

Simultaneous maximum-likelihood calibration of odometry and sensor parameters

Andrea Censi, Luca Marchionni, Giuseppe Oriolo

Abstract—For a differential-drive mobile robot equipped with an on-board range sensor, there are six parameters to calibrate: three for the odometry (radii and distance between the wheels), and three for the pose of the sensor with respect to the robot frame. This paper describes a method for calibrating all six parameters at the same time, without the need for external sensors or devices. Moreover, it is not necessary to drive the robot along particular trajectories. The available data are the measures of the angular velocities of the wheels and the range sensor readings. The maximum-likelihood calibration solution is found in a closed form.

I. INTRODUCTION

The problem of calibrating the odometry of mobile robots has been studied since the 80's. For the formalization and theoretical analysis of the problem, we refer the reader to [1], [2], [3], [4], [5].

As for practical methods used for calibration, there are two main approaches. The first is to drive the robot along especially crafted trajectories, take some sort of external measurement on its pose, and then estimate/correct its parameters. A popular method of this sort is the UMBmark [2], in which a differential-drive robot is driven along a square path for multiple times, clockwise and anti-clockwise. In the same spirit, in [6] there is a generalization to arbitrary trajectories and different kinematics.

Another approach is called auto-calibration or SLAC (Simultaneous Localization And Calibration), in which calibration is part of the normal activity of the robot and happens without human intervention. A possibility is to use an Extended Kalman Filter (EKF) that estimates both the pose of the robot and the odometry parameters [7], [8], [9]. In this case, there are observability issues to take into account: this is considered in [10] for different combinations of sensors and kinematics. In [11] the authors propose a maximum-likelihood model-free approach that does not seek a physical explanation of the odometry errors. In [12], the authors use a maximum-likelihood method for estimating the odometry parameters of a differential-drive robot, using the absolute observations of an external camera. The method is particularly simple because the problem is completely linear, and therefore can be solved via linear least-squares.

A more difficult problem is to simultaneously estimate the odometry parameters plus some other sensor parameters. A



Fig. 1. A team of five Khepera robots. Each robot has an on-board Hokuyo range-finder and associated battery pack.

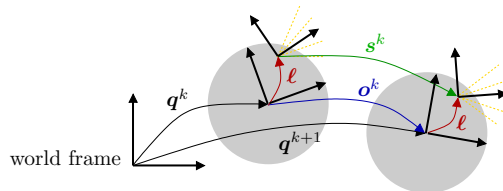


Fig. 2. The robot pose is q^k with respect to the world frame. The sensor pose is ℓ with respect to the robot frame. o^k is the robot displacement between poses, and s^k is the sensor displacement.

first example of this is in [13], where calibration is performed for odometry, a piezo-electric vibrating gyroscope, and a flux-gate magnetic compass. Also in this case there are observability issues: in [14] the authors show that, even assuming the odometry already calibrated, the system is not fully observable if one uses only a bearing sensor.

The present work is the first to consider the simultaneous calibration of odometry and sensor pose (in [14], the sensor pose is calibrated, assuming that odometry parameters are known). Compared to [12], we consider a three-parameters model for a differential-drive robot: we estimate the wheel radii r_L , r_R and the distance b between the wheels. For the sensor, we estimate its pose $\ell = (\ell_x, \ell_y, \ell_\theta)$ with respect to the robot frame. We assume that the sensor is mounted horizontally – an arbitrary orientation would need another two parameters (tilt and yaw). We assume to know the measured wheel velocities, and the output of the range finder. The range readings are passed to a scan-matching algorithm that provides an estimate of the roto-translation of the sensor. We will not discuss the scan-matching algorithm as it is used as a black-box. The method is easy to implement: the robot drives autonomously along arbitrary trajectories, no external measurement is necessary, no nominal parameters must be measured beforehand.

A. Censi is with the Control & Dynamical Systems department, California Institute of Technology, 1200 E. California Blvd., 91125, Pasadena, CA. andrea@cds.caltech.edu

L. Marchionni and G. Oriolo are with the Dipartimento di Informatica e Sistemistica “A. Ruberti”, Università di Roma “La Sapienza”, via Ariosto 25, I-00185 Rome, Italy. {marchionni, oriolo}@dis.uniroma1.it

II. SIMULTANEOUS CALIBRATION OF ROBOT AND SENSOR PARAMETERS

Consider the robot motion along an arbitrary configuration trajectory $\mathbf{q}(t)$. Split the time axis in a number of intervals, each delimited by two range finder readings, and each of (possibly different) duration T^k . In the k -th interval, the robot moves from pose \mathbf{q}^k to pose \mathbf{q}^{k+1} , performing a roto-translation

$$\mathbf{o}^k \triangleq \mathbf{q}^{k+1} \ominus \mathbf{q}^k$$

Such roto-translation depends on the wheel velocities $\omega_L(t)$, $\omega_R(t)$ through the odometry parameters r_L , r_R , b , therefore, at first, \mathbf{o}^k is an unknown quantity. In the same interval, the sensor motion is \mathbf{s}^k in the absolute frame:

$$\mathbf{s}^k \triangleq (\mathbf{q}^{k+1} \oplus \ell) \ominus (\mathbf{q}^k \oplus \ell)$$

Through scan matching, we have an estimate $\hat{\mathbf{s}}^k$ of \mathbf{s}^k .

