

Path Following and Avoiding Obstacle for Mobile Robot Under Dynamic Environments Using Reinforcement Learning

Le Duc Hanh^{1,3}, Vo Duy Cong^{2,3,*}

¹ Department of Mechatronics, Faculty of Mechanical Engineering, Ho Chi Minh City University of Technology (HCMUT), 268 Ly Thuong Kiet Street, District 10, Ho Chi Minh city, Vietnam

² Industrial Maintenance Training Center, Ho Chi Minh City University of Technology (HCMUT), 268 Ly Thuong Kiet Street, District 10, Ho Chi Minh City, Vietnam

³ Vietnam National University Ho Chi Minh City, Linh Trung Ward, Thu Duc City, Ho Chi Minh City, Vietnam

Email: ¹ ldhanh@hcmut.edu.vn, ² congvd@hcmut.edu.vn

*Corresponding Author

Abstract—Obstacle avoidance for mobile robot to reach the desired target from a start location is one of the most interesting research topics. However, until now, few works discuss about working of mobile robot in the dynamic and continuously changing environment. So, this issue is still the research challenge for mobile robots. Traditional algorithm for obstacle avoidance in the dynamic, complex environment had many drawbacks. As known that Q-learning, the type of reinforcement learning, has been successfully applied in computer games. However, it is still rarely used in real world applications. This research presents an effectively method for real time dynamic obstacle avoidance based on Q-learning in the real world by using three-wheeled mobile robot. The position of obstacles including many static and dynamic obstacles and the mobile robot are recognized by fixed camera installed above the working space. The input for the robot is the 2D data from the camera. The output is an action for the robot (velocities, linear and angular parameters). Firstly, the simulation is performed for Q-learning algorithm then based on trained data, The Q-table value is implemented to the real mobile robot to perform the task in the real scene. The results are compared with intelligent control method for both static and dynamic obstacles cases. Through implement experiments, the results show that, after training in dynamic environments and testing in a new environment, the mobile robot is able to reach the target position successfully and have better performance comparing with fuzzy controller.

Keywords—Path Following; Avoiding Obstacle; Mobile Robot; Reinforcement Learning; Q-learning.

I. INTRODUCTION

In recent years, due to their agility, maneuverability, and ability to be deployed in many complex missions, mobile robot has attracted many researchers, particularly in respect of autonomous navigation in the warehouse or restricted area [1-10]. However, methods using fixed line have many drawbacks:

- After a period of use, line will be blurred and make it difficult for the robot to detect.
- It is also susceptible to the impact of the surrounding environment and complicated to change the route.
- It is not flexible when encountering dynamic obstacles and easily causes the robot to have very large errors

with the planned path. This poses a danger when the operating robot escapes from the safety zone in the factory or workshop

Nowadays, there is an expanding demand for complex applications where working environment has the participation of humans, moving robots or sudden obstacles appearance. At this time, humans or robots are considered movable obstacles that form the dynamic environment. Since then, the solution for the “Navigation” problem, which is based on camera application [11-16], has attracted much interest as it can surmount the limitations of tradition line detecting method and also provides the ability to optimize the path. Moreover, the working area is usually a warehouse or limited area, so the robot's ability to operate must be flexible and can run according to the predetermined trajectory and avoid moving obstacles that may appear without leaving from safe planning trajectory [17-20]. Therefore, there are currently numerous traditional researches and approaches such as: RRT* [21-22], A* [23-24], Visibility Graph, Fast Marching Tree, which mainly focused on autonomous path planning for mobile robot where the environment is static and mapped, and obstacle locations are assumed to be known in advance. Besides, [25-34] applied complicated algorithms such as Adaptive Genetic, Bacterial Evolutionary Algorithm, Predictive behavior or Partial swarm to avoid collisions. Those methods are time-consuming in building and updating of the dynamic environment map, these drawbacks make the low accuracy of the prediction.

Recently intelligent control algorithms are also applied to obstacle avoidance, fuzzy logic control method [35-41], network method [42-50], The obstacle avoidance method based on fuzzy logic control forms certain rules according to the prior knowledge. Fuzzy logic control approach demonstrates good robustness property, real-time performance and less dependence on the environment. But there exists the phenomenon of symmetry which cannot be determined. The obstacle avoidance method based on neural network designs controller according to the position of obstacles. But it needs a lot of time to collect data and train the network to find a path.



Traditional obstacle avoidance algorithms are invalid when the information of the obstacle is incomplete or completely unknown. The intelligent control needs data or experience to design [51-52]. Reinforcement learning (RL), unlike other artificial intelligence algorithms, is a learning method that does not require any rules [53-56]. RL is a machine learning method that regards the feedback of the environment as an input and adapts the environment. Q-learning is one of the most popular algorithms in the RL algorithm. The algorithm focuses on value-based reinforcement learning that is updated as the environment is explored by means of the Q-value function [57-60]. Recently, combining intelligent control and Q-learning are also applied [61-65]. However, these studies are only carried out on simulations, as well as experiments with simple static objects. Moreover, designing the controller for robot to follow the processed path (virtual) is not clearly discussed. As we know that Q-learning based on a learning experience of trial and error where the agent goes through numerous failures before actual success. it is very expensive as well as time-consuming. This means that the training is difficult to perform in a real environment and tends to be done mostly in simulation. From [61-65] it can be seen that Q-learning are trainable in virtual environments and afterward transferable to the real world in robot applications. To address these challenges, in this research the training environment for the RL agent in a virtual environment is developed. The virtual environment depicts the actual scenario and enables the user to collect a large number of reactions in various environments. The agent can interact with the environment through the actions and can also be trained with various user-defined rewards and goals. After that the learned table is transfer to the real mobile robot for doing the real experiments.

The contribution of this study is that:

- Unlike other traditional, artificial and intelligence algorithms, the obstacle avoidance methodology for moving obstacle in this paper does not use any prior dataset for training or experience for design the controller.
- The training data are transferred to robot and work real time in the real application.
- From the experiment the RL proved to have better performance than intelligent control algorithm in terms of total time and errors.

