Laboratory 2: Differential direct and inverse kinematics

Gonzalo Peralta Jauregui 8th Semester
gonzalo.peralta@ucb.edu.bo

Leonardo Acha Boiano 8th Semester
leonardo.acha@ucb.edu.bo

Andres Ayala Diaz 8th Semester
andres.ayala@ucb.edu.bo

Bruno Ramiro Rejas Montero 8th Semester
bruno.rejas@ucb.edu.bo

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

I. INTRODUCTION

This report focuses on direct and inverse differential kinematics applied to a mobile robot, with the incorporation of PID control models previously developed in a laboratory. We will see how these concepts are essential to control the movement of the robot and how their combination will allow us to build a mobile robot capable of executing autonomous tasks with precision and efficiency. The report will provide an overview of the theoretical foundations, the implementation of PID controllers in robot motors and the results of experiments, making it a valuable reference for those interested in mobile robotics and its practical application.

II. GOALS

- Control the speed of two motors simultaneously.
- Build a differential two-wheeled mobile robot with energy independence.
- Implement linear direct kinematics and angular direct kinematics of the differential robot built.
- Perform the necessary experiments to obtain the average error.
- Verify the proper functioning of the platform and the robot.

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 speed 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 velocity and angle of the robot. These equations are presented in their matrix form in equation (#) and are solved explicitly in equation (#).

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

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

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 mobile robots, as it allows the translation of motion objectives into commands for the robot's actuators.

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

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

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

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. mobile robot

A mobile robot is a type of robot designed to move and operate in different environments or surfaces autonomously or controlled by an operator. These robots are equipped with locomotion systems that allow them to move, sensors to perceive their environment and often

They include information processing capabilities to make decisions and perform specific tasks. Mobile robots can vary in size, shape and applications, and are used in a wide range of industries and fields, from manufacturing and logistics to space exploration and healthcare.

Some examples of mobile robots are: space exploration robots, autonomous delivery robots, cleaning robots, agricultural robots, medical assistance robots, cleaning robots, rescue robots, and educational robots.

In Figure 1 you can see the mobile robot model based on which the mobile used in the present experiment was designed.

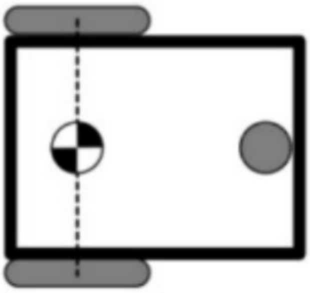

Figure 1. Initial Mobile Robot Prototype

IV. DEVELOPMENT

- **Mobile Robot Design**

The mobile robot was designed and assembled with all the necessary components, including motors, wheels, and the ESP32 control board. The ESP32 was chosen as the main platform due to its effectiveness in simultaneously controlling two motors with encoders, using a PID controller.

A complete and detailed view of the design of the proposed mobile robot is presented in Figure 2, which allowed precise measurements of the motor mounts and wheels. It was necessary to custom manufacture the wheels, since no standard measurements could be found that would fit perfectly to the motor shaft.

