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

***Abstract*—This paper presents the development of a sweeper arm mechanism implemented on the line follower robot. The development is intended to improve the curvature-driven kinematic method currently use on a line follower robot by eliminating the fixed sensor array and replaced with a sweeper arm. The sweeper arm mechanism enables the robot to move freely while following the path without interfering with the robot's movement. The potentiometer inside the sweeper arm enables the robot to read the path more accurately with higher resolution than a conventional fixed IR sensor array. The path sensing algorithm allows the robot to track various path characteristics. The curvature-driven kinematic lets the robot optimize speed based on curve radius. To measure the result, the test track and test bench were created. On one hand, the test bench was used to test the functionality of the sweeper arm. On the other hand, the test track ensures that everything on the robot can work together in a real-world track. The results show that the robot can follow the curve smoothly and is reliable, demonstrating that the approach is more practical for high-speed applications. Although the prototype did not 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

A line follower robot is categorized as a type of automated guided vehicle (AGVs) that can detect and follow the path drawn on the floor. Generally, the path is predetermined and can be visible, like a black line on a white floor, or invisible, like a magnetic field path. Usually, line follower robots are employed in logistic operations; for example, line follower robots transport goods around an industrial plant [1]. In addition, line follower robots require less maintenance and do not require highly skilled engineers to provide maintenance services. Furthermore, technological improvements in logistic management, such as IoT (internet of things) and 4.0 industries, are driving up the demand for AGVs. This demonstrates the significance of line follower robots and AGVs in the progress of the automated industries sector [2].

The curvature-driven kinematic is a type of line follower robot control system that determines path curvature radius and calculates the velocity and angular velocity necessary to follow the path [3]. However, there is some limitation to a fixed sensor configuration with curvature-driven kinematic: the curvature-driven kinematic resolution is inversely proportional to the space between each sensor.

This research aims to develop line follower robots that use a sweeper arm mechanism and curvature-driven kinematic. The sweeper arm eliminates the problem associated with fixed sensors configuration by introducing continuous motion tracking and determining the radius with more resolution. Furthermore, the path sensing algorithm was developed to accommodate the use of the sweeper arm. This algorithm uses standard deviation to evaluate the presence of a path and can stop the robot immediately when it loses contact with the course.

II. Robot Structure

The structure design of the robot was divided 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 that connects all electronic components. 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 printed circuit board (PCB) as a structure; that reduces weight and provides space for electronic components.

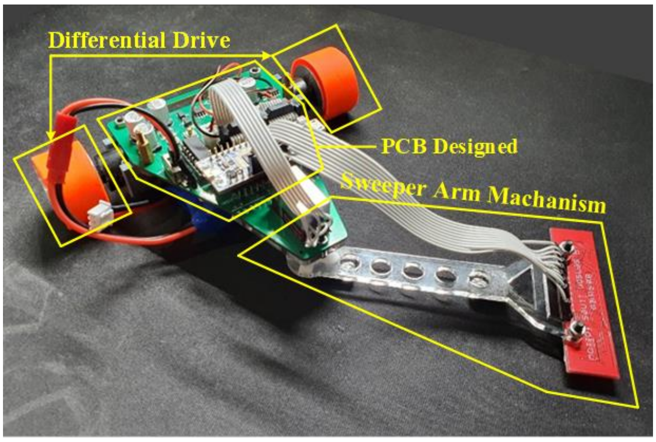

Fig. 1. Line follower robot with sweeper arm mechanism

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 achieve higher speed. The sweeper arm was attached to the modified radio-control servo motor (RC servo motor), enabling the sweeper arm 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 Infrared (IR) sensors, which is attached at the 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 has a range of +/- 90 degrees from the robot center-line. It was made from laser-cut acrylic and has sliding pads attached to avoid scratching from the ground while moving around. speed. The sweeper arm has a range of +/- 90 degrees from the robot center-line. It was made from laser-cut acrylic and has sliding pads attached to avoid scratching from the ground while moving around.

