A Neural Network Controller Design for the Mecanum Wheel Mobile Robot

Trinh Thi Khanh Ly
Faculty of Control and Automation, Electric Power University (EPU), Vietnam
lyttk@epu.edu.vn

Nguyen Hong Thai
Faculty of Mechatronics, Hanoi University of Science and Technology (HUST), Vietnam
thai.nguyenhong@hust.edu.vn
(corresponding author)

Hoang Thien
Faculty of Mechatronics, Hanoi University of Science and Technology (HUST), Vietnam
hoangthien.hust@gmail.com

Nguyen Thi Thanh
Hanoi University of Science and Technology (HUST), Vietnam | University of Economics - Technology for Industries (UNETI), Vietnam
ntthanh.ddt@uneti.edu.vn

Received: 6 February 2023 | Revised: 26 February 2023 | Accepted: 4 March 2023

ABSTRACT

Advanced controllers are an excellent choice for the trajectory tracking problem of Wheeled Mobile Robots (WMRs). However, these controllers pose a challenge to the hardware structure of WMRs due to the controller's complex structure and the large number of calculations needed. In that context, designing a controller with a simple structure and a small number of computations but good real-time performance is necessary in order to improve the tracking accuracy for the WMRs without requiring high hardware architecture. In this work, a neural network controller with a simple structure for the trajectory-tracking of a Mecanum-Wheel Mobile robot (MWMR) based on a reference controller is proposed. A two-layer feedforward neural network is designed as a tracking controller for the robot. The neural network is trained with a sample input-output data set so that the error between the neural network output and the reference control signal of the supervisory controller is minimal. The neural network parameters are trained to update over time. The simulation results verified the effectiveness of the neural network controller, whose parameters are tuned adaptively to ensure a fast convergence to the desired Bézier trajectory.

Keywords-mecanum wheel mobile robot; tracking control; neural networks; Bézier trajectory

I. INTRODUCTION

There is a considerable scientific interest in the area of Wheeled Mobile Robots (WMRs), and especially Mecanum-Wheeled Mobile Robot (MWMRs), which have been widely applied to many fields due to their maneuverability and superior motion ability [1-3]. For control engineers, the trajectory tracking control problem of MWMR has always received significant attention. Trajectory tracking control is the key to realizing the autonomous movement of the MWMRs. However, the trajectory tracking control of the WMRs is a challenging study area because WMR systems are typically subjected to non-holonomic constraints, random disturbances and uncertainties. Many researchers studied control design and development with different controllers to improve the tracking control performance of the MWMRs. A time-varying parameter PID controller has been proposed in [4] to control a four-MWMR along a desired trajectory with minimal error. An advanced intelligent adaptive motion controller was designed in [5] using fuzzy wavelet networks for Mecanum Wheeled Omnidirectional Robots (MWORs) with parameter variations. Authors in [6] presented a novel neural network adaptive sliding mode control system for an omnidirectional vehicle with four mecanum wheels. Network weight adaptation was based on the analysis of the Lyapunov stability. Other controllers used for MWMR for trajectory tracking control used adaptive integral terminal sliding mode [7], robust adaptive control [8], adaptive fuzzy tracking control [9-11], PID controller with time-varying parameters [12], adaptive back stepping control using neural networks [13], predictive control [14], self-tuning fuzzy-PID control [15], and fuzzy adaptive PID control [16]. For robots to operate in a dynamic working environment and meet the required safety, accuracy, and reliability, advanced intelligent control systems are a valuable solution for the trajectory-tracking control problem [17-19]. The use of neural networks for the control of mobile robots has received attention from many researchers because they are self-learning and provide a real-time approximation of nonlinearities in a mathematical model of a robot using network weight adaptation [20-22]. Artificial Neural Networks (ANNs) are increasingly used to solve trajectory tracking problems, such as the adjustment of control coefficients, the choice of the direction of motion, speed correction, and identification of information from sensors.

The above discussion verified that the advanced controllers have excellent trajectory-tracking performance. Even so, they have the disadvantage that the complex controller structure leads to many complex calculations requiring high hardware structure. Therefore, it is necessary to design a controller with a simple structure, a small number of computations and good real-time performance to improve trajectory-tracking performance with a minor error for MWMR. Therefore, this work focuses on designing a simple computational ANN-based model reference controller to accommodate the trajectory tracking of the MWMR with minor errors, simultaneously significantly reducing the hardware's processing speed and capacity versus advanced controllers.

