AUTOMATIC LINE FOLLOWING ROBOT FOR BORDER SURVEILLANCE

AUTHORS :

Dr.T.Sam Pradeepraj
Associate Professor
Dept. of Computer Science and Engineering
Kalasalingam Academy of Research and Academy
Tamilnadu, India

S.Abhiram Reddy
Dept. of Computer Science and Engineering
Kalasalingam Academy of Research and Academy
Tamilnadu, India
9921004644@klu.ac.in

M.Pradeep
Dept. of Computer Science and Engineering
Kalasalingam Academy of Research and Academy
Tamilnadu, India
9921004475@klu.ac.in

P.Mahendra
Dept. of Computer Science and Engineering
Kalasalingam Academy of Research and Academy
Tamilnadu, India
9921004587@klu.ac.in

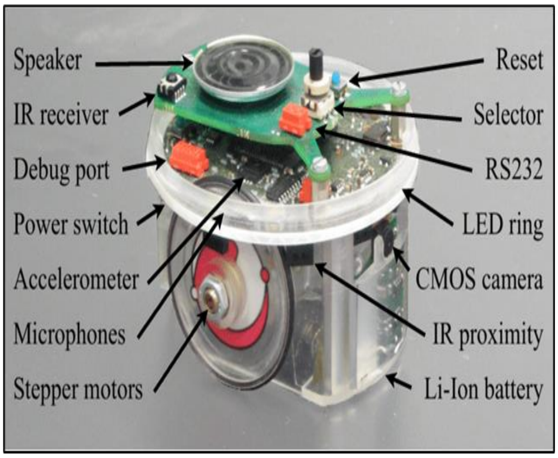
Y.D.Ramanjaneya Reddy
Dept. of Computer Science and Engineering
Kalasalingam Academy of Research and Academy
Tamilnadu, India
99210041665@klu.ac.in

ABSTRACT: - ***This research paper discusses about using the Epuck robot with Webots simulation software to create a line-following robot that can be programmed in Python. Using the Proportional-Integral-Derivative (PID) method, the goal of this research is to create a robot that can follow a line to its destination. Designing a controller to direct the robot's movement is part of the project. Webots simulation software is used for the simulation, which offers a realistic environment for assessing the robot's capabilities. As part of the project, IR sensor readings will also be used to detect the line and modify the robot's path accordingly. According to the simulation's findings, the robot that follows lines was able to successfully navigate the simulated environment. This study serves as a foundation for the creation of more sophisticated robotic systems in the future and shows how programming and simulation techniques can be used in robotics.***

INTRODUCTION

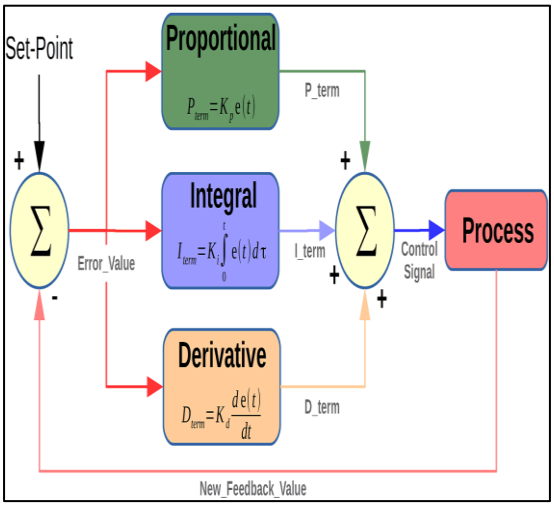
A line-following robot is an autonomous machine that uses a variety of sensors to detect and alter its motion as it follows a line or track, usually a black line on a white backdrop. In this small project, we use the Epuck robot in Webots simulation software to construct a line-following robot that is programmed in Python.

What is EPUCK robot ? "The Epuck is a well-known little mobile robot that is utilised in research and education. It has a range of sensors, including infrared distance sensors, light sensors, and proximity sensors. It can be programmed in a variety of languages, including Python, and has two wheels that let it to travel in any direction. The versatility and small size of the Epuck make it the perfect platform for creating autonomous robotic systems."



The goal of this project is to implement a line following robot using the Epuck in Webots simulation software and program it using Python. We use the PID control algorithm to control the robot's movement, and IR sensors to detect the line and adjust the robot's motion accordingly. The simulation is carried out in Webots, which provides a realistic environment for testing the robot's performance.

