An Improved Artificial Potential Field Method Based on DWA and Path Optimization

Jinwen Hu
School of Automation
Northwestern Polytechnical University
Xi'an, China
hujinwen@nwpu.edu.cn

Changwei Cheng
School of Automation
Northwestern Polytechnical University
Xi'an, China
chengchangwei@mail.nwpu.edu.cn

Ce Wang
School of Automation
Northwestern Polytechnical University
Xi'an, China
18829237740@163.com

Chunhui Zhao
School of Automation
Northwestern Polytechnical University
Xi'an, China
zhaochunhui@nwpu.edu.cn

Quan Pan
School of Automation
Northwestern Polytechnical University
Xi'an, China
quanpan@nwpu.edu.cn

Zhenbo Liu
School of Automation
Northwestern Polytechnical University
Xi'an, China
liuzhenbo1@mail.nwpu.edu.cn

***Abstract*—The artificial potential field (APF) method is a simple and effective path planning approach. However, there is a fatal problem, which is that the robot can fall into local minima easily before reaching destination. Thus in this paper, we propose an improved APF method based on dynamic window approach (DWA) by evaluating points around robot in local minima with evaluation function, and choose the best point as next path point. Besides, we also propose an optimization algorithm to shorten the path. The main idea is to connect the consecutive points of the planned path while leaving enough secure space between robot and obstacles. The results show that our improved method can solve the problem of local minima and the optimization algorithm can plan a shorter path while consuming little computational time.**

***Index Terms*—Path planning, improved APF, DWA, path optimization.**

I. Introduction

Robotic path planning is a hot problem in robotic research, and there are many good methods which are suitable for different environment. They can be classified as classic and heuristic methods [1]. Many path planning methods have their unique advantages and disadvantages, so there is not a method that is suitable for all kinds of environment and practicalities. The common path planning algorithms are simulated annealing [3], artificial potential field [4], RRT [5], A* [6], Q-learning [7] et al..

One of the classic algorithm is the APF method proposed by Khatib in 1986, where the robot walks along the direction of the resultant force produced by obstacles and the target, which is similar to the motion of a charge in an electric field. The APF has been widely used because it is very simple and effective in path planning. But robot is easily trapped in local minima, which is the main disadvantage of APF. There are many solutions to local minima. Reference [8] used a virtual obstacle in the area near local minima to push the robot away. Another improved APF is proposed in [9] integrating the APF with simulated annealing, but the method may fail when there are many obstacles in a small area. Reference [10] modified the repulsive force, and the new repulsive force consists of tangential repulsive force and radial repulsive force. Reference [11] combined APF method and grid method to avoid local minima and optimize the path. There are many good solutions to avoid local minima problem besides these, they can plan a useful path for robot to walk along.

In this paper, we propose an improved APF method based on DWA to avoid local minima problem in a known environment. DWA is a popular path planning method which use the dynamic characteristics of line acceleration and angular acceleration of the robot to produce the trajectories and choose the best trajectory by evaluation function [12]. When robot is trapped in local minima, we evaluate the points around the robot with an evaluation function and choose the best point as the next path point until robot escapes from local minima. Besides, we also propose a optimization algorithm to optimize the planned path. Our main idea is similar to the method in [2], which connected the planned path points to convert the curve path to the path consisting of severe line segment. On their basis, we consider the size of the robot. To avoid that optimized path is too close to obstacle, we expand the line segment to rectangle to leave enough space for robot. Thus we need to judge the relative position of the rectangle and obstacles.

We organize this paper as follows. In Section II, we describe the tradition APF method and local minima briefly. Section III explains the improved APF based on DWA in details. In Section IV, we discuss our optimization algorithm and how to leave enough secure space for the robot. We give some simulation experiments in Section V. We conclude our work and future plans in Section VI.

The research is supported by National Natural Science Foundation of China (61603303, 61803309, 61703343), Natural Science Foundation of Shaanxi Province (2018JQ6070), China Postdoctoral Science Foundation (2018M633574) and Fundamental Research Funds for the Central Universities(3102019ZDHKY02, 3102018JCC003).

