Kurdistan Region – Iraq
Ministry of Higher Education and Scientific Research
University of Salahaddin - Erbil

Review Article:

Line Following Robots

Review, Critiques, and Research Directions/Methods

Submitted to the Department of Electrical Engineering at Salahaddin University-Erbil

Prepared by:

Ahmed Khalid Ahmed

(Ph.D. Student)

Supervised by:

Prof. Dr. Ibrahim Hamarash

Erbil-Kurdistan

January 2021

Table of Contents:

Table of Figures:

1- Abstract

The Line following robots are one of the types of differential drive wheeled mobile robots category. It depends on detecting a white line path on black surfaces or vise versa. Many techniques and methodologies control Robot's type to maintain performance characteristics, robustness, and fair speed limits. Some of these methods include PIDs and Fuzzy logic control rules. Detecting the path problem is another discipline of research and investigation. Some methods to detect the path line are Imaging processing using camera sensors, reflective photodiodes, IR sensors, and optical encoders. The algorithm plays a significant role in solving the line detection. The least-square method and the Quadratic line detection algorithm are examples of algorithms used for line detection and path planning. In this review paper, the basic concepts, mechanism of work, methodologies, path detection techniques, and path planning methods will be presented.

Keywords: *Line following Robot, PID, fuzzy Logic, Arduino, path planning*

2- Introduction

The modern industrial plants are highly equipped with mechanical and automated parts, and even sometimes, they are fully automated. Using Robots is extremely useful in industrial and domestic applications. Because Robot is used widely in different areas of applications, it is crucial to understand the fundamentals and essential concepts of this mechatronic device. Robots generally conclude embedded systems consisting of a microcontroller, software, hardware, control theory, sensors, actuators, and interface or GUI. Designing embedded systems is fundamental to design robots generally and line follower mobile robot, especially. According to (Tayal *et al.*, 2020), line follower robots' primary goals are 1- Accurate speed control on the predefined Line (the constraint is the friction between the wheels and the ground surface also using servo motors ensures that the Robot accurately follows the Line). 2- Line sensing using red infrared sensors (IR).

Nowadays, line-following robot applications are extended to several areas to accomplish the tasks and interact with the human lifestyle. For example, Autonomous human following is one kind of application of mobile followers. The person may walk, and the follower will carry heavy bags. It can also follow a worker in a warehouse to pick a load from one area and autonomously transfer it to another spot. One example of robots capable of following and carrying heavy loads is the Boston Dynamic's LS3 four-legged Robot. In such kinds of applications, the Robot should track and follow the human smoothly and robustly while avoiding moving and static obstacles at the same time (Jin, Fang and Zeng, 2020).

Most of the Line following techniques include the predictive control method, proportional integral derivative PID, model adaptive, fuzzy logic control method, neural network, and fractional order control method (Eleftheriou *et al.*, 2020).

In this article review, the concepts and fundamental techniques used in Line following robots will be presented. Later, several methodologies and applications will be reviewed in the following next topics. At the end of each section, a specific conclusion was formed for the work reviewed in that section.

3-Design and Hardware Application of Line Following Robot

In (Tayal *et al.*, 2020), a cost-effective and straightforward circuit design of a black line follower robot and the implementation have been presented. The block diagram of the Line following Robot is shown below:

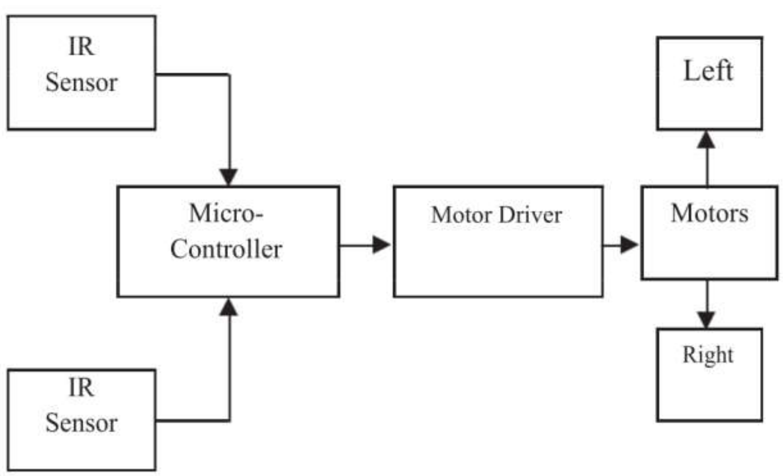

Figure 1: Block diagram of Line following robot structure.

As shown in the **Figure** above, the main parts are IR sensors, Microcontroller, motor driver, left and right motors. Other sensors used in line follower robots are photodiodes and Light dependent resistors (LDR). The IR sensor resistance is inversely proportional to the light reflected from the ground surface to the sensor. When the Robot is on a black line, and the surroundings are white, the amount of reflected light will increase, and the sensor's resistance will increase accordingly. The Robot, in this way, will be controlled and remain on the Line.

