

## COMPUTATIONALLY EFFICIENT DYNAMIC WINDOW APPROACH BASED ON PATTERN SEARCH OPTIMIZATION

Rafal SZCZEPANSKI\*<sup>id</sup>

\*Institute of Engineering and Technology, Faculty of Physics Astronomy and Informatics,  
Nicolaus Copernicus University in Toruń, ul. Wileńska 7, 87-100 Toruń, Poland

[szczepl@umk.pl](mailto:szczepl@umk.pl)

received 27 April 2025, revised 04 November 2025, accepted 05 November 2025

**Abstract:** Collision avoidance is a crucial aspect of autonomous ground vehicles (AGVs). One of the most common algorithms is called the dynamic window approach (DWA). The algorithm enables AGVs to operate with high performance in an unknown environment with a particular emphasis on achieving maximum linear and angular acceleration. However, DWA requires high computational effort to examine all possibilities with high resolution, and then select the best possible pair of control signals, i.e., linear and angular velocities. In this paper, the Pattern Search (PS) optimization algorithm is used to reduce the computational requirement of the DWA. Instead of calculating the DWA objective function for each possibility, the PS is used for algorithmically selecting the next examined pair of control signals. The results obtained demonstrate that a similar resolution of control signals can be achieved with almost two times less computational effort. The proposed approach has been examined in the MATLAB environment, while the source code is available on the MathWorks FileExchange.

**Key words:** autonomous ground vehicle, path planning algorithm, Dynamic Window Approach, optimization algorithm, Pattern Search

### 1. INTRODUCTION

Autonomous ground vehicles (AGVs) are rapidly advancing to provide efficient and safe operations in various industrial sectors, including logistics, manufacturing, and healthcare [1-2]. The navigation of an AGV can be divided into two parts: local and global path planning [3]. The global one is responsible for providing the shortest path to the goal based on the environment map [4]. The AGV then follows the global path using the sensors onboard to avoid unpredictable obstacles – this is known as a local path planning problem [5-6].

In this paper, the local path planning problem is considered. One of the most commonly used algorithms is called the Dynamic Window Approach (DWA) [7]. It is worth noting that DWA is one of the default local path planning algorithms implemented in the Robot Operating System. The algorithm is based on selection control signals (i.e., linear and angular velocity pairs) to minimize the objective function. The procedure is as follows: Generate possible pairs of velocities with particular emphasis on limited dynamics (i.e., linear and angular accelerations). Next, for each pair, predict the future trajectory and calculate the objective function. The objectives are as follows: to maximize the linear velocities, maximize the clearance, and minimize the heading angle of movement to the goal. The algorithm is relatively simple. However, to provide high performance, the number of pairs for evaluation must be relatively high. In such a case, the computational effort is significant.

In recent years, researchers have focused on enhancing the DWA algorithm to achieve the highest performance in AGV operations. However, there is a gap related to the computational performance of the DWA. In [8], the integration of reinforcement learning with an improved dynamic window approach is proposed to plan

the path of mobile robots in unknown environments. Using learning policies (Q-learning algorithm), the AGV can adjust the weight of the objective function of the DWA. This solution enables the authors to enhance the efficiency and effectiveness of robotic navigation in complex and unpredictable terrains. Unfortunately, such a solution increases the computational effort of the algorithm. An improved DWA for unmanned surface vehicles (USVs) is proposed by incorporating environmental factors such as wind, waves, and current into the local path planning process [9]. Additional factors were added to the objective function of the DWA. This modification enables USVs to more effectively adapt their navigation strategies in real-time, thereby significantly enhancing their ability to maneuver through complex and dynamic maritime environments. The computational effort was not considered in the article. In [10], the smooth and safe path is provided using a path planner based on DWA and A\* algorithms. The solution by the fusion algorithm can successfully avoid dynamic obstacles and is effective and feasible in path planning. A similar approach was proposed in [11]. However, the proposed mechanism appears more complex, and enhancements significantly improve global planning efficiency, ensuring smoother paths. In [12], the DWA was combined with the rapidly-exploring random tree (RRT\*) algorithm. The modification was similar to the ones mentioned above. The RRT\* was used to provide a global path, while the DWA algorithm was responsible for tracking the global path. Moreover, the object recognition using the YOLO framework was adopted. The presented results demonstrate that the DWA algorithm is a crucial path-planning algorithm for a broad range of autonomous vehicles. The objective of the work described in [13] is to design an improved DWA algorithm to enhance the obstacle avoidance ability of mobile robot formations when encountering dynamic obstacle interference. In [14], enhanced DWA is proposed to improve performance in environments filled with dense

