Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2017) 25: 666 – 676

doi:10.3906/elk-1504-194

Obstacle avoidance for mobile robot based on improved dynamic window approach

Xiuyun LI[1,2], Fei LIU[3], Juan LIU[4], Shan LIANG[2,*]
[1]Chongqing Vocational Institute of Engineering, Chongqing, P.R. China
[2]College of Automation, Chongqing University, Chongqing, P.R. China
[3]National Center for Robot Testing and Evaluation (Chongqing), Chongqing, P.R. China
[4]China Coal Technology Engineering Group Chongqing Research Institute, Chongqing, P.R. China

Received: 22.04.2015 • **Accepted/Published Online:** 06.12.2015 • **Final Version:** 10.04.2017

Abstract: The dynamic window approach has the drawback that it may result in local minima and nonoptimal motion decision for obstacle avoidance because of not considering the size constraint of a mobile robot. Thus, an improved dynamic window approach is proposed, which takes into account the relation between the size of the mobile robot and the free space between obstacles. A laser range finder is employed to improve the ability of sensing and prediction of the environment, in order to avoid being trapped in a U-shaped obstacle, such as a box canyon. By applying the proposed method, the local minima problem can be solved and the optimal path can be obtained. The effectiveness and superiority is proven by theoretic analysis and simulations.

Key words: Improved dynamic window approach, obstacle avoidance, mobile robot, laser range finder

1. Introduction

For autonomous mobile robots, obstacle avoidance is the basic requirement for moving safely and executing tasks [1]. The two most used methods for avoiding obstacles are the artificial potential field (APF) and grid-based vector field histogram (VFH) [2–5]. The APF will control the motion by use of the sum of attractive and repulsive forces from goal and obstacles, respectively. However, it compresses all the environment information into a virtual force, which causes the loss of valuable information about obstacle distribution and results in the local minima problem [6]. Furthermore, it may lead to an unstable state in narrow spaces [2]. The VFH has better performance in narrow openings, but it does not consider the robot's size and kinematics [4]. Some other methods include the curvature-velocity method and basic dynamic window method [7-10], which may also result in local minima problems due to the lack of information about the connectivity of free space.

The dynamic window method is a widely used method for obstacle avoidance, which has good performance when the mobile robot moves at high velocity [8–10]. However, the local minima problem exists and may prevent the robot from real-time stable and smooth avoidance of the obstacle. Traditional strategies such as wall-following and modified APF methods may not solve this problem fast [1,11,12]. In addition, the size of the robot is not considered in these methods, which may lead to unfeasible paths and brings negative effects for global optimal path planning. In [14], Seder and Petrovic proposed a solution that integrates a focused D* search algorithm and dynamic window local obstacle avoidance algorithm with certain adaptations for the

*Correspondence: lightsun@cqu.edu.cn

avoidance of moving obstacles. Brock and Khatib proposed a global DWA that combines motion planning and real-time obstacle avoidance to achieve robust high-velocity, goal-directed, reactive motion for a robot in unknown and dynamic environments [15]. In [16], Arras et al. introduced an approach to obstacle avoidance and local path planning for polygonal robots, where a reduced dynamic window dealing with robot shape and dynamics is utilized.

Moreover, convergence to the goal was theoretically analyzed in [17–20]. Ögren and Leonard integrated the convergent Koditschek scheme with the fast reactive dynamic window approach (DWA) to make the robot attain the goal configuration [17,18]. Vista et al. proposed an objective function for the DWA that holds convergence to the goal [19]. Berti et al. proposed improvements over the dynamic window approach (I-DWA), utilized for computing in real-time autonomous robot navigation [20]. In [20], an objective function involving Lyapunov stability criteria was presented to guarantee the global and asymptotic convergence to the goal while avoiding collisions. In [21], a modified DWA was proposed for obstacle avoidance where stereovision is used. By defining a strict search space, the error of the stereoprocessing is controlled and steering control is smoothed, thus producing the optimal velocities for the robot. However, the local minima problem may still occur and the stereo increases the time and space complexity.

In this paper, an improved DWA is proposed to solve the obstacle avoidance problem. We first detect and analyze the connectivity of free space in front of the robot by a laser ranger finder (LRF), which can prevent the situation of being trapped in a U-shape obstacle. The basic DWA is then modified by considering the size of the robot in order to improve the safety and save time.

