Proceedings of the 6th International Conference on Computing and Data Science
DOI: 10.54254/2755-2721/95/20241751

Low-cost guide dog robot navigation using Dueling DQN

Feiran Fang

Cape Cod Academy, Sandwich, Massachusetts, USA

vanessafang123@gmail.com

Abstract. Traditional aids for the visually impaired, such as guide dogs, have limitations. They are expensive to train, require a long preparation time, and are not accessible to most people. Guide robots have the potential to address these issues. However, most existing navigation methods focus solely on the robot's trajectory, neglecting the movement of the person. As a result, the paths of the visually impaired individual and the robot may not overlap or may intersect with obstacles. To overcome these challenges, we propose an intelligent guide dog system based on a Deep Q-network (DQN). First, a kinematic model is constructed to calculate the relative positions of the guide robot and the human. Then, the reward function is designed based on this kinematic model. Finally, the Dueling DQN is trained to plan a path and predict the human's possible trajectory, ensuring collision avoidance for both the robot and the human. Our experiments demonstrated that our system performs well in both simulated environments and in real-world scenarios, such as climbing stairs. The robot can accurately find a safe path that considers the trajectories of both the robot and the human, successfully completing tasks safely.

Keywords: Guide Dog Robot, Reinforcement Learning, Navigation, Dueling DQN, Kinematic Model

1. Introduction

Guide dogs play a crucial role in aiding the visually impaired with mobility. However, the selection and training of each guide dog is a labor-intensive and time-consuming process. Additionally, training dogs to perform significant responsibilities often conflicts with their natural instincts. With recent advancements in robotics technology, guide dog robots have emerged as a promising alternative for addressing mobility challenges faced by the visually impaired. In light of these developments, this paper proposes the Intelligent Guide Dog Robot (IGDR), a cost-effective solution designed to better serve individuals with visual impairments.

A primary function of the guide dog robot is navigation and obstacle avoidance. Currently, most existing path planning methods for Intelligent Guide Dog Robots (IGDRs) focus solely on the robot itself and assume that the follower will always move along the IGDR's trajectory[1,2,3]. Kaveh Akbari Hamed et al., proposed a hierarchical control strategy for IGDRs that addresses the issue of overshoot and undershoot in the follower's trajectory[4]. Meanwhile, Kulyukin et al. employed pre-installed RFID tags along frequently traveled routes to guide the IGDR to its destination [5]. Similarly, Tzu-Kuan Chuang et al. , trained a Convolutional Neural Network (CNN) model to detect colorful trails, enabling the tracking car to follow these trails[6]. However, the trajectories of the follower and the IGDR do not always align perfectly. The movement of the follower is dependent on the IGDR's motion, and when the IGDR transitions from linear to circular motion, the angle between the follower and the IGDR converges

DOI: 10.54254/2755-2721/95/20241751

exponentially to a stable state. The paths of the IGDR and the follower form two concentric circles with the same center but different radii. Therefore, IGDR navigation must ensure that the follower avoids obstacles, rather than focusing solely on the robot's path. This makes IGDR navigation a typical multi-constraint, multi-objective optimization problem. Compared to traditional path planning algorithms such as A*, RRT, GA, and PSO, Deep Reinforcement Learning (DRL) offers significant advantages in addressing such complex optimization challenges.

In addition to navigation, two-way interaction between the human and robot is also crucial for the IGDR. J. Taery Kim et al. investigated the effects of rotating rod and rigid harness models on human-robot interactions[7]. Anxing Xiao et al. proposed a hybrid physical Human-Robot Interaction (HRI) model that describes the relationship between the follower and the robot under both taut and slack leash conditions[8]. Shozo Saegusa et al. developed a human-robot interface framework capable of recognizing the follower's walking conditions[9]. Yuanlong Wei et al. introduced a "smart rope" system designed to enhance human-robot interactions[10].

