



Article

Online Odometry Calibration for Differential Drive Mobile Robots in Low Traction Conditions with Slippage

Carlo De Giorgi, Daniela De Palma  and Gianfranco Parlangei * 

Department of Engineering for Innovation, University of Salento, Via per Monteroni, 73100 Lecce, Italy; carlo.degiorgi@unisalento.it (C.D.G.); daniela.depalma@unisalento.it (D.D.P.)

* Correspondence: gianfranco.parlangei@unisalento.it; Tel.: +39-0832-297301

Abstract: This paper addresses a systematic method for odometry calibration of a differential-drive mobile robot moving on arbitrary paths in the presence of slippage and an algorithm encoding it which is well fit for online applications. It exploits the redundancy of sensors commonly available on ground mobile robots, such as encoders, gyroscopes, and IMU, to promptly detect slippage phenomena during the calibration process and effectively address their impact on odometry. The proposed technique has been validated through exhaustive numerical simulations and compared with other available odometry calibration methods. The simulation results confirm that the proposed methodology mitigates the impact of poor calibration, conducted without considering possible slipping phenomena, on reaching a target position, reducing the error by up to a maximum of 35 times. This restores the robot's performance to a calibration condition close to that of a slip-free scenario, confirming the effectiveness of the approach and its robustness against slippage phenomena.

Keywords: calibration and identification; kinematics; wheeled robots



Citation: De Giorgi, C.; De Palma, D.; Parlangei, G. Online Odometry Calibration for Differential Drive Mobile Robots in Low Traction Conditions with Slippage. *Robotics* **2024**, *13*, 7. <https://doi.org/10.3390/robotics13010007>

Academic Editors: Konstantinos Tsintotas, Loukas Bampis, Nitin J Sanket, Antonios Gasteratos, Giulio Sandini and Yiannis Aloimonos

Received: 31 October 2023
Revised: 17 December 2023
Accepted: 22 December 2023
Published: 27 December 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The analysis and development of control systems for autonomous vehicles has been a continuously active area of research. One of the key challenges for mobile devices lies in their capability to achieve a high level of autonomy. Accurate guidance, navigation, and control are of paramount importance for achieving autonomy. They basically rely on the performance of the robot's localization system. Indeed, localization plays a fundamental role in the navigation of mobile ground robots, allowing them to learn their position and orientation in the environment [1]. It is crucial for various applications, including autonomous exploration, path planning, and object manipulation.

To achieve accurate localization, sensor fusion techniques are commonly employed to integrate data from different sources and sensors. Among these techniques, odometry stands out; it is a widely adopted method of estimating the motion and position of a vehicle by combining data from the vehicle's odometer (which relies on wheel revolutions), and kinematic models.

However, odometry has inherent limitations. It assumes that wheel revolutions directly correspond to linear displacement, neglecting factors such as wheel slippage, terrain variations, and mechanical imperfections. Consequently, odometry can accumulate errors over time. Such errors can be categorized into systematic and non-systematic errors [2]. Systematic errors arise due to biases in the robot's kinematic model, wheel size discrepancies, or misalignment between the sensors and the robot's body. Non-systematic errors, on the other hand, are unpredictable and can occur during the robot's operation, such as slippage on low-friction surfaces. Both types of errors contribute to the overall inaccuracy of odometry and hinder the robot's precise localization.

Odometry calibration aims at identifying and correcting the kinematic parameters of the robot's motion model to improve localization accuracy. By accurately calibrating the

odometry system, systematic errors can be minimized, resulting in more reliable estimates of the robot's position and orientation. Nevertheless, during the odometry calibration procedure, non-systematic errors, such as slippage, can introduce additional uncertainties and errors. Slippage occurs when the wheels lose traction with the ground surface, leading to inconsistent wheel revolutions and inaccurate odometry measurements. Addressing and reducing these non-systematic errors, including slippage, is crucial for achieving a reliable and accurate localization system.

1.1. Odometry Calibration Literature Review

Over the years, numerous studies have focused on odometry calibration techniques. Recently, a review article [3] has been published on this topic, presenting a comparative survey of the existing techniques. The literature can be roughly categorized into three approaches [4]:

- (i) Offline One-Time Calibration (Before Use).
- (ii) Online Calibration, separate from Localization.
- (iii) Online Calibration Integrated into Robot Localization Algorithm.

Here, we briefly discuss the features of each approach.

1.1.1. Offline One-Time Calibration before Use

Papers falling into this area propose an offline calibration process that is performed once before deploying the robot, and typically involves predefined test paths or maneuvers to collect data for calibration. The work in [5] introduced the pioneer method for odometry calibration, which served as the basis for subsequent studies. This method, for the first time, enabled the calibration of differential robot parameters by combining rotations and translations in a square path. The authors of [6] further demonstrated that the path size influences the calibrated odometry accuracy, and proposed a 2×2 m square path for calibration. Since then, various approaches have emerged in the quest to identify the optimal calibration path. Researchers have explored different path designs and lengths to enhance the accuracy of calibrated odometry. These investigations have led to the introduction of alternative calibration methods and paths, such as the rotational motion-focused approach [7], the distance-based approach utilizing three points [8], and the bidirectional circular path test (BCPT) [9], as an alternative to square paths, aiming to reduce the probability of slippage and minimize effort during the calibration process. Ivanjko et al. [10] introduced a simplified calibration path consisting of straight-line motions and 180-degree rotations, offering a more straightforward approach with reduced space requirements. All of these methodologies, as exemplified in [11], have demonstrated remarkable effectiveness. It is worth noting, however, that these calibration paths often necessitate substantial space, leading to interruptions in the robot's regular usage. Moreover, conducting calibration tests in diverse environmental conditions, including various surfaces or terrains, poses significant challenges due to the extensive spatial requirements and difficulties in replicating real-world scenarios. These challenges highlight the need for calibration methods that can provide accurate results with minimal disruption to the robot's regular activities.

1.1.2. Online Calibration Separate from Localization

To overcome the limitations of previous methods, researchers have developed Online Calibration methodologies that enable the calibration of robot kinematic parameters during normal usage. In [12] odometry calibration is formulated as a linear problem, utilizing multiple calibration runs with different trajectories to compute an unbiased estimator. In [13] a terminal iterative learning control (TILC) algorithm is employed, requiring only one calibration run but relying on the chosen trajectory for odometry accuracy. However, it is important to consider that these methods encounter challenges in the absence of continuous position measurements along the entire trajectory. They are susceptible to

uncontrolled slippage phenomena, which can impact the calibration process. To enhance robustness against slippage, a significant number of calibration paths are often necessary.

1.1.3. Online Calibration Integrated into Robot Localization Algorithm

Another set of approaches integrates online calibration directly into the robot's localization fusion algorithm. These methods continuously update the calibration parameters based on real-time sensor measurements and fusion algorithms, improving the localization accuracy during robot operation. Ref. [14] presented a method that simultaneously calibrates odometry and extrinsic sensor parameters by leveraging the robot's relative motion throughout the entire trajectory. The work in [4] employed absolute localization systems and utilized the robot's position over the entire trajectory, resulting in a more complex calibration procedure. Additionally, several Kalman Filter-based methods have been proposed for odometry calibration, as for example [15–17] developed approaches that utilize Kalman filters and incorporate laser, DGPS, and inertial/magnetometer data, respectively. However, these approaches often require additional hardware, such as positioning systems for indoor environments like motion capture systems, and rarely provide direct position measurements as those offered by GPS systems for outdoor environments.

1.2. Paper Contribution and Organization

Motivated by the above considerations, in this work we present a novel online calibration method with enhanced robustness against slippage phenomena which is built on arbitrary paths commonly executed by robots during their operations. The proposed approach relies on an *active* slippage compensation during odometry calibration, built on a slippage detection system, which disconnects data collection from encoders for odometry calibration while slipping, and in turn triggers an IMU-based motion reconstruction algorithm to estimate and correct position data during slipping periods for odometry calibration. The basic idea is to maintain the original approach of [12], which is applied to an indoor differential robot, utilizing only proprioceptive sensor data, and eliminating the acquisition from signals affected by slippage when the traditional kinematic model becomes invalid. As long as slipping is detected, motion reconstruction is based on the only correct data available, namely IMU data. It is worth remarking that IMU-based short-time motion reconstruction is recently very active for the wide emerging applications (apart robotics), e.g., medicine, sports, rockfalls, snow avalanches and other risky environments where motion detection can be exploited for incipient disaster prediction [18,19]. The aim of the proposed methodology is to provide an online calibration algorithm that detects early slippage and effectively addresses its impact by utilizing gyroscopes and IMU, without the need for direct position measurements that may not always be available. A peculiar feature of the proposed algorithm is that it does not require a prescribed reference trajectory so that it can be run periodically to check eventual time variations of the kinematic matrix.

The rest of this paper is organized as follows. Section 2 theoretically formalizes the addressed problem. Section 3 presents our novel online calibration method with enhanced robustness against slippage phenomena. Exhaustive numerical simulations are reported in Section 4 to evaluate and validate its performance in different scenarios. Finally, concluding remarks are summarized in Section 5.

2. Problem Modeling and Formulation

2.1. Problem Modeling

Consider a differential-drive mobile robot as illustrated in Figure 1. It has two fixed wheels with a common axis of rotation and a caster wheel with the function of keeping the robot statically balanced. The two fixed wheels are controlled through their angular velocities, while the caster wheel is passive. Consider an inertial fixed reference frame \mathcal{I}_{xy} (see Figure 1) and a body-fixed reference frame $\mathcal{B}_{x^b y^b}$ centered in the midpoint of the line joining the two wheel centers, having the $^b x$ -axis aligned with the frontal direction of the robot body, and the $^b y$ -axis along its left lateral direction.

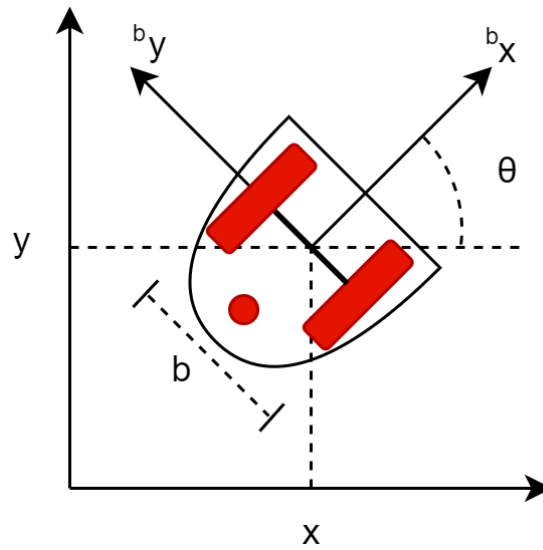


Figure 1. Differential-drive mobile robot.

