Sweeper Arm Mechanism Design on the Line Follower Robot with Path Sensing Algorithm and Curvature-Driven Kinematic

Nattapat Charaspotiratanakul
Darunsikkhalai School for Innovative Learning
King Mongkut's University of Technology Thonburi
Bangkok, Thailand
n.charaspotiratanakul@gmail.com

Watchara Thitayanuwat
Darunsikkhalai School for Innovative Learning
King Mongkut's University of Technology Thonburi
Bangkok, Thailand
watchara.kmutnb@gmail.com

Ittichai Rattanathavorn
Darunsikkhalai School for Innovative Learning
King Mongkut's University of Technology Thonburi
Bangkok, Thailand
ittchai.rat@mail.kmutt.ac.th

***Abstract*— This paper presents the development of a sweeper arm mechanism implemented on the line follower robot. The sweeper arm mechanism enables the robot to move freely while following the path without interfering with the robot's movement. The path sensing algorithm allows the robot to track various path characters. The curvature-driven kinematic lets the robot optimize speed based on curve radius. The results show that the robot can follow the curve smoothly and reliably, proving the concept to be feasible for high-speed applications. Although the prototype didn't test at full speed due to physical limitations, it proves to be functional as intended.**

Keywords*— *line follower robot, curvature-driven kinematic, sweeper arm

I. Introduction

Line follower robot is one of the most common types of automated guided vehicles employed in industrial plants. Because of its cost-efficient and easy-to-understand application usage, this type of robot receives broad interest from various manufacturers. Line follower robots were suited with multiple tasks, ranging from carrying goods, floor cleaning, delivering service, and transportation. Since line follower robots only require a small amount of maintenance and don't require sophisticated engineer personnel to maintain their service, it is widely used as AGVs (automated guided vehicles) robots of choice for many users. The advancements in logistic management such as IoT (internet of thing) and 4.0 industries also drive the demand for this type of AGVs even further. This proves the importance of line follower robots and AGVs in the advancement of automated industries. [1]

Line following robot is a type of AGVs usually employed to carry objects around the industrial plant. This type of robot is operated by using sensors to track the predetermined path and follow the course to the desired destination. Usually, line follower robots were tasked with a relatively low-speed mission, thus doesn't require a sophisticated algorithm to archive the task. However, it is essential to develop a high-speed alternative version of the line follower robot, which will further increase the efficiency and reliability of this existing method.

This paper describes a way to create an alternative approach to line follower robots. By employing new techniques and algorithms, high-speed operation can archive. In this regard, the line follower robots were designed with sensors attached to the sweeper arm, which will follow the path separately from the robot. The robot also uses a potentiometer to measure the relative angle between the robot's sweeper arm and the robot itself and provide information to the curvature-driven kinematic algorithm. This paper also discusses a path sensing algorithm, which can sense the presence of the path and abort the system if necessary.

II. Robot Structure

The robot structure design was separated into three main parts as shown in Fig. 1. The first part was the sweeper arm, which is the main concept of this robot. The second part is the circuit board, which connected every electric component in this robot together. The third part is the differential drive, which drives the robot. The robot design was heavily influenced by the sweeper arm mechanism, which is essential to test the viability of the sweeper arm in a line follower robot. The robot also uses a PCB board as a structure which reduces weight and accommodates spaces for electric components.

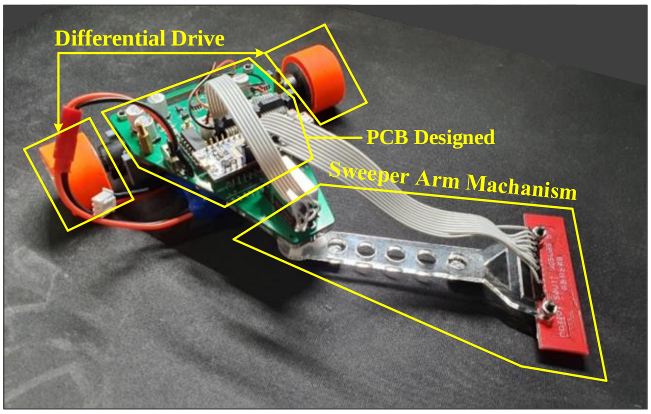

Fig. 1. Line follower robot with sweeper arm mechanism

A. Sweeper Arm Mechanism

The sweeper arm was designed based on the concept that the robot needs to know the radius of the path to archive higher speed. The sweeper arm was attached to the modified RC servo motor, enabling it to move freely while following the path without interfering with the robot's movement. For example, the robot can read the path on the left side while keep moving straight. The modified servo motor also has a potentiometer to measure the angle between the sweeper arm and the robot centerline, providing essential information for curvature-driven kinematic calculation. The movement of the sweeper arm is controlled by an array of IR sensors, which is attached to the other end of the sweeper arm. The IR sensor reads path position and uses that data to determine the servo motor's speed. The sweeper arm was made from laser-cut acrylic. It also has sliding pads attached to avoid scratching from the ground while moving around.