objects. An additional component of the DWA objective function related to object density is introduced. Improvements enable more effective real-time navigation in complex environments, resulting in increased safety and operational efficiency for AGVs. The calculation time of the algorithm was not taken into account. The algorithm proposed in [15] integrates the Particle Swarm Optimization (PSO) algorithm with an improved DWA to achieve effective planning of AGV routes in a dynamic environment. The PSO has been used for global path planning, while the DWA has been used to track the optimal trajectory. Improvement in the DWA was related to modifying the objective function of the DWA to account for dynamic obstacle velocities. The computational effort has not been presented in the paper. Dynamic adaptive DWA was proposed in [16]. The algorithm enhances the DWA by introducing adaptive mechanisms that dynamically adjust its prediction horizon in response to environmental conditions and obstacle information. It allows the authors to improve the DWA algorithm in densely packed environments. Furthermore, the article proposes applying fuzzy logic to adapt the weights of the DWA objective function. Unfortunately, the calculation time of the modification was not presented in comparison to the original DWA.

The original DWA method for selecting the optimal pair of velocities is not computationally efficient. To improve the selection process, the author proposes an approach based on the Pattern Search (PS) optimization algorithm. The algorithm does not require a gradient calculation, and it can efficiently solve the local optimization problem. The algorithm calculates the candidates with a pre-defined step size in each dimension. If the algorithm finds a better solution, the current position is switched to the new one. Otherwise, the step size is reduced (divided by 2) and the procedure is repeated until the predefined resolution or the maximum number of function evaluations is reached. In this paper, the pattern search dynamic window approach (PSDWA) is proposed. The proposed approach enables a significant reduction in the computational effort required by the algorithm.

The paper is organized as follows. Section 2 describes the original Dynamic Window Approach. The proposed PSDWA is presented in Section 3, while the simulation results, along with a comparison to the original DWA, are presented in Section 4. The paper is summarized in Section 5.

## 2. DYNAMIC WINDOW APPROACH

The Dynamic Window Approach is a fundamental local path planning algorithm and is included as one of the default libraries in the Robot Operating System (ROS). DWA incorporates the prediction of future movements based on the specified linear and angular velocities of the AGV. The algorithm itself can be divided into two primary components:

- Search within the solution space,
- Optimization.

The first component is responsible for preparing a discrete set of possible control signals, i.e., linear and angular velocities. It comprises the following three elements:

- Circular trajectories: A key assumption of the dynamic window approach is the consideration of circular trajectories, which can be defined as pairs of linear and angular velocities  $(v, \omega)$ . This assumption reduces the path planning problem to a two-dimensional one.
- Admissible velocities: A given velocity pair is considered only

if the resulting trajectory is deemed safe, which means that it does not result in a collision with any obstacle. The admissibility condition for a velocity pair  $(v, \omega)$  is that the AGV must be able to stop before reaching the closest obstacle on the generated trajectory.

- Dynamic window: Limited AGV accelerations are assumed. Therefore, considering the AGV's current velocities, future velocities are allowed to change only within a bounded range defined by the maximum linear and angular accelerations.

