Bulletin of Electrical Engineering and Informatics
Vol. 11, No. 4, August 2022, pp. 1902~1910
ISSN: 2302-9285, DOI: 10.11591/eei.v11i4.3712

Trajectory tracking control for mecanum wheel mobile robot by time-varying parameter PID controller

Nguyen Hong Thai[1], Trinh Thi Khanh Ly[2], Le Quoc Dzung[2]
[1]Faculty of Mechatronics, Hanoi University of Science and Technology (HUST), Hanoi, Vietnam
[2]Faculty of Automation Technology, Electric Power University (EPU), Hanoi, Vietnam

Article Info

Article history:

Received Feb 21, 2022
Revised May 9, 2022
Accepted Jun 7, 2022

Keywords:

Mecanum wheel
NURBS trajectory
Omnidirectional robot
PID controller
Trajectory tracking

ABSTRACT

This paper presents the design method a proportional integral derivative (PID) controller with time-varying parameters for trajectory tracking of a mecanum-wheeled robot with a minor error. First, a nonlinear kinematic error model of the robot is established and linearized around the working point. Then a PID controller with time-varying parameters is designed based on this linear model. The coefficients of K_P, K_I, and K_D control parameters are determined by trial and error technique to ensure that the robot moves along the desired trajectory with minimal error. A platform mecanum-wheeled mobile robot has been designed and manufactured to demonstrate the proposed controller. Simulation and experimental results are presented to verify the effective and accuracy of the proposed controller. It shows that can apply this research to control of a four mecanum-wheeled mobile robot in logistics services in practice.

Corresponding Author:

Nguyen Hong Thai
Faculty of Mechatronics, Hanoi University of Science and Technology (HUST)
No 1, Dai Co Viet Road, Hanoi, Vietnam
Email: thai.nguyenhong@hust.edu.vn

1. INTRODUCTION

Nowadays, autonomous wheeled mobile robots are used in logistics to transport materials automatically, components, supplies, products, and sell products, in intelligent manufacturing and logistics systems [1]. In consideration of a complicated working environment and tight spaces, which is not enough to design the turning radii for the robot, lead to mecanum-wheeled mobile robot (MWMR) has been developed to satisfy the whole smart logistics systems and the modern industrial environment [2]. In intelligent manufacturing, the MWMR is integrated with an industrial IoT-based system to intra-factory logistics automation [3]. In addition, the MWMR is also integrated with a manipulator to perform different tasks such as: machining large-scale wind turbine blades by the mobile robots [4]; the robot surveillance of the Super Proton Synchrotron accelerator of the European Organization for nuclear research [5]; and the aromatherapy robot delivers therapeutic vaporized essential oils or drugs needed to prevent or treat Covid-19 infections [6]. However, for the MWMRs to work effectively in a system the main problem is to control its movement precisely following the desired trajectory. Therefore, many controllers have been proposed to control the MWMRs to follow the desired trajectory, such as: the linear state feedback controller [7], [8]; recommended controller based on the combination of the robust sliding mode observer with the adaptive controller [9]–[11]; the fuzzy controller [12] or the fuzzy-proportional integral (fuzzy-PI) controller adjusts the parameters to decrease the position and posture errors of the robot due to the dynamic changes and navigation challenges [13], [14], using genetic algorithm to find near-optimal travelling path in logistics task scheduling problem for a mobile robot [15], and robust nonsingular terminal sliding model-based controller design for the path-following problem with higher tracking precision [16], [17]. Thereby it can be seen advanced controllers have demonstrated outstanding potential, but the complex controller structure requires the processor and the memory capacity of the hardware is large. Thus, proportional integral derivative (PID) is still the favourite in mobile robot control applications especially in the case of path tracking control problems for the robot [6], [18], [19]. But, when the trajectory is complex and the desired velocity varies with time, the time-invariant parameter PID controller does not efficiently control the system because the accuracy is not high. A robot cannot be asymptotically stabilized to an arbitrary point by a smooth, time-invariant controller. With the above disadvantage, there has been a constant effort to design higher performance PID controllers for all possible parameter variations, such as the fuzzy PID controller has been proposed in [20], [21]; the controller is designed based on the sliding mode control method combined with PID; the artificial neural network-based adaptive law is introduced to model and estimate the various uncertainties disturbances [22]. These improvements have improved the accuracy and efficiency of traditional PID controllers, simultaneously significantly reducing the hardware's processing speed and capacity versus modern controllers. Which primarily ensures that the MWMR adheres to its predefined desired path is complex with the velocity is continuously changing. Also, to avoid sudden changes in the robot velocities, we use the non-uniform rational B-spline (NURBS) curve to design the simulation and experimental problem path to verify the efficiency of the controller designed by this study.

Compared to the other research, this paper focuses on designing a PID controller with time-varying parameters via the error function to meet the trajectory tracking of the MWMR with minor errors. The remainder of this paper is organized as follows: a time-varying PID controller to control the MWMR to track the NURBS trajectory with minor errors, while the robot linear and angular velocities are time-varying are presented in section 2. Section 3 presents the method of setting up the simulation and experimental parameters for the MWMR. Section 4 presents the designed controller installed on a practical MWMR platform to the NURBS trajectory tracking. Simulation and experimental results are presented to evaluate the usefulness of the designed controller. Finally, we discuss the advantages and limitations of the proposed controller and present our conclusions in section 5.

2. CONTROLLER DESIGN WITH TIME-VARYING PARAMETER

2.1. Kinematic model

Consider a MWMR moving along the trajectory ξ in the global coordinate $\boldsymbol{\vartheta_f}\{\boldsymbol{O_f x_f y_f}\}$, as shown in Figure 1(a). The relationship between the wheels angular velocities and the robot linear and angular velocities is determined by [7].

$$\omega = \mathrm{J}\mathrm{Q}^T(\phi)\dot{q} \quad (1)$$

where $J = \begin{bmatrix} 1/r & 1/r & -(L+d)/2r \\ 1/r & -1/r & (L+d)/2r \\ 1/r & 1/r & (L+d)/2r \\ 1/r & -1/r & -(L+d)/2r \end{bmatrix}$; $Q(\phi) = \begin{bmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix}$;

$\omega(t) = [\omega_1(t) \quad \omega_2(t) \quad \omega_3(t) \quad \omega_4(t)]^T$ and $\dot{q}(t) = [\dot{x}(t) \quad \dot{y}(t) \quad \dot{\phi}(t)]^T = [V_x \quad V_y \quad \Omega]^T$ with r, L, d are the wheel radius, the distance between the two wheels, the distance between the front and rear wheels, respectively. Meanwhile, V_x, V_y and Ω are the linear and angular velocities of the robot, while Q(φ) is the rotation matrix from the local coordinate $\vartheta_R\{Gx_R y_R\}$ (mounted on the robot) to respect the global coordinate ϑ_f. From (1), the forward kinematics equation of the MWMR is as shown in.

$$\dot{q} = Q(\phi)\dot{q}_R = Q(\phi)(J^T J)^{-1} J^T \omega \quad (2)$$

With $q = [x \quad y \quad \phi]^T$ is the vector of generalized coordinates in global coordinate ϑ_f.

2.2. Kinematic error model

Let $q_d(t) = [x_d(t) \quad y_d(t) \quad \phi_d(t)]^T$ be the desired trajectory ξ of the robot in the global coordinate ϑ_f. The vector $e = q_d(t) - q(t) = [e_x \quad e_y \quad e_\phi]^T$ describes the position and posture errors of the MWMR when moving along the desired trajectory ξ (see Figure 1(b)), then transform e from the global coordinate to the local coordinate. The error model can be written as shown in:

$$e_R = [e_{xR} \quad e_{yR} \quad e_{\phi R}]^T = Q^T(\phi)e \quad (3)$$

derivative (3), then combined with (2), results in a nonlinear error model of the MWMR.

$$\dot{e}_R = \tilde{\Omega} e_R + \mathrm{R}\dot{\mathrm{q}}_d - \dot{q} \quad (4)$$

Wherein $\tilde{\Omega} = \begin{bmatrix} 0 & \Omega(t) & 0 \\ -\Omega(t) & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$ and $R = \begin{bmatrix} cos\, e_\phi - sin\, e_\phi\, 0 \\ sin\, e_\phi\, cos\, e_\phi\, 0 \\ 001 \end{bmatrix}$

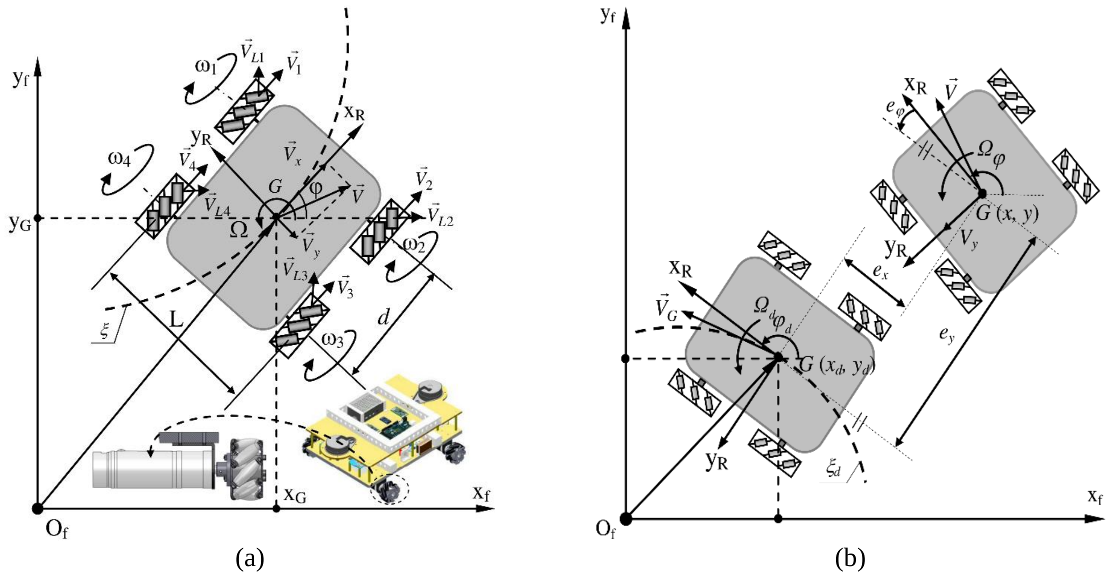


Figure 1. Schematic the kinematic relationship of the MWMR with (a) the robot geometry kinematic and (b) the robot position and posture errors

