https://doi.org/10.20965/jaciii.2021.p0064

Paper:

Path Planning Based on Improved Hybrid A* Algorithm

Bijun Tang, Kaoru Hirota, Xiangdong Wu, Yaping Dai, and Zhiyang Jia†

School of Automation, Beijing Institute of Technology
No.5 Zhongguancun South Street, Haidian District, Beijing 100081, China
E-mail: bijun.tang@bit.edu.cn
†Corresponding author

[Received October 25, 2020; accepted October 29, 2020]

Hybrid A* algorithm has been widely used in mobile robots to obtain paths that are collision-free and drivable. However, the outputs of hybrid A* algorithm always contain unnecessary steering actions and are close to the obstacles. In this paper, the artificial potential field (APF) concept is applied to optimize the paths generated by the hybrid A* algorithm. The generated path not only satisfies the non-holonomic constraints of the vehicle, but also is smooth and keeps a comfortable distance to the obstacle at the same time. Through the robot operating system (ROS) platform, the path planning experiments are carried out based on the hybrid A* algorithm and the improved hybrid A* algorithm, respectively. In the experiments, the results show that the improved hybrid A* algorithm greatly reduces the number of steering actions and the maximum curvature of the paths in many different common scenarios. The paths generated by the improved algorithm nearly do not have unnecessary steering or sharp turning before the obstacles, which are safer and smoother than the paths generated by the hybrid A* algorithm for the autonomous ground vehicle.

Keywords: path planning, hybrid A* algorithm, artificial potential field, ROS platform

1. Introduction

Path planning algorithm is required to be able to plan a path that is collision-free and dynamic-feasible. The widely used path planning algorithms are sample-based algorithm and graph-based algorithm. The probabilistic roadmap (PRM) [1] and the rapidly exploring random trees (RRTs) [2] are the typical sample-based algorithms. To solve the problem that sample-based algorithms often return jerky and dynamically–infeasible paths, Zhu et al. [3] propose a convex optimization approach to smooth trajectories for motion planning. However, since the distance between the path and the obstacle is not considered, the smoothed path generated by the algorithm proposed by Zhu et al. is close to the obstacles. The Kinodynamic-RRT* is proposed for asymptotically optimal motion planning for robots with linear differential constraints [4]. This approach can fit with motion and other constraints in robot navigation by using a fixed-final-state-free-final-time controller to connect any pair of states. Gammell et al. [5] propose an Informed-RRT* to focus the search on an ellipsoidal informed subset of the state space which improves the convergence rate and final solution quality. The typical graph-based path planning algorithms are A* [6], ARA* [7], and jump point search (JPS) [8], but they tend to produce paths that are non-smooth and do not satisfy the non-holonomic constraints of the vehicle. Therefore, Dolgov et al. [9] propose a hybrid A* algorithm and participate with Junior on the DUC. The hybrid A* algorithm is achieved for the KTH RCV in C++ and ROS platform in [10]. The hybrid A* algorithm uses estimates from both constrained and unconstrained heuristics and augments the search with analytic expansions based on Reeds–Shepp model [11]. The path generated by the hybrid A* algorithm is drivable, but it often contains unnecessary steering actions and is close to the obstacles. The APF [12] is also a path planning algorithm whose basic idea is to regard the movement of the robot in the planning space as a kind of force movement in the virtual force field. The APF has the advantages of small amount of calculation and easy to understand. However, there may be some trap areas due to the local minima and oscillations in the paths generated by the APF before the obstacles and in a narrow passage. If the obstacle is near the target, the robot may not be able to reach the target. Therefore, different kinds of improvements for APF have been made [13–16].

In view of the existing problems of the hybrid A* algorithm mentioned above, the APF concept is used to optimize the hybrid A* algorithm. The whole optimization process consists of three main phases. The first phase is to identify a region of the workspace which is collision free. During the optimization process, the optimized path can be adjusted in this collision-free region to ensure the safety of the path. The second phase is to construct an optimization problem about repulsive and attractive forces. The obstacle closest to the waypoint produces a repulsive force to the waypoint, so each point in the path keeps a distance from the closest obstacle. The position of the each point in the path is adjusted by the attractive forces generated by two adjacent points, which straightens the

© Fuji Technology Press Ltd. Creative Commons CC BY-ND: This is an Open Access article distributed under the terms of the Creative Commons Attribution-NoDerivatives 4.0 International License (http://creativecommons.org/licenses/by-nd/4.0/).

path. To make the optimized path still meet the vehicle's non-holonomic constraints, the optimization problem is subject to the turning radius constraint. This optimization problem makes the path smoother and safer, while ensuring that the path still satisfies the turning radius constraint. The third phase is to solve the structural optimization problem and get a smoother and safer path.

Experiments are carried out with the ROS platform under Linux system, using C++ as the programming language. The optimization problem is solved by a general non-linear optimization solver NLopt. And the optimization algorithm we choose is sequential least-squares quadratic programming (SLSQP). In the hybrid A* algorithm, the open motion planning library (OMPL) which consists of many state-of-the-art sampling-based motion planning algorithms is used to realize the Reeds–Shepp curves. In addition, the Rviz which is a visualization tool provided by ROS platform is used to visualize the planned paths.

