Multi-Constraint Enhanced DWA for Robust and Smooth Local Navigation

Yi Tang
Department of Information
Technologies in Automated Systems BSUIR
Minsk, Belarus
tangyijcb@163.com

Dmitry Pertsau
Department of Electronic
Computer Machines BSUIR
Minsk, Belarus
pertsev@bsuir.by

***Abstract.* To address the issues of path discontinuity, control instability, and insufficient global guidance in the traditional Dynamic Window Approach (DWA) for local path planning, this paper proposes a Multi-Constraint Dynamic Window Approach (MC-DWA). The proposed method incorporates multiple constraints in trajectory evaluation, including heading deviation, path adherence, curvature variation, and velocity jerk, thereby improving the smoothness and stability of the planned paths. Extensive simulation experiments demonstrate that MC-DWA achieves better path quality and safety than existing methods, exhibiting enhanced adaptability and robustness.**

***Keywords:* dynamic window approach, global-local synergy, multi-constraint optimization, local path planning**

I. Introduction

With the advancement of intelligent and automated technologies, mobile robots are increasingly deployed in areas such as industry, healthcare, and logistics. As a key component of autonomous navigation, path planning aims to generate an optimal, collision-free, and efficient path from the start to the target [1]. It not only ensures motion safety but also addresses the challenge of navigating in complex and dynamic environments.

Path planning algorithms are typically divided into global and local planning. Global planning assumes a fully known environment and uses methods like A* and Dijkstra to compute an optimal path, making it suitable for static scenarios. Local planning, by contrast, depends on real-time perception to adapt the robot's path in response to environmental changes and obstacles. Given that real-world environments are often dynamic and partially observable, local planning plays a critical role in ensuring navigation safety and task reliability [2].

DWA is a classical local path planning algorithm proposed by Fox et al. in 1997 [3]. This method generates feasible short-term trajectories by simulating the robot's motion in the velocity space in real time. However, the traditional DWA algorithm has limited consideration of the robot's kinematic and dynamic characteristics during trajectory generation. It does not adequately account for factors such as trajectory smoothness, stability, and curvature variation. As a result, the robot may experience abrupt turns or sudden velocity changes during motion, which can negatively impact the stability of the control system and reduce the reliability of trajectory execution.

To address these limitations, this paper proposes a MC-DWA. The method introduces several trajectory evaluation items, including dual-angle heading deviation, global path adherence, curvature variation control, and jerk minimization. These constraints are integrated into a unified multi-objective scoring model, enabling more globally consistent, locally smooth, and dynamically stable trajectory generation. Extensive simulations across various complex scenarios show that MC-DWA significantly outperforms classical DWA and its variants in terms of path quality, safety, and robustness, making it a promising solution for real-time autonomous navigation in dynamic and cluttered environments.

II. Method

A. Basic Principles of the Classic DWA Algorithm

The fundamental idea of DWA is to sample admissible pairs of linear and angular velocities within a constrained dynamic space, simulate the resulting short-term trajectories, and select the optimal control command according to a predefined evaluation function. This enables the robot to navigate toward the target while avoiding obstacles in dynamic environments [3]. The set of admissible velocity commands in DWA is governed by the intersection of three velocity constraint sets: kinematic, dynamic, and safety constraints.

Kinematic constraints V_{kin} define the basic physical limits of the robot's actuators. Specifically, the linear velocity v and angular velocity ω must lie within the hardware-defined bounds as in Equation (1):

$$\{V_{\text{kin}} = (v,\omega) \mid v \in [v_{min}, v_{max}], \omega \in [\omega_{min}, \omega_{max}]\} \quad (1)$$