Therefore, the objectives of this study are to develop an intelligent guide dog robot system that addresses the limitations of traditional aids for the visually impaired. To achieve these goals, we have created a Dueling DQN-based navigation system that integrates a kinematic model to calculate the relative positions of the guide robot and the human, ensuring that neither the robot nor the follower collides with obstacles. Additionally, a cane mounted at the back of the IGDR is used to determine the relative position of the follower and robot, and it facilitates communication between the follower and the robot. This new algorithm has been tested and validated with our prototype guide dog robot.

2. Robot and Kinematic Model

2.1. Hardware and Human-Robot Interactions

The guide dog robot should be designed with flexibility to navigate obstacles such as stairs and small stones. Additionally, to make it accessible to more people, its cost needs to be kept as low as possible. To address these needs, we developed a cost-effective Intelligent Guide Dog Robot (IGDR) based on the Rocker-Bogie suspension system, which is capable of traversing stairs. Nearly all components are 3D printed to keep production costs down.

To improve human-robot interaction, the robot is equipped with a cane attached to its rear. Two Angular Displacement Sensors are mounted on the bottom of the cane to monitor the relative position between the user and the robot based on the sensor data and the length of the cane. Additionally, control buttons on the cane handle allow users to operate the robot. The Intelligent Guide Dog Robot, depicted in Figure 1, also features a Lidar sensor on its back to detect the surrounding environment. The robot employs a six-wheel differential drive system for enhanced steering and maneuverability.

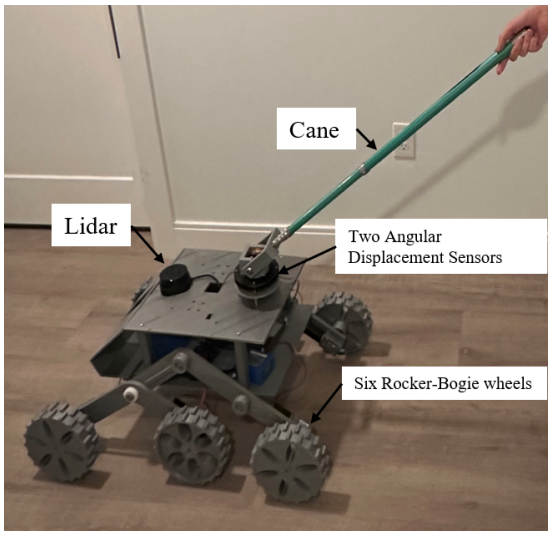


Figure 1. The Intelligent Guide Dog Robot, When the robot moves, a traction force is applied to the individual, and it is assumed that the individual's movement is in the direction of the applied traction force.

DOI: 10.54254/2755-2721/95/20241751

Key parameters of the IGDR are detailed in Table 1. The cane is designed to be over 1.5 meters long to ensure the user maintains a safe distance from the robot, reducing the risk of collision.

Table 1. The main technical parameters of the IGDR

Parameter	Value
Size(Length * Width)(mm)	700*500
Maximum Operating Speed(ms-1)	0.36
Cane Length(m)	1.5
Cost($)	400
Lidar range(m)	10
Battery(mAh)	18000

2.2. Kinematic Model of the Robotic Guide Dog

The robotic guide dog and its user can be modeled as a tractor-trailer wheeled system, connected by a rigid cane. The mathematical model for this system is illustrated in Figure 2. To develop this model, the following assumptions are made:

1. The vehicle operates solely on a two-dimensional plane, ignoring vertical motion.
2. The wheels exhibit pure rolling with no relative sliding against the ground.
3. There is no friction between the joints connecting the vehicle body.
4. The user moves precisely according to the feedback provided by the cane.