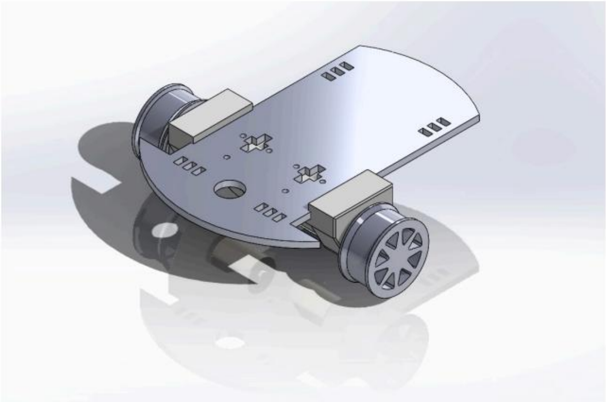

Figure 2. Computer Aided Design of Two Wheeled Differential Robot

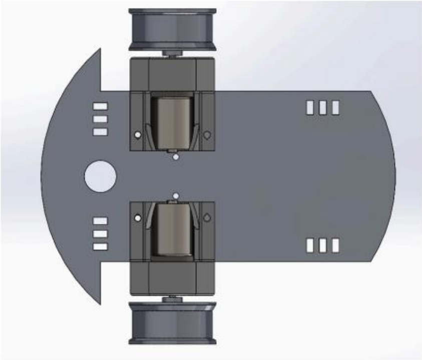

Figure 3. Bottom view of the CAD model of the mobile robot.

In Figure 3, you can see the couplings designed to align and fix the motors on the base of the robot. It is important to highlight that an adequate coupling guarantees that vibrations do not affect the operation of the motors, which contributes to the stability of the system.

As shown in Figure 4, the design of the motor mount has been designed to allow easy insertion of the motor, but with enough pressure to hold it firmly in position. This prevents any play and ensures the stability of the motor. Furthermore, the same design has been applied to the shaft, following the same premise.

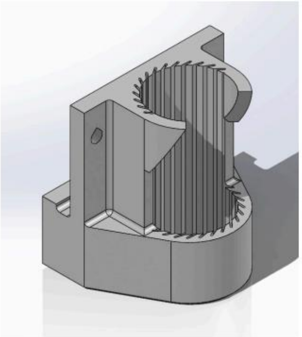

Figure 4. Motor Support.

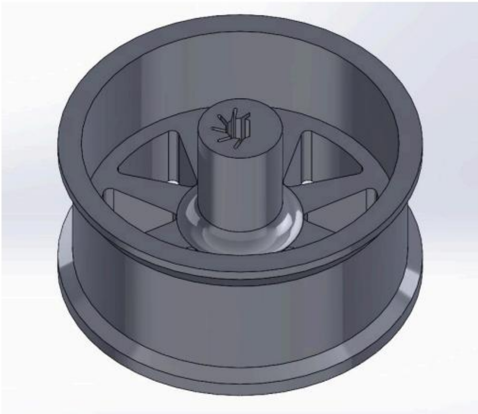

Figure 5. Wheels of the Mobile Robot.

A custom model has been created for the wheel, as it was not possible to find a commercial wheel with a suitable axle for the motor. Figure 5 shows this model, and it is observed that its shaft has the same design to allow easy insertion, but with the necessary firmness to avoid any play.

- **Printed Circuit Board Design:**

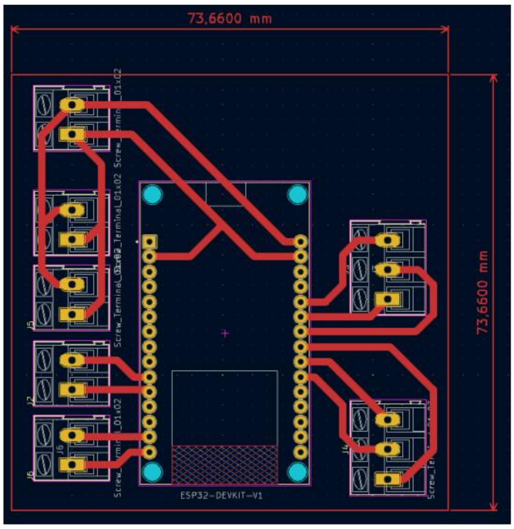

Figure 6. View of the PCB implemented to improve cable organization

Figure 6 presents a careful design of a printed circuit board (PCB) with a clear focus on system organization and efficiency. This PCB neatly houses the ESP32 and provides specific connection terminals for both the power supply and the motors equipped with encoder. This design not only eliminates cable clutter, but also simplifies system installation and maintenance, ensuring smoother operation and greater reliability.