Dynamic constraints V_{dyn} account for the robot's limited acceleration and deceleration capabilities, ensuring that sampled velocities are physically reachable within a single control cycle of duration Δt. Given the robot's current linear and angular velocities (v_0, ω_0), and the maximum allowable linear and angular accelerations $(a_v^{\max}, a_\omega^{\max})$, the velocity change over one time step is bounded as follows in Equation (2):

$$\begin{cases} v \in [v_0 - a_v^{\max}\Delta t, v_0 + a_v^{\max}\Delta t] \\ \omega \in [\omega_0 - a_\omega^{\max}\Delta t, \omega_0 + a_\omega^{\max}\Delta t] \end{cases}. \quad (2)$$

Safety constraints V_{safe} ensure that candidate velocities do not result in collisions. For each candidate velocity, the robot simulates the corresponding trajectory and checks whether it can safely stop before hitting any obstacle. Assuming a maximum deceleration a_{brake} (here, $a_{brake} = a_v^{\max}$), the required braking distance d_{brake} to come to a full stop from velocity v is given:

$$d_{brake} = \frac{v^2}{2a_{brake}}. \quad (3)$$

A candidate velocity is considered safe only if the minimum distance to obstacles along the predicted trajectory, denoted d_{obs}, is bigger than d_{brake}. This constraint effectively filters out velocity pairs that would result in collision under emergency braking conditions. The final dynamic window is defined as the intersection of the three constraint sets in Equation (4):

$$V_{dw} = V_{kin} \cap V_{dyn} \cap V_{safe}. \tag{4}$$

Each velocity pair $(v, \omega) \in V_{dw}$ is forward-simulated over a short time horizon T, generating a corresponding trajectory. The quality of each trajectory is then assessed using a weighted evaluation function: $G = \alpha \cdot \text{heading} + +\beta \cdot \text{clearance} + \gamma \cdot \text{velocity}$. In this formulation, the 'heading' term measures the angular alignment between the trajectory endpoint and the target direction, promoting goal-oriented motion. The 'clearance' term represents the shortest distance between the predicted trajectory and nearby obstacles, encouraging safer paths. The 'velocity' term favors trajectories with higher forward speeds, which contributes to time efficiency. The weights α, β, γ are tuning parameters used to balance these three objectives according to the task requirements. The detailed formula can be seen in [3].

B. Multi-Constraint Enhanced DWA

Although the DWA offers real-time performance and dynamic obstacle avoidance capabilities, it still exhibits notable limitations in complex environments. Specifically, the traditional DWA employs a relatively simplified evaluation function that typically considers only basic factors such as goal direction, velocity magnitude, and obstacle distance. It lacks effective integration of global path guidance, which can lead to trajectory deviations from the global plan and susceptibility to local minima [4]. To address these issues, the proposed MC-DWA incorporates three key categories of constraint metrics and constructs a comprehensive trajectory scoring model to enhance the rationality, smoothness, and robustness of path planning. The specific improvements of MC-DWA are described as follows.

To enhance directional guidance during navigation, MC-DWA introduces a dual-heading consistency evaluation mechanism, which includes goal-oriented consistency and global path alignment. Specifically, let the predicted trajectory endpoint orientation be denoted as θ_{pred}, the direction toward the goal as θ_{goal}, and the direction of the final segment of the global path as θ_{final}. The angular deviations between these directions are calculated using Equation (5):

$$\begin{cases} \Delta\theta_{goal} = |\theta_{pred} - \theta_{goal}| \\ \Delta\theta_{final} = |\theta_{pred} - \theta_{final}| \end{cases}. \tag{5}$$

Based on this, the heading consistency score is computed as shown in Equation (6):

$$\mathrm{H(v, w)} = \lambda_1 \cdot \frac{\pi - \Delta\theta_{goal}}{\pi} + \lambda_2 \cdot \frac{\pi - \Delta\theta_{final}}{\pi}, \tag{6}$$

where λ_1 and λ_2 are weighting factors that balance the contributions of goal direction and path direction alignment, respectively (here we set $\lambda_1 = \lambda_2 = 0.5$).