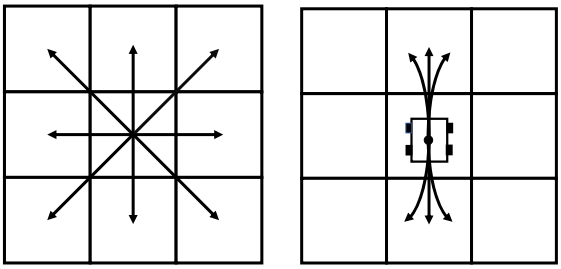
A short overview of the hybrid A* algorithm and the process of using APF to optimize the hybrid A* algorithm are demonstrated in Section 2. In Section 3, the results of experiments on ROS platform are presented to compare the hybrid A* algorithm with the improved hybrid A* algorithm.

2. Improved Hybrid A* Algorithm

2.1. Hybrid A* Algorithm

The A* algorithm can only expand nodes in discrete state space, which does not satisfy the non-holonomic constraints of the vehicle as illustrated in **Fig. 1(a)**. To solve this problem, the hybrid A* algorithm expands nodes in continuous space rather than discrete space. Each continuous state of the hybrid A* algorithm is given by (x, y, θ) which are the location and orientation of the vehicle, respectively. To avoid increasing the size of the search graph, the search space is discretized. The nodes are expanded by the one of the three actions: maximum left steering action, maximum right steering action as well as no steering, illustrated in **Fig. 1(b)**, resulting in an arc of a circle with a lower bound turning radius based on the constraints of the vehicle.

Hybrid A* algorithm uses two heuristics to evaluate the expanded nodes and the maximum of the two heuristics is used for any given state. The constrained heuristic incorporates the vehicle's constraints and ignores the environment, while the unconstrained heuristic disregards the vehicle's constraints and only accounts for obstacles. Firstly, suitable candidates of the constrained heuristic are Dubins curves and Reeds–Shepp curves. The difference between them is that Dubins curves cannot reverse, while the Reeds–Shepp curves can reverse. Since this heuristic takes orientation as well as the turning radius into account, the vehicle approaches the goal with the correct heading. And these curves are the paths of minimal length with an upper bound curvature. Secondly, the estimate of the unconstrained heuristic is based on the shortest distance to the goal. The distance is calculated by the A* algorithm in two dimensions with Euclidean distance. Both heuristics are mathematically admissible, so the maximum of the two can be used. The advantage of the unconstrained heuristic is that it can guide the vehicle away from the dead end and the U-shaped obstacle. The effective usage of heuristics reduces the number of expanded child nodes considerably and makes the search converge toward the solution quickly.



(a) Regular A* search (b) Hybrid A* search

Fig. 1. Node expansion of regular A* and hybrid A* search algorithm.

As the forward search space is discretized, the exact continuous-coordinate goal state is not always reachable by the discretized control actions alone (the accuracy depends on the resolution of the map). So, the search is augmented with analytic expansions based on the Reeds–Shepp model. And the calculated path is checked for collisions with the environment. For computational reasons, the Reeds–Shepp expansion is applied to one of every N nodes, where N decreases as the node gets closer to the goal.

As the node is expanded by applying one of the three steering actions, paths always contain unnatural swerves which need unnecessary steering actions. Another issue of the path obtained by hybrid A* search algorithm is that the distance to the obstacle is close. As the hybrid A* algorithm takes the shortest path as one heuristic, the obtained path tends to be close to the obstacles to pursue the shortest distance. Therefore, the paths generated by the hybrid A* algorithm need to be post-processed by the following optimization procedure.

2.2. Path Optimization Using APF

The basic idea of the APF is to regard the movement of the vehicle in the planning space as a kind of force movement in the virtual force field. The obstacles or threat areas generate repulsive forces to the vehicle, and the target point generates attractive force to it. The vehicle can reach the target and avoid obstacles in response to the resultant force, as illustrated in **Fig. 2**.

Different from the traditional APF for path planning, this method is used in the existed path. The obstacles generate the repulsive forces to the points in the path, and each point is also subject to the attractive forces generated by the two adjacent points. At the same time, in order to

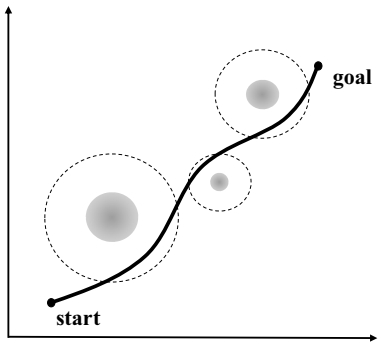


Fig. 2. Path planning based on APF.

make the optimized path satisfy the kinematics constraints of the vehicle, it is necessary to construct an optimization problem of attractive and repulsive forces that is subject to the kinematic constraints (turning radius).

Optimizing the paths generated by the hybrid A* algorithm through the APF is mainly divided into three steps: 1) generate a passable area according to the distance between each point in the reference path and the nearest obstacle; 2) construct an optimization model of the repulsive and attractive forces; 3) solve the optimization problem to optimize the path generated by the hybrid A* algorithm. The flow chart of the overall improved algorithm is shown in **Fig. 3**.