- **Final assembly of the Mobile Robot**

The robot was assembled, supplying an 11.1 V lipo battery as a power source. energy. In Figure 7 you can see the armed robot connected to the battery charger.

Figure 7. Developed robot connected to a charger.

• **Controller Tuning**

After assembling the system, the behavior of the speed controllers implemented for the motors was reviewed. As seen in Figure 8, both drivers were not working satisfactorily on the new system. Something natural if you take into account that the plant dynamics changed, the H-bridge module that previously used one motor is now used by two, the previously available power supply was replaced by a lipo battery, the charging of the wheels, the weight of the mobile and above all the greater use of resources in the microcontroller that the operating system used must handle. Thus, the motor controllers were recalibrated. The results of this calibration can be seen in the step response in Figure 9.

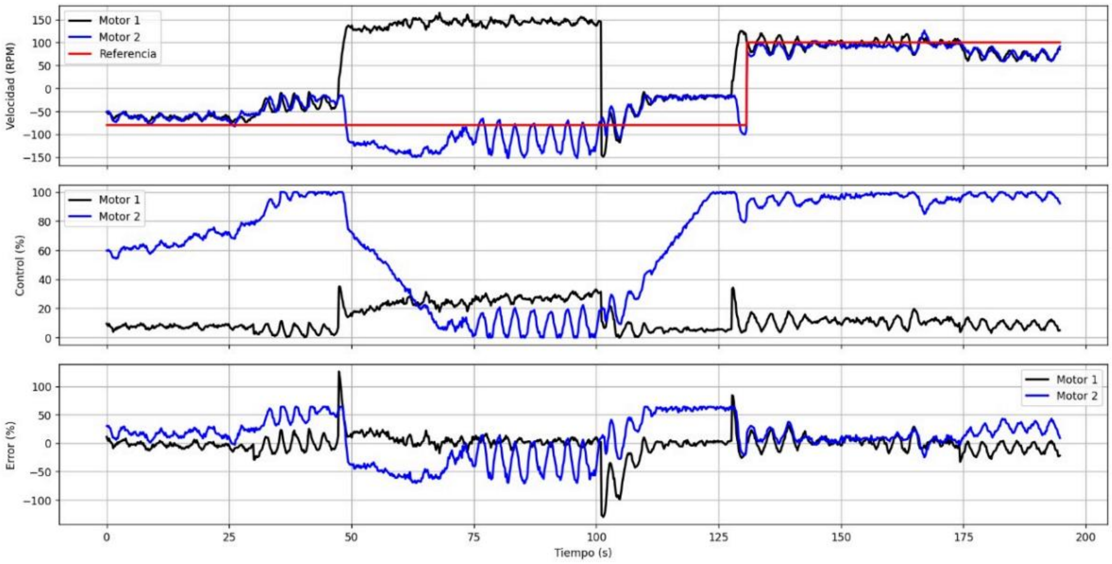

Figure 8. Step Response of the Motors after assembling the system.

