Adaptive Control For Mobile Robots Based On Inteligent Controller

Than Thi Thuong[1] and Vo Thanh Ha[2*]

[1] *Faculty of Electrical Engineering, University of Economics - Technology for Industries, Vietnam*
[2] *Faculty of Electrical and Electronic Engineering, University of Transport and Communications, Vietnam*
[*] *Corresponding author. E-mail: vothanhha.ktd@utc.edu.vn*

Received: Dec. 18, 2022; Accepted: Aug. 19, 2023

The paper presents three position controller designs for a mobile robot. The first is a position controller using a classic PID controller. The second is the position controller is designed based on optimal three coefficients for PID controller by fuzzy logic control (FLC). The last, the mobile robot is moved according to the trajectories set by the FLC controller. All three controllers have two state variables (position error and position deviation derivative and one output variable, velocity) and one velocity output variable of the robot. The robot is moved according to the trajectories set based on the PID-FLC controller flow fuzzy rules with a $7x7$ matrix to the optimal three coefficients of the PID controller. Meanwhile, the FLC controller is done by a 9x9 matrix rule. Evaluated the efficiency of PID-FLC and FLC controllers are compared to classical PID controllers. The correctness of the three controllers is proven through MATLAB/Simulink simulation. The PID-FLC controller has the result better than the other two controllers.

Keywords: Mobile Robot, PID, Fuzzy Logic Control, FLC, PID-Fuzzy

http://dx.doi.org/10.6180/jase.202405_27(5).0012

1. Introduction

The mobile robot is an innovative solution for the future in the era of digitization and industry 4.0. The Self-propelled robot ensures the certainty and flexibility of the product. At the same time, it makes it easier to move goods inside factories and warehouses. Besides, Autonomous Robots also improve automation and solve production continuity problems [1, 2]. In the world, in recent decades, autonomous robot control has received extensive research and development attention, and many methods, from classical control to modern management, have been proposed to apply to self-propelled robots. Previously, most of the publications used the structure of two control loops: the outer kinematic loop uses the Lyapunov function to synthesize the position tracking controller, and the dynamic inner circle controls the speed tracking. Many dynamic loop control methods have been proposed, such as slip control [3–6] and backstepping control [7–9]. When the emotional equation has uncertain parameters, adaptive management is included in the design [10–13], and adaptive control combines with neurons to approximate the result. Unpredictable parts [14–16] or adaptive control combined with fuzzy logic [17–20] gave reasonable control quality, compensating for model error and system input noise.

Although many advanced controllers have been researched and developed, traditional PID controllers are still chosen to be used in the problem of controlling orbital self-propelled robots because of their effectiveness of the controller. This ensures stability and traction. However, the accuracy achieved is not high [21]. Another problem affecting the accuracy of orbital tracking for self-propelled robots is the robot's parameters, such as weight, cargo volume, and motion. Moreover, wheel and environmental friction can change and affect the operation of the whole system, where the PID controller no longer maintains traction control. Therefore, this paper presents the analysis, comparison, and evaluation of Fuzzy control algorithms and the Fuzzy-PID auto-tuning algorithm to find the optimal Kp, Kd, and Ki values to compare with classical PID controller [22, 23] to get better. The PID controller has the advantage of a simple design, but the system is not stable when the noise affecting the system is always changing while the controller parameters are fixed. Independent fuzzy will overcome the system noise change, but it is very difficult to determine the fuzzy rule to eliminate noise compared to the PID set. From that, the author group came up with the idea of designing a Fuzzy-PID controller, taking a Fuzzy set to adjust parameters for PID, to ensure stable performance of the closed-loop system. Research methods are tested by MATLAB/Simulink software and experiment. The PID controller on the self-propelled robot model controls the movement along a predetermined line.

This paper is organized into six main parts. Part 1 and part 2 present the introduction to the target study and kinematics and dynamics model. The fuzzy logic controller is designed in section 3. Part 4 expressed PID-Fuzzy logic controller design Part 5 the simulation and simulation results. The last section is the conclusion.

2. Kinematic and dynamic model

2.1. Kinematic Model

In the plane of the moving medium attach a fixed frame of reference as depicted in Fig. 1.