In the following, for simplicity, we first assume that the wheel velocities are constant over each interval. This leads to simple formulas, an efficient implementation, and good accuracy in our experimental setting. In section II-D we give the equivalent formulas for arbitrary velocities.

A. Kinematic model

Consider the kinematic model of the unicycle:

$$\frac{d}{dt} \begin{bmatrix} q_x(t) \\ q_y(t) \\ q_\theta(t) \end{bmatrix} = \begin{bmatrix} v(t) \cos q_\theta(t) \\ v(t) \sin q_\theta(t) \\ \omega(t) \end{bmatrix} \quad (1)$$

The absolute velocities $v(t)$, $\omega(t)$ depend on the wheel velocities through a linear transformation \mathbf{J} :

$$\begin{bmatrix} v(t) \\ \omega(t) \end{bmatrix} = \mathbf{J} \begin{bmatrix} \omega_L(t) \\ \omega_R(t) \end{bmatrix}$$

Such transformation depends on the odometry parameters:

$$\mathbf{J} = \begin{bmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{bmatrix} = \begin{bmatrix} +r_L/2 & +r_R/2 \\ -r_L/b & +r_R/b \end{bmatrix} \quad (2)$$

In this paper, we always consider each interval separately. Fix $t = 0$ at the beginning of the interval, and let the starting pose be $\mathbf{q}^k = \mathbf{q}(0) = \mathbf{0}$. If we assume that the wheel velocities are constant over one interval ($\omega_L(t) = \omega_L^k$, $\omega_R(t) = \omega_R^k$), then the robot velocities are constant too:

$$v(t) = J_{11}\omega_L^k + J_{12}\omega_R^k = v_0^k \quad (3)$$

$$\omega(t) = J_{21}\omega_L^k + J_{22}\omega_R^k = \omega_0^k \quad (4)$$

and we can compute the roto-translation at the end of the interval $\mathbf{q}^{k+1} = \mathbf{q}(T^k) = \mathbf{o}^k$ in closed form, by integrating exactly the differential equation (1):

$$o_x^k = v_0^k T^k (\sin \omega_0^k T^k) / (\omega_0^k T^k) \quad (5)$$

$$o_y^k = v_0^k T^k (1 - \cos \omega_0^k T^k) / (\omega_0^k T^k) \quad (6)$$

$$o_\theta^k = \omega_0^k T^k \quad (7)$$

If the velocities $\omega(t)$, $v(t)$ are not constant in the interval, the solution can still be easily written, as the differential equations are in cascade form.

SYMBOLS USED IN THIS PAPER

parameters (to be estimated)

r_R, r_L	wheel radii (m)
b	distance between wheels (m)
ℓ	laser pose (w.r.t. robot frame)

measurements

$\hat{\mathbf{s}}^k$	scan matching result
$\hat{\omega}_R, \hat{\omega}_L$	measured wheel velocity (rad/s)

other symbols

\oplus, \ominus As defined in [15], the \oplus operator is the composition of transformations, and \ominus is its inverse.

$$\begin{bmatrix} a_x \\ a_y \\ a_\theta \end{bmatrix} \oplus \begin{bmatrix} b_x \\ b_y \\ b_\theta \end{bmatrix} \triangleq \begin{bmatrix} a_x + b_x \cos a_\theta - b_y \sin a_\theta \\ a_y + b_x \sin a_\theta + b_y \cos a_\theta \\ a_\theta + b_\theta \end{bmatrix}$$

$$\begin{bmatrix} a_x \\ a_y \\ a_\theta \end{bmatrix} \ominus \begin{bmatrix} b_x \\ b_y \\ b_\theta \end{bmatrix} \triangleq \begin{bmatrix} a_x \\ a_y \\ a_\theta \end{bmatrix} \oplus \begin{bmatrix} -b_x \cos b_\theta - b_y \sin b_\theta \\ +b_x \sin b_\theta - b_y \cos b_\theta \\ -b_\theta \end{bmatrix}$$

\mathbf{q}^k	robot pose (w.r.t. world frame)
T^k	time interval between \mathbf{q}^k and \mathbf{q}^{k+1}
\mathbf{o}^k	robot displacement: $\mathbf{o}^k \triangleq \mathbf{q}^{k+1} \ominus \mathbf{q}^k$
\mathbf{s}^k	laser displacement: $\mathbf{s}^k \triangleq (\mathbf{q}^{k+1} \oplus \ell) \ominus (\mathbf{q}^k \oplus \ell)$
$\omega_R(t), \omega_L(t)$	wheel velocities
$v(t), \omega(t)$	linear/angular robot velocities
\mathbf{J}	Linear transformation between wheel and robot velocities.

$$\mathbf{J} = \begin{bmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{bmatrix} = \begin{bmatrix} +r_L/2 & +r_R/2 \\ -r_L/b & +r_R/b \end{bmatrix}$$

B. Estimation of the parameters J_{21} , J_{22}

We first estimate J_{21} and J_{22} by using the scan-matching estimate of the rotation. This part of the method is numerically equivalent to the estimation done in [12]. The difference is that we are considering relative measurements instead of absolute, and very short intervals instead of full trajectories.