The first step is to identify a region of the workspace that is collision free. As shown in **Fig. 4**, the region is thought as a collision-free "tube," which can be constructed by generating a series of circles. The centers of the circles are the positions of the points in the path generated by the hybrid A* algorithm, and the radiuses are the distances between the points in the path and the nearest obstacles to them. During the optimization process, the new path obtained in each iteration of optimization is limited in the collision-free channel.

The second step is to construct an optimization problem about the repulsive and attractive forces. The obstacles generate repulsive forces to the points in the path, and each point is also subject to the attractive forces generated by the two adjacent points. The repulsive forces make the path keep away from the obstacles and the attractive forces make the path smooth without redundant steering. When the resultant force of two attractive forces at one point is equal to zero, the path is a straight line, which is "ideal smoothness." The position of the each point in the path is adjusted by the resultant force in the respective collision-free circle. And in order to make the optimized path satisfy the kinematics constraints of the vehicle, it is necessary to make the optimization problem be subject to the minimum turning radius, the starting point, and the target point constraints.

The each point in the path is subject to the attractive forces generated by two adjacent points, as illustrated in **Fig. 5**. The effect of attractive forces is to straighten the path. When the point in the path moves along the attractive force direction, the attractive force decreases, and the

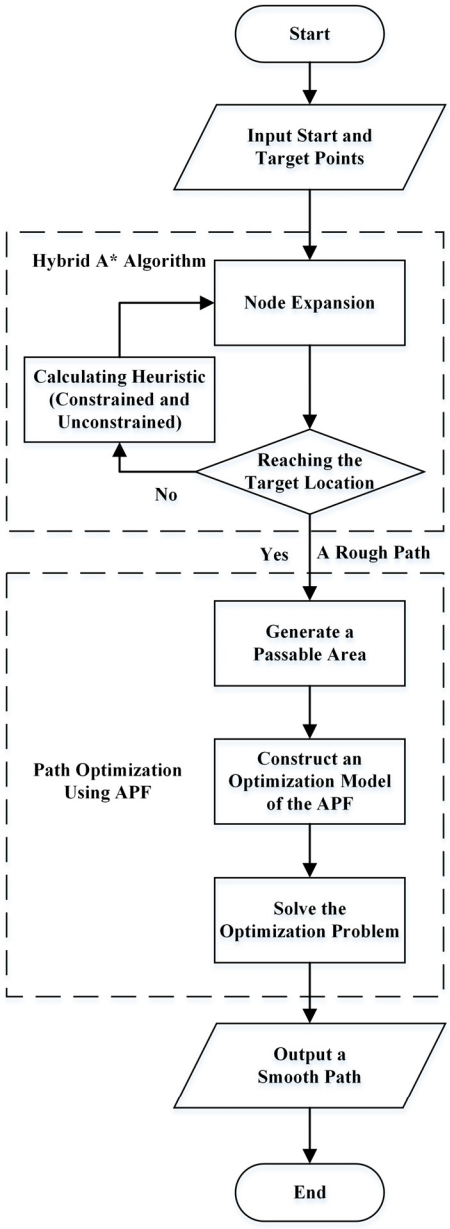


Fig. 3. The flowchart of the improved algorithm.

adjacent three points tend to be on the same line, which means the path is smoother. The attractive force is defined as

$$\begin{aligned} \|F_{\mathrm{att},k}\| &= \|F_{k-1} - F_k\| \\ &= \| \underbrace{(P_k - P_{k-1})}_{F_{k-1}} - \underbrace{(P_{k+1} - P_k)}_{F_k} \|, \quad \ldots \quad (1) \end{aligned}$$

where $F_{\mathrm{att},k}$ denotes the attractive force at the k-th point, which is the vector sum of the two attractive forces.

Although the optimized path is limited in the collision-free "tube" generated in the first step, the attractive forces make the path very close to the obstacle. Therefore, it is necessary to use the repulsive forces to keep the path away from the obstacles. Since the obstacles in the map are large and irregular, we assume that the each point in the

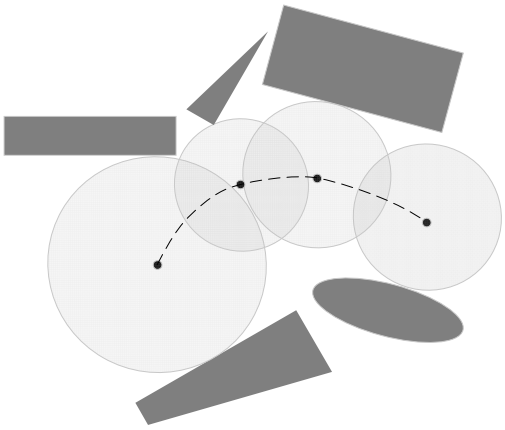


Fig. 4. The collision-free channel.

