.
9. Howard, T. M., & Kelly, A. (2007). Optimal rough terrain trajectory generation for wheeled mobile robots. *The International Journal of Robotics Research*, *26*(2), 141-166.
10. Li, X., Sun, Z., Liu, D., Zhu, Q., & Huang, Z. (2014, October). Combining local trajectory planning and tracking control for autonomous ground vehicles navigating along a reference path. In *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)* (pp. 725-731). IEEE.
11. Li, X., Sun, Z., Kurt, A., & Zhu, Q. (2014, June). A sampling-based local trajectory planner for autonomous driving along a reference path. In *2014 IEEE intelligent vehicles symposium proceedings* (pp. 376-381). IEEE.
12. Li, X., Sun, Z., He, Z., Zhu, Q., & Liu, D. (2015, June). A practical trajectory planning framework for autonomous ground vehicles driving in urban environments. In *2015 IEEE Intelligent Vehicles Symposium (IV)* (pp. 1160-1166). IEEE.
13. Li, X., Sun, Z., Cao, D., He, Z., & Zhu, Q. (2015). Real-time trajectory planning for autonomous urban driving: Framework, algorithms, and verifications. *IEEE/ASME Transactions on mechatronics*, *21*(2), 740-753.
14. Werling, M., Ziegler, J., Kammel, S., & Thrun, S. (2010, May). Optimal trajectory generation for dynamic street scenarios in a frenet frame. In *2010 IEEE International Conference on Robotics and Automation* (pp. 987-993). IEEE.
15. Werling, M., Kammel, S., Ziegler, J., & Gröll, L. (2012). Optimal trajectories for time-critical street scenarios using discretized terminal manifolds. *The International Journal of Robotics Research*, *31*(3), 346-359.
16. Gu, T., Snider, J., Dolan, J. M., & Lee, J. W. (2013, June). Focused trajectory planning for autonomous on-road driving. In *2013 IEEE Intelligent Vehicles Symposium (IV)* (pp. 547-552). IEEE.
17. Zhou, J., He, R., Wang, Y., Jiang, S., Zhu, Z., Hu, J., ... & Luo, Q. (2020). Autonomous Driving Trajectory Optimization with Dual-Loop Iterative Anchoring Path Smoothing and Piecewise-Jerk Speed Optimization. *IEEE Robotics and Automation Letters*.
18. Wei, J., & Dolan, J. M. (2009, June). A robust autonomous freeway driving algorithm. In *2009 IEEE Intelligent Vehicles Symposium* (pp. 1015-1020). IEEE.
19. Dolgov, D., Thrun, S., Montemerlo, M., & Diebel, J. (2008). Practical search techniques in path planning for autonomous driving. *Ann Arbor*, *1001*(48105), 18-80.
20. Dolgov, D., Thrun, S., Montemerlo, M., & Diebel, J. (2010). Path planning for autonomous vehicles in unknown semi-structured environments. *The international journal of robotics research*, *29*(5), 485-501.
21. Mizuno, N., Ohno, K., Hamada, R., Kojima, H., Fujita, J., Amano, H., ... & Tadokoro, S. (2019). Enhanced path smoothing based on conjugate gradient descent for firefighting robots in petrochemical complexes. *Advanced Robotics*, *33*(14), 687-698.
22. Zhang, …………………………..
23. Zhang, Y., Sun, H., Zhou, J., Hu, J., & Miao, J. (2019, October). Optimal trajectory generation for autonomous vehicles under centripetal acceleration constraints for in-lane driving scenarios. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)* (pp. 3619-3626). IEEE.

1. All Authors are with Shanghai Automotive Industry Corporation, Ltd. 201, Anyan Road, Jiading, Shanghai, China. \*Corresponding author: Yuncheng Jiang [jiangyuncheng@saicmotor.com](mailto:jiangyuncheng@saicmotor.com) [↑](#footnote-ref-1)