TEKNOKOM : Jurnal Teknologi dan Rekayasa Sistem Komputer
Vol. 5, No. 1, March 2022, pp. 103 - 112
ISSN: 2686-3219, DOI: 10.31943/teknokom.v5i1.74

THE ACA-BASED PID CONTROLLER FOR ENHANCING A WHEELED-MOBILE ROBOT

Heru Suwoyo[1*], Zhou Thong[2], Yingzhong Tian[2], Andi Adriansyah[1], Muhammad Hafizd Ibnu Hajar[1]
[1]Department of Electrical Engineering, Universitas Mercu Buana, Indonesia
[2]School of Mechanical and Electrical Engineering and Automation, Shanghai University, China

ARTICLE INFO

History of the article:
Received February 15, 2022
Revised March 21, 2022
Accepted March 25, 2022
Published April 07, 2022

Keywords:
Wall-Following Robot
PID Controller
Ant Colony Algorithm
Parameter Optimization

Correspondece:
Heru Suwoyo,
Department of Electrical Engineering,
Universitas Mercu Buana, Indonesia,
Email : heru.suwoyo@mercubuana.ac.id

ABSTRACT

Wall-following control of mobile robot is an important topic in the mobile robot researches. The wall-following control problem is characterized by moving the robot along the wall in a desired direction while maintaining a constants distance to the wall. The existing control algorithms become complicated in implementation and not efficient enough. Ant colony algorithm (ACA), in terms of optimizing parameters, has a faster convergence speed and features that are easy to integrate with other methods. This paper adopts ant colony algorithm to optimize PID controller, and then selects ideal control parameters. The simulation results based on MATLAB show that the control system optimized by ant colony algorithm has higher efficiency than the traditional control systems in term of RMSE.

INTRODUCTION

Mobile Robot is a comprehensive system that integrates environmental perception, planning, behavior control and execution, and is one of the hottest research fields in recent years [1]. With the in-depth development of control theory, artificial intelligence, computer theory, ultra-large scale computer integrated circuit, sensor technology and other research, the functions of mobile robots have been greatly enriched and their performance is more perfect. They have been widely applied to agriculture, industry, transportation, medical assistance, national defense and military industries [2].

Navigation is a basic function that a mobile robot should have. When in an unknown, complex and dynamically changing environment, the robot is guided to the desired position through exploration of the environment while minimizing consumption (such as time or energy) [3]. Among them, path planning technology is one of the key technologies for mobile robots to realize autonomous positioning and path navigation [4], which is an important research direction in the field of mobile robots. Path planning technology is to search for an optimal or nearly optimal collision-free path from the initial state to the target state according to the task requirements [5]. According to the different degree of mobile robots' knowledge of environmental information, they can be divided into two types: global path planning with completely known environmental information and local path planning with completely unknown or partially unknown environmental information. At present, common path planning methods include A* algorithm, artificial potential field method, RRT algorithm, ant colony algorithm, genetic algorithm, particle swarm algorithm, etc [6-10].

Researchers at home and abroad have carried out relevant research on this. Through the improvement and optimization of the above-mentioned methods, many methods used to solve the problems are put forward, which are mainly divided into two categories: traditional methods and intelligent methods [11]. Literature [12-13] uses artificial potential field method to plan the path of robots. The method is simple and fast, but it is easy to fall into local minima and the target cannot be reached. Literature [14-16] searches for paths in environmental space through fast random trees, which is highly random and theoretically complete, but the generated paths are small broken lines with low smoothness, which is not beneficial to the robot. Literature [17-18] models the environment through grid method, and uses A* algorithm to find the optimal path. This method is a heuristic search algorithm. However, in order to store open set and close set in complex environment, this method takes up a lot of memory overhead and has a slow operation speed. Literature [19-20] uses an improved genetic algorithm for path planning, which has the advantages of strong robustness and implicit parallelism, but is prone to premature convergence. Document [21] uses particle swarm algorithm to carry out path planning, which simulates the foraging behavior of birds. This method has the advantages of less parameter adjustment and fast search speed, but it is prone to premature convergence in the later search period.

There are many path planning methods, but each has its own advantages and disadvantages. In this paper, ACA-based PID controller is used to improve the performance of Wall-Following Robot. PID control is currently the most widely used control strategy. With its simple and clear structure, good robustness and wide application range, PID control is favored by the industry and is increasingly paid attention by the control theory circle. However, the control effect of PID controller is closely related to its parameter setting. There are many conventional PID controller parameter tuning methods, which can be summarized into two categories: one is the experimental trial-and-error method, which mainly relies on debugging experience. The method is simple and easy to master, and is widely used in engineering practice. However, the control effect of the controller parameters obtained by the parameter trial-and-error method is often not very ideal, and it takes time and effort to manually adjust the controller parameters to find a better value; The second is the theoretical calculation setting method, which is mainly based on the mathematical model of the system and determines the controller parameters through theoretical calculation. The calculated data obtained by this method must also be adjusted and modified through engineering practice. Therefore, it is very necessary to optimize the controller parameters by using an optimization algorithm. Ant colony algorithm is a heuristic bionic evolutionary algorithm based on population. The positive feedback mechanism and distributed parallel computing mechanism adopted by the algorithm are easy to combine with other methods, and have strong convergence and robustness, especially suitable for solving combinatorial optimization problems [22]. This paper adopts a PID parameter optimization method based on ant colony algorithm, which greatly improves the convergence speed and the efficiency of the control system.

