Calibrating Distance Sensors for Terrestrial Applications Without Groundtruth Information

Anas Alhashimi, *Student Member, IEEE,* Damiano Varagnolo, and Thomas Gustafsson

***Abstract*—This paper describes a new calibration procedure for distance sensors that does not require independent sources of groundtruth information, i.e., that is not based on comparing the measurements from the uncalibrated sensor against measurements from a precise device assumed as the groundtruth. Alternatively, the procedure assumes that the uncalibrated distance sensor moves in space on a straight line in an environment with fixed targets, so that the intrinsic parameters of the statistical model of the sensor readings are calibrated without requiring tests in controlled environments, but rather in environments where the sensor follows linear movement and objects do not move.**

The proposed calibration procedure exploits an approximated EM scheme on top of two ingredients: an heteroscedastic statistical model describing the measurement process, and a simplified dynamical model describing the linear sensor movement. The procedure is designed to be capable of not just estimating the parameters of one generic distance sensor, but rather integrating the most common sensors in robotic applications, such as Lidars, odometers and sonar rangers and learn the intrinsic parameters of all these sensors simultaneously.

Tests in a controlled environment led to a reduction of the MSE of the measurements returned by a commercial triangulation Lidar by a factor between 3 and 6, comparable to the efficiency of other state-of-the art groundtruth-based calibration procedures. Adding odometric and ultrasonic information further improved the performance index of the overall distance estimation strategy by a factor of up to 1.2. Tests also show high robustness against violating the linear movements assumption.

***Index Terms*—Expectation Maximization, distance sensors, intrinsic sensors calibration, heteroscedastic models, simultaneous sensors calibration, triangulation lidars, ultrasonic sensors, odometry**

I. Introduction

THERE are different technologies for measuring distances. Therefore, the distance measurement error distribution and variance varies with different distance sensors. Figure 1 shows three realizations of the distance measurement errors committed by three different and off-the-shelf commercial sensors typically used in robotics applications.

Calibrating the sensors above means finding how to process the raw information so to diminish its instrumental bias and noisiness levels. The standard calibration procedure is then to compare the measurements against some "groundtruth" and then learn through this comparison how the original readings are affected by bias and noise. Obtaining this groundtruth in its turn typically requires first setting up the data collection environment, then collecting the data, and eventually execute some opportunely implemented statistical learning algorithms. Since the characteristics of these sensors may change over time, it is also advisable to perform this procedure periodically.

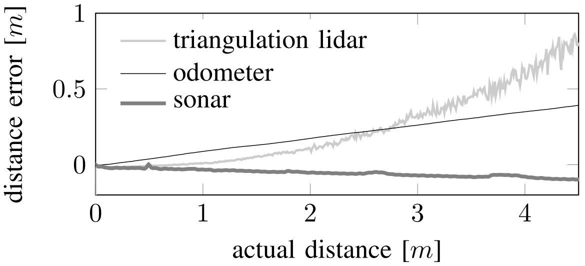

Fig. 1. Realizations of the errors committed by a triangulation Lidar, an odometer and a sonar in measuring the distance between a robot hosting these sensors and a frontal wooden obstacle, for the case where the robot was moving on a flat floor in an indoor artificially illuminated room (more details on the experimental setup are given in Section VI).

The problem with this "standard way" is that setting up a controlled environment may be expensive and time consuming, specially if the sensors are installed in some end-user commercial applications and thus spread around the globe. In this paper we focus on the need for alternative calibration strategies.

Consider then that calibration may refer to two different types of information: learning either the *intrinsic* parameters of the sensor (i.e., parameters that explain the statistics of the sensors readings), or the *extrinsic* parameters (i.e., parameters that result from how the sensor interacts with the environment like, for example, the relative positioning among different objects).

In this paper we specifically consider the problem of estimating the intrinsic parameters of distance sensors that are to be used in terrestrial robotics applications, and thus in situations where the ranges of the measurements are in the order of meters, their precision in the order of millimeters, and their typical usage is on wheeled robots. More precisely we consider how to calibrate distance sensors using, instead of groundtruth information as above, alternative structural assumptions on the statistical model of the sensor readings and on the movement of the sensor in space.

In practice thus we consider the practical need of calibrating a distance sensor mounted on an autonomous terrestrial robot while assuming that: *i)* we do not have access to groundtruth information; *ii)* we have access to the inputs given from

A. Alhashimi, D. Varagnolo and T. Gustafsson are with the Control Engineering Group, Department of Computer Science, Electrical and Space Engineering, Luleå University of Technology, Luleå 97187, Sweden. Emails: { anas.alhashimi | damiano.varagnolo | thomas.gustafsson } @ ltu.se

The research leading to these results was partially supported by Norrbottens Forskningsrd and University of Baghdad.

the robot to the wheels' motors; *iii)* the robot moves on a straight line; *iv)* the surrounding environment does not change. (Optionally, the robot may also be endowed with other non-calibrated distance-measuring sensors such as odometers and ultrasound rangers.) In practice, we consider the archetypal situation of an autonomous vacuum cleaning robot that makes straight moves in an unknown environment.

We also assume that we know the structure of the statistical models of the various sensors, so that calibration problem can be cast as a statistical inference problem (i.e., transform a dataset of distances measured by the sensors plus commands given to the robot's wheels into a (meaningful) estimate of the sensors' intrinsic parameters, under the assumption that the robot moves along a straight line).

Literature review

The most common types of distance sensors used in the terrestrial robotic applications and with the measurement ranges indicated above are Lidars, sonars [1], and odometers [1] [3].

Specific types of Lidars have specific statistical models, and leveraging on these differences different authors proposed different calibration algorithms. E.g., the characterization of the intrinsic parameters of Time of Flight (ToF) Lidars has been analyzed in [4], [5], [6], [7], [8], [9], [10], [11], [12]. More specifically, a characterization study of the Sick LMS200 Lidar with measurement drift over time, and targeted the influence of the surface properties as well as the incidence angle on the measurement process presented in [4], the paper also proposed a probabilistic range measurement model constructed starting from the experimental results. A detailed characterization of the Hokuyo URG-04LX 2D Lidar, a device with issues like time drift effects and dependencies on distance and target properties was presented in [5]. Authors here concluded that the accuracy of the sensor is strongly depending on the target properties and that it is consequently difficult to establish a calibration model. However, recently a computationally inexpensive algorithm for range correction in industrial scenarios that is based on the material surface was proposed and developed in [6]. A comparison between the Sick LMS200 and the Hokuyo URG-04LX for measurement drift over time, the effect of material and color on measurement accuracy was discussed in [7]. The geometry of the emitted beam and the *mixed pixel* effect for LMS200 Lidar were characterized in [8]. A model proposed by [9] for estimating edge loss in Lidar data by considering the impacts of various factors such as scanning distance, density of data and incidence angle on the edge loss. The evaluation results showed that using this model reduces the measurement error. The performance of CSEM SwissRanger2 and CanestaVision DP205 3D Lidars characterized in [10] , examining the effects of target range, reflectance and angle of incidence and mixed pixel effects. We analyzed in our previous work [11] how the temperature stabilization transient constitutes a source of measurement drift over time for the LMS200 Lidar. The paper proposed thus a statistical model that accounts for the bias induced by temperature changes and the laser diode mode-hopping effects, leading to an alternative calibration procedure that is based on Expectation Maximization (EM) strategies. Recently, a probabilistic modeling of Hokuyo UTM-30LX Lidar was presented in [12].

The calibration of the intrinsic parameters of triangulation Lidars has instead been studied in [13], [14], [15], [16]. More specifically, the technology introduced in [13] and performed an early-assessment of the potential for this technology; the nonlinearities affecting the triangulation Lidars produced by Neato discussed by [14]; the effects of the color of the target on the raw data returned by triangulation Lidars analyzed in [15]; and a groundtruth-based calibration procedure based on a statistical model that generalizes the one in [14] proposed in [16] .

An alternative intrinsic parameters calibration strategy specifically dedicated to multi-beam Lidars is to calibrate the various beams against each others see literature review in [17], [18] for more details.

As for the calibration of sonars, the existing literature presents ultrasonic range-finder models in [19] and opportune calibration algorithms in [20]. More specifically, a probabilistic measurement model for the sonar suitable for robot localization proposed by [19]; and a calibration model of an ultrasonic range-finder from POLAROID ultrasonic ranging system built in [20], it is based on experimental results and includes probability directional diagram of the sensor and probability estimation of the sensor measurements.

Eventually, a good literature review on odometer calibration is presented in [21]. We notice that this review presents, among others, an on-line identification algorithm for the odometer parameters for a differential drive mobile robot. Complementing the last cited work there is then [22], that describes how to treat the case of linearized odometer models and considers how the errors from this linearization process propagates to other computations.

Statement of contributions

Differently from all of the manuscripts cited above, we propose a groundtruth-less calibration strategy for inferring the intrinsic parameters of robotics-oriented distance sensors via an approximated EM algorithm. More precisely, the strategy uses the following standing assumptions:

1) the distance sensor follows the generic non-linear and heteroscedastic statistical model summarized in (2);
2) the sensor is mounted on a robot that moves along a line (even with a time varying speed; the important is that the movement is along a line);
3) the robot has knowledge about the actuation signals it gave to its motors (notice that this does not mean that the robot has knowledge of its actual velocity).

