

# Vision-Based Line Following Robot in Webots

Alfian Ma'arif  
 Department of Electrical Engineering  
 Universitas Ahmad Dahlan  
 Yogyakarta, Indonesia  
 alfianmaarif@ee.uad.ac.id

Aninditya Anggari Nuryono\*  
 Department of Electrical Engineering  
 Universitas Gadjah Mada  
 Yogyakarta, Indonesia  
 anindityanuryono@gmail.com  
 \*corresponding author

Iswanto  
 Department of Electrical Engineering  
 Universitas Muhammadiyah  
 Yogyakarta  
 Yogyakarta, Indonesia  
 iswanto\_te@umy.ac.id

**Abstract**— Line following robot is one of the popular robots commonly used for educational purposes. The most widely used sensors for the robots are photoelectric sensors. However, it is irrelevant, along with the development of autonomous vehicles and robotic vision. Robotic vision is a robot that can obtain information through image processing by the camera. The camera installed on the line following robot aims to detect image-based lines and to navigate the robot to follow the path. This paper proposed a method of image preprocessing along with its robot action for line-following robots. This includes image preprocessing such as dilation, erosion, Gaussian filtering, contour search, and centerline definition to detect path lines and to determine the proper robot action. The implementation of the robot is simulated using Webots simulator. OpenCV and Python are utilized to design line detection systems and robot movements. The simulation result shows that the method is implemented properly, and the robot can follow a different type of path lines such as zigzag, dotted, and curved line. The resolution of the cropped-image frame is the fundamental parameter in detecting path lines.

**Keywords**—Webots, line-following, robot, OpenCV, vision

## I. INTRODUCTION

Rapid advances in robotics make robots to be able to do some specific tasks such as monitoring [1], delivering goods [2], football [3][4][5], sorting fruit based on the type of fruit or the fruit color, or automatic watering plants [6]. Another kind of robot which is line following robot is one of the popular robots commonly used for educational purposes [7]. Line-following robots are popular to be used as an introduction for students on how to program the robot in a computer moving in a specific direction or with controlled speed.

Line following robots are robots that can follow or track particular lines autonomously. The most commonly used sensors for the robots are photoelectric sensors such as photodiode [8] or light-dependent resistors (LDR) [9].

However, photoelectric sensors are no longer relevant to be used for advanced line following robots along with the development of autonomous vehicles. Photoelectric sensors have some tricky limitations. The basic design of the photoelectric sensor for detecting lines is profoundly affected by external or additional lights, hard to recognize varying colors and minimal distance range of senses.

Some modified-designs of photoelectric sensors for line following have been proposed. However, it is only suitable for specific conditions [10][11]. Along with the era of robotic vision, optical cameras are now considered as the alternative use for line following robots [12].

This paper proposed an image processing method and its implementation to the line following robot. The robot has a

camera as the main component to be able to move autonomously. Visual-based robot movements have become a research trend in the field of robotics. The visual representation of the environment that is similar to reality is an essential part of image-based control [13]. It is processed as feedback to move the robot from the starting point to the destination point. In the paper, the camera is utilized as a sensor to detect the path or line that will be traversed by the robot. The robot will follow the relative position of black and white processed-image.

Webots simulator is a simulator of a virtual environment that can be used as a robotic vision simulator [14][15]. This paper will also use the Webots simulator virtual environment with KephraIV as line following robot. OpenCV will be used for image processing of robotic cameras.

The control system of the camera robot has an important role in the application of image-based autonomous robot motion systems [16]. Image-based systems can provide detailed information related to the environment and navigation that obtained by image processing.

There were some researches related to the line following robots, such as one that is conducted by Li-Hong Juang et al. [17]. They aimed to get visual navigation for humanoid robots. Their proposed method was a square searching to extract lines from line images. The method not only improves the navigation performance of NAO robots in real-time but also increases the accuracy of detecting lines. The robot can walk along with different line shapes with low deviations using PID control.