The equation describing the kinematics of the mobile robot is expressed in Eq. (1) [1].

$$\dot{q} = \begin{bmatrix} \frac{r}{2}\cos(\theta) & \frac{r}{2}\,\mathrm{d}\cos(\theta) \\ \frac{r}{2}\sin(\theta) & \frac{r}{2}\,\mathrm{d}\sin(\theta) \\ \frac{r}{2a} & \frac{r}{2a} \end{bmatrix} \begin{bmatrix} \dot{\varphi}_r \\ \dot{\varphi}_l \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} \tag{1}$$

Where: r is right and left wheel radius; $2a$ is distance between the actuated wheels and the symmetry axis; $\dot{\varphi}_r, \dot{\varphi}_l$ are angular velocity of the right and left wheels; v, ω are Angular velocity of the right and left wheels; q is linear and angular velocities of robot; θ is orientation angle; $\dot{q}$ is robot speed.

2.2. Dynamic Model

The kinetic energy of the self-propelled robot is calculated by:

$$T_c = \frac{1}{2}m_c\vartheta_c^2 + \frac{1}{2}I_c\dot{\theta}^2 \tag{2}$$

$$T_{\omega R} = \frac{1}{2}m_\omega\vartheta_\omega^2 + \frac{1}{2}I_m\dot{\theta}^2 + \frac{1}{2}I_\omega\dot{\varphi}_r^2 \tag{3}$$

$$T_{\omega L} = \frac{1}{2}m_\omega\vartheta_\omega^2 + \frac{1}{2}I_m\dot{\theta}^2 + \frac{1}{2}I_\omega\dot{\varphi}_l^2 \tag{4}$$

Where: T_c is the kinetic energy of the DWMR without the wheels, $T_{\omega R}$ is the kinetic energy of the actuated wheels in the plane and $T_{\omega L}$ is the kinetic energy of all the wheels considering the orthogonal plane ; m_c is mass of the robot without wheels and motors; m_ω is mass of each wheel and motor assembly; m_t is total mass of the DWMR; I_c is moment of inertia of the DWMR without wheels and motors about the vertical axis through P; I_w is Moment of inertia of each wheel and motor about the wheel axis; I is Total inertia moment of the robot; $\dot{\varphi}_r, \dot{\varphi}_l$ are angular velocity of the right and left wheels; v, ω are Angular velocity of the right and left wheels; θ is orientation angle.

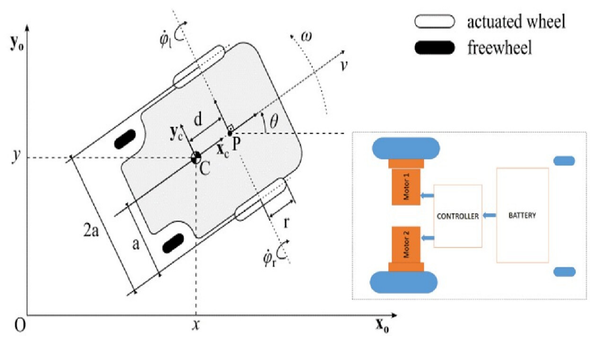

Fig. 1. Kinetic relationship of mobile robots. Where: P is Intersection of the symmetry axis with the axis of the wheels; C is mass center or guidance point; d is distance between C and P; r is right and left wheel radius; $2a$ is distance between the actuated wheels and the symmetry axis; m_c is mass of the robot without wheels and motors; m_ω is mass of each wheel and motor assembly; m_t is total mass of the DWMR; m_t is a moment of inertia of the DWMR without wheels and motors about the vertical axis through P; I_c is moment of inertia of the DWMR without wheels and motors about the vertical axis through P; I_w is Moment of inertia of each wheel and motor about the wheel axis; I is Total inertia moment of the robot; $\dot{\varphi}_r, \dot{\varphi}_l$ are angular velocity of the right and left wheels; v, ω are Angular velocity of the right and left wheels; q is linear and angular velocities of robot; θ is orientation angle.