The rest of this paper is organized as follows: Section 2 describes the models of the environment and the robot. Section 3 describes the proposed improved dynamic window approach. In Section 4, simulations are conducted and the result is presented and discussed. Finally, the conclusion and future work are addressed in Section 5.

2. Problem statement

One of the classic obstacle avoidance methods is the dynamic window approach, which was first proposed by Fox et al. in 1997 [8]. The method is derived directly from the motion dynamics of the robot, in which the limitations of velocity and acceleration are considered. First, a velocity vector $(\nu,\ w)$ is defined where ν is the heading velocity and w is the rotational velocity. Then, to avoid colliding with obstacles, an admissible speed set is defined as

$$V_a = \{(v,w)|v \leq \sqrt{2dist(v,w)\dot{v}_b} \wedge w \leq \sqrt{2dist(v,w)\dot{w}_b}\}, \tag{1}$$

where $dist(\nu,\ w)$ is the closest distance from the robot to the obstacle, and $\dot{v}_b$ and $\dot{w}_b$ are corresponding accelerations for breakage. Eq. (1) can guarantee that the robot stops without hitting an obstacle.

When considering the constrained accelerations of the motors, the whole search can be reduced to the dynamic window that involves only the velocities that can be reached within the next time interval. The dynamic window V_d is defined as

$$V_d = \{(v,w)|v \in [v_a - \dot{v}t, v_a + \dot{v}t] \wedge w \in [w_a - \dot{w}t, w_a + \dot{w}t]\}, \tag{2}$$

where t is the time interval during which $\dot{v}$ and $\dot{w}$ are applied and $(v_a, w_a)$is the actual velocity. Eq. (2) shows that within the next time interval all the area the robot can reach is inside the dynamic window and thus obstacle avoidance outside the window can be ignored. As shown in Figure 1, the restricted search space

is $V_r = V_s \cap V_a \cap V_d$, where V_s contains all the available velocities. Finally, we choose a suitable value from V_r as the velocity in next time interval. To optimize the movement, we build an object function

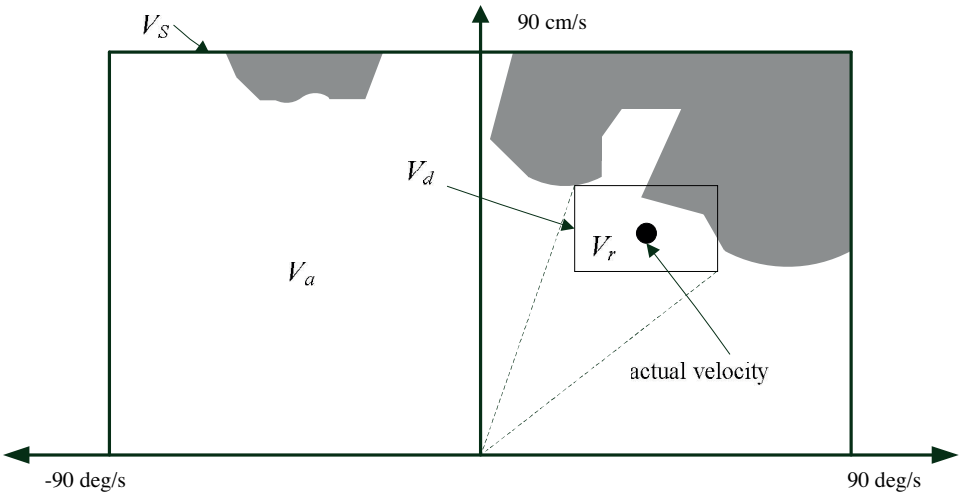

Figure 1. The dynamic window.

$$G(v,w) = \alpha \cdot heading(v,w) + \beta \cdot dist(v,w) + \gamma \cdot vel(v,w), \tag{3}$$

which will be maximized. In Eq. (3), $heading(\nu,\ w)$ is used to measure the progress towards the goal, which is maximal when the robot moves directly to the goal. $dist(\nu,\ w)$ is the closest distance to the obstacle and $vel(\nu,\ w)$ is the forward velocity. α, β, and γ are the weights.

