Three Wheeled Omnidirecional Mobile Robot - Design and Implementation

Darci Luiz Tomasi Junior
Tuiuti University of Paraná
Curitiba - PR, Brazil
darci.junior@utp.br

Eduardo Todt
Federal University of Paraná
Curitiba - PR, Brazil
todt@inf.ufpr.br

ABSTRACT

This article presents a study of the resources necessary to provide movement and localization in three wheeled omnidirectional robots, through the detailed presentation of the mathematical procedures applicable in the construction of the inverse kinematic model, the presentation of the main hardware and software components used for the construction of a functional prototype, and the test procedure used to validate the assembly.

The results demonstrate that the developed prototype is functional, as well as the developed kinematic equation, given the small error presented at the end of the validation procedure.

KEYWORDS

Omnidirectional, Mobile Robot, Odometry, *UMBMark*

1 INTRODUCTION

In the mobile robotics context, one of the most critical tasks performed by an autonomous system is self-localization in an unknown environment, making use of physical and computational resources embedded or not in the base. The input data to execute this localization process are provided by sensors that, when correctly interpreted by a designated system, estimate the current robot positions. This designated system shall understand and implement Kinematic equations previously calculated for each robot construction and configuration.

In this context and concerning a three-wheeled omnidirectional robot, the research presents a detailed procedure to generate the inverse kinematic equations of the robot developed and complementarily validate this equation in a real robot base using the well known *UMBMark* test procedure.

This research is divided into five main sections: Review section presents the basic concepts applicable to kinematic study, Calculation section presents the kinematic development, Implementation section presents the components used, Results section presents the base performance and Conclusion section the final remarks, contributions, and future works.

2 BACKGROUND

In this section, main concepts and definitions are presented in order to provide a theoretical basement to simplify the comprehension of the next sections.

2.1 Ominidirectional wheeled platforms

There are two types of omnidirectional wheeled platforms: one type is using special wheels, and the other type includes conventional wheels. Special wheels, which have been mostly studied for the omnidirectional mobile platforms, have an active tracking direction and a passive moving direction [5].

There are several types of omnidirectional wheels, but in all of them, the principle of function is based on providing traction in the direction normal to the motor axis and the use o inner passive rollers that can slide in the direction of the motor axis. These inner passive wheels, balls, or rollers are placed along the periphery of the main wheels [8].

Fig. 2 shows different types of omnidirectional wheels with other constructions and characteristics[1]. The overlapping parallel model was selected for this development, and VEX Co. supplies the wheels, Ltd [18].

As shown in Fig. 1 the robot is equipped with three motors, and each motor is attached to one wheel, where the center axis pass through the middle point of the base and generates 120° between each wheel.

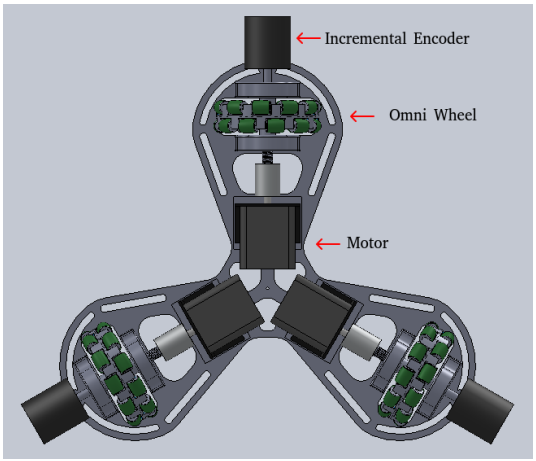

Figure 1: The omnidirectional robot project, developed with three motors, omnidirectional wheels and incremental encoders. Due to components position *DDOF* is 3.

2.2 Differentiate Degrees of Freedom

One of the essential characteristics to be observed in a mobile robot developed to navigate in an unknown environment is the *differentiate degrees of freedom* (*DDOF*) characteristic. *DDOF* describes the number of independent components of robot motion that the

[1]Discontinuous traces are a known vibration source to the system and can generate different kinds of problems and behaviors. To find more information about it, please refer to [20].