METHOD AND MATERIAL

Kinematic Configuration and Differential Steering System

In this paper, the proposed algorithm is used to improve the performance of a wheeled mobile robot accomplishing its task, namely following the wall. In order to satisfy the realistic simulation, the real robot (see Figure 2) is modelled based on the kinematic approach. The configuration of this kinematic geometric is presented as follows.

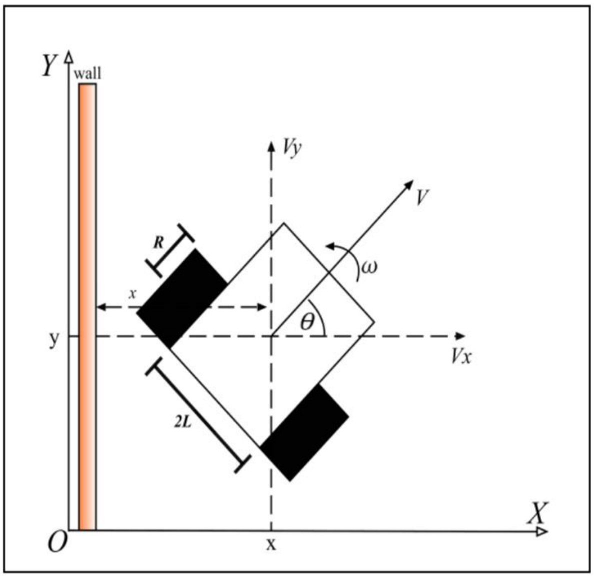

Figure 1. Kinematic Configuration

where, L represents the distance of driven wheel displacement, which is 10 cm, and R refers to the radius of its wheel, which is 3.5 cm. Considering that the robot is placed on the planar cartesian coordinate, the robot pose p can be expressed by the following vector.

$$p(t) = [x(t), y(t), \theta(t)]^T \tag{1}$$

where t represents the discrete time index, and (x, y) is the spatial coordinate of the robot and θ is its heading. Suppose it moves depend on the linear velocity v and the angular velocity ω. Therefore, the dynamic pose of the robot cause by those types of velocity is calculated as follows.

$$\begin{bmatrix} x(t+1) \\ y(t+1) \\ \theta(t+1) \end{bmatrix} = \begin{bmatrix} x(t) \\ y(t) \\ \theta(t) \end{bmatrix} + \begin{bmatrix} \cos\theta(t) & 0 \\ \sin\theta(t) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v(t) \\ \omega(t) \end{bmatrix} \quad (2)$$

where the linear velocity v and angular velocity θ are respectively calculated as follows

$$v = \frac{(\omega_r + \omega_l)}{2R} \quad (3)$$

and

$$\omega = \frac{\omega_r - \omega_l}{2RL} \quad (4)$$

where

$$\omega_r = vR + \omega RL \quad (5)$$

$$\omega_l = vR - \omega RL \quad (6)$$

As noted, both geometry of kinematic configuration and the bundle of equations are based on the real robot form shown in Fig.1.

Figure 2. Real Mobile Robot

Ant Colony Algorithm

A. Introduction to Ant Colony Algorithm

Ant Colony Algorithm (ACA) was first proposed by Marco Dorigo in his doctoral thesis in 1992. Ant colony algorithm is a heuristic search algorithm based on population, and its inspiration comes from the behavior of ants finding paths in the process of searching food. When searching for food sources, ants release a pheromone on their path and can sense pheromones released by other ants. The size of the pheromone concentration indicates the distance of the path. The higher the pheromone concentration, the shorter the distance of the corresponding path. In general, ants will give priority to the path with higher pheromone concentration with greater probability, and release a certain amount of pheromone at the same time to enhance the pheromone concentration on the path, thus forming a positive feedback. Finally, the ant can find the best path from its nest to its food source, that is, the shortest path. Biologists also found that the pheromone concentration on the path would gradually decrease with time.

The ant colony algorithm is applied to solve the optimization problem. Its basic idea is that the feasible solution of the problem to be optimized is represented by the walking path of ants, and all paths of the whole ant colony form the solution space of the problem to be optimized. Ants with shorter paths release more pheromones. As time goes on, the concentration of pheromones accumulated on shorter paths gradually increases, and the number of ants choosing this path also increases. In the end, the whole ant will concentrate on the best path under the action of positive feedback, at which time the corresponding is the optimal solution of the problem to be optimized.

B. Mathematical Principle of Ant Colony Algorithm

Let the number of ants in the whole ant population be m, the number of points be n,t the mutual distance between point I and point j be $d_{ij}(i,j = 1,2,\dots,n)$, and the pheromone concentration on the connection path between point I and point j at time t be. At the initial time, the pheromone concentration on the connection path between each point is the same, and may be set $to = \tau 0$. Ant $k(k = 1,2,\dots,m)$ determines its next access point according to the pheromone concentration on the connection path between each point, and sets the probability of ant k transferring from point I to point j at time t, and its calculation formula is as follows:

$$P_{ij}^k = \begin{cases} \dfrac{[\tau_{ij}(t)]^\alpha . [\eta_{ij}(t)]^\beta}{\sum_{s \in allow_k} [\tau_{is}(t)]^\alpha . [\eta_{is}(t)]^\beta} & s \in allow_k \\ 0 & s \notin allow_k \end{cases} \quad (7)$$

