Probabilistic sensor data processing for robot localization on load-sensing floors

Maxime Rio, Francis Colas, Mihai Andries, François Charpillet

Probabilistic sensor data processing for robot localization on load-sensing floors

Maxime Rio*†‡, Francis Colas‡, Mihai Andries‡§, and François Charpillet‡
† Biozentrum, University of Basel, Klingelbergstrasse 50/70, 4056 Basel, Switzerland
‡ Inria, Villers-lès-Nancy, F-54600, France
CNRS, Loria, UMR n° 7503, Vandœuvre-lès-Nancy, F-54500, France
Université de Lorraine, Loria, UMR n° 7503, Vandœuvre-lès-Nancy, F-54500, France
§ ISIR, Université Pierre et Marie Curie, UMR n° 7222, Paris, France
* Corresponding author: maxime.rio@unibas.ch

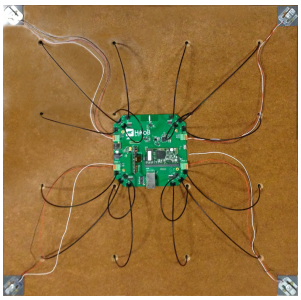
***Abstract*— Load-sensing floors are capable of tracking objects without suffering from occlusions nor posing the same privacy issues as cameras. They have been mostly used to analyze human gait as a way of continuous diagnosis but could also be placed alongside robots to help monitoring in specialized institutions, such as elderly care facilities. However, large-scale deployments necessitate cheap sensors which do not necessarily offer the same precision. With more noisy sensors, lighter robots might be difficult to track and precisely localize. In this article, we investigate various models in order to estimate the position of a robot. We experiment with several robots of different weights and compare the models' estimates against ground truth measurements provided by a motion capture system. We show that with standard-sized tiles of 60 cm, we can track even the lighter robots with less than 4 cm of error.**

I. Introduction

Environments with embedded ambient intelligence are more and more often equipped with sensing floors, due to their inherent advantages. They can localize the objects supported by the floor, they don't suffer from occlusions, and are less intrusive than vision sensors. Although high-precision object tracking is straightforward on high-resolution floors, it is less obvious whether tracking is feasible on cheap, low-resolution, noisy floors. This paper presents a method to process sensor data, allowing for high-precision tracking of robots on a low-resolution, noisy load-sensing floor.

Ever since their introduction in the 1990's [1], [2], sensing-floor prototypes have been mostly developed for tracking and analysing the human gait for medical applications [3], such as continuous diagnosis of the health-state of a person. Different technologies have been used for pressure sensing, such as strain gauge load cells [1], [2], electro-mechanical film [4], force-sensing resistor mats [5], [6], [7], piezo-electric sensors [8], and photo-interruptor sensors [9]. These floors detect and localize objects either directly, by identifying which sensor has measured the pressure, or by calculating the position of the center of pressure in the case of floors with a low density of sensors. The latter, however, have difficulties tracking multiple objects whenever they are too close compared to the density of sensors. Thus, precise localization comes at a cost proportional to the density of load sensors in the floor. We work with a low-resolution modular floor prototype [10] designed as a low-cost sensor for ambient intelligence applications (Figure 1).

(a) Our prototype apartment with the tiled sensing floor.



(b) Underside of a tile with the load sensors in the corners.

Fig. 1: Experimental platform consisting in (a) a two-room apartment whose floor is covered with (b) 90 smart tiles (60 cm wide) with four load cells in the corners.

Given that all the mentioned floors have been designed to fulfill the precision requirements of their applications, no previous work has centered on increasing the precision of the sensor measurements through post-treatment of the pressure and position data. However, efforts were made to improve tracking when doing it for multiple persons on a binary-sensing floor (subject to tracking ambiguity), and on sensing floors that do not cover all the floor, as detailed below.

To perform multi-person tracking on a binary pressure-sensing floor, Suutala *et al.* [11] used Gaussian Process Joint Particle Filtering. Similarly, Murakita *et al.* [12] used a Markov Chain Monte Carlo algorithm: their prediction model contained 2 parts: a generic linear model for predicting human's position (using his previous position, his current velocity, and a Gaussian noise), and a bipedal model of sensor activations while walking (based on which foot was put forward). In order to track inhabitants on a tiled floor with sensorless spacings of 20 cm between tiles, Wen-hau *et al.* [13] used a Probability Data Association (PDA) algorithm, which is based on the Kalman filter.

None of these works are able to provide a high-precision localization of objects. In contrast, this paper presents a technique for fine tracking of objects, in spite of the obstacles posed by noisy sensors, and the low resolution of the floor.

We first present in more detail the apartment, including the load-sensing floors and the robots (section II). Then,

(a) Light robot in the smart apartment (Turtlebot 2, 6.3 kg including notebook). Source: www.turtlebot.com

(b) A heavy robot in the smart apartment (Robosoft robuLAB-10, 35.7 kg). Source: www.robosoft.com

Fig. 2: Robots used in this paper.

we propose several models to localize robots on such a floor (section III). Finally we compare those models on real data taken with two different robots (section IV) before concluding with a discussion about the choice of the models (section V).