Mobile robot speed is calculated by:

$$\vartheta_i^2 = \dot{x}_i^2 + \dot{y}_i^2 \tag{5}$$

The coordinates of the wheels are therefore determined as follows:

$$\begin{cases} x_{\omega r} = x + a\sin\theta \\ y_{\omega r} = y + a\cos\theta \end{cases} \tag{6}$$

$$\begin{cases} x_{\omega l} = x - a\sin\theta \\ y_{\omega l} = y + a\cos\theta \end{cases} \tag{7}$$

From Eq. (2)to Eq. (7), the total kinetic energy:

$$T = \frac{1}{2} m_t \left(\left(\dot{x}^2 + \dot{y}^2 \right) - \dot{y} d\dot{\theta} \cos(\theta) + m_t \dot{x} d\dot{\theta} \sin(\theta) \right) + \frac{1}{2} I\dot{\theta}^2 + \frac{1}{2} I_w \left(\dot{\varphi}_r^2 + \dot{\varphi}_l^2 \right) \tag{8}$$

Where: $m_t = m_c + 2m_\omega; I = m_c d^2 + I_c + 2m_\omega \left(d^2 + a^2\right) + 2I_m$ and $\dot{\theta} = \omega$

The robot's equation of motion is described by the system of equations:

$$\begin{cases} m\ddot{x} - m_c d\ddot{\theta} \sin\theta - md\dot{\theta}^2 \cos\theta = F_1 - C_1 \\ m\ddot{y} - m_c d\ddot{\theta} \cos\theta - m_c d\dot{\theta}^2 \sin\theta = F_2 - C_2 \\ -m_c d \sin\theta \ddot{x} + m_c d \cos\theta \ddot{y} + I\ddot{\theta} = F_3 - C_3 \\ I_\omega \ddot{\varphi}_r = \tau_r - C_4 \\ I_\omega \ddot{\varphi}_l = \tau_l - C_5 \end{cases} \tag{9}$$

The matrix linking the kinematic constraints:

$$\Lambda^T(\mathrm{q}) = \begin{bmatrix} C_1 \\ C_2 \\ C_3 \\ C_4 \\ C_5 \end{bmatrix} \tag{10}$$

From Eqs. (9) and (10) The motion of the robot can be represented by the equation:

$$M(q)\ddot{q} + V(q,\dot{q}) + F(\dot{q}) + G(q) + \tau_d = B(q)\tau - \Lambda^T(q)^\lambda \tag{11}$$

Where: $M(q)$ is positive inertia matrix; $V(q,\dot{q})$ is centripetal Matrix; $F(\dot{q})$ is surface friction; $G(q)$ is gravity acceleration matrix; τ_d is noise component; $B(q)$ is input matrix; $\Lambda^T(q)$ is binding matrix; λ is Lagrange multiplier vector

2.3. Kinematic Model

The kinematic error model q_e of a self-propelled robot is a mathematical equation describing the deviation of the robot's position and posture, when the motion-controlled robot follows a desired trajectory ξd and is defined as following in the original coordinate system:

$$q_e = \begin{bmatrix} x_e \\ y_e \\ \theta_e \end{bmatrix} = R(\theta)\left(q_r - q\right) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_r - x \\ y_r - y \\ \theta_r - \theta \end{bmatrix} \tag{12}$$

The derivative of Eq. (11) combined with the kinematic equation of the mobile robot Eq. (1). The system of error function equations as follows:

$$\dot{q}_e = \begin{bmatrix} \dot{x}_e \\ \dot{y}_e \\ \dot{\theta}_e \end{bmatrix} = \begin{bmatrix} \cos(\theta_e) & 0 \\ \sin(\theta_e) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \vartheta_r \\ \omega_r \end{bmatrix} + \begin{bmatrix} -1 & y_e \\ 0 & -x_e \\ 0 & -1 \end{bmatrix} \begin{bmatrix} \vartheta \\ \omega \end{bmatrix} \tag{13}$$