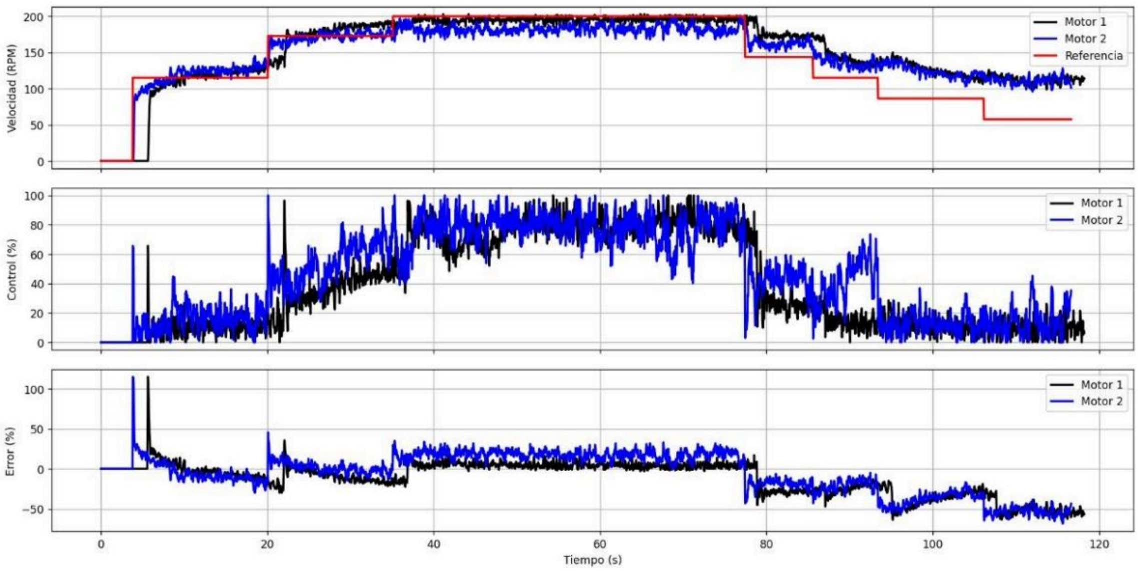

Figure 9. Step Response of the Motors after Retuning.

• **Cinematic Analysis**

The developed system makes use of direct and inverse kinematics as shown in Figure 10. Through http protocols, signals corresponding to the linear and angular speeds that the robot is desired to have are received. Equations 4 and 5 of inverse kinematics are responsible for providing the corresponding values in terms of angular speeds of the wheels; these speeds are transformed into references for the PID controllers of the motors.

Finally, the real speed is processed in a task dedicated to evaluating whether the real speed of the robot corresponds to the speed requested via Wi-Fi and sends these values by UART or by WIFI communication, failing that.

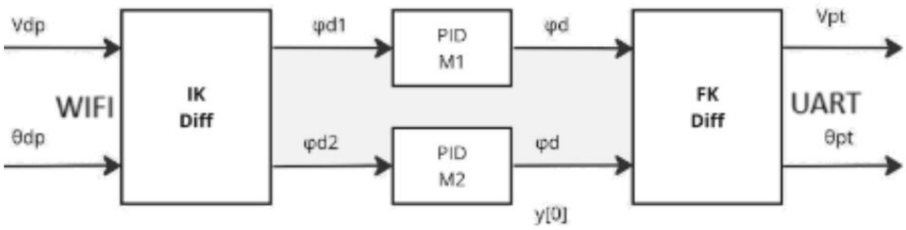

Figure 10. Block Diagram of the Developed System.

In the development of the code, it was taken into account that direct kinematics calculates the current position and orientation of the robot based on the speeds of its wheels.

1) DirectKinematicsTask: • Read wheel speeds from queues (motor1SpeedQueue and motor2SpeedQueue).
   - Calculates the linear speed (robot_speed) and the angular speed (robot_angle) of the robot using the formulas of differential kinematics.
   - Records linear and angular velocity on the console.

In turn, the inverse kinematics block calculates the wheel speeds necessary to achieve a desired speed and orientation of the robot. This code is responsible for handling an HTTP request to update the speed and orientation of the robot.

2) handle_set_robot_speed: • Validates and extracts the speed and orientation of the robot from the http request. • Calculate the wheel speeds (v_left and v_right) using the inverse kinematics formulas.

   • Applies limits to wheel speeds and converts them into percentages. • Sets the speed references for the motors. • Records updated speeds on the console.

The results obtained after testing and plotting the controller and the reference that both motors followed can be seen in Figure 11, 12 and 13.

