

# Two-Phase Online Calibration for Infrared-based Inter-Robot Positioning Modules

Sven Gowal, Amanda Prorok, Alcherio Martinoli

**Abstract**—Multi-robot systems can solve complex tasks that require the coordination of the team-member positions with respect to each other. While the development of ad-hoc relative positioning platforms embedding cheap off-the-shelf components is a practical choice, it leads not only to differences between the platforms themselves, but also to a high sensitivity to external factors. In this paper, we present a novel lightweight online calibration method composed of two phases, capable of running on miniature robots with limited computational capabilities. Furthermore, by exploiting a Gaussian process regression in its second phase, the proposed calibration approach is able to capture deviations from an assumed underlying physical model. We compare the performance of our approach with the theoretical Cramér-Rao lower bound and test its efficiency on real robots equipped with range and bearing modules.

## I. INTRODUCTION

Recent years have seen the development of infrared range and bearing sensors for commercially available robots [12], open robot architectures [4] or customized robots [13]. The main usage of these hardware modules is to acquire the relative position of nearby robots (teammates) to accomplish collaborative tasks. Unfortunately, these modules have several disadvantages: (i) infrared receivers have variable sensitivity, (ii) infrared emitters have variable emitting powers, and (iii) environmental conditions (i.e. background noise) have a high impact on their performance. Moreover, these artifacts, even if mitigated by a careful and manual component selection, are exacerbated when using cheap off-the-shelf components, and thus one can often not fully rely on a priori physical models to calibrate such sensors. Nevertheless, appropriate software can be developed to remedy these issues. In this context, we propose a two-phase calibration method which explicitly addresses this problematic in real-time for a multi-robot system. The first phase of our method consists in applying a standard online stochastic-gradient method, which results in a fairly good estimation of the parameters of the underlying physical model. The second phase of our method consists in refining this first estimate by adapting the underlying physical model to the present observations. Thus the combination of these two phases not only resolves issues (i) and (iii), but also implicitly adapts the a priori underlying physical model. Finally, by utilizing the results

Sven Gowal, Amanda Prorok and Alcherio Martinoli are with the Distributed Intelligent Systems and Algorithms Laboratory, School of Architecture, Civil and Environmental Engineering, Ecole Polytechnique Fédérale de Lausanne. [svenadrian.gowal@epfl.ch](mailto:svenadrian.gowal@epfl.ch), [amanda.prorok@epfl.ch](mailto:amanda.prorok@epfl.ch), [alcherio.martinoli@epfl.ch](mailto:alcherio.martinoli@epfl.ch). This work was partially sponsored by the National Center of Competence in Research on Mobile Information and Communication Systems (NCCR-MICS), a center supported by the Swiss National Science Foundation under grant number 51NF40-111400.

of the first calibration phase and reducing the search space of the second, computationally more complex phase, the methodology is purposefully designed to run on resource-constrained robots.

## A. Related Work

The importance of sensor calibration is widely acknowledged in the robotics community. Many works can be found on camera and laser range scanners calibration [5, 9, 16]. However, to the best of our knowledge, calibration of infrared-based range and bearing modules has not been considered so far. Yet, most recent work using such platforms assume that the relation between the measured signal strength and the distance to the emitters is known [4, 12, 13]. This relation is often established manually by placing the platforms at specific positions and measuring the strength of the modulated infrared light received. Since this procedure is cumbersome and impossible for certain applications, we inspire ourselves from the same motivations that have fostered the literature covering online self-calibration of odometry parameters [8, 14]. Indeed, self-calibration from on-board localization capabilities or even auto-calibration given external ground-truth measurements offer an interesting alternative to manual calibration. Additionally, since the infrared lighting may constantly fluctuate, the ability to calibrate online and in real-time rather than a single time offline is a must.

## B. Problem Statement

Although the calibration strategy presented in this paper is generalizable to a wide range of online calibration problems, we have chosen the case-study of infrared-based range and bearing sensors. Let us assume that we have a team of robots each equipped with a range and bearing module. An example of such a platform is shown on Figure 1(a). The shown module has sixteen evenly-spaced infrared Light Emitting Diodes (LEDs). As there are enough infrared LEDs to emit omni-directionally with respect to the robot carrying this platform (and for the sake of simplicity), we will consider the robot itself as an **emitter**. The shown platform also contains eight evenly-spaced infrared **receivers** capable of measuring the Received Signal Strength Indicator (RSSI) of the incoming infrared light. The measured raw dimensionless RSSI is proportional to the actual magnitude of the modulated light emitted. This platform is also able to broadcast low bit rate communication packets, thus enabling association of RSSI values with a specific emitter. Furthermore, we assume that, at arbitrary moments during their mission, the robots are able to determine their actual (even if approximate)

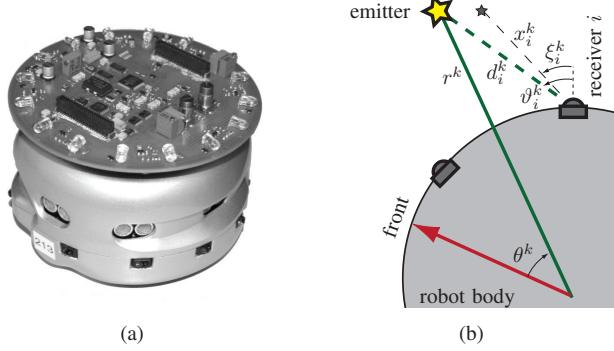


Fig. 1: (a) A Khepera III robot with a range and bearing board attached. This range and bearing module features sixteen infrared light emitting diodes and eight infrared light sensors. (b) At each time step  $k$ , a robot receives infrared light coming from an emitter located at a range  $r^k$  and a bearing  $\theta^k$  from it. In a training phase, the receiving robot obtains an approximated distances  $x_i^k$  and angle of incidence  $\xi_i^k$  between the emitter and each receiver. In an evaluation phase, it estimates the range  $r^k$  and bearing  $\theta^k$  using the received signal strength indicator from its infrared light sensors.

position in a common frame, either through an external system (e.g., overhead camera) or using their own sensing capabilities (e.g., odometry, ultrasound).