This dynamic window approach has been proven to be well suited for robots operating at high speed and thus is widely applied. However, the local minima problem will occur since this method only takes into account

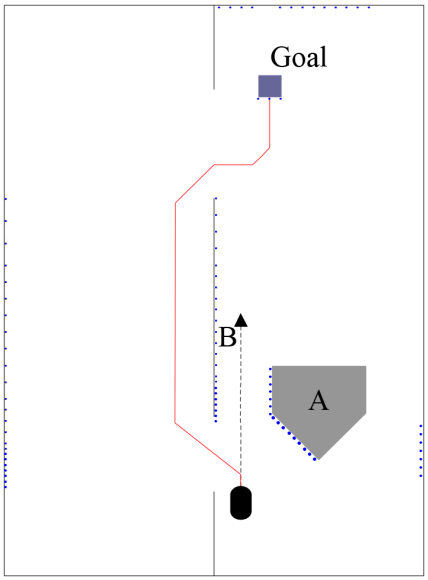

Figure 2. Application of the dynamic window approach.

the admissible velocities within the dynamic window, while the connectivity of the free space is not considered. Moreover, due to not considering the size of the robot, the path planned may not be optimal. In the case shown in Figure 2, when the DWA is utilized, the robot will abandon the preplanned path (the dashed arrow) because of the casual occurrence of obstacle A. Then the robot moves to the goal by following the newly planned path (the red line), which is much longer than the preplanned path. It is feasible for the robot to pass the opening between obstacles A and B. To solve this problem, we will take into account the size of the robot and the real-time sensing information.

3. Improved dynamic window approach

3.1. LRF-based traverse feasibility analysis

The LRF is a time of flight-based sensor that provides precise distance of obstacles within a range. Compared with image-based sensors, the measurement is not influenced by light, and it has the advantage of high accuracy [13]. A typical LRF, SICK LMS200, is used to check the environment information. With each turn the LMS200 can get 181 values from the distance to obstacles with an angular range from 0° to 180° and the resolution is 1°. The maximum measurement distance is 8 m and the distance resolution is 15 mm. The data obtained can be expressed using a polar coordinate method, e.g., the ith distance value s_i is

$$s_i = (\rho_i, \theta_i)^T, i = 1, ..., N, \tag{4}$$

where ρ_i is the distance between the sensor and the obstacle, and θ_i is the relative angle. The model of the LRF is shown in Figure 3.

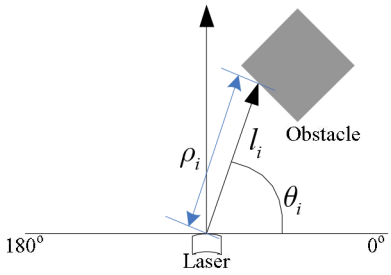

Figure 3. The model of a laser range finder.

The environment is described in a 2-dimension Cartesian coordinate system. Therefore, we use Eqs. (5) and (6) to map s_i into the robot's coordinate system and further convert it into the global coordinate system (GCS):

$$\begin{cases} x' = d + \rho \sin\theta \\ y' = -\rho \cos\theta \end{cases}, \tag{5}$$

$$\begin{cases} x = x_R + \sqrt{x'^2 + y'^2}\cos(\alpha + \alpha') \\ y = y_R + \sqrt{x'^2 + y'^2}\sin(\alpha + \alpha') \end{cases}, \tag{6}$$

where (x', y') is the location of the obstacle in the robot's coordinate system (RCS), and d is a constant that is the difference between the original points of the RCS and the sensor's coordinate system (SCS), which is obtained by the robot's physical structure. (x_R, y_R) and (x, y) represent the robot's coordinates in RCS and GCS. α is the angle of the robot's heading direction, and $\alpha' = \arctan(\frac{x'}{y'})$. The coordinates GCS $(X-O-Y)$,

RCS(X_R−O_R−Y_R), and SCS(X_S−O_S−Y_S) are shown in Figure 4. Finally, the SCS is converted into a grid-based map by

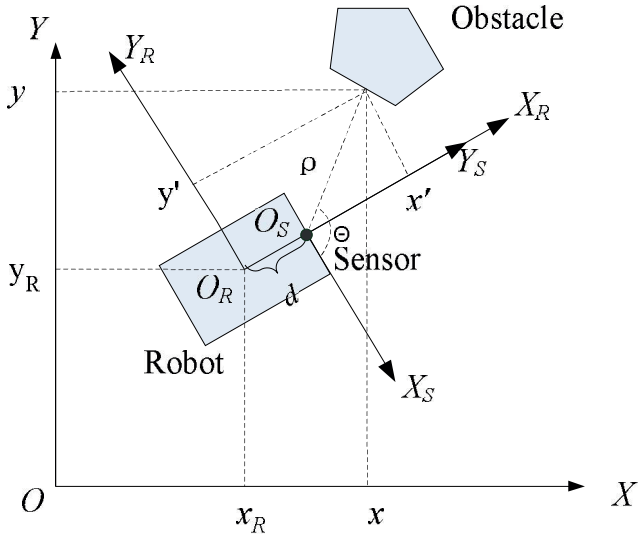

Figure 4. Convention of three coordinates.

$$\begin{cases} x_g = int(\frac{x}{w}) \cdot w + int(\frac{w}{2}) \\ y_g = int(\frac{y}{w}) \cdot w + int(\frac{w}{2}) \end{cases}, \tag{7}$$

where w is the width of the grid and (x_g, y_g) is the grid's coordinate in the GCS.