where, $\eta_{ij}(t)$ is the heuristic function, $\eta_{ij}(t) = 1/d_{ij}$, indicates the expected degree of ant transfer from city i to city j. Meanwhile, $allow_k$ is the set of points to be visited by ant $k = 1,2,\dots,m$. At the beginning, there were $(n-1)$ elements in $allow_k$, including all points except the starting point of ant k. As time goes on, the elements in $allow_k$ are continuously reduced until they become empty sets, which means that all points have been visited. The variable α is the pheromone importance factor, and the larger the value, the greater the effect of pheromone concentration on the transfer. The variable β represents the importance factor of the heuristic function, and the larger its value is, the greater the role of the heuristic function in migration, i.e., ants will transfer to shorter distance points with greater probability.

At the same time when ants release pheromones, the pheromones in the connection path between each point gradually disappear, and the parameter $0 < \rho < 0$ is set to indicate the volatilization degree of pheromones. Therefore, after all ants complete a cycle, the pheromone concentration on the routes between cities needs to be updated in real time, with the specific formula as follows:

$$\begin{cases} \tau_{ij}(t+1) = (1-\rho) * \tau_{ij}(t) + \Delta\tau_{ij} \\ \Delta\tau_{ij} = \sum_{k=1}^{n} \Delta\tau_{ij}^{k} \end{cases}, 0 < \rho < 1 \quad (8)$$

where, $\Delta\tau_{ij}$ represents the pheromone concentration released by the k-th ant on the connection path of point I and point j, and $\Delta\tau_{ij}$ represents the sum of the pheromone concentrations released by all ants on the connection path of point I and point j.

C. Algorithm steps

The algorithm steps of Ant Colony Optimization are listed as follows;

1. Initialization parameters

   At the beginning of calculation, relevant parameters need to be initialized, such as ant colony size m, pheromone importance factor α, heuristic function importance factor β, pheromone volatilization factor ρ, total pheromone release quantity q, maximum iteration number $iter_{max}$, initial iteration number $iter = 1$.

2. Constructing a solution space

   Each ant is randomly placed at different starting points. For each ant k $(k = 1,2,\dots,m)$, the points visited by the next generation are calculated until all ants have visited all the points.

3. Update pheromone

   Calculate the path length $(k = 1,2,\dots,m)$ of each ant, and record the optimal solution (shortest path) in the current iteration number. At the same time, the pheromone concentration on the connection path of each point is updated.

4. Judging whether to terminate or not

   If $iter < iter_{max}$, make $iter = iter + 1$, empty the record table of ant paths, and return to step (2), otherwise, stop the calculation and output the optimized solution.

These steps can be clearly seen from Fig. 3.
Characteristics of Ant Colony Algorithm

In order to ease the process of designing optimization strategy based on Ant Colony Algorithm, the following points have been commonly concerned by scholars [23, 24].

1. Positive feedback mechanism is adopted to make the search process converge continuously and finally approach the optimal solution;
2. Each individual can change the surrounding environment by releasing pheromones, and each individual can sense the real-time changes of the surrounding environment, and individuals communicate indirectly through the environment (pheromones);
3. The search process adopts a distributed computing method, and multiple individuals perform parallel computing at the same time, thus greatly improving the computing capability and operation efficiency of the algorithm;
4. Heuristic probability search method is not easy to fall into local optimum, and is easy to find global optimum solution.

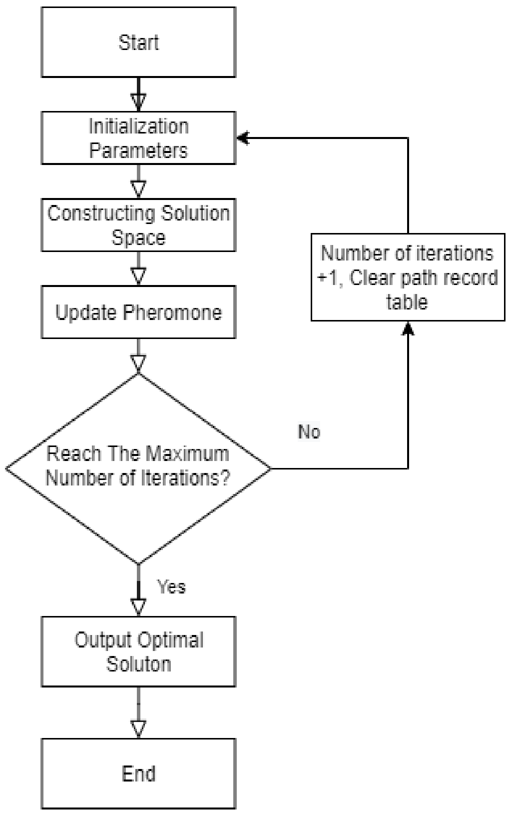

Figure 3. Flowchart of Ant Colony Algorithm

PID Parameter Optimization Based on Ant Colony Algorithm

Using ant colony algorithm to optimize PID parameters is to find the optimal values of K_p, K_i and K_d. K_p, K_i and K_d are taken as a combination. Ants continuously adjust the path in the search space under the guidance of pheromone, and finally find the optimal path. The value corresponding to the optimal path is the optimal parameter of PID controller. The system block diagram is shown in Fig. 3 below.

