Target tracking using machine learning and Kalman filter in wireless sensor networks

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

Target tracking using machine learning and Kalman filter in wireless sensor networks

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

***Abstract*—This paper describes an original method for target tracking in wireless sensor networks. The proposed method combines machine learning with a Kalman filter to estimate instantaneous positions of a moving target. The target's accelerations, along with information from the network, are used to obtain an accurate estimation of its position. To this end, radio-fingerprints of received signal strength indicators (RSSIs) are first collected over the surveillance area. The obtained database is then used with machine learning algorithms to compute a model that estimates the position of the target using only RSSI information. This model leads to a first position estimate of the target under investigation. The kernel-based ridge regression and the vector-output regularized least squares are used in the learning process. The Kalman filter is used afterwards to combine predictions of the target's positions based on acceleration information with the first estimates, leading to more accurate ones. The performance of the method is studied for different scenarios and a thorough comparison to well-known algorithms is also provided.**

***Index Terms*—radio-fingerprinting, Kalman filter, machine learning, RSSI, target tracking, wireless sensor networks.**

I. Introduction

Recently, advances in radio and embedded systems have led to the emergence of Wireless Sensor Networks (WSNs), that have become a major research field during the last few years. These networks are beginning to be deployed at an accelerated pace for many applications, ranging from home monitoring [1] to industrial monitoring [2], and covering medical applications [3].

Target tracking [4], [5] is an interesting research and application field in WSNs, that consists of estimating instantly the position of a moving target. Target tracking can be viewed as a sequential localization problem, thus requiring a real-time location estimation algorithm. Typically, sensors broadcast signals in the network, while targets collect these signals for location estimation. Several types of measurements can be considered, such as received signal strength indicators (RSSIs) [6], angle of arrival (AOA) [7], time difference of arrival (TDOA) [8] and time-of-arrival (TOA) [9]. Previous studies have shown that investigating TOA and TDOA leads to more accurate position estimates compared to other methods [10]. However, implementing these techniques requires high-cost hardware, making them impractical for most applications. Unlike these techniques, the RSSI-based ones achieve acceptable performance, with no extra hardware.

Many RSSI-based tracking techniques have been proposed. For instance, [11] proposes a target tracking technique using a particle filter with the exact RSSI channel model. However, such an approach is not reliable with the highly varying RSSIs, due to the signal fading, the additive noise, etc. Also exploring RSSIs, authors of [12] and [13] propose target tracking methods based on connectivity measurements using the interval analysis or the variational filter. By exploring connectivity, these methods are more robust than the ones using the exact channel model. However, the performance highly depends on the number and the positions of the sensors in the network. In [14], the authors propose a tracking algorithm that works in indoor and outdoor environments. Indeed, it switches between GPS information when the node is outdoors and an existing Wifi-based application when the GPS signals are no more available. This android application yields several position estimates of the node when the Wifi is activated. Then, it is followed by a Gaussian process regression that uses the estimated positions, to reconstruct a smooth trajectory and recover the missing positions. In addition to measurements, tracking algorithms can employ a state-space model to refine the position estimation based on its previous position. For instance, a first-order model is used with a Kalman filter in [15], and with a particle filter in [11], whereas [16] employs a second-order one. However, these models are only reliable for targets having slightly varying velocities or accelerations.

In other contexts, RSSI-based methods have been proposed for nodes localization in WSNs. These methods aim at location estimation by investigating observation information without taking advantage of nodes mobility. One interesting RSSI-based localization approach consists of radio-fingerprinting [17], [18]. Such an approach allows taking into consideration the stationary characteristics of the environment. Several studies have been made for sensors localization using RSSI-based radio-fingerprinting, such as the weighted K-nearest neighbor (WKNN) algorithm [19]. We have recently proposed in [20] and [21] two localization methods using radio-fingerprinting in WSNs, by taking advantage of kernel methods in machine learning. These methods outperform the WKNN approach. In order to perform tracking, the authors of [22] propose to correct the

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

J. Farah is with the Department of telecommunications, Faculty of engineering, Université Saint-Esprit de Kaslik, Lebanon, e-mail: joumanafarah@usek.edu.lb.

This work is supported by the Champagne-Ardenne Region in France, grant WiDiD: Wireless Diffusion Detection.

WKNN estimates using the Kalman filter with a second-order state-space model.

In this paper, we propose a new method for target tracking in WSNs that combines radio-fingerprinting and accelerometer information. The proposed method consists of setting reference positions along the network where RSSI measurements are collected, leading to a radio-fingerprint database. This database is used with machine learning algorithms to define a kernel-based model, whose input is the RSSI vector and whose output is the corresponding position. To estimate this model, we investigate two learning algorithms: the ridge regression and the vector-output regularized least squares. A moving target then measures its RSSIs and instantaneous acceleration. A first position estimate is obtained using the already-defined kernel-based model and the measured RSSIs, then this estimate is combined with the acceleration information, by means of a Kalman filter, to achieve better accuracy. To this end, three different orders of the tracking models are examined. The proposed method outperforms existing methods, especially for hyperactive targets.

The rest of the paper is organized as follows. The proposed tracking approach is presented in Section II. Section III describes three orders of the state-space model, while Section IV defines the observation model and the use of machine learning in our method. In Section V, we examine the performance of the proposed method and compare it to two recently derived methods. Finally, Section VI concludes the paper.

II. Tracking approach

Consider an environment of D dimensions, for instance $D = 2$ for a two-dimensional environment, and N_s stationary sensors having known locations, denoted by $\boldsymbol{s}_i,\ i \in \{1,\ldots,N_s\}$. In the following, all coordinates are D-dimension row vectors. For the sake of clarity, and without loss of generality, only one target with the unknown position $\boldsymbol{x}(k)$ is considered, k being the current time step. Nevertheless, the proposed method could be extended to several moving targets, since they are tracked independently from each other, using their accelerations and information exchanged only with the stationary sensors.

To this end, a linear state-space model is proposed to describe the target's motion as follows:

$$\boldsymbol{x}(k) = \boldsymbol{x}(k-1)\boldsymbol{A} + \boldsymbol{B}(k) + \boldsymbol{\theta}(k), \tag{1}$$

where $\boldsymbol{x}(k-1)$ is the target's previous position, $\boldsymbol{A}$ is a D-by-D state transition matrix that relates the current position of the target to its previous one, $\boldsymbol{B}(k)$ is a control-input vector depending on the accelerations, and $\boldsymbol{\theta}(k)$ is a random vector noise whose probability distribution is assumed to be normal, having zero mean and covariance matrix $\boldsymbol{Q}(k)$, that is $\boldsymbol{\theta}(k) \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q}(k))$. More details about the definitions of $\boldsymbol{A}$, $\boldsymbol{B}(k)$ and $\boldsymbol{Q}(k)$ are given in Section III, where different orders of the state-space model are considered. In addition to its accelerations, the target exchanges information in the network with the stationary sensors at each time step. It therefore collects a set of measurements, stored in $\boldsymbol{z}(k)$. These measurements are described in detail in Section IV. Let the observation equation be given by its linear general formulation as follows:

$$\boldsymbol{z}(k) = \boldsymbol{x}(k)\boldsymbol{H} + \boldsymbol{n}(k), \tag{2}$$

where $\boldsymbol{H}$ is the observation matrix that relates the state $\boldsymbol{x}(k)$ to the measurement $\boldsymbol{z}(k)$ and $\boldsymbol{n}(k) \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{R})$ is the observation noise with normal distribution, zero mean and covariance matrix $\boldsymbol{R}$. This variable is assumed to be independent of $\boldsymbol{\theta}(k)$. The values of $\boldsymbol{H}$ and $\boldsymbol{R}$, as well as the choice of the linear observation model, will also be discussed in detail in Section IV.