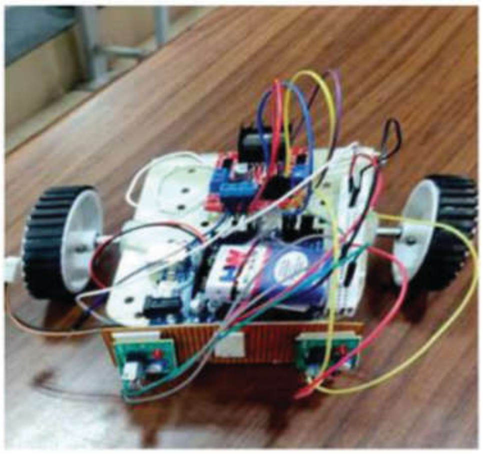

Figure 2: Proposed Design

Figure 3: Arduino Uno

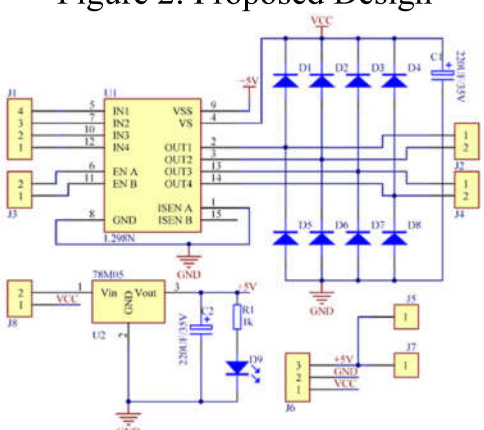

Figure 4: L298N motor driver

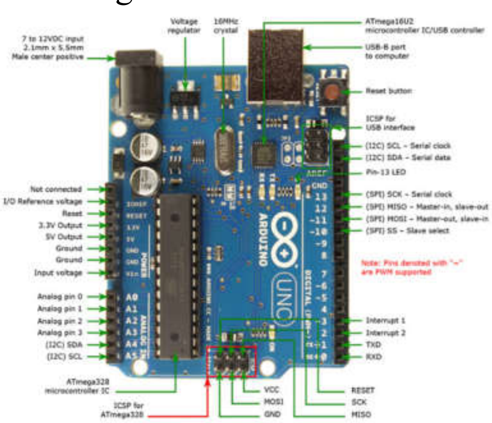

Figure 5: Arduino Pinout Diagram

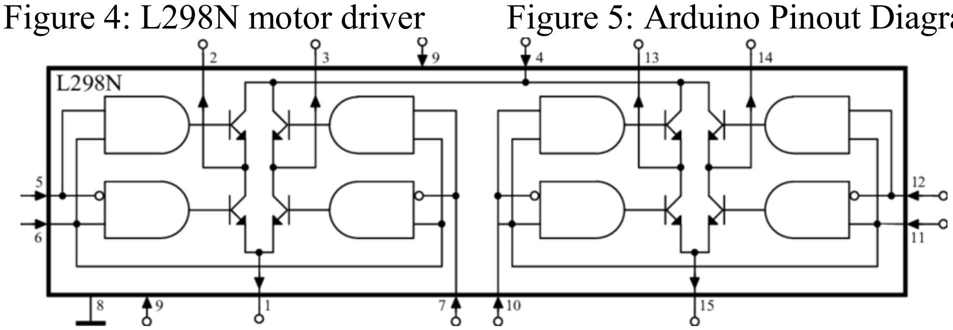

Figure 6: Block Diagram of L299N motor driver

The Microcontroller will process the sensor data, and then the calculated signals will be passed to the driver system.

As shown in **Figure** (3), Arduino Uno is used as the system's brain and can achieve the desired output for a given input signal. The Arduino MCU is programmable, and it can be easily connected to the designed circuit board. The MCU unit can be interfaced with several electrical components, and it can generate outputs according to the component; and based on programming, the pinout diagram is shown in **Figure** (4).

The right and left side motors can be controlled both in direction and speed using the motor driver circuit shown in **Figure** (4) and (6). It consists of dual H-bridge circuits coupled to the two motors for speed and direction control.

The MCU receives the signal from the IR sensors, and then it passes the signals to the motor driver to perform the Robot's required movement. The robot wheels are connected to the motor shaft, and they will help the Robot move quickly and smoothly. At the Robot's front, the caster wheel is used as a support for the whole mechanism. The steering capability of the Robot is the duty of the back left and right wheels.

The motor driver IC connected to the pair pins 1,2 and 3,4, and when pin 1 and 2 is high and low, the motor will rotate in the backward direction, and conversely, when pins 1 and 2 are low and high, the motor will rotate forward. Moreover, the motor rotation will stop when both inputs have the same signal state. Six types of motion for the Robot is declared in **Figure** (7).

S.N	**Robot Movement**	**Motor Rotation**	
		Left	***Right***
1	Left	STOP	ON (Forward)
2	Straight	ON (Forward)	ON (Forward)
3	Sharp Left	ON (Reverse)	ON (Forward)
4	Right	ON (Forward)	STOP
5	Sharp Right	ON (Forward)	ON (Reverse)
6	Reverse	ON (Reverse)	ON (Reverse)

Figure 7: Movement type of mobile robots.

Figure 8: Wheel of the Robot

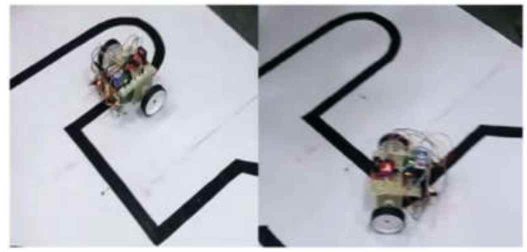

Figure 9: Line following robot movement.

The Line follower Robot in action and movement is shown in **Figure** (9)

4 -Line following Robot based on Fuzzy Logic Control, PID, and Image processing

4-1 Fuzzy Logic Control Background

There are two sorts of fuzzy logic inference systems (FIS) available: Mamdani and Sugeno. These techniques essentially contain the same steps; the only variance is that the Sugeno method outputs are usually linear. In terms of the fuzzy logic processes, there are three main processes, shown in **Figure** 10.

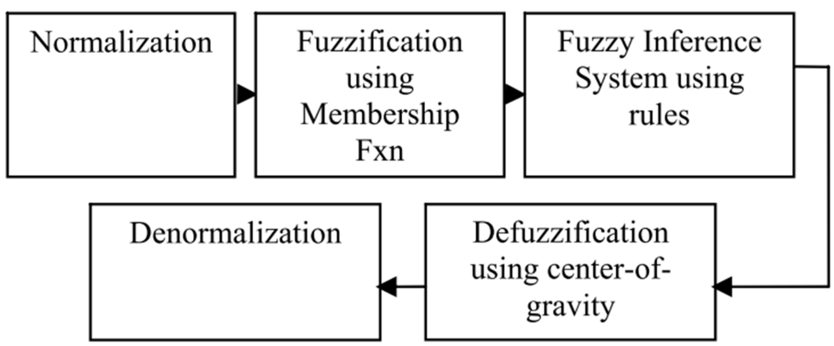

Figure 10: Main processes of fuzzy logic method

In the first primary fuzzification process, a crisp value is "fuzzified," which means that a fuzzy set characterizes it. It is followed by the inference process, whereby the fuzzified inputs described in the form of membership functions are linked and used to demonstrate the experts' knowledge in the form of rules. In the final process known as defuzzification, the output value is "defuzzified," allowing for significant crisp value.

The construction of a complete fuzzy control system consists of the subsequent main parts: -

1. Fuzzification,
2. Knowledge base.
3. Inference engine.
4. Defuzzification.

Figure 11 shows the internal configuration of a fuzzy logic controller.

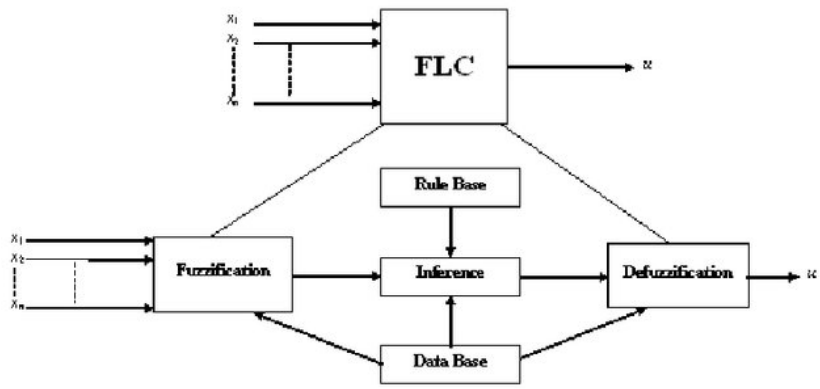

Figure 11: The Internal Configuration Of a Fuzzy Logic Controller

The fuzzification module transforms the crisp states of the control inputs into fuzzy states. A fuzzy variable has values represented by semantic variables (fuzzy sets or subsets) such as Low, Medium, high, prominent, slow, where a steadily altering membership function defines each one.

