2022 The 22nd International Conference on Control, Automation and Systems (ICCAS 2022)
BEXCO, Busan, Korea, Nov. 27~Dec. 01, 2022

Improved Dynamic Window Approach using the jerk model

Ziang Lin[1*] and Ryo Taguchi[2]

[1] Department of Computer Science, Nagoya Institute of Technology,
Nagoya, Japan (z.lin.553@stn.nitech.ac.jp) * Corresponding author
[2] Department of Computer Science, Nagoya Institute of Technology,
Nagoya, Japan (taguchi.ryo@nitech.ac.jp)

Abstract: The Dynamic Window Approach (DWA) method is a collision avoidance strategy for mobile robots that is optimized to take a robot to its goal while maintaining a minimum distance from any obstacle. However, owing to its velocity invariance assumption, it generates narrow paths that may not enable it to avoid obstacles. In addition, it is impossible to control the jerk. To address these issues, we propose a method for generating speed change paths based on jerk models. First, we limited the maximum jerk in the model. Second, we used the model to generate variable speed sequences to obtain a wider range of paths. We added jerk as a term in the evaluation function to prevent consistently large jerks. We verified the effectiveness of the proposed method by comparing the results with the conventional method through simulation experiments.

Keywords: Robot, DWA, Jerk control

1. INTRODUCTION

Against the background of labor shortages, the usage of robots has increased significantly as they are utilized extensively in various fields. Robots are not only used in conventional AGVs and robot arms in factories [1-2], but also as guiding robots and catering robots in airports and restaurants [3]. Because the general environment is usually more complex than a factory environment, it is important to guarantee the safety of the autonomous movement of robots.

Autonomous control consists of three elements: mapping, self-position estimation, and path planning [4-6]. Path planning consists of both global and local path planning. Global path planning plans a route from a starting point to a target on a map. A*[7], D*[8], and RRT*[9] are well-known algorithms for global path planning. Local path planning generates collision-free paths in real time, following a global path. The vector field histogram (VFH) and dynamic window approach (DWA) are frequently used as local path-planning algorithms [10-11]. The DWA generates candidate speed commands for the robot, considering the kinematic parameters of the robot. It then determines the coordinates at which each command will be executed and selects the optimal speed based on the evaluation results of the coordinates.

However, the DWA uses the robot's current speed as a reference and samples velocity pairs within the range of speeds that can be attained in a short time. When velocity pairs are used to predict a robot's path in a short period, the generated paths, and their arrival points are concentrated within a certain range because a constant speed is maintained. Therefore, it may not be possible to generate a path that avoids all the obstacles. Kobayashi proposed a method that combined virtual manipulators and DWA [12]. This method generates two types of virtual manipulators. One is called the Leader manipulator and is responsible for path tracking. The other is called the Assistant manipulator, which avoids collisions by providing speed in the opposite direction of the obstacle when the distance between the robot and obstacle is smaller than a threshold value. In this method, when the DWA cannot generate an avoidance path, the assistant manipulator avoids collision. However, the avoidance speed may cause the robot to stop or decelerate suddenly that is, jerk.

The DWA uses an acceleration model and does not control the jerk. When switching between acceleration and deceleration, a large jerk may occur. If the jerk exceeds a certain value, the robot may fall over. Haschke proposed a method for controlling a robot arm using a jerk model [13]. In their method, the motion of the robot arm is divided into several states, the initial position, velocity, and time of the current state and the initial velocity, position, and time of the next state are known; the velocity change between the two states is planned by using a jerk model. This method requires an understanding of the overall motion of the robot. However, this method cannot be applied to local path algorithms to avoid dynamic obstacles.

This paper proposes a method to solve the problem wherein the paths generated by DWA are centralized and the problem wherein DWA cannot control jerk by using the jerk model. Section 2 describes the DWA, Section 3 describes the proposed method, and Section 4 evaluates the effectiveness of the proposed method through comparative experiments between the proposed method and original methods in each environment.

