Improving a Wall-Following Robot Performance with a PID-Genetic Algorithm Controller

Heru Suwoyo
School of Mechatronic Engineering and Automation,
Shanghai University,
Shanghai, P.R China
herusuwoyo14@yahoo.com

Yingzhong Tian
School of Mechatronic Engineering and Automation,
Shanghai University,
Shanghai, P.R China
troytian@163.com

Chenwei Deng
School of Information and Communication Engineering,
Beijing Institute of Technology,
Beijing, Shi, China
cwdeng@bit.edu.cn

Andi Adriansyah
Departement of Electrical Engineering,
Universitas Mercu Buana,
Jakarta, Indonesia
andi@mercubuana.ac.id

***Abstract*—A wall-following robot needs a robust controller that navigate robot based on the specified distance from the wall. The usage of PID controller has been successfully minimizing the dynamic error of wall-following robot. However, a manual setting of three unknown parameters of PID-controller often precisely increase instability. Hence, recently there are many approaches to solve this issue. This paper presents an approach to obtaining those PID parameters automatically by utilizing the role of Genetic Algorithm. The proposed method was simulated using MATLAB and tested in a real robot. Based on several experiments results it has been showing the effectiveness of reducing the dynamic error of the wall-following robot.**

***Keywords*—Wall-Following Robot, PID Controller, Genetic Algorithm, PID-GA Controller**

I. INTRODUCTION

A wall-following robot is a robot that has the primary task to follow the wall by maintaining its movement. The robot will move closer to the wall, move forward and move further from the wall when its position is far from the wall, on the predetermined path and close to the wall, respectively. Thus, apparently, by utilizing the mounted distance sensors, the robot has to sense the distance between the wall and itself along the operation. Besides, these distances are processed to generate the proper movement with the involvement of a specific controller. The success of this robot lies in its capability to follow the predetermined path by concerning quickness and accuracy, that apply in automatically wheelchair, electronic goods transportation, and automated guided vehicle (AVG) [1]-[4].

However, the accuracy of all sensors is affected by limited precision. Evenly the best sensors still have a degree imprecision. Consequently, the role of the controller in maintaining the movement has been becoming the core of the problem to be accomplished. There are many control methods have been attempted to address the common issue of the wall-following robot [5], such as Fuzzy Logic [6][7], Genetic Algorithm, Neural Network or hybrid of them [8]. But, most of these approaches have to utilize the orientation sensor featured in the robot (compass, GPS, etc.) [9].

The well-known controller proposed method to address these problems is a PID Controller that generating the proper velocity for each wheel referring to the rule of pulse width modulation (PWM) [10][11]. As known, generally, the performance of PID-controller lies on a determination of its three necessary parameters. Each parameter of the PID controller has a particular influence on the process. Consequently, their precision should be concerned carefully.

There have been many methods proposed in producing these parameters automatically. The previous PID tuning is using Ziegler-Nichols Algorithm [12]. But the tuning method is needed difficult robot modeling for satisfaction one. In another case, the necessary parameters were obtained automatically involving an algorithm that can derive the original data to be proper optimum parameter respect to the movement. Fuzzy-PID is one of the tuning methods of PID using Fuzzy Logic [13][2]. Tang et al. [14] and Takahashi [15] also proposed a method to tune PID automatically using Neural Network. Unfortunately, these the process is too many computations works that make a control process operated in hard.

This paper proposed a method to improve the performance of a wall-following robot with PID Controller which the parameters tuned automatically by Genetic Algorithm (GA). GA is a metaheuristic inspired by the process of natural selection used to optimize and search for some variables [16] [17].

The robot considered here is a self-designed robot which is completed by some distance sensors. The robot will maneuver use a differential-wheel system of the mobile robot [18]-[20]. These sensors were installed and placed on the precise direction which aims to be ably sensing the left-side, right-side and front-side wall accurately.

This paper is organized as follows. The related basic-knowledge is presented in Section 2. Section 2 also describes the design flow of the proposed method. Section 3 gives the experimental result and discussion. Finally, the experiment will be concluded in Section 4.

II. METHOD

A designed robot that completed by three distance sensors is used to implement the proposed method. This robot was intended to be able working automatically utilizing the capability microcontroller, Arduino, which calculates the distance and control two DC motors connected to an odometer.

