Design real-time embedded optimal PD fuzzy controller by PSO algorithm for autonomous vehicle mounted camera

Cite as: AIP Conference Proceedings **2188**, 030008 (2019); https://doi.org/10.1063/1.5138401
Published Online: 17 December 2019

N. X. Chiem, N. D. Anh, A. D. Lukianov, P. D. Tung, H. D. Long, and N. D. Linh

Design Real-Time Embedded Optimal PD Fuzzy Controller by PSO Algorithm For Autonomous Vehicle Mounted Camera

N.X. Chiem [1, a)], N.D.Anh[2], A.D. Lukianov [3], P.D. Tung [4], H.D. Long[1], N.D. Linh[1]

[1]*Department of Automation and Computing Techniques, Le Quy Don Technical University, Hanoi, Vietnam*
[2]*Viet nam academy of science and technology, Vietnam*
[3]*Don State Technical University, Rostov on Don, Russia*
[4]*Department of Technology, equipment and aerospace, Le Quy Don Technical University, Hanoi, Vietnam*

[a)]Corresponding author: nguyenxuanchiem83@gmail.com.

Abstract. The paper presents methods for designing embedded controller for autonomous robots following the lane. Robots use cameras to detect lanes and use frame replacing techniques to determine the path of the robot. The controller is designed using the PSO - PD fuzzy logic controller. PSO - PD fuzzy logic is the controller that combines the Fuzzy Controller with the optimal parameter based on a swarm intelligence algorithm. This controller shows the effectiveness for the effects of disturbance or when the information of desired trajectory is interrupted. The simulation of the robot's trajectory tracking showed the quality and capability of this design method. Experimental results show the method's effectiveness when building on real-time embedded systems.

INTRODUCTION

Autonomous vehicles are widely used in many different fields in reality and play an important role in the traffic environment with high risk of accidents such as steep, slippery hills or smart transportation systems. Automatic motion control can help reduce accidents as well as traffic congestion [1-2].

Autonomous vehicles are widely used in many fields and their models are widely used in laboratories (Figure 1) [11-14]. In principle, a simple AVG system will consist of two main components: the preprocessor and the control unit. One of the most difficult research problems in robotic systems is the Autonomous vehicle motion control to follow a line with the best accuracy possible. There have been many studies of control methods but all have its advantages and disadvantages [5-9]. This paper presents how to determine the motion trajectory and the method of building a PSO-PD fuzzy logic controller for robots that follow the desired trajectory. The quality of the control system is shown on simulation results and experimental models.

THE MATHEMATICAL MODEL OF TWO-WHEELED AUTONOMOUS VEHICLE

Kinematic and Dynamic Modeling of Two-Wheeled Autonomous Vehicle [5,7,12]. The model of autonomous vehicle is built as Figure 1 and the geometric structure of the vehicle in the form of 1b. The goal of the car dynamics model is to find the vehicle's speed through the wheel speed and the vehicle's geometric parameters.

XV International Scientific-Technical Conference "Dynamic of Technical Systems" (DTS-2019)
AIP Conf. Proc. 2188, 030008-1–030008-7; https://doi.org/10.1063/1.5138401
Published by AIP Publishing. 978-0-7354-1935-3/$30.00

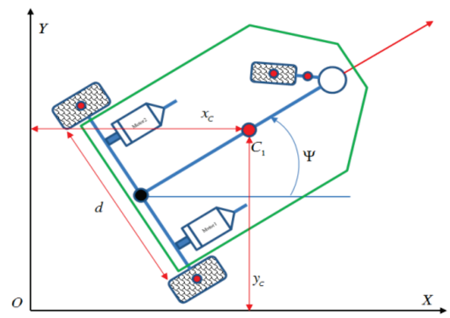


FIGURE 1. The geometric structure of Two-Wheeled Mobile Robot

Suppose the vector of the autonomous vehicle is presented as $p=[x \quad y \quad \psi]^T$, where x and y are the position of the autonomous vehicle and ψ is the angle between the X-coordinate axis and the direction of the autonomous vehicle (Figure 2). Autonomous vehicles are controlled by the angular velocity of the wheels ω_L, ω_R. There are connections between angular velocities ω_L, ω_R and velocity V_L, V_R [13]:

$$V_L = R\omega_L; \quad V_R = R\omega_R; \tag{1}$$

where R is the radius of the wheels.

