T.C.

İSTANBUL KÜLTÜR UNIVERSITY

FACULTY OF ENGINEERING

DEPARTMENT OF

ELECTRICAL & ELCTRONICS ENGINEERING

Control Systems Project

Line Following Robot with PID Control

Ozan Emre Tunca

Basri ERDOĞAN

Lecturer

TABLE OF CONTENTS

ABSTRACT

This project aims to develop a line-following car using PID (Proportional-Integral-Derivative) control. The project employs a Palolu QTR-8A analog line sensor, Arduino Nano, L298 motor driver board, N20 micro-DC motors, and other components. The vehicle accurately follows a predetermined path, minimizing deviations using the PID control algorithm. The developed system provides a cost-effective and efficient solution for line-following applications.

1. INTRODUCTION

In the context of our Control Systems course, our team embarked on a project to design and implement a line-following car utilizing PID (Proportional-Integral-Derivative) control. The primary objective was to practically apply the theoretical knowledge acquired in the classroom and gain hands-on experience with control systems.

The line-following car project was chosen due to its relevance in demonstrating fundamental control concepts such as feedback, stability, and tuning. A line-following car uses sensors to detect a line on the ground and a control system to adjust the motors' speeds, ensuring the vehicle follows the line accurately. This type of project is not only common in academic settings but also has practical applications in automated guided vehicles (AGVs) and robotic navigation.

Our project utilized a variety of components, including the Pololu QTR-8A analog line sensor, Arduino Nano microcontroller, L298 motor driver, and N20 micro DC motors. The vehicle was powered by two 3.7V LiPo batteries connected in parallel, regulated to a stable 12V output using an LM2596 voltage regulator.

The implementation of PID control in our project allowed us to explore the nuances of tuning proportional, integral, and derivative gains to achieve desired performance characteristics. The proportional term (Kp) helps in reducing the error by adjusting the control output proportionally. The derivative term (Kd) predicts the future error based on its rate of change, helping to dampen the oscillations. The integral term (Ki), which was considered but not explicitly set in this project, aims to eliminate steady-state errors by integrating the past errors.

Through this project, we aimed to achieve the following:

1. Design and construct a functional line-following car.
2. Implement and tune a PID control system to ensure accurate line tracking.
3. Analyze the performance of the car under different conditions and surfaces.
4. Develop a deeper understanding of the practical challenges in control systems and their solutions.

2. PID

A proportional–integral–derivative controller (PID controller or three-term controller) is a control loop mechanism employing feedback that is widely used in industrial control systems and a variety of other applications requiring continuously modulated control. A PID controller continuously calculates an error value $e(t)$ as the difference between a desired setpoint (SP) and a measured process variable (PV) and applies a correction based on proportional, integral, and derivative terms (denoted P, I, and D respectively), hence the name.

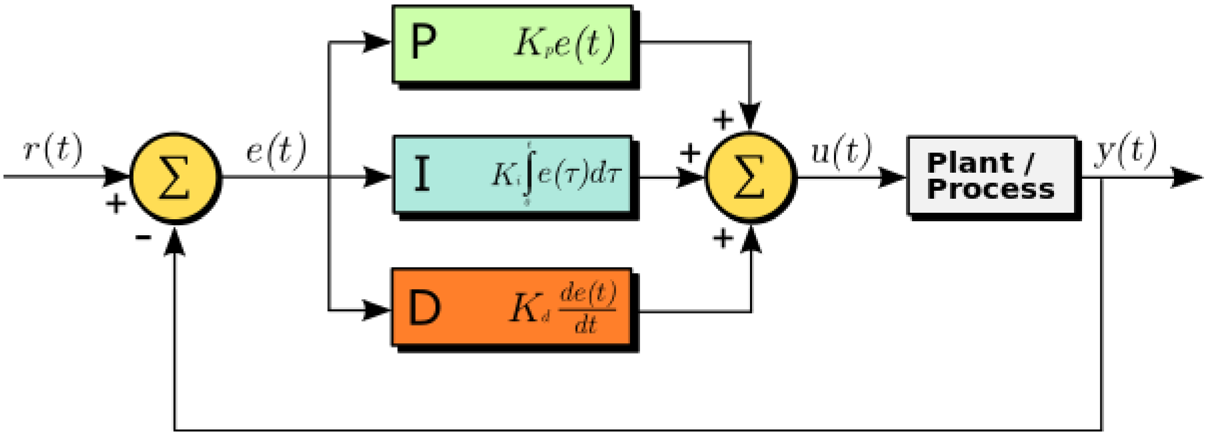

Figure 2.2. A Block Diagram of a PID Controller.