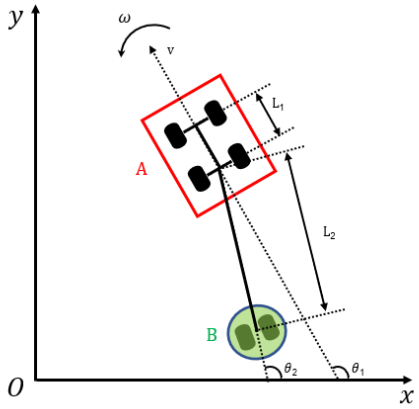


Figure 2. Kinematic Model of the Robotic Guide Dog.

Rectangle A represents the guide dog robot, while circle B represents the follower. L1 denotes the wheelbase, or the distance between the front and rear wheels of the robot, and L2 is the distance between the follower and the linkage point on the robot. Angles θ1 and θ2 represent the azimuth angles of the follower and the robot, respectively, indicating the orientation of each relative to the fixed coordinate system. The parameters φ, v, and w represent the steering angle, driving speed, and steering rate of the robot, respectively. Based on the geometric relationships, the positional relationship between the follower and the robot is given by:

$$\begin{aligned} x_2 &= x_1 - L_2 \cos\ \theta_2 \\ y_2 &= y_1 - L_2 \sin\ \theta_2 \end{aligned} \tag{1}$$

The kinematic function is:

$$\begin{aligned} \dot{x}_1 &= v \cos\ \theta_1 \\ \dot{y}_1 &= v \sin\ \theta_1 \\ \dot{\theta}_1 &= \omega \end{aligned} \tag{2}$$

DOI: 10.54254/2755-2721/95/20241751

$$\dot{\theta}_2 = \omega - \frac{v}{L}\sin(\theta_1 - \theta_2)$$

3. DRL-Based Path Planning

3.1. Reinforcement Learning and Markov Chains

Reinforcement Learning (RL) combined with Markov Chains utilizes the Markov property, which states that the future state (X_{n+1}) of a system depends only on the current state Xn, not on past states (, X_{n-1}, X_0)[11]. Markov Chains offer a probabilistic framework for modeling state transitions within a system, while RL seeks to identify the optimal strategy (or policy) to maximize the cumulative reward over time through interactions with the environment.

$$P\left(X_{n+1=j|X_n=t_n,X_{n-1}=t_{n-1},\ldots,X_0=t_0}\right) = P(X_{n+1} = j|X_n = i) = p_{ij} \tag{3}$$

And $\sum_{j=1}^{m} p_{ij} = 1$,for all i

In the above equation, the current state represents the surrounding environment of the robot. When the robot takes an action, the environment provides feedback (reward) and then transitions to the next state based on the probability p_{ij}. In this study, the state is defined by the data from the LiDAR sensor and the position of the follower, which is computed using the kinematic model described earlier. The action space consists of {left, idle, right, fast, slow}. A positive reward is given when the robot aligns with the direction of the target, while a negative reward is assigned if the robot deviates from the target's direction, with the penalty increasing as the angle between the robot's orientation and the target grows. Additionally, if the robot or follower collides with an obstacle, the robot receives a significant negative reward.

$$R(s,a) = a_1 \frac{v - v_{min}}{v_{max} - v_{min}} - a_2 robot\_collision - a_3 follower\_collision \tag{4}$$

where v, vmin, vmax are the current, minimum and maximum speed of the robot respectively, a_1, a_2 and a_3 are three coefficients.

Q-learning is a reinforcement learning algorithm that uses feedback from an agent's actions to determine optimal behavior. It involves updating the value of state-action pairs to reflect the expected cumulative reward. The choice of actions at each step is guided by the value function, which is updated using the Bellman equation to optimize decision-making.

$$Q(s,a) \leftarrow R + \gamma \max_{a'} Q(s',a') \tag{5}$$

3.2. Q-value Approximation Using Dueling DDQN

In Q-learning, robots understand their environment through the Markov decision process, where they store behavior values for all states in a table. However, traditional methods often lack theoretical performance guarantees, particularly for high-dimensional problems, which require substantial storage. Additionally, large nonlinear approximations of the behavior value function can lead to slower learning rates.