However, when the robot is far from the target, heading-based scoring alone may be insufficient to provide strong attraction toward the goal [5]. To address this, a goal distance factor is incorporated into the evaluation function. An exponential attraction function is used to introduce a nonlinear response to distance variations, as defined in Equation (7):

$$G_{att}(v, w) = -exp(\alpha \cdot d_g(v, \omega)), \tag{7}$$

where $d_g(v, w)$ represents the Euclidean distance between the predicted trajectory endpoint and the goal, and $\alpha = 0.5$ is a tunable parameter that controls the strength of the attractive force. Therefore, the heading guidance and target alignment evaluation function is denoted as $\mathrm{O}(v, \omega)$, and its computation is given:

$$\mathrm{O}(v, \omega) = \mathrm{H}(v, \omega) + G_{att}(v, \omega). \tag{8}$$

Then, to address the issue in classical DWA where local path planning may deviate significantly from the global reference path, which causes the robot to fall into local minima. We introduce a global path adherence constraint. Specifically, let the global path generated by a global planner (e.g., A*) be denoted as $P_{global} = \{P_1, P_2, \ldots, P_L\}$ where each $P_i = (x_i, y_i)$ represents a waypoint in the global path. At each control cycle, the algorithm first identifies the index i^* of the global path point closest to the robot's current position $X_c = (x_c, y_c)$. Then, a local global path segment P_{local} consist of m consecutive nodes starting from P_{global} is selected. Then, given a predicted trajectory consisting of T points, denoted as $T_{pred} = \{t_1, t_2, \ldots, t_N\}$, the path adherence metric is defined as the maximum deviation between the predicted trajectory and the selected local global path segment. The computation process is described:

$$\mathrm{i}^* = \mathrm{argmin}_{1 \le i \le L} \| \mathrm{P_i} - \mathrm{X_c} \|_2, \tag{9}$$

$$\mathrm{P_{local}} = \{\mathrm{P_{i^*}}, \mathrm{P_{i^*+1}}, \ldots, \mathrm{P_{i^*+M-1}}\}, \tag{10}$$

$$\mathrm{D}(v, \omega) = -\mathrm{max}_{1 \le i \le N} \mathrm{min}_{p \in \mathrm{P_{local}}} \| \mathrm{t_i} - \mathrm{p} \|_2. \tag{11}$$

In classical DWA, the lack of temporal continuity in decision-making often causes abrupt changes in velocity and steering, leading to discontinuous trajectories and unstable robot motion [4]. To address this issue, MC-DWA introduces a penalty term based on the variation in motion state from the previous time step during candidate velocity selection. In addition, MC-DWA enhances the trajectory evaluation function by incorporating constraints on trajectory curvature variation and jerk (i.e., the rate of change of acceleration), optimizing trajectory smoothness and dynamic stability from both spatial and dynamic perspectives. Specifically, the curvature variation is defined as the second-order difference between consecutive trajectory points, and is computed using:

$$C_{var}(v, \omega) = -\frac{1}{N-2} \sum_{i=1}^{N-2} \left| \frac{y_{i+2} - 2y_{i+1} + y_i}{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 + \varepsilon} \right|, \tag{12}$$

where ε is a small constant to prevent division by zero. This expression captures the local bending intensity of the trajectory. Furthermore, jerk metrics are introduced for both linear and angular velocities to suppress control shocks during execution can be calculated:

$$\begin{cases} J_v(v,\omega) = -\frac{1}{N-2}\sum_{i=1}^{N-2} \left| \frac{a_{v,i+1}-a_{v,i}}{\Delta t} \right| \\ J_\omega(v,\omega) = -\frac{1}{N-2}\sum_{i=1}^{N-2} \left| \frac{a_{\omega,i+1}-a_{\omega,i}}{\Delta t} \right| \end{cases}. \tag{13}$$

This design limits abrupt changes in velocity commands, improving trajectory controllability and execution smoothness, particularly for mobile platforms requiring high trajectory continuity. The overall smoothness and stability evaluation function denoted by $S(v,\omega)$, and is computed using (14):

$$S(v,\omega) = C_{var}(v,\omega) + J_v(v,\omega) + J_\omega(v,\omega) \tag{14}$$