Having defined both the state-space model and the observations, we now propose to solve the tracking problem by using a Kalman filter [23], [24]. To this end, the proposed filter first predicts the unknown position using the previous estimated position and the state-space equation (1). Then, the predicted position is corrected using the observation (2) in the following step.

Now let $\hat{\boldsymbol{x}}(k-1)$ denote the target's position estimated with the Kalman filter at time step $k-1$. Therefore, the predicted position can be written as:

$$\hat{\boldsymbol{x}}^{-}(k) = \hat{\boldsymbol{x}}(k-1)\,\boldsymbol{A} + \boldsymbol{B}(k). \tag{3}$$

At $k = 0$, $\hat{\boldsymbol{x}}(0)$ is assumed to be known. Then, the Kalman filter updates the D-by-D predicted estimation covariance as follows:

$$\boldsymbol{T}^{-}(k) = \boldsymbol{A}\,\boldsymbol{T}(k-1)\boldsymbol{A}^{\top} + \boldsymbol{Q}(k), \tag{4}$$

where $\boldsymbol{T}(k-1)$ is the final covariance estimation at time step $k-1$ and $\boldsymbol{T}(0)$ is null since the initial state is known. Then, the predicted quantities $\hat{\boldsymbol{x}}^{-}(k)$ and $\boldsymbol{T}^{-}(k)$ are corrected using the observation equation (2) as follows:

$$\hat{\boldsymbol{x}}(k) = \hat{\boldsymbol{x}}^{-}(k) + (\boldsymbol{z}(k) - \hat{\boldsymbol{x}}^{-}(k)\,\boldsymbol{H})\,\boldsymbol{G}_K(k) \tag{5}$$

$$\boldsymbol{T}(k) = (\mathbf{I}_D - \boldsymbol{G}_K(k)\,\boldsymbol{H})\,\boldsymbol{T}^{-}(k), \tag{6}$$

where $\mathbf{I}_D$ is the D-by-D identity matrix and $\boldsymbol{G}_K(k)$ is the optimal Kalman gain given by:

$$\boldsymbol{G}_K(k) = \boldsymbol{T}^{-}(k)\,\boldsymbol{H}^{\top}\,(\boldsymbol{H}\,\boldsymbol{T}^{-}(k)\,\boldsymbol{H}^{\top} + \boldsymbol{R})^{-1}. \tag{7}$$

In the following section, the state-space model is described in detail by writing the model (1) in three different forms with the corresponding covariance matrix $\boldsymbol{Q}(k)$. As for the choice of the observation $\boldsymbol{z}(k)$ in (2), it is explained in Section IV.

III. State-space models

This section highlights the definition of the state-space-model of the tracking problem, where the target is assumed to be equipped with an accelerometer, yielding at each time step its current D accelerations. The target is assumed to be fixed at a known position $\boldsymbol{x}(0)$ at the beginning of the tracking. The objective consists then of relating the current position of the target $\boldsymbol{x}(k)$ to its previous position $\boldsymbol{x}(k-1)$, using its measured accelerations. To do this, three orders of the state-space model are described: (i) a first-order, where

the velocities are assumed to be constant between any two consecutive time steps; (ii) a second-order, where the accelerations are assumed to be constant between any two consecutive time steps; and finally, (iii) a third-order, where the accelerations are assumed to vary linearly between any two consecutive time steps. In all these cases, the target's trajectory is described by the equation (1). In the following, $\boldsymbol{\nu}(k)$ denotes the estimated velocity vector of the target at the time step k, $\boldsymbol{\gamma}(k)$ denotes its measured acceleration vector at the time step k, and Δt is the tracking period, that is the time period separating two consecutive time steps.

A. First-order state-space model

The first-order model makes two assumptions on the motion of the target. It first assumes that the acceleration vector of the target is constant between two consecutive time steps $k-1$ and k, and equal to $\boldsymbol{\gamma}(k)$. It thus computes the target's velocity vector iteratively by:

$$\boldsymbol{\nu}(k) = \boldsymbol{\nu}(k-1) + \boldsymbol{\gamma}(k)\,\Delta t, \tag{8}$$

with $\boldsymbol{\nu}(0)$ null since the target is assumed to be fixed at the beginning of the tracking. It then assumes that the velocity between $k-1$ and k is constant and equal to $\boldsymbol{\nu}(k)$. This leads to:

$$\boldsymbol{B}(k) = \boldsymbol{\nu}(k)\,\Delta t, \tag{9}$$

in the state-space equation (1) with $\boldsymbol{A}$ equal to the D-by-D identity matrix. It is obvious that this model only works for slightly-varying-velocity targets ($\boldsymbol{\gamma} \simeq \mathbf{0}$). With more active targets, the performance of this first-order model degrades very fast, because of its two assumptions.

As for the model noises, the acceleration noise is assumed to be independent with zero-mean normal distribution, having known variances $\sigma^2_{\gamma,d}$, $d = 1, ..., D$. Their values can be estimated by performing a calibration of the accelerometer before the tracking stage. With noisy accelerations, the estimated velocities have noise, with a zero-mean normal distribution having the covariance matrix $\boldsymbol{Q}_{\boldsymbol{\nu}}(k)$ updated recursively as follows:

$$\boldsymbol{Q}_{\boldsymbol{\nu}}(k) = \boldsymbol{Q}_{\boldsymbol{\nu}}(k-1) + \Delta t^2\,\mathrm{Diag}\left(\boldsymbol{\sigma}^2_{\boldsymbol{\gamma}}\right), \tag{10}$$

where $\mathrm{Diag}\left(\boldsymbol{\sigma}^2_{\boldsymbol{\gamma}}\right)$ is the D-by-D diagonal matrix with entries $\sigma^2_{\gamma,d}$, $d = 1, ..., D$, and $\boldsymbol{Q}_{\boldsymbol{\nu}}(0)$ is null. The state noise $\boldsymbol{\theta}(k)$ is then normally distributed with zero-mean and the covariance matrix $\boldsymbol{Q}(k)$ given as follows:

$$\begin{aligned}\boldsymbol{Q}(k) &= \mathrm{Cov}\left(\boldsymbol{x}(k-1) + \boldsymbol{\nu}(k)\,\Delta t\right), \\ &= \boldsymbol{Q}(k-1) + \boldsymbol{Q}_{\boldsymbol{\nu}}(k-1)\,\Delta t^2,\end{aligned} \tag{11}$$

where $\boldsymbol{Q}(0)$ is null since there is no uncertainty over the target's position at time step $k = 0$. It is worth noting that the covariance matrices of the target's velocity and state noises are diagonal since noises over the coordinates are assumed to be independent.

