Optimization of PID Control for Line-Following Robot

Fatine Azzabi*, Abdul Malik*, Danyal Ahmed*
Department of Electrical-Electronics Engineering, Abdullah Gul University
Kayseri, Turkey
Email : fatine.azzabi@agu.edu.tr, abdul.malik@agu.edu.tr , danyal.ahmed@agu.edu.tr

Abstract-- Line-following robots are essential tools in robotics education and industrial automation, representing an accessible platform for developing and testing control algorithms. These robots utilize sensors to follow predefined tracks and dynamically adjust their movements using feedback control. This report explores the design and optimization of a line-following robot using a Proportional-Integral-Derivative (PID) controller. By integrating methodologies from two comprehensive projects, this study demonstrates how cost-effective components like the STM32F103C8 microcontroller, L298N motor drivers, and QTR-8RC reflectance sensors can be used to create a high-performance robot. Furthermore, innovative strategies such as adaptive tuning, noise reduction, and PWM scaling are introduced to enhance precision and reliability. The challenges encountered, such as handling sharp turns and sensor noise, are analyzed, along with recommendations for future improvements.

I. INTRODUCTION

Line-following robots serve as a cornerstone in robotics, providing practical and educational applications for control theory, sensor integration, and real-time feedback systems. These robots detect and follow a visual path, often marked by a contrasting black or white line, using a combination of sensors and actuators. Adjustments are driven by feedback loops, ensuring accurate alignment with the predefined path.

Such robots are widely used in academia as learning tools and in industries for automated tasks like material transportation and guided navigation. However, achieving efficient performance is challenging when relying on low-cost components. Key issues include sensor calibration, maintaining stability during sharp turns, and real-time control of movements.

This report combines methodologies from two research projects to design a robot capable of addressing these challenges. The robot includes innovative features such as manual joystick control and dynamic error correction. The central goal is to create a reliable, cost-effective robot capable of navigating complex tracks with minimal external intervention. Optimizing the PID controller plays a critical role in achieving these objectives.

II. SYSTEM ARCHITECTURE

A. Hardware Components

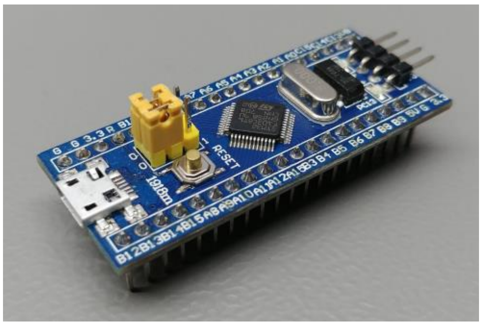

Figure 1: STM32F103C8T6 Blue Pill

A.1 STM32F103C8 Microcontroller

The STM32F103C8 microcontroller is the brain of the robot, chosen for its high efficiency, low power consumption, and rich feature set. Operating at 72 MHz, this ARM Cortex-M3-based microcontroller offers ample processing power for real-time applications. Key features include:

- General Purpose Input/Output (GPIO) pins for interfacing with sensors and motors.
- Advanced timer modules for generating precise PWM signals.
- 12-bit Analog-to-Digital Converter (ADC) for accurate sensor data processing.
- Communication interfaces like USART and I²C for potential future expansion.

The STM32F103C8 is programmed using low-level embedded C code, ensuring precise control over hardware functionality. Its compact size and flexibility make it ideal for integration into robotics projects.

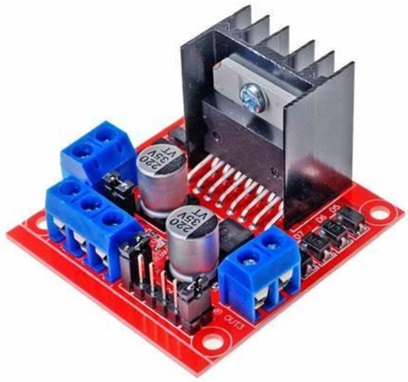

Figure 2: L298N Motor driver with a power transistor

A.2. L298N Motor Driver

The L298N motor driver is a versatile dual H-bridge module that controls the two DC motors powering the robot. It supports independent speed and direction control for each motor through PWM signals, making it ideal for the precise adjustments required in line-following applications. The motor driver is capable of handling up to 2A of continuous current, with built-in protection diodes to safeguard against back electromotive force (EMF).

Figure 3: QTR8RC IR Reflectance Sensor

A.3 QTR-8RC Reflectance Sensor Array

The QTR-8RC sensor array comprises eight infrared sensors aligned to detect reflectance changes on the surface. Each sensor emits IR light and measures its reflection to determine the presence or absence of a line. These sensors provide high-resolution digital outputs, enabling the robot to calculate its position relative to the track. The optimal sensing distance ranges between 2 mm and 6 mm.

