Enhanced DWA algorithm for local path planning of mobile robot

Xin Lai
School of Mechanical and Electrical Engineering, Southwest Petroleum University, Chengdu, China; Oil and Gas Equipment Technology Sharing and Service Platform of Sichuan Province, Chengdu, China and The State Key Laboratory of Oil and Gas Reservoir Geology and Exploitation, Southwest Petroleum University, Chengdu, China, and

Dan Wu, Di Wu, Jia He Li and Hang Yu
School of Mechanical and Electrical Engineering, Southwest Petroleum University, Chengdu, China

Abstract

Purpose – The purpose of this study is to solve the problems of poor stability and high energy consumption of the dynamic window algorithm (DWA) for the mobile robots, a novel enhanced dynamic window algorithm is proposed in this paper.

Design/methodology/approach – The novel algorithm takes the distance function as the weight of the target-oriented coefficient, and a new evaluation function is presented to optimize the azimuth angle.

Findings – The jitter of the mobile robot caused by the drastic change of angular velocity is reduced when the robot is closer to the target point. The simulation results show that the proposed algorithm effectively optimizes the stability of the mobile robot during operation with lower angular velocity dispersion and less energy consumption, but with a slightly higher running time than DWA.

Originality/value – A novel enhanced dynamic window algorithm is proposed and verified. According to the experimental result, the proposed algorithm can reduce the energy consumption of the robot and improves the efficiency of the robot.

Keywords DWA algorithm, Evaluation function, Dynamic window, Path planning, Azimuth, Autonomous robots

Paper type Research paper

1. Introduction

With the development of computer, 5 G and artificial intelligence technologies, the intelligence of monitoring systems in the oilfield petrochemical and other industries is gradually improving, and oilfield inspection robots have become one of the indispensable tools to improve the oilfield automation. The inspection robot relies on the on-board equipment to perform inspection operation independently, and path planning is one of the pivotal technologies of the robot. The inspection robot can avoid various obstacles in the dynamic and unknown environment through local planning and reach the destination smoothly according to the specified route.

At present, the local path planning algorithms include the potential field algorithm (Chen *et al.*, 2016; Chen *et al.*, 2015), vector domain histogram algorithm (Borenstein and Koren, 1991; Chen *et al.*, 2019), neural network algorithm (Yang and Luo, 2004), virtual force field algorithm (Oussama, 1986; Borenstein and Koren, 1989), genetic algorithm (Ahmed and Deb, 2013; Luo *et al.*, 2020), sampling-based planning algorithm (LaValle and Kuffner, 2000; Lin *et al.*, 2017) and dynamic window algorithm (DWA) (Fox *et al.*, 1997). The DWA algorithm, proposed by Fox in 1997, is a local reaction avoidance technique based on velocity space. By sampling groups of velocities in velocity space, the trajectory of mobile robot in a certain time under these velocities is simulated and evaluated, and then the velocity corresponding to the optimal trajectory is selected to drive the robot. The inspection robot often performs the goal-oriented tasks in the dynamic and unknown environments, and the working environment is complex, so the smoothness of the robot operation is particularly important. However, DWA algorithms in real-time obstacle avoidance mostly focus on avoiding obstacles and the velocity of the robot when passing, ignoring the stability of its own operation.

The performance of DWA algorithm will decrease when the obstacle moves dynamically, numerous improved algorithms have been proposed based on DWA algorithms. Lee *et al.* (2021) proposed the finite distribution estimation-based dynamic window approach, and it can estimate the velocity, position and distribution of obstacles through finite memory filtering, predict the state of obstacles and enable the robot to maintain good performance even in dynamic obstacle environments. Chou and Lian (2009) proposed a local reactive method (DWA*) based on

The current issue and full text archive of this journal is available on Emerald Insight at: **https://www.emerald.com/insight/0143-991X.htm**

Industrial Robot: the international journal of robotics research and application
50/1 (2023) 186–194

[DOI 10.1108/IR-05-2022-0130]

This work was supported in part by the Open Fund of the State Key Laboratory of Oil and Gas Reservoir Geology and Exploitation under Grant PLN2020-10, in part by the Sichuan Science and Technology Department Application Foundation Project 2019YJ0311, in part by the Extracurricular Open Experiment Program of Southwest Petroleum University 2021KSZ03022.

Received 14 March 2022
Revised 3 July 2022
26 July 2022
Accepted 28 July 2022

DWA algorithm. Region analysis technology is applied to filter out inappropriate commands, and A* algorithm is used to directly select the optimal velocity command from the reserved candidate speeds, which can effectively avoid the robot driving into local minima area and trapping in complex environment. Liu *et al.* (2019) introduced the evaluation factor about direction change into the path evaluation function of the original DWA algorithm. This method suppresses the excessive influence of a particular factor on the evaluation function in a specific situation and reduces the unnecessary steering frequency of the robot, which makes the robot turning path smoother and the path planning more reasonable.

The weights of the original DWA algorithm are constant, but the environment where robot works is dynamic and complex, and different situations require different weights. Zhang *et al.* (2015) proposed a modified DWA combined with fuzzy logic, and used the fuzzy logic to determine the appropriate weights in real time by analyzing the information of targets and obstacles, making the robot motion safer and more stable. Chang *et al.* (2021) proposed an improved dynamic window method based on Q-learning. The algorithm adds two new functions and combines Q-learning with DWA to adjust the weights of each evaluation function in DWA, and corrects the deficiencies of the original evaluation function, so that the robot has higher navigation efficiency and success rate in complex and unknown environments. However, the existing DWA algorithms ignore the chattering problem of mobile robots caused by drastic changes in angular velocity when they are close to the target point.