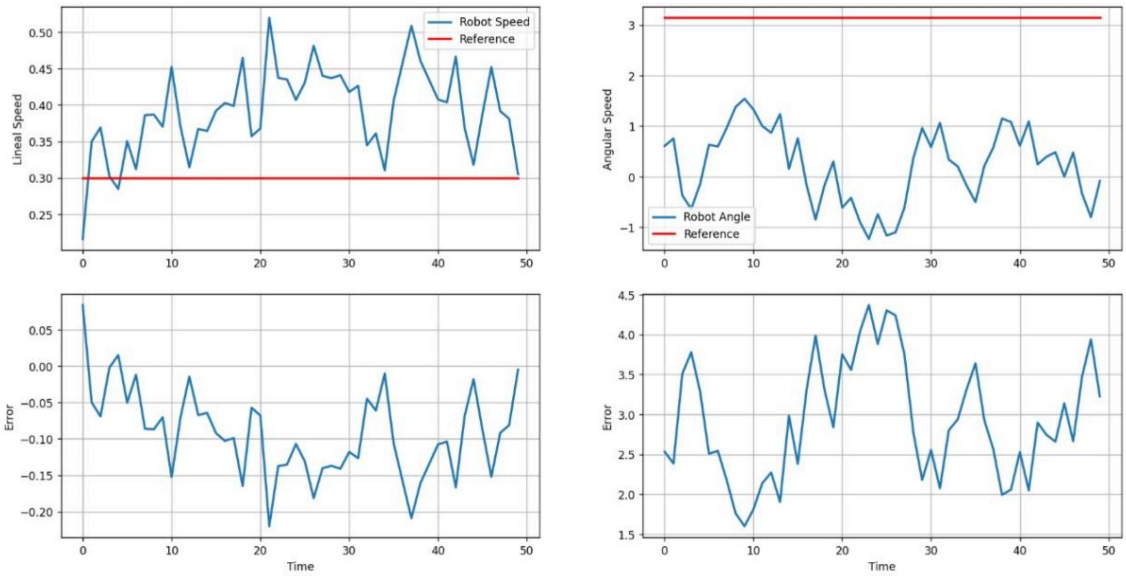

Figure 11. Step Response of the Motors after assembling the system.

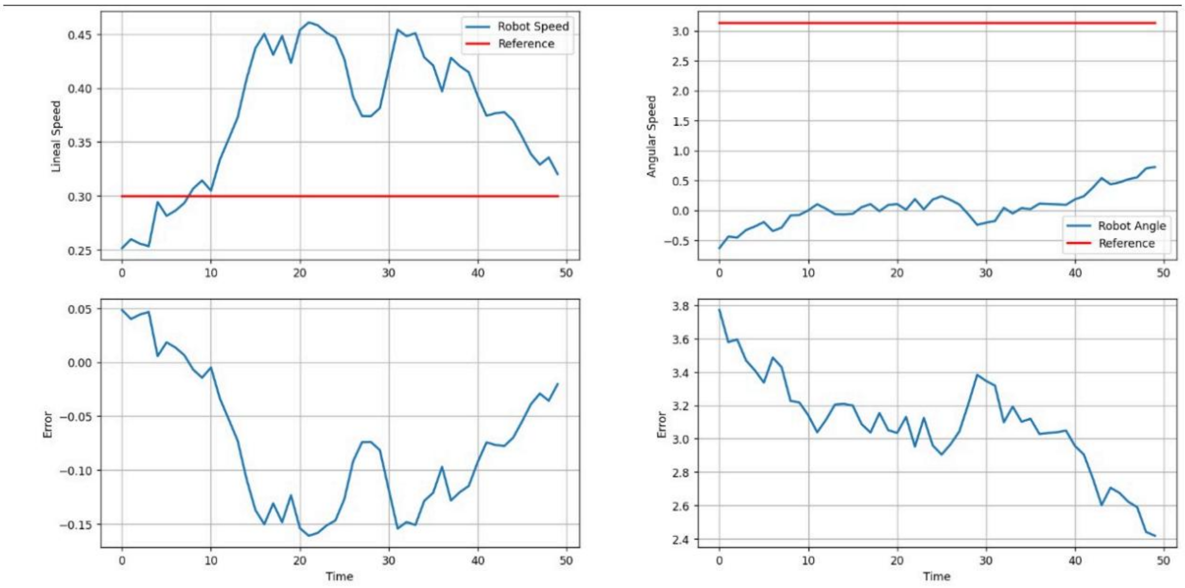

Figure 12. Step Response of the Motors after assembling the system.