A kinetic model of mobile robots can be based on the following equation:

$$\dot{x} = V\cos(\psi); \quad \dot{y} = V\sin(\psi); \quad \dot{\psi} = \omega; \quad V = \frac{V_L + V_R}{2}; \quad \omega = \frac{V_L - V_R}{D}; \tag{2}$$

where D is the distance between two –wheeled control.

The dynamic model of this robot is described by the following equation system:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \frac{\cos(\psi)}{2} & \frac{\cos(\psi)}{2} \\ \frac{\sin(\psi)}{2} & \frac{\sin(\psi)}{2} \\ \frac{D}{2} & \frac{D}{2} \end{bmatrix} \begin{bmatrix} V_L \\ V_R \end{bmatrix} \tag{3}$$

Dynamic equations of the robot motion in the form [6,7]:

$$\begin{cases} \dot{V} = \frac{nc}{r}(U_1 + U_2) + \frac{m_1 a\omega^2}{m} - \frac{2n^2c^2}{r^2Rm} - \mu V \\ \dot{\omega} = \frac{ncD}{2rRJ}(U_1 - U_2) - \frac{m_1 aV\omega}{J} - \frac{2n^2c^2D^2}{4r^2Rm}\omega - \mu\frac{D^2}{4r^2}\omega \end{cases} \tag{4}$$

where n - gear transmission ratio; $m = m_1 + 2m_k + 2J_y r^{-2}$; $J = J_1 + (m_1 + 2m_k)a^2 + 2J_y r^{-2} l^2$; m_1 - mass of the stator and the bracket; m_k - mass of gear and motor rotor; J_1 - the inertial moment of the robot compared to the vertical axis through the center C_1; $J_y = J_{ky} + n^2 J_{ry}$ - the inertial moment of the wheels; J_{ky} – the inertial moment by horizontal axis; J_{ry} – the inertial moment of motor rotor; $a=AC_1$; c – coefficient of electromechanical interactions; μ – the coefficient of friction; R – resistance of roto; U_1, U_2 – control voltage applied to 2 electric motors.

BUILD THE LANE DETECTION ALGORITHM

The image processing algorithms for lane detection can be divided into three main groups: (1) based on the region (region-based), (2) based on feature analysis (feature-based) and (3) based on the mathematical modeling of lane (model-based). Most studies follow this third approach, using mathematical models for lane detection [1-4]. The algorithm for our lane detection method is shown in the diagram in Figure 2. The algorithm for lane detection method is shown in the diagram in Figure 2.

After the image pre-processing according to the algorithm in Figure 1, when the binary image is received and the lane start point is detected through the histogram. Using lane detection techniques as a sliding window technique, placed around the center of the line to find and track the lines up to the top of the frame. The sliding window technique can be displayed as shown in the image below (Figure 3). Next, the curvature of the lane and the vehicle position for the center line of the lane and stacking the lane boundaries are detected back into the original image. Finally, determine the center of the lane that the vehicle needs to follow (Figure 4).