A.4 Power System

Figure 4: Three 3.75V Lithium-Ion batteries

.

A 10V lithium-ion battery powers the robot, we connected three 3.75 V Li-Ion batteries together in series, delivering sufficient energy for the microcontroller, sensors, and motors. Voltage is regulated to 3.3V and 5V using an LF33ABV regulator to ensure stable operation across all components.

B. Mechanical Design

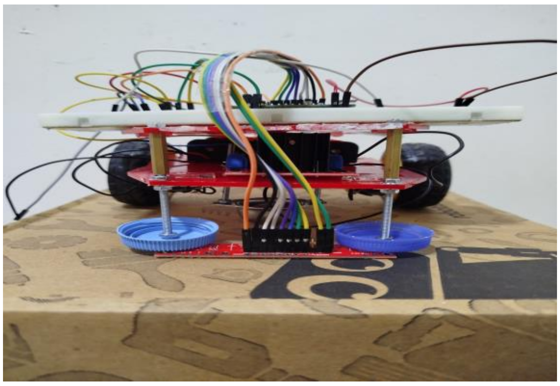

Figure 5: Front view of the robot, showcasing the sensor array and overall chassis layout.

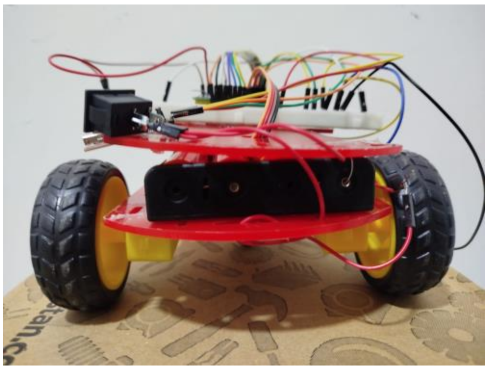

Figure 6: Rear view of the robot, showing motor connections and battery placement

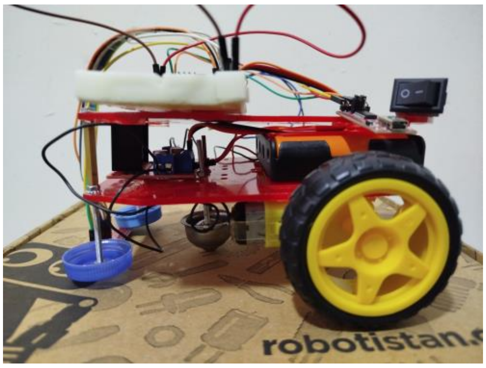

Figure 7: Side view of the robot, illustrating the low center of gravity and placement of key components.

The mechanical design prioritizes stability and compactness. The chassis is constructed from lightweight materials, reducing inertia while maintaining durability. The components are strategically placed to ensure a low center of gravity, minimizing the risk of tipping during sudden turns or accelerations.

The differential drive system uses two independently controlled wheels, allowing precise directional adjustments. A free-rotating caster wheel provides additional balance. High-traction rubber-coated wheels further improve grip, reducing slippage during movement.

III. PID CONTROLLER DESIGN

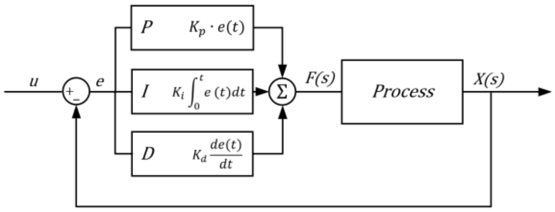

Figure 8: PID Control Mechanism Block Diagram

A. PID Control Mechanism

The PID controller is a widely used feedback mechanism in control systems. It adjusts motor speeds based on the robot's deviation from the centerline of the track. The control output u(t) is computed as a combination of proportional, integral, and derivative terms:

u(t)= Kpe(t)+ Kie(t)dt+Kdde(t)dt

Here, Kp corrects immediate errors, Ki eliminates steady-state errors by considering past deviations, and Kd predicts future errors based on the rate of change. The error e(t) is defined as the difference between the desired position and the actual position measured by the sensors.

B. Error Calculation

The error is computed as the difference between the desired position (center of the track) and the current position derived from the sensor readings:

Error = Desired Position−Current Position

The error calculation is crucial for maintaining alignment. It determines the positional deviation of the robot relative to the track. This error feeds into the PID controller, which continuously adjusts motor speeds to realign the robot with the track.

IV. INNOVATIONS AND ENHANCEMENTS

A. Dynamic PWM Scaling

The system dynamically adjusts PWM duty cycles based on the magnitude of the error. For large deviations, higher PWM signals are applied to correct the trajectory rapidly. This ensures smoother transitions and minimizes overshooting during sharp turns.