The second component, namely optimisation, involves maximizing a cost function expressed by the following equation:

$$G(v, \omega) = \sigma(\alpha \cdot \text{heading}(v, \omega) + \beta \cdot \text{dist}(v, \omega) + \gamma \cdot \text{vel}(v, \omega)) \quad (1)$$

where:  $\alpha$ ,  $\beta$ , and  $\gamma$  are weighting parameters of the cost function, and  $\sigma$  denotes a smoothing function to keep the subobjectives (i.e. heading, dist and vel functions) in range  $<0,1>$  for entire DWA iteration. This cost function comprises the following components:

- Target heading: A reward is assigned for movement directed toward the target. The motion toward the goal gives the maximum value.
- Clearance: The dist function represents the minimum distance to an obstacle along a given trajectory. The smaller the distance to the obstacle, the higher the likelihood that the AGV will attempt to maneuver around it.
- Velocity: A reward is provided for a higher linear velocity in the cost function.

First, the algorithm generates possible linear and angular velocities within the maximum capabilities of the AGV. Additionally, assuming that the robot is currently moving with velocities  $(v_{AGV}, \omega_{AGV})$  and that the maximum linear and angular accelerations are  $(a^{max}, \epsilon^{max})$ , the upper and lower bounds for admissible velocities can be defined as:

$$v_{AGV}^{max}(k+1) = v_{AGV}(k) + a^{max} \cdot T_s \quad (2)$$

$$v_{AGV}^{min}(k+1) = v_{AGV}(k) - a^{max} \cdot T_s \quad (3)$$

$$\omega_{AGV}^{max}(k+1) = \omega_{AGV}(k) + \epsilon^{max} \cdot T_s \quad (4)$$

$$\omega_{AGV}^{min}(k+1) = \omega_{AGV}(k) - \epsilon^{max} \cdot T_s \quad (5)$$

where:  $T_s$  denotes the time at the  $k$ -th sample time. In this manner, the dynamic window defining the admissible range of linear and angular velocities is obtained. When this dynamic window is intersected with the maximum allowed AGV velocities, the set of permissible velocities (that is, the search space dictated by dynamics) is obtained. Therefore, two-dimensional search space is obtained by discretizing this range with a given resolution or dividing it into  $N_v$  and  $N_\omega$  values for linear and angular velocities, respectively.

The next step involves predicting future positions (that is, the trajectory of the AGV) using the following equations:

$$\theta(k+1) = \theta(k) + \omega_{AGV}(k) \cdot T_s \quad (6)$$

$$x_{AGV}(k+1) = x_{AGV}(k) + v_{AGV}(k) \cdot \cos(\theta(k+1)) \cdot T_s \quad (7)$$

$$y_{AGV}(k+1) = y_{AGV}(k) + v_{AGV}(k) \cdot \sin(\theta(k+1)) \cdot T_s \quad (8)$$

where:  $\theta$  denotes the AGV's orientation, and  $x_{AGV}$  and  $y_{AGV}$  are the robot's positions along the  $x$  and  $y$  axes respectively. Predictions are carried out over predefined time interval ( $H^{prediction}$ ). If a trajectory has been generated for every pair  $(v, \omega)$ , it can then

be verified whether the trajectory is collision-free, and whether the AGV is capable of stopping before encountering an obstacle. The second condition can be expressed as the following condition:

$$v \leq \sqrt{2 \cdot \text{dist}(v, \omega) \cdot a_b} \quad (9)$$

where:  $a_b$  denotes the deceleration values during braking. For a simplification the following assumption can be made:  $a_b = a^{\max}$ . Next, by excluding collision-prone trajectories, a set of linear and angular velocity pairs is obtained for the optimization process. For each pair, a quality index value is computed using eq. (1), and subsequently, the pair with the maximum value among all evaluated options is selected. This determines the new linear and angular velocity of the robot. The procedure of the DWA algorithm is presented in Fig. 1 in graphical representation for a single linear velocity for better readability.

