XLIII CILAMCE
21–25, November, 2022
FOZ DO IGUAÇU – PR – BRAZIL

Odometry and speed control of a 4WD mobile robot integrated with ROS 2

Leonardo G. Batista[1], Pablo F. Salarolli[1], Daniel F. T. Gamarra[2], Gustavo M. de Almeida[1], Rafael P. D. Vivacqua[1], Marco A. S. L. Cuadros[1]

[1]*Master's program in control and automation engineering, Federal Institute of Espírito Santo*
Av. dos Sabiás, 330 - Morada de Laranjeiras, 29166-630, Serra-ES, Brazil
leonardo-baptista@live.com, pablosalarolli@gmail.com, gmaia@ifes.edu.br, rafsat@ifes.edu.br, marcoantonio@ifes.edu.br

[2]*Control and Automation Engineering Course, Federal University of Santa Maria*
Av. Roraima, 1000 - Cidade Universitária, 97105-900, Santa Maria - RS, Brazil
fernandotg99@yahoo.com

Abstract. With the advancement of technology and artificial intelligence, robots are increasingly used in various applications. There are various types of mobile robots, such as air, land and sea. Among the various types of wheeled mobile land robots, the differential drive 4WD mobile robot stands out for its better performance in navigating outdoor environments, mainly due to the presence of four-wheel drive. However, the differential drive 4WD robot exhibits greater wheel slippage especially in the moments of rotation. This type of behavior complicates the accuracy of classical localization methods, such as odometry. In this paper, a real case of odometry implementation is presented, as well as wheel speed modeling and speed controller implementation in a 4WD robot. To enable the use of more advanced libraries and later the development of various applications, the odometry and linear and angular speed control of the robot were integrated into the ROS (Robot Operating System). Practical results in internal and external environments are presented at the end of the paper using graphs and a video with an access link.

Keywords: Odometry, 4WD robot, ROS.

1 Introduction

Robots are devices capable of performing a range of tasks with a high degree of autonomy. Nowadays, there are robots that perform tasks in the home, in industry, and even in space [1], [2], with some of them completely replacing humans in high-risk activities such as underwater and space exploration, defusing explosive devices, working in radioactive environments, etc. [3], [4]. Robots can also be used for repetitive tasks that require a high degree of precision and are difficult or impossible for humans to perform [5]. In order to perform tasks accurately, the mobile robot must be able to navigate reliably. Navigating wheeled mobile robots is not a trivial problem because it is difficult to determine the exact location (position and orientation) of the mobile robot [6].

In [7] the authors define mobile robot localization as the problem of estimating the position of a robot with respect to the environment in which it is located. Therefore, it is a critical problem for mobile robotics. In [8] the author states that location is the most fundamental problem in mobile robotics and that it is unlikely to make a mobile robot autonomous without an accurate location system. Location of mobile robots can be divided into two groups of measures: relative measures, also known as dead reckoning, and absolute measures, known as reference guidance [9].

Dead reckoning based methods use internal sensor sources of the robot that are independent of an external reference. Therefore, odometry is one of the most common methods, where the robot's position is estimated using information from encoders connected to the wheels of the vehicle [9], [10]. However, along the motion of the

robot, odometry accumulates errors in an unbounded manner. In this paper, we present the results of odometry of a 4WD mobile robot based on practical tests, as well as the modeling, tuning and implementation of wheel speed controllers integrated in ROS 2.

This article is organized as follows: in section 2 the 4WD mobile robot is presented, in section 3 some necessary methods are described, in section 4 the obtained results are presented and finally in section 6 the conclusions and final considerations are discussed.

2 4WD Mobile Robot

The development of practical robotics work requires robotic structures capable of performing the above tests. However, importing structures that have already been validated by large companies can be a costly and infeasible task. In this sense, the development of robotic structures is a viable solution, but it is exhausting and complex due to the errors and successes that occur during the process.

