Vietnam Journal of Science and Technology ***62*** *(2) (2024) 374-386*

doi:10.15625/2525-2518/18066

Design of neural network-PID controller for trajectory tracking of differential drive mobile robot

Trinh Thi Khanh Ly[1]**, Nguyen Hong Thai**[2, *]**, Luu Thanh Phong**[2]

[1]*Faculty of Automation Technology, Electric Power University (EPU), 235 Hoang Quoc Viet,*
Bac Tu Liem, Ha Noi, Viet Nam

[2]*Department of Mechatronics, Hanoi University of Science and Technology (HUST),*
No. 1 Dai Co Viet, Hai Ba Trung, Ha Noi, Viet Nam

[*]Emails: *thai.nguyenhong@hust.edu.vn*

Received: 7 February 2023; Accepted for publication: 24 February 2024

Abstract. This paper proposes the design of a neural network controller based on a sample controller for controlling the trajectory-tracking motion of a differential drive mobile robot (DDMR). Firstly, the trajectory tracking model for DDMR is established based on position error. Next, a perceptron neural network is designed with three hidden layers to control the trajectory tracking of DDMR. The backpropagation algorithm is used to train the neural network with training data obtained from the PID controller with time-varying parameters. The authors have developed this approach and experimentally verified it with minor tracking errors. The neural network's weight matrix (**W**) and bias vector (**b**) are updated in real-time, providing an advantage over other methods. The effectiveness of the proposed controller is demonstrated by the DDMR's NURBS trajectory tracking error, which does not exceed 2.17 cm, and the DDMR's motion error, with linear and angular velocities not exceeding 0.004 m/s and 0.0007 rad/s, respectively. The proposed controller can supplement traditional controllers in controlling the trajectory of autonomous mobile robots, thereby improving the ability to generate local trajectories to avoid dynamic obstacles by the neural network.

Keywords: differential drive mobile robot, trajectory tracking control, neural network, train the neural network, NURBS trajectory.

Classification numbers: 5.2.1, 5.3.3, 5.3.5.

1. INTRODUCTION

In recent years, wheeled mobile robots have found a wide range of applications in various fields, such as borehole drilling robots in tunnelling construction [1], supermarkets [2], hospital transportation [3], autonomous guided vehicles in smart factories [4 - 6], and more. Among wheeled mobile robots, differential drive mobile robots (DDMR) are popular due to their simple structure and easy control. The continual improvement of automation and information processing technology has led to the development of new control systems to increase the autonomy and operability of autonomous mobile robots. The primary focus is on resolving trajectory-tracking issues and enhancing tracking accuracy.

Hitherto, two main challenges for researchers have been the optimal path planning and improvement of controllers to ensure high accuracy in the movement of autonomous mobile robots. Numerous methods for motion planning and robot control have been suggested, such as improved PID controllers [7 - 9], improved linear state feedback controllers [10], a fuzzy controller, and a neural network (NN) controller [11, 12]. Xuehao Sun *et al*. [13] proposed a motion plan with a smooth curve, enabling the robot to move at optimal speed and avoid unforeseen dynamic obstacles. Hsiu-Ming Wu *et al*. [14] developed a nonlinear trajectory tracking controller that asymptotically reduces the position error of the robot to zero, with the stability of the system verification by the Lyapunov stability method. Dino Zivojevic *et al*. [15] applied a random tree algorithm to plan the path of mobile robots with differential constraints using the Dubin method. The method addresses the challenge of finding a feasible path between a starting and a destination point in environments with static obstacles. Talebi Abatari *et al*. [16] employed a PID-Fuzzy controller to replace the standard PID controller in tracking the robot's trajectory. The results indicated that the proposed controller outperforms the standard controller with a faster convergence rate for arbitrary initial states. Shang *et al.* [17] proposed to design a state feedback controller with a time-varying parameter for the mobile robots to follow the desired trajectory. HaoWang *et al*. [18] employed a Fuzzy-NN for the path planning of a mobile robot, resulting in an optimal path from the starting point to the destination point. The evaluation analysis above shows that modern controllers have been proven to be superior to both classic and improved traditional controllers or Na *et al*. [19] utilized a combination of neural networks, fuzzy logic, and reinforcement learning to achieve path planning and optimize path length. Nevertheless, modern controllers also have a disadvantage because a large number of calculations leads to high requirements for the hardware structure of the robot, which increases the cost of the system.

In this work, a NN-PID controller is proposed to control the trajectory tracking of DDMR. The proposed controller combines the strengths of both a NN and a PID controller to advance accurate and stable trajectory tracking. The input/output sample data set used to train the neural network is determined from the signals of the sample controller or manual control signals during the system setup. The structure of the paper includes the following contents: first, the trajectory tracking model of the DDMR is established. Second, the Perceptron Neural Network structure is designed using a trial and error method. Then, the network is trained using the back-propagation algorithm and a sample data set consisting of the input ($\mathbf{e} = \mathbf{q}_d - \mathbf{q}_f$) and the output (the signal $\mathbf{u}_m$ of the sample controller). The sample controller used in this research is an improved PID controller with time-varying parameters developed in the literature [7, 8]. The paper concludes with the presentation of simulation results of DDMR tracking along a NURBS curve and an evaluation and discussion to demonstrate the effectiveness of the proposed controller.