II. ROBOT MODELING

Fig. 1 show the parameters of two-wheel mobile robot, where $R(x_R, y_R)$ is the goal point, $G(x_G, y_G)$ is the control point of robot, d is radius of wheels, b is the distance between two wheels and e_1, e_2, e_3 are the errors between robot and goal point.

From Fig. 1, the error between the center and the goal point as shown in equations (1) and (2).

$$\begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} \cos\phi & \sin\phi & 0 \\ -\sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_R - x_C \\ y_R - y_C \\ \phi_R - \phi \end{bmatrix} \quad (1)$$

Derivative of the errors we have

$$\begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \end{bmatrix} = \begin{bmatrix} -1 & e_2 \\ 0 & -e_1 - d \\ 0 & -1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (2)$$

with two control parameters are velocity v and angular velocity ω , objective of the needed controller is to eliminate the errors $[e_1, e_2, e_3]^T = [0, 0, 0]^T$. The Lyapunov's stability method is used for the controller.

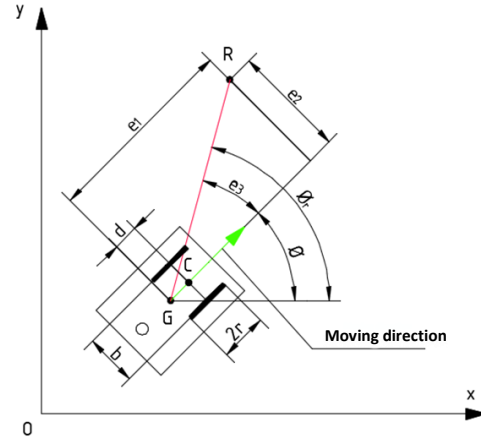


Fig. 1. Robot modelling

Theorem: consider a system which is described by a state equation $\dot{x} = f(x_1, \dots, x_n)$. If exist a positive-definite $V(x)$ with all state variables, so that its time derivative is a negative-definite function then that system is stable [66].

Choose a positive-definite Lyapunov function shown in (3) and value of derivative.

$$V = \frac{1}{2} e_1^2 + \frac{1}{2} e_2^2 + (1 - \cos e_3) \quad (3)$$

Derivative of V :

$$\dot{V} = e_1 \dot{e}_1 + e_2 \dot{e}_2 + \dot{e}_3 \sin e_3 = -e_1 v - \omega \sin e_3 - d \omega e_2$$

Choose (v, ω) so that \dot{V} is negative-definite:

$$\begin{cases} v = k_1 e_1 \\ \omega = k_2 \sin e_3 \end{cases} \quad \text{with } k_1, k_2 > 0$$

Substitute v and ω into (3), we have:

$$\dot{V} = -k_1 e_1^2 - k_2 \sin^2 e_3 - d \omega e_2 \leq 0$$

From Fig. 1 the velocity of each wheel is calculated in equation (4) and (5).

$$\omega_l = \frac{v_l}{r} = \frac{v - \omega \frac{b}{2}}{r} = \frac{v}{r} - \frac{\omega b}{2r} \quad (4)$$

$$\omega_r = \frac{v_r}{r} = \frac{v + \omega \frac{b}{2}}{r} = \frac{v}{r} + \frac{\omega b}{2r} \quad (5)$$

III. CONTROL DESIGN

A. Q-learning

Q-learning is a modelless reinforcement learning algorithm [67-68]. The goal of Q-learning is to learn the rules, the rules that tell the machine what action to take under what circumstances. The algorithm does not require a model of the

environment and can handle random changes without adjustment [69-71].

Specifically, in case the robot approaches an obstacle, the robot will compare the relative position of the obstacle and then take an action turn left, turn right, go straight. Every time the robot passes or hits an obstacle, it will receive the corresponding reward. This process repeats until the robot finds the actions to optimize the reward received, in other words, finds the best rule to avoid the obstacle.

The relative position of the obstacle relative to the robot will be divided into 8 angles (G_i with $i \in \{1, 8\}$) corresponding to 8 states. Each angle G_i has a magnitude of 45° , starting from the direction coincident with the current direction of the robot, in the positive trigonometric direction as shown in Fig. 2. The robot calculates the angle (the angle between the robot's current direction and the line connecting the control point to the center of the obstacle) and then considers what state it is in and then gives the corresponding action for that state

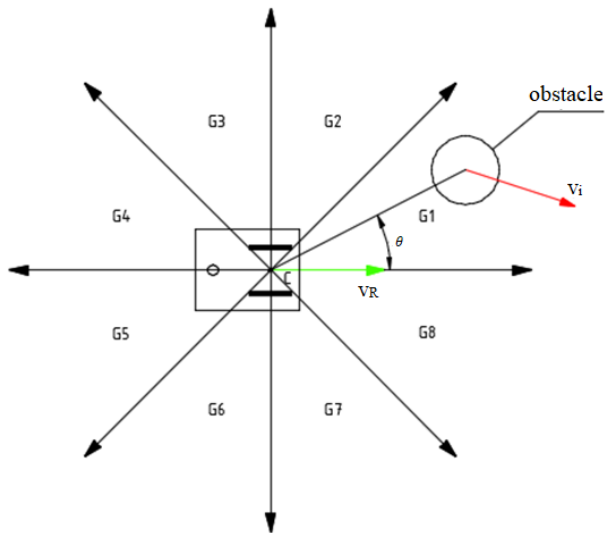


Fig. 2. The action states the robot

In the Q-Learning algorithm, the corresponding action for each state is calculated from the Q-table. The rows of the table include the states reached by the robot during the learning process, the columns are the actions that the robot performs in the respective states, each action is predefined v and ω for the action. Initially, the value of the cells in the table is set to 0, then it will be updated gradually during the learning process. The value in each cell is calculated by the formula in equation (6).

$$Q(s_t, a_t) = r(s_t, a_t) + \gamma \text{Max}(Q, s_{t+1}) \quad (6)$$

Where, s_t, a_t are state and action at time t respectively, $Q(s_t, a_t)$ is the value at state s_t and action a_t , $r(s_t, a_t)$ is the reward received when performing the action at in state s_t , $\text{Max}(Q, s_{t+1})$ is the largest value in the row corresponding to the state at the next time in the Q-table, γ is the attenuation factor, with a value less than 1 to ensure that the further away from the target, the smaller the value.

