Non-parametric and semi-parametric RSSI/distance modeling for target tracking in wireless sensor networks

Sandy Mahfouz, Farah Mourad-Chehade, Paul Honeine, Joumana Farah, and Hichem Snoussi

Abstract—This paper introduces two main contributions to the wireless sensor network (WSN) society. The first one consists of modeling the relationship between the distances separating sensors and the received signal strength indicators (RSSIs) exchanged by these sensors in an indoor WSN. In this context, two models are determined using a radio-fingerprints database and kernel-based learning methods. The first one is a non-parametric regression model, while the second one is a semi-parametric regression model that combines the well-known log-distance theoretical propagation model with a non-linear fluctuation term. As for the second contribution, it consists of tracking a moving target in the network using the estimated RSSI/distance models. The target's position is estimated by combining acceleration information and the estimated distances separating the target from sensors having known positions, using either the Kalman filter or the particle filter. A fully comprehensive study of the choice of parameters of the proposed distance models and their performances is provided, as well as a study of the performance of the two proposed tracking methods. Comparisons to recently proposed methods are also provided.

Keywords—*Distance estimation, Kalman filter, machine learning, particle filter, radio-fingerprints, RSSI, target tracking.*

I. INTRODUCTION

Recently, wireless sensor networks (WSNs) have received much concern because of their practical applications in nowadays world. WSNs are rapidly gaining importance in many fields, especially in the healthcare and industrial domains [1], [2]. They are also being deployed to track enemy vehicles in military applications [3], and to follow the movement of wild animals in environmental monitoring [4].

In all applications, awareness of location information is fundamental, since collected data are meaningless without any geographical context. Typically, stationary sensors broadcast signals in the network, while targets collect these signals for location estimation. Many localization algorithms using stationary sensors have been proposed. They are mainly based on estimating the distances between the stationary sensors and the targets to be localized. Estimating these distances can be based on several types of measurements, such as received signal strength indicators (RSSIs) [5], angle of arrival [6],

S. Mahfouz, F. Mourad-Chehade, and H. Snoussi are with the Institut Charles Delaunay (CNRS), Université de Technologie de Troyes, France, e-mail: {sandy.mahfouz,farah.mourad,hichem.snoussi}@utt.fr

P. Honeine is with the LITIS lab, Université de Rouen, Rouen, France, e-mail: paul.honeine@univ-rouen.fr

J. Farah is with the Faculty of Engineering of the Lebanese University, Roumieh, Lebanon, e-mail: joumanafarah@ul.edu.lb

time difference of arrival [7] and time-of-arrival [8]. The RSSI-based techniques exploit the attenuation of the signal strength with the traveled distance to estimate the distances separating the emitters from the receivers. These techniques exhibit favorable properties with respect to power consumption, size and cost. Nevertheless, it is important to highlight that distance estimation using RSSI can be really challenging, since the measurements of signals' powers could be significantly altered by the presence of additive noise, multi-path fading, shadowing, and other interferences. One popular model used to characterize the relationship between the RSSIs and the distances is the log-distance propagation model, also known as the Okumura-Hata model [9], [10], [11]. Even though this model has many limitations, it is still widely used because of its simplicity. However, this model is basically for outdoors, since it predicts the signal strength without taking the surrounding environment into account, such as the walls and the floors. Therefore, it becomes inaccurate in cases where there is no line of sight between sensors. Different models have been proposed to overcome this problem, such as modified versions of the log-distance model [12], in which the attenuations due to floors and walls are explicitly included. Several researchers have determined a mathematical relationship between RSSIs and distances, without taking physical properties into consideration [13]. To this end, an empirical model is estimated with a polynomial regression.

Once the distances are estimated from the RSSIs, they are then combined using for instance triangulation [14] or trilateration [15] to find the sensors' positions. One can also use methods based on filtering, since they can help in smoothing the target's trajectory, reducing thus the estimation error. One widely used filter is the Kalman filter [16]. However, it is known that this filter is only reliable for systems which are almost linear. To overcome the non-linear estimation problem, the extended Kalman filter (EKF) and unscented Kalman filter (UKF) have been proposed. A tracking approach using the EKF with the distances and acceleration information is proposed in [17]. The main disadvantage of such an approach is that the linearization and the approximations made might lead to sub-optimal performance and sometimes divergence of the filter [18]. Another tracking approach using the Kalman filter is described in [19], where the authors first find the target's positions using the weighted K-nearest neighbor (WKNN) algorithm. Then, they combine these estimates with acceleration information using the Kalman filter with a second-order state-space model. The authors of [20] also propose a tracking method that uses the Kalman filter with a third-order state-space model to enhance the position of the target estimated using a kernel-based model proposed by the authors.

Another well-known filter used for tracking is the particle filter (PF) [21]. The tracking methods described in [22], [23] take advantage of such filter, by using the target's motion and the distances as well.

In this paper, we propose two original regression models that estimate the distances separating sensors using their received signal strength indicators (RSSIs), in an indoor wireless sensor network. We then propose to solve the tracking problem using two new methods. Our approach consists first of a collection phase, aiming at building a training database composed of RSSI measures and distances based on the radio-fingerprinting concept [24]. The collection phase is then followed by an offline training phase where the distance models are defined using the collected database. Defining the distance models is an essential step in the tracking process; in fact, in the following, the estimated distances obtained using one of the proposed distance models will be used to determine the target's position. The first proposed distance model is a non-parametric model without any prior knowledge on the physical system, while the second one is a semi-parametric regression model, combining the well-known log-distance theoretical propagation model with a non-linear fluctuation term, estimated within the framework of kernel-based machines. Each model takes the RSSIs as input and yields the distance to the corresponding sensor as output. To the best of our knowledge, both models have not yet been proposed in the literature. Then, in the online tracking phase, a moving target collects RSSI measures and uses them with the computed kernel-based models to estimate the distances separating it from the sensors having known locations. Based on these distances, the target's position is estimated using two different methods. The first one consists of combining these distance estimates with the acceleration information, by means of a Kalman filter [25], [16], to determine the target's position. The second one consists of localizing the target using a particle filter [21] with the distance estimates and the acceleration information. Compared to [26] and [27], the proposed methods take advantage of the target's mobility to enhance the obtained position estimate. Compared to [20], where a kernel-based position model is used, we propose in this paper two kernel-based distance models to find the distances separating the target from the sensors. Because of the logarithmic relationship between the RSSIs and the distances, the use of distance models gives more relevant results than position models. Moreover, the most important reason for switching to distance models is that, by doing so, the sensors are allowed to move in the network, without affecting the tracking as it will be shown throughout this paper.

The rest of the paper is organized as follows. The proposed tracking methods are described in Section II. In Section III, we define the two kernel-based distance models using machine learning. In Section IV, we solve the tracking problem using the distances and acceleration information by means of the Kalman filter, while in Section V, we find the target's positions using the particle filter. In Section VI, we examine the performance of the two proposed distance models, and in Section VII, we evaluate the proposed tracking methods, while providing a comparison to state-of-the-art tracking methods. Finally, Section VIII concludes the paper.

II. Description of the tracking approach

Consider an environment of δ dimensions, with for instance $\delta = 2$ for a two-dimensional environment, and N_s sensors having known locations. The sensors are allowed to move in the network, and their locations are considered to be known at any given time step. For simplicity, we denote the sensors' locations by $\boldsymbol{s}_i = (s_{i,1} \ldots s_{i,\delta})^\top$, $i \in \{1, \ldots, N_s\}$. In the collection phase, in order to construct a radio-fingerprints database, N reference positions, denoted by $\boldsymbol{p}_\ell = (p_{\ell,1} \ldots p_{\ell,\delta})^\top, \ell \in \{1, \ldots, N\}$, are either generated on a uniform grid or randomly in the studied environment. The sensors continuously broadcast signals in the network at a fixed initial power, and a sensor is placed consecutively at the reference positions to detect the broadcasted signals and measure their RSSIs. Let $\rho_{\boldsymbol{s}_i, \boldsymbol{p}_\ell}$, $i \in \{1, \ldots, N_s\}$, denote the power of the signal received from the sensor at position $\boldsymbol{s}_i$ by the sensor at reference position $\boldsymbol{p}_\ell$, and let $d_{\boldsymbol{s}_i, \boldsymbol{p}_\ell}$, $i \in \{1, \ldots, N_s\}$, denote the distance separating these sensors. The distance $d_{\boldsymbol{s}_i, \boldsymbol{p}_\ell}$ is then given by:

$$d_{\boldsymbol{s}_i, \boldsymbol{p}_\ell} = \sqrt{\sum_{v=1}^{\delta} \left(s_{i,v} - p_{\ell,v} \right)^2}. \tag{1}$$

In this way, N_s training sets of N pairs $(\rho_{\boldsymbol{s}_i, \boldsymbol{p}_\ell}, d_{\boldsymbol{s}_i, \boldsymbol{p}_\ell})$ are obtained. The training phase consists then in finding a set of N_s models $\chi_i \colon \mathbb{R} \to \mathbb{R}$, $i \in \{1, \ldots, N_s\}$; each model takes as input the RSSI received from the sensor $\boldsymbol{s}_i$ at a reference position $\boldsymbol{p}_\ell, \ell \in \{1, \ldots, N\}$, and yields as output the distance $d_{\boldsymbol{s}_i, \boldsymbol{p}_\ell}$ separating $\boldsymbol{s}_i$ from $\boldsymbol{p}_\ell$. These distance models can be non-parametric or semi-parametric; details will be provided in the following section. Note that the database construction and the computation of the models χ_i are performed only once at a computation center, before the tracking phase, regardless of the chosen model. The models are then communicated to the target, that performs all the following computations.