In this paper, a novel enhanced DWA (EDWA) algorithm is proposed to reduce the jitter caused by the unstable robot operation, a new evaluation function is used to optimize the azimuth, and the performance of the improved algorithm is verified by simulation. The proposed algorithm optimizes its azimuth evaluation function based on DWA algorithm. The contribution points of this work are as follows:

- A new direction angle function is put forward by considering the factors of direction angle and distance. The distance is taken as the evaluation coefficient to control the azimuth function, and the method effectively avoids the instability of the mobile robot caused by excessive increase of angular velocity to reduce the angle difference.
- The proposed evaluation function is used to optimize the directional angle for local path planning, and the distance function is used as the weight coefficient of the target orientation to regulate the whole evaluation function, which alleviates the jitter of the mobile robot caused by the dramatic change of angular velocity when the DWA algorithm is close to the target point.
- The experimental results show that, compared with the DWA algorithm, the EDWA algorithm enables the robot to effectively avoid dynamic obstacles in dynamic obstacle environments, while making the running route smoother. The algorithm is applicable to sudden motion obstacles (MO) in complex environments, with higher quality of path planning as well as better stability.

2. Dynamic window algorithm

DWA algorithm (Fox *et al.*, 1997) is to sample multiple sets of velocities in the velocity space by considering the kinematic performance of the robot, and simulate the trajectories of these velocities within a certain period of time. Multiple sets of trajectories are obtained, and then evaluation function is established to score these trajectories. The optimal speed velocity of the mobile robot at the next moment is screened out, and the optimal velocity is sent to the lower computer to drive the robot along the smooth collision free path corresponding to the velocity group. The evaluation function mechanism is pivotal in obtaining the optimal simulated motion trajectory of the robot.

The core components of the DWA algorithm include a mobile robot motion model, velocity sampling and evaluation functions. The trajectory can be simulated by sampling speed and evaluation functions. After obtaining the sampled velocities in the velocity vector space under the constraints, an evaluation function mechanism is required to filter the optimal velocity of the mobile robot at the next moment, and obtain the optimal simulated motion trajectory.

Because there are many velocity combinations in the velocity vector space, it is necessary to restrict to reduce the complexity of velocity sampling from the following three aspects.

1. Because the mobile robot is limited by the maximum velocity (v_{max}, ω_{max}) and the minimum velocity (v_{min}, ω_{min}), the sampling speed is limited as follows:

$$v_c = \{v \in [v_{max}, \omega_{max}], [v_{min}, \omega_{min}]\} \tag{1}$$

2. The maximum acceleration and deceleration speed of mobile robot is limited to (a_{max}, δ_{max}) by motor torque due to the influence of motor performance. During the period Δt when the mobile robot is simulated, because of the dynamic window, the actual velocity V_d of the mobile robot is:

$$\begin{aligned} V_d = \{v \in [v_0 - a_{max}\ \Delta t, v_0 + a_{max}\ \Delta t], \\ \omega \in [\omega_0 - \delta_{max}\ \Delta t, \omega_0 + \delta_{max}\ \Delta t]\} \end{aligned} \tag{2}$$

where (v_0, ω_0) represents the current velocity of the mobile robot.

3. To avoid collisions with obstacles, mobile robot needs to reduce velocity to zero before colliding with obstacles in practice. According to the above constraints, the velocity of the mobile robot is limited to a maximum range, which can be expressed as follows:

$$V_m = \{v \le \sqrt{2 * dist(v, \omega) * a_{max}},\ \omega \le \sqrt{2 * dist(v, \omega) * \delta_{max}}\} \tag{3}$$

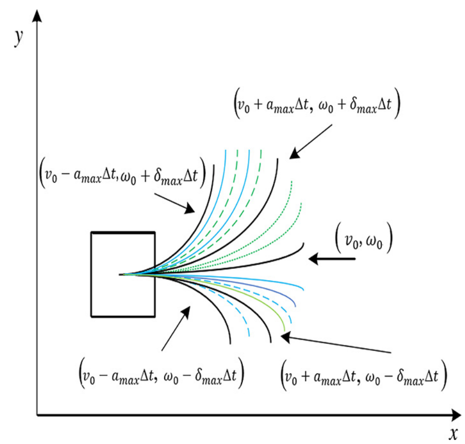
where $dist$ (v, ω) represents the shortest distance between the mobile robot and the obstacles at (v, ω) velocity.

The generated smaller velocity vector space is represented by V_r as follows:

$$Vr = V_c \cap V_d \cap V_m \tag{4}$$

To simplify the calculation, the velocity of the robot in the time domain of the forward simulated trajectory is set to be constant until a new velocity task is reassigned. The sampling trajectory of the dynamic window is shown in Figure 1.

Figure 1 Sampling trajectory of the dynamic window



The evaluation function of DWA is defined as:

$$G(v, \omega) = \alpha \cdot heading(v, \omega) + \beta \cdot dist(v, \omega) + \gamma \cdot vel(v, \omega) \quad (5)$$

where α, β, γ represent the weight coefficients corresponding to the direction angle function, distance function and velocity function of the evaluation function, respectively, and their value range is [0, 1].*dist* (v, ω) denotes the degree of distance between the robot and the nearest obstacle on the simulated trajectory corresponding to the velocity (v, ω). The larger the value, the less likely the mobile robot collides with the nearest obstacle *dist* (v, ω) can be defined by:

$$dist(v, \omega) = \begin{cases} l/L, & 0 < l < L \\ 1, & l \geq L \end{cases} \quad (6)$$

where l indicates the distance from the nearest obstacle on the current simulated trajectory and L indicates the setting safe distance of the mobile robot.

