Robotics Laboratory 4

KINEMATICS AND DIFFERENTIAL MOTION FOR MOBILE ROBOTS

Seth Paul C. Babayson
BSECE-4A
College of Engineering
Samar State University

***Abstract*—This experiment focused on testing a robot's ability to travel exactly 1 meter and turn 10°. Using a 50% PWM signal, the robot achieved a mean distance of 0.9703 meters, with a distance error of only 2.97%, meeting the objective of staying within a 5% margin. A recalculation for the 10° turn was necessary as the initial speed was too low, leading to motor instability. After adjusting the PWM and reducing the turn time, the robot successfully completed the 10° turn. The findings demonstrate that real-world adjustments are crucial for achieving precise robotic movements.**

I. RATIONALE

Understanding how mobile robots move and turn is important for building smart and automatic robots. This experiment teaches the basics of how differential drive robots move, which is a common and simple way to control a robot direction. By programming the robot to move straight and turn accurately, I get practical experience using what I have learned about motion control, sensor feedback, and computer simulation.

Using the Webots simulation software, it tests and improves the program in a safe and virtual environment. This helps them see what works and what doesn't when controlling robot movement. Trying to keep the error in distance below 5% and position error within 5 cm and 10 degrees shows how important accuracy is in real robotic systems.

This experiment helps fill a gap between learning theory and making robots work in the real world. As Siegwart and others (2011) explained, learning how control code affects real robot behavior is a key step toward building reliable, autonomous systems. Using sensor data to fix movement errors also matches what researchers are doing in making robots that can move on their own in complex environments (Thrun et al., 2005).

II. OBJECTIVES

A. Program differential drive kinematics to move a robot in different directions, achieving a position error within 5 cm in linear travel and 10° for turns.

This objective focuses on writing a code that let a robot move forward, backward, and turn accurately using two motors. The goal is to make sure the robot doesn't miss its target position by more than 5 cm or turn more than 10 degrees off from the desired angle.

Identify applicable funding agency here. If none, delete this.

B. Integrate wheel encoders to achieve precise control of movement, with a distance error less than 5% over 1 meter.

This portion uses sensors on the wheels (encoders) to measure how far the robot moves. The aim is to keep the robot's distance error small—no more than 5 cm off when it tries to travel 1 meter.

C. Use Webots to simulate and analyze robot motion, ensuring 90% accuracy in path tracking.

To test the robot in the Webots simulation and check if it follows the planned path correctly. The goal is for the robot to stay on track at least 90% of the time during the simulation.

III. MATERIALS AND SOFTWARE

- Materials:
  - Arduino Uno R3
  - DC Motor
  - Wheel Encoder(Alternative: Hall effect and magnet
  - L298N Dual H-Bridge Motor Driver Module
  - Breadboard and Jumper Wires
  - 3x 3.7V Li-ion Battery
- Software:
  - Arduino IDE
  - Webots Simulator

IV. PROCEDURES

TABLE I
MOTOR DRIVERS/ULTRASONIC SENSOR/SERVO MOTOR TO ARDUINO UNO CONNECTIONS

Components Pin Connections	
Motor Drivers/Ultrasonic Sensor/Servo Motor Pins	***Arduino UNO Pins***
ENA	D5
ENB	D3
IN2/IN1S (FRONT LEFT)	D4
IN4/IN3 (FRONT RIGHT)	D12
IN1/IN2 (REAR LEFT)	D8
IN3/IN4 (REAR RIGHT)	D7
Wheel Encoder(A3144 Hall Effect)	D2

- Motor Driver Setup
  - For the supply, connect VCC and GND to 5V and GND respectively of Arduino Uno. Connect logic pin to arduino digital pins. And also the ENA and

ENB pin to the PWM pins to vary the speed of the motor.

- Setting hall effect IC with magnet as Wheel encoder setup:
  - Connect hall effect output pin to arduino's digital pin D2 with 10k pull-up resistor.
  - Since there is only one magnet attached to a motor, this should measure the rotation per second to determine the motor's speed and distance traveled. By using Linear Speed and Angular Speed formula. (See Data Analysis for the calculations.)
- Webots Simulation
  - Simulate and observe the behavior of the tractor based on the adjustments of its speed.

