2023 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)
June 28-30, 2023. Seattle, Washington, USA

DQDWA: Dynamic Weight Coefficients Based on Q-learning for Dynamic Window Approach Considering Environmental Situations

Masato Kobayashi[1], Hiroka Zushi[2], Tomoaki Nakamura[2], and Naoki Motoi[2*]

***Abstract*— Autonomous mobile robots are used in a wide range of industrial application. Dynamic window approach (DWA) is one of effective local path planning methods considering collision avoidance and kinematic constraints. DWA selects the optical path from path candidates from velocity space by using an evaluation function with fixed weight coefficients. These fixed weight coefficients are designed for the specific environmental situation. Therefore, if the environmental situation such as congestion, road width, and obstacles changes, the evaluation function with fixed weight coefficients may select the inefficient path or path with the collision. To address this issue, this paper proposes the dynamic weight coefficients based on Q-learning for DWA considering environmental situations (DQDWA). Q-learning is one of reinforcement learning methods. The Q-table in DQDWA consists of states of robot and environmental situations, and actions of weight coefficients in DWA evaluation function. By using the learned Q-table, DQDWA dynamically selects weight coefficients and, the optimal path considering environmental situations is generated. The effectiveness of the proposed method was confirmed through simulations.**

I. INTRODUCTION

In recent years, the declining birthrate and aging population have been considered to be a serious problem. Therefore, workforce creation through autonomous mobile robots is desired in various situations. To work in any environment, robots are required to move the goal position while avoiding obstacles. For the realization of such movement, autonomous mobile technology should be developed. Autonomous mobile technology consists of localization [1], mapping [2], perception [3], and path planning [4].

This paper focuses on path planning. Path planning is divided into two parts; one is global path planning, and the other is local path planning [5]. Global path planning generates the subgoals from the start position to the goal position. Local path planning generates the robot motion to the subgoal considering collision avoidance in real-time.

This paper develops local path planning in static environments such as factories and warehouses. Dynamic window approach (DWA) is widely used as the local path planning method [6]. Many methods related to improving DWA have been reported [7]–[9]. DWA generates path candidates from robot velocity space considering collision avoidance and kinematic constraints. The evaluation function selects the optimal path from path candidates considering the goal position, obstacle distance, and robot velocity. The optimal path depends on the weight coefficients of the evaluation function. The fixed weight coefficients are suitable for a specific situation. However, if the environment situation is changed, the inefficient path or path with collision may be selected.

To solve this problem, dynamic weight coefficients in DWA have been developed [11], [12]. These methods adjusted weight coefficients in real-time by using fuzzy logic with analysis of goal positions and obstacles. In addition, dynamic weight coefficients with Q-learning was proposed [13]. Q-learning is one of the reinforcement learning methods [14]. This method improved the DWA evaluation functions and adjusted the weight of each subfunction by using a trained Q-learning agent.

For these features, we focus on the Q-learning method to deal with the adjustment of weight coefficients. In the conventional method [13], the area of the spaces and congestion rate were not considered as environmental situations. To address this issue, this paper proposes the dynamic weight coefficients based on Q-learning for DWA considering environmental situations (DQDWA). DQDWA defines environmental situations as goal distance, goal direction, velocity, visible area, and congestion. DQDWA dynamically adjusts the weight coefficients of the evaluation function for environmental situations by using Q-learning.

This paper consists of seven sections including this one. Section II shows the modeling of the mobile robot. DWA and Q-learning are explained as the conventional methods in Sections III and IV. Section V proposes DQDWA to address the environmental change. In Sections VI, simulation results are shown to confirm the usefulness of the proposed method. Section VII concludes this paper.

II. COORDINATE SYSTEM

Fig. 1 shows the coordinate system of the robot. As shown in Fig. 1, there are two coordinate systems; one is the local coordinate system Σ_{LC}, and the other is the global coordinate system Σ_{GB}. The superscript ${}^{GB}\bigcirc$ means the value in Σ_{GB}, and the variables in Σ_{LC} do not use the superscript. The origins in Σ_{GB} and Σ_{LC} are set as an initial robot position and the center point of both wheels. The direction of X-axis matches the forward direction of the robot, and the direction of Y-axis is the vertical left of X-axis. $({}^{GB}x, {}^{GB}y)$ and ${}^{GB}\theta$ refer to the position and angle of the robot in the global coordinate system. L^{rob} is the radius of the robot.