B. Second-order state-space model

The second-order model assumes that the acceleration vector is constant between two consecutive time steps $k-1$ and k, and equal to $\boldsymbol{\gamma}(k)$. The velocity vector is estimated as for the first-order model using equation (8). However, the control vector $\boldsymbol{B}(k)$ of the state-space equation is modified as follows:

$$\boldsymbol{B}(k) = \boldsymbol{\nu}(k-1)\,\Delta t + \boldsymbol{\gamma}(k)\,\frac{\Delta t^2}{2}, \tag{12}$$

with the transition matrix $\boldsymbol{A}$ equal to identity.

Here, the covariance matrix $\boldsymbol{Q}(k)$ is diagonal, given by:

$$\begin{aligned}\boldsymbol{Q}(k) &= \mathrm{Cov}\left(\boldsymbol{x}(k-1) + \boldsymbol{\nu}(k-1)\,\Delta t + \boldsymbol{\gamma}(k)\,\frac{\Delta t^2}{2}\right), \\ &= \boldsymbol{Q}(k-1) + \boldsymbol{Q}_{\boldsymbol{\nu}}(k-1)\,\Delta t^2 + \frac{1}{4}\Delta t^4\,\mathrm{Diag}\left(\boldsymbol{\sigma}^2_{\boldsymbol{\gamma}}\right),\end{aligned} \tag{13}$$

where $\boldsymbol{Q}(k)$ is null at time step $k = 0$ since the target's position is initially known, and $\boldsymbol{Q}_{\boldsymbol{\nu}}(k)$ is given by (10) as for the first-order model.

The second-order state-space model outperforms the first-order one, since it considers less approximations and assumptions. This model performs well with slightly varying accelerations motions. However, it is not well-adapted to trajectories with abrupt changes in accelerations, since estimates might be significantly deviated from the exact trajectory due to cumulative model errors over time.

C. Third-order state-space model

This model considers that the target's accelerations vary linearly between two consecutive time steps, that is the acceleration vector varies from $\boldsymbol{\gamma}(k-1)$ at $k-1$ to $\boldsymbol{\gamma}(k)$ at k with a slope equal to $\frac{\boldsymbol{\gamma}(k)-\boldsymbol{\gamma}(k-1)}{\Delta t}$. According to this assumption, the velocity vector of the target at time step k is estimated recursively by:

$$\boldsymbol{\nu}(k) = \boldsymbol{\nu}(k-1) + \boldsymbol{\gamma}(k-1)\,\Delta t + \frac{\boldsymbol{\gamma}(k) - \boldsymbol{\gamma}(k-1)}{\Delta t}\,\frac{\Delta t^2}{2}, \tag{14}$$

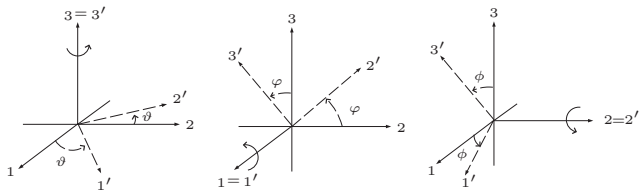
where the target is also assumed to be fixed at the beginning of the tracking (*i.e.,* $\boldsymbol{\nu}(0) = \mathbf{0}$) with null acceleration (*i.e.,* $\boldsymbol{\gamma}(0) = \mathbf{0}$), and at a known position $\boldsymbol{x}(0)$. Then, the vector $\boldsymbol{B}(k)$ in (1) is given by:

$$\begin{aligned}\boldsymbol{B}(k) = \boldsymbol{\nu}(k-1)\,\Delta t \; &+ \; \boldsymbol{\gamma}(k-1)\,\frac{\Delta t^2}{2} \\ &+ \; \frac{\boldsymbol{\gamma}(k) - \boldsymbol{\gamma}(k-1)}{\Delta t}\,\frac{\Delta t^3}{6},\end{aligned} \tag{15}$$

with the transition matrix $\boldsymbol{A}$ equal to identity, as for the other two models.

The covariance matrix $\boldsymbol{Q}(k)$ is also diagonal, given by the following:

$$\boldsymbol{Q}(k) = \boldsymbol{Q}(k-1) + \boldsymbol{Q}_{\boldsymbol{\nu}}(k-1)\,\Delta t^2 + \frac{11}{36}\Delta t^4\,\mathrm{Diag}\left(\boldsymbol{\sigma}^2_{\boldsymbol{\gamma}}\right), \tag{16}$$



(a) (b) (c)

Fig. 1: Rotations in a three-dimensional environment.

where $\boldsymbol{Q}(0)$ is null since there is no uncertainty at time step $k = 0$, and $\boldsymbol{Q}_{\boldsymbol{\nu}}(k)$ is given by:

$$\boldsymbol{Q}_{\boldsymbol{\nu}}(k) = \boldsymbol{Q}_{\boldsymbol{\nu}}(k-1) + \frac{3}{2}\Delta t^2 \operatorname{Diag}\left(\boldsymbol{\sigma}_{\boldsymbol{\gamma}}^2\right), \qquad (17)$$

with $\boldsymbol{Q}_{\boldsymbol{\nu}}(0)$ also null. This model outperforms the other models, since it brings the estimated trajectory closer to the real one compared to the others, especially for hyperactive targets having highly varying accelerations.

Remark 1. *In the previous paragraphs, the target is assumed to be rotationally constrained. Indeed, the accelerations measured by the accelerometer in the target's coordinate system are used directly in the equations, as if they are measured in the world coordinate system. However, during its motion in real applications, the target could rotate, and the coordinate system, where the accelerations are given, might change. The solution to this problem is to equip each target with a gyroscope, which yields its orientations with respect to the world coordinate system.*