Let $\boldsymbol{x}(k) = (x_1(k) \ldots x_\delta(k))^\top$ denote the target's position at a given time step k. Then, in the tracking phase, the moving target collects the RSSIs of the signals received from the N_s sensors, at time step k, and stores them into the vector $\boldsymbol{\rho}_{\boldsymbol{x}(k)} = (\rho_{\boldsymbol{s}_1, \boldsymbol{x}(k)} \cdots \rho_{\boldsymbol{s}_{N_s}, \boldsymbol{x}(k)})^\top$. An estimate of the distance $\widehat{d}_{\boldsymbol{s}_i, \boldsymbol{x}(k)}, i \in \{1, \ldots, N_s\}$, separating the sensor at position $\boldsymbol{s}_i$ from the target is then obtained using the already-defined distance model χ_i, as follows:

$$\widehat{d}_{\boldsymbol{s}_i, \boldsymbol{x}(k)} = \chi_i(\rho_{\boldsymbol{s}_i, \boldsymbol{x}(k)}). \tag{2}$$

The target is assumed to be equipped with a gyroscope and an accelerometer that yield respectively its instantaneous orientation and accelerations. Combining this information leads to the instantaneous accelerations of the target over the δ coordinates in the network coordinates system. We propose then to solve the positioning problem in two ways. The first solution is given in Section IV, where the distance estimates are combined with the accelerometer information to obtain an estimate of the target's positions using the Kalman filter. The second solution is given in Section V, where the target's position estimates are obtained by combining the accelerations and the distance estimates using the particle filter.

III. DEFINITION OF χ_i USING KERNEL METHODS

The objective of this section is to define the set of models χ_i, $i \in \{1, \ldots, N_s\}$, using the constructed training database. Each model χ_i is defined in such a way that it associates to each RSSI measure $\rho_{\boldsymbol{s}_i, \boldsymbol{p}_\ell}$, $\ell \in \{1, \ldots, N\}$, the corresponding distance $d_{\boldsymbol{s}_i, \boldsymbol{p}_\ell}$. To this end, we propose two distance models. In the first subsection, we define the set of models χ_i, $i \in \{1, \ldots, N_s\}$, as non-parametric models, i.e., they are taken as linear combinations of kernels centered on the training samples. The second models, described in the second paragraph of this section, are semi-parametric regression models that combine the physical log-distance propagation model of [9], [10] with a non-linear fluctuation term, defined in a reproducing kernel Hilbert space. This non-linear term compensates for the missing factors in the log-distance model, therefore allowing a better modeling of the RSSI/distance relationship.

A. Non-parametric regression models

In this paragraph, the distance model is derived using only the training database. No assumptions on the model or prior knowledge of the model are considered. Therefore, determining χ_i requires solving a non-linear regression problem. We take advantage of kernel methods, that have been remarkably successful for solving non-linear regression problems. To this end, we consider the kernel ridge regression [28] to determine the functions χ_i.

Each function χ_i is determined by minimizing the mean quadratic error between the model's outputs $\chi_i(\rho_{\boldsymbol{s}_i, \boldsymbol{p}_\ell})$ and the desired outputs $d_{\boldsymbol{s}_i, \boldsymbol{p}_\ell}$, as follows:

$$\min_{\chi_i \in \mathcal{H}} \frac{1}{N} \sum_{\ell=1}^{N} (d_{\boldsymbol{s}_i, \boldsymbol{p}_\ell} - \chi_i(\rho_{\boldsymbol{s}_i, \boldsymbol{p}_\ell}))^2 + \eta_i \|\chi_i\|_{\mathcal{H}}^2, \quad (3)$$

where $\mathcal{H}$ is a reproducing kernel Hilbert space, and η_i is a regularization parameter that controls the tradeoff between the training error and the complexity of the solution. According to the representer theorem [29], the optimal function can be written as follows:

$$\chi_i(\cdot) = \sum_{\ell=1}^{N} \alpha_{\ell,i} \, \kappa(\rho_{\boldsymbol{s}_i, \boldsymbol{p}_\ell}, \cdot), \quad (4)$$

where " $\cdot$ " is the function's input, $\kappa : \mathbb{R} \times \mathbb{R} \mapsto \mathbb{R}$ is a reproducing kernel, and $\alpha_{\ell,i}, \ell \in \{1, \ldots, N\}$, are parameters to be determined. Let $\boldsymbol{\alpha}_i$ be the $N \times 1$ vector whose ℓ-th entry is $\alpha_{\ell,i}$. By injecting (4) in (3), a dual optimization problem in terms of $\boldsymbol{\alpha}_i$ is obtained, whose solution is given by taking the derivative of the corresponding cost function with respect to $\boldsymbol{\alpha}_i$ and setting it to zero. One can easily find the following form of the solution:

$$\boldsymbol{\alpha}_i = (\boldsymbol{K}_i + \eta_i N \mathbf{I}_N)^{-1} \boldsymbol{d}_{\boldsymbol{s}_i}, \quad (5)$$

where $\mathbf{I}_N$ is the $N \times N$ identity matrix, $\boldsymbol{d}_{\boldsymbol{s}_i} = (d_{\boldsymbol{s}_i, \boldsymbol{p}_1} \ldots d_{\boldsymbol{s}_i, \boldsymbol{p}_N})^\top$, $i \in \{1, \ldots, N_s\}$, and $\boldsymbol{K}_i$ is the $N \times N$ matrix whose (u, u')-th entry is $\kappa(\rho_{\boldsymbol{s}_i, \boldsymbol{p}_u}, \rho_{\boldsymbol{s}_i, \boldsymbol{p}_{u'}})$, for $u, u' \in$ $\{1, \ldots, N\}$. For appropriate values of the regularization parameter η_i, the matrix between parentheses is always non-singular.

Notice that this non-parametric model does not consider any prior knowledge on the physical properties of the relationship between the RSSIs and the distances. Indeed, the model does not take into account the fact that this RSSI/distance relationship is logarithmic. In the following subsection, we propose a semi-parametric regression model in such a way that the prior knowledge of the logarithmic RSSI/distance relationship is taken into consideration.

B. Semi-parametric regression models

As we already stated, the theoretical log-distance signal propagation model alone is insufficient to characterize the RSSI/distance relationship, since it neglects all other factors in the environment, such as physical obstacles, multi-path propagation, additive noises, etc. This model characterizes the RSSI/distance relationship as follows:

$$\rho_{\boldsymbol{s}_i, \boldsymbol{p}_\ell} = \rho_{0i} - 10 \, n_{\mathrm{P}} \log_{10} \frac{d_{\boldsymbol{s}_i, \boldsymbol{p}_\ell}}{d_{0i}}, \quad (6)$$

where $\rho_{\boldsymbol{s}_i, \boldsymbol{p}_\ell}$ is the power received from the sensor at position $\boldsymbol{s}_i$ by the node at position $\boldsymbol{p}_\ell$, that is the i-th entry of the vector $\boldsymbol{\rho}_\ell$, ρ_{0i} is the power at the reference distance d_{0i}, and n_{P} is the path-loss exponent. In this paragraph, we propose a new propagation model by adding to this model a non-linear fluctuation term φ_i, that represents a combination of all unknown factors affecting the RSSI measures, as follows:

$$\rho_{\boldsymbol{s}_i, \boldsymbol{p}_\ell} = \rho_{0i} - 10 \, n_{\mathrm{P}} \log_{10} \frac{d_{\boldsymbol{s}_i, \boldsymbol{p}_\ell}}{d_{0i}} + \varphi_i, \quad (7)$$

This term provides the log-distance physical model with further flexibility, resulting in a more accurate model. Equation (7) can be written as follows:

$$\log_{10} d_{\boldsymbol{s}_i, \boldsymbol{p}_\ell} = \frac{\rho_{0i}}{10 \, n_{\mathrm{P}}} + \log_{10} d_{0i} - \frac{\rho_{\boldsymbol{s}_i, \boldsymbol{p}_\ell}}{10 \, n_{\mathrm{P}}} + \frac{\varphi_i}{10 \, n_{\mathrm{P}}}. \quad (8)$$

One can see that (8) is a combination of a linear model in terms of $\rho_{\boldsymbol{s}_i, \boldsymbol{p}_\ell}$ and a non-linear model. Let $\psi_i(\cdot)$ denote the model that associates to each RSSI $\rho_{\boldsymbol{s}_i, \boldsymbol{p}_\ell}$ the logarithm of the distance $\log_{10}(d_{\boldsymbol{s}_i, \boldsymbol{p}_\ell})$. This model $\psi_i(\cdot)$ can be decomposed into two terms: a linear term and a non-linear fluctuation term, such that:

$$\psi_i(\cdot) = \psi_{i,\mathrm{lin}}(\cdot) + \psi_{i,\mathrm{nlin}}(\cdot),$$

where we have the following:

$$\begin{cases} \psi_{i,\mathrm{lin}}(\rho_{\boldsymbol{s}_i, \boldsymbol{p}_\ell}) = a_i \, \rho_{\boldsymbol{s}_i, \boldsymbol{p}_\ell} + b_i, \\ \psi_{i,\mathrm{nlin}}(\rho_{\boldsymbol{s}_i, \boldsymbol{p}_\ell}) = \frac{\varphi_i}{10 \, n_{\mathrm{P}}}, \end{cases} \quad (9)$$

with $a_i = -\frac{1}{10 \, n_{\mathrm{P}}}$ and $b_i = \frac{\rho_{0i}}{10 \, n_{\mathrm{P}}} + \log_{10} d_{0i}$.

We now consider that the non-linear term, given by $\psi_{i,\mathrm{nlin}}(\cdot)$, lies in a reproducing kernel Hilbert space denoted by $\mathcal{H}$ and induced by a positive definite kernel function $\kappa(\cdot, \cdot)$. It is

estimated by minimizing the regularized mean quadratic error on the training set, given by:

$$\frac{1}{N} \sum_{\ell=1}^{N} \varepsilon_\ell^2 + \eta_i \|\psi_{i,nlin}\|_\mathcal{H}^2, \tag{10}$$

where the quantity η_i is a regularization parameter that controls the tradeoff between the training error and the complexity of the solution, and the cost function $\varepsilon_{\ell,i}$ is given by:

$$\varepsilon_{\ell,i} = \log_{10} d_\ell - a_i \rho_{\boldsymbol{s}_i,\boldsymbol{p}_\ell} - b_i - \psi_{i,\text{nlin}}(\rho_{\boldsymbol{s}_i,\boldsymbol{p}_\ell}),$$

where $\ell \in \{1,\ldots,N\}$. From the semi-parametric representer's theorem [29], $\psi_{i,\text{nlin}}(\cdot)$ can be written as a linear combination of kernels:

$$\psi_{i,\text{nlin}}(\cdot) = \sum_{\ell=1}^{N} \beta_{\ell,i}\, \kappa(\rho_{\boldsymbol{s}_i,\boldsymbol{p}_\ell}, \cdot),$$

where $\kappa : \mathbb{R} \times \mathbb{R} \mapsto \mathbb{R}$, and $\beta_{\ell,i}$, with $\ell \in \{1,\ldots,N\}$, are parameters to be estimated.