Visual based line detection was also studied by Jianye et al. [18] using multiple line detection algorithm based on pixel histogram. Lines are the bird view to obtain bend pixels and ignore horizontal lines that intersect each other. Image-based robot control was also carried out by Chun et al. [19], who proposed the IP camera and MATLAB toolbox as image processing to detect black lines. The cascade intelligent control system is also used to stabilize the robot and to achieve the desired tracking performances.

The following line robots also had a task as assistant robots, such as one conducted by Abdul Kader et al. [20] to deliver particular objects. This robot uses the PID controller and infrared to detect paths [21][22]. Line following robots using artificial neural networks proposed by Roy et al. [23] based on infrared sensors. The robot can follow the curve path smoothly. Besides Arduino [24][25], line-following robots can also use the microcontroller ATmega32A [26][27] and ATmega328 Pro [28].

The first section of the paper is about the introduction. The second section is about modeling a mobile robot. The third section discusses the proposed method. The fourth section is

about results and discussion. The last section is conclusions and future work.

## II. PROPOSED METHOD

The research used Python 3.7 and OpenCV 4 to build the visual simulation of KheperaIV as a line following robot. The proposed method consists of three main steps, which are initialization, image processing, and robot action.

Initialization is done based on parameters of the robot and the environment, such as the camera selection and the generated path line. In other words, initialization is done to define each parameter used in the simulation, including the robot's and virtual environment parameters.

In image preprocessing, the image of the path line is captured by the robot's camera. The image is used as an input and then cropped to a specific width and length of the frame. After that, the image which is still in RGB color, is converted into a greyscale image.

The RGB image is converted into greyscale by using (1), where variable *Red* is the red value, variable *Green* is the green value, and the variable *Blue* is the blue value of each pixel. The result is shown in Figure 1.

$$Gray = 0.3 * Red + 0.59 * Green + 0.11 * Blue \quad (1)$$

The next step is to reduce the noise found in the image. Gaussian blur filtering is used for blurring in image preprocessing. It creates the autofocus effect by reducing details and adding the foggy effect.

The equation of Gaussian filtering can be written as (2), where variable  $\sigma$  is the standard deviation. The result of image preprocessing based on the applied 2-dimensional Gaussian filter equation is shown in Figure 2.

$$G = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (2)$$

Then, the grayscale image is converted into binary form. The binary threshold is used for segmentation so that the robot's path line can be obtained from a black and white image. The result is shown in Figure 3.

The next step is the dilation and erosion process. Dilation is the process of adding pixels to the edges of objects in the image, while erosion is the process of removing pixels. The equation of the erosion process is shown as (3).

$$A \ominus B = \{z \in E | B_z \subseteq A\} \quad (3)$$

Where  $B_z$  is  $B$  translation of the vector  $z$ ,  $E$  is Euclidean,  $A$  is a binary image.



Fig. 1. Grey image



Fig. 2. Gaussian blur image

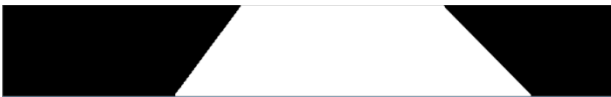


Fig. 3. Binary threshold result

Meanwhile, the binary image dilation equation can be written as (4), where  $A_b$  is the translation of  $A$  with respect to  $b$ ,  $A$  is a binary image, and  $B$  is an element of the Euclidean distance.

$$A \oplus B = \bigcup_{b \in B} A_b \quad (4)$$

The result of the dilation and erosion process is the result of image preprocessing. The data will get into image processing steps. After the dilation and erosion process, the contour values will be determined. In other words, this is the process to find and to detect the edge contour of any shapes found in the image. The contour values are used as a curve that connects all continuous points, which has the same color and light intensity.