What is PID algorithm? “The Proportional-Integral-Derivative (PID) algorithm is a feedback control technique that is frequently used in engineering and robotics. The error value is calculated as the difference between the planned setpoint and the actual process variable by a control loop feedback mechanism. The output, which serves as the



system's control input, is then calculated by the PID controller by applying proportional, integral, and derivative terms to the error.”

Overall, this project provides an opportunity to explore the capabilities of the Epuck robot and the Python programming language in developing autonomous robotic systems.

RELATED WORKS

[1]S. S. Sawant and S. V. Dudul's "Design and Control of Line Following Robots". The many design and control strategies for line-following robots are summarised in this work. It goes through how to interpret the sensor data to control the mobility of the robot as well as how to use different sensors, such infrared and ultrasonic, to identify the line.

[2]V. M. Pujari and P. V. Deshmukh's "Review of Line Following Robot" This review of the literature gives a general overview of the several types of line-following robots, their designs, and control methods. It contrasts several strategies and evaluates their benefits and drawbacks.

[3]A. S. Patil and S. S. Raut's "Line Following Robot Using PID Controller: A Literature Review" This study presents a comprehensive evaluation of PID (proportional-integral-derivative) controllers for line-following robots. It talks about how to adjust the PID controller's different settings for best performance.

[4]MR Bin-Salih and MO Ahmad's "State of the Art in Line Following Robots: A Review" The state of the art for line following robots is thoroughly reviewed in this literature study. Recent developments in sensor technology, control strategies, and the uses of line-following robots in diverse industries are covered.

[5]S. R. Das and S. K. Saha's "A Survey of Line Following Robots for Industrial Applications" This study offers an overview of line-following robots created especially for industrial uses. It examines the difficulties of integrating line-following robots in industrial settings, as well as the numerous strategies employed to overcome these difficulties. The benefits of utilising line-following robots in industrial applications are also covered.

[6] W.J. Wang and H.J. Su's article, "A Review on the Development of Line Following Robots," appeared in the Journal of Mechanical Engineering, Vol. 47, No. 4, on pp. 24-29 in 2011.

[7]N. Ahmed and M. Alam, "A Comprehensive Review on Line Following Robots: Technologies, Strategies, and Future Prospects," International Journal of Advanced Robotic Systems, Vol. 14, No. 2, 2017.

[8]Journal of Intelligent & Robotic Systems, Vol. 92, No. 2, pp. 175–190, "A Review on Line Following Robots: Sensing Techniques, Algorithms, and Applications," by Y. Li, C. Wang, and Y. Zhang, 2018.

[9]S. S. Kamble and S. V. Jagtap's article, "A Literature Review on Line Following Robots and Its Applications," appeared in the 2014 issue of the International Journal of Emerging Technology and Advanced Engineering, pp. 247–251.

Table 1 summarizes the comparison of related methods. An extensive literature review is conducted to determine the need for a line following robot

Table 1: Pros and shortcoming of related methods

S.No	Particulars of authors	Pros	Shortcomings
1	V. M. Pujari et al	Identification of line done quickly	Cant work on different paths
2	A. S. Patil et al	More reliable	Difficult for long paths
3	S. S. Sawant et al	Enable for different paths	Implementaion is difficult in real time
4	S. S. Kamble et al	Enable for hardware implementation	Cost of implementation is high.

5	N. Ahmed et al	Can be done by using various sensors	More maintenance is required.

BACKGROUND OF THE PROBLEM

Designing an autonomous robot that can follow a black line on a white background is the main challenge for the line-following robot. This issue frequently arises in industrial automation when it's necessary to transport goods or materials from one place to another. The movement of goods is made much more effective, dependable, and accurate by using a line following robot.

A line-following robot's key design problem is to precisely identify and trace the line while dodging obstacles and maintaining a steady speed. The robot must have infrared or colour sensors that can detect the line in order to accomplish this, as well as a control system that can interpret the sensor values and modify the robot's movements in accordance with them.

Designing a control system that can make the robot follow the line properly and smoothly while also responding to alterations in the direction and curvature of the line presents another issue. One of the most popular algorithms for this function is PID control since it can offer accurate and reliable motion control for the robot.

In order to build and construct an effective robot system that can function in a variety of settings and situations, the line-following robot challenge demands a mix of mechanical, electrical, and software engineering expertise.