3. Pid - fuzzy logic controller design

Fuzzy controller structure diagram for robot is showed at Fig. 2. A fuzzy logic controller is used to set parameters for the PID controller. And the PID controller will control the robot. The input to the fuzzy controller is still the model error and the model error derivative, but the output here is the coefficient for the PID controller. Use the normalized formula for all 3 parameters, K_p, K_I, K_D, assuming all 3 parameters are bounded:

$$K_P^{\min} \le K_p \le K_P^{\max}; K_I^{\min} \le K_I \le K_I^{\max}; K_D^{\min} \le K_D \le K_D^{\max}$$
$$K_P = \frac{K_p - K_P^{\min}}{K_P^{\max} - K_P^{\min}}; K_I = \frac{K_I - K_I^{\min}}{K_I^{\max} - K_I^{\min}}; K_D = \frac{K_D - K_D^{\min}}{K_D^{\max} - K_D^{\min}} \tag{14}$$

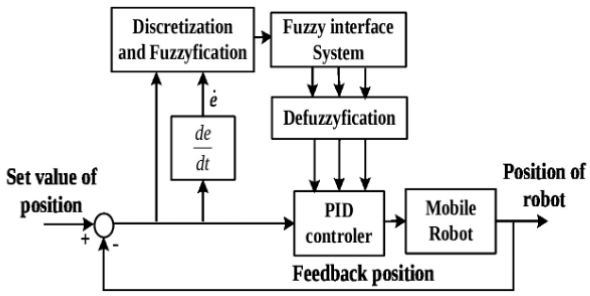

Fig. 2. Fuzzy controller structure diagram for robot

Then the Mallesham - Rajani fuzzy tuner will be calculated by blurring two inputs $(e, de/dt)$ and three outputs (K_p, K_I, K_D) (see Figs. 3 and 4 and Table 1).

4. Fuzzy logic controller design

Using a preprocessor, the inputs that were in the form of crisp values generated from feedback error (e) and change of error (de) were conditioned in terms of multiplying by constant gains before entering into the main control block, such as in Figs. 5 and 6. From the rule based commands, the Mamdani-type inference engine determined the capability of degree of employed rules and returned a fuzzy set for defuzzification block where the fuzzy output data were taken and crisp values were returned. The transformed fuzzy block matches the data with the conditions of the given

Table 1. Fuzzy rule table for KP/KI/KD

e/ce	NB	NS	**Z**	PS	PB
NB	VB/S/S	VB/M/S	S/M/VB	S/M/B	M/S/S
NS	VB/B/S	B/B/S	S/B/B	S/B/B	B/B/S
Z	B/B/S	M/B/M	S/VB/B	S/VB/M	VB/VB/S
PS	M/B/M	S/B/B	S/B/B	M/B/S	VB/B/S
PB	S/S/M	S/S/VB	S/M/VB	B/M/S	VB/S/S