2. DYNAMIC WINDOW APPROACH (DWA)

DWA is a local path-planning algorithm. The path is planned considering the speed parameters of the robot. In the DWA, the current velocity v_t of the robot is first used as a reference, under the limits of acceleration a and angular acceleration according to Eq. (1), the lower

and upper limits of speed v_{t+1} that can be attained in a short time constitute the dynamic window (DW).

$$v_{t+1} = \begin{cases} v_t + a \cdot \Delta t & n = 0 \\ v_t & n \neq 0 \end{cases} \quad a \in [-a_{max}, a_{max}] \qquad (1)$$

Δt is the time interval between the prediction steps. An image of the DW is shown in Fig. 1. The DW is a continuous velocity interval that requires discretization of the velocity space because velocity sampling is required to generate the trajectory. As shown in Fig. 2, the velocity up to t_0 is indicated by the black line, sampled at t_0, accelerated between $t_0 \sim t_1$, and the velocity remains unchanged after t_1. The possible velocity changes from t_0 are indicated by dashed lines. By selecting multiple velocity and angular velocity pairs (v, ω) in the discretized DW, and combining the predictions of Δt in multiple steps using the kinematic model in Eqs. (2) ~ (4), the coordinates at time t can be predicted.

$$x_{t+1} = x_t + v\Delta t\cos(\theta_t) \qquad (2)$$
$$y_{t+1} = y_t + v\Delta t\sin(\theta_t) \qquad (3)$$
$$\theta_{t+1} = \theta_t + \omega\Delta t \qquad (4)$$

The current state of the robot is described as (x_t, y_t, θ_t), and the next time state is described as $(x_{t+1}, y_{t+1}, \theta_{t+1})$. As shown in Fig. 3, the set of coordinates represents the predicted trajectory.

DWA evaluates the predicted trajectory using the evaluation function defined in Eq. (5). The evaluation function consists of the difference between the robot's direction and the target's azimuth angle heading(v, ω), distance dist (v, ω) from the obstacle, and velocity vel(v, ω).

$$G(v, \omega) = \sigma(\alpha \cdot \text{heading}(v, \omega) + \beta \cdot \text{dist}(v, \omega) + \gamma \cdot \text{vel}(v, \omega)) \qquad (5)$$

The original methods assumed that the speed of the robot was invariant during the forecast period. This causes the arrival points of the generated paths to concentrate within a certain range. Therefore, the generated paths may not allow the robot to avoid obstacles by accelerating or decelerating. In addition, the acceleration model did not limit the change in acceleration per unit of time. This can cause large acceleration changes. Considering Eqs. (6) ~ (8) and the second law of motion, a large acceleration causes an increase in the force. In particular, acceleration changes in a short period may provide a large impact force to the robot that may cause the robot to sway or fall over [14-16].

$$a = \int j\,dt \qquad (6)$$
$$I = \int F\,dt \qquad (7)$$
$$F = ma \qquad (8)$$

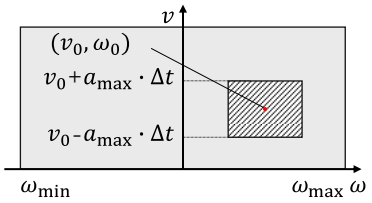


Fig. 1. Conceptual diagram of DW

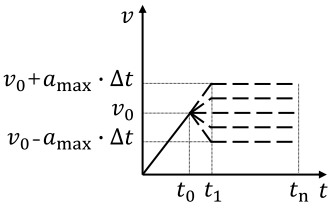


Fig. 2. Relationship of DW and total velocity

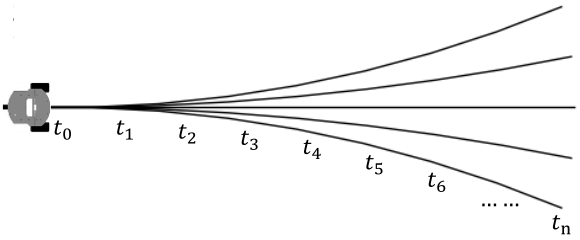


Fig. 3. Conceptual diagram of predict path

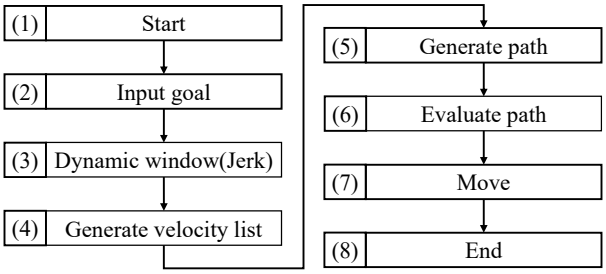


Fig. 4. Algorithm flow chart

The robot must be able to move efficiently during autonomous driving; however, it must not be subject to excessive acceleration. There is a trade-off between the evaluation of speed and acceleration. Because the evaluation function of the original method does not include an acceleration evaluation, they tend to select the highest speed [17-20].