When the robot is in the state s_t , the robot will find the maximum value of the corresponding actions in that state and

perform the action, then update the value at that cell $Q(s_t, a_t)$. Continue until the learning process is over, the robot has learned the rules (through the Q-table) to be able to avoid obstacles in many different cases.

B. Fuzzy Control

Fuzzy Logic is a control algorithm that mimics the processing of ambiguous information and decision-making by humans [72-77]. Specifically, in case the robot approaches an obstacle, the input value includes not only greater or less than the safe distance, but also additional values such as slightly far, far, slightly close, close, between combined with predefined rules to get the corresponding output values. Number of inputs consist of angle θ which is the angle made by the robot's current direction and the line connecting the control point to the center of the obstacle and the variation of the angle θ ($\Delta\theta$). The number of outputs includes robot angular speed ω as shown in Fig. 3. Center average is used for Defuzzification.

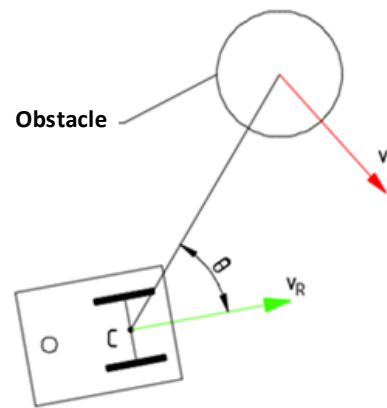


Fig. 3. Relative position of robot and obstacle

The angle θ is partitioned into three fuzzy sets with the range from -60 to 60 degree and the change rate of θ is partitioned into three fuzzy sets from -8 to 8 . The output ω have five fuzzy set from -20 to 20 rad/s. All the parameters are shown in Fig. 4 and Fig. 5.

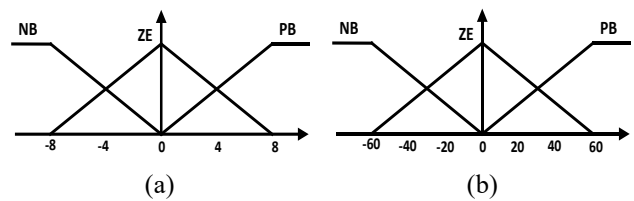


Fig. 4. The membership function (a) theta, (b) change rate of theta

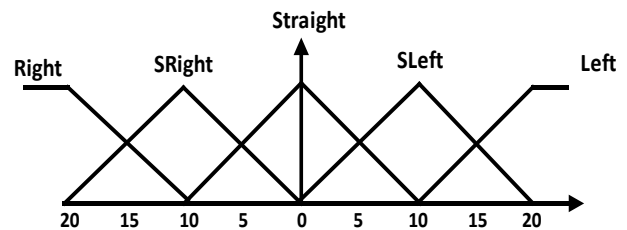


Fig. 5. The membership function of output ω

One important step in the fuzzy logic controller design is to make a rule table. In this work, three fuzzy subsets of θ

and three subsets $\Delta\theta$, there exist nine rules, and therefore, the rule base is represented in a 3x3 matrix, as shown in Table 1.

TABLE I. RULE FUZZY TABLE

	NB	ZE	PB
NB	Straight	SRight	Right
ZE	Sright	Straight	SLeft
PB	Left	SLeft	Straight

IV. SIMULATION RESULTS

For designing the Q-learning controller, the simulation has two Phase: Training and Testing. The simulation is to investigate the ability of mobile robot to accomplish the task without hit the obstacles in different environment. The Q learning and Fuzzy Q learning is applied in Training Phase. In Testing Phase, Mobile robot is placed in another environment and three algorithms discussed above are performed and compared.

Training Phase: Mobile robot is trained to reach the target in an environment with eight static circle (black color) and six dynamic obstacles (blue color) with 10cm diameter are used. The dynamic one is generated randomly and run different velocity and trajectory as shown in Fig. 6. There are six random start-points for the training to make sure that robot can operate in different environment. The velocity of robot is assumed constant as 0.5(m/s). By using the controller discussed in Section 3, the robot will keep tracking the shortest path (dot line) to reach the target while avoiding the obstacle. After twenty-two epochs, in different positions, robot can reach the target without hitting any obstacles. The result of the training performance and learning table is shown in Fig. 7.

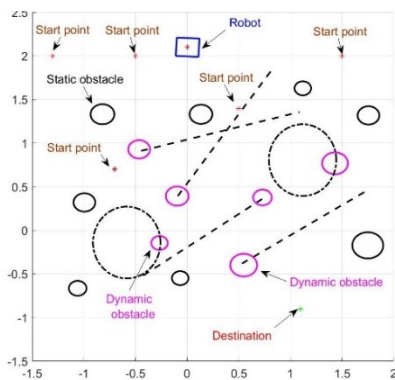


Fig. 6. Training environment for mobile robot

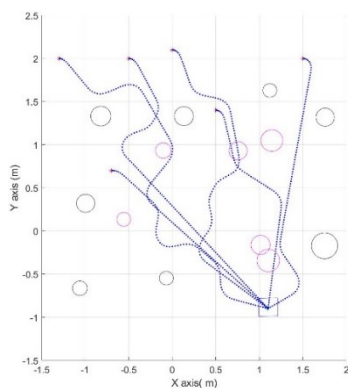


Fig. 7. Training performance and learning table result

Testing Phase: The robot reuses the map as shown in Fig. 6, but the position of the robot's starts are changed. This position change is used to create different operating environments for the robot. The robot is placed in the starting positions. Fuzzy and Q learning algorithms are performed at those locations, respectively. The performance of these algorithms are illustrated in Fig. 8 and Fig. 9. The average error between the shortest path the simulation path of each algorithm, the time average for each simulation is summarized as in table 2.