II. KINEMATIC ERROR MODEL OF MWMR

A. *Kinematic Model*

Consider an MWMR moving along the Bézier trajectory Σ with the assumption of no longitudinal and lateral slip in the global coordinate system $\vartheta_f\{O_f x_f y_f\}$, as shown in Figure 1. The angular velocities relationship of wheels to the linear and angular velocities of the MWMR are determined by [4]:

$$\boldsymbol{\omega} = \mathbf{J}\mathbf{Q}^T(\varphi)\dot{\mathbf{q}}_f \qquad (1)$$

where:

$$\boldsymbol{\omega} = [\omega_1(t) \quad \omega_2(t) \quad \omega_3(t) \quad \omega_4(t)]^T$$

$$\mathbf{J} = \begin{bmatrix} 1/r & 1/r & -(L+d)/2r \\ 1/r & -1/r & (L+d)/2r \\ 1/r & 1/r & (L+d)/2r \\ 1/r & -1/r & -(L+d)/2r \end{bmatrix}$$

$$\mathbf{Q}(\varphi) = \begin{bmatrix} \cos\varphi(t) & -\sin\varphi(t) & 0 \\ \sin\varphi(t) & \cos\varphi(t) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\dot{\mathbf{q}}_f = [\dot{x}(t) \quad \dot{y}(t) \quad \dot{\varphi}(t)]^T = [V_x \quad V_y \quad \Omega]^T$$

and r, L and d are the radii of the wheel, the distance in the x_R-axis and the y_R-axis of two wheels, V_x, V_y, and Ω are the linear and angular velocities of the MWMR in the global coordinate ϑ_f, respectively.

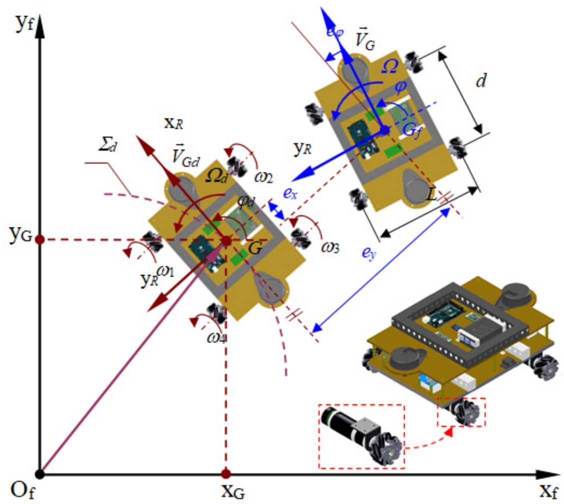


Fig. 1. Illustration og the kinematic error of MWMR.

From (1), the forward kinematics equation of the MWMR is given by:

$$\dot{\mathbf{q}}_f = \mathbf{Q}(\varphi)\dot{\mathbf{q}}_R = \mathbf{Q}(\varphi)\mathbf{J}^{+}\boldsymbol{\omega} \qquad (2)$$

where $\mathbf{J}^{+} = \left(\mathbf{J}^T\mathbf{J}\right)^{-1}\mathbf{J}^T$ is the pseudo-inverse of $\mathbf{J}$.

B. *Kinematic Error Model*

The error model describes the variation in position and orientation of the MWMR when moving along the desired trajectory Σ, defined by the position error vector $\mathbf{e}$ [23]:

$$\mathbf{e} = \mathbf{q}_d - \mathbf{q}_f = \begin{bmatrix} e_x & e_y & e_\varphi \end{bmatrix}^T \qquad (3)$$

$\mathbf{q}_d = [x_d(t) \quad y_d(t) \quad \varphi_d(t)]^T$ and $\mathbf{q}_f = [x_f(t) \quad y_f(t) \quad \varphi_f(t)]^T$ are the MWMR's desired and actual motion pose vectors in the global coordinate ϑ_f. The kinematic error model of the MWMR in the local coordinate system $\vartheta_R\{Gx_R y_R\}$ attached to the MWMR is given by:

$$\mathbf{e}_R = \begin{bmatrix} e_{xR} & e_{yR} & e_{\varphi R} \end{bmatrix}^T = \mathbf{Q}^T(\varphi)\mathbf{e} \qquad (4)$$

Derivative (4) is obtained by:

$$\dot{\mathbf{e}}_R = \tilde{\boldsymbol{\Omega}}\mathbf{e}_R + \mathbf{A}\dot{\mathbf{q}}_d - \dot{\mathbf{q}}_f \qquad (5)$$

where:

$$\tilde{\boldsymbol{\Omega}} = \begin{bmatrix} 0 & \Omega(t) & 0 \\ -\Omega(t) & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \text{ and } \mathbf{A} = \begin{bmatrix} \cos e_\varphi & -\sin e_\varphi & 0 \\ \sin e_\varphi & \cos e_\varphi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

III. NEURAL NETWORK CONTROLLER DESIGN FOR THE MECANUM WHEEL MOBILE ROBOT

The proposed Neural Network (NN) control system to track the desired motion trajectory is described in Figure 2. The control system consists of the NN controller, the kinematic model of MWMR, the reference controller and the desired trajectory Σ. The NN's parameters are adapted online according to the reference model described in Figure 2, in which the reference controller is the kinematic controller proposed in [23] to generate the reference control signals $\mathbf{u}_r$ to determine the reference linear and angular velocities (V_{rR}, Ω_{rR}) for the MWMR to follow the desired trajectory Σ. In [23], the control gains K_1, K_2 and K_3 of the reference controller are time-varying parameters for MWMR to achieve high tracking accuracy. The parameters for the proposed ANN controller are given in Table I. The inputs to the ANN are the position errors of MWMR $\mathbf{e}$, and the ANN's outputs are $\mathbf{u}_n$ (the velocities of MWMR).

