Laboratory 3: Odometry

Leonardo Acha Boiano
Mechatronic Engineering: IMT 342
Universidad Católica Boliviana Santa
Cruz de la Sierra, Bolivia
leonardo.acha@ucb.edu.bo

Andres Movladi Ayala Diaz
Mechatronic Engineering: IMT 342
Universidad Católica Boliviana 8th
Semester
andres.ayala@ucb.edu.bo

Gonzalo Peralta Jauregui
Mechatronic Engineering: IMT 342
Bolivian Catholic University 8th
Semester
gonzalo.peralta@ucb.edu.bo

Bruno Ramiro Rejas Montero
Mechatronic Engineering: IMT 342
Universidad Católica Boliviana 8th
Semester
bruno.rejas@ucb.edu.bo

MSc. Jose Jesus Cabrera Pantoja
Robotics (IMT-342)
jcabrera@ucb.edu.bo
Department of Mechatronic Engineering

INTRODUCTION

This report addresses the importance of odometry in mobile robotics, highlighting its key role in estimating the position and orientation of a robot based on the speeds of its motors. The process of measuring motor speeds is described and the integration of knowledge and control models previously developed in previous laboratories is emphasized. This report seeks to delve deeper into the principles of odometry and its practical application to improve the precision and efficiency in the navigation and control of mobile robots.

II. GOALS

• Control the speed and direction of a motor with an encoder • Ensure a stable connection between arduino and the motor • Create a mathematical model that allows controlling the motor • Verify proper operation of the PWM

III. THEORETICAL FRAMEWORK

to. PID controller

The PID controller (Proportional Integral Derivative) and its corresponding variants are the most used controllers in modern industry thanks to their ease of use and usefulness when working with plants whose mathematical model is unknown. In this laboratory, the parallel PID controller was implemented, which is given by the following formula:

( ) = ( ) + ÿ (ÿ) ÿ + — ( ) **(1)**

b. Direct kinematics

Differential direct kinematics, in the context of mobile robotics, refers to the process of calculating the linear and angular velocity of a mobile robot based on the individual speeds of its wheels or motors. In other words, it is used to determine how a robot moves and turns in space in response to speeds applied to its wheels or motors.

This kinematics is essential for trajectory planning and motion control of a mobile robot, as it provides a direct relationship between velocity inputs and motion outputs.

In this experiment, the following formulas were applied to calculate the linear and angular velocity of the robot. These equations are presented in their matrix form in equation (1) and are solved explicitly in equation (2).

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} r/2 & r/2 \\ 0 & 0 \\ r/2b & -r/2b \end{pmatrix} * \begin{pmatrix} \dot{\varphi}_r \\ \dot{\varphi}_l \end{pmatrix} \quad \textbf{(2)}$$

$$\begin{aligned} \dot{x} &= \frac{r}{2}(\dot{\varphi}_r + \dot{\varphi}_l) \\ \dot{\theta} &= \frac{r}{2b}(\dot{\varphi}_r - \dot{\varphi}_l) \end{aligned} \quad \textbf{(3)}$$

c. Inverse kinematics

Differential inverse kinematics, in the context of mobile robotics, refers to the process of calculating the individual speeds of a mobile robot's wheels or motors necessary to achieve a desired linear and angular speed or to follow a specific trajectory. In other words, differential inverse kinematics answers the question of how a robot's wheels or motors should move to reach a given speed and direction or to follow a predefined path.

This kinematics is essential to achieve precise control and autonomous navigation of robots mobile, since it allows moving objectives to be translated into commands for the robot's actuators.