B. Printed Circuit Board (PCB) Design

The circuit board was designed based on the requirement as shown in Fig. 2. The robot needs to include a modified RC servo motor, two brushed DC motors, two L293D motor drivers, two switches, and two indicators LED. The robot uses QTR-8RC analog IR sensor array from Pololu, which is connected to the main processing unit via a circuit board. The microcontroller unit is Nucleo-L432KC which has 7 analog pins and the same footprint as Arduino nano. The design also included low battery warning indicators, which will light up when battery voltage drops below 6.8 volts (from operation battery voltage at 7.4 volts). The PCB schematic was designed on Kicad as shown in Fig. 3.

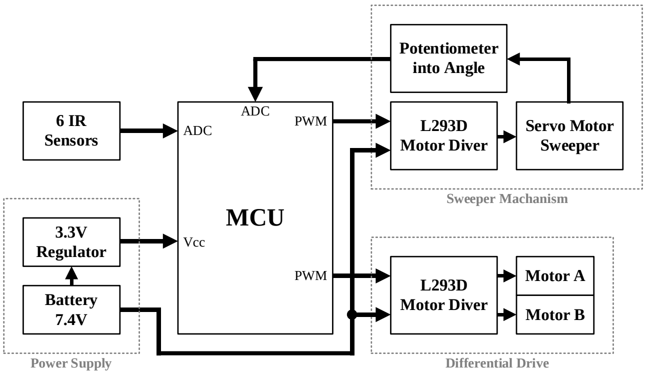

Fig. 2. Block diagram of overall hardware

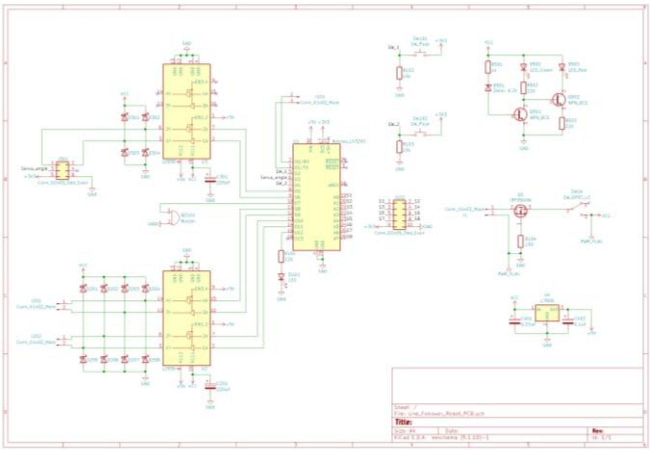

Fig. 3. Circuit schematic

C. Differential Drive

This line follower robot uses a simple differential drive configuration which consists of two motors driving the wheel on the same axis at the back of the robot. The robot uses DC brushed 7.4 volts which have planetary gear transmission and can archive a top rotation speed at around 1500 rpm (no-load). Connected to the driving shaft is an aluminum wheel with silicone tires, which create enough fiction to run the robot properly.

III. Robot Control Software

The robot control software was separated into five main parts, as shown in Fig. 4. The first part is the setup, in which the robot memorizes essential values from the sensor. The second part is the path sensing algorithm, which can determine the presence of the path. The third part is the path determination algorithm, which determines path position relative to the sweeper arm. The fourth part is sweeper arm position control system, which moves the sweeper arm according to the value from the previous part. The last part is curvature-driven kinematic, which calculates differential drive motor speed to move the robot.