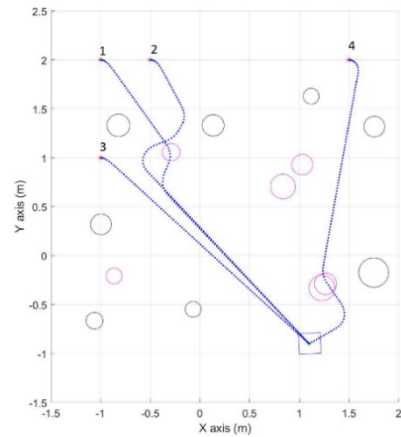


Fig. 8. Testing performance of Q learning

From Fig. 8 and Fig. 9, it can be seen that under the same condition, the robot can complete the task with both algorithms. However, with the Q learning algorithm, the robot has better performance when avoiding obstacles, the robot has less to change direction many times, which is very important when the robot carries objects in a warehouse or working area.

From the Table 2 and Table 3 it can be seen that at every position, the Q-learning algorithm have smaller total time and the absolute error between the shortest path planning (the line connecting the start and the goal point) and the real path is smaller. Q learning algorithm helps the robot to follow the planned path better, so that the robot does not get out of the safe zone around the planned path

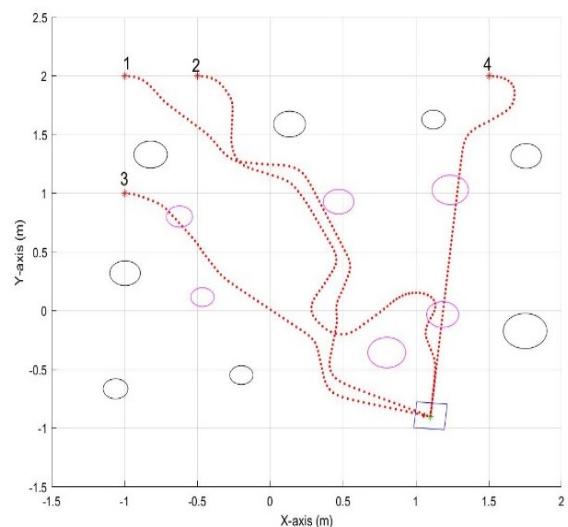


Fig. 9. Testing performance of Fuzzy control

TABLE II. PERFORMANCE TABLE OF FUZZY CONTROL

Position	Total time (s)	Average Absolute Error with reference line (m)
1	62	0.68
2	79	0.72
3	20	0.21
4	16	0.14
Average	44.25	0.44

TABLE III. PERFORMANCE TABLE OF Q-LEARNING

Position	Total time (s)	Average Absolute Error with reference line (m)
1	40	0.21
2	60	0.34
3	14	0.05
4	12	0.12
Average	31.5	0.18

V. EXPERIMENT

Robot is brought to another operating environment similar to the real experiment to simulate and compare the results with the experiment. The Q- learning table which obtained in the simulation is also reused in the real experiment. The robot attached with AR marker above will operate on flat terrain, camera fixed above working space with frame rate 30fps and fixed lighting conditions to create a map as shown in Fig. 10. The robot (marked with a red “R”) will move from the starting point to the destination point (marked in red), the robot's initial direction is 0o. Obstacles will be fixed-sized objects of the same height as the robot. Each obstacle is mounted on an AR marker (static obstacles are marked with red “SO”, moving obstacles are marked with red “MO”). The number of obstacles includes two static obstacles and one dynamic obstacle. The starting point is marked in black; the destination point is marked in green. Static obstacles are black circles; dynamic obstacles are purple circles. The overall controller for the mobile robot is shown in Fig. 11. Firstly the AR marker on robot will give the position of the robot and the error pose between the initial location and target pose $[e_1, e_2, e_3]^T$ are calculated. After that the velocity of the wheel robot is calculated for robot to move to the target position. on the way to the destination, mobile robot will use the Q-learning algorithm to avoid the obstacles. The working result of simulation and experiment are also shown in Fig. 12.



Fig. 10. Experiment apparatus

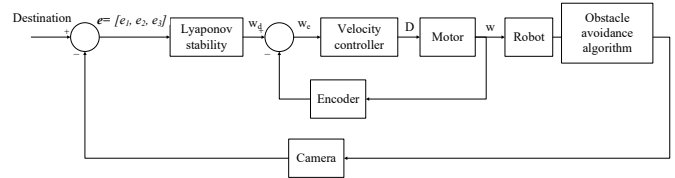


Fig. 11. The overall control flowchart

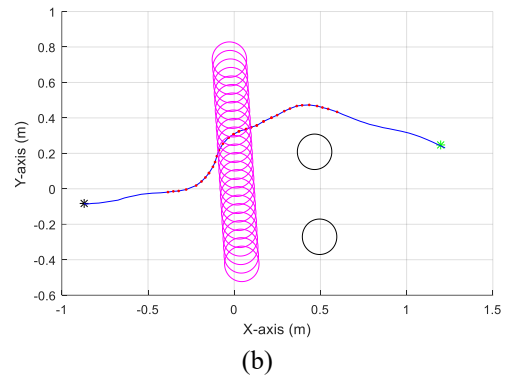
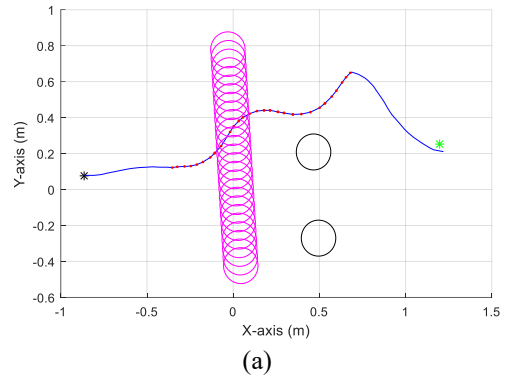
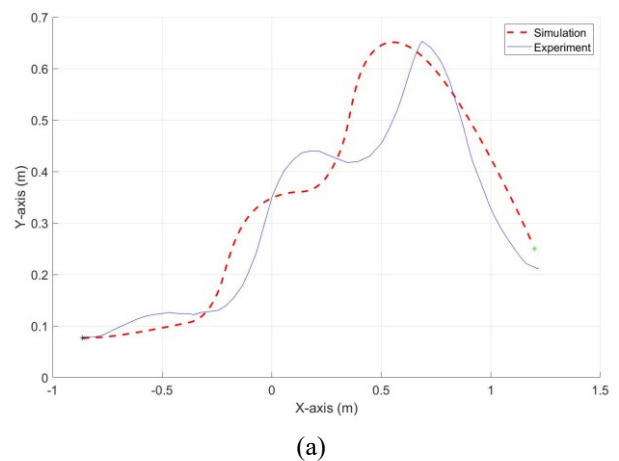
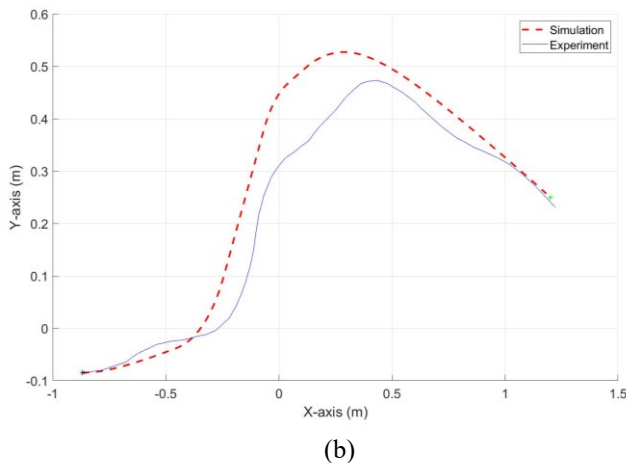