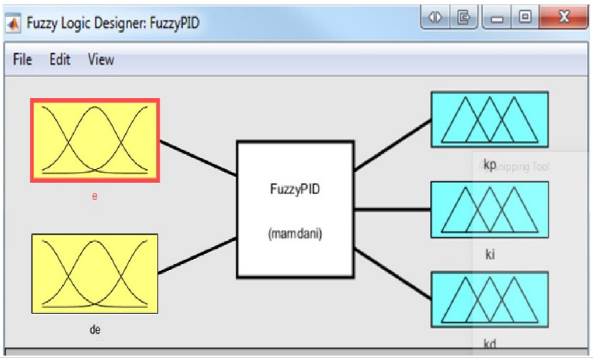

Fig. 3. Structure of the PID - Fuzzy logic controller on MATLAB/Simulink

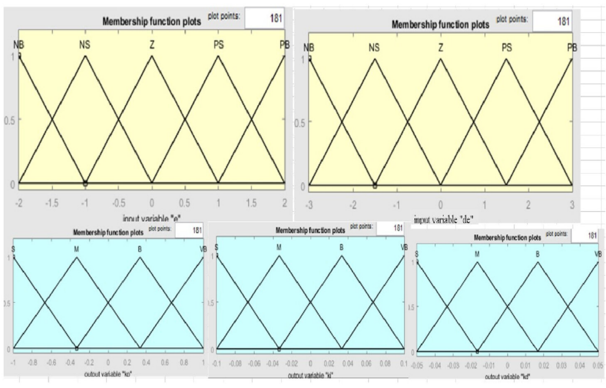

Fig. 4. Structure of the PID - Fuzzy logic controller on MATLAB/Simulink

fuzzy rule. The output of the fuzzy set is converted to the clarity values through the centroid defuzzification method and converted into a control signal, as in Fig. 7. The FLC controller is controlled by rules and makes control efforts based on several if-then statements about (e) and (de), i.e., if the error is equal to Negative Big (NB) and the change of error is equal to negative medium (NM), then the change in control (c) is positive big (PB). The numbers of these if-then statements were determined based on the experiment and tuning of the system. Plots of fuzzy logic membership function for the two inputs variables (e) and (de) and the output (c) are shown in Figs. 5 to 7, and correspond to the rules of Table 2.

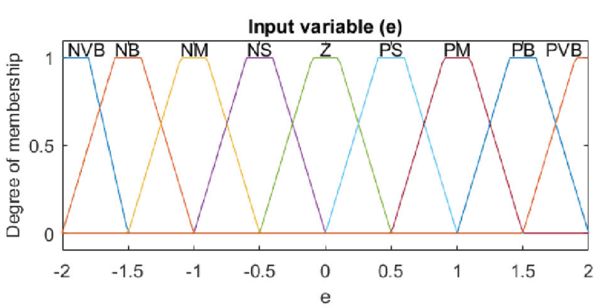

Fig. 5. Input of bias variable *e*

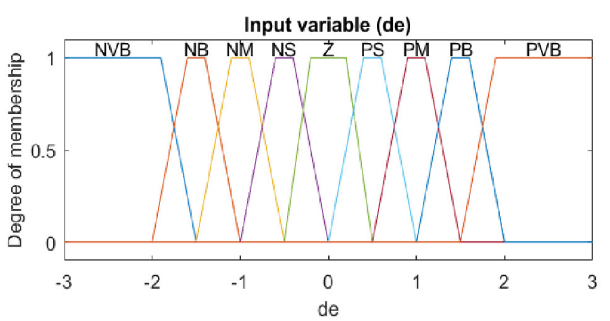

Fig. 6. Deviated variable derivative input *de*

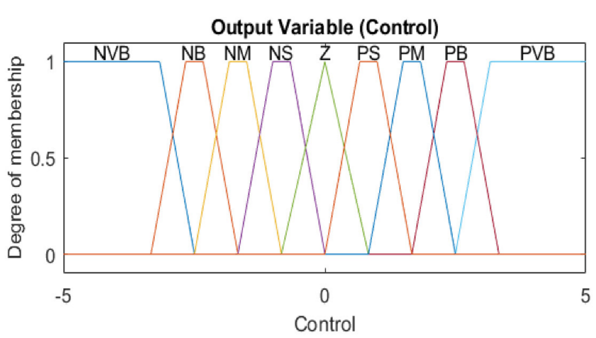

Fig. 7. Output variable

Table 2. The rule control for FLC controller

De/e	NVB	NB	NM	NS	Z	PS	PM	PB	PVB
NVB	PVB	PVB	PVB	PB	PM	PM	PS	Z	Z
NB	PVB	PVB	PB	PM	PS	PS	PS	Z	Z
NM	PVB	PB	PM	PS	PS	Z	Z	Z	NS
NS	PB	PM	PM	PS	PS	Z	Z	NS	NS
Z	PM	PM	PS	Z	Z	Z	NS	NS	NM
PS	PM	PS	PS	Z	NS	NS	NM	NM	NB
PM	PS	PS	Z	NS	NS	NM	NB	NB	NB
PB	PS	Z	Z	NS	NM	NM	NB	NVB	NVB
PVB	Z	Z	NS	NM	NM	NB	NB	NVB	NVB

5. Simulation results on matlab/simulink

The control structure of the self-propelled robot is expressed in Fig. 8.