In fuzzy set vocabulary, all the potential values that a variable can assume are called the universe of discourse, and the fuzzy sets (described by membership functions) include the whole universe of discourse. The form of fuzzy sets can be triangular, trapezoidal.

A fuzzy control embeds a human operator's intuition and experience, and sometimes those of a designer and researcher. The database and the laws form the knowledge base used to get the deducing relation R. The database includes a representation of input and output variables using fuzzy sets. The rule base is typically the control policy of the system. It is usually acquired from expert knowledge or heuristics. It holds a collection of fuzzy conditional statements represented as a set of IF-THEN rules, such as:

$R^{(i)}$: If x_1 is F_1 and x_2 is F_2 … and x_n is F_n THEN Y is $G^{(i)}, i = 1, \dots, M$ (1)

Where: $(x_1, x_2, \dots, x_n)$ is the input variables vector, Y is the control variable, M is the number of rules, n is the number of fuzzy variables, (F1, F2… Fn) is the fuzzy sets.

The mathematical method of converting fuzzy values into brittle values is known as 'defuzzification.' The option of defuzzification methods depends on the application and the available processing power. A standard fuzzy classifier splits the signal x into five fuzzy levels: -

1. LP: x is large, positive.
2. X is medium positive.
3. S: x is small
4. MN: x is medium negative.
5. LN: x is a large negative.

(Bickraj *et al.*, 2006)(Ghani and Tahour, 2012)(Sharma, 2011).

Fuzzy Logic needs numerical parameters to function, such as a notable error and a critical rate-of-change-of-error error. Still, exact values of these numbers are usually not critical unless a very sympathetic performance is required, in which case practical tuning would determine them. A fuzzy expert system is a system that utilizes fuzzy Logic rather than Boolean Logic. They are used in various wide-ranging areas, including (Singh and Mishra, 2015):

1. Linear and nonlinear control.
2. Pattern recognition.
3. Financial systems.

4-2 Fuzzy Control Approaches

As mentioned by (Sharma, Disha), there are two main approaches for designing Fuzzy Logic Controllers, and they are:

1. A Model-Free approach
2. A Model-Based approach
3. Adaptive Fuzzy control

A model Free approach of fuzzy logic control operates for trajectory tracking for a standard, even complicated, dynamic system that does not have a specific mathematical model.

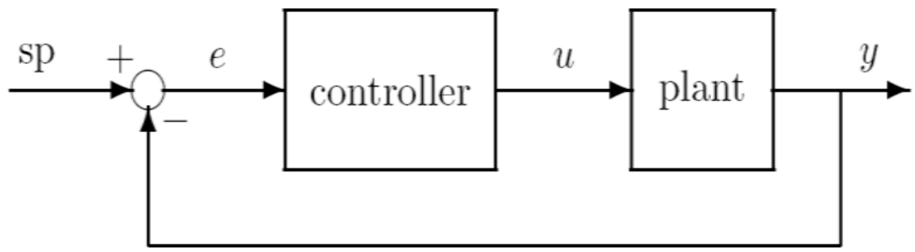

Figure 12: A typical setpoint tracking control system

The goal here is to establish a controller to achieve the goal *e(t)* → 0 as t → ∞, without any mathematical formula of the plant except that its inputs and outputs are measurable by sensors on Line. It is one of the advantages of the Fuzzy Logic approach in that it enables the design of a controller without knowing the system's mathematical description.

If a mathematical model of the system, or a reasonably good estimate of it, is available, one may design a fuzzy logic controller with more excellent results such as performance specifications and ensure stability; this constitutes a model-based fuzzy control approach.

When the parameters are immediately adjusted according to some adaptive law, reduce the plant's output and the reference model's variance.

Many aspects concerning Fuzzy Logic system modeling, structure identification, and parameters identifications cab be found in (Sharma, 2011).

Based on the essential background and approaches mentioned above, one can implement a Line follower robot using the fuzzy logic controller, and during the past decades, many researchers investigated designing and controlling line follower robots utilizing fuzzy control rules.

4-3 Line Following based on PID and Fuzzy Logic

(Eleftheriou *et al.*, 2020) worked on optimizing the line following and tracking control techniques used in IoT supported Line following Robot. This work proposed using computer vision, an array of IR sensors, and optimized PID based feedback. The PID controller has several advantages, including good control effect, simplicity in structure, easy implementation, and robustness.

Using a pure line following algorithm could lead to inconsistent results considering that the Robot will face several constraints during the task assigned to it. The constraints may include turning at the curved paths, endpoints, lighting effects, surface structure, noise, and bouncing wheels. The Fuzzy approach is an alternative to the approach mentioned above that provides several linguistic rules that describe the problem at hand very simply and fast.

One of the main advantages of a Fuzzy logic controller is that the mathematical model or the system dynamics is not necessarily known for most cases. Instead of this, some heuristic control knowledge is a necessary inference part of the control system.

This work will utilize PID in conjunction with the fuzzy logic and computer vision-based controller. The system's performance will increase because it relies on detecting the Robot's Line while moving. It calculates Robot's desired orientation before getting close to the constraints mentioned above by accelerating or decelerating the Robot's speed for better maneuverability.

The principle of work of this system can be explained as follows:

1. The IR sensors collect data concerning the Robot's position referred to the Line and send it to the MCU to be processed.
2. At the same time that IR sensors collect data, an onboard camera will take a picture of the area around the Robot, and the Microcontroller on the Camera calculates the Line's length.
3. The outputs of both IR and camera systems will be the input to the fuzzy controller.
4. The fuzzy controller generates the required output signal concerning the Robot's current position and next behavior.
5. Based on the fuzzy controller's output, optimal PID gains, and proper speed will be calculated.

For this work, a custom robot was developed, and the design is lightweight, low cost, and low computationally demanding. It can move freely by collecting data from several sensors and processing it onboard. The new system blocks are shown in **Figure** below:

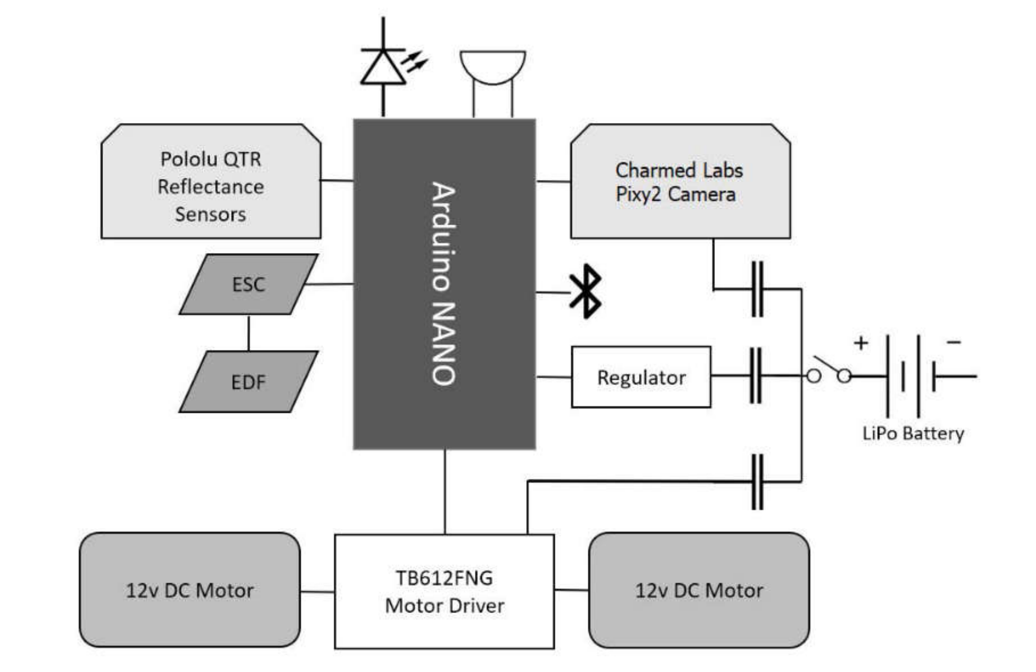

Figure 13: The proposed System Block diagram.

For this design, 8 IR sensors were used with a Pixy2 Camera from charmed Labs. Pixy2 Camera is robust, lightweight, and it can be easily interfaced with Arduino using standard communication ports such as USB. A new PCB was designed and fabricated to hold all the components used in this work.

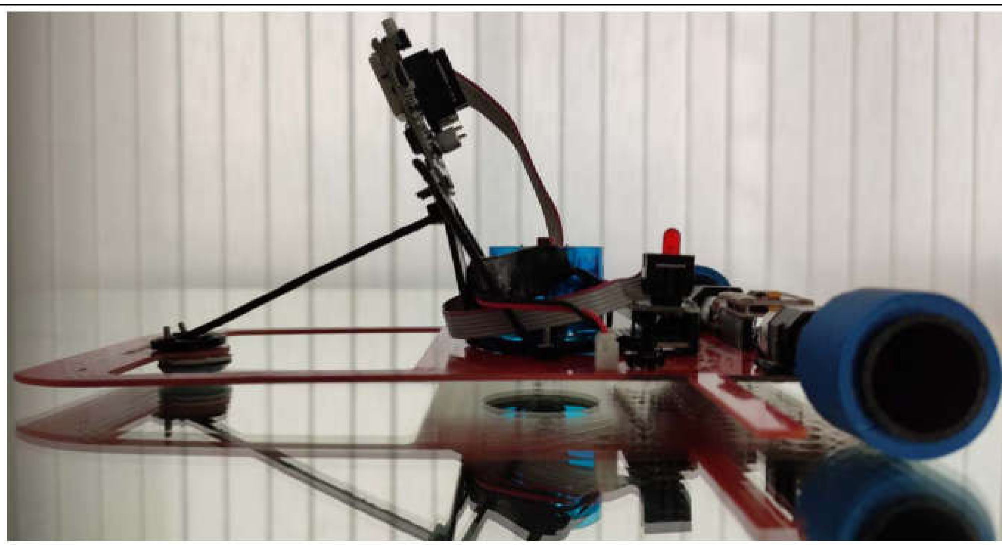
Figure 14: The side view of the proposed robot design.

The PID controller is used for the proposed robot algorithm and can be used successfully for low speeds and easy navigating paths.

The PID control can be expressed in this way:

$$u(t) = K_{\mathrm{p}} e(t) + K_{\mathrm{i}} \int_0^t e(t')\, dt' + K_{\mathrm{d}} \frac{de(t)}{dt} \qquad (2)$$

K_d, K_p, $K_{i,}$ and K_i are all non-negative quantities. U(t) is the input, and e(t) is the error. PID can be applied robustly to derive the Robot at a speed of one meter per second. However, when the speed required is more, and the path requires complicated maneuvers, PID only controller is not sufficient to accomplish the task. PID controllers suffer from this downside, and in the case of line follower robots, the controller can be used only for optimized and wide-angle paths.

This issue can be tested by turning the PID controller and use it in the line follower robot on the right angle path. One can easily decide that the Robot fails to turn and follow the path.

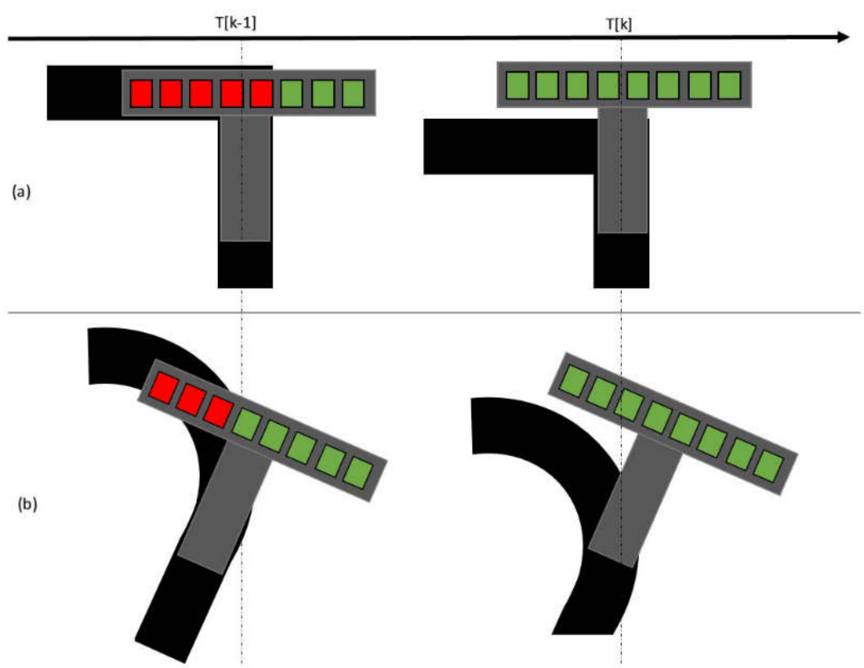

Figure 15: Line loss cases.

From the **Figure** shown above, when the path curve changes dramatically, the PID controller will lose the path due to its dependency on feedback. The shortcomings of PID can be compensated by using computer vision and fuzzy logic controller. So in parallel with PID, the proposed controller will be used to overcome the issues and limitations mentioned above.