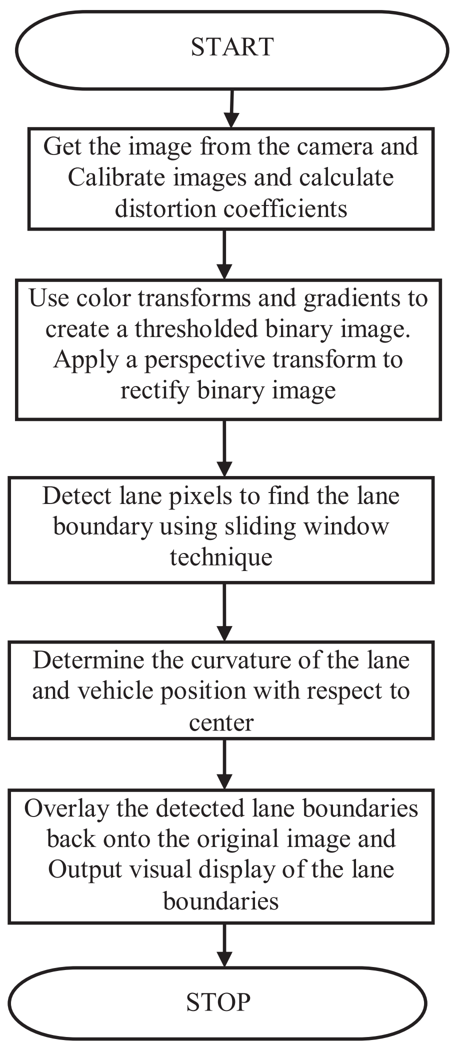


FIGURE 2. Flowchart of image processing algorithm

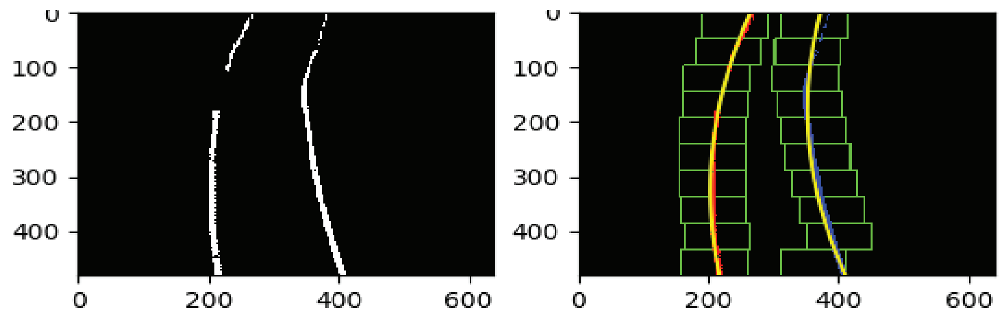


FIGURE 3. Illustration of lane detection method with sliding window

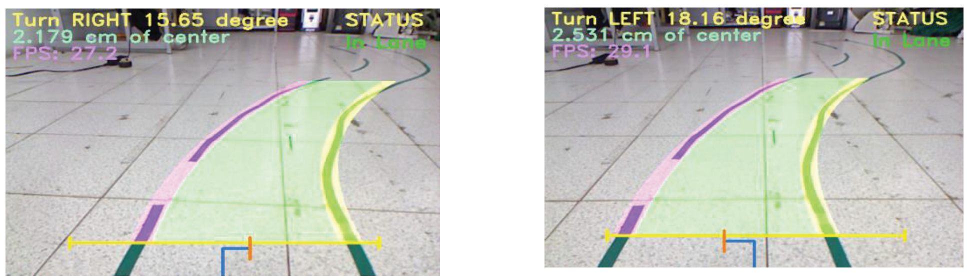


FIGURE 4. Output of the image processing

DESIGN CONTROLLERS COMBINE SWARM ALGORITHMS FOR AUTONOMOUS VEHICLE PATH TRACKING

Diagram of control structure of autonomous vehicles

In the paper, the authors designed two controllers (Figure 2). The speed stabilizer controller uses the traditional PD control law based on the speed error with the set point V_{sp} and the derivative of speed error. The directional controller is designed to be a Fuzzy Logic PD Controller with two inputs being angular error and the derivative of angular error. This controller is executed once so the input parameter is optimized through the PSO algorithm that adjusts these parameters.

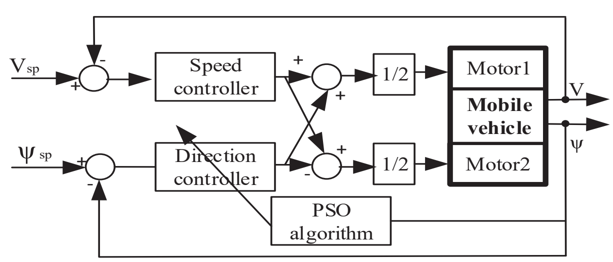


FIGURE 5. Structure diagram of direction controller for autonomous vehicle.

Design PSO-PD Fuzzy Logic Controller for autonomous vehicles

In the paper, the controller is designed based on fuzzy logic theory and the traditional controller PD stabilizes the vehicle speed with the structure shown in Figure 6.a. The input of fuzzy controller is angular error (e) and the angular error derivative (), is is fuzzificated by 5 language variables {NB, NS, ZE, PS, PB}. The fuzzy set of input, output is denoted as follows: NB is Negative Big, NS is Negative Small, ZE is Zero, PS is Positive Small, PB is Positive Big. The functions of the input are built within [-1: 1] (Figure 6.b), these membership functions are designed once for both inputs. The change of the boundary value of the membership functions is standardized and optimized by the Kp and Kd coefficients using the PSO algorithm. From the basis of knowledge experience about the dynamic characteristics of autonomous vehicles, the law of the controller is designed based on experiment (Table 1). The defuzzyfication method of fuzzy controller is calculated according to the defuzzyfication principle of Tagaki-Sugeno fuzzy model.