B. *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 a QTR-8RC analog IR sensor array from Pololu, which is connected to the microcontroller unit (MCU) via a circuit board. The MCU is Nucleo-L432KC which has seven analog pins and the same footprint as Arduino nano. The design also features 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.

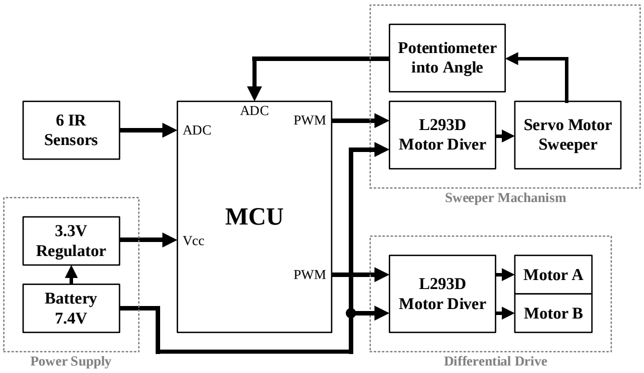

Fig. 2. Block diagram of overall hardware

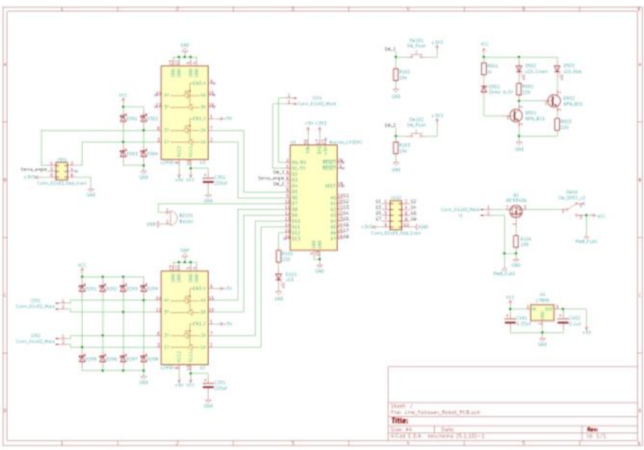

Fig. 3. Circuit schematic

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 achieve a top rotation speed at around 1500 rpm (no-load). Two aluminum wheels with silicone tires are attached to the driving shaft. The motors are controlled by the MCU and L293D motor driver.

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 that 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 based on the previous part's value. The last part is the curvature-driven kinematic, which calculates differential drive motor speed to move the robot.

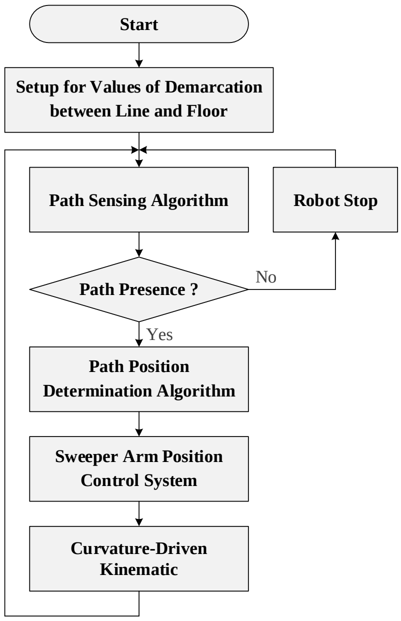

Fig. 4. Flowchart of robot control software

A. *Setup for Values of Demarcation between Line and Floor*

The block diagram of the setup process is shown in fig. 5. During the setup phase, the robot measure the IR value using its sensor from the line and the floor. 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.

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

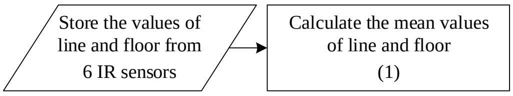

Fig. 5. Measure and calculate the mean values of line and floor.