A new algorithm will be applied to sum up the advantages of PID with computer vision based on fuzzy logic rules. Now let us consider the following decision vector:

$$\mathbf{x}(k) \equiv \left[x_k^1, x_k^2, \ldots, x_k^N\right]^{\mathcal{T}} \qquad (3)$$

(N) is denotes the number of variables used for decision making of the Robot of k iterations. These variables are the variables that can be controlled, for example, the speed of the wheels. The vector that contains the external measurements look like the following:

$$\mathbf{y}(k) \equiv \left[y_k^1, y_k^2, \ldots, y_k^M\right]^{\mathcal{T}} \qquad (4)$$

M is the Robot's number of sensors at (k) iterations, two sensors that repeatedly read the measurements over time, and 120 frames per second Camera. The optimal $x(k)$ can be derived from two functions, one containing $y(k)$ entries related to the infrared IR sensor and denoted by $\hat{y}(k)$. The second function contains the entries of the camera sensor denoted by $\dot{y}(k)$.

$$x(k) = F_1(\hat{y}(k)) + F_2(\dot{y}(k)) \qquad (5)$$

In other words, F_1 takes care of the steering problem, and F_2 manipulates the speed parameter and adjusts the gains of the PID controller. The mechanism of navigation can be written as follows:

Algorithm 1: The proposed Algorithm for navigation of the Robot.

Result: Augmented Decision Vector $x(k)$

initialization;
while *True* **do**
  Capture IR measurements;
  Capture an Image Frame;
  Detect Lines;
  if *Only one Line Detected* **then**
    Calculate Length of the Line;
  else
    Calculate the Intersection Point;
    Retrieve Position of the Intersection Point;
    Calculate Length of the Line;
  end
  Based on the current speed of the motors, calculate $\mathcal{X}_1$;
  Based on the Length of the Line, calculate $\mathcal{X}_2$;
  Use the Fuzzy Ruled-Based System to Calculate $\mathscr{F}_2$;
  Retrieve the optimal K_p, K_i and K_d values for $\mathscr{F}_2$;
  Calculate $\mathscr{F}_1$ based on the IR sensor's measurements;
  Calculate $x(k)$;
end

In this implementation, the output of function F_2 is based on the fuzzy interface. This word uses Mamdani Fuzzy rules, which consists of two inputs and one output.

The first input to the fuzzy controller is the current maximum speed of the motors and the second input is the length of the detected Line. Both can be calculated using the equations below:

$$\mathcal{X}_1 = \frac{\max\{\mathbf{x}_k^j\} \times 100}{|X'|},\ j \in [1, \dots, N] \quad (6)$$

$$\mathcal{X}_2 = \frac{L_k \times 100}{|L'|} \quad (7)$$

The L hat component denotes the maximum length of the path that the sensor can measure.

For detecting the black lines and calculate their length on the white surface, a specific algorithm operated on the Pixy2 camera sensor's MCU to detect the line segment and length. The camera sensor detects all the lines ahead and their intersections concerning a primary line defined under Robot's front point. In most cases, only one or two lines will be detected at every time step.

A voting system is used to detect more than one Line by selecting the intersection of the lines with the primary Line. In this way, the calculation of the line segment and length will be more straightforward.

In the **Figure** below, some of the common cases necessary to measure the line segment and length were illustrated.

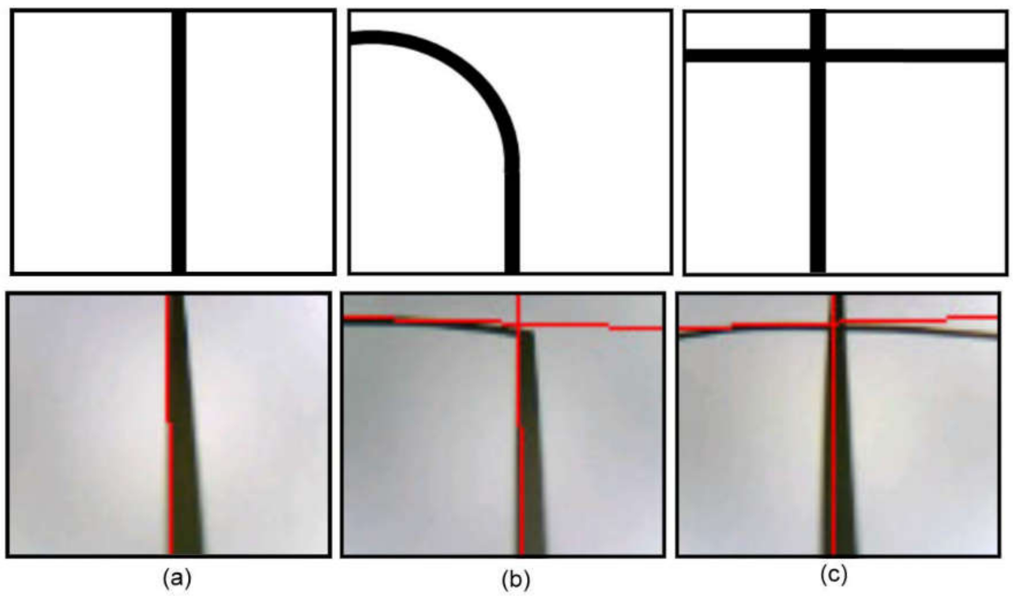

Figure 16: Some cases of detected lines and intersection points.

The controller's fuzzy output is produced from the fuzzy inputs by combining them using six IF AND THEN formats. The list of rules can be explained in the following **Figure**:

IF $\mathcal{X}_1$	**AND** $\mathcal{X}_2$	**THEN**	**IF** $\mathcal{X}_1$	**AND** $\mathcal{X}_2$	**THEN**
Low	Close	LC	Medium	Far	MF
Low	Far	LF	High	Close	HC
Medium	Close	MC	High	Far	HF

Figure 17: The fuzzy input rules (Inference rules).

The results of combining the input rules are a set of fuzzy output variables. The six membership functions produced are shown in the **Figure** below:

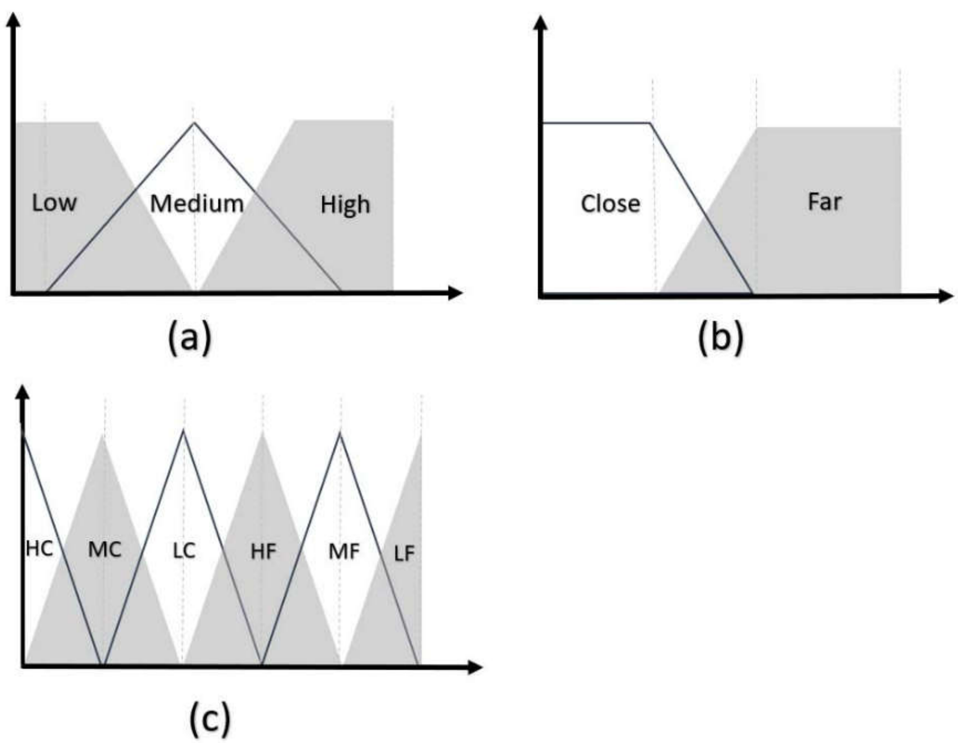

Figure 18: The fuzzy output triangular membership functions.