Fig. 12. The working result of simulation and real experiment, (a) fuzzy control, (b) Q-learning control.

From Fig. 12 it can be seen that, When the robot moves to the destination, the moving obstacle (pink circle) also moves to cut the mobile robot's direction, then the robot must use algorithms designed to avoid obstacles.

The performance of the controllers is showed in Fig. 13 and Table 4. The result shown in blue is the robot's trajectory. The red points are the position of the robot when performing the obstacle avoidance algorithm.





(b)

Fig. 13. Performance of controller, (a) Fuzzy controller, (b) Q-learning controller.

TABLE IV. PERFORMANCE TABLE COMPARISON

Position	Total time (s)	Average absolute error between simulation and experiment (m)
Fuzzy	8.5	0.15
Q-learning	7	0.14

From Fig. 12 it can be seen that, the response of the control algorithm in simulation and experiment is similar. Both simulation and experiment in the new environment, Q-learning algorithm always gives better results when the trajectory changes less direction.

From table 4, it can be seen that, the total time and error of Q learning is smaller. This happens because the moving trajectory of the Q-learning algorithm is smoother

VI. CONCLUSION

In this paper, the path following algorithm for mobile robot and obstacle avoidance using Lyapunov stability, and Q-learning has been implemented. The series of simulation and experiment are investigated to make sure that the obstacle avoidance algorithm can help robot to avoid obstacles without leaving the planned path. These algorithms can all be applied in factories or pre-planned areas with fixed cameras from above. Series of simulation and experiment show that the Q learning algorithm can simulate in a virtual environment before applying it in the real environment. By comparing the total time and error from Table 4, the performance for the obstacle avoidance algorithm using Q-learning controller is better than the fuzzy controller in terms of total time performing and errors with the planning path. This happens because the moving trajectory of the Q-learning algorithm is smoother

For future improvement, the error between the simulation and the real experiment can be reduced by implementing visual servoing control algorithm.

ACKNOWLEDGMENT

We acknowledge Ho Chi Mi City University of Technology (HCMUT), VNU-HCM for supporting this study.

REFERENCES

- [1] A. Mora, A. Prados, A. Mendez, R. Barber, and S. Garrido, "Sensor Fusion for Social Navigation on a Mobile Robot Based on Fast Marching Square and Gaussian Mixture Model," *Sensors*, vol. 22, no. 22, p. 8728, 2022.
- [2] R. M. Sousa, D. B. Aranibar, J. D. Amado, R.E. P. Escarcina, and R. M. P. Trindade, "A New Approach for Including Social Conventions into Social Robots Navigation by Using Polygonal Triangulation and Group Asymmetric Gaussian Functions," *Sensors*, vol. 22, no. 12, p. 4602, 2022.
- [3] L. C. Santos, A. S. Aguiar, F. N. Santos, A. Valente, and M. Petry, "Occupancy Grid and Topological Maps Extraction from Satellite Images for Path Planning in Agricultural Robots," *Robotics*, vol. 9, no. 4, p. 77, 2020.
- [4] Y. Gao, C. Bai, R. Fu, and Q. Quan, "A non-potential orthogonal vector field method for more efficient robot navigation and control," *Robotics and Autonomous Systems*, vol. 159, p. 104291, 2023.
- [5] M. Santilli, P. Mukherjee, R. K. Williams, and A. Gasparri, "Multirobot Field of View Control with Adaptive Decentralization," *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2131-2150, 2022.
- [6] Y. Zhao, T. Wang and W. Bi, "Consensus Protocol for Multiagent Systems with Undirected Topologies and Binary-Valued Communications," in *IEEE Transactions on Automatic Control*, vol. 64, no. 1, pp. 206-221, 2019.
- [7] D. Gadjov and L. Pavel, "A Passivity-Based Approach to Nash Equilibrium Seeking Over Networks," in *IEEE Transactions on Automatic Control*, vol. 64, no. 3, pp. 1077-1092, 2019.
- [8] Y. Hua, X. Dong, Q. Li, and Z. Ren, "Distributed Fault-Tolerant Time-Varying Formation Control for Second-Order Multi-Agent Systems with Actuator Failures and Directed Topologies," in *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 65, no. 6, pp. 774-778, 2018.
- [9] A. Carrio, J. Tordesillas, S. Vemprala, S. Saripalli, P. Campoy and J. P. How, "Onboard Detection and Localization of Drones Using Depth Maps," in *IEEE Access*, vol. 8, pp. 30480-30490, 2020.
- [10] H. Seo, G. Cho, S. -J. Kim, J. -H. Chun, and J. Choi, "Multievent Histogramming TDC With Pre-Post Weighted Histogramming Filter for CMOS LiDAR Sensors," in *IEEE Sensors Journal*, vol. 22, no. 23, pp. 22785-22798, 2022.
- [11] Z. Tahir, A. H. Qureshi, Y. Ayaz, and R. Nawaz, "Potentially guided bidirectionalized RRT* for fast optimal path planning in cluttered environments," *Robotics and Autonomous Systems*, vol. 108, pp. 13-27, 2018.
- [12] K. M. I. Khalilullah, S. Ota, T. Yasuda, and M. Jindai, "Road area detection method based on DBNN for robot navigation using single camera in outdoor environments," *Industrial Robot*, vol. 45, no. 2, pp. 275-286, 2018.
- [13] G. Lei, R. Yao, Y. Zhao, and Y. Zheng, "Detection and Modeling of Unstructured Roads in Forest Areas Based on Visual-2D Lidar Data Fusion," *Forests*, vol. 12, no. 7, p. 820, 2021.
- [14] F. Amorós, L. Payá, W. Mayol-Cuevas, L. M. Jiménez, and O. Reinoso, "Holistic Descriptors of Omnidirectional Color Images and Their Performance in Estimation of Position and Orientation," in *IEEE Access*, vol. 8, pp. 81822-81848, 2020.
- [15] A. W. L. Yao and H. C. Chen, "An Intelligent Color Image Recognition and Mobile Control System for Robotic Arm," *International Journal of Robotics and Control Systems*, vol. 2, no. 1, pp. 97-104, 2022.
- [16] I. Hassani, I. Ergui, and C. Rekiq, "Turning Point and Free Segments Strategies for Navigation of Wheeled Mobile Robot," *International Journal of Robotics and Control Systems*, vol. 2, no. 1, pp. 172-186, 2022.
- [17] G. Farid *et al.*, "Modified A-Star (A*) Approach to Plan the Motion of a Quadrotor UAV in Three-Dimensional Obstacle-Cluttered Environment," *Appl. Sci.*, vol. 12, no. 12, p. 5791, 2022.
- [18] T. -Y. Lin, K. -R. Wu, Y. -S. Chen, and Y. -S. Shen, "Collision-Free Motion Algorithms for Sensors Automated Deployment to Enable a Smart Environmental Sensing-Net," in *IEEE Transactions on Automation Science and Engineering*, vol. 19, no. 4, pp. 3853-3870, 2022.