*Assume that the localization is performed in a three-dimensional environment (*i.e., $D = 3$*). Consider that ϑ, φ, and ϕ are the angles of the counter-clockwise rotation of the target, given by its gyroscope at a given time step around the third coordinate axis of the world system, the first one and the second one respectively. The plots (a), (b) and (c) of Fig. 1 illustrate the single rotations around the third, the first, and the second axes respectively, 1,2, and 3 being the world coordinate axes, and 1',2', and 3' the target's ones. Let $\boldsymbol{\gamma} = (\gamma_1 \;\; \gamma_2 \;\; \gamma_3)$ be the acceleration vector of the target in the world coordinate system at the same time step and let $\boldsymbol{\gamma}' = (\gamma_1' \;\; \gamma_2' \;\; \gamma_3')$ be its measured one in its coordinate system. Then, having the rotation angles, $\boldsymbol{\gamma}$ is computed as follows:*

$$\boldsymbol{\gamma} = \boldsymbol{\gamma}' \, \mathfrak{R}, \qquad (18)$$

where the first column of the three-dimensional rotation matrix $\mathfrak{R}$ is defined by:

$$\begin{pmatrix} \cos\vartheta\cos\phi \\ -\sin\vartheta\cos\phi \\ \sin\phi \end{pmatrix}, \qquad (19)$$

its second column is defined by:

$$\begin{pmatrix} \cos\vartheta\sin\varphi\sin\phi + \sin\vartheta\cos\varphi \\ -\sin\vartheta\sin\varphi\sin\phi + \cos\vartheta\cos\varphi \\ -\sin\varphi\cos\phi \end{pmatrix}, \qquad (20)$$

and its third column is defined by:

$$\begin{pmatrix} -\cos\vartheta\cos\varphi\sin\phi + \sin\vartheta\sin\varphi \\ \sin\vartheta\cos\varphi\sin\phi + \cos\vartheta\sin\varphi \\ \cos\varphi\cos\phi \end{pmatrix}. \qquad (21)$$

*In a two-dimensional environment (*i.e., $D = 2$*), where $\boldsymbol{\gamma} = (\gamma_1 \;\; \gamma_2)$, the rotation is only possible in the plane with the rotation angle ϑ. By setting $\phi = \varphi = 0$, one gets the following transformation:*

$$\boldsymbol{\gamma} = \boldsymbol{\gamma}' \begin{pmatrix} \cos\vartheta & \sin\vartheta \\ -\sin\vartheta & \cos\vartheta \end{pmatrix}. \qquad (22)$$

During the tracking, the target measures its acceleration vector in its coordinate system at each time step, then finds its orientations using the gyroscope. Its accelerations in the world coordinate system can then be computed and used in the localization algorithm. For simplicity, we only consider rotationally constrained targets in our paper. However, as just explained, the computations could be easily modified to consider the target's rotations.

IV. Observation model

The aim of this section is to define the observation model (*i.e.,* $z(k)$ in (2)), based on the information gathered by the target from the stationary sensors in the network. The proposed method is a radio-fingerprinting approach using the Received Signal Strength Indicators (RSSIs) of the signals exchanged between the target and the stationary sensors. It is worth noting that the target is assumed to be active and cooperative in the proposed approach, that is, it exchanges information with its neighborhood. Based on radio-fingerprinting, the approach needs then a configuration phase, before the tracking. To this end, N_p reference positions, denoted by $\boldsymbol{p}_\ell$, $\ell \in \{1, \ldots, N_p\}$, are generated uniformly or randomly in the studied region. All stationary 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 $\boldsymbol{\rho}_\ell = (\rho_{\boldsymbol{s}_1,\boldsymbol{p}_\ell} \;\; \ldots \;\; \rho_{\boldsymbol{s}_{N_s},\boldsymbol{p}_\ell})^\top$ be the vector of RSSIs sent by all N_s sensors and received at the position $\boldsymbol{p}_\ell$, $\ell \in \{1, \ldots, N_p\}$. In this way, a set of N_p pairs $(\boldsymbol{\rho}_\ell, \boldsymbol{p}_\ell)$ is obtained. This radio-fingerprint database is considered in the estimation of the observation model, that is $z(k)$.

A. Definition of the observations

While moving, the target collects the sensors signals and measures their RSSIs. Instead of using the target's RSSIs as observations, the proposed approach consists of finding a function $\boldsymbol{\psi}\colon \mathbb{R}^{N_s} \mapsto \mathbb{R}^D$, based on the radio-fingerprint database, that associates to each RSSI vector $\boldsymbol{\rho}_\ell$ the corresponding position $\boldsymbol{p}_\ell$, with the advantage of not having to estimate the channel model. Kernel methods in machine learning [25], [26], [27] provide an elegant framework to define the function $\boldsymbol{\psi}(\cdot)$, as it will be shown in the following subsection. It is worth noting that the database construction and the computation of $\boldsymbol{\psi}(\cdot)$ are

performed only once, before the tracking phase. Once the model is available, the target is able to perform all tracking computations and determine its own position. Indeed, consider the moving target collecting RSSIs in the network. At a given time step k, it stores them into a vector $\boldsymbol{\rho}(k)$, and then uses the defined model $\boldsymbol{\psi}(\cdot)$ to compute its position. The first estimated coordinates of the target at time step k are then given by $\boldsymbol{\psi}(\boldsymbol{\rho}(k))$. This estimate is considered as an observation of the desired value, namely

$$\boldsymbol{z}(k) = \boldsymbol{\psi}(\boldsymbol{\rho}(k)). \tag{23}$$

Both the observation and the state-space model (1) are then used in the Kalman filter to compute a more accurate position estimation, as shown in Section II. Determining the model $\boldsymbol{\psi}(\cdot)$ is explained in the following paragraph.

Following the definition of $\boldsymbol{z}(k)$, one can see that the matrix $\boldsymbol{H}$ of (2) is set to identity. As for $\boldsymbol{n}(k) \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{R})$, an approximation of the value of its covariance matrix $\boldsymbol{R}$ is done by generating a new set of reference pairs, and by localizing the positions according to the defined model $\boldsymbol{\psi}(\cdot)$. The error on the new set is computed and stored into a vector, then the matrix $\boldsymbol{R}$ is determined by computing the covariance of the error vector. This matrix is considered to be constant over time and for all targets.

B. Definition of $\boldsymbol{\psi}(\cdot)$ using kernel methods

In this paragraph, the objective is to determine the aforementioned function $\boldsymbol{\psi}(\cdot)$ that associates to each RSSI vector $\boldsymbol{\rho}_\ell$ the corresponding position $\boldsymbol{p}_\ell$. Determining $\boldsymbol{\psi}(\cdot)$ requires solving a nonlinear regression problem. We take advantage of kernel methods [25], [26], that have been remarkably successful for solving such problems. Let the vector-valued function $\boldsymbol{\psi}(\cdot)$ be decomposed into D real-valued functions, namely $\boldsymbol{\psi}(\cdot) = (\psi_1(\cdot) \ \ldots \ \psi_D(\cdot))$, where $\psi_d \colon \mathbb{R}^{N_a} \mapsto \mathbb{R}, d \in \{1, \ldots, D\}$, estimates the d-th coordinate in $\boldsymbol{p}_\ell = (p_{\ell,1} \ \ldots \ p_{\ell,D})$, for an input $\boldsymbol{\rho}_\ell$. Let $\boldsymbol{P} = (\boldsymbol{p}_1^\top \ \ldots \ \boldsymbol{p}_{N_p}^\top)^\top$. The matrix $\boldsymbol{P}$ is then of size N_p-by-D having $p_{\ell,d}$ for the (ℓ, d)-th entry, and $\boldsymbol{p}_\ell$ for the ℓ-th row. In the following, we denote $\boldsymbol{p}_\ell$ by $\boldsymbol{P}_{\ell,*}$ and the d-th column of $\boldsymbol{P}$ by $\boldsymbol{P}_{*,d}$. Therefore, the vector $\boldsymbol{P}_{*,d}$ holds all N_p points for a fixed coordinate d.