In practice, the line and the floor values are virtually the same. The line floor values will later be represented as color#1 and color#2. Since the robot can automatically differentiate between line and floor using its algorithm, these values are only used to determine the clarity level between the line and the floor that the robot will experience. The clarity level between the line and the floor is represented by the expected standard deviation value (Expected S.D. value). 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 [4] as in (4).

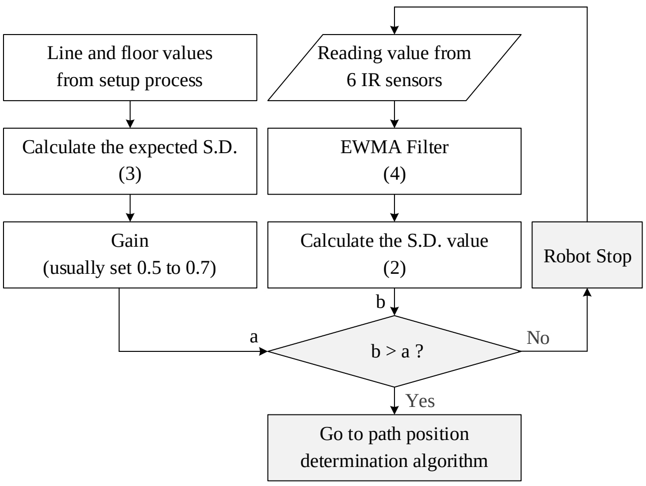

Fig. 6. Flow diagram of path sensing algorithm

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

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

$$EWMA(n) = (1-\alpha)*x(t) + (1-\alpha)*EWMA(n-1) \tag{4}$$

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 present). 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, color#1 will be identified as the floor according to (5), and color#2 will be identified as the line according to (6) when the sensor array observes more color#1 than color#2. The robot then gives each sensor a binary result.

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 divided 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.

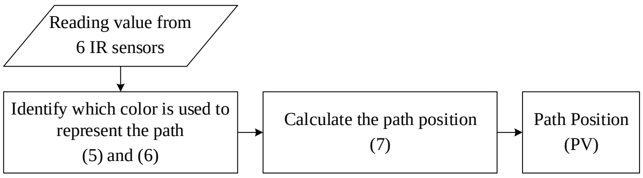

Fig. 7. Flow diagram of path position determination algorithm

Color selection:

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

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

Path position calculation:

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

$f(V_n)$ represents the function that differentiates between the line and floor on sensor n. The values resulting 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.

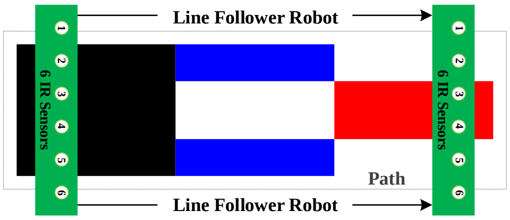

Fig. 8. Path chracteristics

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 placing an IR sensor in front of the robot; at the end of the sweeper arm, which is coupled to a modified RC servo motor, so that they are virtually reading the line ahead, and the robot follows it. 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.

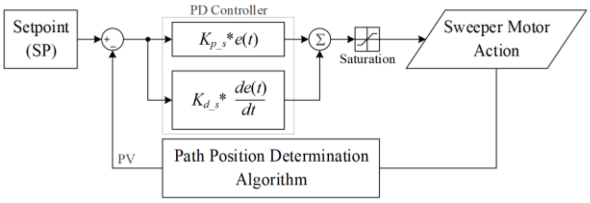

Fig. 9. Block diagram of sweeper arm position control system

In practice, this system constantly adjusts sweeper arm position to the setpoint (SP), which is equal to 3.5 or the 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. With the knowledge of trigonometry, this angle can be used to calculate the current turning radius of the robot, 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 multiplying the constant (k_r) and the maximum operateing 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 the data to calculate the operational speed of its two motors with differential drive inverse kinematic and set the motor speed according to (12) and (13) respectively. The algorithm structure is shown in Fig. 11. [5], [6]