II. Platform

This work is a contribution to an innovative platform for favoring research in assistance for elderly people at home. This platform consists of a two-room apartment (Figure 1a), with numerous sensing and connected devices such as mobile robots, a motion capture system, a network of RGB-D cameras, and a load-sensing floor made up of 90 tiles, all connected with the Robotic Operating System (ROS) ([14]).

A. Tiles

The load-sensing floor is composed of square 60×60 cm tiles, each equipped with two ARM processors (Cortex m3 and a8), four load cells in the corners (Figure 1b), 16 LEDs, and a wired ethernet connection. There is also a magnetometer, that could be used to detect variations of magnetic field induced by a moving metallic object such as a robot, and an accelerometer to detect shocks. Both of these are not used in this paper.

B. Motion Capture system

We installed a motion capture system with 8 Qualisys oqus 700 cameras allowing to track the position of robots at 150 Hz with high accuracy (less than 1 mm). In this paper, this tracking is considered as the ground truth for the localization (see subsection IV-A for more details).

C. Robots

An other objective of the platform is to develop companion robots able to interact with people and monitoring their activities while detecting emergency situations. The chosen approach is to propose services which can run on low-cost robotic platforms such as Turtlebots (Figure 2a) or Robosoft Robulab mobile robot (Figure 2b).

III. Models

In this section, we present various estimators with increasing complexity: a full direct estimate, a direct estimate on selected tiles, a Kalman Filter (KF) and an Extended Kalman Filter (EKF).

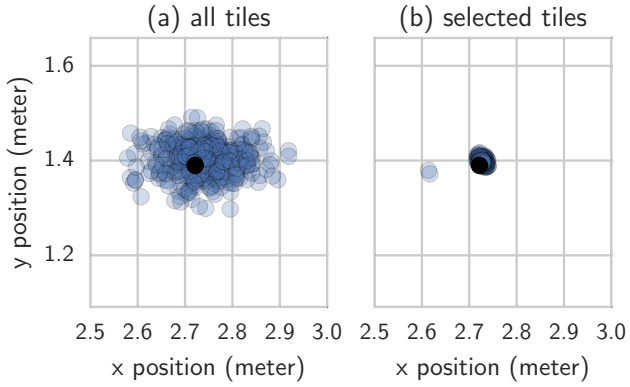


Fig. 3: Estimates of the Robulab position using sensor data from (a) all available tiles of the floor and (b) a subset of the tiles selected with a hypothesis test on the measured force, at $\alpha = 10^{-8}$. Theoretical position is displayed as a black circle.

A. Direct estimation

The simplest way to estimate the center-of-pressure (CoP) position of an object placed on a tile is to compute the barycenter of the load measured by the gauge sensors:

$$\begin{pmatrix} \hat{x}_t \\ \hat{y}_t \end{pmatrix} = \frac{\sum_i \tilde{s}_{i,t} . \begin{pmatrix} x_i \\ y_i \end{pmatrix}}{\sum_i \tilde{s}_{i,t}}, \quad (1)$$

where $\hat{x}_t$ and $\hat{y}_t$ are the estimated coordinates of the CoP at time frame t, $\tilde{s}_{i,t}$ the load measured[1] by sensor i and (x_i, y_i) the coordinates of sensor i. In case the object is spanning several tiles, the CoP can be computed as a weighted average of the estimates of each tile, which simplifies to Equation 1 considering the sensors of the tiles all together. We can even use the same equation with all sensors of the sensitive floor as non-loaded tiles should not record any force.

However, as we use noisy sensors, this ideal case provides poor estimates of the CoP location. Indeed, the summation terms in the numerator and denominator of Equation 1 imply that the variance of these quantities grows with the number of noisy measures summed. Consequently, the variance of the ratio also grows [15], [16] (Figure 3a).

B. Selection of tiles

In order to reduce the noise level of direct estimates, one idea is to select at each time frame the right subset of tiles that actually have a significant load. The selection is made independently for each tile and not each sensor because on the one hand sensors of a tile are mechanically coupled (since our tiles are rigid) and on the other hand we do not have a precise enough model of the noise of a single sensor. To decide whether there is a load on a tile, we do a statistical test on the total force measured by its sensors, with a null hypothesis H_0 of no force. The tile is selected if H_0 is

[1] We have to subtract the load when the tile is empty, which is estimated as the smallest of the means of the values in a given time-window.

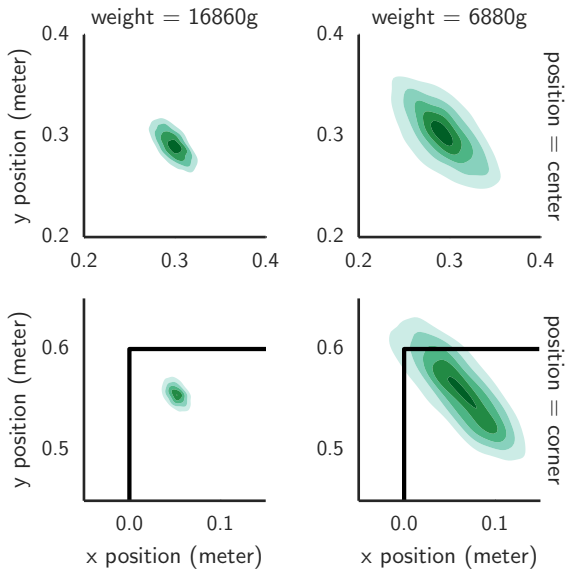


Fig. 4: Empirical distributions of the observed CoP $\tilde{\mathbf{z}}$, for two different positions and weights of a static punctiform object placed on one tile. Top row: the object is placed in the center of a tile; bottom row: the object is placed in a corner. Left column: a heavy object; right column: a light object. Black thick lines represent the tile contours.

