Open Access

*Iraqi Journal for Electrical and Electronic Engineering*
*Original Article*

# Design and Implementation of Line Follower Arduino Mobile Robot Using Matlab Simulink Toolbox

**Mazin Majid Abdulnabi Alwan, Anwar Abdulrazzaq Green, Abdulazez Safaa Noori and Ammar A. Aldair \***
Electrical Engineering Department, University of Basrah, Basrah, Iraq

**Correspondence**
\* Ammar A. Aldair
Electrical Engineering Department,
University of Basrah, Basrah, Iraq
Email: ammar.abdulhameed@uobasrah.edu.iq

**Abstract**
*The main problem of line follower robot is how to make the mobile robot follows a desired path (which is a line drawn on the floor) smoothly and accurately in shortest time. In this paper, the design and implementation of a complex line follower mission is presented by using Matlab Simulink toolbox. The motion of mobile robot on the complex path is simulated by using the Robot Simulator which is programed in Matlab to design and test the performance of the proposed line follower algorithm and the designed PID controller. Due to the complexity of selection the parameters of PID controller, the Particle Swarm Optimization (PSO) algorithm are used to select and tune the parameters of designed PID controller. Five Infrared Ray (IR) sensors are used to collect the information about the location of mobile robot with respect to the desired path (black line). Depending on the collected information, the steering angle of the mobile robot will be controlled to maintain the robot on the desired path by controlling the speed of actuators (two DC motors). The obtained simulation results show that, the motion of mobile robot is still stable even the complex maneuver is performed. The hardware design of the robot system is perform by using the Arduino Mobile Robot (AMR). The Simulink Support Package for Arduino and control system toolbox are used to program the AMR. The practical results show that the performances of real mobile robot are exactly the same of the performances of simulated mobile robot.*

**KEYWORDS: Arduino Mobile Robot, Line follower, Infrared Ray (IR) sensors, Particle Swarm Optimization(PSO) algorithm, PID controller.**

## I. INTRODUCTION

Nowadays, the researchers have made significant effort in the field of line follower robot because it is used in many applications such as: medical assistance, transportation, educational, marketing, and industrial applications. The line follower mobile robot is one of the types of the autonomous robot that can perform a specific desired mission without human intervention. The operations of the line follower robot can be conclude as follow: the robot used the IR sensors, which are mounted at front ends of the robot, to capture the line position that is drawn on the floor. Depending on the IR sensors reading, the control system will send the command signals to the actuators (DC motors) to govern the steering angle of the mobile robot to track the line smoothly and accurately in shortest time. So that, the sensing process of the line requires high robustness and high resolution.

In reference [1], the authors designed a line follower robot for public transport to solve the problems of number of passengers in Turkey. The web server is used to control the motion of line follower mobile robot to track the desired trajectory, where the PID controller is used to perform this mission [2]. The IR sensor to collect the data and it to the microcontroller, then the microcontroller generate the suitable signals to actuate the motors to and maintain the robot follow the desired path [3-5].

The main issue in the design of the line follower mobile robot is the design the suitable controller to govern the steering angle of the robot to keep it on the desire path. The PID control scheme is used successfully in many industrial applications in last five decades [6]. The PID controller is most widely used in the industrial application over the other controller types due to the following reasons: its structure is simple, robustness in wide range of operation, and just three parameters should be adjusted to design its structure [7]. The PID controller system has three parameters that should be tuned to enhance the performance of the controlled plant. The task of adjusting the parameters of PID controller is quite difficult and there are many method to perform this task. The Ziegler–Nichols tuning method is

a heuristic method of tuning a PID controller. It was developed by John G. Ziegler and Nathaniel B. Nichols [8]. Unfortunately, this classical method does not return an optimal parameters for the PID controller, so that by using this method to design the controller, the performance of controlled system is still insufficient. Some other classical methods are used to tune the parameters of PID controller such as Cohen-Coon method, rule-based method and model-based method. Each of those methods has its advantages and drawbacks [9]. The drawbacks of those methods are they utilize only for first order models including large process delays and they require experienced persons in industrial applications.