[1]Cybermedia Center, Osaka University, Toyonaka, Japan kobayashi.masato.cmc@osaka-u.ac.jp
[2]Graduate School of Maritime Sciences, Kobe University, Kobe, Japan motoi@maritime.kobe-u.ac.jp

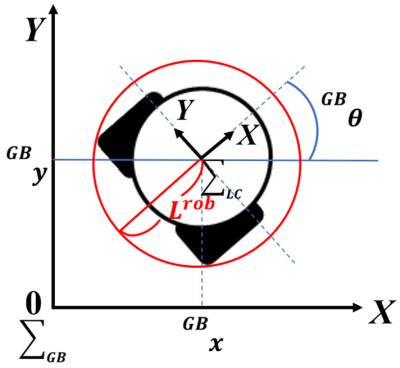
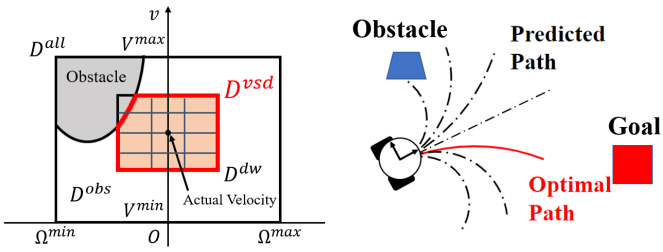


Fig. 1. Modeling of Robot



(a) VSD (b) Optimal Path of DWA

Fig. 2. Image of DWA

III. Dynamic Window Approach (DWA)

Dynamic Window Approach (DWA) is one of the actual local path planning methods [6]. Firstly, the velocity space with dynamic constraints (VSD) is searched based on the current velocities of the robot. The velocity space D^{vsd} as shown in Fig. 2 (a) is defined as follows.

$$D^{vsd} = D^{all} \cap D^{dw} \cap D^{obs} \tag{1}$$

where D^{all} and D^{dw} represents the velocity range determined from the robot specifications, and velocity range that can be generated at the next time step. D^{obs} indicates the velocity range without the collision.

Secondly, the optimal path is selected from the VSD using an evaluation function at each time step. As shown in Fig. 2 (b), the predicted trajectories are calculated from the velocities in D^{vsd} as path candidates. By maximizing evaluate function J, the optimal velocity pair which consists of the translational and angular velocity is chosen from the dynamic window.

$$J = W^{gol} \cdot c^{gol} + W^{vel} \cdot c^{vel} + W^{obs} \cdot c^{obs} \tag{2}$$

where c^{gol}, c^{vel}, and c^{obs} represent distance between the robot position and goal position, the current translational velocity, and the distance from the robot to the nearest obstacle, respectively. W^{gol}, W^{vel} and W^{obs} are weighting coefficients.

The details of the DWA are further elaborated in [6].