In this work, a mobile robot with four-wheel drive was used. This robot was designed to operate in internal and external environments, so its mechanical structure was reinforced. To reduce the number of motors needed, the robot was equipped with two motors. The transmission to the other wheels was done by a belt and pulley mechanism, as shown in Figure 1.

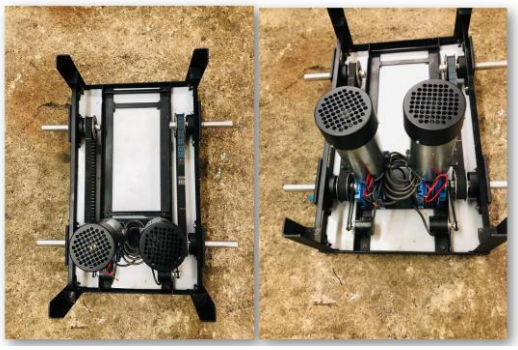

Figure 1 - Chassis developed for the 4WD differential robot.

The Figure 2 shows a diagram of how the connections between the devices were made and presents how each device was inserted into the mobile robot. The use of the wifi router allowed an external laptop to be connected to the system, through which a graphical interface was developed to display graphs and data in real time. Velocities were transmitted to the robot via a video game controller integrated into ROS 2. In this way, it was possible to generate velocity references (standard geometry_msgs/Twist.msg) depending on its analog joystick.

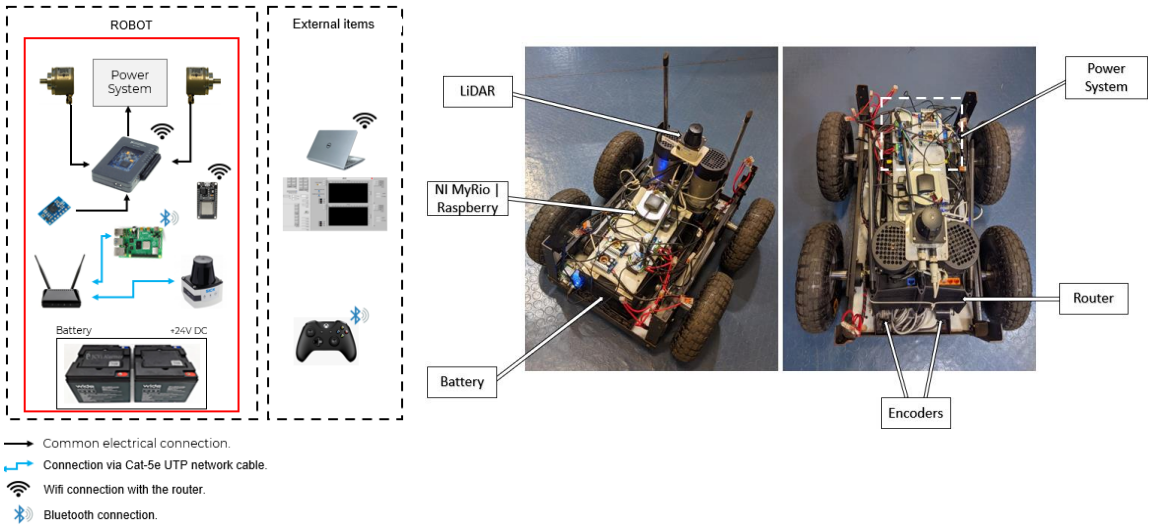

Figure 2 - Device integration.

3 Methodology

3.1 Odometry

The odometry technique uses robot kinematics to determine position. The kinematic model describes only the velocities of the vehicle, without considering its mass and the forces or torques that cause these velocities [11]. The discretization of the kinematic model makes it possible to find the expression for updating the pose at each iteration. Given the numerical integration by the first-order Euler approximation the estimate of the robot position is given by Equation (1). Here, x and y represent the position of the point of interest (in this case, at the centre of the virtual axis connecting the drive wheels), the orientation θ of the robot (i.e., the direction of the axis), v and ω is the linear or angular velocity of the robot. Figure 3 shows the attitude of the single-wheeled robot for this case.