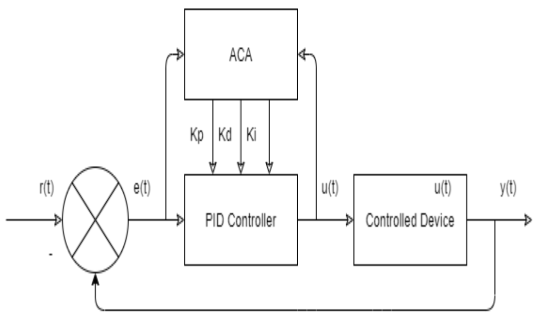

Figure 4. PID Control System Block Diagram Base on Ant Colony Algorithm

The establishment of nodes and paths

In order to realize the ant optimization process, the nodes and paths needed by ant optimization must be established. K_p, K_i and K_d are taken as parameters to be optimized, so that these three variables have 5 significant digits, and K_p, K_i and K_d are all accurate to 4 decimal places. In order to describe the method more intuitively, the above three parameters are put into the two-dimensional plane XOY for description. Detailed description is as follows: Draw 15-line segments $L_1, L_2, \dots, L_{15}$ with equal spacing and perpendicular to the x-axis, as shown in Fig. 4. The line segments 1 to 5, 6 to 10, and 11 to 15 correspond to the values in the first to fifth digits of K_p, T_i, and T_d, respectively. In this way, the range of x-axis variable is expressed as 1 to 15. At the same time, divide these line segments 9 equally, so that the variable range of the y-axis is 0 to 9. Nodes on the plane are represented by $knot(xi, yi, j)$, where x_i is the abscissa of L_i and y_i, j is the ordinate of L_i node j. If ant K starts from the origin and when it crawls to any point on L_{15} segment, it completes a cycle, the crawling path of the ant can be expressed as:

$$
\begin{aligned}
&\text{starting point} \to knot(x_1, y_{1,j}) \to knot(x_2, y_{2,j}) \to knot(x_3, y_{3,j}) \to knot(x_4, y_{4,j}) \\
&\to knot(x_5, y_{5,j}) \to knot(x_6, y_{6,j}) \to knot(x_7, y_{7,j}) \to knot(x_8, y_{8,j}) \\
&\to knot(x_9, y_{9,j}) \to knot(x_{10}, y_{10,j}) \to knot(x_{11}, y_{11,j}) \to knot(x_{12}, y_{12,j}) \\
&\to knot(x_{13}, y_{13,j}) \to knot(x_{14}, y_{14,j}) \to knot(x_{15}, y_{15,j})
\end{aligned}
$$

K_p, K_i and K_d can be found according to equation (11):

$$
\begin{cases}
Kp = y_{1,j} \times 10^0 + y_{2,j} \times 10^{-1} + y_{3,j} \times 10^{-2} + y_{4,j} \times 10^{-3} + y_{5,j} \times 10^{-4} \\
Ki = y_{6,j} \times 10^0 + y_{7,j} \times 10^{-1} + y_{8,j} \times 10^{-2} + y_{9,j} \times 10^{-3} + y_{10,j} \times 10^{-4} \\
Kd = y_{11,j} \times 10^0 + y_{12,j} \times 10^{-1} + y_{13,j} \times 10^{-2} + y_{14,j} \times 10^{-3} + y_{15,j} \times 10^{-4}
\end{cases} \tag{11}
$$

From equation (11), K_p, K_i and K_d in Fig. 4 can be calculated as 42.546, 3.5137 and 6.4354 respectively

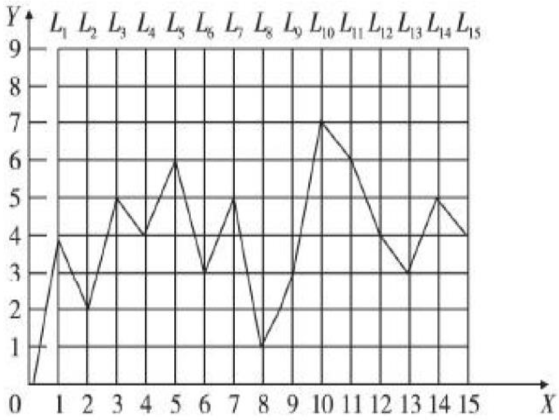

Figure 5. Node and Path Generation Schematic Diagram

Path Selection and Pheromone Update

Ants constantly select and adjust paths in the search space under the guidance of pheromones. Proper methods for both the state transition probability and pheromone update determine the performance of the algorithm to a large extent.

1) The state transition probability of the path is as follows:

$$P(x_i,y_{ij},t) = \frac{\tau[x_i,y_{ij},t]^{\alpha}\eta[x_i,y_{ij},t]^{\beta}}{\sum_{y_{ij}=0}^{9}\tau[x_i,y_{ij},t]^{\alpha}\eta[x_i,y_{ij},t]^{\beta}} \quad (12)$$

In formula (6): t is the current time, $\tau(x_i,y_{ij},t)$ is the pheromone left over from node $C(x_i,y_{ij})$ at time t. Meanwhile, $\eta(x_i,y_{ij},t)$ is the visibility of information on node $C(x_i,y_{ij})$ at time t, determined according to equation (10), α is the importance of residual information, β is the importance of heuristic information.

$$\eta(x_i,y_{ij},t) = \eta(x_i,y_{ij},t) + \Delta\eta(x_i,y_{ij},t) \quad (13)$$

In equation (13), it is the change amount of information visibility on node $knot(x_i,y_{ij})$ at time t, and is determined according to equation (14):

$$\Delta\eta(x_i,y_{ij},t) = \frac{10-|y_{ij}-y_{ij}^{best}|}{10} \quad (14)$$

In equation (14), y_{ij}^{best} is the ordinate of each node corresponding to the current optimal path.