The most computationally complex part of the algorithm is to predict the future trajectory and calculate the objective function for each pair of possible linear and angular velocities. Commonly, the fixed number of division of linear and angular velocities ( $N_v$  and  $N_\omega$ , respectively) are used. For example, the default values for Robot Operating System are as follows:  $N_v = 3$  and  $N_\omega = 10$ . Therefore, the 30 trajectories must be predicted to calculate the objective functions and select the best.

### 3. THE PROPOSED COMPUTATIONALLY EFFICIENT DYNAMIC WINDOW APPROACH

The proposed approach aims to reduce the computational effort of the DWA algorithm by integrating it with the Pattern Search optimization algorithm. The PS algorithm is employed to efficiently select new control signals for the AGV, rather than evaluating all possible solutions.

The Pattern Search algorithm is a direct method of finding the extremum of a function. It is a gradient-free optimization algorithm. The initial guess position and the initial step-size ( $\Delta$ ) are the only parameters of PS. Next, the neighborhood of the current point is examined. In the case of a better value in the neighborhood, the algorithm moves the current position to this point. When no improvement in the function's value is observed, the algorithm decreases the step size by a factor of two. The process is repeated iteratively until a stopping criterion is reached. PS is characterized by simplicity of implementation and robustness to the lack of smoothness of the objective function. However, its effectiveness may be limited for problems with high dimensionality or in the presence of local minima.

The most time-consuming part of the DWA algorithm is related to the optimization process. The original DWA algorithm determines the possible velocities and then calculates the objective function for each of these possibilities. To improve this part of the algorithm, the above-described PS optimization algorithm is applied. Therefore, the following possibilities are examined only if necessary. The procedure of the proposed PSDWA is as follows:

- Calculate the possible linear and angular velocities using eq. (2)-(5).
- Set the initial position for PS as the current linear and angular velocities.
- Set the initial step size as a quarter of the linear and angular range.
- Run PS, while the maximum number of evaluated solutions are not reached ( $\text{iter}^{\max}$ ).

- Set the current position of the PS algorithm as linear and angular velocities.

It should be noted that the DWA's objective function of DWA (eq. (1)) requires a smoothing function ( $\sigma$ ). Such a function normalizes the values of the subobjectives to range  $<0,1>$  using the minimum and maximum value obtained in the examined solutions. In the proposed approach, the objective functions are calculated and compared sequentially. In such a case the smoothing function is not simple to determine, due to lack of information of the maximum and minimum value of examined objective functions. To address this issue, Author proposes to normalize the components using constant values: maximum allowed AGV's linear velocity ( $v^{\max}$ ) for velocity,  $\pi$  for heading angle, and obstacle distance reaction ( $d_{\text{obstacle}}^{\min}$ ) for clearance.