By multiplying (10) by N for convenience and writing it in matrix form, one gets the following:

$$\begin{aligned}
&\boldsymbol{\Lambda}_i^\top \boldsymbol{\Lambda}_i + a_i^2\, \boldsymbol{\rho}_i^\top \boldsymbol{\rho}_i + Nb_i^2 + \boldsymbol{\beta}_i^\top \boldsymbol{K}_i^\top \boldsymbol{K}_i \boldsymbol{\beta}_i \\
&- 2a_i\boldsymbol{\Lambda}_i^\top \boldsymbol{\rho}_i - 2b_i \mathbf{1}^\top \boldsymbol{\Lambda}_i - 2\boldsymbol{\Lambda}_i^\top \boldsymbol{K}_i \boldsymbol{\beta}_i + 2a_ib_i \mathbf{1}^\top \boldsymbol{\rho}_i \\
&+ 2a_i\boldsymbol{\rho}_i^\top \boldsymbol{K}_i \boldsymbol{\beta}_i + 2b_i \mathbf{1}^\top \boldsymbol{\beta}_i \\
&+ \eta_i N \boldsymbol{\beta}_i^\top \boldsymbol{K}_i \boldsymbol{\beta}_i, \tag{11}
\end{aligned}$$

where $\boldsymbol{\beta}_i = (\beta_{1,i} \ldots \beta_{N,i})^\top$, $\boldsymbol{K}_i$ is the $N \times N$ matrix whose (u,u')-th entry is $\kappa(\rho_{\boldsymbol{s}_i,\boldsymbol{p}_u}, \rho_{\boldsymbol{s}_i,\boldsymbol{p}_{u'}})$, for $u,u' \in \{1,...,N\}$, $\boldsymbol{\rho}_i$ is the $N \times 1$ vector whose ℓ-th entry is equal to $\rho_{\boldsymbol{s}_i,\boldsymbol{p}_\ell}$, with $\ell \in \{1,\ldots,N\}$, $\mathbf{1}$ is the vector of ones of appropriate size, and $\boldsymbol{\Lambda}_i$ is the $N \times 1$ vector whose ℓ-th entry is equal to $\log_{10} d_{\boldsymbol{s}_i,\boldsymbol{p}_\ell}$.

By taking the partial derivatives in matrix form of (11) with respect to a_i, b_i and $\boldsymbol{\beta}_i$ and setting them to zero, we get the following linear system having the form $\boldsymbol{U}_i = \boldsymbol{W}_i \boldsymbol{Y}_i$, where

$$\boldsymbol{Y}_i = \begin{bmatrix} a_i \\ b_i \\ \boldsymbol{\beta}_i \end{bmatrix}, \boldsymbol{U}_i = \begin{bmatrix} \mathbf{1}^\top \boldsymbol{\Lambda}_i \\ \boldsymbol{\Lambda}_i \\ \boldsymbol{\Lambda}_i^\top \boldsymbol{\rho}_i \end{bmatrix},$$

$$\boldsymbol{W}_i = \begin{bmatrix} \mathbf{1}^\top \boldsymbol{\rho}_i & N & \mathbf{1}^\top \boldsymbol{K}_i \\ \boldsymbol{\rho}_i & \mathbf{1} & \boldsymbol{K}_i + \eta_i N \mathbf{I}_N \\ \boldsymbol{\rho}_i^\top \boldsymbol{\rho}_i & \mathbf{1}^\top \boldsymbol{\rho}_i & \boldsymbol{\rho}_i^\top \boldsymbol{K}_i \end{bmatrix} \tag{12}$$

The solution is then given by:

$$\boldsymbol{Y}_i = (\boldsymbol{W}_i^\top \boldsymbol{W}_i)^{-1} \boldsymbol{W}_i^\top \boldsymbol{U}_i. \tag{13}$$

After computing the model's parameters using (13), one can find the logarithm of any distance separating a sensor at position $\boldsymbol{s}_i$ and the target at position $\boldsymbol{x}(k)$ in the network using only the RSSI information, as follows:

$$\begin{aligned}
\log_{10} d_{\boldsymbol{s}_i,\boldsymbol{x}(k)} &= \psi_i(\rho_{\boldsymbol{s}_i,\boldsymbol{x}(k)}) \\
&= a_i \rho_{\boldsymbol{s}_i,\boldsymbol{p}_\ell} + b_i + \sum_{\ell=1}^{N} \beta_\ell\, \kappa(\rho_{\boldsymbol{s}_i,\boldsymbol{p}_\ell}, \rho_{\boldsymbol{s}_i,\boldsymbol{x}(k)}).
\end{aligned}$$

Then, the model $\chi_i(\cdot)$ yields the distance as follows:

$$\chi_i(\rho_{\boldsymbol{s}_i,\boldsymbol{x}(k)}) = 10^{\psi_i(\rho_{\boldsymbol{s}_i,\boldsymbol{x}(k)})}. \tag{14}$$

IV. TRACKING USING THE KALMAN FILTER

In this section, we propose to find the target's position by combining its instantaneous accelerations with the distances separating it from the sensors in the network, by means of a Kalman filter. These distances are estimated using either one of the kernel-based distance models defined in Section III The first and most challenging part is to find a model that describes the target's motion. To overcome cases where the target does not follow a predictable path, we propose to use a third-order state-space model. This model is based on the assumption that the target's accelerations vary linearly between any two consecutive time steps $k-1$ and k with a slope equal to $\frac{\gamma(k)-\gamma(k-1)}{\Delta t}$, where $\boldsymbol{\gamma}(k) = (\gamma_1(k)\ldots\gamma_\delta(k))^\top$ is the acceleration vector at time step k, and Δt the time period separating two consecutive time steps. Accordingly, one can estimate recursively the velocity vector $\boldsymbol{\nu}(k)$ of the target at time step k as follows:

$$\boldsymbol{\nu}(k) = \boldsymbol{\nu}(k-1) + \boldsymbol{\gamma}(k-1)\,\Delta t + \frac{\boldsymbol{\gamma}(k) - \boldsymbol{\gamma}(k-1)}{\Delta t}\, \frac{\Delta t^2}{2}. \tag{15}$$

Then, by taking the primitive integral of (15), the position of the target can be written as follows:

$$\begin{aligned}
\boldsymbol{x}(k) = \boldsymbol{x}(k-1) + \boldsymbol{\nu}(k-1)\,\Delta t + \boldsymbol{\gamma}(k-1)\frac{\Delta t^2}{2} \\
+ \frac{\boldsymbol{\gamma}(k) - \boldsymbol{\gamma}(k-1)}{\Delta t}\, \frac{\Delta t^3}{6}. \tag{16}
\end{aligned}$$

Assuming that the target is at a fixed known position $\boldsymbol{x}(0)$ at the beginning of the tracking and having $\boldsymbol{\nu}(0)$ and $\boldsymbol{\gamma}(0)$ null, one can then find $\boldsymbol{x}(k)$ at any time step k using the measured accelerations. This model yields an accurate approximation of $\boldsymbol{x}(k)$ for small values of Δt.

Now let $\boldsymbol{X}(k) = (\boldsymbol{x}(k)\ \ \boldsymbol{\nu}(k))^\top$ denote the unknown state of the target at time step k. The considered state-space equation is then given by the following:

$$\boldsymbol{X}(k) = \boldsymbol{A}\ \boldsymbol{X}(k-1) + \boldsymbol{B}(k) + \boldsymbol{\epsilon}_{\text{KF}}(k), \tag{17}$$

where $\boldsymbol{A}$ is the state transition matrix and $\boldsymbol{B}(k)$ is a noisy control-input vector depending on the accelerations given by:

$$\boldsymbol{A} = \begin{pmatrix} \mathbf{I}_\delta & \Delta t \mathbf{I}_\delta \\ \mathbf{0} & \mathbf{I}_\delta \end{pmatrix}$$

$$\boldsymbol{B}(k) = \begin{pmatrix} \boldsymbol{\gamma}(k-1)^\top \frac{\Delta t^2}{3} \\ \boldsymbol{\gamma}(k-1)^\top \frac{\Delta t}{2} \end{pmatrix} + \begin{pmatrix} \boldsymbol{\gamma}(k)^\top \frac{\Delta t^2}{6} \\ \boldsymbol{\gamma}(k)^\top \frac{\Delta t}{2} \end{pmatrix},$$

where $\mathbf{I}_\delta$ is the $\delta \times \delta$ identity matrix, and $\mathbf{0}$ is the matrix of zeros of appropriate size. As for the quantity $\boldsymbol{\epsilon}_{\text{KF}}(k)$, it is the state equation error, whose probability distribution is assumed to be normal, having zero mean and covariance matrix $\boldsymbol{V}$ of size $2\delta \times 2\delta$.

Remark. In the previous paragraphs, the target is assumed to be rotationally constrained. However, during its motion in real applications, the target could rotate, leading to changes in the coordinate system, where the accelerations are given. The solution to this problem is to equip each target with a gyroscope, which yields its orientations with respect to the world coordinate system. At each time step, the target

measures its acceleration vector in its coordinate system, then finds its orientations using the gyroscope. Its accelerations in the world coordinate system can then be computed using the rotation matrix given in [20]. For simplicity, we only consider rotationally constrained targets in our paper. However, as just explained, the computations could be easily modified to consider the targets rotations and find the accelerations in the world coordinate system.