TABLE I. THE NEURAL NETWORK PARAMETERS

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

$f_1(n) = \dfrac{1}{1+e^{-n}}$ and $f_2(n) = n$ are the activation functions of the 1st and 2nd hidden layer, respectively. The output $\mathbf{u}_n$ is compared with the reference control signal $\mathbf{u}_r$, and the error is used to adapt the weights W and bias b in order to reduce the NN's error on the output:

$$\mathbf{e}_u = (\mathbf{u}_r - \mathbf{u}_n) \rightarrow 0 \qquad (6)$$

$f_1(n) = \dfrac{1}{1+e^{-n}}$ and $f_2(n) = n$ are the activation functions of the 1st and 2nd hidden layer, respectively. The output $\mathbf{u}_n$ is compared with the reference control signal $\mathbf{u}_r$, and the error is used to adapt the weights W and bias b in order to reduce the NN's error on the output:

$$\mathbf{e}_u = (\mathbf{u}_r - \mathbf{u}_n) \rightarrow 0 \qquad (6)$$

IV. NEURAL NETWORK CONTROLLER

A. *The Neuron Network Controller Model*

It is defined by:

$$\mathbf{u}_n = f_2(\mathbf{W}_2^{\,T} f_1(\mathbf{W}_1^{\,T}\mathbf{e}^T + \mathbf{b}_1^T) + \mathbf{b}_2^T) \qquad (7)$$

where $\mathbf{W}_2, \mathbf{W}_1, \mathbf{b}_1, \mathbf{b}_2$ are the weight matrix and bias vector of the NN, respectively, and $\mathbf{e}$ is given by (2). On the other hand, considering the MWMR's local coordinate system, we have:

$$\dot{\mathbf{q}}_n = \dot{\mathbf{q}}_{dR} - \mathbf{u}_n \qquad (8)$$

where:

$$\dot{\mathbf{q}}_{dR} = \mathbf{T}\dot{\mathbf{q}}_d \text{ with } \mathbf{T} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}^T$$

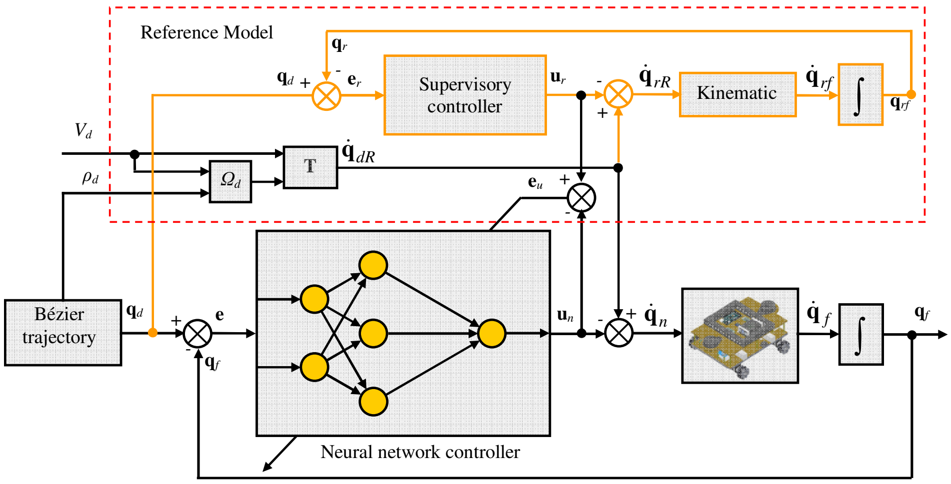


Fig. 2. Structure of neurocontrolled MWMR with supervised learning.

B. *Neural Network Training*

The parameters of the NN, including the weight matrix $\mathbf{W}$ and the bias vector $\mathbf{b}$, are updated as the time by the back-propagation method according to the following law:

$$\begin{cases} \mathbf{W}_m(k+1) = \mathbf{W}_m(k) - \alpha \mathbf{s}_m \mathbf{a}_{m-1}^T \\ \mathbf{b}_m(k+1) = \mathbf{b}_m(k) - \alpha \mathbf{s}_m \end{cases} \qquad (9)$$

where:

$$\mathbf{s}_m(k) = \dot{\mathbf{F}}_m(\mathbf{W}_m \mathbf{a}_{m-1} + \mathbf{b}_n)\mathbf{W}_{m+1}^T \mathbf{s}_{m+1},$$

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

$$\mathbf{a}_n = f_n(\mathbf{W}_n \mathbf{a}_{n-1} + \mathbf{b}_n),$$

$$\dot{f}_m(n_{m,i}) = \frac{\partial f_m(n_{m,i})}{\partial n_{m,j}}.$$

where m = 1, 2,..N, N is the number of layers of the NN, $a \in [0,1]$ is the learning rate, f_n is the activation function of the m^{th} layer. The weights values of the NN are adjusted so that the cost function given by (10) is minimized:

$$E = \frac{1}{2}\mathbf{e}_u^T\mathbf{e}_u \to \min \qquad (10)$$

V. SIMULATION SETUP

A. The MWMR Dimension Setup

The dimensions of the MWMR are: length × width × height of 380mm × 260mm×165mm, L = 316mm, d = 270mm, and r = 30mm is the radius of wheels.