Complementary with the strategy above, we also propose two ancillary results: the first is a description of how to integrate in the calibration scheme more than one distance sensors (e.g., calibrate simultaneously Lidars, odometry and ultrasonic sensors, so to enjoy of the potential synergies in the information that these sensors provide). The second

[1] One may stretch this list and add radars [2] to it, even if they tend to be used for deeper ranges than the ones considered here.

is a description of how to use the results coming from a groundtruth-less calibration procedure to perform Kalman smoothing during normal operations where a first calibration has been performed and re-calibration is not yet needed. Notice that we ignore temporal calibration problems, i.e., ignore the effects of uncertainties in the timings of the measurements (see [23] for a treatment of this type of problems).

We eventually quantify and compare how these novel groundtruth-less calibration strategies perform compared to the groundtruth-based strategies proposed in [16], plus investigate the gains obtained combining Lidars, odometers and sonars.

Organization of the manuscript

Section II presents the statistical models of the sensors that we considered here, plus generalizes these models so that other authors with other sensors may tailor our strategy to their particular case. Section III models the dynamics of the robot. Section IV derives our calibration procedure. Section V presents a Kalman smoothing based strategy useful to exploit the results from the calibration procedure during the normal operations of the sensors. Section VI numerically compares the statistical performances of the various estimators and a Monte Carlo analysis of the sensitivity of the proposed strategies to the standing assumption of the sensor moving along a line. Section VII draws some conclusions and describes our future research efforts.

II. Sensors Modeling

Sections II-A, II-B and II-C describe the specific statistical models of the various sensors used in our EM calibration strategy. Section II-D instead the statistical models of Sections II-A, II-B and II-C and presents information useful to re-derive the equations of our EM calibration strategy for other types of ranging sensors.

A. Triangulation Lidar sensors models

In our previous work [16] we derived and validated, starting from a combination of physical and statistical considerations, a model of the measurements produced by triangulation Lidars that accounts for pinhole lens radial distortions effects and nonlinearities induced by the geometry of the laser-CCD system. The model is then

$$y_k^l = \underbrace{\alpha_0^l + \alpha_1^l d_k + \alpha_2^l d_k^2}_{\text{nonlinear bias}} + \underbrace{\beta_2^l d_k^2 e_k^l}_{\text{heteroscedastic noise}}$$

where y_k^l is the measurement at time k, d_k is the true distance, $\alpha_0^l, \alpha_1^l, \alpha_2^l$ are the parameters defining the (nonlinear) sensor bias, $e_k^l \sim \mathcal{N}(0,1)$ is independent and identically distributed (iid), and the term $\beta_2^l d_k^2$ implies that the measurement noise is heteroscedastic.

B. Odometry sensors model

Several different models describe the statistical properties of the odometers' measurements, e.g., [3], [24]. There seems to be consensus in considering, in the common case where robots have two independent traction wheels, separate errors in the translation of each wheel that increase linearly with the distance traveled and with the number of input commands given to the robot, i.e., heteroscedastic models.

Unfortunately our experiments suggest to use a homoscedastic noises (see, e.g., Figure 1). For this reason we considered in our calibration procedure the model

$$y_k^o = \underbrace{\alpha_0^o + \alpha_1^o d_k}_{\text{linear bias}} + \underbrace{\beta_0^o e_k^o}_{\text{homoscedastic noise}} \tag{1}$$

with $e_k^o \sim \mathcal{N}(0,1)$. Notice that this choice has been driven by our specific hardware, and thus has no claim of generality; nonetheless we derived our procedure using general formulas, so that if readers need to change (1) in favor of more complicated dependencies on e_k^o they can easily do so.

C. Ultrasonic ranging sensors model

Ultrasonic sensors are affected by an affine bias accounting for installation offsets and scaling of the actual distance induced by the dependency of the sound propagation speed in air on the air temperature (safely assumable constant during a calibration procedure). The measurement noise is instead typically generated from robot shaking and floor surface variations effects that generate mechanical vibration of the robot body [25]. Thus the statistical model that we consider is

$$y_k^u = \underbrace{\alpha_0^u + \alpha_1^u d_k}_{\text{affine bias}} + \underbrace{\beta_0^u e_k^u}_{\text{homoscedastic noise}}$$

where the notation and assumptions on e_k^u are similar to the ones in II-A.

D. Generic sensor model

Generalizing the results obtained in Sections II-A, II-B and II-C, we may consider a generic heteroscedastic sensor model

$$y_k^{(s)} = \underbrace{\sum_{i=0}^{N_\alpha^{(s)}-1} \alpha_i^{(s)} d_k^i}_{\text{bias}} + \underbrace{\sum_{i=0}^{N_\beta^{(s)}-1} \beta_i^{(s)} d_k^i e_k^{(s)}}_{\text{noise}} \tag{2}$$

where d_k is the noiseless distance, (s) is the sensor label, $e_k^{(s)} \sim \mathcal{N}(0,1)$ i.i.d., and the coefficients $\alpha_i^{(s)}$, $\beta_i^{(s)}$, $N_\alpha^{(s)}$ and $N_\beta^{(s)}$ define the type of bias and noise affecting that specific sensor type. This model thus captures the statistical behavior of the most typical distance sensor mentioned in our literature review.

III. Model of the dynamics of the robot

Assume that a robot is endowed with S different sensors, and that each of these sensors satisfy a measurement model like the one proposed in (2). Calibrating these sensors then means to estimate the coefficients of the various polynomials of the type (2). From statistical identifiability perspectives there is thus the need for models of the dynamics of the robot that depend on all the parameters involved in the polynomials above. In other words, it is possible to do calibration of all the parameters $\alpha_i^{(s)}$ and $\beta_i^{(s)}$ only if they are all present in the model expressing the dynamics of the overall system.

Let then the transition model for the actual distance d_k be linear Gaussian, i.e.,

$$d_{k+1} = d_k + u_k + \nu_k \tag{3}$$

with u_k the scalar input representing the motion commands given to the robot, $\nu_k \sim \mathcal{N}\left(0, \sigma_d^2\right)$ with σ_d^2 for simplicity assumed known. Define thus N_{max} as the maximum order of the polynomials appearing in (2), i.e.,

$$N_{\text{max}} := \max_s \left\{ N_\alpha^{(s)}, N_\beta^{(s)} \right\}_{s \in \{1,\dots,S\}},$$

and let the (redundant) state vector describing the robot's motion as $x_k := [1, d_k, \cdots, d_k^{N_{\text{max}}-1}]^T$. The associated (non-minimal) dynamical model is thus

$$\begin{bmatrix} x_{k+1} \\ y_k \end{bmatrix} = \begin{bmatrix} A_k & B_k \\ C & \mathbb{0} \end{bmatrix} \begin{bmatrix} x_k \\ 1 \end{bmatrix} + \begin{bmatrix} w_k \\ v_k(x_k) \end{bmatrix} \tag{4}$$

with $y_k := [y_k^{(1)}, \dots, y_k^{(S)}]^T$ the measurements vector, $\mathbb{0}$ a vector of zeros with opportune dimensions, and with (padding when necessary the various coefficients with zeros)

$$A_k := \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots \\ 0 & 1 & 0 & 0 & \cdots \\ 0 & 2u_k & 1 & 0 & \cdots \\ 0 & 3u_k^2 & 3u_k & 1 & \cdots \\ \vdots & \vdots & \ddots & \ddots & \ddots \end{bmatrix} \qquad B_k := \begin{bmatrix} 0 \\ u_k \\ \vdots \\ u_k^{N_{\text{max}}-1} \end{bmatrix}$$

$$C := \begin{bmatrix} \alpha_0^{(1)} & \cdots & \alpha_{N_{\text{max}}-1}^{(1)} \\ \vdots & & \\ \alpha_0^{(S)} & \cdots & \alpha_{N_{\text{max}}-1}^{(S)} \end{bmatrix} \tag{5}$$

with

- u_k (assumed known) conveniently absorbed into the various model matrices;
- the i-th row of A_k exhibiting the coefficients of the binomial formula $(d_k + u_k)^{i-1}$ (but the first element, that is always 0);
- the measurement noise v_k satisfying, given model (2), $v_k \sim \mathcal{N}(0, R(x_k))$ with

  $$R(x_k) := \operatorname{diag}\left(r^{(1)}(x_k), \dots, r^{(S)}(x_k) \right)$$

  and

  $$r^{(s)}(x_k) := \left(\begin{bmatrix} \beta_0^{(s)} & \cdots & \beta_{N_{\text{max}}-1}^{(s)} \end{bmatrix} x_k \right)^2.$$

We notice that the definition of x_k and the assumption $\nu_k \sim \mathcal{N}\left(0, \sigma_d^2\right)$ in (3) implies that the process noise w_k in (4) should not be Gaussian. This, unfortunately, hinders numerical tractability of the estimation processes based on (4). To overcome this problem we simplify w_k and assume it to be Gaussian, $w_k \sim \mathcal{N}(\mathbb{0}, Q)$ with Q known and diagonal with small entries. Moreover, again for the same sake of simplification, we assume the initial state to be Gaussian too, i.e., $x_1 \sim \mathcal{N}(\mu_d, \Sigma_d)$ with μ_d known and $\Sigma_d = \mathbb{0}$.

We also notice that we may have alternatively defined x_k as $[d_k, \cdots, d_k^{N_{\text{max}}-1}]^T$, but this would have led us into the need for estimating the matrices B_k (and thus system identifiability problems).

IV. An EM-based groundtruth-less calibration procedure

Let $\boldsymbol{u} := [u_1, \dots, u_N]^T$, and

$$y_k := \begin{bmatrix} y_k^{(1)} \\ \vdots \\ y_k^{(S)} \end{bmatrix}, \; k = 1, \dots, N, \qquad \boldsymbol{y} := [y_1, \dots, y_N]$$

$$x_k := \begin{bmatrix} 1 \\ d_k \\ \vdots \\ d_k^{N_{\text{max}}-1} \end{bmatrix}, \; k = 1, \dots, N+1, \qquad \boldsymbol{x} := [x_1, \dots, x_{N+1}].$$

Assuming $\boldsymbol{u}$ to be known, model (4) is fully described by the set of parameters

$$\begin{aligned} \theta := \Bigg\{ & \left\{ \alpha_0^{(s)}, \dots, \alpha_{N_{\text{max}}-1}^{(s)} \right\}_{s \in \{1,\dots,S\}}, \\ & \left\{ \beta_0^{(s)}, \dots, \beta_{N_{\text{max}}-1}^{(s)} \right\}_{s \in \{1,\dots,S\}} \Bigg\}. \end{aligned} \tag{6}$$

Our first aim is thus to estimate θ from a dataset of measurements $\boldsymbol{y}$, $\boldsymbol{u}$ collected in a non-controlled environment. In other words, we want to find a statistically meaningful map of the kind

$$\{\boldsymbol{y}, \boldsymbol{u}\} \mapsto \widehat{\theta} \tag{7}$$

where the unique additional assumption that we pose is that the robot follows a straight line and the surrounding environment does not change in the while.

We assume θ to be a deterministic unknown quantity. Given this assumption we strive for finding the Maximum Likelihood (ML) estimator for θ given $\{\boldsymbol{y}, \boldsymbol{u}\}$ in (7). Since the likelihood $p(\boldsymbol{y}, \boldsymbol{x} \; ; \; \theta)$ depends on the unknown $\boldsymbol{x}$, the natural strategy would then be to maximize the marginal likelihood of the outputs $\boldsymbol{y}$ with respect to θ, i.e., solve

$$\widehat{\theta}_{ML} := \arg\max_{\theta \in \Theta} p(\boldsymbol{y} \; ; \; \theta) \tag{8}$$

where Θ is the (assumed closed) set of admissible candidate parameters vectors, and where $p(\boldsymbol{y} \; ; \; \theta)$ is obtained integrating out from $p(\boldsymbol{y}, \boldsymbol{x} \; ; \; \theta)$ the latent r.v. $\boldsymbol{x}$. Since solving numerically (8) in our specific case is not trivial (given that the marginalization task is not trivial) we attempt to solve (8) numerically by means of an opportune EM scheme. Before presenting the specific equations of our strategy in Sections IV-B and IV-C, and for completeness of the treatment, we briefly discuss the basic machineries behind EM algorithms in the following Section IV-A. For more details on the EM algorithm see, e.g., [26].

A. The Expectation Maximization (EM) algorithm in the general case

The strategy is founded on the basic relationship

$$p(\boldsymbol{y} \; ; \; \theta) = \frac{p(\boldsymbol{y}, \boldsymbol{x} \; ; \; \theta)}{p(\boldsymbol{x} \,|\boldsymbol{y} \;\; ; \; \theta)}$$

and computes $\widehat{\theta}$ iterating the two steps (with t being the EM iteration index):

E step: given $\widehat{\theta}^{(t)}$ i.e., the estimate of the parameters at iteration t, compute

$$\ell\left(\theta,\ \widehat{\theta}^{(t)}\right) = \mathbb{E}_{p\left(\boldsymbol{x}|\boldsymbol{y}\ ;\ \widehat{\theta}^{(t)}\right)}\left[\log p\left(\boldsymbol{y}, \boldsymbol{x}\ ;\ \theta\right)\right];$$

M step: compute

$$\widehat{\theta}^{(t+1)} = \arg\max_{\theta} \ell\left(\theta,\ \widehat{\theta}^{(t)}\right).$$

The EM algorithm is ensured to make $\widehat{\theta}^{(t)}$ asymptotically converge, by iterating the two steps above, to a potentially local maximum of $p\left(\boldsymbol{y}\ ;\ \theta\right)$. Among the various plausible stopping criteria, the most common ones are to stop either when $\left\|\widehat{\theta}^{(t+1)} - \widehat{\theta}^{(t)}\right\|$ is below a given threshold, or after a pre-fixed number of iterations. The following two subsections describe in detail how to implement the EM steps in our specific case, while we report the equations of generic EM algorithms in the appendix for completeness.

B. The Expectation step in our specific case

Computing $\ell\left(\theta,\ \widehat{\theta}^{(t)}\right)$ requires to find $\log p\left(\boldsymbol{y}, \boldsymbol{x}\ ;\ \theta\right)$; consider thus that model (4) implies

$$\begin{bmatrix} x_{k+1} \\ y_k \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} A_k & B_k \\ C & \mathbb{0} \end{bmatrix}\begin{bmatrix} x_k \\ 1 \end{bmatrix}, \begin{bmatrix} Q & \mathbf{0} \\ \mathbf{0}^T & R\left(x_k\right) \end{bmatrix}\right). \tag{9}$$

with $\mathbf{0}$ a matrix of zeros with opportune dimensions, $x_1 \sim \mathcal{N}\left(\mu_d, \Sigma_d\right)$ and with μ_d and Σ_d known. Defining

$$\Sigma_k := \begin{bmatrix} Q & \mathbf{0} \\ \mathbf{0}^T & R\left(x_k\right) \end{bmatrix} \tag{10}$$

and using both the Bayes rule and the Markovianity of (4) we get

$$p\left(\boldsymbol{y}, \boldsymbol{x}\ ;\ \theta\right) = p\left(x_1\ ;\ \theta\right) \prod_{k=1}^{N} p\left(y_k \left| x_k \right.\ ;\ \theta\right) p\left(x_{k+1} \left| x_k \right.\ ;\ \theta\right)$$

that leads immediately to

$$\begin{aligned} \log p\left(\boldsymbol{y}, \boldsymbol{x}\ ;\ \theta\right) &= \log p\left(x_1\ ;\ \theta\right) + \\ &\sum_{k=1}^{N} \log p\left(y_k \left| x_k \right.\ ;\ \theta\right) + \sum_{k=1}^{N} \log p\left(x_{k+1} \left| x_k \right.\ ;\ \theta\right). \end{aligned}$$

Given (9) the joint log likelihood thus can be written as

$$\begin{aligned} \log p\left(\boldsymbol{y}, \boldsymbol{x}\ ;\ \theta\right) &\propto \mathcal{C} + \log\det \Sigma_d^{-1} - \left\|x_1 - \mu_d\right\|_{\Sigma_d}^2 + \\ &\sum_{k=1}^{N}\left(\log\det \Sigma_k^{-1} - \left\|\begin{bmatrix} x_{k+1} \\ y_k \end{bmatrix} - \begin{bmatrix} A_k & B_k \\ C & \mathbb{0} \end{bmatrix}\begin{bmatrix} x_k \\ 1 \end{bmatrix}\right\|_{\Sigma_k}^2\right) \end{aligned}$$

where $\left\|\star\right\|_{\diamond}^2 := \star^T \diamond^{-1} \star$ and $\mathcal{C}$ is a constant independent of the variables $\boldsymbol{y}$, $\boldsymbol{x}$, and θ. Applying then the conditional expectation $\mathbb{E}_{p\left(\boldsymbol{x}|\boldsymbol{y}\ ;\ \widehat{\theta}^{(t)}\right)}\left[\cdot\right]$ on both sides, expanding the norms opportunely and ignoring multiplicative factors yields to (see also [27])

$$\ell\left(\theta,\ \widehat{\theta}^{(t)}\right) = \mathcal{C} + \sum_{k=0}^{N}\left(\log\det {\Sigma_k}^{-1} - \operatorname{tr}\left(\mathcal{E}_k\right)\right) \tag{11}$$

with, for $k = 0$, $\Sigma_0 = \Sigma_d$ and

$$\mathcal{E}_0 := \mathbb{E}_{p\left(\boldsymbol{x}|\boldsymbol{y}\ ;\ \widehat{\theta}^{(t)}\right)}\left[\Sigma_d^{-1}\left(x_1 - \mu_d\right)\left(x_1 - \mu_d\right)^T\right]$$

and, for $k = 1, \ldots, N$,

$$\begin{aligned} \mathcal{E}_k :=\ & \mathbb{E}_{p\left(\boldsymbol{x}|\boldsymbol{y}\ ;\ \widehat{\theta}^{(t)}\right)}\left[\Sigma_k^{-1}\left(\begin{bmatrix} x_{k+1} \\ y_k \end{bmatrix} - \begin{bmatrix} A_k & B_k \\ C & \mathbb{0} \end{bmatrix}\begin{bmatrix} x_k \\ 1 \end{bmatrix}\right)\right. \\ &\left.\left(\begin{bmatrix} x_{k+1} \\ y_k \end{bmatrix} - \begin{bmatrix} A_k & B_k \\ C & \mathbb{0} \end{bmatrix}\begin{bmatrix} x_k \\ 1 \end{bmatrix}\right)^T\right]. \end{aligned} \tag{12}$$

Exploiting the fact that Σ_k in (10) is block diagonal, expanding we find that calculating $\operatorname{tr}\left(\mathcal{E}_k\right)$ requires computing the following quantities

$$\begin{gathered} \mathbb{E}_{p\left(\boldsymbol{x}|\boldsymbol{y}\ ;\ \widehat{\theta}^{(t)}\right)}\left[x_{k+1}x_k^T\right] \\ \mathbb{E}_{p\left(\boldsymbol{x}|\boldsymbol{y}\ ;\ \widehat{\theta}^{(t)}\right)}\left[x_k x_k^T\right] \\ \mathbb{E}_{p\left(\boldsymbol{x}|\boldsymbol{y}\ ;\ \widehat{\theta}^{(t)}\right)}\left[R\left(x_k\right)^{-1} C x_k x_k^T C^T\right] \\ \mathbb{E}_{p\left(\boldsymbol{x}|\boldsymbol{y}\ ;\ \widehat{\theta}^{(t)}\right)}\left[R\left(x_k\right)^{-1} y_k x_k^T C^T\right]. \end{gathered} \tag{13}$$

Given that $R\left(x_k\right)$ in (13) depends on x_k, the quantities above cannot be computed in closed form, but rather requires numerical integration procedures. Since we aim at algorithms that can be implemented on cheap hardware, we seek for approximating $\ell\left(\theta,\ \widehat{\theta}^{(t)}\right)$ in (11) with an alternative approximated version $\widetilde{\ell}\left(\theta,\ \widehat{\theta}^{(t)}\right)$ with closed-form computability qualities.

To this point we notice that if the covariances $R\left(\cdot\right)$ were independent of x_k then we would be in the very same situation of [27], and thus we would be able to compute (13) by means of a dedicated Kalman smoother. We thus follow this approach, and approximate $R\left(\cdot\right)$ by considering x_k being equal to its past estimated value.

More precisely, assume to be at iteration t of the EM algorithm; this means that at time $t-1$ we have computed both an estimate of the parameters $\widehat{\theta}^{(t)}$ and an estimate of the state $\widehat{x}_k^{(t-1)}$ (with initial condition $\widehat{x}_k^{(0)} = \left[1, \text{mean}(y_k), \ldots, \text{mean}(y_k)^{N_{\max}-1}\right]$). Define thus

$$R_k^{(t)} := \operatorname{diag}\left({r^{(1)}}^{(t)}\left(\widehat{x}_k^{(t-1)}\right), \ldots, {r^{(S)}}^{(t)}\left(\widehat{x}_k^{(t-1)}\right)\right)$$

$${r^{(s)}}^{(t)}\left(\widehat{x}_k^{(t-1)}\right) := \left(\begin{bmatrix} {\beta_0^{(s)}}^{(t)} & \cdots & {\beta_{N_{\max}-1}^{(s)}}^{(t)} \end{bmatrix}\widehat{x}_k^{(t-1)}\right)^2$$

and ${\beta_0^{(s)}}^{(t)}, \ldots, {\beta_{N_{\max}-1}^{(s)}}^{(t)}$ the set of parameters modeling sensor s estimated at time t by the EM algorithm. $R_k^{(t)}$ is thus a statistically meaningful approximation of the actual noise covariance $R\left(x_k\right)$, and with this we can approximate the quantities in (13) (and thus $\ell\left(\theta,\ \widehat{\theta}^{(t)}\right)$) by means of the following Algorithm 1. For convenience we indicate with $C^{(t)}$ the estimate of matrix C in (5) and the process noise covariance defined by the current estimate of the parameters $\widehat{\theta}^{(t)}$.

Algorithm 1 Kalman smoother for the Expectation step

```
Requires: $C^{(t)}$, $R_1^{(t)}, \dots, R_N^{(t)}$
set (initial conditions for the forward pass)
    $P_{1|1} = \Sigma_1 \qquad \widehat{x}_{1|1} = \mu_1$
compute, for $k = 1, \dots, N$ (forward pass)
    $P_{k|k-1} = A_k P_{k-1|k-1} A_k^T + Q$
    $K_k = P_{k|k-1} {C^{(t)}}^T \left( C^{(t)} P_{k|k-1} {C^{(t)}}^T + R_k^{(t)} \right)^{-1}$
    $P_{k|k} = P_{k|k-1} - K_k C^{(t)} P_{k|k-1}$
    $\widehat{x}_{k|k-1} = A_k \widehat{x}_{k-1|k-1} + B_k 1$
    $\widehat{x}_{k|k} = \widehat{x}_{k|k-1} + K_k \left( y_k - C^{(t)} \widehat{x}_{k|k-1} \right)$
set (initial conditions for the backwards pass)
    $M_{N|N} = \left( I - K_N C^{(t)} \right) A_k P_{N-1|N-1}$
compute, for $k = N, \dots, 1$ (backwards pass)
    $J_k = P_{k|k} A^T P_{k+1|k}^{-1}$
    $P_{k|N} = P_{k|k} + J_k \left( P_{k+1|N} - P_{k+1|k} \right) J_k^T$
    $\widehat{x}_{k|N} = \widehat{x}_{k|k} + J_k \left( \widehat{x}_{k+1|N} - A_k \widehat{x}_{k|k} - B_k 1 \right)$
    $M_{k|N} = P_{k|k} J_{k-1}^T + J_k \left( M_{k+1|N} - A_k P_{k|k} \right) J_{k-1}^T$
    (the last equation being performed only when $k \neq N$)
```

Exploiting the results in [28], thus, we can claim that

$$
\begin{aligned}
\mathbb{E}_{p\left(\boldsymbol{x}|\boldsymbol{y}\ ;\ \widehat{\theta}_t\right)} \left[x_k x_k^T \right] &\approx \widehat{x}_{k|N} \widehat{x}_{k|N}^T + P_{k|N} \\
\mathbb{E}_{p\left(\boldsymbol{x}|\boldsymbol{y}\ ;\ \widehat{\theta}_t\right)} \left[x_{k+1} x_k^T \right] &\approx \widehat{x}_{k+1|N} \widehat{x}_{k|N}^T + M_{k+1|N} \\
\mathbb{E}_{p\left(\boldsymbol{x}|\boldsymbol{y}\ ;\ \widehat{\theta}_t\right)} \left[y_k x_k^T \right] &\approx y_k \widehat{x}_{k|N}^T .
\end{aligned}
\tag{14}
$$

Approximating $R\left(x_k\right)$ with $R_k^{(t)}$ thus leads to approximate the expectations (13) with (14), and thus to approximate $\ell\left(\theta,\ \widehat{\theta}^{(t)}\right)$ with an opportune $\widetilde{\ell}\left(\theta,\ \widehat{\theta}^{(t)}\right)$ obtainable expanding $\mathcal{E}_k$ in (12) into single factors and exploiting the fact that $R_k^{(t)}$ does not depend on x_k. Moreover when Algorithm 1 terminates we can also set $\widehat{\boldsymbol{x}}^{(t)} = \left[\widehat{x}_{1|N}, \dots, \widehat{x}_{N|N} \right]$.

We notice that approximating ℓ with $\widehat{\ell}$ may theoretically disrupt the convergence properties of our EM strategy (something that we never experienced, though); proving the stability of the proposed scheme is nonetheless out of scope here and currently under analytical investigation.

C. The Maximization step in our specific case

Given the discussion above, we solve the Maximization step by searching that parameter vector that maximizes $\widetilde{\ell}\left(\theta,\ \widehat{\theta}^{(t)}\right)$, i.e., by computing

$$
\widehat{\theta}^{(t+1)} = \arg\max_{\theta \in \Theta} \widetilde{\ell}\left(\theta,\ \widehat{\theta}^{(t)}\right) . \tag{15}
$$

by means of closed form equations and considering the latent variables $\boldsymbol{x}$ to be equal to that $\boldsymbol{x}^{(t)}$ computed in the Expectation step. Given definition (6), estimating θ means finding the matrix C (that contains the various $\left\{ \alpha_0^{(s)}, \dots, \alpha_{N_{\max}-1}^{(s)} \right\}_{s \in \{1,\dots,S\}}$), and the matrix $R\left(x_k^{(t)}\right)$ (that contains the various $\left\{ \beta_0^{(s)}, \dots, \beta_{N_{\max}-1}^{(s)} \right\}_{s \in \{1,\dots,S\}}$).

As shown in the next subsections, the actual equations for solving (15) depend on which combination of sensors one uses.

1) The Maximization step when using just a triangulation Lidar: Having a triangulation Lidar only, and considering the likelihood (12), our aim is to estimate $C = \left[\alpha_0^l, \alpha_1^l, \alpha_2^l \right]$ and the variance parameter β_2^l, given the current estimate of the state provided by the previous Expectation step $\widehat{\boldsymbol{x}}$.

Considering the structure of (12), this requires to compute

$$
\begin{aligned}
C^{(t+1)} = &\left(\sum_{k=1}^{N} \mathbb{E}_{p\left(\boldsymbol{x}|\boldsymbol{y}\ ;\ \widehat{\theta}^{(t)}\right)} \left[R(x_k)^{-1} y_k x_k^T \right] \right) \\
&\left(\sum_{k=1}^{N} \mathbb{E}_{p\left(\boldsymbol{x}|\boldsymbol{y}\ ;\ \widehat{\theta}^{(t)}\right)} \left[R(x_k)^{-1} x_k x_k^T \right] \right)^{-1} .
\end{aligned}
\tag{16}
$$

Unfortunately, since $R(x_k)^{-1}$ is a function of the states, solving (16) analytically proves to be very complicated. To reduce the computational burden associate to this step we repeat the same approximation we performed before on $R\left(x_k\right)$, and consider it equal to $R(\widehat{x}_k^{(t-1)})$. Therefore we rewrite (16) by substituting $R(x_k)$ with $\left([0, 0, \beta_2^l] \widehat{x}_k^{(t-1)} \right)^2 = \left(\beta_2^l \widehat{x}_k^{(t-1)}[3] \right)^2$ where $\widehat{x}_k^{(t-1)}[3]$ is the third element in the vector $\widehat{x}_k^{(t-1)}$. Since this quantity is not a function of x_k anymore, we can safely take it out of the expectations in (16), apply (14) and thus approximate the Maximization step with

$$
\begin{aligned}
C^{(t+1)} \approx &\left(\sum_{k=1}^{N} (\widehat{x}_k^{(t-1)}[3])^{-2} y_k \widehat{x}_{k|N}^T \right) \\
&\left(\sum_{k=1}^{N} (\widehat{x}_k^{(t-1)}[3])^{-2} \widehat{x}_{k|N} \widehat{x}_{k|N}^T + (\widehat{x}_k^{(t-1)}[3])^{-2} P_{k|N} \right)^{-1} .
\end{aligned}
$$

To complete the Maximization step, we apply a similar procedure for the computation of β_2^l and obtain

$$
\begin{aligned}
\left(\beta_2^{l\ (t+1)} \right)^2 = &\frac{1}{N} \sum_{k=1}^{N} (\widehat{x}_k^{(t-1)}[3])^{-2} y_k y_k^T \\
&- \sum_{k=1}^{N} (\widehat{x}_k^{(t-1)}[3])^{-2} C^{(t+1)} \widehat{x}_k y_k^T .
\end{aligned}
$$

2) The Maximization step when using both a triangulation Lidar and an odometer: In this case the matrix C is equal to

$$
C = \begin{bmatrix} C[1] \\ C[2] \end{bmatrix} = \begin{bmatrix} \alpha_0^l & \alpha_1^l & \alpha_2^l \\ \alpha_0^o & \alpha_1^o & \alpha_2^o \end{bmatrix}
$$

where the parameters of the triangulation Lidar are given by $C[1]$ and β_2^l, and the ones of the odometer are given by $C[2]$ and β_0^o.

Given the $\widehat{\boldsymbol{x}}^{(t)}$ computed in the E step, and the fact that the two sensors are independent, the estimation of the two sets of parameters can be performed independently. Thus for $C[1]$ and β_2^l we can proceed as in Section IV-C1, while for $C[2]$ and β_0^o we proceed considering that, given model (1) and using again ML interpretations,

$$C^{(t+1)}[2] = \left(\sum_{k=1}^{N} y_k \widehat{x}_{k|N}^T\right)\left(\sum_{k=1}^{N} \widehat{x}_{k|N}\widehat{x}_{k|N}^T + P_{k|N}\right)^{-1}$$

$$\left(\beta_0^{o(t+1)}\right)^2 = \frac{1}{N}\sum_{k=1}^{N} y_k y_k^T - \sum_{k=1}^{N} C^{(t+1)}[2]\widehat{x}_k y_k^T$$

(17)

3) The Maximization step when using a triangulation Lidar, an odometer and a sonar: In this case the matrix C is equal to

$$C = \begin{bmatrix} C[1] \\ C[2] \\ C[3] \end{bmatrix} = \begin{bmatrix} \alpha_0^l & \alpha_1^l & \alpha_2^l \\ \alpha_0^o & \alpha_1^o & \alpha_2^o \\ \alpha_0^u & \alpha_1^u & \alpha_2^u \end{bmatrix}$$

where the parameters of the triangulation Lidar are given by $C[1]$ and β_2^l, the ones of the odometer are given by $C[2]$ and β_0^o, and the ones of the ultrasonic ranger are given by $C[3]$ and β_0^u. The situation is as before, where sensors' parameters can be learned independently; one may then repeat the procedures in Sections IV-C1 and IV-C2, and then apply strategy (17) for the particular case of the ultrasonic data $\boldsymbol{y}^u$ and $C[3]$.

V. Using the results of the EM calibration algorithm for testing purposes

The EM algorithm in Section (IV) returns two different quantities:

- an estimate $\widehat{\boldsymbol{x}}$ of the latent variables $\boldsymbol{x}$, from which one can also estimate the various d_ks;
- an estimate $\widehat{\theta}$ of the calibration parameters θ.

The EM strategy thus can be directly used to transform the raw measurements $\boldsymbol{y}$ into some statistical estimate of the distances d_k. Nonetheless the EM algorithm may be computationally demanding, and one may prefer to run it only when strictly necessary.

Assume thus to have run the EM calibration algorithm fully once, and want now to process some new raw data with a more lightweight estimation strategy. Given that we have $\widehat{\theta}$, at this point to obtain an estimated $\widehat{\boldsymbol{x}}$ (and thus $\widehat{d}_k$s) one simply has to run just once the Kalman smoother defined in Algorithm 1 (again with initial conditions $\widehat{x}_k^{(0)} = \left[1, \text{mean}(y_k), \ldots, \text{mean}(y_k)^{N_{\text{max}}-1}\right]$).

As will be shown in Section VI, this heuristic is fast and provides results similar to a dedicated complete EM algorithm when applied to a dataset with small sample size.

VI. Numerical results

We consider datasets where the same robot moves with different constant speeds (0.1, 0.2, and 0.3 m/s) towards a fixed wooden target starting at a distance of 0.5m and ending at a distance of 4m. The experiment setup is shown in Figure 2. The robot mounts a Lidar, an odometer, and a sonar sensors, and we collect the groundtruth distances d_ks using a Vicon motion capture system. We measure the statistical performance of an estimate $\widehat{d}_1, \ldots, \widehat{d}_N$ with the normalized empirical Mean Squared Error (MSE)

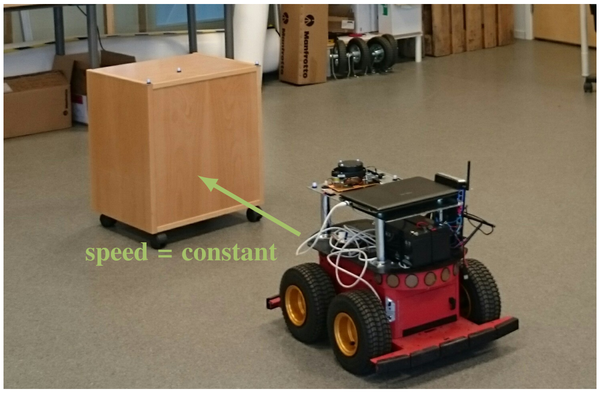

Fig. 2. Photo of the experimental setup used for collecting the dataset. The photo shows the robot, the Lidar and the wooden target.

$$\text{NMSE} := \frac{1}{N}\sum_{k=1}^{N} \frac{\left\|\widehat{d}_k - d_k\right\|^2}{\left\|d_k\right\|^2}.$$

A. Testing the strategy in Section IV

We now analyze how the groundtruth-less EM calibration procedure compares w.r.t. a groundtruth-based one, and what is the influence of using more than one sensor. More specifically, Figure 3 plots the normalized MSE for the full EM strategy in Section IV. As the legend indicates, the figure considers different combinations of sensors or calibration algorithms: indeed *raw Lidar data* indicates the MSE of the Lidar measurements y_k^l, while *groundtruth-based* indicates the MSE obtained when training[2] the estimator proposed in [16]. The three remaining signals indicate different combinations of sensors. Eventually the figure presents the results obtained for different robot speeds.

The collected evidence indicates that in our setup combining odometry measurements does not improve the estimation outcomes while adding the sonar does. Moreover increasing the speed of the robot (that in practice corresponds to diminishing the number of samples in the dataset) leads, as expected, to a generalized worsening of the estimation performance.

Figure 4 eventually gives a rough indication of how the computational complexity of our EM strategy depends on the number of samples in the dataset and on the number of sensors used. More precisely the figure plots the convergence time of the algorithm implemented in Matlab on a standard laptop.

B. Testing the strategy in Section V

We then analyze if it is necessary to run the full EM algorithm every time, or if we can do just one calibration once and after that use the Kalman smoother. Beside this, we check if and (in case) how the performance of the Kalman smoother depends on the size of the test dataset.