- [19] K. Li, Q. Hu, and J. Liu, "Path planning of mobile robot based on improved multiobjective genetic algorithm," *Wireless Communications and Mobile Computing*, vol. 2021, pp. 1-12, 2021.
- [20] K. Hao, J. Zhao, B. Wang, Y. Liu, and C. Wang, "The application of an adaptive genetic algorithm based on collision detection in path planning of mobile robots," *Computational Intelligence and Neuroscience*, vol. 2021, pp. 1-20, 2021.
- [21] I. Noreen, A. Khan, K. Asghar, and Z. Habib, "A path-planning performance comparison of RRT*-AB with MEA* in a 2-dimensional environment," *Symmetry*, vol. 11, no. 7, pp. 945-960, 2019.
- [22] J. Wang, W. Chi, C. Li, C. Wang, and M. Q. -H. Meng, "Neural RRT*: Learning-Based Optimal Path Planning," in *IEEE Transactions on Automation Science and Engineering*, vol. 17, no. 4, pp. 1748-1758, 2020.
- [23] Y. Bai, G. Li, and N. Li, "Motion Planning and Tracking Control of Autonomous Vehicle Based on Improved A* Algorithm," *Journal of Advanced Transportation*, vol. 2022, pp.1-14, 2022.
- [24] H. Min, X. Xiong, P. Wang, and Y. Yu, "Autonomous driving path planning algorithm based on improved algorithm in unstructured environment," *Proceedings of the Institution of Mechanical Engineers - Part D: Journal of Automobile Engineering*, vol. 235, no. 2-3, pp. 513-526, 2021.
- [25] B. Fu *et al.*, "An improved A* algorithm for the industrial robot path planning with high success rate and short length," *Robotics and Autonomous Systems*, vol. 106, pp. 26-37, 2018.
- [26] N. B. A. Latip, R. Omar, and S. K. Debnath, "Optimal path planning using equilateral spaces-oriented visibility graph method," *Intl. J. Electr. Comput. Eng.*, vol. 7, no. 6, pp. 3046-3051, 2017.
- [27] L. Janson, E. Schmerling, A. Clark, and M. Pavone, "Fast marching tree: A fast marching sampling-based method for optimal motion planning in many dimensions," *The International journal of robotics research*, vol. 34, no. 7, pp. 883-921, 2015.
- [28] O. Montiel, U. Orozco-Rosas, and R. Sepveda, "Path planning for mobile robots using bacterial potential field for avoiding static and dynamic obstacles," *Expert Syst. Appl.*, vol. 42, no.12, pp. 5177-5191, 2015.
- [29] A.H. Karami and M. Hasanzadeh, "An adaptive genetic algorithm for robot motion planning in 2D complex environments," *Computers & Electrical Engineering*, vol. 43, pp. 317-329, 2015.
- [30] F. Kamil, T.S. Hong, W. Khaksar, M.Y. Moghrabiah, N. Zulkifli, and S.A. Ahmad, "New robot navigation algorithm for arbitrary unknown dynamic environments based on future prediction and priority behavior," *Expert Syst. Appl.*, vol. 86, pp. 274-291, 2017.
- [31] M. Alajlan, I. Chaari, A. Koubaa, H. Bennaceur, A. Ammar, and H. Youssef, "Global robot path planning using GA for large grid maps: Modelling performance and experimentation," *International Journal of Robotics and Automation*, vol. 31, no. 6, pp. 484-495, 2016.
- [32] Y. Zhao, X. Liu, G. Wang, S. Wu, and S. Han, "Dynamic Resource Reservation Based Collision and Deadlock Prevention for Multi-AGVs," *IEEE Access*, vol. 8, pp. 82120-82130, 2020.
- [33] L. Yue and H. Fan, "Dynamic Scheduling and Path Planning of Automated Guided Vehicles in Automatic Container Terminal," in *IEEE/CAA Journal of Automatica Sinica*, vol. 9, no. 11, pp. 2005-2019, 2022.
- [34] H. Xiao, X. Wu, D. Qin, and J. Zhai, "A Collision and Deadlock Prevention Method with Traffic Sequence Optimization Strategy for UGN-Based AGVs," in *IEEE Access*, vol. 8, pp. 209452-209470, 2020.
- [35] M. Faisal, R. Hedjar, M. Al-Sulaiman, and K. Al-Mutib, "Fuzzy Logic Navigation and Obstacle Avoidance by a Mobile Robot in an Unknown Dynamic Environment," *International Journal of Advanced Robotic Systems*, vol. 10, no. 1, 2013.
- [36] A. Shitsukane, W. Cheruiyot, C. Otieno and M. Mvurya, "Fuzzy Logic Sensor Fusion for Obstacle Avoidance Mobile Robot," *2018 IST-Africa Week Conference (IST-Africa)*, pp. 1-8, 2018.
- [37] S. -Y. Chiang, "Vision-based obstacle avoidance system with fuzzy logic for humanoid robots," *The Knowledge Engineering Review*, vol. 32, no. E9, 2017.