3. PROPOSED METHOD

This study proposes a dynamic window approach using the jerk model (DWA-J), a method for generating speed change paths based on jerk models. Fig. 4 explains the algorithm used. Compared with the original method, Step (3) samples the jerk and angular velocity pairs. In

Step (4), the velocity series is generated from the sampled jerk by jerk model. In Step (6), the jerk scores are added to the evaluation function.

3.1 Velocity control method

The velocity series $v_{(1,2,3\ldots n)}$ within the forecast period was used to generate the path. The original method used the acceleration model shown in Eq. (1). In the acceleration model, the hypothesis of velocity invariance was eliminated, resulting in Eq. (9).

$$v_{t+1} = v_t + a \cdot \Delta t \qquad a \in [-a_{max}, a_{max}] \quad (9)$$

The velocity at the destination of the path generated using Eq. (1) is $v_0 + a \cdot \Delta t$; the velocity in Eq. (9) is $v_0 + \sum_0^t a \cdot \Delta t$. It was possible to generate a velocity series with a wide velocity range for the arrival point in the generated path. Figs. 5 and 6 show the relationship between acceleration and velocity. Fig. 5 shows the relationship between the velocity and acceleration using the original method. Fig. 6 shows the relationship between the velocity and acceleration using a method that eliminates the hypothesis of velocity invariance. The robot began its motion at t_0. In the proposed method the robot only accelerates between t_0 and t_1. The velocity becomes invariant from t_1. A method that eliminates the hypothesis of velocity invariance, the robot, continues to accelerate between t_0~t_n.

However, in the acceleration model, there is no restriction on the choice of acceleration; therefore, a large jerk may occur. Limiting the change in acceleration per unit time results in an acceleration model as given by Eq. (10).

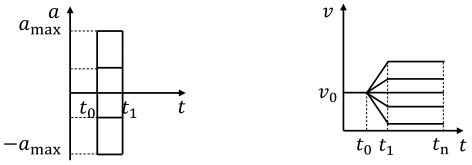
$$\begin{aligned} v_{t+1} = v_t + a_t \cdot \Delta t \quad & a_t \in [-a_{max}, a_{max}] \\ + j_{t+1} \cdot \Delta t^2 \quad & j_t \in [-j_{max}, j_{max}] \end{aligned} \quad (10)$$

The current velocity and acceleration of the robot are denoted as v_t and a_t; the acceleration provided by this is shown as j_{t+1}. The velocity at the end of the path, generated using Eq. (10) is $v_0 + \sum_0^{t-1} a_t \cdot \Delta t + \sum_0^t j_t \cdot \Delta t^2$. The relationship between jerk, acceleration, and velocity is shown in Fig. 7.

The jerk model generates a velocity series while controlling the maximum jerk. However, if n is the number of prediction steps, the total number of velocity series is the nth power of the number of j_n selections, and the increase in n makes it difficult to compute in real time. Therefore, in this study, we assumed that jerk acceleration was invariant during the prediction interval, as shown in Fig. 8(a). Even with this assumption, the acceleration continues to vary, as shown in Fig. 9(b), allowing for nonlinear velocity control, as shown in Fig. 10(c).

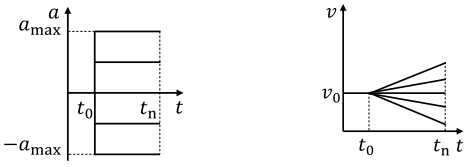
3.2 Evaluate function

The jerk value is added to the evaluation function shown in Eq. (5). The evaluation function for jerk is given by Eq. (11). Different functions are used to evaluate the velocity, depending on whether the vehicle should decelerate. Deceleration was determined based on the distance from the target. The distance from the target is indicated by d_{to_goal} and the deceleration distance by $d_{deaccel}$. The evaluation equation for the speed is given by Eq. (12). The overall evaluation function is given by Eq. (13).