A. Robot Kinematic Model

Fig. 1 shows the steering system of the nonholonomic wheeled mobile robot. The turning ability is dominantly due to two powered wheels referring to the rule of the differential drive system [17]. The robot has a dimension such as L=10 cm R=3.5 cm

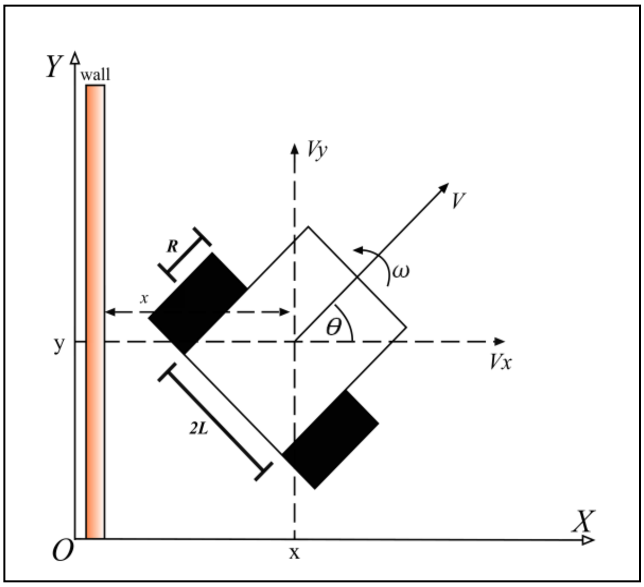


Fig. 1. Robot Pose in a Globa Coordinate System

By considering that the robot is operated in a planar environment, the robot's pose p comprises two-dimensional planar coordinates (x,y), and an orientation θ. Thus, the prior robot’s pose can be described by the following vector,

$$p_{(t)} = [x_{(t)} \quad y_{(t)} \quad \theta_{(t)}]^T \tag{1}$$

Since the linear velocity v and angular velocity ω are considered as the control reference, we have the dynamic,

$$\dot{p} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta_{(t)} & 0 \\ \sin\theta_{(t)} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{2}$$

where $\dot{p}$ is the transition of the robot. Then the posterior robot’s pose becomes

$$p_{(t+1)} = p_{(t)} + \dot{p} \tag{3}$$

Additionally, involving the term angular velocities for each powered wheels, the control reference can be equated as follows:

$$v = \frac{(\omega_r + \omega_l)}{2R} \tag{4}$$

$$\omega = \frac{(\omega_r - \omega_l)}{2RL} \tag{5}$$

$$\omega_r = vR + \omega RL \tag{6}$$

$$\omega_l = vR - \omega RL \tag{7}$$

where ω_r and ω_l are the angular velocity of the right and left the wheel, respectively.

B. PID-GA Controller Design

PID-controller is a well-known controller widely used to solve a nonlinear problem of the specific system. It adjusts three necessary parameters as condition change dynamically. These non-zero parameters are commonly termed as proportional gain Kp, integral gain Ki, and derivative gain Kd and due to their influence to the output controller, obtaining their accurate values have been regarded as the crucial task of the usage PID controller. These specific values can be obtained by performing particular high-level tuning.

Generally, the optimum setting of these parameters involves the role of accumulated error, e(t), which is calculated from actual process value pv and predetermined value termed as set point, sp(t). Considering u(t) as the control variable, then the analogy of the PID controller will be depicted as Fig. 2. It can be modeled mathematically as follows [9] [10],

$$u(t) = Kp.e(t) + Ki.\int_0^t e(\tau)d\tau + Kd\frac{d}{dt}e(t) \tag{8}$$

where *t* indicates that the controller is operated in a time domain.

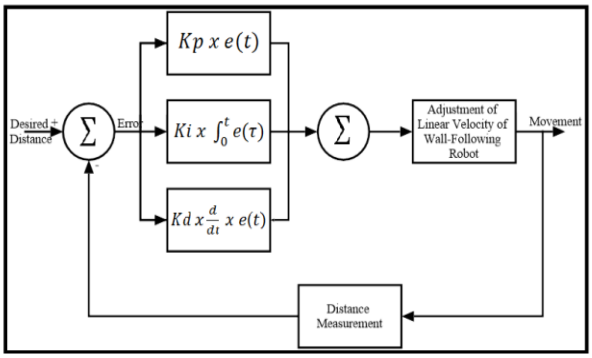


Fig. 2. The PID Controller Diagram

Obtaining the non-zero value of the proper amount of three-term controller Kp, Ki and Kd have been challenging many researchers. Ineffectively it has been accomplished by manual tuning.

Genetic Algorithm (GA) is an optimization method based on the natural selection process. It modifies the initial population of individual solution till obtained the proper optimum solution to the target system. A term named as fitness function represents the quality of target optimized system. The optimal solution offered by GA can be perceived by adjusting the generated random or known as initial population value with natural process termed as crossover and mutation. To produce the best solution, the process iteration can be predetermined as many as possible [15] [16].