In the recent decade, the researchers focused on developing and proposing the optimization methods to obtain the optimal parameters for PID controller to enhance the performances of the controlled system. The emergence of intelligence and optimization algorithms such as genetic algorithm (GA), Particle Swarm Optimization (PSO) method, Ant Colony Optimization (ACO) method, provides new techniques to tuning the PID parameters successfully [10].

In this paper, the line follower algorithm is proposed and the PID controller is designed for the mobile robot. The PSO algorithm has been applied to find the optimal parameters of the designed PID controller. Undoubtedly, the cost of line follower mobile robots are relatively expensive. Hence, a precise controller should be properly designed and it performance should be thoroughly studied before buying a physical robot. Therefore, many simulation programs are designed to help the researchers to evaluate the performance of designed controlled system. For such reasons, the usage of a simulator program becomes advantageous as it can save time and cost effectively. Therefore, in this study, the Robot Simulator and Simulink package are used to simulate the tracking of the controlled mobile robot for the desired path that is drawn on the floor. The obtained simulation results proved that the performance of the designed PID controller is very accurate. Then, the Arduino Mobile Robot (AMR) is implemented and programed to follow the desired trajectory. Five Infrared Ray (IR) sensors are used to collect the information about the location of mobile robot with respect to the desired path (black line). Depending on the collected information, the steering angle of the mobile robot will be controlled to maintain the robot on the desired path by controlling the speed of actuators (two DC motors). The ardouin and control system toolboxes (PID controller Blocks), which are constructed in Matlab, are used to program the AMR. The practical results show that the performances of real mobile robot are exactly the same of the performances of simulated mobile robot.

## II. LINE FOLLOWER MOBILE ROBOT

The Line Follower Mobile Robot (LFMR) is an autonomous robot that can capture the desired path drawn on the floor. The LFMR is used many application such as military application, hospital services application, delivery services system, transportation systems and blindness helping systems. The LFMR uses several optical sensors

(such as IR sensors) that are fixed on the front end of the robot to detect the existence of the drawn line. The operation principle of the IR sensor depends on the Planck's radiation law [10]. The portions of reflection, absorption and scattering light depend on the color of the indecent surfaces. When the light ray falls on the black surface, the surface will absorb all the power of the light and no light will reflect back. Whereas, if the light ray is incident on white surface, the surface will reflect back all the amount of the light. For colored surfaces, some amount of the light absorbs and the rest reflects back. Figure (1) shows the operation principle of IR sensor.
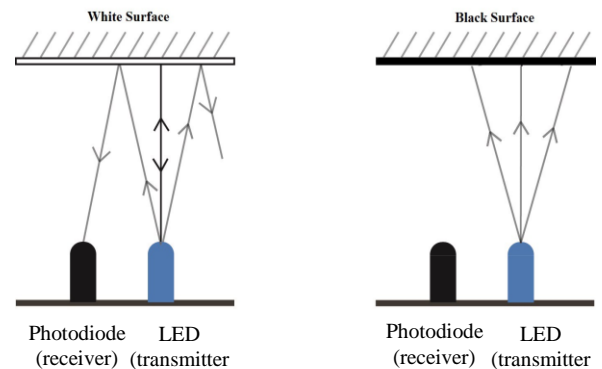