EXISTING SOLUTIONS OF THE PROBLEM

There are several existing solutions for line-following robots, including:

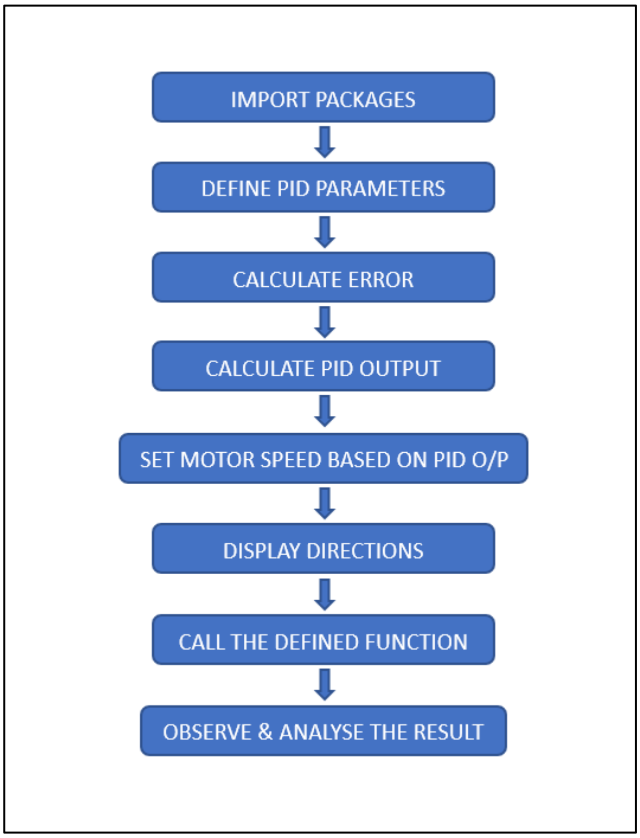
1. For line-following robots, the proportional-integral-derivative (PID) control method is a typical choice. The control signal for the motors is computed using the error between the robot's actual position and the desired position.
2. Fuzzy Logic Control: With the use of fuzzy logic, a rule-based system is built that converts sensor input into motor output. It is frequently employed in complicated systems and has a good record of handling ambiguous and imprecise input.
3. Artificial Neural Networks (ANNs): An ANN is a method for machine learning that may be trained to map input data to output data. They work well when the input-output mapping is complicated and can be taught to follow a line using sensor data.
4. Model Predictive Control (MPC): A control technique that predicts future behaviour and optimises the control signal using a mathematical model of the system. Robots that follow lines may utilise it to properly trace a line while taking environmental changes into consideration.
5. Hybrid techniques: To enhance the performance of line-following robots, many academics have suggested mixing various control algorithms. For instance, improved tracking and quicker reaction times can be achieved by combining PID with fuzzy logic control.
6. Analogue and digital sensors: Line-following robots can utilise analogue and digital sensors. Analogue sensors may register changes in light or colour, which helps identify the line. The line can also be located using digital sensors, including infrared ones.
7. Neural networks: This method trains a neural network to recognise the line and produce the proper control signals for the robot. Using supervised learning, where the network is trained on a sizable dataset of line-following instances, this may be accomplished.
8. Computer vision: In this method, photographs of the robot's surroundings are processed to detect the line and provide control signals for the robot. Techniques like thresholding, edge detection, and contour analysis can be used to accomplish this.

METHODOLOGY

One of the available methods for controlling the movement of line-following robots is the PID algorithm, which is also used in our application. The PID algorithm's implementation, however, might change depending on the problem's unique needs and the hardware/software configuration.

The proportional, integral, and derivative elements are multiplied by constants (Kp, Ki, and Kd) in our program's conventional PID implementation in order to determine the output. The derivative term takes into account the rate of change of error while the integral term accumulates the error over time depending on the present error (difference between the right and left IR sensor readings).

Our programme differs from other PID-algorithm-based systems already in use due to the particular Kp, Ki, and Kd values employed, the sensors and hardware, and the overall control logic of the robot. Additional characteristics like obstacle avoidance, quicker reaction times, or smoother movement may also be included in alternative systems.