978-1-7281-3792-6/19/$31.00 ©2019 IEEE

II. Problem Description

A. Traditional APF

In APF method, we assume the robot as a point, thus we don't need to consider the size of robot. The robot walks in the virtual force field, which is similar to electric field. The virtual force field $U(q)$ is composed of attractive field $U_{att}(q)$ and repulsive field $U_{rep}(q)$. The attractive force $F_{att}(q)$ and repulsive force $F_{rep}(q)$ are the negative gradient of $U_{att}(q)$ and $U_{rep}(q)$, respectively. The resultant force $F(q)$ are the sum of $F_{att}(q)$ and $F_{rep}(q)$. The direction of robot's movement is the direction of resultant force $F(q)$ (Fig. 1). In the real environment, there may be a lot of obstacles, so $F_{rep}(q)$ should be the sum of the obstacles near the robot, that is $F_{rep}(q) = \sum_{i=1}^{n} F_{repi}(q)$.

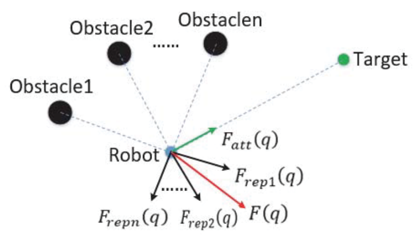


Fig. 1: Force in the APF method.

B. Local Minima

The robot is easily trapped in local minima when we use the APF method. There are many reasons for it. When $F_{att}(q)$ and $F_{rep}(q)$ are equal in magnitude and opposite in direction, $F(q)$ is zero and the robot will stop before reaching the target (Fig. 2). Besides, when there are many obstacles in a small area, this area will easily form a potential well. The robot will be easily trapped in it while $F(q)$ is larger than zero (Fig. 3).

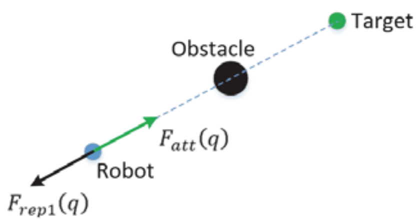


Fig. 2: The resultant force $F(q)$ may be zero when robot, obstacles and target point are on the same line.

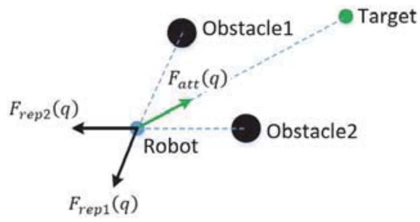


Fig. 3: If there are many obstacles in a small area, the area will form a potential well under the influence of these obstacles and the target. Then the robot is easily trapped around here.

III. Improved Method

A. Modification of Attractive Field and Attractive Force

In the traditional APF method, $F_{att}(q)$ is proportional to the distance from robot to target. In other words, when robot is far away from the target, $F_{att}(q)$ can be very large. Therefore, the robot will ignore the repulsive force $F_{rep}(q)$ and collide with obstacles. Thus, we use the modified the potential field and the force [2]

$$U_{att}(q) = \begin{cases} \frac{1}{2} k_{att}(q - q_g)^2, & ||q - q_g|| \le d_0; \\ k_{att} d_0 (q - q_g), & ||q - q_g|| > d_0 \end{cases} \tag{1}$$

and

$$F_{att}(q) = \begin{cases} -k_{att}(q - q_g), & ||q - q_g|| \le d_0; \\ -k_{att} d_0 \frac{(q - q_g)}{||q - q_g||}, & ||q - q_g|| > d_0, \end{cases} \tag{2}$$

where k_{att} is a attract constant, $q = (x, y)$ is the current coordinate of the robot, $q_g = (x_g, y_g)$ is the coordinate of the target. d_0 is the largest influence distance of the target. If the distance for the robot to the target is larger than d_0, $F_{att}(q)$ will be a constant.

The graphs of $U_{att}(q)$ and $F_{att}(q)$ after modification are seen in Fig. 4.

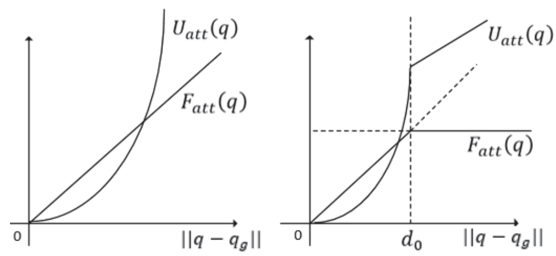


Fig. 4: On the left is the attractive field and attractive force before the modification, and on the right is the modified.