On the other hand, dynamic stability control is especially critical during high-speed turns. For mobile platforms with significant velocity or center of mass height (e.g., differential-drive robots or automated forklifts), sharp turning at high speeds may cause skidding or even rollover. To mitigate this, MC-DWA incorporates constraints on curvature radius and lateral acceleration during the velocity space filtering stage. Specifically, the maximum admissible angular velocity under a given linear velocity is restricted based on the lateral acceleration threshold a_{lat}, as formulated:

$$V_{lat} = \{(v,\omega) | v \cdot \omega < a_{lat}|\}. \tag{15}$$

This constraint ensures that the resulting centrifugal force during turning does not exceed the platform's stability limit, thereby enhancing dynamic controllability and operational safety. Accordingly, in MC-DWA, the admissible velocity space of the dynamic window is computed as Equation (16):

$$V_{dw} = V_{kin} \cap V_{dyn} \cap V_{safe} \cap V_{lat}. \tag{16}$$

Finally, to effectively integrate the aforementioned constraints, MC-DWA constructs the following comprehensive trajectory evaluation function in (17):

$$\mathrm{Score}(v,\omega) = w_O \cdot O(v,\omega) + w_D \cdot D(v,\omega) + w_S \cdot S(v,\omega) \tag{17}$$

Here, w_O, w_D and w_S are weighting factors, which can be changed to adapted to specific applications. All the values are set to 1 in this paper.

III. Experiment

A. Experiment Setups

The MC_DWA algorithm framework was implemented on the MATLAB R2024a platform. To systematically evaluate its path planning performance and control stability, a total of 50 two-dimensional grid maps were designed, covering typical navigation challenges such as dense obstacle regions and long corridor intersections. Each map includes a defined start and goal point, with a guaranteed feasible global reference path.

All evaluation metrics were statistically derived from these 50 representative test scenarios, which feature varying map structures, obstacle densities, and dynamic obstacle patterns. This diverse task setup ensures strong generalization and realistic challenge levels, thereby providing a more objective and comprehensive validation of the proposed method's robustness and adaptability in complex dynamic environments. The robot's control and physical parameters used during the simulation are summarized in Table I.

TABLE I. Simulation Parameters

Name	*Value*	Name	*Value*
Time step (Δt)	0.1s	Robot radius	0.4
Horizon (T)	2.0s	Safe margin	0.2
Max step	500	Maximum lateral acceleration (a_{lat})	1.0 m/s^2
Maximum linear velocity (v_{max})	2.0m/s	Maximum angular velocity (ω_{max})	1.57 rad/s
Minimum linear velocity (v_{min})	0.0 m/s	Minimum angular velocity (ω_{min})	-1.57 rad/s
Maximum linear acceleration (a_v^{max})	0.4 m/s^2	Maximum angular acceleration (a_ω^{max})	0.78 rad/s^2
Minimum linear acceleration (a_v^{min})	-0.4 m/s^2	Minimum angular acceleration (a_ω^{min})	-0.78 rad/s^2

B. Comparison with other improved DWA algorithms

To comprehensively evaluate the practical performance of the proposed MC-DWA, we conducted a systematic comparison with the classical DWA algorithm as well as three representative improved variants [6–8].

To ensure fairness, all algorithms were evaluated under the same initial conditions and map settings, using a unified framework to assess performance across multiple dimensions. The evaluation metrics include Minimum Safe Distance (MSD), Time Step (TS), Path Length (PL), and Path Curvature (PC), which collectively reflect path safety, efficiency, smoothness, and control stability. The quantitative results are summarized in Table II.