2. TRAJECTORY TRACKING MODEL FOR DDMR

2.1. Kinematic model of DDMR

Assuming that the moving surface is flat, the scenario considered is a non-slip rolling motion of DDMR following a path trajectory ξ in a global coordinate system $\vartheta_f \{O_f x_f y_f\}$ attached to the moving surface, as illustrated in Figure 1. With the above definitions, according to [6], the relationship between the speed of the DDMR and the angular velocity of the two differential driving wheels is given by:

$$\begin{cases} V_G = \dfrac{V_2 + V_1}{2} = r\dfrac{(\omega_2 + \omega_1)}{2} \\ \dot{\theta} = \dfrac{V_2 - V_1}{L} = r\dfrac{(\omega_2 - \omega_1)}{L} \end{cases} \tag{1}$$

wherein *r* and *L* represent the wheel radius and the distance between the two driving wheels; V_G and $\dot{\theta} = \Omega$ are the velocities and angular velocity of the DDMR; $V_1 = \omega_1 r$ and $V_2 = \omega_2 r$ denote the angular velocity of the two driving wheels.

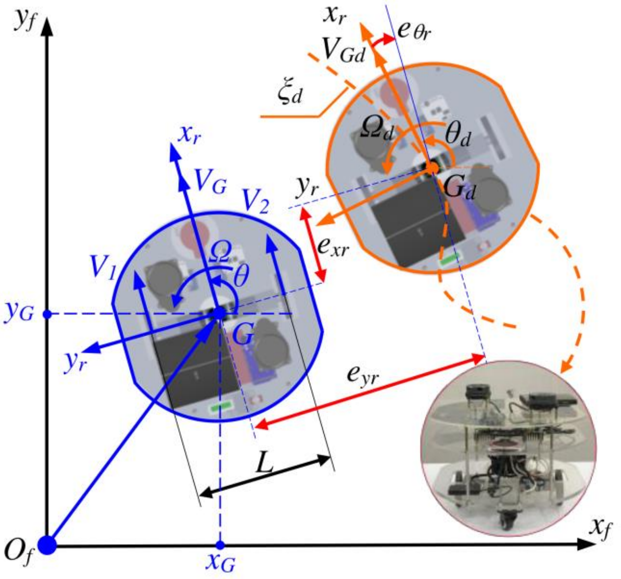

Figure 1. Illustration of the trajectory tracking motion of DDMR.

Thus, the kinematic model of DDMR in ϑ_f is expressed by:

$$\dot{\mathbf{q}}_f = \begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{\theta}(t) \end{bmatrix} = \begin{bmatrix} \cos\theta(t) & 0 \\ \sin\theta(t) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} V_G(t) \\ \dot{\theta}(t) \end{bmatrix} = \begin{bmatrix} \cos\theta(t) & 0 \\ \sin\theta(t) & 0 \\ 0 & 1 \end{bmatrix} \dot{\mathbf{q}}_r \tag{2}$$

here, $\mathbf{q}_r = \begin{bmatrix} \dot{x}_r & \dot{y}_r & \dot{\theta} \end{bmatrix}^T$ in the local coordinate system $\vartheta_r\{Gx_r y_r\}$, ϑ_r is attached to the geometric center G of the DDMR.

2.2. Trajectory tracking model of DDMR

Set **e** is the position error vector between the desired trajectory *{ξ}* and the current trajectory of motion in the global coordinate system ϑ_f. Thus, we have:

$$\mathbf{e} = \mathbf{q}_d - \mathbf{q}_f = \begin{bmatrix} e_x & e_y & e_\theta \end{bmatrix}^T \tag{3}$$

The desired trajectory *{ξ}* of the DDMR in the global coordinate system ϑ_f is represented by $\mathbf{q}_d = [x_d(t) \quad y_d(t) \quad \theta_d(t)]^T$.

Thus, in the local coordinate ϑ_r , the vector $\mathbf{e}_r$ is expressed by:

$$\mathbf{e}_r = \begin{bmatrix} e_{xr} & e_{yr} & e_{\theta r} \end{bmatrix}^T = \mathbf{R}^T \mathbf{e} \tag{4}$$

with

$$\mathbf{R} = \begin{bmatrix} \cos\theta(t) & -\sin\theta(t) & 0 \\ \sin\theta(t) & \cos\theta(t) & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

By taking the derivative of equation (4) and combining it with equations (2, 3), the motion trajectory tracking model of the DDMR is obtained as follows:

$$\dot{\mathbf{e}}_r = \begin{bmatrix} \dot{e}_{xr} \\ \dot{e}_{yr} \\ \dot{e}_{\theta r} \end{bmatrix} = \begin{bmatrix} V_d(t)\cos e_{\theta r} - V_f(t) + \dot{\theta}(t) e_{yr} \\ V_d(t)\sin e_{\theta r} - \dot{\theta}(t) e_{xr} \\ \dot{\theta}_d(t) - \dot{\theta}(t) \end{bmatrix}. \tag{5}$$

Rewrite equation (5) as a matrix:

$$\dot{\mathbf{e}}_r = \begin{bmatrix} 0 & \dot{\theta}(t) & 0 \\ -\dot{\theta}(t) & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} e_{xr} \\ e_{yr} \\ e_{\theta r} \end{bmatrix} + \begin{bmatrix} \cos e_{\theta r} & 0 \\ \sin e_{\theta r} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} V_d(t) \\ \dot{\theta}_d(t) \end{bmatrix} + \begin{bmatrix} -1 & 0 \\ 0 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} V_G(t) \\ \dot{\theta}(t) \end{bmatrix}. \tag{6}$$

3. DESIGN OF NN CONTROLLER FOR TRAJECTORY TRACKING OF DDMR

The structure of a NN-PID controller designed to control trajectory tracking for DDMR is shown in Figure 2.