B. The Bézier Moving Trajectory of the Robot

The Bézier curve [24] is defined by trajectory interpolation points and is given by:

$$P(t) = \sum_{i=0}^{n} B_i J_{n,i}(t), 0 \le t \le 1 \qquad (12)$$

where B_i represents the coordinates of interpolated points, as described in Figure 3, i is the ordinal number of the interpolation points, and $J_{n,i}(t)$ is a Bernstein polynomial of degree n, given by:

$$J_{n,i}(t) = \binom{n}{i} t^i (1-t)^{n-i} \qquad (13)$$

where $\binom{n}{i} = \frac{n!}{i!(n-i)!}$ is the the Pascal coefficient, and n is the Bernstein polynomial degree and the curve's degree. From the above Bézier curve design method, we have the desired moving trajectory of the robot shown in Figure 3 with the points A_i (i = 1-14) being the interpolation points.

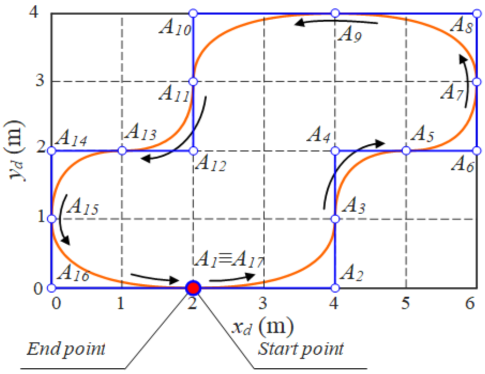


Fig. 3. The desired Bézier trajectory of the MWMR.

C. MWMR Linear and Angular Velocity Determination

The desired velocity $V_d(t)$ of the MWMR is determined by the Bézier motion trajectory Σ_d as follows:

$$V_d(t) = \frac{\Delta S}{\Delta t} = \frac{\sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}}{t_i - t_{i-1}} \qquad (14)$$

The desired angular velocity Ω_d of the MWMR is determined by:

$$\Omega_d(t) = V_d(t) / \rho(t) \qquad (15)$$

where $\rho_i \in [\rho_{\min}, \rho_{\max}]$ is the radius of the desired trajectory (i= 1, 2, .), obtained by:

$$\rho_i = \left| \frac{\left(\dot{x}_i^{\,2} + \dot{y}_i^{\,2}\right)^{3/2}}{\dot{x}_i \ddot{y}_i - \dot{y}_i \ddot{x}_i} \right| \qquad (16)$$

where:

$$\begin{cases} \dot{x}_i = \dfrac{\Delta x}{\Delta t} = \dfrac{x_i - x_{i-1}}{t_i - t_{i-1}} \\ \dot{y}_i = \dfrac{\Delta y}{\Delta t} = \dfrac{y_i - y_{i-1}}{t_i - t_{i-1}} \end{cases}, \quad \begin{cases} \ddot{x}_i = \dfrac{\Delta \dot{x}}{\Delta t} = \dfrac{\dot{x}_i - \dot{x}_{i-1}}{t_i - t_{i-1}} \\ \ddot{y}_i = \dfrac{\Delta \dot{y}}{\Delta t} = \dfrac{\dot{y}_i - \dot{y}_{i-1}}{t_i - t_{i-1}} \end{cases}.$$

Clearly, from (14)-(16) and the data of the trajectory interpolation points given in Figure 3, we have the desired linear velocity V_d and angular velocity $\mathbf{\Omega}_d$ of the MWMR described in Figure 4 when the MWMR moves along the Bézier trajectory Σ with Δt = 0.1s and maximal allowed speed V_{dmax} = 0.3m/s. Thus, the desired $\dot{\mathbf{q}}_d$ vector is defined by $\dot{\mathbf{q}}_d = [V_d \quad \Omega_d]^T$.

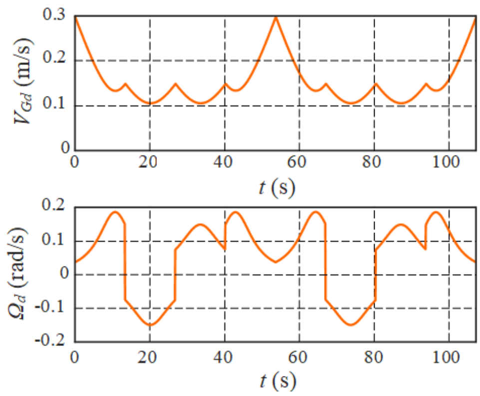


Fig. 4. The desired linear and angular velocites of the MWMR.

D. Setting Parameters of Neural Network Training

The initialization values at the input/output of the NN while training are: $\mathbf{a}_0 = \mathbf{e}$, $\mathbf{a}_2 = \mathbf{u}$ and $\mathbf{s}_2(k) = -\dot{\mathbf{F}}_2(\mathbf{W}_2\mathbf{a}_1 + \mathbf{b}_2)\mathbf{e}_u$. The learning coefficient α = 1 is determined by trial and error, and is such that the jump does not exceed the optimal value. The parameter matrices are updated online to control the MWMR trajectory tracking with a minor tracking error.