These data will be processed in further image processing steps to obtain the parameter of the centerline point, which will determine the robot action or movement. The equation used to find the central or centroid line equation with  $375 \times 752$  resolution can be written as (5) and (6).

$$m_{ji} = \sum_{x,y} (array(x,y) \cdot (x - \bar{x})^j \cdot (y - \bar{y})^i) \quad (5)$$

$$\bar{x} = \frac{m_{01}}{m_{00}}, \bar{y} = \frac{m_{10}}{m_{00}} \quad (6)$$

Where  $\bar{x}$  and  $\bar{y}$  are center of  $x$ -axis and center of  $y$ -axis respectively,  $m_{ji}$  is the central momentum. The result of the process is shown in Figure 4.

The simulation in Webots displays KheperaIV as the robot, the black-colored line as the path to be followed by the robot, and the camera display as the robot vision of the environment. The robot vision and the line detection algorithm are shown in Figure 5 and Figure 6.

Based on Figure 6, robot action is the robot movement, such as move forward, turn left, turn right, and turn around as the response to detected lines based on the result of the image

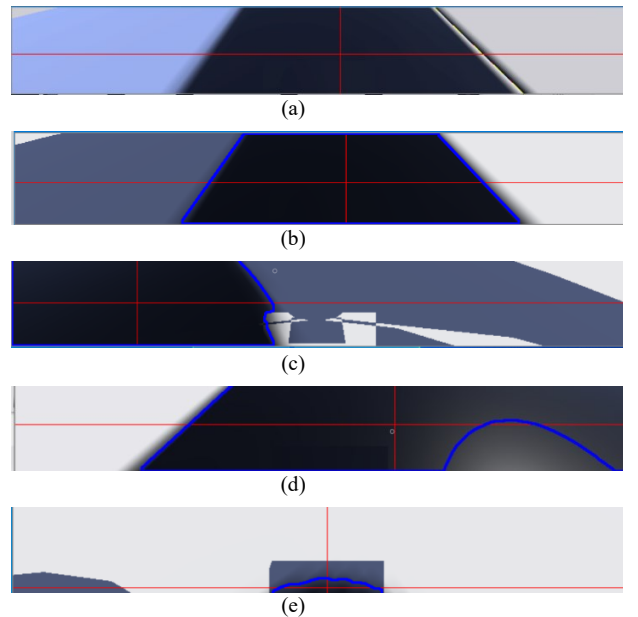


Fig. 4. Centerline detection with  $375 \times 752$  resolution: (a) center of  $x$  and  $y$  axis, (b) center of  $x$ -axis: move forward, (c) center of  $x$  axis: turn left (d) center of  $x$ -axis: turn right, (e) center of  $y$ -axis: turn around