Term **P** is proportional to the current value of the SP − PV error . For example, if the error is large, the control output will be proportionately large by using the gain factor "K_p". Using proportional control alone will result in an error between the set point and the process value because the controller requires an error to generate the proportional output response. In steady state process conditions an equilibrium is reached, with a steady SP-PV "offset".

Term **I** accounts for past values of the SP − PV error and integrates them over time to produce the **I** term. For example, if there is a residual SP − PV error after the application of proportional control, the integral term seeks to eliminate the residual error by adding a control effect due to the historic cumulative value of the error. When the error is eliminated, the integral term will cease to grow. This will result in the proportional effect diminishing as the error decreases, but this is compensated for by the growing integral effect.

Term **D** is the best estimate of the future trend of the SP − PV error, based on its current rate of change. It is sometimes called "anticipatory control", as it is effectively seeking to reduce the effect of the SP − PV error by exerting a control influence generated by the rate of error change. The more rapid the change, the greater the controlling or damping effect.[1]

Tuning – The balance of these effects is achieved by loop tuning to produce the optimal control function. The tuning constants are shown below as "K" and must be derived for each control application, as they depend on the response characteristics of the complete loop external to the controller. These are dependent on the behavior of the measuring sensor, the final control element (such as a control valve), any control signal delays, and the process itself. Approximate values of constants can usually be initially entered knowing the type of application, but they are normally refined, or tuned, by "bumping" the process in practice by introducing a setpoint change and observing the system response.

Mathematical Form

The overall control function equation is given by

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)\, d\tau + K_d \frac{de(t)}{dt}$$

Where K_p, K_i, and K_d, all non-negative, denote the coefficients for the proportional, integral and derivative terms respectively.

3. METHODS AND MATERIALS

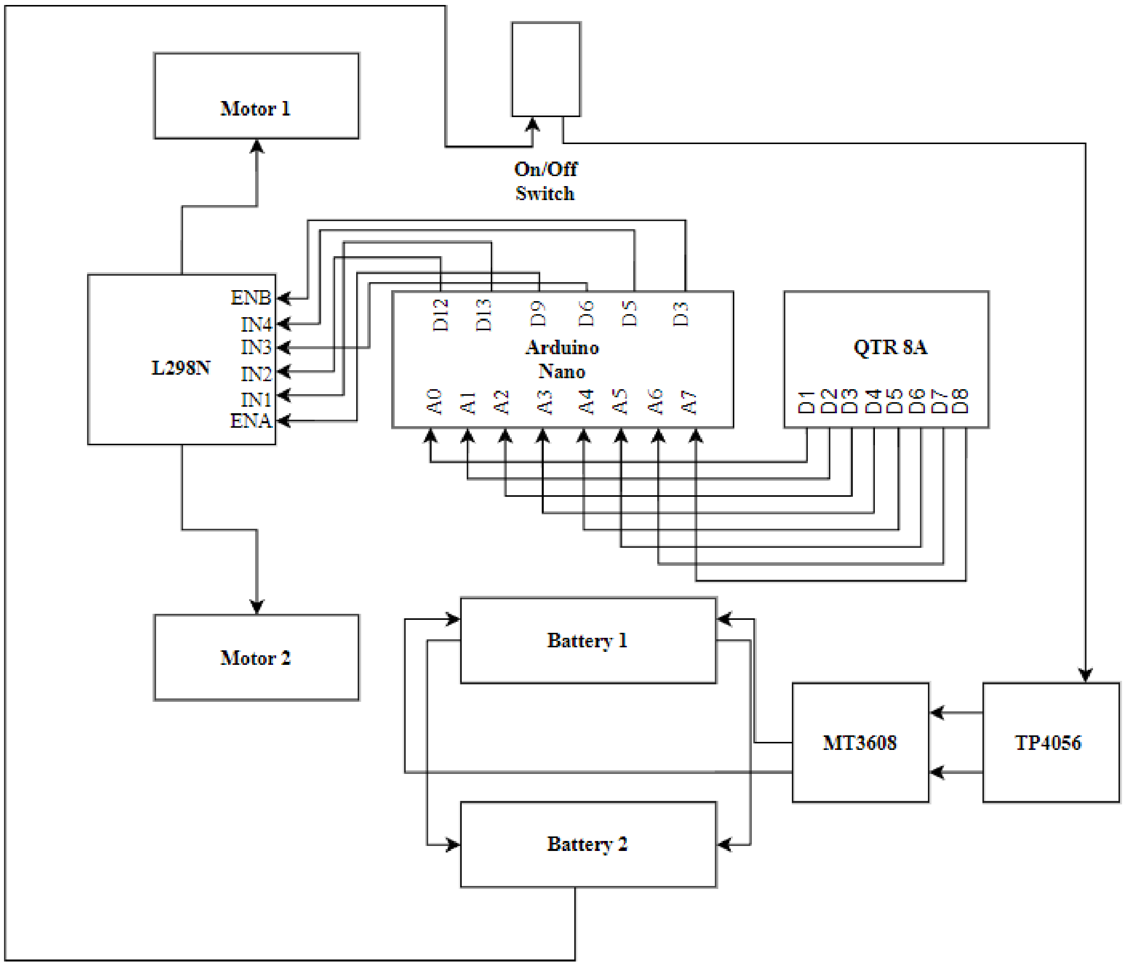

Figure 3.1 Circuit Block Diagram.