2) The update of node pheromone is shown in equation (15):

$$\tau(x_n,y_{nj},t) = (1-\gamma)\tau(x_n,y_{nj},t) + \Delta\tau(x_n,y_{nj},t) \quad (15)$$

In formula (9), γ is the pheromone volatilization coefficient, $\Delta\tau(x_n,y_{nj},t)$ is the total change amount of pheromone on node $C(x_i,y_{ij})$ at time t, and is determined according to equation (16) ：

$$\Delta\tau(x_n,y_n,t) = \sum_{x_i=1}^{x_n}\Delta\tau(x_i,y_{ij},t) \quad (16)$$

In equation (10), $\Delta\tau(x_i,y_{ij},t)$ is the change amount of pheromone on node $C(x_i,y_{ai})$ after each ant climbs over, as shown in equation (16):

$$\Delta\tau(x_i,y_{ij},t) = \frac{Q}{|y_{ij}-y_{ij}^{best}|+1} \quad (17)$$

In Formula (17), Q is the pheromone concentration.

PID Parameter Optimization Process Based on Ant Colony Algorithm

1) Initialization

Generating a node matrix, setting an ant colony scale m, an importance degree α of legacy pheromones, an importance degree β of heuristic information, a pheromone volatilization coefficient gamma, a pheromone intensity q, and a maximum iteration number NC_{max}; Setting the remaining pheromone τ and the pheromone visibility η as constants;

2) Optimization

1. Putting ants at the origin and starting crawling, and calculating the state transition probability p of each node to be accessed according to (12);
2. Generates random numbers satisfying [0,1], looks for nodes whose state

transition probability is greater than the random numbers, and selects the first node as the next crawling node;

3. When the ant climbs any point on the L15 line segment, completing a cycle and recording the ordinate of the climbed node;
4. When all ants complete a crawl, calculate K_p, K_i and K_d according to (11) and assign them to PID controller;
5. Operating the control system model to obtain the objective function value, returning and recording the optimal result;
6. Update pheromone according to (15), $NC = NC + 1$;
7. Entering the next cycle until NC_{max} is reached, and outputting optimal K_p, K_i and K_d. The flow chart is as follows:

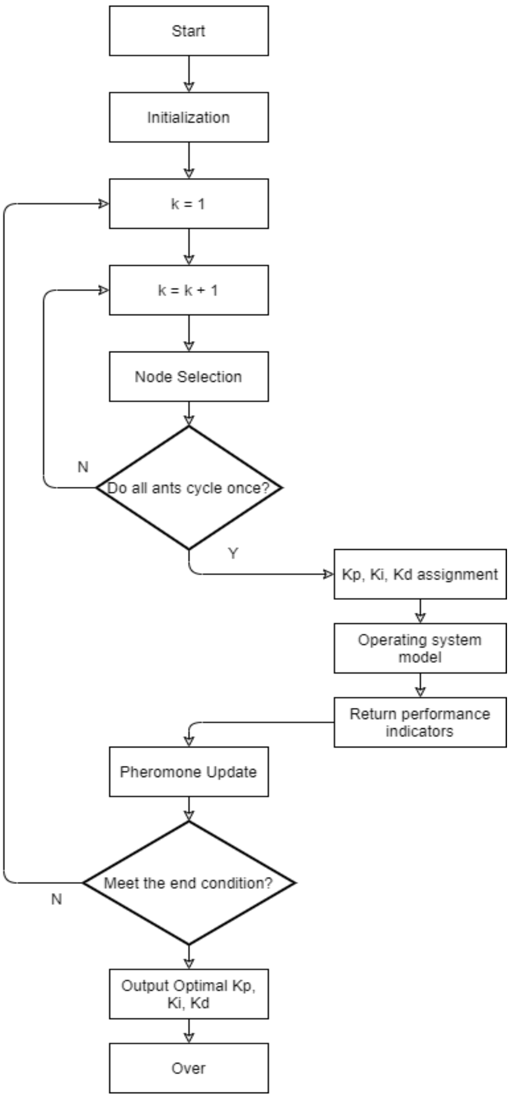

Figure 6. PID Parameter Optimization Flowchart Based on Ant Colony Algorithm

RESULTS AND DISCUSSION

The simulation experiment is carried out in MATLAB. The ant colony algorithm M-file is used to call the PID control model, assign values to K_p, K_i and K_d, and the PID control system model is run and the objective function value is returned, which provides the basis for the ant colony algorithm to judge the current result and find the optimal result.

The initial parameters are set as follows: ant number $m = 50$, maximum iteration number $iter_{max} = 300$, pheromone importance factor $\alpha = 1$, heuristic function importance factor $\beta = 5$, pheromone volatilization factor $\gamma = 0.1$, and pheromone intensity $Q = 0.5$. The parameters K_p, K_i and K_d all retain 5 significant digits. According to experience, the solution is $K_p = 0.04311$, $K_i = 0.07023$ and $K_d = 0.19805$ with the fitness value of 10.2104 obtained before the number of generations reaches 50.