Having defined the state-space model, we now need to define the observation in order to use the Kalman filter. However, by using the equations of the estimated distances separating the target from the sensors, the obtained observation model is non-linear. To overcome this problem, we propose in this paper an alternative linear observation model based on the distances. Indeed, using (1), one can write the following:

$$d_{\boldsymbol{s}_i,\boldsymbol{x}(k)}^2 = \sum_{v=1}^{\delta}(s_{i,v} - x_v(k))^2$$
$$= \sum_{v=1}^{\delta} s_{i,v}^2 + \sum_{v=1}^{\delta} x_v^2(k) - 2\sum_{v=1}^{\delta} s_{i,v}\, x_v(k).$$

By subtracting $d_{\boldsymbol{s}_{N_s},\boldsymbol{x}(k)}^2$ from $d_{\boldsymbol{s}_i,\boldsymbol{x}(k)}^2$, $i \in \{1,...,N_s - 1\}$, we get the following:

$$d_{\boldsymbol{s}_i,\boldsymbol{x}(k)}^2 - d_{\boldsymbol{s}_{N_s},\boldsymbol{x}(k)}^2 = \sum_{v=1}^{\delta} s_{i,v}^2 - \sum_{v=1}^{\delta} s_{N_s,v}^2$$
$$+ 2\sum_{v=1}^{\delta} x_v(k)(s_{N_s,v} - s_{i,v}).$$

After rearranging the terms in this expression, it becomes possible to set a linear observation model as follows:

$$\boldsymbol{z}_{\mathrm{KF}}(k) = \boldsymbol{C}\boldsymbol{X}(k) + \boldsymbol{n}(k), \qquad (18)$$

where $\boldsymbol{n}(k) \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{R})$ is an additive noise, assumed to have a normal distribution, with zero mean and covariance matrix $\boldsymbol{R}$, and the observation vector $\boldsymbol{z}_{\mathrm{KF}}(k)$ can be written as follows:

$$\begin{pmatrix} d_{\boldsymbol{s}_1,\boldsymbol{x}(k)}^2 - d_{\boldsymbol{s}_{N_s},\boldsymbol{x}(k)}^2 + \sum_{v=1}^{\delta} s_{N_s,v}^2 - \sum_{v=1}^{\delta} s_{1,v}^2 \\ \vdots \\ d_{\boldsymbol{s}_{N_s-1},\boldsymbol{x}(k)}^2 - d_{\boldsymbol{s}_{N_s},\boldsymbol{x}(k)}^2 + \sum_{v=1}^{\delta} s_{N_s,v}^2 - \sum_{v=1}^{\delta} s_{N_s-1,v}^2 \end{pmatrix}$$

The observation matrix $\boldsymbol{C}$ is then given by:

$$2\begin{pmatrix} s_{N_s,1} - s_{1,1} & \cdots & s_{N_s,\delta} - s_{1,\delta} \\ \vdots & & \vdots & \boldsymbol{0}_{(N_s-1)\times\delta} \\ s_{N_s,1} - s_{N_s-1,1} & \cdots & s_{N_s,\delta} - s_{N_s-1,\delta} \end{pmatrix},$$

To approximate $\boldsymbol{R}$, a new set of reference positions is generated, and their distances to the sensors are estimated using χ_i. Then, the differences between their observation vectors, using their estimated distances, and the product of the matrix $\boldsymbol{C}$ by their exact positions are computed and stored into a vector, that is the error vector. The matrix $\boldsymbol{R}$, assumed to be constant over time, is then determined by computing the covariance of the error vector.

Now that we have defined the two linear equations (17) and (18), we can use the Kalman filter to find the target's position at each time step k. The Kalman filter first predicts the unknown position using the previous estimated position and the state-space equation (17), as follows:

$$\hat{\boldsymbol{X}}^-(k) = \boldsymbol{A}\,\hat{\boldsymbol{X}}(k-1) + \boldsymbol{B}(k),$$

where $\hat{\boldsymbol{X}}(0)$ is assumed to be known. Then, the predicted estimation covariance is updated by the following:

$$\boldsymbol{T}^-(k) = \boldsymbol{A}\,\boldsymbol{T}(k-1)\boldsymbol{A}^\top + \boldsymbol{Q}(k),$$

where $\boldsymbol{T}(k-1)$ is the covariance estimation at time step $k-1$, and $\boldsymbol{T}(0)$ is null since the initial state is known. The matrix $\boldsymbol{Q}(k)$ is the covariance matrix of $\boldsymbol{X}(k)$ given $\boldsymbol{X}(k-1)$, namely

$$\boldsymbol{Q}(k) = \mathrm{Cov}(\boldsymbol{X}(k)|\boldsymbol{X}(k-1) = \mathrm{Cov}(\boldsymbol{B}(k) + \boldsymbol{\epsilon}(k))$$
$$= \mathrm{Cov}(\boldsymbol{B}(k)) + \mathrm{Cov}(\boldsymbol{\epsilon}(k))$$
$$= \mathrm{Cov}(\boldsymbol{B}(k)) + \boldsymbol{V},$$

where $\mathrm{Cov}(\cdot)$ computes the covariance matrix of its argument. Each coordinate acceleration noise is assumed to be independent with zero-mean normal distribution, having known variances $\sigma_{\gamma,d}^2$, $d = 1,...,\delta$. Their values can be estimated by performing a calibration of the accelerometer before the tracking stage. We also assume that all the δ coordinates are statistically independent. The matrix $\boldsymbol{Q}(k)$ is then given by:

$$\boldsymbol{Q}(k) = \boldsymbol{V} + \begin{pmatrix} \frac{5\Delta t^4}{36}\,\mathrm{Diag}\left(\boldsymbol{\sigma}_\gamma^2\right) & \boldsymbol{0} \\ \boldsymbol{0} & \frac{\Delta t^2}{2}\,\mathrm{Diag}\left(\boldsymbol{\sigma}_\gamma^2\right) \end{pmatrix}.$$

The final step of the filter is the correction of the predicted quantities $\widehat{\boldsymbol{X}}^-(k)$ and $\boldsymbol{T}^-(k)$ using the observation equation (18) as follows:

$$\widehat{\boldsymbol{X}}(k) = \widehat{\boldsymbol{X}}^-(k) + \boldsymbol{G}_{\mathrm{KF}}(k)\left(\boldsymbol{z}_{\mathrm{KF}}(k)^\top - \boldsymbol{C}\widehat{\boldsymbol{X}}^-(k)\right),$$
$$\boldsymbol{T}(k) = (\mathbf{I}_{2\delta} - \boldsymbol{G}_{\mathrm{KF}}(k)\,\boldsymbol{C})\,\boldsymbol{T}^-(k),$$

where $\boldsymbol{G}_{\mathrm{KF}}(k)$ is the optimal Kalman gain given by:

$$\boldsymbol{G}_{\mathrm{KF}}(k) = \boldsymbol{T}^-(k)\,\boldsymbol{C}^\top\left(\boldsymbol{C}\,\boldsymbol{T}^-(k)\,\boldsymbol{C}^\top + \boldsymbol{R}\right)^{-1}.$$

V. TRACKING USING THE PARTICLE FILTER

The objective in this section is to find the target's position estimates using the particle filter, the distances estimated by either one of the kernel-based distance models of Section III and the acceleration information of the target. Unlike the Kalman filter, the particle filter makes no restrictive assumption about the dynamics of the state-space or the density function. The observation model can be non-linear, and the initial state and noise distributions can take any required form. For this reason, the observations here are taken equal to the estimated distances separating the target from the sensors, that is:

$$\boldsymbol{z}_{\mathrm{PF}}(k) = \boldsymbol{d}_{x(k)},$$

where $\boldsymbol{d}_{x(k)} = (d_{\boldsymbol{s}_1,\boldsymbol{x}(k)}\ldots d_{\boldsymbol{s}_{N_s},\boldsymbol{x}(k)})^\top$, obtained using the kernel-based model. Like the Kalman-based method, the tracking problem is also defined by the third-order state-space model. In order to solve the problem using the particle filter,

N_{PF} particles are first generated at the initial position of the target $\boldsymbol{x}(0)$, considered to be known. All the particles are given equal weights of $\frac{1}{N_{\mathrm{PF}}}$. It is important to note here that the higher N_{PF} is, the more computations we need. Let $\boldsymbol{x}^m(0), m \in \{1, \ldots, N_{\mathrm{PF}}\}$, denote the initial particles positions, and $\boldsymbol{\nu}^m(0)$ their initial null velocities; then the particles positions at time step k are denoted by $\boldsymbol{x}^m(k), m \in \{1, \ldots, N_{\mathrm{PF}}\}$, and their velocities by $\boldsymbol{\nu}^m(k)$. Starting from the initial position, the particles positions and their velocities are recursively predicted using (15) and (16) as follows:

$$\boldsymbol{\nu}^m(k) = \boldsymbol{\nu}^{m'}(k-1) + \boldsymbol{\gamma}(k-1)\,\Delta t + \frac{\boldsymbol{\gamma}(k) - \boldsymbol{\gamma}(k-1)}{\Delta t}\,\frac{\Delta t^2}{2},$$

$$\boldsymbol{x}^m(k) = \boldsymbol{x}^{m'}(k-1) + \boldsymbol{\nu}^{m'}(k-1)\,\Delta t + \boldsymbol{\gamma}(k-1)\,\frac{\Delta t^2}{2}$$
$$+ \frac{\boldsymbol{\gamma}(k) - \boldsymbol{\gamma}(k-1)}{\Delta t}\,\frac{\Delta t^3}{6} + \boldsymbol{\epsilon}_{\mathrm{PF}}(k),$$

where $\boldsymbol{x}^{m'}(k-1)$ is one of the particles of time step $k-1$, selected randomly according to the discrete distribution of their weights, $\boldsymbol{\nu}^{m'}(k-1)$ is the velocity associated to $\boldsymbol{x}^{m'}(k-1)$, and $\boldsymbol{\epsilon}_{\mathrm{PF}}(k) \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{H})$ is the state equation error whose probability distribution is assumed to be normal, having zero mean and covariance matrix $\boldsymbol{H}$ of size $\delta \times \delta$. The particles' weights are updated according to the observed distances. Indeed, a particle is more valuable if its distances to the sensors are closer to the observed ones [23]. This update rule is formulated as follows:

$$w^m(k) = \frac{1}{\|\boldsymbol{d}_{\boldsymbol{x}^m(k)} - \boldsymbol{z}_{\mathrm{PF}}(k)\|}\, w^{m'}(k-1),$$

where $\boldsymbol{d}_{\boldsymbol{x}^m(k)} = (d_{\boldsymbol{s}_1, \boldsymbol{x}^m(k)} \ldots d_{\boldsymbol{s}_{N_s}, \boldsymbol{x}^m(k)})^\top$, and $\|\boldsymbol{d}_{\boldsymbol{x}^m(k)} - \boldsymbol{z}_{\mathrm{PF}}(k)\|$ is the distance between the vectors $\boldsymbol{z}_{\mathrm{PF}}(k)$ and $\boldsymbol{d}_{\boldsymbol{x}^m(k)}$. Then, the weights are normalized using the following:

$$w^m(k) = \frac{w^m(k)}{\sum_{m=1}^{N_{\mathrm{PF}}} w^m(k)}.$$

Finally, the last step in the filtering process is the resampling. In this step, the particles with negligible weights are replaced by new particles in the proximity of the particles with higher weights. It is applied only when the effective number of particles N_{eff} becomes less than a threshold value N_{th}, with N_{eff} given by:

$$N_{\mathrm{eff}} = \frac{1}{\sum_{m=1}^{N_{\mathrm{PF}}} (w^m(k))^2}.$$

The threshold N_{th} is usually taken equal to 10% of the particles number N_{PF}. Having calculated the weights of the particles, the target's position at time step k is then given using the following:

$$\widehat{\boldsymbol{x}}(k) = \sum_{m=1}^{N_{\mathrm{PF}}} \boldsymbol{x}^m(k)\, w^m(k).$$

Compared to the Kalman-based method, this algorithm needs more computations, due to the generation and resampling of the particles; however, it is more robust when the distances errors are non Gaussian.