Fig. (1) The operation principle of IR

The IR sensor uses this principle to recognize the black line that is drawn on the white surface. The IR sensor has two main parts: LED (transmitter) and photodiode (receiver). The LED sends the light to detect the color on the surface under the robot. If large amount of the light return back to the photodiode that is mean the white surface is detected. While, if small amount of the light return back to the photodiode that is mean the black line is detected. The collected data from IR sensors will send to the microcontroller to generate the suitable control signals that is used to steer the LFMR and maintain it on the desired path. To stabilize the tracking motion, the PID controller is used. In this work, the Arduino Mobile Robot (AMR) is used. Figure (2) shows the out view of AMR. The AMR comes with a different types of sensors such as: a digital compass and five floor sensors and a two potentiometers. It also has two DC motors with built in motor drive. The Simulink Package for Arduino Hardware that is installed with Matlab Simulink is used to control all these sensors and actuators. The AMR has Leonardo (ATmega32u4) based boards: the Control Board (top) and the Motor Board (bottom) as shown in Figure (3).
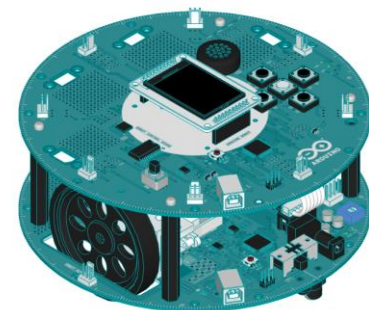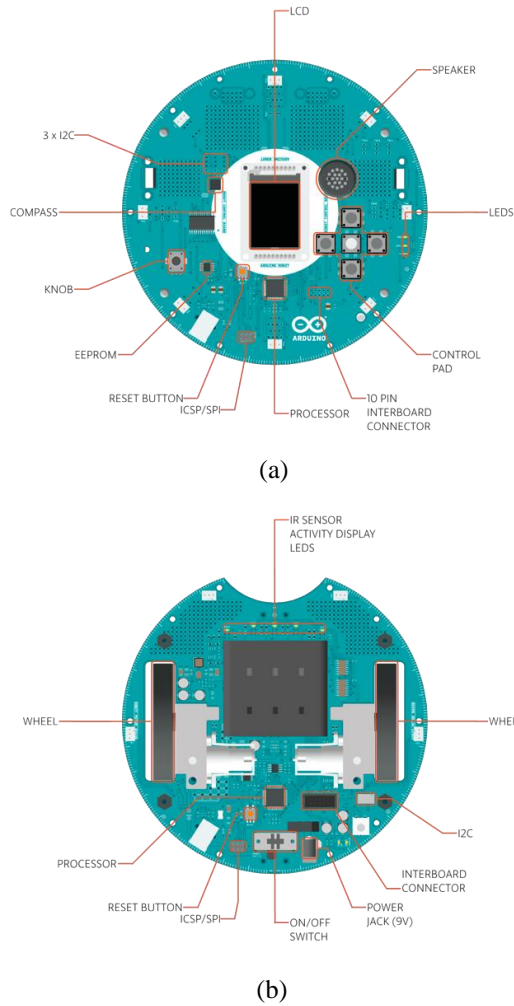


Fig. (2) The out view of AMR

named (SA, SB and SC) are connected to digital output pins of the ATmega32u4 microcontroller. Therefore, microcontroller is read just one IR sensor at a time that is applied to analog input pin INA. Table 1 shows the selected IR sensor reading according to the selectors command.
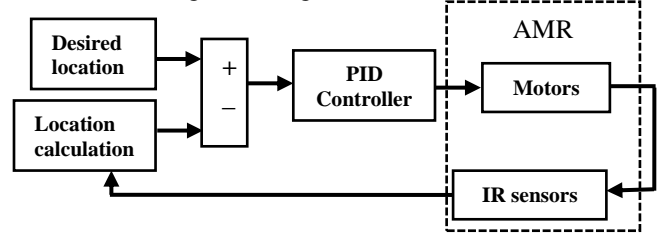


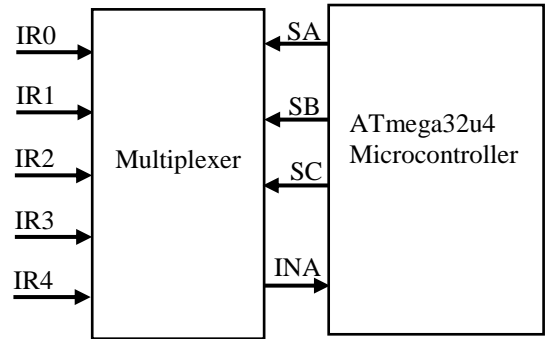Fig. (4) the mechanism of line follower for the AMR