heading (v, ω) denotes the azimuth function and can be expressed as:

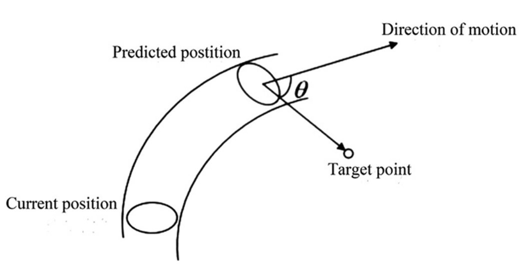
$$heading(v, \omega) = 1 - \theta/\pi \quad (7)$$

where θ is the angle difference. The directional angular motion of the robot is shown in Figure 2.*vel* (v, ω) is the velocity evaluation function and defined as:

$$vel(v, \omega) = v/v_{max} \quad (8)$$

where v is the current sampling velocity of the mobile robot and v_{max} is the maximum velocity in the velocity vector space.

Figure 2 Schematic diagram of directional angular motion



3. Enhanced dynamic window algorithm

The difference between the proposed algorithm and the DWA algorithm lies in the evaluation function. The EDWA algorithm optimizes its azimuth evaluation function based on DWA algorithm. A novel directional angle functions are proposed considering both directional angle and distance effects.
The proposed evaluation function of EDWA is defined as:

$$G(v, \omega) = \alpha \cdot Dh(v, \omega) + \beta \cdot dist(v, \omega) + \gamma \cdot vel(v, \omega) \quad (9)$$

$Dh(v, \omega)$ denotes a new direction angle evaluation function, which can be represented as:

$$Dh(v, \omega) = Dist(v, \omega) \cdot heading(v, \omega) \quad (10)$$

$$Dist(v, \omega) = (dist(v, \omega))/(\max\ dist(v, \omega)\ ) \quad (11)$$

where *Dist* (v, ω) represents the new distance function, it is the distance between the end of the trajectory corresponding to the velocity (v, ω) and the target position, *max dist* (v, ω) represents the distance from the set starting point to the end point.

Because the evaluation function of the mobile robot is discontinuous in the running process, the value of the evaluation function of one of the items is more obvious. The evaluation function is normalized as:

$$normal\_head(i) = heading(i)/\sum_{i=1}^{n} heading(i) \quad (12)$$

$$normal\_dist(i) = dist(i)/\sum_{i=1}^{n} dist(i) \quad (13)$$

$$normal\_vel(i) = vel(i)/\sum_{i=1}^{n} vel(i) \quad (14)$$

where i is the current trajectory that needs to be evaluated, and N is the number of all samples. The higher the value of the evaluation function, the better the trajectory. With the fusion of three factors in the evaluation function: directional angle function, distance function and velocity evaluation function, a collision-free optimal path with the smallest angular direction to the target position and the fastest speed is selected.

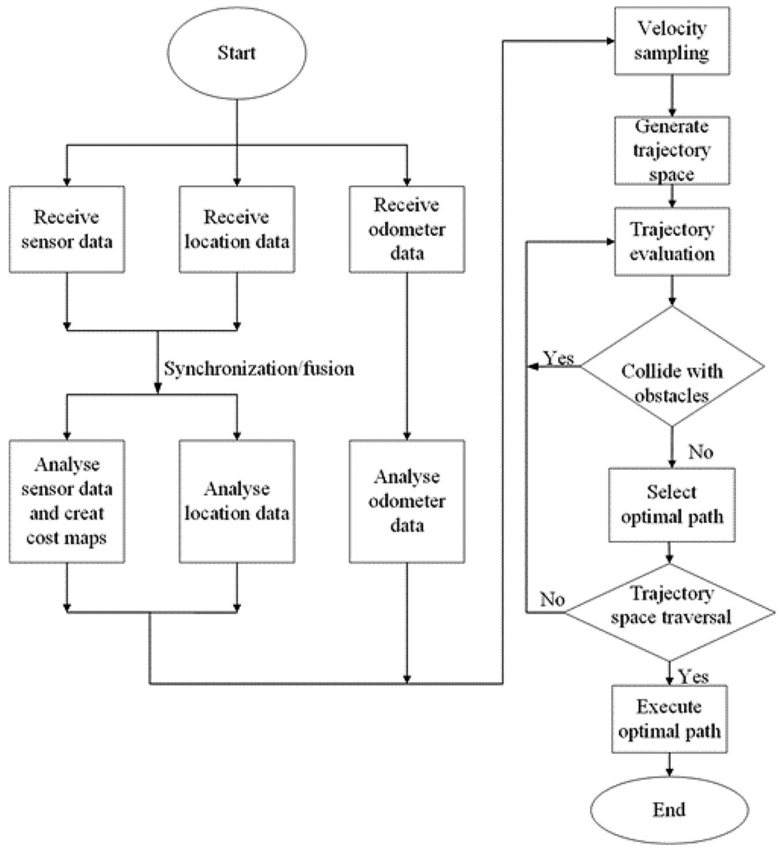
The proposed algorithm enables robots to plan a more reasonable obstacle avoidance path in the complex multidynamic and unknown environments, and makes the path smoother and the mobile robot more stable, which reduces the energy consumption and improves the stability of mobile robots. The execution flow of the proposed algorithm is shown in Figure 3.

4. Simulation

To verify the effectiveness of the EDWA algorithm, simulation is carried out on MATLAB. The values of parameters in the paper are set as follows: the maximum value of linear velocity v_{max} = 1.0 (m/s), the maximum value of angular velocity ω_{max} = 2.0 (m/s), acceleration a = 3.0 (m/s^2), rotational acceleration b = 4.0 (rad/s^2); the value of dv is 5 m/s for linear velocity resolution and $d\omega$ is 6 rad/s for angular velocity resolution; the unit time dt = 0.1 s, and the values of each factor coefficient in the evaluation function are: α = 0.4, β = 0.3, γ = 0.4.