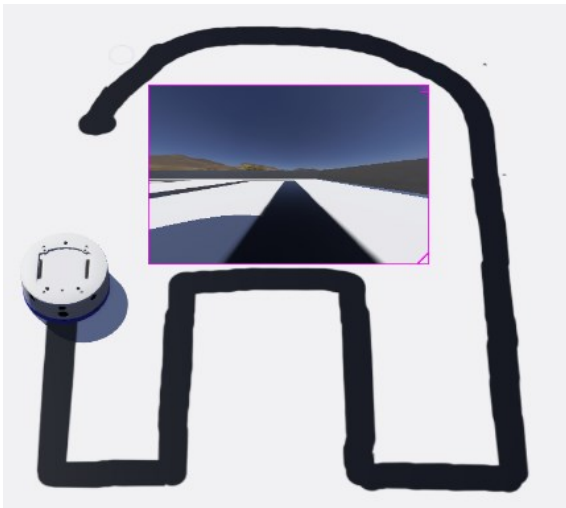


Fig. 5. Line path and KheperaIV robot vision

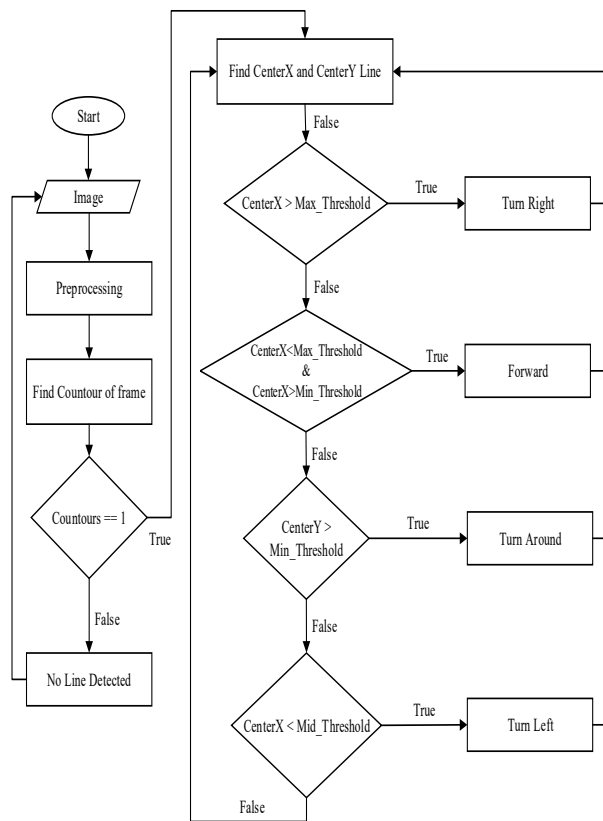


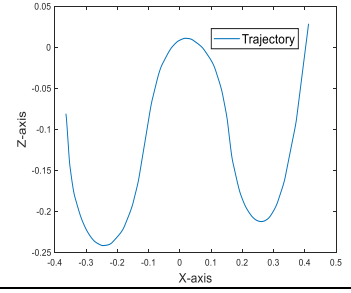

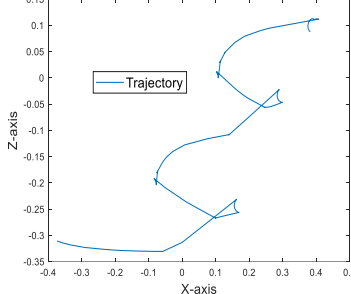

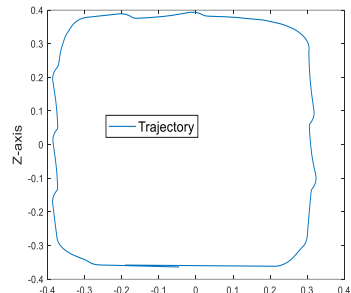
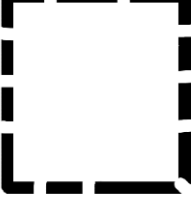
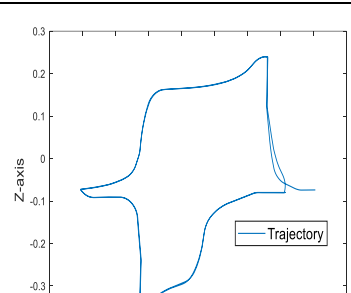
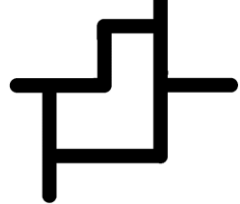
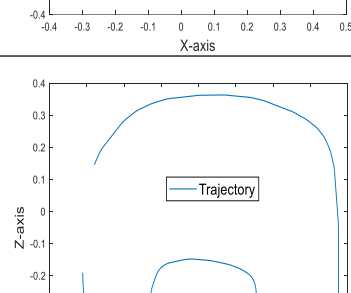
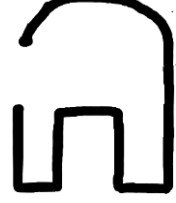
Fig. 6. Line following detection system diagram

processing. It depends on the detected line center value. When the center value on the  $x$ -axis of the line is greater than the maximum threshold, the robot will turn right. The robot will move forward when the  $x$ -axis center value of the line is between the range of maximum and the minimum threshold value. It will turn around when the robot meets a dead-end or does not detect any line, and the center value of the  $x$ -axis is greater than the minimum threshold value. It will turn left when the  $x$ -axis center value is smaller than the minimum threshold.