2.3. Controller design

The linear error model of the MWMR is established by linearizing (4) around the working point with the desired values $\phi_R(t) \approx \phi_{dR}(t)$ and $\Omega_R(t) \approx \Omega_{dR}(t)$. Thus, $(cos\, e_\phi \approx 1,\ sin\, e_\phi \approx e_\phi)$ then substituting into the (4) the linear error model in the local coordinate ϑ_R is given.

$$\dot{e}_R = \mathrm{Ae}_R + I_{3\times3} u \quad (5)$$

Where $A = \begin{bmatrix} 0 & \Omega(t) & -V_y(t) \\ -\Omega(t) & 0 & V_x(t) \\ 0 & 0 & 0 \end{bmatrix}$, $u = [e_{Vx} \quad e_{Vy} \quad e_\Omega]^T = [V_{xd} - V_x \quad V_{yd} - V_y \quad \Omega_d - \Omega]^T$ is the control variable vector and I_{3x3} is the unit matrix. The control law to reference trajectory tracking of the MWMR is designed as shown in [23]:

$$\begin{cases} e_{VxR} = K_P e_{xR} + K_I \int e_{xR} dt + K_D \frac{de_{xR}}{dt} \\ e_{VyR} = K_P e_{yR} + K_I \int e_{yR} dt + K_D \frac{de_{yR}}{dt} \\ e_\Omega = K_P e_{\phi R} + K_I \int e_{\phi R} dt + K_D \frac{de_{\phi R}}{dt} \end{cases} \quad (6)$$

with parameters K_P, K_I, K_D of the time-varying PID controller depend on the error function $e = \sqrt{e_x^2 + e_y^2}$ when the MWMR moves along the desired trajectory ξd is described by [24].

$$\begin{cases} K_P = a_1 + a_2 e \\ K_I = b_1 - b_2 e \\ K_D = c_1 + c_2 e \end{cases} \quad (7)$$

Here a_i, b_i, c_i are the control coefficients (i=1,2). Assume that the wheels and road surface are free of hysteresis and the robot has no longitudinal and transverse slip. Based on the kinematic error model and the control law established above, the trajectory tracking controller with the time-varying parameter is designed as shown in Figure 2. In which the transform block is given by (3), PID controller block is the control law is represented by (6), the kinematic block is described by (2).

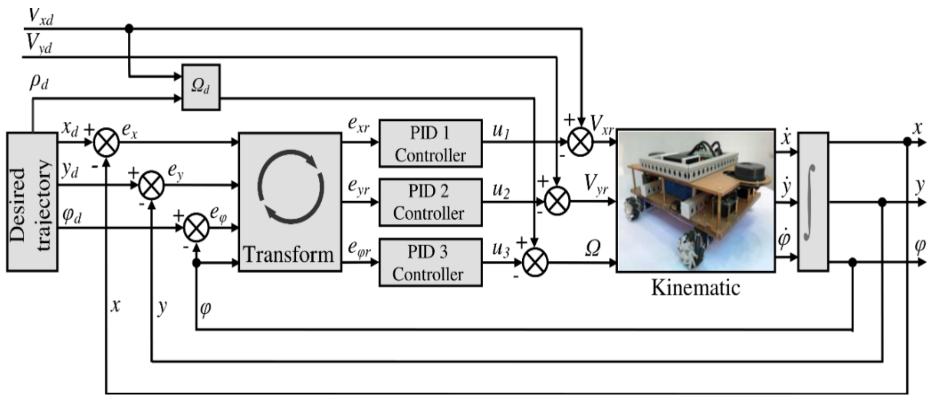


Figure 2. Block diagram of the time-varying parameter PID controller

3. SIMULATION AND EXPERIMENTAL SETUP

3.1. Simulation parameters setup

3.1.1. Dimensional of the robot

Length x width x height (380 mm x 260 mm x 165 mm), *L*=316 (mm), *d*=270 (mm), radius of wheels *r*=30 (mm).

3.1.2. Motion trajectory design

Nowadays, there have been many different solutions to design motion trajectories for mobile robots, as presented in the introduction. In this study, to generalize and smooth the motion trajectory to prevent the robot from having sudden speed changes at bends, we NURBS curve interpolation method to design the desired motion trajectory of the robot [25]. The dashed line in Figure 3 shows the desired motion trajectory of the designed robot after performing interpolation in MATLAB software with grid nodes A_i (i=1-14).