The mobile robot has a task to follow the desired path by maintaining the rotational velocity of each wheel as shown in Fig. 3. These velocities are generated from the linear velocity produced by the PID controller.

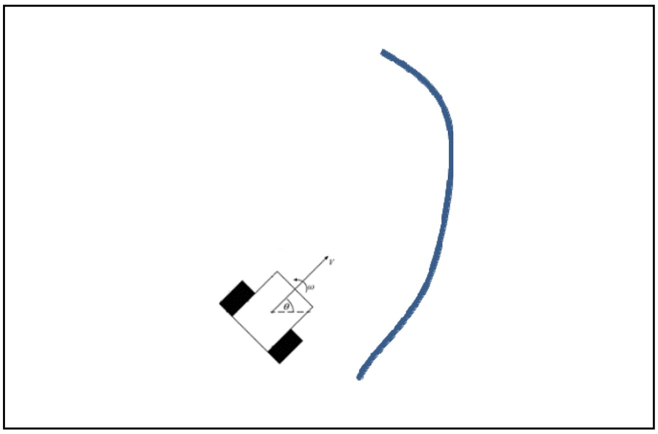


Fig. 3. Wall-Following Process

In order to utilize the general equation of PID, initially the error was designed as follows:

$$e(t+1) = sp - pv(t) \quad (9)$$

Then, considering that the angular velocity ω has no much effect to the robot's dynamic error, the angular velocity v is regarded to be equal as the half of prior orientation. Hence, the PID controller only produces the linear velocity of the robot v that will be used to obtain each angular velocity of the wheel ω by applying (6)-(7).

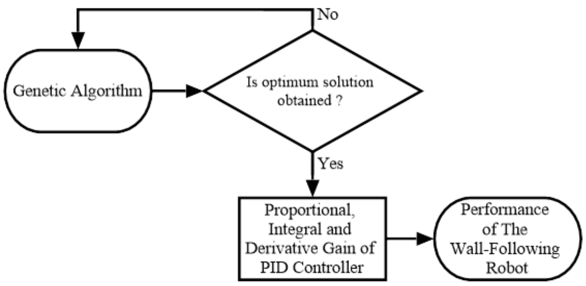


Fig. 4. The PID-GA Controller Diagram

Furthermore, the GA was used to generate three basic parameters of PID (i.e. Kp, Ki, and Kd). To know the quality of the gained solution, the fitness function is constructed as follows:

$$\sum_{it=0}^{max} e(it)^2 \quad (10)$$

where it represents all error along the movement. Each iteration stores an optimum solution of a generation and the next generation will store the new solution if it is better than the previous solution otherwise the process in the current generation will be restarted. The last representation shows us the best solution of three basic parameters of PID controller produced by GA.

Finally, the output of the PID controller $u(t)$ can be used to recalculate the linear velocity of the stable pose $e(t) = 0$ denoted by v_{st}. Then, the final linear velocity $v(t)$ is derived as follows:

$$v(t) = v_{st} - u(t) \quad (11)$$

III. Results and Discussion

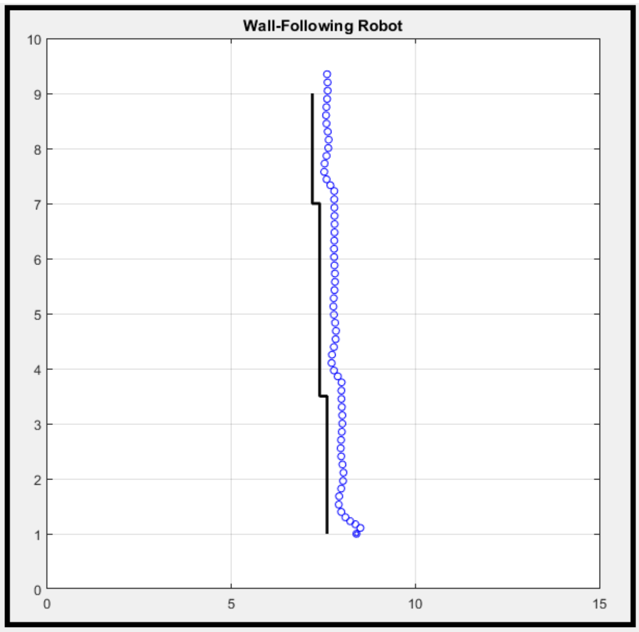
The real wall-following robot has been designed and shown in Fig. 5. The proposed method of PID-GA controller was tested by simulating the robot kinematic model using MATLAB.