B. Solution to Local Minima

Local minima is the main problem faced by APF method. Inspired by DWA algorithm, we evaluate all feasible points around local minima with evaluation function, then choose the best point which has the highest evaluation score as the next path point. In our improved APF method, if the robot is trapped in local minima, we start our improved method. The details are as follows.

The robot takes the current position point of itself as the center of the circle, the $Stepsize$ as the radius, and takes one point every $\pi/180$ on the circle, totaling 360 points. You can change the number of the points according to your needs. Then we evaluate all feasible points by the evaluation function. After getting the scores of these points, we select the highest evaluation score point as the next path point. The evaluation function is

$$\widetilde{G}(x,y) = \alpha \cdot heading(x,y) + \beta \cdot dist(x,y), \quad (3)$$

- $\boldsymbol{heading(x,y)}$ **:** let θ be the angle between line l_1 and line l_2 (Fig. 5). The value of $heading(x,y)$ should be large when the value of θ is small, that is, the robot moves toward the target. The relationship between $heading(x,y)$ and θ is $heading(x,y) = 180 - \theta$.

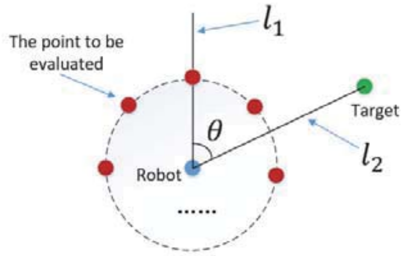


Fig. 5: Angle θ to the target.

- $\boldsymbol{dist(x,y)}$ **:** the value of $dist(x,y)$ is the distance to the closest obstacle for the robot. If $dist(x,y)$ is very large, set it as a constant value $dist(x,y)_{max}$.

where α and β are the weights of $heading(x,y)$ and $dist(x,y)$, respectively.

After getting $heading(x,y)$ and $dist(x,y)$, it's time to set the value of α and β. In traditional DWA, α and β are fixed. In our paper, We fuzzy α and β. When the value of $heading(x,y)$ is large, which means the direction robot moves forward is good, we can set the value of β to a larger value to encourage this direction. The detail is: set α=1, and the relationship between β and $heading(x,y)$ is

$$\beta = \begin{cases} 4, & if \quad heading(x,y) > 135; \\ 3, & if \quad heading(x,y) > 90; \\ 2, & if \quad heading(x,y) > 45; \\ 1, & if \quad heading(x,y) > 0. \end{cases} \quad (4)$$

Then some progress on $heading(x,y)$ and $dist(x,y)$ should be done. Generally speaking, the value of $heading(x,y)$ is in $[0, 180°]$ while the value of $dist(x,y)$ is below 15 (in our simulation environment), which makes that $\widetilde{G}(x,y)$ is mainly determined by $heading(x,y)$. So we normalize $heading(x,y)$ and $dist(x,y)$:

$$heading(i) = \frac{heading(i)}{\sum_{i=1}^{n} heading(i)}, \quad dist(i) = \frac{dist(i)}{\sum_{i=1}^{n} dist(i)}. \quad (5)$$

So the final evaluation function $G(x,y)$ is decided by the following function:

$$G(x,y) = \alpha \cdot heading(x,y) + \beta \cdot dist(x,y). \quad (6)$$

IV. Path Optimization

The improved APF method can solve the problem of local minima in traditional APF method, but the planned path consists of many points, which make the path seem like a curve. The curve path is not optimal or close to optimal, which seriously restricts the use of the method, especially for robots that are particularly constrained by time and energy. So in this section, we propose an optimization algorithm to quickly shorten the path points obtained by the improved APF method based on DWA. After optimizing, the path consists of severe line segments.

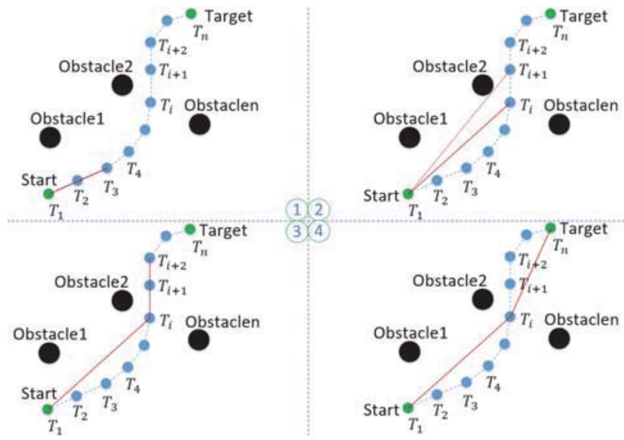


Fig. 6: Strategy of optimization algorithm.