Formally, each robot has  $M$  receivers, and at each time step  $k$ , the  $i$ -th receiver can analyze the incoming infrared light and measure a corresponding RSSI  $y_i^k$ . The measured RSSIs ( $y_1^k, \dots, y_M^k$ ) are then associated to a specific emitter (at each time step, one and only one emitter is associated to the RSSIs). When localization information is available, actual distances  $x_i^k$  and angles of incidence  $\xi_i^k$  between the detected emitter and each receiver  $i$  can be determined. Note that  $x_i^k$  and  $\xi_i^k$  do not need to be the true distances  $d_i^k$  and angles  $\vartheta_i^k$ . When localization information is not available, robots need to correctly estimate the range  $r^k$  and bearing  $\theta^k$  to neighboring teammates given the currently measured RSSIs  $y_1^k, \dots, y_M^k$ . In other words, the training data  $(y_1^k, \dots, y_M^k, x_1^k, \dots, x_M^k)$  may only be available at some unknown point in time, whereas the evaluation of  $d_i^k$  may be required at every point in time. Figure 1(b) shows a schematic illustration of the system.

This paper is organized as follows. In Section II, we introduce the underlying physical model relating the distance and the RSSI measurement and elaborate the individual calibration methods constituting our approach. We conclude the section by combining the introduced methods in a two-phase range calibration procedure. Section III extends the latter for range and bearing. Finally, our approach is validated on real robots in real-time, and results are shown in a comprehensive comparison with standard offline calibration methods.

## II. RANGE ESTIMATION

Throughout this section, we will concentrate on the calibration of a single receiver in terms of range estimation (i.e. all  $\vartheta_i^k$  and  $\xi_i^k$  are equal to zero). For brevity, we will omit the receiver index (e.g.  $x_i^k$  becomes  $x^k$ ).

### A. Preliminaries

1) *Underlying Physical Model:* Detecting the range between two robots equipped with range and bearing modules requires a comparison between the strength of the received infrared light and the one of the emitted light. It is clear that an equation relating the emitted signal strength with the received strength depends on the distance traveled by the light. Here, inspired by the Beer-Lambert law [6], we assume an exponential decay such that:

$$y^k = \alpha + \beta \cdot e^{-\gamma d^k} + \epsilon_y^k \quad \text{with } \epsilon_y^k \sim \mathcal{N}(0, \sigma_y^2) \quad (1)$$

where  $y^k$  is the RSSI measured by a specific receiver at the  $k$ -th time step,  $\alpha$  is the receiver offset and/or background noise,  $\beta$  incorporates the receiver sensitivity on one hand and the emitted strength on the other (which can vary depending on the hardware),  $\gamma$  is the absorption coefficient which can vary slightly due to environmental conditions (humidity, etc.) and  $d^k$  is the distance between the emitter and receiver. Finally,  $\epsilon_y^k$  is a white noise component sampled from a normal distribution with standard deviation  $\sigma_y$  at time  $k$ . Note that the decay of the signal strength with respect to the distance, given here by the term  $e^{-\gamma d^k}$ , may also be given by a lookup table as in [12].

2) *Observation Model:* As mentioned in Section I-B, an actual distance  $x^k$  between the emitter and the receiver is arbitrarily available and follows the equation:

$$x^k = d^k + \epsilon_x^k \quad \text{with } \epsilon_x^k \sim \mathcal{N}(0, \sigma_x^2) \quad (2)$$

where  $d^k$  is the true distance and  $\epsilon_x^k$  is a white noise component sampled from a normal distribution with standard deviation  $\sigma_x$  at time  $k$ . Importantly,  $x^k$  is an unbiased estimator for  $d^k$  (i.e. its expectancy is  $d^k$ ). The use of a biased estimator may lead to an unsatisfactory calibration process (explained in Section II-C).

### B. Calibration Methods

This section presents the two main components of our two-phase method by detailing two standard calibration algorithms capable of estimating the parameters of the physical model in Equation 1. For both calibration methods, we assume that during a training phase,  $N$  data points containing the approximate range values  $x^k$  and their corresponding RSSI values  $y^k$  are gathered. We will denote by  $\mathbf{x} = [x^1, \dots, x^N]^T$  the vector aggregating all ranges and by  $\mathbf{y} = [y^1, \dots, y^N]^T$  the vector of corresponding RSSI values.