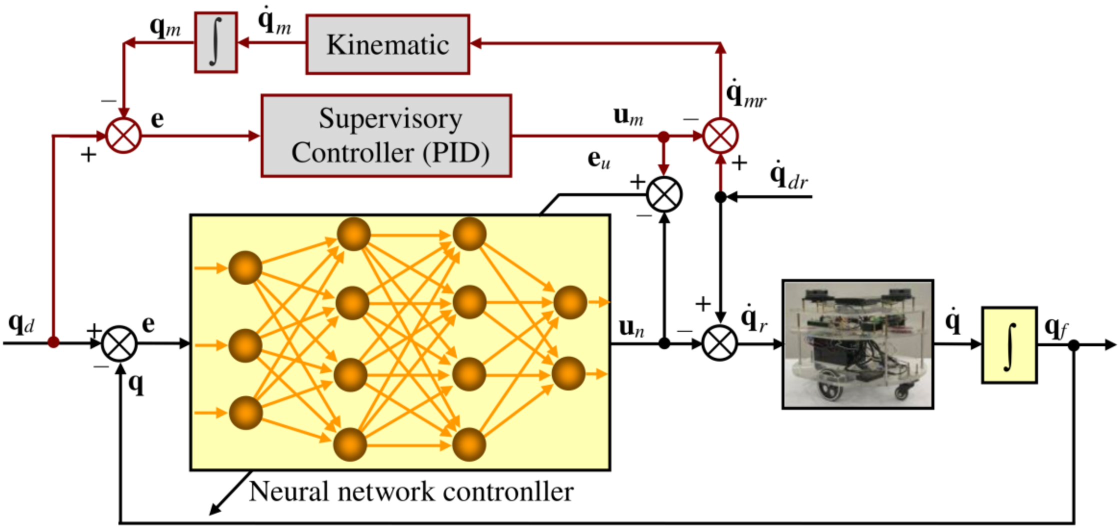

Figure 2. The diagram of the proposed controller structure.

The sample model is an improved PID controller with time-varying parameters, as proposed in [7], which generates control signals $\mathbf{u}_m$ to determine the linear velocity V_m and angular velocity $\Omega_m = \dot{\theta}$ for DDMR to follow the desired trajectory $\{\xi\}$. These signals can also be obtained through manual DDMR teaching. The structural parameters of the perceptron network are determined through a trial and error method, as described in Table 1. The inputs to the neural network are the reference state $\mathbf{q}_d$ and the current state $\mathbf{q}_f$, while the output is the signal $\mathbf{u}_n$.

Table 1. The neural network parameters.

Parameter	Value
Number of input neurons	3
Number of output neurons	2
Number of hidden layers	3
Number of neurons in hidden layer 1	4
Number of neurons in hidden layer 2	4
Number of neurons in hidden layer 3	2

wherein $f_1(n) = \dfrac{1-e^{-n}}{1+e^{-n}}$, $f_2(n) = \dfrac{1}{1+e^{-n}}$, and $f_3(n) = n$ are activation functions of the first, second, and third hidden layers. The trained neural network parameters are updated online so that:

$$\mathbf{e}_u = (\mathbf{u}_m - \mathbf{u}_n) \to 0 \tag{7}$$

with $\mathbf{u}_m$ is the control signal from the sample controller and $\mathbf{u}_n$ is the control signal from the neural network.

3.1. Neural Network Controller

The NN model is defined by:

$$\mathbf{u}_n = f_3(\mathbf{W}_3^T f_2(\mathbf{W}_2{}^T f_1(\mathbf{W}_1{}^T \mathbf{e}^T + \mathbf{b}_1^T) + \mathbf{b}_2^T) + \mathbf{b}_3^T)\,. \tag{8}$$

On the other hand, in the local coordinate system ϑ_r of DDMR, the control signal $\mathbf{u}_n$ is represented as follows:

$$\mathbf{u}_n = \dot{\mathbf{q}}_{dR} - \dot{\mathbf{q}}_n \tag{9}$$

here $\dot{\mathbf{q}}_{dr} = \mathbf{T}\dot{\mathbf{q}}_d$ with $\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}^T$; and $\mathbf{W}_i$, $\mathbf{b}_i$ are the weight matrix and bias vector of the neural network, respectively.

3.2. Neural network training

The parameters of the neural network, including the weight matrix **W** and the bias vector **b**, are trained according to:

$$\begin{cases} \mathbf{W}_m(k+1) = \mathbf{W}_m(k) - \alpha \mathbf{s}_m \mathbf{a}_{m-1}^T \\ \mathbf{b}_m(k+1) = \mathbf{b}_m(k) - \alpha \mathbf{s}_m \end{cases} \tag{10}$$

where $\mathbf{s}_m(k) = \dot{\mathbf{F}}_m(\mathbf{W}_m \mathbf{a}_{m-1} + \mathbf{b}_n)\mathbf{W}_{m+1}^T \mathbf{s}_{m+1}$ with m = 1, 2, …N; N is the number of layers of the NN, α is the learning rate.

$$\dot{\mathbf{F}}_m(n_m) = diag\left[\dot{f}_m(n_{m,1}),\ \dot{f}(n_{m,2}),\ldots,\dot{f}(n_{m,sm})\right];$$

$\mathbf{a}_n = f_m(\mathbf{W}_n\mathbf{a}_{n-1} + \mathbf{b}_n)$; f_m is the activation function of the m^{th} layer, $\dot{f}_m(n_{m,i}) = \dfrac{\partial f_m(n_{m,i})}{\partial n_{m,j}}$.