TABLE II. Comparison with other DWA methods

Method	*MSD*	*PL*	*PC*	*TS*
Classic-DWA[3]	0.20	35.67	0.706	357
E-DWA[6]	0.19	30.41	0.513	304
Fuzzy-DWA[7]	0.16	35.15	0.792	352
Pred-DWA[8]	0.22	33.04	0.763	330
MC-DWA(ours)	0.53	37.97	0.323	380

Table presents a comparative analysis of the proposed MC-DWA against the classical DWA and three improved variants, namely Energy-Efficient DWA (E-DWA), Fuzzy Adaptive DWA (Fuzzy-DWA), and Predictive DWA (Pred-DWA). Overall, MC-DWA demonstrates superior performance in terms of path safety, smoothness, and control stability, validating the effectiveness of its multi-constraint fusion strategy. For MSD, MC-DWA achieves a minimum obstacle clearance of 0.53, significantly higher than all other methods, indicating enhanced safety and better avoidance behavior. This is attributed to the global path adherence constraint, which helps maintain safe distances in dense environments. In PC, MC-DWA achieves the lowest average curvature (0.323), while other methods exceed 0.7, showing that MC-DWA produces much smoother trajectories. This improvement results from the use of curvature and jerk constraints, which reduce oscillations and improve control consistency. Regarding PL, MC-DWA generates a slightly longer trajectory (37.97) than E-DWA and Pred-DWA but remains comparable to Classic DWA and Fuzzy-DWA. The marginal increase in path length is a reasonable trade-off for improved smoothness and directional consistency enabled by the dual-heading guidance. In TS, MC-DWA requires 380 steps to reach the goal, slightly higher than others, with E-DWA being the most time-efficient. However, the improved stability and safety justify this minor cost, especially in dynamic or risk-prone environments.

C. Visualization comparison

To provide an intuitive comparison of path planning performance across different environments, experimental results were visualized.

As shown in Fig 1, the paths generated by the proposed MC-DWA algorithm, the classical DWA, and A* algorithm are illustrated under normal and densely obstructed environments, respectively. It can be observed from plot that in the normal environment, although the classical DWA is able to reach the target successfully, its generated path tends to closely follow the boundaries of static obstacles, posing a higher risk of collision. In more complex environments with dense obstacles, the classical DWA fails to complete navigation due to the lack of effective global guidance and dynamic obstacle avoidance strategies. In contrast, the proposed MC-DWA successfully completes the navigation task in both environments. Generated trajectories maintain good adherence to the global path and consistently keep a safe distance from surrounding obstacles throughout the motion process.

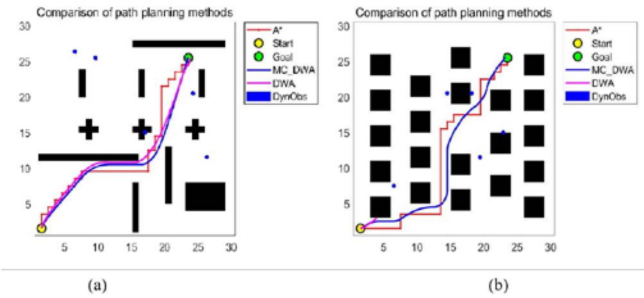

Fig. 1. Visualization of comparison with classic DWA and global A* methods in different maps

Furthermore, Fig. 2 provides a visual comparison in two typical complex environments.

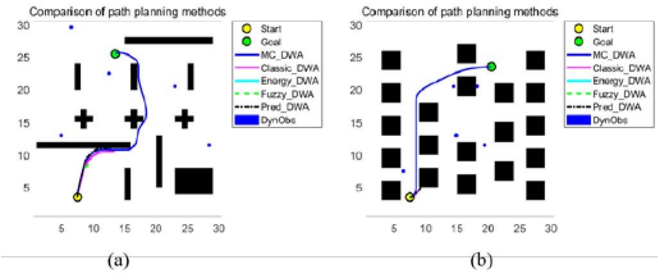

Fig. 2. Visualization of comparison with different DWA methods in different challenging maps