VI. EVALUATION OF THE ACCURACY OF THE PROPOSED DISTANCE MODELS

We now propose to evaluate the accuracy of the proposed distance models, in the case of real data and simulated data. The proposed models are compared to the log-distance propagation model, and to the polynomial model introduced in [13]. The polynomial regression in [13] determines a mathematical relation between RSSIs and distances, without taking physical properties into consideration. In such a case, the signal propagation model is the q-th degree polynomial given by the following:

$$d_{\boldsymbol{s}_i, \boldsymbol{p}_\ell} = a_0 + a_1\, \rho_{\boldsymbol{s}_i, \boldsymbol{p}_\ell} + a_2\, \rho^2_{\boldsymbol{s}_i, \boldsymbol{p}_\ell} + \cdots + a_q\, \rho^q_{\boldsymbol{s}_i, \boldsymbol{p}_\ell}, \quad (19)$$

where a_j, $j \in \{0, \ldots, q\}$, are the polynomial's coefficients to be determined. These parameters a_j are chosen in a way to fit the q-th degree polynomial through the training set, using the least squares method.

A. Evaluation of the distance models on real data

To evaluate the proposed distance models, we use the set of collected measurements available from [30]. In this work, the measurements are carried out in a room of approximately 10 m $\times$ 10 m, where 48 uniformly distributed EyesIFX sensor nodes are deployed. Furniture and people in the room cause multi-path interferences affecting the collected RSSI values. Now, for our evaluation, we consider that 5 sensors over the 48 are denoted as fixed sensors. The 43 remaining are used to collect the measurements for the training phase. Fig. 1 shows the topology of the testbed. Note that it is the collected measurements that are needed in the test phase with the 5 fixed sensors, not the 43 remaining ones. It is also worth noting here that the training data for our algorithm could be collected by using one sensor, as explained in Section II, and this only once at the beginning of the experiments. Indeed, by placing consecutively one sensor over the reference positions, one can collect RSSI/distance pairs and construct the training database.

It is important to note that the average values over time of the RSSIs are used here. In fact, the RSSIs vary significantly with respect to time and movements. These variations are known as short-term or multi-path fading. On the other hand, the local average of the signal varies slowly. These slow fluctuations depend mostly on environmental characteristics, and they are known as long-term fading. Therefore, it is more suitable to use the average values of the RSSIs than to use all the collected values [31]. Now the objective is to find the RSSI/distance relationship; in other words, we need to find the three distance models, i.e., one model per sensor. Each model has a different set of training data to be used, and different parameters that need to be found. In fact, the RSSIs of the signals exchanged between each fixed sensor and the 43 other sensors are used along with the distances separating this sensor from the other sensors. This information is then used in the training phase as described in III to compute the models' parameters. In the following, we use the Gaussian kernel given by:

$$\kappa(\rho_{\boldsymbol{s}_i, \boldsymbol{p}_u}, \rho_{\boldsymbol{s}_i, \boldsymbol{p}_{u'}}) = \exp\left(-\frac{\|\rho_{\boldsymbol{s}_i, \boldsymbol{p}_u} - \rho_{\boldsymbol{s}_i, \boldsymbol{p}_{u'}}\|^2}{2\sigma_i^2} \right),$$
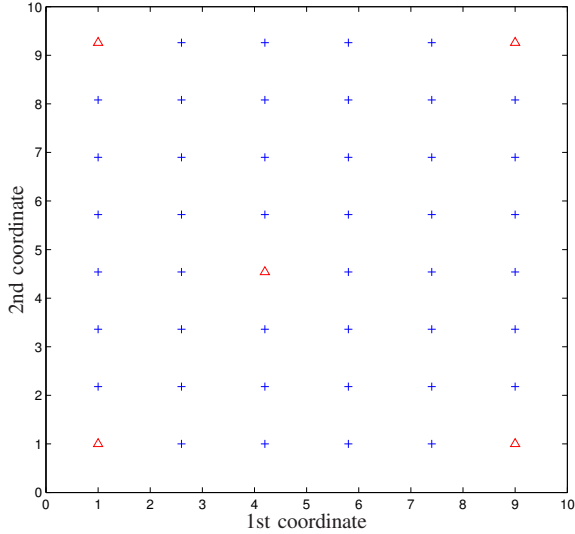
Fig. 1: Topology of the testbed (real data), where $\triangle$ represents the sensors and $+$ represents the training positions.

TABLE I: Comparison between models (error in meters) for real data.

Considered model	Mean training error	Mean LOO error
PM	1.36	1.41
PolyM, $q = 2$	1.30	1.40
PolyM, $q = 3$	1.29	1.48
PolyM, $q = 4$	1.29	1.74
NPM	1.21	1.33
SPM	1.17	1.33

TABLE II: Estimation errors for simulated data in the case of the first scenario.

Considered model	Mean training error	Mean test error
PM	2.26	2.59
PolyM, $q = 2$	2.74	2.94
PolyM, $q = 3$	2.23	2.57
PolyM, $q = 4$	2.22	2.60
NPM	2.18	2.57
SPM	2.09	2.59

where σ_i is its bandwidth that controls, together with the regularization parameter η_i, the degree of smoothness, noise tolerance, and generalization of the solution. The kernel parameters are chosen in a way to minimize the error on the training set. The value of this error for the computed models is given in Table I, along with the error on the training set for the physical log-distance propagation model used in [9], [10], [11] and for the polynomial model of [32], for several degrees q. We also use the leave-one-out (LOO) technique in order to evaluate the performance of the proposed model in the case of data that are not part of the training set. The LOO technique involves using a single observation from the collected data as the validation data, and the remaining observations as the training data. This is repeated 43 times, such that each observation is used once as the validation data. This technique is interesting because it allows us to compare the proposed models to the log-distance model, even though the set of collected data is not really large. Finally, the mean estimation error (in meters) is stored in Table I for the computed models. For convenience, in the comparison tables, let us denote the non-parametric model of Subsection III-A by NPM, and the semi-parametric model of Subsection III-B by SPM. We also denote the log-distance propagation model, that is known for being a physical model, by PM and the polynomial model by PolyM. One can see from Table I that the proposed models yield better results than the log-distance model, when comparing the mean estimation error. Moreover, the two proposed models yield really close results.

B. Evaluation of the distance models on simulated data

In this subsection, we evaluate the accuracy of the distance models on simulated data, in the case of two different scenarios. In the first paragraph, we present the first scenario,

where an area without walls is considered. We then compare the results obtained using the proposed distance models to the results obtained with the log-distance propagation model. In the second paragraph, we consider a different scenario, where the signals are attenuated because of the presence of walls in the proposed topology. As it will be shown in the following, such topology allows a better comparison between models.

We start with the first scenario, where a 100 m $\times$ 100 m area is considered, with $N_s = 16$ sensors and $N_p = 100$ reference positions distributed over the area. Fig. 2 illustrates the considered topology, where no walls or obstacles are present. The RSSIs for the training phase are obtained using the theoretical log-distance propagation model [9] given in (6), with n_P set to 4 as often given in the literature, and ρ_0 set to 1 dBm. As for the test phase, 100 positions are randomly generated in the studied area, and their RSSIs are also obtained using (6). Finally, a zero mean additive white noise $\varepsilon_{i,\ell}$ is added to all the RSSIs, with σ_ρ being its standard deviation. Here, we take σ_ρ equal to 1 dBm. Based on the study given in Section III, we define $N_s = 16$ non-parametric distance models and $N_s = 16$ semi-parametric distance models. Then, in the test phase, we estimate the distances using these models, the log-distance propagation model, and the polynomial model with $q = 2$, $q = 3$ and $q = 4$. The mean estimation errors for the 16 models (in meters) are stored in Table II. One can see from this table that the results are really close, especially for the test error, which is of much great importance than the training error. This result was expected, since we are generating the RSSIs using the log-distance propagation model, and the noise is a zero mean additive noise. Therefore, the proposed distance models are trying to model the noise, which can not be learned nor modeled, because of its randomness.

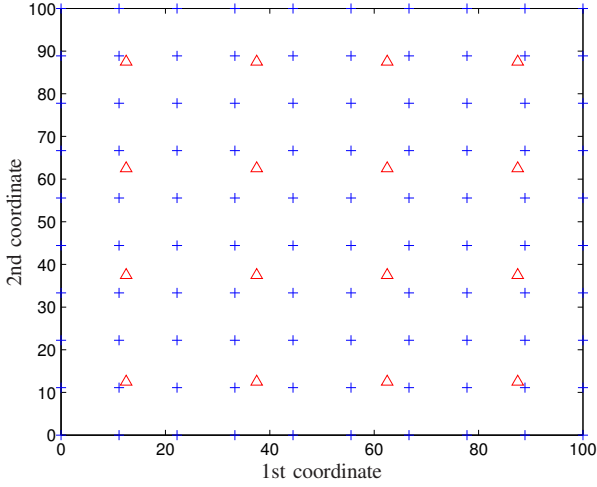We now propose another scenario given in Fig. 3, where we

Fig. 2: Topology of the testbed, $\triangle$ represents the sensors and $+$ represents the training positions.

TABLE III: Estimation errors for simulated data in the case of the second scenario, with σ_ρ =0.5 dBm.

Considered model	Mean training error	Mean test error
PM	1.26	1.37
PolyM, $q = 2$	0.56	0.62
PolyM, $q = 3$	0.55	0.61
PolyM, $q = 4$	0.54	0.61
NPM	0.20	0.32
SPM	0.22	0.34

TABLE IV: Estimation errors for simulated data in the case of the second scenario, with σ_ρ =1 dBm.