3.2. The improved dynamic window approach

The key to the improved dynamic window approach is the sensing of obstacles. We use a LRF, SICK LMS200, to measure the distance to obstacles. By using the scanned data, the analysis of traverse feasibility is conducted as shown in Figures 5 and 6. In Figure 5, the LMS200 is mounted in the front of the robot. The scanning area is divided into 6 regions, which are A, A', B, B', C, and C' respectively. For simplicity, we use AA', BB', and CC' to describe the regions A and A', B and B', and C and C', respectively. Thus, the angular ranges of AA', BB', and CC' are obtained as

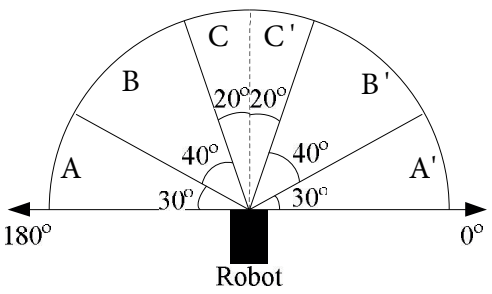

Figure 5. Partition of scanning regions.

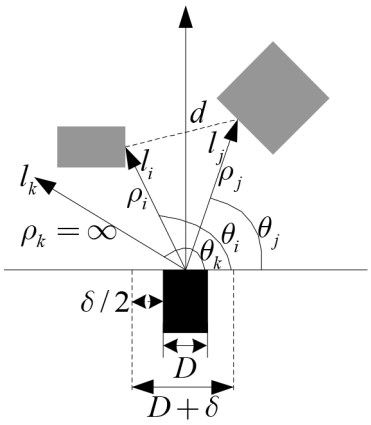

Figure 6. The robot and the laser beams.

$$\begin{cases} Ra(AA') = [0, 30^\circ) \cup (150^\circ, 180^\circ] \\ Ra(BB') = [30^\circ, 70^\circ) \cup (110^\circ, 150^\circ] \\ Ra(CC') = [70^\circ, 90^\circ) \cup (90^\circ, 110^\circ] = [70^\circ, 110^\circ] \end{cases}, \tag{8}$$

where $Ra(\cdot)$ represents the angular range of the corresponding region.

Figure 6 shows the model of the robot and laser's beam. The robot is modeled as rectangular, which is a typical model for robots. It is approximated to be a cylinder of size $D \times L$ and a safety clearance radius δ for movement, where δ depends on the velocity vector (v, w). In addition, three beams, l_i, l_j, and l_k, are shown in Figure 6. We define d as the distance between the two points produced by beams l_i and l_j shooting on obstacles, and then we obtain

$$d = \sqrt{\rho_i^2 + \rho_j^2 - 2\cos(\theta_j - \theta_i)\rho_i\rho_j}. \tag{9}$$

However, if one beam (e.g., l_k) cannot shoot anything, it means that the obstacle is out of the laser's range, and then we denote $\rho_k = \infty$.

As shown in Figure 7, with the partition of scanning regions AA', BB', and CC', the process of traverse feasibility analysis is executed as follows:

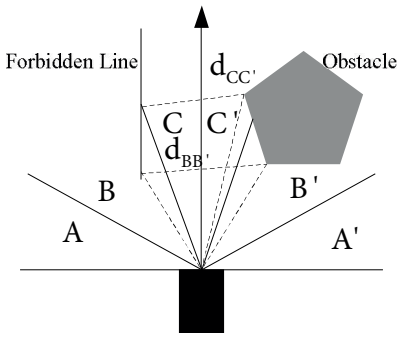

Figure 7. Traverse feasibility analyses.