The motion of the differential drive mobile robot in \mathcal{I} is described by a set of equations as:

$$\begin{cases} \dot{x} = v \cos \theta \\ \dot{y} = v \sin \theta \\ \dot{\theta} = \omega \end{cases} \quad (1)$$

where (x, y) are the coordinates of the midpoint of the two wheel centers, θ is the orientation of the fixed wheels, (hence, of the vehicle body), and v and ω represents, respectively, the driving and steering velocity of the vehicle. Under the assumption of pure rolling, these quantities can be expressed as a function of the wheel angular velocities:

$$\begin{cases} v = \frac{r_R \omega_R + r_L \omega_L}{2} \\ \omega = \frac{r_R \omega_R - r_L \omega_L}{b} \end{cases} \quad (2)$$

where ω_R and ω_L are the speed of the right and left wheel, respectively, r_R and r_L are the radii of the right and left wheel and b is the distance of their centers. Equation (2) can be rewritten in compact form as:

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = C \begin{bmatrix} \omega_R \\ \omega_L \end{bmatrix} \quad (3)$$

where $C \in \mathbb{R}^{2 \times 2}$ is given by:

$$C = \begin{bmatrix} \frac{r_R}{2} & \frac{r_L}{2} \\ \frac{r_R}{b} & \frac{r_L}{b} \end{bmatrix} \quad (4)$$

Denote by $\mathbf{q}_k = (x_k, y_k, \theta_k)$ the configuration of the robot, by (v_k, ω_k) the set of inputs at the sampling time t_k through Equation (2). The odometric localization of the vehicle is typically implemented by forward integration of the kinematic model in (1). Using the second-order Runge–Kutta integration method leads to [20]:

$$\begin{cases} x_{k+1} = x_k + \Delta_t v_k \cos(\theta_k + \Delta_t \omega_k / 2) \\ y_{k+1} = y_k + \Delta_t v_k \sin(\theta_k + \Delta_t \omega_k / 2) \\ \theta_{k+1} = \theta_k + \Delta_t \omega_k \end{cases} \quad (5)$$

being Δ_t the sampling period. The set of velocities (v_k, ω_k) is computed from the measurements of the wheel angular velocities $(\omega_{R,k}, \omega_{L,k})$ at t_k using Equation (3).

2.2. Problem Formulation

The above kinematic model, and in particular Equation (3), is valid only under the pure rolling condition, which cannot be assumed for granted in low friction or unknown environments. Considering such a realistic scenario, the goal of this paper is to propose an algorithm exploiting other sensors usually onboard on a ground mobile robot, and perform active online odometric calibration with corrected data whenever slippage condition is detected. More precisely, in this paper we assume that the robot is equipped with an IMU sensor which provides data on linear accelerations (on three axes in space), angular velocities and attitude.

In the following, we analyze calibration techniques for the estimation of kinematic parameters contained in the matrix C .

3. Online Calibration in Presence of Slippage

Inspired by the Least-Squares calibration approach [12] briefly summarized in Appendix A, we propose a methodology that takes into consideration slippage phenomena that may occur during the calibration process, making the analysis in [12] no longer valid. As in [12], we consider the following assumptions:

- A1. The path is divided into N steps.
- A2. Each step is completed in a fixed time Δ_t .
- A3. We have a constant ω_R, ω_L in each steps.
- A4. The initial and final poses are known from measurements (for instance, by using exteroceptive sensors at the start and the end of the considered trajectory).

As for example, consider an arbitrary path where slippage phenomena occur during some steps (see Figure 2). Without loss of generality, we assume that the slippage area, which is part of the main path, has a length of m steps, with $m < N$ where N is the number of steps of the total path length (see A1). In the following, we use the superscript ‘ ns ’ to indicate instances without slippage, and the superscript ‘ s ’ to indicate segments consisting of multiple steps of time Δ_t where there happens some slippage in that area.

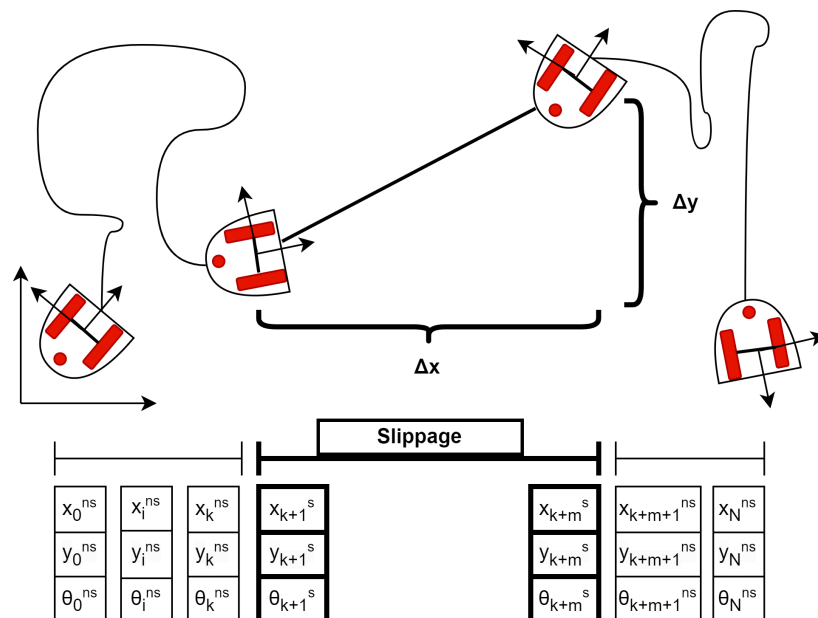


Figure 2. Example of trajectory with slippage and timeline.

In order to exploit the methodology expressed in [12] in the presence of slippage, Equation (5) can be rewritten as follows:

$$\begin{cases} x_{k+1} = x_k + \Delta x_k \\ y_{k+1} = y_k + \Delta y_k \\ \theta_{k+1} = \theta_k + \Delta \theta_k \end{cases} \quad (6)$$

where increments are defined as

$$\begin{aligned} \Delta x_k &= \begin{cases} \Delta x_k^{ns} = \Delta x_k^{odo} & \text{in absence of slippage,} \\ \Delta x_k^s = \Delta x_k^{imu} & \text{otherwise.} \end{cases} \\ \Delta y_k &= \begin{cases} \Delta y_k^{ns} = \Delta y_k^{odo} & \text{in absence of slippage,} \\ \Delta y_k^s = \Delta y_k^{imu} & \text{otherwise.} \end{cases} \\ \Delta \theta_k &= \begin{cases} \Delta \theta_k^{ns} = \Delta \theta_k^{odo} & \text{in absence of slippage,} \\ \Delta \theta_k^s = \Delta \theta_k^{imu} & \text{otherwise.} \end{cases} \end{aligned}$$

and they can be computed either through ordinary odometry inference, whenever slippage is absent, or, in the presence of slippage, exploiting different other sensors (as described in the following of this Section).

3.1. Slippage Detection

Slippage can cause inconsistent measurements between the instantaneous displacement obtained from odometry and the actual displacement value. Slippage is primarily caused by lateral and longitudinal forces resulting from the vehicle's motion, and low coefficient of friction. Though we can attempt to reduce these forces, slipping is inevitable when there is an unexpected change in the static coefficient of friction. There are several methodologies to detect slipping, such as comparing the kinematic model with pure rolling to another one that includes slipping.

Assuming to have access to an inertial measurement unit (IMU) mounted on board the vehicle, and assuming that the reference system of the IMU is aligned with the robot, an acceleration measurement ${}^b \mathbf{a}_k$ can be acquired from the system. In case of lateral slipping, the analysis of the acceleration ${}^b \mathbf{a}_k = [{}^b a_{x,k}, {}^b a_{y,k}]^\top$ obtained from the IMU can reveal a non-zero value inconsistent with the differential model:

$${}^b a_{y,k} - [0 \ 1 \ 0] {}^b \mathbf{R}_k \mathbf{g} = 0 \quad (7)$$

being ${}^b \mathbf{R}_k \in SO(3)$ the rotation matrix for the coordinates' transformation from the reference frame \mathcal{I} to the body-fixed reference frame \mathcal{B} obtained from angular velocity measurements, and \mathbf{g} the gravity vector in \mathcal{I} [18].

On the other hand, the presence of frontal slipping, which may occur due to braking or acceleration, can be verified by checking the consistency between the differential model and the data from inertial accelerations: It could be verified whether there is consistency between the data of inertial acceleration in the presence of frontal slipping, which may occur due to braking or acceleration:

$$c_{2,1}(\omega_{R,k} - \omega_{R,k-1}) + c_{2,2}(\omega_{L,k} - \omega_{L,k-1}) - {}^b a_{x,k} \Delta t = 0 \quad (8)$$

The violation of one of these conditions denotes the presence of slippage in the considered sub-track.

Remark 1. It should be noted that zero equality in the Equations (7) and (8) is only theoretical. From a practical point of view, a certain tolerance should be considered regarding the measurement error, namely:

$$|{}^b a_{y,k}^{IMU} - [0 \ 1 \ 0] {}^b \mathbf{R}_k^{IMU} \mathbf{g}| \leq \epsilon_L^{IMU} \quad (9)$$

and

$$|c_{2,1}(\omega_{R,k} - \omega_{R,k-1}) + c_{2,2}(\omega_{L,k} - \omega_{L,k-1}) - {}^b a_{x,k}^{IMU} \Delta t| \leq \epsilon_F^{IMU} \quad (10)$$

where $\epsilon_L^{IMU}, \epsilon_F^{IMU}$ are suitable thresholds that can be set based on the IMU data sheet, combining the in-run bias stability and noise, as explained in detail in [21]. Such parameters are indeed a criterion for the selection of an IMU mostly adequate for the proposed algorithm (which is expected to last an amount of time of the order of a minute). Finally, to avoid false alarms caused by noise, violation of conditions (9) and (10) is checked for few steps instead of only one.

3.2. Slippage Estimation

Whenever a slippage condition is detected, the algorithm triggers an IMU-based motion reconstruction procedure to exploit the only correct data available and estimate the induced spatial displacements to further data correction. To this end, in this Section, we derive a procedure for the motion reconstruction based on IMU data only which has been recently developed in [18,19], and we briefly describe in the following. The inertial Unit provides the vehicle acceleration measurement ${}^b \mathbf{a}_{k\ imu}$ expressed in the body-fixed reference frame \mathcal{B} , while, conversely, modern gyroscopes are able to provide absolute orientation with accuracy of the order of 0.1 degrees, allowing direct measurement of the angle θ in \mathcal{I} , and consequently to quickly identify the value of $\Delta\theta$. For sake of clarity, we assume that slipping occurs in a single area consisting of m segments, namely:

$$\begin{cases} \Delta x^s = \sum_{i=k+1}^{k+m} \Delta x_i^s = \sum_{i=k+1}^{k+m} \Delta x_i^{imu} \\ \Delta y^s = \sum_{i=k+1}^{k+m} \Delta y_i^s = \sum_{i=k+1}^{k+m} \Delta y_i^{imu} \\ \Delta\theta^s = \sum_{i=k+1}^{k+m} \Delta\theta_i^s = \sum_{i=k+1}^{k+m} \Delta\theta_i^{gyro} \end{cases} \quad (11)$$

We denote θ_k and v_k resp. the orientation and forward velocity of the robot at the beginning of each slip segment. Since during slipping, there is no friction between the robot wheels and the ground, to reconstruct the motion in these m segments, we adopt a uniformly accelerated model based on IMU data and we recur to the approach of [18]:

$$\mathbf{v}_i = \mathbf{R}_k \begin{bmatrix} {}^b v_k \\ 0 \end{bmatrix} + \sum_{h=k+1}^i \Delta t (\mathbf{R}_h {}^b \mathbf{a}_h - \mathbf{g}), \quad i = k + 1, \dots, k + m \quad (12)$$

$$\begin{bmatrix} \Delta x^s \\ \Delta y^s \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \left(\sum_{i=k+1}^{k+m} \Delta t \mathbf{v}_i + \frac{\Delta t^2 (\mathbf{R}_i {}^b \mathbf{a}_i - \mathbf{g})}{2} \right) \quad (13)$$

being $\mathbf{R} \in SO(3)$ the rotation matrix for the coordinates' transformation from the body-fixed reference frame \mathcal{B} to the reference frame \mathcal{I} obtained from angular velocity measurements [18], ${}^b v_k$ the vehicle's velocity in the body-fixed reference frame \mathcal{B} obtained from encoders measurements, ${}^b \mathbf{a}_h$ the vehicle's acceleration measurement in body reference frame \mathcal{B} obtained from IMU, and \mathbf{g} the gravity vector in \mathcal{I} [18].

Equations (11)–(13) together with Equation (17), allow for a proper exploitation of the heterogeneous data gathered from the sensors, together with the kinematics of the robot, to have an accurate estimation of robot localization through Equations (22) and (23) notwithstanding the possible presence of slippage.

Remark 2. Also, in this case, Equations (12) and (13) are computed using data $\mathbf{R}_h^{IMU}, {}^b \mathbf{a}_h^{IMU}$ obtained through IMU acquisition. Combining Equations (12) and (13) with a sensor output model as Equation (1) of \mathbf{v}_i [21] allow also for easy computation of the maximum estimation error of $\mathbf{v}_i, \Delta x^s$ and Δy^s starting from the IMU measurement of $\mathbf{R}_h, {}^b \mathbf{a}_h$.

3.3. Slippage-Compensated Odometric Parameters Estimation

Consider now the overall displacement, from time 0 to $N \cdot \Delta t$ along the path. Under the pure rolling condition, it is expressed by Equation (A1) in Appendix A. Taking into account

the possible presence of slippage, Equation (A1) can be reorganized introducing one term that encloses the possible presence of slippage:

$$\theta_N - \theta_0 = \Delta_t c_{21} \sum_{i=0}^k \omega_{Ri} + \Delta_t c_{22} \sum_{i=0}^k \omega_{Li} + \Delta\theta^s + \sum_{k+m+1}^{N-1} \omega_{Ri} + \Delta_t c_{22} \sum_{k+m+1}^{N-1} \omega_{Li} \quad (14)$$

$$\begin{aligned} x_N - x_0 = & \Delta_t c_{11} \sum_{i=1}^k \omega_{Ri} \cos(\theta_i + \Delta_t \omega_i / 2) + \Delta_t c_{12} \sum_{i=1}^k \omega_{Li} \cos(\theta_i + \Delta_t \omega_i / 2) + \Delta x^s + \\ & + \Delta_t c_{11} \sum_{i=k+m+1}^{N-1} \omega_{Ri} \cos(\theta_i + \Delta_t \omega_i / 2) + \Delta_t c_{12} \sum_{i=k+m+1}^{N-1} \omega_{Li} \cos(\theta_i + \Delta_t \omega_i / 2) \end{aligned} \quad (15)$$

$$\begin{aligned} y_N - y_0 = & \Delta_t c_{11} \sum_{i=1}^k \omega_{Ri} \sin(\theta_i + \Delta_t \omega_i / 2) + \Delta_t c_{12} \sum_{i=1}^k \omega_{Li} \sin(\theta_i + \Delta_t \omega_i / 2) + \Delta y^s + \\ & + \Delta_t c_{11} \sum_{i=k+m+1}^{N-1} \omega_{Ri} \sin(\theta_i + \Delta_t \omega_i / 2) + \Delta_t c_{12} \sum_{i=k+m+1}^{N-1} \omega_{Li} \sin(\theta_i + \Delta_t \omega_i / 2) \end{aligned} \quad (16)$$

It is worth noticing that the displacements Δx^s , Δy^s , $\Delta\theta^s$ in Equations (14)–(16), do not depend on the unknown odometric parameters, but they can be considered known quantities under the assumptions of measurements from different sensors onboard. This allows us to adapt the original Least-Squares approach by keeping the information of all the non-corrupted sections, and by correcting the final position through the increments measured in the slippage sections only.

In order to exploit the above logic line, we introduce the following notation:

$$\begin{cases} \theta' = \theta_N - \Delta\theta^s - \theta_0 \\ x' = x_N - \Delta x^s - x_0 \\ y' = y_N - \Delta y^s - y_0 \end{cases} \quad (17)$$

the following equations hold, which resemble the proposed no-slippage approach:

$$\theta' = \Phi'_\theta \begin{bmatrix} c_{21} \\ c_{22} \end{bmatrix} \quad (18)$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \Phi'_{xy} \begin{bmatrix} c_{11} \\ c_{12} \end{bmatrix} \quad (19)$$

where the regressors Φ'_{xy} and Φ'_θ can be easily deduced from Equations (14)–(16).

The procedure is then repeated for multiple experiments collecting data from P suitable trajectories, measurement data are stacked, and the new regressor is obtained (similarly to the original approach that led to Equations (A6) and (A7)):

$$\begin{bmatrix} x'_1 \\ y'_1 \\ \vdots \\ x'_p \\ y'_p \end{bmatrix} = \overline{\Phi'}_{xy} \begin{bmatrix} c_{11} \\ c_{12} \end{bmatrix} \quad (20)$$

$$\begin{bmatrix} \theta'_1 \\ \vdots \\ \theta'_p \end{bmatrix} = \overline{\Phi'}_\theta \begin{bmatrix} c_{21} \\ c_{22} \end{bmatrix} \quad (21)$$

Finally, through a suitable number of random trajectories, it is possible to estimate the parameters c_{ij} :

$$\begin{bmatrix} \hat{c}_{11} \\ \hat{c}_{12} \end{bmatrix} = \left(\overline{\Phi'_{xy}}^T \overline{\Phi'_{xy}} \right)^{-1} \overline{\Phi'_{xy}}^T \begin{bmatrix} x'_1 \\ y'_1 \\ \vdots \\ x'_P \\ y'_P \end{bmatrix} \tag{22}$$

$$\begin{bmatrix} \hat{c}_{21} \\ \hat{c}_{22} \end{bmatrix} = \left(\overline{\Phi'_{\theta}}^T \overline{\Phi'_{\theta}} \right)^{-1} \overline{\Phi'_{\theta}}^T \begin{bmatrix} \theta'_1 \\ \vdots \\ \theta'_P \end{bmatrix} \tag{23}$$

An overview of the whole calibration procedure described in this section is shown in the flowchart in Figure 3.

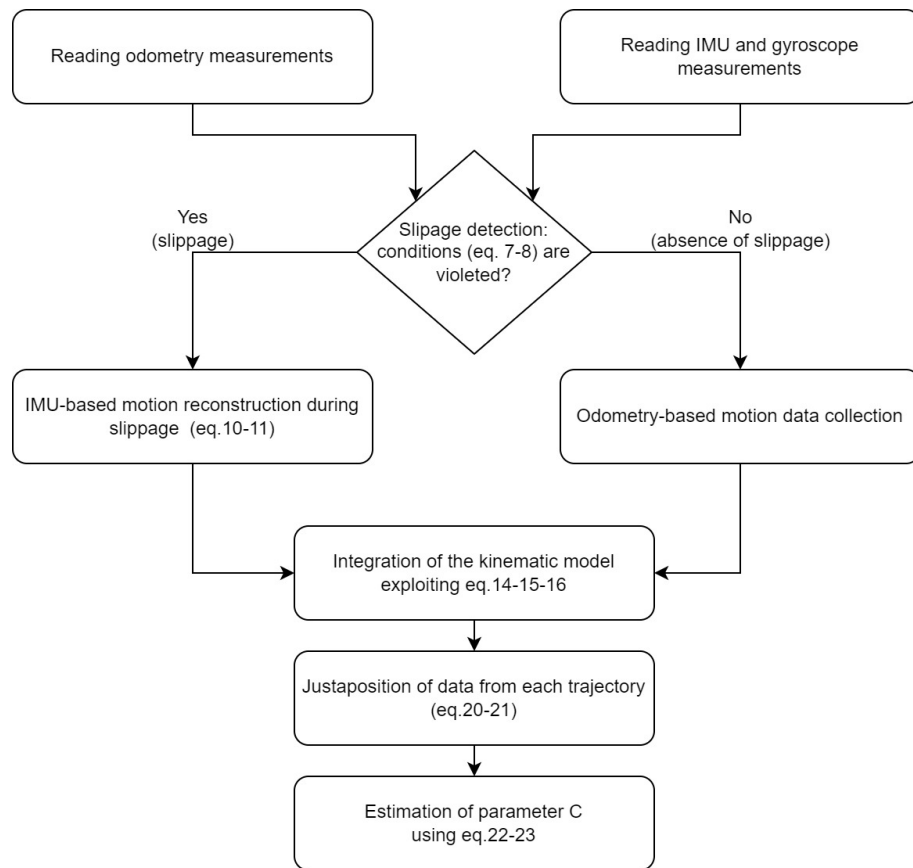


Figure 3. Overview of the calibration procedure that derives slippage-compensated odometric parameters estimation.

Remark 3. If IMU data sheets are available, Equations (17), (22) and (23) can be equipped with the maximum error induced by the IMU measurements. Indeed, following the direction of Remark 2, setting $\theta' = \theta'^{IMU} + \delta_{\theta}$ where θ'^{IMU} is computed using the measured values and δ_{θ} is the estimation error (whose bound can be computed as suggested in Remark 2), using (23) it is possible to find the maximum estimation error of \hat{c}_{21} and \hat{c}_{22} caused by the IMU measurements. Similar considerations hold for x' and y' as well, making it possible, using (22), to find the maximum estimation error of \hat{c}_{11} and \hat{c}_{12} caused by the IMU measurements .

4. Numerical Simulations

The proposed methodology is validated through MATLAB simulations, providing important insights for the next experimental phase. Indeed, the final objective of this numerical simulation campaign is to assess the performance of the proposed approach in a controlled environment where the ground truth is exactly known.

We are interested in testing the online calibration method on ‘arbitrary’ paths that could be executed during standard operating activities of a mobile robot. To this aim, a two-dimensional grid including obstacles is considered, and a set of P free paths from an initial to a target position are simulated. We assume, without loss of generality, that the robot is equipped with a path planner to autonomously build these paths; the most common planners typically used in robotic applications are the rapidly exploring random tree (RRT) planner and the Probabilistic Road map (PRM) planner.

In order to test the proposed method, an arbitrary slippage is introduced by modifying the nominal absolute velocities of the robot proportionally to an arbitrary value α . Figure 4 shows one example of the P trajectories. The slip-free trajectory is depicted in blue, while the trajectory affected by the presence of slippage is depicted in red. In particular, the red path is obtained by arbitrarily modifying the nominal absolute velocities \dot{x} (increased by a factor $\alpha = 2$) and \dot{y} (reduced by a factor $\alpha = -0.2$). The resulting velocities generate frontal and lateral accelerations inconsistent with the odometric model as described in Section 3.1. This allows a direct comparison between the ideal case in the absence of slippage (denoted by ‘*ns*’ in the following) and the two case studies where: (i) slippage is present but unidentified (denoted by ‘*suc*’), and (ii) slippage is present, properly estimated and managed with the proposed methodology (denoted by ‘*sc*’).

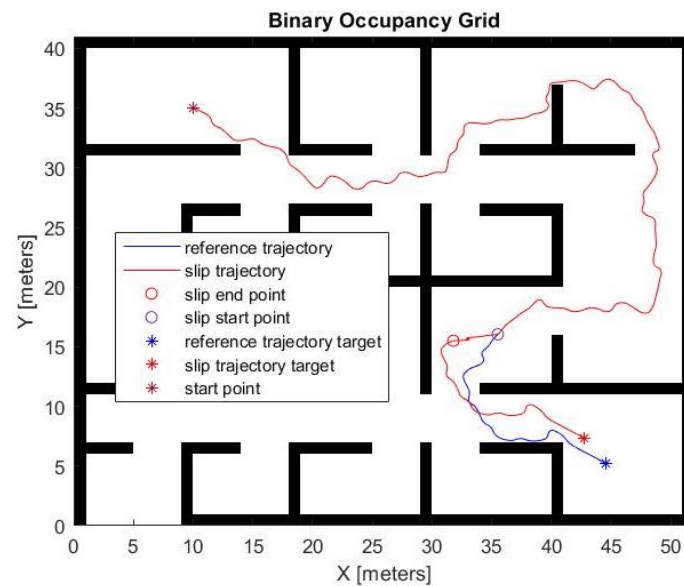


Figure 4. Example of trajectory with slippage, the slip-free trajectory is depicted in blue, while the trajectory affected by the presence of slippage is depicted in red.

The presence of slippage is simulated for an interval T_s , after that the robot proceeds in both simulated scenarios with the same nominal speed control signal, reaching two final target positions translated by a quantity Δ_x, Δ_y .

In order to ensure comparability of the trajectories, it is assumed that both trajectories reach their final target position and that the starting point and the two endpoints are not subject to any measurement error.

To perform a realistic test of the algorithm, nominal inputs and sensor measurements are perturbed by introducing a zero mean white Gaussian noise to the signals, as described in the following. The acquisition of the θ signal is simulated with a signal-to-noise ratio of 30 dB. The estimation of the slippage Δ_x^s, Δ_y^s from IMU and gyro measurements is simulated

with a signal-to-noise ratio of 30 dB. The acquisition of the wheels' angular velocity ω_R and ω_L signals from encoders is simulated with a signal-to-noise ratio of 50 dB. The simulation is undertaken assuming a sampling period $T = 0.1$ [s] and the matrix C containing the odometric parameters is equal to:

$$C = \begin{bmatrix} 0.0750 \text{ m} & 0.0750 \text{ m} \\ 0.0833 & -0.0833 \end{bmatrix}. \quad (24)$$

In the three scenarios (ideal '*ns*', presence of slippage but unidentified '*suc*', and presence of slippage properly estimated and managed '*sc*'), we compare the results obtained in the presence of noise measurements, or in its absence, considering the two different planners commonly used in robotic applications, i.e., the PRM planner and the RRT-Dubins planner (namely, the RRT planner where each path is made of Dubins curves with maximum turning radius equal to 0.5 m).

Table 1 shows the estimations of the odometric parameters \hat{C} and the corresponding estimation error $\tilde{C} = C - \hat{C}$ obtained in the absence of measurement noise considering $P = 12$ trajectories generated by the RRT-Dubins planner.

Table 1. Odometric parameter estimation \hat{c}_{ij} in absence and presence of slippage for $T_s = 10$ s considering $P = 12$ noise-free trajectories generated using the RRT-Dubins planner.

Scenario	\hat{C}	\tilde{C}
<i>ns</i>	$\begin{bmatrix} 0.0757 \text{ m} & 0.0742 \text{ m} \\ 0.0833 & -0.0833 \end{bmatrix}$	$\begin{bmatrix} 7.5 \times 10^{-4} & -7.0 \times 10^{-4} \\ 3.3 \times 10^{-5} & -3.3 \times 10^{-5} \end{bmatrix}$
<i>suc</i>	$\begin{bmatrix} 0.1216 \text{ m} & 0.0225 \text{ m} \\ 0.0833 & -0.0833 \end{bmatrix}$	$\begin{bmatrix} 4.6 \times 10^{-2} & -5.2 \times 10^{-2} \\ 3.3 \times 10^{-5} & -3.3 \times 10^{-5} \end{bmatrix}$
<i>sc</i>	$\begin{bmatrix} 0.0751 \text{ m} & 0.0746 \text{ m} \\ 0.0833 & -0.0833 \end{bmatrix}$	$\begin{bmatrix} 1.7 \times 10^{-4} & -3.1 \times 10^{-4} \\ 7.3 \times 10^{-5} & -7.1 \times 10^{-5} \end{bmatrix}$

As expected, in the presence of slippage, proper detection of slippage occurrences and proper management of the measurements acquired during such steps using the proposed method allows for estimating the odometric parameters with higher precision than the conventional approach that does not take into account (by construction) any possible occurrence of slippage.

The effectiveness of the proposed methodology is also clear in the presence of noisy measurements. Figures 5 and 6 report an example of the noisy angular measurements acquired during a simulated trajectory, and Figure 7 displays the relative displacements.

Table 2 shows the estimated values of the odometric parameters \hat{C} and the corresponding estimation error \tilde{C} , obtained in the presence of measurement noise considering $P = 12$ trajectories generated by the RRT-Dubins planner in the three different scenarios: ideal with noisy measurements (denoted as '*ns-noisy*'), presence of slippage but unidentified with noisy measurements (denoted as '*suc-noisy*'), and the presence of slippage properly estimated and managed with noisy measurements ('*sc-noisy*').

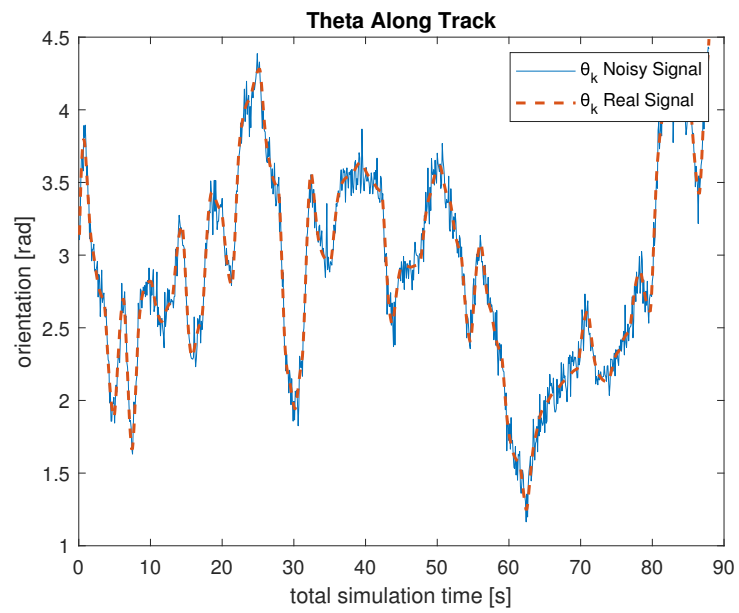


Figure 5. Example of θ noisy measurements acquired during a trajectory.

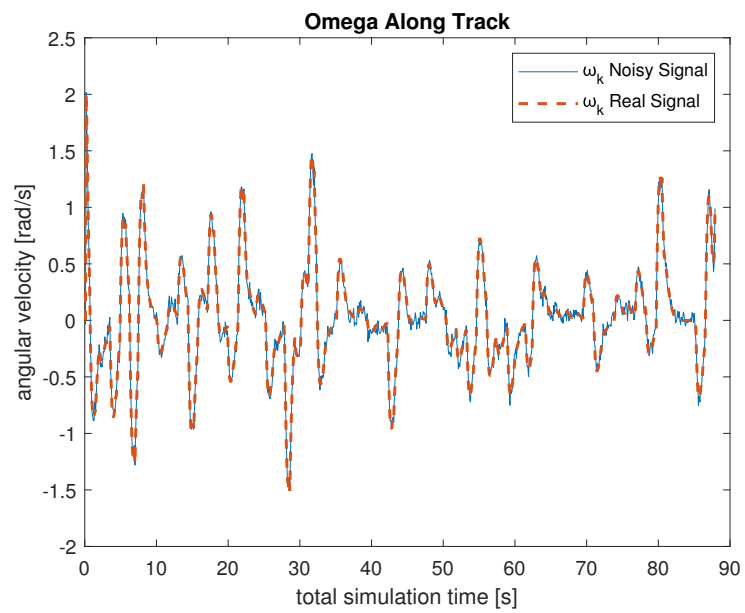


Figure 6. Example of ω noisy measurements acquired during a trajectory.

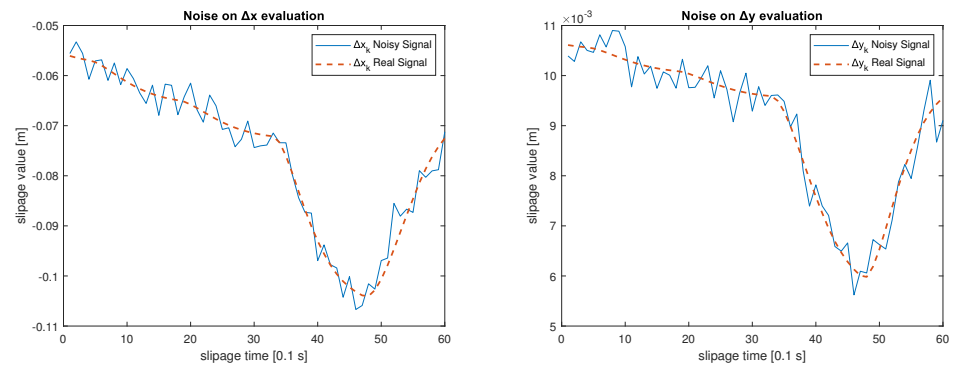


Figure 7. Example of Δ_x , Δ_y noisy measurements acquired during a trajectory.

Table 2. Odometric parameter estimation \hat{c}_{ij} in the absence and presence of slippage for $T_s = 10$ s considering $P = 12$ noisy trajectories generated using the RRT-Dubins planner.

Scenario	\hat{C}	\tilde{C}
<i>ns – noisy</i>	$\begin{bmatrix} 0.0753 \text{ m} & 0.0756 \text{ m} \\ 0.0833 & -0.0833 \end{bmatrix}$	$\begin{bmatrix} 3.7 \times 10^{-4} & -6.0 \times 10^{-4} \\ 7.8 \times 10^{-5} & -7.7 \times 10^{-5} \end{bmatrix}$
<i>suc – noisy</i>	$\begin{bmatrix} 0.1325 \text{ m} & 0.0130 \text{ m} \\ 0.0833 & -0.0833 \end{bmatrix}$	$\begin{bmatrix} 5.7 \times 10^{-2} & -6.1 \times 10^{-2} \\ 7.8 \times 10^{-5} & -7.7 \times 10^{-5} \end{bmatrix}$
<i>sc – noisy</i>	$\begin{bmatrix} 0.0763 \text{ m} & 0.0740 \text{ m} \\ 0.0833 & -0.0833 \end{bmatrix}$	$\begin{bmatrix} 1.3 \times 10^{-3} & -9.8 \times 10^{-4} \\ 4.8 \times 10^{-5} & -5.3 \times 10^{-5} \end{bmatrix}$

Tables 3 and 4 show the results of analogous simulations (both in the absence and presence of measurement noise for the different scenarios), when a PRM planner is adopted.

Table 3. Odometric parameter estimation \hat{c}_{ij} in the absence and presence of slippage for $T_s = 10$ s considering $P = 12$ noise-free trajectories generated by the PRM motion planner.

Scenario	\hat{C}	\tilde{C}
<i>ns</i>	$\begin{bmatrix} 0.0753 \text{ m} & 0.0745 \text{ m} \\ 0.0833 & -0.0833 \end{bmatrix}$	$\begin{bmatrix} 3.8 \times 10^{-4} & -4.5 \times 10^{-4} \\ 3.3 \times 10^{-5} & -3.3 \times 10^{-5} \end{bmatrix}$
<i>suc</i>	$\begin{bmatrix} 0.1335 \text{ m} & 0.0161 \text{ m} \\ 0.0833 & -0.0833 \end{bmatrix}$	$\begin{bmatrix} 5.8 \times 10^{-2} & -5.8 \times 10^{-2} \\ 3.3 \times 10^{-5} & -3.3 \times 10^{-5} \end{bmatrix}$
<i>sc</i>	$\begin{bmatrix} 0.0753 \text{ m} & 0.0745 \text{ m} \\ 0.0833 & -0.0833 \end{bmatrix}$	$\begin{bmatrix} 3.7 \times 10^{-4} & -4.8 \times 10^{-4} \\ 2.3 \times 10^{-5} & -2.8 \times 10^{-5} \end{bmatrix}$

Table 4. Odometric parameter estimation \hat{c}_{ij} in the absence and presence of slippage for $T_s = 10$ s considering $P = 12$ noisy trajectories generated by the PRM motion planner.

Scenario	\hat{C}	\tilde{C}
<i>ns – noisy</i>	$\begin{bmatrix} 0.0740 \text{ m} & 0.0767 \text{ m} \\ 0.0833 & -0.0833 \end{bmatrix}$	$\begin{bmatrix} 9.0 \times 10^{-4} & -1.7 \times 10^{-3} \\ 4.6 \times 10^{-5} & -2.8 \times 10^{-5} \end{bmatrix}$
<i>suc – noisy</i>	$\begin{bmatrix} 0.1322 \text{ m} & 0.0183 \text{ m} \\ 0.0833 & -0.0833 \end{bmatrix}$	$\begin{bmatrix} 5.7 \times 10^{-2} & -5.6 \times 10^{-2} \\ 4.6 \times 10^{-5} & -2.8 \times 10^{-5} \end{bmatrix}$
<i>sc – noisy</i>	$\begin{bmatrix} 0.0800 \text{ m} & 0.0710 \text{ m} \\ 0.0833 & -0.0833 \end{bmatrix}$	$\begin{bmatrix} 5.0 \times 10^{-3} & -3.9 \times 10^{-3} \\ -4.4 \times 10^{-5} & 4.9 \times 10^{-5} \end{bmatrix}$

Figures 8 and 9 report an example of possible trajectories, both in the presence and absence of slippage, generated by the RRT-Dubins motion planner and the corresponding vehicle's orientation during the motion. Next, Figures 10 and 11 report an example of possible trajectories, in the presence and absence of slippage, generated by the PRM motion planner and the corresponding vehicle's orientation during the motion.

Analyzing the results obtained in the different scenarios with the two motion planners, we can infer that if the number of analyzed paths increases, the use of different motion planners has a reduced impact on the estimation. However, when paths are characterized by significant variations in the orientation θ , a little improvement in estimation can be observed. The results of these trials show how the use of curved trajectories, such as those generated by the RRT-Dubins motion planner, allows for slightly better calibration results. For this reason, in the following, we proceeded with the simulation campaign considering only the RRT-Dubins motion planner.

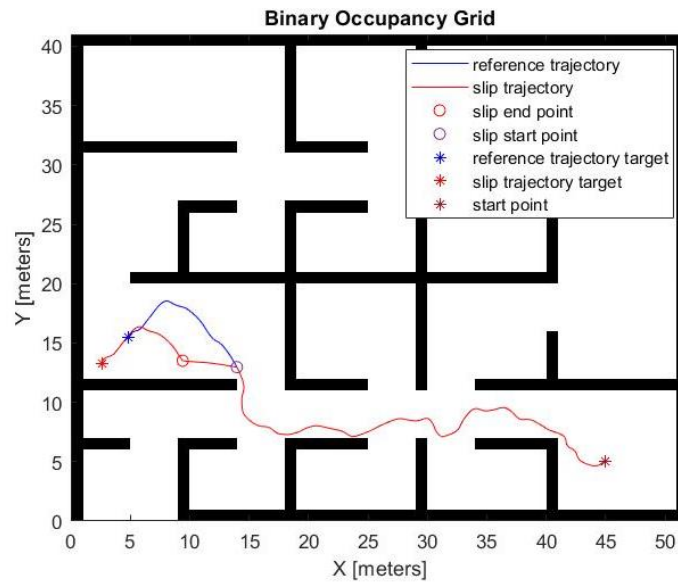


Figure 8. Example of path generated by the RRT-Dubins motion planner.

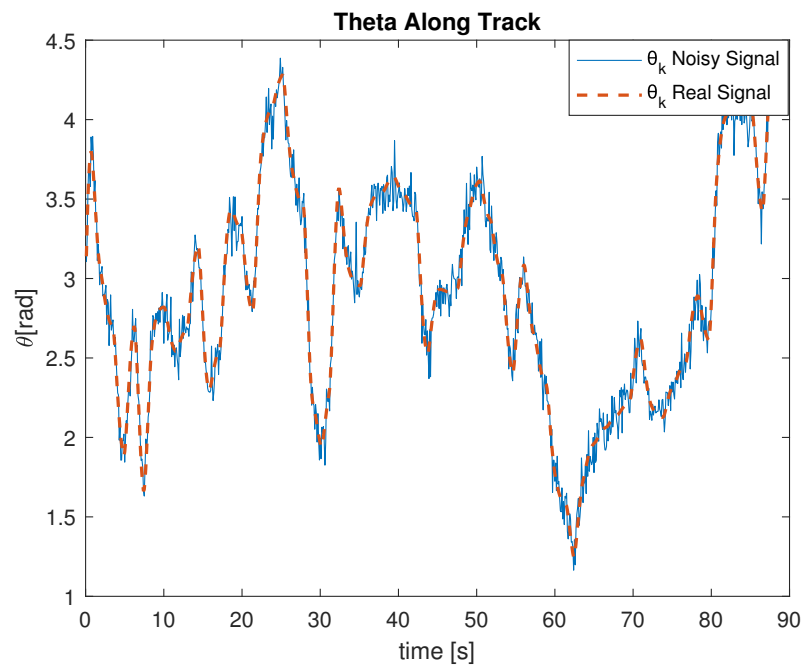


Figure 9. Vehicle’s orientation θ during the trajectory in Figure 8 generated by the RRT-Dubins motion planner.

To conclude the analysis, we investigated the impact of the methodology in the presence of slippage for an interval T_s of varying entities. In particular, in order to assess the resilience of the two methodologies in the absence and presence of slippage and noise measurement, we estimated the matrix C assuming different values for $T_s = 10, 6,$ and 4 s . The results are reported in Table 5. Analyzing the proposed methodology as T_s varies, it is observed that an increase in the slip time has the effect of a deterioration of the estimation. Nevertheless, even in the presence of considerable slippage and noisy signals, the proposed methodology allows obtaining an estimation of matrix C_{sc} close to the real one, and thus performances similar to the estimation in the scenario without slipping C_{ns} .

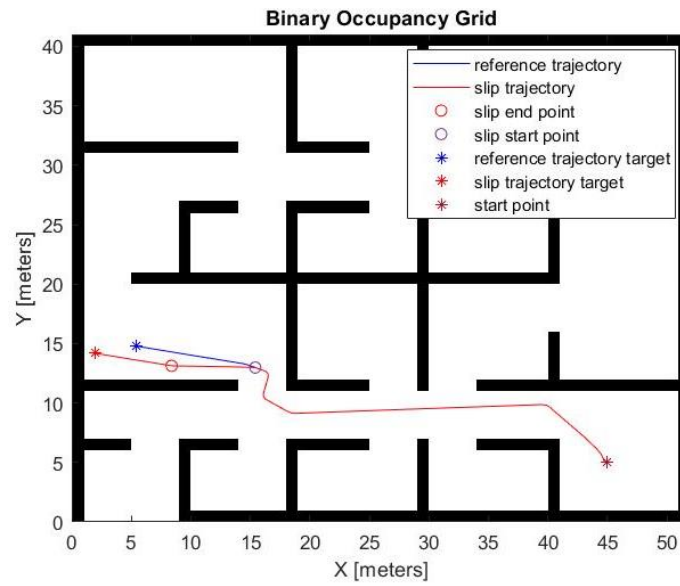


Figure 10. Example of path generated by the PRM motion planner.

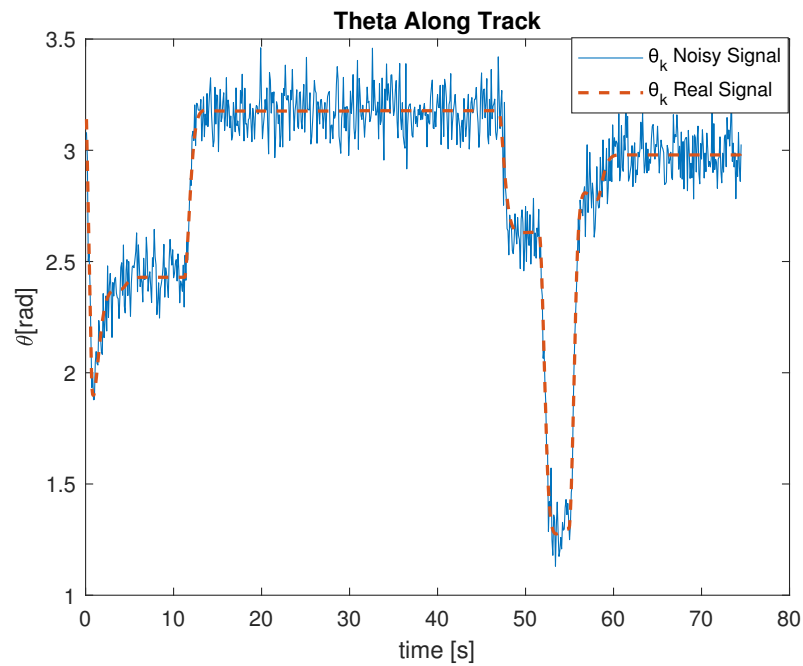


Figure 11. Vehicle’s orientation θ during the trajectory in Figure 10 generated by the PRM motion planner.

As a final remark, it is worth highlighting that exhaustive numerical experiments revealed a significant detrimental effect on the calibration accuracy when the measurement noise on the orientation θ , which is related to the gyroscope signals, increases, and the same holds considering the slippage Δ_x^s, Δ_y^s , which is related to the measurements obtained through both gyroscope and IMU. On the other hand, the impact is minimal when the measurement noise increases on a single wheel velocity.

Table 5. Odometric parameter estimation \hat{c}_{ij} in the absence and presence of slippage for a varying time interval T_s considering $P = 12$ arbitrary trajectories generated by the RRT-Dubins motion planner.

Scenario	T_s	$T_s = 10\text{ s}$	$T_s = 6\text{ s}$	$T_s = 4\text{ s}$
\hat{C}_{ns}		$\begin{bmatrix} 0.0757\text{ m} & 0.0742\text{ m} \\ 0.0833 & -0.0833 \end{bmatrix}$	$\begin{bmatrix} 0.0755\text{ m} & 0.0744\text{ m} \\ 0.0833 & -0.0833 \end{bmatrix}$	$\begin{bmatrix} 0.0754\text{ m} & 0.0756\text{ m} \\ 0.0833 & -0.0833 \end{bmatrix}$
\hat{C}_{suc}		$\begin{bmatrix} 0.1216\text{ m} & 0.0225\text{ m} \\ 0.0833 & -0.0833 \end{bmatrix}$	$\begin{bmatrix} 0.1158\text{ m} & 0.03180\text{ m} \\ 0.0833 & -0.0833 \end{bmatrix}$	$\begin{bmatrix} 0.0978\text{ m} & 0.0481\text{ m} \\ 0.0833 & -0.0833 \end{bmatrix}$
\hat{C}_{sc}		$\begin{bmatrix} 0.0751\text{ m} & 0.0746\text{ m} \\ 0.0833 & -0.0833 \end{bmatrix}$	$\begin{bmatrix} 0.0755\text{ m} & 0.0744\text{ m} \\ 0.0833 & -0.0833 \end{bmatrix}$	$\begin{bmatrix} 0.0755\text{ m} & 0.0745\text{ m} \\ 0.0833 & -0.0833 \end{bmatrix}$
$\hat{C}_{ns-noisy}$		$\begin{bmatrix} 0.0753\text{ m} & 0.0756\text{ m} \\ 0.0833 & -0.0833 \end{bmatrix}$	$\begin{bmatrix} 0.0777\text{ m} & 0.0725\text{ m} \\ 0.0833 & -0.0833 \end{bmatrix}$	$\begin{bmatrix} 0.0792\text{ m} & 0.0711\text{ m} \\ 0.0833 & -0.0833 \end{bmatrix}$
$\hat{C}_{suc-noisy}$		$\begin{bmatrix} 0.1325\text{ m} & 0.1301\text{ m} \\ 0.0833 & -0.0833 \end{bmatrix}$	$\begin{bmatrix} 0.1178\text{ m} & 0.0301\text{ m} \\ 0.0833 & -0.0833 \end{bmatrix}$	$\begin{bmatrix} 0.1021\text{ m} & 0.0440\text{ m} \\ 0.0833 & -0.0833 \end{bmatrix}$
$\hat{C}_{sc-noisy}$		$\begin{bmatrix} 0.0763\text{ m} & 0.0740\text{ m} \\ 0.0833 & -0.0833 \end{bmatrix}$	$\begin{bmatrix} 0.0735\text{ m} & 0.0765\text{ m} \\ 0.0833 & -0.0833 \end{bmatrix}$	$\begin{bmatrix} 0.0763\text{ m} & 0.0741\text{ m} \\ 0.0833 & -0.0833 \end{bmatrix}$

To evaluate the algorithm’s robustness in challenging conditions, including uneven terrain, electrical noise, and the presence of bumps or irregularities, we conducted tests involving the introduction of non-Gaussian noise to the Inertial Measurement Unit (IMU) signal. The ‘Heavy-tailed’ type noise was applied to the IMU using the Matlab command ‘makedist’ with parameters (‘LocationScale’, mu = 0, sigma = 1, nu = 3) and a noise amplitude of 0.1. as showed in Figure 12 Additionally, we performed tests by introducing ‘Unknown But Bounded’ noise to the IMU signal. In this assessment, the Matlab function ‘rand’ with a maximum percentage amplitude of 0.1 was utilized to introduce the specified noise type. These evaluations, conducted over a sliding time of $T_s = 6$ on the 12 paths previously examined, demonstrated the algorithm’s sustained effectiveness, even in the presence of random peaks, as outlined in the following Table 6.

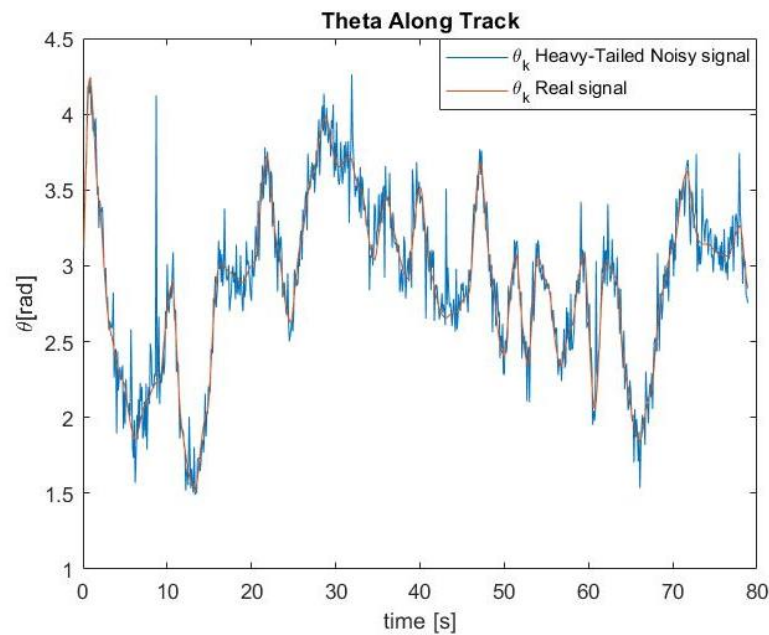


Figure 12. Example of Vehicle’s orientation θ during a trajectory in the presence of ‘heavy-tailed’ noise.

Table 6. Odometric parameter estimation \hat{c}_{ij} with Non-Gaussian noise for $T_s = 6$ s considering $P = 12$.

Scenario	“Heavy-Tailed” Noise	“UBB” Noise
$\hat{C}_{suc-noisy}$	$\begin{bmatrix} 0.1005 \text{ m} & 0.0477 \text{ m} \\ 0.0833 & -0.0833 \end{bmatrix}$	$\begin{bmatrix} 0.1038 \text{ m} & 0.0459 \text{ m} \\ 0.0833 & -0.0833 \end{bmatrix}$
$\hat{C}_{sc-noisy}$	$\begin{bmatrix} 0.0784 \text{ m} & 0.0731 \text{ m} \\ 0.0835 & -0.0835 \end{bmatrix}$	$\begin{bmatrix} 0.07331 \text{ m} & 0.0775 \text{ m} \\ 0.0832 & -0.0832 \end{bmatrix}$

Validation

Once we estimate the odometric parameters \hat{C} , it becomes possible to assess the impact of each estimation on the robot localization’s performance. Hence, a numerical validation is undertaken to evaluate the calibration accuracy. The simulation is based on a set of 12 arbitrary paths generated by applying a sequence of control signals (u, ω) to the vehicle, ensuring a condition free from slippage and noise. After averaging the previously estimated odometric parameters \hat{C} , the velocity commands are converted into specific ω_R and ω_L values for each case. Finally, exploiting the kinematic model in Equation (A1), the motion of the vehicle is characterized by the real matrix C in Equation (24).

In order to quantify the impact of the estimation on an arbitrary path, the specific path computed through the actual C in Equation (24) is employed as reference nominal path. The behavior of the robots, starting from a prescribed initial position and moving according to the wheel velocity control signals induced by the different estimation \hat{C} , is evaluated using the following metrics:

- the average distance error over paths \overline{PE} ,

$$\overline{PE} = \frac{1}{P \cdot N} \sum_{p=1}^P \sum_{i=1}^N \|[\hat{x}_{i,p} - x_{i,p}, \hat{y}_{i,p} - y_{i,p}]^\top \|, \tag{25}$$

- the average final distance error \overline{PE}_n ,

$$\overline{PE}_n = \frac{1}{P} \sum_{p=1}^P \|[\hat{x}_{N,p} - x_{N,p}, \hat{y}_{N,p} - y_{N,p}]^\top \|, \tag{26}$$

- the average orientation error over the paths \overline{OE} ,

$$\overline{OE} = \frac{1}{P \cdot N} \sum_{p=1}^P \sum_{i=1}^N | \hat{\theta}_{i,p} - \theta_{i,p} |, \tag{27}$$

- and the average final orientation error \overline{OE}_n ,

$$\overline{OE}_n = \frac{1}{P} \sum_{p=1}^P | \hat{\theta}_{N,p} - \theta_{N,p} |, \tag{28}$$

being n the total number of samples over the paths. The results obtained are presented in Table 7.

The simulation results reveal that as the estimation of matrix C deteriorates, positioning errors progressively increase with higher T_s values, ultimately causing positioning errors exceeding one meter when uncompensated slipping occurs. However, numerical experiments also demonstrate that the proposed methodology greatly reduces both positioning and orientation errors, even in the case of noisy sensors. The effect of the proposed methodology is also highlighted in Figure 13, where it is possible to compare the positioning and orientation errors, over 12 trajectories, between the $\hat{C}_{suc-noisy}$ scenario

and the $\hat{C}_{sc-noisy}$ scenario. While there is a slight increase in orientation error, a substantial reduction in positioning error is observed, reaching a decrease of an order of magnitude as presented in Table 8. This confirms the effectiveness of the proposed approach when slippage phenomena occur during the online calibration procedure.

Table 7. Position and orientation error, in [m] and [rad], respectively, in the absence and presence of slippage for a varying time interval T_s considering $P = 12$ arbitrary trajectories generated by the RRT-Dubins motion planner.

Scenario	Metric	$T_s = 10$ s	$T_s = 6$ s	$T_s = 4$ s
\hat{C}_{ns}	\overline{PE} [m]	0.0118	0.0083	0.0080
	\overline{PE}_n [m]	0.0116	0.0088	0.0101
	\overline{OE} [rad]	2.57×10^{-14}	7.47×10^{-14}	1.05×10^{-13}
	\overline{OE}_n [rad]	3.22×10^{-14}	1.09×10^{-13}	1.60×10^{-13}
\hat{C}_{suc}	\overline{PE} [m]	1.3226	0.8039	0.8432
	\overline{PE}_n [m]	1.7727	0.9941	1.1743
	\overline{OE} [rad]	2.96×10^{-14}	7.66×10^{-14}	1.08×10^{-13}
	\overline{OE}_n [rad]	3.67×10^{-14}	1.10×10^{-13}	1.68×10^{-13}
\hat{C}_{sc}	\overline{PE} [m]	0.0371	0.0597	0.0468
	\overline{PE}_n [m]	0.0675	0.1083	0.08108
	\overline{OE} [rad]	7.97×10^{-4}	0.0029	0.0016
	\overline{OE}_n [rad]	0.0013	0.0050	0.0023
$\hat{C}_{ns-noisy}$	\overline{PE} [m]	0.1247	0.0590	0.0751
	\overline{PE}_n [m]	0.2057	0.0657	0.0941
	\overline{OE} [rad]	9.26×10^{-4}	5.08×10^{-4}	9.66×10^{-4}
	\overline{OE}_n [rad]	0.0012	7.50×10^{-4}	0.0017
$\hat{C}_{suc-noisy}$	\overline{PE} [m]	1.2951	0.7967	0.8396
	\overline{PE}_n [m]	1.5737	0.9616	1.1451
	\overline{OE} [rad]	9.10×10^{-4}	5.05×10^{-4}	9.90×10^{-4}
	\overline{OE}_n [rad]	0.0012	7.45×10^{-4}	0.0018
$\hat{C}_{sc-noisy}$	\overline{PE} [m]	0.0562	0.0507	0.0777
	\overline{PE}_n [m]	0.1051	0.0977	0.1133
	\overline{OE} [rad]	0.0017	0.0025	6.67×10^{-4}
	\overline{OE}_n [rad]	0.0028	0.0046	0.0012

Table 8. Enhanced robot positioning achieved through algorithm implementation, utilizing the data from Table 7.

Scenario	Metric	$T_s = 10$ s	$T_s = 6$ s	$T_s = 4$ s
\hat{C}_{suc} vs. \hat{C}_{sc}	$\overline{PE}_{suc}/\overline{PE}_{sc}$	35.64	13.46	18.01
	$\overline{PE}_n_{suc}/\overline{PE}_n_{sc}$	26.26	9.18	14.48
$\hat{C}_{suc-noisy}$ vs. $\hat{C}_{sc-noisy}$	$\overline{PE}_{suc-noisy}/\overline{PE}_{sc-noisy}$	23.04	15.71	10.80
	$\overline{PE}_n_{suc-noisy}/\overline{PE}_n_{sc-noisy}$	14.97	9.84	10.11

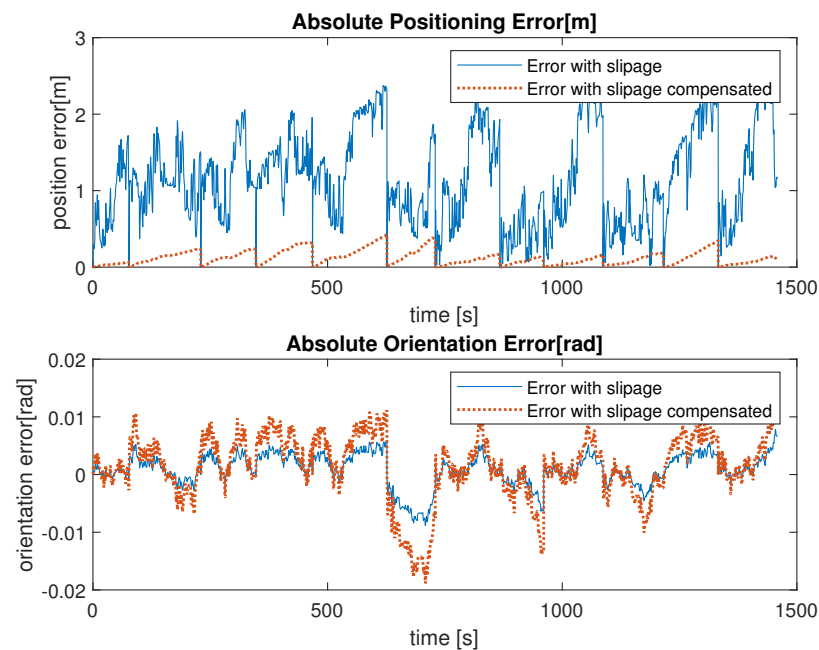


Figure 13. Example absolute position and orientation error over $P = 12$ paths.

5. Conclusions

This paper proposes an algorithm for online odometry calibration for differential-drive mobile robots subject to slippage. Different from the existing methods, the proposed approach explicitly takes into consideration the presence of slippage during the calibration process, and it mitigates its effect on the estimation of the odometric parameters. Moreover, the approach is designed with reference to an ‘arbitrary’ path, that could be executed during standard operating activities of a mobile robot and hence does not require specific or complex calibration paths. The rationale behind the proposed approach is to exploit measurements from common on-board sensors such as gyroscopes and IMUs to detect and estimate the presence of slippage, and properly address its impact on the odometry calibration. Exhaustive numerical experiments in several scenarios have been undertaken to evaluate the performance of the approach. The algorithm has been evaluated in simulation to have a clear and complete vision of its functioning, based on complete access to all the information (e.g., the slipping time, the exact value of displacements, and the actual values of the odometry matrix). However, a further investigation based on experimental testing is essential to definitely assess its usefulness in practice. Numerical results confirm the effectiveness of the proposed method and its accuracy in the estimation of the odometric parameters, even in the case of noisy sensors and in the presence of considerable slippage. Further research on this topic includes the experimental stage to determine the advantages and drawbacks of its use in real applications, as well as its integration with other fault detection and compensation algorithms currently under investigation, to achieve an overall higher resilience and endurance. Further research directions include the extension of the approach to the kinematic models of various robotic vehicles, such as tricycle, omnidirectional, and Ackerman. The possibility of using sensors other than the IMU for slippage estimation may also be investigated.

Author Contributions: Conceptualization, C.D.G., D.D.P., G.P.; methodology, C.D.G., D.D.P., G.P.; software, C.D.G., D.D.P.; validation, C.D.G., D.D.P., G.P.; formal analysis, D.D.P., G.P.; investigation, C.D.G., D.D.P., G.P.; resources, C.D.G., D.D.P., G.P.; data curation, C.D.G.; writing—original draft preparation, C.D.G., D.D.P., G.P.; writing—review and editing, G.P.; visualization, C.D.G., D.D.P.; supervision, G.P.; project administration, G.P.; funding acquisition, G.P. All authors have read and agreed to the published version of the manuscript.

Funding: The APC was funded by the National Basic Research Program. This research activity is within the Regional Program POC Puglia ‘RIPARTI’ FESR/FSE 2014–2020 grant no. 57.

Data Availability Statement: The datasets generated for this study are available on request to the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Least-Squares Online Calibration

Matrix C in Equation (4) is computed using the estimated values \hat{r}_R , \hat{r}_L , and \hat{b} of the odometric parameters. However, this leads to an identification problem that aims to minimize the error given by the mismatch between the actual and estimated parameters in C . Here we describe the original approach for this estimation problem proposed in [12].