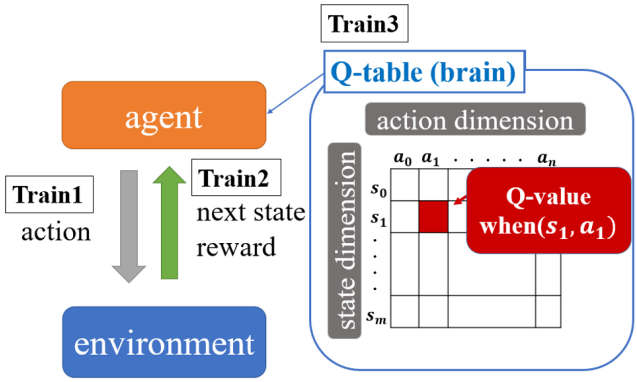


Fig. 3. Concept of Q-learning

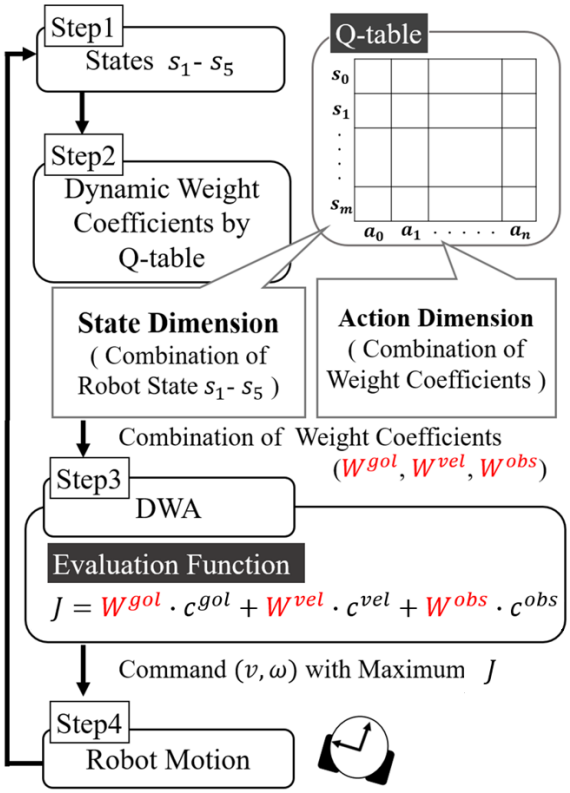


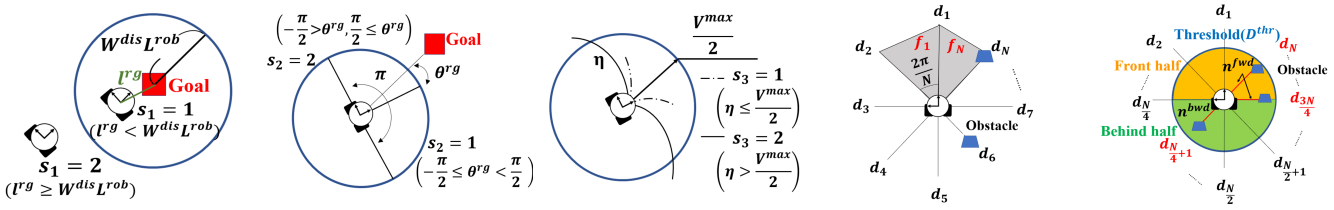
Fig. 4. Overview of DQDWA

IV. Q-learning

This section introduces Q-learning as one of the reinforcement learning methods [14]. Fig. 3 shows the concept of Q-learning. As shown in Fig. 3, there are three steps. In Train1, the agent chooses the action a of the current state s. The ϵ-greedy method with Q-table is used for a. In Train2, the agent receives the next state and reward R from the environment. In Train3, the Q-value in the Q-table is updated.

$$Q(s,a) = (1-\alpha)Q(s,a) + \alpha[R(s,a) + \gamma Q(s',a)] \tag{3}$$

where α and γ are the learning rate and the discount rate. $R(s,a)$ and $Q(s',a)$ represent the reward of the agent and the



(a) s_1 (b) s_2 (c) s_3 (d) s_4 (e) s_5

Fig. 5. Image of State in Proposed Method

maximum Q-value in the next state. The Q-table is a $m \times n$ matrix, where m and n are the numbers of actions. Trains1-3 are repeated until the Q-table converges to a threshold value.

V. Proposed Method (DQDWA)

This section proposes the dynamic weight coefficients of the evaluation function for DWA considering environmental situations (DQDWA). In the conventional method [13], the area of the spaces and congestion rate were not considered as environmental situations. Therefore, an inefficient path or path with a collision may be selected, if the environmental situation is changed. To address this issue, we define area and congestion as environmental situations.

Fig. 4 shows the image of DQDWA. In Step1, the state $\boldsymbol{s} = [s_1\ s_2\ s_3\ s_4\ s_5]^T$ is calculated. Fig. 5 shows the image of $\boldsymbol{s}$. s_1, s_2, s_3, s_4 have 2 patterns and s_5 has 4 patterns. Therefore, the size of the state dimension is 64. Each definition of the states is described.

s_1 is the state related to the distance between the robot position and the goal position. s_1 is defined as follows.

$$s_1 = \begin{cases} 1 & \text{if} \quad l^{rg} < W^{dis}L^{rob} \\ 2 & \text{otherwise} \end{cases} \tag{4}$$

where W^{dis} is the weight coefficient of goal distance. l^{rg} is the distance between the robot position and the goal position.

s_2 is the state to represent the angle difference between the robot and the goal position. s_2 is defined as follows.