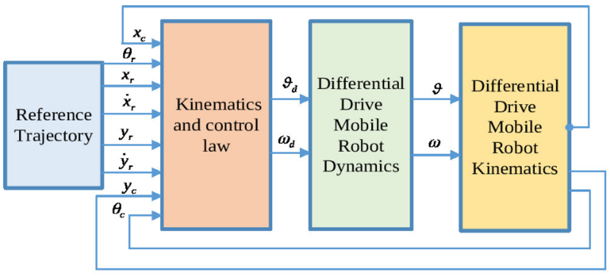

Fig. 8. The control structure of self-propelled robot

The PID - Fuzzy controller is compared with FLC and PID controllers. The parameters of the PID set are determined through the tuning simulation method on MATLAB/Simulink as $K_p = 0.7; K_I = 0.6; K_D = 0.01$.

Case 1: the trajectory is a circular orbit with radius 1 , center is origin. The results of the three controllers when the robot follows the same circular trajectory and the simulated response is shown in Figs. 9 to 11.

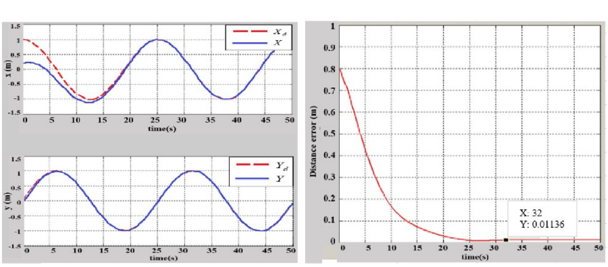

Fig. 9. x, y position and system error using PID controller

The results of the 3 controllers that respond to the simulation are shown in Figs. 9 to 11. From these figures, it is shown that all 3 controllers satisfy the stability for the preset trajectory tracking robot. However, the error of robot position for control method using PID controller is the largest (0.01) with response time of 32 s, then to fuzzy logic controller (0.006) with response time of 25 s, and the error is at least almost zero when using the PID - Fuzzy controller (0.004) and the response time is also very fast 10 s.

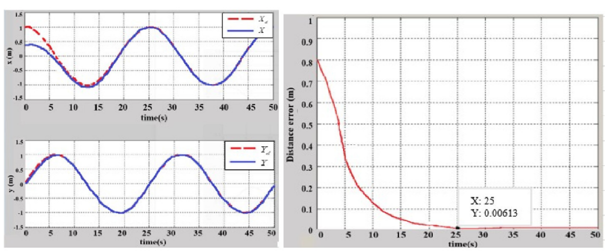

Fig. 10. x,y position and system error using FLC controller

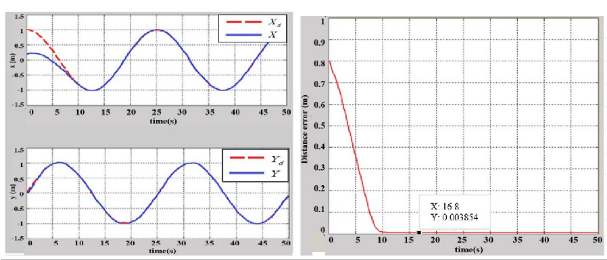

Fig. 11. x,y position and system error using PID – Fuzzy logic controller

Case 2: The trajectory is simulated, which is the crackling trajectory. The position response of the two controls is expressed in Figs. 12 to 14. All three controls react to the stability of a moving robot following a predetermined course. The PID controller, however, has a minor robot position error (0.01) and a slower reaction time (3s). However, even though the FLC controller's system error is just 0.006, it responds more quickly than the PID controller, which takes 2s. Much more optimal is the PID-Fuzzy logic controller error is almost zero.