Recently, Deep Q-Networks (DQN) have emerged as a solution for approximating the behavior value function, significantly improving capabilities for handling multi-constraint and multi-objective optimization problems. DQN introduces two key innovations: experience replay buffers and target networks. These modifications help break temporal correlations and stabilize learning.

The comprehensive learning process of the DQN algorithm is illustrated in Figure 3.

DOI: 10.54254/2755-2721/95/20241751

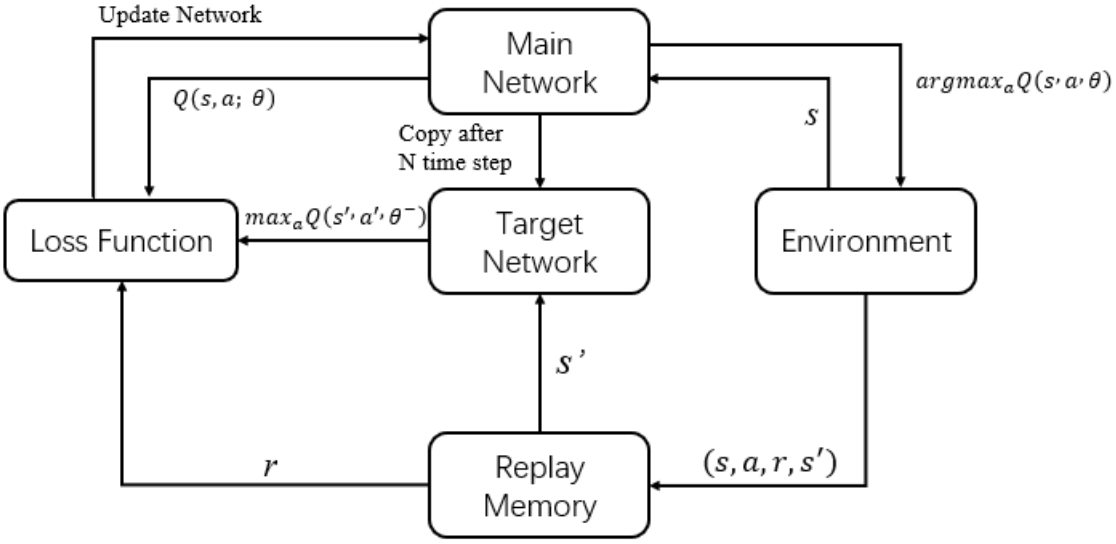


Figure 3. The comprehensive learning process of the DQN algorithm

In the process of updating Q-values, DQN employs a maximization operation, which can lead to an overestimation of Q-values, potentially affecting the convergence and performance of the algorithm. Double DQN addresses this overestimation issue by decoupling the action selection from the action evaluation, as outlined in [12]:

$$y_t^{DDQN} = r_{t+1} + \gamma Q(s_{t+1}, \text{argmax} Q(s_{t+1}, a'; \theta); \theta^- \quad (6)$$

where θ represents the parameters of the main network, and θ^- represents the parameters of the target network. The main network is responsible for selecting actions, while the target network is used to estimate Q-values.

To further enhance the performance of DQN, Dueling DQN [13] has been introduced. Dueling DQN improves learning efficiency by providing a more accurate estimation of the contribution of different actions for a given state. In Dueling DQN, the Q-value function Q(s,a) for each valid action a in a given state s is approximated by combining the estimated state value function V(s) and the advantage function A(s,a) . The key idea is to separate the learning of the value of being in a state (captured by V(s)) from the advantage of taking a particular action in that state (captured by A(s,a)).