### III. RESULT AND DISCUSSION

The study has several scenarios for the evaluation of the proposed method. The evaluation is conducted by testing the applied proposed method with a different type of path lines.

TABLE I. ROBOT TRAJECTORY AND TRACK LINE

| Robot Trajectory   | Track Line   |
|--|--|
|    | <br>Curved        |
|    | <br>Zigzag        |
|   | <br>Dotted       |
|  | <br>Line path 1 |
|  | <br>Line path 2 |

The result of the evaluation is shown in Table I. The track lines are the path lines or the virtual environment, which the robot shall pass. Meanwhile, the robot trajectory is the path

of robot movement from one point to another by following a predetermined path. In other words, the trajectory is the result of the robot's action based on the applied proposed method of image processing of the track lines as a robot's environment. In Webots simulator, the  $x$ -axis and  $y$ -axis is used since the  $z$ -axis is not used because there is no coordinates' alteration of the robot's position in the Cartesian  $z$ -axis.

Based on the figures in Table I, the robot is able to move according to the line paths in general. The first track is the curve line. It is shown that the robot can follow the line with some arches quite different from the original curve track. The second track is to evaluate how the robot moves in the path with corners. The result shows that the robot does not move accurately while meeting corners or sharp turns.

Meanwhile, the third, fourth, and fifth track are to test the robot while following elbow tracks. The result shows that the movement does not form the elbow well. However, the robot still can follow the path.

Based on the simulation result, the robot can follow various track lines generally. However, the robot is not able to move along the path while using frame-cropping image resolution higher than  $475 \times 725$  dpi, as shown in Figure 7. This has happened because the camera detects lines in the outside area.

#### IV. CONCLUSION AND FUTURE WORK

The paper presented the visual of navigation system for KheperaIV robot as line following robot, which is simulated in the Webots simulator. KheperaIV Robot used a camera as the primary sensor for visualization. Several steps of image preprocessing, such as image cropping, binary conversion, grayscaleing, Gaussian blur filtering, erosion, and dilation, are carried out for the path detection process. The proposed method, which was applied to KheperaIV robot, is able to make the robot to move and to follow different path lines properly. Some types of path lines used for evaluation are zigzag, dotted lines, and a curved line. A robot is able to perform the appropriate robot action regarding the path line based on visualization from the camera. However, a path detection can be failed because of the higher resolution of the

cropped-image frame. Therefore, the frame resolution of the cropped-image becomes a significant parameter for detecting path lines.

Some possible future researches can be done based on the paper. In the paper, the robot only used a simple controller, which is the if-else rule. Hence, the speed of the robot cannot be controlled well. It creates possible future work in regards to speed control, which can be done by applying the Proportional Integral Derivative (PID) Controller. Moreover, the hardware implementation is needed to explore the behavior or characteristics of the proposed method. In a hardware implementation, the noise of the image from the camera becomes another problem in the visualization. It may come from the shock that happened when the robot is moving. Kalman Filter may be added to overcome this issue.