- [38] S. Ayub, N. Singh, M. Z. Hussain, M. Ashraf, D. K. Singh, and A. Haldorai, "Hybrid approach to implement multi-robotic navigation system using neural network, fuzzy logic, and bio-inspired optimization methodologies," *Computational Intelligence*, pp. 1- 15, 2022.
- [39] K. Farah and M. Y. Mohammed, "Multilayer Decision-Based Fuzzy Logic Model to Navigate Mobile Robot in Unknown Dynamic Environments", *Fuzzy Information and Engineering*, vol. 14, no. 1, pp. 51-73, 2021.
- [40] H. Muhammad *et al.*, "Robust mobile robot navigation in cluttered environments based on hybrid adaptive neuro-fuzzy inference and sensor fusion," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 10, pp. 9060-9070, 2022.
- [41] F. Á. Szili, J. Botzheim, and B. Nagy, "Bacterial Evolutionary Algorithm-Trained Interpolative Fuzzy System for Mobile Robot Navigation", *Electronics*, vol. 11, no. 11, p. 1734, 2022.
- [42] S. Zubair, S. ABU, R. Ruzairi, A. Andi, and H. Mohd, "Non-Verbal Human-Robot Interaction Using Neural Network for The Application of Service Robot," *IJUM Engineering Journal*, vol. 24, no. 1, pp. 301-318, 2023.
- [43] U.A. Syed, F. Kunwar, and M. Iqbal, "Guided autowave pulse coupled neural network (GAPCNN) based real time path planning and an obstacle avoidance scheme for mobile robots," *Robotics and Autonomous Systems*, vol. 62, no. 4, pp. 474-486, 2014.
- [44] K. -H. Chi and M. -F. R. Lee, "Obstacle avoidance in mobile robot using Neural Network," *2011 International Conference on Consumer Electronics, Communications and Networks (CECNet)*, pp. 5082-5085, 2011.
- [45] A. Medina-Santiago, J. L. Camas-Anzueto, J. A. Vazquez-Feijoo, H. R. Hernández-de León, R. Mota-Grajales, "Neural Control System in Obstacle Avoidance in Mobile Robots Using Ultrasonic Sensors," *Journal of Applied Research and Technology*, vol. 12, no. 1, pp. 104-110, 2014.
- [46] A. N. A. Rafai, N. Adzhar, and N. I. Jaini, "A Review on Path Planning and Obstacle Avoidance Algorithms for Autonomous Mobile Robots," *Journal of Robotics*, vol. 2022, pp. 1-14, 2022.
- [47] F. Qu, W. Yu, K. Xiao, C. Liu, and W. Liu, "Trajectory Generation and Optimization Using the Mutual Learning and Adaptive Ant Colony Algorithm in Uneven Environments", *Applied Sciences*, vol. 12, no. 9, p. 4629, 2022. Doi: 10.3390/app12094629.
- [48] A. Muhammad, M. A. H. Ali, and I. H. Shanono, "Path Planning Methods for Mobile Robots: A systematic and Bibliometric Review", *ELEKTRIKA- Journal of Electrical Engineering*, vol. 19, no. 3, pp. 14-34, 2020.
- [49] D. Zhu, C. Tian, B. Sun, and C. Luo, "Complete Coverage Path Planning of Autonomous Underwater Vehicle Based on GBNN Algorithm," *Journal of Intelligent & Robotic Systems*, vol. 94, no. 237-249, 2019.
- [50] C. Sun, W. He, and J. Hong, "Neural Network Control of a Flexible Robotic Manipulator Using the Lumped Spring-Mass Model," in *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 8, pp. 1863-1874, 2017.
- [51] A. Kumar, P. B. Kumar, and D. R. Parhi, "Intelligent Navigation of Humanoids in Cluttered Environments Using Regression Analysis and Genetic Algorithm," *Arabian Journal for Science and Engineering*, vol. 43, pp. 7655-7678, 2018.
- [52] A. K. Abbas, Y. A. Mashhadany, M. J. Hameed, and S. Algburi, "Review of Intelligent Control Systems with Robotics," *Indonesian Journal of Electrical Engineering and Informatics (IJEEI)*, vol. 10, no. 4, pp. 734-753, 2022.
- [53] M. Naeem, S. T. H. Rizvi, and A. Coronato, "A Gentle Introduction to Reinforcement Learning and its Application in Different Fields," in *IEEE Access*, vol. 8, pp. 209320-209344, 2020.
- [54] M. Naeem, A. Coronato, Z. Ullah, S. Bashir, and G. Paragliola, "Optimal User Scheduling in Multi Antenna System Using Multi Agent Reinforcement Learning," *Sensors*, vol. 22, no. 21, p. 8278, 2022.
- [55] W. Kumwilaisak, S. Phikulngoeng, J. Piriyyataravet, N. Thatphithakkul, and C. Hansakunbuntheung, "Adaptive Call Center Workforce Management with Deep Neural Network and Reinforcement Learning," in *IEEE Access*, vol. 10, pp. 35712-35724, 2022.
- [56] A. Alwarafy, M. Abdallah, B. S. Çiftler, A. Al-Fuqaha, and M. Hamdi, "The Frontiers of Deep Reinforcement Learning for Resource Management in Future Wireless HetNets: Techniques, Challenges, and