Two different machine learning techniques are investigated in the following: the ridge regression and the vector-output regularized least squares. The kernel-based ridge regression is considered in Subsection IV-B1, where D optimization problems are set separately to define the D models $\psi_1(\cdot), \ldots, \psi_D(\cdot)$. In Subsection IV-B2, we explore multi-task learning to determine a vector-output model $\boldsymbol{\psi}(\cdot)$ that estimates simultaneously all D coordinates.

1) Ridge regression: The kernel-based ridge regression is considered in this paragraph to determine the D models, $\psi_1(\cdot), \ldots, \psi_D(\cdot)$, by setting D separate optimization problems. Indeed, each function $\psi_d(\cdot)$ is estimated by minimizing the mean quadratic error between the model's outputs $\psi_d(\boldsymbol{\rho}_\ell)$ and the desired outputs $p_{\ell,d}$:

$$\min_{\psi_d \in \mathcal{H}} \frac{1}{N_p} \sum_{\ell=1}^{N_p} ((p_{\ell,d} - \psi_d(\boldsymbol{\rho}_\ell))^2 + \eta \|\psi_d\|_{\mathcal{H}}^2, \tag{24}$$

where η is a positive tunable parameter that controls the tradeoff between the fitness error and the complexity of the solution, as measured by the norm in the Reproducing Kernel Hilbert Space $\mathcal{H}$. According to the representer theorem [28], [26], the optimal function can be written as follows:

$$\psi_d(\cdot) = \sum_{\ell=1}^{N_p} \alpha_{\ell,d} \, \kappa(\boldsymbol{\rho}_\ell, \cdot), \tag{25}$$

where $\kappa : \mathbb{R}^{N_s} \times \mathbb{R}^{N_s} \mapsto \mathbb{R}$ is a reproducing kernel, and $\alpha_{\ell,d}, \ell \in \{1, \ldots, N_p\}$, are parameters to be estimated. Let $\boldsymbol{\alpha}$ be the $N_p \times D$ matrix whose (ℓ, d)-th entry is $\alpha_{\ell,d}$, and whose d-th column is denoted by $\boldsymbol{\alpha}_{*,d}$ and ℓ-th row by $\boldsymbol{\alpha}_{\ell,*}$.

By injecting (25) in (24), we get a dual optimization problem in terms of $\boldsymbol{\alpha}_{*,d}$, whose solution is given by taking its derivative with respect to $\boldsymbol{\alpha}_{*,d}$ and setting it to zero. One can easily find the following form of the solution:

$$\boldsymbol{\alpha}_{*,d} = (\boldsymbol{K} + \eta N_p \mathbf{I}_{N_p})^{-1} \boldsymbol{P}_{*,d}, \tag{26}$$

where $\mathbf{I}_{N_p}$ is the N_p-by-N_p identity matrix, and $\boldsymbol{K}$ is the $N_p \times N_p$ matrix whose (i, j)-th entry is $\kappa(\boldsymbol{\rho}_i, \boldsymbol{\rho}_j)$, for $i, j \in \{1, ..., N_p\}$. For an appropriate value of the regularization parameter η, the matrix between parenthesis is always non-singular.

One can see that the same matrix $(\boldsymbol{K} + \eta N_p \mathbf{I}_{N_p})$ needs to be inverted in order to estimate each coordinate. To reduce the computational complexity, all D estimations are collected in a single matrix inversion problem, as follows:

$$\boldsymbol{\alpha} = (\boldsymbol{K} + \eta N_p \mathbf{I}_{N_p})^{-1} \boldsymbol{P}. \tag{27}$$

We then define a model that allows us to estimate all D coordinates at once, using equation (25) and the definition of the vector of functions $\boldsymbol{\psi}(\cdot)$, as follows:

$$\boldsymbol{\psi}(\cdot) = \sum_{\ell=1}^{N_p} \boldsymbol{\alpha}_{\ell,*} \, \kappa(\boldsymbol{\rho}_\ell, \cdot). \tag{28}$$

2) Vector-output regularized least squares: In this paragraph, we take advantage of multi-task learning by using the vector-output regularized least squares (vo-RLS) algorithm [29] to estimate all D coordinates at once. Instead of estimating the set of functions $\psi_d(\cdot)$, we now determine a 1-by-D vector-output function $\boldsymbol{\psi}(\cdot)$.

In multi-task learning, $\boldsymbol{\psi}(\cdot)$ takes the form:

$$\boldsymbol{\psi}(\cdot) = \sum_{\ell=1}^{N_p} \beta_\ell \, \boldsymbol{P}_{\ell,*} \, \kappa(\boldsymbol{\rho}_\ell, \cdot), \tag{29}$$

where $\beta_\ell, \ell \in \{1, \ldots, N_p\}$, are parameters to be defined. As for the optimization problem, the objective stays the same. Indeed, the function $\boldsymbol{\psi}(\cdot)$ is determined by minimizing the mean quadratic error between the model's outputs $\boldsymbol{\psi}(\boldsymbol{\rho}_\ell)$ and the desired outputs $\boldsymbol{P}_{\ell,*}$, namely

$$\min_{\boldsymbol{\psi}} \frac{1}{N_p} \sum_{\ell=1}^{N_p} \|\boldsymbol{P}_{\ell,*} - \boldsymbol{\psi}(\boldsymbol{\rho}_\ell)\|^2 + \eta \|\boldsymbol{\beta}\|^2, \tag{30}$$

where $\boldsymbol{\beta} = \begin{pmatrix}\beta_1 & \dots & \beta_{N_p}\end{pmatrix}^\top$. Substituting the expressio of $\boldsymbol{\psi}(\cdot)$ from (29) in the optimization problem (30), we ge in matrix form, the following problem formulation:

$$\min_{\boldsymbol{\beta}} \operatorname{tr}(\boldsymbol{P}\boldsymbol{P}^\top) - 2\,\boldsymbol{\xi}^\top\boldsymbol{\beta} + \boldsymbol{\beta}^\top\boldsymbol{G}\boldsymbol{\beta} + \eta N_p\boldsymbol{\beta}^\top\boldsymbol{\beta}, \quad (31$$

where $\operatorname{tr}(\cdot)$ is the matrix trace operator, $\boldsymbol{G}$ is the N_p-by-N matrix whose (j,k)-th entry is

$$\boldsymbol{P}_{j,*}\,\boldsymbol{P}_{k,*}^\top \sum_{i=1}^{N_p} \kappa(\boldsymbol{\rho}_j,\boldsymbol{\rho}_i)\,\kappa(\boldsymbol{\rho}_k,\boldsymbol{\rho}_i),$$

and $\boldsymbol{\xi}$ is the N_p-by-1 vector whose j-th entry is

$$\sum_{k=1}^{N_p} \boldsymbol{P}_{j,*}\,\boldsymbol{P}_{k,*}^\top\,\kappa(\boldsymbol{\rho}_j,\boldsymbol{\rho}_k).$$