The weight value **W** and the bias vector value **b** of the neural network are adjusted so that the cost function, as given by equation (10), is minimized:

$$E = \frac{1}{2}(e_{u1}^2 + e_{u2}^2 + e_{u3}^2) \rightarrow \min \tag{11}$$

The online training sample data for the neural network includes the input **e**, which is determined by equation (3), and the corresponding output $\mathbf{u}_m$, which is determined by:

$$\mathbf{u}_m = \dot{\mathbf{q}}_{dr} - \dot{\mathbf{q}}_{mr} \tag{12}$$

Here $\mathbf{u}_m, \mathbf{q}_{mr}$ represent the output and control signals of the sample controller. The sample controller used in this research is an improved PID controller with time-varying parameters, as described in the literature [7].

4. SETTING SIMULATION PARAMETERS

Dimensions of DDMR

The distance between the two differential wheels is L = 0.3 m, and the radius differential wheels r = 0.0475 m.

Desired motion trajectory

Table 2. Interpolation points data for desired NUBRS motion trajectory *{ξ}*.

A_1	A_2	A_3	A_4	A_5	A_6	A_7
$\begin{bmatrix}0\\1.0\end{bmatrix}$	$\begin{bmatrix}0\\1.5\end{bmatrix}$	$\begin{bmatrix}0.5\\2.0\end{bmatrix}$	$\begin{bmatrix}1.5\\2.0\end{bmatrix}$	$\begin{bmatrix}2.5\\0\end{bmatrix}$	$\begin{bmatrix}3.5\\0\end{bmatrix}$	$\begin{bmatrix}4.0\\0.5\end{bmatrix}$
A_8	A_9	A_{10}	A_{11}	A_{12}	A_{13}	A_{14}
$\begin{bmatrix}4.0\\1.5\end{bmatrix}$	$\begin{bmatrix}3.5\\2.0\end{bmatrix}$	$\begin{bmatrix}1.5\\2.0\end{bmatrix}$	$\begin{bmatrix}1.5\\0\end{bmatrix}$	$\begin{bmatrix}0.5\\0\end{bmatrix}$	$\begin{bmatrix}0\\0.5\end{bmatrix}$	$\begin{bmatrix}0\\1.0\end{bmatrix}$

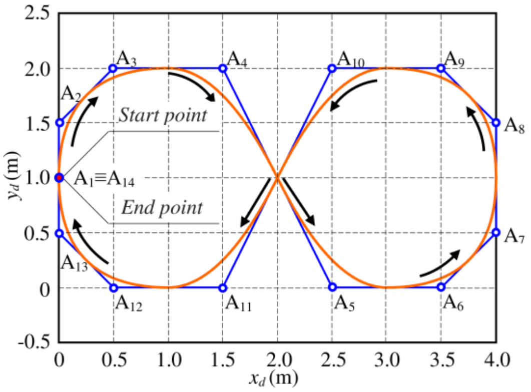

Figure 3. The desired trajectory of DDMR.

The red line in Figure 3 represents the desired motion trajectory, which is a 3rd order NURBS curve [10]. The aim is to ensure a smooth motion path and eliminate sudden changes in velocity and points of zero velocity of the DDMR. Table 2 provides the interpolation point data for the desired motion trajectory.

Motion parameters

With the desired trajectory ξ defined above, the time-varying velocity $V_d(t)$ is determined by:

$$V_d(t) = \frac{\Delta S}{\Delta t} = \frac{\sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}}{t_i - t_{i-1}} \tag{13}$$

where $V_d(t) \leq V_{dmax}$, and the desired angular velocity Ω_d of the DDMR is expressed by:

$$\Omega_d = \frac{V_d(t)}{\rho(t)} \tag{14}$$

Here $\rho_i \in [\rho_{min}, \rho_{max}]$ is the radius of curvature of the motion trajectory ξ with $i = 1,2, \ldots, n$ and is defined by:

$$\rho_i = \left| \frac{\left(\dot{x}_i^{\,2} + \dot{y}_i^{\,2}\right)^{3/2}}{\dot{x}_i \ddot{y}_i - \dot{y}_i \ddot{x}_i} \right| \tag{15}$$

Set $\Delta t = 0.1\,s$ and $V_{dmax} = 0.3$ *m/s*.

5. SIMULATION RESULTS AND DISCUSSION

The setup parameters encompass: (1) The robot's dimensions; (2) The structure of the neural network and the weight matrix update rule of training, and (3) The DDMR's desired trajectory and motion parameters. Below are the results controlling the trajectory-tracking motion of DDMR after 892 times of online update training.

5.1. Neural network training

Figure 4 shows the online training results based on 892 interpolation points of the NURBS trajectory.

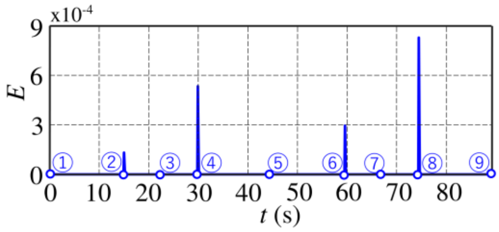

Figure 4. The cost function value.