- Research Directions,” in *IEEE Open Journal of the Communications Society*, vol. 3, pp. 322-365, 2022.
- [57] B. Jang, M. Kim, G. Harerimana, and J. W. Kim, “Q-Learning Algorithms: A Comprehensive Classification and Applications,” in *IEEE Access*, vol. 7, pp. 133653-133667, 2019.
- [58] N. Sutisna, A. M. R. Ilmy, I. Syafalni, R. Mulyawan, and T. Adiono, “FARANE-Q: Fast Parallel and Pipeline Q-Learning Accelerator for Configurable Reinforcement Learning SoC,” in *IEEE Access*, vol. 11, pp. 144-161, 2023.
- [59] N. Sutisna, Z. N. Arifuzzaki, I. Syafalni, R. Mulyawan, and T. Adiono, “Architecture Design of Q-Learning Accelerator for Intelligent Traffic Control System,” *2022 International Symposium on Electronics and Smart Devices (ISESD)*, pp. 1-6, 2022.
- [60] A. Chandrakar and P. Paliwal, “An Intelligent Mechanism for Utility and Active Customers in Demand Response Using Single and Double Q Learning Approach,” *Smart Energy and Advancement in Power Technologies*, vol. 926, pp. 397-413, 2023.
- [61] J. Raajan, P.V. Srihari, J. P. Satya, B. Bhikkaji, and R. Pasumarthy, “Real Time Path Planning of Robot using Deep Reinforcement Learning,” *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 15602-15607, 2020.
- [62] R. Cimurs, J. H. Lee, and I. H. Suh, “Goal-Oriented Obstacle Avoidance with Deep Reinforcement Learning in Continuous Action Space,” *Electronics*, vol. 9, no. 3, p. 411, 2020.
- [63] L. Huang, H. Qu, M. Fu, and W. Deng, “Reinforcement Learning for Mobile Robot Obstacle Avoidance Under Dynamic Environments,” *PRICAI 2018: Trends in Artificial Intelligence*, vol. 11012, 2018.
- [64] A. D. Pambudi, T. Agustinah, and R. Effendi, “Reinforcement Point and Fuzzy Input Design of Fuzzy Q-Learning for Mobile Robot Navigation System,” *2019 International Conference of Artificial Intelligence and Information Technology (ICAIIIT)*, pp. 186-191, 2019.
- [65] S. Wen, X. Hu, Z. Li, H.K. Lam, F. Sun, and B. Fang, “NAO robot obstacle avoidance based on fuzzy Q-learning,” *Industrial Robot*, vol. 47, no. 6, pp. 801-811, 2020.
- [66] D. Lj. debelicković, V. B. Bajić, T. N. Erić, and S. A. Milinković, “A Lyapunov analysis of stability robustness for discrete linear descriptor systems,” *IMA Journal of Mathematical Control and Information*, vol. 15, no. 1, pp. 53-62, 1998.
- [67] J. Chang and G. Kwon, “Analysis of Hyper-parameters in Solving Sokoban using Q-learning,” *The Journal of Korean Institute of Information Technology*, vol. 20, no. 11, pp. 67-72, 2022.
- [68] S. Suratkar *et al.*, “An adaptive honeypot using Q-Learning with severity analyzer,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 13, no. 10, pp. 4865-4876, 2022.
- [69] K. Cheena, T. Amgoth, and G. Shankar, “Deep Q-probabilistic algorithm based rock hyraxes swarm optimization for channel allocation in CRSN smart grids,” *Wireless Networks*, vol. 28, no. 6, pp. 2553-2565, 2022.
- [70] A. M. Rahmani *et al.*, “An energy-aware and Q-learning-based area coverage for oil pipeline monitoring systems using sensors and Internet of Things,” *Scientific Reports*, vol. 12, no. 1, 2022.
- [71] J. Lansky *et al.*, “A Q-learning-based routing scheme for smart air quality monitoring system using flying ad hoc networks,” *Scientific Reports*, vol. 12, no. 1, 2022.
- [72] C. M. Horváth, J. Botzheim, T. Thomessen, and P. Korondi, “Bead geometry modeling on uneven base metal surface by fuzzy systems for multi-pass welding,” *Expert Systems with Applications*, vol. 186, pp. 115356, 2021.
- [73] M. Al-Mallah, M. Ali, and M. Al-Khawaldeh, “Obstacles Avoidance for Mobile Robot Using Type-2 Fuzzy Logic Controller,” *Robotics*, vol. 11, no. 6, p. 130, 2022.
- [74] M. Khairudin, R. Refalda, S. Yatmono, H. S. Pramono, A. K. Triatmaja, and A. Shah, “The Mobile Robot Control in Obstacle Avoidance Using Fuzzy Logic Controller,” *Indones. J. Sci. Technol.*, vol. 5, no. 3, pp. 334-353, 2020.
- [75] S. Ayub, N. Singh, M. Z. Hussain, M. Ashraf, D. K. Singh, and A. Haldorai, “Hybrid approach to implement multi-robotic navigation system using neural network, fuzzy logic, and bio-inspired optimization methodologies,” *Computational Intelligence*, pp. 1-15, 2022.
- [76] N. Zagradjanin, A. Rodic, D. Pamucar, and B. Pavkovic, “Cloud-Based Multi-Robot Path Planning in Complex and Crowded Environment Using Fuzzy Logic and Online Learning,” *Information Technology and Control*, vol. 50, no. 2, pp. 357-374, 2021.
- [77] V. Sangeetha *et al.*, “A Fuzzy Gain-Based Dynamic Ant Colony Optimization for Path Planning in Dynamic Environments,” *Symmetry*, vol. 13, no. 2, p. 280, 2021.