[2] Notice that for this case we considered the training MSE and not the test one so to be even more unfavorable comparisons against our novel procedure.

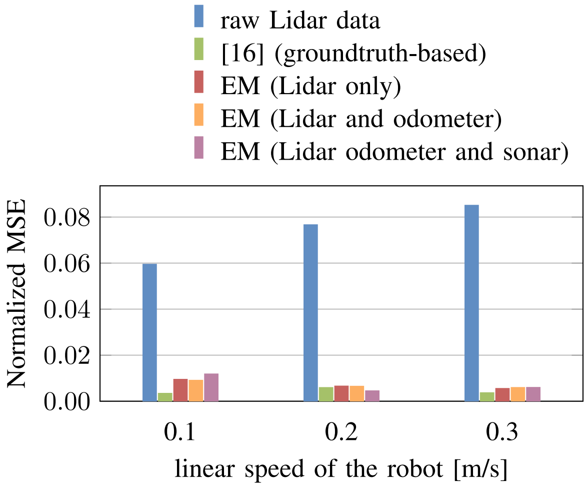

Fig. 3. Comparison of the normalized empirical MSE for various types of estimators testing using different combinations of Lidar, odometer and sonar sensors.

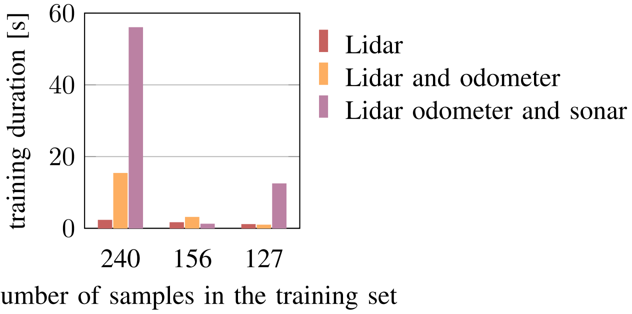

Fig. 4. Dependency of the convergence time of the algorithm on the number of samples in the dataset used for training purposes.

To answer these questions we use one dataset recorded at 0.1 m/s for performing the training step (i.e., we use the richest dataset in terms of number of samples), and we then apply the Kalman smoother as indicated in Section V to the other datasets.

Figure 5 then shows the normalized empirical MSE indexes obtained for various speeds and estimators. In this figure *raw Lidar data* indicates once again the MSE of the raw Lidar measurement y_k^l, and *groundtruth-based* indicates the MSE obtained testing the estimator proposed in [16] and trained as in Figure 3. The other three entries instead refer to using the Kalman smoother on test sets windows that are 12 samples long; for these estimators, each bar in the plot thus represents the average of the MSEs calculated along the whole dataset at a given speed. We notice how the novel strategy compares favorably against the groundtruth-based one, indicating that it reaches good generalization capabilities. Also in this plot, combining the odometer measurements with the Lidar measurements does not improve the MSE in most of the cases.

Figure 6 eventually answers the question about how the performance of the Kalman smoother depends on the size of the training set, focusing just on the case where one uses only the Lidar sensor. We see that larger sample sizes almost always return a smaller MSE, and that also increasing the robot linear speed seems to decrease the normalized MSE for speeds smaller than 0.4 m/s.

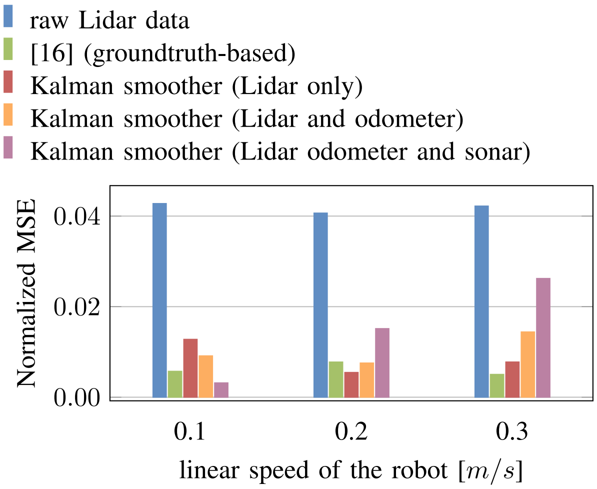

Fig. 5. Comparison of the test-set performance of various estimators.

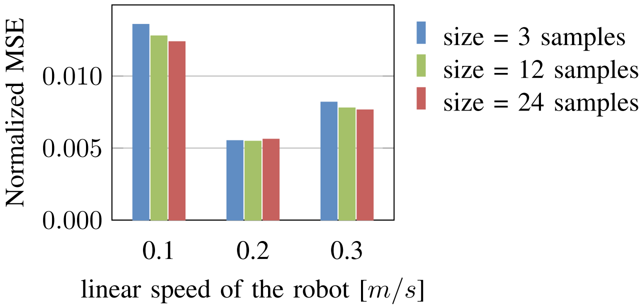

Fig. 6. Dependency of the performance of the Kalman smoothing strategy on the size of the test set.

C. Analysis of the sensitivity of the results against errors in the heading angles

Theoretically it may be possible to design opportune hypothesis testing algorithms for detecting if the robot is turning while moving, i.e., if it is violating our standing assumption of traveling along a line. This issue, however, is both outside the scope of this manuscript plus it deserves a thorough discussion and analysis; we thus neglect this problem for now and consider it as a future work. We are instead interested in understanding how sensitive the results we presented above are on the violation of the standing assumption of movement along a line (something that, being ideal, will never actually occur in practice).

To this aim we first verify how much our experiments violate this assumption. We thus plot in Figure 7 how much the actual heading of our robot changed in time when applying to it one movement-command, and this for several experiments with different speeds. As it can be noticed from the figure, the maximal absolute robot heading error was below 0.3°.

In order to quantify the effect of this heading angle error on both the estimate of the model parameters and on the final estimate of the distance between the robot and the target we use Monte Carlo approach where we run 100 independent simulations for different heading angle errors comprised between $-5°$ and $5°$ and with an angular separation of $0.25°$. We then show graphically the influence of these heading angle errors on the outcomes of both the parameter estimation and distance estimation algorithms in Figures 9 and 8 respectively. Remarkably, and according to the simulation results, non negligible initial heading errors lead to negligible final effects. A posteriori this is actually expected, since the cosine of the non-null heading angles errors will be almost equal to 1 (specially for heading angles smaller than $0.3°$).

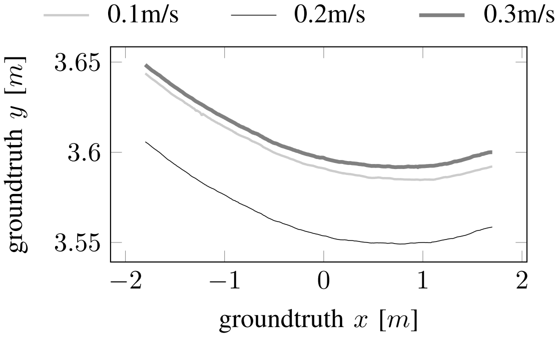

Fig. 7. The actual path that our robot performed during some of our lab experiments. In these cases the robot was given a single command of performing a linear-movement for different translation speeds.

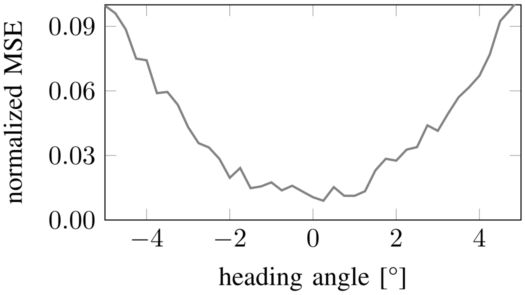

Fig. 8. Monte Carlo simulations showing the effect of the error on the heading angle on the normalized mean squared distance estimation error.

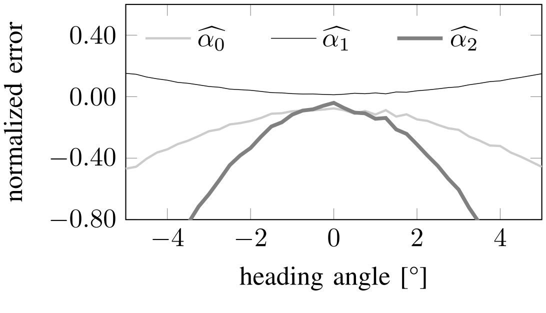

Fig. 9. Monte Carlo simulations showing the effect of the error on the heading angle on the final estimated model parameters.

VII. Conclusions

We designed and tested an approximated EM intrinsic parameters calibration strategy that can calibrate sets of either homoscedastic or heteroscedastic sensors such as Lidars, sonars and odometers. The algorithm does not require knowledge on groundtruth information, but alternatively exploits assumptions of motion along a line and knowledge about which actuation inputs have been given to the motors of the robot carrying the distance sensors.

The purpose of this effort is to make it possible (and computationally cheap) for terrestrial robots to autonomously recalibrate their sensors whenever they feel they need it, without having to go back to the factory or requiring the aid of technicians.

The proposed calibration procedure has also been shown to compete with alternative groundtruth-based strategies, validating thus our efforts at least in our experimental setup.

The numerical results indeed indicate that:

1) the novel groundtruth-less calibration strategy leads to results that are similar to the ones that can be obtained with groundtruth-based strategies (and sometimes even better). This is not totally surprising: we indeed postulate that the information on the actuation signal given to the robots' wheels (that was not used, e.g., in [16]) compensates for the loss of the groundtruth;
2) adding odometers and sonars tend to improve the overall estimation performance, but the improvement also tends to be quite contained, indicating that the additional information brought from these sensors is minimal;
3) violating the standing assumption of the movement being confined to a line is not disrupting the proposed estimation procedures, since the sensitivity of these errors on the heading error has been numerically evaluated as very low.

Despite promising, the strategy hasn't been fully developed yet: future research efforts are then directed to both prove the theoretical convergence properties of the overall EM scheme, plus generalize the strategy towards also aerial robotic applications, where our assumptions on the robot moving on a line are not valid anymore. Moreover an interesting research direction is on understanding how to infer if the robot is not moving along a line from the data and potentially additional assumptions on the structure of the surrounding environment. This indeed would help developing a control signal that guarantees straight-line motions, that is of particular interest in robotics applications.

Appendix
Equations of Generic EM Algorithm

Consider the following generic model

$$\begin{bmatrix} x_{k+1} \\ y_k \end{bmatrix} = \begin{bmatrix} A & B \\ C & \mathbb{0} \end{bmatrix} \begin{bmatrix} x_k \\ 1 \end{bmatrix} + \begin{bmatrix} w_k \\ v_k \end{bmatrix} \tag{18}$$

with $w_k \sim \mathcal{N}(\mathbb{0}, Q)$ and iid, $v_k \sim \mathcal{N}(\mathbb{0}, \beta R_k)$ and iid, $\mathbb{0}$ a vector of zeros with opportune dimensions, Q and R_k are diagonal matrices and β^2 is unknown scalar.

Given a measurement sequence $\{y_k\}$ and the fully known initial condition $x_1 \sim \mathcal{N}(\mu_d, \Sigma_d)$, we would like to apply the EM to estimate the matrix C and the scalar β^2.

A. Compute $\ell\left(\theta,\ \widehat{\theta}^{(t)}\right)$

Computing $\ell\left(\theta,\ \widehat{\theta}^{(t)}\right)$ requires finding $\log p\left(\boldsymbol{y},\boldsymbol{x}\ ;\ \theta\right)$; given the above model and follow the same procedure described in Subsection IV-B gives

$$\ell\left(\theta,\ \widehat{\theta}^{(t)}\right) = \mathcal{C} + \sum_{k=0}^{N}\left(\log\det{\Sigma_k}^{-1} - \operatorname{tr}\left(\mathcal{E}_k\right)\right) \tag{19}$$

with, for $k=0$, $\Sigma_0 = \Sigma_d$ and

$$\mathcal{E}_0 := \mathbb{E}_{p\left(\boldsymbol{x}|\boldsymbol{y}\ ;\ \widehat{\theta}^{(t)}\right)}\left[\Sigma_d^{-1}\left(x_1-\mu_d\right)\left(x_1-\mu_d\right)^T\right]$$

and, for $k = 1,\dots,N$,

$$\begin{aligned}\mathcal{E}_k :=\ & \mathbb{E}_{p\left(\boldsymbol{x}|\boldsymbol{y}\ ;\ \widehat{\theta}^{(t)}\right)}\left[\Sigma_k^{-1}\left(\begin{bmatrix}x_{k+1}\\ y_k\end{bmatrix} - \begin{bmatrix}A & B\\ C & \mathbb{0}\end{bmatrix}\begin{bmatrix}x_k\\ 1\end{bmatrix}\right)\right.\\ & \left.\left(\begin{bmatrix}x_{k+1}\\ y_k\end{bmatrix} - \begin{bmatrix}A & B\\ C & \mathbb{0}\end{bmatrix}\begin{bmatrix}x_k\\ 1\end{bmatrix}\right)^T\right]\end{aligned}$$

and

$$\Sigma_k := \begin{bmatrix}Q & \mathbf{0}\\ \mathbf{0}^T & \beta^2 R_k\end{bmatrix}$$

since we are interested in finding C and β, calculating $\operatorname{tr}\left(\mathcal{E}_k\right)$ requires computing the following quantities

$$\mathbb{E}_{p\left(\boldsymbol{x}|\boldsymbol{y}\ ;\ \widehat{\theta}^{(t)}\right)}\left[x_k x_k^T\right]$$
$$\mathbb{E}_{p\left(\boldsymbol{x}|\boldsymbol{y}\ ;\ \widehat{\theta}^{(t)}\right)}\left[y_k x_k^T\right].$$

We follow the same procedure in [28] and using the identity

$$\operatorname{cov}\left(x_k, x_k^T\right) := \mathbb{E}_p\left[x_k x_k^T\right] - \mathbb{E}_p\left[x_k\right]\mathbb{E}_p\left[x_k^T\right]$$

or

$$P_{k|N} = \mathbb{E}_{p\left(\boldsymbol{x}|\boldsymbol{y}\ ;\ \widehat{\theta}_t\right)}\left[x_k x_k^T\right] - \widehat{x}_{k|N}\widehat{x}_{k|N}^T$$

also

$$\begin{aligned}\mathbb{E}_{p\left(\boldsymbol{x}|\boldsymbol{y}\ ;\ \widehat{\theta}_t\right)}\left[y_k x_k^T\right] &= y_k\mathbb{E}_{p\left(\boldsymbol{x}|\boldsymbol{y}\ ;\ \widehat{\theta}_t\right)}\left[x_k^T\right]\\ &= y_k\widehat{x}_{k|N}^T.\end{aligned}$$

Therefore we need to find the terms $\widehat{x}_{k|N}^T$ and $P_{k|N}$ in the E step

B. E step

given $\widehat{\theta}^{(t)}$ i.e., the estimate of the parameters at iteration t, compute

$$\ell\left(\theta,\ \widehat{\theta}^{(t)}\right) = \mathbb{E}_{p\left(\boldsymbol{x}|\boldsymbol{y}\ ;\ \widehat{\theta}^{(t)}\right)}\left[\log p\left(\boldsymbol{y},\boldsymbol{x}\ ;\ \theta\right)\right]$$

which is finding the smoothed state $\widehat{x}_{k|N}^T$ and the smoothed covariance $P_{k|N}$. This can be done optimally using appropriate Kalman smoother as explained in Algorithm 2

C. M step

compute $\widehat{\theta}^{(t+1)} := \{\widehat{C}^{(t+1)}, \widehat{\beta^2}^{(t+1)}\}$ using

$$\widehat{\theta}^{(t+1)} = \arg\max_{\theta}\ell\left(\theta,\ \widehat{\theta}^{(t)}\right).$$

We present here a detailed derivation of the maximization step for the generic model (18). Starting with Equation (19)

Algorithm 2 Kalman smoother for the Expectation step

```
Requires: $C^{(t)}$, $(\beta^2)^{(t)}$
set (initial conditions for the forward pass)
    $P_{1|1} = \Sigma_1 \qquad \widehat{x}_{1|1} = \mu_1$
compute, for $k = 2,\dots,N$ (forward pass)
    $P_{k|k-1} = AP_{k-1|k-1}A^T + Q$
    $K_k = P_{k|k-1}{C^{(t)}}^T\left(C^{(t)}P_{k|k-1}{C^{(t)}}^T + R_k(\beta^2)^{(t)}\right)^{-1}$
    $P_{k|k} = P_{k|k-1} - K_kC^{(t)}P_{k|k-1}$
    $\widehat{x}_{k|k-1} = A\widehat{x}_{k-1|k-1} + B1$
    $\widehat{x}_{k|k} = \widehat{x}_{k|k-1} + K_k\left(y_k - C^{(t)}\widehat{x}_{k|k-1}\right)$
set (initial conditions for the backwards pass)
    $M_{N|N} = \left(I - K_NC^{(t)}\right)AP_{N-1|N-1}$
compute, for $k = N,\dots,1$ (backwards pass)
    $J_k = P_{k|k}A^TP_{k+1|k}^{-1}$
    $P_{k|N} = P_{k|k} + J_k\left(P_{k+1|N} - P_{k+1|k}\right)J_k^T$
    $\widehat{x}_{k|N} = \widehat{x}_{k|k} + J_k\left(\widehat{x}_{k+1|N} - A\widehat{x}_{k|k} - B1\right)$
    $M_{k|N} = P_{k|k}J_{k-1}^T + J_k\left(M_{k+1|N} - AP_{k|k}\right)J_{k-1}^T$
    (the last equation being performed only when $k \neq N$)
```

since we would like to estimate C and β, we keep only the second row of $\mathcal{E}_k$, we call it $\bar{\mathcal{E}}_k$ for $k = 1,\dots,N$

$$\ell\left(\theta,\ \widehat{\theta}^{(t)}\right) \propto -\sum_{k=1}^{N}\log\beta^2 - \sum_{k=1}^{N}\operatorname{tr}\left(\bar{\mathcal{E}}_k\right)$$

$$\bar{\mathcal{E}}_k := \mathbb{E}_{p\left(\boldsymbol{x}|\boldsymbol{y}\ ;\ \widehat{\theta}^{(t)}\right)}\left[\beta^{-2}R_k^{-1}\left(y_k - Cx_k\right)\left(y_k - Cx_k\right)^T\right]$$