VI. RESULTS AND DISCUSSION

The installed parameters consist of (1) the size of the MWMR, (2) the parameters and the training parameters of the neural network, and (3) the motion trajectory and the parameters of MWMR. The motion trajectory of the MWMR, after 1072 online training updates, is depicted in Figure 5, with the orange line as the desired trajectory and the blue line as the motion trajectory of the robot controlled by the proposed neural network controller. The position and posture errors of MWMR are depicted in Figure 6.

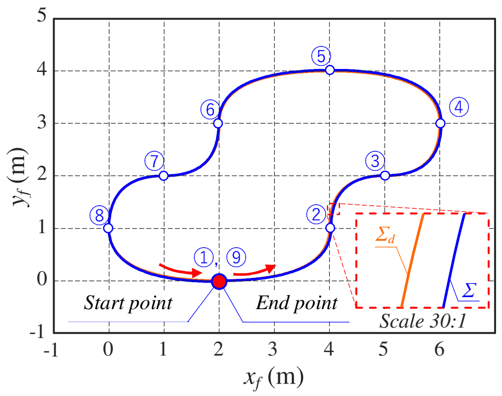


Fig. 5. The moving trajectory of the MWMR.

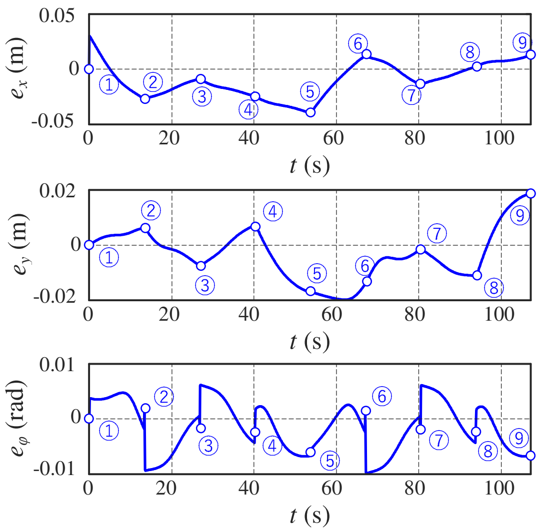


Fig. 6. The position error of the G-point and posture error of the robot.

Figure 6 shows that the proposed controller has the maximum error in the x-direction (e_{xmax}) not more than 11mm, and the maximum position error in the y - direction (e_{ymax}) is 17mm at position 9. The maximum posture error of the robot $e_{\varphi max}$ does not exceed 0.34°. The points marked from 1 to 9 in Figure 6 correspond to where the MWMR changes direction in Figures 5 and 3. In general, significant position errors often occur at the points where the MWMR changes direction. These are the points where the MWMR changes from clockwise to counterclockwise rotation and vice versa. Figures 5 and 6 clearly show that the proposed controller can track the desired trajectory with minor errors.

Figure 7 shows the cost function value when training the neural network to update the weight matrices **W** and bias vector **b** in real time. From Figure 7, it is easy to see that at the points where the direction of the MWMR change, the control signal of the neural network has a deviation from the signal of the reference control model. Still, it was trained in time to minimize the error to asymptotically reach zero.

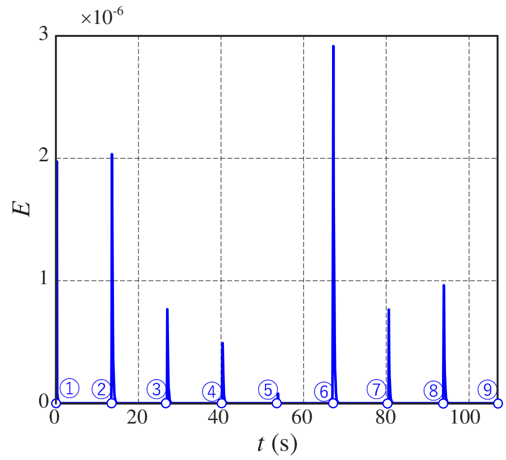


Fig. 7. The cost fucntion.

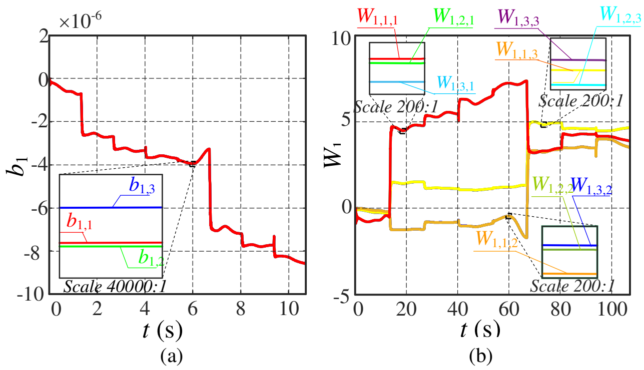
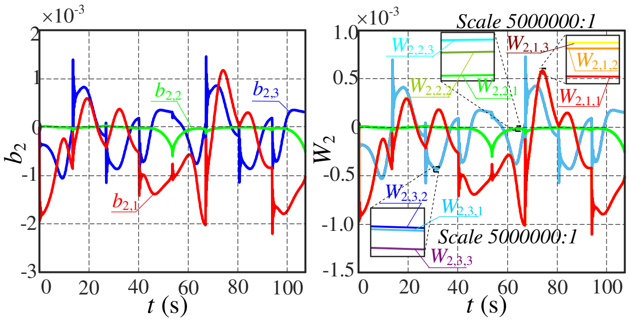


Fig. 8. The network parameters of the first hidden layer: (a) The network bias $\mathbf{b}_1$, (b) The network weight matrix $\mathbf{W}_1$.