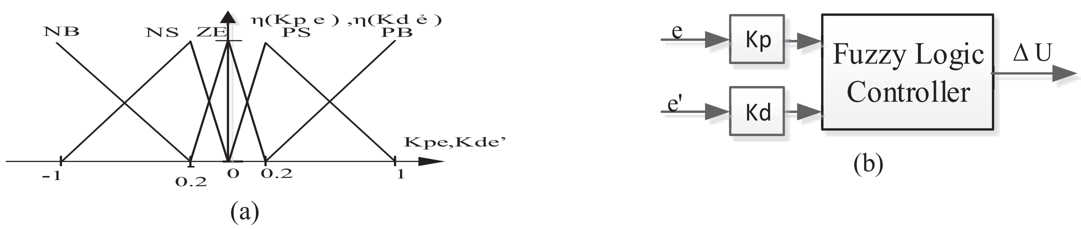


FIGURE 6. Selection type of fuzzy membership function and controller structure

TABLE 1. The fuzzy rule sets

e \ e'	**NB**	**NS**	**ZE**	**PS**	**PB**
NB	PB	PB	PB	PS	ZE
NS	PB	PB	PS	ZE	NS
ZE	PB	PS	ZE	NS	NB
PS	PS	ZE	NS	NB	NB
PB	ZE	NS	NB	NB	NB

Particle swarm optimization (PSO) was developed by Kennedy and Eberhart in 1995 based on swarm behavior in nature, such as fish and bird schooling. The PSO algorithm searches the space of an objective function by adjusting the trajectories of individual agents, called particles, as the piecewise paths formed by positional vectors in a quasi-stochastic manner [15,16]. The movement of a swarming particle consists of two major components: a stochastic component and a deterministic component. PSO algorithm is implemented as follows::

Particle Swarm Optimization
Objective function $f(x), x = (x_1, ..., x_d)^T$
Initialize locations x_i and velocity v_i of n particles.
Find g^* from min$\{f(x_1), ..., f(x_n)\}$ (at $t = 0$)
while (criterion)
for loop over all n particles and all d dimensions
Generate new velocity v_i^{t+1} using equation $v_i^{t+1} = v_i^t + \alpha e_1 [g^* - x_i^t] + \beta e_2 [x_i^* - x_i^t]$
Calculate new locations $x_i^{t+1} = x_i^t + v_i^{t+1}$
Evaluate objective functions at new locations x_i^{t+1}
Find the current best for each particle x_i^*
end for
Find the current global best g^*
Update $t = t + 1$ (pseudo time or iteration counter)
end while
Output the final results x_i^* and g^{*P}

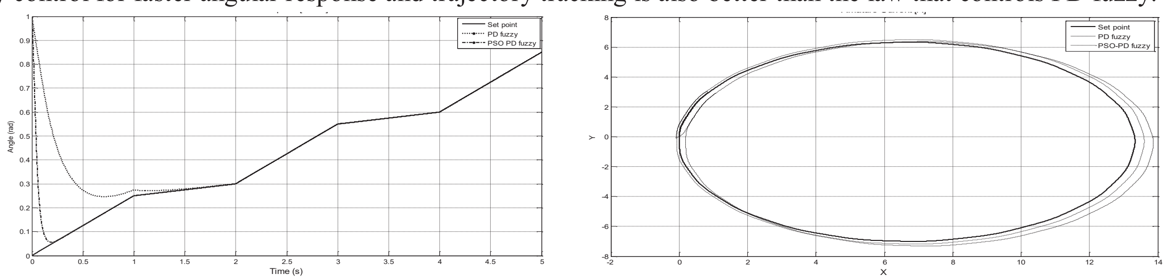
The target function of the PSO algorithm is chosen as the ITAE function:

$$ITAE: \quad g = \int t|e(t)|dt \tag{5}$$

SIMULATION RESULTS AND EXPERIMENTS ON REAL MODELS

Simulation results: The parameters are selected for the model when simulating the approximate real model value: m_l=5.0 (kg), m_k=1.4 (kg), J_l=0.9 (kg/m^2); J_{kz}=0.0002 (kg/m^2); J_{ky}=0.0003(kg/m^2); J_{ry}=0.00004(kg/m^2); μ=0.1; n=30; c=2; l=0.3(m); R=1.5(om); a=0.2(m); r=0.07(m). With the initial condition that the vehicle is on the desired trajectory, the vehicle speed is 0 (m/s) and has an angular error of 1 (rad).