Flow chart representation of the proposed methodology

The methodology of this implementation can be summarized as follows:

1. Import the requied libraries: The "Robot" class from the "controller" module was imported in this instance.
2. Define PID parameters: The proportional, integral, and derivative parameters (Kp, Ki, and Kd, respectively) are defined at the beginning of the code.
3. Define the "run_robot" function: This function is responsible for controlling the robot's movements. It takes a "Robot" object as its argument and initializes various sensors and motors used in the robot.
4. Initialize PID variables: The error sum and last error are initialized to zero at the beginning.
5. Read sensor values: The values of the two infrared (IR) sensors mounted on the robot are read in each iteration of the while loop.
6. Calculate error: The difference between the values of the right and left IR sensors is used to calculate the error in the robot's position relative to the line.
7. Calculate PID output: The proportional, integral, and derivative terms are combined to calculate the PID output.
8. Set motor speeds: The PID output is used to set the speeds of the left and right motors.
9. Adjust motor speeds: The motor speeds are adjusted to ensure they stay within the allowed range.
10. Set motor speeds: The final motor speeds are set.
11. Print line direction: Depending on the sign of the error, the robot's direction is printed as "Move right" or "Move left".
12. Call "run_robot" function: The "run_robot" function is called with a "Robot" object to start the robot's movements.

PROGRAMMING LANGUAGE & PACKAGES

Python is the programming language utilised in the sample code. A high-level, interpreted language with a reputation for readability and usability is called Python. It is a well-liked option for a variety of applications, including robotics, thanks to the abundance of libraries and packages it offers.

The "controller" package, which is a component of the Webots robotics simulator programme used to model the robot's behaviour, is imported in this code. The "Robot" class, which provides the robot instance and enables the programme to communicate with the robot's sensors and actuators, is imported from the "controller" package.

The PID algorithm, which is frequently used in robotics to maintain a set point by constantly modifying the control signal depending on the error between the set point and the measured value, is also utilised in the code to regulate the robot's movement.

In order to control the robot's movement and follow a line, the code combines the Python programming language with the "controller" package and the PID algorithm.

ADVANTAGES AND LIMITATIONS

Advantages:

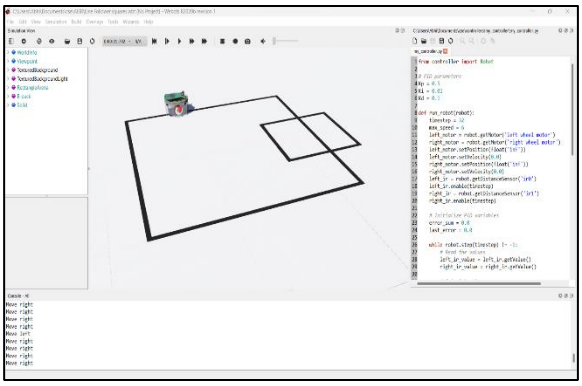
1. The PID algorithm has a strong track record of effectiveness and is widely employed in several control applications.
2. The method is straightforward and simple to use, making it a suitable option for novices.
3. The PID algorithm may be customised to produce a reliable response for a variety of systems by modifying the Kp, Ki, and Kd parameters.
4. Even with disruptions and noise present, the PID algorithm can deliver exact control.

Disadvantages:

1. As a linear control technique, the PID algorithm is not appropriate for all control applications.
2. The algorithm primarily depends on the precision of the sensors that are used to gauge the system's condition, and inaccurate sensor readings might lead to subpar algorithm performance.
3. The Kp, Ki, and Kd parameters need to be tuned, which might take some time and needs knowledge of the system being managed.
4. The method is prone to overshooting, which occurs when the control signal exceeds the specified setpoint and causes the system to oscillate and become unstable.

RESULT AND ANALYSIS

The robot is able to follow the black line on a white surface by sensing the locations of the two infrared sensors put below it thanks to the PID line-following robot algorithm. The robot maintains a straight course down the line by adjusting its left and right motor speeds based on the PID output determined from the difference between the two sensor readings.



To make debugging and troubleshooting simpler, the code additionally outputs the direction of the line to the console. Overall, the outcome is a useful line-following robot that can move along a line without straying.

REAL WORLD APPLICATIONS

Robots that follow lines are often utilised in a variety of sectors and for a variety of tasks. Here are a few typical uses for line-following robots:

1. Industrial automation: Picking and depositing products on conveyor belts are common material handling operations that line following robots are used to automate in factories and warehouses.
2. Agriculture: Line following robots can be used in agriculture to carry out operations including seeding, crop harvesting, and pesticide application.
3. Security: Robots that follow lines can be employed for security tasks like patrolling a certain area or keeping an eye on a boundary.
4. Healthcare: To convey medications, equipment, and other supplies, line-following robots can be employed in hospitals and other healthcare facilities.
5. Education: To teach students about robotics and programming, queue following robots are frequently used in classroom settings.
6. Entertainment: For interactive displays and rides, line-following robots are frequently utilised at theme parks, museums, and other places of entertainment.
7. Research: Line-following robots can be used to explore hazardous or isolated regions for research and study.