In this experiment, the following formulas presented in their matrix form were applied in equation (4) and are solved explicitly in equation (5).

$$\begin{pmatrix} \dot{\varphi}_r \\ \dot{\varphi}_l \end{pmatrix} = \begin{pmatrix} 1/r & 0 & b/r \\ 1/r & 0 & -b/r \end{pmatrix} * \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} \quad \textbf{(4)}$$

$$\begin{aligned} \dot{\varphi}_r &= \frac{1}{r}(\dot{x} + \dot{\theta}b) \\ \dot{\varphi}_l &= \frac{1}{r}(\dot{x} - \dot{\theta}b) \end{aligned} \quad \textbf{(5)}$$

d. FreeRTOS

FreeRTOS is an open source real-time operating system for microcontrollers that makes it easy to program, deploy, secure, connect, and manage small, low-power peripheral devices.

and. Odometry

Odometry involves the study of estimating the position of wheeled vehicles during navigation. To carry out this estimation, information about the rotation of the wheels is used to calculate changes in position over time. In this approach, they rely on simple equations that can be implemented easily and that make use of the data.

of encoders installed on the robot wheels. However, odometry is based on the assumption that wheel revolutions can be converted into a linear displacement relative to the ground.

IV. DEVELOPMENT

1. **OdometryTask Function Description** The

OdometryTask function is a task that calculates the odometry of a robot. It uses the speed of two motors (motor1 and motor2) to determine the position and orientation of the robot in a two-dimensional plane (x, y) with respect to a reference point. Odometry is based on the movement of the robot's wheels and is commonly used in robotics for robot localization and tracking.

2. **Theoretical Explanation**

Odometry is a method for estimating the position and orientation of a mobile robot based on measuring the speeds of the robot's wheels. In this code, the speeds of the motors (in revolutions per minute) are used to calculate the distance traveled by each wheel. With this information, the robot's displacement and its change in orientation are estimated.

The odometry algorithm used is based on the differential kinematics of the robot. The distances traveled by each wheel are calculated based on the speeds of the motors and the diameter of the wheels. This information is then used to calculate the distance and change in orientation of the robot based on the left and right wheels.

3. **Code Analysis**

Initialized Variables:

x, y, and theta: Represent the initial position and orientation of the robot in the (x, y) plane and the theta angle in radians.

motor1_speed and motor2_speed: Store the speeds of motors 1 and 2 in revolutions per minute.

4. **Calculation of Distances and Change of Orientation:**

- The distances traveled by each wheel (deltaSLeft and deltaSRight) are calculated using the formula for the circumference of a wheel.
- The change in the position and orientation of the robot (deltaS and deltaTheta) is calculated using the differential kinematics formulas.

Position and Orientation Update:

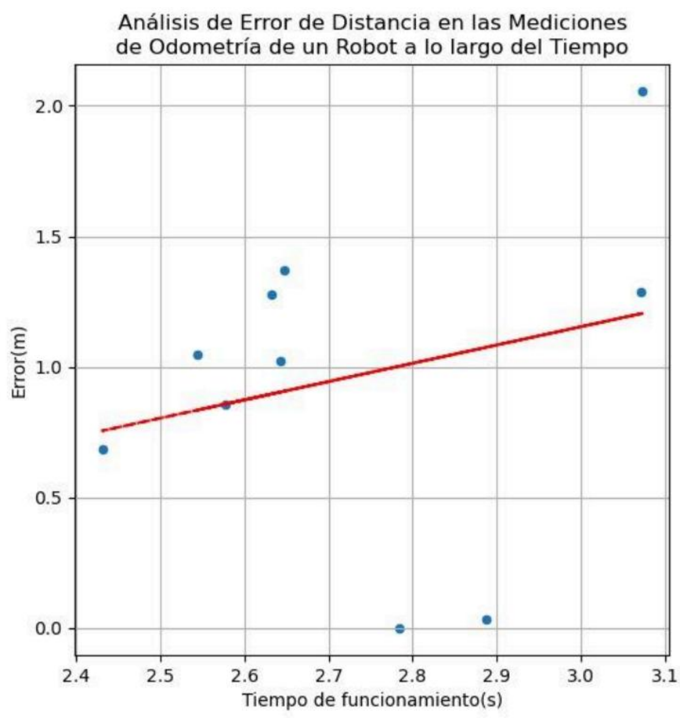

Figure 1 . Distance Error Analysis Chart in measurements

- The robot's position (xyy) and orientation (theta) are updated using changes in the calculated position and orientation.