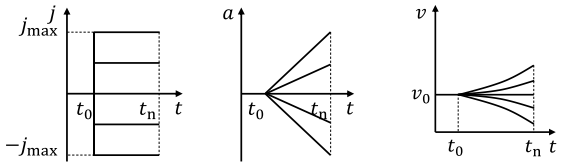
(a) Acceleration (b) Velocity

Fig. 5. Conceptual diagram of acceleration model with constant velocity hypothesis



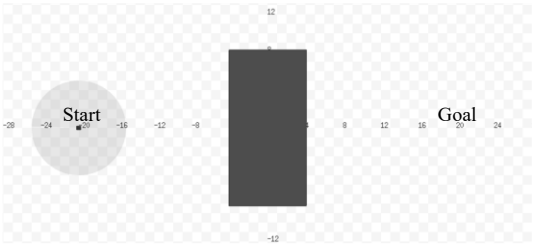
(a) Acceleration (b) Velocity

Fig. 6. Conceptual diagram of acceleration model

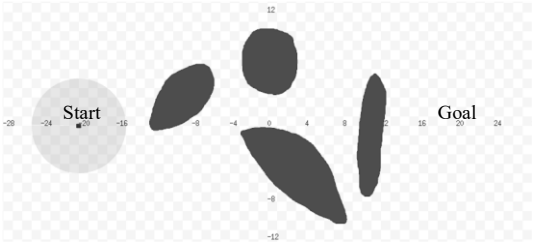


(a) Jerk (b) Acceleration (b) Velocity

Fig. 7. Conceptual diagram of jerk model



(a) Simple environment



(b) Complex environment

Fig. 8. Experiment environment

$$\text{jerk}(j) = 1 - \left|\frac{j}{j_{\max}}\right| \tag{11}$$

$$\text{vel}(v,\omega) = \begin{cases} \dfrac{v}{v_{\max}} & d_{\text{togoal}} > d_{\text{deaccel}} \\ 1 - \dfrac{v}{v_{\max}} & d_{\text{togoal}} \le d_{\text{deaccel}} \end{cases} \tag{12}$$

$$\begin{aligned} G(v,\omega) = \sigma(&\alpha \cdot \text{heading}(v,\omega) + \beta \\ &\cdot \text{dist}(v,\omega) + \gamma \cdot \text{vel}(v,\omega) + \delta \cdot \text{jerk}(j)) \end{aligned} \tag{13}$$

4. EXPERIMENT

We verified the effectiveness of the proposed method through simulations.

4.1 Experiment environment

The experimental environment and control program were built based on the Robot Operating System (ROS) [21]. In this experiment, we used a two-dimensional (2D) simulator called Stage [22], because the robot moved only in a plane.

We conducted an autonomous driving experiment in a simple environment to verify the effectiveness of the proposed method in controlling jerk, and compare the effect of range of the arrival point of the paths on autonomous driving in a complex environment. The experimental environment is displayed in Fig. 8.

4.2 Experiment1: Verification of the effectiveness of the proposed method in controlling jerk

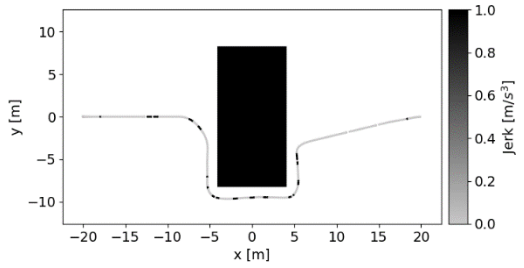
Fig. 8(a) shows the environment used to verify the effectiveness of the proposed method in controlling jerk. The maximum velocity of the robot was set to 2.0[m/s], maximum acceleration to 1.0[m/s^2], maximum jerk in the proposed method to 0.5[m/s^3], robot's initial position to (-20, 0), and the target to (20, 0). The unit of the coordinates is[m]. It is possible in DWA that the robot may not reach the goal because of a local minimum in the process of optimizing the evaluation function. Therefore, we used the Tangentbug [23] to generate a sub-goal, which is a local path algorithm that can provide an optimal sub-goal within the sensor range. It was considered that the robot has reached the target if it enters within a radius of 0.2 [m] around the target. The deceleration distances for the original and proposed methods were set to 1.0[m] and 2.0[m], respectively. The weights a, β, γ, δ of the evaluation function are set to 1.0, 2.0, 0.1, and 0.001, respectively.