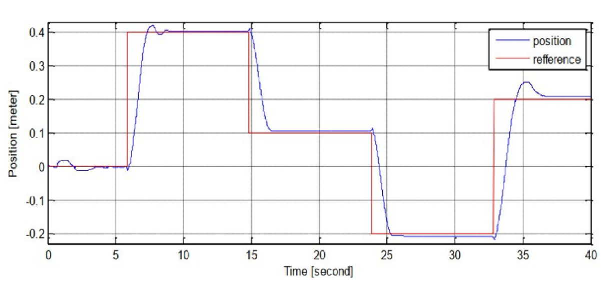

Fig. 12. Position using traditional PID controller

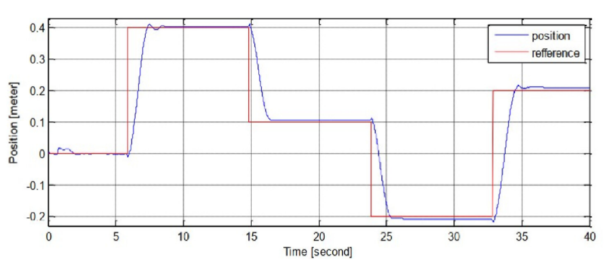

Fig. 13. Position using FLC logic controller

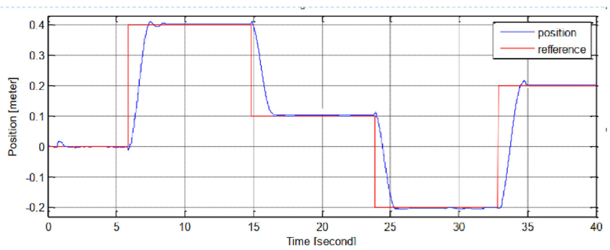

Fig. 14. Position using PID – Fuzzy logic controller

6. Conclusion

The paper has proposed a kinematic and dynamic model for a mobile robot with differential actuator based on the Lagrange dynamic approach. Mobile Robot is moved according to the trajectory set by the PID-Fuzzy logic controller. This controller has the advantages of simple design and better performance than PID and FLC controllers with orbital error of 0.004, setting time 10s. However, in order to improve the moving quality of robot mobile with zero error and faster, it is necessary to use hybrid control methods such as fuzzy logic combined with neural network, or Sliding mode control with fuzzy logic, or a combination controller artificial intelligence image processing.

References

[1] J. C. Alexander and J. H. Maddocks. "On the Kinematics of Wheeled Mobile Robots". In: *Autonomous Robot Vehicles*. Ed. by I. J. Cox and G. T. Wilfong. New York, NY: Springer New York, 1990, 5–24. DOI: 10.1007/978-1-4613-8997-2_1.

[2] J. Barraquand and J.-C. Latombe, (1993) *"Nonholonomic multibody mobile robots: Controllability and motion planning in the presence of obstacles"* **Algorithmica** ***10***(2-4): 121–155. DOI: 10.1007/BF01891837.

[3] G. Campion, G. Bastin, and B. D'Andréa-Novel, (1996) *"Structural properties and classification of kinematic and dynamic models of wheeled mobile robots"* **IEEE Transactions on Robotics and Automation** *12*(1): 47–62. DOI: 10.1109/70.481750.

[4] H. Maaref and C. Barret, (2002) *"Sensor-based navigation of a mobile robot in an indoor environment"* **Robotics and Autonomous Systems** *38*(1): 1–18. DOI: https://doi.org/10.1016/S0921-8890(01)00165-8.

[5] S. Thongchai, S. Suksakulchai, D. Wilkes, and N. Sarkar. "Sonar behavior-based fuzzy control for a mobile robot". In: *Smc 2000 conference proceedings. 2000 ieee international conference on systems, man and cybernetics. 'cybernetics evolving to systems, humans, organizations, and their complex interactions' (cat. no.0*. **5**. 2000, 3532–3537 vol.5. DOI: 10.1109/ICSMC.2000.886556.

[6] I. Kolmanovsky and N. McClamroch, (1995) *"Developments in nonholonomic control problems"* **IEEE Control Systems Magazine** *15*(6): 20–36. DOI: 10.1109/37.476384.

[7] F. L. Lewis, D. M. Dawson, and C. T. Abdallah. *Robot manipulator control: theory and practice*. CRC Press, 2003.

[8] Y. Li, L. Zhu, and M. Sun, (2013) *"Adaptive neural-network control of mobile robot formations including actuator dynamics"* **Applied Mechanics and Materials** ***303-306***: 1768–1773. DOI: 10.4028/www.scientific.net/AMM.303-306.1768.