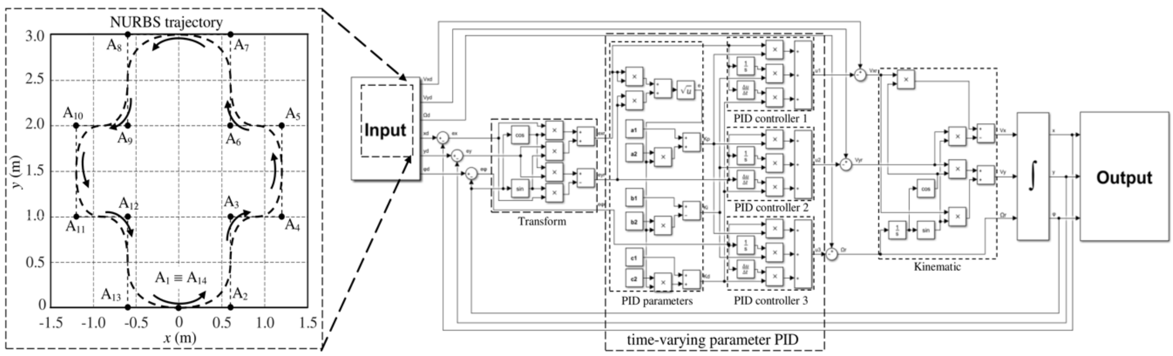


Figure 3. Simulation model for NURBS trajectory tracking of the MWMR in Matlab/Simulink

3.1.3. Determination of motion parameters

From the desired motion path ξd, the time-varying velocity $Vd(t)$ of the robot is determined by:

$$V_d(t) = \frac{\Delta S}{\Delta t} = \frac{\sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}}{t_i - t_{i-1}} \quad (8)$$

with V_{dmax}=0,25 m/s is maximal allowable velocity. Thus, the desired angular velocity Ωd of the robot when moving on curved arcs is given by:

$$\Omega_d(t) = \frac{V_d(t)}{\rho_d(t)} \quad (9)$$

in which, $\rho_i(t) \in [\rho max_{min}$ is the radii of curvature of the desired trajectory ξd moving with i=1,2. n and is defined:

$$\rho_i = \left|(\dot{x}_i{}^2 + \dot{y}_i{}^2)^{3/2}(\dot{x}_i\ddot{y}_i - \dot{y}_i\ddot{x}_i)^{-1}\right| \quad (10)$$

here $\dot{x}_i = \frac{\Delta x}{\Delta t} = \frac{x_i - x_{i-1}}{t_i - t_{i-1}}$, $\dot{y}_i = \frac{\Delta y}{\Delta t} = \frac{y_i - y_{i-1}}{t_i - t_{i-1}}$, $\ddot{x}_i = \frac{\Delta \dot{x}}{\Delta t} = \frac{\dot{x}_i - \dot{x}_{i-1}}{t_i - t_{i-1}}$, $\ddot{y}_i = \frac{\Delta \dot{y}}{\Delta t} = \frac{\dot{y}_i - \dot{y}_{i-1}}{t_i - t_{i-1}}$ with Δt=0.05 s.

3.1.4. Determine the coefficients of the controller

The trial and error technique to minimize error is applied to determine the coefficients a_i, b_i, c_i (i=1,2) by the simulation model as shown in Figure 3. Table 1 is the data of the controller coefficients that have been selected to minimize the NURBS trajectory tracking error. The data in Table 1 and Figure 4 describe the time-varying parameters of the PID controller for the MWMR to follow the NURBS trajectory with minimal error.

Table 1. The coefficients of the controller

a_1	a_2	b_1	b_2	c_1	c_2
0.001	0.001	0.001	0.581	0.001	0.001

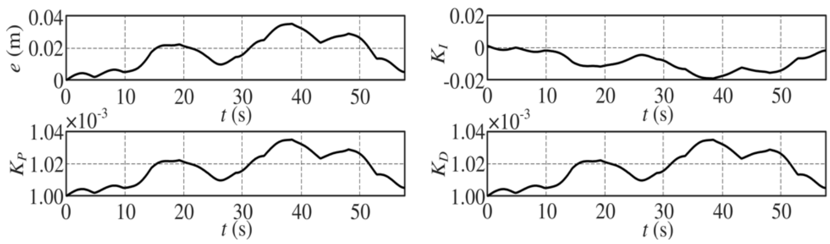


Figure 4. Time-varying parameter of the PID controller according to the error function e

3.2. Experimental setup

3.2.1. Hardware architecture

The hardware architecture is distributed over two frames, as shown in Figure 5, where each frame is 5 mm thick mica to ensure lightness and rigidity. In which i) the lower frame includes the following devices: 1) drive: four mecanum wheels connected to integrated DC servo motor with integrated planetary gear reducer with gear ratio 1:14 and encoder 500 counts per revolution (CPR) (2 channels A-B, 500 pulses); 2) four 43 A H-bridge motor driver modules (BTS7960); and 3) the robot is electrically powered by four Lithium batteries Lifepo4 33140 (3.2 V-15 Ah), which enable it to operate for about 5 h; ii) the upper frame includes the following devices: 1) the higher-level controller is an embedded computer (Raspberry Pi 4 model b) for multi-sensor data processing; 2) the lower-level controller is a module Arduino (ATmega 2560 R3) for motion control; and 3) the INS and obstacle detection system includes a Waveshare 6 degrees of the freedom (DOF) inertial measurement unit (IMU) sensor (SEN0386) and two LiDAR delta 2 A positioning scanners; iii) the upper frame includes the following devices: 1) the central processor is an embedded computer Raspberry Pi 4 model b; 2) control module Arduino ATmega 2560 R3; and 3) the INS and obstacle detection system includes a Waveshare 6 DOF IMU sensor (SEN0386) and two LiDAR delta 2 A positioning scanners; iv) communication: connect between host and robot by module WiFi integrated with an embedded computer; and v) robot control software is developed on ROS operating system.