Industrial Robot: the international journal of robotics research and application
Volume 50 · Number 1 · 2023 · 186–194

Figure 3 EDWA algorithm execution steps



Simulation experiments are conducted for the EDWA and DWA algorithms. The map size is 10 m × 10 m, the coordinates of global target points are (10,10), the number of obstacles is 20 and the influence radius of obstacles is 0.5 m. The simulation moves in different directions with $\omega = [0,2 * pi]$ rad/s, $v = [0,0.5]$ m/s. The initial state of the mobile robot at the starting position: the linear velocity $v = 0$ m/s, the angular velocity $\omega = 0$ rad/s, the attitude angle $\theta = 0.1\ \pi$. The simulation results of the evaluation functions of EDWA and DWA are shown in Figures 4 and 5, the purple circles represent the dynamic obstacles in the environment map in the figures.

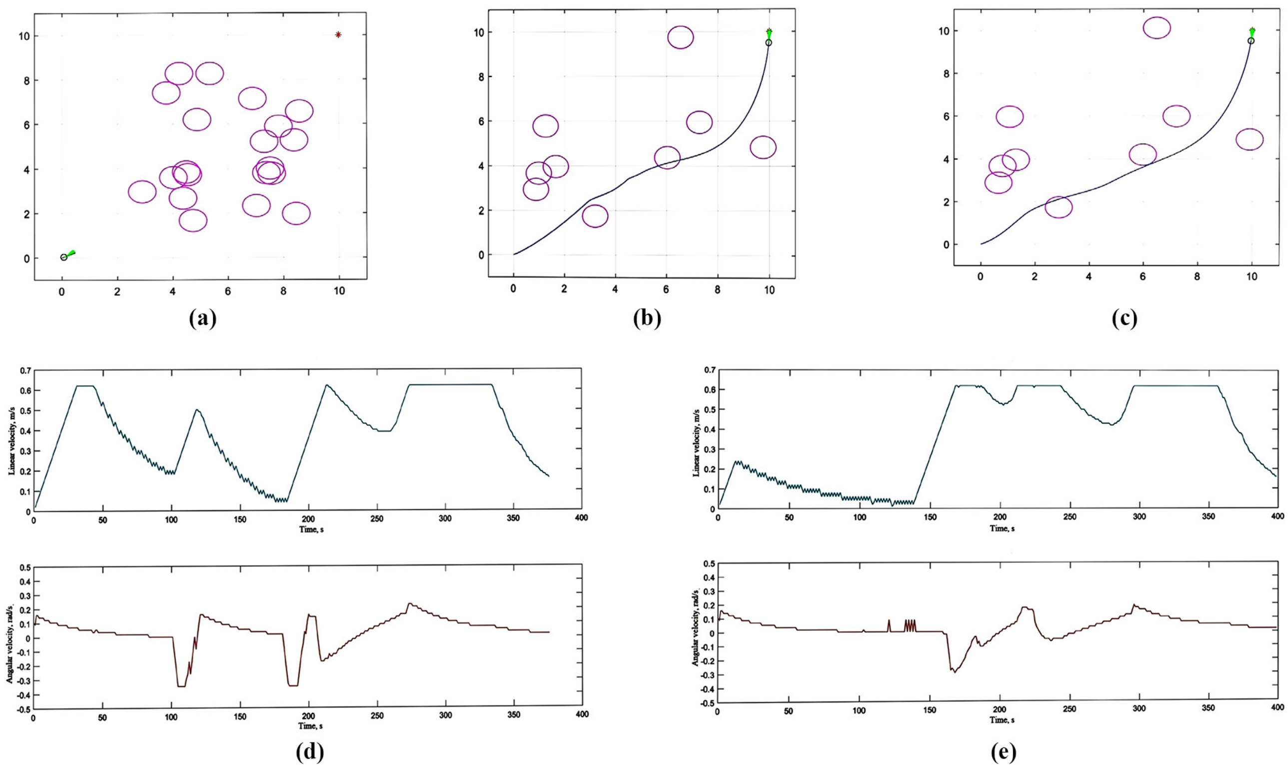
It can be seen from Figure 4(b) and 4(c) that the proposed EDWA algorithm enables the robot to travel along a smoother path to reach the target point. We can see from Figure 4(d) that when the DWA algorithm is used, the angular velocity of the robot changes dramatically twice in a row over time period 40–190 s when it encounters an obstacle after a period of motion, and the linear velocity also changes simultaneously with the angular velocity. In Figure 4(e), when the proposed algorithm is used, the angular velocity shows a more stable decrease and the linear velocity is also in a more stable floating state when the robot encounters an obstacle in the initial motion. From the initial state of Figure 4(a), it can be seen that the obstacles are denser near the half-way of travel. The angular and linear velocity change dramatically in the DWA algorithm, whereas in the EDWA algorithm, the variation of angular velocity and linear velocity is significantly reduced, and the situation is greatly improved. When approaching the target point, there is no obvious change in the comparison of angular velocity and linear velocity changes under the two algorithms, maybe due to fewer obstacles.