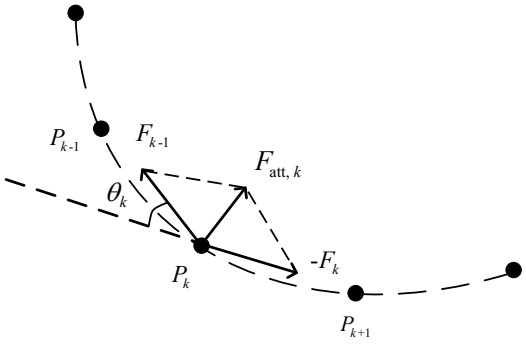


Fig. 5. Attractive forces from two adjacent points.

path is only affected by the obstacle closest to it, as shown in **Fig. 6**. The role of the repulsive forces is to keep the path away from obstacle. If the distance between the point in the path and the obstacle is less than the threshold, the obstacle will generate a repulsive force to the point. If the distance is greater than the threshold, the repulsive force generated by the obstacle is considered negligible to the path. Thus the repulsive force should be a non-negative and monotonically decreasing function in domain $[0, d_{\text{th}})$ and equal to zero when $d(P_k, O_k) \geq d_{\text{th}}$. When the point in the path moves in the direction of the repulsive force, the repulsive force decreases and the path maintains a safe distance from the obstacles. So, the repulsive force is defined as

$$\|F_{\text{rep},k}\| = \begin{cases} K_{\text{rep}} \left(1 - \dfrac{d(P_k, O_k)}{d_{\text{th}}}\right), & d(P_k, O_k) < d_{\text{th}}, \\ 0, & d(P_k, O_k) \geq d_{\text{th}}, \end{cases} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2)$$

where $d(P_k, O_k)$ denotes the distance between the k-th point in the path and the obstacle closest to it; d_{th} denotes the threshold of the distance; K_{rep} denotes the maximum repulsive force.

Although the attractive forces smooth the path, in some cases such as U-turns, it causes the path to be pulled out of a tip. So, the optimization problem is subject to the turning radius constraint. To relate $F_{\text{att},k}$ with R_k

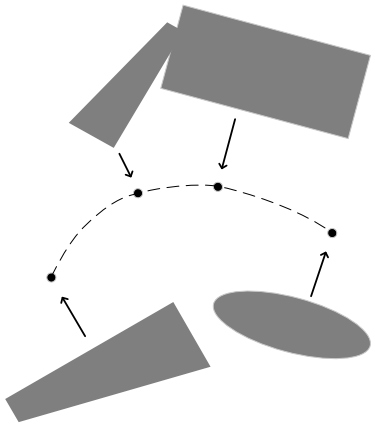


Fig. 6. Repulsive forces from the closest obstacles.

as [3], there are some approximations which are valid when the waypoints are uniformly spread over a path and dense "enough": 1) $\|F_{k-1}\| = \|F_k\|$, 2) θ_k is small, and 3) $\theta_k \cong \|P_{k+1} - P_k\|/R_k$. The points in the path generated by the hybrid A* algorithm can meet the above assumptions, so we can obtain the relationship between $F_{\text{att},k}$ and R_k as follows:

$$\begin{aligned} \|F_{\text{att},k}\| &= \|F_{k-1} - F_k\| \cong 2\|F_k\| \cdot \sin\left(\frac{\theta_k}{2}\right) \\ &\cong \|P_{k+1} - P_k\|\theta_k \cong \frac{\|P_{k+1} - P_k\|^2}{R_k}. \quad \ldots (3) \end{aligned}$$

To make the above inequality a quadratic constraint, the average length of any two adjacent points in the path is used in the inequality to replace the length of each band $P_{k+1} - P_k$. The average length d is defined as

$$d = \frac{\sum_{k=1}^{n-1} \|Q_{k+1} - Q_k\|}{n-1}, \quad \ldots\ldots\ldots\ldots (4)$$

where Q_i is the i-th point in the path generated by the hybrid A* algorithm. Therefore, the minimum turning radius constraint is defined as

$$\|F_{\text{att},k}\| \leq \frac{d^2}{R_{\min}}. \quad \ldots\ldots\ldots\ldots\ldots (5)$$

To constrain the positions and orientations of the start and target points of the path, the constraints $P_1 = Q_1$, $P_2 = Q_2$, $P_{n-1} = Q_{n-1}$, and $P_n = Q_n$ are added to the optimization problem. The whole optimization problem is defined as

$$\begin{aligned} &\min \sum_{k=2}^{n-1} (\|F_{\text{att},k}\|^2 + \|F_{\text{rep},k}\|) \\ &\text{s.t.} \quad P_1 = Q_1, \quad P_2 = Q_2, \\ &\qquad P_{n-1} = Q_{n-1}, \quad P_n = Q_n, \\ &\qquad P_k \in C_k, \quad k = 3, \ldots, n-2, \\ &\qquad \|F_{\text{att},k}\| \leq \frac{d^2}{R_{\min}}, \quad k = 3, \ldots, n-2. \quad \ldots (6) \end{aligned}$$

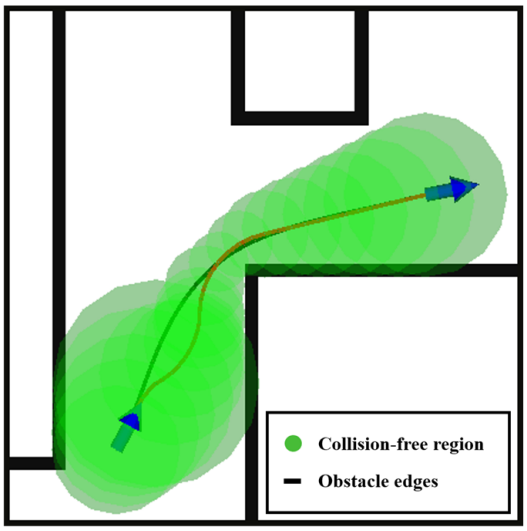


Fig. 7. The collision-free region.