Fig. 5. The Real Wall-Following Robot

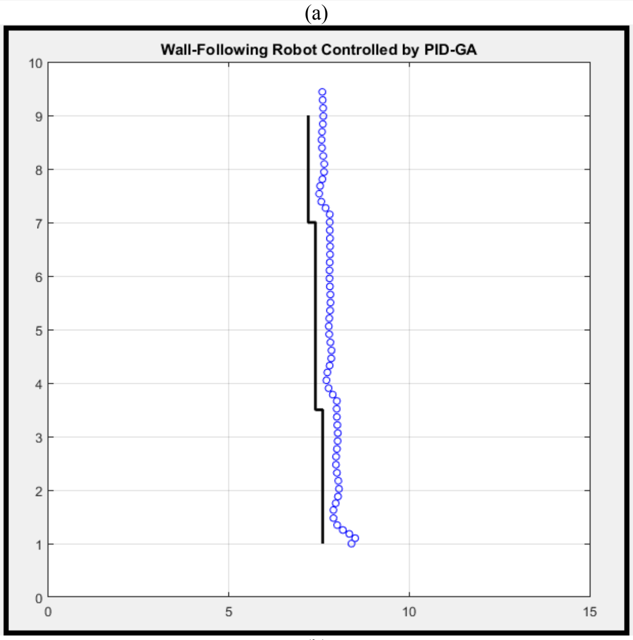
Automatically, a GA has been approached to generating the parameters of the PID controller. The effectiveness includes the quick process and the guaranteed precision. To determine the satisfied parameters, which are non-negative constants, it is required to involve the optimization method. Two different experiments of the wall-following robot performance have been performed either normal with constant linear velocity or controlled by PID-GA Controller.

The mobile robot with different those two type of the controller was positioned at the same coordinate position (8.4,1,π/4). Then, both were operated at the same time span of 100 seconds. The robot should follow the wall in a distance of 6 cm from the wall. By applying 60 iterations or approximately its operation within a minute, the robot performance with the constant linear velocity of 0.15 cm/s is seen in Fig. 6(a). By applying same starting point, the performance of wall-following robot controlled by the GA-PID Controller as seen in Fig. 6(b). The comparison of two different performances can be presented by its generated error on a scale of dm and the last coordinate as shown in Table 1.

Furthermore, the end position that reached by the robot and the total number of errors expressed by (4) have been observed. The position achievement of the robot with PID-GA controller farther from the starting position. It indicates that the robot with PID-GA controller has a better working speed compared to the others. While the value of the total error with PID-GA controller lest than other. It also indicates that PID-GA controller has a better level of following the desired path.



(a)



(b)

Fig. 6. The Result of the Performance of Wall-Following Robot: (a). PID Controller without GA. (b) GA-PID Controller

TABLE I. TABLE TYPE STYLES PERFORMANCE OF WALL-FOLLOWING ROBOT

Characteristic	Generated Error	Final Position
Normal Movement	0.83625	9.1986
Based on PID-GA	0.78543	9.2867

IV. CONCLUSION

The PID-GA Controller has implemented to control the wheeled-mobile robot to follow the desired path along the wall. The role of PID-GA Controller is to produce the proper linear velocity of the wall-following robot. The reduce of the error number shows the effectiveness of the proposed method. It can be seen from the experimental result that shows the robot has faster speed and smoother movement than the before. The proposed controller able to improve the performance of wall following robot better. The GA able to optimize the PID parameters simpler and with optimization time faster than optimize manually.

ACKNOWLEDGMENT

The authors would like to gratefully acknowledge the funding from the Research Program of Electrical Engineering Department, Faculty of Engineering, Universitas Mercu Buana.

REFERENCES

[1] Y. T. Lee, C. S. Chiu and I. T. Kuo, "Fuzzy wall-following control of a wheelchair," Joint 17th World Congress of International Fuzzy Systems Association and 9th International Conference on Soft Computing and Intelligent Systems (IFSA-SCIS), Otsu, Japan, pp. 1-6, 2017.

[2] A. Adriansyah and H.M.A. Shamsudin, "Wall-Following Behavior-Based Mobile Robot Using Particle Swarm Fuzzy Controller," Jurnal Ilmu Komputer dan Informasi, Vol. 9, No.1, pp. 9-16, 2016.

[3] A.A. Wardana, A. Widyotriatmo, and A. Turnip, "Wall following control of a mobile robot without orientation sensor," 3rd International Conference on Instrumentation Control and Automation (ICA), Ungasan, Indonesia, pp. 212-215, 2013.