Figure 4 shows that the cost function's E value consistently approaches zero once trained. Figures 5 to 7 below are graphs updating weights matrix **W** and bias vector **b** during NN training. Figure 4, combined with Figure 8, reveals that points 1 to 8 are where the DDMR changes direction from clockwise to counterclockwise or vice versa. Furthermore, Figures 5 to 7 indicate that the weight **W** and bias vector **b** are updated between the first 15 to 18 s.

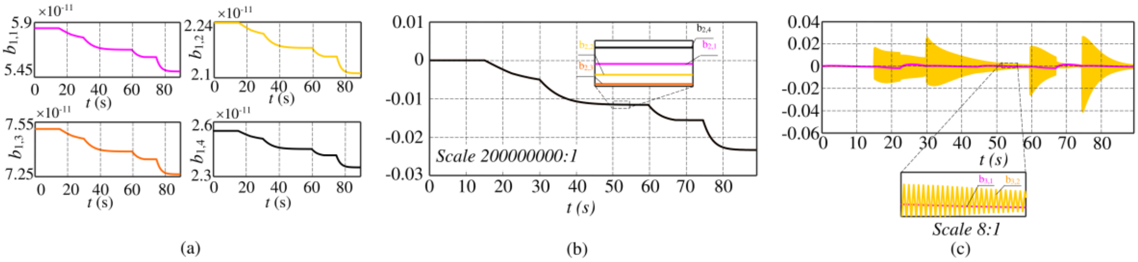

Figure 5. Update bias vector **b** in real-time of NN with: (a) hidden layer 1, (b) hidden layer 2 and (c) hidden layer 3.

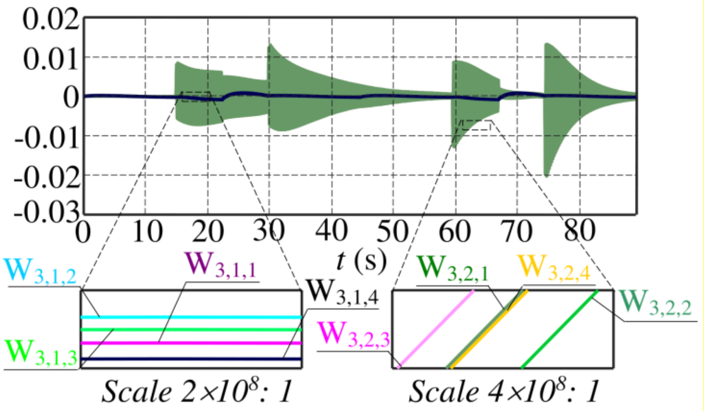

Figure 6. Update weight **W** in real-time of hidden layer 3.

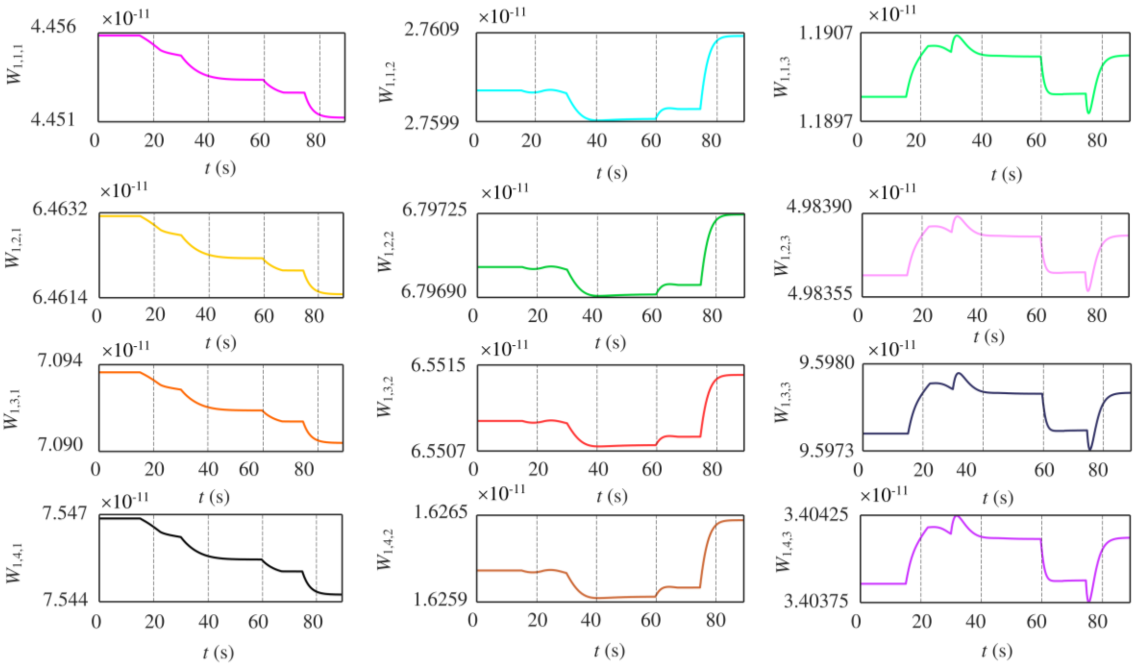

Figure 7a. Update weights **W** in real-time with: hidden layer 1.