Figures 8 and 9 verify the online updating of the network weight matrix **W** and bias vector **b** of the proposed controller for the MWMR to follow the Bézier trajectory. The NN weights from zero initial values changed during the adaptation process and stabilized at specific values. Since the neural network structure is rather simple, the proposed neural network can find the optimum values of the network parameters quickly and accurately. Figure 10 shows that the velocities of the MWMR are adjusted around the desired value such that the MWMR position to an asymptote value is '**0**'. Figure 10 shows that the velocity error $\mathbf{V}_{xr}$ ranges from 0 to 0.0011m/s, the velocity error $\mathbf{V}_{yr}$ is adjusted around the desired value 0 to 0.0009m/s, and the maximum angular velocity error of MWMR is more than 0.00054rad/s. Figure 10 shows that the proposed controller is precisely following the reference linear and angular velocities. Figures 11 and 12 show that the angular velocities of the wheels always follow the desired values, and the angular velocities error varies from 0 to 0.05rad/s. The angular velocity errors were limited and had the most significant values at the beginning and at the points where the MWMR changed direction.



(a) (b)

Fig. 9. The ANN parameters of the second hidden layer 2: (a) The network bias $\mathbf{b}_2$, (b) The network weight matrix $\mathbf{W}_2$.

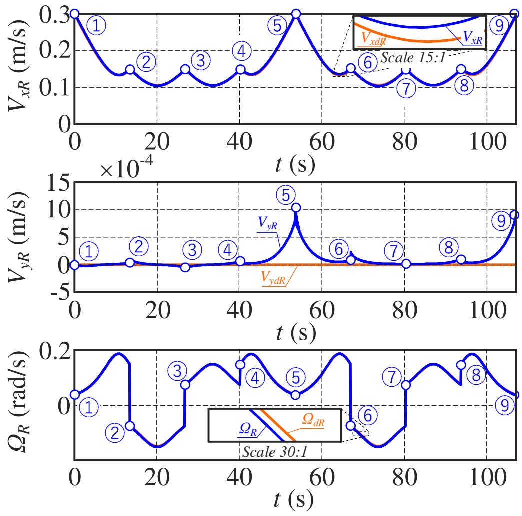


Fig. 10. Linear and angular velocities of the MWMR.

The efficacy of the proposed controller can also be proved from the velocity and error curves, which show that the control system based on the NN controller can track the desired trajectory with minimum tracking error, and the control inputs strictly follow the reference velocities. Additionally, the neural network has overcome the disadvantages of our previous studies [4], where the time-varying PID controller was used to control the MWMR, such as (i) the time-varying PID controller requires the linear model of the robot and (ii) the PID controller needs to know the functions of the controller gains and the gains are often assumed as the linear function of the robot error.

VII. CONCLUSIONS

The following conclusions can be derived from the results and discussion of the current study.

An artificial neural network-based model reference controller for the Bézier trajectory tracking of the mecanum-wheeled mobile robot with a minor tracking error is proposed in this paper. Compared to other controllers, the neural network in the proposed controller has a simple structure, but it effectively improves performance with minor error. This primarily ensures that a small number of computations increases the convergence speed to update the control gains online. The adapted gains are derived by satisfying the condition that the error between the actual and controller outputs is zero asymptotically.

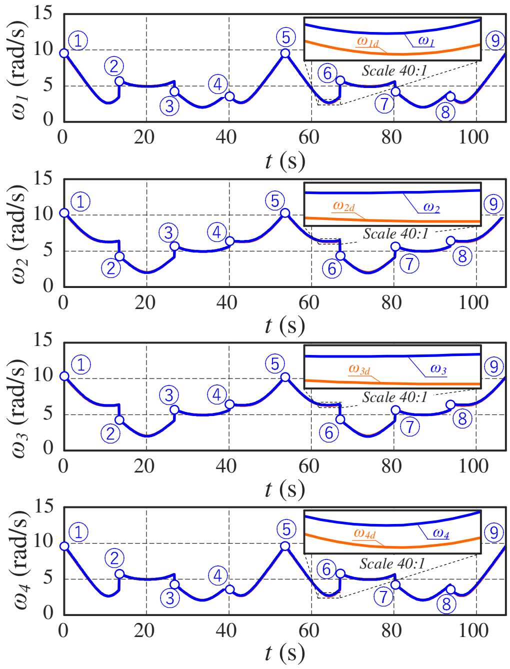


Fig. 11. The angular velocities of the four wheels.