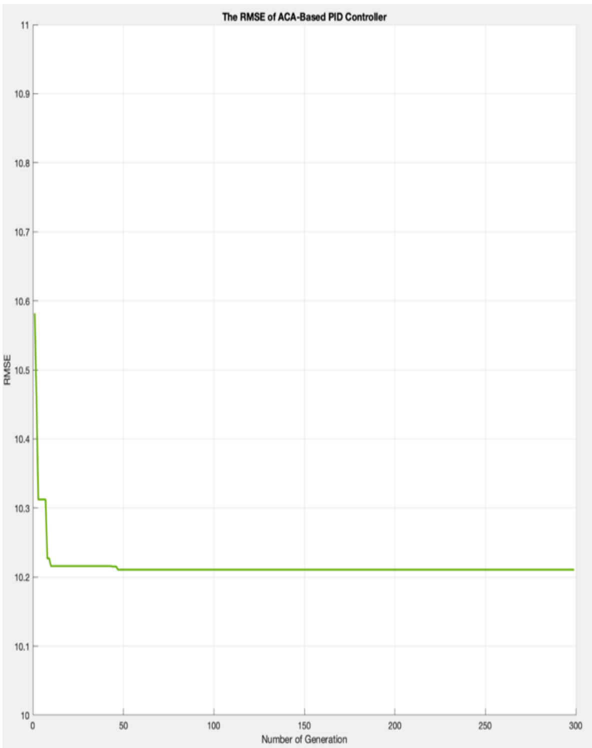

Figure 7. Convergence Curve of PID Controller Base on ACA

In order to verify the superiority of the method proposed in this paper, the above method is compared with genetic algorithm to optimize PID parameters. And based on our previous achievement [24], the solution is $K_p = 0.05431$, $K_i = 0.02023$ and $K_d = 0.25406$ with the fitness value of 10.234 obtained after the number of generations reaches 238.

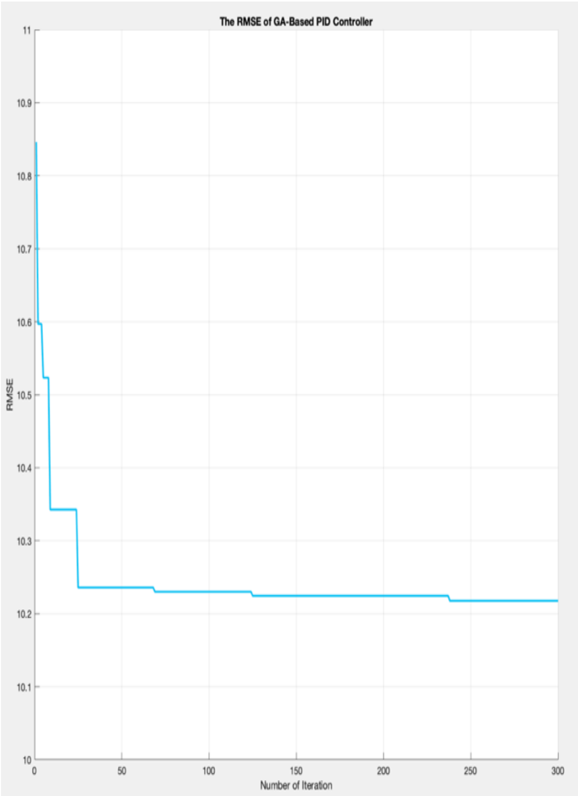

Figure 8. Convergence Curve of PID Controller Base on GA

As note, the fitness function used as benchmark in Fig. 6 and Fig.7 is the Root Mean Square Error for a complete process of a wheeled mobile robot. It is calculated as (12)

$$RMSE = \frac{1}{t_{max}} \sum_{t=1}^{t_{max}} e(iter)^2 \qquad (18)$$

Where $e(iter)^2$ represent the square error of each movement of the robot based on the sensed distance and the setpoint 0.

Finally, using optimal solution given by ACA, the performance of a wall-following robot can be graphically presented as follows

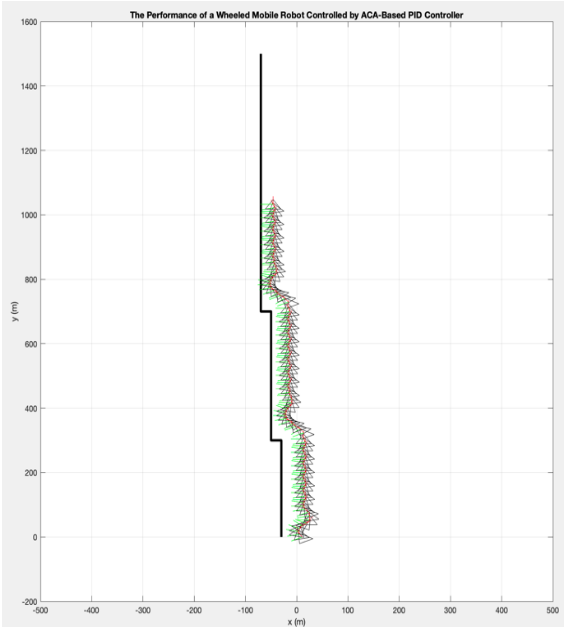

Figure 9. Path Diagram of Wall Following Robot (Using ACA Based PID Controller)

As can be seen from Fig. 8, the performance of a wheeled mobile robot satisfying the smoothness. It has been overcoming the wobble problem of the conventional performance. This graph also proves that it is better than GA-based PID Controller for the same generation number, $iter_{max} = 300$.