1) *Least Squares:* Our objective is to adjust the parameters  $\Theta_m = [\alpha, \beta, \gamma]^T$  assuming the model below:

$$\begin{cases} \mathbf{x} = \mathbf{d} + \epsilon_{\mathbf{x}} \\ \mathbf{y} = \alpha + \beta \cdot \mathbf{f}(\mathbf{d}) + \epsilon_{\mathbf{y}} \end{cases} \quad (3)$$

where  $\mathbf{d}$  is a vector of unknown values and  $\mathbf{f}(\mathbf{d})$  is a vectorial function and is equal to  $[e^{-\gamma d^1}, \dots, e^{-\gamma d^N}]^T$ . The values  $\epsilon_{\mathbf{x}}$  and  $\epsilon_{\mathbf{y}}$  are two vectors of independent zero-mean random variables with unknown variances  $\sigma_x^2$  and  $\sigma_y^2$ , respectively. Given this model and since we do not have full knowledge on the conditional probability density function

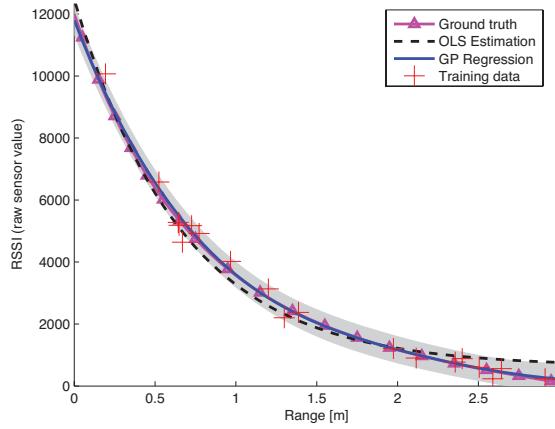


Fig. 2: Example of a Gaussian process regression (GPR) on 20 data points. We observe that the GPR estimates better the true function ( $\alpha + \beta e^{-\gamma d} - Kd$ ) than the ordinary least square (OLS) regression when the model assumed by both the OLS and the GPR is not the true model (here, only  $\alpha + \beta e^{-\gamma d}$ ). The gray area denotes the  $\pm 1.96\sigma$  boundary found by the GPR. We have set  $\alpha = 200$ ,  $\beta = 10000$ ,  $\gamma = 1.5$  and  $K = 650$ .

$P(d|x, y)$  (in particular, we do not know  $\sigma_x$  and  $\sigma_y$ ), we can resort to a least-squares estimator. Obenchain [10] showed that least squares estimation always provides estimates with a minimum Mean Square Error (MSE) risk (even when the assumed model is wrong). Hence our goal will be to minimize the sum of the squared residuals

$$\hat{\Theta}_m = \underset{\Theta_m = [\alpha, \beta, \gamma]^T}{\operatorname{argmin}} \left( S = \sum_{i=1}^N (\alpha + \beta \cdot e^{-\gamma x^i} - y^i)^2 \right) \quad (4)$$

such that  $\alpha, \beta, \gamma \geq 0$ .

There is no closed-form solution to the above optimization problem, but we can use the derivatives of  $S$  with respect to the parameters to perform gradient descent and find an approximation  $\hat{\Theta}_m$  of the optimal parameters. If the problem is assumed to be linear (i.e. by fixing  $\gamma$  which varies only slightly in time), the problem takes the form of an Ordinary Least Squares (OLS) estimation problem and the closed-form solution for  $\hat{\Theta}_\mu = [\alpha, \beta]^T$  is then

$$\hat{\Theta}_\mu = (X^\top X)^{-1} X^\top \mathbf{y} \quad (5)$$

where  $X = [\mathbf{1}, \mathbf{f}(\mathbf{x})]$  with  $\mathbf{1} = [1, 1, \dots, 1]^\top$ . Note that Equation 5 is relatively cheap to compute as the matrix to be inverted has a size equal to the number of parameters to estimated.

Unfortunately, the use of least-squares estimation is constrained to the model proposed in Equation 3. Firstly, if we fix  $\gamma$  to avoid a costly and maybe suboptimal numerical optimization procedure, it might still be that  $\gamma$  is different from the one provided. Secondly, the sensor response might be nonlinear, thus invalidating the original model. It is important to note that Equation 1, although being a fairly good approximation of reality, does not match it perfectly. Hence, it is useful to devise a regression procedure that can approximate reality rather than a theoretical model.

2) *Gaussian Processes*: Due to the above issues, we resort to another type of regression, namely Gaussian Process Regression (GPR). GPR allows the incorporation of noisy measurements from an unknown process in a probabilistically sound way, thus also enabling the recovery from wrong a priori underlying models. We demonstrate these benefits in Figure 2: we simulated observations employing a purposely altered physical model and then attempted an OLS and GP estimation which were based on the original physical model. We observe that the GPR yields a better estimate of the altered model than the OLS estimation.

For our specific case-study, a GP that enables us to predict the range  $x$  given an RSSI  $y$  is described as a distribution over functions so that the mean function  $\mu(y)$  and covariance function  $k(y_1, y_2)$  of a process are [1]

$$\mu(y) = \mathbb{E}[f(y)] \quad (6)$$

$$k(y_1, y_2) = \mathbb{E}[(f(y_1) - \mu(y_1))(f(y_2) - \mu(y_2))] \quad (7)$$

and the Gaussian process is then

$$f(y) \sim \mathcal{GP}(\mu(y), k(y_1, y_2)). \quad (8)$$

Given a set of training data points  $\mathbf{x}$  and  $\mathbf{y}$ , a GPR can predict the value  $x_*$  of a new data point  $y_*$ :

$$x_* = \mu(y_*) + K(y_*, \mathbf{y}) \cdot K(\mathbf{y}, \mathbf{y})^{-1} \cdot (\mathbf{x} - \mu(\mathbf{y})) \quad (9)$$

where  $K(\mathbf{y}_1, \mathbf{y}_2)$  is the covariance matrix relating the  $n$  points of  $\mathbf{y}_1$  with the  $m$  points of  $\mathbf{y}_2$ :

$$K(\mathbf{y}_1, \mathbf{y}_2) = \begin{bmatrix} k(y_1^1, y_2^1) & \cdots & k(y_1^1, y_2^m) \\ \vdots & \ddots & \vdots \\ k(y_1^n, y_2^1) & \cdots & k(y_1^n, y_2^m) \end{bmatrix}. \quad (10)$$

The GPR also yields the uncertainty about the new point  $x_*$ :

$$\sigma_*^2 = k(y_*, y_*) - K(y_*, \mathbf{y}) \cdot K(\mathbf{y}, \mathbf{y})^{-1} \cdot K(\mathbf{y}, y_*). \quad (11)$$

The mean function  $\mu(y)$  and covariance function  $k(y_1, y_2)$  can vary and may depend on additional parameters (called hyper-parameters of the Gaussian process). For our case-study we have

$$\mu(y) = -\frac{1}{\gamma} \log \left( \frac{y - \alpha}{\beta} \right) \quad (12)$$

$$k(y_1, y_2) = \sigma_f^2 e^{-\frac{\|y_1 - y_2\|^2}{2\lambda^2}} + \sigma_n^2 \cdot \delta(y_1 - y_2) \quad (13)$$

where  $\sigma_f$  models the amplitude of the process variance,  $\lambda$  the length scale of process variation and  $\sigma_n$  the observation noise. The value  $\delta(y)$  is 1 when  $y = 0$  and 0 otherwise. The mean function reflects the empirical model for light propagation of Equation 1 and the covariance function is the standard exponential covariance function. Note that all parameters  $\Theta_m = [\alpha, \beta, \gamma]^T$  and  $\Theta_k = [\sigma_f, \lambda, \sigma_n]^T$  control the shape of each function and thus affect the behavior of the GP. In order to optimize and find the hyper-parameters it is useful to compute the marginal log-likelihood of the model with respect to the training data

$$\log P(\mathbf{x}|\mathbf{y}, \Theta_m, \Theta_k) = -\frac{1}{2} (\mathbf{x} - \mu(\mathbf{y}))^\top K(\mathbf{y}, \mathbf{y})^{-1} (\mathbf{x} - \mu(\mathbf{y})) - \frac{1}{2} \log |K(\mathbf{y}, \mathbf{y})| - \frac{N}{2} \log 2\pi. \quad (14)$$

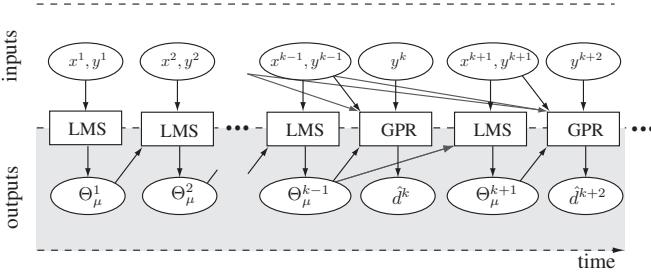


Fig. 3: Example of a two-phase calibration procedure. Training and estimation are interleaved depending on whether ground-truth is available.

Finally, one can derive the log-likelihood with respect to the different hyper-parameters and perform gradient ascent to maximize it.

In conclusion, the advantage of a GPR is that it can account for characteristics present in the data that are not visible in the given explicit prior on the mean function. Hence, it is able to find the nonlinearities present at the level of the receiver sensor. On the other hand, finding optimal values for  $\Theta_m$  or  $\Theta_k$  is cumbersome as it requires a numerical optimization procedure that needs to invert the covariance matrix of size  $N$ . Note that in [3], authors successfully use GPR to estimate the position of a radio emitter with a *Scarab* robot. The real-time requirements of their approach are maintained thanks to a 2.5 GHz processor running a well-optimized available C++ library capable of handling a hundreds of training data points. In our case-study, we use the *Khepera III* robot which has a 400 MHz processor and not even enough disk space to hold the C++ standard library.

### C. Two-Phase Online Calibration

This section presents the core of this paper by combining the two previous calibration methods to estimate the parameters of the model in Equation 1 and by refining that model over time.

At each time step  $k$ , a receiver gathers a range  $x^k$  and an RSSI value  $y^k$ . Starting with the model described by Equations 1 and 2, we try to find the optimal values for  $\Theta_\mu = [\alpha, \beta]^\top$  whilst keeping  $\gamma$  fixed. Typically,  $\gamma$  was found to be around 1.5 with only very mild variations throughout an experimental run and across multiple receivers. We transform Equation 5 of the OLS estimation into a standard online stochastic-gradient method, the least-mean-squares (LMS) algorithm [15]. Many variations of the LMS algorithm exist, but we will only consider its standard variant. At each time step  $k$ , we can make a new estimation of  $\Theta_\mu$  using the update

$$\hat{\Theta}_\mu^k = \hat{\Theta}_\mu^{k-1} + \nu \cdot [1, e^{-\gamma x^k}]^\top \cdot (y^k - [1, e^{-\gamma x^k}] \cdot \hat{\Theta}_\mu^{k-1}) \quad (15)$$

where  $\nu$  is a fixed positive step-size (usually small) and  $\hat{\Theta}_\mu^0$  is an initial guess for  $\alpha$  and  $\beta$ . Note that in this case the LMS algorithm iteration requires 5 multiplications and 5 additions, thus being computationally lightweight. The negative aspect of this approach is its lack to adapt quickly to changes and the fact that it is limited to the given model (exponential

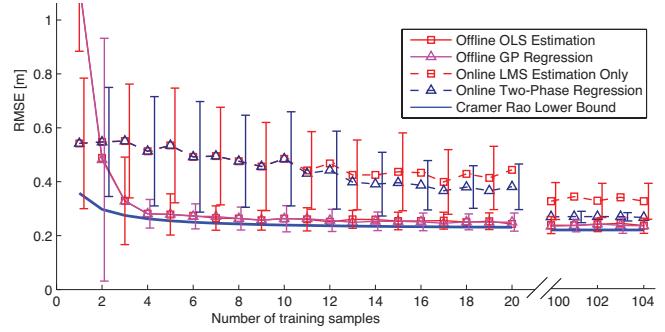


Fig. 4: Simulated performance of the offline and online calibration depending on the number of training data points. We have set  $\alpha = 1000$ ,  $\beta = 10000$ ,  $\gamma = 1.5$ ,  $\sigma_x = 0.05$ m,  $\sigma_y = 200$ ,  $d_{\min} = 0$ m and  $d_{\max} = 3$ m.

decay and fixed decay rate). To overcome these limitations we use the parameters  $\hat{\Theta}_\mu^k$  computed by the LMS, and bootstrap a GPR that uses Equation 12 as its mean function with a fixed  $\gamma$ . The hyper-parameters  $\sigma_f$ ,  $\sigma_n$  and  $\lambda$  are also fixed (as in [3]) and can be easily tuned by hand depending on the given hardware. We will see in Section IV that the performance of the two-phase calibration is not sensitive to moderate changes of either  $\gamma$  or the hyper-parameters.

We use Equation 9 to predict the range  $\hat{d}^k$  corresponding to the new observation  $y^k$  when  $x^k$  is not available:

$$\hat{d}^k = \mu(y^k) + K(y^k, \mathbf{y}) \cdot K(\mathbf{y}, \mathbf{y})^{-1} \cdot (\mathbf{x} - \mu(\mathbf{y})) \quad (16)$$

where  $\mathbf{x}$  and  $\mathbf{y}$  contain the training data gathered so far. Note however, that this step requires the inversion of the covariance matrix. To reduce the complexity of this inversion, we select a handful of training data points, in our case 10, instead of using all available data points. The selection of these data points may be crucial for certain applications and a multitude of selection methods could be envisioned. In this paper, the different selection schemes will not be presented, and we resort to the last 10 available training data points. As we will see later in Section IV, even this simplistic selection performs reasonably well. An example of the two-phase calibration procedure is reported in Figure 3. The figure shows how the two phases of the calibration are interleaved, depending on whether ground-truth information is available.

### D. Comparison with the Cramér-Rao Lower Bound

In order to validate the performance of our algorithm, we compare it with the Cramér-Rao lower bound (CRLB) [2]. The CRLB states that the variance (or mean square error) of an estimator is at least as high as the inverse of the Fisher Information. If an estimator achieves this lower bound, it is said to be efficient. The complete derivation of the CRLB is available online at [http://disalw3.epfl.ch/publications/IROS11\\_cramerrao.pdf](http://disalw3.epfl.ch/publications/IROS11_cramerrao.pdf).

Figure 4 shows the average Root MSE (RMSE) of an offline OLS estimation, an offline GPR, an online LMS algorithm and our online two-phase calibration method with respect to the number of training data points observed.

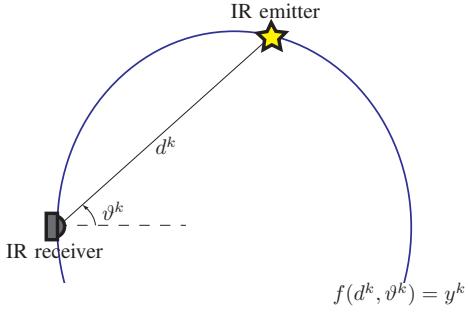


Fig. 5: An infrared receiver sensing a RSSI  $y^k$  from an omni-directional infrared emitter.  $h(d^k, \vartheta^k) = y^k$  is the isoline of all position where the emitter could be such that the receiver senses an RSSI  $y^k$ .

Observations are generated from our underlying physical model from Equations 1 and 2. These performances are compared with the CRLB. We observe that the RMSE of the offline calibrations is close the Cramér Rao lower bound. Online calibrations reach the same performance as that of the offline calibrations but only after more data points have been collected. In particular, we also see that the additional GPR phase is very useful as the error is decreased with respect to an online LMS estimation only, even when the model provided to the LMS estimation is the true model.

### III. RANGE AND BEARING ESTIMATION

In this section, we will extend the general calibration process explained in Section II to account for the incident angle, thus enabling the evaluation of the bearing direction. Indeed, by thus adapting our framework, the following elaborations are specific to the range and bearing sensor case-study.

#### A. Single Receiver Calibration

As already observed in [12], the RSSI  $y^k$  sensed by a given receiver at time  $k$  is highly dependent on the angle of incidence  $\vartheta^k$  of the emitted light. Hence, just like in the previous section, we assume the existence of a relation  $g(d^k) = y^k$ . We extend it by assuming that there exists a function  $h(d^k, \vartheta^k) = y^k$  that relates the incidence angle and distance to the receiver with the RSSI measured at the receiver. We schematize this interaction in Figure 5.

By including the angle of incidence, we would have to calibrate in an augmented dimensional space. This would increase the computational requirements as well as the number of training data points to acquire before good performance is achieved. Fortunately, we conclude from [12] that the relation  $h(d^k, \vartheta^k) = y^k$  can be decomposed into  $g(d^k) \cdot \cos \vartheta^k = y^k$ . This allows us to train the distance response of each receiver on  $y^k / \cos \vartheta^k$  and  $x^k$ , and perform the calibration procedures explained in Section II-B or II-C.

#### B. Computing the Range and the Bearing

In this section, to conclude the procedure, we explain how to find the range and bearing estimate given the above calibration method. As explained in Section I-B, each robot has  $M$  receivers that we are now able to calibrate (using either an offline or online procedure). Hence, each receiver  $i$

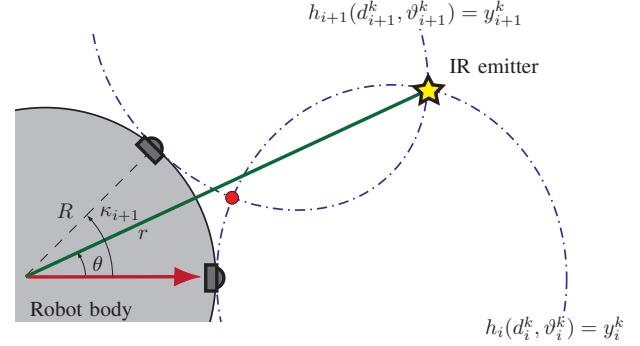


Fig. 6: Two infrared receivers mounted on a robot body sensing RSSI  $y_i^k$  and  $y_{i+1}^k$  from an omni-directional infra-red emitter. The emitter should line at the intersection between the two isoline  $h_i(d_i^k, \vartheta_i^k) = y_i^k$  and  $h_{i+1}(d_{i+1}^k, \vartheta_{i+1}^k) = y_{i+1}^k$ . Note how in this example one of the intersections can safely be eliminated due the geometrical constraints (i.e. radius of a robot).

has a known (estimated) response  $\hat{h}_i(d, \vartheta) = \hat{g}_i(d) \cos \vartheta = y_i^k$  relating the distance and angle of incidence to the receiver and its measured RSSI value. Given the position of each receiver with respect to the receiving robot we can then compute the intersections of the isolines  $\hat{h}_i(d, \vartheta) = y_i^k$ . Figure 6 shows the intersection of two isolines generated from two contiguous receivers. Note however, that a better estimate of the range and bearing  $[r, \theta]^T$  can be obtained by taking the three receivers  $i-1, i, i+1$  measuring the highest sum of RSSI values and minimizing:

$$[\hat{r}, \hat{\theta}]^T = \underset{r, \theta}{\operatorname{argmin}} \sum_{j=-1}^1 \left( \hat{g}_{i+j}^{-1} \left( \frac{y_i}{\cos \vartheta_{i+j}} \right) - d_{i+j} \right)^2 \quad (17)$$

where  $\hat{g}_i^{-1}(y)$  is given by Equation 16. If each receiver  $i$  is positioned around the robot at a distance  $R$  from the center and a bearing  $\kappa_i$  (as shown on Figure 6), we have:

$$d_i = \sqrt{r^2 - 2Rr \cos(\theta - \kappa_i) + R^2} \quad (18)$$

$$\cos \vartheta_i = \frac{r \cos(\theta - \kappa_i) - R}{d_i} \quad (19)$$

The choice of this method to compute the range and bearing estimate is arbitrary and many other methods are possible. In particular, by assuming that the range is much greater than the robot radius, simple trigonometric relations can be found [12].

### IV. EXPERIMENTS

Experiments were performed using Khepera III robots [11], developed by K-Team in collaboration with the Distributed Intelligent and Algorithms Laboratory (DISAL) at EPFL. This robot has a diameter of 12 cm, making it appropriate for multi-robot indoor experiments. As shown on Figure 1(a), we equip each robot with a range and bearing module. We perform two sets of experiments, one set with two robots and another one with four robots. For each set, five runs of five minutes each are made, making a total of 25 minutes worth of evaluation. Robots move randomly in a  $3 \times 3 \text{m}^2$  arena. Their ground truth position and orientation is monitored with SwisTrack [7], an

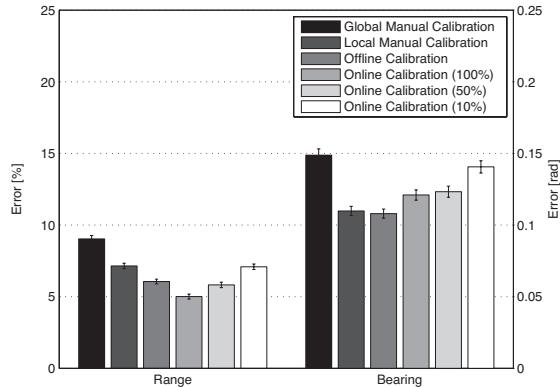


Fig. 7: Average error in the estimation of the range and bearing with 2 robots in the arena. The error bars define the 95% confidence intervals.

open-source tracking software. We evaluate the performance of four calibration methods:

- 1) *Global Manual Calibration*: In a preliminary step, robots are placed pairwise in the arena, and moved manually to different known locations, to gather 50 data points per receiver over a range of distances from 0.3 to 2 meters. Once all data points for all receivers of one robot have been gathered, we perform a least mean square regression on the model in Equation 1 to estimate the best overall  $\alpha$ ,  $\beta$  and  $\gamma$  values for all receivers of each single robot. During the experimental runs, if a robot detects a neighboring robot, it will estimate the range and bearing to the detected robot. Simultaneously, we calculate the error with the true range and bearing values (measured by the tracking system).
- 2) *Local Manual Calibration*: Same as above, except that  $\alpha$ ,  $\beta$  and  $\gamma$  values are estimated for each receiver (instead of for each robot).
- 3) *Offline Calibration*: In a preliminary step, either two or four robots (depending on the experimental set) are placed in the arena. The robots move randomly for ten minutes and training data points are gathered using the range and bearing module in conjunction with the camera tracking system. For each receiver  $i$ , a GPR is performed to estimate its response  $\hat{g}_i^{-1}$ . During the experimental runs, the performance is evaluated in the same manner as above.
- 4) *Online Calibration*: No preliminary step is performed. During the experimental runs, if a robot detects a neighboring robot and measures the corresponding RSSI values it will estimate the range and bearing to the detected robot. Only after the estimation of these values, it will receive in 100%, 50% or 10% of the cases the true range and bearing from the tracking system in order to improve the calibration of its receivers.

Each robot estimates the range and bearing to other robots using all aforementioned methods at the same time on the same RSSI values. At the end of all runs, the average RMSE between the true and estimated range and bearing is computed for each method.

Figure 7 shows the average performance for the first set of experiments conducted with two robots. For the range, the error is presented in percent, and for the bearing, in radians.

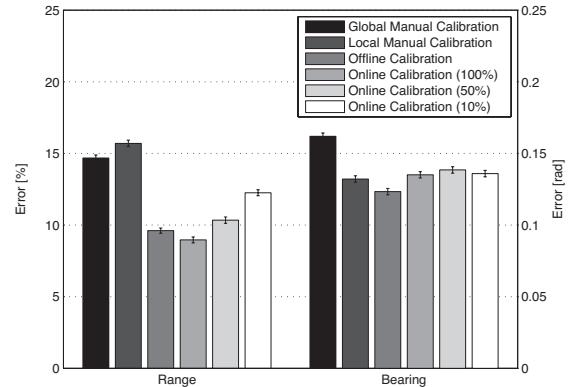


Fig. 8: Average error in the estimation of the range and bearing with 4 robots in the arena. The error bars define the 95% confidence intervals.

As expected, we observe that the global calibration performs the worst. The local and offline calibration methods perform quite similarly showing only a slight benefit of using the GPR calibration. The online calibration performs the best when all data points are used and this is also expected since it can account for changing conditions (e.g. presence of a corner or a wall that reflects additional infrared light). When using half of the data points it reaches the same performance as that of the offline calibration and finally degrades to the performance of the local calibration when using only a tenth of the data points. This is quite exceptional since a tenth of the data points represents about 50 points for all receivers per run (instead of 50 points per receiver per run).

Figure 8 shows the average performance for the second set of experiments conducted with four robots. The performances have all degraded, especially the local calibration which now has a similar performance as that of the global calibration. In fact, when calibrating the robots in a pairwise manner, they are exposed to different external influences than they actually undergo during experimentation. By adding two robots in the experimental arena, the environmental conditions drastically change, worsening results for the manual calibration methods. The offline calibration method again shows very good results as it was performed with four robots in identical conditions. On the other hand, if other environmental factors would have changed, its performance would have degraded significantly. Finally, the two-phase online calibration shows very good results overall.

These results are very symptomatic of the constant environmental changes happening in real-time, even in a highly controlled experimental arena. Performing a calibration on the actual test-bed where the robots will be evaluated improves the performance of the system. Unfortunately, this is rarely possible in reality. Hence, the advantages of an online calibration procedure are obvious and as the results show, an online method is able to perform better than manual offline calibration methods, even when using only a fraction of the number of training data points.

Finally, we perform a sensitivity analysis by varying each of the GPR hyper-parameters  $\sigma_n$ ,  $\sigma_f$  and  $\lambda$  as well as  $\gamma$ . We re-compute the range and bearing errors with the data

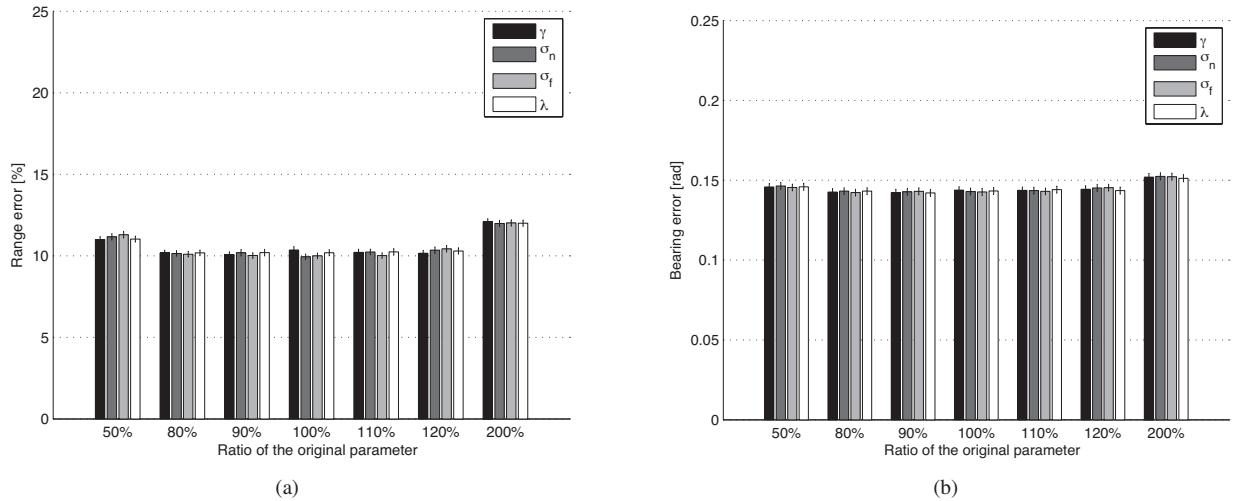


Fig. 9: Sensitivity of the average (a) range and (b) bearing error with 4 robots in the arena using the online calibration at 50%. The GPR hyper-parameters as well as  $\gamma$  are varied in turn between 50% and 200% of their original values. The error bars define the 95% confidence intervals.

gathered experimentally. Each parameters is modified in turn, while the others remain unchanged. Figure 9 reports the resulting errors. We observe that the performance of the two-phase calibration method is hardly affected by moderate changes of the parameters. We, thus, conclude that the parameters can be tuned by hand without compromising performance.

## V. CONCLUSION

A standalone least-mean-squares algorithm is not able to capture even the smallest deviation from the assumed underlying physical model. In this paper, we have proposed a novel online two-phase calibration technique based on the sequential combination of a least-mean-squares algorithm and a Gaussian process regression. We have shown how, by using the estimation results of the least-mean-squares algorithm to define the mean function of the Gaussian process regression, we are able to significantly reduce its search space and thus its computational complexity. Hence, the two-phase method is ideal for miniature or resource-bounded robots. We compared our approach with the Cramér-Rao lower bound showing that its estimation of the true parameters is efficient. Finally, we performed real robots experiments which show improved performance when compared with standard offline calibration procedures.

## REFERENCES

- [1] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006. ISBN 978-0387310732.
- [2] H. Cramér. *Mathematical Methods of Statistics*. Princeton University Press, 1946. ISBN 978-0691005478.
- [3] J. Fink and V. Kumar. Online methods for radio signal mapping with mobile robots. In *IEEE International Conference on Robotics and Automation*, pages 1940–1945, 2010.
- [4] A. Gutierrez, A. Campo, M. Dorigo, J. Donate, F. Monasterio-Huelin, and L. Magdalena. Open e-puck range & bearing miniaturized board for local communication in swarm robotics. In *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, pages 3111–3116, 2009.
- [5] J. Heikkila and O. Silven. A four-step camera calibration procedure with implicit image correction. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1106–1112, 1997.
- [6] J. D. Ingle and S. R. Crouch. *Spectrochemical Analysis*. Prentice Hall, 1988. ISBN 978-0138268763.
- [7] T. Lochmatter, P. Roduit, C. Cianci, N. Correll, J. Jacot, and A. Martinoli. Swistrack - a flexible open source tracking software for multi-agent systems. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4004–4010, 22–26 2008.
- [8] A. Martinelli, N. Tomatis, and R. Siegwart. Simultaneous localization and odometry self calibration for mobile robot. *Autonomous Robots*, 22:75–85, 2007.
- [9] F. Mirzaei and S. Roumeliotis. A kalman filter-based algorithm for imu-camera calibration. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2427–2434, 2007.
- [10] R. L. Obenchain. Residual optimality: Ordinary vs. weighted vs. biased least squares. *Journal of the American Statistical Association*, 70:375–379, 1975.
- [11] A. Prorok, A. Arfire, A. Bahr, J. Farserotu, and A. Martinoli. Indoor navigation research with the Khepera III mobile robot: An experimental baseline with a case-study on ultra-wideband positioning. In *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2010. doi: 10.1109/IPIN.2010.5647880.
- [12] J. Pugh, X. Raemy, C. Favre, R. Falconi, and A. Martinoli. A fast onboard relative positioning module for multirobot systems. *Mechatronics, IEEE/ASME Transactions on*, 14(2):151–162, 2009.
- [13] J. Roberts, T. Stirling, J.-C. Zufferey, and D. Floreano. 2.5d infrared range and bearing system for collective robotics. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 3659–3664, 2009.
- [14] N. Roy and S. Thrun. Online self-calibration for mobile robots. In *IEEE International Conference on Robotics and Automation*, volume 3, pages 2292–2297, 1999.
- [15] A. H. Sayed. *Fundamentals of Adaptive Filtering*. Wiley, 2003. ISBN 978-0471461265.
- [16] Q. Zhang and R. Pless. Extrinsic calibration of a camera and laser range finder (improves camera calibration). In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 3, pages 2301–2306, 2004.