(1) Region AA' is a protection area, which is used to prevent the robot from bumping into the obstacles on its two sides. Denote ρ_1 and ρ_{181} as the two measured distances of two beams of 0° and 180°. If $\rho_1 = \infty$ and $\rho_{181} = \infty$, the robot can move safely. If $\rho_1 = \infty$ and $\rho_{181} \neq \infty$, the robot should rotate left, and if $\rho_1 \neq \infty$ and $\rho_{181} = \infty$, the robot should rotate right. When $\rho_1 \neq \infty$ and $\rho_{181} \neq \infty$, the robot needs to stop or move backwards if

$$(\rho_1 + \rho_{181}) \leq c(D + \delta), \tag{10}$$

where c is a parameter that is determined by the robot's physical structure. When $(\rho_1 + \rho_{181}) > c(D + \delta)$, if $\rho_1 \leq \rho_{181}$, the robot will rotate left; otherwise, it will rotate right.

(2) Region BB' is a proximal prediction area. First we find out the minimal values $\rho_B^{\min}$ and $\rho_{B'}^{\min}$ in divisions B and B', respectively. Then with Eq. (9) we can obtain $d_{BB'}$. Then the robot can move forward if

$$d_{BB'} > \sqrt{c^2(D + \delta)^2}. \tag{11}$$

Otherwise, the robot cannot pass this narrow area and path replanning is needed.

(3) Region CC' is a distal prediction area, which implies the connectivity and describes the width $d_{CC'}$ that the robot can pass through the narrow area between obstacles of the distal region in the robot's forward motion. In order to obtain $d_{CC'}$, it is necessary to search the first noninfinity values ρ_C^1 and $\rho_{C'}^1$ in region C and C', respectively. If $\rho_C^1 \leq \rho_{C'}^1$, we find out the minimal value $\rho_{C'}^{\min}$ in region C' and thus we obtain $d_{CC'}$ with ρ_C^1 and $\rho_{C'}^{\min}$ and corresponding angles θ_C^1 and $\theta_{C'}^{\min}$. However, if $\rho_C^1 > \rho_{C'}^1$, $d_{CC'}$ is computed with $\rho_C^{\min}$ and $\rho_{C'}^1$ and the corresponding angles $\theta_{C'}^1$ and $\theta_C^{\min}$. Similarly, if

$$d_{CC'} > \sqrt{c^2(D+\delta)^2}, \tag{12}$$

the robot can move forward; otherwise, it needs to replan the path. If a noninfinity value is not found in region CC', the free space in this region is wide enough to be passed through. However, if this happens in regions BB' and CC', there may exist a local minimum area, and thus the robot should replan a local path to avoid the local minima.

When a dynamic obstacle is found that does not exist in the environment map, the obstacle avoidance mechanism works, which can be described as follows:

Step 1: The robot avoids collision with unforeseen obstacles by analyzing region AA'.

Step 2: Predict the feasibility of passing through the planned path by analyzing region BB'. If neither $\rho_B^{\min}$ nor $\rho_{B'}^{\min}$ exists, go to step 3. If both $\rho_B^{\min}$ and $\rho_{B'}^{\min}$ exist and Eq. (11) becomes true, go to step 3; otherwise, go to step 4.

Step 3: If neither $\rho_C^{\min}$ nor $\rho_{C'}^{\min}$ exists, go to step 4. If both $\rho_C^{\min}$ and $\rho_{C'}^{\min}$ exist and Eq. (12) is true, then go to step 5; otherwise, go to step 4.

Step 4: Replan the local path to avoid collision with obstacles since the opening gap between the obstacles is too narrow to pass through.

Step 5: Return to step 2.

In order to improve the ability to pass the openings between obstacles by traverse feasibility analysis above, a new feasibility evaluation function, $f(v,w,s)$, is defined as

$$f(v,w,s) = \alpha \cdot heading(v,w) + \beta \cdot dist(v,w) + \gamma \cdot vel(v,w) + \varepsilon width(v,w,s), \tag{13}$$

where $heading(v,w)$ is the heading direction, $dist(v,w)$ is the closest distance to obstacles, and $vel(v,w)$ is the velocity vector. s is the sensing data, ε is the weight, and $width(v,w,s)$ is

$$width(v,w,s) = \begin{cases} l, & if \quad d_{AA'} > c(D+\delta) or (\rho = \infty) \in AA' \\ 0, & if \quad d_{AA'} \leq c(D+\delta) \end{cases}, \tag{14}$$

where l is

$$l = \begin{cases} \frac{c^2(D+\delta)^2}{d_{BB'} d_{CC'}}, & if \quad d_{BB'} > c(D+\delta) and\, d_{CC'} > c(D+\delta); \\ \frac{c(D+\delta)}{d_{BB'}}, & if \quad d_{BB'} > c(D+\delta) and\, (\rho = \infty) \in CC'; \\ 0, & \text{else.} \end{cases} \tag{15}$$

The parameter $width(v,w,s)$ determines the possibility to pass through the opening area between obstacles. The situation $width(v,w,s) = 0$ implies that the robot cannot pass through the opening area between obstacles, and thus it needs to replan a new path. When $width(v,w,s) \neq 0$, the robot is able to pass through and the motion is controlled by parameters α, β, γ, and ε.