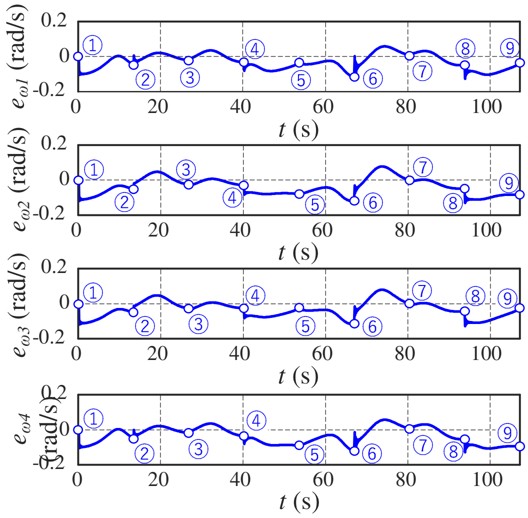


Fig. 12. The angular velocity error of four wheels.

The simulation results demonstrate the effectiveness of the proposed controller. They show that the errors in the x- and y-directions do not exceed 11mm and 17mm, respectively, and the direction error does not exceed 0.34°. In addition, the position and orientation errors of the MWMR are often significant at the points where the MWMR changes direction to the desired trajectory tracking. We believe this will yield guarantees on the tracking performance of the MWMR. Additionally, the proposed controller exhibited motion error, including the linear velocity error, limited to 0.0014m/s and angular velocity error limited to 0.00054rad/s.

Likewise, the proposed controller can be applied to any desired trajectory with minimal control action and perfect orientation with low hardware structure cost. Nevertheless, this research does not account for the longitudinal and lateral slip, friction at the wheel-ground contact points, friction between the roller rotation and the roller's shaft on the wheels, and inertia forces. These are considered parts of our future research goals.

ACKNOWLEDGEMENT

This research is funded by Hanoi University of Science and Technology (HUST) under project number T2022-PC-030. Master's student Hoang Thien was funded by the Master Scholarship Programme of Vingroup Innovation Foundation (VINIF), Code: VINIF.2022.ThS.128.

REFERENCES

[1] M. Szeremeta and M. Szuster, "Neural Tracking Control of a Four-Wheeled Mobile Robot with Mecanum Wheels," *Applied Sciences*, vol. 12, no. 11, Jan. 2022, Art. no. 5322, https://doi.org/10.3390/app12115322.

[2] J. Qian, B. Zi, D. Wang, Y. Ma, and D. Zhang, "The Design and Development of an Omni-Directional Mobile Robot Oriented to an Intelligent Manufacturing System," *Sensors*, vol. 17, no. 9, Sep. 2017, Art. no. 2073, https://doi.org/10.3390/s17092073.

[3] N. H. Thai, T. T. K. Ly, H. Thien, and L. Q. Dzung, "Trajectory Tracking Control for Differential-Drive Mobile Robot by a Variable Parameter PID Controller," *International Journal of Mechanical Engineering and Robotics Research*, pp. 614–621, 2022, https://doi.org/10.18178/ijmerr.11.8.614-621.

[4] N. H. Thai, T. T. K. Ly, and L. Q. Dzung, "Trajectory tracking control for mecanum wheel mobile robot by time-varying parameter PID controller," *Bulletin of Electrical Engineering and Informatics*, vol. 11, no. 4, pp. 1902–1910, Aug. 2022, https://doi.org/10.11591/eei.v11i4.3712.

[5] Ching-Chih Tsai, Hsiao-Lang Wu, and Ying-Ru Lee, "Intelligent Adaptive Motion Controller Design for Mecanum Wheeled Omnidirectional Robots with Parameter Variations," *International Journal of Nonlinear Sciences and Numerical Simulation*, vol. 11, no. Supplement, pp. 91–96, Dec. 2010, https://doi.org/10.1515/IJNSNS.2010.11.S1.91.

[6] X. Lu, X. Zhang, G. Zhang, J. Fan, and S. Jia, "Neural network adaptive sliding mode control for omnidirectional vehicle with uncertainties," *ISA Transactions*, vol. 86, pp. 201–214, Mar. 2019, https://doi.org/10.1016/j.isatra.2018.10.043.

[7] Z. Sun, S. Hu, D. He, W. Zhu, H. Xie, and J. Zheng, "Trajectory-tracking control of Mecanum-wheeled omnidirectional mobile robots using adaptive integral terminal sliding mode," *Computers & Electrical Engineering*, vol. 96, Dec. 2021, Art. no. 107500, https://doi.org/10.1016/j.compeleceng.2021.107500.

[8] V. Alakshendra and S. S. Chiddarwar, "Adaptive robust control of Mecanum-wheeled mobile robot with uncertainties," *Nonlinear Dynamics*, vol. 87, no. 4, pp. 2147–2169, Mar. 2017, https://doi.org/10.1007/s11071-016-3179-1.

[9] T. Zhao, X. Zou, and S. Dian, "Fixed-time observer-based adaptive fuzzy tracking control for Mecanum-wheel mobile robots with guaranteed transient performance," *Nonlinear Dynamics*, vol. 107, no. 1, pp. 921–937, Jan. 2022, https://doi.org/10.1007/s11071-021-06985-0.