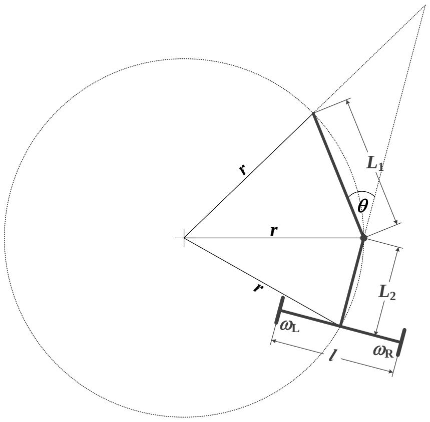

Fig. 10. Radius of path curvature drive using kinematic

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

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

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

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

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

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

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, which is equal to 0.01625 m. l is a wheelbase length, which is equal to 0.13 m. L_1 is an arm length, which is equal to 0.11 m. L_2 is a robot length, which is equal to 0.08 m.

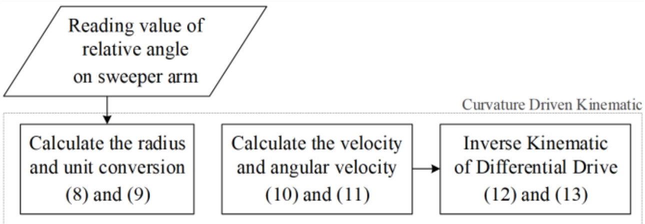

Fig. 11. Flow diagram of curvature driven kinematic algorithm

IV. Experiment Method and Result

The experiment method was designed to test the viability of the sweeper arm mechanism on the line follower robot. The result can be obtained by analyzing the robot movement during the test. The experiment is separated into two parts. The first part tested only the sweeper arm, and the second part tested the line follower robot with the sweeper arm on the test track.

A. Path Characteristic Test

The path characteristic test aimed to test the robot's path position determination algorithm and sweeper arm. The path was made from electrical tape stuck on a white flat surface. The test was carried out on the test bench, which has three different path styles and three colors, as shown in Fig. 12. The path length was 45 cm, that divided into three equal sections and each with different characteristics and colors. The test was separated into two parts. The first part tested the sweeper arm and path position determination algorithm separately. The second part tested the robot on the same test bench with all five robot control software mentioned earlier on the paper working in conjunction.

The first part of the test was done by pivoting the robot around the path. The robot should respond by moving the sweeper arm to correct the off-center position and remain in contact with the path. Repeat this action multiple times, and wait for the sweeper arm to stop moving. Then, proceed the second part tested which The robot had setup one time only for working.

The second part of the testing, the robot setup for the values of demarcation between line and floor was completed on the red section of the path and on the white floor. This is because the robot needs to know the minimum difference in IR values between line and floor during the operation, as stated in the setup process. The colors red and white were chosen because their IR readings on the test bench differ the least.

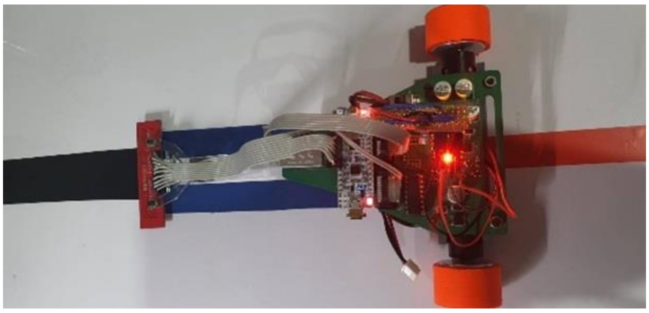

Fig. 12. Path characteristics test bench

The results from the first part of the path characteristic test show that the robot can move the sweeper arm toward the center of the path as the robot pivot around the path. Furthermore, the sweeper arm can track the path in all three styles and three colors in one setup only. The test was repeated multiple times, and the sweeper arm reliably tracked the path course every time. However, during testing on the wider path, the sweeper's arm oscillated a few times and jiggled while in transition between each section of the test bench (both colors and characteristics).