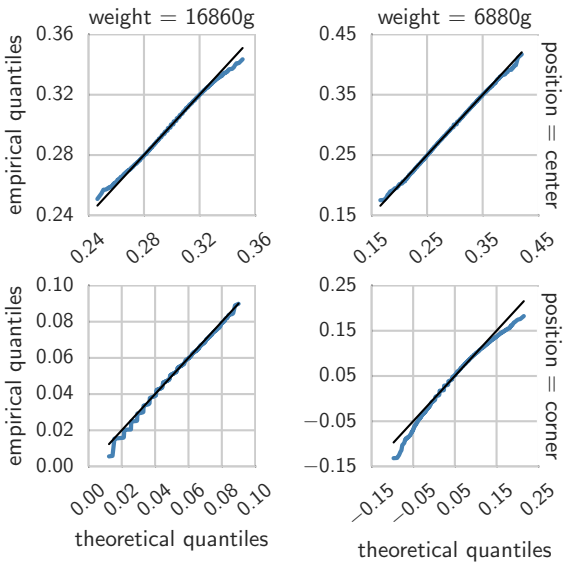


Fig. 5: Q-Q plots of the distributions of the observed CoP $\tilde{x}$, for the same scenarios as in Figure 4. Top row: the object is placed in the center of a tile; bottom row: the object is placed in a corner. Left column: a heavy object; right column: a light object. Straight lines represent theoretical curves under a Gaussian distribution assumption.

rejected at a fixed α level. Otherwise, we ignore it for this time frame.

Let F_t be the random variable associated with the total force measured on the tile. Assuming noise of each sensor is independent additive white noise of variance σ_i^2, we approximate the distribution of F_t under H_0 with a zero-mean Gaussian distribution of variance $\sigma^2 = \sum_i \sigma_i^2$. Therefore, the probability of measuring a force as high as the observed load $\tilde{f}_t = \sum_i \tilde{s}_{i,t}$ under H_0 is written as follows:

$$P(F_t \geq \tilde{f}_t | H_0) = \Phi(-\tilde{f}_t / \sqrt{\sigma^2}), \tag{2}$$

where Φ stands for the cumulative density function of the standard normal distribution.[2] The null hypothesis H_0 at is rejected at level α if $p(F_t \geq \tilde{f}_t | H_0) < \alpha$.

The α level corresponds to a false positive rate considered as acceptable. To set it, we take into account the multiple hypothesis tests issue, as one statistical test is made per tile and per time frame. Thus, to control the familywise error rate (FWER), i.e. the probability to have one false positive for all tests together, we add a Bonferroni correction to α, dividing it by the total number of tests made for a fixed period of time [17]. Figure 3 highlights the benefits of this tile selection approach.

C. Filtering

The objects we are tracking with such a sensing floor are moving robots. Therefore, we can use the temporal information to reduce the noise level of the CoP position estimates, considering a model of the robot movements.

The simplest possible approach is to employ a KF to track the position of the robot through time from the noisy direct estimates. The hidden state we seek to estimate is the CoP position $\mathbf{x}_t = (x_t, y_t)$ in the global reference frame. Observations are the direct estimates $\tilde{\mathbf{z}}_t = (\hat{x}_t, \hat{y}_t)$, as computed by Equation 1. We chose basic transition and measurement models:

$$\mathbf{x}_t \sim \mathcal{N}(\mathbf{x}_{t-1}, \mathbf{Q}_t) \quad \text{and} \quad \mathbf{z}_t \sim \mathcal{N}(\mathbf{z}_t, \mathbf{R}),$$

where $\mathcal{N}(\mu, \mathbf{\Sigma})$ is the Gaussian distribution with mean μ and covariance matrix $\mathbf{\Sigma}$.

The matrix $\mathbf{Q}_t$ is specified as a scaled identity matrix, that allows for the motion of the robot through the noise component of the transition model. Our system is asynchronous and to account for the various time arrivals of the sensor data, the $\mathbf{Q}_t$ matrix is scaled by the time interval Δt: $\mathbf{Q}_t = \Delta t \mathbf{Q}_0$, where $\mathbf{Q}_0$ is the covariance expected after 1 s. As an approximation of real phenomena, the matrix $\mathbf{R}$ is also specified as a scaled identity matrix tuned to reflect the expected spread of the direct estimates in both x and y directions.

The KF might not perform optimally as the noise assumptions in the measurement model do not reflect the real noise of the direct estimates. Indeed, as shown by Figure 4, the noise of the observations is correlated, position dependent and weight dependent. The Q-Q plots in Figure 5 highlight a non-Gaussianity of the noise, as the curve deviates from the theoretical one in the tails of the distribution. This effect

[2]Even if the distribution of noise within each sensor is not Gaussian, in practice the sum is well approximated by a Gaussian distribution, as a result of the central-limit theorem.