Figure 18 (a),(b) is the input, and (c) is the output of the proposed system.

Finally, the brittle output will be generated by the defuzzification process known as centroid or center of gravity.

The efficiency of the system was tested using experiments. The track line of the experiment was set up on the three-meter by three-meter white PVC. The black line length is calculated to be 20 meters in length. The path is designed to mirror reality by including sharp angle turns and complex line paths. Then to assess the proposed system, four identical systems were formulated as shown in the results below.

Setup	Avg Speed	Success
Classical PID Controller	1.25 m/s	35/100
PID Controller with EDF	1.65 m/s	43/100
Proposed Method with PID Controller	1.55 m/s	92/100
Proposed Method with PID Controller and EDF	1.95 m/s	90/100

Figure 19: The experimental result of four identical systems with different control techniques.

One hundred tests were performed for each system to follow the complete path, and the speed was registered, as shown in the above **Figure**. The success rate shows that the proposed method with a PID controller has 92 success out of 100 trials at a speed of 1.55 meters per second. The last system has a lower success rate but a higher speed than the proposed method.

4-3 Line Following Robot based on Image Processing

(Sarwade *et al.*, 2019) studied the Line following Robots based on image processing. The camera sensor is used to take the path's image, and then it will be converted to a bitmap image format. The least-square method is used to follow the predefined path, and by calculating the slope of the Line, the Robot steering can be guided along the path.

The proposed system blocks are shown in the **Figure** below. The system consists of the following parts:

1- Camera Sensor.
2- Hellobot MCU.
3- Stepper motors.

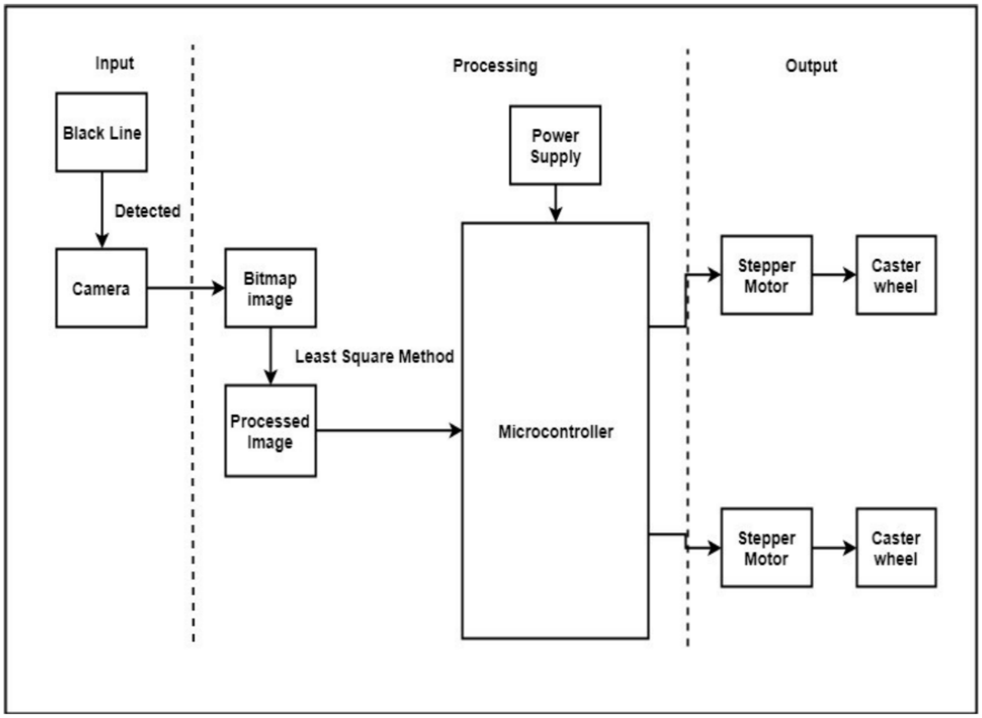

Figure 20: The parts of the proposed system.

The principle work of the system can be summarized in the following steps:

1- The Camera captures the image of the path.
2- The image will be sent to the MCU.
3- The steering of the Robot will be calculated by the Least square method.

The stepper motor can be driven according to the MCU's bitmap image from the camera sensor. The least-square method algorithm will calculate the central Line of the path.

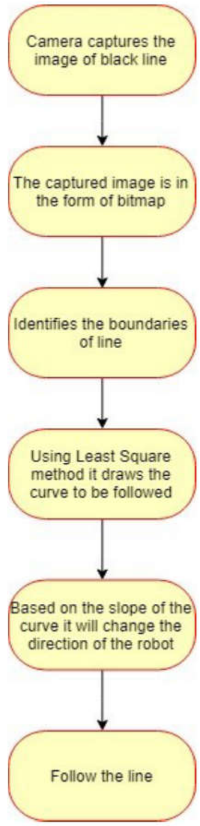

Figure 21: the flow chart proposed work

The image in the **Figure** below captured by the Camera and then processing shows that the centerline is successfully calculated.

a: Input line b: processed Line.

Figure 22: the standard input line and the processed image.

5-Line follower Robot based on Arduino for hospitals

(Chaudhari, Desai and Gavarskar, 2019) worked on designing a line following Robot based on Arduino for transporting medical materials from one location to another location inside the hospital or any medical facility or industrial plant. This work aims to implement the Line following Robot by tuning the control system's parameter and achieving better performance. Integrating robots in health centers and hospitals is to automate and support the healthcare staff in case of emergency and transferring goods from one place to another place efficiently and accurately.

In this work, IR sensors, Arduino MCU, motor drivers, and motors were used.