$$s_2 = \begin{cases} 1 & \text{if} \quad \theta^{rg} \in [-\frac{\pi}{2}, \frac{\pi}{2}) \\ 2 & \text{otherwise} \end{cases} \tag{5}$$

where θ^{rg} is the angle between the robot and the goal position.

s_3 is the state related to the traveled distance between the current position and the position after one second. Traveled distance η is calculated as follows.

$$\eta = \begin{cases} \frac{2v}{\omega} & \text{if} \quad |\omega| > \pi \\ v & \text{else if} \omega = 0 \\ \frac{2v}{\omega}\sin(\frac{\omega}{2}) & \text{otherwise} \end{cases} \tag{6}$$

s_3 is defined as follows.

$$s_3 = \begin{cases} 1 & \text{if} \quad |\eta| \leq \dfrac{V^{max}}{2} \\ 2 & \text{otherwise} \end{cases} \tag{7}$$

where V^{max} is the maximum translational velocity of the robot.

s_4 is the state to quantify the area of the space. For the state s_4, the divided area f_i is defined as follows.

$$f_i = \frac{1}{2} d_i d_{i+1} \sin(\frac{2\pi}{N}) \tag{8}$$

where d_i is the i-th distance data measured by the distance sensor. N is the number of distance data. Note that if $i = N$, d_1 is used instead of d_{i+1}. d_1 is the distance data directly in front of the robot. From there, distance data is obtained in $\frac{2\pi}{N}$ radian increments counter-clockwise. The summation of the divided area f^{all} is calculated as follows.

$$f^{all} = \sum_{i=1}^{N} f_i \tag{9}$$

s_4 is defined as follows.

$$s_4 = \begin{cases} 1 & \text{if} \quad f^{all} \leq W^{are}F^{max} \\ 2 & \text{otherwise} \end{cases} \tag{10}$$

where W^{are} is the weight coefficient of the area. F^{max} is the maximum summation of the divided area.

s_5 is the state related to the congestion. s_5 is defined according to the number of obstacles around the robot.

$$s_5 = \begin{cases} 1 & \text{if} \quad n^{fwd} > \dfrac{N}{4} \\ 2 & \text{else if} \quad n^{bwd} > \dfrac{N}{4} \\ 3 & \text{else if} \quad n^{all} > \dfrac{N}{4} \\ 4 & \text{otherwise} \end{cases} \tag{11}$$

where n^{fwd} and n^{bwd} are the number of sensor data within the distance threshold D^{thr} in the front and behind half of the robot. n^{all} is number of all sensor data ($n^{all} = n^{fwd} + n^{bwd}$).

A. Definition of reward

To adjust the weight coefficients of the evaluation function with Q-learning, the reward R is defined as follows.

$$R = R_1 + R_2 + R_3 \tag{12}$$

where R_1, R_2, and R_3 are rewards related to the goal or collision, distance from the goal position, and distance from

TABLE I

CONTROL PARAMETERS

V^{max}	Maximum Translational Velocity	0.22 $[m/s]$
V^{min}	Minimize Translational Velocity	-0.1 $[m/s]$
Ω^{max}	Maximum Angular Velocity	1.5 $[rad/s]$
Ω^{min}	Minimize Angular Velocity	-1.5 $[rad/s]$
$\dot{V}^{max}$	Maximum Translational Acceleration	2.5 $[m/s^2]$
$\dot{\Omega}^{max}$	Maximum Angular Acceleration	3.2 $[rad/s^2]$
T^{max}	Maximum Predicted Time	4.0 $[s]$
ΔT	Time Step	0.2 $[s]$
D^{thr}	Distance Threshold on s_5	1.5$[m]$
L^{rob}	Robot Radius	0.13$[m]$
W^{are}	Weight Coefficient of Area	0.3
W^{dis}	Weight coefficient of Goal Distance	3
F^{max}	Max Area in s_4	38$[m^2]$
N	Number of Sensor Data	24

the obstacle.

$$R_1 = \begin{cases} 5000 & \text{if reach goal} \\ -200 & \text{else if collide obstacle} \\ -2 & \text{otherwise} \end{cases} \quad (13)$$

$$R_2 = \begin{cases} 10 & \text{if approach goal} \\ -10 & \text{Otherwise} \end{cases} \quad (14)$$

$$R_3 = \begin{cases} -5 & \text{if approach obstacle} \\ 5 & \text{Otherwise} \end{cases} \quad (15)$$

B. Definition of action dimension

The weight coefficients of position, velocity, and obstacles are each selected from 1, 2, and 3. Note that we removed the set {2,2,2}, {3,3,3} because it has the same meaning as {1,1,1}. Finally, the 25 combinations are obtained and they constitute the action dimension.