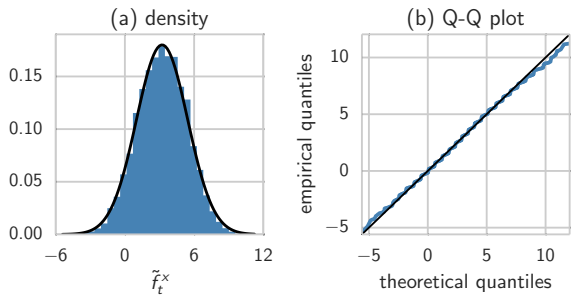


Fig. 6: Comparisons of the empirical distribution of $\tilde{f}_t^x$ with a fitted Gaussian distribution, for the scenario from Figure 4 of a light punctiform object located in the corner of a tile. Black curves represent the theoretical curves.

is particularly visible in case of a light weight placed in a corner.

Thus, setting $\mathbf{R}$ to a sensible value might turn out to be a difficult task, which is not feasible without knowing the weight of the tracked object. Besides, it will not fix the violation of the Gaussianity noise implied by the KF model.

D. Observations transformation

The particular issue of the non-Gaussian distribution stems from the division in the computation of the barycenter. More precisely, both numerator and denominator terms in Equation 1 are made of random values as explained above. These terms corresponding to weighted sums of finite-variance random variables, by virtue of the central-limit theorem, their distributions are approximately Gaussian, correlated by construction. [15], [16] provides the analytical form for the probability density function of a ratio of two correlated Gaussian variables which proves to be unwieldy.

We propose to change the space of observation $\mathbf{z}_t$ of the filter. The objective is to find the simplest observation space for which the Gaussianity assumption reasonably holds. Our solution is to separate the denominator from the numerators and add it as a third dimension:

$$\mathbf{z}_t = \begin{pmatrix} f_t^x \\ f_t^y \\ f_t \end{pmatrix} = \begin{pmatrix} \sum_i s_i x_i \\ \sum_i s_i y_i \\ \sum_i s_i \end{pmatrix} = \mathbf{C}\mathbf{s}_t, \tag{3}$$

where, for N sensors,

$$\mathbf{C} = \begin{pmatrix} x_1 & \dots & x_N \\ y_1 & \dots & y_N \\ 1 & \dots & 1 \end{pmatrix} \quad \text{and} \quad \mathbf{s}_t = \begin{pmatrix} s_1 \\ \vdots \\ s_N \end{pmatrix}.$$

As illustrated by Figure 6, the observation values in this new input space tend to be more Gaussian, even for the most problematic case of a light object placed in a corner of a tile.

With these observations, the state of the filter has to incorporate the force f_t at time t, and becomes $\mathbf{x}_t = (x_t, y_t, f_t)$. The transition model of the filter can be written similarly as above:

$$\mathbf{x}_t \sim \mathcal{N}(\mathbf{x}_{t-1}, \mathbf{Q}_t),$$

Algorithm 1 EKF with tile selection

Require: sensors measures $\{\tilde{s}_{i,t}\}_{i=1}^N$ at time t
Require: previous estimate $\hat{\mathbf{x}}_{t-1}$
Require: previous covariance $\hat{\mathbf{P}}_{t-1}$
  for all tiles j **do** ▷ individual tiles selection
    $\tilde{f}_{j,t} \leftarrow \sum \tilde{s}_{i,f}$ if sensor i belongs to tile j
    if $\Phi(-\tilde{f}_{j,t}/\sigma_j) < \alpha$ **then**
      reject tile j
    end if
  end for
  $\tilde{\mathbf{z}}_t \leftarrow \mathbf{C}\tilde{\mathbf{s}}_t$ ▷ observation on remaining sensors
  $\mathbf{R}_t \leftarrow \mathbf{C}\mathbf{S}_t\mathbf{C}^\intercal$
  $\begin{cases} \hat{\mathbf{x}}_t & \leftarrow & \hat{\mathbf{x}}_{t-1} \\ \hat{\mathbf{P}}_t & \leftarrow & \hat{\mathbf{P}}_{t-1} + \Delta t \mathbf{Q}_0 \end{cases}$ ▷ filter prediction
  if $\Phi((\hat{f}_t - \tilde{f}_t)/\sqrt{\mathbf{R}_t^{3,3}}) \geq \beta$ **then** ▷ observation rejection
    $\mathbf{H}_t \leftarrow \mathbf{H}_{\hat{\mathbf{x}}_t}$ ▷ linearization of observation model
    $\tilde{\mathbf{y}}_t \leftarrow \tilde{\mathbf{z}}_t - g(\hat{\mathbf{x}}_t)$
    $\mathbf{S}_t \leftarrow \mathbf{H}_t\mathbf{P}_t\mathbf{H}_t^\intercal + \mathbf{R}_t$
    $\mathbf{K}_t \leftarrow \mathbf{P}_t\mathbf{H}_t\mathbf{S}^{-1}$
    $\begin{cases} \hat{\mathbf{x}}_t & \leftarrow & \hat{\mathbf{x}}_t + \mathbf{K}_t\tilde{\mathbf{y}}_t \\ \hat{\mathbf{P}}_t & \leftarrow & (\mathbf{I}_3 - \mathbf{K}_t\mathbf{H}_t)\hat{\mathbf{P}}_t \end{cases}$ ▷ filter update
  end if
  return state estimate $\hat{\mathbf{x}}_t$ and its covariance matrix $\hat{\mathbf{P}}_t$

but the measurement model does change:

$$\mathbf{z}_t \sim \mathcal{N}(g(\mathbf{x}_t), \mathbf{R}_t) \quad \text{with} \quad g(\mathbf{x}_t) = \begin{pmatrix} f_t x_t \\ f_t y_t \\ f_t \end{pmatrix}.$$

The noise terms are still unbiased Gaussian noise with covariance matrices $\mathbf{Q}_t$ and $\mathbf{R}_t$. Interestingly, $\mathbf{R}_t$ can be deduced from Equation 3 as:

$$\mathbf{R} = \mathbf{C}\mathbf{S}_t\mathbf{C}^\intercal \quad \text{with} \quad \mathbf{S}_t = \begin{pmatrix} \sigma_1^2 & & 0 \\ & \ddots & \\ 0 & & \sigma_N^2 \end{pmatrix},$$

where σ_i^2 are the sensors noise variance. We have observed that the individual sensors noise do not depend on the weight (which does not prevent the CoP depending on it), thus $\mathbf{R}_t$ does not depend on the weight. But note that, due to the tile selection procedure, $\mathbf{R}_t$ changes at each time step according to the retained sensors. Besides, the measurement model is no longer linear w.r.t. the state vector and we thus need to implement an EKF which involves $\mathbf{H}_\mathbf{x}$, the Jacobian of g at a value $\mathbf{x}$.

Finally, as this filter tracks the object weight, it can be affected by the selection algorithm presented in subsection III-B. Indeed, when a light robot spreads its load on several tiles, there is a risk that only a subset of these tiles get selected, providing an incorrect value for $\mathbf{z}_t$. To prevent this issue, we added a test after the prediction step of the EKF, discarding the update step if the probability of observing a value as low as $\tilde{f}_t$ is smaller than a fixed threshold β:

$$P(F_t \leq \tilde{f}_t) = \Phi((f_t - \tilde{f}_t)/\sqrt{\mathbf{R}_t^{3,3}}) < \beta,$$

Turtlebot	Methods		
Scenarios	DE-TS	KF	EKF
all	6.0 cm (±6.1)	3.8 cm (±2.6)	5.1 cm (±5.8)
static	5.0 cm (±4.2)	3.5 cm (±1.8)	3.7 cm (±1.9)
rotation	6.9 cm (±7.5)	5.5 cm (±2.9)	5.6 cm (±2.7)
straight line	6.3 cm (±8.2)	3.7 cm (±2.9)	6.0 cm (±8.3)
rectangle	5.8 cm (±5.6)	3.8 cm (±2.6)	5.1 cm (±6.0)
figure eight	6.5 cm (±5.9)	3.7 cm (±2.7)	5.0 cm (±5.1)
Robulab	Methods		
Scenarios	DE-TS	KF	EKF
all	2.1 cm (±2.8)	2.1 cm (±2.9)	3.9 cm (±7.2)
static	1.3 cm (±1.0)	1.2 cm (±0.3)	1.4 cm (±0.7)
rotation	2.0 cm (±1.2)	1.9 cm (±0.8)	2.0 cm (±0.9)
straight line	1.9 cm (±1.3)	1.7 cm (±0.8)	4.1 cm (±6.9)
rectangle	2.2 cm (±4.0)	2.4 cm (±4.2)	5.8 cm (±9.7)
figure eight	2.2 cm (±1.7)	2.2 cm (±1.5)	1.8 cm (±1.2)

TABLE I: Summary values for the evaluation of the methods, as errors mean (± sd.) in centimeters. Top: with the light robot. Bottom: with the heavy robot.

where noise covariance matrix $\mathbf{R}_t$ can change at each time step depending on the involved tiles.

The specific equations of the EKF combined with the other steps of the model are summarized by Algorithm 1.

IV. Applications

In order to evaluate the relevance of the models presented in the previous section, we designed a set of experiments with robots of different weights executing different types of trajectories.

A. Evaluation

Our dataset is made of 10 replications – per robot – of several kinds of trajectories: static robot, rotation on the spot, straight line, rectangle, and driven in a figure eight. These trajectories have been recorded in a rectangular area covered with 3 by 5 sensing tiles. The location of the robot was tracked with the Qualysis motion capture system.

To investigate the effect of the weight on the estimated trajectories, we employed an heavy Robulab robot (35.7 kg) and a lighter Turtlebot robot (6.3 kg).

Our criterion to assess the quality of the estimated trajectories is absolute error of the CoP. It was therefore necessary to calibrate both the transformation between the motion capture and the tiles reference frames, and the CoP position of the robot w.r.t. the motion capture object. This was done using least-square estimation on a subset of the figure-eight scenarios with all three methods.

The methods evaluated are: the direct estimates with tile selection (DE-TS), the KF and the EKF. For the KF method, several values of $\mathbf{R}$ were tested. Similarly, we tested several values of β parameter of EKF. Both filters used the same $\mathbf{Q}_0$ values for (x_t, y_t) states, fixed at 0.1 m, to reflect the range of speed of both robots. We fixed $\alpha = 10^{-8}$ for the selection of tiles.[3]

[3]This value corresponds to an expected false positive rate of less than one per hour for a set of 100 tiles with a sampling frequency of 50 Hz.