Limitation of Guidance:

- It is ensured that the orientation angle (theta) is maintained in the range of [-pi, pi]. • Position and Orientation Record: • The current position and orientation of the robot is recorded in the log.

Task Execution Delay:

The task sleeps for 10 milliseconds to control the execution rate of the task.
This code runs in a continuous loop, updating the robot's position and orientation based on motor speeds and applying differential kinematics to calculate the robot's odometry.

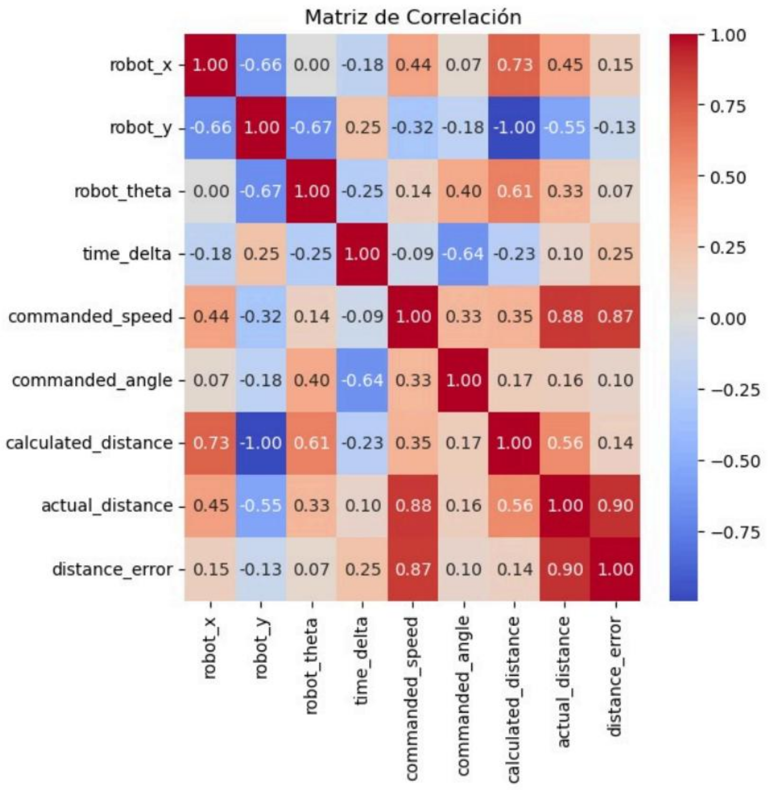

Figure 2. Coorelation Matrix Graph

The results obtained from the analysis of the Correlation Matrix reveal the interconnections and dependencies between various variables in the examined data set. It has been noted that as the position of the robot in the In contrast, it is seen that the increase in robot_x is associated with a noticeable decrease in the position of the robot on the Y axis (robot y) and the real distance. In particular, the strong negative relationship between robot_y and the calculated distance has been highlighted, indicating that an increase in the robot's position on the Y axis leads to a proportional decrease in the calculated distance. Furthermore, the marked correlation between the commanded speed and the actual distance, as well as the distance error, has been highlighted.

These findings provide insightful understanding into the complex interactions and dynamics that these variables exhibit within the data set under consideration.

V. CONCLUSION

The position on the X axis is positively related to speed, indicating greater speed when moving in this direction. The increase in X position is linked to a greater calculated distance, but also a decrease in Y position and actual distance. There is an inverse relationship between the position on the Y axis and the calculated distance. Speed impacts the accuracy of distance measurements, evidenced by the correlation with actual distance and distance error. In conclusion, the position of the robot in the X and Y axes, along with its orientation, influences speed and distance measurements, providing essential information to optimize its movement and precision.