Formally, the aggregating operation that combines V(s) and A(s, a) to approximate Q(s, a) is given by:

$$Q(s,a;w) \triangleq V(s;w^V) + A(s,a;w^A) - \text{mean}_{a \in A}\ A(s,a;w^A) \quad (7)$$

where:

s is the current state.

a is a valid action in state s.

w are the shared parameters between the two streams (up to the point where they split).

w^V are the parameters specific to the state value function stream.

w^A are the parameters specific to the advantage function stream.

A is the set of all valid actions in state s.

The term $\text{mean}_{a \in A}\ A(s,a;w^D)$ represents the mean advantage across all actions in state s . Subtracting this mean from the advantage of each action ensures that the estimated Q-values preserve their correct relative rankings, while allowing the network to independently represent the value of the state regardless of action advantages.

DOI: 10.54254/2755-2721/95/20241751

In this design, the state value function V(s) reflects the general value of being in state s, independent of the specific actions taken. Conversely, the advantage function A(s, a) captures the relative benefit of taking action a in state s compared to other actions. The aggregation of these two components allows the network to approximate the Q-value for each action more effectively, leading to improved learning efficiency and stability.

Additionally, Dueling DQN employs Huber loss instead of Mean Squared Error (MSE) to further enhance the stability and effectiveness of training.

3.3. Duelling-DQN Network Architecture

To accurately approximate the Q-function, we propose a Deep Double Dueling Q-Network (D3QN), as illustrated in Figure 4. This network processes a stacked sequence of the last N frames (where N = 4 , but adjustable) from our situation assessment model. Its output is a vector of Q-values for all valid actions.

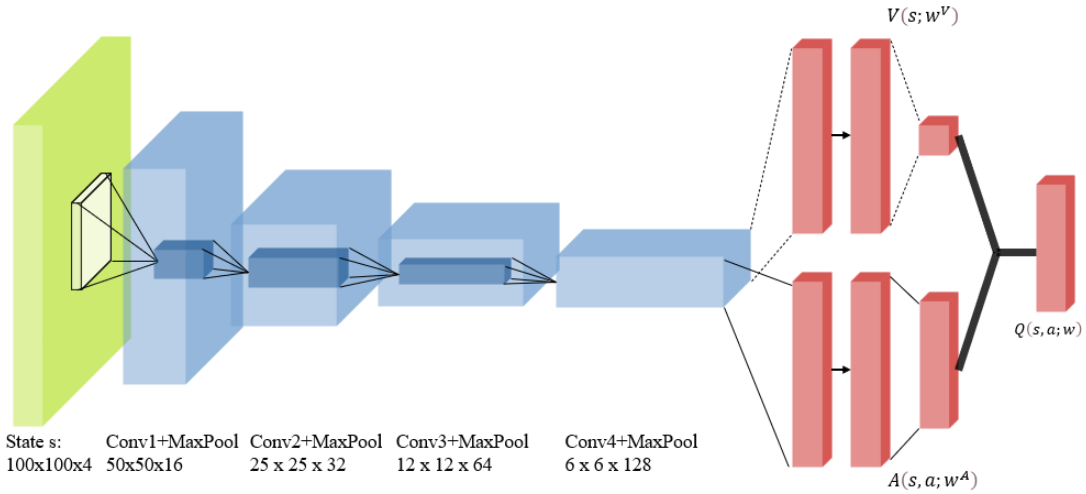


Figure 4. The Architecture of Dueling DQN

The network architecture includes two main components: a convolutional neural network (CNN) and a dueling neural network. The CNN section comprises four convolutional layers (Conv1 to Conv4) with progressively smaller filter sizes and strides to extract features from the input frames. Each convolutional layer is followed by a ReLU activation function to introduce non-linearity.

The dueling neural network is designed to separate the estimation of state value and action advantage. It consists of two parallel streams of fully-connected layers (FC1 and FC2) for each stream. Both streams feature an intermediate layer with 256 hidden units and ReLU activation. The value stream concludes with a single output node representing the state value, while the advantage stream produces eight outputs, corresponding to the number of valid actions.