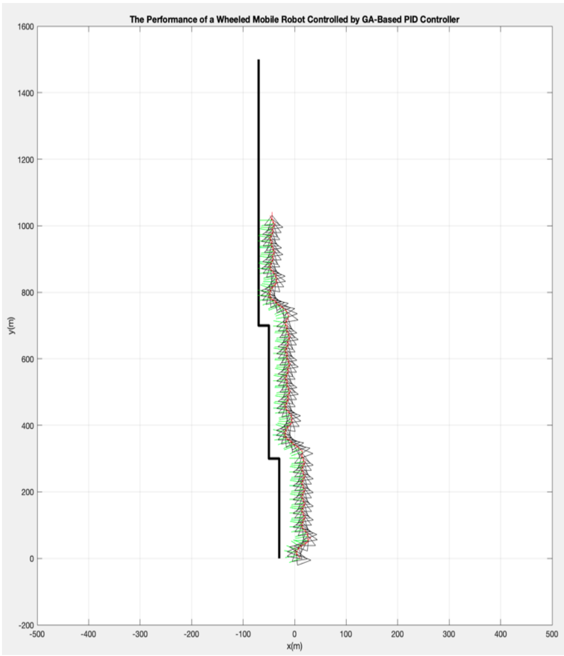

Figure 10. Path Diagram of Wall Following Robot (Using GA Based PID Controller)

As shown in Fig 6 and 7, optimizing PID parameters with genetic algorithm requires at least 30 iterations to stabilize, while optimizing PID parameters with ant colony algorithm requires only about 10 times to achieve the same purpose. In addition, by observing the dynamic process of convergence of the two algorithms, it can be observed that compared with optimizing PID parameters with genetic algorithm, optimizing PID parameters with ant colony algorithm can make the convergence speed faster. This result shows that the adjustment time of PID control system optimized by ant colony algorithm is shorter.

As shown in the figure 8 and 9, by comparing the path of two mobile robots along the wall, it can be seen that the path of ant colony algorithm optimizing PID parameters is smoother, with less fluctuation at the corner of the wall and shorter adjustment time. This result shows that the PID control system optimized by ant colony algorithm has better stability and shorter adjustment time. Besides from Fig.8 and Fig.9, the comparation can also be analyzed from Fig.10, which depicts the error of wheeled mobile robot with respect to the step integration.

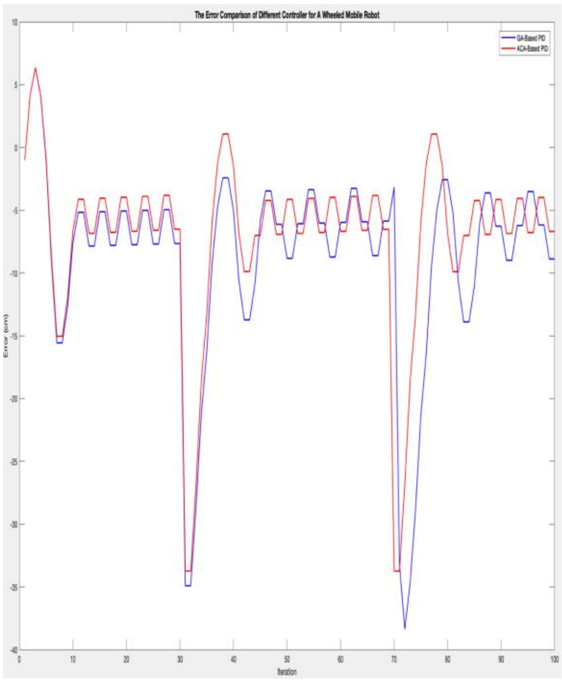

Figure 11. The Error Value (cm) for Wheeled Mobile Robot in Each Step

The codes are tested MATLAB on Processor 2.3 GHz Dual-Core Intel Core i5 with Memory 8 GB 2133 MHz LPDDR3. Elapsed time for each generation is 5.375713 second for ACA-based Controller and 60.441706 second for GA-based PID Controller.

CONCLUSION

In this paper, ant colony algorithm and PID controller are combined to improve the performance of Wall-Following Robot. Through the simulation experiment on MATLAB platform, compared with the PID control system optimized by genetic algorithm, the PID parameter tuning method based on ant colony algorithm can quickly and accurately find the optimal parameters of PID controller, with better responsiveness and stability, which verifies the feasibility of the method. In addition, the tuning method also provides an effective alternative method for manual empirical adjustment of PID controller parameters.

ACKNOWLEDGMENT

The research was supported by Special Plan of Major Scientific Instruments and Equipment of the State (Grant No.2018YFF01013101), National Natural Science Foundation of China (61704102), the IIOT Innovation and Development Special Foundation of Shanghai (2017- GYHLW01037), Project named “Key Technology Research and Demonstration Line Construction of Advanced Laser Intelligent Manufacturing Equipment” from Shanghai Lingang Area Development Administration, Universitas Mercu Buana, Jakarta, Indonesia.