The experimental design in Figure 5 is made based on the calculated data according to Figure 3. Practical measurement data position (x_G, y_G) and robot posture angle (φ) are determined through IMU and encoder. Meanwhile, the observed angular velocity data of the four wheels is determined by the encoder integrated with the motors.

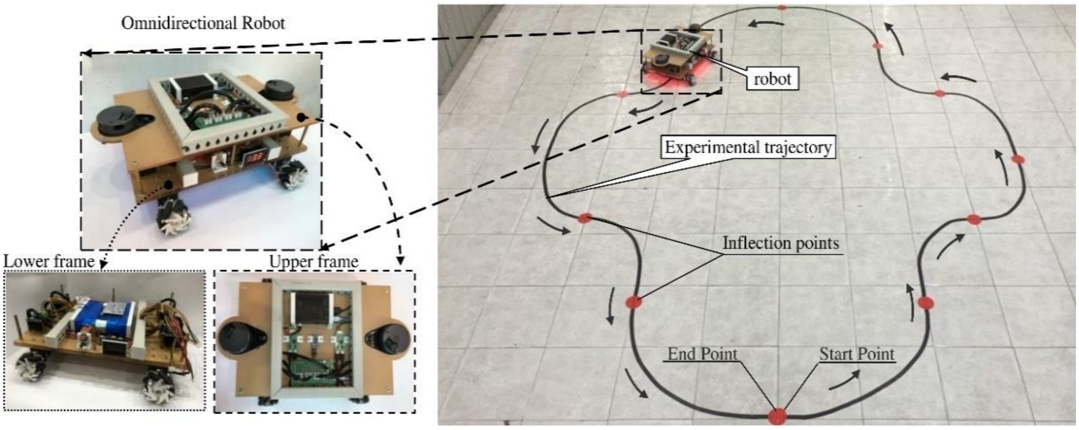
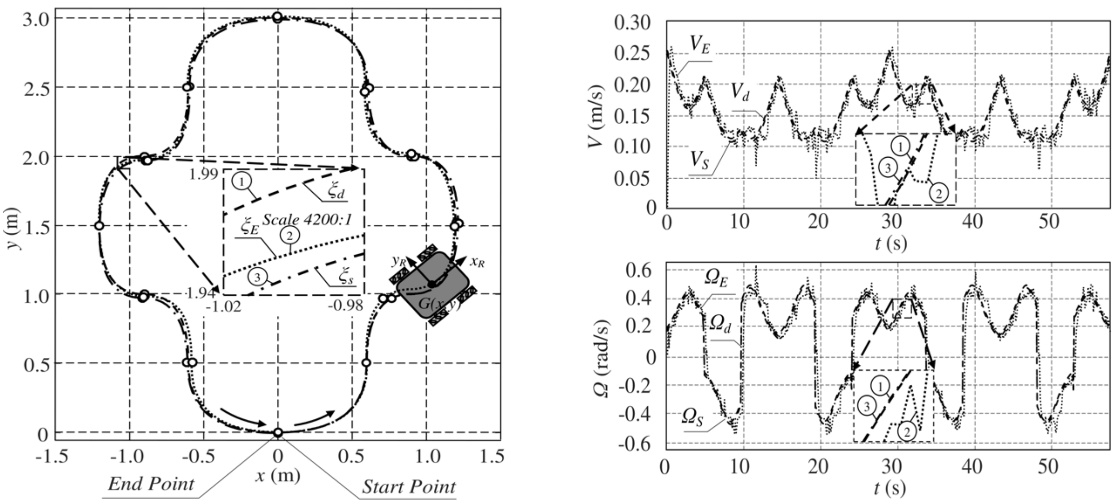


Figure 5. NURBS trajectory experiment setup for the robot

4. RESULTS AND DISCUSSION

To illustrate the application and relevance of the time-varying parameter PID controller, we are going to compare the simulation and experiment results of the MWMR with the setting parameters in section 3. Figure 6(a) shows the motion trajectory followed by the robot includes desired, simulated and measured values includes desired, simulated and measured values. Here, the line no. 1 is the desired value, the line no. 2 is the measured value and the line no. 3 is the simulated value. Figure 6(b) presents the desired, simulated and measured values of linear and angular velocities. It is clear that the measured and simulated robot velocities are closed to the desired ones.