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ \theta_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} + \begin{bmatrix} v_k * \cos(\theta_k) \\ v_k * \sin(\theta_k) \\ \omega_k \end{bmatrix} \tag{1}$$

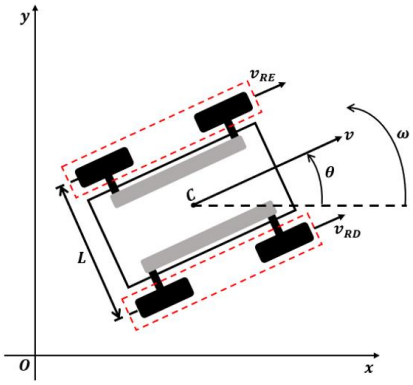

Figure 3 - Differential mobile robot configuration.

3.2 Software Architecture

Figure 4 shows a simplification of the software architecture used. As shown, NI myRIO was responsible for determining the odometry estimates and executing the linear velocity control loops for each wheel. Point-to-point communication via the TCP-IP socket allowed NI myRIO to publish the odometry information and read the linear and angular velocity setpoints. The Raspberry Pi was responsible for running ROS 2, in addition to reading data from the LiDAR sensor and joystick. After the whole implementation, it was possible to control the mobile robot with the joystick. This fact allowed a better positioning of the robot during the odometry tests and during the validation of the tuning of the speed controllers.

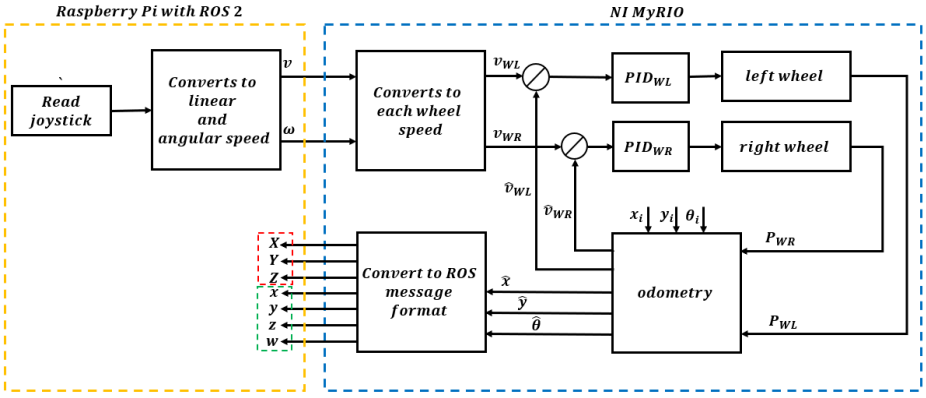

Figure 4 - Software Architecture.

4 Results

To obtain the model of the two sets of wheels, the robot was positioned in the test environment. The step tests were performed with the robot wheels in contact with the ground to obtain a better representation of the system dynamics. Open circuit tests were then performed and data collected. The extracted curves allowed the determination of the model parameters of each wheel set. Internal model control (IMC) was chosen as the tuning method for the PIDs. Once the controller setting was determined, the next step after implementation was to test the robot with the proposed setting. Figure 5 shows that the control loop for the left wheel worked as expected, showing zero error in the steady state and a response that matched the specified one. The same happened with the right wheel assembly (Figure 5). After the whole implementation, it was possible to control the mobile robot with the joystick. This fact allowed a better positioning of the robot during the odometry tests and during the validation of the tuning of the speed controllers.