[4] A. Adriansyah, Y. Gunardi, B. Badaruddin and E. Ihsanto, "Goal-seeking Behavior-based Mobile Robot using Particle Swarm Fuzzy Controller," TELKOMNIKA, Vol. 13, No. 2, pp. 528-538, 2015.

[5] A. Pandey, S. Pandey, S. and D.R. Parhi, "Mobile Robot Navigation and Obstacle Avoidance Techniques: A Review," International Robotics & Automation Journal, Vol. 2, No. (3), pp. 00022, 2017.

[6] I. Emmanuel, "Fuzzy Logic-Based Control for Autonomous Vehicle: A Survey," International Journal of Education and Management Engineering, Vol. 2, pp. 41-49, 2017.

[7] N. P. Varma, V. Aivek and V. R. Pandi, "Intelligent wall following control of differential drive mobile robot along with target tracking and obstacle avoidance," 2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT), Kannur, pp. 85-91, 2017.

[8] N.H. Singh, and K. Thongam, "Fuzzy Logic-Genetic Algorithm-Neural Network for Mobile Robot Navigation: A Survey," International Research Journal of Engineering and Technology (IRJET), Vol. 4, No. 8, pp. 24-45, 2017.

[9] A. Mohamed., M. El-Gindy and J. Ren, "Advanced control techniques for unmanned ground vehicle: literature survey," International Journal of Vehicle Performance, Vol 4, No. 1, 2018.

[10] A.S. Wardoyo, S. Hendi, D. Sebayang, I. Hidayat, A. Adriansyah, "An investigation on the application of fuzzy and PID algorithm in the two wheeled robot with self balancing system using microcontroller," In 2015 IEEE International Conference on Control, Automation and Robotics (ICCAR), Singapore, pp. 64-68, 2015.

[11] I. Ardiyanto, "Task oriented behavior-based state-adaptive pid (proportional integral derivative) control for low-cost mobile robot," 2010 Second IEEE International Conference on Computer Engineering and Applications (ICCEA), Bali, Indonesia, Vol. 1, pp. 103-107, 2010

[12] C.B. Prakash and R.S. Naik, "Tuning of PID Controller by Ziegler-Nichols Algorithm for Position Control of DC Motor," International Journal of Innovative Science, Engineering & Technology, Vol. 1, No. 3, pp. 379-382, 2014.

[13] C.Y. Chou and C.F. Juang, "Navigation of an Autonomous Wheeled Robot in Unknown Environments Based on Evolutionary Fuzzy Control," Inventions, Vol. 3, No. 3, pp. 2-14, 2018.

[14] Z. Tang, M. Yang, and Z. Pei, "Self-Adaptive PID Control Strategy Based on RBF Neural Network for Robot Manipulator," 2010 First International Conference on Pervasive Computing, Signal Processing and Applications, Harbin, pp. 932-935, 2010.

[15] K. Takahashi, Y. Hasegawa and M. Hashimoto, "Remarks on Quaternion Neural Network-Based Self-Tuning PID Controller," 2016

International Symposium on Computer, Consumer, and Control (IS3C), Xi'an, China, pp. 231-234, 2016.

[16] M. Li, C. Wang, Z. Chen, X. Lu, M. Wu and P. Hou, "Path planning of mobile robot based on genetic algorithm and gene rearrangement," 2017 Chinese Automation Congress (CAC), Jinan, China, pp. 6999-7004, 2017.

[17] D. Sahu and A. K. Mishra. "Mobile Robot Path Planning by Genetic Algorithm with Safety Parameter," International Journal of Engineering Science and Computing, Vol. 7, No. 8, pp. 14723-14727, 2017.

[18] A.V. Chavan and J. L. Minase, "Design of a Differential Drive Mobile Robot Platform for use in constrained environments," International Journal of Innovations in Engineering Research and Technology (IJIERT), Vol. 2, No. 6, pp. 1-9, 2015

[19] Y. Gunardi, D. Hanafi, F. Supegina and A. Adriansyah, "Mathematic base for navigation mobile robot using reachability Petri net," Journal of Telecommunication, Electronic and Computer Engineering, Vol. 10, Issue 1-9, pp. 65-69, 2018

[20] A. Adriansyah, B. Sulle, E. Ihsanto, and Y. Gunardi, "Optimization of circular robot size using behavior based architecture," Journal of Telecommunication, Electronic and Computer Engineering, Vol. 9, Issue 3-7, pp. 67-72, 2017