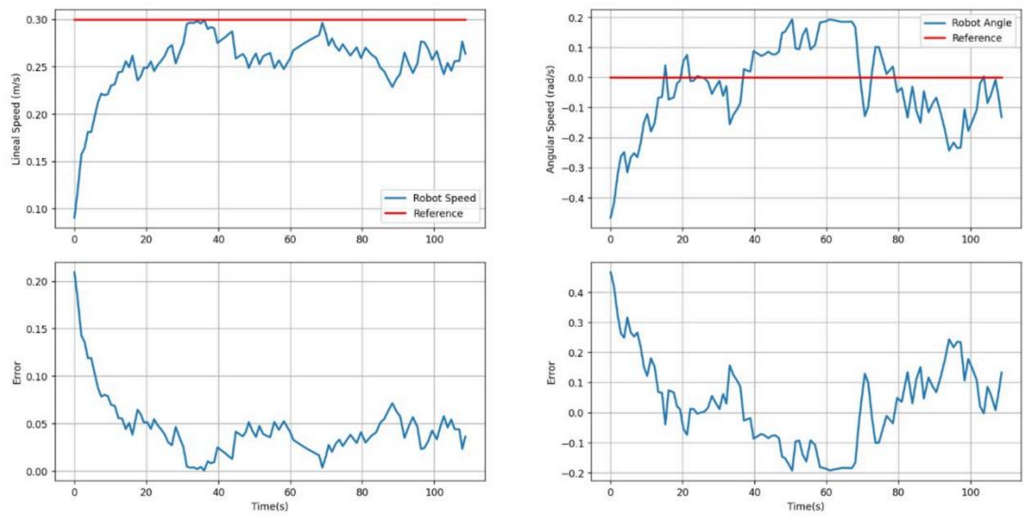

Figure 13. Step Response of the Motors after assembling the system.

The first kinematics analysis, shown in Figure 11, reveals notable differences between the reference signal (represented in red) and found by the direct kinematics equations (in blue). It is evident that direct kinematics presents superior performance in linear velocity compared to angular velocity. In a second experiment, represented in Figure 12, a trend towards convergence with the reference is evident as the time period extends. Finally, Figure 13 shows that the linear speed tends to equal the reference when the robot angle is kept at 0.

The numerical results reveal an average error in linear velocity of -0.1143 m/s and an average error in angular velocity of 2.9437 rad/s. These data indicate the need for recalibration and adjustment of task periods and priorities, in order to guarantee consistent results, even in the face of variations in reference values. These adjustments are considered essential to achieve accurate and consistent system response, a goal that will remain the focus of analysis and development in subsequent research.

V. CONCLUSION

In conclusion, the speeds of two motors were successfully controlled simultaneously using PID controllers. A two-wheeled differential mobile robot was built with energy independence, taking care of the design details of the components, such as motors, wheels and ESP32 control board. Then, direct kinematics equations were implemented to calculate the linear and angular speed of the robot based on the individual wheel speeds.

In the experiments, the robot's responses at reference speeds were evaluated. Numerical results revealed an average error in linear velocity of -0.1143 m/s and an average error in angular velocity of 2.9437 rad/s compared to the reference velocities.

These values indicate the need for adjustments and calibration to achieve a more precise and coherent response of the system, especially when faced with variations in the reference values. These adjustments are considered essential to ensure optimal control of the robot in future developments. In summary, the experiments demonstrated the successful application of forward and inverse kinematics in the control of a mobile robot, although they also highlighted the importance of continuous improvement to obtain more accurate and consistent results.