IJSRD - International Journal for Scientific Research & Development| Vol. 11, Issue 5, 2023 | ISSN (online): 2321-0613

Design and Construction of Fastest Line Following Robot Using QTR-8RC Array Sensor

Ayush Hatwar[1] Sachidanand Kulkarni[2] Ravindra Vidhate[3] Kirti Nikhar[4] Yasin Sayyed[5]
[1,2,3,4,5]UG Students
[1,2,3,4,5]Department of Mechanical Engineering
[1,2,3,4,5]Government College of Engineering and Research Avasari(kh), Pune, India

Abstract — The automated guide vehicles which follow the assigned path using various sensors are the key technology for mobility in industries. In the presented research work, the fastest line-following robot has been designed and integrated which predominately follows the black line path. The QTR infrared sensor of eight arrays is implemented for sensor input toward the micro-control system. Robot fully working on PID algorithm. Through enough interaction the efficiency and the accuracy can be increased for the robot had been 3D printed with different additive manufacturing materials. Overall, the unique robot design with greater accuracy is achieved.

Keywords: QTR sensor, PID algorithm, Blackline follower, 3D Printed, Traction

I. INTRODUCTION

In Industry 4.0, multinational product design and manufacturing companies mostly rely on automated guided robots in order to transport the product or material from one location to another within the working area. A line follower robot is an autonomous machine that is designed to follow a specific path or track marked by a contrasting color or a line. It utilizes sensors and a control system to detect and track the line's position, allowing it to navigate along the predetermined path. The introduction aims to provide an original and plagiarism-free overview of a line follower robot, discussing its components, working principle, and potential applications.

Abbreviation-

- QTR - Charge transmitter Resistance Sensor
- PID - Proportional Integral Derivative
- CAD – Computer-Aided Design
- PLA - Polylactic Acid

Ebiesuwa O. et. al represented microcontrollers that received signals and controlled robots through programs. An optocoupler was used to combine the infrared light and the phototransistor. Generate an analog signal. They used a CMOS (89C52) based 8051 microcontroller. It has an optical sensor that recognizes colors and calculates the shortest distance. This process takes only a few nanoseconds. This will be one of the fastest line-following robots. They also use a speed detector for this [1]. They have used two IR sensors that are connected to the Arduino Uno. They shared their algorithm for the robot. They also proposed the idea of automotive parking. They have used sharp IR sensors, flags, and buzzers for performing this operation. In flags, they discussed resident and visitor slots [2].

Chaudhari et al., The usage of LED and photodiodes, which emit and measure IR radiation intensity, was made. They suggested a robot's operating system and algorithm. Differentiated between surface quality and evaluated robot performance based on lane curvature and vehicle speed [3]. The robot was constructed using an Arduino, two IR sensors, four LEDs, and a phototransistor. The IR sensor transmits messages to the Arduino after detecting the robot at that precise instant and identifying various colors. They discussed algorithms that could provide information. They included details on a variety of hardware, including motor drivers, sensors, regulators, batteries, comparators, etc. A truth table was used to present the experimental outcomes of the robot moments [4].

II. CONSTRUCTION OF THE LINE FOLLOWING THE ROBOT

A. Parametric CAD Model

Figure 1 depicts the parametric CAD model of the Robot body with electronic equipment. Geometric modeling software licensed to Solid Works was used to create the entire model. The mounting points for OEM-based motors and other electronic modules give the robot body an ergonomic shape.

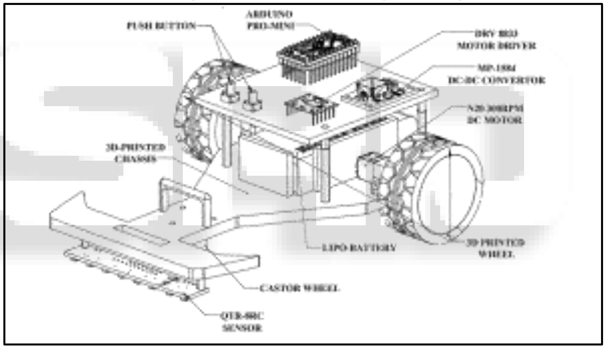

Fig. 1: Parametric CAD Model

The Physical parameters of the proposed robot are shown in Table 1.

No.	Parameter	Value
1.	Wheel Base	15 cm
2.	Track Width	13 cm
3.	Height	8 cm
4.	Max rpm	300 rpm
5.	Sensing ranges from ground	3 mm

Table 1. Physical Parameters

B. 2D Layout with Component placement

The 2D layout of component placement is shown in Figure 2.