By taking the gradient of the objective function in (31) wit respect to $\boldsymbol{\beta}$, namely $-\boldsymbol{\xi} + \boldsymbol{G}\boldsymbol{\beta} + \eta N_p\boldsymbol{\beta}$, and setting it t zero, we obtain the final solution:

$$\boldsymbol{\beta} = (\boldsymbol{G} + \eta N_p \mathbf{I}_{N_p})^{-1}\boldsymbol{\xi}.$$

V. Practical simulations and results

In this section, we evaluate the performance of our method on simulated data. In the first paragraph, several trajectories with different orders for the state-space model are examined. In the second paragraph, we study the impact of the noises standard deviations $\boldsymbol{\sigma}_{\boldsymbol{\gamma}}$ and $\sigma_{\boldsymbol{\rho}}$ on the estimation error. In the third paragraph, we study the impact of the number of stationary sensors and the number of reference positions on the estimation error. Finally, results are compared to ones obtained with the WKNN algorithm combined with a Kalman filter [30] and tracking using particle filtering [31].

The same practical setup is considered for the two following paragraphs, given as follows. We consider a $100m \times 100m$ area, and generate 16 stationary sensors and 100 reference positions uniformly distributed over the area. The RSSI values are obtained using the well-known Okumura-Hata model [32] given by:

$$\rho_{\boldsymbol{s}_i,\boldsymbol{p}_\ell} = \rho_0 - 10\,n_P\,\log_{10}\|\boldsymbol{s}_i - \boldsymbol{p}_\ell\| + \varepsilon_{i,\ell}, \quad (32)$$

where $\rho_{\boldsymbol{s}_i,\boldsymbol{p}_\ell}$ (in dBm) 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_0 is the initial power (in dBm) set to 100, n_P is the path-loss exponent set to 4 as often given in the literature, $\|\boldsymbol{s}_i - \boldsymbol{p}_\ell\|$ is the Euclidian distance between the position $\boldsymbol{p}_\ell$ of the considered node and the position $\boldsymbol{s}_i$ of a stationary sensor, and $\varepsilon_{i,\ell}$ is the noise affecting the RSSI measures with $\sigma_{\boldsymbol{\rho}}$ its standard deviation. We also generate a trajectory and calculate the RSSI values collected by the moving target using (32). For the definition of $\boldsymbol{\psi}(\cdot)$ using kernel methods, we consider the Gaussian kernel given by:

$$\kappa(\boldsymbol{\rho}_u,\boldsymbol{\rho}_{u'}) = \exp\left(\frac{-\|\boldsymbol{\rho}_u - \boldsymbol{\rho}_{u'}\|^2}{2\sigma^2}\right),$$

where σ is its bandwidth that controls, together with the regularization parameter η, the degree of smoothness, noise

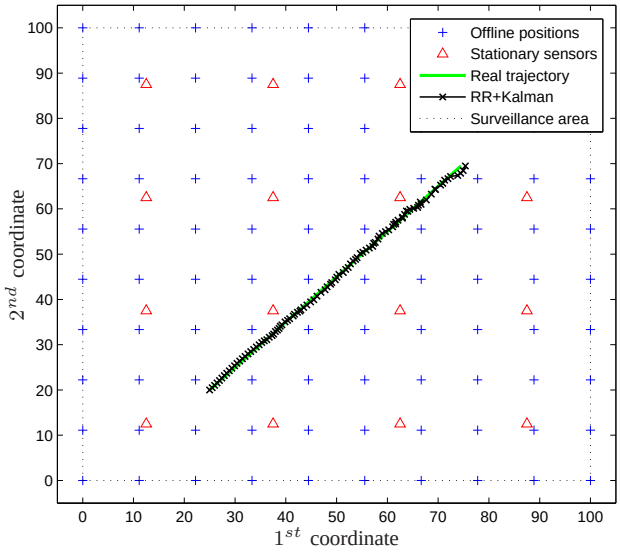


Fig. 2: Estimation of the first trajectory.

tolerance, and generalization of the solution. The choice of the values for η and σ is done using a grid search on $\eta N_p = 2^r$ with $r \in \{-20,-19,\cdots,-1\}$ and $\sigma = 2^{r'}$ with $r' \in \{1,2,\cdots,10\}$, where the corresponding error is estimated using the 10-fold cross-validation scheme. This scheme consists of dividing the data into 10 folds: 9 for training the model and the remaining one for validating it [33].

A. Evaluation of our method on three trajectories

We consider three different trajectories of 100 points with $\Delta t = 1s$. For the trajectory illustrated in Fig. 2, the accelerations are assumed equal to zero, leading to constant velocities. As for the second and the third trajectories of Fig. 3 and Fig. 4 respectively, their respective accelerations are given in the top plots and in the bottom plots of Fig. 5, γ_1 and γ_2 being the first and the second acceleration coordinates respectively. One can see that the accelerations of the third trajectory have more variations than the accelerations of the second trajectory. The coordinates expressions are obtained by taking twice the primitive integral of the accelerations. By taking these three trajectories, the performance of the proposed method is evaluated for different types of scenarios, considering first a monotonously moving target, then more hyperactive ones.

Since a noiseless setup is not realistic in a practical environment, we consider that noises are present in all scenarios. Here, we take both components of $\boldsymbol{\sigma}_{\boldsymbol{\gamma}}$ equal to $0.01m/s^2$, and $\sigma_{\boldsymbol{\rho}}$ equal to $1dB$. Let the estimation error be evaluated by the root mean squared distance between the exact positions and the estimated ones. Fig. 2, Fig. 3, and Fig. 4 show the estimated trajectories when using the proposed method with the ridge regression (RR) for the third-order state-space model described in Section III. Table I shows the average over 50 simulations of the estimation

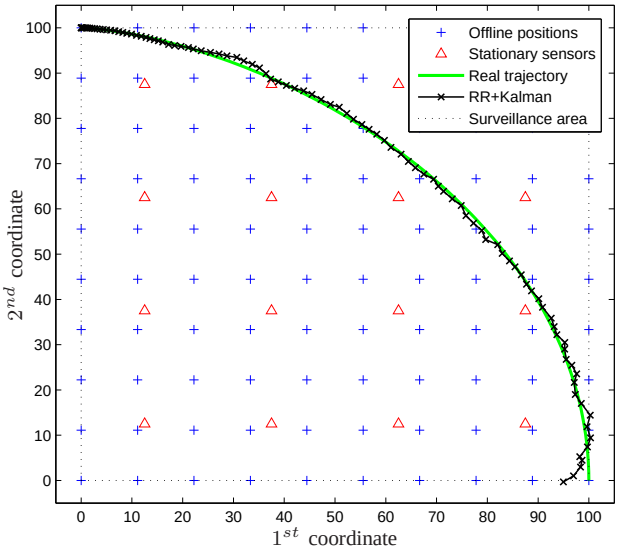


Fig. 3: Estimation of the second trajectory.