(a) (b)

Figure 6. Simulation and experiment results of MWMR obtained from time varying paremeter PID controller (a) the desired, simulated and measured trajectory and (b) the desired, simulated, and measured values of linear and angular velocities

From Figure 6, it can be seen that the trajectory tracking error between measured and desired values: the maximum position error e_{max} is less than 5.5 cm at t=53 s, and the maximum posture error $e_{\varphi max}$ is approximately 10^o at t=49 s. The cause of the errors between the measured and the desired values are the variation of the linear and angular velocities of the MWMR at the inflection points, as shown in Figure 6. In addition, there is slippage between the wheel and the road surface, and the problem of robot dynamics has not been considered. In contrast, the trajectory tracking error between the simulated and desired values is minor: the position error e_{max} is less than 2 cm at t=38 s, while the maximum posture error $e_{\varphi max}$ is approximately 1.5^o at t=53 s. Figure 7 makes this more clear, in which Figure 7(a) is the error of linear and angular velocities between simulated and desired values with e_{Vmax}=0.00185 m/s at t=48 s and $e_{\Omega max}$=0.0025 rad/s at

t=19 s. Figure 7(b) is the error of linear and angular velocities between measurement and simulation with e_{Vmax}=0.07 m/s at t=19 s and $e_{\Omega max}$=0.8 rad/s at t=38 s. Figure 7(c) describes the robot position and posture errors during the whole process of trajectory tracking.

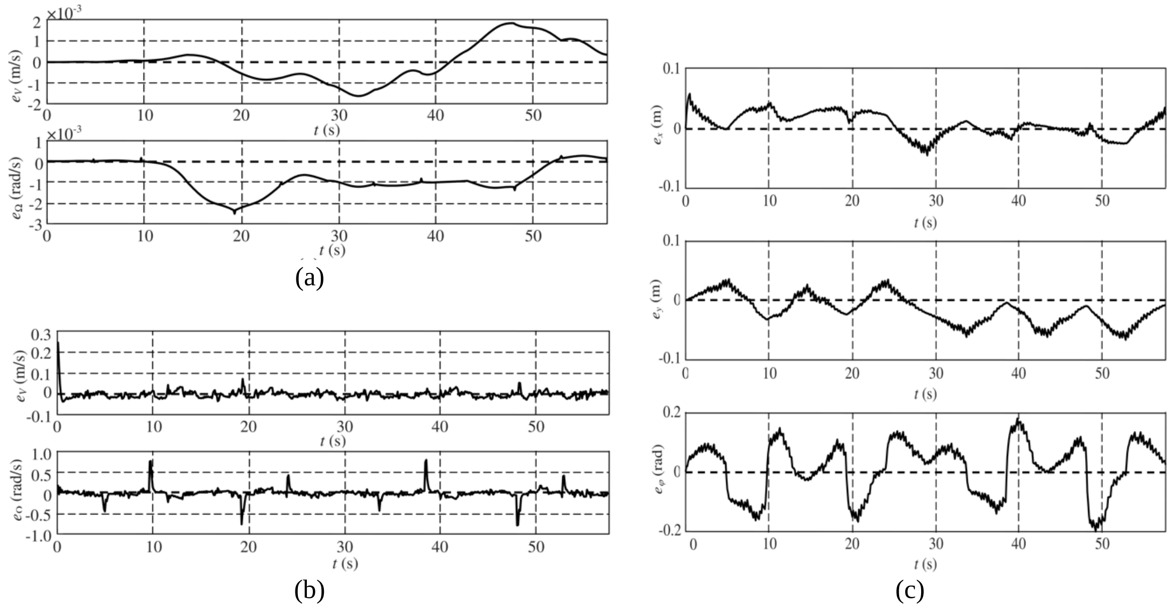


Figure 7. Evolution of the MWMR errors (a) the velocity errors between simulated and desired values, (b) the velocity errors between measured and simulated values, and (c) the position and posture errors between measured and simulation

Figures 6(b), 7(b) and 7(c) show that the measured and simulated values of the robot linear and angular velocities are always around the desired value and symmetrically in two arcs: the first arc: 0 -> 7 (t=0 s to t=30 s) and the second arc 7 -> 0 (t=30 s to t=60 s). The experimental and simulation results versus the desired values in Figures 6 and 7 verified that the robot tracks the desired trajectory with minor errors. That shows the simplicity and efficiency of the PID controller designed with the time-varying parameters proposed by this study. Furthermore, this proposal has overcome the disadvantages of some previous studies when using the time-invariant parameter PID controller to control the mobile robot, such as: i) ineffective system control because the accuracy is not high; ii) when the robot moves on a complex trajectory with variable linear and angular velocities cannot be stabilized asymptotically to an arbitrary point; and iii) the traditional PID controller must be combined with another complex modern controller to satisfy the system. The speed of the four wheels, as shown in Figure 8, illustrates that the proposed controller has the efficiency to control for the robot to perform the above motion since measured and simulated values are always around the desired value.

