UNIVERSITY of DENVER

Engineering Integration I

Student : Ali Behbehani

Course : Engineering Integration I

Section : Friday 1:00 - 3:50

Lab 03: Integrating Mechanical, Electrical, and Computing Components

---

Documentation Statement

I certify that this lab report represents my own work and that all procedures were completed according to the lab instructions and academic integrity guidelines.

Introduction

The objective of this laboratory assignment was the successful integration of mechanical, electrical, and computing components to construct a functional Two-Wheel Drive (2WD) vehicle. In this experiment, a Teensy 4.1 microcontroller was used with an L298 Motor Driver Module, a high-power electrical distribution system was created, and the frame was assembled. Understanding differential drive system mechanics, applying pulse width modulation (PWM) for open-loop control, and examining how wheel geometry affects vehicle maneuverability were among the main learning objectives.

PRE-LAB

There was no pre-lab for lab03

Experiment 1 - Soldering

Objective

The point of this activity was to get some hands-on experience soldering wires to the motor terminals so the electrical connections would be solid and wouldn't vibrate loose.

Procedure

I cut the wires to about 10 inches and stripped the ends . I then threaded the wire through the little holes on the motor tabs, made a hook, and soldered them in place . I made sure the solder flowed nicely over the whole connection so it was strong.

Results

After I finished, I showed my work to the TA, and they signed off on it. The connections were clean and held up perfectly during the rest of the lab.

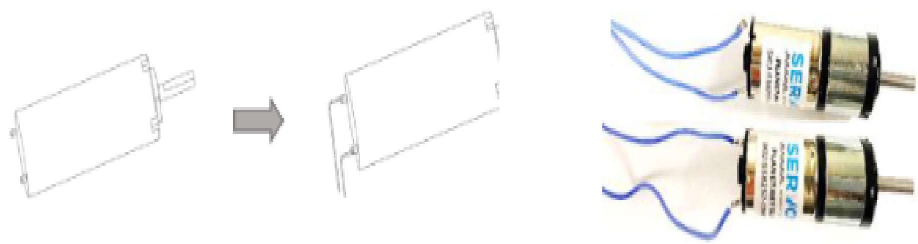

Figure 1: Soldered solid core wire connections to the DC motor terminals.

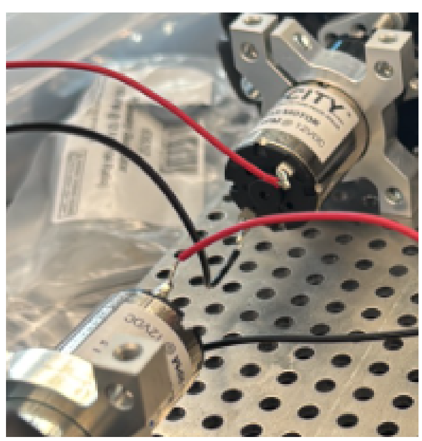

Figure 2: Soldered solid core wire connections (In Lab)

Experiment 2 - Vehicle Assembly and Wiring

Objective

The goal here was to actually build the car and get all the electronics wired up so the Teensy could control the motors.

Procedure

1. I started by clamping the DC motors into their mounts and bolting them onto the metal grid plate .

2. I added the ball caster to the front of the plate so the car wouldn't tip over .

3. I mounted the L298 motor driver in the middle. I was careful to use nylon bolts and plastic spacers because if the bottom of the driver touched the metal plate, it would have shorted out the whole system .

4. Finally, I wired the driver to the Teensy pins (9, 8, 7 for one side and 3, 4, 5 for the other) and connected the 12V battery .

Results

The car ended up feeling very solid. I kept the wiring organized so nothing would get caught in the wheels while the car was moving .

Conclusion:

Proper mechanical alignment during assembly is crucial for predictable vehicle motion. The use of spacers for the L298 module was a critical step in preventing electrical damage to the system.

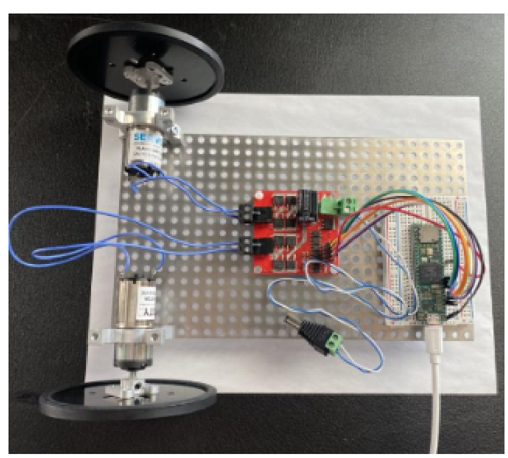

Figure 3: Fully assembled vehicle chassis featuring the grid plate, L298 motor driver, and Teensy 4.1 microcontroller.