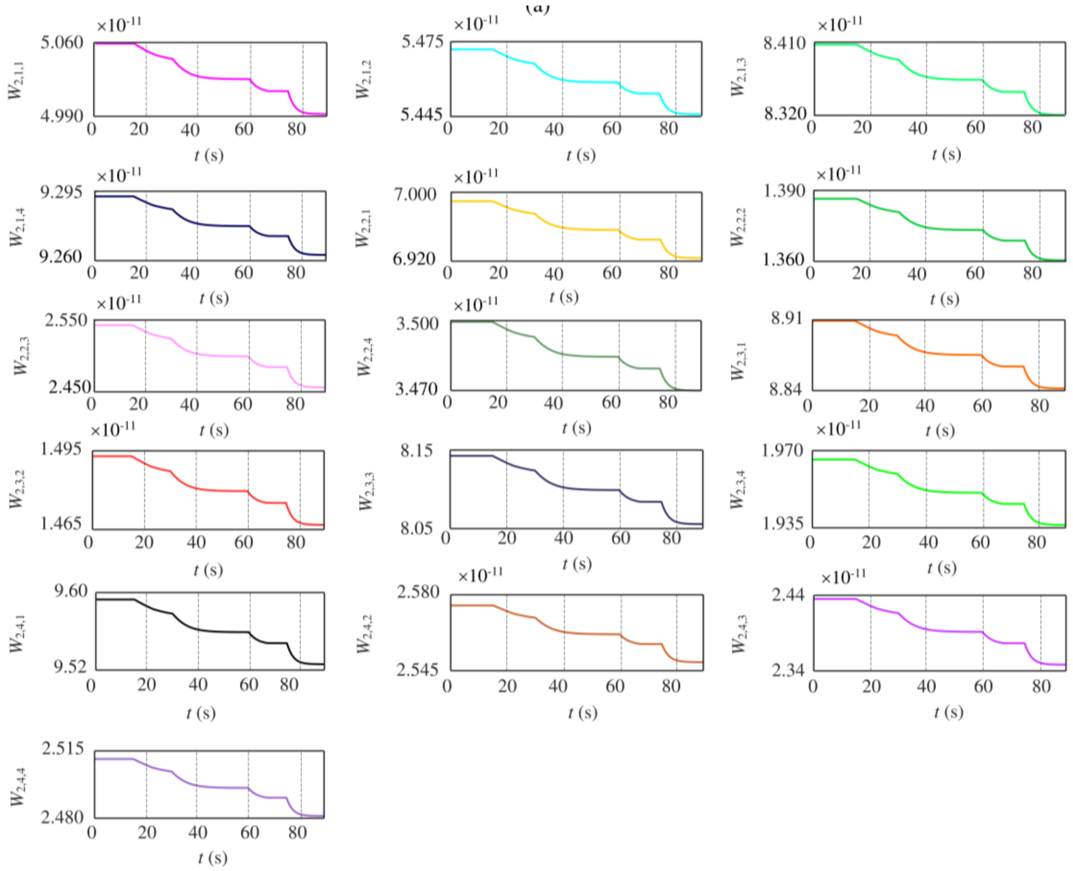

Figure 7b. Update weights **W** in real-time with: hidden layer 2.

5.2 The results of the trajectory tracking for DDMR

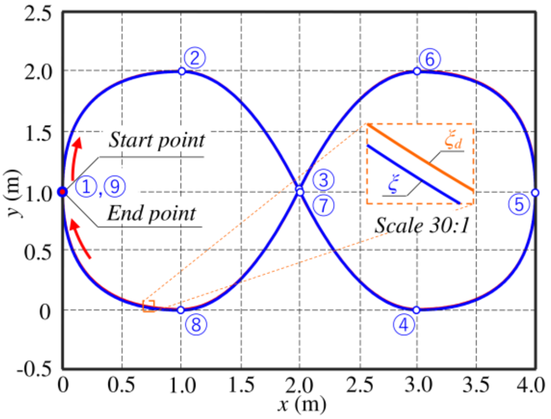

Figure 8. The moving trajectory tracking of the DDMR.

Once trained, the blue line is the motion trajectory of DDMR, while the desired trajectory is the orange line, as described in Figure 8. Figure 9 shows the position and posture errors of DDMR controlled by the proposed controller.

Figure 3, combined with Figures 8 and 9, shows that at the points where the DDMR changes direction (from clockwise to counterclockwise or vice versa), there will be a more significant error with verification data, as described in Table 3.

Table 3. The position error value at the point DDMR changes direction.

point	e_x (*mm*)	e_y (*mm*)	δ (*mm*)
1	0.000	0.000	0.000
2	12.737	1.657	12.844
3	8.318	20.048	21.705
4	1.330	4.247	13.963
5	1.855	16.706	16.809
6	11.184	4.347	11.999
7	6.616	17.117	18.351
8	11.337	7.216	13.439
9	0.203	19.624	19.626

Where the position error $\delta = \sqrt{e_x^2 + e_y^2}$ does not exceed 21.7 mm, and the posture error does not exceed 0.021 rad.