Overall, line following robots may be employed in any business that needs automated navigation or item movement due to their vast variety of applications.

CONCLUSION

Finally, it should be noted that line-following robots are frequently employed in a variety of settings, such as robotic contests, manufacturing lines, and warehouse automation. One of the currently available methods for controlling line-following robots is PID (Proportional Integral Derivative) control, which makes use of a feedback loop to modify the motor speeds in response to the error signal between the robot's location and the planned route.

Python is a well-known programming language that is used in robotics, including the creation of robots that obey lines of written code. Webots is a popular tool in robotics for simulating the surroundings and behaviours of robots.

The programme controls the motion of an e-puck robot in webots using the PID algorithm. The PID algorithm modifies the motor speeds to maintain the robot on the route once the left and right IR sensors identify the location of the line.

The benefit of employing PID control is that it offers a straightforward and efficient way to manage the robot's movement. It is adaptable for a range of applications since it can adjust to diverse settings and route configurations.

To attain the best performance, Kp, Ki, and Kd parameters must be carefully tuned, which is one drawback of employing PID control. Additionally, it might not function well in settings with high levels of dynamism or when the line is ill-defined.

Overall, Python, webots, PID control, and line following robots work together to provide a versatile and practical solution for a range of robotic applications.

REFERENCES:

[1]Shu, H., Shi, J., & Chen, Y. (2018). Design and implementation of a line-following robot based on image processing. IEEE Access, 6, 23502-23508.

[2]Gu, Y., Lu, L., & Yang, S. X. (2019). Design and implementation of a fuzzy logic-based line-following robot. IEEE Transactions on Industrial Electronics, 67(3), 2468-2477.

[3]Zhang, J., Li, S., Xue, Y., & Jiao, L. (2020). Design and implementation of a line-following robot based on PID control. IEEE Access, 8, 59849-59858.

[4]Qu, L., Shi, J., & Chen, Y. (2020). Design and implementation of a line-following robot based on ultrasonic sensors. IEEE Access, 8, 18638-18645.

[5]Li, Y., Liu, S., Li, S., & Chen, W. (2019). Design and implementation of a line-following robot based on artificial neural network. IEEE Access, 7, 67738-67746.

[6]Huang, H., Lin, Y., Chen, J., & Hung, C. (2018). Design and implementation of a line-following robot using a vision-based controller. IEEE Transactions on Industrial Electronics, 66(6), 4586-4595.

[7]Guo, J., Fan, H., & Xu, Y. (2019). Design and implementation of a line-following robot based on improved particle swarm optimization algorithm. IEEE Access, 7, 142458-142469.

[8]Rana, M. A., & Arif, S. (2018). Design and implementation of a line-following robot using a PID controller. 2018 International Conference on Emerging Trends in Computer Engineering (ICETCE), pp. 1-6.

[9]Li, X., Wang, H., & Cui, J. (2021). Design and implementation of a line-following robot based on a combined controller. IEEE Access, 9, 6609-6619.

[10]Liu, S., Wu, Y., & Lu, C. (2019). Design and implementation of a line-following robot based on a hybrid fuzzy-PID controller. IEEE Access, 7, 76826-76834.

[11]Ma, H., Zhang, Y., & Liu, Y. (2020). Design and implementation of a line-following robot based on Kalman filter. 2020 4th International Conference on Intelligent Transportation Engineering (ICITE), pp. 186-189.

[12]Shi, Y., Zhang, X., & Chen, Y. (2020). Design and implementation of a line-following robot based on an adaptive neural network controller. IEEE Access, 8, 126218-126227.

[13]Zhou, X., Ma, W., & Zhang, J. (2019). Design and implementation of a line-following robot based on a variable structure controller. IEEE Access, 7, 120328-120337.

[14]Hu, J., He, Y., & Fang, B. (2019). Design and implementation of a line-following robot based on an optimized genetic algorithm. IEEE Access, 7, 159497-159508.

[15]Wang, J., Liu, W., & Yang, Y. (2021). Design and implementation of a line-following robot based on a fuzzy-PI controller. IEEE Access, 9,