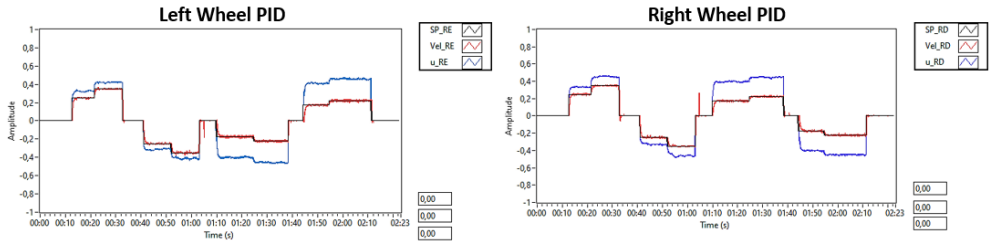

Figure 5 - Result of tuning the controllers.

Odometry assessment tests were performed in a controlled environment. First, a starting point was established, and the robot was slowly moved to a new position using the joystick. A total of 17 different positioning points were used, which allowed the robot to be tested on straight lines and curves. At each of these points, the ground truth of the pose was recorded very carefully to be later compared with the data estimated by the odometry of the robot. As for the orientation error, the graph in Figure 6a shows that the odometry was not successful, especially after the first turn. This problem was to be expected given the slippage that the wheels have when turning in this structure. The same effect was confirmed in the following curves, so that the estimated orientation was completely different from the actual one. The severe orientation problems affected the position estimation, which was already expected since these measurements depend on the orientation of the mobile robot as seen in its kinematic model. Figure 6b helps to better understand the fact described in the previous section. Six cases have been defined to compare the estimated and the actual position and to understand the odometry load.

In the first case, the robot is on a straight path. We can see that the estimated pose is close to the real pose, which confirms the fact that the odometry of this robot works relatively well for displacements in straight lines. The second evaluation point shows the inadequacy of the orientation estimation caused by the slippage of the wheels. The third point did not show a major orientation error, as the robot moved only on a distance close to a straight line. The fourth and fifth points show and reinforce the deficiency in odometry for cornering conditions by giving the impression that the robot is in a position that does not match the actual position. And the sixth and last points, like the previous ones, show the deficiency in the odometry of this robot to a greater extent, with a final estimate that is completely disoriented and displaced from the real one.

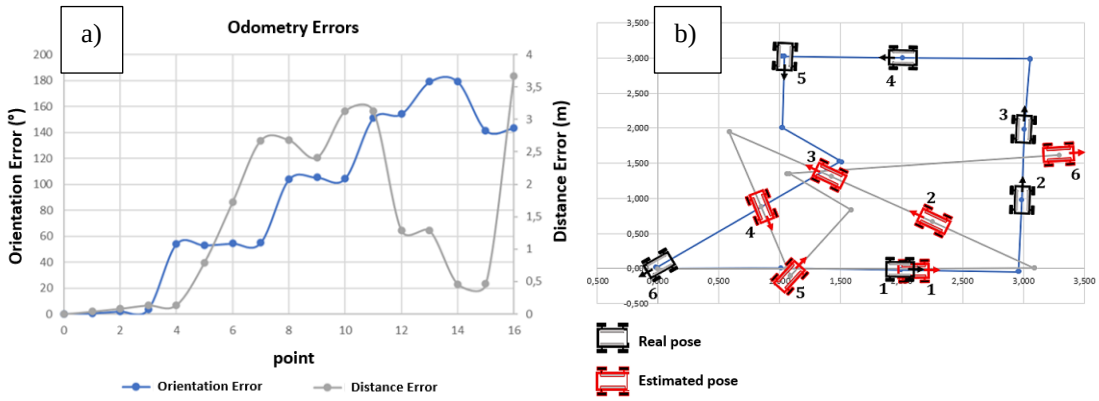

Figure 6 - Odometry Erros.

5 Conclusions

The teleoperated navigation tests have shown that the developed robotic structure is suitable for indoor and outdoor navigation. Up to this stage, a number of problems have been solved, many of which replace mechanical and electrical components, such as the reduction gears and drives. The fact that it is a 4WD structure made it much easier for the robot to navigate on different floors and terrains. A in video of the robot navigating indoors is shown in [12] and outdoors is shown in [13] proving the facts stated here.