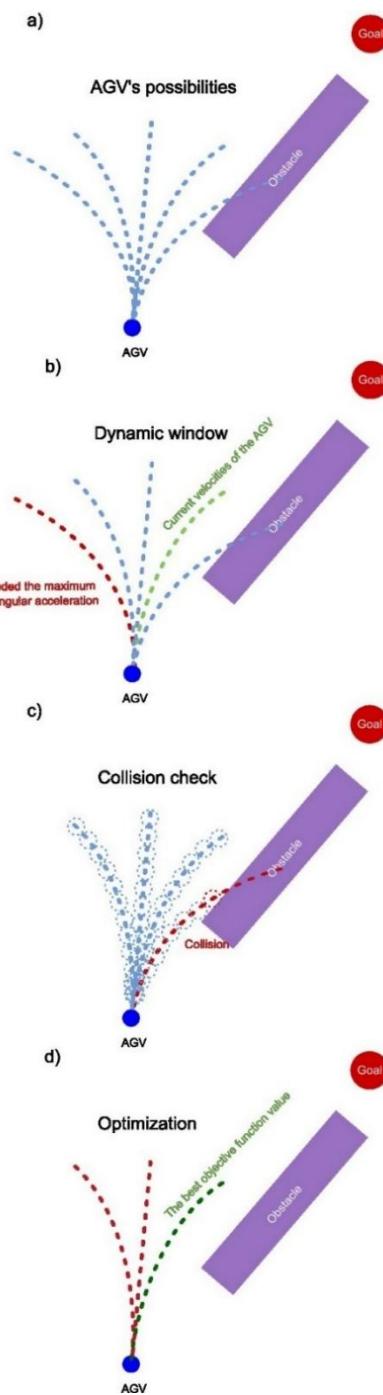


Fig. 1. Visualisation of the DWA procedure for a single linear velocity

## 4. RESULTS

The effectiveness of the proposed PSDWA test and the comparison with the original DWA will be presented. The validation was executed in a MATLAB environment. The source code has been published on MathWorks FileExchange [18]. To ensure that the proposed approach allows AGV to reach similar performance, the examination has been provided in a thousand randomly generated environments with ten static obstacles. The algorithms' parameters are presented in Tab. 1.

**Tab. 1.** Parameters of the examined local path planning algorithms

Parameter	Symbol	Value
Sampling period	$T_s$	0.01 s
Horizon of prediction	$H_{prediction}$	1.0 s
Maximum linear acceleration	$a^{max}$	1.0 m/s <sup>2</sup>
Maximum linear velocity	$v^{max}$	0.5 m/s
Maximum angular acceleration	$\epsilon^{max}$	1.0 · $\pi$ rad/s <sup>2</sup>
Maximum angular velocity	$\omega^{max}$	0.5 · $\pi$ rad/s
Obstacle distance reaction	$d_{obstacle}^{min}$	2.0
DWA: number of samples for linear velocity	$N_v$	3
DWA: number of samples for angular velocity	$N_\omega$	10
PSDWA: maximum number of examinations	$iter^{max}$	15
PSDWA: Initial step size	$\Delta^{initial}$	$\frac{1}{4} \left[ \frac{v_{AGV}^{max} - v_{AGV}^{min}}{\omega_{AGV}^{max} - \omega_{AGV}^{min}} \right]$

For comparison, statistical indicators commonly used for path planning algorithms have been used [15]:

- path length:  $L = \sum_{i=1}^{M-1} d(q_i^{x,y}, q_{i+1}^{x,y})$ ,
- smoothness:  $S = \sqrt{\frac{1}{M-2} \sum_{i=1}^{M-1} (q_i^\theta - q_{i+1}^\theta)^2}$ ,

where:  $M$  is number of samples,  $q_i^{x,y}$  is AGV position in  $i$ -th iteration,  $q_i^\theta$  is AGV orientation in  $i$ -th iteration, and  $d(\dots)$  is a function that calculated euclidean distance between two points. The obtained results are presented in Tab. 2. One can see that the quality indicators are very similar for the original DWA and the proposed PSDWA except for the mean computation time of the algorithm. The proposed method has limited the trajectory examination to 15, while the DWA parameters requires computation unit to examine 30. Therefore, the mean computation time of the algorithm is almost two times smaller. The difference between the path indicators is related to unpredictable resolution of the linear and angular velocities - the count of step-size reductions at each iteration may differ. Nevertheless, the speed-up of the proposed approach is close to 2 (i.e., mean computation time of DWA divided by mean computation time of the proposed PSDWA), and the path quality indicators are almost the same.

**Tab. 2.** Comparison of quality indicators obtained for DWA and the proposed PSDWA examined in a thousand randomly generated environments with ten obstacles

Quality indicator	DWA	PSDWA
Mean path length [m]	3.972	3.996
Mean smoothness [rad]	0.989	1.009
Mean goal-reaching time [s]	8.114	8.162
Mean computation time of the algorithm [ms]	22.97	11.60
Speed-up	1.00	1.98

The maximum number of examinations has been selected empirically in the above-described comparison. The specified value has been chosen to minimize computational effort while maintaining the high performance of the algorithm. However, the computational effort can be reduced if the lower performance is acceptable, or the higher performance can be achieved if the higher computational effort is possible.