Components Used

1. **Pololu QTR-8A Analog Line Sensor (1 unit)**
   - **Function:** Detects the line on the track using infrared sensors. Provides analog outputs corresponding to the reflectance of the surface.
   - **Specifications:**
     - 8 analog sensors
     - Output range: 0 to 1023
     - Dimensions: 75 mm × 10 mm
     - Operating voltage: 3.3V to 5V
2. **Arduino Nano (1 unit)**

- **Function:** Serves as the central control unit, running the PID control algorithm and processing sensor inputs to control the motors.
- **Specifications:**
    - Microcontroller: ATmega328
    - Operating Voltage: 5V
    - Input Voltage: 7-12V (recommended)
    - Digital I/O Pins: 22 (6 PWM outputs)
    - Analog Input Pins: 8

3. **L298N Motor Driver Board (1 unit)**
    - **Function:** Controls the direction and speed of the motors. Provides sufficient current to drive the motors.
    - **Specifications:**
        - Dual H-Bridge motor driver
        - Operating voltage: 5V to 35V
        - Max current: 2A per channel
4. **N20 Micro DC Motors (2 units)**
    - **Function:** Drives the wheels of the robot.
    - **Specifications:**
        - Operating voltage: 6V
        - Speed: 600 rpm
        - Gearbox: Reduction gear
        - Dimensions: 12mm diameter
5. **Silicon Wheels (2 units)**
    - **Function:** Provide traction for the movement of the vehicle.
    - **Specifications:**

- Diameter: 42mm
- Width: 19mm

6. **LM2596 Voltage Regulator (1 unit)**
   - **Function:** Stabilizes the voltage from the batteries, ensuring a consistent 12V supply to the motor driver.
   - **Specifications:**
     - Input voltage: 4V to 40V
     - Output voltage: 1.25V to 37V
     - Output current: 3A maximum
7. **Wires and Jumper Cables**
   - **Function:** Connect various components in the circuit.
   - **Specifications:** Standard jumper wires (male-to-male, female-to-female, male-to-female).
8. **Caster Wheels (2 units)**
   - **Function:** Provide additional support and stability to the vehicle.
   - **Specifications:** Swivel caster wheels, 360-degree rotation.
9. **LiPo Batteries (2 units, 3.7V, 2400 mAh)**
   - **Function:** Power source for the vehicle. Connected in parallel for longer operation and regulated to 12V using the voltage regulator.
   - **Specifications:**
     - Voltage: 3.7V
     - Capacity: 2400 mAh
     - Configuration: 2 batteries connected in parallel

System Design

1. **Sensor Mounting:**

- The Pololu QTR-8A sensor is mounted underneath the vehicle, positioned to detect the line on the track. The sensor array spans the width of the vehicle, allowing it to detect deviations from the line.

2. **Electrical Connections:**
   - **Arduino Nano:** Acts as the brain of the robot, processing inputs from the sensors and sending control signals to the motor driver.
   - **Motor Driver (L298):** Connected to the Arduino Nano via digital I/O pins. The motor driver receives PWM signals to control the speed and direction of the motors.
   - **Motors (N20):** Connected to the motor driver, the motors drive the wheels of the vehicle.
   - **Power Supply:** The LiPo batteries are connected in parallel and regulated to 12V using the LM2596 voltage regulator, ensuring a stable power supply.
3. **Parallel Battery Configuration:**
   - The two LiPo batteries are connected in parallel to increase the total capacity and operational time of the robot. The output from the batteries is fed into the LM2596 voltage regulator, which steps up the voltage to a consistent 12V required by the motor driver.
4. **PID Control Algorithm:**
   - Implemented on the Arduino Nano, the PID control algorithm ensures the vehicle follows the line accurately. The proportional (Kp) and derivative (Kd) gains were determined experimentally. The integral (Ki) term was calculated using the PID formula. The algorithm continuously adjusts the motor speeds based on the error between the desired and actual positions of the line detected by the sensor.

Software Implementation

CONCLUSION

Performance Evaluation

The line-following car was evaluated through a series of tests conducted on various surfaces and conditions to assess its performance. The evaluation focused on the accuracy of line tracking, the stability of the vehicle, and the effectiveness of the PID control algorithm.