We use Fig. 6 to illustrate the optimization algorithm. In Fig. 6, it is assumed that the points $T_i \in \{T_1, T_2, T_3 ... T_i, T_{i+1} ... T_n\}$ are path points planned by using the APF method based on DWA. T_1 is the start point while T_n is the target point. The robot moves along $T_1, T_2 ... T_n$ in order. First, connect T_1 and T_3, and obtain a line segment $L_{1,3}$, then calculate whether $L_{1,3}$ collides with obstacles. If $L_{1,3}$ does not collide with obstacles, then connect T_1 and T_4 and obtain a line segment $L_{1,4}$. Calculate whether $L_{1,4}$ collides with the obstacles or not. Repeat above steps until $L_{1,i+1}$ collides with the obstacles or is too close to obstacles. So $L_{1,i+1}$ should be abandoned while $L_{1,i}$ can be accepted. By now, the optimization of the first stage is over. In the second stage, choose T_i as the next start point. Then connect T_i and T_{i+2}, and obtain a line segment $L_{i,i+2}$. Continue to judge whether $L_{i,i+2}$ collides with obstacles...Repeat the above operation until reaching T_n and obtain $L_{i,n}$. In each optimization stage, we connect starting point T_i and T_{i+2} instead of T_{i+1}, because $L_{i,i+1}$ has be checked when planning the path.

After using the optimization algorithm, the path consists of several line segments instead of many points (Before optimization, the path consists of many points, thus the path is more like a curve). Thus the length of the path has been shorten a lot. In the optimization process, we need to judge whether the line segment collides with obstacles or not.

In the simulation environment, we regard obstacles as a circle, and the robot as a point. Considering that robot is not a point in the actual scene, so we expand the line segment into a rectangle to leave a safe space for the robot. Judging whether the line segment collides with obstacles or not equals judging the relative position of the rectangle and the circle. Detailed steps are as follows.

Step 1: expand the line segment to rectangle.

After obtaining line segment $L_{i,i+k}$, expand $L_{i,i+k}$ towards its two sides to obtain rectangle $ABCD$ (Fig. 7). The length of AT_{i+k} depends on robot's size and we set $AT_{i+k} = 0.2$. In Fig. 7, the point O is the midpoint of $L_{i,i+k}$, and the coordinate of point O is

$$\begin{cases} X_O = \frac{1}{2}(X_i + X_{i+k}); \\ Y_O = \frac{1}{2}(Y_i + Y_{i+k}), \end{cases} \tag{7}$$

where (X_i, Y_i) and (X_{i+k}, Y_{i+k}) are the coordinates of point T_i and point T_{i+k}, respectively. The angle between line segment $L_{i,i+k}$ and the x-axis is θ_1, and the angle between line OP and the x-axis is θ_2 (Point P is the center point of the obstacle).

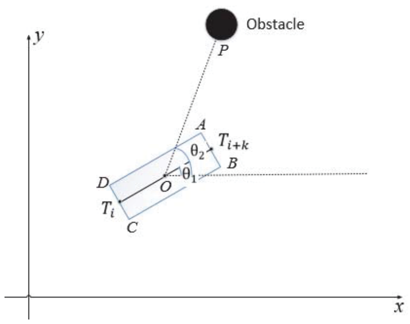


Fig. 7: Expand the line segment to rectangle.

Step 2: translation and rotation transformation.