The second part of the test aimed to test the entire robot working on the test bench with three different path characteristics and colors. The purpose is to analyze the robot's response while experiencing various path characteristics. The robot was setup in the same configuration as the first experiment. However, the robot was programmed to test the interoperability of all five parts for robot control software.

The results of the second part, the path characteristic test, shows that the robot can successfully complete the test bench. With just one setup, the robot was able to move through all three different characteristics and colors. Similar to the first part experiment, the robot exhibited a few oscillations during the transition between each section.

B. Path Follower Robot Test

The path-tracking robot test was intended to test all five parts of robot control software mentioned earlier in this paper working in conjunction. The test was carried out on a track constructed from electrical tape on a strip wooden floor, as shown in Fig. 13. The track tested that includes six different curves and six straight lines with an overall length of 7.2 m, as shown in Fig. 14. The test was performed by running the setup and placing the robot on the track. Then, run the robot was measure for the time it took to complete a lap. The result can be observed by analyzing the robot response to curves. The sweeper arm should tangent to the path (Fig. 10) as it moves through the curves and the conclusion can be made by observing the robot on the test track.

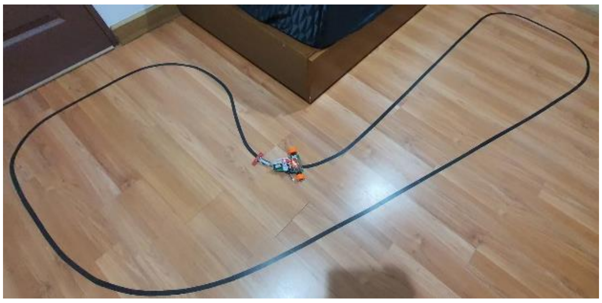

Fig. 13. Path follower robot on the test track

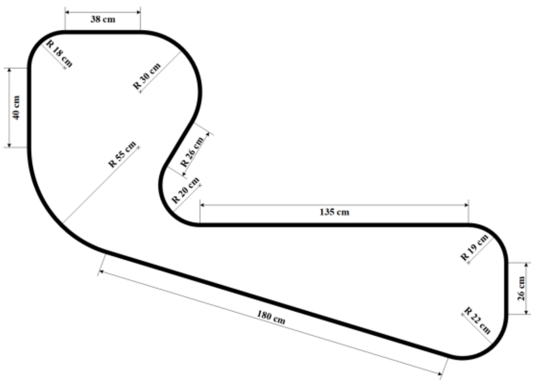

Fig. 14. Test track dimension

The path follower robot test showed that the robot was able to track the path successfully. The robot took 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 and the sweeper arm is tangent to the curve on most occasions.

V. Conclusion

The result from the path characteristic test showed 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 not handle the sudden change in color that occurs during the transition. Thus, creating a little spike in the IR value stored in the filter.

Although the path follower robot experiment was done at a relatively modset speed (average speed equal 0.41 m/s), it proved the concept of a sweeper arm line follower robot. This shows that the robot can track the line smoothly with its arm following the curve, as shown in Fig. 13. However, the robot could not achieve higher speed because of the physical limitation of the robot. The brush DC motor can not provide sufficient torque to brake or suddenly change the robot's speed. This can be problematic when using a curvature-driven kinematic, that dose not account for the lag between input and output of the system. Further improvement can be made by introducing a step response control with an overshoot characteristic on an open-loop control DC motor. Therefore, 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 features of the robot.

In conclusion, the results showed that sweeper arm mechanism design on the line follower robot with path sensing algorithm and curvature-driven kinematic concept was viable for high-speed application. Although the prototype did not test at full speed due to physical limitations, It was proved to functional as intended.

REFERENCES

[1] Suman Kumar Das, "Design and Methodology of Line Follower Automated Guided Vehicle-A Review," International Journal of Science Technology and Engineering (IJSTE), vol. 2, April 2016.

[2] 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.

[3] 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.

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

[5] 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.

[6] 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.