The travel times and maximum jerks for the original and proposed methods are listed in Table 1. The path and variation of the acceleration on the path are shown in Figs. 9(a) and 10(a), respectively. The changes in the velocity, acceleration, and acceleration over time are displayed in Figs. 9(b) and 10(b). The time interval Δt is set to 0.1[s], such that the frame rate is 10[fps]. Hereafter, the duration of the experimental results is specified in terms of frames.

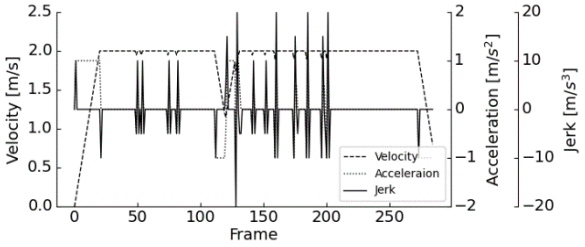
Figs. 9(a) and 10(a) show that, for both the original and proposed methods, the speed was changed to avoid obstacles. In the original method, the robot decelerated to avoid obstacles at 120 frames, as shown in Fig. 9(b).

Table 1. Comparison of original method and proposed method

	Original method	Proposed method
Total frames	279	289
Maximum jerk[m/s^3]	20.0	0.5
Frames of cruise period	247	230
Average velocity of cruise period[m/s]	1.99	1.98

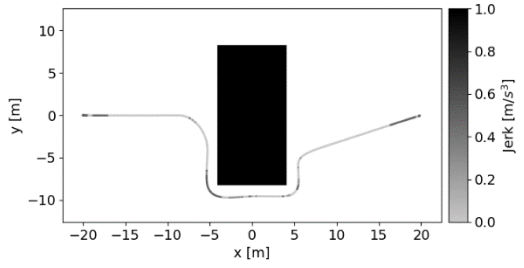


(a) Migration path

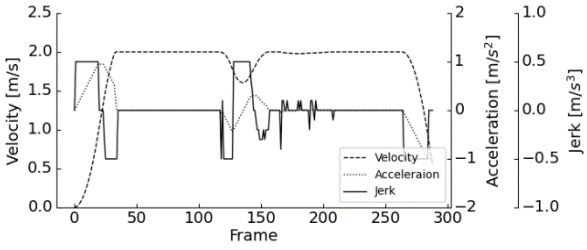


(b) Velocity–acceleration jerk graph

Fig. 9. Result of original method



(a) Migration path



(b) Velocity–acceleration jerk graph

Fig. 10. Result of proposed method

After avoidance, the acceleration changed suddenly from -1.0 to 1.0[m/s^2]. During the avoidance process, a large jerk occurs and the velocity changes significantly. Because the proposed method controls jerk, the robot can avoid obstacles even by slight deceleration.

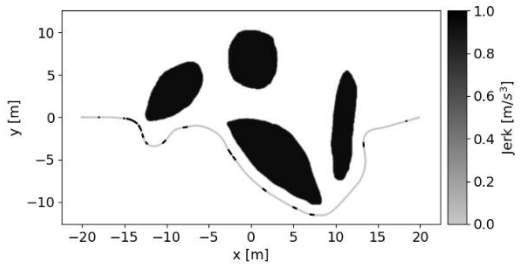
Robot speed changes are divided into three major periods: an initial acceleration period, mid-cruise period, and final deceleration period. The differences in the efficiency of the speed control methods were primarily reflected in the acceleration and deceleration periods. As the robot's travel time increased, the difference in the efficiency of each method decreased. Because the velocity curve of the proposed method with the jerk model is s-shaped, the acceleration time is longer than that of the original method using the trapezoidal velocity control with the acceleration model. To remove the effect of the acceleration/deceleration period, the average speed and travel time during the cruise period were used to compare the efficiency of the proposed and original methods. The cruise periods for the original and proposed methods ranged from 20 to 267 frames and from 33 to 263 frames, respectively. As listed in Table 1, the maximum jerk of the proposed method is smaller than that of the original method, with a similar average speed and travel time.