#### REFERENCES

- [1] W. Rahmiani and A. Wicaksono, "Design and Implementation of a Mobile Robot for Carbon Monoxide Monitoring," *Journal of Robotics and Control (JRC)*, vol. 2, no. 1, 2020.
- [2] L. K. Amifia, M. I. Riansyah, and P. D. Putra, "Design of Logistic Transporter Robot System," *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika*, vol. 6, no. 1, p. 19, 2020.
- [3] R. T. Yunardi, D. Arifianto, F. Bachtari, J. I. Prananingrum, and U. Airlangga, "Holonomic Implementation of Three Wheels Omnidirectional Mobile Robot using DC Motors," *Journal of Robotics and Control (JRC)*, vol. 2, no. 2, pp. 65–71, 2021.
- [4] N. S. Widodo and A. Pamungkas, "Machine Vision-based Obstacle Avoidance for Mobile Robot," *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika*, vol. 5, no. 2, p. 77, 2020.
- [5] N. Rinanto, I. Marzuqi, A. Khumaidi, and S. T. Sarena, "Obstacle Avoidance using Fuzzy Logic Controller on Wheeled Soccer Robot," *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika*, vol. 5, no. 1, pp. 26–35, 2019.
- [6] A. Hassan *et al.*, "A Wirelessly Controlled Robot-based Smart Irrigation System by Exploiting Arduino," *Journal of Robotics and Control (JRC)*, vol. 2, no. 1, pp. 29–34, 2020.
- [7] P. V. S. G. De Lima *et al.*, "Improving Early Robotics Education Using a Line-Following Robot Simulator," *Proceedings - 15th Latin American Robotics Symposium, 6th Brazilian Robotics Symposium and 9th Workshop on Robotics in Education*, pp. 554–561, 2018.
- [8] D. Nikolov, G. Zafirov, I. Stefanov, K. Nikov, and S. Stefanova, "Autonomous navigation and speed control for line following robot," *2018 IEEE 27th International Scientific Conference Electronics, ET 2018 - Proceedings*, pp. 1–4, 2018.
- [9] K. M. Hasan, Abdullah-Al-Nahid, and A. Al Mamun, "Implementation of autonomous line follower robot," in *2012 International Conference on Informatics, Electronics and Vision, ICIEV 2012*, 2012, pp. 865–869.
- [10] M. Shah, V. Rawal, and J. Dalwadi, "Design Implementation of High-Performance Line Following Robot," *International Conference on Transforming Engineering Education*, 2018.
- [11] G. Sonal, P. Raninga, and H. Patel, "Design and implementation of RGB color line following robot," in *Proceedings of the International Conference on Computing Methodologies and Communication, ICCMC 2017*, 2018, vol. 2018-January, pp. 442–446.
- [12] J. Sarwade, S. Shetty, A. Bhavsar, M. Mergu, and A. Talekar, "Line following robot using image processing," in *Proceedings of the 3rd International Conference on Computing Methodologies and Communication, ICCMC 2019*, 2019, pp. 1174–1179.
- [13] H. Omrane, M. S. Masmoudi, and M. Masmoudi, "Neural controller of autonomous driving mobile robot by an embedded camera," *2018 4th International Conference on Advanced Technologies for Signal and Image Processing, ATSIP 2018*, no. Figure 1, pp. 1–5, 2018.
- [14] O. Aviles, O. G. Rubiano, M. Mauledoux, A. Valencia, and R. Jiménez, "Simulation of a Mobile Manipulator on Webots," *Int. J. Online Eng.*, vol. 14, pp. 90–102, 2018.
- [15] A. C. Stan and M. Oprea, "A Case Study of Multi-Robot Systems Coordination using PSO simulated in Webots," *Proceedings of the 11th International Conference on Electronics, Computers and Artificial Intelligence, ECAI 2019*, 2019.
- [16] X. Du, K. K. Tan, and K. K. Htet, "Vision-based lane line detection

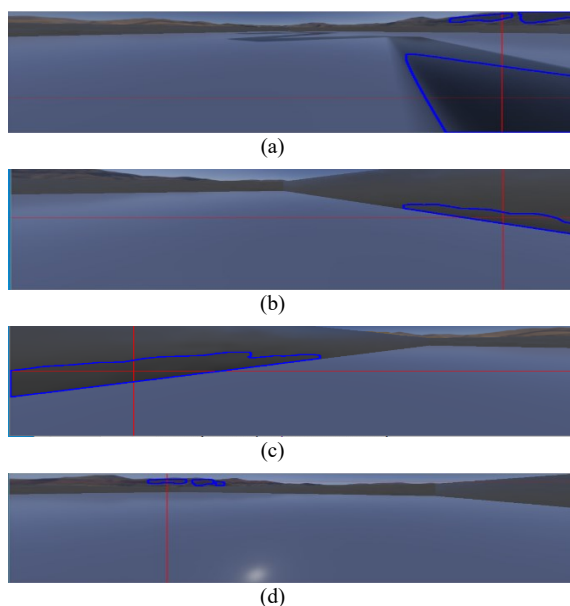


Fig. 7. Failure line detection with higher resolution cropping image with  $475 \times 725$  resolution