Fig. (5) the connection of the IR sensors and ATmega32u4 controller via multiplexer

TABLE (1)

The selected IR sensor reading according to the selectors command

| SA | SB | SC | IR sensor NO. |
|----|----|----|---------------|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 2 |
| 0 | 1 | 1 | 3 |
| 1 | 0 | 0 | 4 |

The connection of ATmega32u4 microcontroller with two wheel motor via Motor Driver is shown in Figure (6). The speed and direction of the left and right wheel are adjusted by the motor drive. Four digital output pins (OA1, OA2, OB1 and OB2) of the microcontroller are connected to inputs of motor driver (INA1, INA2, INB1 and INB2). The inputs INA1 and INB1 of the motor driver corresponds to the forward movement of the left and right motor respectively. INA2 and INB2 corresponds to the reverse movement of the left and right motor respectively. A pulse width modulated (PWM) signals are used to control the motors. A positive input to the motor driver from the microcontroller corresponds to forward movement of the motors, whereas a negative input corresponds to reverse movement of the motors. For forward movement of both the motors, you have to apply the PWM signals to INA1 and INB1 and zero values to INA2 and INB2. Similarly for reverse movement, you have to apply the PWM signals only to INA2 and INB2 and zero values to INA1 and INB1.

The line follower algorithm is constructed as closed loop controlled system with PID controller. The parameters



Fig. (3) (a) The Control Board, (b) The Motor Board

In this work, Simulink model is used to program and run the AMR as line follower by applying the line follower algorithm on the motor board and accessing the IR sensors and motors.

### III. THE LINE FOLLOWER ALGORITHM

Figure 4 shows the mechanism of line follower for the AMR. The current location of the mobile robot to the respect to the center black line is recognized and calculated by the reading of IR sensors to compute the error between the current location and the desired path. Then the computed error is used as input to the digital PID controller. The Pulse Width Modulated (PWM) control signals are generated to control and drive the motors connected to the Motor Board of AMR to adjust the steering angle of the robot and to bring it back to the center of the black line.

The Multiplexer logic circuit is used to interface between ATmega32u4 Microcontroller of the Arduino board and the five IR sensors to read just one data that come from IR sensors set according to command signal sent from ATMega32u4 controller. Figure 5 shows the connection of the IR sensors and ATmega32u4 microcontroller via multiplexer. The five IR sensors (IR0-IR4) are connected to the inputs of multiplexer. The selectors of the multiplexer

of PID controller is tuned by using PSO algorithm. The control action depends on the collected data from IR sensors. Figure 6 shows the description of the line follower algorithm.

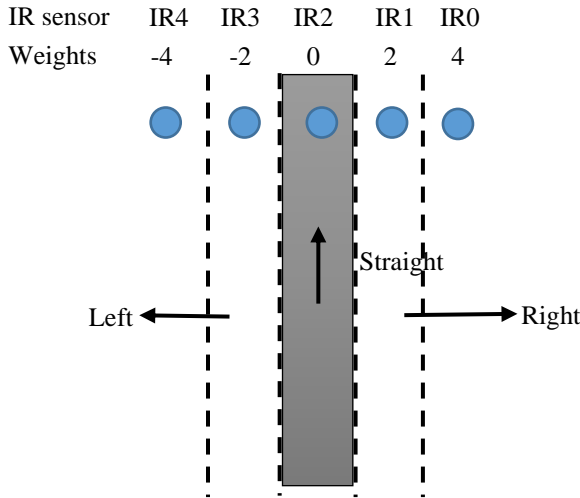| IR sensor | IR4 | IR3 | IR2 | IR1 | IR0 |
|-----------|-----|-----|-----|-----|-----|
| Weights | -4 | -2 | 0 | 2 | 4 |