Thus, the Q-table consists of actions of weight coefficients in the evaluation function, states of the robot, and environmental situations. By using the learned Q-tables, DQDWA selects the optimal path by using dynamic weight coefficients considering environmental situations.

VI. SIMULATION

A. Simulation Setup

The simulation system was implemented by Robot Operating System (ROS) and Gazebo. In this simulation, we simulated the four patterns; DQDWA, and DWA with fixed weight coefficients such as DWA I, DWA II, and DWA III. The fixed weight coefficients $\{W^{gol}, W^{vel}, W^{obs}\}$ of DWA I, DWA II, and DWA III were set as {1,1,2}, {1,2,1}, and {2,1,1}, respectively. Table I shows simulation parameters.

B. Pre-Train of Q-table

Fig. 6 (a)-(e) show the environments used in the learning process. Environments and goal positions were randomly selected at the beginning of each trial as shown in Table II. ${}^{GB}x^{gol}$ and ${}^{GB}y^{gol}$ are the X and Y coordinates of the goal positions. $([-1.2, 1.2], [-1.2, 1.2])$ means ${}^{GB}x^{gol}$ and ${}^{GB}y^{gol}$ which are randomly selected from range of $[-1.2, 1.2]$. The red-filled area in Fig. 6 indicated the goal position. As shown in Fig. 6 (a)-(c), Env. 1-Env. 3 were set up to verify differences in robot behavior due to crowding in a restricted space. As shown in Fig. 6 (d), Env. 4 was to examine robot behavior in a wide space with many obstacles. In Fig. 6 (e), Env. 5 was to evaluate robot behavior with obstacles and humans. All environments were designed assuming the use in warehouses and factories. The learning was continued until the Q-table was updated 30,000 times.

TABLE II

GOAL POSITIONS IN EACH ENVIRONMENT OF LEARNING PHASE

Environment	$({}^{GB}x^{gol}, {}^{GB}y^{gol})$
$Env.$ 1	$([-1.2, 1.2], [-1.2, 1.2])$ (resolution: 0.1)
$Env.$ 2	$([-1.2, 1.2], [-1.2, 1.2])$ (resolution: 0.1)
$Env.$ 3	(1.3,1.6), (-1.3,1.5), (0.5,-1.5), (-0.5,-1.5)
$Env.$ 4	(0.0,4.0), (-0.5,3.0), (0.0,3.0), (-1.5,2.5), (-2.0,0.5), (0.5,-3.0), (1.0,-3.0), (1.5,-2.5)
$Env.$ 5	(0.0,8.0)

TABLE III

CASE S1 RESULTS (1TIME IN EACH ENVIRONMENT)

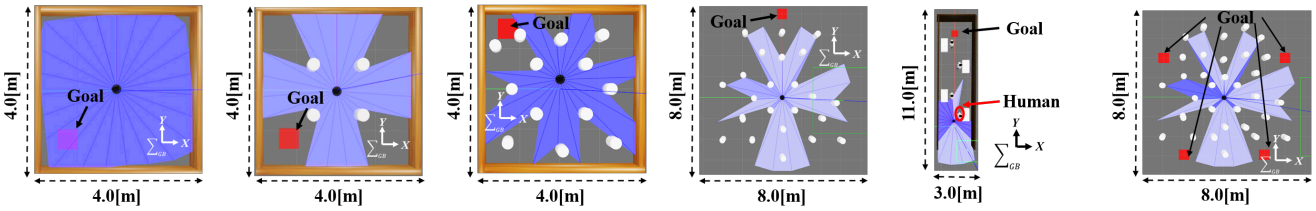
Environment	Method	Success Rate [%]	Time [sec]	TL [m]	PD [rad]
$Env.$ 1 (1 Time)	DWA I	100	19.9	1.75	3.88
	DWA II	100	14.9	1.98	2.95
	DWA III	100	16.4	2.06	10.4
	DQDWA	100	14.3	1.98	3.35
$Env.$ 2 (1 Time)	DWA I	100	15.4	2.02	9.47
	DWA II	100	15.3	2.19	9.38
	DWA III	100	18.2	2.25	4.19
	DQDWA	100	14.0	2.04	3.12
$Env.$ 3 (1 Time)	DWA I	100	21.5	2.32	4.33
	DWA II	100	15.1	2.14	1.60
	DWA III	0	-	-	-
	DQDWA	100	15.5	2.22	2.72
$Env.$ 4 (1 Time)	DWA I	100	26.6	4.14	3.35
	DWA II	100	24.3	3.96	2.32
	DWA III	0	-	-	-
	DQDWA	100	23.8	3.96	2.04
$Env.$ 5 (1 Time)	DWA I	100	46.7	7.84	5.33
	DWA II	100	44.0	7.65	4.34
	DWA III	100	43.6	7.55	3.72
	DQDWA	100	42.8	7.55	3.83