Taking the zero of the score with respect to C and using the matrix derivative identity $\frac{\partial}{\partial X}\operatorname{tr}\left(\left(AXB+C\right)\left(AXB+C\right)^T\right) = 2A^TAXBB^T + 2A^TCB^T$. In this case $A = I$, $C = -y_k$ and $B = x_k$, yields

$$\begin{aligned}\frac{\partial\ell\left(\theta,\ \widehat{\theta}^{(t)}\right)}{\partial C} &= 2\sum_{k=1}^{N}\mathbb{E}_{p\left(\boldsymbol{x}|\boldsymbol{y}\ ;\ \widehat{\theta}^{(t)}\right)}\left[\beta^{-2}R_k^{-1}Cx_kx_k^T\right]\\ &-2\sum_{k=1}^{N}\mathbb{E}_{p\left(\boldsymbol{x}|\boldsymbol{y}\ ;\ \widehat{\theta}^{(t)}\right)}\left[\beta^{-2}R_k^{-1}y_kx_k^T\right]\end{aligned}$$

for single output case (R_k is 1×1), we can solve for C to obtain the estimator

$$\begin{aligned}C^{(t+1)} = &\left(\sum_{k=1}^{N}\mathbb{E}_{p\left(\boldsymbol{x}|\boldsymbol{y}\ ;\ \widehat{\theta}^{(t)}\right)}\left[R_k^{-1}y_kx_k^T\right]\right)\\ &\left(\sum_{k=1}^{N}\mathbb{E}_{p\left(\boldsymbol{x}|\boldsymbol{y}\ ;\ \widehat{\theta}^{(t)}\right)}\left[R_k^{-1}x_kx_k^T\right]\right)^{-1}.\end{aligned} \tag{22}$$

Now to find β^2, we use the identity $\operatorname{tr}\left(aX\right) = a\operatorname{tr}\left(X\right)$ for scalar a, then taking the zero of the score with respect to β^2

$$\frac{\partial \ell\left(\theta,\ \widehat{\theta}^{(t)}\right)}{\partial \beta^2} = -\frac{N}{\beta^2} + \beta^{-2} \sum_{k=1}^{N} \operatorname{tr}\left(\bar{\mathcal{E}}_k\right) = 0 \qquad \text{or}$$

$$(\beta^2)^{(t+1)} = \frac{1}{N} \sum_{k=1}^{N} \mathbb{E}_{p\left(\boldsymbol{x}|\boldsymbol{y}\ ;\ \widehat{\theta}^{(t)}\right)} \left[R_k^{-1}(y_k - Cx_k)\left(y_k - Cx_k\right)^T\right]$$

for the case of single output, we can substitute for $C^{(t+1)}$ from Equation (C) in the last term of the bracket expansion, then simplify to obtain

$$(\beta^2)^{(t+1)} = \frac{1}{N} \sum_{k=1}^{N} \mathbb{E}_{p\left(\boldsymbol{x}|\boldsymbol{y}\ ;\ \widehat{\theta}^{(t)}\right)} \left[R_k^{-1}\left(y_k y_k^T - C^{(t+1)} x_k y_k^T\right)\right]$$

References

[1] H. Hua, Y. Wang, and D. Yan, “A low-cost dynamic range-finding device based on amplitude-modulated continuous ultrasonic wave,” *IEEE transactions on instrumentation and measurement*, vol. 51, no. 2, pp. 362–367, 2002.

[2] A. Foessel-Bunting, “Radar sensor model for three-dimensional map building,” in *Intelligent Systems and Smart Manufacturing*. International Society for Optics and Photonics, 2001, pp. 127–138.

[3] K. S. Chong and L. Kleeman, “Accurate odometry and error modelling for a mobile robot,” in *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, vol. 4. IEEE, 1997, pp. 2783–2788.

[4] C. Ye and J. Borenstein, “Characterization of a 2-d laser scanner for mobile robot obstacle negotiation,” in *ICRA*, 2002, pp. 2512–2518.

[5] L. Kneip, F. Tâche, G. Caprari, and R. Siegwart, “Characterization of the compact hokuyo urg-04lx 2d laser range scanner,” in *Robotics and Automation, 2009. ICRA’09. IEEE International Conference on*. IEEE, 2009, pp. 1447–1454.

[6] C. N. MacLeod, R. Summan, G. Dobie, and S. G. Pierce, “Quantifying and improving laser range data when scanning industrial materials,” *IEEE Sensors Journal*, vol. 16, no. 22, pp. 7999–8009, 2016.

[7] K.-H. Lee and R. Ehsani, “Comparison of two 2d laser scanners for sensing object distances, shapes, and surface patterns,” *Computers and electronics in agriculture*, vol. 60, no. 2, pp. 250–262, 2008.

[8] R. Sanz-Cortiella, J. Llorens-Calveras, J. R. Rosell-Polo, E. Gregorio-Lopez, and J. Palacin-Roca, “Characterisation of the lms200 laser beam under the influence of blockage surfaces. influence on 3d scanning of tree orchards,” *Sensors*, vol. 11, no. 3, pp. 2751–2772, 2011.

[9] P. Tang, B. Akinci, and D. Huber, “Quantification of edge loss of laser scanned data at spatial discontinuities,” *Automation in Construction*, vol. 18, no. 8, pp. 1070–1083, 2009.

[10] D. Anderson, H. Herman, and A. Kelly, “Experimental characterization of commercial flash ladar devices,” in *International Conference of Sensing and Technology*, vol. 2, 2005.

[11] A. Alhashimi, D. Varagnolo, and T. Gustafsson, “Joint temperature-lasing mode compensation for time-of-flight lidar sensors,” *Sensors*, vol. 15, no. 12, pp. 31 205–31 223, 2015.

[12] M. Dekan, F. Duchoň, A. Babinec, P. Hubinskỳ, M. Kajan, and M. Szabová, “Versatile approach to probabilistic modeling of hokuyo utm-30lx,” *IEEE Sensors Journal*, vol. 16, no. 6, pp. 1814–1828, 2016.

[13] K. Konolige, J. Augenbraun, N. Donaldson, C. Fiebig, and P. Shah, “A low-cost laser distance sensor,” in *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*. IEEE, 2008, pp. 3002–3008.

[14] J. Lima, J. Gonçalves, and P. J. Costa, “Modeling of a low cost laser scanner sensor,” in *CONTROLO’2014–Proceedings of the 11th Portuguese Conference on Automatic Control*. Springer, 2015, pp. 697–705.

[15] D. Campos, J. Santos, J. Gonçalves, and P. Costa, “Modeling and simulation of a hacked neato xv-11 laser scanner,” in *Robot 2015: Second Iberian Robotics Conference*. Springer, 2016, pp. 425–436.

[16] A. Alhashimi, D. Varagnolo, and T. Gustafsson, “Statistical modeling and calibration of triangulation lidars,” in *ICINCO*, 2016.

[17] C.-Y. Chen and H.-J. Chien, “On-site sensor recalibration of a spinning multi-beam lidar system using automatically-detected planar targets,” *Sensors*, vol. 12, no. 10, pp. 13 736–13 752, 2012.

[18] M. Gordon and J. Meidow, “Calibration of a multi-beam laser system by using a tls-generated reference,” *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences II-5 W*, vol. 2, pp. 85–90, 2013.

[19] A. Burguera, Y. González, and G. Oliver, “Sonar sensor models and their application to mobile robot localization,” *Sensors*, vol. 9, no. 12, pp. 10 217–10 243, 2009.

[20] S. Noykov and C. Roumenin, “Calibration and interface of a polaroid ultrasonic sensor for mobile robots,” *Sensors and Actuators A: Physical*, vol. 135, no. 1, pp. 169–178, 2007.

[21] C. U. Dogruer, “Online identification of odometer parameters of a mobile robot,” in *International Joint Conference SOCO14-CISIS14-ICEUTE14*. Springer, 2014, pp. 195–206.

[22] A. Kelly, “Linearized error propagation in odometry,” *The International Journal of Robotics Research*, vol. 23, no. 2, pp. 179–218, 2004.

[23] J. Kelly and G. S. Sukhatme, “A general framework for temporal calibration of multiple proprioceptive and exteroceptive sensors,” in *Experimental Robotics*. Springer, 2014, pp. 195–209.

[24] 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.

[25] L. Kleeman, “Advanced sonar and odometry error modeling for simultaneous localisation and map building,” in *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, vol. 1. IEEE, 2003, pp. 699–704.

[26] S. Gibson and B. Ninness, “Robust maximum-likelihood estimation of multivariable dynamic systems,” *Automatica*, vol. 41, no. 10, pp. 1667–1682, 2005.

[27] R. H. Shumway and D. S. Stoffer, “An approach to time series smoothing and forecasting using the em algorithm,” *Journal of time series analysis*, vol. 3, no. 4, pp. 253–264, 1982.

[28] ——, *Time series analysis and its applications: with R examples*. Springer Science & Business Media, 2006.