4. Simulation and result

Simulations are conducted in MATLAB and parameters are set as $\alpha = 0.5$, $\beta = 0.2$, $\gamma = 0.1$, $\varepsilon = 0.2$, $\omega = 10$ cm, $D = 6$ cm, and $L = 10$ cm. The robot moves at the speed of 15 cm/s when the free space in front of the robot is wide enough to pass. When closing to any obstacle, the robot moves at the speed of 10 cm/s, and when the robot needs to rotate, the forward velocity and rotation velocity are set as 5 cm/s and $\pi/4$ rad s^{-1}. In simulations, the DWA and the proposed improved dynamic window approach (IDWA) are applied respectively for each case. Simulation results are shown in Figures 8–10.

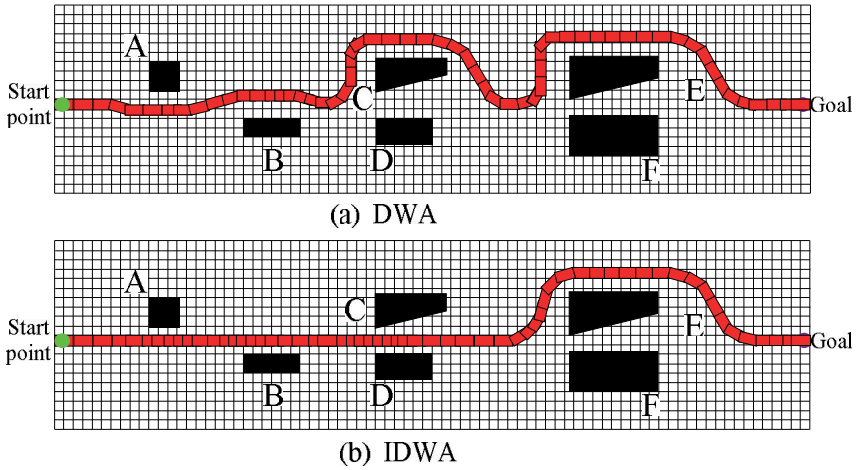

Figure 8. Result of simulation I.

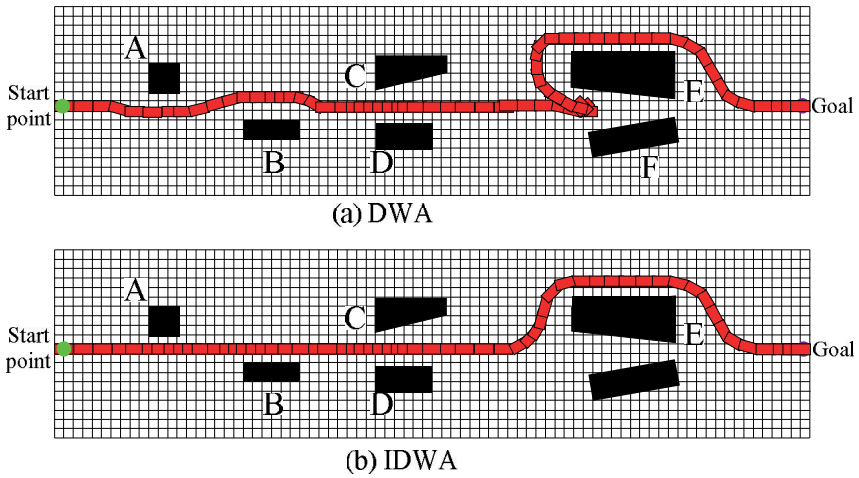

Figure 9. Result of simulation II.

Figures 8a and 8b show the result that applies DWA and IDWA, respectively. The major difference occurs when facing obstacles C and D. By using DWA, the robot moves around obstacle C according to the optimal decision. However, using IDWA, Figure 8b shows that the robot can pass the narrow passage between obstacles C and D after taking into account the size of robot. Thus, the path becomes much longer according to DWA.