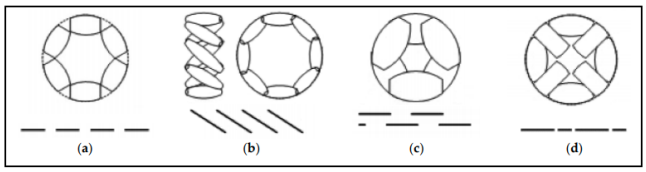

Figure 2: (a) Multiple passive rollers; (b) Overlapping passive rollers; (c) Overlapping parallel wheels; (d) Alternated passive rollers with different size and shape [8].

.

robot system can control [16], represented as a velocity space vector. Considering the structure illustrated in Fig. 1, *DDOF* will be three due to 90-degree passive rollers arrangement, and it means that the robot can achieve any pose in the environment (X, Y, θ), directly achieving the goal positions of all three axes simultaneously [16]. This advantage not only improves the flexibility of the robot greatly in order to achieve fast target tracking and obstacle avoidance but also provides more references for robot motion control methods [19].

Non-omnidirectional robots are widely used on robotics research, industrial and non-professional applications as presented in [17], [4], [9]. However, generally they require more complex maneuvers and path planning than omnidirectional bases, such as Fig. 3 shows.

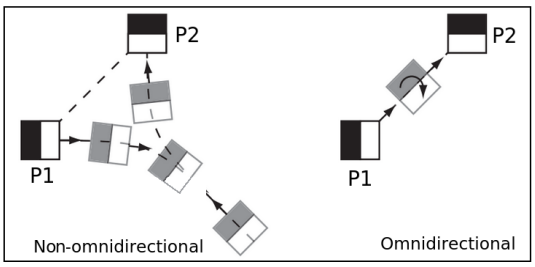

Figure 3: Omnidirectional and non-omnidirectional bases performing the same trajectory from *P1* to *P2*. Omnidirectional base needs less space and simple path planning strategy to reach the goal. Modified from [7].

2.3 Forward and Inverse Kinematic

To define the path that the robot follow, the planning control shall be able to understand the mechanical behavior of the entire system, doing it through the kinematic equations. These equations are expressed as forward kinematic and inverse kinematic. Generalizing the definitions, forward kinematic is a function that receives wheel speed and returns the estimate *Cartesian* position, and inverse kinematic receives the *Cartesian* position and estimates the Wheel speed to attempt the target.

Generally, to manipulate the robot in an unknown environment, the inverse kinematic is most attractive to the motion control system due to the origin of the problem [10], or in other words, before attempting to throw a basketball, it is easier to first look at the basket and throw it, than simply throw the ball and than look to the basket.

2.4 Proprioceptive and Exteroceptive Measurements

Sensors used in robotics can be divided into two classes, proprioceptive and exteroceptive sensors. The proprioceptive sensors measure internal values to the system (robot). For example, engine speed, wheel load, angle of robot arms, and battery voltage. On the other hand, the exteroceptive sensors acquire information about the environment where the robot travels. For example, distance measurements, luminous intensity, features in images, and sound amplitude [17].

In the context of location for mobile robotics, the proprioceptive sensors stand out due to the ease of reading and a high degree of precision. Thus, an estimate of the current location of the base is possible through an odometry system. Concerning the omnidirectional structure illustrated in Fig. 1, each displacement system (motor + wheel) is coupled with an incremental encoder to measure the motor speed. This data is periodically collected and integrated using equations presented in Section 3 to estimate the position and orientation (POSE) of the robot on time interval t.

In a perfect world, odometry results are the only information needed to perform navigation tasks. Unfortunately, sensors are susceptible to measure not only true data but erroneous information as well, and due to integration characteristics, these error accumulates, increasing the localization uncertainty over time.

One task generally applied to reduce uncertainty is to incorporate exteroceptive sensors in robot systems to estimate POSE based on the environment where the robot is, and incorporate probability techniques to increase the position accuracy, like the cyclic estimation-measure steps found in SLAM algorithms using Kalman filters [6].

To enable this localization task, robot incorporate one monocular camera supplied by Logitech Co. Ltd [11], to be used as images source with any visual odometry algorithms such as presented in [2], [12], [1], [15].