V. Observations and Data Collection

Entry	RPS	Speed (m/s)
1	3.00	0.471
2	2.00	0.314
3	3.00	0.471
4	2.00	0.314
5	3.00	0.471
6	3.00	0.471
7	2.00	0.314
8	3.00	0.471
9	2.00	0.314
10	3.00	0.471
11	2.00	0.314
12	3.00	0.471
13	2.00	0.314
14	3.00	0.471
15	2.00	0.314
16	3.00	0.471
17	2.00	0.314
18	2.00	0.314
19	2.00	0.314
20	2.00	0.314
21	3.00	0.471
22	2.00	0.314
23	2.00	0.314
24	2.00	0.314
25	2.00	0.314
26	3.00	0.471
27	2.00	0.314
28	2.00	0.314
29	2.00	0.314
30	2.00	0.314
31	2.00	0.314
32	3.00	0.471
33	2.00	0.314
34	2.00	0.314
35	2.00	0.314
36	2.00	0.314
37	3.00	0.471
38	2.00	0.314
39	2.00	0.314
40	2.00	0.314
41	2.00	0.314
42	3.00	0.471
43	2.00	0.314
Mean Average Speed		**0.365 m/s**

TABLE II

SPEED READINGS AND RPS VALUES RECORDED AT 50% PWM.

During testing of the motor's speed using hall effect sensor as wheel encoder, speed and rotation per second (RPS) values were recorded from the DC motors set at 50% PWM duty cycle. These values are presented in the Table above, which displays the RPS and the corresponding calculated speed in meters per second for each entry. A total of 43 measurements were taken, alternating between 2.00 RPS (0.314 m/s) and 3.00 RPS (0.471 m/s). The calculated mean average speed from all recorded values is approximately 0.365 m/s.

In actual testing, the robot's distance traveled was 1 meter and has plus and minus 0.

Using this average speed, I computed the time required to travel a distance of one meter with the formula $t = \frac{d}{s}$, where $d = 1$ meter and $s = 0.365$ m/s. Substituting the values gives $t = \frac{1}{0.365} \approx 2.74$ seconds. Therefore, the motors should run for approximately 2.74 seconds to move the robot forward by one meter.

It has compute the necessary wheel speed for a 10° in-place turn. Given a wheel radius of 25 mm (0.025 m), a wheelbase of 127 mm (0.127 m), and a desired turn angle of $\theta = 10^\circ \approx 0.1745$ radians, I assumed a time duration of 1 second for the maneuver. The required angular velocity was calculated using $\omega = \frac{\theta}{t} = 0.1745$ rad/s. For an in-place rotation where the wheels turn in opposite directions, the linear speed of each wheel is derived from the equation $\omega = \frac{v_R - v_L}{L} = \frac{2v}{L}$, giving $v = \frac{0.1745 \cdot 0.127}{2} \approx 0.0111$ m/s.

However, during the actual test, the motors didn't work smoothly at this calculated speed. The motors didn't have enough power to move the robot properly, and the motion was not stable. This is because at very low speeds, DC motors don't get enough power to rotate, which can cause them to struggle.

After recalculating the speed for a 10° turn in just 200 milliseconds, I found that the robot would need to rotate faster, with an angular speed of about $\omega = 0.8725 rad/s$. This means each wheel would have to move at a speed of around $0.0554 m/s$ in opposite directions to complete the turn within the given time. (See calculations in the Data Analysis)

VI. Data Analysis

Time Calculation to Travel 1 Meter

To determine the time required to travel 1meter at a speed of 0.365meter/second, I use the formula:

$$t = \frac{d}{s}$$

Substituting the values:

$$t = \frac{1}{0.365} \approx 2.74 second$$

Theoretically, the DC motors should run for approximately 2.74 seconds to move the robot forward by 1 meter at the given speed.

In this section of the experiment, the goal was to evaluate the robot's ability to travel exactly 1 meter in a straight line using differential drive control. The PWM signal applied to the motors was set to 50%, and based on earlier measurements, this corresponded to an average linear speed of approximately 0.365 m/s. Using the formula $t = \frac{d}{s}$, the calculated time required to move the robot forward by 1 meter was determined to be approximately 2.74 seconds. This delay time was programmed into the system for each trial run.

A total of 20 trials were conducted, and the distance traveled by the robot was recorded in inches and then converted to meters. The values in a table was used to compute the mean distance traveled.

TABLE III
DISTANCE TRAVELED BY ROBOT (CONVERTED FROM INCHES TO METERS)

Sample No.	Distance (in)	Distance (m)
1	33	0.8382
2	37	0.9398
3	38	0.9652
4	36	0.9144
5	36	0.9144
6	38	0.9652
7	39	0.9906
8	39	0.9906
9	39	0.9906
10	40	1.0160
11	38	0.9652
12	41	1.0414
13	39	0.9906
14	38	0.9652
15	39	0.9906
16	41	1.0414
17	38	0.9652
18	39	0.9906
19	38	0.9652
20	38	0.9652
Mean Average		**0.9703 m**

- **Target Distance:** 1.0000 meters
- **Average Distance from 20 Trials:** 0.9703 meters
- **Distance Error:** $1.0000 - 0.9703 = 0.0297$ meters
- **Percentage Error:** $\frac{0.0297}{1.0000} \times 100 \approx \mathbf{2.97\%}$

According to the objective of the experiment, the robot must achieve a distance error of less than **5%** over a 1-meter travel. The calculated percentage error of **2.97%**. This confirms that the robot meets the objectives. With the motor calibration at 50% PWM, it provides sufficient speed and control for the robot to move forward.