C. Simulation Environment

In this simulation, the following two types of simulations were conducted. Case S1 is 1 time simulation in each environment (Env. 1-5). Case S2 is 30 times simulations in unlearned environment (Env. 6). The start position was $({}^{GB}x^{sta}, {}^{GB}y^{sta}) = (0.0, 0.0)$ in Case S1-S2. In Case S1, the goal positions of Env. 1-5 $({}^{GB}x^{gol}, {}^{GB}y^{gol})$ were set $(-1.2, -1.2)$, $(-1.2, -1.2)$, $(-1.3, 1.5)$, $(0.0, 4.0)$, and $(0.0, 8.0)$. In Case S2, the goal positions were randomly chosen from red-filled areas as shown in Fig. 6 (f). Env. 6 was an environment with even more obstacles than Env. 4. It tests the robot's ability to handle obstacles that were not present during the learning phase.

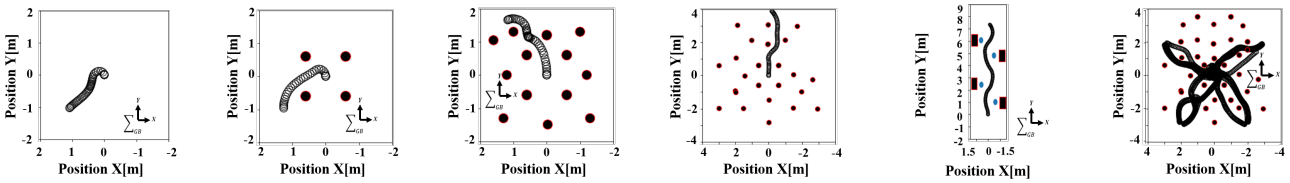
D. Simulation Results

1) Case S1: Table III and Table IV show the results of Case S1. TL and PD mean the trajectory length and the movement posture displacement. Table IV shows the collision numbers and average of time, TL, and PD. Figs. 7-10 show the trajectories in each environment.



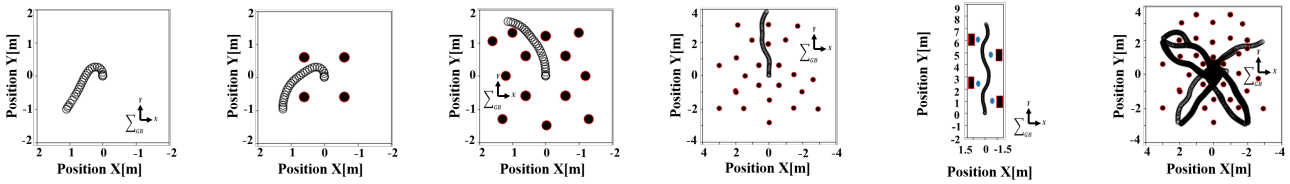
(a) Env. 1 (b) Env. 2 (c) Env. 3 (d) Env. 4 (e) Env. 5 (f) Env. 6