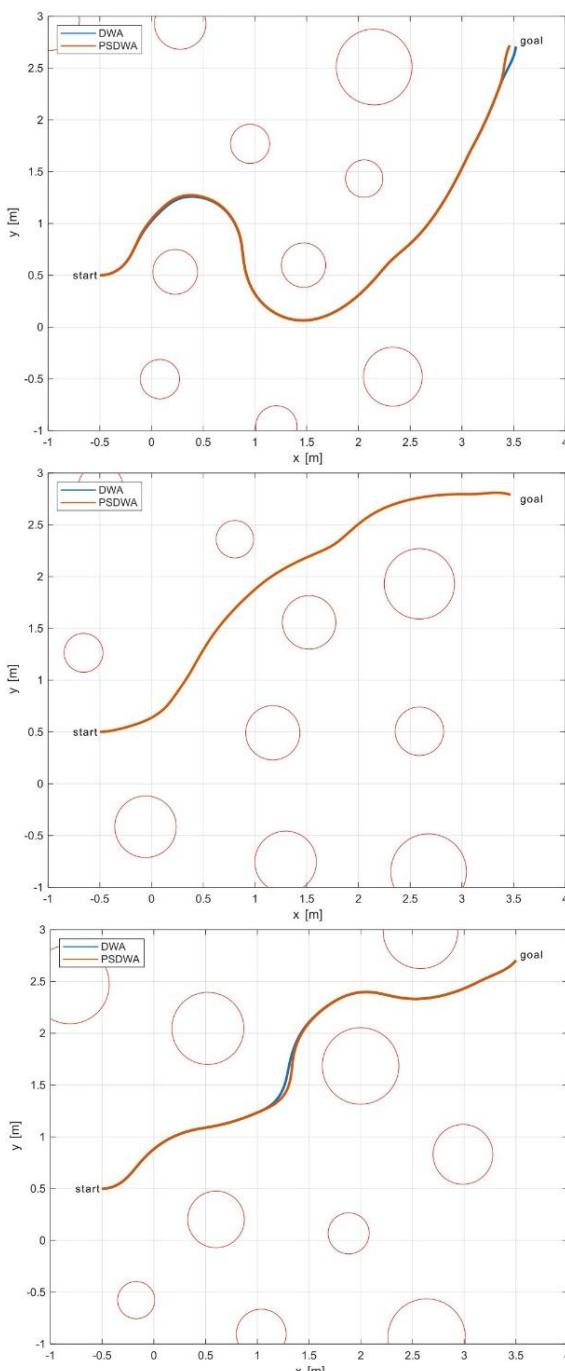
In Fig. 2, examples of paths obtained by DWA and PSDWA are presented. The graphical representation of the AGV's paths obtained for the original DWA and the proposed PSDWA confirms the performance of the PSDWA regardless of the reduced computational requirements.

It is worth noting that the computational effort of the DWA is strongly dependent on the density of obstacles. Therefore, a performance sensitivity analysis under different obstacle distributions has been conducted, and the results are presented in Table 3. The higher number of obstacles in the environment significantly increases computational time due to the prediction and optimization processes. However, the proposed modification decreases the number of predicted paths. In such a case, the speed-up indicator appears to be constant, regardless of the number of obstacles. For the two obstacles, the speed-up indicator is slightly lower due to the other parts of the algorithm, such as the prediction path and optimization process.