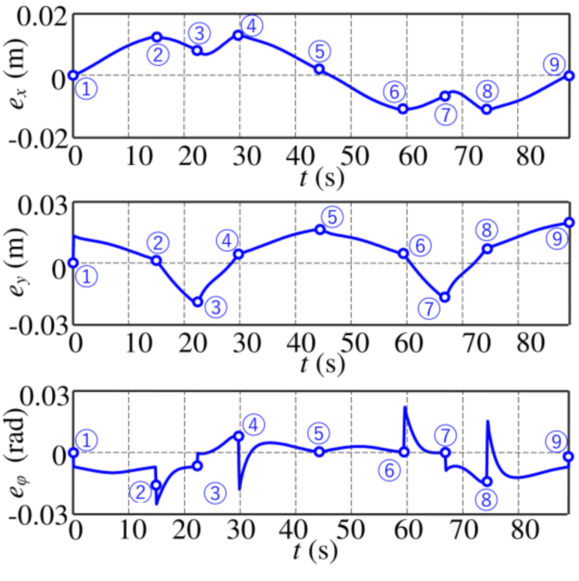

Figure 9. Position and posture error of DDMR.

Figure 10 displays the linear and angular velocities of the DDMR while following the NURBS motion trajectory. Figure 11 shows the linear and angular velocities error between the desired and controlled values.

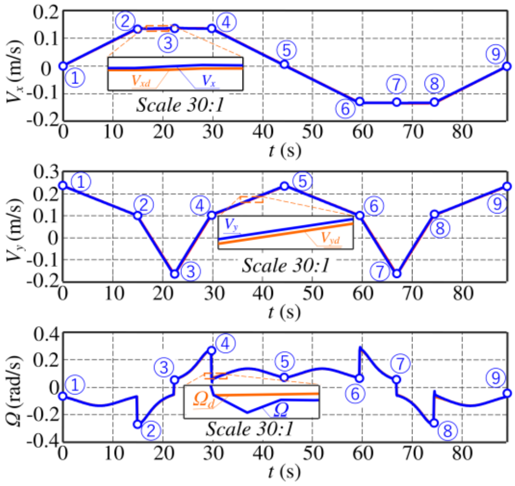

Figure 10. The linear and angular velocities of the DDMR.

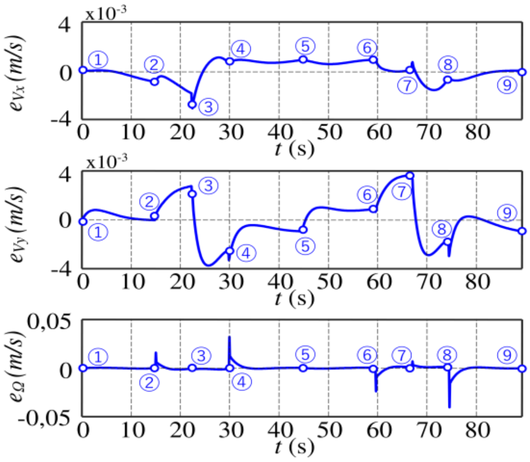

Figure 11. The errors in linear and angular velocity.

Figure 11 depicts the motion error of DDMR between the current value and the desired value. At points where the DDMR changes direction, the linear velocity error does not exceed 0.004 m/s, while the maximum angular velocity error is 0.0007 rad/s. This verifies the effectiveness of the proposed controller in ensuring that DDMR follows the NURBS trajectory.

Figure 12 describes the controlled angular velocity values of the two drive wheels, which are used to control DDMR to follow the desired trajectory. The red line represents the desired angular velocity of the driving wheels, and the blue line depicts their actual control state when DDMR changes direction, the error $\delta\omega$ between the actual angular velocity of the driving wheels and the desired value of 0.07 rad/s.

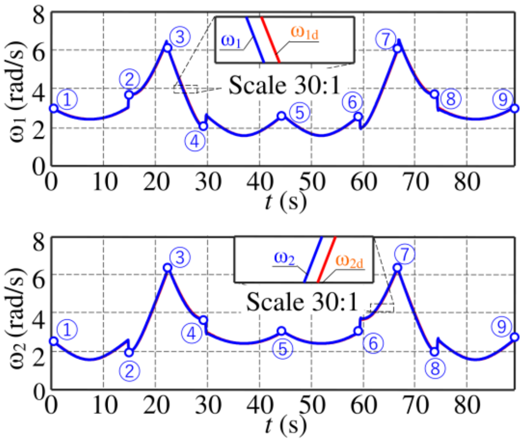

Figure 12. The angular velocities of differential wheels.

6. CONCLUSIONS

From the simulation results, discussion and evaluation above, this research achieved the following results: (1) The proposed NN-PID controller for DDMR tracked the trajectory with minor errors; (2) The simulation results verified the effectiveness of the proposed controller, which reduces computational complexity, achieves fast convergence, and updates weights online to ensure DDMR follows the desired trajectory. The proposed controller exhibited a tracking error not exceeding 2.17 cm and a maximum posture error of 0.021 rad. While the motion error includes the linear velocity error δV was limited to 0.004 m/s, and the angular velocity error $\delta\Omega$ was limited to 0.0007 rad/s. These results show the application potential of the proposed controller in trajectory tracking control for mobile robots with low hardware structure costs. Additionally, future research goals will include addressing the challenges of developing static and unexpected obstacle avoidance algorithms based on neural networks and integrating them with the controller.

Acknowledgement. This research is funded by Electric Power University under Research 2023.

CRediT authorship contribution statement. Trinh Thi Khanh Ly and Nguyen Hong Thai contributed to the idea's inception, theoretical modeling, implementation plan, and manuscript writing. Luu Thanh Phong performed simulations and consulted with Nguyen Hong Thai on essential aspects. All authors participated in the discussion of results and reviewed and approved the final manuscript.

