Design and Implementation of Closed-Loop Tracking Vehicle with Discrete H-Bridge Driver

Author: Annie Huang (National Taipei University of Technology)

Date: January 2026

Abstract

This project presents the design of an autonomous tracking vehicle featuring a **custom discrete H-Bridge driver**. Departing from conventional module-based solutions (e.g., L298N), this system implements a mixed-transistor driver circuit using **S8550 (PNP)** and **C1384 (NPN)** BJTs to achieve physical isolation between control logic and power actuation. The control strategy employs a **Bang-Bang controller with deadband**, integrated with a **median filter** for robust signal processing. This report details the hardware architecture, control algorithms, and a comprehensive failure analysis case study.

Discrete Driver Circuit Design

The core innovation of this project is the "First Principles" approach to motor driving. The custom H-Bridge circuit is designed to handle high current loads while protecting the microcontroller.

- Speed Control (Low-side Switching): The PWM signal from the Arduino drives the base of a C1384 (NPN) power transistor. This transistor modulates the connection to the ground, effectively controlling the average voltage across the motor and thus its speed.
- Direction Control (High-side Switching): A DPDT Relay is used to reverse the polarity of the motor. Since the Arduino GPIO cannot supply sufficient current for the relay coil, an S8550 (PNP) transistor is utilized as a driver switch.
- Protection Mechanism: 1N4001 Flyback Diodes are installed in parallel with the relay coils to suppress inductive voltage spikes (Back-EMF) during switching, preventing damage to the transistors and the MCU.

(Note: Please refer to the schematic diagram in the repository for circuit details.)

Engineering Failure Analysis & Troubleshooting

During the development phase, several critical issues were encountered. The following engineering approach was adopted: **Identify Issue → Analyze Root Cause → Implement Solution**.

Issue	Root Cause Analysis	Engineering Solution
Sensor Cross-talk	Simultaneous triggering of left and right HC-SR04 sensors caused echo interference.	Time-Division Multiplexing: Introduced a 15ms delay between left and right sensor readings to isolate signals.
Signal Jitter	Low-cost sensors produced outliers due to environmental noise.	Median Filter: Implemented a 5-sample median filter to statistically eliminate spikes before control logic processing.
Right Turn Instability	The vehicle hesitated during right turns. Swap testing confirmed the issue followed a specific sensor.	Physical Masking: Identified a wide beam angle defect in one sensor. Applied a physical mask to the sensor emitter to narrow the detection cone.
Oscillation in Tracking	The vehicle "snaked" (oscillated L/R) while tracking a straight line due to over-sensitive steering logic.	Differential Fine-Tuning: Modified logic to maintain relay polarity (forward) but adjust L/R PWM speeds for minor corrections, rather than triggering a full turn.
Power Drop	Motor startup current caused voltage sags affecting the MCU.	Power Isolation: Separated logic and motor power rails and added bulk capacitance to stabilize the voltage bus.

Control Logic: Bang-Bang with Deadband

To prevent mechanical wear on the relays and ensure smooth operation, a hierarchical state machine is implemented:

1. Priority 1: Orientation Correction: If the difference between Left/Right sensors > 10cm (Turn Gap), execute a Spot Turn.

2. Priority 2: Distance Control: If the average distance is outside the 8cm ± 3cm target range:

    - **> 11cm:** Move Forward (with differential correction).

    - **< 5cm:** Move Backward.

3. Priority 3: Deadband (Stop): If the distance is within 8cm ± 3cm, cut power to motors to prevent hunting/chattering around the setpoint.