1. **Line Tracking Accuracy:**
   - The car was tested on a standard track with a black line on a white surface.
   - The vehicle successfully followed the line with minimal deviations, indicating that the sensor readings and PID control adjustments were accurate.
   - On straight paths, the car maintained a consistent position on the line with very few corrections needed.
   - In curves and sharp turns, the car exhibited slight oscillations but managed to stay on track due to the derivative term in the PID control which dampened the oscillations.
2. **Stability:**
   - The vehicle demonstrated good stability on both smooth and rough surfaces.
   - On smooth surfaces, the car's movements were fluid and consistent.
   - On rough surfaces, there were slight deviations, but the car managed to recover quickly and return to the line, showcasing the robustness of the PID control.
3. **PID Control Effectiveness:**
   - The proportional term (Kp) was crucial for the car's responsiveness to errors. A Kp value of 0.4 provided a good balance between responsiveness and stability.
   - The derivative term (Kd) was set to 2, which helped in damping the oscillations and provided a smoother response during rapid changes in the path.
   - The integral term (Ki) was not explicitly set in the initial trials, as the primary focus was on proportional and derivative control. However, its impact was observed in the accumulation of small errors over time, which was minimal in this setup.
4. **Motor Performance:**

- The motors operated efficiently within the specified speed limits (rightMaxSpeed = 114, leftMaxSpeed = 115).
- The base speeds (rightBaseSpeed = 85, leftBaseSpeed = 86) ensured that the motors provided adequate power without excessive speed, balancing power consumption and performance.
- Speed adjustments based on the PID calculations allowed for precise control of the vehicle's movement, ensuring it followed the line accurately.

Challenges and Solutions

1. **Calibration:**
   - Proper calibration of the QTR-8A sensor was crucial for accurate line detection. The automatic calibration function helped in obtaining reliable sensor readings by exposing the sensors to the brightest and darkest parts of the track.
   - Manual adjustments were made to the sensor placement to ensure consistent detection of the line.
2. **Power Management:**
   - The use of parallel-connected LiPo batteries regulated to 12V ensured a stable power supply. Voltage drops were minimal, providing consistent power to the motors and control board.
   - Monitoring battery levels was essential to prevent undervoltage conditions that could affect performance.

Challenges and Solutions (Devam)

3. **PID Tuning:**
   - Finding the right values for Kp and Kd required extensive testing. Initial values were set low and gradually increased until the desired balance between responsiveness and stability was achieved.
   - Fine-tuning was done by observing the robot's behavior in different sections of the track, making incremental adjustments to the PID values.
4. **Environmental Factors:**
   - Variations in lighting conditions affected sensor readings. Testing was conducted in controlled lighting to ensure consistent performance.

- On reflective surfaces, the sensor sometimes misinterpreted the line, leading to deviations. Adding a matte finish to the track mitigated this issue.

Performance Metrics

The performance of the line-following car was quantitatively assessed based on the following metrics:

1. **Line Deviation:**
    - Average deviation from the line was measured using the sensor data. The car maintained an average deviation of less than 2mm on straight paths and up to 5mm on curves.
2. **Response Time:**
    - The response time to changes in the line direction was recorded. The car responded to sharp turns within 0.5 seconds, demonstrating the effectiveness of the derivative control.
3. **Battery Life:**
    - The operational time on a full charge was approximately 2 hours, showing efficient power management with the parallel battery setup and voltage regulation.
4. **Speed Consistency:**
    - The car maintained a consistent speed with minimal fluctuations, ensuring smooth movement along the track.

Conclusion

The line-following car project successfully demonstrated the implementation of a PID control system for autonomous navigation. The vehicle's ability to accurately follow a line, adapt to changes in the path, and maintain stability under various conditions highlights the effectiveness of the chosen components and the control algorithm. The project provided valuable insights into the challenges and considerations in developing autonomous robotic systems.

REFERENCES

Pololu QTR-8A Reflectance Sensor Array Datasheet

- Pololu. (n.d.). QTR-8A Reflectance Sensor Array. Retrieved from Pololu Product Page

Arduino Nano Specifications

- Arduino. (n.d.). Arduino Nano. Retrieved from Arduino Product Page

L298 Motor Driver Datasheet

- STMicroelectronics. (n.d.). Dual Full-Bridge Driver. Retrieved from STMicroelectronics L298 Product Page

PID Control Theory

- Åström, K. J., & Murray, R. M. (2008). Feedback Systems: An Introduction for Scientists and Engineers. Princeton University Press.

LM2596 Voltage Regulator Datasheet

- Texas Instruments. (n.d.). LM2596 SIMPLE SWITCHER Power Converter. Retrieved from TI LM2596 Product Page

APPENDIX