This separation allows the network to independently assess the value of being in a state and the relative benefits of specific actions. Detailed parameters for the layers, including the number of filters, filter sizes, strides, and hidden units, are provided in Table 2. This integrated architecture combines convolutional feature extraction with the stability and efficiency of the dueling network structure to precisely estimate Q-values for improved decision-making.

Table 2. The details of our Network

Layer	Size(kernel)	Stride
CONV2	16@3 x 3	1
POOL1(MAX)	2 x 2	2

DOI: 10.54254/2755-2721/95/20241751

CONV2	32@3 x 3	1
POOL2(MAX)	2 x 2	2
CONV3	32@3 x 3	1
POOL3(MAX)	2 x 2	2
CONV4	64@3 x 3	1
POOL4(MAX)	2 x 2	2
FC1(V)	4608	1
FC2(V)	512	1
FC3(A)	4608	1
FC4(A)	512	1

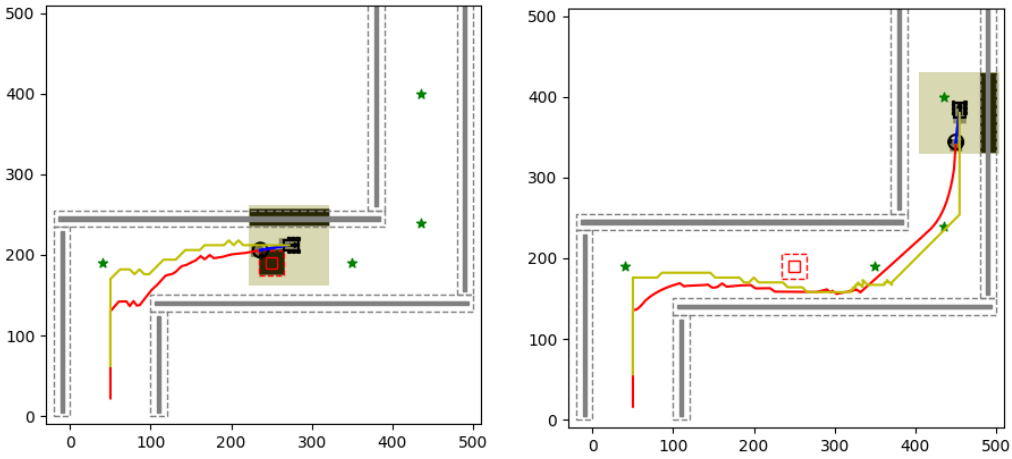
4. Experiments

In this section, we demonstrate that the agent trained using the D3QN algorithm effectively prevents both the follower and the robot from colliding with obstacles. The proposed robotic guide dogs were then tested in an indoor environment, as shown in Figure 5.

Figure 5. we test our method in the corridor. Because of narrow space, robot has to often change its direction.

The trajectory of robot is shown in Figure 6. We compare the trajectories of two agents. The first agent only considers its own interests, while the second agent cares both about itself and a follower. When the follower follows the first agent, it frequently collides with obstacles, whereas following the second agent results in fewer collisions.

DOI: 10.54254/2755-2721/95/20241751



(a) agent only cares itself (b) agent cares itself and followers

Figure 6. The trajectory of robot(yellow) and follower(red). Gray rectangles are obstacle zones.

The red rectangle is pedestrian. and the gray and red dotted lines are the area after the obstacle has expanded. The green stars are local target position, for providing movement direction to agent.

We choose Adam optimizer, because of its fast convergence speed and good adaptability. the other hyperparameter settings detailed is shown in Table 3.

Table 3. Hyperparameter settings

Parameter	Value
Batch Size	64
Buffer Size	10000
Epsilon	0.2
Gamma	0.99
Learning Rate	1e-4
Max Episode	10000

The software environment for this study was based on Python 3.10, open-source deep learning framework PyTorch, ROS2.0 Humble, and Ubuntu 22.04. The agent was trained on a laptop equipped with an NVIDIA GTX 4060 GPU.

Figure 7 displays the averaged reward during the training processes of the D3QN. These curves, representing the average performance over ten separate training runs, were obtained by training each algorithm across one million episodes under identical conditions with a map size of 100x100.

DOI: 10.54254/2755-2721/95/20241751

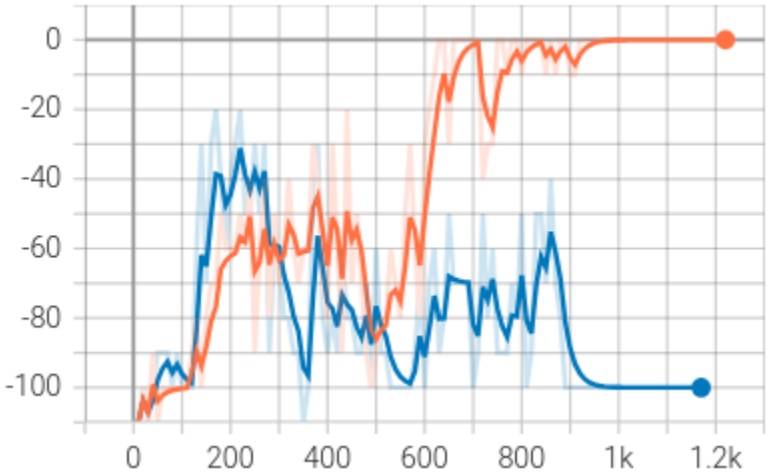


Figure 7. Reward agent gets every episode. The orange curve represents the reward received by the agent when trained to avoid collisions between both the follower and the robot with obstacles. In contrast, the blue curve illustrates the reward obtained by the agent when it only needs to consider its own collision avoidance.

Then, Figure 8 shows the curve of loss function. It shows the loss function value gradually decreases from around 2.5 to nearly 0.

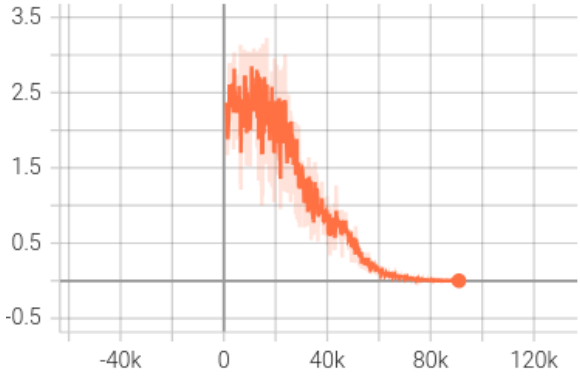


Figure 8. The curve of loss function

5. Conclusions

This article proposes a low-cost intelligent guide dog robot and develops a control method for it based on the Dueling DQN algorithm. This method incorporates the kinematic relationship between the robot and the follower and includes considerations for the follower's obstacle avoidance, not just the robot's, during the training of the control agent. This ensures enhanced safety by preventing collisions. The experimental indicate that if the navigation algorithm solely focuses on obstacle avoidance for the robot, the follower's trajectory is prone to frequent collisions with obstacles. And the experimental results show the proposed method effectively avoids the follower and robot collisions with obstacle. However, the current training environment is relatively simple. Future research should focus on refining the navigation algorithm to handle more complex environments and improve both navigation and obstacle avoidance capabilities.

DOI: 10.54254/2755-2721/95/20241751

6. References

[1] Galatas, Georgios & McMurrough, Christopher & Mariottini, Gian & Makedon, Fillia. (2011). eyeDog: An assistive-guide robot for the visually impaired. ACM International Conference Proceeding Series. 58. 10.1145/2141622.2141691.

[2] Shaojun Cai, Ashwin Ram, Zhengtai Gou, Mohd Alqama Wasim Shaikh, Yu-An Chen, Yingjia Wan, Kotaro Hara, Shengdong Zhao, and David Hsu. 2024. Navigating Real-World Challenges: A Quadruped Robot Guiding System for Visually Impaired People in Diverse Environments. In Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI '24). Association for Computing Machinery, New York, NY, USA, Article 44, 1–18. https://doi.org/10.1145/3613904.3642227

[3] Weiss, Martin, Simon Chamorro, Roger Girgis, Margaux Luck, Samira Ebrahimi Kahou, Joseph Paul Cohen, Derek Nowrouzezahrai, Doina Precup, Florian Golemo and Christopher Joseph Pal. "Navigation Agents for the Visually Impaired: A Sidewalk Simulator and Experiments." Conference on Robot Learning (2019).

[4] K. A. Hamed, V. R. Kamidi, W. -L. Ma, A. Leonessa and A. D. Ames, "Hierarchical and Safe Motion Control for Cooperative Locomotion of Robotic Guide Dogs and Humans: A Hybrid Systems Approach," in IEEE Robotics and Automation Letters, vol. 5, no. 1, pp. 56-63, Jan. 2020, doi: 10.1109/LRA.2019.2939719.

[5] V. Kulyukin, C. Gharpure, J. Nicholson, and G. Osborne, "Robot assisted wayfinding for the visually impaired in structured indoor environments," Autonomous Robots, vol. 21, no. 1, pp. 29–41, 2006.

[6] T. -K. Chuang et al., "Deep Trail-Following Robotic Guide Dog in Pedestrian Environments for People who are Blind and Visually Impaired - Learning from Virtual and Real Worlds," 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia, 2018, pp. 5849-5855, doi: 10.1109/ICRA.2018.8460994.

[7] J. Taery Kim, Wenhao Yu, Jie Tan, Greg Turk, and Sehoon Ha. 2023. How to Train Your Guide Dog: Wayfinding and Safe Navigation with Human-Robot Modeling. In Companion of the 2023 ACM/IEEE International Conference on Human-Robot Interaction (HRI '23). Association for Computing Machinery, New York, NY, USA, 221–225. https://doi.org/10.1145/3568294.3580076.

[8] Xiao, Anxing & Tong, Wenzhe & Yang, Lizhi & Zeng, Jun & Li, Zhongyu & Sreenath, Koushil. (2021). Robotic Guide Dog: Leading a Human with Leash-Guided Hybrid Physical Interaction. 11470-11476. 10.1109/ICRA48506.2021.9561786.

[9] Saegusa, S., Yasuda, Y., Uratani, Y. et al. Development of a guide-dog robot: human–robot interface considering walking conditions for a visually handicapped person. Microsyst Technol 17, 1169–1174 (2011). https://doi.org/10.1007/s00542-010-1219-1

[10] Y. Wei, X. Kou and M. C. Lee, "Smart rope and vision based guide-dog robot system for the visually impaired self-walking in urban system," 2013 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, Wollongong, NSW, Australia, 2013, pp. 698-703, doi: 10.1109/AIM.2013.6584174. keywords: {Robot sensing systems;Acoustics;Mobile robots;Force;Fuzzy logic;Visually impaired;Guide-dog robot;Hall-sensor joystick;ultrasonic sensor;Fuzzy logic control;Adaboosting;Template matching;Traffic lights;Zebra crossing},

[11] Thie, Paul R. "Markov decision processes." Comap, Incorporated, 1983.

[12] Van Hasselt, H., Guez, A., Silver, D.: Deep reinforcement learning with double Q-learning. In: Proceedings of AAAI Conference on Artificial Intelligence, pp. 2094–2100 (2015).

[13] Wang, Z., Schaul, T., Hessel, M., Van Hasselt, H., Lanctot, M., De Freitas, N.: Dueling network architectures for deep reinforcement learning. In: Proceedings of International Conference on Machine Learning (ICML), pp. 1995–2003 (2016).