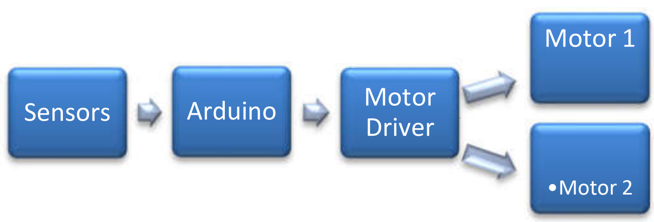

Figure 23: Block diagram of the proposed design

The Line following Robot detects signals from two IR sensors, one placed on the right side and the second is placed on the left side. Once the Arduino MCU receives the reading, the board decides the Robot's movement on the path. The working principle is easy and straightforward; if the left and right sensors are on the white path, the Robot will move forward. If the left sensor is on the black path, the Robot will move to the left, and conversely, if the right sensor is on the black path, the Robot moves to the right. Finally, when both the right and left sensors are on the black path, the Robot will stop moving.

In conjunction with ID sensors, photodiodes are mounted next to the IR sensor to detect the reflected light from the path.

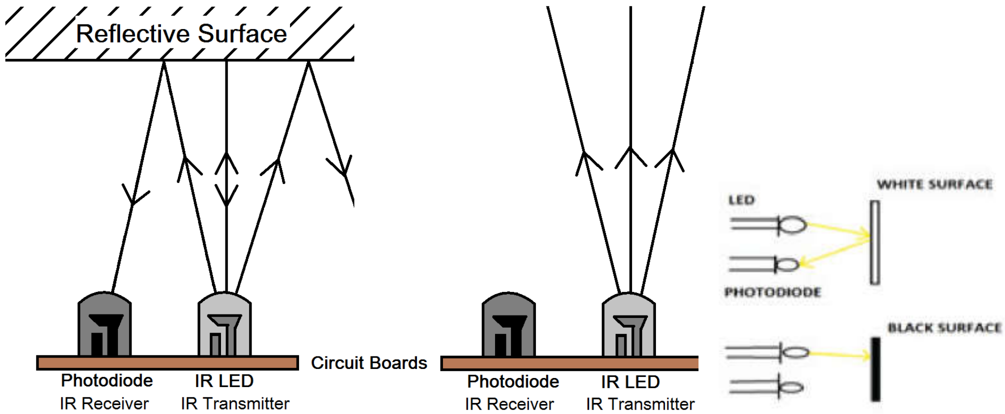

Figure 24: The IR sensor working mechanism.

The proposed design tested on the uniform and non-uniform paths to test the Robot's performance while working. The test shows that the Robot will be disturbed while it is moving on the non-uniform path.

The algorithm used to drive the Robot is shown below:

1. START
2. Read LM and RM
3. If LM and RM both on white surface.
4. Move forward (rotate both motor on full speed)
5. Go to step-2
6. If LM on black line
7. Move left (reduce left motor speed to half)
8. Go to step 2
9. If RM on black line
10. Move right (reduce right motor speed to half)
11. Go to step 2
12. If LM and RM on black line
13. Stop

This work shows that if the Robot moves forward, the speed is constant, and it can pass three meters (3 meters) in thirty-one seconds (31 seconds). When the Robot moves on curved paths, the performance will decrease, and it can pass three meters (e meters) in thirty-six seconds (36 seconds). When the path has an s-shape, the Robot takes a longer time, and when the shape is a sharp-conic path, the Robot will be out of the Line and stop.

6- Path Planning of Line Follower Robot

(Engin and Engin, 2012) worked on developing a line following Robot that uses LM3S811 MCU, which is an arm cortex-3 Microcontroller. The design uses PID to increase the reliability of the navigation of the Robot. This Robot will be used in the competition, and thus it has required to be lightweight and fast. High-speed motors were used with high sensitivity signal conditioning circuits. The photo of the top and bottom of the proposed design is shown in the **Figure** below:

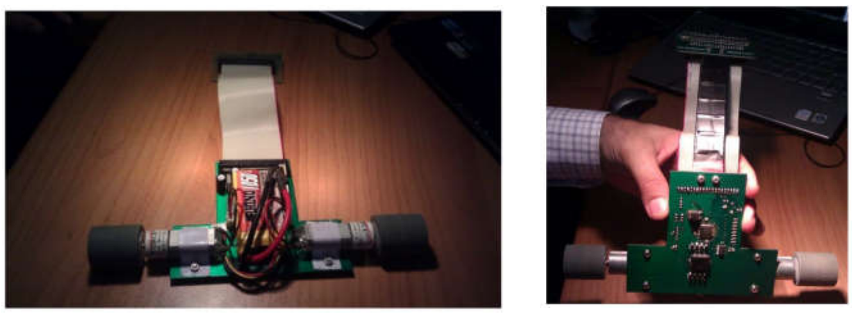

Figure 25: Top and front the views of the proposed design.

The movement mechanism is straightforward. When the two motors are rotating at the same speed, the Robot will move forward, and when the left motor is working and the right motor is turned OFF, the Robot will rotate to the right and vise versa.

For detecting the Line, the quadratic line detection algorithm was used instead of classical detection methods. Eight reflective optical sensors were used with this technique, and the most left sensor is marked to have a coordinate zero (0).

When three sequential reflective optical sensor readings are higher than the others, it means that the Line has been detected, as shown in **Figure** (26). Let us assume that the coordinate of the three sensors is X1, X1+1, and X1+2. Then one can construct a relationship between the three sensors to be:

$$y_1 = ax_1^2 + bx_1 + c \qquad (8)$$

$$y_2 = a(x_1 + 1)^2 + b(x_1 + 1) + c \qquad (9)$$

$$y_3 = a(x_1 + 2)^2 + b(x_1 + 2) + c \qquad (10)$$

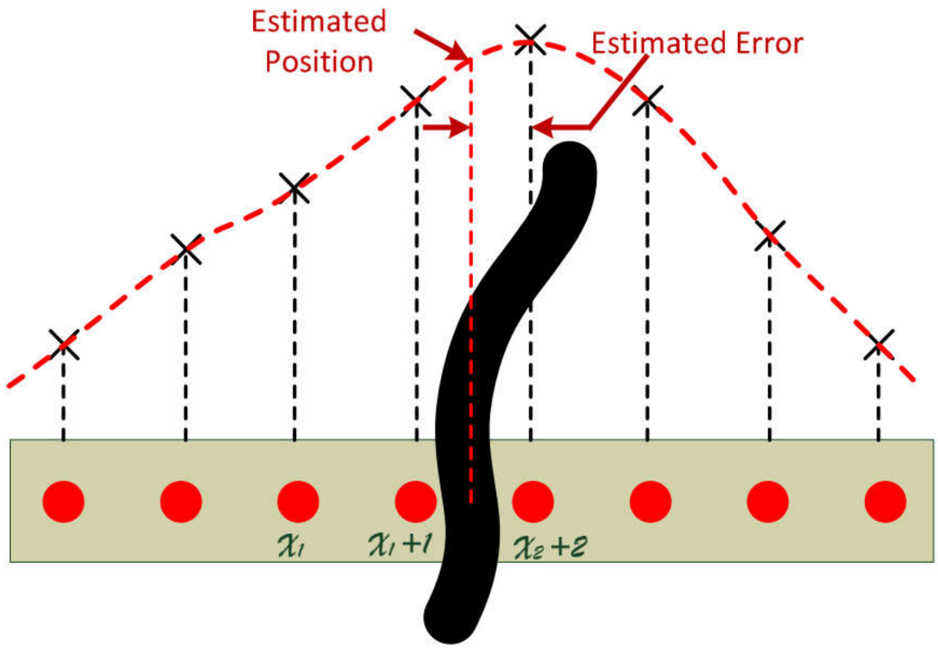

Figure 26: Line detection using the quadratic line detection method

To minimize the error in path detection and planning, the PID controller is introduced to follow the Robot's racing path efficiently. The error between the path and the center point of the sensor array can be calculated and minimized by the PID controller.