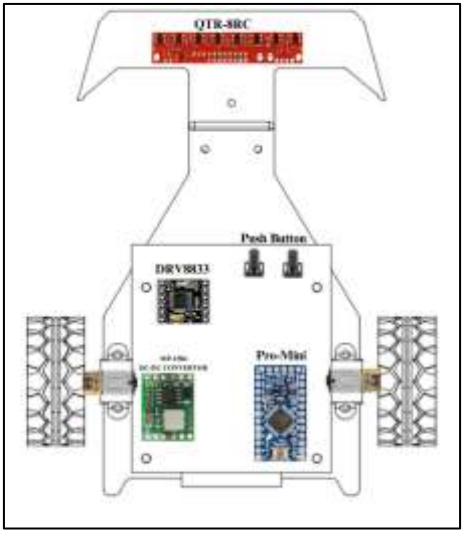

Fig. 2: Two-dimensional layout

C. Material Selection

The robot's structural body is composed entirely of PLA pro material. Additional mechanical subsystems and connections are also made with the aid of 3D printing and high-strength PLA pro quick prototyping material. Processing this material is simpler. It might decompose into a byproduct that is not harmful. Being biocompatible and degradable helps reduce the amount of plastic waste.

III. CONTROL SYSTEMS AND MANEUVERABILITY

A. Components Selection

The main components required for robot development are listed below.

1) Arduino mini pro -

The Arduino Pro Mini is a small, inexpensive microcontroller board based on the ATmega328P microcontroller, designed for projects with a smaller footprint and less power.

2) DRV883 Motor driver -

A motor driver integrated circuit (IC) called the DRV883 is used in electronic applications to control stepper motors or DC motors. Its voltage range ranges from 0V to 11V or from 0V to 36V. PWM inputs or straightforward logic-level inputs can be used as the DRV883 ICs' control interface to control speed. To safeguard the motors and the IC, they frequently contain built-in overcurrent and thermal protection devices. When motors are not in use, some models can include sleep mode or low-power standby mode to conserve electricity.

3) QTR-8rc -

It is nothing but an array of IR transmitter LED and photodiode. Transmitters transmit infrared radiation. Photodiodes absorb radiation accordingly and send signals to the Arduino. It is very precise and calibratable. It has an operating voltage between 3.3v to 5v.

4) Mp1584dc dc converter -

It has an input voltage between 4.5 v to 28 v. it has a current range from 0 to 3 amp. it has an output voltage between 0.8 v to 19(experimental data). It is very small in size.

5) Push button -

It is used for only calibration purposes. With the help of the push button, we can on and off the calibration mode of the line following bot.

B. PID Algorithm working

A PID (Proportional-Integral-Derivative) algorithm is commonly used in line-following robots to control their movement along a specified path. The algorithm utilizes sensors to detect the position of the line and calculates an appropriate control signal to keep the robot on track. The basic outline of how a PID algorithm can be implemented for a line-following robot is given as,

1) Sensor Reading: The robot is equipped with QTR RC line sensors, that detect the line beneath it.
2) Error Calculation: Calculate the error term, which represents the deviation of the robot from the desired path. This can be done by subtracting the desired position from the sensor readings.
3) PID Terms: The PID algorithm consists of three components: proportional, integral, and derivative. These terms are calculated as follows:
   - Proportional (Kp) Term
   - Integral (Ki) Term
   - Derivative (Kd) Term
4) Control Signal Calculation: Combine the proportional, integral, and derivative terms to calculate the control signal that will be sent to the robot's motors. The control signal determines the speed and direction of the robot's movement. The basic formula for calculating the control signal is:

   Control Signal = (P * Kp) + (I * Ki) + (D * Kd)
5) Motor Control: Apply the calculated control signal to the robot's motors to adjust their speeds and direction of rotation. The specific motor control mechanism will depend on the hardware and motor driver being used.

C. Electronic Block Diagram and Flow Chart

The block diagram of the connection of the components is shown in Figure 3.

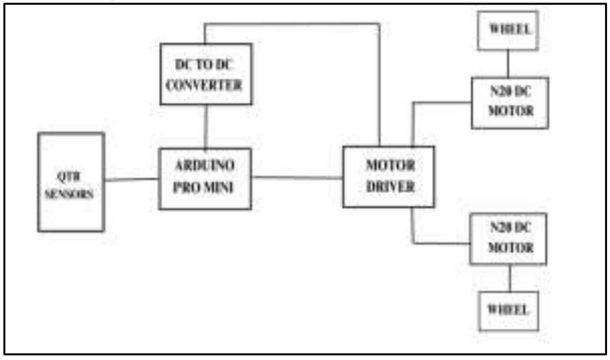

Fig. 3: Block Diagram

1) Process Chart

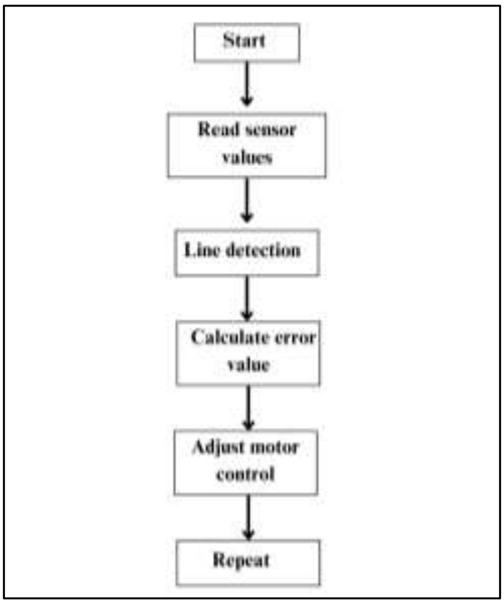

Fig. 4: Process chart

Explanation of the flowchart:

- Start: The program starts.
- Read sensor values: The QTR sensors' values are read using appropriate functions or methods.
- Line detection: Based on the sensor readings, determine which sensors are over the line and which are not.
- Calculate error value: Use the sensor readings to calculate an error value, indicating how far the robot is deviating from the desired line position.
- Adjust motor control: Based on the error value, adjust the motor control signals to steer the robot back to the line.
- Repeat: Continuously repeat steps 2 to 5 in a loop to monitor and adjust the robot's position.
- End: The program ends.