The variables are the positions of the waypoints P_k, which can be placed anywhere within the collision-free region generated in the step one.

The third step is to solve the optimal problem constructed in the second step. The most important thing in this step is to choose a suitable algorithm to solve the constructed optimization problem. The principles for selecting the optimization algorithm are as follows: 1) Since the optimization variables are the positions of all points in the path, this optimization algorithm must be able to quickly locate the solution in high-dimensional problems. 2) This algorithm must be able to support nonlinear inequality constraints. Therefore, the optimization algorithm we choose is SLSQP which is a sequential quadratic programming (SQP) algorithm for nonlinearly constrained gradient-based optimization. And NLopt which is a free and open-source library for nonlinear optimization includes SLSQP, so we use NLopt to solve the optimal problem constructed in the second step.

3. Experimental Results

The effectiveness of this improved algorithm is verified by simulation on the ROS platform. The improved hybrid A* algorithm is implemented in C++ with a general nonlinear optimization solver NLopt. All the experiments run on a quad-core 2.70 GHz Intel i7-7500U processor, which has 8 GB RAM. The paths generated by the hybrid A* algorithm and the improved hybrid A* algorithm are visualized by Rviz. The input of the map is a binary occupancy grid.

As shown in **Fig. 7**, the collision-free regions are the green circle areas. During the optimization process, the new path obtained at each iteration of optimization is limited in this collision-free area. In **Figs. 8–10**, the red curves are the paths planned by the hybrid A* algorithm. The black curves are the paths planned by the improved hybrid A* algorithm. The blue arrows are the start points and the target points. Three experimental scenarios are mainly considered: line changing, maze with rectangular-shaped obstacles, and other four common scenarios. **Tables 1–5** summary the results of the paths generated by the two algorithms in the different scenarios. The experiments in the three scenarios are introduced as follows.

3.1. Path Planning in Line Changing Scenario

In this scenario, we consider a road line with rectangular-shaped obstacles in it. The width and height of the map are 100 and 10 pixels, respectively. And the resolution of the map is 1 m per pixel. As shown in **Fig. 8**, the path obtained by the hybrid A* algorithm obviously has some unnecessary steering actions and makes sharp turning just before the obstacle, which is dangerous and even impossible when the vehicle is driving in high speed. The path generated by the improved hybrid A* algorithm keeps a distance from the obstacles without unnecessary steering, which is smoother and safer than the path generated by the hybrid A* algorithm.

From the experimental results of the first scenario, we can see that hybrid A* algorithm mainly has the following two problems: 1) Since the algorithm pursues the shortest path, the path is close to the obstacle and it turns sharply when approaching the obstacle. There is a tendency to "hug" the obstacle, which is not conducive to the safety of the vehicle and is easy to collide the obstacle in reality. 2) The nodes in the path are expanded by one of three actions. Therefore, some unnecessary steering actions occur in the path, which increases the friction loss of the vehicle and reduces the driving efficiency and safety during actual driving.

3.2. Path Planning in Maze

In this scenario the workspace is a maze whose width and height are both 100 pixels, and the resolution is 1 m per pixel. As shown in **Fig. 9**, the path obtained by the hybrid A* algorithm has some unnecessary steering actions and is close to the obstacles. The path obtained by the improved algorithm nearly does not have unnecessary steering actions and can keep a comfortable distance to the obstacles.

In the experiment, we compare and analyze the results of the paths generated by the two algorithms in **Fig. 9**. There are 181 waypoints in the path, which means that the number of variables in this optimization problem is 362. The entire optimization process is iterated 1441 times, and the objective function value drops to 0.105 eventually. As can be seen from the data in the **Table 1**, the number of steering actions in the path obtained by the improved hybrid A* algorithm is almost reduced to 10% of the number of steering actions in the path obtained by the hybrid A* algorithm. By calculating the curvature of each point on the two paths separately, the maximum curvature of the paths generated by the two algorithms is compared. The maximum curvature of the path generated by the improved hybrid A* algorithm is also almost reduced to 10%

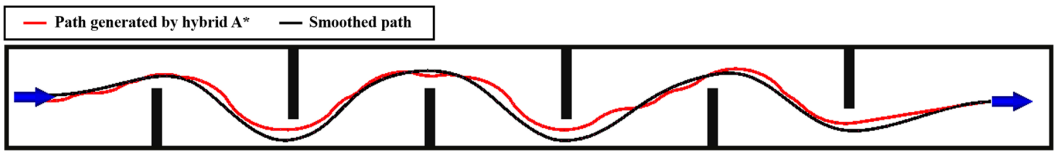


Fig. 8. Optimization results in lane changing scenario.

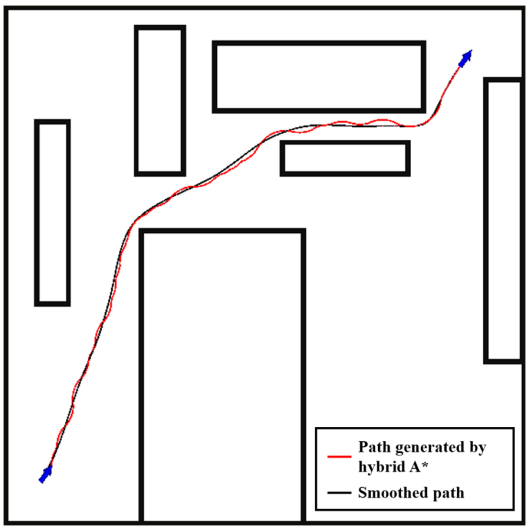


Fig. 9. Comparison of hybrid A* algorithm and improved hybrid A* algorithm in a maze-like environment.

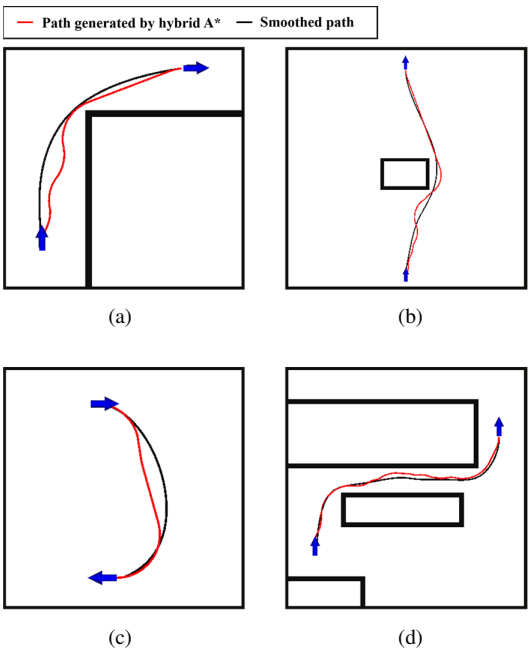


Fig. 10. Paths obtained in four common scenarios. (a) shows passing through a 90° corner; (b) shows bypassing obstacles; (c) shows U-turns; (d) shows passing a narrow passage.

of the maximum curvature of the path generated by the hybrid A* algorithm.

Table 1. Comparison of results of the paths in **Fig. 9**.

	The number of steering	The max curvature of the path
Improved Algorithm	3	0.339
Hybrid A*	31	3.145

Table 2. Comparison of results of the paths in **Fig. 10(a)**.

	Number of steering	Max curvature of the path	Time [ms]
Improved Algorithm	1	0.172	2443.422
Hybrid A*	5	3.145	1340.156

Table 3. Comparison of results of the paths in **Fig. 10(b)**.

	Number of steering	Max curvature of the path	Time [ms]
Improved Algorithm	3	0.102	6464.229
Hybrid A*	8	3.145	3756.872

Table 4. Comparison of results of the paths in **Fig. 10(c)**.

	Number of steering	Max curvature of the path	Time [ms]
Improved Algorithm	1	0.177	965.982
Hybrid A*	2	3.145	110.338

Table 5. Comparison of results of the paths in **Fig. 10(d)**.

	Number of steering	Max curvature of the path	Time [ms]
Improved Algorithm	2	0.226	4498.08
Hybrid A*	9	3.145	3107.996

The smaller the maximum curvature and the fewer turns, the more comfortable and effective of driving. And the path generated by the improved hybrid A* algorithm is easier and safer to track for a lower-level controller.

3.3. Path Planning in Four Common Scenarios

In order to verify the effectiveness of the improved hybrid A* algorithm in more cases, several paths in some common scenarios are depicted in **Figs. 10(a)–(d)**. There are four common scenarios: passing through a 90° corner, bypassing obstacles, U-turns, and passing a narrow passage. In **Fig. 10(a)**, the both paths have 49 waypoints, the entire optimization process is iterated 531 times, and the objective function value drops to 0.048 eventually. In **Fig. 10(b)**, the both paths have 79 waypoints, the entire optimization process is iterated 1001 times, and the objective function value drops to 0.049 eventually. In **Fig. 10(c)**, the both paths have 38 waypoints, the entire optimization process is iterated 388 times, and the objective function value drops to 0.176 eventually. In **Fig. 10(d)**, the both paths have 73 waypoints, the entire optimization process is iterated 261 times, and the objective function value drops to 0.237 eventually.

In both **Figs. 10(a)** and **(b)**, the paths generated by the hybrid A* algorithm contain unnecessary steering actions and tend to "hug" the obstacles. The paths generated by the improved algorithm do not turn sharply in front of the obstacle and do not have unnecessary steering actions in both **Figs. 10(a)** and **(b)**. In **Fig. 10(c)**, the path generated by the hybrid A* algorithm has two large-angle steering actions while the path generated by the improved hybrid A* algorithm only have one slightly steering action. In **Fig. 10(d)**, there are oscillations in the path generated by the hybrid A* algorithm in the narrow passage, which is unsafe and unstable for the vehicle in actual driving.

Tables 2–5 list the results of the paths obtained by the two algorithms in **Figs. 10(a)–(d)**. As can be seen from the data in the **Tables 2–5**, the number of steering actions in the paths generated by the improved algorithm are reduced to the minimum number of steering actions to reach the goal. And the maximum curvature of the paths generated by the improved hybrid A* algorithm is reduced by more than ten times. The running time of the improved hybrid A* algorithm is higher than that of the hybrid A* algorithm. And the number of iterations basically increases as the number of points in the path increases. This is because that the difficulty of solving the optimization problem is related to the number of variables. The more variables, the more difficult it is to solve the optimization problem.

4. Conclusion

In order to solve the problems existing in the hybrid A* search algorithm, an improved hybrid A* algorithm is presented by constructing an optimization problem about repulsive and attractive forces. The repulsive forces keep the paths away from obstacles, and attractive forces make the paths smoother without unnecessary steering actions. The experiments, which compare the hybrid A* algorithm and the improved hybrid A* algorithm, are implemented in C++ with a general non-linear optimization solver NLopt. The simulation results show that the improved hybrid A* algorithm solves some problems existing in the hybrid A* algorithm.

The experiments show that although path obtained by hybrid A* algorithm is drivable, it always contains unnatural swerves that require unnecessary steering actions. Another issue of the paths generated by the hybrid A* algorithm is that the distance to the obstacles is close. The path tends to "hug" the obstacles to achieve the minimum path length which is suboptimal for safety distance to the obstacles. The path generated by the improved algorithm nearly does not have unnecessary steering actions and keeps a comfortable distance to the obstacles, which is safer and more efficient during actual driving than the path obtained by the hybrid A* algorithm in all scenarios. **Table 1** shows that the improved hybrid A* algorithm reduces the number of steering actions and the maximum curvature by nine out of ten in the maze with rectangular-shaped obstacles in it, which increases driving comfort and effectiveness. **Tables 2–5** show that the maximum curvature and the number of steering actions in the path generated by the improved hybrid A* algorithm are greatly reduced in the four scenarios. The experiments verify the effectiveness of the improved algorithm in line changing, path planning in a maze, passing through a 90° corner, bypassing obstacles, U-turns, and passing a narrow passage scenarios.

In the future, we plan to deploy the improved algorithm on a real self-driving car and try to extend this improved algorithm to other dynamic systems, such as unmanned aerial vehicle or underwater glider.

Acknowledgements

This work is supported in part by the National Talents Foundation under Grant No. WQ20141100198, in part by the Beijing Municipal Natural Science Foundation under Grant No. 3192028.

References:

[1] M. Baumann, S. Léonard, E. A. Croft, and J. J. Little, "Path Planning for Improved Visibility Using a Probabilistic Road Map," IEEE Trans. on Robotics, Vol.26, No.1, pp. 195-200, 2010.

[2] M. Kothari and I. Postlethwaite, "A Probabilistically Robust Path Planning Algorithm for UAVs Using Rapidly-Exploring Random Trees," J. of Intelligent & Robotic Systems, Vol.71, Issue 2, pp. 231-253, 2013.

[3] Z. Zhu, E. Schmerling, and M. Pavone, "A Convex Optimization Approach to Smooth Trajectories for Motion Planning with Car-Like Robots," Proc. of the 2015 54th IEEE Conf. on Decision and Control (CDC), pp. 835-842, 2015.

[4] D. J. Webb and J. Van Den Berg, "Kinodynamic RRT*: Asymptotically Optimal Motion Planning for Robots with Linear Differential Dynamics," Proc. of the 2013 IEEE Int. Conf. on Robotics and Automation, pp. 5054-5061, 2013.

[5] J. D. Gammell, S. S. Srinivasa, and T. D. Barfoot, "Informed RRT*: Optimal Sampling-based Path Planning Focused via Direct Sampling of an Admissible Ellipsoidal Heuristic," Proc. of the 2014 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp. 2997-3004, 2014.

[6] A. V. Le, V. Prabakaran, V. Sivanantham, and R. E. Mohan, "Modified A-Star Algorithm for Efficient Coverage Path Planning in Tetris Inspired Self-Reconfigurable Robot with Integrated Laser Sensor," Sensors, Vol.18, No.8, Article No.2585, 2018.

[7] M. Likhachev, G. J. Gordon, and S. Thrun, "ARA*: Anytime A* with Provable Bounds on Sub-optimality," Proc. of the 16th Int. Conf. on Neural Information Processing Systems, pp. 767-774, 2004.