It can be seen from Figure 5(d) that during the first half of the robot motion, the angular velocity undergoes two large fluctuations over the time intervals [0–150] due to the dense obstacles, and the linear velocity experiences the same two fluctuations. There are almost no obstacles in the middle part of the route, and the angular velocity decreases more smoothly, and the linear velocity also shows a stable state, with only slight fluctuations over the time intervals [230–250]. It can be seen from Figure 5(e) that the proposed algorithm does not show large fluctuations in the angular velocity of the robot in this path, and there is only a slight fluctuation at 240 s. Comparing the linear velocity curves in Figure 5(d) and 5(e), it can be seen that the linear velocity of the proposed algorithm is generally more stable than that of the original DWA algorithm, and the fluctuation range is smaller. In Scene 2, the number of line velocity changes is less and the magnitude is smaller because the number of obstacles is less than that of Scene 1.

It can be concluded that, taking the distance function as the weight coefficient of target orientation, the overall change of the robot's angular velocity in the EDWA algorithm is smoother than that in the original evaluation function when encountering obstacles, and the angular velocity does not appear to fluctuate up and down drastically. The linear velocity of the robot is controlled within a stable range even if it changes. Especially the linear velocity of the robot in Scene 2, the EDWA algorithm

Industrial Robot: the international journal of robotics research and application
Volume 50 · Number 1 · 2023 · 186–194

Figure 4 Simulation of path planning of the proposed EDWA and DWA algorithm (Scene 1)



Notes: (a) Original state; (b) DWA algorithm trajectory; (c) EDWA algorithm trajectory; (d) The linear velocity and angular velocity of DWA algorithm; (e) The linear velocity and angular velocity of EDWA algorithm

enables the linear velocity hardly changes in the second half of the robot's operation, and the curve is relatively smooth during the deceleration process near the destination. It can be seen that the overall operation stability has been greatly increased. Compared with the DWA's evaluation function, the simulation shows that the proposed evaluation function can better control the velocity of the robot in driving and considerably improve the stability of the robot operation. Table 1 shows the experimental data of the EDWA algorithm and the DWA algorithm.

The standard deviation reflects the degree of dispersion of the data set, and smaller data indicates less dispersion of the data and less fluctuation of the angular velocity.

The improvement of angular velocity stability is calculated by:

$$\xi = (\sigma 1 - \sigma 2)/\sigma 1 \tag{15}$$

where σ_1 and σ_2 represent the DWA standard deviation of angular velocity and the EDWA standard deviation of angular velocity, respectively. Figure 6 shows the motion stability and time consumption of the EDWA algorithm and DWA algorithm.

It can be seen from Figure 6 that in two experiment scenes, the angular velocity stability of the proposed algorithm increases about 31.7% and 21.9%, respectively, and the running time also increases 6.12% and 9.71%, respectively, compared to that of the DWA algorithm. The proposed algorithm makes the angular velocity of the robot path planning more stable but higher computational complexity, which reduces the robot's energy and improves the robot's efficiency.

We counted the number of drastic changes in linear velocity and angular velocity from Figures 4 and 5. The threshold value for angular velocity is 0.005 rad/s and for linear velocity is 0.005 m/s. The number of dramatic changes in angular velocity and linear velocity are shown in Table 2.

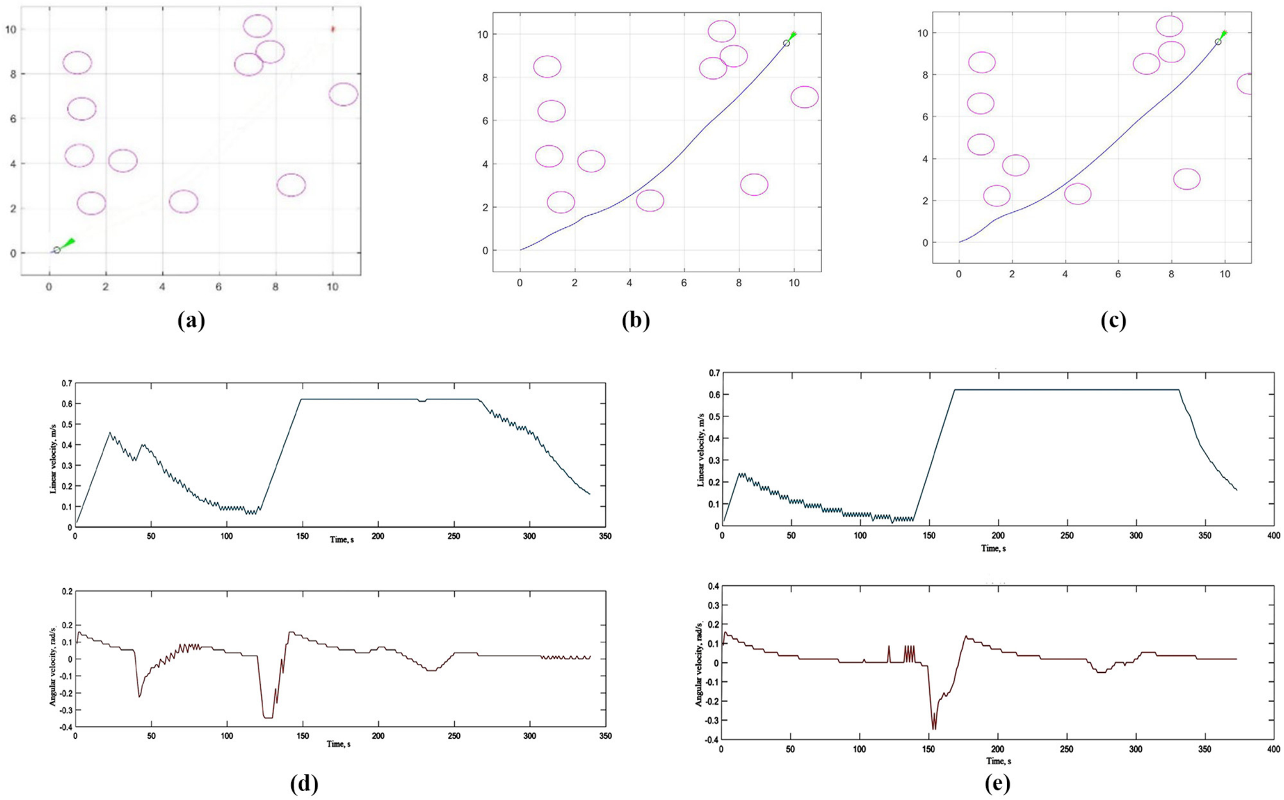
As shown in Figure 7, the number of violent changes in angular velocity is reduced by about 14.3% and 50% in Scene 1 and Scene 2, respectively, corresponding to a reduction of about 16.7% and 60% in the number of violent changes in angular velocity, respectively. It can be concluded that the reduction of the number of velocity changes of the robot improves the stability of the robot.