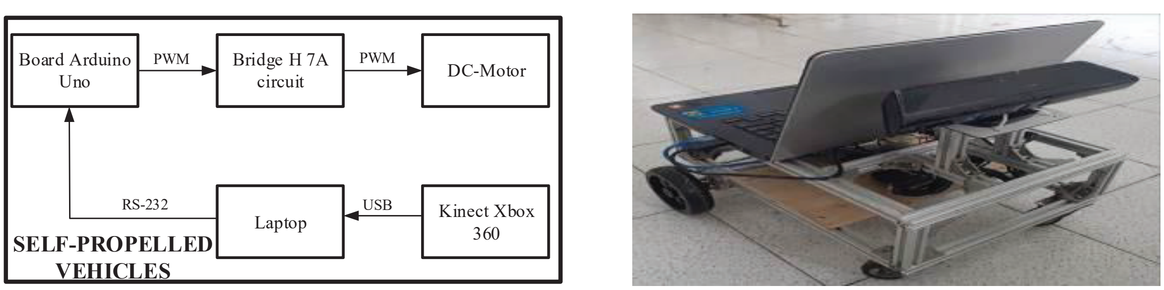
Figure 7a, 7b show the angular response and actual trajectory of vehicle to the desired trajectory with two controllers PD fuzzy and PSO-PD fuzzy. From the results we can conclude that according to the law of PSO-PD fuzzy control for faster angular response and trajectory tracking is also better than the law that controls PD fuzzy.



(a) (b)

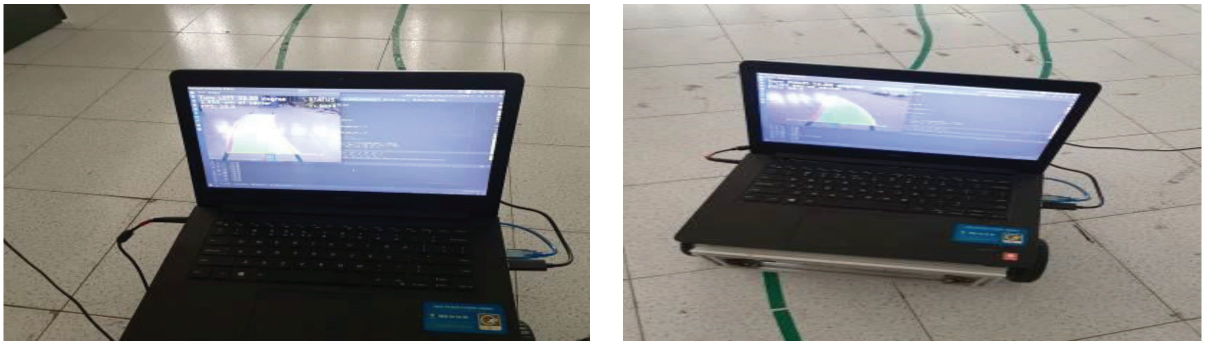
FIGURE 7. Simulation results of direction and trajectory tracking.

Experimental results: Block diagram and autonomous vehicle model are built as shown in Figure 8. Control the speed and direction of the vehicle are two motors with wheels behind. Camera Kinect Xbox 360 captures photos of the path to the computer via USB. The computer processes images from cameras that determine the position and angular error from the desired trajectory. The Arduino Uno embedded board receives information about the angular error transmitted from the computer using the PSO-PD fuzzy algorithm that gives the motor control signal via the H-bridge circuit to the motors. The experimental results in Figure 9 show the embedded control system with PSO-PD fuzzy controller gave good results. Stable driving, detecting and tracking lanes with acceptable errors.



(a) (b)

FIGURE 8. Block diagram and autonomous vehicle model



(a) (b)

FIGURE 9. Experimental results on autonomous vehicle model

CONCLUSION

The paper presents a method for designing embedded control systems of autonomous vehicle with cameras based on the control rules of PSO-PD fuzzy controller on the direction channel. The simulation results showed better control quality of the controller when compared to the PD-fuzzy controller. Experimental results show the enforcement of control rules on real systems. Future studies will supplement the impact of disturbance in the mathematical model, combine neural networks in control systems, design nonlinear control rules, but also new algorithms in image processing and detect lane.