2.5 *UMBmark* test procedure

UMBmark [3] was developed as a procedure to reduce the odometry error in differential drive robots, measuring and compensating these errors through a calibration process. However, the methodology used to verify the performance of the robot is widely used by mobile robot researchers [19], [17], [4], [9], and is used in this research to verify the consistence of the kinematic equations.

The procedure is based on the execution of a sequence of tests that makes use of a previously known and calibrated standard, which is a square of 4x4m in length. It is mentioned in [3] the need to construct the square near a reference wall so that the robotic base is always positioned at the same starting point, ensuring consistency in the tests. In this study, the robot was developed with four local reference points, allowing the measurement of initial, intermediary, and final position without any previous standard.

The test is performed at the clockwise and counterclockwise orientation, five times for each side, and at the end of each run, the error is calculated using Equations 1 and 2. The terms *Xabs* and *Yabs* are the base stop positions after each run, with respect to the coordinate starting point X and Y, respectively. The measurements of these terms are performed manually, usually using a caliper or a

laser gauge. The terms *Xcalc* and *Ycalc* are the base stop position calculated by odometry, at the X and Y coordinates, respectively.

$$eX = Xabs - Xcalc \tag{1}$$

$$eY = Yabs - Ycalc \tag{2}$$

Obtaining the data sets, the averages for the X and Y axes are calculated using the Equations 3 and 4. The objective of using the averages of the acquired values is to decrease the dispersion of the points arising from non-systematic errors inherent to the test environment. The variable n is the number of tests performed for each orientation (usually five times), and i each measurement performed.

$$mX_{cw/ccw} = \frac{1}{n} \times \sum_{i=1}^{n} eX(i) \tag{3}$$

$$mY_{cw/ccw} = \frac{1}{n} \times \sum_{i=1}^{n} eY(i) \tag{4}$$

The test result is obtained by calculating the *Euclidean* distance of the mean points X and Y calculated from the Equations 3 and 4, with respect to the start and end of measurements according to Equations 5 and 6.

$$euc_{cw} = \sqrt{(mX_{cw})^2 + (mY_{cw})^2} \tag{5}$$

$$euc_{ccw} = \sqrt{(mX_{ccw})^2 + (mY_{ccw})^2} \tag{6}$$

3 CALCULATION

Concerning mobile robots such as Fig. 1 shows, all movement and constrain are imposed by wheels. Hence the calculation begins observing the omnidirectional wheel such as Fig. 4 shows, inserted in a *Cartesian* plane to show global references. The elements X, Y and O refer to a *Cartesian* plane positioned at the center of rotation of the robotic base. x', y' and o' refer to a *Cartesian* plane positioned at the wheel's center of rotation, where α_i is the angle of the passive elements in relation to the wheel's center of rotation and, β_i is the angle formed between the imaginary lines $\overline{oo'}$ and $\overline{ox}$, where i is the identification of each wheel.

The wheel placement relative to the xoy plane is described by (l_{ix}, l_{iy} and θ_i) where:

$$l_{ix} = l_i \cos(\beta_i) \tag{7}$$

$$l_{iy} = l_i \sin(\beta_i) \tag{8}$$

Fig. 5 shows the local references of each wheel, where r is the wheel radius in meters[m], V_{ir} is the linear wheel roller center speed vector in meters per second[m/s] and ω_i the wheel rotation speed relative to the center of rotation in rad per second[rad/s].

Beginning the analysis, first converting angular to the linear displacement of the wheel[1]:

[1]Omnidirectional overlapping parallel wheels generally are designed to present α_i value like 90°, and due to it V_{ir} will always be 0. Otherwise, when it is not the case, Equation 9 shall be used

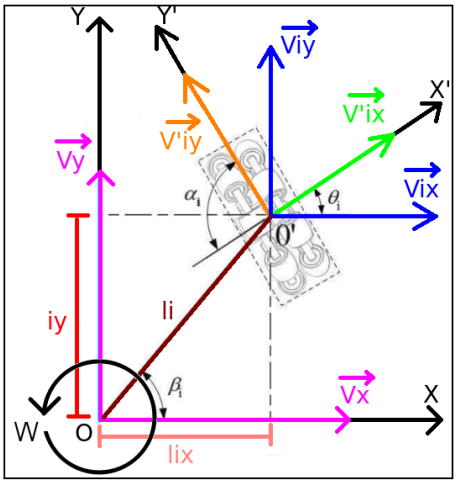

Figure 4: Omnidirectional wheel described as global references. Pink vectors illustrate the base center of rotation displacement, and blue vectors illustrate the wheel center of rotation displacement, both referred to as global frames. Green and orange vectors illustrate the wheel center of rotation referred to as the local frame. Modified from: [19].

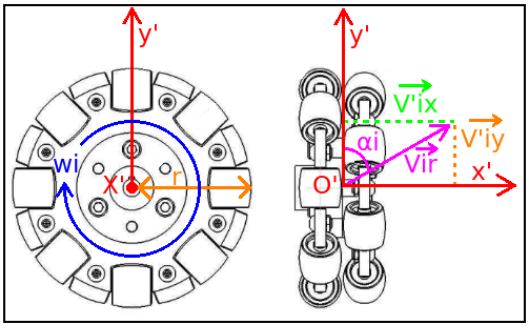

Figure 5: Omnidirectional wheel described as local references. Blue vector illustrates wheel rotational speed and pink vector wheel linear displacement. Depending on wheel construction, α_i can be different than 90°, which generates the need for green and orange vectors. Modified from: [19].

$$\begin{aligned} V_{ir} &= \frac{\Delta s}{\Delta t} \\ &= \frac{2\pi}{t} r \\ &= wi \times r \end{aligned} \tag{9}$$

Due to α_i value, V_{ir} vector must be decomposed into perpendicular components in order to analyze wheel displacement on local reference plane (X', Y'), when the *Cartesian* reference is located on wheel center of rotation such as Fig. 5 shows:

$$\begin{cases} V'_{ix} = \sin(\alpha i) \times V_{ir} \\ V'_{iy} = \cos(\alpha i) \times V_{ir} + (wi \times r) \end{cases}$$

$$\begin{bmatrix} V'_{ix} \\ V'_{iy} \end{bmatrix} = Ki1 \begin{bmatrix} wi \\ V_{ir} \end{bmatrix}, Ki1 = \begin{bmatrix} 0 & \sin(\alpha i) \\ r & \cos(\alpha i) \end{bmatrix} \quad (10)$$

Once V'_{ix} and V'_{iy} components are calculated to wheel reference, it is needed to transfer the *Cartesian* reference to the base center of speed, such as Fig. 4 shows, to measure base displacement due to individual contribution of each wheel.

$$\begin{cases} V_{ix} = (\cos(\theta i) \times V'_{ix}) - (\sin(\theta i) \times V'_{iy}) \\ V_{iy} = (\sin(\theta i) \times V'_{ix}) + (\cos(\theta i) \times V'_{iy}) \end{cases}$$

$$\begin{bmatrix} V_{ix} \\ V_{iy} \end{bmatrix} = Ki2 \begin{bmatrix} V'_{ix} \\ V'_{iy} \end{bmatrix}, Ki2 = \begin{bmatrix} \cos(\theta i) & -\sin(\theta i) \\ \sin(\theta i) & \cos(\theta i) \end{bmatrix} \quad (11)$$

Replacing 10 in 11:

$$\begin{bmatrix} V_{ix} \\ V_{iy} \end{bmatrix} = Ki2 \times Ki1 \times \begin{bmatrix} wi \\ V_{ir} \end{bmatrix} \quad (12)$$

As shows Fig. 4 each wheel is located β_i degrees from base reference, and l_i meters from base center of rotation point. This point displacements (V_x and V_y) will suffer the displacement due to V_{ix}, V_{iy} components, based on β_i and l_i positions. Therefore, V_{ix} and V_{iy} components are over again calculated, but this time observing the wheel position when viewed by the overall system components.

$$l_{ix} = \cos(\beta i) \times l_i \quad (13)$$

$$l_{iy} = \sin(\beta i) \times l_i \quad (14)$$

Representing V_{ix} and V_{iy} on global reference, where ω is rotational displacement of base center point O in rad per second [rad/s]

$$\begin{cases} V_{ix} = V_x + (V_y \times 0) - \omega \times l_{iy}) \\ V_{iy} = (V_x \times 0) + V_y + \omega \times l_{ix}) \end{cases}$$

$$\begin{bmatrix} V_{ix} \\ V_{iy} \end{bmatrix} = Ki3 \begin{bmatrix} V_x \\ V_y \\ \omega \end{bmatrix}, Ki3 = \begin{bmatrix} 1 & 0 & -l_{iy} \\ 0 & 1 & l_{ix} \end{bmatrix} \quad (15)$$

Matching 12 and 15:

$$Ki2 \times Ki1 \times \begin{bmatrix} wi \\ V_{ir} \end{bmatrix} = Ki3 \times \begin{bmatrix} V_x \\ V_y \\ \omega \end{bmatrix} \quad (16)$$

$$\begin{bmatrix} wi \\ V_{ir} \end{bmatrix} = Ki1^{-1} \times Ki2^{-1} \times Ki3 \times \begin{bmatrix} V_x \\ V_y \\ \omega \end{bmatrix} \quad (17)$$

Calculating the inverse matrices:

$$Ki1^{-1} = \begin{bmatrix} \left(-\frac{1}{r} \times \frac{\cos(\alpha i)}{\sin(\alpha i)}\right) & \frac{1}{r} \\ \frac{1}{\sin(\alpha i)} & 0 \end{bmatrix} \quad (18)$$

$$Ki2^{-1} = \begin{bmatrix} \cos\theta i & \sin\theta i \\ -\sin\theta i & \cos\theta i \end{bmatrix} \quad (19)$$

Calculating the multiplication between the matrices, considering $\gamma i = \theta i - \alpha i$ and replacing l_{ix} 13 and l_{iy} 14:

$$\begin{bmatrix} wi \\ V_{ir} \end{bmatrix} = \frac{1}{-r \sin(\alpha i)} \times T \times \begin{bmatrix} V_x \\ V_y \\ \omega \end{bmatrix} \quad (20)$$

$$T = \begin{bmatrix} \cos(\gamma i) & \sin(\gamma i) & l_i \sin(\gamma i - \beta i) \\ -r\cos(\theta i) & -r\sin(\theta i) & (r \times l_i \times \sin(\theta i - \beta i)) \end{bmatrix}$$

There is no way to control the rotational speed of the passive rollers due to omnidirectional wheel construction. Hence only the rotational speed of the center of each wheel (ω_i) is observed to determine the displacement vectors of the center of the base $[V_x V_y w]^T$. The equation is expressed as a function of ω generalizing for all wheels.

$$\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} = \frac{1}{-\begin{bmatrix} r_1 & 0 & 0 \\ 0 & r_2 & 0 \\ 0 & 0 & r_3 \end{bmatrix}} E \begin{bmatrix} V_x \\ V_y \\ w \end{bmatrix} \quad (21)$$

$$E = \begin{bmatrix} \frac{\cos(\gamma 1)}{\sin(\alpha 1)} & \frac{\sin(\gamma 1)}{\sin(\alpha 1)} & \frac{l_1 \sin(\gamma 1 - \beta 1)}{\sin(\alpha 1)} \\ \frac{\cos(\gamma 2)}{\sin(\alpha 2)} & \frac{\sin(\gamma 2)}{\sin(\alpha 2)} & \frac{l_2 \sin(\gamma 2 - \beta 2)}{\sin(\alpha 2)} \\ \frac{\cos(\gamma 3)}{\sin(\alpha 3)} & \frac{\sin(\gamma 3)}{\sin(\alpha 3)} & \frac{l_3 \sin(\gamma 3 - \beta 3)}{\sin(\alpha 3)} \end{bmatrix}$$

In order to define base position in *Cartesian* coordinate $[X_c, Y_c]$, center position vector must be observed based on linear displacement velocity of each wheel $[v_1 v_2 v_3]^T$, as a function of base center displacement vectors $[V_x V_y w]^T$.

$$v = r \times w$$

$$\begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} r_1 & 0 & 0 \\ 0 & r_2 & 0 \\ 0 & 0 & r_3 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix}$$

$$\begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} r_1 & 0 & 0 \\ 0 & r_2 & 0 \\ 0 & 0 & r_3 \end{bmatrix} \frac{1}{-\begin{bmatrix} r_1 & 0 & 0 \\ 0 & r_2 & 0 \\ 0 & 0 & r_3 \end{bmatrix}} E \begin{bmatrix} V_x \\ V_y \\ w \end{bmatrix}$$

$$\begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = -E \begin{bmatrix} V_x \\ V_y \\ w \end{bmatrix} \tag{22}$$

Fig. 6 shows the state of the omnidirecional three wheeled robot in *Cartesian* coordinate $[X_c, Y_c]$, where V_x and V_y are vectors to represent the center speed of the base, and θc the angle between axis X_c and vector V_x.

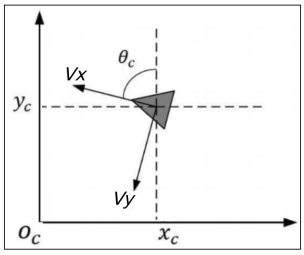

Figure 6: Omnidirectional robot state in *Cartesian* coordinate. Modified from: [19]

Thus:

$$\begin{bmatrix} V_x \\ V_y \\ \omega \end{bmatrix} = H^{-1} \begin{bmatrix} X_c \\ Y_c \\ \omega_c \end{bmatrix} \tag{23}$$

$$H = \begin{bmatrix} \cos(\theta_c) & -\sin(\theta_c) & 0 \\ \sin(\theta_c) & \cos(\theta_c) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Finally replacing 23 in 22, the inverse kinematic equation is:

$$\begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = -EH^{-1} \begin{bmatrix} X_c \\ Y_c \\ \omega_c \end{bmatrix} \tag{24}$$

4 IMPLEMENTATION

The implementation step concerned hardware and mechanical components to develop the omnidirectional robot, such as Fig. 1 shows. An embedded computer has CPU ARM Cortex-A53 - 1.2GHz, GPU Broadcom VideoCore IV, 1GB DDR2 memory, and 16GiB microSd storage space.

To provide rotational displacement to each wheel and additionally measure angular speed to apply in equation 24, three-step motors Nema 17 and three 600ppr incremental encoders are the essential mechanical components.

Three dedicated microcontrollers were used to simplify the software development and provide hardware modularization, one to each displacement group such as Fig 7 shows. Arduino NANO boards based on ATMega328P microcontroller were selected.

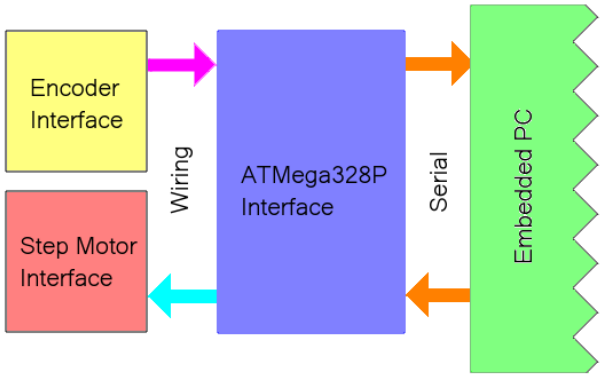

Figure 7: Hardware interface to each displacement group. The red box is comprehended as Step Motor and Power Drive interface, while the yellow box illustrates the interface with incremental encoder hardware. Both interfaces require hardware and software low-level access.

Software implementation was performed using well known Robot Operating System (ROS) framework [14], due to programming language flexibility, easy-to-use libraries, variety of tools, and open-source access. Concerning the entire software implementation, it is possible to cite at least *rosserial* library [13] and *rviz* tool [21] as fundamental to the development.

5 RESULTS

In order to verify equation 24 effectiveness and hardware and software development, *UMBmark* test procedure was performed such as presented in secton 2.5.