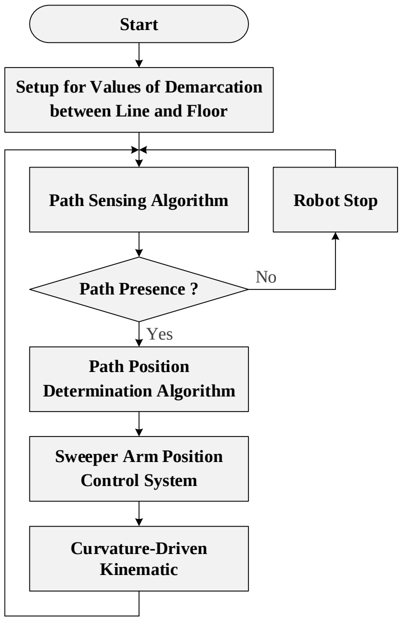

Fig. 4. Flowchart of robot control software

A. Setup for Line and Floor Values

First, during the setup process, the robots memorize IR sensor values measured from the line and the floor as shown in Fig. 5. Then, the robot calculates the means of those two values and stores them in the memory according to (1). These values will later be used to separate line and floor in the path position determination algorithm. Lastly, the robot communicates everything that it gains from the setup process back to the computer, which users can review for safety purposes.

$$Mean\{IR_1,\ \dots,\ IR_6\} = \frac{\text{Line}\{IR_1,\dots,IR_6\}+\text{Floor}\{IR_1,\dots,IR_6\}}{2} \quad (1)$$

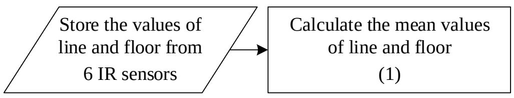

Fig. 5. Setup for values of Demarcation between line and floor color

In practicality, line and floor values are virtually the same, which will later represent as color#1 and color#2 instead. Since the robot can automatically differentiate between line and floor using its algorithm, these values are only used to determine the clarity level (expected S.D. value) between the line and the floor that the robot will experience. This is also an integral part of the line determination system, and the line position determination system.

B. Path Sensing Algorithm

Using the path sensing algorithm as shown in Fig. 6, the robot can sense the line underneath and its presence. This is done by calculating the S.D. value as in (2), a quantity calculated to indicate the extent of deviation of the data group, from the sensor and compared to the expected S.D. value as in (3) stored during the setup phase. If the S.D. value falls below the expected S.D. value, the robot assumes that all sensors detected the same value, which indicates that the path is not present, therefore, aborting the tasks. In practice, this system stops the robot immediately after losing contact with the path and effectively prevents any damage to the robot and its surroundings. The robot also uses an exponential weight moving average (EWMA) filter to minimize data fluctuation [2] as in (4).

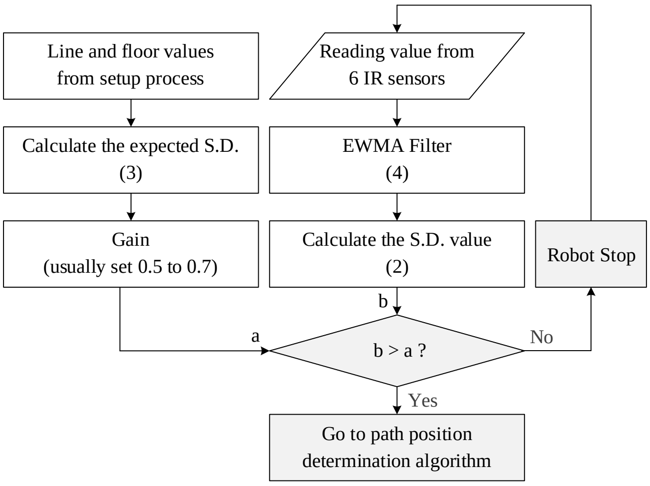

Fig. 6. Flow diagram of path sensing algorithm

$$S.D. = \sqrt{\frac{\sum_{n=1}^{6}|S_n+\bar{s}|^2}{6}} \quad (2)$$

$$Expected\ S.D. = \frac{\sum_{n=1}^{6}|Color\#1_n - Color\#2_n|}{6} \quad (3)$$

$$EWMA(n) = (1-\alpha)*x(t) + (1-\alpha)*EWMA(n-1) \quad (4)$$

α is the degree weighting decrease constant. n is a sensor position ranging from 1 to 6. S_n is a value read from sensor n. $\bar{s}$ is an average value across all sensors.