Calculating Wheel Speed for a 10° Turn

Given:

- Wheel radius, $R = 25mm = 0.025m$
- Wheelbase, $L = 127mm = 0.127m$
- Desired rotation angle: $\theta = 10^\circ = \frac{10\pi}{180} \approx 0.1745rad$
- Time to complete turn: $t = 1sec$ (assumed)

Compute Required Angular Velocity

$$\omega = \frac{\theta}{t} = \frac{0.1745}{1} = 0.1745rad/sec$$

Step 2: Derive Linear Wheel Speed For an in-place turn (left wheel backward, right wheel forward at same speed):

$$\omega = \frac{v_R - v_L}{L} = \frac{2v}{L} \quad \Rightarrow \quad v = \frac{\omega \cdot L}{2}$$

Substitute the known values:

$$v = \frac{0.1745 \cdot 0.127}{2} \approx 0.0111m/s$$

But during the actual testing, the DC motors couldn't operate smoothly with that kind of calculated value of speed. That's why the PWM value was adjusted and increased, and also the time to run at that speed decreased. If the time to complete the turn is reduced to $t = 200ms = 0.2s$, I update the angular velocity calculation as follows:

Recalculate Angular Velocity

$$\omega = \frac{\theta}{t} = \frac{0.1745}{0.2} = 0.8725rad/s$$

Recalculate Linear Wheel Speed
Using the formula:

$$v = \frac{\omega \cdot L}{2}$$

Substitute the updated values:

$$v = \frac{0.8725 \cdot 0.127}{2} \approx 0.05537m/s$$

With a time of 200ms, each wheel should move at approximately 0.0554m/s in opposite directions to achieve a 10° in-place turn.

The PWM signal applied to the motors was reset to 59% instead of 50% to attain the recalculated speed

VII. DISCUSSION AND INTERPRETATIONS

The goal of this experiment was to test how accurately the robot could move forward by 1 meter and make a small 10° turn. Based on earlier measurements, I calculated that at 50% PWM, the robot moved at an average speed of 0.365 m/s. Using this speed, as calculated, it would take around 2.74 seconds to travel 1 meter. In actual tests, the robot traveled an average distance of 0.9703 meters over 20 trials. This gave a small error of about 2.97%, which is well within the given distance error of 5%. This shows that the calculations were mostly accurate, and the robot performed well at the tested speed.

For the turning part, I calculated that the robot's wheels needed to move at about 0.0111 m/s to make a 10° turn in 1 second. However, the motors didn't work properly at such a low speed—likely because DC motors need a higher voltage to move smoothly. To fix this, I changed the time to make the turn to just 200 milliseconds, which increased the required speed to about 0.0554 m/s. I also increased the PWM from 50% to 59%. After doing this, the robot was able to complete the 10° turn more accurately and consistently.

Some differences between our calculated and actual results can be explained by a few things. First, the motors may not be perfectly identical—one might be slightly faster than the other.

The floor surface can also affect movement, and small bumps or friction can change the results. The Hall effect sensor I used for measuring speed might also have small errors. Lastly, motors don't always respond perfectly to changes in PWM, especially at low speeds.

Overall, even with these small issues, the robot met the goals of the experiment. It was able to move forward almost exactly 1 meter and make the small turn with good accuracy. The results show that using calculations for speed and turn timing is useful, but in real-world tests, adjustments are needed to match the actual performance of the motors and robot.

VIII. Conclusion

In this experiment, it is tested how well the robot could move 1 meter and turn 10°. The results were very close to what we expected. The robot only had a 2.97% error in distance, which is good since the goal was to keep it under 5%. At first, the calculated speed for the 10° turn didn't work well because the motors couldn't move the robot smoothly. After increasing the PWM and reducing the turn time, the robot was able to complete the turn as expected. This shows that while calculations help, real testing and adjustments are needed to get the best results. Overall, the experiment was successful and met its goals.

References

[1] Siegwart, R., Nourbakhsh, I. R., & Scaramuzza, D. (2011). Introduction to Autonomous Mobile Robots. MIT Press.

[2] Thrun, S., Burgard, W., & Fox, D. (2005). Probabilistic Robotics. MIT Press.

IX. Appendix

https://github.com/seth-paul/Elective-2-Robotics-Technology/tree/main/Laboratory%204