This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3111004, IEEE Access

Digital Object Identifier Xxx

Improved tracking and docking of Industrial Mobile Robots through UKF vision based kinematics calibration

STEFANO MUTTI[1], NICOLA PEDROCCHI[1]
[1]Institute of Intelligent Industrial Technologies and Systems for Advanced Manufacturing, National Research Council of Italy, 20133, Milan, Italy
Corresponding author: Stefano Mutti (e-mail: stefano.mutti@stiima.cnr.it).

ABSTRACT Performing an open-loop movement, or docking, for an industrial mobile robot (IMR), is a common necessary procedure when relying on environmental sensors is not possible. This procedure precision and outcome, solely depend on the IMR forward kinematic and odometry correctness, which is tied to the kinematics parameters, depending on the IMR kind. Calibrating the kinematic parameters of an IMR is a time consuming and mandatory procedure, since the mechanical tolerances and the assembly procedure may introduce a large variation from the nominal parameters. Furthermore, calibration inaccuracies might introduce severe inconsistencies in tasks such as localization, mapping, and navigation in general. In this work, we focus on the so-called kinematic parameter calibration. We propose the use of the unscented Kalman filter to perform a calibration procedure of the geometrical kinematic parameters of a mobile platform. The mobile platform is externally tracked during the calibration phase, using a fixed temporary external sensor that retrieves the position of a visual tag fixed to the platform. The unscented Kalman filter, using the calibration phase collected data, estimates the enlarged system state, which is comprised of the parameters that have to be estimated, the platform odometry and the visual tag position.
The method can either be used online, to identify parameters and monitor their value while the system is operating, or offline, on logged data. We validate this method on two different devices, a 4 mecanum-wheel IMR , and a Turtlebot 3, using a camera to track the movement trough a reference chessboard, for then comparing the original path to its corrected version.

INDEX TERMS Mobile robot calibration, Unscented Kalman filter

I. INTRODUCTION

Generally, industrial mechanical systems need to have a good parameter calibration to perform accordingly to the standard, hence, a calibration procedure is needed. This calibration procedure, or parameter estimation, is obtainable by means of general purpose algorithms, or specifically tailored methods for particular mechanical systems. Concerning IMRs, the value of the kinematic parameters incorporated in the model, hugely modify the performance of the system in all of its uses, from the mapping phase, where the IMR position infers the map conception, in navigation, where the planning algorithms use the IMR kinematics to devise a path and control the robot movements, and in localization, where the relative position and velocity are used to estimate the displacement progression in an environment. The overall impact of the IMR odometry correctness arises in specific cases such as docking, in which we cannot rely on localization algorithms (e.g. highly de-structured environments, dynamic environments or lacking of localization sensors), and high precision relative displacements are performed in open loop on the IMR kinematic model. Generally, the constant parameters that are taken into account in an IMR kinematic are the wheels radius, which is usually the same for all the wheels, and a geometric length that depends of the IMR structure (e.g omnidirectional mobile platform, car-like, three wheels). Several works related to the general subject of odometry calibration are present in the literature, some focusing on the intrinsic kinematic parameter calibration, some on the calibration of extrinsic parameters related to mounted sensors, or both problems simultaneously. The majority of the works present are designed for differential drive IMRs, while few are specific to omnidirectional mobile platforms, furthermore, most of the methods are offline and based on the repetition of fixed calibration paths. Offline calibration is more commonly employed in standard robotics, but is not suitable to systems where the kinematic parameters change

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3111004, IEEE Access

significantly over time (inflatable wheels, robots with great dynamic loads, re-configurable mobile platforms), while on-line calibration requires real time capability in the measure system and in the computation, which might be hard to accomplish for complex vision related measure systems.

Works dealing with the topic of parameter estimation and calibration are vastly present in literature, for generic robotic chains, many classical methods have been devised, from a classical Denavit – Hartenberg approach [1], to rejecting-sampling methods [2], virtually closed kinematic chains [3], and Kalman filtering based methods, such as in [4], [5], where Kalman theory is used alongside a particle filter to estimate serial manipulator parameters. Recent works [6]–[8] apply visual based measurement system for the estimation of manipulator parameters, [9] deals with a complete $AXB = YCZ$ problem, while deep learning based methods have also been used extensively on the subject [10], [11]. For an extensive survey on the overall topic, refer to [12]. Despite IMRs can be treated as a generic kinematic chains, many works focus on the different kinds existing. On the subject, systematic odometry errors have been initially studied by [13]–[18], while more recent works on the subject use iterative learning approaches [19], least squares estimation [20], and 'Effective Kinematic Parameters' [21]. All the mentioned methods, follow a trajectory based optimization, where the aim is to minimize the positional error on a given goal path that is repeated constantly, and where the information needed are related only to the end points. In such works, the calibration performance is affected by the path parameters, hence the number of trials and the length of the paths [22], while the paths have to be manually designed and the measurements of the final error carried out manually, with the possibility to bring inaccuracies and measurement errors. Many works take in consideration a possible kinematic flaw, such has an unequal wheel diameter [23]–[25], generic optimization methods tackle the model inaccuracies by means of corrective coefficients [26], [27], while dynamic wheel model and lateral dynamics and taken into account in [28]. Specific works have also been carried out regarding the type of IMR: car-like robot [29]–[31], differential drive [22], [32] and omnidirectional robots [25], [27], [33]–[35]. Many methods based on Kalman filtering algorithms have been used, where the localization and calibration are solved as a single combined task , using the available sensors on the IMR [36]–[38]. The advantage in using Kalman filtering techniques is that the calibration can be depleted online, while the mobile platform performs its tasks. For an extensive survey on IMRs odometry calibration methods, refer to [39]. The contribution of this paper is to propose the usage of an external sensor system to track the IMR state, using a visual tag fixed on it, while reading the values of the wheels encoders, in order to perform a model parameter identification using the collected data, by means of an unscented Kalman filter. The usage of an external visual sensor as a tracking method has many advantages on the calibration system, nowadays industrial cameras reach a high level of fidelity, given by the combination of high resolution, advanced calibration techniques, and high frame rate, which allows a smooth tracking of the IMR movement. Furthermore, this method doesn't require the design of a specific path, nor requires hardware changes to the IMR, making it easily integrable in every scenario. In order to identify the parameters, a system model that includes the mobile platform position, velocity, and the relative tag position is needed, where the unknown relative position between the reference tag and the mobile platform reference system corresponds to an hand-eye problem, which parameters will be included in the system model and estimated.

II. BACKGROUND

A. MOBILE PLATFORM KINEMATIC AND ODOMETRY

Consider a generic IMR as shown in Fig 1, where W is the globally fixed world frame, MR is the frame fixed to the IMR, and

$$T^W_{MR} = \begin{pmatrix} R^W_{MR} & \mathbf{t}^W_{MR} \\ 0 & 1 \end{pmatrix} \tag{1}$$

is the transformation matrix, composed by the rotational R^W_{MR} and translational $\mathbf{t}^W_{MR} = [p_x \ p_y \ p_z]^T$ parts, from the MR frame to the W. The robot position in the world is described by its translational and rotational components, hence the six element vector $[p_x, p_y, p_z, \phi_1, \phi_2, \phi_3] = [\mathbf{p}_W, \boldsymbol{\phi}_W]$, prompt by the requirement of describing also rough and uneven terrains, uneven wheels diameters, and other environment irregularities.

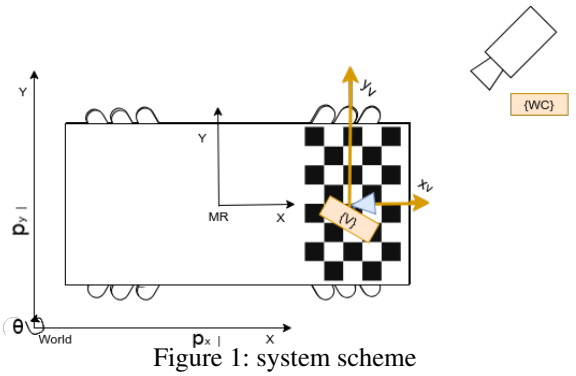

Figure 1: system scheme

The IMR model generally includes the j-th wheel radius r_j,with $\mathbf{r} = [r_0, ..., r_j]$, a geometric distance depending on the robot kind l, a general purpose parameter vector $\mathbf{q}$, and the wheels rotational speed $\dot{\boldsymbol{\theta}}$, composed by a j number of elements $\dot{\boldsymbol{\theta}} = [\dot{\theta}_0, \ldots, \dot{\theta}_j]$, computed by the wheels encoder sensors.

Depending on the specifics, the wheels radius might be assumed equal, or different for each wheel. Due to the digital nature of IMR controllers and encoders, the system will be described with a discrete time model. The IMR velocities $\mathbf{v}$ and $\boldsymbol{\omega}$ at time t are expressed with respect to the MR frame as the vector $\mathbf{v}^t_{MR} = [v^t_x, v^t_y, v^t_z]^T$ and $\boldsymbol{\omega}^t_{MR} = [\omega^t_1, \omega^t_2, \omega^t_3]^T$, as computed by the forward kinematic of the mobile robot :

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3111004, IEEE Access

$$\begin{bmatrix}\mathbf{v}_{MR}^t\\ \omega_{MR}^t\end{bmatrix} = \mathbf{f}\left(\dot{\boldsymbol{\theta}}^t, \mathbf{r}, l, \mathbf{q}\right) \tag{2}$$

where the function $\mathbf{f}$ depends on the mobile platform type(e.g. for car-like robots $v_y^t = 0$), while the same velocities referred to the W are:

$$\begin{bmatrix}\mathbf{v}_{W}^t\\ \boldsymbol{\omega}_{W}^t\end{bmatrix} = \begin{pmatrix} R_{MR}^W & [\mathbf{t}]_X R_{MR}^W \\ 0 & R_{MR}^W \end{pmatrix} \begin{bmatrix}\mathbf{v}_{MR}^t\\ \boldsymbol{\omega}_{MR}^t\end{bmatrix} \tag{3}$$

where $[\cdot]_X$ is the skew operator.

Referred to the frame W at time $t+1$, the position and the orientation (odometry) of the IMR is then computed as the discrete-time integral given dt as the update time, using position, orientation, linear velocity and angular velocity at time t (passing from angular velocity to Cardan angles is done accordingly to [40]).

B. UNSCENTED KALMAN FILTER

Consider a discrete time nonlinear dynamical system:

$$\zeta^{t+1} = F(\zeta^t, u^t, v^t, e) \tag{4}$$
$$y^t = G(\zeta^t, n^t, b) \tag{5}$$

where t is the time index, ζ is the state vector, u is the system input, y is the system output, and e and b are generic constant parameters. The model includes v and n which are respectively the state noise, in order to include model uncertainties, and the measurement noise, both are zero-mean white noises with known covariance matrix. Both the system dynamic model F and the output function G, are known, and include some unknown constant parameters that have to be estimated. In order to use the Kalman filtering theory to estimate the constant parameters, they have to be given a fake dynamic, and to be included in the state vector:

$$x^{t+1} = \begin{bmatrix}\zeta^{t+1}\\ e^{t+1}\\ b^{t+1}\end{bmatrix} = \begin{bmatrix}F(\zeta^t, u^t, v^t)\\ e^t\\ b^t\end{bmatrix} \tag{6}$$

where x is the enlarged state, and the parameters e and b, that have to be estimated, do not have an evolution because their value at instant $t+1$ is the same as at instant t.

Kalman filtering theory can be optimally used in linear dynamical systems to estimate the state value, while, in case of a nonlinear system, variants of the original filter exist, such as the extended Kalman filter (EKF) or the Unscented Kalman filter (UKF). Considering the strong non-linearity of the system, and the additional complexity brought by the hand-eye problem, an UKF is employed, due to its increased reliability in non-linear system identification [41], and robustness to parameter initial value [42]. While the inaccuracies of the EKF, come from the linearization of the dynamic model, the UKF keeps the original nonlinear dynamic model, using the unscented transform methodology [43] to propagate the state uncertainty[1]. Following [41], denote $x \in \mathbb{R}^L$, $\bar{x}$, P_x as an L dimensional random variable, its mean value, and its co-variance matrix respectively, and denote $y = g(x)$ as a generic nonlinear map. The distribution of y, its mean value $\bar{y}$ and its co-variance P_y can be approximated using a number $2L+1$ of sigma row vectors χ_i, which form a matrix χ, and associated weights W_i as following

$$y_i \equiv g(\chi_i) \qquad i = 0, ..., 2L \tag{7}$$
$$\bar{y} \approx \sum_{i=0}^{2L} W_i^m y_i \tag{8}$$
$$P_y \approx \sum_{i=0}^{2L} W_i^c (y_i - \bar{y})(y_i - \bar{y})^T. \tag{9}$$

The set of vectors χ_i and weights W_i are computed as

$$\begin{aligned}
\chi_0 &= \bar{x},\\
\chi_i &= \begin{cases} \bar{x} + (\sqrt{(L+\lambda)P_x})_i & \text{if} \quad i = 1, ..., L \\ \bar{x} - (\sqrt{(L+\lambda)P_x})_{i-L} & \text{if} \quad i = L+1, ..., 2L \end{cases}\\
W_0^m &= \lambda/(L+\lambda)\\
W_0^c &= \lambda/(L+\lambda) + (1 - \mu^2 + \nu)\\
W_i^m &= W_i^c = 1/(2(L+\lambda)).
\end{aligned} \tag{10}$$

where $(.)_i$ extracts the i-th row of the resulting matrix, and $\lambda = \mu^2(L+\kappa) - L$ is the gamma vectors scaling parameter, with μ, ν and κ tuned accordingly to the process.

The UKF, extends the unscented transform, including it into the Kalman filter iterative estimation, the UKF steps are reported in algorithm 1, for an extensive survey, refer to [41].

III. METHODOLOGY

The system is comprised by an IMR endowed with wheel encoders, a visual tag fixed on the platform, and an external tracking camera to estimate the position of the visual tag, as shown in Fig 1. The system model includes the IMR kinematic and odometry, the tag frame position, the fixed transformation between the IMR frame and the tag frame, and the geometrical parameters related to the kinematic formulation.

Four reference systems are defined prior to the model, the frame W, a fixed global frame to which the mobile platforms position are referred, equal to MR, the mobile platform frame, at the starting time, and VW, the visual world fixed reference system, equal to V, the tag frame, at starting time, refer to Fig 2 for a system representation.

The tag position results in

$$T_{V|t}^{WV} = H_{MR}^V \, T_{MR|t}^W \, H_V^{MR} \tag{11}$$
$$\tag{12}$$

where H_{MR}^V is the constant transformation matrix between the MR frame and the tag frame V, and $T_{MR|t}^W$ is the odometry transformation.

In order to properly include the H_{MR}^V and T_V^{WV} matrix in the model, its formulation from the Euler xyz angles is used :

[1]The unscented transformation is a method to compute the evolution of a random variable through a nonlinear function.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3111004, IEEE Access

Algorithm 1: UKF

$x^a \leftarrow [x^T\ v^T\ n^T]$
$\chi^a \leftarrow [(\chi^x)^T(\chi^v)^T(\chi^n)^T]^T$
x^a is the concatenation of original state and noise variables
P_v is the process noise co-variance matrix
P_n i the measurement noise co-variance matrix
x_k^- is the optimal prediction of such variable.

procedure UKF BASED CALIBRATION
  $\hat{x_0} \leftarrow E[x_0]$
  $P_0 \leftarrow E[(x_0 - \hat{x}_0)(x_0 - \hat{x}_0)^T]$
  $\hat{x}_0^a \leftarrow E[x^a] \leftarrow [\hat{x}_0^T\ 0\ 0]^T$
  $P_0^a \leftarrow E[(x_0^a - \hat{x}_0^a), (x_0^a - \hat{x}_0^a)^T] = \begin{bmatrix} P_0 & 0 & 0 \\ 0 & P_v & 0 \\ 0 & 0 & P_n \end{bmatrix}$
  for *every step k* **do**
  compute sigma points:
  $\chi_{k-1}^a \leftarrow [\hat{x}_{k-1}^a \quad \hat{x}_{k-1}^a \pm \sqrt{(L+\lambda)P_{k-1}^a}]$
  time update:
  $\chi_{k|k-1}^x \leftarrow F[\chi_{k-1}^x, \chi_{k-1}^v]$
  $\hat{x}_k^- \leftarrow \sum_{i=0}^{2L} W_i^m \chi_{i,k|k-1}^x$
  $P_k^- \leftarrow \sum_{i=0}^{2L} W_i^c [\chi_{i,k|k-1}^x - \hat{x}_k^-][\chi_{i,k|k-1}^x - \hat{x}_k^-]^T$
  $y_{k|k-1} \leftarrow H[\chi_{k|k-1}^x, \chi_{k-1}^n]$
  $\hat{y}_k^- \leftarrow \sum_{i=0}^{2L} W_i^m y_{i,k|k-1}$
  Measurement update:
  $P_{\tilde{y}_k,\tilde{y}_k} \leftarrow \sum_{i=0}^{2L} W_i^c [y_{i,k|k-1} - \hat{y}_k^-][y_{i,k|k-1} - \hat{y}_k^-]^T$
  $P_{x_k,y_k} \leftarrow \sum_{i=0}^{2L} W_i^c][\chi_{i,k|k-1} - x_k^-[y_{i,k|k-1} - \hat{y}_k^-]^T$
  $\kappa \leftarrow P_{x_k,y_k} P_{\tilde{y}_k\tilde{y}_k}^{-1}$
  $\hat{x}_k \leftarrow \hat{x}_k^- + \kappa(y_k - \hat{y}_k^-)$
  $P_x \leftarrow P_k^- - \kappa P_{\tilde{y}_k\tilde{y}_k} \kappa^T$
end procedure

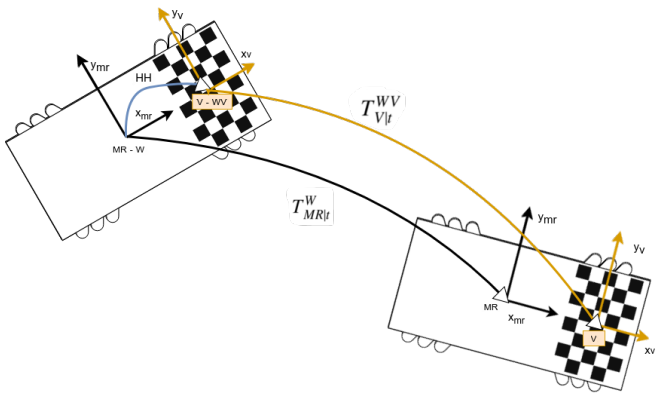

Figure 2: Coordinates systems in two different instant times.

$$H_{MR}^{V} = \left(\begin{array}{ccc|c} & R_z(\gamma)R_y(\beta)R_x(\alpha) & & \begin{matrix} h_x \\ h_y \\ h_z \end{matrix} \\ \hline 0 & 0 & 0 & 1 \end{array}\right) \tag{13}$$

$$T_{V}^{WV} = \left(\begin{array}{ccc|c} & R_z(\delta)R_y(\epsilon)R_x(\eta) & & \begin{matrix} c_x \\ c_y \\ c_z \end{matrix} \\ \hline 0 & 0 & 0 & 1 \end{array}\right) \tag{14}$$

Hence, defining $\widetilde{\mathbf{h}}$ and $\widetilde{\mathbf{c}}$ as

$$\widetilde{\mathbf{h}}^t = [h_x^t, h_y^t, h_z^t, \alpha^t, \beta^t, \gamma^t]^T \tag{15}$$

$$\widetilde{\mathbf{c}}^t = [c_x^t, c_y^t, c_z^t, \eta^t, \epsilon^t, \delta^t]^T \tag{16}$$

the state of the system becomes :

$$\left[\begin{array}{cccccccc} \mathbf{v}_W & \mathbf{p}_W & \widetilde{\mathbf{c}} & \widetilde{\mathbf{h}} & \mathbf{r} & l & \mathbf{q} \end{array}\right]^T \tag{17}$$

.

where the last elements are the constant parameters that have to be estimated:

$$\begin{bmatrix} \vdots \\ \widetilde{\mathbf{h}}^{\mathbf{t+1}} \\ \mathbf{r}^{t+1} \\ l^{t+1} \\ \mathbf{q}^{t+1} \end{bmatrix} = \begin{bmatrix} \vdots \\ \widetilde{\mathbf{h}}^{\mathbf{t}} \\ \mathbf{r}^{t} \\ l^{t} \\ \mathbf{q}^{t} \end{bmatrix} \tag{18}$$

The measurable output y^t of the system is the visual tag position, computed by an external tracking system as shown in Fig. 3. In order to retrieve the position of the mobile tag frame by means of the external sensor, the vision tag displacement at time t is always referred to the position at instant 0, using the following: $T_{V|t}^{WV} = T_{WC}^{WV|t=0} T_{V|t}^{WC}$. The output y^t is hence expressed as

$$y^t = \widetilde{\mathbf{c}}^t \tag{19}$$

The parameters are then estimated online, or offline, using the output readings and encoders value to evolve the model dynamic.

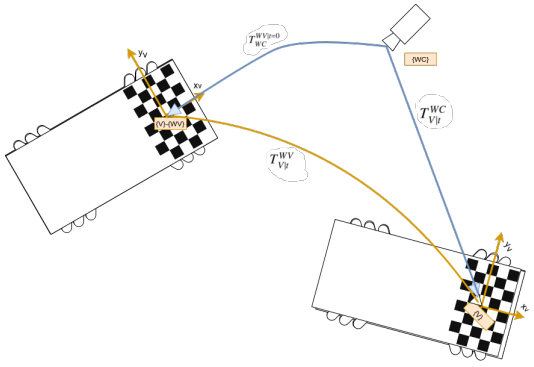

Figure 3: Reference tag measure

IV. TEST CASES

A. DESIGN OF EXPERIMENTS

Tests have been performed in a 2D planar environment, with two different IMRs, one omnidirectional and one with differential drive, in order to asses its effectiveness in two different scenarios. As a reference, the omniwheel platform is shown in Fig. 4, while the differential drive Turtlebot 3 waffle is shown in Fig. 5. The parameters to be estimated

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3111004, IEEE Access

are the radius of the platform wheels, considered to be equal for wheels of the same platform, and a geometric parameter, which is l_{xy} for the omniwheel platform, and the parameter l_d for the differential drive. The parameters are initialized with the CAD values for the omniwheel internally developed platform, and with data-sheet values for the differential drive commercial platform. The platforms have been teleoperated, using a joystick as a remote manual speed controller, along 10 different paths, each with an average duration of 50 seconds, as a reference, the nominal paths performed by the omnidirectional robot are shown in Fig. 6, we refer to the nominal path as the one measured by the external camera.

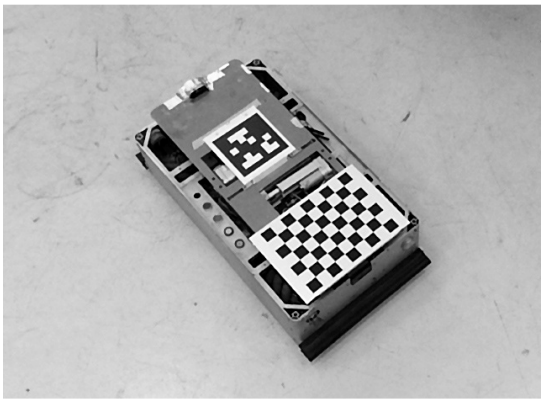

Figure 4: Omniwheel platform during the acquisition

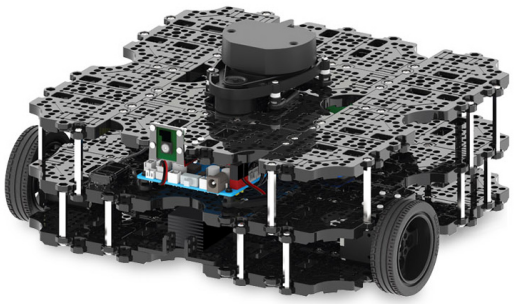

Figure 5: Turtlebot3 waffle

In order to ease the estimation, we suppose the tag frame to be planar to the platform xy frame plane (i.e. $\alpha, \beta = 0$), and for both the frames to have the same z component in the world frame (i.e. h_z = 0). All the needed data are logged during the runs, and then used offline to evolve the dynamic of the discrete-time system, at every step(time instant), and improve the estimation of the state, repeating the overall estimation procedure for a number of iterations. The data is shuffled between each estimation trial, where half of the data are used to simulate the model and estimate the parameters, while the other half to validate the trial and assess improvement. Parameters relative to the UKF are tuned according to [43], [44], specifically the values relative to the Unscented transform are: number of points(14), α(0.1), β(2.0) and κ(-11). Regarding the initial covariances of the UKF, the numerous trials we performed offline suggested that the initial values mostly affect the convergence time, and not the actual estimation value. Specifically, the value of the state covariance matrix P is set to $0.01 * I$, where I is the identity matrix, the state noise covariance matrix Q is set to $0.01 * I$, aside from the values corresponding to parameters that have to be estimated, which is set to zero, while the output noise covariance matrix R is $0.0025 * I$, which has been approximately computed using the camera resolution, the camera field of view and the average distance from camera to the tracked object. The identification algorithm is run in two different forms:

- parallelly, performing an estimation for each of 5 estimation data sets, using 10 iterations of the whole data per run, for then taking the average of the values. In order to assess the convergence to common values of the estimations on the data sets.
- on the 5 data sets in series, for 10 iterations, for having a single estimate of the parameters

furthermore, the outcomes are compared, to establish the more appropriate approach to the problem.

Each estimation is repeated 50 times, randomly shuffling the estimation and validation paths in order to have statistically relevant results.

The IMRs controllers are synchronized with the camera acquisition system using the NTP protocol. Both robots are moved at a maximum velocity of $0.15\frac{m}{s}$, while externally tracked at 30Hz, using a calibrated Kinect V2 camera, hence allowing a maximum displacement of $0.005m$ per frame. Specifically, the Kinect V2 camera has a 1920 x 1080 RGB resolution, and , most importantly, adopt a global shutter, which reduces blurs and improves the measure quality. The wheels encoders are then down-sampled at 30 Hz in order to have the same time scale. The relative position between the tag and the camera is computed using the perspective-n-point algorithm in the OpenCv libraries [45]. For the algorithm development, Python3 has been used, while the UKF library is provided by [46]. The estimation algorithm has been run using an Intel Core i7-7700HQ processor. Given this setup, the running time of a single estimation instance on the offline data, for both methods, was of 35 seconds. In order to measure the performance improvements, 2 indexes will be used, one that takes in account the whole movement along the paths, and one which aim is to measure the docking precision, that relies only on the final position after a path is performed. We define a path $\mathcal{P}_. = \{\mathbf{p_0}, ..., \mathbf{p_{N_p}}\}$ as an ordered collection of IMR positions $\mathbf{p}$, where each position is composed by two translation(x, y) and one rotation(ω), being our experiment setup modelled in a 2D environment. Then, we will define a nominal path $\mathcal{P}_n$, as the one measured by the external camera, an original path $\mathcal{P}_o$, as the one computed using the initial parameters values by the IMR kinematics, and a corrected $\mathcal{P}_c$ path, as the one re-computed with the newly estimated parameters. In order to measure the improvement given by the estimation of the parameters, we compare the different paths, using the summation of the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3111004, IEEE Access

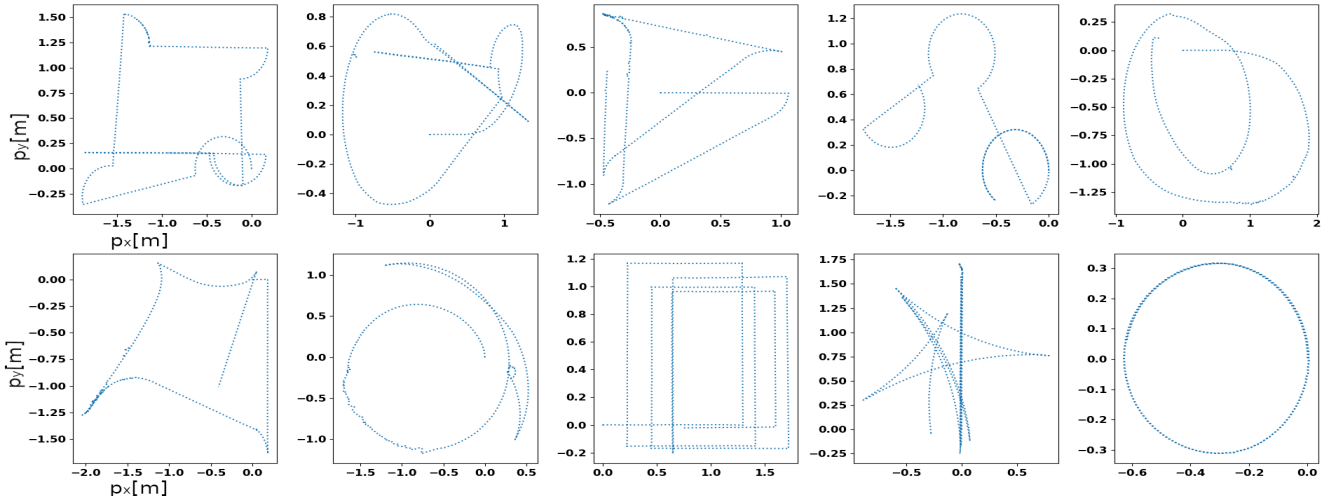

Figure 6: Nominal paths used for the omnidirectional platform

euclidean distance between each $j-th$ point $\mathcal{P}(j) = p_j$ in $\mathcal{P}_o$ and $\mathcal{P}_c$, with respect to the $j-th$ point in $\mathcal{P}_n$ as:

$$\mathcal{E}_o = \sum_{j}^{N_p} \|\mathcal{P}_n(j) - \mathcal{P}_o(j)\| \qquad \mathcal{E}_c = \sum_{j}^{N_p} \|\mathcal{P}_n(j) - \mathcal{P}_c(j)\| \tag{20}$$

where N_p is the equal number of points of the paths, for then computing the percentage improvement of the corrected path as

$$\sigma_c = \frac{\mathcal{E}_o - \mathcal{E}_c}{\mathcal{E}_o} \tag{21}$$

B. TEST CASE A - OMNIWHEEL MOBILE PLATFORM

The test is performed on an omniwheel mobile platform, endowed with 4 mecanum wheels with an axis of rotation at $45°$ to the wheel plane and at $45°$ to the axle line, each wheel has a high frequency (1kHz) encoder to compute the speed and position. For this specific robot, equations 2 and 3 take this form:

$$\begin{bmatrix} v_x^t \\ v_y^t \\ v_z^t \\ \omega_1^t \\ \omega_2^t \\ \omega_3^t \end{bmatrix} = \frac{r}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & -1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -\frac{1}{l_{xy}} & \frac{1}{l_{xy}} & -\frac{1}{l_{xy}} & \frac{1}{l_{xy}} \end{bmatrix} \begin{bmatrix} \dot{\theta}_0^t \\ \dot{\theta}_1^t \\ \dot{\theta}_2^t \\ \dot{\theta}_3^t \end{bmatrix}$$

where l_{xy} is half-distance between front wheels and rear wheels plus half-distance between left wheels and the right wheels, refer to [47] for a complete survey on mobile robot kinematics.

Regarding the parallel estimation approach, Fig. 7 shows the convergence of the radius estimation, while Fig. 8 shows the relation between the nominal, original and corrected paths for the test dataset, in the same way, for the series approach, Fig. 9 shows the convergence of the radius and l_{xy} estimations, while Fig. 10 shows the paths comparison.

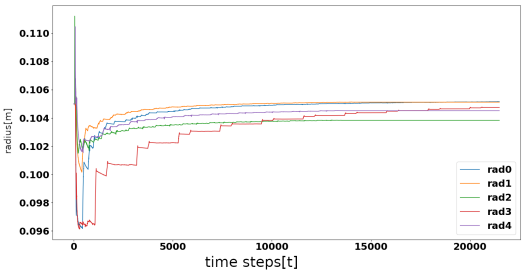

Figure 7: radius trend, for a single instance of the parallel estimation method. Standard deviation of estimated radius: 0.00033m

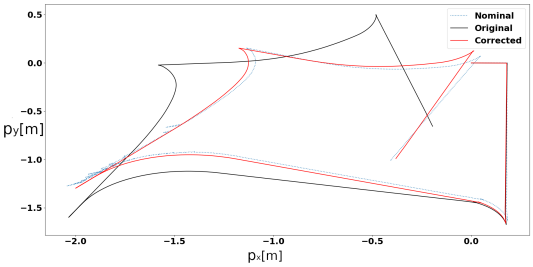

Figure 8: Nominal, original and corrected paths, for a single instance of the parallel estimation case.Value of $\sigma_c = 0.57$, as computed in equation 21

C. TEST CASE B - DIFFERENTIAL DRIVE MOBILE PLATFORM

The test is performed on a Turtlebot3 Waffle commercial mobile platform, endowed with 2 active wheels (100Hz encoders) and 2 ball casters.

For this specific robot, equation 2 takes this form:

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3111004, IEEE Access

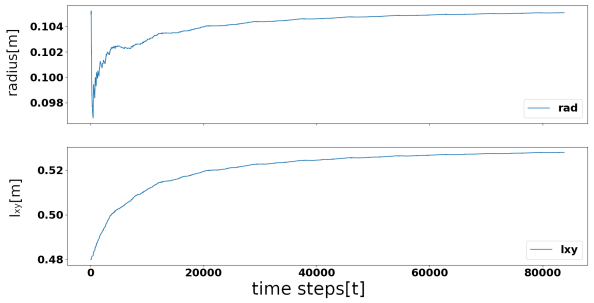

Figure 9: radius and l_{xy} trends , for a single instance of series estimation method

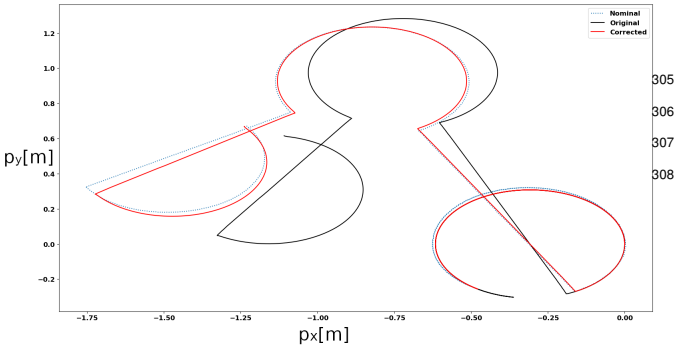

Figure 10: Nominal, original and corrected path, for a single instance of the series estimation case.Value of $\sigma_c = 0.48$, as computed in equation 21

$$\begin{bmatrix} v_x^t \\ v_y^t \\ v_z^t \\ \omega_1^t \\ \omega_2^t \\ \omega_3^t \end{bmatrix} = r \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ -\frac{1}{l_d} & \frac{1}{l_d} \end{bmatrix} \begin{bmatrix} \dot{\theta}_0^t \\ \dot{\theta}_1^t \end{bmatrix} \tag{22}$$

where l_d is the distance between left wheel and right wheel, refer to [47] for a complete survey on mobile robot kinematics.

An example of the radius value trend during the parallel estimation in this case is shown in Fig. 11.

D. RESULTS

The results are presented in Table 2 for the omniwheel robot, and in Table 3 for the Turtlebot3 waffle, where Std refers to the standard deviation of the estimated parameter, and $\overline{\sigma_c}$,$\overline{r}$ and $\overline{l_.}$ are the mean values, over the 50 trials. While the overall odometry error decreased in both cases, it is noticeable that the series estimation brought better results in both the experiments. The estimation in the series case, is more precise, due to higher $\overline{\sigma_c}$, and more reliable, due to a smaller $Std\ r$ and $Std\ l_.$. In order to assess the docking capability improvement, we also compare the final position distance between the nominal path, the original computed odometry and the corrected one, using the same concept as in 21, the results are shown in 1. The overall resulted accuracy is higher in the omniwheel robot, due to a better manufacture and devices quality .

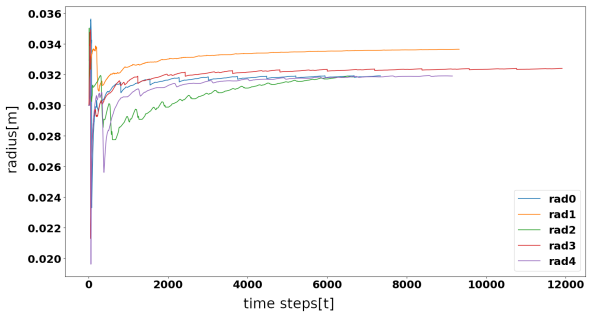

Figure 11: radius trend, for a single instance of the parallel method.Standard deviation of estimated radius: 0.00059m

Table 1: Docking performance improvement for both mobile robots

$\overline{\sigma_c}$(eq 21)	Parallel	Series
Omniwheel	0.76	0.82
Turtlebot3	0.37	0.51

Table 2: Omniwheel robot estimation results, on 20 trials

	Parallel	Series
$\overline{r}[m]$	0.101	0.104
$Std\ r[m]$	0.0031	0.00035
$\overline{l_{xy}}[m]$	0.495	0.525
$Std\ l_{xy}[m]$	0.0086	0.0017
$\overline{\sigma_c}$(ref eq 21)	0.26	0.428

V. CONCLUSIONS AND FURTHER DEVELOPMENT

In this work, a novel method to improve the IMR docking quality and calibrate the kinematic parameters of a mobile platform is presented. An Unscented Kalman Filter is used to estimate the parameters of the dynamic model of the mobile platform, tracked using an external camera, and wheels encoders. The method is tested on two different IMRs, gathered results show a positive outcome. In order to improve the method, future works will focus on filtering the data, to raise the value of information-rich data, and filter out the data that is deemed not important. This approach can be pursued using the Fisher information theory. Furthermore, the method will be tested in a 3D scenario, including problematics related to 3D outdoor navigation and mobile platforms with suspensions. Other improvements can be done to lessen the computation time, in order to make the algorithm more suitable for online use.

Table 3: Turtlebot3 estimation results, on 20 trials

	Parallel	Series
$\overline{r}[m]$	0.027	0.031
$Std\ r[m]$	0.0064	0.004
$\overline{l_d}[m]$	0.332	0.340
$Std\ l_d[m]$	0.021	0.012
$\overline{\sigma_c}$(ref eq 21)	0.122	0.27

References

[1] J.-W. Lee, G.-T. Park, J.-S. Shin, and J.-W. Woo, "Industrial robot calibration method using denavit—hatenberg parameters," in 2017 17th International Conference on Control, Automation and Systems (ICCAS). IEEE, 2017, pp. 1834–1837.

[2] Y. Pichugin, G. Alferov, and O. Malafeyev, "Parameters estimation in mechanism design," Contemporary Engineering Sciences, vol. 9, pp. 175–185, 2016.

[3] C. S. Gatla, R. Lumia, J. Wood, and G. Starr, "An automated method to calibrate industrial robots using a virtual closed kinematic chain," IEEE Transactions on Robotics, vol. 23, no. 6, pp. 1105–1116, 2007.

[4] G. Du and P. Zhang, "Online serial manipulator calibration based on multisensory process via extended kalman and particle filters," IEEE Transactions on Industrial Electronics, vol. 61, no. 12, pp. 6852–6859, 2014.

[5] Z. Jiang, W. Zhou, H. Li, Y. Mo, W. Ni, and Q. Huang, "A new kind of accurate calibration method for robotic kinematic parameters based on the extended kalman and particle filter algorithm," IEEE Transactions on Industrial Electronics, vol. 65, no. 4, pp. 3337–3345, 2017.

[6] J. M. S. Motta, G. C. De Carvalho, and R. McMaster, "Robot calibration using a 3d vision-based measurement system with a single camera," Robotics and Computer-Integrated Manufacturing, vol. 17, no. 6, pp. 487–497, 2001.

[7] Y. Meng and H. Zhuang, "Autonomous robot calibration using vision technology," Robotics and Computer-Integrated Manufacturing, vol. 23, no. 4, pp. 436–446, 2007.

[8] X. Zhang, Y. Song, Y. Yang, and H. Pan, "Stereo vision based autonomous robot calibration," Robotics and Autonomous Systems, vol. 93, pp. 43–51, 2017.

[9] L. Wu, J. Wang, L. Qi, K. Wu, H. Ren, and M. Q.-H. Meng, "Simultaneous hand–eye, tool–flange, and robot–robot calibration for comanipulation by solving the axb= ycz problem," IEEE TRansactions on robotics, vol. 32, no. 2, pp. 413–428, 2016.

[10] H. Su, C. Yang, H. Mdeihly, A. Rizzo, G. Ferrigno, and E. De Momi, "Neural network enhanced robot tool identification and calibration for bilateral teleoperation," IEEE Access, vol. 7, pp. 122 041–122 051, 2019.

[11] H.-N. Nguyen, J. Zhou, and H.-J. Kang, "A calibration method for enhancing robot accuracy through integration of an extended kalman filter algorithm and an artificial neural network," Neurocomputing, vol. 151, pp. 996–1005, 2015.

[12] A. Elatta, L. P. Gen, F. L. Zhi, Y. Daoyuan, and L. Fei, "An overview of robot calibration," Information Technology Journal, vol. 3, no. 1, pp. 74–78, 2004.

[13] M. Muniandy and K. Muthusamy, "An innovative design to improve systematic odometry error in non-holonomic wheeled mobile robots," Procedia Engineering, vol. 41, pp. 436–442, 2012.

[14] T. Abbas, M. Arif, and W. Ahmed, "Measurement and correction of systematic odometry errors caused by kinematics imperfections in mobile robots," in 2006 SICE-ICASE International Joint Conference. IEEE, 2006, pp. 2073–2078.

[15] J. Borenstein and L. Feng, "Correction of systematic odometry errors in mobile robots," vol. 3. IEEE, 1995, pp. 569–574.

[16] M. Bak, T. D. Larsen, N. A. Andersen, and O. Ravn, "Autocalibration of systematic odometry errors in mobile robots," in Mobile Robots XIV, vol. 3838. International Society for Optics and Photonics, 1999, pp. 252–263.

[17] J. Borenstein and L. Feng, "Measurement and correction of systematic odometry errors in mobile robots," IEEE Transactions on robotics and automation, vol. 12, no. 6, pp. 869–880, 1996.

[18] Y. Maddahi, "Off-line calibration of autonomous wheeled mobile robots," in Handbook of Research on Biomimetics and Biomedical Robotics. IGI Global, 2018, pp. 375–389.

[19] S. Mondal, Y. Yun, and W. K. Chung, "Terminal iterative learning control for calibrating systematic odometry errors in mobile robots," in 2010 IEEE/ASME International Conference on Advanced Intelligent Mechatronics. IEEE, 2010, pp. 311–316.

[20] F. Galasso, D. L. Rizzini, F. Oleari, and S. Caselli, "Efficient calibration of four wheel industrial agvs," Robotics and Computer-Integrated Manufacturing, vol. 57, pp. 116–128, 2019.

[21] E. Savaee and A. R. Hanzaki, "A new algorithm for calibration of an omni-directional wheeled mobile robot based on effective kinematic parameters estimation," Journal of Intelligent & Robotic Systems, vol. 101, no. 2, pp. 1–11, 2021.

[22] K. Lee, C. Jung, and W. Chung, "Accurate calibration of kinematic parameters for two wheel differential mobile robots," Journal of mechanical science and technology, vol. 25, no. 6, p. 1603, 2011.

[23] C. Jung and W. Chung, "Accurate calibration of two wheel differential mobile robots by using experimental heading errors," in 2012 IEEE International Conference on Robotics and Automation. IEEE, 2012, pp. 4533–4538.

[24] D. L. Tomasi and E. Todt, "Rotational odometry calibration for differential robot platforms," in 2017 Latin American Robotics Symposium (LARS) and 2017 Brazilian Symposium on Robotics (SBR). IEEE, 2017, pp. 1–6.

[25] K.-L. Han, H. Kim, and J. S. Lee, "The sources of position errors of omni-directional mobile robot with mecanum wheel," in 2010 IEEE International Conference on Systems, Man and Cybernetics. IEEE, 2010, pp. 581–586.

[26] Y. Maddahi, N. Sepehri, A. Maddahi, and M. Abdolmohammadi, "Calibration of wheeled mobile robots with differential drive mechanisms: An experimental approach," Robotica, vol. 30, no. 6, pp. 1029–1039, 2012.

[27] P. Lin, D. Liu, D. Yang, Q. Zou, Y. Du, and M. Cong, "Calibration for odometry of omnidirectional mobile robots based on kinematic correction," in 2019 14th International Conference on Computer Science & Education (ICCSE). IEEE, 2019, pp. 139–144.

[28] M. Fazekas, P. Gáspár, and B. Németh, "Calibration and improvement of an odometry model with dynamic wheel and lateral dynamics integration," Sensors, vol. 21, no. 2, p. 337, 2021.

[29] K. Lee and W. Chung, "Calibration of kinematic parameters of a car-like mobile robot to improve odometry accuracy," in 2008 IEEE International Conference on Robotics and Automation. IEEE, 2008, pp. 2546–2551.

[30] K. Yoo and W. Chung, "Convergence analysis of kinematic parameter calibration for a car-like mobile robot," in 2009 IEEE/ASME International Conference on Advanced Intelligent Mechatronics. IEEE, 2009, pp. 740–745.

[31] D. Jung, J. Seong, C.-b. Moon, J. Jin, and W. Chung, "Accurate calibration of systematic errors for car-like mobile robots using experimental orientation errors," International Journal of Precision Engineering and Manufacturing, vol. 17, no. 9, pp. 1113–1119, 2016.

[32] E. Ivanjko, I. Komsic, and I. Petrovic, "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.

[33] Y. Maddahi, A. Maddahi, and N. Sepehri, "Calibration of omnidirectional wheeled mobile robots: method and experiments," Robotica, vol. 31, no. 6, pp. 969–980, 2013.

[34] Y. Tu and H. Min, "Calibration method of mecanum wheeled mobile robot odometer," in 2019 Chinese Automation Congress (CAC). IEEE, 2019, pp. 3014–3019.

[35] H. Nie, W. Ren, B. Li, G. Liu, K. Xiong, and H. Zhang, "Multi-node calibration motion control system of three-wheel omnidirectional mobile robot," in 2021 5th International Conference on Robotics and Automation Sciences (ICRAS). IEEE, 2021, pp. 60–64.

[36] T. D. Larsen, M. Bak, N. A. Andersen, and O. Ravn, "Location estimation for autonomously guided vehicle using an augmented kalman filter to autocalibrate the odometry," in FUSION98 Spie Conference, 1998.

[37] A. Martinelli, N. Tomatis, and R. Siegwart, "Simultaneous localization and odometry self calibration for mobile robot," Autonomous Robots, vol. 22, no. 1, pp. 75–85, 2007.

[38] L. Cantelli, S. Ligama, G. Muscato, and D. Spina, "Auto-calibration methods of kinematic parameters and magnetometer offset for the localization of a tracked mobile robot," Robotics, vol. 5, no. 4, p. 23, 2016.

[39] R. B. Sousa, M. R. Petry, and A. P. Moreira, "Evolution of odometry calibration methods for ground mobile robots," in 2020 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC). IEEE, 2020, pp. 294–299.

[40] R. Pio, "Euler angle transformations," IEEE Transactions on automatic control, vol. 11, no. 4, pp. 707–715, 1966.

[41] E. A. Wan and R. V. D. Merwe, "The unscented kalman filter for nonlinear estimation." Institute of Electrical and Electronics Engineers Inc., 2000, pp. 153–158.
[42] T. Fiorenzani, C. Manes, G. Oriolo, and P. Peliti, "Comparative study of unscented kalman filter and extended kalman filter for position/attitude estimation in unmanned aerial vehicles," Inst. for Systems Analysis and Computer Science (IASI-CNR), Rome, Italy, Rept, pp. 08–08, 2008.
[43] S. Julier, J. Uhlmann, and H. F. Durrant-Whyte, "A new method for the nonlinear transformation of means and covariances in filters and estimators," IEEE Transactions on automatic control, vol. 45, no. 3, pp. 477–482, 2000.
[44] S. J. Julier, "The scaled unscented transformation," in Proceedings of the 2002 American Control Conference (IEEE Cat. No. CH37301), vol. 6. IEEE, 2002, pp. 4555–4559.
[45] G. Bradski, "The OpenCV Library," Dr. Dobb's Journal of Software Tools, 2000.
[46] Roger Labbe, Retrieved August 18, 2020, from https:github.comrlabbefilterpy, 2020.
[47] S. G. Tzafestas, Introduction to mobile robot control. Elsevier, 2013.