Fig. 6 the description of the line follower algorithm

As shown in the Fig. 6 the AMR is located at the center of black line and the five IR sensors is represented by blue dots. In this case, just the IR2 reads logic 0 (black background) and the rest of IR sensors read white logic 1 (white background). Each sensor has weight value, as shown in Fig. 6, with higher values to the extreme ones providing higher control signals when higher deviation is detected. The weighted Current Location (CL) of the AMR is computed as follow:

$$CL = \frac{\sum_{i=0}^{4}(\text{reading of } i^{th} \text{ sensor} * \text{weight of } i^{th} \text{sensor})}{\text{number of sensors reading } 0} \quad (1)$$

According to the reading of IR sensors, the microcontroller generates the PWM control signal to the motor drive to control the steering angle of the AMR as shown in Table 2:

TABLE 2
Different conditions for movement of AMR

| IR sensor reading | Command | Motion of Left Motor | Motion of Right Motor |
|-------------------|---------|---------------------|----------------------|
| 10001 | Straight | Forward | Forward |
| 11011 | Straight | Forward | Forward |
| 11111 | Straight | Forward | Forward |
| 10000 | Turn Right | Forward | Backward |
| 11000 | Turn Right | Forward | Backward |
| 11100 | Turn Right | Forward | Backward |
| 11110 | Turn Right | Forward | Backward |
| 00001 | Turn Left | Backward | Forward |
| 00011 | Turn Left | Backward | Forward |
| 00111 | Turn Left | Backward | Forward |
| 01111 | Turn Left | Backward | Forward |

## IV. SIMULATION RESULTS

To study the effectiveness of the proposed line follower algorithm and the robustness of the designed PID controller, the simulation package of the Matlab is used as shown in Figure 7. The PSO algorithm is used to tune the parameters of PID controller. The complex path (has sharp bends) is used to test the simulated mobile robot. The result show that the proposed line follower algorithm and the designed PID controller succeed to keep the robot track the desired trajectory as shown in Fig. 8. The error signal, the control signal, the Angular velocity of left wheel and angular velocity of the right wheel are shown in Figure 9, 10, 11 and 12, respectively. The hardware implementation is performed in this work. The AMR is connected with the host computer using USB cable to program the Motor Board of AMR. Simulink Support Package for Arduino Hardware is installed to Matlab program for programming the AMR. The Matlab Simulink is used to design the multiplexer and the flow chart to read the IR sensors to generate suitable PWM signals for the motor driver depending on the proposed line follower algorithm. The same complex path that is used in simulation part is used to test the real mobile robot. The practical results show that the AMR succeeds to track the desired trajectory even a complex path is used. Figure 13 shows the practical results.
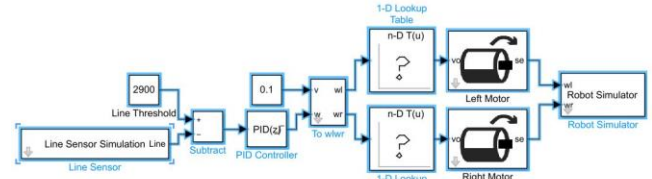