[9] L. Yan-dong, Z. Ling, and S. Ming, (2013) *"Adaptive RBFNN formation control of multi-mobile robots with actuator dynamics"* **TELKOMNIKA Indonesian Journal of Electrical Engineering** *11*(4): 1797–1806. DOI: 10.11591/telkomnika.v11i4.2334.

[10] R. DeCarlo, (1996) *"Variable structure sliding-mode controller design"* **Control Handbook**: DOI: 10.1201/b10384-59.

[11] E. Derks, M. S. Pastor, and L. Buydens, (1995) *"Robustness analysis of radial base function and multi-layered feed-forward neural network models"* **Chemometrics and Intelligent Laboratory Systems** *28*(1): 49–60. DOI: 10.1016/0169-7439(95)80039-c.

[12] F. P. Freire, N. A. Martins, and F. Splendor, (2018) *"A simple optimization method for tuning the gains of PID controllers for the autopilot of cessna 182 aircraft using model-in-the-loop platform"* **Journal of Control, Automation and Electrical Systems** ***29***: 441–450. DOI: 10.1007/s40313-018-0391-x.

[13] W. Gao and J. C. Hung, (1993) *"Variable structure control of nonlinear systems: A new approach"* **IEEE transactions on Industrial Electronics** ***40***(1): 45–55. DOI: 10.1109/41.184820.

[14] F. Lewis, S. Jagannathan, and A. Yesildirak. *Neural network control of robot manipulators and non-linear systems*. CRC press, 1998.

[15] F. L. Lewis, D. M. Dawson, and C. T. Abdallah. *Robot manipulator control: theory and practice*. CRC Press, 2003.

[16] Y. Li, S. Qiang, X. Zhuang, and O. Kaynak, (2004) *"Robust and adaptive backstepping control for nonlinear systems using RBF neural networks"* **IEEE Transactions on Neural Networks** ***15***(3): 693–701. DOI: 10.1109/TNN.2004.826215.

[17] J. Keighobadi and Y. Mohamadi. "Fuzzy Sliding Mode Control of non-holonomic Wheeled Mobile Robot". In: *2011 IEEE 9th International Symposium on Applied Machine Intelligence and Informatics (SAMI)*. 2011, 273–278. DOI: 10.1109/SAMI.2011.5738888.

[18] M. Begnini, D. W. Bertol, and N. A. Martins, (2017) *"A robust adaptive fuzzy variable structure tracking control for the wheeled mobile robot: Simulation and experimental results"* **Control Engineering Practice** ***64***: 27–43. DOI: https://doi.org/10.1016/j.conengprac.2017.04.006.

[19] M. Begnini, W. Bertol, and N. A. Martins, (2018) *"Design of an adaptive fuzzy variable structure compensator for the nonholonomic mobile robot in trajectory tracking task"* **Control and Cybernetics** ***47***(3): 239–275. DOI: 10.5281/zenodo.4282917.

[20] M. Begnini, D. W. Bertol, and N. A. Martins, (2018) *"Practical implementation of an effective robust adaptive fuzzy variable structure tracking control for a wheeled mobile robot"* **Journal of Intelligent & Fuzzy Systems** ***35***(1): 1087–1101. DOI: 10.3233/JIFS-17699.

[21] N. A. Martins, E. R. De Pieri, U. F. Moreno, et al., (2014) *"PD-super-twisting second order sliding mode tracking control for a nonholonomic wheeled mobile robot"* **IFAC Proceedings Volumes** ***47***(3): 3827–3832. DOI: 10.3182/20140824-6-ZA-1003.02210.

[22] J. A. Goguen, (1973) *"L. A. Zadeh. Fuzzy sets. Information and control, vol. 8 (1965), pp. 338–353. - L. A. Zadeh. Similarity relations and fuzzy orderings. Information sciences, vol. 3 (1971), pp. 177–200."* **The Journal of Symbolic Logic** ***38***(4): 656–657. DOI: 10.2307/2272014.

[23] J. Zhang, N. Wang, and S. Wang. "A developed method of tuning PID controllers with fuzzy rules for integrating processes". In: *Proceedings of the 2004 American Control Conference*. **2**. IEEE. 2004, 1109–1114. DOI: 10.23919/ACC.2004.1386720.