5. Experiment

The robot used for the experiments in this paper is a small two-wheel differential speed mobile robot developed based on ROS. The EDWA and DWA algorithms are used for path planning to avoid the sudden appearance of dynamic basketball obstacles during robot motion, as shown in Figure 8. By collecting the environment information by LiDAR in real time, the A* algorithm is used as the global planning algorithm, and the proposed DWA algorithm and DWA algorithm are used as the real-time obstacle avoidance algorithm for local planning.

Figure 9 shows the local path planning of the robot when it encounters a dynamic obstacle. When the dynamic obstacle

Figure 5 Simulation of path planning of the proposed EDWA and DWA algorithm (Scene 2)

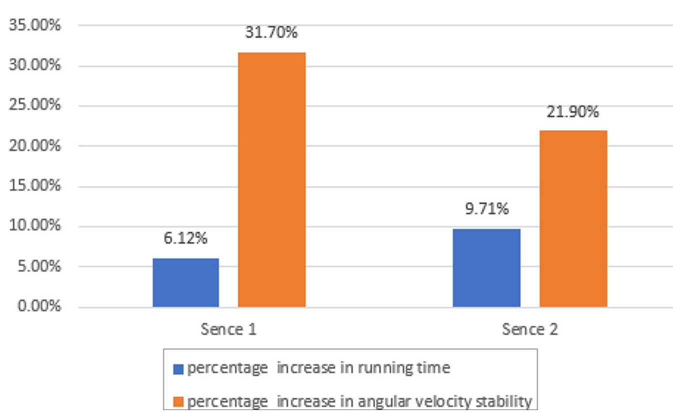


Notes: (a) Original state; (b) DWA algorithm trajectory; (c) EDWA algorithm trajectory; (d) The linear velocity and angular velocity of DWA algorithm; (e) The linear velocity and angular velocity of EDWA algorithm

Table 1 EDWA algorithm and DWA algorithm experimental data

Algorithm	Experiment scene 1	Experiment scene 2
DWA standard deviation of angular velocity	0.1123	0.0834
EDWA standard deviation of angular velocity	0.0767	0.0651
DWA running time (s)	376	340
EDWA running time (s)	399	373

Figure 6 Comparison of motion stability and time consumption of EDWA algorithm



Industrial Robot: the international journal of robotics research and application
Volume 50 · Number 1 · 2023 · 186–194

Table 2 EDWA algorithm and DWA algorithm number of dramatic changes in angular velocity and angular velocity

Algorithm	Experiment scene 1	Experiment scene 2
DWA number of dramatic changes in angular velocity	7	4
EDWA number of dramatic changes in angular velocity	6	2
DWA number of dramatic changes in angular velocity	6	5
EDWA number of dramatic changes in linear velocity	5	2

Figure 7 Comparison of the number of changes in angular velocity and linear velocity of EDWA algorithm

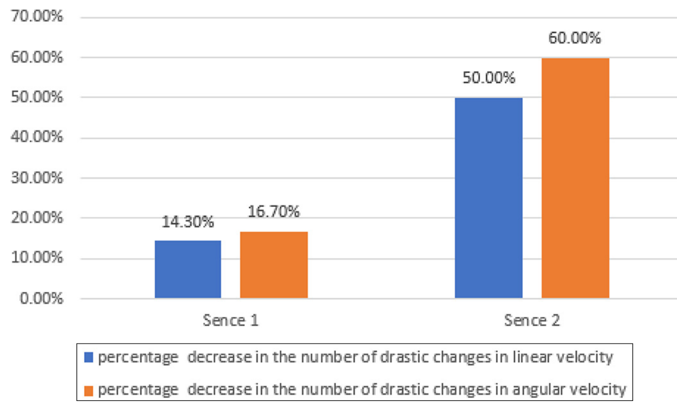


Figure 8 Experiment environment with obstacle

appears, the robot executes the DWA algorithm to avoid the obstacle. When the dynamic obstacle avoidance is completed, the A^* algorithm needs to be called again and so on until the robot reaches the target location. Figure 9(a)–(f) successively shows the action response and path planning trajectory of the robot when it encounters a dynamic obstacle. The robot decelerates when a sudden obstacle is detected, and stops briefly and then retreats a long distance along the oblique rear when the obstacle approaches. The robot deflects slightly to the left and then drives forward over the obstacle to the end point. MO is the dynamic obstacle scanned by LiDAR, the black curve is the local planning path, and the blue curve is the global planning path by A^* algorithm.