REFERENCES

1. M.Asif, M.R.Arshad, and P.A.Wilson, 2007. "AGV Guidance System: An Application of Simple Active Contour for Visual Tracking", Proceeding world academy of science engineering and technology, Vol. 6, No. 2, pp. 664 –667.
2. Yue Wang, Eam Khwang Teoh, Dinggang Shen, 2004. "Lane detection and tracking using Bsnake", Journal of Image and Vision Computing, Vol. 22, No. 1, pp. 269–280.
3. Jiang Ruyi, Klette Reinhard, Vaudrey Tobi, Wang Shigang, 2011. "Lane detection and tracking using a new lane model and distance transform", Journal of Machine Vision and Application, Vol. 22, No. 4, pp. 721–737.
4. Mohamed Aly, 2008. "Real time Detection of Lane Markers in Urban Streets", International Conference on Intelligent Vehicles Symposium, pp.7-12.
5. Amin Abbasi, Ata Jahangir Moshayedi, Trajectory Tracking of Two-Wheeled Mobile Robots, Using LQR Optimal Control Method, Based On Computational Model of KHEPERA IV. Journal of Simulation & Analysis of Novel Technologies in Mechanical Engineering 10 (3) (2017), pp. 41-50.
6. Sevil A. Ahmed. Michail G. Petrov. Trajectory Control of Mobile Robots using Type-2 Fuzzy-Neural PID Controller. IFAC-PapersOnLine 48-24 (2015) 138–143
7. Petrov M., I. Ganchev, A. Taneva. Fuzzy PID Control of Nonlinear Plants. Proceedings of the IEEE International Symposium on "Intelligent Systems", Varna, Bulgaria, 10 -13 September, 2002, IEEE Catalogue Number 02EX499, ISBN 0-7803-7601-3, Vol. 1, pp. 30 - 35.
8. Emina Petrović, Vlastimir Nikolić, Ivan Ćirić, Miloš Simonović, Saša Pavlović, Marko Mančić, Boban Rajković Kinematic. Model and control of mobile robot for trajectory tracking. Annals of faculty engineering hunedoara – International journal of engineering Tome xiv [2016] – fascicule 2 -may. Pp. 161-164.
9. Wai, R.-J., Liu, C.-M. (2009). Design of dynamic Petri recurrent fuzzy neural network and its application to path-tracking control of nonholonomic mobile robot. IEEE transactions on Industrial Electronics, vol. 56, no.7, pp. 2667–2683.
10. Lee, J.H., Lin, C., Lim, Lee J.M. (2009). Sliding Mode Control for Trajectory Tracking of Mobile Robot in the RFID Sensor Space. International Journal of Control, Automation, and Systems, vol. 7, no.3, pp.429-435.
11. Xu,Q., Kan, J., Chen, S., Yan, S. (2014). Fuzzy PID Based Trajectory Tracking Control of Mobile Robot and its Simulation in Simulink. International Journal of Control and Automation, vol.7, no.8, pp. 233- 244.
12. Zhongbao Luo, Wei Li: Tracking of mobile robot expert PID controller design and simulation, 2014 International Symposium on Computer, Consumer and Control.
13. Li. Kunpeng, Wang. Xuewen, Yuan. Mingxin, Li. Xiaohu, and Wang. Sunan: Adaptive Sliding Mode Trajectory Tracking Control of Mobile Robot with Parameter Uncertainties, IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA 2009) - Daejeon, Korea (South) (2009.12.15-2009.12.18)] 2009.
14. Nguyen X.C. Phan N.H. Hoang D.L. Truong D.K. Tran C.P. Do N.S. Advance in the efficiency of type 2 fuzzy controller in embedded systems and application for control two-wheel self-balancing robot. E-journal "Engineering journal of Don", Number 3, 2018.
15. Kennedy J, Eberhart RC. Particle swarm optimization. In: Proceedings of the IEEE international conference on neural networks, Piscataway, NJ, USA; 1995. p. 1942–48.
16. Kennedy J, Eberhart RC, Shi Y. Swarm intelligence. London, UK: Academic Press; 2001.