Using the results of odometry, it was possible to verify the problem caused by the landslides in the estimates of this method. On the straight lines, the odometry proved to be satisfactory with the real pose, but in the curves, the wheels turned more than necessary, resulting in orientation values that did not match the execution. Furthermore, these errors were passed on as the robot navigated through the environment. In the tests performed, the robot had a final range error of more than 3.5 m and an absolute orientation error of more than 140°. This fact proves the need to combine odometry with other techniques to obtain a more accurate localization system.

The modeling and tuning of the controllers proved to be satisfactory, being able to ensure the control of the velocities in unfavorable situations of the model. ROS was important to allow the control of the robot with a joystick and to favor future implementations.

Acknowledgements. The development of this work was supported by the Graduate Program in Control and Automation Engineering (ProPECAut) of the Federal Institute of Espírito Santo - Campus Serra in partnership with GainTech Tecnologia LTDA.

References

[1] R. Goel and P. Gupta, “Robotics and Industry 4.0,” 2020, pp. 157–169. doi: 10.1007/978-3-030-14544-6_9.

[2] H. Kalita, A. S. Gholap, and J. Thangavelautham, “Dynamics and Control of a Hopping Robot for Extreme Environment Exploration on the Moon and Mars,” in *2020 IEEE Aerospace Conference*, 2020, pp. 1–12. doi: 10.1109/AERO47225.2020.9172617.

[3] L. D. L. Barker *et al.*, “Scientific Challenges and Present Capabilities in Underwater Robotic Vehicle Design and Navigation for Oceanographic Exploration Under-Ice,” *Remote Sensing*, vol. 12, no. 16, p. 2588, Aug. 2020, doi: 10.3390/rs12162588.

[4] A. Ibarra, A. Cóndor, P. Martínez, and E. Tipán, “Control reengineering used for rehabilitation of Andros Remotec bomb disposal robot,” in *2020 IEEE ANDESCON*, 2020, pp. 1–6. doi: 10.1109/ANDESCON50619.2020.9272161.

[5] S. S. Khan and A. S. Khan, “A Brief Survey on Robotics,” 2017. Accessed: Aug. 07, 2019. [Online]. Available: www.ijcsmc.com

[6] D. Fox, W. Burgard, and S. Thrun, *Probabilistic robotics*. 2005.

[7] S. Thrun, D. Fox, W. Burgard, and F. Dellaert, “Robust Monte Carlo localization for mobile robots,” *Artificial Intelligence*, 2001, doi: 10.1016/S0004-3702(01)00069-8.

[8] D. Fox, “Adapting the sample size in particle filters through KLD-sampling,” *International Journal of Robotics Research*, 2003, doi: 10.1177/0278364903022012001.

[9] J. Borenstein, H. R. Everett, L. Feng, and others, “Where am I? Sensors and methods for mobile robot positioning,” *University of Michigan*, vol. 119, no. 120, p. 27, 1996.

[10] M. B. Alatise and G. P. Hancke, “A Review on Challenges of Autonomous Mobile Robot and Sensor Fusion Methods,” *IEEE Access*, vol. 8, pp. 39830–39846, 2020, doi: 10.1109/ACCESS.2020.2975643.

[11] R. Fierro and F. L. Lewis, “Control of a nonholomic mobile robot: Backstepping kinematics into dynamics,” *Journal of Robotic Systems*, 1997, doi: 10.1002/(sici)1097-4563(199703)14:3<149::aid-rob1>3.3.co;2-n.

[12] GainTech, “G ROBOT IO - Teste de navegação em ambientes internos - YouTube,” 2022. https://www.youtube.com/watch?v=f8YQrBDPh34 (accessed May 11, 2022).

[13] GainTech, “Projeto G-ROBOT IO - YouTube,” 2022. https://www.youtube.com/watch?v=_cWDR9GNiJw&t=43s (accessed May 09, 2022).