Most algorithms for vehicle odometry focus on each parameter r_R , r_L , and b separately. However, these parameters are nonlinearly combined in the expression of C , leading to a nonlinear identification problem. To address this, the work in [12] proposes identifying the four elements c_{ji} of C instead of the three odometric parameters. While the result is numerically equivalent to estimating the individual odometry parameters for implementing calibrated odometry, this approach leads to a linear identification problem. However, by treating the four elements of C as independent, we ignore their mutual relationship, even though the unknowns are only three. Let us rewrite the equations in (5) so as to exploit linearity in the four parameters c_{ji} , and iterate it from the first up to the final N -th time sample of a generic trajectory, the result is:

$$\begin{cases} x_N - x_0 = \Delta_t c_{11} \sum_{i=0}^{N-1} \omega_{Ri} \cos(\theta_i + \Delta_t \omega_i / 2) + \Delta_t c_{12} \sum_{i=0}^{N-1} \omega_{Li} \cos(\theta_i + \Delta_t \omega_i / 2) \\ y_N - y_0 = \Delta_t c_{11} \sum_{i=0}^{N-1} \omega_{Ri} \sin(\theta_i + \Delta_t \omega_i / 2) + \Delta_t c_{12} \sum_{i=0}^{N-1} \omega_{Li} \sin(\theta_i + \Delta_t \omega_i / 2) \\ \theta_N - \theta_0 = \Delta_t c_{21} \sum_{i=0}^{N-1} \omega_{Ri} + \Delta_t c_{22} \sum_{i=0}^{N-1} \omega_{Li} \end{cases} \quad (A1)$$

The initial and final poses are assumed to be available from measurements (as for example by using exteroceptive sensors at the start and the end of the considered trajectory).

By defining the regressors:

$$\Phi_{xy} = \Delta_t \begin{bmatrix} \sum_{i=0}^{N-1} \omega_{Ri} \cos(\theta_i + \Delta_t \omega_i / 2) & \sum_{i=0}^{N-1} \omega_{Li} \cos(\theta_i + \Delta_t \omega_i / 2) \\ \sum_{i=0}^{N-1} \omega_{Ri} \sin(\theta_i + \Delta_t \omega_i / 2) & \sum_{i=0}^{N-1} \omega_{Li} \sin(\theta_i + \Delta_t \omega_i / 2) \end{bmatrix}, \quad (A2)$$

$$\Phi_{\theta} = \Delta_t \begin{bmatrix} \sum_{i=0}^{N-1} \omega_{Ri} & \sum_{i=0}^{N-1} \omega_{Li} \end{bmatrix}, \quad (A3)$$

Equation (A1) can be rewritten in compact form as:

$$\begin{bmatrix} x_N - x_0 \\ y_N - y_0 \end{bmatrix} = \Phi_{xy} \begin{bmatrix} c_{11} \\ c_{12} \end{bmatrix} \quad (A4)$$

$$\theta_N - \theta_0 = \Phi_{\theta} \begin{bmatrix} c_{21} \\ c_{22} \end{bmatrix} \quad (A5)$$

As usual in batch least squares estimation, it is important to collect sufficient data from multiple experiments. By assuming the execution of P suitable trajectories and stacking the results, one obtains:

$$\begin{bmatrix} x_{N,1} - x_{0,1} \\ y_{N,1} - y_{0,1} \\ \vdots \\ x_{N,P} - x_{0,P} \\ y_{N,P} - y_{0,P} \end{bmatrix} = \begin{bmatrix} \Phi_{xy,1} \\ \vdots \\ \Phi_{xy,P} \end{bmatrix} \begin{bmatrix} c_{11} \\ c_{12} \end{bmatrix} = \overline{\Phi}_{xy} \begin{bmatrix} c_{11} \\ c_{12} \end{bmatrix} \quad (\text{A6})$$

$$\begin{bmatrix} \theta_{N,1} - \theta_{0,1} \\ \vdots \\ \theta_{N,P} - \theta_{0,P} \end{bmatrix} = \begin{bmatrix} \Phi_{\theta,1} \\ \vdots \\ \Phi_{\theta,P} \end{bmatrix} \begin{bmatrix} c_{21} \\ c_{22} \end{bmatrix} = \overline{\Phi}_{\theta} \begin{bmatrix} c_{21} \\ c_{22} \end{bmatrix} \quad (\text{A7})$$

Through a suitable number of random trajectories, it is possible to estimate in a least squares sense the parameters c_{ij} :

$$\begin{bmatrix} \hat{c}_{11} \\ \hat{c}_{12} \end{bmatrix} = \left(\overline{\Phi}_{xy}^T \overline{\Phi}_{xy} \right)^{-1} \overline{\Phi}_{xy}^T \begin{bmatrix} x_{N,1} - x_{0,1} \\ y_{N,1} - y_{0,1} \\ \vdots \\ x_{N,P} - x_{0,P} \\ y_{N,P} - y_{0,P} \end{bmatrix} \quad (\text{A8})$$

$$\begin{bmatrix} \hat{c}_{21} \\ \hat{c}_{22} \end{bmatrix} = \left(\overline{\Phi}_{\theta}^T \overline{\Phi}_{\theta} \right)^{-1} \overline{\Phi}_{\theta}^T \begin{bmatrix} \theta_{N,1} - \theta_{0,1} \\ \vdots \\ \theta_{N,P} - \theta_{0,P} \end{bmatrix} \quad (\text{A9})$$

The C matrix estimated in this way may not satisfy the physical constraint of $c_{11}/c_{12} = -c_{21}/c_{22}$. It is worth noting that it may be feasible to incorporate this constraint into the estimation process after obtaining \hat{c}_{11} and \hat{c}_{12} [12].

References

1. Borenstein, J.; Everett, H.; Feng, L. *Where Am I? Sensors and Methods for Mobile Robot Positioning*; University of Michigan: Ann Arbor, MI, USA, 1996.
2. Borenstein, J.; Feng, L. Measurement and correction of systematic odometry errors in mobile robots. *IEEE Trans. Robot. Autom.* **1996**, *12*, 869–880. [CrossRef]
3. Sousa, R.B.; Petry, M.R.; Moreira, A.P. Evolution of Odometry Calibration Methods for Ground Mobile Robots. In Proceedings of the 2020 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC), Ponta Delgada, Portugal, 15–17 April 2020; pp. 294–299. [CrossRef]
4. Goronzy, G.; Hellbrueck, H. Weighted online calibration for odometry of mobile robots. In Proceedings of the 2017 IEEE International Conference on Communications Workshops (ICC Workshops), Paris, France, 21–25 May 2017; pp. 1036–1042. [CrossRef]
5. Borenstein, J.; Feng, L. *UMBmark: A Method for Measuring, Comparing, and Correcting Dead-Reckoning Errors in Mobile Robots*; University of Michigan: Ann Arbor, MI, USA, 1994.
6. Lee, K.; Jung, C.; Chung, W. Accurate calibration of kinematic parameters for two wheel differential mobile robots. *J. Mech. Sci. Technol.* **2011**, *25*, 1603–1611. [CrossRef]
7. Tomasi, D.L.; Todt, E. Rotational odometry calibration for differential robot platforms. In Proceedings of the 2017 Latin American Robotics Symposium (LARS) and 2017 Brazilian Symposium on Robotics (SBR), Curitiba, Brazil, 8–11 November 2017; pp. 1–6. [CrossRef]
8. Bostani, A.; Vakili, A.; Denidni, T. A novel method to measure and correct the odometry errors in mobile robots. In Proceedings of the 2008 Canadian Conference on Electrical and Computer Engineering, Niagara Falls, ON, Canada, 4–7 May 2008; pp. 897–900. [CrossRef]
9. Abbas, T.; Arif, M.; Ahmed, W. Measurement and Correction of Systematic Odometry Errors Caused by Kinematics Imperfections in Mobile Robots. In Proceedings of the 2006 SICE-ICASE International Joint Conference, Busan, Republic of Korea, 18–21 October 2006; pp. 2073–2078. [CrossRef]
10. Ivanjko, E.; Komšić, I.; Petrović, I. Simple off-line Odometry Calibration of Differential Drive Mobile Robots. In Proceedings of the 16th International Workshop on Robotics in Alpe-Adria-Danube Region—RAAD 2007, Ljubljana, Slovenia, 7–9 June 2007.

11. Jung, C.; Chung, W. Calibration of Kinematic Parameters for Two Wheel Differential Mobile Robots by Using Experimental Heading Errors. *Int. J. Adv. Robot. Syst.* **2011**, *8*, 68. [\[CrossRef\]](#)
12. 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.* **2005**, *21*, 994–1004. [\[CrossRef\]](#)
13. Mondal, S.; Yun, Y.; Chung, W.K. Terminal iterative learning control for calibrating systematic odometry errors in mobile robots. In Proceedings of the 2010 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, Montreal, QC, Canada, 6–9 July 2010; pp. 311–316. [\[CrossRef\]](#)
14. Censi, A.; Franchi, A.; Marchionni, L.; Oriolo, G. Simultaneous Calibration of Odometry and Sensor Parameters for Mobile Robots. *IEEE Trans. Robot.* **2013**, *29*, 475–492. [\[CrossRef\]](#)
15. Martinelli, A.; Tomatis, N.; Tapus, A.; Siegwart, R. Simultaneous localization and odometry calibration for mobile robot. In Proceedings of the Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453), Las Vegas, NV, USA, 27–31 October 2003; Volume 2, pp. 1499–1504. [\[CrossRef\]](#)
16. Caltabiano, D.; Muscato, G.; Russo, F. Localization and self-calibration of a robot for volcano exploration. In Proceedings of the IEEE International Conference on Robotics and Automation, New Orleans, LA, USA, 26 April–1 May 2004; Volume 1, pp. 586–591. [\[CrossRef\]](#)
17. 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* **2016**, *5*, 23. [\[CrossRef\]](#)
18. Neurauter, R.; Gerstmayr, J. A novel motion-reconstruction method for inertial sensors with constraints. *Multibody Syst. Dyn.* **2022**, *2*, 181–209. [\[CrossRef\]](#) [\[PubMed\]](#)
19. Neurauter, R.; Hergel, P.; Gerstmayr, J. Evaluation of inertial measurement units for short time motion tracking. In Proceedings of the International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Online, 17–19 August 2021; American Society of Mechanical Engineers: New York, NY, USA, 2021; Volume 85468, p. V009T09A045.
20. Siciliano, B.; Sciavicco, L.; Villani, L.; Oriolo, G. *Robotics: Modelling, Planning and Control*; Springer: London, UK, 2009. [\[CrossRef\]](#)
21. Borodacz, K.; Szczepański, C.; Popowski, S. Review and selection of commercially available IMU for a short time inertial navigation. *Aircr. Eng. Aerosp. Technol.* **2022**, *94*, 45–59. [\[CrossRef\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.