Laboratory 4*
KINEMATICS AND DIFFERENTIAL MOTION FOR MOBILE ROBOTS COMPONENTS

Romeo P. Aboganda Jr
Bachelor of Science in Electronics Engineering
Samar State University
Catbalogan City, Philippines
romeoaboganda25@gmail.com

***Abstract*—This laboratory activity explores the principles of kinematics and differential motion in mobile robots. Using a differential drive robot model, the experiment involves programming the robot's motion through kinematic equations and controlling its speed and trajectory. The study emphasizes the relationship between wheel velocities, robot movement, and the control logic required to achieve desired paths. Observations indicate that effective control strategies, such as tuning wheel velocities, are essential for precise movement, which has applications in fields like automation, navigation, and delivery systems.**

I. RATIONALE

The goal of this experiment is to understand the mechanics of mobile robots, particularly the kinematics and differential drive systems. Students will apply kinematic equations to calculate robot motion and implement algorithms that control wheel speeds for accurate navigation. This experiment will equip students with knowledge of differential motion control, which is critical in real-world mobile robot applications.

II. OBJECTIVES

- To explore the kinematic principles governing mobile robots, particularly the differential drive system.
- To program a mobile robot using differential motion control to navigate along specific paths.
- To develop an understanding of velocity control, trajectory planning, and motion coordination.
- To simulate and test different robot paths and refine control algorithms for real-world application.
-

III. MATERIALS AND SOFTWARE.

- Simulation Software (e.g., Gazebo, Webots) – for virtual robot modeling and testing
- Programming Language (e.g., Python, C++) – for robot control algorithm
- Mobile Robot Model – differential drive robot with two or more wheels
- Personal Computer or Laptop – for running simulations and writing code
- Internet Connection – for software updates or accessing resources

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

A. Procedure

1) **Set up the hardware**
    - Connect the Arduino Uno to the DC motors, servo motors, and stepper motor. Use the L298N motor driver to control the DC motors.
2) **Generate PWM signals**
    - Use PWM signals to control the speed of the DC motor. Vary the PWM signal to observe the motor speed change between 50 percent and 100 percent power.
    - Control the direction of the DC motor by adjusting the logic levels on the motor driver pins.
3) **Write the control code**
    - Program the Arduino using the Arduino IDE to control all three actuators. Implement the necessary code to change the speed and direction of the DC motor, and to position the servo and stepper motors accurately
4) **Simulate in Webots**
    - Simulate the actuators in Webots to test if the motors are moving correctly and maintaining the correct speed. Ensure that the motor behavior matches the expected results, achieving a success rate of 95 percent.

B. Observations and Data Collection

- **DC Motor Control:**By varying the PWM signal from 50 percent to 100 percent, the motor speed should change smoothly. The direction of the motor can be controlled by switching the polarity of the motor driver.
- **Servo Motor Control:**The servo motor should be able to rotate within a specified range (usually 0°

to 180°) depending on the control signals sent from the Arduino.
- **Stepper Motor Control:** The stepper motor should rotate in precise steps, with the ability to change direction by modifying the input signals.

C. Data Analysis.

The analysis of robot movement shows that precise control of wheel velocities is crucial for accurate navigation. The kinematic equations allowed for reliable predictions of robot paths, though slight discrepancies were observed when actual performance deviated from theoretical predictions. Data gathered from the simulation shows that altering wheel speeds affects both the robot's linear and angular velocities, underlining the relationship between motor control and robot movement.

WALKING GAIT EQUATION

In this experiment, we apply the basic equations governing the kinematics and motion of a differential drive mobile robot. These include forward kinematics, odometry for position estimation, and control laws for motion.

1. Forward Kinematics for Differential Drive Robot

The forward kinematics equations describe the robot's movement in terms of the wheel velocities and the robot's linear and angular velocities. These equations are:

$$v = \frac{r}{2} \times (\omega_L + \omega_R)$$

where:

- v is the linear velocity of the robot.
- r is the radius of the robot's wheels.
- ω_L and ω_R are the angular velocities of the left and right wheels, respectively.

The robot's angular velocity is given by:

$$\omega = \frac{r}{L} \times (\omega_R - \omega_L)$$

where:

- ω is the angular velocity of the robot.
- L is the distance between the two wheels (the wheelbase).

2. Odometry for Position Estimation

Odometry is used to estimate the robot's position and orientation over time. The change in position and orientation is calculated using the following equations:

$$\Delta x = v \times \cos(\theta) \times \Delta t$$

$$\Delta y = v \times \sin(\theta) \times \Delta t$$

$$\Delta\theta = \omega \times \Delta t$$

where:

- Δx and Δy are the changes in the robot's X and Y positions, respectively.
- θ is the robot's orientation (yaw).
- Δt is the time interval during which the movement occurs.

3. Control Law for Motion

To control the robot's movement, we may implement a control law such as the Proportional-Integral-Derivative (PID) controller. A typical PID control law for adjusting wheel velocities is:

$$\omega_L = \text{PID}_L(v_{\text{desired}}, v_{\text{actual}})$$

$$\omega_R = \text{PID}_R(v_{\text{desired}}, v_{\text{actual}})$$

where:

- v_{desired} is the target velocity.
- v_{actual} is the current velocity.
- PID controllers adjust the left and right wheel velocities based on the error between desired and actual velocities.

FIGURES AND TABLES

Trial No.	Left Wheel Angular Velocity (L)	Right Wheel Angular Velocity (R)
1	1.5 rad/s	1.5 rad/s
2	1.7 rad/s	1.5 rad/s
3	2.0 rad/s	1.5 rad/s
4	2.0 rad/s	1.3 rad/s
5	1.5 rad/s	1.7 rad/s

TABLE I
WHEEL VELOCITIES AND ROBOT MOVEMENT DATA

Distance Measurement Accuracy

Trial No.	Measured Distance (m)	Expected Distance (m)	Accuracy (%)	Er
1	0.9	1.0	90.00	
2	1.0	1.1	90.91	
3	1.2	1.25	96.00	
4	1.1	1.1	100.00	
5	0.9	0.95	94.74	

TABLE II
DISTANCE MEASUREMENT ACCURACY FOR ROBOT MOVEMENT