[8] D. Harabor and A. Grastien, "Online Graph Pruning for Pathfinding on Grid Maps," Proc. of the 25th AAAI Conf. on Artificial Intelligence (AAAI'11), pp. 1114-1119, 2011.
[9] D. Dolgov, S. Thrun, M. Montemerlo, and J. Diebel, "Path Planning for Autonomous Vehicles in Unknown Semi-structured Environments," The Int. J. of Robotics Research, Vol.29, Issue 5, pp. 485-501, 2010.
[10] K. Kurzer, "Path Planning in Unstructured Environments: A Real-time Hybrid A* Implementation for Fast and Deterministic Path Generation for the KTH Research Concept Vehicle," Master Thesis, KTH Royal Institute of Technology, 2016.
[11] J. A. Reeds and L. A. Shepp, "Optimal Paths for a Car That Goes Both Forwards and Backwards," Pacific J. of Mathematics, Vol.145. No.2, pp. 367-393, 1990.
[12] O. Khatib, "Real-time Obstacle Avoidance for Manipulators and Mobile Robots," Proc. of the 1985 IEEE Int. Conf. on Robotics and Automation, pp. 500-505, 1985.
[13] Z. Jin, B. Yan, and R. Ye, "The Flight Navigation Planning Based on Potential Field Ant Colony Algorithm," Proc. of the 2018 Int. Conf. on Advanced Control, Automation and Artificial Intelligence (ACAAI 2018), pp. 200-204, 2018.
[14] T. Weerakoon, K. Ishii, and A. A. F. Nassiraei, "An Artificial Potential Field Based Mobile Robot Navigation Method to Prevent from Deadlock," J. of Artificial Intelligence and Soft Computing Research, Vol.5, No.3, pp. 189-203, 2015.
[15] Z.-Z. Yu, J.-H. Yan, J. Zhao, Z.-F. Chen, and Y.-H. Zhu, "Mobile Robot Path Planning Based on Improved Artificial Potential Field Method," J. of Harbin Institute of Technology, Vol.43, No.1, pp. 50-55, 2011 (in Chinese).
[16] N. Zhang, Y. Zhang, C. Ma, and B. Wang, "Path Planning of Six-DOF Serial Robots Based on Improved Artificial Potential Field Method," Proc. of the 2017 IEEE Int. Conf. on Robotics and Biomimetics (ROBIO), pp. 617-621, 2017.

Name:
Bijun Tang

Affiliation:
School of Automation, Beijing Institute of Technology

Address:
No.5 Zhongguancun South Street, Haidian District, Beijing 100081, China

Brief Biographical History:
2018 Received B.S. degree from University of Science and Technology Beijing
2018- M.S. Candidate, School of Automation, Beijing Institute of Technology

Main Works:
• Path planning

Name:
Kaoru Hirota

Affiliation:
School of Automation, Beijing Institute of Technology

Address:
No.5 Zhongguancun South Street, Haidian District, Beijing 100081, China

Brief Biographical History:
1982-1995 Professor, College of Engineering, Hosei University
1995-2015 Professor, Tokyo Institute of Technology
2015- Professor, Beijing Institute of Technology

Main Works:
• Image pattern recognition, intelligent robotics, fuzzy control, artificial intelligence
• K. Chen, F. Yan, K. Hirota, and J. Zhao, "Quantum Implementation of Powell's Conjugate Direction Method," J. Adv. Comput. Intell. Intell. Inform., Vol.23, No.4, pp. 726-734, 2019.
• K. Uehara and K. Hirota, "Noise Reduction with Inference Based on Fuzzy Rule Interpolation at an Infinite Number of Activating Points: Toward Fuzzy Rule Learning in a Unified Inference Platform," J. Adv. Comput. Intell. Intell. Inform., Vol.22, No.6, pp. 883-899, 2018.
• L. Chen, M. Zhou, M. Wu, J. She, Z. Liu, F. Dong, and K. Hirota, "Three-Layer Weighted Fuzzy Support Vector Regression for Emotional Intention Understanding in Human–Robot Interaction," IEEE Trans. on Fuzzy Systems, Vol.26, No.5, pp. 2524-2538, 2018.

Membership in Academic Societies:
• The Institute of Electrical and Electronics Engineers (IEEE), Life Member
• International Fuzzy Systems Association (IFSA), Fellow, Past-President
• Japan Society for Fuzzy Theory and Intelligent Informatics (SOFT), Honorable Member, Past-President

Name:
Xiangdong Wu

Affiliation:
School of Automation, Beijing Institute of Technology

Address:
No.5 Zhongguancun South Street, Haidian District, Beijing 100081, China

Brief Biographical History:
2013 Received B.S. degree from Electronic Information Engineering, Hengshui University
2017 Received M.S. degree from Control Science and Engineering, Yanshan University
2018- Ph.D. Candidate, School of Automation, Beijing Institute of Technology

Main Works:
• Intelligent optimization algorithms, motion planning

Name:
Yaping Dai

Affiliation:
School of Automation, Beijing Institute of Technology

Address:
No.5 Zhongguancun South Street, Haidian District, Beijing 100081, China

Brief Biographical History:
1990-1994 Ph.D., Beijing Institute of Technology
2002- Professor, Beijing Institute of Technology

Main Works:
• Networked control, internet of things and e-experiment, data fusion, targets identification and tracking in video, smart city relative
• She has published 1 translation book and more than 80 papers, won "quality course certification" from Beijing Municipal Education Commission.

Membership in Academic Societies:
• Beijing Automation Association, Vice-Chief-Director

Name:
Zhiyang Jia

Affiliation:
School of Automation, Beijing Institute of Technology

Address:
No.5 Zhongguancun South Street, Haidian District, Beijing 100081, China

Brief Biographical History:
2010 Received B.S. degree from Electrical Engineering and Automation, Northwestern Polytechnical University
2013 Received M.S. degree from Control Science and Engineering, Beijing University of Technology
2017 Received Ph.D. degree from Electrical and Computer Engineering, University of Connecticut
2017-2019 Post-Doctoral Researcher, School of Automation, Beijing Institute of Technology
2019- Assistant Professor, School of Automation, Beijing Institute of Technology

Main Works:
• Intelligent manufacturing, production system engineering, operational research, management and decision

Membership in Academic Societies:
• The Institute of Electrical and Electronics Engineers (IEEE)