Considered model	Mean training error	Mean test error
PM	1.29	1.39
PolyM, $q = 2$	0.68	0.66
PolyM, $q = 3$	0.67	0.65
PolyM, $q = 4$	0.67	0.66
NPM	0.39	0.51
SPM	0.43	0.52

consider a 25 m $\times$ 5 m area, 2 fixed sensors and 45 known positions for the training phase. This figure shows that there are 5 rooms in the area, meaning that the signal penetrates a maximum of 4 walls during its propagation. Therefore, we consider the average walls model described in [33] to generate the RSSI measures. This model is a modified version of the log-distance model that explicitly takes into account the attenuations due to walls. The received signal strength indicator is then given by the following:

$$\rho_{\boldsymbol{s}_i,\boldsymbol{p}_\ell} = \rho_0 - 10\,n_{\mathrm{P}}\log_{10} d_{\boldsymbol{s}_i,\boldsymbol{p}_\ell} - N_{w_i}L_{w_i} + \varepsilon_{i,\ell}, \quad (20)$$

where the quantities L_{w_i} and N_{w_i} denote respectively the loss due to walls and the number of penetrated walls. The quantity L_{w_i} is taken equal to 6.9 dBm, since we consider the case of heavy thick walls [33]. Now for the test phase, 100 positions are randomly generated in the studied area, and their RSSIs are obtained using (20). Then, the distances are estimated using the obtained models from Section III, and the log-distance propagation model. The estimation errors for the 2 models, i.e. a model for each sensor, are stored in Table III, when the standard deviation of $\varepsilon_{i,\ell}$ is taken equal to 0.5 dBm. Table IV yields the estimation errors when the noise's standard deviation is increased to 1 dBm. One can see from both tables that the two proposed distance models outperforms the log-distance propagation model in terms of accuracy. Moreover, it is the non-parametric model that yields the best results.

VII. EVALUATION OF THE PERFORMANCE OF THE TRACKING METHODS

This section evaluates the performance of the two proposed tracking methods on simulated data. In the first subsection, we evaluate the performance of the proposed methods in terms of accuracy for fixed values of the noises standard deviations $\boldsymbol{\sigma}_\gamma$ and σ_ρ. Then, we study the impact of varying

the number of sensors N_s on the estimation error. Next, in the third subsection, we study the impact of varying the noises standard deviations $\boldsymbol{\sigma}_\gamma$ and σ_ρ on the estimation error. Then, we compare our results to ones obtained with two recently proposed positioning method; the first one is based on the WKNN algorithm combined with a Kalman filter [34], while the second one makes use of the ridge regression learning method to find the position estimates [26]. Finally, we compare the accuracy of the proposed methods to the tracking method proposed in [20] when the sensors change their locations. In the following, we consider the setup of Fig. 2, and the RSSI measures are generated using (6), with n_{P} set to 4 and ρ_0 set to 1 dBm. Now for the choice of the distance model, one can see from the results of Section VI that the non-parametric model yields the best results in all scenarios. Therefore, this model will be used in this section. As we already explained, the kernel parameters η_i and σ_i are chosen in such a way to minimize the error on the training set. As for the application of the particle filter in all our simulations, the number of particles N_{PF} is set to 50.

A. Evaluation of the proposed methods

We consider a moving target in the defined area. The target's accelerations are given in Fig. 4, γ_1 and γ_2 being the first and the second acceleration coordinates respectively. By taking twice the primitive integrals of the accelerations, we compute the coordinates expressions of the target. The trajectory obtained is then of 100 points with $\Delta t = 1$ s, and it is illustrated in Fig. 5. We consider that noises are present in all scenarios, since a noiseless setup is not realistic in a practical environment. To this end, we take σ_ρ equal to 10% of standard deviation of the RSSI measures, i.e., $\sigma_\rho = 1.08$ dBm, and we take $\boldsymbol{\sigma}_\gamma$ equal to 5% of the standard deviation of
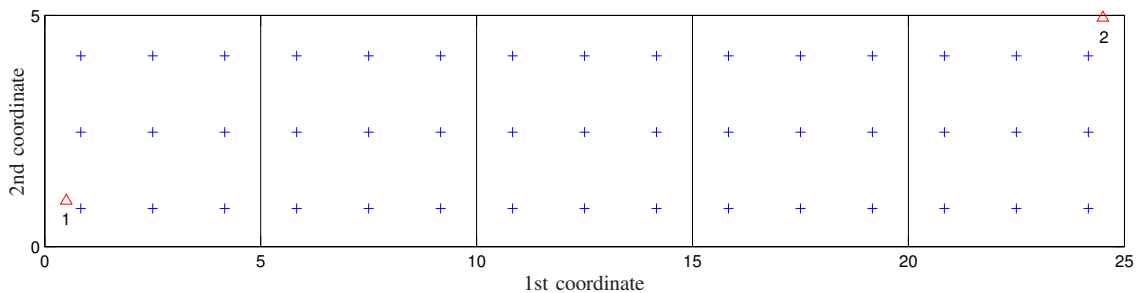
Fig. 3: Topology of the simulated area for the second scenario, $\triangle$ represents the sensors and $+$ represents the training positions.
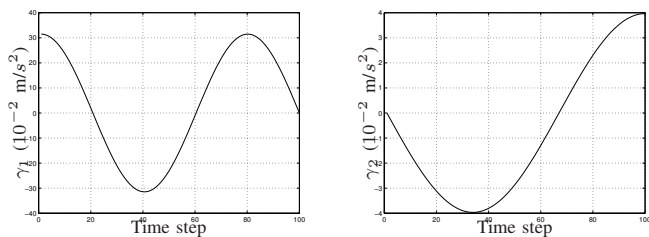

Fig. 4: Acceleration signals of the target.

Fig. 5: Trajectory estimation.

the accelerations. Let the estimation error be evaluated by the root mean squared distance between the exact positions and the estimated ones. Fig. 5 shows the estimated trajectories when using the two proposed methods. The results are averaged over 50 Monte-Carlo simulations. The mean error obtained when using the Kalman filter of Section IV is equal to 1.03 m. As for the mean error obtained with the particle filter of Section V, it is equal to 0.68 m. One can see that both methods allow an accurate tracking of the target, with a better estimation error when considering the particle filter with this setup.

B. Impact of $\boldsymbol{\sigma}_\gamma$ and σ_ρ

In this subsection, we study the impact of the noises standard deviations $\boldsymbol{\sigma}_\gamma$ and σ_ρ on the estimation error. First, we take a fixed value for σ_ρ, equal to 10% of standard deviation of the RSSI measures. Then, different percentages of the standard deviation of the acceleration are considered, ranging from 1% to 10%. The estimation errors are averaged over 50 Monte-Carlo simulations. Fig. 6 shows the impact of the variation of $\boldsymbol{\sigma}_\gamma$ on the estimation error. This figure shows that both filters have similar performances when the noise on the accelerations is small; however, with higher noise values, the particle filter outperforms the Kalman filter.

Next, we take a fixed value for $\boldsymbol{\sigma}_\gamma$ equal to 5% of the standard deviation of the accelerations, with several percentages of the standard deviation of the RSSI measures, going from 0% to 50%. Fig. 7 shows the impact of the variation of σ_ρ on the estimation error for both methods. One can see here as well that the particle filter yields better results than the Kalman
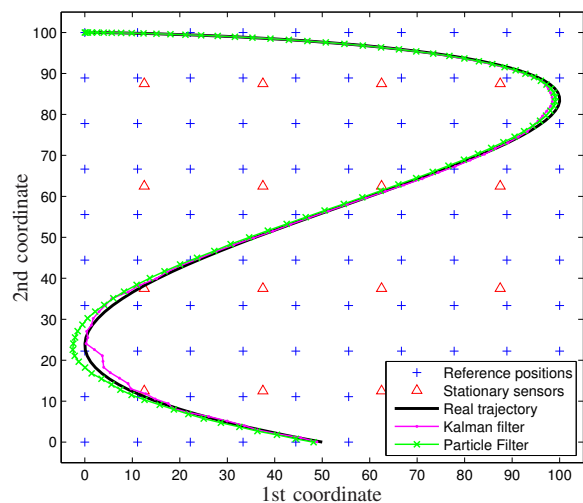
filter. This is due to the distribution of the observation errors, assumed to be Gaussian in the Kalman filter. Indeed, small RSSI errors yield slightly varying observation errors. However, with higher RSSI errors, the noise distribution gets farther from a Gaussian one, and the particle filter performs better. Nevertheless, one can see that both methods have relatively small estimation errors, compared to the space dimensions.

C. Comparison to other methods

The objective now is to first compare the proposed approach to two recently proposed positioning methods [19], [34], [26]. We consider the same setup as the one in Subsection VII-A, and the same values for $\boldsymbol{\sigma}_\gamma$ and σ_ρ. Then, taking the same setup, we consider the case where sensors are moving in the surveillance area, and we compare the performance of the proposed methods to the tracking method of [20] in such a case.

We first describe briefly the methods in [19], [34]. It consists of estimating the position using the weighted K-nearest
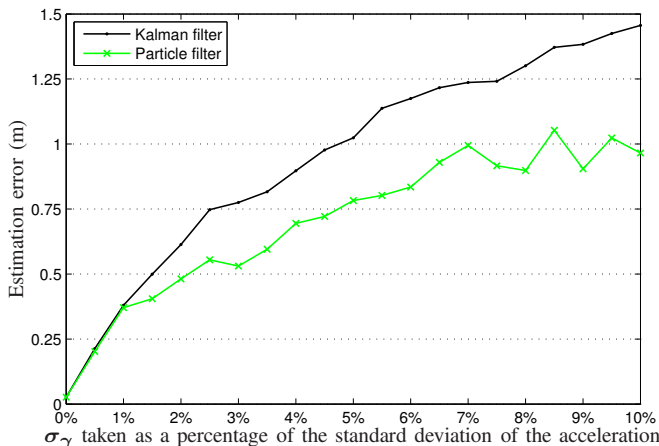
Fig. 6: Estimation error as a function of the noise on the accelerations with σ_ρ equal to 10% of the standard deviation of the RSSIs.
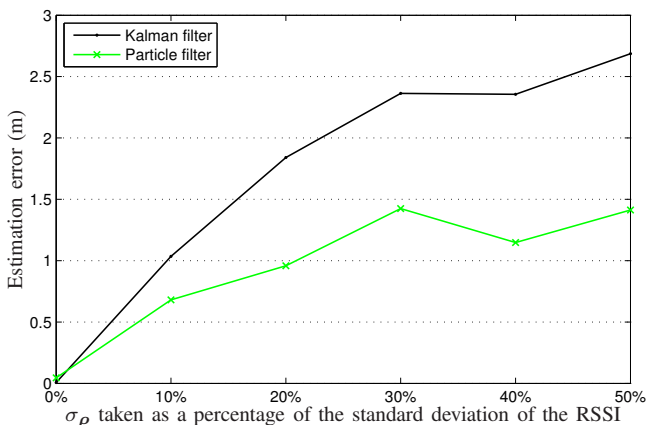

Fig. 7: Estimation error as a function of the noise on the RSSI.

TABLE V: Estimation error (in meters) for the different methods.

Tracking method	Estimation error
Proposed Kalman filter	**1.03**
Proposed particle filter	**0.68**
RR	2.98
WKNN	4.57
WKNN + Kalman	2.15
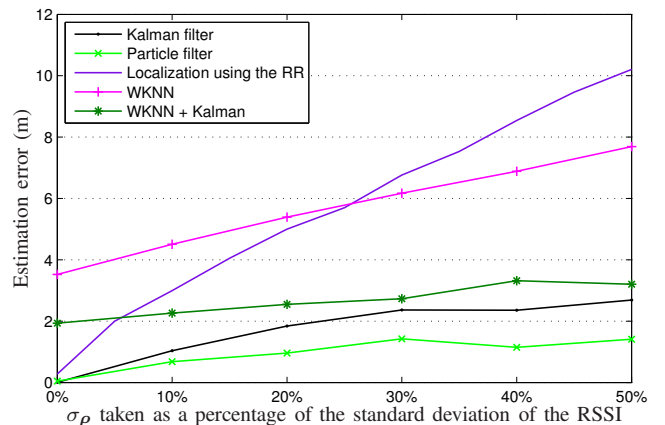
Fig. 8: Estimation error as a function of the noise on the RSSI, using different methods.

neighbor (WKNN) algorithm, then applying the Kalman filter to enhance the estimation. A target's first position estimate using WKNN is given by weighted combinations of the K nearest neighboring positions from the training database, with the nearness indicator being based on the Euclidean distance between RSSIs. The number of neighbors K is taken equal to 8 as in the simulations of [19], where more details can be found. As for the position enhancement using the Kalman filter, a second-order state-space model is used. The estimation errors (in meters) obtained when using this algorithm (WKNN + Kalman) are computed 50 times, and the mean estimation error is stored in Table V. The table also shows the mean estimation error of this algorithm without using the acceleration information (WKNN) and the errors standard deviations as well. One can see that the two proposed methods outperform the WKNN-based methods in terms of accuracy.

As for the localization method of [26], a radio-fingerprinting database and the ridge regression learning method are used to define a model that takes as input the RSSI measures and gives as output the position of the target. However, this method does not take into account the acceleration information. Table V also shows the estimation error obtained using this method (RR). The errors obtained with the proposed methods are significantly smaller than the one obtained with the method of [26].

Next, several percentages of the standard deviation of the RSSI measures are taken using the methods in [19] and [26]. The value of σ_γ is taken equal to 5% of the standard deviation of the accelerations as in the scenario of Fig. 7. The estimation error obtained in this case is shown in Fig. 8. This figure shows that the two proposed methods outperform the other methods for all values of σ_ρ, compared to the results obtained in Fig. 7.

We now compare these methods for several values of the number of sensors N_s, for a fixed number of reference positions $N = 100$. σ_ρ is taken equal to 10% of the standard deviation of the RSSI measures and σ_γ equal to 5% of the standard deviation of the accelerations. Fig. 9 shows the estimation error as a function of N_s. One can see that the two proposed methods yield the smallest errors for all values of the number of sensors; moreover, at $N_s = 4$, the other methods do not give such accurate results. Another important thing to notice is that the proposed methods are more robust to the changes in the number of sensors and they present less
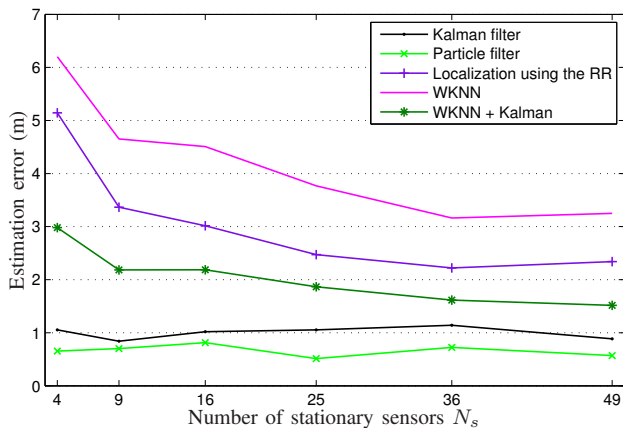
Fig. 9: Estimation error as a function of the number of stationary sensors.

Fig. 10: Estimation error as a function of the radius defining the movement of the sensors.

variations than the other methods.

Finally, we consider that the $N_s = 16$ sensors are moving in the surveillance area, and we compare the performance of the proposed tracking methods of this paper to the performance of the method of [20]. In [20], the target is first localized using a predefined model that takes as input the RSSI measures and gives as output the position of the target such as in [26]. The estimated positions are then combined with acceleration information using a Kalman filter along with a third-order state-space model. This approach gives accurate position estimates in the case of stationary sensors. However, if the sensors were to move, the results would not be accurate anymore, since the sensors' positions are included in the learning process. The advantage of the proposed methods is that we find the RSSI/distance relationship, and thus find directly the distance from the RSSI measures. When the sensors change their positions, this relationship does not change, thus we do not need to reconfigure the model, i.e. a new training phase is not needed. Now we consider that the sensors have circular movements. The trajectory of Fig. 5 is estimated using the three methods for different radii ranging from 0 to $30m$ and for random angles going from 0 to 2π (radians). The results are then averaged over 50 Monte-Carlo simulations. Fig. 10 shows the mean estimation error as a function of the radius for the three tracking methods. One can see that all three methods yield close results at the beginning, when the radius is equal to zero, that is when the sensors remain stationary. However, when the value of the radius increases, the estimation error also increases for the method of [20], while the estimation error almost remains the same for the two proposed methods in this paper. This result proves that the new proposed methods are more robust to changes in the initial configuration of surveillance area than the method of [20].

VIII. CONCLUSION

In this paper, we proposed two original regression models that relate the received signal strength indicators (RSSIs) to
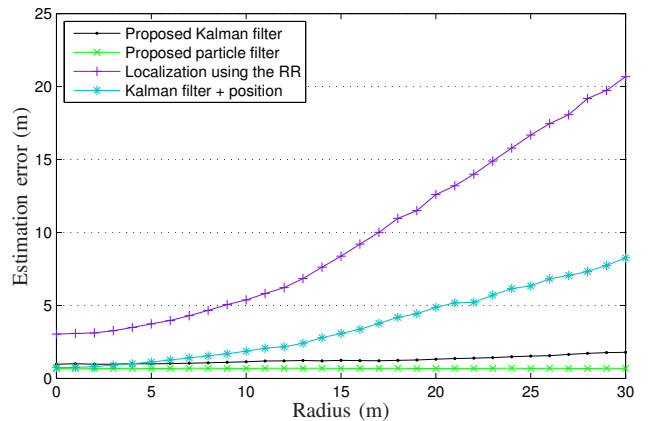
the distances separating sensors in a wireless sensor network. Then, we solved the tracking problem using the estimated distances and two new methods that take into account the target's motion. We provided a fully comprehensive study of the proposed distance models and their performances. Simulation results show that our models yield accurate distance estimation. Results also show that our tracking methods allow accurate position estimation, and are proved to be robust in the case of noisy data. Both proposed tracking methods outperform tracking using recently developed methods based on the WKNN method and the Kalman filter or some other learning strategy. Future work will handle further improvements of this work, such as wisely choosing a group of sensors instead of using all the available distance information. Solutions to cases where zones of the surveillance area are not covered by all sensors could also be provided.

REFERENCES

[1] P. Honeine, F. Mourad, M. Kallas, H. Snoussi, H. Amoud, and C. Francis, "Wireless sensor networks in biomedical: body area networks," in *Proc. 7th International Workshop on Systems, Signal Processing and their Applications*, Algeria, 09–11 May 2011.

[2] F. Salvadori, M. De Campos, P. Sausen, R. De Camargo, C. Gehrke, C. Rech, M. Spohn, and A. Oliveira, "Monitoring in industrial systems using wireless sensor network with dynamic power management," *IEEE Transactions on Instrumentation and Measurement*, vol. 58, no. 9, pp. 3104–3111, 2009.

[3] S. H. Lee, S. Lee, H. Song, and H. S. Lee, "Wireless sensor network design for tactical military applications : Remote large-scale environments," in *Military Communications Conference*, 2009, pp. 1 –7.

[4] A. Tovar, T. Friesen, K. Ferens, and B. McLeod, "A dtn wireless sensor network for wildlife habitat monitoring," in *23rd Canadian Conference on Electrical and Computer Engineering (CCECE)*, 2010, pp. 1–5.

[5] E.-E.-L. Lau and W.-Y. Chung, "Enhanced rssi-based real-time user location tracking system for indoor and outdoor environments," in *International Conference on Convergence Information Technology*, 2007, pp. 1213–1218.

[6] L. Zhang, Y. H. Chew, and W.-C. Wong, "A novel angle-of-arrival assisted extended kalman filter tracking algorithm with space-time correlation based motion parameters estimation," in *9th International Wireless Communications and Mobile Computing Conference (IWCMC)*, 2013, pp. 1283–1289.

[7] J. Wendeberg, J. Muller, C. Schindelhauer, and W. Burgard, "Robust tracking of a mobile beacon using time differences of arrival with simultaneous calibration of receiver positions," in *Int. Conf. on Indoor Positioning and Indoor Navigation (IPIN)*, 2012, pp. 1–10.

[8] E. Xu, Z. Ding, and S. Dasgupta, "Target tracking and mobile sensor navigation in wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 12, no. 1, pp. 177–186, 2013.

[9] A. Medeisis and A. Kajackas, "On the use of the universal Okumura-Hata propagation prediction model in rural areas," *Vehicular Conference Proceedings*, vol. 3, 2000.