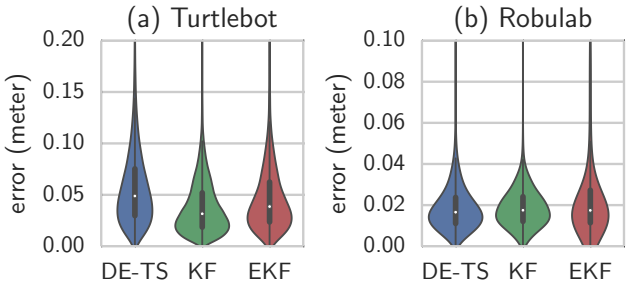


Fig. 7: Comparison of error distributions for all models. Left: on the light robot; right: on the heavy robot.

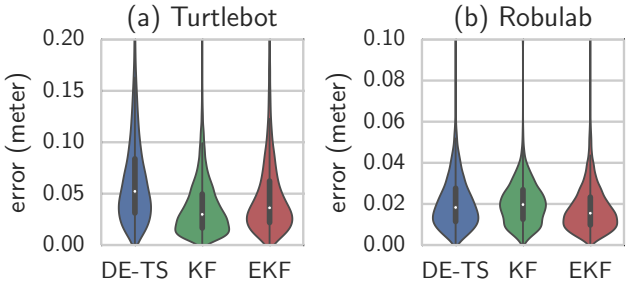


Fig. 8: Comparison of error distributions for all models, in the figure eight scenario. Left: on the light robot; right: on the heavy robot.

B. Global results

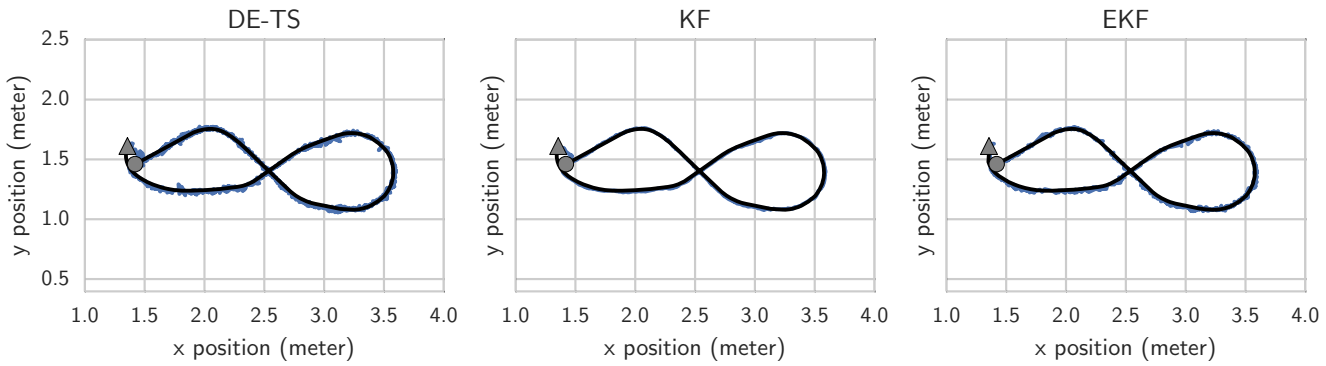
The results with our dataset are summarized per robot in Table I, for all scenarios pooled together and per scenario. For the KF and EKF models, which depend on parameter values, only the best results are reported ($\mathbf{R} = 0.1\,\mathbf{Id}$ and $\beta = 0.1$), an extended analysis of their sensitivity to their parameter being presented in the following sections.

The three models (DE-TS, KF, EKF) present similar results when pooling data from all scenarios (see Figure 7 shown using violin plots [18] with inner boxplots). They are all capable of tracking a robot with a maximum average error of 6 cm. The KF model outperforms the two other models for the lightest robot, with 1 cm less in average error.[4] The same pattern arises when analyzing each scenario individually, as illustrated in Figure 8 for the figure eight scenario.

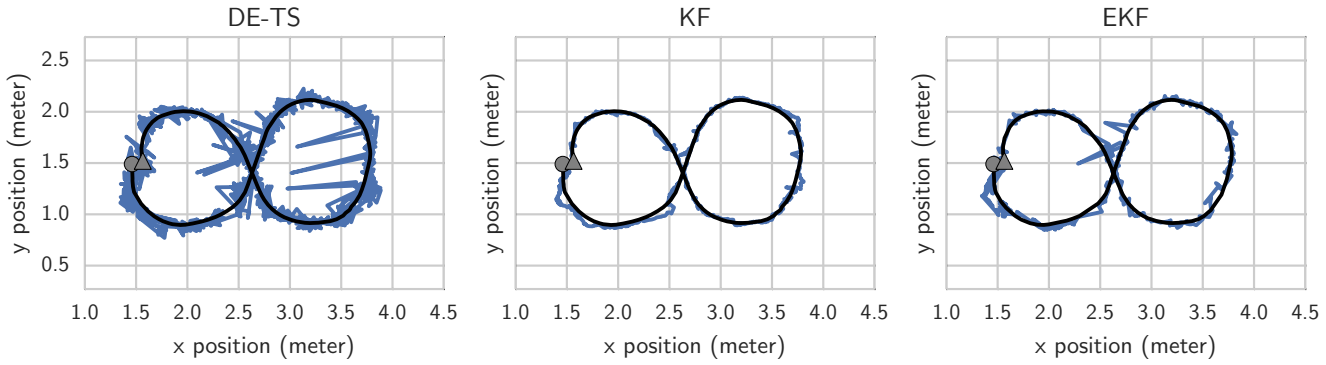
Even if most differences of average errors between methods are statistically significant, these differences are in an order of magnitude of 1 cm, while the standard deviation of the errors is up to 7 cm for the worst case. Therefore, the mean errors can be considered – for practical purposes – as sensibly identical for these methods, and a more valuable analysis is found in looking at the spread of the errors.

In this case, we can see that DE-TS performs worse than KF and similarly to EKF, with a standard deviation of 6.1 cm, compared to respectively 2.6 cm and 5.8 cm, when dealing with the light Turtlebot robot. This implies that trajectories

[4]This result is statistically significant at p-value ≤ 0.01, assessed by a bootstrap test for the difference between two sample means (2000 samples).



(a) Robulab



(b) Turtlebot

Fig. 9: Estimated trajectories in a figure eight scenario. Black line: ground truth trajectory; blue line: estimated trajectory. Grey circle: starting point; grey triangle: stopping point. Top row: the heavy robot; bottom row: the light robot.

estimated by DE-TS and EKF are noisier, which can be seen in a sample trajectory in Figure 9.

Concerning the effect of the robot weight, the performances of DE-TS, KF and EKF methods respectively degrade by 3.9 cm, 1.7 cm and 1.2 cm between the heavy robot and the light robot. Furthermore, robot lightness drastically increases the standard deviation of the error for DE-TS.

C. *Influence of* $\mathbf{R}$ *tuning in KF*

One concern that arose while designing the KF model was the value of $\mathbf{R}$. Assuming this matrix is a scaled identity matrix, we tried different values for the diagonal element σ_{xy}^2, i.e. the variance of CoP position direct estimates.

It turns out that the KF is robust to the setting of the same parameter for both robots, as shown in Figure 10. Best results are obtained with $\sigma_{xy} = 0.05$ for Robulab and $\sigma_{xy} = 0.10$ with Turtlebot.

Underestimating the measurement noise has a mild effect in the Turtlebot case, slightly increasing the variance of the errors, but improves results in the Robulab case in terms of average error and variance. Overestimating the measurement noise increases both mean and variance of the errors, in the cases of both robots.

D. *Influence of* β *tuning in EKF*

Our EKF model includes a β parameter used to prevent poor estimates due to partial information from the selected

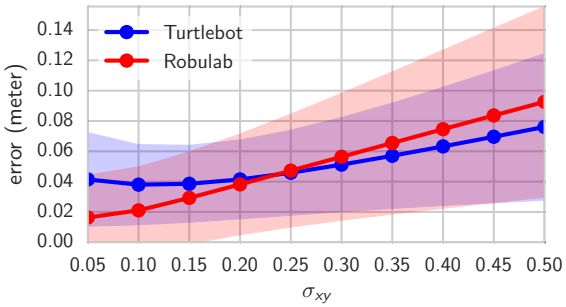


Fig. 10: Errors of KF depending on the σ_{xy} parameter ($R = \sigma_{xy}^2\mathbf{I}$). Thick lines represent the mean error. Shades represent one standard deviation around the mean. In blue: the light robot; in red: the heavy robot.

tiles. As it acts as a threshold on probabilities, we used a logarithmic sampling to explore the influence of this parameter. Results are graphically transposed in Figure 11.

The effect of this parameter is negligeable in the Robulab case. It is highly likely that very few time frames are affected by the issue of incorrectly rejected tiles, due to the good signal to noise ratio brought by the robot weight.

On the other hand, β affects the results for the Turtlebot. Up to $\beta = 10^{-1}$, this parameter decreases the mean error.

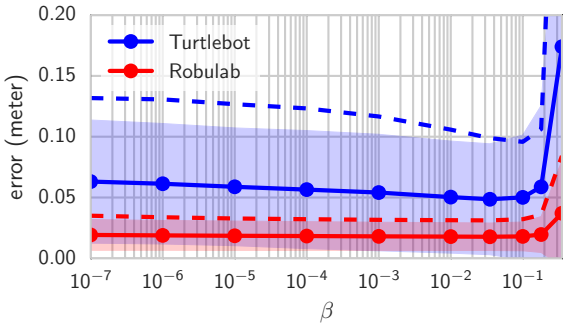


Fig. 11: Errors of EKF depending on β parameter. Thick lines represent the mean error. Shades represent one standard deviation around the mean. Dashed lines represent 90th percentile of the error. In blue: the light robot; in red: the heavy robot.

Furthermore, it lowers the extreme values of the error, as shown by the 90th percentile displayed in Figure 11.

For too high values such as $\beta = 0.35 \approx 10^{-0.5}$, the EKF rejects too many observations to be able to track reliably the Turtlebot. Recall, when the EKF only predicts the position, without incorporating evidence from the observation, it replicates the last estimates, as our transition model does not incorporate any speed component.