The local path planning of EDWA algorithm is smoother as shown in Figure 10. The EDWA algorithm enables the robot decelerates when a sudden obstacle is detected. The robot stops for a short period of time and then backs up a short distance when the obstacle is approaching. The robot keeps the original direction and drives forward over the obstacle to reach the destination when the obstacle passes in front of the robot. Comparing Figure 9(c)–(f) with Figure 10(c)–(f), it can be clearly seen that DWA algorithm make the robot move backward a large distance, and then plans a relatively tortuous path through local path planning. However, the EDWA algorithm moves backward only a small distance, and the path is smoother after the replanning. The angular velocity and linear velocity of the robot are shown in Figure 11. It can be seen from that the robot motion is more stable with the proposed DWA algorithm, and the linear and angular velocities do not fluctuate as much as the DWA algorithm, indicating that the path is smooth and easy for the robot to travel. The robot can successfully reach the set target position, and the robot runs more stably during dynamic obstacle avoidance under the EDWA.

6. Conclusion

The proposed EDWA algorithm is proposed to improve the evaluation function for the problem of the jitter and unstable driving of the mobile robot. The new azimuthal evaluation function is used to optimize the directional angle function to make the robot run more smoothly, and the distance function is used as the weight coefficient of target orientation to avoid the jitter vibration. Simulation results verify that the EDWA algorithm can effectively avoid the dynamic obstacles along a smoother route in the dynamic obstacle environments. The proposed algorithm is applicable to the sudden appearance of moving obstacles in the complex environments, which makes the path quality of the path planning algorithm higher and improves the stability of the local path planning of the mobile robot.

Figure 9 Local path planning based on DWA algorithm

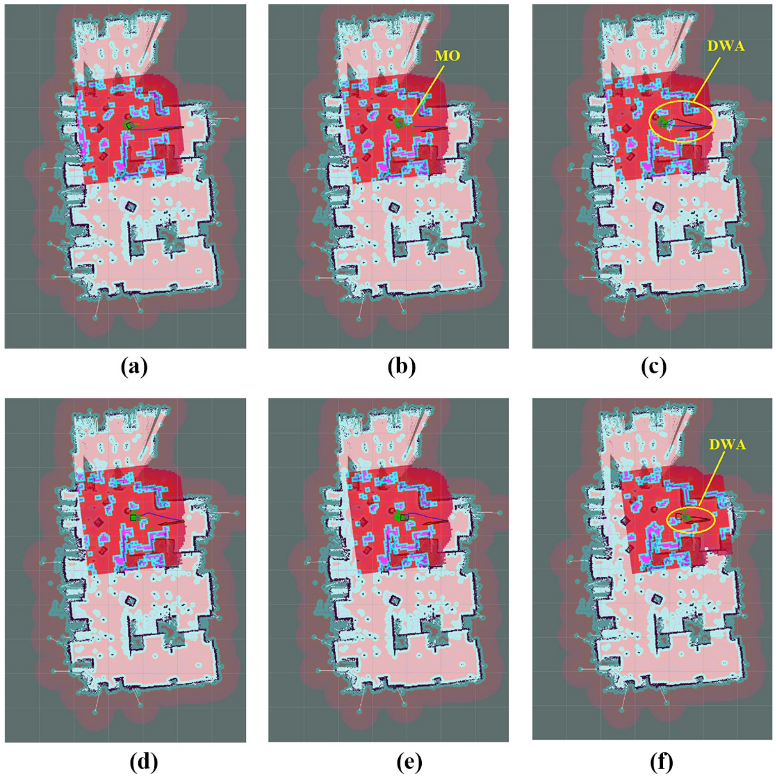


Figure 10 Local path planning based on EDWA algorithm

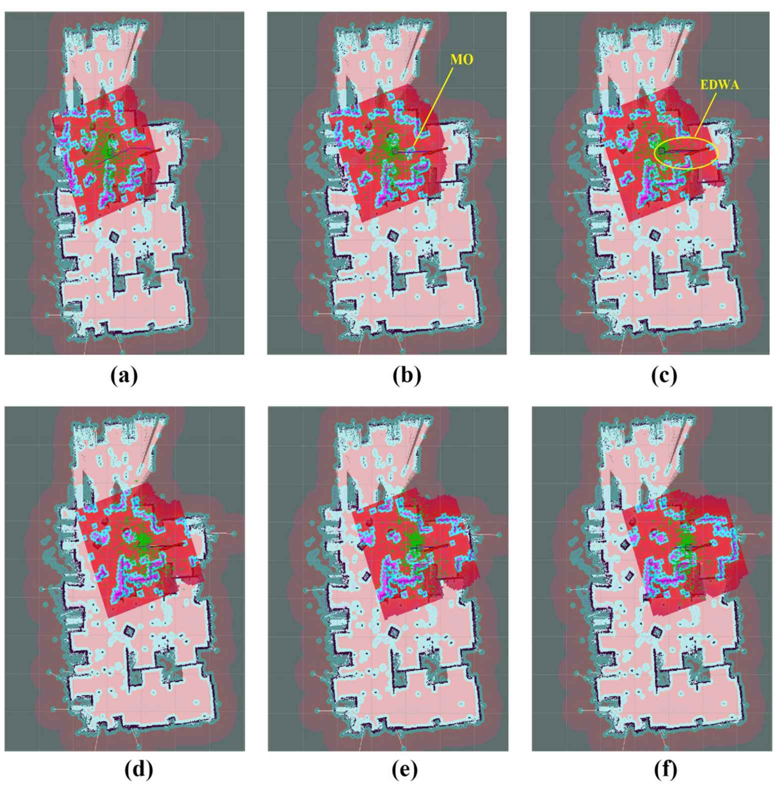
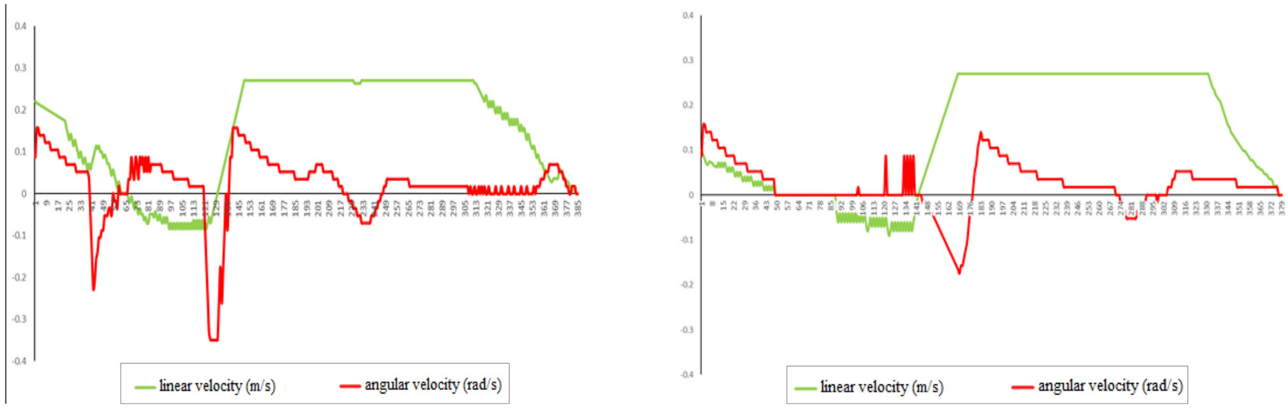