Experiment 3 - Vehicle Testing with Disk Wheels

Objective

The aim of this experiment was divided into two parts: first, to verify that the hardware and software were integrated correctly, and second, to characterize the vehicle's forward movement to enable open-loop control .

Part 1: Vehicle Operation

Procedure

I began by putting the vehicle on the floor so the wheels could spin freely without the car driving off the table . After powering on the 12V battery, I uploaded the VehicleTestProgram to Teensy . I watched the wheel rotations carefully to see if they matched the programmed logic.

Results & Discussion

- **[E 3.1] Software Integrity:** It is essential that I didn't modify the VehicleTestProgram code. Because this script is a known "constant," it allows me to troubleshoot. If the wheels spin the wrong way, I know it's a wiring mistake on the L298 driver rather than a bug I accidentally wrote into the code .
- **[E 3.2 & 3.3] Test Observations:** Looking down from above, the vehicle successfully executed the test sequence: it moved forward, rotated right, rotated left, and then moved backward. Since it performed exactly as expected, I documented this as a "successful test" of my vehicle's construction.

Part 2: Forward Distance PWM Testing

Procedure

Once the operation was verified, I placed the vehicle on the floor to run the FwdDistanceTest. I ran the car for exactly 2 seconds at different PWM duty cycles, starting with a 35% baseline . I measured the distance traveled for each test to build a "PWM to Distance" transform table

Results & Discussion

- **[E 3.4] Straight Line Calibration** Initially, the vehicle drifted to the side due to inherent motor imbalances. I "trimmed" the PWM values in the code to correct this; for example, when running at a 20% setting, I used a PWM value of 51 for the left motor and a slightly adjusted value for the right motor to ensure a straight trajectory. After this adjustment, the car traveled as documented in Table 2.

Test #	PWM Setting (% of max)	PWM Value (int)	Distance (cm)	Speed (m/s)
Test 1	10%	25.5	0.00	0.00
Test 2	20%	51.0	2.90	1.45
Test 3	30%	76.5	6.60	3.30
Test 4	40%	102.0	8.20	4.10
Test 5	60%	153.0	9.40	4.70
Test 6	70%	178.5	10.00	5.00

- **[E 3.5] Motion Threshold / Dead-band** In Test 1, the vehicle recorded a distance of 0.00 cm at a PWM setting of 10% (value 25.5). This indicates that the electrical power was insufficient to overcome the static friction of the gearboxes and the inertia of the 2WDD chassis. Movement only began once the PWM reached the 20% (value 51.0) threshold.

- **[E 3.6] Correlation Analysis** The data demonstrates a clear positive correlation between the PWM duty cycle and distance traveled over 2 seconds. However, as shown in the Best Fit Curve, the relationship is non-linear. The distance increases rapidly between 20% and 40% PWM but begins to plateau as it approaches 70%, likely due to the mechanical limits of the DC motors or battery voltage stabilization.

- **[E 3.7] Open-Loop Control Logic** With this characterization data, I can implement open-loop control. By using the polynomial equation from Figure 9, I can predict the required PWM for a target distance. For instance, to travel 8.00 cm, I would program the vehicle to run at a PWM value of approximately 98 for exactly 2 seconds.

Conclusion

I successfully characterized the vehicle's forward movement by establishing a mathematical relationship between PWM duty cycles and travel distance. By analyzing the data in Table 2, I determined that my vehicle requires a minimum of 20% PWM to overcome static friction and begin moving. This experiment provided me with the necessary open-loop control logic to predict the PWM values required for specific navigation distances.

Experiment 4 - Vehicle Testing with Omni-Wheels

Objective

The goal was to observe how omni-wheels affect vehicle kinematics.

Procedure

I replaced the disk wheels with omni-wheels and re-ran the VehicleTestProgram.

Results & Discussion

- **[E 4.1] Observations:** The omni-wheels made rotations much smoother, but the vehicle was more prone to lateral drifting during forward movement.

- **[E 4.2] Explanation:** This occurs because the rollers on omni-wheels allow for perpendicular slip, reducing the lateral friction found in disk wheels.

- **[E 4.3] Design Theory:** To mimic disk wheels, I would need more complex control or a 4WD configuration to stabilize lateral motion.

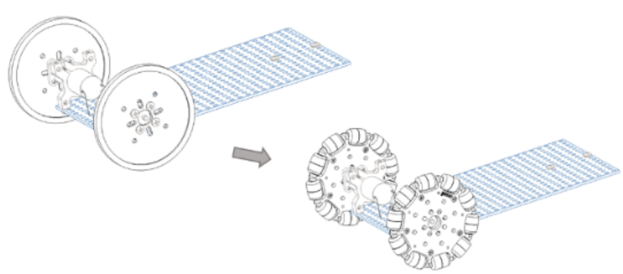

Figure 4: Comparison of standard disk wheels and the omni-wheel configuration used for kinematic testing