Translate the center point O of rectangle $ABCD$ to the origin point and get a new point O. Move obstacles to the corresponding position in the same manner. Then, rotate rectangle $ABCD$ and the obstacle with the center point O as the rotation point, and rotate the θ_1 clockwise (Fig. 8). The coordinates of each point are

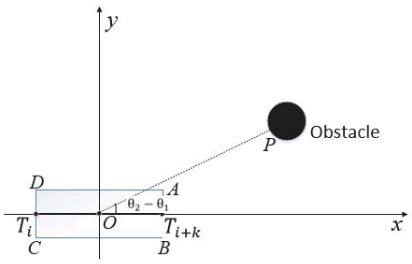


Fig. 8: Translation and rotation transformation.

$$\begin{cases} X_A = ||OT_{i+k}||; \\ Y_A = ||T_{i+k}A||, \end{cases} \tag{8}$$

and

$$\begin{cases} X_{Obs1} = ||OP||cos(\theta_2 - \theta_1); \\ Y_{Obs1} = ||OP||sin(\theta_2 - \theta_1). \end{cases} \tag{9}$$

After the translation and rotation transformation, we obtain the obstacle coordinate (X_{Obs1}, Y_{Obs1}). It is necessary to project the obstacle to the first quadrant for the convenience of calculation, and the obstacle coordinate after projecting is

$$\begin{cases} X_{Obs} = |X_{Obs1}|; \\ Y_{Obs} = |Y_{Obs1}|. \end{cases} \tag{10}$$

Step 3: judge the relative position of the rectangle and the circle.

Firstly, we get the vector $\overrightarrow{AP} = (X_{Obs} - X_A, Y_{Obs} - Y_A)$, and then determine the relative position of the obstacle to the rectangle according to the vector $\overrightarrow{AP}$. There are four relative relationships between rectangle $ABCD$ and the circle (Fig. 9):

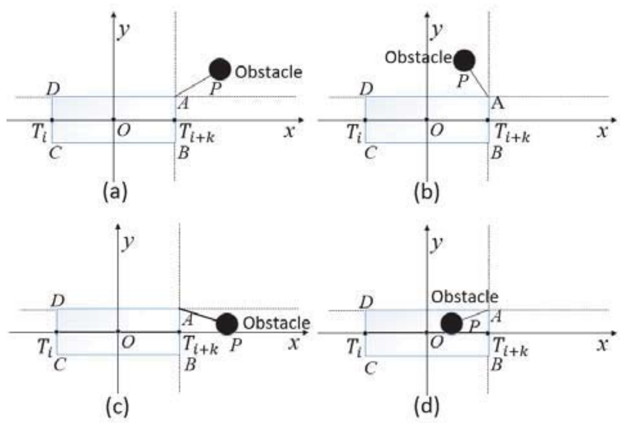


Fig. 9: The four relative relationships between rectangle $ABCD$ and the circle.

(a) If $X_{Obs} - X_A > 0, \quad Y_{Obs} - Y_A > 0$,

$$\begin{cases} x = X_{Obs} - X_A; \\ y = Y_{Obs} - Y_A; \\ d = \sqrt{x^2 + y^2}, \end{cases} \tag{11}$$

(b) If $X_{Obs} - X_A < 0, \quad Y_{Obs} - Y_A > 0$,

$$\begin{cases} x = 0; \\ y = Y_{Obs} - Y_A; \\ d = \sqrt{x^2 + y^2}, \end{cases} \tag{12}$$

(c) If $X_{Obs} - X_A > 0, \quad Y_{Obs} - Y_A < 0$,

$$\begin{cases} x = X_{Obs} - X_A; \\ y = 0; \\ d = \sqrt{x^2 + y^2}, \end{cases} \tag{13}$$

(d) If $X_{Obs} - X_A < 0, \quad Y_{Obs} - Y_A < 0$,

$$d = 0. \tag{14}$$

The value of d, which is calculated according to the above process, is the shortest distance between all the obstacles and the rectangle. We can judge the relative position of the obstacle to the rectangle by comparing d with the obstacle's radius r:

$$\begin{cases} d > r, \quad not \quad collide; \\ d \le r, \quad collide. \end{cases} \tag{15}$$

V. Simulation and Results

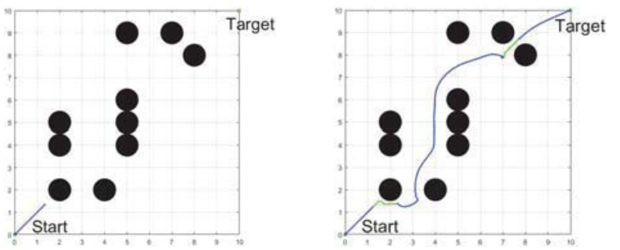
Numbers of simulations are done for Section III and Section IV using MATLAB 2018a, a 2.80GHZ CPU (Core i5; Inter Corp) with the Windows 10 OS. The simulation environment is a square with 10 width. All the obstacles' locations are known in advance. Table I shows simulation parameters of the improved APF method.

TABLE I: Simulation parameters of the improved APF method.

k_{att}	k_{rep}	$Stepsize$	ρ_0	d_0	$dist(x,y)_{max}$	r
5	15	0.1	2	2	2	0.5

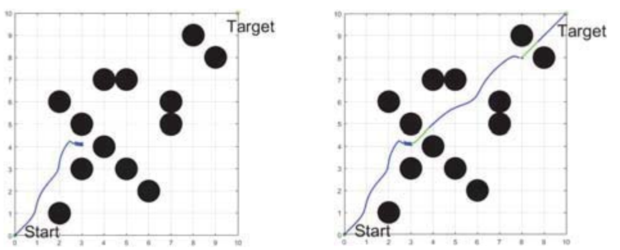
where k_{rep} is the repulsion constant, $Stepsize$ is the length of one step, ρ_0 represents the largest influence distance of a single obstacle, r the radius of the obstacles in the simulation environment.

In Fig. 10-12, the blue line represents the path planned by traditional APF method. When robot is trapped in local minima, improved APF method based on DWA is used and the path planned by the method is the green line. After robot escapes from local minima, we continue to use the traditional APF method because this method is simple and fast. We can see that the traditional APF method is trapped in local minima easily and can't reach the target, while the improved APF method based on DWA can escape from local minima, which proves that our improved method is effective to overcome the local minima problem in traditional APF method.



(a) Traditional APF. (b) Improved APF based on DWA.

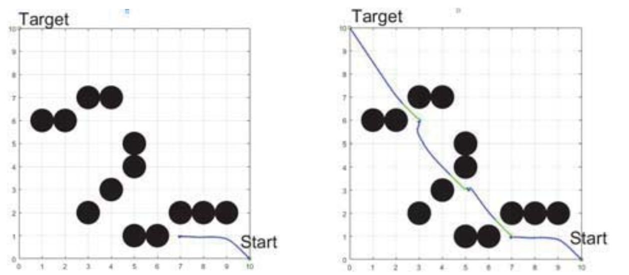
Fig. 10: Results of the two methods.



(a) Traditional APF. (b) Results of the two methods.

Fig. 11: Results of the two methods.

After obtaining the path, we optimize it to shorten its distance in four different scenes (Fig. 13). The blue and green line is original path before optimization while the red line is optimized path. The blue line consists of many points, so it seems like a curve. There are some twists and turns in the blue line, which is't suitable for robot to move along. The red line consist of several line segments. It is easier to control robot's movement along straight line than curve.



(a) Traditional APF. (b) Improved APF based on DWA.

Fig. 12: Results of the two methods.

Though our algorithm optimizes the planned path and shortens its distance, it only consume little time. As we can see in Fig. 14(a), the two methods take similar computational time, which means that our optimization algorithm doesn't need to consume much time. In Fig. 14(b), the distance of the path planned by our optimization algorithm is evidently shorter than the traditional APF. The distances of the path after optimizing are 90.96% , 87.70%, 87.84% and 93.29% of the path before optimization in four scenes, respectively.

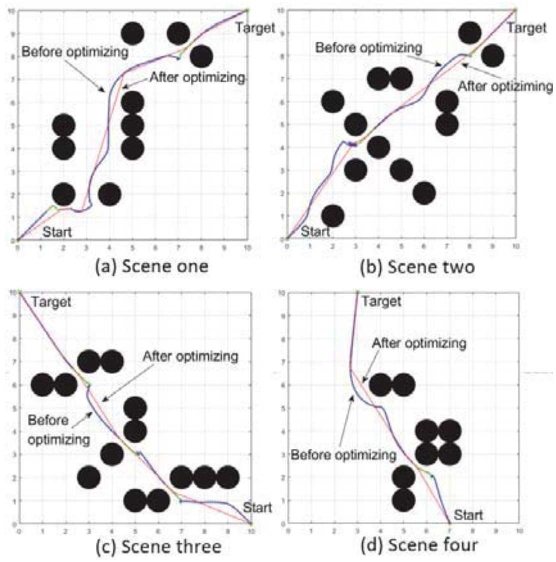
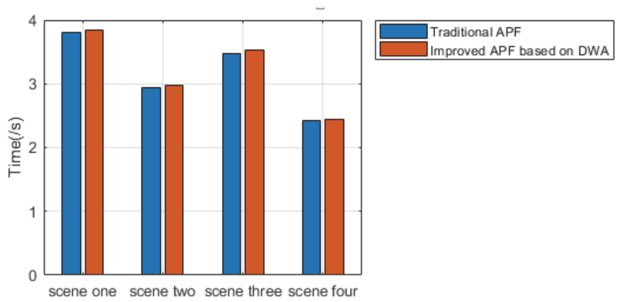


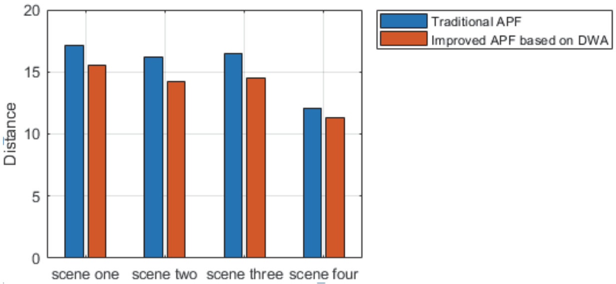
Fig. 13: Path optimization in four scenes.

VI. Conclusion

We proposed an improved APF method based on DWA, which can solve the local minima problem faced by traditional APF method. Besides, we also proposed an optimization algorithm to shorten the path while leaving enough secure space for the robot. The simulation results showed that the optimization algorithm can plan a shorter path while consuming little computational time. In the future works, we attend to modify



(a) Computational time of the two methods.



(b) Distance of the two methods.

Fig. 14: Computational time and distance of the two methods.

our method to plan path in an unknown environment and use it for the real robot.

REFERENCES

[1] E. Masehiam, and D. Sedighizadeh, "Classic and Heuristic Approaches in Robot Motion Planning - A Chronological Review," Word Academy of Science, Engineering and Technology 29, pp. 101-106, 2007.

[2] G. H. Li, S. G. Tong, G. Lv, R. Y. Xiao and F. Y. Cong, "An Improved Artificial Potential Field-based Simultaneous FORward Search (Improved APF-based SIFORS) method for Robot Path Planning," The 12th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI, 2015), pp. 330 335, 2015.

[3] H. E. Romeijn and R. L. Smith . "Simulated Annealing for Constrained Global Optimization," Journal of Global Optimization, pp. 101-124, 1994.

[4] O. Khatib, "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots," Autonomous Robot Vehicles, pp. 396-404, 1986.

[5] S. M. Lavalle, "Rapidly-exploring random trees: A new tool for path planning," ,1998.

[6]] F. Duchon, A. Babinec, M. Kajan and P. Beno, "Path planning with modified A star algorithm for a mobile robot," Procedia Engineering, pp. 59-69, 2014.

[7] C .J. Watkins and P. Dayan, "Q-learning," Machine learning, vol. 8, no. 3-4, pp. 279-292, 1992.

[8] M. G. Park and M. C. Lee, "Artificial potential field based path planning for mobile robots using a virtual obstacle concept," Proceedings of the 2003 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM 2003), pp. 735-740, 2003.

[9] M. G. Park, J. H. Jeon and M. C. Lee, "Obstacle avoidance for mobile robots using artificial potential field approach with simulated annealing," ISIE, Pusan Korea, pp. 1530-1535, 2001.

[10] P. Yan, Z. Yan, H. X. Zheng and J. F. Guo, "Real Time Robot Path Planning Method Based on Improved Artificial Potential Field Method," Proceedings of the 37th Chinese Control Conference, pp. 4814-4820, 2018.

[11] Q. S. Zhang, D. D. Chen, "An Obstacle Avoidance Method of Soccer Robot Based on Evolutionary Artificial Potential Field," 2012 International Conference on Future Energy, Environment, and Materials, pp. 1792-1798, 2012.

[12] D. Fox, W Burgard and S. Thrun, "The Dynamic Window Approach to Collision Avoidance," IEEE Robotics Automation Magazine, pp. 23-33, 1997.