The result of simulation II is shown in Figures 9a and 9b, which use DWA and IDWA, respectively. Compared with simulation I, the space between C and D becomes larger and there is a U-shaped trap between

obstacles E and F. We can see from Figure 9 that the robot can pass through the gap between C and D in both cases. However, when facing the U-shaped local minima, DWA cannot make the robot sense this trap before entering it, which is shown in Figure 9a. Thus, the robot must turn around and replan the path to the goal, which costs more time and makes the robot traverse longer. In Figure 9b, by employing IDWA, the robot will find the U-shaped trap before moving into that area and replan the path to the goal in a timely manner, which saves a lot of time.

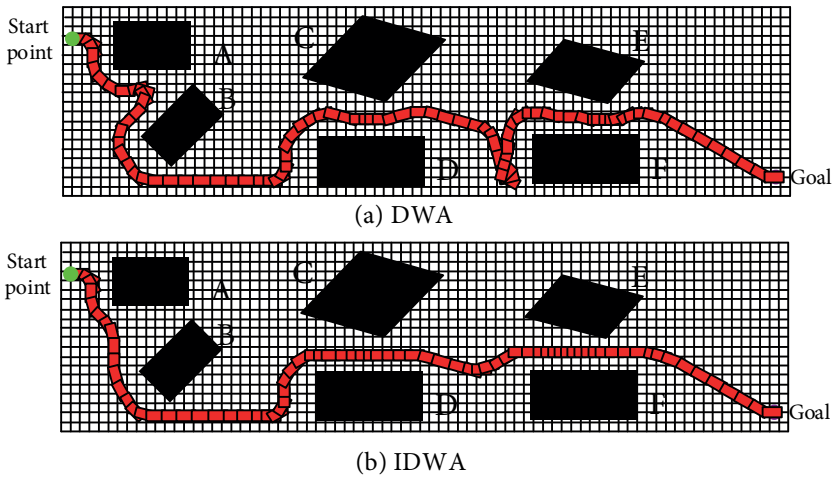

Figure 10. Result of simulation III.

Figure 10 shows the result of simulation III, in which the start point, the goal, and the distribution of obstacles are different from simulations I and II. The result indicates that when facing obstacles A and B, by using IDWA, the robot can avoid moving into the local minima and thus uses less time and path length than when using DWA. Furthermore, Figure 10a shows that the robot finds it impossible to pass through the gap between obstacle F and the border. However, Figure 10b shows that by considering the size and the detection of the gap between obstacle F and the border, the robot can avoid wasting time to get to the border and find the feasible path between E and F.

The three simulations above show that IDWA can solve the local minima problem while avoiding obstacles and thus produces an optimal path to the goal. Moreover, we list the time cost and the total path length in each case in the Table, in which comparison implies that by applying IDWA, the robot spends less time, and the path length is much shorter. In all, IDWA proves to be better than DWA.

Table. Comparison of simulation results.

Simulations	DWA		IDWA	
	Path length (cm)	Time (s)	Path length (cm)	Time (s)
I	512	54	430	42
II	486	46	430	37
III	544	68	506	52

5. Conclusion and future work

This paper proposes an improved DWA to solve the problem of obstacle avoidance. The major modification lies in the consideration of the size of robot and the development of a new feasibility evaluation function of the width of the opening space between obstacles. The simulations show that it can solve the local minima problem

and improve the ability to deal with narrow passages, and thus proves to be more capable of global optimal path planning while avoiding obstacles. In future work, a more complicated environment will be employed to verify and improve the proposed strategy. Finally, we will apply it to a real robot.

Acknowledgment

This work was financially supported by the Natural Science Foundation of the Chongqing Municipal Education Commission, China (Grant No. KJ1403209).

References

[1] Liu F, Liang S, Xian X, Bi H. Oscillation elimination for mobile robot based on behavior-memorizing. ICCI Express Letters 2011; 5: 3109-3115.

[2] Koren Y, Borenstein J. Potential field methods and their inherent limitations for mobile robot navigation. In: Proceedings of the IEEE Conference on Robotics and Automation; 7-12 April 1991; Sacramento, CA, USA. New York, NY, USA: IEEE. pp. 1398-1404.

[3] Borenstein J, Koren Y. The vector field histogram-fast obstacle avoidance for mobile robots. IEEE J Robot Autom 1991; 7: 278-288.