If the robot rotates by an angle o_θ^k , also the sensor will rotate by the same angle: $s_\theta^k = o_\theta^k$. Therefore, through the scan matching estimate \hat{s}_θ^k , we can observe $o_\theta^k = \omega_0^k T^k$. From (4) and (7) it follows that:

$$(J_{21}\omega_L + J_{22}\omega_R)T^k = s_\theta^k$$

Rewrite this in matrix form with the measured quantities: \hat{s}_θ^k being the estimate of s_θ^k given by scan-matching, and $\hat{\omega}_L, \hat{\omega}_R$ being the measured wheel velocities:

$$\begin{bmatrix} \hat{\omega}_L^k T^k & \hat{\omega}_R^k T^k \end{bmatrix} \begin{bmatrix} J_{21} \\ J_{22} \end{bmatrix} = \hat{s}_\theta^k + \text{errors} \quad (8)$$

For simplicity, we do not write the details of the ‘errors’ terms. As for the terms $\hat{\omega}_L^k T^k$ and $\hat{\omega}_R^k T^k$, note that they are simply the encoder increments in the k -th interval; their error is at most one encoder tick. The variance of \hat{s}_θ^k can be taken into account in the least-squares solution, if the scan matching algorithm provides such data (most do not; see [16] for computing the covariance of ICP). Stack all the equations

for each interval k :

$$\begin{bmatrix} \vdots & \vdots \\ \hat{\omega}_L^k T^k & \hat{\omega}_R^k T^k \\ \vdots & \vdots \end{bmatrix} \begin{bmatrix} J_{21} \\ J_{22} \end{bmatrix} = \begin{bmatrix} \vdots \\ \hat{s}_\theta^k \\ \vdots \end{bmatrix} + \text{errors}$$

From this, one can find an estimate \hat{J}_{21} , \hat{J}_{22} by linear least-squares. Observability of J_{21} , J_{22} can be checked via the condition number of the constraint system.

Note that, at this point, we only know an estimate of the ratios $J_{21} = -r_L/b$ and $J_{22} = +r_R/b$, but we do not know the value of the single physical parameters r_L , r_R , b . In the following, we will estimate the four parameters b , ℓ_x , ℓ_y , ℓ_θ ; then, the estimate of r_L , r_R will be given by $\hat{r}_L = -\hat{b}\hat{J}_{21}$ and $\hat{r}_R = +\hat{b}\hat{J}_{22}$.

C. Estimation of the other parameters b , ℓ_x , ℓ_y , ℓ_θ

From geometric inspection of Fig. 2, one obtains this constraint for the values of ℓ , \mathbf{o}^k and \mathbf{s}^k :

$$\ell \oplus \mathbf{s}^k = \mathbf{o}^k \oplus \ell \quad (9)$$

Rewrite it in the extended form, for the x , y components:

$$\ell_x + s_x^k \cos \ell_\theta - s_y^k \sin \ell_\theta = o_x^k + \ell_x \cos o_\theta^k - \ell_y \sin o_\theta^k \quad (10)$$

$$\ell_y + s_x^k \sin \ell_\theta + s_y^k \cos \ell_\theta = o_y^k + \ell_x \sin o_\theta^k + \ell_y \cos o_\theta^k \quad (11)$$

At this point, we already have an estimate of o_θ^k , therefore we know an estimate for the terms $\cos o_\theta^k$, $\sin o_\theta^k$. Now we give an expression for o_x^k , o_y^k that depends on the other parameter b . Then, the relations (10)-(11) will contain all four remaining parameters (b , ℓ_x , ℓ_y , ℓ_θ). From (2), note that J_{11} and J_{12} can be written in terms of J_{21} , J_{22} and b :

$$J_{11} = -\frac{b}{2}J_{21} \quad J_{12} = \frac{b}{2}J_{22}$$

Therefore, the linear velocity v_0^k can be estimated as:

$$v_0^k = -\frac{b}{2}J_{21}\omega_L^k + \frac{b}{2}J_{22}\omega_R^k \quad (12)$$

By substituting (12) into (5)-(6), we find that o_x^k , o_y^k are proportional to b through two constants c_x and c_y :

$$o_x^k = c_x b \quad o_y^k = c_y b \quad (13)$$

The constants c_x and c_y depend on quantities that have already been estimated:

$$c_x = \frac{1}{2}T^k(-J_{21}\omega_L^k + J_{22}\omega_R^k) \frac{\sin \omega_0^k T^k}{\omega_0^k T^k}$$

$$c_y = \frac{1}{2}T^k(-J_{21}\omega_L^k + J_{22}\omega_R^k) \frac{1 - \cos \omega_0^k T^k}{\omega_0^k T^k}$$

If the velocities ω_L^k , ω_R^k are not constant, c_x and c_y have a more complicated expression, discussed in section II-D.