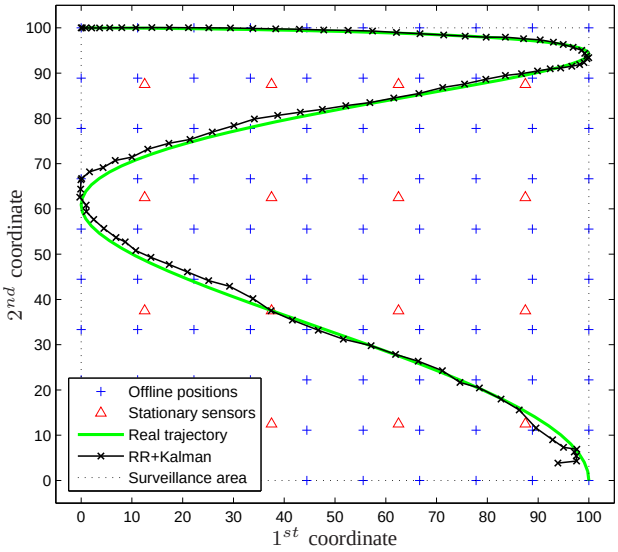


Fig. 4: Estimation of the third trajectory.

errors for the three trajectories and the three different state-space models, using the RR and the vo-RLS in the learning process. The three models yield almost the same results for the first two trajectories. However, for the third trajectory, the smallest estimation error is obtained when using the third-order state-space model. This result is expected since the accelerations in this trajectory have high variations, and as explained in Section III, the third-order state-space model is well suited for such cases.

B. Impact of $\boldsymbol{\sigma_\gamma}$ and $\sigma_{\boldsymbol{\rho}}$

In this section, we will test our method using the trajectory of Fig. 4, where the general case of a hyperactive target

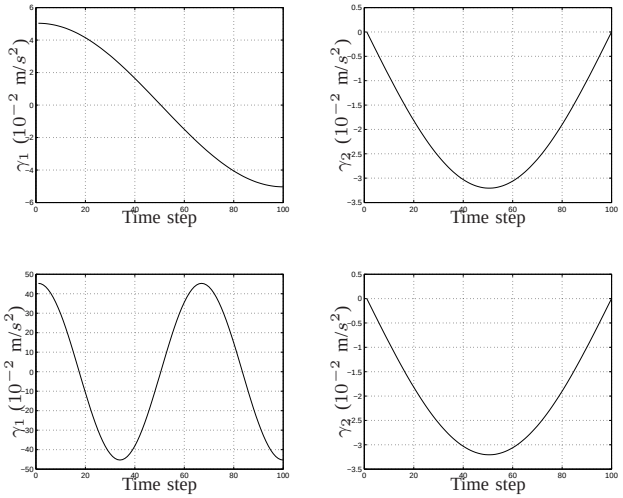


Fig. 5: Acceleration signals for the second trajectory in the top plots and for the third trajectory in the bottom plots.

TABLE I: Estimation errors (in meters) for different orders of the state-space models and the three trajectories.

	Traj. 1	Traj. 2	Traj. 3
RR + model order 1	0.94	1.12	2.15
RR + model order 2	0.94	1.10	1.97
RR + model order 3	0.93	1.09	1.09
vo-RLS + model order 1	1.17	1.41	2.43
vo-RLS + model order 2	1.16	1.38	2.25
vo-RLS + model order 3	1.14	1.36	1.21

is considered. The third-order state-model from Section III-C is used since it yields the best results as shown in the previous section. Indeed, even though the first-order model and the second-order model yield good results for the trajectories of Fig. 2 and 3, the estimation error increases significantly compared to the third-order model when the target is hyperactive (Fig. 4) as shown in Table I.

Let us now study the impact of the noises standard deviations $\boldsymbol{\sigma_\gamma}$ and $\sigma_{\boldsymbol{\rho}}$ on the estimation error. We first take different percentages of the standard deviation of the acceleration, going from 1% to 10%, along with a fixed $\sigma_{\boldsymbol{\rho}}$ equal to 5% of standard deviation of the RSSI measures, and repeat the simulation 50 times to obtain an average value of the estimation errors. The average value of the estimation errors is stored. It is worth noting that the standard deviation of the RSSI is equal to $10.79dBm$; therefore, $\sigma_{\boldsymbol{\rho}}$ is equal to $0.54dBm$. Fig. 6 shows the impact of the variation of $\boldsymbol{\sigma_\gamma}$ on the estimation error. One can see that the results obtained in Fig. 6 with the ridge regression and the vo-RLS are independent from the acceleration noise, whereas estimations using only accelerometer information are highly affected by the variations of $\boldsymbol{\sigma_\gamma}$. The RR combined with

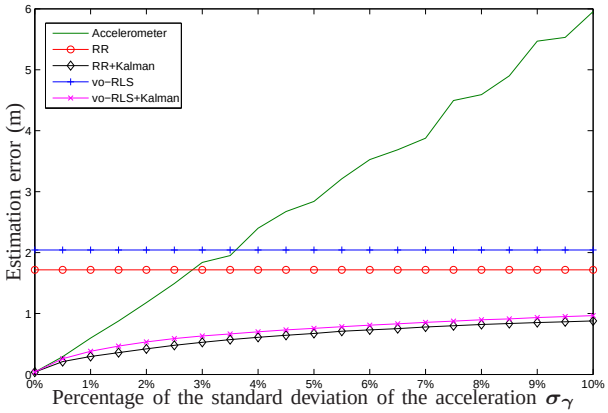


Fig. 6: Estimation error as a function of the noise on the accelerations.

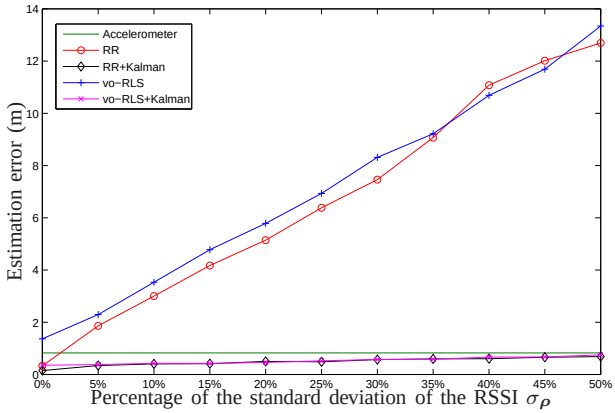


Fig. 7: Estimation error as a function of the noise on the RSSI.

the Kalman filter yields the best results. In fact, the filter corrects the results, and the error is always smaller than the error in the case of the ridge regression alone, and around $\boldsymbol{\sigma_\gamma}$ equal to 7% of the standard deviation of the acceleration, the error becomes constant.