First, the PID controller calculates the Robot's current location concerning the path line, then when the error is high, the controller makes the motors make a high turn, and if the error is small, the turn will be smaller.

The pseudo-code can be written in this way:

```
Kp = 10
Ki = 1
Kd = 100
offset = 45      ! Initialize the variables Tp = 50
integral = 0    ! the place where integral value will be stored
lastError = 0   ! place where last error value will be stored
derivative = 0 ! place where derivative value will be stored
Loop forever
    LightValue = read sensors        ! read sensors.
   error = -x        ! calculate the error using equation (7).
   integral = integral + error       ! calculate the integral
   derivative = error - lastError    ! calculate the derivative
   Turn = Kp*error + Ki*integral + Kd*derivative
    powerA = Tp + Turn              ! power level for motor A
    powerB = Tp - Turn              ! power level for motor B
   MOTOR A direction=forward power=PowerA
   MOTOR B direction=forward power=PowerB
   lastError = error         ! save the current error
   end loop forever          ! do it again.
```

The gain tuning and adjusting are not comfortable, and there is no method to measure these quantities. The only solution is to play with the gains to find suitable values for the design.

To test the Robot performance programmed with PID, the Robot programmed with a simple on/off method for comparison. As a result, the comparison shows that the PID controlled Robot can complete the whole path faster, smoother, and with a low tendency to stray from the Line than the simple on/off method.

7- Conclusion

1- In (Eleftheriou *et al.*, 2020) work, the line follower performance, efficiency, and speed can be improved by integrating more sensors and better controlling technology. Fuzzy Logic was integrated into this work to overcome the issues and downside of classical PID controllers; some of these disadvantages were demonstrated in this review, and the main disadvantage was that PID suffers from overshoot and dependency on feedback.

2- In (Sarwade *et al.*, 2019) work**,** the algorithm can find a path and detect the path's centreline. Nevertheless, the performance and efficiency of the system are not calculated. One can implement the same system and calculate this system's performance, and then it can be compared to the systems presented in **Figure** (19).

3**-** In (Chaudhari, Desai and Gavarskar, 2019) work, Placing Robots in preserving and conveying goods from one place to another place autonomous is an immeasurable idea. This work shows that the robot work is not robust and smooth when the path is disturbed, especially in s-shaped and conic-sharp paths. It will cause problems in supporting and delivering goods inside the hospital.

The results of this work depend on measuring the speed of the Robot during the path. However, the authors were supposed to include testing the Robot under load conditions too. Because this type of Robot should carry some payloads from one place to another, in this case, the Robot should be able to carry the payload, and the performance should be robust and controllable throughout the path. Technically this work contains low-quality images and repeated sentences and paragraphs.

4- In (Engin and Engin, 2012) work, the Line following Robot in this work combines two techniques for detecting the path and guiding the Robot throughout the path. The result is better performance, high completion of the path, and the path line's low tendency in straying from the path compared to another method, the on/off method.

5- In (Tayal et al., 2020) presented some general concepts of Line Follower Robots, and the authors tried to keep it simple without going more in-depth in implementing the work. In the paper, low-quality images were used in the Figures, which are supposed to be of better quality. Some of the images used above are the same copy of the paper's images but in higher quality. The overall schematic diagram, which shows the physical connection between the parts used in this paper, is not presented, which is the main downside of this work. When the authors confer about how the IR Sensor can be used in the first Figure, the statement is not clear, and they could show more effort to define the sensor's actual function. During the review, it is clear that the paper focuses on the subject technically and without mentioning any mathematics, kinematics, or dynamic equation of the mobile robots. For example, the type of battery used is not a crucial point, or the advantages of one type over the other types are not necessary.

However, it mentioned that Lipo batteries are used in work, but if one looks at the Robot's image in Figure (2), you can see that the battery used is a regular 9-volt carbon-zinc battery.

As mentioned above, the paper did not refer to any dynamic equations and not even one single algorithm.

8- References:

Bickraj, K. *et al.* (2006) 'Fuzzy Logic Based Integrated Controller for Unmanned Aerial Vehicles', *Florida Conference on Recent Advances in Robotics*.

Chaudhari, J., Desai, A. and Gavarskar, S. (2019) 'Line following robot using arduino for hospitals', *2019 2nd International Conference on Intelligent Communication and Computational Techniques, ICCT 2019*, pp. 330–332. doi: 10.1109/ICCT46177.2019.8969022.

Eleftheriou, G. *et al.* (2020) 'A Fuzzy Rule-Based Control System for Fast Line-Following Robots', *Proceedings - 16th Annual International Conference on Distributed Computing in Sensor Systems, DCOSS 2020*, pp. 388–395. doi: 10.1109/DCOSS49796.2020.00068.

Engin, M. and Engin, D. (2012) 'Path planning of line follower robot', *EDERC 2012 - Proceedings of the 5th European DSP in Education and Research Conference*, pp. 1–5. doi: 10.1109/EDERC.2012.6532213.

Ghani, A. and Tahour, A. (2012) 'Application of Fuzzy Logic in Control of Electrical Machines', *Fuzzy Logic - Controls, Concepts, Theories and Applications*, (March 2012). doi: 10.5772/35770.

Jin, D., Fang, Z. and Zeng, J. (2020) 'A Robust Autonomous following Method for Mobile Robots in Dynamic Environments', *IEEE Access*, 8, pp. 150311–150325. doi: 10.1109/ACCESS.2020.3016472.

Sarwade, J. *et al.* (2019) 'Line following robot using image processing', *Proceedings of the 3rd International Conference on Computing Methodologies and Communication, ICCMC 2019*, (Iccmc), pp. 1174–1179. doi: 10.1109/ICCMC.2019.8819826.

Sharma, D. (2011) 'Designing and Modeling Fuzzy Control Systems', *International Journal of Computer Applications*, 16(1), pp. 46–53. doi: 10.5120/1973-2644.

Singh, B. and Mishra, A. K. (2015) 'Fuzzy Logic Control System and its Applications', *International Research Journal of Engineering and Technology (IRJET)*, 2(8), pp. 743–746.

Tayal, S. *et al.* (2020) 'Line Follower Robot: Design and Hardware Application', *ICRITO 2020 - IEEE 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)*, pp. 10–13. doi: 10.1109/ICRITO48877.2020.9197968.