D. Calibration

Set the sensor threshold value that distinguishes between the line and the background. Determine the minimum and maximum sensor readings by moving the robot over the line and the background. Adjust the threshold value to a suitable range within the minimum and maximum readings to ensure accurate line detection.

IV. RESULTS

The results of a black line follower robot using QTR sensors can vary depending on the specific implementation and environmental conditions.

1) Line tracking: The robot should be able to accurately track the black line and follow its path if the line is properly detected by the QTR sensors.
2) Speed and responsiveness: The speed at which the robot can track the line and respond to deviations will depend on factors such as the processing power of the microcontroller, the algorithm used for motor control, and the frequency at which sensor readings are processed. Generally, the robot should be able to maintain a reasonable speed while effectively tracking the line.
3) Precision and accuracy: The accuracy of the line-following robot using QTR sensors can vary based on the calibration of the sensors, the line detection algorithm, and the precision of the motor control. Proper calibration and fine-tuning of the algorithm can improve the precision and accuracy of the robot's line tracking.
4) Adaptability to line variations: The robot should be able to handle minor variations in the line, such as curves, intersections, and gaps. However, the performance may degrade if the line becomes too faint, discontinuous, or if there are abrupt turns that exceed the capabilities of the robot's turning radius.

V. CONCLUSION

The use of QTR sensors provides advantages such as simplicity, cost-effectiveness, and ease of integration with microcontrollers. The calibration process allows the sensors to be optimized for the specific line and background conditions, enhancing the robot's ability to differentiate between the line and its surroundings. With proper calibration and algorithm implementation, the black line follower robot using QTR sensors can achieve precise line tracking and maintain a desired path. By continuously monitoring the sensor readings, calculating error values, and adjusting motor control, the robot can make real-time corrections to stay on the line even in the presence of minor line variations or environmental changes.

It is important to note that the performance of the robot may be influenced by factors such as lighting conditions, line complexity, and the quality of sensor calibration. It is recommended to test and fine-tune the robot's parameters, including sensor threshold values, motor control algorithm, and PID control if implemented, to optimize its line-following capabilities.

Overall, a black line follower robot using QTR sensors is a reliable and versatile solution for applications such as maze solving, industrial automation, and educational projects. With careful design and implementation, the robot can successfully navigate and track black lines, demonstrating efficient line-following behavior.

ACKNOWLEDGMENT

We would like to thank Dr. S.V. Kshirsagar, the department head, for his invaluable assistance to the group. We also appreciate Prof. N. D. Padwale, an assistant professor, for his invaluable advice along the way. We thank everyone who contributed to the creation and design of the robot.

REFERENCES

[1] Ebiesuwa O. O, Adekunle Y. A, Akinyemi L. A, Oyerinde O. D, 2013, Line Follower Robot Using A Sophisticated Sensor Approach, International Journal of Engineering Research & Technology (IJERT) Volume 02, Issue 07 (July 2013).

[2] Md Younus , Pooja Gadekar, Adhiraj Walse, 'Line Follower Using Arduino And Its Application', International Journal Of Applied Engineering Research ISSN 0973-4562 Volume 14, Number 13, 2019.

[3] J. Chaudhari, A. Desai and S. Gavarskar, "Line Following Robot Using Arduino for Hospitals," 2019

2nd International Conference on Intelligent Communication and Computational Techniques (ICCT), Jaipur, India, 2019, pp. 330-332.

[4] Mr. Shivanshu Gautam, Ms. Shefali Sharma ,Mr.Vikram Singh, Mr. Sheetal Kumar Dixit, 'Case Study: Line Follower Robot Analysis', International Journal of Creative Research Thoughts (IJCRT), | Volume 9, Issue 2 February 2021

[5] Roy, Abhishek, and Mathew Mithra Noel. "Design of a High-Speed Line Following Robot That Smoothly Follows Tight Curves." Computers & Electrical Engineering 56 (November 2016): 732–47.

[6] Pakdaman, Mehran, M. Mehdi Sanaatiyan, and Mahdi Rezaei Ghahroudi. "A Line Follower Robot from Design to Implementation: Technical Issues and Problems." The 2nd International Conference on Computer and Automation Engineering (ICCAE), February 2010.

[7] Patil, Samruddhi, Ameya Wagh, Mitali Sawant, Saurav Panda, and Aditya Bhopale. "Design and Implementation of Advanced Auto Calibrating Line Following Sensor for Coloured Surfaces with a White Line." 2016 IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES), July 2016.