The expected S.D. value was calculated by averaging the difference in value between color#1 (floor) and color#2 (line) on six sensors during the setup phase. This gives the exact S.D. value expected to be calculated from the sensors during the regular operation of the robot (The line is presence). It is also to be noted that the difference between line and floor values also dictates the accuracy and preciseness of the operation, with more different equate more accuracy.

C. Path Position Determination Algorithm

The path position determination algorithm is used to determine the exact position of the path relative to the center position of the IR sensor array as shown in Fig. 7. The process starts by identifying whether color#1 or color#2 are more abundant throughout the array of sensors. For example, if the sensor array sees color#1 more than color#2, color#1 will be identified as the floor according to (5), and color#2 will be identified as the line according to (6). Then, the robot gives a binary result for each sensor.

After that, the robot calculates the path position (PV) by summing the position number (1 to 6) at which the sensors detected the line and divides those numbers by the numbers of sensors identified as seeing lines according to (7). For example, if sensors 3 and 4 detect the line, the calculated position will be equal to 3.5 Because that is the addition of 3 plus 4 and then divide by 2. The result of this operation is the exact path position relative to the array of sensors. Furthermore, the path position value becomes independent from the path width, thus ensuring an accurate reading over the course. The path position will later be used to command the sweeper arm movement. In practice, this enables the robot to track the path that is shown in Fig. 8.

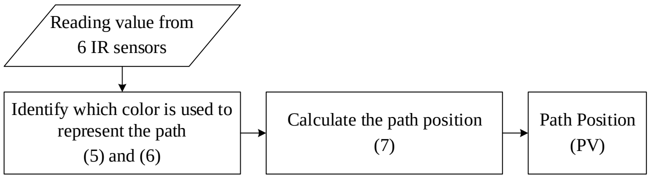

Fig. 7. Flow diagram of path position determination algorithm

Color selection:

$$\text{Floor is } f(V_n) = 0 \quad (5)$$

$$\text{Path is } f(V_n) = 1 \quad (6)$$

Path position calculation:

$$Path\ Position = \frac{\sum_{n=1}^{6} f(V_n)*n}{\sum f(V_n)} \quad (7)$$

$f(V_n)$ represents the function that differentiate between line and floor on sensor n. The value result from this function were shown in (5) and (6). While n is the IR sensor position ranging from 1 to 6 as shown in Fig. 8. Characteristic of the path that the robot can follow:

1) Color of the path and floor must have enough differences.
2) Path must be continuous, but color can switch or change to some extent.
3) Path width must cover 2 to 4 IR sensors at the same time.

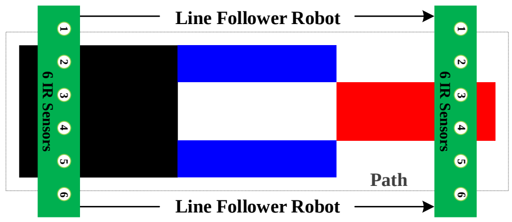

Fig. 8. Path chracteristics

D. Sweeper Arm Position Control System

Sweeper arm positions were controlled according to the path position relative to the sweeper with PD feedback loop control [3] as shown in Fig. 9. The purpose of this system is to increase the operation speed of the robot by increasing the continuality regarded to line reading. This system work by putting an IR sensor in front of the robot, located at the end of the sweeper arm attached to the modified RC servo motor, so they are virtually reading the line ahead, and the robot follows its afterword. This also eliminates the separation between the robot and its sensors during the curve path. Since the robot is not required to move its entire body to follow the line, line reading can be a lot smoother and finer than the conventional configuration. The system also has a potentiometer used to measure the angle between the sweeper arm and the robot itself, which is later used to calculate the angular velocity of the robot.

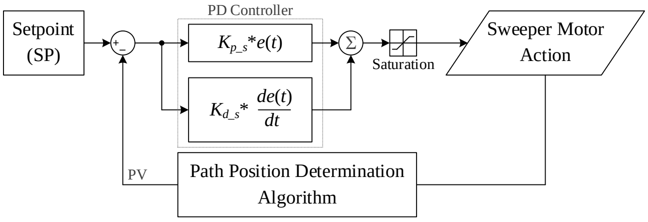

Fig. 9. Block diagram of sweeper arm position control system

In practice, this system constantly adjusts sweeper arm position to the setpoint (SP) which is equal to 3.5 or center position.