V. Conclusion

In this paper, we have proposed several models to estimate the CoP position of wandering mobile robots on a low-cost load-sensing floor. We have evaluated them on several datasets, recorded with two robots and included ground truth. The results have shown that we are able to track heavy and light robots up to 4 cm.

This work highlights issues with load-sensing floors for tracking light targets, namely that the constant noise level on the individual sensors translates into an increased noise on the CoP position. Furthermore, the statistical distribution of such noise is highly correlated and becomes non-Gaussian, which has to be taken into consideration in the estimation process.

Interestingly, all these problems are non-issues in case of heavy robots. In these cases, a basic robust approach (DETS) performs very well.

We proposed two more advanced models, namely KF and EKF which are capable of accurately tracking light targets as well. They represent different trade-offs in complexity and parameter tuning. The KF model is simpler but requires a reasonably good estimate of the scale of the measurement noise covariance matrix. This implies knowledge of the weights of the robots one tracks on such a floor. In contrast, the EKF model is more complex but the parameters that need to be fixed do not depend on the tracked weight.

Both of these models can serve as basis for advanced models, which could include a motion model, kinematic constraints and control inputs at the expense of losing the generality of our approach. Other extensions would include multi-robot tracking in a mixed human-robot environment. For this purpose, one extension could be to segregate areas for each moving object before position estimation, using a more advanced tile selection process including temporal information.

To conclude, we believe that load-sensing floors, deployed for elderly assistance, could be useful both as gait analysis tools and tracking devices to help robots navigate in their environment.

Acknowledgements

The authors wish to thank Pauline Houlgatte and Mélanie Lelaure for their help and participation in the experimental data gathering.

References

[1] M. Addlesee, A. Jones, F. Livesey, and F. Samaria, "The ORL active floor," *Personal Communications, IEEE*, vol. 4, no. 5, pp. 35–41, 1997.

[2] R. J. Orr and G. D. Abowd, "The smart floor: a mechanism for natural user identification and tracking," in *CHI '00 Extended Abstracts on Human Factors in Computing Systems*. ACM, 2000, pp. 275–276.

[3] L. Ballaz, M. Raison, and C. Detrembleur, "Decomposition of the vertical ground reaction forces during gait on a single force plate," *Journal of musculoskeletal & neuronal interactions*, vol. 13, no. 2, pp. 236–243, 2013.

[4] J. Suutala and J. Röning, "Methods for person identification on a pressure-sensitive floor: Experiments with multiple classifiers and reject option," *Information Fusion*, vol. 9, no. 1, pp. 21–40, 2008.

[5] L. Middleton, A. A. Buss, A. Bazin, and M. S. Nixon, "A floor sensor system for gait recognition," in *IEEE Workshop on Automatic Identification Advanced Technologies*, 2005, pp. 171–176.

[6] G. Qian, J. Zhang, and A. Kidané, "People identification using floor pressure sensing and analysis," *Sensors Journal, IEEE*, vol. 10, no. 9, pp. 1447–1460, 2010.

[7] I. Al-Naimi, C. B. Wong, P. Moore, and X. Chen, "Advanced approach for indoor identification and tracking using smart floor and pyroelectric infrared sensors," in *International Conference on Information and Communication Systems (ICICS)*, 2014, pp. 1–6.

[8] R. Vera-Rodriguez, J. Mason, J. Fierrez, and J. Ortega-Garcia, "Comparative analysis and fusion of spatiotemporal information for footstep recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 4, pp. 823–834, 2013.

[9] J. Yun, "User identification using gait patterns on UbiFloorII," *Sensors*, vol. 11, no. 3, pp. 2611–2639, 2011.

[10] M. Andries, F. Charpillet, and O. Simonin, "Localisation of humans, objects and robots interacting on load-sensing floors," *IEEE Sensors Journal*, vol. 6, no. 4, pp. 1026–1037, 2015.

[11] J. Suutala, K. Fujinami, and J. Roning, "Persons tracking with Gaussian process joint particle filtering," in *IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, Aug 2010, pp. 160–165.

[12] T. Murakita, T. Ikeda, and H. Ishiguro, "Human tracking using floor sensors based on the Markov chain Monte Carlo method," in *International Conference on Pattern Recognition (ICPR)*, vol. 4, 2004, pp. 917–920.

[13] L. Wen-Hau, W. Chao-Lin, and F. Li-Chen, "Inhabitants tracking system in a cluttered home environment via floor load sensors," *IEEE Transactions on Automation Science and Engineering*, vol. 5, no. 1, pp. 10–20, 2008.

[14] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: an open-source robot operating system," in *ICRA Workshop on Open Source Software*, 2009.

[15] E. C. Fieller, "The distribution of the index in a normal bivariate population," *Biometrika*, vol. 24, no. 3/4, pp. 428–440, 1932.

[16] D. V. Hinkley, "On the ratio of two correlated normal random variables," *Biometrika*, vol. 56, no. 3, pp. 635–639, 1969.

[17] O. J. Dunn, "Multiple comparisons among means," *Journal of the American Statistical Association*, vol. 56, no. 293, pp. 52–64, 1961.

[18] J. L. Hintze and R. D. Nelson, "Violin plots: A box plot-density trace synergism," *The American Statistician*, vol. 52, no. 2, 1998.