REFERENCES

[1] Mohamed, A., El-Gindy, M., & Ren, J. (2018). Advanced control techniques for unmanned ground vehicle: literature survey. International journal of vehicle performance, 4(1), 46-73.

[2] Kai, Z., Shengjin, L., & Juan, S. (2013). Research on Improved Ant Colony Algorithm in mobile robot path planning [J]. Microcomputer and application, 32(4), 67-70.

[3] Mu, Y. Z., Dong, C. Y., Chen, Q. M., Li, B. C., & Fan, Z. Q. (2020, April). Research on Navigation and Path Planning of Mobile Robot Based on Vision Sensor. In Proceedings of the 2020 6th International Conference on Computing and Artificial Intelligence (pp. 519-524)

[4] Bin-Qiang, Y., Ming-Fu, Z., & Yi, W. (2011, July). Research of path planning method for mobile robot based on artificial potential field. In 2011 International Conference on Multimedia Technology (pp. 3192-3195). IEEE.

[5] Zhang, H. D., Zheng, R., & Cen, Y. W. (2005). Present situation and future

development of mobile robot path planning technology. Acta Simulata Systematica Sinica, 17(2), 439-443.
[6] Xin, Y., Liang, H. W., Du, M. B., Mei, T., Wang, Z. L., & Jiang, R. H. (2014). An improved A* algorithm for searching infinite neighbourhoods. Robot, 36(5), 627-633.
[7] Jian-Ying, Z. H. A. N. G., & Tun, L. (2007). Optimized path planning of mobile robot based on artificial potential field. Acta Aeronaut. Astronaut. Sinica, 28(2), 183-188.
[8] Jing, Y., Jiao, M., Chen, Y., Zheng, W., Huang, J., & Niu, B. (2020, July). Mobile Robot Path Planning Based on Improved Ant Colony Optimization Algorithm. In 2020 39th Chinese Control Conference (CCC) (pp. 1559-1564). IEEE.
[9] Wang, G., Wang, H., Li, Z., & Li, J. (2019). Path planning for mobile robots based on optimized genetic algorithm. Machine Tool & Hydraulics, Papers, 47(03), 44-47.
[10] Zhi-yong, C. H. E. N. (2017). Research on Multi-objective Vehicle Route Program Based on IPSO Algorithm. Journal of Shandong Agricultural University (Natural Science Edition), 02..
[11] Yang, J. C., Li, S. X., & Cai, Z. Y. (2017). Research and development of path planning algorithm [J]. Control engineering, 24(07), 1473-1480. (in Chinese)
[12] HAN, Y., ZHANG, L., TAN, H., XUE, X., GUO, R., & GUO, Q. Mobile robot path planning based on improved particle swarm optimization algorithm.
[13] Gao, T. T., Zhang, L., Li, B. D., & GAO, Q. (2016). Study on path planning of soccer robot based on improved particle swarm algorithm. Journal of Xi'an Polytechnic University, 30(5), 609-615. (in Chinese)
[14] Cui, R., Li, Y., & Yan, W. (2015). Mutual information-based multi-AUV path planning for scalar field sampling using multidimensional RRT. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 46(7), 993-1004.
[15] Liu, C., Han, J., & An, K. (2017). Dynamic path planning based on an improved RRT algorithm for RoboCup robot. Robot, 39(1), 8-15. (in Chinese)
[16] Zhenghua, H., Qiubo, Z., Chunya, T., & Kuai, W. (2017, August). Application of rapidly exploring random tree algorithm in the scene of road network with scheduled tracks. In 2017 2nd International Conference on Advanced Robotics and Mechatronics (ICARM) (pp. 311-315). IEEE.,
[17] Zhao, X., Wang, Z., Huang, C. K., & Zhao, Y. W. (2018). Mobile robot path planning based on an improved A* algorithm. Robot, 40(6), 903-910. (in Chinese)
[18] Guruji, A. K., Agarwal, H., & Parsediya, D. K. (2016). Time-efficient A* algorithm for robot path planning. Procedia Technology, 23, 144-149.
[19] Wang, L., Li, M., Tang, D. B., & Cai, J. (2016). Dynamic path planning for mobile robot based on improved genetic algorithm. Journal of Nanjing University of Aeronautics and Astronautics, 48(6), 841-846. (in Chinese)
[20] Song, B., Wang, Z., & Sheng, L. (2016). A new genetic algorithm approach to smooth path planning for mobile robots. Assembly Automation.
[21] Das, P. K., Behera, H. S., & Panigrahi, B. K. (2016). A hybridization of an improved particle swarm optimization and gravitational search algorithm for multi-robot path planning. Swarm and Evolutionary Computation, 28, 14-28.
[22] Yin, H. P., & Chai, Y. (2007). Parameters optimization design of PID controller based on ant colony algorithms. Jisuanji Gongcheng yu Yingyong(Computer Engineering and Applications), 42(17), 4-7..
[23] Jacknoon, A., & Abido, M. A. (2017, January). Ant Colony based LQR and PID tuned parameters for controlling Inverted Pendulum. In 2017 International Conference on Communication, Control, Computing and Electronics Engineering (ICCCCEE) (pp. 1-8). IEEE.
[24] Suwoyo, H., Tian, Y., Deng, C., & Adriansyah, A. (2018, October). Improving a wall-following robot performance with a PID-genetic algorithm controller. In 2018 5th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI) (pp. 314-318). IEEE