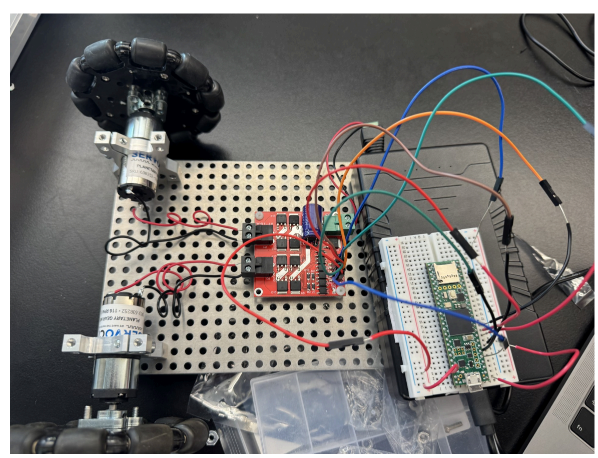

Figure 5: OMNI-wheels configured in class

.

Experiment 4 conclusion

I successfully evaluated the impact of wheel geometry on vehicle kinematics by replacing the standard disk wheels with omni-directional wheels. I observed that the omni-wheels significantly improved the vehicle's rotational agility, allowing for much smoother spins compared to the disk wheels. However, this gain in maneuverability came at the cost of straight-line stability, as the vehicle was much more prone to lateral drifting during forward movement

---

APPENDIX

```
/////////////////////////////////////////////////////////////////////////////////
// UNIVERSITY OF DENVER: ENGR 2610
// Lab-03: Forward Distance Characterization Program

// Purpose: This program drives the 2WDD vehicle forward for a fixed 2.00 second
// duration at a specific PWM duty cycle. This allows for open-loop calibration
// of distance vs. power settings.
/////////////////////////////////////////////////////////////////////////////////

[cite_start]// L298 Motor Controller Pin Assignments [cite: 145-153]
int enA = 9;  // Speed control for Motor A (Left)
int in1 = 8;  // Direction control for Motor A
int in2 = 7;  // Direction control for Motor A

int enB = 3;  // Speed control for Motor B (Right)
int in3 = 4;  // Direction control for Motor B
int in4 = 5;  // Direction control for Motor B

void setup() {
  // Initialize pins as outputs
  pinMode(enA, OUTPUT); pinMode(enB, OUTPUT);
  pinMode(in1, OUTPUT); pinMode(in2, OUTPUT);
  pinMode(in3, OUTPUT); pinMode(in4, OUTPUT);

  // Default state: Motors Off
  digitalWrite(in1, LOW); digitalWrite(in2, LOW);
  digitalWrite(in3, LOW); digitalWrite(in4, LOW);
}

void loop() {
  // Apply PWM duty cycle
  analogWrite(enA, 89);
  analogWrite(enB, 86);

  // Set direction to Forward
  digitalWrite(in1, HIGH); digitalWrite(in2, LOW);
  digitalWrite(in3, LOW);  digitalWrite(in4, HIGH);

  delay(2000); // 2.00 second test interval

  // Emergency Stop
  analogWrite(enA, 0); analogWrite(enB, 0);
  while(1);
}


  // Set direction to Forward
  digitalWrite(in1, HIGH); digitalWrite(in2, LOW);
  digitalWrite(in3, LOW);  digitalWrite(in4, HIGH);

  delay(2000); // 2.00 second test interval

  // Emergency Stop / Completion Halt
  analogWrite(enA, 0); analogWrite(enB, 0);
  while(1); // Stop execution to prevent repeat loops
}
```

Final Conclusion

This lab demonstrated how to integrate hardware and software for mobile robotics. I learned that hardware characterization is vital for open-loop control and that wheel geometry significantly impacts vehicle handling. Minor errors may have occurred due to floor surface friction variations and battery voltage fluctuations during testing.

SIGN OFF SHEET

ENGR 2610 – Integration I   Lab-03 – 2WDD Vehicle

Print Name (first last): Ali Behbehani

Date: 30 January   Lab Section Day: Friday   Lab Section Time: 1-3:50

Integration I: Lab-03

Integrating Mechanical, Electrical, and Computing Components – Building a Small 2WD Vehicle

- Each student is **individually responsible** for the work on each lab assignment.
- Review course Lecture Notes, then **carefully** read through the lab assignment.
- Follow the lab instructions **step-by-step**. Use the Lecture Notes as a reference.
- If you have questions, ask your Instructor, TA, or collaborate with your classmates.

Performance Sign-off

______ (Optional) **Experiment 1: Soldering**

Comments:

______ **Experiment 2: 2WD Vehicle Assembly and Wiring**

Comments:

______ **Experiment 3: Vehicle Testing with Disk Wheels**

______ **Part 1 – Vehicle Operation Test (move forward, rotate right, rotate left, move backward)**

______ **Part 2 – Vehicle Forward Movement Testing**

Comments:

______ **(Extra Credit) Characterizing Vehicle Rotation Capability**

______ **Experiment 4: Vehicle Testing with Omni-Wheels**

Comments: