



# OptiOdom: a Generic Approach for Odometry Calibration of Wheeled Mobile Robots

Ricardo B. Sousa<sup>1,2</sup> · Marcelo R. Petry<sup>2</sup> · Paulo G. Costa<sup>1,2</sup> · António Paulo Moreira<sup>1,2</sup>

Received: 9 June 2021 / Accepted: 5 April 2022 / Published online: 8 June 2022  
© The Author(s), under exclusive licence to Springer Nature B.V. 2022

## Abstract

Odometry calibration adjusts the kinematic parameters or directly the robot's model to improve the wheeled odometry accuracy. The existent literature considers in the calibration procedure only one steering geometry (differential drive, Ackerman/tricycle, or omnidirectional). Our method, the OptiOdom calibration algorithm, generalizes the odometry calibration problem. It is developed an optimization-based approach that uses the improved Resilient Propagation without weight-backtracking (iRprop-) for estimating the kinematic parameters using only the position data of the robot. Even though a calibration path is suggested to be used in the calibration procedure, the OptiOdom method is not path-specific. In the experiments performed, the OptiOdom was tested using four different robots on a square, arbitrary, and suggested calibration paths. The OptiTrack motion capture system was used as a ground-truth. Overall, the use of OptiOdom led to improvements in the odometry accuracy (in terms of maximum distance and absolute orientation errors over the path) over the existent literature while being a generalized approach to the odometry calibration problem. The OptiOdom and the methods from the literature implemented in the article are available in GitHub as an open-source repository.

**Keywords** Calibration · Dead reckoning · Mobile robots · Robot kinematics · Wheeled odometry

## 1 Introduction

Localization is one of the most critical problems in autonomous navigation. Indeed, autonomous navigation is only achieved if it is possible to determine the robot's pose in the environment in which is inserted. Two

basic localization methods commonly applied together using sensor fusion are absolute and relative localization. Absolute localization usually depends on map matching, identification of active or passive landmarks or beacons. Relative localization is usually based on wheel, visual and/or laser odometry [3, 12, 23, 29].

Wheel odometry uses the robot's kinematic model and the displacement of the wheels to estimate the robot's pose. The estimated pose is relative to a previous instant. Usually, wheel odometry is combined with other localization methods using sensor fusion techniques. Furthermore, the kinematic model is specific to each steering geometry. The model's parameters are the physical dimensions of the robot, such as diameters of the wheels, steering angle offsets, among others. [31]. As for the wheels displacement, it is estimated, e.g., from the measurement of wheel revolutions (usually obtained from optical encoders) and/or steering angles. In comparison with other localization methods, wheel odometry is computationally inexpensive, high sampling rate, and easy to apply in real-time [3].

A well-known disadvantage of odometry is the accumulation of errors. Some of these errors are systematic, i.e., they are deterministic and constantly accumulate over time

---

This work is financed by the ERDF – European Regional Development Fund through the Operational Programme for Competitiveness and Internationalisation – COMPETE 2020 Programme, and by National Funds through the Portuguese funding agency, FCT – Fundação para a Ciência e a Tecnologia, within project SAICTPAC/0034/2015- POCI-01-0145-FEDER-016418. This work is also financed by the ERDF – European Regional Development Fund through the Operational Programme for Competitiveness and Internationalisation – COMPETE 2020 Programme within project POCI-01-0145-FEDER-006961, and by National Funds through the FCT – Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) as part of project UID/EEA/50014/2013.

---

✉ Ricardo B. Sousa  
up201503004@fe.up.pt

(e.g., uncertainty of the robot's kinematic parameters) [3, 16]. Calibration can be used to reduce the uncertainty of the kinematic parameters and the effect of systematic errors.

So, odometry calibration is essential to improve the accuracy of wheel odometry. The main goal is to estimate the kinematic parameters of the robot, increasing the accuracy of the odometry's pose estimation. In the literature, several methods were proposed for the most frequent steering geometries: differential drive ([1–6, 8, 13, 16, 17, 25–27, 34, 37]), Ackerman/tricycle ([7, 11, 18, 19]), and omnidirectional ([14, 22, 24]). The existent works on odometry calibration illustrate three main trends: path-specific algorithms with closed-form equations, optimization-based methods, and methods that implement Augmented Kalman Filters (AKF). However, all of them only focus on estimating the kinematic parameters for a specific steering geometry.

This paper presents the OptiOdom odometry calibration method. OptiOdom is an optimization-based method, and has the following contributions:

1. generic odometry calibration algorithm (tested in real environments for the differential drive, Ackerman/tricycle, and omnidirectional steering geometries);
2. comparison of OptiOdom with popular calibration methods for the differential drive, Ackerman/tricycle, and omnidirectional steering geometries.

Our previous works [32, 33] are the basis for the discussion of the related work, and provides a comprehensive literature review on odometry calibration methods. Furthermore, to the best of our knowledge, the OptiOdom is the only odometry calibration method that intends to calibrate simultaneously three different steering geometries (differential drive, Ackerman/tricycle, and omnidirectional).

The rest of the paper is organized as follows. Section 2 presents the related work. Section 3 defines the kinematics for the differential drive, Ackerman/tricycle, and omnidirectional steering geometries. Section 4 formulates the OptiOdom method for wheeled odometry calibration. Section 5 analyses the results obtained from the experiments made. Section 6 presents the conclusions of the work performed.

## 2 Related Work

### 2.1 Path-specific with Closed-form Equations

Borenstein and Feng [3] was the pioneer method in odometry calibration. The authors proposed the University of Michigan Benchmark (UMBmark) for the differential drive

steering geometry. The method used a 4x4m bi-directional square as a calibration path. Closed-form equations formulated in [3] estimate the kinematic parameters. Similar to [3], Jung and Chung [17] used a square calibration path, but changed its length to 2m. Also, new closed-form equations were formulated for the kinematic parameters. The authors determined the square length by demonstrating that the path size influences the calibrated odometry accuracy (for the robot used in their experiments). However, it was not proven that the proposed path size effectively improves the odometry for all robots when compared to the 4x4m square calibration path. The methodology proposed by Tomasi and Todt [34] is also based on UMBmark [3]. However, instead of using a square path, it uses rotational motions only: 180° and 360° rotations in both clockwise (CW) and counterclockwise (CCW) directions.

Bostani et al. [4] proposed moving a differential drive robot back and forward, with an intermediate 180° rotation (CW and CCW directions). The closed-form equations defined in [4] considered the initial, intermediate and final position of the robot through the path. Maddahi et al. [25] also used a straight-line as a calibration path, but only with forward motion. The final position is used to estimate the kinematic parameters of differential drive robots. The adaptation of [25] for omnidirectional robots was formulated by Maddahi et al. [24]. As an alternative to square and straight-line calibration paths for differential drive robots, Abbas et al. [1] proposed the Bi-directional Circular Path Test (BCPT). The measurements from the circumferences diameters (after each CW and CCW runs) allow the estimation of the kinematic parameters.

As for the Ackerman/tricycle steering geometry, De Cecco [7] proposed a self-calibration algorithm. The method required for the robot to go through a specific path defined in [7]. Both position and orientation of the robot at intermediate and final waypoints were used to estimate the kinematic parameters. The method proposed by Jung et al. [18] is also intended for the Ackerman/tricycle steering geometry. It required the robot to go through a 1.75m straight-line and a 1.75m diameter semicircle to the CW direction, repeating these two motions to finish ideally in the starting pose. The path is also repeated on the CCW direction. The final orientation is used to calibrate the kinematic parameters.

The main problem of path-specific methods with closed-form equations is that they are specific to a given steering geometry. Indeed, either the path or equations are required to be modified to implement the method on other steering geometries. Furthermore, the convergence of the estimated kinematic parameters could depend on the path size and the initial estimation for the kinematic parameters [36].

## 2.2 Optimization-based Methods

As for optimization-based methods, Antonelli et al. [2] was the first to use the linear least-squares algorithm to minimize the final pose (position and orientation) error relative to the absolute and odometry estimations. This method formulated the least-squares algorithm specifically for differential drive robots. This formulation was possible because the velocity of a differential drive robot is linearly related to the motors' angular speed. Due to the use of least-squares, [2] required several runs with different paths to compute an unbiased estimator. It is proposed the use of open and long paths. The adaptation of [2] for omnidirectional robots was formulated by Lin et al. [22]. Given that the velocity of omnidirectional robots is linearly related to the motors' angular speed, [22] also used the least-squares algorithm. In contrast to [2], Lin et al. [22] adjusts directly the kinematic model and not each kinematic parameter. Also, it has more Degrees of Freedom (DoFs) than the robot's kinematic model: 6 vs 4 DoFs and 8 vs 5 DoFs for three and four-wheel omnidirectional robots, respectively. Han et al. [14] also used the linear least-squares algorithm to calibrate the odometry of omnidirectional robots. Han et al. [14] adjusted directly the kinematic model by minimizing the error between the expected robot's velocity with the one measured from the wheels' encoders. However, the method only considered error sources specific to omnidirectional robots (slippage, bearing and/or axle friction, and point contact friction). Goronzy and Hellbrueck [13] implemented a weighted non-linear least-squares algorithm for differential drive robots to estimate the kinematic parameters. The main goal of this method is online odometry calibration for differential drive robots with "noisy" ground-truth data (e.g., ultrawideband or QRpos).

Another work for the differential drive steering geometry that used the linear least-squares algorithm was proposed by Censi et al. [8]. This method performs simultaneously odometry and extrinsic sensor calibration for a laser scanner. However, the two calibrations cannot be performed independently of each other, and it is required for the robot to have constant wheels' velocities in each path segment. Based on [8], Kallasi et al. [19] formulated the odometry and extrinsic sensor calibration independently for tricycle robots. This method implements a linear least-squares algorithm using the orientation displacement as input data. Also, [19] proposed a circular-based calibration path (circular motion with decreasing radius) to generate the input data for the calibration algorithm. Similar to [8], this method requires constant wheels velocities in path segments. The equivalent method of [19] for the Ackerman (front-wheel drive) and dual-drive (Ackerman, but with

rear-wheel drive) steering geometries is formulated in the work of Galasso et al. [11]. Both works [11, 19] did not estimate the distance between front and rear wheels, assuming that this distance is known.

Ivanjko et al. [16] minimized the final pose error for differential drive robots by using an optimization algorithm (it is suggested the Gauss-Newton or Nelder-Mead algorithms). First, [16] estimated the diameter of the wheels from the straight-line path. Then, the distance between the two wheels is estimated from the 180° rotation motions in both CW and CCW. This calibration path specific for differential drive robots assumed that linear motion is affected mostly by unequal wheels diameters and rotational motion by distance between wheels. Mondal et al. [27] developed a terminal iterative learning control algorithm to calibrate differential drive robots. This method proposes only one run through an arbitrary path.

The optimization-based methods found in the literature are all specific to one steering geometry. Also, only Mondal et al. [27] and Censi et al. [8] compared their works with another odometry calibration method. The comparison in both works was to UMBmark [3]. Mondal et al. [27] improved the odometry accuracy of a differential drive robot over [3]. Although Censi et al. [8] obtained similar odometry accuracy compared to [3], the method also estimated the extrinsic parameters of a laser scanner.

## 2.3 Augmented Kalman Filter

The methods based on Augmented Kalman Filters (AKF) implement Extended Kalman Filters (EKF) that simultaneously estimate the robot pose and its kinematic parameters [26]. These methods can perform the calibration procedure while the robot is performing tasks, i.e., performing online odometry calibration without the need of the robot going through specific paths. The AKF-based methods differ mostly in terms of the filter's update step. Martinelli et al. [26] proposed an AKF that needed a known map (e.g., landmarks) to fuse the laser scanner with odometry data and update the AKF. Caltabiano et al. [5] required Differential Global Positioning System (DGPS) data to update the filter. Although [5] is similar to [26], the data fusion in [5] also considered attitude and heading measurements from an inertial measurement unit. Lastly, Yun et al. [37] did not require external sensors given that it is focused on "home positioning", i.e., the robot returning to its "home" or charging spot after it goes through an arbitrary path. The loop closure updates the AKF and corrects the kinematic parameters.

Even though the AKF-based methods can perform online odometry calibration without the need of the robot going through specific paths, the works found

were only formulated for differential drive robots. Also, only Caltabiano et al. [5] performed comparisons with UMBmark [3] obtaining better odometry accuracy over the last. Indeed, the rest of the AKF-based methods did not make any comparisons with other works proposed in the literature.

### 3 Kinematics

This section presents the forward and inverse kinematics of the steering geometries considered in this paper: differential drive, Ackerman/tricycle, and omnidirectional (three and four wheels). The forward kinematics are required to formulate the OptiOdom odometry calibration method. As for the inverse kinematics, these are presented here for reproducibility reasons (relative to the experimental results). The inverse kinematics formulated in this section were used to perform all the experiments presented in this paper.

#### 3.1 Differential Drive Steering Geometry

A differential drive robot is illustrated in Fig. 1. The two driving wheels rotate around the center of the robot. One or more ground contact points can be added to improve the stability of the robot. If it is added more than one contact point, it should be ensured that the traction wheel is always in contact with the floor. The type of contact points can be a spherical ball or a caster wheel [31]. The kinematic model of this steering geometry is characterized by the following parameters: wheelbase of the robot ( $b$ ), and the diameters of the left and right wheels ( $D_i$ ).

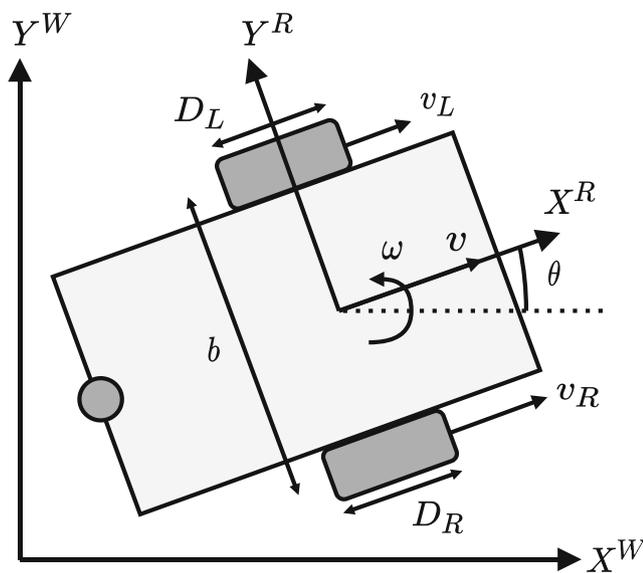


Fig. 1 Differential drive steering geometry

#### 3.1.1 Forward Kinematics

The linear displacement of a wheel  $i$  at a time instant  $k$  ( $\Delta d_{i,k}$ ) is defined in Eq. 1. It depends on the diameter of the wheel ( $D_i$ ), the gear reduction ratio ( $n$ ), the encoders' resolution ( $C_e$ ), and the impulses increment ( $\#i_{i,k}$ ) measured by the respective encoder.

$$\Delta d_{i,k} = \frac{\pi D_i}{nC_e} \#i_{i,k} \tag{1}$$

Next, Eqs. 2 and 3 are relative to the linear ( $\Delta d_k$ ) and angular ( $\Delta \theta_k$ ) displacements of the robot on its frame ( $\{X^R, Y^R\}$ ), respectively.

$$\Delta d_k = \frac{\Delta d_{R,k} + \Delta d_{L,k}}{2} \tag{2}$$

$$\Delta \theta_k = \frac{\Delta d_{R,k} - \Delta d_{L,k}}{b} \tag{3}$$

Finally, the robot's pose relative to the world frame ( $\{X^W, Y^W\}$ ) can be computed by the odometry equations using the centered discrete approximation presented in [31]. This approximation is illustrated in the set of Eq. 4 for differential drive robots.

$$\begin{aligned} x_k &= x_{k-1} + \Delta d_k \cdot \cos(\theta_{k-1} + \Delta \theta_k/2) \\ y_k &= y_{k-1} + \Delta d_k \cdot \sin(\theta_{k-1} + \Delta \theta_k/2) \\ \theta_k &= \theta_{k-1} + \Delta \theta_k \end{aligned} \tag{4}$$

#### 3.1.2 Inverse Kinematics

The inverse kinematics is necessary to control the robot through a specific path. As illustrated in Eq. 5, the linear velocities of the right and left ( $v_{i,k}$ ) wheels can be formulated as dependent on the robot's linear ( $v_k$ ) and angular ( $\omega_k$ ) velocities at a certain time instant  $k$  [31].

$$\begin{aligned} v_{R,k} &= v_k + \frac{\omega_k b}{2} \\ v_{L,k} &= v_k - \frac{\omega_k b}{2} \end{aligned} \tag{5}$$

Lastly, the motors' angular velocity ( $\dot{\phi}_{i,k}$ ) is computed depending on the desired wheels' linear velocity ( $v_{i,k}$ ), diameter of the wheel ( $D_i$ ), and on the gear reduction ratio ( $n$ ), as illustrated in Eq. 6.

$$\dot{\phi}_{i,k} = \frac{2n}{D_i} \cdot v_{i,k} \tag{6}$$

#### 3.2 Ackerman/Tricycle Steering Geometry

The Ackerman and tricycle steering geometries are very similar. An Ackerman robot can have multiple wheels, but

must have only a single Instantaneous Center of Rotation (ICR) to avoid slipping. A single ICR can be achieved by having different steering angles on the steering wheels. As for the tricycle geometry, it always has only one ICR given that tricycle robots have only one steering wheel. So, these two steering geometries are kinematically equivalent [30, 31].

In this paper, we analyze the tricycle geometry, illustrated in Fig. 2. The kinematic parameters of tricycle robots are the diameter of the front-driven wheel ( $D$ ), the steering angle offset ( $\alpha_{off}$ ), and the distance between the front and rear wheels ( $l$ ).

### 3.2.1 Forward Kinematics

The linear displacement of the front wheel ( $\Delta d_{f,k}$ ) is computed as in Eq. 1. Next, the wheel’s linear displacement and steering angle ( $\alpha_k$ ) compute the robot’s linear ( $\Delta d_k$ ) and angular ( $\Delta \theta_k$ ) displacement, as illustrated in Eqs. 7 and 8 [30]. Note that the steering angle offset ( $\alpha_{off}$ ) influences both displacements, offsetting the value of the real steering angle of the robot ( $\alpha_k + \alpha_{off}$ ).

$$\Delta d_k = \cos(\alpha_k + \alpha_{off}) \cdot \Delta d_{f,k} \tag{7}$$

$$\Delta \theta_k = \frac{\sin(\alpha_k + \alpha_{off})}{l} \cdot \Delta d_{f,k} \tag{8}$$

The robot’s pose can be estimated using the same odometry equations defined in Eq. 4 for the differential drive steering geometry [31].

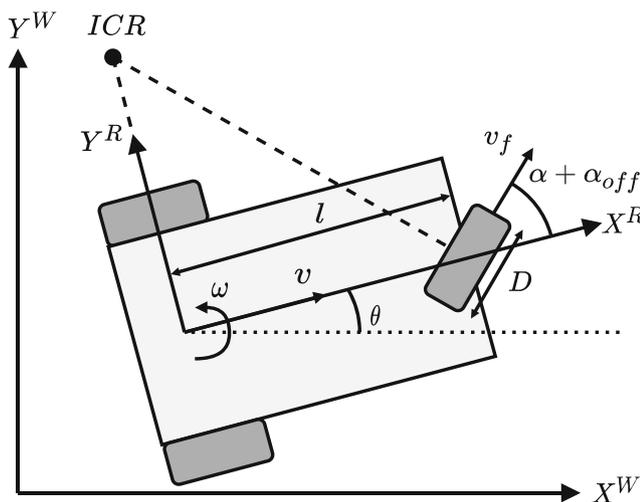


Fig. 2 Tricycle steering geometry

### 3.2.2 Inverse Kinematics

The steering angle ( $\alpha_k$ ) and the wheel’s linear velocity ( $v_f$ ) are computed by Eqs. 9 and 10, respectively. Note that, in the case of  $v_k = 0$ , the steering angle have two possible values. The reference for the steering angle is set by the closest angle relative to the current one. The angular velocity of the front wheel ( $\dot{\phi}_{f,k}$ ) can be calculated as in Eq. 6 [31].

$$\alpha_k = \begin{cases} \alpha_{k-1} & \text{if } \omega_k = 0 \wedge v_k = 0 \\ \pm \frac{\pi}{2} - \alpha_{off} & \text{if } v_k = 0 \\ \arctan\left(\frac{l\omega_k}{v_k}\right) - \alpha_{off} & \text{other cases} \end{cases} \tag{9}$$

$$v_{f,k} = \begin{cases} \text{sign}\{\alpha_k + \alpha_{off}\} \cdot l\omega_k & \text{if } v_k = 0 \\ \frac{v_k}{\cos(\alpha_k + \alpha_{off})} & \text{other cases} \end{cases} \tag{10}$$

## 3.3 Omnidirectional Steering Geometry

The three and four-wheeled omnidirectional robots studied in this paper are illustrated in Fig. 3a and b, respectively. In terms of the kinematic model, the diameter of the wheels ( $D_i$ ) and the distance between the robot’s geometrical center and the wheels ( $l$ ) characterize the model for the three-wheeled omnidirectional robot. The difference between the three and the four-wheeled robot’s model is that the last does not depend directly on  $l$ . The four-wheeled geometry is characterized by the sum of the distance between the front and rear wheels ( $l_1$ ) plus the distance between the left and right wheels ( $l_2$ ).

### 3.3.1 Forward Kinematics

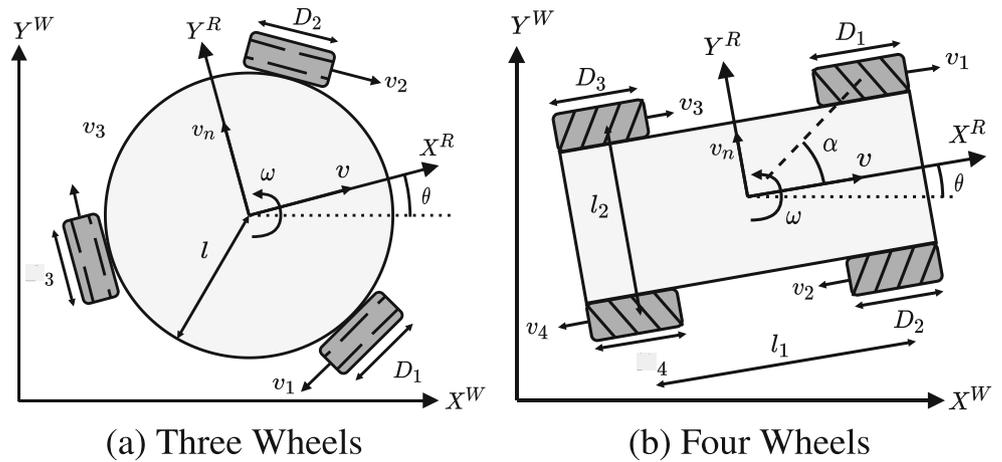
The kinematic model of an omnidirectional robot can be formulated through the motion constraints of the individual wheels. As analysed in Siegwart et al. [31] and illustrated in Fig. 4, an omnidirectional wheel can be characterized by four parameters:

- $(l_i, \alpha_i)$ : polar coordinates of the wheel  $i$  relative to the robot coordinate frame;
- $\beta_i$ : angle of the wheel  $i$  plane relative to the robot chassis;
- $\gamma_i$ : angle between the roller axis and the wheel  $i$  plane.

Furthermore, Tables 1 and 2 defines the four parameters for both three and four-wheeled omnidirectional robots, respectively. Note that the indexes  $i$  are the same as the ones illustrated in Fig. 3a and b.

Then, the inverse kinematics defined by the matrix  $J_{omni_m}^{-1}$  is formulated with the four parameters of the omnidirectional wheel, as illustrated in Eq. 11 [24, 31].

**Fig. 3** Omnidirectional steering geometry



Also, note that Eqs. 12 and 13 define the matrix  $J_{omni_m}^{-1}$  for three and four-wheeled omnidirectional robots, respectively.

in Eqs. 15 and 16. These two matrices were computed using the Monroe-Penrose pseudo-inverse.

$$J_{omni_m}^{-1} = \begin{bmatrix} \frac{s_{\alpha_1+\beta_1+\gamma_1}}{c_{\gamma_1}} & \frac{-c_{\alpha_1+\beta_1+\gamma_1}}{c_{\gamma_1}} & \frac{-lc_{\beta_1+\gamma_1}}{c_{\gamma_1}} \\ \vdots & \vdots & \vdots \\ \frac{s_{\alpha_m+\beta_m+\gamma_m}}{c_{\gamma_m}} & \frac{-c_{\alpha_m+\beta_m+\gamma_m}}{c_{\gamma_m}} & \frac{-lc_{\beta_m+\gamma_m}}{c_{\gamma_m}} \end{bmatrix} \quad (11)$$

$$\begin{bmatrix} \Delta d_k \\ \Delta d_{n,k} \\ \Delta \theta_k \end{bmatrix} = J_{omni_m} \cdot \begin{bmatrix} \Delta d_{1,k} \\ \vdots \\ \Delta d_{m,k} \end{bmatrix} \quad (14)$$

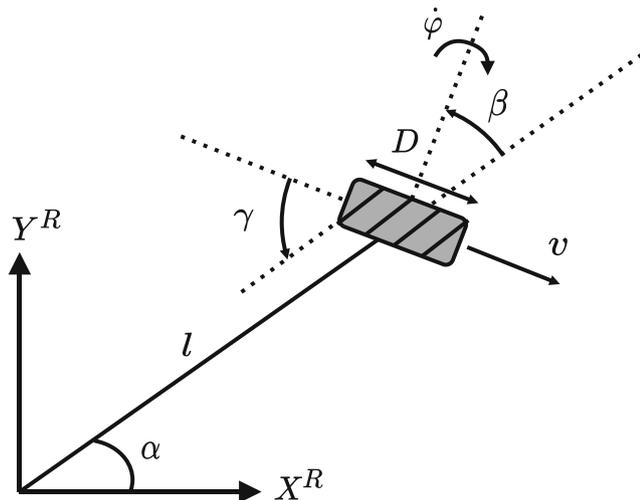
$$J_{omni_3}^{-1} = \begin{bmatrix} -\frac{\sqrt{3}}{2} & -\frac{1}{2} & -l \\ \frac{\sqrt{3}}{2} & -\frac{1}{2} & -l \\ 0 & 1 & -l \end{bmatrix} \quad (12)$$

$$J_{omni_3} = \begin{bmatrix} -\frac{\sqrt{3}}{3} & \frac{\sqrt{3}}{3} & 0 \\ -\frac{1}{3} & -\frac{1}{3} & \frac{2}{3} \\ -\frac{1}{3l} & -\frac{1}{3l} & -\frac{1}{3l} \end{bmatrix} \quad (15)$$

$$J_{omni_4}^{-1} = \begin{bmatrix} 1 & -1 & -\frac{1}{2} \cdot (l_1 + l_2) \\ -1 & -1 & -\frac{1}{2} \cdot (l_1 + l_2) \\ 1 & 1 & -\frac{1}{2} \cdot (l_1 + l_2) \\ -1 & 1 & -\frac{1}{2} \cdot (l_1 + l_2) \end{bmatrix} \quad (13)$$

Next, Eq. 14 illustrates that the linear ( $\Delta d_k$ ), normal ( $\Delta d_{n,k}$ ), and angular ( $\Delta \theta_k$ ) displacements of the robot can be computed using the matrices  $J_{omni_3}$  and  $J_{omni_4}$ , defined

$$J_{omni_4} = \begin{bmatrix} \frac{1}{4} & -\frac{1}{4} & \frac{1}{4} & -\frac{1}{4} \\ -\frac{1}{4} & -\frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ -\frac{1}{2 \cdot (l_1+l_2)} & -\frac{1}{2 \cdot (l_1+l_2)} & -\frac{1}{2 \cdot (l_1+l_2)} & -\frac{1}{2 \cdot (l_1+l_2)} \end{bmatrix} \quad (16)$$



**Fig. 4** Parameters of an omnidirectional wheel

Finally, the odometry equations using a centered discrete approximation can be formulated as illustrated in Eq. 17

**Table 1** Wheels' parameters of the three-wheeled omnidirectional robot

$i$	$l$ (m)	$\alpha$ (rad)	$\beta$ (rad)	$\gamma$ (rad)
1	$l$	$-\pi/3$	0	0
2	$l$	$\pi/3$	0	0
3	$l$	$\pi$	0	0

**Table 2** Wheels’ parameters of the four-wheeled omnidirectional robot

$i$	$l$ (m)	$\alpha$ (rad)	$\beta$ (rad)	$\gamma$ (rad)
1	$\sqrt{(l_1/2)^2 + (l_2/2)^2}$	$\alpha = \text{atan}\left(\frac{l_2}{l_1}\right)$	$\pi/2 - \alpha$	$-\pi/4$
2	$\sqrt{(l_1/2)^2 + (l_2/2)^2}$	$-\alpha$	$-\pi/2 + \alpha$	$\pi/4$
3	$\sqrt{(l_1/2)^2 + (l_2/2)^2}$	$\pi - \alpha$	$-\pi/2 + \alpha$	$\pi/4$
4	$\sqrt{(l_1/2)^2 + (l_2/2)^2}$	$\pi + \alpha$	$\pi/2 - \alpha$	$-\pi/4$

[31]. These equations apply to any omnidirectional robot.

if  $(\omega = 0)$   
 $x_k = x_{k-1} + \Delta d_k c_{\theta_{k-1}} - \Delta d_{n,k} s_{\theta_{k-1}}$   
 $y_k = y_{k-1} + \Delta d_k s_{\theta_{k-1}} + \Delta d_{n,k} c_{\theta_{k-1}}$   
 else  
 $x_k = x_{k-1} +$   
 $+ (\Delta d_k s_{\Delta\theta_k} + \Delta d_{n,k} (c_{\Delta\theta_k} - 1)) \frac{c_{\theta_{k-1} + \Delta\theta_k/2}}{\Delta\theta_k}$   
 $- (\Delta d_k (1 - c_{\Delta\theta_k}) + \Delta d_{n,k} s_{\Delta\theta_k}) \frac{s_{\theta_{k-1} + \Delta\theta_k/2}}{\Delta\theta_k}$  (17)  
 $y_k = y_{k-1} +$   
 $+ (\Delta d_k s_{\Delta\theta_k} + \Delta d_{n,k} (c_{\Delta\theta_k} - 1)) \frac{s_{\theta_{k-1} + \Delta\theta_k/2}}{\Delta\theta_k}$   
 $+ (\Delta d_k (1 - c_{\Delta\theta_k}) + \Delta d_{n,k} s_{\Delta\theta_k}) \frac{c_{\theta_{k-1} + \Delta\theta_k/2}}{\Delta\theta_k}$   
 end  
 $\theta_k = \theta_{k-1} + \Delta\theta_k$

### 3.3.2 Inverse Kinematics

As already mentioned, the inverse kinematics of any omnidirectional robot (Eq. 18) depend on the matrix  $J_{\text{omni}_m}^{-1}$ . Lastly, the angular velocity of the wheels ( $\dot{\varphi}_{i,k}$ ) can be calculated as in Eq. 6 [31].

$$\begin{bmatrix} v_{1,k} \\ \vdots \\ v_{m,k} \end{bmatrix} = J_{\text{omni}_m}^{-1} \cdot \begin{bmatrix} v_k \\ \omega_k \end{bmatrix} \tag{18}$$

## 4 OptiOdom Calibration Method

### 4.1 Optimization-based Algorithm

As shown in Section 3, the odometry equations define the current robot’s pose relative to the previous instant. However, these equations could be formulated to a time instant in which it is possible to obtain a known pose from a tracking system. For exemplification purposes, this formulation is illustrated next for the differential drive or

Ackerman/tricycle steering geometries:

$$\begin{aligned} x_j &= x_i + \sum_{k=i+1}^j \Delta d_k \cdot \cos(\theta_{k-1} + \Delta\theta_k/2) \\ y_j &= y_i + \sum_{k=i+1}^j \Delta d_k \cdot \sin(\theta_{k-1} + \Delta\theta_k/2) \\ \theta_j &= \theta_i + \sum_{k=i+1}^j \Delta\theta_k \end{aligned} \tag{19}$$

where the time instants  $i$  and  $j$  are relative to known poses of the robot. Note that the formulation illustrated in the set of Eq. 19 is also possible for omnidirectional robots. The only difference is that each element of the sum is dependent on the angular velocity of the robot.

Therefore, it is possible to define the final position error ( $\epsilon_{xy,s}$ ) of the odometry ( $x/y_{o,j(s)}$ ) and the ground-truth ( $x/y_{gt,j(s)}$ ) estimations (acquired at time instants  $i$  and  $j$ ) on each path segment  $s$ , as illustrated in Eq. 20. Given a set of segments ( $s \in \{1, 2, \dots, N_{seg}\}$ ), we estimate the kinematic parameters by minimizing a cost function  $f$ , as illustrated in Eq. 21. This cost function represents the sum of squares of the position errors. In terms of performing the minimization procedure, any iterative optimization algorithm could be used. The one used in this article was the improved Resilient Propagation without weight-backtracking (iRprop-) formulated by Igel and Hüsken [15]. The advantages of using a Rprop algorithm is that it is only a first-order optimization algorithm, highly accurate and robust, simple to implement, and provides good results in terms of convergence speed [20].

$$\epsilon_{xy,s} = \sqrt{(x_{o,j(s)} - x_{gt,j(s)})^2 + (y_{o,j(s)} - y_{gt,j(s)})^2} \tag{20}$$

$$\min f = \sum_{s=1}^{N_{seg}} \epsilon_{xy,s}^2 \tag{21}$$

Note that the calibration algorithm formulated is independent on the robot’s steering geometry. Even though the steering geometries considered in the experiments were the differential drive, Ackerman/tricycle, and omnidirectional ones, the algorithm could be adapted to other steering geometries. The only requirement is that the forward kinematics must be defined as dependent on the kinematic parameters.

### 4.2 Calibration Path

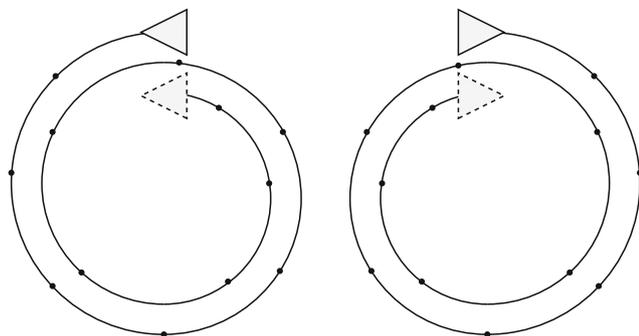
The OptiOdom calibration method is not path-specific. Indeed, the odometry data used for the calibration can be acquired during the robot’s operation. However, the choice of the calibration path should be one that the odometry data related to each kinematic parameter are

uncorrelated [8]. For example, Borenstein and Feng [3] demonstrated that a differential drive robot can obtain the same final position error on a unidirectional square path with different systematic errors (i.e., unequal wheels diameters or uncertainty on the wheelbase) between runs. In a practical view, if a robot only performs linear motion, it will not be possible to retrieve information for kinematic parameters that only contribute to angular motion, and vice-versa. So, the path should combine these two types of motions independently or simultaneously.

Therefore, we propose the use of a circular path with a decreasing radius after  $180^\circ$  turns in both CW and CCW directions, just as in the works of Kallasi et al. [19] and Galasso et al. [11], and illustrated in Fig. 5. This type of calibration path combines linear and angular motion with different wheels' velocities after  $180^\circ$  turns. The points on the path in Fig. 5 represent the ground-truth data acquisition instants. In the case of omnidirectional robots, the robot should execute the calibration path with linear velocity not only in the direction of  $X^R$  but also in  $Y^R$  both clock and counterclockwise. Note that omnidirectional robots can drive sideways, unlike differential or Ackerman/tricycle robots. So, we should also retrieve odometry data in the direction of  $Y^R$  for observability purposes of all kinematic parameters.

As for acquiring ground-truth data, Kallasi et al. [19] and Galasso et al. [11] specified that after  $180^\circ$  curves it should be measured the robot's pose. This metric is more suitable on paths highly dependent on angular movement. Even though we used the same path for odometry calibration, we defined that should be measured the robot's position (the method does not require the orientation of the robot) after a certain value of linear displacement. This metric is suitable not only for the calibration path used but also for square or arbitrary paths, given that the robot usually performs at least linear motion.

In terms of processing the ground-truth and the odometry data, Fig. 6a and b illustrates two possible different approaches. The first approach defines a path segment as the



**Fig. 5** Calibration path used in this article, Kallasi et al. [19] and Galasso et al. [11]

sub-path between the initial position and the data acquisition points along the path, while the second approach defines a sub-path between the data acquisition points and the final pose. Note that the only matching segment between these two is the entire path. However, segments of these two approaches can be correlated with each other if the robot performs the same motion over the segment. In order to have only uncorrelated data for the optimization procedure, we implemented the approach illustrated in Fig. 6a. However, the one illustrated in Fig. 6b could also be used.

### 4.3 Calibration Procedure

Next, it is enumerated the steps required to perform our calibration algorithm with the circular-based calibration path:

1. Measure the absolute position of the robot and initialize the odometry system with that position (the orientation can be initialized with  $0^\circ$ )
2. Run the robot through a curvilinear motion in CW direction ( $\omega < 0$ ):
  - Set the initial radius ( $R_0$ ) taking into consideration the available space ( $\omega_0 = \frac{v_0/v_{n,0}}{R_0}$ )
  - Set a radius decreasing ratio ( $r\%$ )
  - Update the linear velocities of the robot after each  $l$   $180^\circ$  curve ( $v_{l+1}/v_{n,l+1} = r\% \cdot v_l/v_{n,l}$ )
  - Measure the robot's absolute position after the robot has gone through 0.5m linear displacement (based on odometry)
3. Repeat steps 1-2 if needed (for noise robustness)
4. Repeat steps 1-3 in CCW direction
5. Omnidirectional robot: repeat steps 1-4 with linear velocity in direction of  $v_n$
6. Perform the optimization procedure
7. Adjust the robot parameters given the result from the previous step

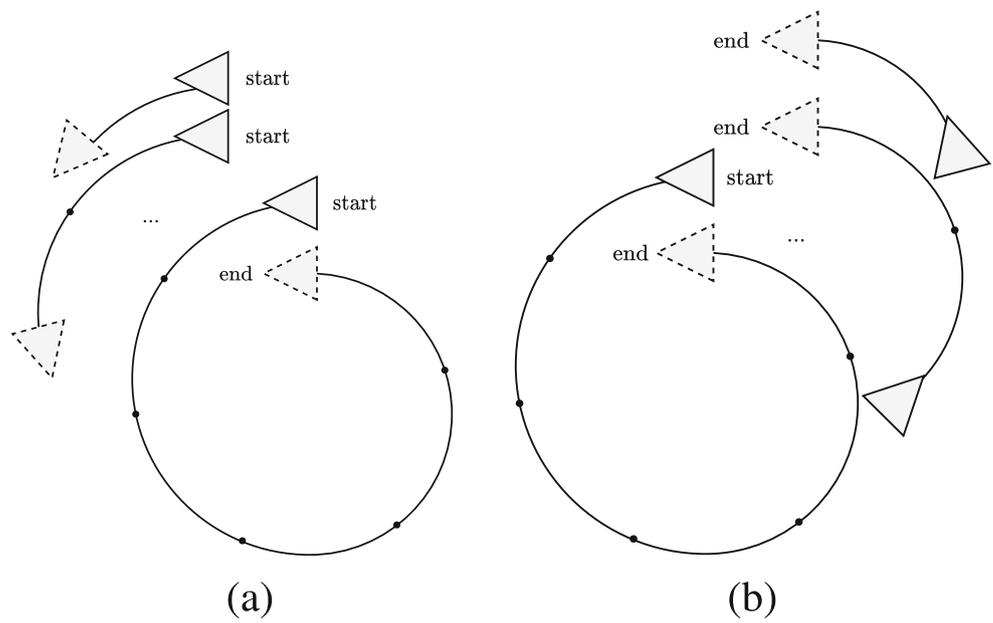
However, we underline that the OptiOdom optimization-based algorithm also works with other types of calibration paths. The experimental results presented in the next section show that it is possible to use OptiOdom for odometry calibration with square and arbitrary paths.

## 5 Experiments

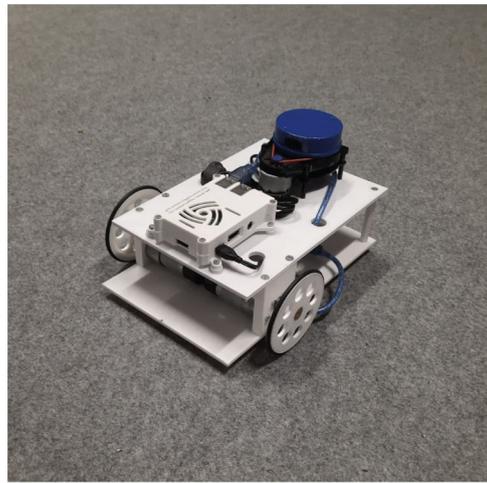
### 5.1 Robots Used in the Experiments

OptiOdom has been tested on four different robots: differential drive, tricycle, and three and four-wheeled omnidirectional. The use of these robots illustrated in Fig. 7

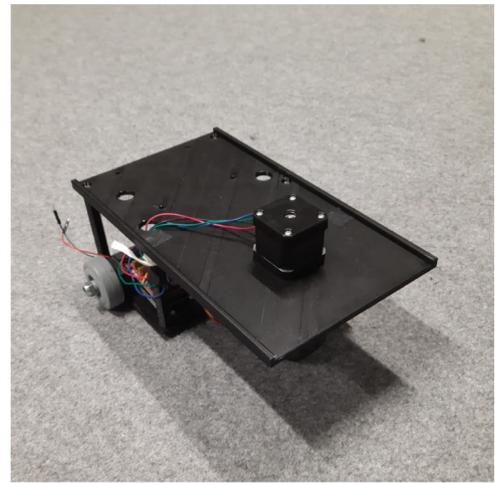
**Fig. 6** Different definitions for the path segments



**Fig. 7** Robots used in the experiments



(a) Differential drive



(b) Tricycle



(c) Three-wheeled omni



(d) Four-wheeled omni

intends to test the odometry calibration algorithm on the three most used steering geometries (differential drive, Ackerman/tricycle, omnidirectional) and compare its results with traditional calibration methods.

For comparison purposes, all the calibration algorithms implemented in this article will use the same initial estimations for the kinematic parameters. These initial estimations measured with a metric tape are described for the four robots used, as follows:

- Differential drive:  $b = 0.2\text{m}$ ,  $D_R = D_L = 0.084\text{m}$
- Tricycle:  $l = 0.15\text{m}$ ,  $D = 0.065\text{m}$
- Three-wheeled omnidirectional:  $l = 0.195\text{m}$ ,  
 $D_1 = D_2 = D_3 = 0.102\text{m}$
- Four-wheeled omnidirectional:  $l_1 = l_2 = 0.2\text{m}$ ,  
 $D_1 = D_2 = D_3 = D_4 = 0.06\text{m}$

## 5.2 Ground-truth

The OptiTrack [21] motion capture system is used as a ground-truth referencing system. Our system is composed of 6 Flex3 InfraRed (IR) cameras (one of them shown in Fig. 8a) that can locate the robot in an area of approximately  $3 \times 2.5\text{m}$ , although the total area available for the experiments (illustrated in Fig. 8b) is  $6 \times 2.5\text{m}$ . This motion capture system is widely used for human tracking, as exemplified in the literature review of Nagymáté and Kiss [28]. In robotics, the works of Yang and Shen [35], Furtado et al. [10], and Dudsizk [9] are examples of applications of the OptiTrack [21] tracking system for ground and aerial vehicles. The synchronized data of OptiTrack [21] and odometric estimations for the robot's pose are used to obtain all error metrics results and graphs (illustrating the pose of the robot in the space XY) presented in this article.

## 5.3 Evaluation of the OptiOdom Calibration Method

Our main goal was to evaluate the calibration accuracy of OptiOdom using the circular-based calibration path described in Section 4. Also, we evaluated the OptiOdom's accuracy using other types of calibration paths: square (the same used in UMBmark [3]) and an arbitrary one (i.e., using arbitrary motions for the robot). In the case of the square path, a simple proportional controller on the distance error to the closest point to a line was used for controlling the robot's linear and angular velocities to follow each side of the square. For the other paths used in the experiments, the wheels' angular velocity is directly controlled based on the desired velocity for the robot. The angular velocities are computed based on the robot's inverse kinematics presented in Section 3. The use of square and arbitrary paths allows evaluating the method's accuracy of the estimated kinematic parameters when using other calibration paths than the one

proposed in this paper. Note that OptiOdom is not path-specific (see Section 4.2). The experimental procedures for each path used in this article are available in videos on a YouTube playlist.<sup>1</sup>

Furthermore, the different methods implemented for all steering geometries are tested on three different types of paths: circular path described in Section 4.2, square, and an arbitrary one. If a method from the literature is not specific to a certain path, we use these three calibration paths with that method to estimate the odometry parameters. Each set of kinematic parameters estimated by OptiOdom are represented by a symbol depending on the calibration path used for the optimization procedure: \* (circular), † (square), and ‡ (arbitrary). This terminology also applies to the literature methods that are not path-specific. The tests with the initial estimation are represented by the symbol 0.

Finally, the proposed algorithm is compared to other methods presented in the literature using the following metrics:

- $\epsilon_{\max,d}$ : maximum distance error over the path
- $\epsilon_{\max,|\theta|}$ : maximum absolute orientation error over the path
- $\epsilon_{\max,\text{fin},d}$ : maximum final distance error
- $\epsilon_{\max,\text{fin},|\theta|}$ : maximum final absolute orientation error

The evaluation of the results obtained from the different methods is presented in comparison tables and categorized by the robot used in the experiments. When an arbitrary path is used, the path is presented on an XY graph.

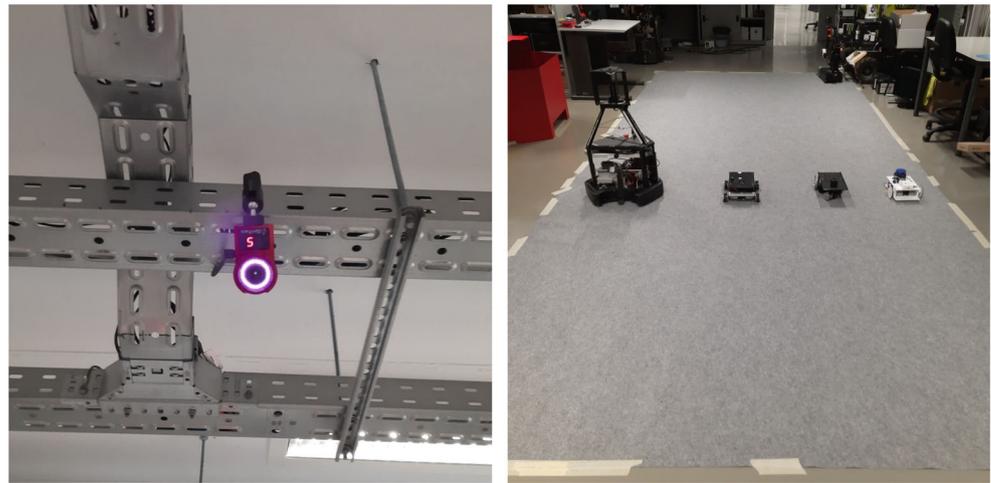
### 5.3.1 Differential Drive

The differential drive robot was first tested with the measured kinematic parameters (0), as indicated in Section 5.1. Next, we used the circular calibration path (\*) (described in Section 4.2 and illustrated in Fig. 9a) with OptiOdom to estimate the kinematic parameters of the robot. This calibration path is defined by its initial radius ( $R_0$ ), decreasing radius ratio ( $r\%$ ), number of  $180^\circ$  curves ( $\#180^\circ$ ), and total runs ( $N$ ). We also used a LxLm square (†) and an arbitrary (‡) calibration paths (illustrated in Figs. 9a and 10, respectively) for the OptiOdom calibration algorithm (described in Section 4.1). Then, we retrieved the estimated kinematic parameters relative to each one of these paths.

The works from the literature considered for comparison purposes were Borenstein and Feng [3], Jung and Chung [17], and Ivanjko et al. [16]. The first [3] for historical reasons, the [17] due to the improvements presented in their experiments over the methods [3, 4], and [16] given that it is an optimization-based and uses

<sup>1</sup><https://www.youtube.com/watch?v=-aL24R5CkGw&list=PLpCUb5UCOnV09bBFDVtbkEs0w-YA9eP4S>

Fig. 8 Experimental setup



(a) Flex3 IR camera

(b) Area available

a simple calibration path. In terms of calibration paths required for these methods, Borenstein and Feng [3] and Jung and Chung [17] use a square path (illustrated in Fig. 9b) and Ivanjko et al. [16] uses a specific path proposed in their work (illustrated in Fig. 9c).

The experimental results obtained for the differential drive robot are presented in Table 3. These results represent the evaluation of the estimated kinematic parameters on different test paths: circular (same as the one described in Section 4.2), square, and an arbitrary one. Each experiment is represented in Table 3 either by a symbol (type of calibration path used for the OptiOdom optimization algorithm described in Section 4.1) or a bibliographic reference (literature methods). The kinematic parameters estimated by each calibration method and the calibration path used for obtaining these parameters are presented as follows:

- OptiOdom – circular (\*) – Fig. 9a:
  - $R_0 = 0.85\text{m}$ ,  $\#180^\circ = 4$ ,  $r\% = 90\%$
  - $N = 6$  (3 CW + 3 CCW)
  - $b = 0.20150\text{m}$
  - $D_R = 0.08340\text{m}$ ,  $D_L = 0.08346\text{m}$
- OptiOdom – square (†) – Fig. 9b:
  - $L = 1.7\text{m}$
  - $N = 6$  (3 CW + 3 CCW)
  - $b = 0.20127\text{m}$
  - $D_R = 0.08349\text{m}$ ,  $D_L = 0.08356\text{m}$
- OptiOdom – arbitrary (‡) – Fig. 10:
  - $b = 0.20104\text{m}$
  - $D_R = 0.08348\text{m}$ ,  $D_L = 0.08353\text{m}$
- Borenstein and Feng – UMBmark [3] – square – Fig. 9b:

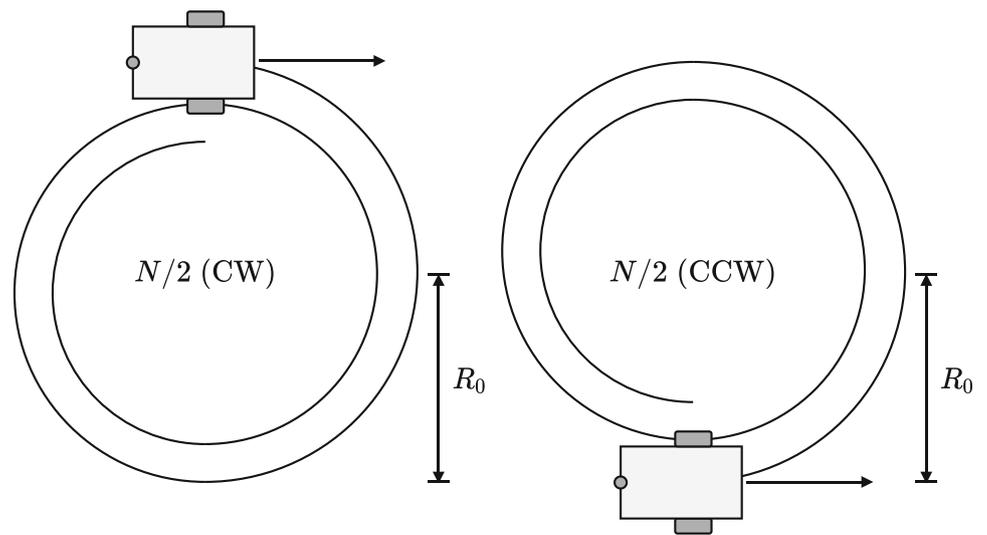
- $L = 1.7\text{m}$
- $N = 6$  (3 CW + 3 CCW)
- $b = 0.20170\text{m}$
- $D_R = 0.08398\text{m}$ ,  $D_L = 0.08402\text{m}$

- Jung and Chung [17] – square – Fig. 9b:
  - $L = 1.7\text{m}$
  - $N = 6$  (3 CW + 3 CCW)
  - $b = 0.20083\text{m}$
  - $D_R = 0.08399\text{m}$ ,  $D_L = 0.08401\text{m}$
- Ivanjko et al. [16] – straight-line + 180° rot. – Fig. 9c:
  - $L = 2\text{m}$
  - $N = 9$  (3 straight-line + 3 180° rot. CW + 3 180° rot. CCW)
  - $b = 0.20132\text{m}$
  - $D_R = 0.08396\text{m}$ ,  $D_L = 0.08390\text{m}$

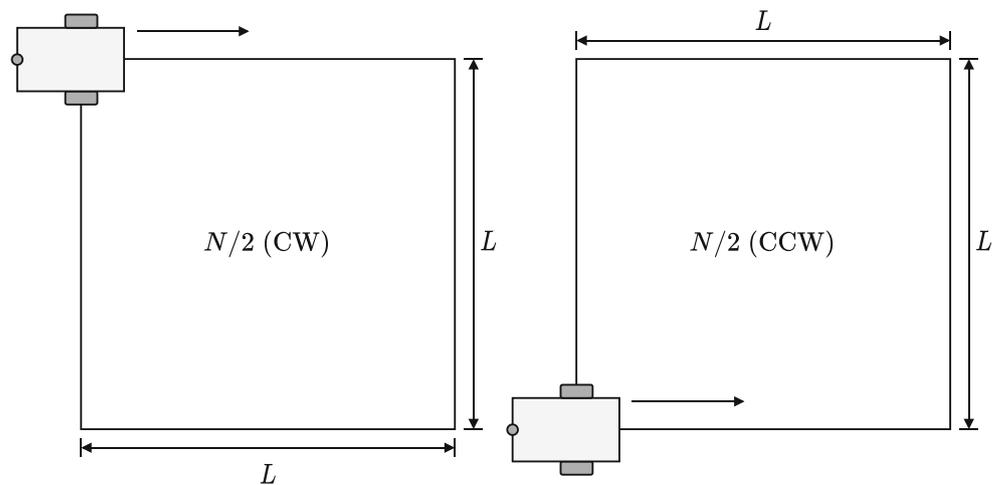
As for the discussion of the results, first, it should be noted that the OptiOdom had similar or lower maximum distance and orientation errors than the methods proposed in the literature. For example, comparing the proposed method using the circular (\*) path's data with the UMBmark [3], the last only had a lower maximum final distance error than OptiOdom (0.03964m vs 0.06527m, respectively).

Furthermore, the difference of the error metrics between using the circular (\*) or square (†) calibration paths for estimating the parameters with OptiOdom is very low. Indeed, this difference was less than 0.035m and 1° when we evaluate the estimated parameters on the different test paths (circular, square, and arbitrary). So, it shows that the calibration path described in Section 4.2 did not lead to overfitting the kinematic parameters only to circular motions. Also, it shows that the square path could be another possibility to use with the OptiOdom for the differential drive robot. As for using the arbitrary (‡) path to estimate

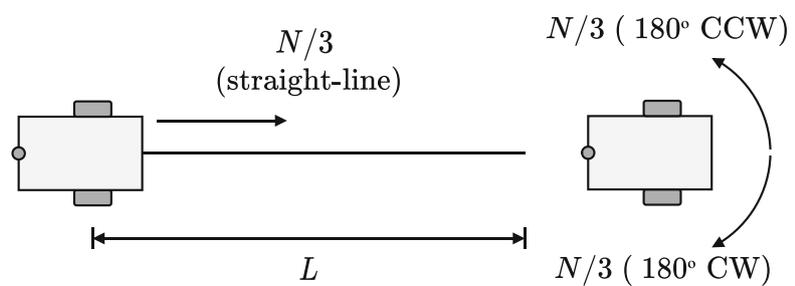
**Fig. 9** Calibration and test specific paths used in the experiments with the differential drive robot (paths (a) and (b) are used also for the experiments with the tricycle robot)



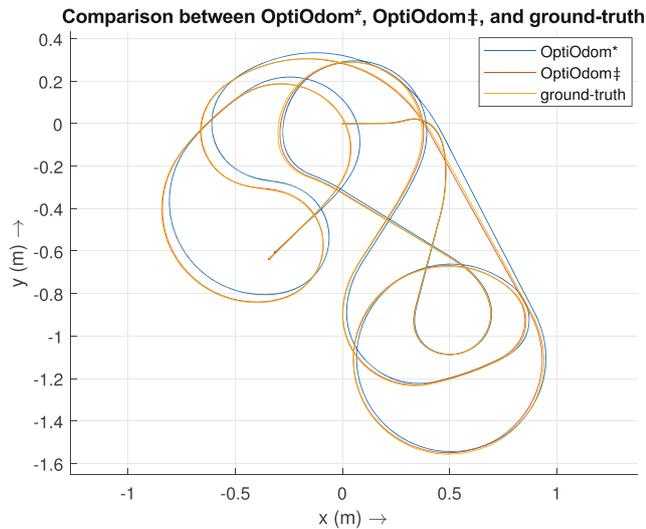
(a) Circular



(b) Square



(c) Ivanjko et al. [16]



**Fig. 10** Arbitrary path tested on the differential drive robot using the kinematic parameters obtained with OptiOdom\* and OptiOdom‡

the kinematic parameters, the results demonstrate that the path illustrated in Fig. 10 was also suitable for this robot and OptiOdom. Even though it seemed that it occurred

overfitting due to the error metrics being below 0.02m and 2° in its own path (‡), the difference for circular (\*) or square (†) data estimations on their respective paths is less than 0.03m and 3°.

Lastly, the UMBmark [3] obtained lower error metrics than Jung and Chung [17]. Even though the kinematic parameters for each method were retrieved from the same dataset, this result was not expected because the comparison results shown in [17] concluded that [3] should be worse. However, we point out that the square path finishes with a 90° on-the-spot rotation, and [17] uses the final orientation of the robot to estimate the parameters. Consequently, while the position should remain the same, the final orientation could be affected by minor wheels slipping or the inertia of the robot itself. As for Ivanjko et al. [16], it was the method that had worse results. The main reasons are the following ones: the method assumes that unequal wheels diameters only affect the straight-line motion and the uncertainty on the wheelbase only affects the on-the-spot rotations; also, it occurs the same problem as Jung and Chung [17] because Ivanjko et al. [16] uses the final orientation after the 180° rotations to estimate the wheelbase.

**Table 3** Experimental results using the differential drive robot

Test path	Calibration method	$\epsilon_{\max,d}$ (m)	$\epsilon_{\max, \theta }$ (°)	$\epsilon_{\max,fin,d}$ (m)	$\epsilon_{\max,fin, \theta }$ (°)
circular	0	0.16160	14.46810	0.15530	13.63979
	OptiOdom*	<u>0.02930</u>	<u>3.00911</u>	<u>0.02862</u>	<u>1.51651</u>
	OptiOdom†	0.04175	3.93018	0.03392	2.81830
	OptiOdom‡	0.05680	5.15923	0.04856	4.09603
	[3]	0.07914	7.27295	0.07291	6.29723
	[17]	0.12131	10.86131	0.11504	10.02437
	[16]	0.12392	11.06147	0.11735	10.22426
square	0	0.13823	9.44422	0.11389	7.48425
	OptiOdom*	<u>0.07088</u>	<u>3.79842</u>	0.06527	2.23851
	OptiOdom†	0.07766	4.19466	0.04385	<u>1.62312</u>
	OptiOdom‡	0.07460	4.89713	0.04659	2.39684
	[3]	0.07946	5.85275	<u>0.03964</u>	3.56040
	[17]	0.10839	7.61089	0.07563	5.54945
	[16]	0.12429	8.41763	0.08849	6.20836
arbitrary	0	0.27740	11.36851	0.16488	6.02200
	OptiOdom*	0.07196	2.60562	0.04311	1.73422
	OptiOdom†	0.04097	2.44194	0.01832	1.54702
	OptiOdom‡	<u>0.01276</u>	<u>2.09365</u>	<u>0.00594</u>	<u>0.19647</u>
	[3]	0.07159	3.75981	0.04449	1.18289
	[17]	0.17649	7.61779	0.10614	3.62400
	[16]	0.16948	8.55794	0.08728	6.84406

### 5.3.2 Tricycle

The side-length of the square had to be reduced to 1.5m to prevent the tricycle’s path exceeding the available space. The method presented by Kallasi et al. [19] was implemented to present another comparison reference other than the initial estimation of the kinematic parameters. This calibration method is formulated considering that the circular calibration path (described in Section 4.2 and the same path as the one illustrated in Fig. 9a) is used to obtain the kinematic parameters.

The experimental results are presented in Table 4, and the estimated parameters of each method were the following ones:

- OptiOdom – circular (\*) – Fig. 9a:
  - $R_0 = 0.85\text{m}$ ,  $\#180^\circ = 4$ ,  $r\% = 90\%$
  - $N = 6$  (3 CW + 3 CCW)
  - $l = 0.15106\text{m}$
  - $D = 0.06172\text{m}$
  - $\alpha_{off} = -1.14443^\circ$
- OptiOdom – square (†) – Fig. 9b:
  - $L = 1.5\text{m}$
  - $N = 6$  (3 CW + 3 CCW)
  - $l = 0.15788\text{m}$
  - $D = 0.06414\text{m}$
  - $\alpha_{off} = -1.15044^\circ$
- OptiOdom – arbitrary (‡) – Fig. 11:
  - $l = 0.15046\text{m}$

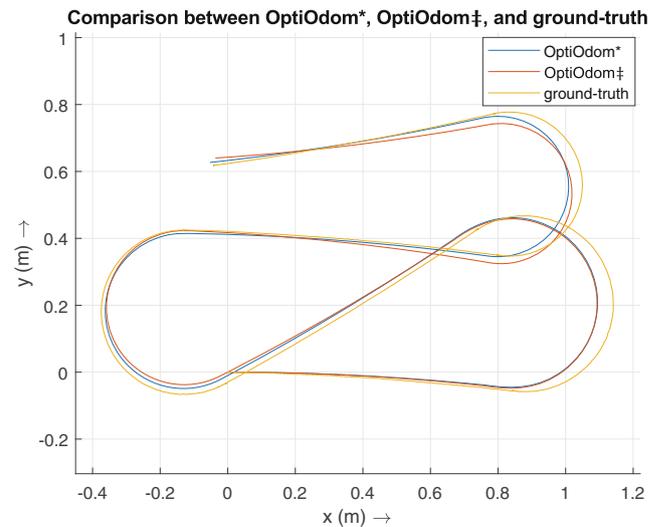


Fig. 11 Arbitrary path tested on the tricycle robot using the kinematic parameters obtained with OptiOdom\* and OptiOdom‡

- $D = 0.06154\text{m}$
- $\alpha_{off} = -1.21378^\circ$
- Kallasi et al. [19] – circular – Fig. 9a:
  - $R_0 = 0.85\text{m}$ ,  $\#180^\circ = 4$ ,  $r\% = 90\%$
  - $N = 6$  (3 CW + 3 CCW)
  - $l = 0.15000\text{m}$
  - $D = 0.06138\text{m}$
  - $\alpha_{off} = -1.10158^\circ$

In terms of the experiments performed with the tricycle robot, using the circular (\*) or the square (†) calibration

Table 4 Experimental results using the tricycle robot

Test path	Calibration method	$\epsilon_{\max,d}$ (m)	$\epsilon_{\max, \theta }$ (°)	$\epsilon_{\max,fin,d}$ (m)	$\epsilon_{\max,fin, \theta }$ (°)
circular	0	1.23668	107.77487	1.07760	101.38498
	OptiOdom*	<u>0.12748</u>	8.20463	<u>0.04407</u>	7.99657
	OptiOdom†	0.12751	12.04124	0.12726	11.83317
	OptiOdom‡	0.13490	11.62948	0.06517	11.42141
	[19]	0.14871	<u>6.22815</u>	0.05576	<u>4.54606</u>
square	0	1.31560	88.17917	1.28562	65.60522
	OptiOdom*	0.20796	25.22979	0.11053	4.42432
	OptiOdom†	<u>0.17762</u>	23.33737	<u>0.10906</u>	<u>4.38200</u>
	OptiOdom‡	0.21098	<u>22.95523</u>	0.14249	7.39547
	[19]	0.23960	27.27458	0.11799	4.57535
arbitrary	0	0.82386	56.60013	0.82359	56.46539
	OptiOdom*	<u>0.05440</u>	<u>4.38828</u>	<u>0.01385</u>	2.43165
	OptiOdom†	0.08286	5.45476	0.08242	3.72467
	OptiOdom‡	0.05447	5.14007	0.02238	4.96001
	[19]	0.07254	5.01587	0.02871	<u>0.46385</u>

path for estimating the parameters with the OptiOdom led to similar results in terms of the error metrics. The only exceptions are the maximum absolute orientation errors on the circular test path, and the maximum final distance on the circular and the arbitrary test paths (differences greater than  $4^\circ$  and 0.07m, respectively). Indeed, the estimated kinematic parameters using the circular (\*) calibration path with OptiOdom seemed to be more suitable to the tricycle geometry than the square (†) calibration path. As for using an arbitrary (‡) calibration path (illustrated in Fig. 11) with OptiOdom, the estimated kinematic parameters led to similar error metrics to the circular (\*) data’s parameters.

Another observation is relative to the fact of the maximum distance and orientation errors (over the entire path, not the final error metrics) being greater than 0.15m and  $20^\circ$  on the square test path, even after calibrating the robot. This result is due to unmodelled effects for the tricycle robot. For example, the intersection of the steering rotation axis with the ground and the point of contact with it of the wheel are slightly different because of mechanical imperfections. This difference is more noticeable when the robot is performing on-the-spot rotations, such as the  $90^\circ$  ones required for square paths. The robot moves slightly while rotating upon itself, while the odometry is estimating only rotational motion.

The OptiOdom using the circular (\*) data’s parameters had similar error metrics when compared to Kallasi et al. [19]. However, it should be noted that the last cannot estimate the distance between the rear and front wheels ( $l$ ). Indeed, [19] assumes that this distance is known.

Ultimately, the parameter that most influenced the difference between the initial estimation and the calibrated robot was the steering angle offset ( $\alpha_{off}$ ). Both OptiOdom and Kallasi et al. [19] led similar estimations for this kinematic parameter ( $-1.21^\circ < \alpha_{off} < -1.10^\circ$ ).

### 5.3.3 Three-wheeled Omnidirectional

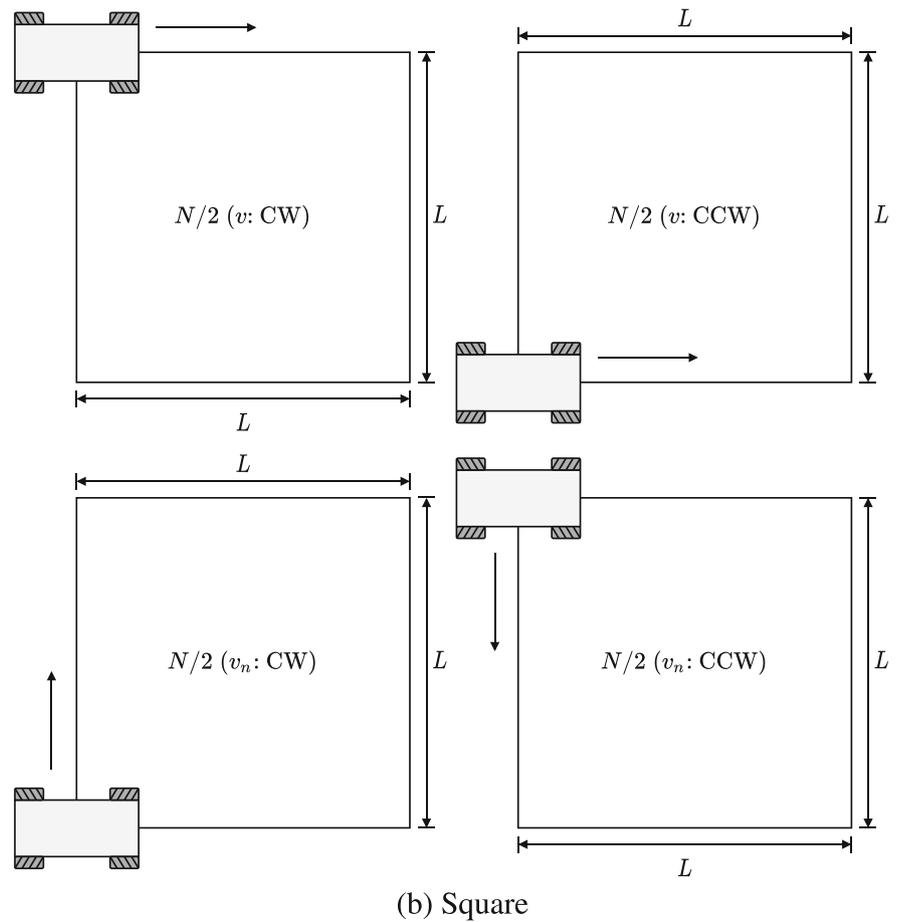
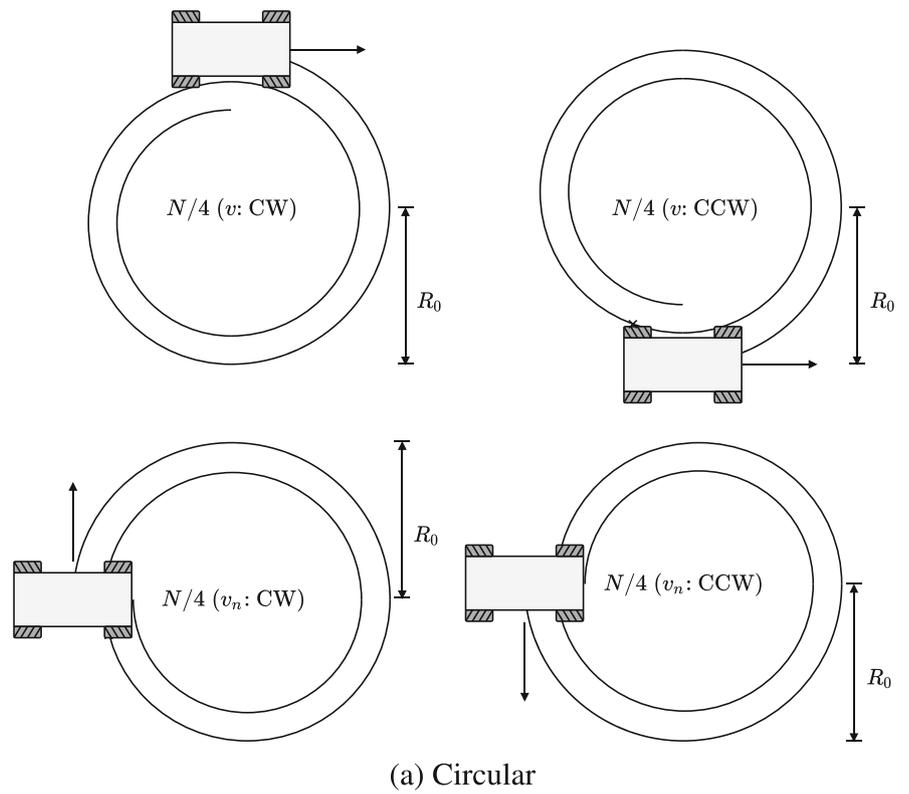
The difference to the square path tested using the tricycle robot is that it was possible to increase the square side-length from 1.5m to 1.6m. As for methods in the literature intended for an omnidirectional robot, it was implemented the method proposed by Lin et al. [22]. This calibration method is not path-specific. So, similar to OptiOdom, we used three types of calibration paths – circular (\*), square (†), and an arbitrary (‡) paths illustrated in Fig. 12a and b, and 13, respectively – to calibrate the kinematic model with Lin et al. [22]’s method. Note that both circular and square calibration paths were performed in  $v$  and  $v_n$  directions. As already analyzed in Section 2.2, Lin et al. [22] adjusts directly the kinematic model (specifically, the matrix  $J_{omni_3}$ ), instead of estimating the kinematic parameters.

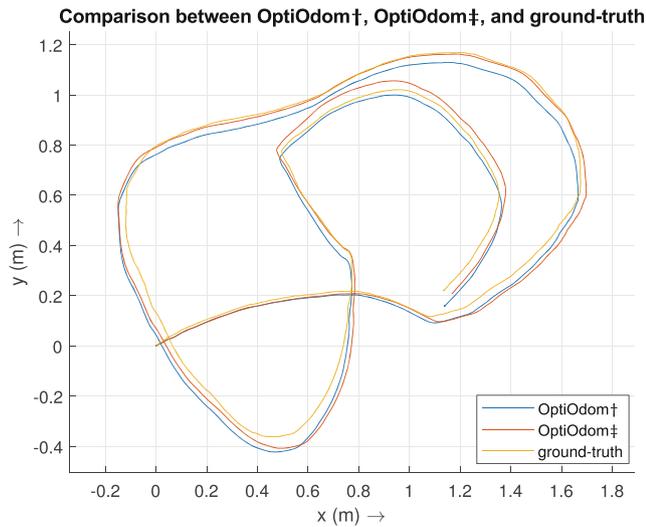
Table 5 presents the experimental results for the three-wheeled omnidirectional robot, and the estimated parameters relative to each method are presented next:

- OptiOdom – circular (\*) – Fig. 12a:
  - $R_0 = 0.85\text{m}, \#180^\circ = 4, r\% = 90\%$
  - $N = 12 (v: 3 \text{ CW} + 3 \text{ CCW}; v_n: 3 \text{ CW} + 3 \text{ CCW})$
  - $l = 0.19133\text{m}$
  - $D_1 = 0.09785\text{m}, D_2 = 0.09667\text{m}, D_3 = 0.09697\text{m}$
- OptiOdom – square (†) – Fig. 12b:
  - $L = 1.6\text{m}$
  - $N = 12 (v: 3 \text{ CW} + 3 \text{ CCW}; v_n: 3 \text{ CW} + 3 \text{ CCW})$
  - $l = 0.19145\text{m}$
  - $D_1 = 0.09951\text{m}, D_2 = 0.09779\text{m}, D_3 = 0.09853\text{m}$
- OptiOdom – arbitrary (‡) – Fig. 13:
  - $l = 0.19375\text{m}$
  - $D_1 = 0.10202\text{m}, D_2 = 0.09947\text{m}, D_3 = 0.10023\text{m}$
- Lin et al. [22] – circular (\*) – Fig. 12a:
  - $R_0 = 0.85\text{m}, \#180^\circ = 4, r\% = 90\%$
  - $N = 12 (v: 3 \text{ CW} + 3 \text{ CCW}; v_n: 3 \text{ CW} + 3 \text{ CCW})$
  - $J_{omni_3} = \begin{bmatrix} -0.557474 & 0.526741 & 0 \\ -0.321858 & -0.304114 & 0.669212 \\ -1.684883 & -1.658193 & -1.667353 \end{bmatrix}$
- Lin et al. [22] – square (†) – Fig. 12b:
  - $L = 1.6\text{m}$
  - $N = 12 (v: 3 \text{ CW} + 3 \text{ CCW}; v_n: 3 \text{ CW} + 3 \text{ CCW})$
  - $J_{omni_3} = \begin{bmatrix} -0.568508 & 0.538931 & 0 \\ -0.328228 & -0.311152 & 0.665076 \\ -1.709092 & -1.673875 & -1.692844 \end{bmatrix}$
- Lin et al. [22] – arbitrary (‡) – Fig. 13:
  - $J_{omni_3} = \begin{bmatrix} -0.565847 & 0.582452 & 0 \\ -0.326692 & -0.336279 & 0.628914 \\ -1.753347 & -1.654622 & -1.752061 \end{bmatrix}$

The OptiOdom seemed to have overfitted the kinematic parameters when using the circular (\*) path’s data for the optimization procedure. The overfit is noticed when the respective kinematic parameters are evaluated on the square test path with maximum distance and absolute orientation errors greater than 0.25m and  $12^\circ$ , respectively.

**Fig. 12** Calibration and test specific paths used in the experiments with omnidirectional robots





**Fig. 13** Arbitrary path tested on the three-wheeled omnidirectional robot using the kinematic parameters obtained with OptiOdom† and OptiOdom‡

The OptiOdom using the square (†) data’s parameters seemed to be the most suitable calibration path for the three-wheeled omnidirectional robot used in the experiments. In terms of the parameters estimated from the arbitrary (‡) path

(illustrated in Fig. 13) using the OptiOdom, it had worse error metrics than the other experiments on both circular and square test paths. This observation leads to the conclusion that the arbitrary (‡) calibration path was not suitable for calibrating the three-wheeled omnidirectional robot.

Lin et al. [22] obtained the best result on the circular test path with a maximum distance and absolute orientation error less than 0.132m and 6.2°, respectively. This result was obtained using the circular (\*) calibration path. Similar to OptiOdom, the kinematic parameters estimated with Lin et al. [22] using the circular (\*) calibration path seemed to have overfitted. When tested on the square path, these same kinematic parameters lead to maximum distance and absolute orientation errors greater than 0.17m and 8°, respectively. Note that these error metrics are lower than the ones obtained with OptiOdom. Indeed, when either the circular (\*) or the square (†) calibration paths are used, Lin et al. [22] can achieve similar or improved accuracy over OptiOdom. One possible explanation is the greater number of DoFs estimated with Lin et al. [22] compared to OptiOdom (6 versus 4 DoFs, respectively). However, the main disadvantage is the model’s overfitting when the data does not allow the computation of an unbiased estimator. Analyzing the results when the arbitrary calibration (‡) path is used (tested on circular and square paths),

**Table 5** Experimental results using the three-wheeled omnidirectional

Test path	Calibration method	$\epsilon_{max,d}$ (m)	$\epsilon_{max, \theta }$ (°)	$\epsilon_{max,fin,d}$ (m)	$\epsilon_{max,fin, \theta }$ (°)
circular	0	0.30354	26.16780	0.26510	25.95029
	OptiOdom*	0.13349	13.32930	0.13292	13.23028
	OptiOdom†	0.19542	12.75036	0.13647	12.26674
	OptiOdom‡	0.25487	18.24839	0.18152	16.57484
	[22]*	<u>0.13103</u>	<u>6.12798</u>	<u>0.07426</u>	<u>5.79114</u>
	[22]†	0.24231	18.19126	0.18912	17.69994
	[22]‡	0.60563	54.37142	0.55068	53.85433
square	0	0.28213	15.28968	0.28172	13.21701
	OptiOdom*	0.25265	12.77193	0.25255	12.55824
	OptiOdom†	<u>0.12232</u>	<u>5.81393</u>	0.10434	4.67215
	OptiOdom‡	0.15615	8.66265	0.10884	4.97658
	[22]*	0.17240	8.90042	0.17230	8.26980
	[22]†	0.12997	5.89576	<u>0.04339</u>	<u>3.04288</u>
	[22]‡	0.50766	29.57256	0.46031	25.82433
arbitrary	0	0.16019	<u>13.93151</u>	0.08301	5.81438
	OptiOdom*	0.13752	14.02916	0.09684	0.74009
	OptiOdom†	0.13554	13.98898	0.06129	0.41018
	OptiOdom‡	<u>0.13297</u>	13.97173	<u>0.03535</u>	2.80621
	[22]*	0.13504	14.01217	0.10402	1.04475
	[22]†	0.13392	13.97440	0.08682	<u>0.30449</u>
	[22]‡	0.16232	13.93299	0.05125	1.25880

Lin et al. [22] achieved worse odometry accuracy than OptiOdom: maximum distance and absolute orientation errors over the entire path of 0.61m and 55° versus 0.26m and 19°, respectively.

Lastly, it should be noted that the three-wheeled omnidirectional robot used in the experiments has a wheel that can have 2 different and distinct contact points with the ground when the robot is moving. So, this type of wheel increases the uncertainty on the odometry estimation and the accuracy of the distance between the wheels and the robot’s geometric center ( $l$ ).

### 5.3.4 Four-wheeled Omnidirectional

A circular and a square paths (illustrated in Fig. 12a and b, respectively) were used to compare the OptiOdom calibration algorithm to Lin et al. [22]’s work on the four-wheeled omnidirectional robot, just as for the three-wheeled omnidirectional robot.

The results obtained in the experiments performed are presented in Table 6. The parameters estimated by each method are presented next:

- OptiOdom – circular (\*) – Fig. 12a:
  - $R_0 = 0.8\text{m}, \#180^\circ = 4, r\% = 90\%$
  - $N = 12 (v: 3 \text{ CW} + 3 \text{ CCW}; v_n: 3 \text{ CW} + 3 \text{ CCW})$
  - $l_1 + l_2 = 0.4106\text{m}$
  - $D_1 = 0.06259\text{m}, D_2 = 0.06354\text{m}, D_3 = 0.06355\text{m}, D_4 = 0.06323\text{m}$

- OptiOdom – square (†) – Fig. 12b:
  - $L = 1.7\text{m}$
  - $N = 12 (v: 3 \text{ CW} + 3 \text{ CCW}; v_n: 3 \text{ CW} + 3 \text{ CCW})$
  - $l_1 + l_2 = 0.42264\text{m}$
  - $D_1 = 0.06715\text{m}, D_2 = 0.06737\text{m}, D_3 = 0.06734\text{m}, D_4 = 0.06767\text{m}$
- OptiOdom – circular (\*) + square (†) – Fig. 12a + b:
  - $l_1 + l_2 = 0.41564\text{m}$
  - $D_1 = 0.06409\text{m}, D_2 = 0.06452\text{m}, D_3 = 0.06451\text{m}, D_4 = 0.06469\text{m}$
- Lin et al. [22] – circular (\*) – Fig. 12a:
  - $R_0 = 0.8\text{m}, \#180^\circ = 4, r\% = 90\%$
  - $N = 12 (v: 3 \text{ CW} + 3 \text{ CCW}; v_n: 3 \text{ CW} + 3 \text{ CCW})$
  - $J_{\text{omni}_4} = \begin{bmatrix} 0.26403 & -0.27877 & 0.24872 & -0.26705 \\ -0.26403 & -0.27877 & 0.24872 & 0.26705 \\ 8.34430 & -10.90538 & -10.90500 & 8.33169 \end{bmatrix}$
- Lin et al. [22] – square (†) – Fig. 12b:
  - $L = 1.7\text{m}$
  - $N = 12 (v: 3 \text{ CW} + 3 \text{ CCW}; v_n: 3 \text{ CW} + 3 \text{ CCW})$
  - $J_{\text{omni}_4} = \begin{bmatrix} 0.24137 & -0.261887 & 0.28703 & -0.31733 \\ -0.24137 & -0.261887 & 0.28703 & 0.31733 \\ -2.09739 & -0.533079 & -0.53361 & -2.10845 \end{bmatrix}$
- Lin et al. [22] – circular (\*) + square (†) – Fig. 12a + b:

**Table 6** Experimental results using the four-wheeled omnidirectional

Test path	Calibration method	$\epsilon_{\text{max},d}$ (m)	$\epsilon_{\text{max}, \theta }$ (°)	$\epsilon_{\text{max},\text{fin},d}$ (m)	$\epsilon_{\text{max},\text{fin}, \theta }$ (°)
circular	0	0.33397	33.29796	0.33308	32.92056
	OptiOdom*	0.08201	<u>7.38059</u>	0.08034	<u>6.46070</u>
	OptiOdom†	0.35689	36.35712	0.35468	35.30545
	OptiOdom*, †	0.12575	13.43768	0.12339	12.23475
	[22]*	<u>0.07098</u>	8.06643	<u>0.06814</u>	6.59450
	[22]†	0.29078	29.06596	0.28646	28.03468
	[22]*, †	0.13074	10.90957	0.12363	8.31950
square	0	0.54130	28.12519	0.53934	28.10249
	OptiOdom*	0.32550	<u>19.21303</u>	0.32413	15.25876
	OptiOdom†	<u>0.13032</u>	21.09370	<u>0.07962</u>	5.75179
	OptiOdom*, †	0.27185	19.59780	0.27050	12.39728
	[22]*	0.33138	19.66074	0.33015	14.94858
	[22]†	0.13815	20.73310	0.12115	<u>4.25336</u>
	[22]*, †	0.29754	19.84756	0.29624	14.94723

$$- J_{\text{omni}_4} = \begin{bmatrix} 0.24870 & -0.27379 & 0.26489 & -0.28242 \\ -0.24870 & -0.27379 & 0.26489 & 0.28242 \\ -18.39197 & 15.82822 & 15.82570 & -18.40799 \end{bmatrix}$$

The kinematic parameters estimated with OptiOdom from the square (†) calibration path seemed to be overfitted to this type of path. When testing the same parameters on the circular test path, the resulting error metrics were even worse than the initial estimation (0). Although the parameters estimated from the circular (\*) path’s data did not have worse error than the initial estimation (0), the maximum distance and absolute orientation errors on the square path were greater than 0.32m and 19°, respectively. In comparison to Lin et al. [22], OptiOdom obtains similar accuracy when we compared these two methods using the same calibration and test paths. Lin et al. [22] was slightly better overall than OptiOdom on the circular test path, in terms of the error metrics present in Table 6; and vice versa for the square path.

In comparison with the other robots, additional experiments were considered for the four-wheeled omnidirectional robot. Moreover, these experiments considered the data retrieved from both circular and square (\*, †) calibration paths for odometry calibration. After estimating the kinematic parameters with OptiOdom and Lin et al. [22], we evaluated the odometry accuracy on the circular and square test paths. Analyzing the results obtained for both methods, we conclude that these experiments reduced the overfitting. Indeed, the error metrics of these experiments are a middle ground compared to considering the circular and square calibration paths individually for the optimization procedure.

The consideration of both circular and square (\*, †) calibration paths for odometry calibration originate a dataset with all possible combinations of linear (in both  $v$  and  $v_n$  directions) and angular motions, similar to an arbitrary path. The square calibration path does not combine simultaneously linear and angular motions, and the circular one combines these two types of motion together. Also, note that in Section 4.2 we defined that the calibration path (even if an arbitrary path was used) should combine linear and angular motions. So, we considered that it was not necessary to test the four-wheeled omnidirectional robot on an arbitrary path. This type of path is already analyzed for the omnidirectional steering geometry (three-wheeled omnidirectional robot).

Lastly, we point out that the combination of different types of motions as input data for the optimization procedure reduces the probability of overfitting. This approach can be applied to all steering geometries using OptiOdom. The method does not require a specific type of path.

## 6 Conclusions and Future Work

In conclusion, this article proposes a generalized approach (in terms of the steering geometry) for odometry calibration of wheeled mobile robots. To the best of our knowledge, the OptiOdom is the only calibration algorithm that intends to calibrate three different steering geometries: differential drive, Ackerman/tricycle, and omnidirectional.

Our previous work [32, 33] allowed us to analyzed thoroughly the literature on odometry calibration. The main conclusions taken from this work were the more recent trend of developing optimization-based algorithms, and the fact that none of the works found calibrated more than one distinct steering geometry. Then, we analyzed the kinematic model for the differential drive, Ackerman/tricycle, and omnidirectional geometries. This analysis showed that the estimation of the kinematic parameters could be done by implementing an interactive optimization algorithm, given that the calibration problem is usually a non-linear one. The cost function defined for the optimization procedure is not specific to a certain type of motion or path. Although we propose the use of a specific calibration path for odometry calibration, OptiOdom is not path-specific. As for the requirements of the method, they are the following ones: the robot must perform linear and angular motions when acquiring data for calibration, and the kinematic model should be described as a relation with the kinematic parameters. The latter would be already needed for obtaining the odometry estimation for the robot’s pose.

The results obtained in the experiments demonstrate that OptiOdom is similar or better (in terms of maximum distance and absolute orientation errors along the path) than the methods from the literature implemented in this article ([3, 16, 17, 19, 22]). Furthermore, OptiOdom was not only tested with the suggested calibration path (the circular path with decreasing radius proposed by [19]) but also with a square and arbitrary calibration paths for the three different steering geometries considered for tests. The tests on these three different types of paths confirm that the method is not path-specific.

As for the four mobile platforms used in the tests (differential drive, tricycle, and three and four-wheeled omnidirectional), OptiOdom achieved maximum distance and absolute orientation errors lower than 0.10m and 5° on the differential drive robot. For tricycle robots, the proposed method accomplished error metrics below 0.13m and 13° independently of the type of calibration path used for odometry calibration. Even though the error metrics were above 0.20m and 20° in the square test path, these metrics are affected by mechanical imperfections of the robot. The results obtained for the three and the four-wheeled

omnidirectional robots were very similar. In both cases, it was occurring overfitting of the estimated kinematic parameters. The solution studied and tested was to consider both circular and square data for the optimization procedure.

Lastly, the OptiOdom odometry calibration method accomplished its main goal of being a generalized approach for odometry calibration while achieving similar or improved odometry accuracy over the current literature. The OptiOdom and the methods from the literature considered in this article are implemented in MATLAB and available in a GitHub repository.<sup>2</sup> However, the code can be easily adapt to another language. The implementation available in the repository allows the execution of the odometry calibration methods tested in this article (OptiOdom plus the ones from the literature). The only requirement is the availability of synchronized ground-truth and odometry data. The next developments will be studying the online implementation of OptiOdom (executing it while the robot is in operation) and evaluate the use of this algorithm for fault detection (evaluation of derail of localization algorithms).

**Author Contributions** Ricardo B. Sousa: conceptualization, methodology, formal analysis and investigation, software, data acquisition, writing – original draft preparation. Marcelo R. Petry: conceptualization, methodology, writing – review and editing, supervision. Paulo G. Costa: data acquisition, software, writing – review and editing, resources. Antônio Paulo Moreira: conceptualization, methodology, funding, software, writing – review and editing, supervision, resources.

**Funding** This work is financed by the ERDF – European Regional Development Fund through the Operational Programme for Competitiveness and Internationalisation – COMPETE 2020 Programme, and by National Funds through the Portuguese funding agency, FCT – Fundação para a Ciência e a Tecnologia, within project SAICTPAC/0034/2015- POCI-01-0145-FEDER-016418. This work is also financed by the ERDF – European Regional Development Fund through the Operational Programme for Competitiveness and Internationalisation – COMPETE 2020 Programme within project POCI-01-0145-FEDER-006961, and by National Funds through the FCT – Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) as part of project UID/EEA/50014/2013.

**Availability of data and material** The authors confirm that the data supporting the findings of this study are available in a GitHub repository referenced in the text.

**Code availability** All code generated or used during the study are available in a GitHub repository referenced in the text.

## Declarations

**Ethics approval** Not applicable (this article does not contain any studies with human participants or animals performed by any of the authors).

<sup>2</sup><https://github.com/inesc-tec-robotics/optiodom>

**Consent for publication** All authors have approved the manuscript and agreed with its publication on the Journal of Intelligent & Robotic Systems.

**Consent to participate** Not applicable (this article does not contain any studies with human participants or animals performed by any of the authors).

**Conflict of interest** The authors have no financial or proprietary interests in any material discussed in this article.

## References

1. Abbas, T., Arif, M., Ahmed, W.: Measurement and correction of systematic odometry errors caused by kinematics imperfections in mobile robots. In: 2006 SICE-ICASE International Joint Conference, pp. 2073–2078. <https://doi.org/10.1109/SICE.2006.315554> (2006)
2. Antonelli, G., Chiaverini, S., Fusco, G.: A calibration method for odometry of mobile robots based on the least-squares technique: theory and experimental validation. *IEEE Trans. Robot.* **21**(5), 994–1004 (2005). <https://doi.org/10.1109/TRO.2005.851382>
3. Borenstein, J., Feng, L.: Measurement and correction of systematic odometry errors in mobile robots. *IEEE Trans. Robot. Autom.* **12**(6), 869–880 (1996). <https://doi.org/10.1109/70.544770>
4. Bostani, A., Vakili, A., Denidni, T.A.: A novel method to measure and correct the odometry errors in mobile robots. In: 2008 Canadian Conference on Electrical and Computer Engineering, pp. 897–900. <https://doi.org/10.1109/CCECE.2008.4564665> (2008)
5. Caltabiano, D., Muscato, G., Russo, F.: Localization and Self-Calibration of a Robot for Volcano Exploration. In: IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004, Vol. 1, pp. 586–591 (2004). <https://doi.org/10.1109/ROBOT.2004.1307212>
6. Cantelli, L., Ligama, S., Muscato, G., Spina, D.: Auto-calibration methods of kinematic parameters and magnetometer offset for the localization of a tracked mobile robot. *Robotics* **5**(4). <https://doi.org/10.3390/robotics5040023> (2016)
7. Cecco, M.D.: Self-calibration of AGV inertial-odometric navigation using absolute-reference measurements. In: IMTC/2002. Proceedings of the 19th IEEE Instrumentation and Measurement Technology Conference (IEEE Cat. No.00CH37276), vol. 2, pp. 1513–1518 (2002). <https://doi.org/10.1109/IMTC.2002.1007183>
8. Censi, A., Franchi, A., Marchionni, L., Oriolo, G.: Simultaneous calibration of odometry and sensor parameters for mobile robots. *IEEE Trans. Robot.* **29**(2), 475–492 (2013). <https://doi.org/10.1109/TRO.2012.2226380>
9. Dudzik, S.: Application of the motion capture system to estimate the accuracy of a wheeled mobile robot localization. *Energies* **13**(23). <https://doi.org/10.3390/en13236437> (2020)
10. Furtado, J.S., Liu, H., Lai, G., Lacheray, H., Desouza-Coelho, J.: Comparative analysis of Optitrack motion capture systems. In: Advances in Motion Sensing and Control for Robotic Applications, pp. 15–31. [https://doi.org/10.1007/978-3-030-17369-2\\_2](https://doi.org/10.1007/978-3-030-17369-2_2) (2019)
11. Galasso, F., Rizzini, D.L., Oleari, F., Caselli, S.: Efficient calibration of four wheel industrial AGVs. *Robot. Comput. Integr. Manuf.* **57**, 116–128 (2019). <https://doi.org/10.1016/j.rcim.2018.11.005>
12. Ganganath, N., Leung, H.: Mobile robot localization using odometry and kinect sensor. In: 2012 IEEE International Conference on Emerging Signal Processing Applications, pp. 91–94. <https://doi.org/10.1109/ESPA.2012.6152453> (2012)

13. Goronzy, G., Hellbrueck, H.: Weighted online calibration for odometry of mobile robots. In: 2017 IEEE International Conference on Communications Workshops (ICC Workshops), pp. 1036–1042. <https://doi.org/10.1109/ICCW.2017.7962795> (2017)
14. Han, K., Kim, H., Lee, J.S.: The Sources of Position Errors of Omni-Directional Mobile Robot with Mecanum Wheel. In: 2010 IEEE International Conference on Systems, Man and Cybernetics, pp. 581–586 (2010). <https://doi.org/10.1109/ICSMC.2010.5642009>
15. Igel, C., Hüsken, M.: Improving the rprop learning algorithm. In: Proceedings of the Second International ICSC Symposium on Neural Computation (NC 2000), pp. 115–121. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.17.3899&rep=rep1&type=pdf> (2000)
16. Ivanjko, E., Komšić, I., Petrović, I.: Simple off-line odometry calibration of differential drive mobile robots. In: Proceedings of 16th Int. Workshop on Robotics in Alpe-Adria-Danube Region - RAAD 2007. <https://www.researchgate.net/publication/268411270> (2007)
17. Jung, C., Chung, W.: Accurate calibration of two wheel differential mobile robots by using experimental heading errors. In: 2012 IEEE International Conference on Robotics and Automation, pp. 4533–4538. <https://doi.org/10.1109/ICRA.2012.6224660> (2012)
18. Jung, D., Seong, J., Moon, C., Jin, J., Chung, W.: Accurate calibration of systematic errors for car-like mobile robots using experimental orientation errors. *Int. J. Precis. Eng. Manuf.* **17**(9), 1113–1119 (2016). <https://doi.org/10.1007/s12541-016-0135-4>
19. Kallasi, F., Rizzini, D.L., Oleari, F., Magnani, M., Caselli, S.: A novel calibration method for industrial AGVs. *Robot. Auton. Syst.* **94**, 75–88 (2017). <https://doi.org/10.1016/j.robot.2017.04.019>
20. Lauer, M., Lange, S., Riedmiller, M.: Calculating the perfect match: an efficient and accurate approach for robot self-localization. In: RoboCup 2005: Robot Soccer World Cup IX, pp. 142–153. [https://doi.org/10.1007/11780519\\_13](https://doi.org/10.1007/11780519_13) (2006)
21. Leyard: Optitrack - Motion Capture Systems. <https://optitrack.com/>. Accessed on 17 Jan 2021
22. Lin, P., Liu, D., Yang, D., Zou, Q., Du, Y., Cong, M.: Calibration for odometry of omnidirectional mobile robots based on kinematic correction. In: 2019 14Th International Conference on Computer Science Education (ICCSE), Pp. 139–144. <https://doi.org/10.1109/ICCSE.2019.8845402> (2019)
23. Liu, J., Gao, W., Hu, Z.: Visual-Inertial Odometry Tightly Coupled with Wheel Encoder Adopting Robust Initialization and Online Extrinsic Calibration. In: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 5391–5397. <https://doi.org/10.1109/IROS40897.2019.8967607> (2019)
24. Maddahi, Y., Maddahi, A., Sepehri, N.: Calibration of omnidirectional wheeled mobile robots: Method and experiments. *Robotica* **31**(6), 969–980 (2013). <https://doi.org/10.1017/S0263574713000210>
25. Maddahi, Y., Sepehri, N., Maddahi, A., Abdolmohammadi, M.: Calibration of wheeled mobile robots with differential drive mechanisms: An experimental approach. *Robotica* **30**(6), 1029–1039 (2012). <https://doi.org/10.1017/S0263574711001329>
26. Martinelli, A., Tomatis, N., Tapus, A., Siegwart, R.: Simultaneous localization and odometry calibration for mobile robot. In: Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453), vol. 2, pp. 1499–1504. <https://doi.org/10.1109/IROS.2003.1248856> (2003)
27. Mondal, S., Yun, Y., Chung, W.K.: Terminal iterative learning control for calibrating systematic odometry errors in mobile robots. In: 2010 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, pp. 311–316. <https://doi.org/10.1109/AIM.2010.5695734> (2010)
28. Nagymáté, G., Kiss, R.M.: Application of Optitrack motion capture systems in human movement analysis: a systematic literature review. *Recent Innovations in Mechatronics* **5**(1), 1–9 (2018). <https://doi.org/10.17667/riim.2018.1/13>
29. Nemeč, D., Šimk, V., Janota, A., Hruboš, M., Bubeníková, E.: Precise localization of the mobile wheeled robot using sensor fusion of odometry, visual artificial landmarks and inertial sensors. *Robot. Auton. Syst.* **112**, 168–177 (2019). <https://doi.org/10.1016/j.robot.2018.11.019>
30. Siciliano, B., Sciavicco, L., Villani, L., Oriolo, G. *Robotics: Modelling, Planning and Control*, 1st edn. Springer, London (2009). <https://doi.org/10.1007/978-1-84628-642-1>
31. Siegwart, R., Nourbakhsh, I.R., Scaramuzza, D. *Introduction to Autonomous Mobile Robots*, 2nd edn. The MIT Press, Cambridge, Massachusetts (2011)
32. Sousa, R.B.: Odometry and Extrinsic Sensor Calibration on Mobile Robots. Master's Thesis, Faculty of Engineering of the University of Porto (FEUP), INESC TEC – Institute for Systems and Computer Engineering, Technology and Science, Porto, Portugal. <https://doi.org/10.13140/RG.2.2.27052.28802> (2020)
33. Sousa, R.B., Petry, M.R., Moreira, A.P.: Evolution of odometry calibration methods for ground mobile robots. In: 2020 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC), pp. 294–299. <https://doi.org/10.1109/ICARSC49921.2020.9096154> (2020)
34. Tomasi, D.L., Todt, E.: Rotational odometry calibration for differential robot platforms. In: 2017 Latin American Robotics Symposium (LARS) and 2017 Brazilian Symposium on Robotics (SBR), pp. 1–6. <https://doi.org/10.1109/SBR-LARS-R.2017.8215315> (2017)
35. Yang, Z., Shen, S.: Monocular visual-inertial state estimation with online initialization and camera-IMU extrinsic calibration. *IEEE Trans. Autom. Sci. Eng.* **14**(1), 39–51 (2017). <https://doi.org/10.1109/TASE.2016.2550621>
36. Yoo, K., Chung, W.: Convergence analysis of kinematic parameter calibration for a car-like mobile robot. In: 2009 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, pp. 740–745. <https://doi.org/10.1109/AIM.2009.5229924> (2009)
37. Yun, Y., Park, B., Chung, W.K.: Odometry calibration using home positioning function for mobile robot. In: 2008 IEEE International Conference on Robotics and Automation, pp. 2116–2121. <https://doi.org/10.1109/ROBOT.2008.4543519> (2008)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Ricardo B. Sousa** obtained a M.Sc. degree in electric and computers engineering at Faculty of Engineering of the University of Porto (FEUP), in 2020. He is currently working towards the Ph.D. degree in electrical and computer engineering with FEUP, and he has a graduate research scholarship from the Centre for Robotics in Industry and Intelligent Systems at INESC TEC. His research interests include robotics, sensor fusion, and localisation and mapping for autonomous robots.

**Marcelo R. Petry** is a robotics researcher and educator who joined INESC TEC's Centre for Robotics in Industry and Intelligent Systems in 2019. His research interests are in robot perception and localization, and the application of robots to logistics, inspection and human assistance. He earned his Ph.D. in Informatics Engineering from University of Porto in 2013. Previously, he was Assistant Professor at Federal University of Santa Catarina, and researcher at INESC P&D Brazil (2014 to 2019).

**Paulo G. Costa** received the M.Sc. and Ph.D. in Electrical and Computer Engineering on Faculty of Engineering of the University of Porto (FEUP), Portugal in 95 and 2000. He joined FEUP in 1992, and currently he is a Professor in the Electrical Engineering Department. He is also a senior researcher in Centre for Robotics in Industry and Intelligent Systems group of the INESC TEC. He has published more than a hundred papers in international scientific journals and conference proceedings. In addition, he participated and organized many autonomous mobile robotics competitions. Moreover, his research interests are in the field of robotics and automation: simulation, path planning, artificial vision, mobile robot localization and navigation, obstacle avoidance and perception.

**António Paulo Moreira** graduated with a degree in electrical engineering at the University of Oporto, in 1986. Then, he pursued graduate studies at University of Porto, obtaining a M.Sc. degree in electrical engineering – systems in 1991 and a Ph.D. degree in electrical engineering in 1998. Presently, he is Associate Professor with tenure at the Faculty of Engineering of the University of Porto and researcher and head of the Centre for Robotics in Industry and Intelligent Systems at INESC TEC. His main research interests are process control and robotics.

## Affiliations

**Ricardo B. Sousa**<sup>1,2</sup>  · **Marcelo R. Petry**<sup>2</sup>  · **Paulo G. Costa**<sup>1,2</sup>  · **António Paulo Moreira**<sup>1,2</sup> 

Marcelo R. Petry  
marcelo.petry@inesctec.pt

Paulo G. Costa  
paco@fe.up.pt

António Paulo Moreira  
amoreira@fe.up.pt

<sup>1</sup> Electrical Engineering Department, Faculty of Engineering of the University of Porto, Porto, 4200-465, Portugal

<sup>2</sup> CRIIS – Centre for Robotics in Industry and Intelligent Systems, INESC TEC – Institute for Systems and Computer Engineering, Technology and Science, Porto, 4200-465, Portugal