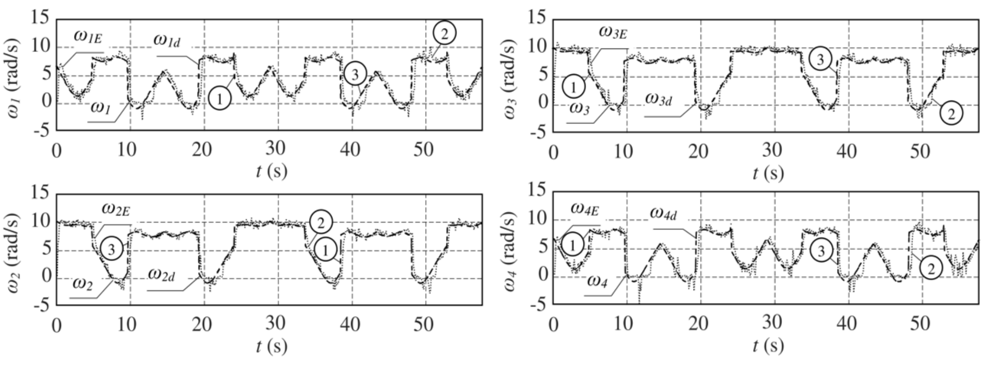


Figure 8. Compare the angular velocities of the four wheels

5. CONCLUSION

From the above results of calculation, simulation, experiment, and discussions this paper achieves the following points: i) a time-varying PID controller based on a function of tracking error has been developed for the MWMR to complex trajectory tracking. This controller with a simple structure but effectively improves performance with a minor error. The performance of the proposed controller is verified via simulations and experiments on the platform MWMR. Therefore, this is the difference between this study versus previous studies; ii) we believe that the investigation method to determine the time-varying coefficients of the PID controller with variable set value in this study will yield guarantees on the tracking performance for the MWMR. Similarly, can apply research results of this paper for any desired trajectory with minimal control action and perfect orientation of the MWMR. Also, the problems of dynamic control, working load, electromechanical interaction between the robot and the environment, and friction between wheel and road surface, to improve and improve the accuracy of the robot will be considered part of our future research goals.

ACKNOWLEDGEMENTS

This research was funded by the Ministry of Industry and Trade in a ministeriallevel scientific and technological research project, conducted in 2020, code: DTKHCN.076/20.

REFERENCES

[1] 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, p. 2073, Sep. 2017, doi: 10.3390/s17092073.

[2] T. T. K. Ly, N. H. Thai, L. Q. Dzung, and N. T. Thanh, "Determination of Kinematic Control Parameters of Omnidirectional AGV Robot with Mecanum Wheels Track the Reference Trajectory and Velocity," in *International Conference on Engineering Research and Applications*, vol. 178, pp. 319–328, Nov. 2020, doi: 10.1007/978-3-030-64719-3_36.

[3] D. Mišković, L. Milić, A. Čilag, T. Berisavljević, A. Gottscheber, and M. Raković, "Implementation of Robots Integration in Scaled Laboratory Environment for Factory Automation," *Applied Sciences*, vol. 12, no. 3, p. 1228, Jan. 2022, doi: 10.3390/app12031228.

[4] B. Tao, X. Zhao, S. Yan, and H. Ding, "Kinematic modeling and control of mobile robot for large-scale workpiece machining," *Part B: Journal of Engineering Manufacture*, vol. 236, no. 1-2, pp. 29-38, Jul. 2020, doi: 10.1177/0954405420933708.

[5] C. P. Sesmero, L. R. Buonocore, and M. Di Castro, "Omnidirectional Robotic Platform for Surveillance of Particle Accelerator Environments with Limited Space Areas," *Applied Sciences*, vol. 11, no. 14, p. 6631, Jul. 2021, doi: 10.3390/app11146631.

[6] Iswanto *et al*., "PID-based with Odometry for Trajectory Tracking Control on Four-wheel Omnidirectional Covid-19 Aromatherapy Robot," *Emerging Science Journal*, vol. 5, pp. 157-181, 2021, doi: 10.28991/esj-2021-SPER-13.

[7] N. H. Thai, T. T. K. Ly, N. T. Long, and L. Q. Dzung, "Trajectory Tracking Using Linear State Feedback Controller for a Mecanum Wheel Omnidirectional," in *IFToMM Asian conference on Mechanism and Machine Science. Advances in Asian Mechanism and Machine Science*, vol. 113, pp. 411-421, 2022, doi: 10.1007/978-3-030-91892-7_39.

[8] 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 *International Conference on Engineering Research and Applications*, vol. 366, pp. 685-696, Dec. 2021, doi: 10.1007/978-3-030-92574-1_71.

[9] M. R. R. Khoygani, R. Ghasemi, and P. Ghayoomi, "Robust Observer-based Control of Nonlinear Multi-Omnidirectional Wheeled Robot Systems via High Order Sliding-mode Consensus Protocol," *International Journal of Automation and Computing*, vol. 18, no. 5, pp. 787–801, Feb. 2021, pp. 787-801, doi:10.1007/s11633-020-1254-z.

[10] 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 and Electrical Engineering*, vol. 96, p. 107500, Dec. 2021, doi: 10.1016/j.compeleceng.2021.107500.