[4] Morales Y, Carballo A, Takeuchi E, Aburadani A, Tsubouchi T. Autonomous robot navigation in outdoor cluttered pedestrian walkways. J Field Robots 2009; 26: 609-635.

[5] Ulrich I, Borenstein J. VFH: Local obstacle avoidance with look ahead verification. In: IEEE International Conference on Robotics and Automation; 24–28 April 2000; San Francisco, CA, USA. New York, NY, USA: IEEE. pp. 2505-2511.

[6] Cai Z, Zheng M, Zou X. Real-time obstacle avoidance for mobile robots strategy based on laser radar. J Cent South Univ 2006; 37: 324-329.

[7] Simmons R. The curvature-velocity method for local obstacle avoidance. In: Proceedings of the 1996 IEEE International Conference on Robotics and Automation; 1996; Piscataway, NJ, USA. New York, NY, USA: IEEE. pp. 3375-3382.

[8] Fox D, Burgard W, Thrun S. The dynamic window approach to collision avoidance. IEEE Robot Autom Mag 1997; 4: 23-33.

[9] Kiss D, Tevesz G. Advanced dynamic window based navigation approach using model predictive control. In: 2012 International Conference on Methods and Models in Automation and Robotics; 27–30 August 2012; Międzyzdroje, Poland. New York, NY, USA: IEEE. pp. 148-453.

[10] Saranrittichai P, Niparnan N, Sudsang A. Robust local obstacle avoidance for mobile robot based on dynamic window approach. In: International Conference on Electrical Engineering and Electronics, Computer, Telecommunications and Information Technology; 15–17 May 2013; Krabi, Thailand. New York, NY, USA: IEEE. pp. 1-4.

[11] Yan Y, Zhang Y. Collision avoidance planning in multi-robot based on improved artificial potential field and rules. In: Proceedings of the 2008 IEEE International Conference on Robotics and Biomimetics; 21–26 February 2009; Bangkok Thailand. New York, NY, USA: IEEE. pp. 1026-1031.

[12] Nooraliei A, Iraji R. Robot path planning using wavefront approach with wall-following. In: International Conference on Computer Science and Information Technology; 8–11 August 2009; Beijing, China. New York, NY, USA: IEEE. pp. 417-420.

[13] Ren L. Obstacle perception and obstacle-avoiding strategy research of mobile robot based on laser range finder. Heilongjiang, China: Harbin Institute of Technology, 2007.

[14] Seder M, Petrovic I. Dynamic window based approach to mobile robot motion control in the presence of moving obstacles. In: Proceedings of IEEE International Conference on Robotics and Automation; 10–14 April 2007; Rome, Italy. New York, NY, USA: IEEE. pp. 1986-1991.

[15] Brock O, Khatib O. High-speed navigation using the global dynamic window approach . In: IEEE International Conference on Robotics and Automation; 10–15 May 1999; Detroit, MI, USA. New York, NY, USA: IEEE. pp. 341-346.

[16] Arras K O, Persson J, Tomatis N, Siegwart R. Real-time obstacle avoidance for polygonal robots with a reduced dynamic window. In: Proceedings of the 2002 IEEE International Conference on Robotics & Automation; 11–15 May 2002; Washington, DC, USA. New York, NY, USA: IEEE. pp. 3050-3055.

[17] Ögren P, Leonard NE. A convergent dynamic window approach to obstacle avoidance. IEEE T Robotic Autom 2005; 21: 188-195.

[18] Ögren P, Leonard NE. A tractable convergent dynamic window approach to obstacle avoidance. In: IEEE/RS International Conference on Intelligent Robots and Systems; 29 October–3 November 2002; Lausanne, Switzerland. New York, NY, USA: IEEE. pp. 595-600.

[19] Vista FP 4th, Singh AM, Lee DJ, Chong KT. Design convergent dynamic window approach for quadrotor navigation. Int J Precision Eng 2014; 15: 2177-2184.

[20] Berti H, Sappa AD, Amennoni OE. Improved dynamic window approach by using Lyapunov stability criteria. Lat Am Appl Res 2008; 38: 289-298.

[21] Zhang HQ, Dou LH, Fang H, Chen J. Autonomous indoor exploration of mobile robots based on door-guidance and improved dynamic window approach. In: Proceedings of the 2009 IEEE International Conference on Robotics and Biomimetics; 19–23 December, 2009; Guilin, China. New York, NY, USA: IEEE. pp. 408-413.