[10] N. Patwari, J. Ash, S. Kyperountas, A. Hero, R. Moses, and N. Correal, "Locating the nodes: cooperative localization in wireless sensor networks," *IEEE Signal Processing Magazine*, vol. 22, no. 4, pp. 54–69, July 2005.

[11] A. Zanella and A. Bardella, "Rss-based ranging by multichannel rss averaging," *IEEE Wireless Communications Letters*, vol. 3, no. 1, pp. 10–13, February 2014.

[12] A. Sandeep, Y. Shreyas, S. Seth, R. Agarwal, and G. Sadashivappa, "Wireless network visualization and indoor empirical propagation model for a campus wi-fi network," *World Academy of Science, Engineering and Technology*, vol. 2, no. 6, pp. 706 – 710, 2008.

[13] J. Yang and Y. Chen, "Indoor localization using improved rss-based lateration methods," in *IEEE Global Telecommunications Conference GLOBECOM.*, Nov 2009, pp. 1–6.

[14] J. Esteves, A. Carvalho, and C. Couto, "Generalized geometric triangulation algorithm for mobile robot absolute self-localization," in *IEEE International Symposium on Industrial Electronics, ISIE '03*, vol. 1, 2003, pp. 346–351 vol. 1.

[15] D. Manolakis, "Efficient solution and performance analysis of 3-D position estimation by trilateration," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 32, no. 4, p. 12391248, 1996.

[16] G. Welch and G. Bishop, "An Introduction to the Kalman Filter," http://www.cs.unc.edu, UNC-Chapel Hill, TR95-041, February 2001.

[17] A. Shareef and Y. Zhu, "Localization using extended kalman filters in wireless sensor networks," *Graduate Student Scholarly and Creative Submissions. Paper 5.*, 2009.

[18] R. V. A. UmaMageswari, J. Joseph Ignatious, "A comparitive study of Kalman filter, extended Kalman filter and unscented Kalman filter for harmonic analysis of the non-stationary signals," *International Journal of Scientific & Engineering Research*, vol. 3, July 2012.

[19] E. Chan, G. Baciu, and S. C. Mak, "Using wi-fi signal strength to localize in wireless sensor networks," in *WRI Int. Conf. on Communications and Mobile Computing*, vol. 1, 2009, pp. 538–542.

[20] S. Mahfouz, F. Mourad-Chehade, P. Honeine, J. Farah, and H. Snoussi, "Target tracking using machine learning and kalman filter in wireless sensor networks," *IEEE Sensors Journal*, vol. 14, no. 10, pp. 3715–3725, Oct 2014.

[21] N. G. Branko Ristic, Sanjeev Arulampalam, *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech House Publishers, February 2004.

[22] H. Wang, H. Lenz, A. Szabo, J. Bamberger, and U. Hanebeck, "Wlan-based pedestrian tracking using particle filters and low-cost mems sensors," in *Positioning, Navigation and Communication, 2007. WPNC '07. 4th Workshop on*, March 2007, pp. 1–7.

[23] M. Farmani, H. Moradi, and M. Asadpour, "A hybrid localization approach in wireless sensor networks using a mobile beacon and inter-node communication," in *IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems*, May 2012, pp. 269–274.

[24] J. Robles, M. Deicke, and R. Lehnert, "3D fingerprint-based localization for wireless sensor networks," in *7th Workshop on Positioning Navigation and Communication (WPNC)*, march 2010, pp. 77 –85.

[25] R. E. Kalman, "A new approach to linear filtering and prediction problems," *J. of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[26] S. Mahfouz, F. Mourad-Chehade, P. Honeine, H. Snoussi, and J. Farah, "Kernel-based localization using fingerprinting in wireless sensor networks," in *IEEE 14th Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 2013, pp. 744–748.

[27] ——, "Decentralized localization using fingerprinting and kernel methods in wireless sensor networks." in *Proc. 21th European Conference on Signal Processing (EUSIPCO)*, 2013.

[28] C. Saunders, A. Gammerman, and V. Vovk, "Ridge regression learning algorithm in dual variables," in *In Proceedings of the 15th International Conference on Machine Learning*. Morgan Kaufmann, 1998, pp. 515–521.

[29] B. Schölkopf, R. Herbrich, and A. J. Smola, "A generalized representer theorem," in *Proc. of the 14th Annual Conference on Computational Learning Theory and 5th European Conference on Computational Learning Theory*. London, UK: Springer-Verlag, 2001, pp. 416–426.

[30] G. Zanca, F. Zorzi, and A. Zanella, "RSSI measurements in indoor wireless sensor networks." [Online]. Available: http://telecom.dei.unipd.it/pages/read/59/

[31] A. Neskovic, N. Neskovic, and G. Paunovic, "Modern approaches in modeling of mobile radio systems propagation environment," *IEEE Communications Surveys Tutorials*, vol. 3, no. 3, pp. 2–12, Third Quarter 2000.

[32] L. Wang, Y. Liu, X. Xu, and X. Wang, "Wsn multilateration algorithm based on landweber iteration," in *International Conference on Electronic Measurement Instruments*, August 2009, pp. 1–250–1–254.

[33] C. Andrade and R. Hoefel, "IEEE 802.11 WLANs: A comparison on indoor coverage models," in *Electrical and Computer Engineering (CCECE), 2010 23rd Canadian Conference on*, May 2010, pp. 1–6.

[34] D. Liu, Y. Xiong, and J. Ma, "Exploit kalman filter to improve fingerprint-based indoor localization," in *Computer Science and Network Technology (ICCSNT), 2011 International Conference on*, vol. 4, 2011, pp. 2290–2293.

Sandy Mahfouz was born in Fidar, Lebanon, on November 27, 1989. She received the diploma degree in computer and communication engineering, majoring in Telecommunications, in 2012, from the Holy Spirit University of Kaslik, Lebanon. She received the Ph.D. degree in Systems Optimization and Security in 2015 from the University of Technology of Troyes, France (UTT), where she is currently a temporary lecturer and research assistant. Her current research interests include wireless and mobile sensor networks, machine learning and signal processing.

Farah Mourad-Chehade was born on January 15, 1984. She received the diploma degree in electrical engineering from the Lebanese University, Faculty of Engineering, Tripoli, Lebanon, in 2006. She also received the Master degree, in 2007, and the Ph.D., in 2010, in systems optimization and security from the University of Technology of Troyes, France (UTT). Since September 2011, she has been an associate professor at the UTT. She has supervised two PhD theses till now. She serves as reviewer for several journals (IEEE Transactions on Signal Processing, IEEE Transactions on Robotics, IEEE Transactions on Vehicular Technology and Elsevier Signal Processing) and conferences (EUSIPCO, WOSSPA and ROADEF). Her research interests include wireless and mobile sensor networks, nonlinear signal analysis, machine learning and biomedical applications.

Joumana Farah received the B.E. degree in Electrical Engineering from the Lebanese University, in 1998, the M.E. degree in Signal, Image, and Speech processing, in 1999, and the PhD in third generation mobile communication systems, in 2002, from the Polytechnic Institute of Grenoble, France. Since January 2010, she holds an Accreditation to Supervise Research (HDR) from University Pierre and Marie Curie (Paris VI), France. She is currently a full-time faculty member of the faculty of engineering, at the Lebanese University. She has supervised a large number of Master and PhD theses. She has been the recipient of several research grants from the Lebanese National Council for Scientific Research and the Franco-Lebanese CEDRE program. She has four registered patents and software and has co-authored a research book and more than seventy papers in international journals and conferences. She was the General Chair of the 19th International Conference on Telecommunications (ICT 2012), and serves as TPC member and reviewer for several journals (IEEE Journal on Selected Areas in Communications, IEEE Communications Letters, Signal Processing: Image Communication, Digital Signal Processing, Annals of Telecommunications, etc.) and conferences (IEEE VTC, IEEE Globecom, IEEE ICECS, EUSIPCO, ICT, etc). Her current research interests include channel coding techniques, distributed video coding, multi-carrier systems, cooperative and wireless sensor networks, and resource allocation techniques.

Hichem Snoussi was born in Bizerta, Tunisia, in 1976. He received the diploma degree in electrical engineering from the Ecole Superieure d'Electricite (Supelec), Gif-sur-Yvette, France, in 2000. He also received the DEA degree and the Ph.D. in signal processing from the University of Paris-Sud, Orsay, France, in 2000 and 2003 respectively. He has obtained the HdR from the University of Technology of Compigne in 2009. Between 2003 and 2004, he was postdoctoral researcher at IRCCyN, Institut de Recherches en Communications et Cyberntiques de Nantes. He has spent short periods as visiting scientist at the Brain Science Institute, RIKEN, Japan and Olin Neuropsychiatry Research Center at the Institute of Living in USA. Between 2005 and 2010, he has been associate professor at the University of Technology of Troyes. Since September 2010, he has been appointed a Full Professor position at the same university. He is in charge of the regional research program S3 (System Security and Safety) of the CPER 2007-2013 and the CapSec plateform (wireless embedded sensors for security). He is the principal investigator of an ANR-Blanc project (mv-EMD), a CRCA project (new partnership and new technologies) and a GDR-ISIS young researcher project. He is partner of many ANR projects, GIS, strategic UTT programs. He obtained the national doctoral and research supervising award PEDR 2008-2012 and the PES (Scientific Excellence Award) 2013-2017.

Paul Honeine (M'07) was born in Beirut, Lebanon, on October 2, 1977. He received the Dipl.-Ing. degree in mechanical engineering in 2002 and the M.Sc. degree in industrial control in 2003, both from the Faculty of Engineering, the Lebanese University, Lebanon. In 2007, he received the Ph.D. degree in Systems Optimization and Security from the University of Technology of Troyes, France, and was a Postdoctoral Research associate with the Systems Modeling and Dependability Laboratory, from 2007 to 2008. From September 2008 till August 2015, he was an assistant Professor at the University of Technology of Troyes, France. Since September 2015, he is a full professor at the lab LITIS of the University of Rouen, France. His research interests include nonstationary signal analysis and classification, nonlinear and statistical signal processing, sparse representations, machine learning. Of particular interest are applications to (wireless) sensor networks, biomedical signal and image processing, hyperspectral imagery and nonlinear adaptive system identification. He is the co-author (with C. Richard) of the 2009 Best Paper Award at the IEEE Workshop on Machine Learning for Signal Processing. Over the past 5 years, he has published more than 100 peer-reviewed papers.