4.3 Experiment2: Compare the effect of range of the arrival point of the paths on autonomous driving

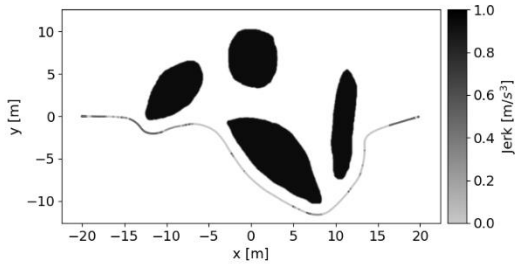
Fig. 8(b) shows the environment used to compare the effect of range of the arrival point of the paths on autonomous driving by adjusting sensor range. The result when sensor range is 5[m] and 2[m] are shown in Figs. 11 and 12, respectively.

Both the original and proposed methods can clear autonomous driving task when sensor range is 5[m].

However, under the condition of 2[m] sensor range, the robot based on the original method stopped halfway through, because the original method cannot generate an avoidance path. On the other hand, the proposed method can generate avoidance path through acceleration and deceleration, the robot can stably accomplish autonomous driving task with narrow sensor range.

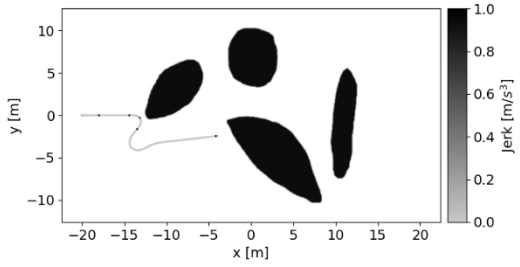


(a) Original method

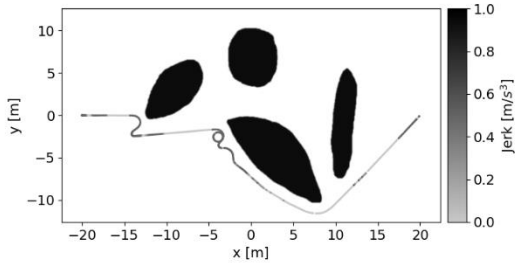


(b) Proposed method

Fig. 11. Result when sensor range is 5[m]



(a) Original method



(b) Proposed method

Fig. 12. Result when sensor range is 2[m]

5. CONCLUSION

The objective of this study was to smooth the motion of an autonomous robot. To solve the problems of the original DWA, such as narrow predicted paths, inability to generate avoidance paths, and inability to control jerk, we proposed a method for generating variable-speed paths based on a jerk model.

The proposed method uses a jerk model to limit maximum jerk. The variable-speed motion based on the velocity series extends the range of the predicted paths and reduces the possibility of failure in generating avoidance paths. Simulator experiments validate the effectiveness of the proposed method.

In DWA for the generation of predictive paths, the velocity and angular velocity are assumed invariant. In this study, instead of assuming that the velocity is invariant, we assume that jerk is invariant. This allows the generation of a wider range of paths. However, the angular velocity is still assumed to be invariant. The application of jerk model to angular velocity is expected to generate a wider range. Because there is a trade-off between the evaluation of the velocity and jerk, it is necessary to optimize the weights of the evaluation function.

REFERENCES

[1] P. Pratama, T. Nguyen, H. Kim, D. Kim and S. Kim: "Positioning and Obstacle Avoidance of Automatic Guided Vehicle in Partially Known Environment," *International Journal of Control, Automation and Systems*, vol. 14, pp. 1572-1581, 2016.

[2] C. Yang, Y. Jiang, J. Na, Z. Li, L. Cheng and C. Su: "One result for "Finite-Time Convergence Adaptive Fuzzy Control for Dual-Arm Robot with Unknown Kinematics and Dynamics", *IEEE Transactions on Fuzzy Systems*, vol. 27, pp. 574-588, 2019.

[3] V. Thanh, D. Vinh, N. Nghi, L. Nam and D. Toan: "Restaurant Serving Robot with Double Line Sensors Following Approach," *IEEE International Conference on Mechatronics and Automation*, pp. 235-239, 2019.