As shown in Fig 2a, the scenario includes a distant obstacle that poses a critical hindrance to the planned path. The results indicate that, except for MC-DWA, all other algorithms fail to navigate around the obstacle and eventually become stuck. This issue primarily stems from the lack of effective target-oriented guidance, causing traditional algorithms to lose directional cues in the vicinity of the obstacle. In contrast, MC-DWA incorporates an exponential attraction mechanism into its trajectory evaluation, which enhances the goal-directed driving force while maintaining safety, enabling the robot to traverse complex terrain and proceed smoothly toward the target. Fig 2b presents a highly complex environment composed of densely distributed obstacles. In this scenario, the classical DWA and other improved methods are generally limited by local information and tend to fall into local minima, halting progress. MC-DWA, however, introduces a global path adherence constraint, enabling local planning to better align with global navigation intent. This mechanism allows the robot to escape local traps and successfully reach the goal. Throughout the process, the generated path maintains safe distances from obstacles while exhibiting good continuity and dynamic stability.

IV. CONCLUSION

This paper presents an improved MC-DWA to address key limitations of the traditional DWA in local path planning. MC-DWA introduces several constraints into a unified scoring framework, significantly enhancing trajectory quality, smoothness, and environmental robustness while maintaining real-time performance. Experiments on various simulated maps demonstrate that MC-DWA outperforms the classical DWA and representative improved methods in several evaluation metrics.

Nonetheless, MC-DWA remains a manually designed heuristic method, with fixed scoring functions and weight parameters, limiting its scalability in highly dynamic or high-dimensional tasks. Future work will explore integrating MC-DWA with Deep Reinforcement Learning (DRL), using its constraint structure as policy guidance or initialization to combine rule-based reliability with data-driven adaptability [9]. We also plan to deploy MC-DWA on platforms such as ROS2 and Gazebo to further assess its generalization and real-world applicability.

REFERENCES

[1] L. Yang, P. Li, S. Qian, H. Quan, J. Miao, M. Liu, and E. Memetimin, "Path planning technique for mobile robots: A review," Machines, vol. 11, no. 10, p. 980, 2023.

[2] J. R. Sánchez-Ibáñez, C. J. Pérez-del-Pulgar, and A. García-Cerezo, "Path planning for autonomous mobile robots: A review," Sensors, vol. 21, no. 23, p. 7898, 2021.

[3] 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, Mar. 1997, DOI: 10.1109/100.580977.

[4] X. Gong, Y. Gao, F. Wang, D. Zhu, W. Zhao, F. Wang, and Y. Liu, "A local path planning algorithm for robots based on improved DWA," Electronics, vol. 13, no. 15, p. 2965, 2024.

[5] L. S. Liu, J. F. Lin, J. X. Yao, D. W. He, J. S. Zheng, J. Huang, and P. Shi, "Path planning for smart car based on Dijkstra algorithm and dynamic window approach," Wireless Communications and Mobile Computing, vol. 2021, no. 1, p. 8881684, 2021.

[6] C. Henkel, A. Bubeck, and W. Xu, "Energy efficient dynamic window approach for local path planning in mobile service robotics," IFAC-PapersOnLine, vol. 49, no. 15, pp. 32–37, 2016.

[7] A. Liu, C. Liu, L. Li, R. Wang, and Z. Lu, "An improved fuzzy-controlled local path planning algorithm based on dynamic window approach," J. of Field Robotics, vol. 42, no. 2, pp. 430–454, 2025.

[8] M. Missura and M. Bennewitz, "Predictive collision avoidance for the dynamic window approach," in Proc. IEEE Int. Conf. on Robotics and Automation, Montreal, QC, Canada, May 2019, pp. 8620–8626.

[9] C. Wu, W. Yu, G. Li, and W. Liao, "Deep reinforcement learning with dynamic window approach based collision avoidance path planning for maritime autonomous surface ships," Ocean Engineering, vol. 284, p. 115208, 2023.