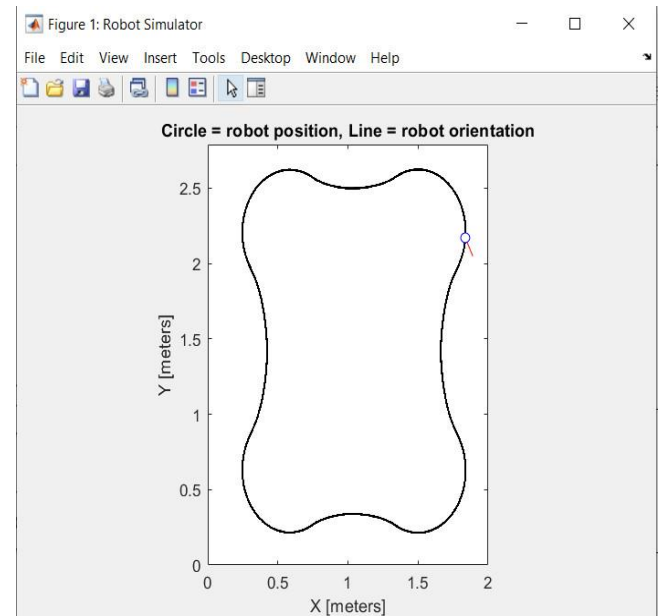Fig. (7) Simulation block diagram of the line follower mobile robot
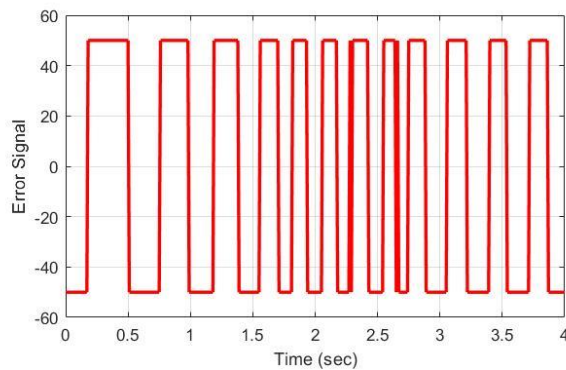


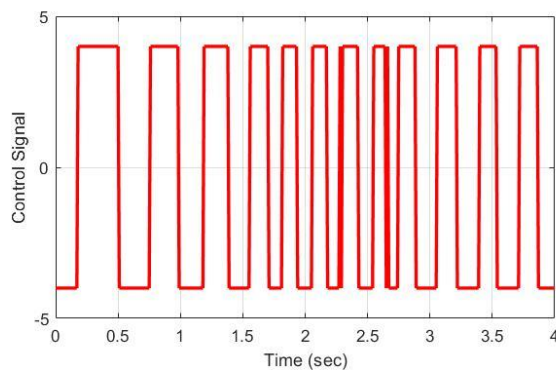Fig. (8) Robot simulator

Fig. (9) The Error Signal
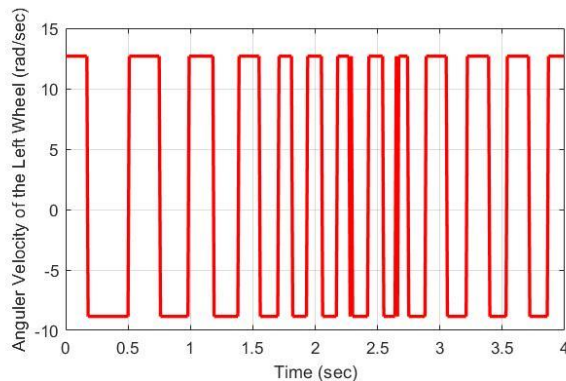


Fig. (10) Control Signal



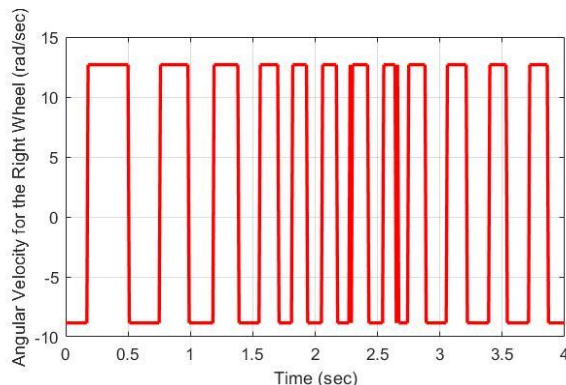Fig. (11) The Angular velocity for the left wheel