E. Curvature-Driven Kinematic

The relative angle between the centerline of the robot and the sweeper arm can be measured using a potentiometer attached to the modified servo motor. This angle can be used to calculate the current turning radius of the robot with the knowledge of trigonometry [4], which is the radius (r) of the curvature according to (8). The robot further calculates the appropriate velocity (v) according to (9) that should be set at a particular curvature. The robot velocity is calculated by multiply the constant (k_r) and the maximum operate velocity of the robot ($V_{\max}$). The constant (k_r) can be calculated by using standard logistic function according to (10). This, combined with the robot mechanics model as shown in Fig. 10, can later be calculated into the robot angular velocity according to (11) needed to use during the curve. The robot then used these data to calculate the operation speed of its two motors with differential drive inverse kinematic and set the motor speed according to (12) and (13) respectively [5]. The algorithm structure was shown in Fig. 11.

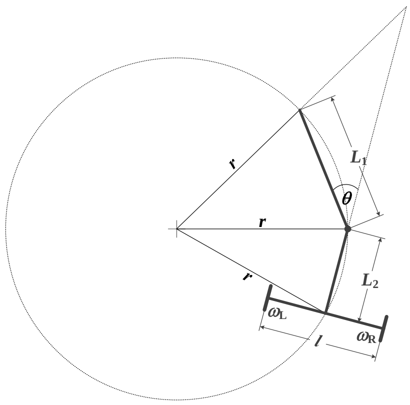

Fig. 10. Radius of path curvature drive using kinematic

$$r = \sqrt{\frac{L_1^2+L_2^2+L_1L_2\cos(\theta)}{2[1-\cos(2\theta)]}} \tag{8}$$

$$v = k_r V_{\max} \tag{9}$$

$$k_r = \left(\frac{1}{1+e^{-r}}\right) \tag{10}$$

$$\omega = \frac{v}{r} \tag{11}$$

$$\omega_L = \frac{1}{r_w}\left(v - \frac{\omega l}{2}\right) \tag{12}$$

$$\omega_R = \frac{1}{r_w}\left(v + \frac{\omega l}{2}\right) \tag{13}$$

ω_L is an angular velocity of the left wheel. ω_R is an angular velocity of the right wheel. r_w is a wheel radius, in this case equal 0.01625 m. l is a wheelbase length, in this case equal 0.13 m. L_1 is an arm length, in this case equal 0.11 m. And L_2 is a robot length, in this case equal 0.08 m.

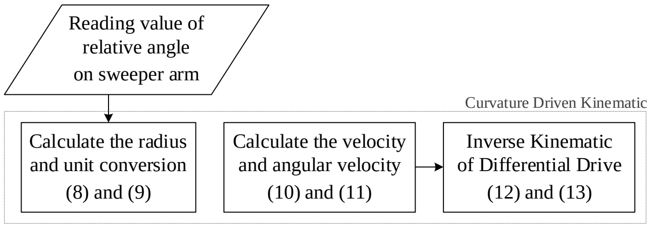

Fig. 11. Flow diagram of curvature driven kinematic algorithm

IV. Results

The experiment was separated into two parts. The first part tested the path characteristic test on the test bench. The second part is the path follower robot test, which tested all systems working in conjunction on the test track.

The path characteristic test have tested the path position determination algorithm and the sweeper arm position control system. The test is carried on a path characteristic test bench made from electrical tape stick to a white flat surface. The test bench has three different path styles and three colors, as shown in Fig. 12. The test is done by moving the robot with a hand and observing the sweeper arm response to the position change of path. To prove that the system work, the robot needs to continuously track path on the test bench in one setup. However, The robot velocity was set to zero and the robot movement was controlled by hand. Because it is not possible for the robot to move due to spaces limitation on the test bench. Only the sweeper arm and path position determination algorithm were tested in the first experiment.

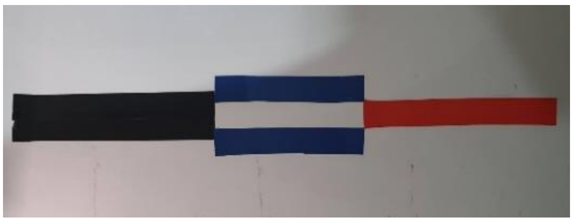

Fig. 12. Path characteristics test bench

The result shows that the robot can move the sweeper arm toward the center of the path in all three different styles and three different colors continuously in one setup. However, the sweeper arm oscillates a little more when tested on the wider path and jiggle during the transition of path characteristics (both colors and styles).

The path follower robot test tested all five parts of robot control software mentioned earlier in this paper working in conjunction. The test is carried on a track constructed from electrical tape on a strip wooden floor as shown in Fig. 13. The test track includes six different curves and six straight lines with an overall length of 7.2 m as shown in Fig. 14. The test is done by running the setup for line and floor values, and placing the robot on the track. Then, run the robot and measure the time it takes to complete the lap.

The robot's maximum operation speed was set to 18% duty cycle (0.9 m/s) during the test. The robot takes an average lap time of 17.6 sec, which is equal to an average speed of 0.41 m/s. The robot can follow the curve smoothly and reliably.

Fig. 13. Path follower robot on the test track

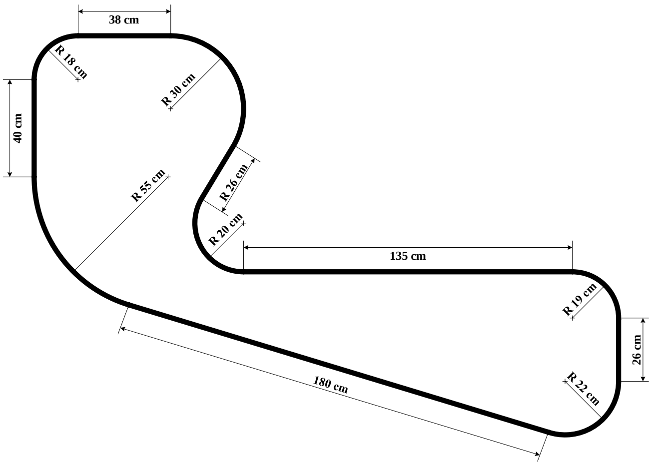

Fig. 14. Test track dimension

V. CONCLUSION

The result from the path characteristic test shows that the robot can operate in different path characteristics. The robot can automatically differentiate between path and floor with its algorithm. However, there is oscillating behavior during the transition between different path characteristics. This is because the EWMA filter can't handle the sudden change in color that occurs during the transition. Thus, create a little spike in the IR value stored in the filter.

Although the path follower robot experiment is done at a relatively low speed (avg. speed of 0.41 m/s), it proved the concept of a sweeper arm line follower robot. This is because the robot shows that it can track the line smoothly with its arm following the curve as shown in Fig. 13. However, the robot can't archive higher speed because of the physical limitation of the robot. The brush DC motor can't provide sufficient torque to brake or suddenly change the robot's speed. This can be problematic when using curvature-driven kinematic, which didn't account for the lag between input and output of the system. Further improvement can be done by introducing step response control with an overshoot characteristic on an open-loop control DC motor. This should compensate for the lag between input and output of the system. The step response control can reduce the rise time of the system, which comes from the physical feature of the robot.

In conclusion, the results show that sweeper arm mechanism design on the line follower robot with path sensing algorithm and curvature-driven kinematic concept is viable for high-speed application. Although the prototype didn't test at full speed due to physical limitations, it proves to be functional as intended.

REFERENCES

[1] Lorenzo Sabattini, Valerio Digani, Cristian Secchi, Giuseppina Cotena Davide Ronzoni, Matteo Foppoli and Fabio Oleari, "Technological roadmap to boost the introduction of AVG in industrial application," IEEE 9th International Conference on Intelligent Computer Communication and Processing (ICCP), pp. 203–208, September 2013.

[2] Marcus B. Perry, "The Exponentially Weighted Moving Average," Wiley Encyclopedia of Operations Research and Management Science, June 2010.

[3] Sourav Sutradhar, Viswanatha V, Siddhant kumar, Shivam Kumar, "Interlligent Line Follower Robot using MSP 430G2ET for Industrial Applications," HELIX: Peer Reviewed Bimonthly International Journal, vol. 10, pp. 232-237, April 2020.

[4] Jitendra Singh and Prashant Singh Chouhan, "A new approach for line following robot using radius of path curvature and differential drive kinematics," IEEE 6th International Conference on Computer Applications In Electrical Engineering-Recent Advances (CERA), October 2017.

[5] Farshad Arvin, Khairulmizam Samsudin, and M.Ali Nasserir, "Design of a Differential-Drive Wheeled Robot Controller with Puls-Width modulation," Conference on Innovation Technologies in Intelligent Syswtems and Industirial Application (CITISIA), pp 143-147, August 2009.