**Tab. 3.** Comparison of computation time of algorithms for various obstacle density in the environment

Number of obstacles	Mean computation time [ms]		Speed-up
	DWA	PSDWA	
2	$4.97 \pm 1.49$	$2.68 \pm 0.81$	1.85
6	$13.26 \pm 3.77$	$6.76 \pm 1.93$	1.97
10	$22.97 \pm 4.67$	$11.60 \pm 3.59$	1.98
15	$42.92 \pm 5.78$	$21.40 \pm 4.49$	2.01
30	$65.67 \pm 7.12$	$33.07 \pm 5.13$	1.99

The speed-up indicator is close to two due to the DWA number of velocity pairs being equal to 30, while for the proposed PSDWA, the maximum number of examinations was set to 15. This number can be increased or decreased. The higher number enables higher performance in local path planning, but it requires more computational resources. To illustrate its relationships, a comparison of path quality indicators and the maximum number of examinations is provided in Table 4.



**Fig. 2.** Example trajectories of the AGV obtained for the original DWA and the proposed PSDWA

**Tab. 4.** Comparison of quality indicators obtained for DWA and the proposed PSDWA with different number of examinations

Quality indicator	DWA	PSDWA		
		Maximum number of examinations ( $iter^{max}$ )		
		5	15	30
Mean path length [m]	6.23	7.03	6.19	6.18
Mean smoothness [rad]	1.063	1.253	1.059	1.056
Mean goal-reaching time [s]	12.60	14.57	12.58	12.56
Speed-up	1.00	5.31	1.93	0.92

## 5. CONCLUSIONS

In this paper, the combination of the Dynamic Window Approach and the Pattern Search optimization algorithm was proposed. To reduce the computational effort of the DWA algorithm, the procedure for selecting the next control signals (i.e., linear and angular velocities of the AGV) is based on a gradient-free optimization algorithm, specifically PS. The simulation examinations demonstrate that the proposed approach enables a reduction in calculation time by approximately two times, while maintaining the provided solution's smoothness as in the original DWA implementation. Moreover, the validation of the provided speed-up indicator for the proposed approach has been provided using environments with different obstacle densities. The maximum number of examinations has been examined to present possibilities for achieving a higher performance of the AGV with a similar calculation time to the original DWA.

It should be noted that the proposed approach can be implemented with more complex modifications of the DWA. The literature review in the Introduction section demonstrates that the improvement of the DWA algorithm primarily relies on modifications to the objective functions. Therefore, the selection based on the PS algorithm is still applicable to achieve a lower computational effort and higher AGV efficiency.

Future research will focus on experimental validation of the proposed approach using a real AGV and implementing PS with more complex DWA improvements proposed in the literature.

## REFERENCES

1. Bouhamatou Z, Abedssemed F. Fuzzy Synergetic Control for Dynamic Car-Like Mobile Robot. *Acta Mech Autom.* 2022 Jan 1;16(1):48–57. <https://doi.org/10.2478/ama-2022-0007>
2. Oyekanlu EA, Smith AC, Thomas WP, Mulroy G, Hitesh D, Ramsey M. A Review of Recent Advances in Automated Guided Vehicle Technologies: Integration Challenges and Research Areas for 5G-Based Smart Manufacturing Applications. *IEEE Access.* 2020; 8: 202312-53. <https://doi.org/10.1109/access.2020.3035729>
3. Kwaśniewski KK, Gosiewski Z. 3K Method: Time-optimal Path Planning for Field Robot. *Acta Mech Autom.* 2025 Mar 1;19(1):172–6. <https://doi.org/10.2478/ama-2025-0020>
4. Kwaśniewski KK, Gosiewski Z. Genetic Algorithm for Mobile Robot Route Planning with Obstacle Avoidance. *Acta Mech Autom.* 2018 Jun 1;12(2):151–9. <https://doi.org/10.2478/ama-2018-0024>
5. Howie C, Lynch KM, Hutchinson S. Principles of robot motion: theory, algorithms, and implementations. Cambridge (MA): MIT Press; 2005.
6. Cieślik J, Gosiewski Z, Ambroziak L, Romaniuk S. Computationally Simple Obstacle Avoidance Control Law for Small Unmanned Aerial Vehicles. *Acta Mech Autom.* 2015 Mar 1;9(1):50–6. <https://doi.org/10.1515/ama-2015-0010>
7. Fox D, Burgard W, Thrun S. The dynamic window approach to collision avoidance. *IEEE Robot Autom Mag.* 1997 March;4(1):23–33. <https://doi.org/10.1109/100.580977>
8. Chang L, Shan L, Jiang C, Dai Y. Reinforcement based mobile robot path planning with improved dynamic window approach in unknown environment. *Auton Robots.* 2020 September 30; 45:51-76. <https://doi.org/10.1007/s10514-020-09947-4>
9. Wang Z, Liang Y, Gong C, Zhou Y, Zeng C, Zhu S. Improved Dynamic Window Approach for Unmanned Surface Vehicles' Local Path Planning Considering the Impact of Environmental Factors. *Sensors.* 2022 Jul 11;22(14):5181. <https://doi.org/10.3390/s22145181>
10. Li C, Yao L, Mi C. Fusion Algorithm Based on Improved A\* and DWA for USV Path Planning. *J Mar Sci Appl.* 2023 Nov 25;24(1):224-37. <https://doi.org/10.1007/s11804-024-00434-1>