[10] H. Medjoubi, A. Yassine, and H. Abdelouahab, "Design and Study of an Adaptive Fuzzy Logic-Based Controller for Wheeled Mobile Robots Implemented in the Leader-Follower Formation Approach," *Engineering, Technology & Applied Science Research*, vol. 11, no. 2, pp. 6935–6942, Apr. 2021, https://doi.org/10.48084/etasr.3950.

[11] B. Kasmi and A. Hassam, "Comparative Study between Fuzzy Logic and Interval Type-2 Fuzzy Logic Controllers for the Trajectory Planning of a Mobile Robot," *Engineering, Technology & Applied Science Research*, vol. 11, no. 2, pp. 7011–7017, Apr. 2021, https://doi.org/10.48084/etasr.4031.

[12] T. Nguyen Hong, Thai, H. Trinh Thi Thu, and K. Ly, "Path tracking control for car-like robots by pid controller with time-varying parameters," *Science & Technology Development Journal – Engineering and Technology*, vol. 5, no. 3, pp. 1642–1650, Dec. 2022, https://doi.org/10.32508/stdjet.v5i3.961.

[13] M. Jiang, L. Chen, Y. Wang, and H. Wu, "Adaptive Backstepping Control for Mecanum-Wheeled Omnidirectional Vehicle Using Neural Networks," *IEEJ Transactions on Electrical and Electronic Engineering*, vol. 17, no. 3, pp. 378–386, 2022, https://doi.org/10.1002/tee.23521.

[14] D. Wang, W. Wei, Y. Yeboah, Y. Li, and Y. Gao, "A Robust Model Predictive Control Strategy for Trajectory Tracking of Omni-directional Mobile Robots," *Journal of Intelligent & Robotic Systems*, vol. 98, no. 2, pp. 439–453, May 2020, https://doi.org/10.1007/s10846-019-01083-1.

[15] E. Malayjerdi, H. Kalani, and M. Malayjerdi, "Self-Tuning Fuzzy PID Control of a Four-Mecanum Wheel Omni-directional Mobile Platform," in *Iranian Conference on Electrical Engineering (ICEE)*, Mashhad, Iran, Feb. 2018, pp. 816–820, https://doi.org/10.1109/ICEE.2018.8472568.

[16] G. Cao, X. Zhao, C. Ye, S. Yu, B. Li, and C. Jiang, "Fuzzy adaptive PID control method for multi-mecanum-wheeled mobile robot," *Journal of Mechanical Science and Technology*, vol. 36, no. 4, pp. 2019–2029, Apr. 2022, https://doi.org/10.1007/s12206-022-0337-x.

[17] M. Boukens and A. Boukabou, "Design of an intelligent optimal neural network-based tracking controller for nonholonomic mobile robot systems," *Neurocomputing*, vol. 226, pp. 46–57, Feb. 2017, https://doi.org/10.1016/j.neucom.2016.11.029.

[18] D. Janglová, "Neural Networks in Mobile Robot Motion," *International Journal of Advanced Robotic Systems*, vol. 1, no. 1, Mar. 2004, Art. no. 2, https://doi.org/10.5772/5615.

[19] P. Bozek, Y. L. Karavaev, A. A. Ardentov, and K. S. Yefremov, "Neural network control of a wheeled mobile robot based on optimal trajectories," *International Journal of Advanced Robotic Systems*, vol. 17, no. 2, Mar. 2020, Art. no. 1729881420916077, https://doi.org/10.1177/1729881420916077.

[20] R. Fierro and F. L. Lewis, "Control of a nonholonomic mobile robot using neural networks," *IEEE Transactions on Neural Networks*, vol. 9, no. 4, pp. 589–600, Jul. 1998, https://doi.org/10.1109/72.701173.

[21] J. Velagic, N. Osmic, and B. Lacevic, "Neural Network Controller for Mobile Robot Motion Control," *International Journal of Intelligent Systems and Technologies*, vol. 3, no. 3, pp. 127–132, Jan. 2008.

[22] M. Fouzia, N. Khenfer, and N. E. Boukezzoula, "Robust Adaptive Tracking Control of Manipulator Arms with Fuzzy Neural Networks," *Engineering, Technology & Applied Science Research*, vol. 10, no. 4, pp. 6131–6141, Aug. 2020, https://doi.org/10.48084/etasr.3648.

[23] N. H. Thai and T. T. K. Ly, "NURBS Curve Trajectory Tracking Control for Differential-Drive Mobile Robot by a Linear State Feedback Controller," in *Advances in Engineering Research and Application*, 2022, pp. 685–696, https://doi.org/10.1007/978-3-030-92574-1_71.

[24] L. T. K. Trịnh and H. Thien, "Bézier trajectory tracking control of The Omnidirectional Mobile Robot based on a linear time- varying state feedback controller," *VNUHCM Journal of Science and Technology Development*, vol. 25, no. 2, pp. 2444–2452, Aug. 2022, https://doi.org/10.32508/stdj.v25i2.3914.