Substitute (13) in (10), (11) to obtain the following relationships, which contain the 4 remaining unknowns:

$$\ell_x + s_x^k \cos \ell_\theta - s_y^k \sin \ell_\theta = c_x b + \ell_x \cos o_\theta^k - \ell_y \sin o_\theta^k \quad (14)$$

$$\ell_y + s_x^k \sin \ell_\theta + s_y^k \cos \ell_\theta = c_y b + \ell_x \sin o_\theta^k + \ell_y \cos o_\theta^k \quad (15)$$

Define the vector of the unknown parameters as \mathbf{x} :

$$\mathbf{x} = [b \quad \ell_x \quad \ell_y \quad \cos \ell_\theta \quad \sin \ell_\theta]^T$$

and treat $\cos \ell_\theta$ and $\sin \ell_\theta$ as two separate variables x_4 and x_5 , with the additional constraint $x_4^2 + x_5^2 = 1$. Rewrite (14)-(15) in matrix form $\mathbf{L}_k \mathbf{x} = \mathbf{0}$:

$$\begin{bmatrix} -c_x (1 - \cos \hat{o}_\theta^k) + \sin \hat{o}_\theta^k & +\hat{s}_x^k & -\hat{s}_x^k \\ -c_y & -\sin \hat{o}_\theta^k (1 - \cos \hat{o}_\theta^k) & +\hat{s}_y^k & +\hat{s}_x^k \end{bmatrix} \mathbf{x} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (16)$$

The matrix \mathbf{L}_k depends on known quantities. By defining $\mathbf{M} = \sum_k \mathbf{L}_k^T \mathbf{L}_k$, the least-squares problem to solve is:

$$\min \mathbf{x}^T \mathbf{M} \mathbf{x} \quad (17)$$

$$\text{subject to } x_4^2 + x_5^2 = 1 \quad (18)$$

For this problem, for each solution $\bar{\mathbf{x}}$, also $-\bar{\mathbf{x}}$ is a solution. As an additional constraint, we impose that the distance b must be positive:

$$x_1 > 0 \quad (19)$$

D. If velocities $\omega_R(t)$ and $\omega_L(t)$ are not constant

If $\omega_R(t)$ and $\omega_L(t)$ are not constant, change (8) to

$$\begin{bmatrix} \int_0^{T^k} \hat{\omega}_L(t) dt & \int_0^{T^k} \hat{\omega}_R(t) dt \end{bmatrix} \begin{bmatrix} J_{21} \\ J_{22} \end{bmatrix} = \hat{s}_\theta^k + \text{errors}$$

Once \hat{J}_{21} and \hat{J}_{22} have been estimated, one can estimate the evolution of $o_\theta^k(t)$ as:

$$\hat{o}_\theta^k(t) = \int_0^t \left(\hat{J}_{21} \hat{\omega}_L(\tau) + \hat{J}_{22} \hat{\omega}_R(\tau) \right) d\tau \quad (20)$$

As before, $v(t)$ is known up to a constant (the distance b):

$$\hat{v}(t) = b \left(-\frac{1}{2} \hat{J}_{21} \hat{\omega}_L(t) + \frac{1}{2} \hat{J}_{22} \hat{\omega}_R(t) \right)$$

Therefore, the value of \hat{o}_x^k , \hat{o}_y^k at time T^k is given by

$$o_x^k = c_x b \quad o_y^k = c_y b$$

where the two constants c_x and c_y can be estimated by:

$$c_x = \int_0^{T^k} \left(-\frac{1}{2} \hat{J}_{21} \hat{\omega}_L(t) + \frac{1}{2} \hat{J}_{22} \hat{\omega}_R(t) \right) \cos \hat{o}_\theta^k(t) dt$$

$$c_y = \int_0^{T^k} \left(-\frac{1}{2} \hat{J}_{21} \hat{\omega}_L(t) + \frac{1}{2} \hat{J}_{22} \hat{\omega}_R(t) \right) \sin \hat{o}_\theta^k(t) dt$$

From this point on, the computation is the same as in the previous case.

E. Solving the constrained least-squares problem

We solve the constrained least-squares problem by using Lagrange's multipliers. The constraint (18) is written in matrix form as

$$\mathbf{x}^T \mathbf{W} \mathbf{x} = 1, \quad \text{with } \mathbf{W} \triangleq \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 2} \\ \mathbf{0}_{2 \times 3} & \mathbf{I}_{2 \times 2} \end{bmatrix} \quad (21)$$

Constraint (19) is ignored for now: we will just flip the solution if \hat{x}_1 happens to be negative.

Using Lagrange's multipliers method, a necessary condition for optimality is that

$$(\mathbf{M} + \lambda \mathbf{W}) \mathbf{x} = \mathbf{0}$$

One needs to find a λ such that the matrix $(\mathbf{M} + \lambda \mathbf{W})$ is singular, and then choose the solution \mathbf{x} in the kernel of such matrix. The value of λ can be found by solving the equation

$$\det(\mathbf{M} + \lambda \mathbf{W}) = 0$$

In our problem, the matrix \mathbf{M} has a particular structure, with some zeros and repeated entries:

$$\mathbf{M} = \begin{bmatrix} m_{11} & 0 & m_{13} & m_{14} & m_{15} \\ * & m_{22} & 0 & m_{35} & -m_{34} \\ * & * & m_{22} & m_{34} & m_{35} \\ * & * & * & m_{44} & 0 \\ * & * & * & * & m_{44} \end{bmatrix}$$

For this matrix, the determinant $\det(\mathbf{M} + \lambda \mathbf{W})$ is a second-order polynomial ($a\lambda^2 + b\lambda + c$) where the values of a , b , and c are as follows:

$$\begin{aligned} a &= m_{11}m_{22}^2 - m_{22}m_{13}^2 & (22) \\ b &= 2m_{13}m_{22}m_{35}m_{15} - m_{22}^2m_{15}^2 - 2m_{11}m_{22}m_{35}^2 \\ &\quad + 2m_{13}m_{22}m_{34}m_{14} - 2m_{22}m_{13}^2m_{44} - m_{22}^2m_{14}^2 \\ &\quad + 2m_{11}m_{22}^2m_{44} + m_{13}^2m_{35}^2 - 2m_{11}m_{22}m_{34}^2 \\ &\quad + m_{13}^2m_{34}^2 \\ c &= -2m_{13}m_{35}^3m_{15} - m_{22}m_{13}^2m_{44}^2 + m_{11}m_{22}^2m_{44}^2 \\ &\quad + m_{13}^2m_{35}^2m_{44} + 2m_{13}m_{22}m_{34}m_{14}m_{44} \\ &\quad + m_{13}^2m_{34}^2m_{44} - 2m_{11}m_{22}m_{34}^2m_{44} \\ &\quad - 2m_{13}m_{34}^3m_{14} - 2m_{11}m_{22}m_{35}^2m_{44} \\ &\quad + 2m_{11}m_{35}^2m_{34}^2 + m_{22}m_{14}^2m_{35}^2 \\ &\quad - 2m_{13}m_{35}^2m_{34}m_{14} - 2m_{13}m_{34}^2m_{35}m_{15} \\ &\quad + m_{11}m_{34}^4 + m_{22}m_{15}^2m_{34}^2 \\ &\quad + m_{22}m_{35}^2m_{15}^2 + m_{11}m_{35}^4 \\ &\quad - m_{22}^2m_{14}^2m_{44} + 2m_{13}m_{22}m_{35}m_{15}m_{44} \\ &\quad + m_{22}m_{34}^2m_{14}^2 - m_{22}^2m_{15}^2m_{44} \end{aligned}$$

The number $\bar{\lambda}$, such that the matrix $(\mathbf{M} + \bar{\lambda} \mathbf{W})$ is singular, can be found in closed form by using the well-known formula for the roots of a second-order polynomial. After having found $\bar{\lambda}$, the choice of \mathbf{x} is unique given the constraints (18) and (19). Call \mathbf{v} any vector in the kernel of $(\mathbf{M} + \bar{\lambda} \mathbf{W})$. To obtain the estimate $\hat{\mathbf{x}}$, scale \mathbf{v} by $\sqrt{v_4^2 + v_5^2}$ to respect constraint (18), then flip it by the sign of v_1 to respect constraint (19):

$$\hat{\mathbf{x}} = \frac{\text{sign } v_1}{\sqrt{v_4^2 + v_5^2}} \mathbf{v}$$

Finally, all six parameters have been recovered:

$$\begin{aligned} \hat{b} &= \hat{x}_1 \\ \hat{r}_L &= +\hat{x}_1 \hat{J}_{21} \\ \hat{r}_R &= -\hat{x}_1 \hat{J}_{22} \\ \hat{\ell}_x &= \hat{x}_2 \\ \hat{\ell}_y &= \hat{x}_3 \\ \hat{\ell}_\theta &= \text{atan2}(\hat{x}_5, \hat{x}_4) \end{aligned}$$

III. EXPERIMENTS

In this section, we present our experience in calibrating Khepera III mobile robots¹ with on-board Hokuyo². A video attached to this submission shows the setting used for the experiments. All the experimental data and the software used is available for download from <http://purl.org/censi/2007/calib>

A. Equipment

The Kheperas III, with respect to the Kheperas II used in [17], have a better encoder resolution of about 7 ticks per degree, therefore we could get good smooth estimates of the velocities ω_L^k, ω_R^k .

The Hokuyo URG-04LX is a small lightweight range-finder sensor. It is the new model with respect to the sensor described in [18], but the under-lying technology is the same. It provides 681 rays over a 240° field of view, with a resolution of 1mm, and a standard deviation of about 3mm. It is not an ideal sensor. The measurements are highly correlated, with every ray's error being correlated with its 3-4 neighbors: this is probably a symptom of some post-processing (interpolation) to bump up the resolution to the nominal 1024/360 rays/degrees. There is temporal bias: readings change as much as 20mm over a period of 5 minutes — this could be due to the battery power, or the change in temperature. There is also a spatial bias which is a function of the distance [18]: in practice, a rectangular environment appears slightly curved to the sensor.

B. Problems

It is surprisingly challenging to gather the perfectly tidy experimental data that one would expect for calibration. We outline the main practical problems we faced, and the solutions we tried.

Data synchronization: We experienced some problems in interfacing with the sensor, probably due to the complicated communication protocol and the not-so-reliable serial-to-USB link. As a result, during the first period of experimentation, we were not able to retrieve the correct timestamp for the scans (this affects datasets k06–k21). We figured that, to mitigate this kind of problem, one should use trajectories

¹<http://www.k-team.com/>

²<http://www.hokuyo-aut.jp/>

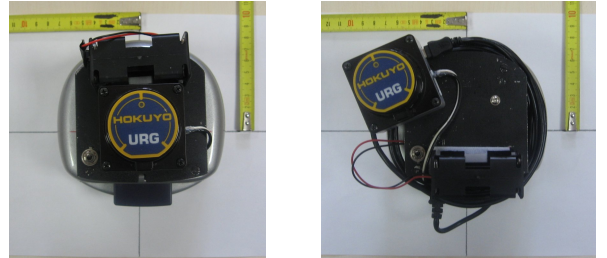


Fig. 3. Two configurations for the Hokuyo. In the first one, the sensor's pose is $\ell \simeq \mathbf{0}$ with respect to the robot frame (approximately indicated by the black lines on the white sheet). In the second, the sensor is translated and rotated.

TABLE I
FINAL CALIBRATION RESULTS

dataset(s)	# samples	b (m)	r_L (m)	r_R (m)	ℓ_x (m)	ℓ_y (m)	ℓ_θ (rad)	J_{11}	J_{12}	J_{21}	J_{22}
k07	766	0.09288	0.02057	0.02113	-0.00446	0.00000	-0.00209	0.01057	0.01028	-0.22752	0.22144
k08	350	0.08785	0.02008	0.01993	-0.00592	0.00020	-0.02371	0.00997	0.01004	-0.22692	0.22853
k12	540	0.08829	0.01987	0.02000	-0.00575	-0.00241	-0.00362	0.01000	0.00993	-0.22658	0.22501
k13	1032	0.08863	0.01958	0.01957	-0.00513	-0.00124	-0.01569	0.00979	0.00979	-0.22086	0.22088
k14	1172	0.08529	0.01868	0.01861	-0.00516	-0.00411	-0.00670	0.00931	0.00934	-0.21825	0.21905
k15	1585	0.07042	0.01519	0.01508	-0.00856	0.06416	0.04799	0.00754	0.00760	-0.21416	0.21578
k06-k15	5728	0.08586	0.01906	0.01901	-0.00441	0.00013	-0.01292	0.00951	0.00953	-0.22143	0.22196
k22	2243	0.09213	0.02109	0.02106	-0.00726	0.00286	-0.01298	0.01053	0.01055	-0.22861	0.22892
k23	2190	0.08543	0.01978	0.01978	-0.00659	0.00034	0.00692	0.00989	0.00989	-0.23159	0.23157
k27	2135	0.08822	0.02017	0.02016	-0.00403	0.00071	-0.01700	0.01008	0.01008	-0.22852	0.22860
k28	4184	0.08741	0.02002	0.01999	-0.00495	0.00223	-0.00816	0.01000	0.01001	-0.22870	0.22902
k29	229	0.08771	0.02020	0.02010	-0.00224	0.01104	-0.01830	0.01005	0.01010	-0.22918	0.23032
k22-k29	10987	0.08682	0.01999	0.01998	-0.00603	0.00125	-0.00156	0.00999	0.01000	-0.23012	0.23029
k40	3784	0.09018	0.02080	0.02074	0.02499	0.03859	0.35894	0.01037	0.01040	-0.22994	0.23066
k42	3516	0.08822	0.02033	0.02033	0.02538	0.03607	0.37073	0.01017	0.01016	-0.23048	0.23041
k43	1845	0.08742	0.02032	0.02026	0.02350	0.04180	0.35594	0.01013	0.01016	-0.23174	0.23242
k44	1335	0.08730	0.02039	0.02029	0.02307	0.04285	0.38065	0.01014	0.01019	-0.23241	0.23352
k45	3496	0.08924	0.02050	0.02047	0.02670	0.04435	0.30744	0.01024	0.01025	-0.22940	0.22974
k40-k45	16473	0.08790	0.02035	0.02029	0.02442	0.04087	0.36093	0.01015	0.01017	-0.23088	0.23146

TABLE II
FINAL CALIBRATION RESULTS (UMBMARK)

	b (m)	r_L (m)	r_R (m)	E_b	E_d	E_{\max}	J_{11}	J_{12}	J_{21}	J_{22}
nominal	0.090	0.021	0.021	1.0	1.0	17.8238	0.0105	0.0105	-0.23333	0.23333
1	0.08971	0.02100	0.02099	0.996854	0.99963	8.1993	0.010501	0.01050	-0.23411	0.23403
2	0.08970	0.02101	0.02099	0.996722	0.99883	6.5975	0.010506	0.01049	-0.23424	0.23396
3	0.08965	0.02102	0.02098	0.996180	0.99848	4.4295	0.010507	0.01049	-0.23441	0.23405
4	0.08962	0.02101	0.02099	0.995824	0.99868	3.8147	0.010506	0.01049	-0.23447	0.23416

with long tracts of constant speed. Only after installing the new firmware, released in August 2007, we could get the correct timestamps (datasets k22–k45).

Wheels slippage: From time to time, we observed severe slip of the wheels. From the measured data, slip is evident as the odometry typically gives an estimate of zero for the rotation, while scan matching detects a non-zero rotation. Slippage seriously impacts calibration. However, a simple post-processing using trimming can eliminate outliers.

In the first step, the parameters are estimated using all available measurement tuples $\langle \hat{s}^k, \hat{\omega}_L^k, \hat{\omega}_R^k \rangle$. Then, the residuals are computed as:

$$e^k \triangleq \hat{\ell} \oplus \hat{s}^k - \hat{o}^k \oplus \hat{\ell}$$

A fixed percentage of the samples having large residuals are considered outliers and discarded; then, the parameters are estimated again with the remaining samples. In our implementation, we discarded the worst 5% for 3 times in a row. Our method is convenient because we can compensate the poor quality of the data with an extremely large quantity of samples; compare, for example, with UMBmark, which requires very precise observation of some small set of data.

C. Datasets

We consider three sets of logs:

- From log ‘k06’ to ‘k21’, we were still using the old Hokuyo firmware, and we used the Khepera’s clock for timestamping the laser data upon reception.

- From log ‘k22’ to ‘k30’: laser and odometry data are properly synchronized. The laser is mounted at $\ell \simeq \mathbf{0}$.
- From log ‘k40’ to ‘k45’: the laser is mounted with a different pose (Fig. 3). We measured the new pose of the laser and obtained approximately (45mm, 35m, 0.40rad).

Note that some of the logs did not excite all parameters. For example, dataset k06 consists of only identical turns, and the linear least-squares problem is under-constrained. However, such data can be used in conjunction with other logs. In general, each log is seen by our method as just a large set of measurements tuples $\langle \hat{s}^k, \hat{\omega}_L^k, \hat{\omega}_R^k \rangle$; even the order of the tuples is not important. Therefore a single log can be split in multiple parts, or multiple logs can be joined together.

D. Results

The calibration results are shown in Table I. In general, there is much variation in the estimate of the physical parameters: the estimate of r_L varies in the (0.087m, 0.092m) interval, and b varies in the (0.019m, 0.021m) interval. However, note that the ratio of such parameters (J_{21}) varies in a much smaller interval. This is because J_{21} is directly observable, while the single parameters are not. A study of the uncertainty of the estimate would clarify such matters, however it is not trivial in this case because of the nonlinearity in the problem.

For the logs ‘k40’ to ‘k45’, the Hokuyo was mounted in the second configuration (Fig. 3). The estimated pose for the Hokuyo was slightly different than our measured

pose. We were puzzled by these results, so we checked the distribution of the residuals. The residuals are interesting because they are the sum of all errors involved in the process: the scan-matching error, the encoder measurements error, any synchronization error, and the error due to the constant-speed approximation. All three components are distributed along the expected bell-shaped curves — the long tails are the very few outliers that get ignored in the estimation. The x, y errors are in the order of millimeters, and the θ errors are much less than 1° . Therefore, we can conclude that the estimation is ‘healthy’, and the final estimation errors are due to the limited information contained in the data. It could be that the logs we used did not excite enough the ℓ parameter: this could be clarified by doing an in-depth observability analysis of the problem.

E. Comparisons

This method is the first to calibrate both odometry and sensor parameters, therefore any comparison can only be qualitative. The method described in [12] estimates the matrix \mathbf{J} as 4 independent numbers, here instead we use the three physical parameters (r_L, r_R, b) — this is actually harder, as there is one constraint more to consider. Anyway, their method could be easily extended to add one constraint to \mathbf{J} , and our method could easily use independent elements. It is unclear which approach is the best.

We report here an anecdote about our use of the UMBmark. Before even thinking about the method described in this paper, we needed to calibrate our robots, and we chose the UMBmark, using a ceiling-mounted camera as the external sensor. The nominal parameters, and the updated parameters after each trial, are reported in Table II. We used 21mm as the nominal radius. One problem with UMBmark is that it constrains the radii average to be fixed at the nominal radius. After running our method, we see that in reality the radii should be slightly less than 21mm, and therefore the UMBmark estimate we used is biased.

IV. CONCLUSIONS AND FUTURE WORK

In this paper we presented a simple and practical method for simultaneous calibration of the odometry and sensor parameters. The method has some interesting characteristics:

- It can run unattended, with no human intervention.
- No apparatus has to be calibrated a priori.
- There is no need for nominal parameters.
- Trajectories can be freely chosen.

Future work concerns characterizing the uncertainty of the estimate. In [12], the author could obtain an estimate of the covariance because they were facing a completely linear problem. Here, two things differ: we use a 3 parameter model, instead of estimating an arbitrary 4-element matrix, and, of course, the other three parameters introduce a non-linearity ($\cos \ell_\theta, \sin \ell_\theta$). Hence it is not trivial to compute the uncertainty.

We have also concerns about different forms of bias. The sensor that we use suffers of both temporal and spatial bias. Moreover, the ICP algorithm used for scan-matching is

biased, and its bias is hard to measure. Even if the sensor and the scan-matching algorithm were unbiased, the maximum-likelihood minimization we use is biased itself. Our intuition is that these four (independent!) sources of bias constitute a non-negligible part of the final error, and should be object of further study.

REFERENCES

- [1] C. M. Wang, “Location estimation and uncertainty analysis for mobile robots,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, (Philadelphia, PA, USA), pp. 1231–1235, Apr. 1988.
- [2] J. Borenstein and L. Feng, “Measurement and correction of systematic odometry errors in mobile robots,” *IEEE Transactions on Robotics and Automation*, vol. 12, December 1996.
- [3] A. Kelly, “General solution for linearized systematic error propagation in vehicle odometry,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, (Maui, Hawaii), November 2001.
- [4] A. Martinelli, “The odometry error of a mobile robot with a synchronous drive system,” *IEEE Transaction on Robotics and Automation*, vol. 18, pp. 399–405, June 2002.
- [5] A. Martinelli, “Evaluating the odometry error of a mobile robot,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1, (Lausanne, Switzerland), pp. 853–858, September 2002.
- [6] A. Kelly, “Fast and easy systematic and stochastic odometry calibration,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 4, pp. 3188–3194, Sept./Oct. 2004.
- [7] T. D. Larsen, M. Bak, N. A. Andersen, and O. Ravn, “Location estimation for an autonomously guided vehicle using an augmented Kalman filter to autocalibrate the odometry,” in *First International Conference on Multisensor-Multisensor Information Fusion (FUSION’98)*, 1998.
- [8] D. Caltabiano, G. Muscato, and F. Russo, “Localization and self-calibration of a robot for volcano exploration,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 1, pp. 586–591, Apr./May 2004.
- [9] A. Martinelli, N. Tomatis, and R. Siegwart, “Simultaneous localization and odometry self calibration for mobile robot,” *Autonomous Robots*, vol. 22, pp. 75–85, 2007.
- [10] A. Martinelli and R. Siegwart, “Observability properties and optimal trajectories for on-line odometry self-calibration,” in *45th IEEE Conference on Decision and Control*, pp. 3065–3070, Dec. 2006.
- [11] N. Roy and S. Thrun, “Online self-calibration for mobile robots,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, 1999.
- [12] G. Antonelli, S. Chiaverini, and G. Fusco, “A calibration method for odometry of mobile robots based on the least-squares technique: theory and experimental validation,” *IEEE Transactions on Robotics*, vol. 21, pp. 994–1004, Oct. 2005.
- [13] H. J. Von der Hardt, R. Husson, and D. Wolf, “An automatic calibration method for a multisensor system: application to a mobile robot localization system,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 4, (Leuven, Belgium), pp. 3141–3146, May 1998.
- [14] A. Martinelli and D. Scaramuzza, “Automatic self-calibration of a vision system during robot motion,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, (Orlando, Florida), 2006.
- [15] R. C. Smith and P. Cheeseman, “On the representation and estimation of spatial uncertainty,” *International Journal of Robotics Research*, vol. 5, no. 4, pp. 56–68, 1987.
- [16] A. Censi, “An accurate closed-form estimate of ICP’s covariance,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, (Rome, Italy), pp. 3167–3172, Apr. 2007.
- [17] G. Antonelli and S. Chiaverini, “Experimental odometry calibration of the mobile robot Khepera II based on the least-squares technique,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, (Barcelona, Spain), April 2005.
- [18] H. Kawata, A. Ohya, S. Yuta, W. Santosh, and T. Mori, “Development of ultra-small lightweight optical range sensor system,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1078–1083, Aug. 2005.