11. Zhang J, Ling H, Tang Z, Song W, Lu A. Path planning of USV in confined waters based on improved A\* and DWA fusion algorithm. *Ocean Eng.* 2025 Jan 31;322:120475.  
<https://doi.org/10.1016/j.oceaneng.2025.120475>
12. Ayalew W, Menebo M, Merga C, Negash L. Optimal path planning using bidirectional rapidly-exploring random tree star-dynamic window approach (BRRT\*-DWA) with adaptive Monte Carlo localization (AMCL) for mobile robot. *Eng Res Express.* 2024 Jul 25;6(3):035212.  
<https://doi.org/10.1088/2631-8695/ad61bd>
13. Cao Y, Nor NM. An improved dynamic window approach algorithm for dynamic obstacle avoidance in mobile robot formation. *Decis Anal J.* 2024 May 1;11:100471.  
<https://doi.org/10.1016/j.dajour.2024.100471>
14. Mai X, Li D, Ouyang J, Luo Y. An improved dynamic window approach for local trajectory planning in the environment with dense objects. *J Phys Conf Ser.* 2021 Apr 1;1884(1):012003.  
<https://doi.org/10.1088/1742-6596/1884/1/012003>
15. Yang Z, Li N, Zhang Y, Li J. Mobile Robot Path Planning Based on Improved Particle Swarm Optimization and Improved Dynamic Window Approach. *J Robot.* 2023 May 1; 2023:6619841.  
<https://doi.org/10.1155/2023/6619841>
16. Dobrevski M, Skočaj D. Dynamic Adaptive Dynamic Window Approach. *IEEE Trans Robot.* 2024;40:3068–81.
17. Hooke R, Jeeves TA. ``Direct Search'' Solution of Numerical and Statistical Problems. *J ACM.* 1961 Apr 1;8(2):212–29.  
<https://doi.org/10.1145/2F321062.321069>
18. Pattern Search Dynamic Window Approach - MATLAB Central File Exchange [Internet]. Mathworks.com. 2025 [cited 2025 Apr 27]. <https://www.mathworks.com/matlabcentral/fileexchange/180896-patternsearch-dynamic-window-approach>
19. Szczepański R. Safe Artificial Potential Field - Novel Local Path Planning Algorithm Maintaining Safe Distance From Obstacles. *IEEE Robot Autom Lett.* 2023 Aug 1;8(8):4823–30.  
<https://doi.org/10.1109/LRA.2023.3290819>

Rafal Szczepanski:  <https://orcid.org/0000-0002-7363-1618>



This work is licensed under the Creative Commons BY-NC-ND 4.0 license.