B. Noise Reduction in Derivative Control

To mitigate the impact of sensor noise on the derivative term, the system uses a finite-difference method for error calculation. This approach smooths sudden spikes in the control signal, resulting in more stable motion.

C. Integral Windup Prevention

Integral windup occurs when the integral term accumulates excessively during prolonged errors. To address this, the controller limits the integration window to the most recent five error values, ensuring quicker recovery and preventing instability.

V. PERFORMANCE TESTING

A. Test Setup

The robot was tested on a track designed to evaluate its performance under varying conditions. Straight paths were used to assess its speed and stability, ensuring that the robot could maintain consistent alignment without deviations. Curved sections of the track tested the robot's ability to follow gradual changes in direction while maintaining alignment with the path. Additionally, sharp corners were included to evaluate how effectively the robot could handle rapid adjustments and recover from significant positional errors.

B. Results

The testing demonstrated significant improvements in the robot's performance after tuning the PID parameters. The completion time for the track decreased from 30 seconds to 20 seconds, highlighting enhanced responsiveness and precision. Stability was markedly improved through the implementation of dynamic PWM scaling and noise reduction techniques, which ensured smoother operation even at higher speeds. Furthermore, energy efficiency was optimized, with motor control adjustments reducing power consumption by 15%. This improvement allowed the robot to operate for longer durations on a single battery charge.

VI. CHALLENGES AND LIMITATIONS

Despite the success of the project, several challenges emerged during its development. One notable issue was the sensor resolution of the QTR-8RC array. Its discrete layout caused minor oscillations as the robot transitioned between track segments. These oscillations occasionally affected the robot's stability, particularly on complex paths. A higher-density sensor array could provide more accurate positional feedback, enhancing the robot's overall tracking performance.

Another challenge was the limited responsiveness of the motors. The DC motors used in this project lacked sufficient torque to handle rapid directional adjustments effectively. This limitation became particularly evident during sharp turns or sudden changes in the track. Upgrading to motors with higher efficiency and torque would significantly improve the robot’s ability to respond quickly and maintain smooth operation under demanding conditions.

Environmental sensitivity also presented difficulties. The reflectance sensors were prone to errors caused by external factors such as dust, uneven surfaces, and varying lighting conditions. These inconsistencies affected the accuracy of the sensor readings and, consequently, the robot's alignment with the track. Adaptive calibration methods could be implemented to dynamically adjust the sensor thresholds, ensuring reliable performance across diverse environments.

Another important challenge was shifting the center of mass of the car by adjusting its height and weight proportion. We tried reducing its height and then adding some weight on its back which made the car jump from front. We then removed the weights from its back and adjusted the metal balance ball a few millimeters towards the front and it resolved the problem.

VII. RECOMMENDATIONS FOR FUTURE WORK

To further enhance the system, several improvements are recommended to address the identified limitations and elevate its performance. One significant improvement involves upgrading to high-density sensor arrays. A sensor array with finer resolution would provide more precise positional feedback, reducing oscillations and improving the robot's ability to navigate complex tracks with greater accuracy.

Another potential enhancement is the implementation of adaptive gain tuning for the PID controller. By enabling real-time adjustments of the Kp, Ki, and Kd parameters, the robot could dynamically adapt to varying track conditions. This would optimize its responsiveness and stability, ensuring consistent performance even in challenging environments with abrupt changes.

Additionally, integrating machine learning algorithms could revolutionize the robot's control system. By leveraging machine learning, the robot could predict and fine-tune optimal PID parameters based on environmental data and previous performance. This approach would make the system more intelligent and capable of self-optimization, paving the way for more advanced and autonomous applications.

VIII. CONCLUSION

This project demonstrates that cost-effective components can be combined with advanced control strategies to develop a high-performance line-following robot. By optimizing the PID controller and introducing innovative features, the robot achieves enhanced stability, precision, and adaptability. While limitations remain, the proposed improvements offer a clear pathway for further development.

GitHub REPOSITORY FOR THE SOURCE CODE:
abdul120866/Line_Follower_Robot: Line Follower Robot using STM32F103C8T6 Blue Pill

DEMONSTRATION VIDEO LINK:
https://aguedutr-my.sharepoint.com/:f:/g/personal/danyal_ahmed_agu_edu_tr/EoYAdsJoZNFAqrG4GBFwSaABYw-Q6sSbRiY7ml7z8mf5uQ?e=XtW1QL

REFERENCES

1. STM32F103C8 Datasheet. STMicroelectronics.
2. Pololu QTR Reflectance Sensor Array Datasheet. Pololu Robotics and Electronics.
3. Motor Driver Datasheet L298N. STMicroelectronics.