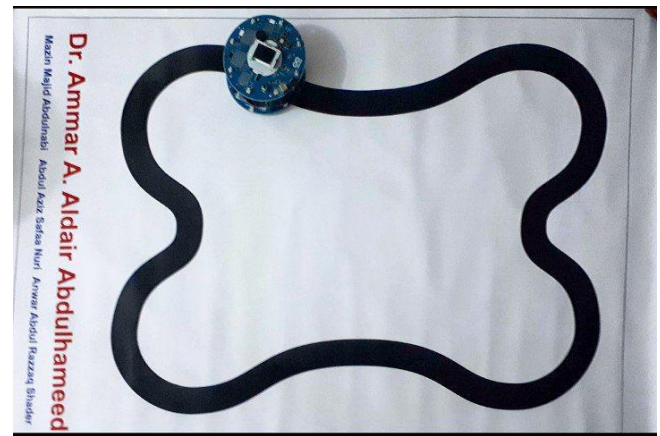Fig. (12) The Angular velocity for the right wheel



Fig. (13) Arduino Mobile Robot Tracks the desired trajectory

## V. CONCLUSION

The line follower algorithm is proposed for mobile robot to drive it on the complex path. The simulator is used to test the effectiveness of the proposed algorithm. The robust PID controller is design for omitting the oscillation of the mobile robot on the sharp bend. The PSO algorithm is used to select the optimal parameters of PID controller. The Arduino Mobile Robot is used for implementing the practical system. Simulink Support Package for Arduino Hardware is installed to Matlab program for programming the AMR. The practical and simulation results are prove that the propose line follower algorithm and designed PID controller succeed to drive the mobile robot on the complex path.

## CONFLICT OF INTEREST

The authors have no conflict of relevant interest to this article.

## REFERENCES

[1] O. Gumus, M. Topaloglu and D. Ozcelik, "The use of computer controlled line follower robots in public transport" 12th International Conference on Application of Fuzzy Systems and Soft Computing, Austria, pp. 202-208, 2016.

[2] M. B. Bhuiya, "Controlling line follower robot with the remote web server" Proceedings of SAI Intelligent Systems Conference, pp. 560-582, 2016.

[3] M. Engin and D. Engin, "Path planning of line follower robot" Proceedings of the 5th European DSP Education and Research Conference, pp. 1-5, 2012.

[4] K. M. Hasan, A. AI-Nahid, A. Al Mamun, "Implementation of autonomous line follower robot" IEEE/OSA/IAPR International Conference on Informatics, Electronics & Vision, pp. 865-868, 2012.

[5] J. Chaudhari, A. Desai and S. Gavarskar, "Line Following Robot Using Arduino for Hospitals" 2nd International Conference on Intelligent Communication and Computational Techniques Manipal University Jaipur, pp. 28-29, 2019.

[6] R. ÇOBAN, Ö. ERÇIN, "Multi-objective Bees Algorithm to Optimal Tuning of PID Controller", Cukurova University Journal of the Faculty of Engineering and Architecture, Vol. 27, No. 2 , pp.13-26, 2012.

[7] W. Liao, Y. Hu, H. Wang, "Optimization of PID control for DC motor based on artificial bee colony algorithm", IEEE International Conference on Advanced Mechatronic Systems, pp. 23-27, 2014.

[8] Y. Sonmez1, O. Ayyildiz, H. T. Kahraman, U. Guvenc, S. Duman, "Improvement of Buck Converter Performance Using Artificial Bee Colony Optimized-PID Controller", Journal of Automation and Control Engineering, Vol. 3, No. 4, pp. 304-310, 2015.

[9] B. N. Abdul Samed, A. A. Aldair, and A. Al-Mayyahi, "Robust Trajectory Tracking Control and Obstacles Avoidance Algorithm for Quadrotor Unmanned Aerial Vehicle," J. Electr. Eng. Technol., vol. 15, no. 2, pp. 855–868, 2020.

[10] A. A. Aldair and A. Al-Mayyahi, "Maze Maneuvering and Colored Object Tracking for Differential Drive Mobile Robot," Iraqi J. Electr. Electron. Eng., vol. 15, no. 1, pp. 47–52, 2019.

[11] P. Heryati, A. Aghagani, "Science of Robot Design and Build Robot", Azarakhsh Publication, 2008.