Figure 11 Speed sampling for local path planning based on DWA algorithm and EDWA algorithm



(a) (b)

Notes: (a) DWA algorithm; (b) EDWA algorithm

References

Ahmed, F. and Deb, K. (2013), "Multi - objective optimal path planning using elitist non-dominated sorting genetic algorithms", *Soft Computing*, Vol. 17 No. 7, pp. 1282-1299.

Borenstein, J. and Koren, Y. (1989), "Real-time obstacle avoidance for fast mobile robots", *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 19 No. 5, pp. 1179-1187.

Borenstein, J. and Koren, Y. (1991), "The vector field histogram-fast obstacle avoidance for mobile robots", *IEEE Transactions on Robotics and Automation*, Vol. 7 No. 3, pp. 278-288.

Chang, L., Shan, L., Jiang, C. and Dai, Y. (2021), "Reinforcement based mobile robot path planning with improved dynamic window approach in unknown environment", *Autonomous Robots*, Vol. 45 No. 1, pp. 51-76.

Chen, Y., Luo, G., Mei, Y., Yu, J. and Su, X. (2016), "UAV path planning using artificial potential field method updated by optimal control theory", *International Journal of Systems Science*, Vol. 47 No. 6, pp. 1407-1420.

Chen, W., Wang, N., Liu, X. and Yang, C. (2019), "VFH* based local path planning for mobile robot", *IEEE VFH* Based Local Path Planning for Mobile Robot*, pp. 18-23.

Chen, W., Wu, X. and Lu, Y. (2015), "An improved path planning method based on artificial potential field for a mobile robot", *Cybernetics and Information Technologies*, Vol. 15 No. 2, pp. 181-191.

Chou, C. and Lian, F. (2009), "Velocity space approach with region analysis and look-ahead verification for robot navigation verification for robot navigation", *Joint 48th IEEE Conference on Decision and Control (CDC)/28th Chinese Control Conference (CDC/CCC 2009)*, pp. 5971-5976.

Fox, D., Burgard, W. and Thrun, S. (1997), "The dynamic window approach to collision avoidance", *IEEE Robotics & Automation Magazine*, Vol. 4 No. 1, pp. 23-33.

LaValle, S. and Kuffner, J. (2000), "Rapidly-exploring random trees: progress and prospects", *Algorithmic and Computational Robotics: New Directions*, pp. 293-308.

Lee, D., Lee, S., Ahn, C., Shi, P. and Lim, C. (2021), "Finite distribution estimation-based dynamic window approach to reliable obstacle avoidance of mobile robot", *IEEE Transactions on Industrial Electronics*, Vol. 68 No. 10, pp. 9998-10006.

Lin, D., Shen, B., Liu, Y., Alsaadi, F.E. and Alsaedi, A. (2017), "Genetic algorithm-based compliant robot path planning: an improved Bi-RRT-based initialization method", *Assembly Automation*, Vol. 37 No. 3, pp. 261-270.

Liu, T., Lian, F., Yan, R. and Wei, G. (2019), "Local path planning algorithm for blind-guiding robot based on improved DWA algorithm", *31st Chinese Control and Decision Conference (CCDC)*, pp. 6169-6173.

Luo, Q., Wang, H., Zheng, Y. and He, J. (2020), "Research on path planning of mobile robot based on improved ant colony algorithm", *Neural Computing and Applications*, Vol. 32 No. 6, pp. 1555-1566.

Oussama, K. (1986), "Real-time obstacle avoidance for manipulators and mobile robots", *The International Journal of Robotics Research*, Vol. 5 No. 1, pp. 90-98.

Yang, S. and Luo, C. (2004), "A neural network approach to complete coverage path planning", *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, Vol. 34 No. 1, pp. 718-724.

Zhang, H., Sun, C., Zheng, Z., An, W. and Zhou, D. (2015), "A modified dynamic window approach to obstacle avoidance combined with fuzzy logic", *International Symposium on Distributed Computing and Applications for Business, Engineering and Science*, pp. 523-526.

Corresponding author

Xin Lai can be contacted at: xinlai06@163.com