Robot parameters as wheel diameter and the distance between wheel center and robot center were set, considering standard components values, that in this case are: 82,5mm to wheel diameter and 113mm to distance between center points.

Fig. 8 shows calculated *Euclidian* distance errors to clockwise and counterclockwise movement, and Fig. 9 real position after each run performed. First evaluating *Euclidian* distance results, it is possible to say that in both clockwise and counterclockwise conditions, the difference between each measurement is less than 3mm, what is very small when compared with test path performed (4x4m square).

Considering real position error after each run it is possible to observe that to counterclockwise condition, X positioning error is bigger and positioning error is smaller when compared with clockwise condition. This behavior is strongly related to the lack of calibration of the base parameters, and tends to be constant in all test cases. However, even in this case the positioning error is very small when compared with test path.

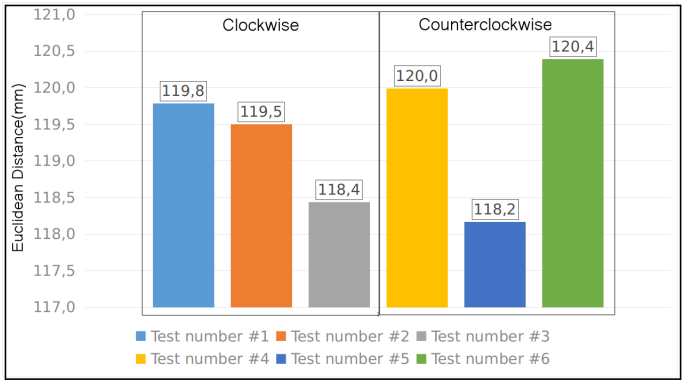

Figure 8: Euclidian distance error analysis. Both clockwise and counterclockwise conditions, the difference between each measurement is less than 3mm.

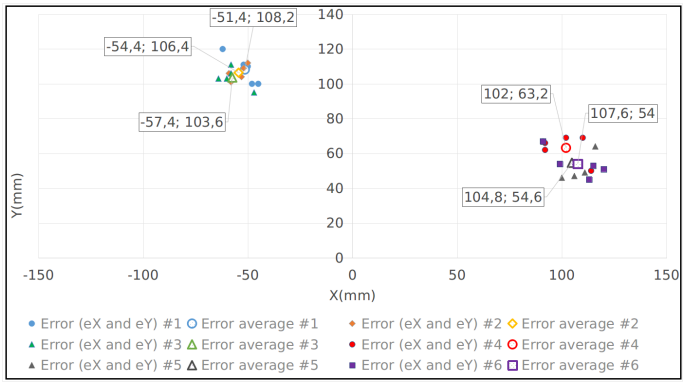

Figure 9: *UMBmark* test results. Observed positioning error highlights the lack of calibration of the base.

6 CONCLUSION

This research work presented the necessary steps for the development of the kinematic equation of an omnidirectional robotic base with three wheels, observing only the characteristics and arrangement of the components in the construction of the mechanical structure. It also presented the validation of the equation through the use of a real robotic base, using the *UMBMark* test procedure.

The results obtained demonstrated the effectiveness of the developed equation, given the small error observed, even for the condition of the base without calibration in which the tests were performed and the repeatability of the results obtained for each evaluated condition.

Finally, this research contributes to the development of omnidirectional robotic bases, considering all kinematics involved, and explains the application of the mathematical concepts necessary to deduce kinematic equations. This theme is relevant to any mobile robotic system, and in the vast majority of cases, it's poorly explained in detail and not wholly understood by the newcomers.

REFERENCES

[1] Mohammad O. A. Aqel, M. H. Marhaban, M. I. Saripan, and N. Ismail. 2016. Review of visual odometry: types, approaches, challenges, and applications. *SpringerPlus* 5 (2016).

[2] Houssem Benseddik, Oualid Djekoune, and Mahmoud Belhocine. 2014. SIFT and SURF Performance Evaluation for Mobile Robot-Monocular Visual Odometry. *Journal of Image and Graphics* 2 (01 2014), 70–76. https://doi.org/10.12720/joig.2.1.70-76

[3] J. Borenstein and L. Feng. 1995. UMBmark: A Benchmark Test for Measuring Odometry Errors in Mobile Robots. In *SPIE Conference on Mobile Robots*. Philadelphia - USA, 1–12.