[11] D. Wang *et al.*, "Formation control of multiple mecanum-wheeled mobile robots with physical constraints and uncertainties," *Applied Intelligence*, vol. 52, pp. 2510–2529, 2022, doi: 10.1007/s10489-021-02459-3.

[12] M. Muhammad, A. A. Bature, U. Zangina, S. Buyamin, A. Ahmad, and M. A. Shamsudin, "Velocity control of a two-wheeled inverted pendulum mobile robot: a fuzzy model-based approach," *Bulletin of Electrical Engineering and Informatics*, vol. 8, no. 3, pp. 808-817, Sep. 2019, doi: 10.11591/eei.v8i3.1594.

[13] M. S. Masmoudi, N. Krichen, M. Masmoudi, and N. Derbel, "Fuzzy logic controllers design for omnidirectional mobile robot navigation," *Applied Soft Computing*, vol. 49, pp. 901–919, Dec. 2016, doi: 10.1016/j.asoc.2016.08.057.

[14] A. M. Alshorman, O. Alshorman, M. Irfan, A. Glowacz, F. Muhammad, and W. Caesarendra, "Fuzzy-Based Fault-Tolerant Control for Omnidirectional Mobile Robot," *Machines*, vol. 8, no. 3, p. 55, Sep. 2020, doi:10.3390/machines8030055.

[15] S. Harun and M. F. Ibrahim, "A genetic algorithm based task scheduling system for logisticsservice robots," *Bulletin of Electrical Engineering and Informatics*, vol. 8, no. 1, pp. 206-213, Mar. 2019, doi: 10.11591/eei.v8i1.1437.

[16] Z. Sun, H. Xie, J. Zheng, Z. Man, and D. He, "Path-following control of Mecanum-wheels omnidirectional mobile robots using nonsingular terminal sliding mode," *Mechanical Systems and Signal Processing*, vol. 147, p. 107128, Jan. 2021, doi: 10.1016/j.ymssp.2020.107128.

[17] X. Wu and Y. Huang, "Adaptive fractional-order non-singular terminal sliding mode control based on fuzzy wavelet neural networks for omnidirectional mobile robot manipulator," *ISA transactions*, vol 121, pp 258-267, Feb. 2022, doi: 10.1016/j.isatra.2021.03.035.

[18] M. Korkmaz, Ö. Aydoğdu, and H. Doğan, "Design and performance comparison of variable parameter nonlinear PID controller and genetic algorithm based PID controller," *2012 International Symposium on Innovations in Intelligent Systems and Applications*, 2012, pp. 1-5, doi: 10.1109/INISTA.2012.6246935.

[19] O. Aydogdu and M. Korkmaz, "A Simple Approach to Design of Variable Parameter Nonlinear PID Controller," *International Conference on Advancements in Information Technology*, vol. 20, pp. 81-85, 2011,

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

[21] Q. Jia, C. Chang, S. Liu, L. Zhang, and S. Zhang, "Motion Control of Omnidirectional Mobile Robot Based on Fuzzy PID," *2019 Chinese Control and Decision Conference (CCDC)*, 2019, pp. 5149-5154, doi: 10.1109/CCDC.2019.8833047.

[22] 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, doi: 10.1016/j.isatra.2018.10.043.

[23] S. G. Tzafestas, *Introduction to mobile robot control*. Elsevier, 2013.

[24] G. Klancar, A. Zdesar, S. Blazic, and I. Skrjanc, *Wheeled mobile robotics: from fundamentals towards autonomous systems*. Butterworth-Heinemann, 2017.

[25] D. F. Rogers, *An introduction to NURBS: with historical perspective*. Morgan kaufmann, 2001.

BIOGRAPHIES OF AUTHORS

Nguyen Hong Thai received a B.E. in 1999 at Hanoi University of Science and Technology, Vietnam. He received an M.Sc. in 2002 and Dr Eng. in 2010 in Hanoi University of Science and Technology, Vietnam. Since 2000, he has been employed as a lecturer at the Department of Design of Machines and Robots, School of Mechanical Engineering, Hanoi University of Science and Technology. His field of research are machine design, dynamics and control of mechatronic systems, theory of gearing. He can be contacted at email: thai.nguyenhong@hust.edu.vn.

Trinh Thi Khanh Ly received a B.E. in 2002 at Hanoi University of Science and Technology. She received the M.Sc degree in Instrument and control and the Ph.D. degree in Control Engineering and Automation from Hanoi University of Science and Technology, in 2004 and 2017, respectively. Currently, she is a lecturer at the Faculty of Automation Technology, Electric Power University in Hanoi, Vietnam. Her research interests include modelling, identification, optimazition and control, dynamics and control of mechatronic systems. She can be contacted at email: lyttk@epu.edu.vn.

Le Quoc Dzung received a B.E in 2007 at Hanoi University of Science and Technology, Vietnam. He received an M.S in 2009 at Hanoi University of Science and Technology, Vietnam. Since 2007, he has been employed as a lecturer at Faculty of Control and Automation, Electric Power University, Vietnam. His field of research is robotics, power electronics, electrical drives, modeling, and simulation. He can be contacted at email: dunglq2016@gmail.com.