[4] U. Reiser, C. Connette, J. Fischer, J. Kubacki, A. Bubeck, F. Weisshardt, T. Jacobs, C. Parlitz, M. Hägele and A. Verl: "Care-o-bot® 3-creating a product vision for service robot applications by integrating design and technology," *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1992-1998, 2009.

[5] S. Tzafestas: "Mobile robot control and navigation: A global overview," *Journal of Intelligent & Robotic Systems*, vol. 17, pp. 35-58, 2018.

[6] Q. Wu, Y. Liu and C. Wu: "An overview of current situations of robot industry development," *ITM Web of Conferences*, vol. 17, 2018.

[7] P. Hart, N. Nilsson and B. Raphael: "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," *IEEE Trans. Syst. Sci. Cybernetics*, vol. 4, pp. 100-107, 1968.

[8] A. Stentz: "Optimal and Efficient Path Planning for Partially-Known Environments," *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, vol. 4, pp. 3310-3314, 1994.

[9] S. Karaman and E. Frazzoli: "Incremental Sampling-based Algorithms for Optimal Motion Planning," *Robotics: Science and Systems VI*, pp. 267-276, 2010.

[10] J. Borenstein and Y. Koren: "The vector field histogram-fast obstacle avoidance for mobile robots," *IEEE transactions on robotics and automation*, vol. 7, pp. 278-288, 1991.

[11] D. Fox, W. Burgard and S. Thrun: "The dynamic window approach to collision avoidance," *IEEE Robotics & Automation Magazine*, vol. 4, no. 1, pp. 23-33, 1997.

[12] M. Kobayashi and N. Motoi: "Local Path Planning Method Based on Virtual Manipulators and Dynamic Window Approach for a Wheeled Mobile Robot," *2021 IEEE/SICE International Symposium on System Integration*, pp. 499-504, 2021.

[13] R. Haschke, E. Weitnauer and H. Ritter: "On-line planning of time-optimal, jerk-limited trajectories," *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3248-3253, 2008.

[14] A. Deshmukh, B. Mulani, N. Jadhav and A. Parihar: "Study of frequency characteristics of vehicle motions for the derivation of inherent jerk," *SAE International Journal of Passenger Cars-Mechanical Systems*, vol. 9, pp. 419-424, 2016.

[15] H. Hayati, D. Eager, A. Pendrill and H. Alberg: "Jerk within the Context of Science and Engineering—A Systematic Review," *Vibration*, vol. 3, no. 4, pp. 371-409, 2020.

[16] C. Bertolin, A. Caratelli, M. Grimaldi and M. Massi: "Analysis of Jerk as a novel tree-fails hazard index: A case study applied to tree monitoring in the archaeological park of the Colosseum in Rome (Italy)," *International Journal of Disaster Risk Reduction*, vol. 56, 2021.

[17] M. Seder and I. Petrović: "Dynamic window based approach to mobile robot motion control in the presence of moving obstacles," *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pp. 1986-1991, 2007.

[18] K. Goto, K. Kon and F. Matsuno: "Motion Planning of an Autonomous Mobile Robot Considering Regions with Velocity Constraint," *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3269-3274, 2010.

[19] D. A. Lima and G. Pereira: "Navigation of an Autonomous Car Using Vector Fields and the Dynamic Window Approach," *Journal of Control, Automation and Electrical Systems*, vol.24, pp. 106-116, 2013.

[20] D. Teso-Fz-Betoño, E. Zulueta, U. Fernández-Gámiz, A. Saenz-Aguirre and R. Martínez: "Predictive Dynamic Window Approach Development with Artificial Neural Fuzzy Inference Improvement," *Electronics*, vol. 8, 2019.

[21] M.Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, A. Ng: "ROS: an open-source Robot Operating System," *ICRA Workshop on Open Source Software*, vol. 3, no. 3.2, pp. 5, 2009.

[22] R. Vaughan: "Massively multi-robot simulation in stage," *Swarm Intelligence*, vol. 2, pp. 189–208, 2008.

[23] I. Kamon, E. Rimon and E. Rivlin: "Tangentbug: A range-sensor-based navigation algorithm," *The International Journal of Robotics Research*, vol. 17, no. 9, pp. 934-953, 1998.