[4] Kok Seng Chong and Lindsay Kleeman. 1997. Accurate Odometry and Error Modelling for a Mobile Robot. In *IEEE International Conference on Robotics and Automation*. Albuquerque - USA, 2783–2788.

[5] Ioan Doroftei, Victor Grosu, and V. Spinu. 2007. *Omnidirectional Mobile Robot - Design and Implementation*. https://doi.org/10.5772/5518

[6] Hugh Durrant-Whyte and Tim Bailey. [n.d.]. *Simultaneous Localization and Mapping(SLAM): Part I The Essential Algorithms*. Technical Report.

[7] R. Haendel. 2005. Design of an omnidirectional universal mobile platform. *Journal of Educational Psychology - J EDUC PSYCHOL* (01 2005).

[8] Javier Javier Moreno, E. Clotet, Ruben Lupiañez, Marcel Tresanchez, Dani Martinez, Tomàs Pallejà, Jordi Casanovas, and Jordi Palacín. 2016. Design, Implementation and Validation of the Three-Wheel Holonomic Motion System of the Assistant Personal Robot (APR). *Sensors* 16 (10 2016), 1658. https://doi.org/10.3390/s16101658

[9] Changbae Jung and Woojin Chung. 2012. Accurate Calibration of Two Wheel Differential Mobile Robots by Using Experimental Heading Erros. In *IEEE International Conference on Robotics and Automation*. Minnesota - USA.

[10] Serdar Kucuk and Z. Bingul. 2006. *Robot Kinematics: Forward and Inverse Kinematics*. https://doi.org/10.5772/5015

[11] Logitech. 2020. C270 Videochamada de HD 720p plug and play. https://www.logitech.com/pt-br/product/hd-webcam-c270. Acessado em 26/10/2020.

[12] Annalisa Milella, Giulio Reina, and Roland Siegwart. 2006. Computer Vision Methods for Improved Mobile Robot State Estimation in Challenging Terrains. *Journal of Multimedia* 1 (11 2006). https://doi.org/10.4304/jmm.1.7.49-61

[13] R. Reignier. 2018. rosserial - ROS Wiki. http://wiki.ros.org/rosserial. Acessado em 05/11/2020.

[14] ROS.org. 2020. ROS.org|Powering the world's robots. https://www.ros.org/. Acessado em 03/11/2020.

[15] Davide Scaramuzza and Friedrich Fraundorfer. 2011. Visual Odometry [Tutorial]. *IEEE Robot. Automat. Mag.* 18 (12 2011), 80–92. https://doi.org/10.1109/MRA.2011.943233

[16] R. Siegwart and Illah R. Nourbakhsh. 2004. *Introduction to Autonomous Mobile Robots*. The MIT Press.

[17] D. L. Tomasi and E. Todt. 2017. Rotational odometry calibration for differential robot platforms. In *2017 Latin American Robotics Symposium (LARS) and 2017 Brazilian Symposium on Robotics (SBR)*. 1–6. https://doi.org/10.1109/SBR-LARS-R.2017.8215315

[18] VEX. 2020. Wheels - VEX Robotics. https://www.vexrobotics.com/edr-wheels.html. Acessado em 26/10/2020.

[19] Chengcheng Wang, Xiaofeng Liu, Xianqiang Yang, Fang Hu, Aimin Jiang, and Chenguang Yang. 2018. Trajectory Tracking of an Omni-Directional Wheeled Mobile Robot Using a Model Predictive Control Strategy. *Applied Sciences* 8 (02 2018), 231. https://doi.org/10.3390/app8020231

[20] Avi Weiss, R.G. Langlois, and M. Hayes. 2014. Dynamics and vibration analysis of the interface between a non-rigid sphere and omnidirectional wheel actuators. *Robotica* 72 (05 2014). https://doi.org/10.1017/S0263574714001088,

[21] W. Woodall. 2018. rviz - ROS Wiki. http://wiki.ros.org/rviz. Acessado em 05/11/2020.