Fig. 6. Image of Each Environment



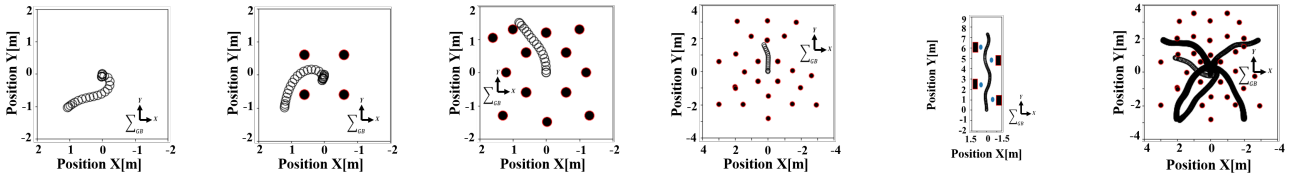
(a) Env. 1 (b) Env. 2 (c) Env. 3 (d) Env. 4 (e) Env. 5 (f) Env. 6

Fig. 7. Trajectories of DWA I ($\{W^{gol}, W^{vel}, W^{obs}\}$ ={1,1,2})



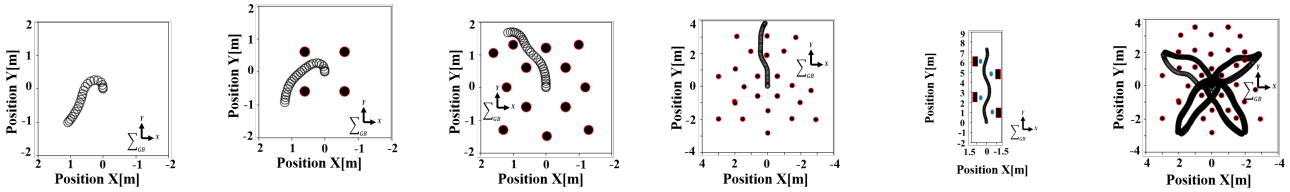
(a) Env. 1 (b) Env. 2 (c) Env. 3 (d) Env. 4 (e) Env. 5 (f) Env. 6

Fig. 8. Trajectories of DWA II ($\{W^{gol}, W^{vel}, W^{obs}\}$ ={1,2,1})



(a) Env. 1 (b) Env. 2 (c) Env. 3 (d) Env. 4 (e) Env. 5 (f) Env. 6

Fig. 9. Trajectories of DWA III ($\{W^{gol}, W^{vel}, W^{obs}\}$ ={2,1,1})



space-4mm

(a) Env. 1 (b) Env. 2 (c) Env. 3 (d) Env. 4 (e) Env. 5 (f) Env. 6

Fig. 10. Trajectories of the Proposed Method (DQDWA)

In DWA I-III, though favorable results in some environments were shown, the robot sometimes had the collision. In addition, it took a long time to reach the goal position. Simulation results in DWA I-III were dependent on the environmental situation since these methods used fixed weight coefficients.

In DQDWA, it succeeded in reaching the goal position in the shortest time and the smallest TL and PD. It is because DQDWA considers space and congestion. In other words, optimal weight coefficients depending on each environment were selected. DQDWA allows for more efficient routing while maintaining safety and not wandering in the same place.

2) Case S2: Table V shows the results of Case S2. Fig. 7-10 (f) show the trajectories in Case S2. In this simulation, the goal position was randomly set at the beginning of each

TABLE IV

CASE S1 RESULTS (AVERAGE IN EACH ENV.)

Method	Collision [-]	Time Average [sec]	TL Average [m]	PD Average [rad]
DWA I	0	26.0	3.61	5.27
DWA II	0	22.7	3.58	4.12
DWA III	2	26.1	3.95	6.10
DQDWA	0	22.1	3.55	3.01

TABLE V

CASE S2 RESULTS (30 TIMES)

Environment	Method	Success Rate [%]	Time [sec]	TL [m]	PD [rad]
$Env.\ 6$ (30 Times)	DWA I	93	29.1	4.27	11.76
	DWA II	80	26.8	4.25	8.76
	DWA III	70	27.5	4.03	7.48
	DQDWA	93	27.3	4.10	7.83

trial. The goal position in Env. 6 is selected from four points; $({}^{GB}x^{gol}, {}^{GB}y^{gol})$ were $(2.0, -3.0)$, $(3.0, 2.0)$, $(-2.0, -3.0)$, and $(-3.0, 2.0)$.

DWA I took a long time to reach the goal position instead of having a high success rate. DWA II and DWA III recorded small time, TL, and PD. However, their success rate was relatively low, since the translational velocity and goal distance were given as high priority compared with avoiding obstacles.

DQDWA recorded the high success rate. In addition, it took almost the same time as the DWA II which prioritizes translational velocity. TL and PD were also as small as DWA III which prioritizes the goal distance. Therefore, DQDWA selected efficient paths while maintaining safety in unlearned environments.

The effectiveness of the proposed method was confirmed by the simulation results of Case S1 and Case S2.

VII. CONCLUSION

This paper proposed DQDWA; the dynamic weight coefficients based on Q-learning for DWA considering environmental situations. We focused on state definition for Q-learning and added definitions for the area of spaces considering the crowded areas. With DQDWA, the robot could select optimal paths thanks to the adjustment of weight coefficients. The effectiveness of the proposed method was demonstrated in simulations.

ACKNOWLEDGMENT

This work was partly supported by the International Affairs Special Funding from the Graduate School of Maritime Sciences, Kobe University.

REFERENCES

[1] S. Zhang, J. Shan, and Y. Liu, "Variational Bayesian Estimator for Mobile Robot Localization With Unknown Noise Covariance", *IEEE/ASME Transactions on Mechatronics*, Vol. 27, No. 4, pp. 2185-2193, 2022.

[2] R. Liu, Y. He, C. Yuen, B.P.L. Lau, R. Ali, W. Fu, and Z. Cao, "Cost-Effective Mapping of Mobile Robot Based on the Fusion of UWB and Short-Range 2-D LiDAR", *IEEE/ASME Transactions on Mechatronics*, Vol. 27, No. 3, pp. 1321-1331, 2022.

[3] A. Bonci, P.D.C. Cheng, M. Indri, G. Nabissi, and F. Sibona, "Human-Robot Perception in Industrial Environments: A Survey", *Sensors*, Vol.21, No. 521, pp. 1571-1579, 2021.

[4] J. Wang, and M.Q.H. Meng, "Socially Compliant Path Planning for Robotic Autonomous Luggage Trolley Collection at Airports", *Sensors,* vol. 19, no. 12, pp. 2759-2773, 2019.

[5] P. Marin-Plaza, A. Hussein, D. Martin, and A.D.L. Escalera, "Global and Local Path Planning Study in a ROS-Based Research Platform for Autonomous Vehicles", *Journal of Advanced Transportation*, Vol. 2018, pp. 1-10, 2018.

[6] 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.

[7] M. Kobayashi, N. Motoi, "Local Path Planning: Dynamic Window Approach with Virtual Manipulators Considering Dynamic Obstacles", *IEEE Access*, Vol. 10, pp. 17018-17029, 2022.

[8] U. Patel, N.K.S. Kumar, A.J. Sathyamoorthy, and D. Manocha, "DWA-RL: Dynamically Feasible Deep Reinforcement Learning Policy for Robot Navigation among Mobile Obstacles", *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 6057-6063, 2021.

[9] Z. Huo, S. Dai, M. Yuan, X. Chen, and X. Zhang, "A Reinforcement Learning Based Multiple Strategy Framework for Tracking a Moving Target", *Proceedings of IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pp. 1292-1297, 2020.

[10] M. Kobayashi and N. Motoi, "Local Path Planning: Dynamic Window Approach With Virtual Manipulators Considering Dynamic Obstacles", *IEEE Access*, Vol. 10, pp. 17018–17029, 2022.

[11] O.A. Abubakr, M.A. Jaradat, and M.F. Abdel-Hafez, "Intelligent Optimization of Adaptive Dynamic Window Approach for Mobile Robot Motion Control Using Fuzzy Logic", *IEEE Access*, Vol. 10, pp. 119368-119378, 2022.

[12] Z. Hong, S. Chun-Long, Z. Zi-Jun, A. Wei, Z. De-Qiang, and W. Jing-Jing, "A Modified Dynamic Window Approach to Obstacle Avoidance Combined with Fuzzy Logic", *Proceedings of International Symposium on Distributed Computing and Applications for Business Engineering and Science*, pp. 523-526, 2015.

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

[14] C.J.C.H. Watkins, "Learning from delayed rewards", King's College, 1989.

[15] A.S. Mignon, and R.L.A. Rocha, "An adaptive implementation of ϵ-greedy in reinforcement learning", *Procedia Computer Science*, Vol. 109, pp. 1146-1151, 2017.