- for autonomous vehicle navigation and guidance,” *2015 10th Asian Control Conference: Emerging Control Techniques for a Sustainable World*, 2015.
- [17] L. H. Juang and J. Sen Zhang, “Robust visual line-following navigation system for humanoid robots,” *Artificial Intelligence Review*, vol. 53, no. 1, pp. 653–670, 2020.
- [18] J. He, S. Sun, D. Zhang, G. Wang, and C. Zhang, “Lane detection for track-following based on histogram statistics,” *2019 IEEE International Conference on Electron Devices and Solid-State Circuits*, pp. 1–2, 2019.
- [19] C. F. Hsu, C. T. Su, W. F. Kao, and B. K. Lee, “Vision-Based Line-Following Control of a Two-Wheel Self-Balancing Robot,” in *Proceedings - International Conference on Machine Learning and Cybernetics*, 2018, vol. 1, pp. 319–324.
- [20] M. A. Kader, M. Z. Islam, J. Al Rafi, M. R. Islam, and F. S. Hossain, “Line Following Autonomous Office Assistant Robot with PID Algorithm,” *2018 International Conference on Innovations in Science, Engineering and Technology*, no. October, pp. 109–114, 2018.
- [21] N. Hasanah, A. H. Alasiry, and B. Sumantri, “Two Wheels Line Following Balancing Robot Control using Fuzzy Logic and PID on Sloping Surface,” *2018 International Electronics Symposium on Engineering Technology and Applications*, pp. 210–215, 2019.
- [22] Z. U. Abideen, M. B. Anwar, and H. Tariq, “Dual purpose cartesian infrared sensor array based PID controlled line follower robot for medical applications,” *2018 International Conference on Electrical Engineering*, pp. 1–7, 2018.
- [23] A. Roy and M. M. Noel, “Design of a high-speed line following robot that smoothly follows tight curves,” *Computers and Electrical Engineering*, vol. 56, pp. 732–747, Nov. 2016.
- [24] M. S. Bin Mostafa, A. K. M. Masum, M. S. Uddin, M. K. A. Chy, and S. M. T. Reza, “Amphibious Line following Robot for Product Delivery in Context of Bangladesh,” *2nd International Conference on Electrical, Computer and Communication Engineering*, pp. 7–9, 2019.
- [25] R. I. Putra, S. Sunardi, and R. D. Puriyanto, “Monitoring Tegangan Baterai Lithium Polymer pada Robot Line Follower Secara Nirkabel,” *Buletin Ilmiah Sarjana Teknik Elektro*, vol. 1, no. 2, p. 73, 2019.
- [26] A. Maarif, S. Iskandar, and I. Iswanto, “New Design of Line Maze Solving Robot with Speed Controller and Short Path Finder Algorithm,” *International Review of Automatic Control (IREACO)*, vol. 12, no. 3, p. 154, 2019.
- [27] A. Latif, H. A. Widodo, R. Rahim, and K. Kunal, “Implementation of Line Follower Robot based Microcontroller ATmega32A,” *Journal of Robotics and Control (JRC)*, vol. 1, no. 2, pp. 70–74, 2020.
- [28] Y. Tian and G. Du, “Infrared Line Following and Ultrasonic Navigating Robot with ATMEGA328 Pro,” *Proceedings of 2019 IEEE 3rd Advanced Information Management, Communicates, Electronic and Automation Control Conference*, no. Imcecc, pp. 856–860, 2019.