LABORATORY 4

KINEMATICS AND DIFFERENTIAL MOTION FOR MOBILE ROBOTS

Ma. Gellan A. Caples
College of Engineering
Bachelor of Science in Electronics Engineering
Samar State University
magellancaples@gmail.com

***Abstract*—This experiment focuses on exploring kinematics and differential motion in mobile robotic systems. Using Arduino, a differential drive robot was programmed to move precisely by calculating and setting appropriate wheel velocities. Wheel encoders were incorporated to provide feedback, enhancing the accuracy of the robot's movement. The system's performance was evaluated through Webots simulation, confirming minimal positional and angular errors during navigation tasks.**

***Index Terms*—Differential Drive, Kinematics, Wheel Encoders, Mobile Robots, Path Tracking.**

I. Rationale

This experiment introduces the kinematics of mobile robots, particularly those with differential drive systems. Students will use this understanding to program the robot's movement.

II. Objectives

- Program differential drive kinematics to move a robot in different directions, achieving a position error within 5 cm for linear travel and within 10° for turns.
- Integrate wheel encoders to achieve precise control of movement, with a distance error less than 5% over 1 meter.
- Use Webots to simulate and analyze robot motion, ensuring 90% accuracy in path tracking compared to the planned trajectory.

III. Materials and Software

A. Materials

- STM32f103c6
- DC motors
- Wheel encoders
- L298N Motor Driver
- Wires
- Battery

B. Software

- Arduino IDE
- Webots simulation environment

IV. Procedures

1) Connect Arduino to DC motors and wheel encoders.
2) Implement differential drive kinematics in the code to control movement.
3) Write feedback algorithms to adjust robot motion based on encoder data.
4) Test robot movements in Webots, comparing them to the planned trajectory.

V. Results

In this experiment, the robot was programmed to travel forward for a distance of 1 meter and execute precise 90° turns at specified locations. Encoder feedback was actively used during movement to detect and correct any deviations from the intended path, enhancing motion accuracy.

During straight-line movement, the positional error consistently remained within 3.8 cm, demonstrating good linear precision. For rotational actions, the angular error was kept within 7°, confirming that the robot could perform reliable and consistent turns with minimal drift or overshooting.

VI. Discussion

The addition of wheel encoders greatly improved the robot's movement accuracy by allowing real-time corrections to each wheel's velocity. Some challenges, such as encoder slip and signal noise, were encountered during the process. However, these issues were effectively addressed by using a lower-value pull-up resistor and adding a capacitor to minimize signal bounce, resulting in more stable and reliable encoder readings.

Moreover, applying a differential drive model made it possible to accurately estimate the robot's position and orientation over time. This contributed to more consistent and predictable navigation, ensuring the robot maintained its intended path with greater precision throughout its movements.

VII. Conclusion

The experiment successfully met its objectives, with the robot demonstrating the capability to move forward and perform turns while keeping positional and angular errors within the designated tolerance limits. Simulations carried out in Webots further confirmed the robot's high accuracy in path tracking, supporting the effectiveness of the system design.

For future enhancements, integrating Proportional-Integral-Derivative (PID) controllers could be investigated to achieve even smoother and more reliable motion control. Additional improvements could involve modifying the system to perform well on non-planar or uneven surfaces, broadening its potential use in more challenging and varied environments.

REFERENCES

[1] STM32f103c6 Documentation, ”STM32f103c6 ”, https://www.st.com/en/microcontrollers-microprocessors/stm32f103c6.html.

[2] Cyberbotics Ltd., ”Webots User Guide”, 2024, https://cyberbotics.com/.

[3] R. Siegwart, I.R. Nourbakhsh, D. Scaramuzza, ”Introduction to Autonomous Mobile Robots”, 2nd ed., MIT Press, 2011.

APPENDIX

Differential Drive Control Code

```
// Motor pins (Driver 1 and Driver 2)
int pinlist[] = {PA0, PC15, PC14, PC13, PB10, PB1, PB0, PA7};
volatile int counter_isr = 0;
volatile bool stop_flag = false;

void moveForward() {
  digitalWrite(PA0, HIGH);  digitalWrite(PC15, LOW); // Motor 1
  digitalWrite(PC14, HIGH);  digitalWrite(PC13, LOW);  // Motor 2
  digitalWrite(PB10, HIGH);  digitalWrite(PB1, LOW);  // Motor 3
  digitalWrite(PB0, HIGH);  digitalWrite(PA7, LOW);  // Motor 4
}

void stopAllMotors() {
  for (int i = 0; i < 4; i++) {
    digitalWrite(motors[i].pin_on, LOW);
    digitalWrite(motors[i].pin_off, LOW);
  }
}


void counter_func(){

        counter_isr++;
        if(counter_isr >= 90)stop_flag = true;

}


void setup() {
  // Setup motors
  for (int i = 0; i < sizeof(pinlist)/sizeof(pinlist[0]); i++) {
    pinMode(pinlist[i], OUTPUT);
    digitalWrite(pinlist[i], LOW);
  }

        pinMode(PA3, INPUT_PULLUP);  // or INPUT, depending on your wiring
        attachInterrupt(PA3, counter_func, RISING);

}


void loop() {

        if(!stop_flag){ //pan move aprox 1m

        moveForward();

        }else{

        stopAllMotors();

        }

}
```