Declaration of competing interest. We have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

REFERENCES

1. Wang Yujun, Fang C., Jiang Q., Ahmed S. N. - The automatic drilling system of 6R-2P mining drill jumbos, Advances in Mechanical Engineering **7** (2) (2015) 504861. https://doi.org/10.1155/2015/504861.
2. Thai N. H., Ly T. T. K., Long N. T., Thuong T. T. - Obstacle Avoidance Algorithm for Autonomous Mobile Robots in the Indoor Environment. In Advances in Engineering Research and Application, Lecture Notes in Networks and Systems, Springer, Cham. **602** (2022) 752-763. https://doi.org/10.1007/978-3-031-22200-9_79.
3. Marko Pedan, Milan Gregor, Dariusz Plinta - Implementation of Automated Guided Vehicle system in healthcare facility, Procedia Engineering **192** (2017) 665-670. https://doi.org/10.1016/j.proeng.2017.06.115.
4. Jun Qian, Bin Zi, Daoming Wang, Yangang Ma, Dan Zhang - The Design and Development of an Omni-Directional Mobile Robot Oriented to an Intelligent Manufacturing System, Sensors **17** (9) (2017). https://doi.org/10.3390/s17092073.
5. Tsai, Huan Liang Tsai Liang, Huynh Cao Tuan - Development of directional algorithm for three-wheel omnidirectional autonomous mobile robot, Vietnam Journal of Science and Technology **59** (3) (2021) 345-356.https://doi.org/10.15625/2525-2518/59/3/15583.
6. Ly, T.T.K., Thien, H., Nhan, D.K. and Thai, N.H.- Dynamic Simulation of Differential-Driven Mobile Robot Taking Into Account The Friction between The Wheel and The Road Surface. In International Conference on Material, Machines and Methods for Sustainable Development, (2022) 367-375.https://doi.org/10.1007/978-3-031-31824-5_44.
7. Thai N. H., Ly T. T. K., Thien H., Dzung L. Q. - Trajectory Tracking Control for Differential Drive Moblie Robot by a Variable Parameter PID Controller, International Journal of Mechanical Engineering and Robotics Research **11** (8) (2022) 614-621. https://doi.org/10.18178/ijmerr.11.8.614-621.
8. Cvejn J. and Tvrdík J. - Learning control of a robot manipulator based on a decentralized position-dependent PID controller, 21st International Conference on Process Control, (2017) 167-172. https://doi.org/ 10.1109/PC.2017.7976208.

9. Pour P. D., Alsayegh K. M., Jaradat M. A. - Type-2 Fuzzy Adaptive PID Controller for Differential Drive Mobile Robot: A Mechatronics Approach, Advances in Science and Engineering Technology International Conferences (ASET), IEEE (2022) 1-6. https://doi.org/10.1109/ASET53988.2022.9734882.
10. Thai N. H., Ly T. T. K. - NURBS curve trajectory tracking control for Differential-Drive Mobile Robot by a linear state feedback controller, Advances in Engineering Research and Application, Lecture Notes in Networks and Systems, Springer, Cham. **366** (2022) 685-696, https://doi.org/10.1007/978-3-030-92574-1_71.
11. Hassan Najva, and Abdul Saleem - Neural network-based adaptive controller for trajectory tracking of wheeled mobile robots, IEEE Access **10** (2022)13582-13597. https://doi.org/ 10.1109/ACCESS.2022.3146970.
12. Raeisi Y., Shojaei K., Chatraei A. - Output feedback trajectory tracking control of a car-like drive wheeled mobile robot using RBF neural network, The 6th Power Electronics, Drive Systems & Technologies Conference (2015) 363-368. https://doi.org/10.1109/ PEDSTC.2015.7093302.
13. Sun X., Deng S., Zhao T., Tong B. - Motion planning approach for car-like robots in unstructured scenario, Transactions of the Institute of Measurement and Control **44** (4) (2022) pp. 754-765. https://doi.org/10.1177/0142331221994393.
14. Wu H. M. - Nonlinear Trajectory -Tracking Control of a Car-Like Mobile Robot in the Presence of Input Saturations and a Pulse Disturbance, 58th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE), (2019) pp. 1498-1502. https://doi.org/10.23919/SICE.2019.8859890.
15. Živojević D., Velagić J. - Path Planning for Mobile Robot using Dubins-curve based RRT Algorithm with Differential Constraints, International Symposium ELMAR, (2019) pp. 139-142. https://doi.org/10.1109/ELMAR.2019.8918671.
16. Talebi Abatari H., Dehghani Tafti A. - Using a fuzzy PID controller for the path following of a car-like mobile robot, First RSI/ISM International Conference on Robotics and Mechatronics (ICRoM), (2013) pp. 189-193. https://doi.org/10.1109/ICRoM. 2013.6510103.
17. Jin Shang, Jian Zhang, Chengchun Li. - Trajectory tracking control of AGV based on time-varying state feedback, EURASIP journal on Wireless Communications and Networking **162** (2021) pp. 1-12. https://doi.org/10.1186/s13638-021-02034-x.
18. Wang H., Duan J., Wang M., Zhao J., Dong Z. - Research on Robot Path Planning Based on Fuzzy Neural Network Algorithm, 2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), (2018) 1800-1803. https://doi.org/10.1109/IAEAC.2018.8577599.
19. Guo, N., Li, C., Gao, T., Liu, G., Li, Y., Wang, D. - A fusion method of local path planning for mobile robots based on LSTM neural network and reinforcement learning, Mathematical Problems in Engineering 2021 (2021) 1-21. https://doi.org/ 10.1155/2021/5524232.