We then take several percentages of the standard deviation of the RSSI measures, going from 0% to 50%, with $\boldsymbol{\sigma_\gamma}$ fixed to 1% of the standard deviation of the acceleration. $\sigma_{\boldsymbol{\rho}}$ is then varying from $0dBm$ to $5.40dBm$. The simulation is also repeated 50 times, and the errors are averaged over the 50 simulations. Fig. 7 shows the impact of the variation of σ_ρ on the estimation error. One can see that localization using only accelerometer information is independent from the noise on the RSSIs, which is expected. The RR and the vo-RLS are highly affected by the noise variations, since they use these RSSI measurements for the estimation. As for the method combining the RR with the Kalman filter, it outperforms the method using only accelerations. It is interesting here to see the effectiveness of the Kalman filter. Indeed, one can see in Fig. 7 that the RR used alone yields better results than the vo-RLS also used alone; however, after adding the Kalman filter, the results of the two techniques become very similar and the error becomes almost constant for both methods when $\sigma_{\boldsymbol{\rho}}$ exceeds 30% of the standard deviation of the RSSI measures.

C. Impact of N_s and N_p

We now study the impact of the number of stationary sensors N_s and the number of reference positions N_p on the performance of the tracking method. We consider the trajectory of Fig. 4, with both components of $\boldsymbol{\sigma_\gamma}$ equal to $0.01m/s^2$ and $\sigma_{\boldsymbol{\rho}}$ equal to $1dB$. The ridge regression is used in this section. Nevertheless, it is worth noting that varying N_s and N_p has the same impact on the tracking method if we use the vo-RLS in the learning process.

We first vary the number of stationary sensors ($N_s = 1^2, \ldots, 15^2$), while keeping a fixed number for the reference positions ($N_p = 100$). Fig. 8 shows the evolution of the estimation error in terms of the number of stationary sensors. We then take a fixed number of stationary sensors equal to 16, and we vary the number of reference positions, $N_p = 5^2, \ldots, 25^2$. Fig. 9 shows the evolution of the estimation error in terms of the number of reference positions. By comparing the obtained results, one can notice that both the increase in the number of stationary sensors and in the number of reference positions yield a better estimation of the target's positions. Indeed, Fig. 8 shows that when using 16 stationary sensors, the average over 50 simulations of the estimation error is $1.10m$ compared to an error of $0.82m$ when using $11^2 = 121$ stationary sensors. Fig. 9 shows that for $N_p = 100$, the average over 50 simulations of the estimation error is $1.08m$ compared to an error of $0.68m$ when increasing N_p to $24^2 = 576$. In fact, with a higher number of stationary sensors and reference positions, we get better coverage and knowledge of the environment, which explains the improvement in the results. However, increasing the number of stationary sensors increases the total cost in material, while increasing the number of reference positions induces a significant increase in the algorithm's complexity. Therefore, depending on the practical system constraints, a tradeoff should be found between the algorithm's accuracy and the computational load.

D. Comparison to other tracking techniques

The objective in this section is to compare the proposed method to two recently proposed tracking methods. For the first comparison, we use the method proposed in [22], that also makes use of the Kalman filter to correct the trajectory estimated by radio-fingerprints. We then compare our method to the centralized version of the method described in [31], which involves the use of a particle filter and RSSI measurements. We consider the three trajectories described in Section V-A for our comparisons. In order to have a fair comparison of our technique towards these two methods, we consider a setup that is the closest possible to the one the authors use in their papers. For this purpose, we take the number of stationary sensors $N_s = 4$, even though taking

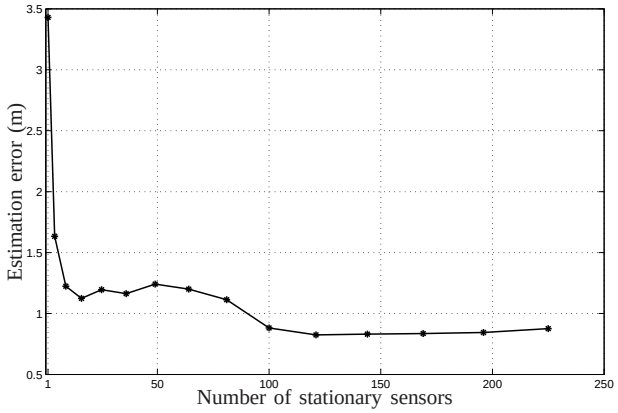


Fig. 8: Estimation error as a function of the number of stationary sensors.

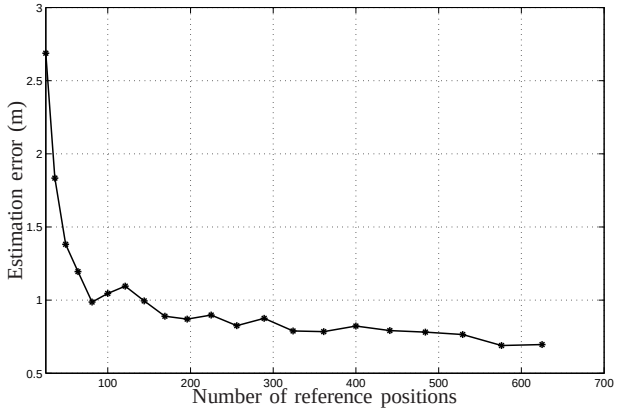


Fig. 9: Estimation error as a function of the number of reference positions.

$N_s = 16$ gives better results in the case of our method as one can see from Fig. 8. We take $\sigma_\rho = 1dB$ and both components of $\boldsymbol{\sigma_\gamma}$ equal to $0.01m/s^2$.

We proceed by briefly describing the method in [22]. It consists of estimating the position using the weighted K-nearest 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 weight used for the WKNN algorithm in [22] is given by:

$$w_n = \frac{1/\delta_n}{\sum_{z \in I} 1/\delta_z},$$

where δ_n is the Euclidean distance between the RSSI vector $\boldsymbol{\rho}(k)$ of the target at time step k and $\boldsymbol{\rho}_n$, $n \in I$, and I is the set of indices of $\boldsymbol{\rho}_\ell$ of the database yielding the K smallest distances (i.e., K nearest neighbors) δ_ℓ at time step k. The estimated target's position is then given by: $\sum_{n \in I} w_n \, \boldsymbol{p}_n$. The number of neighbors K is taken equal to 8 as in the simulations of [22]. As for the correction using the Kalman filter, the authors use a second-order state-space model similar to the one in Section III-B. The estimation errors (in meters) obtained when using this algorithm for the three trajectories of Section V-A are computed 50 times for each case, and their averages are shown in Table II. Our method clearly outperforms the one in [22]. Indeed, the estimation error obtained with the proposed method is significantly smaller than the one obtained with the WKNN algorithm followed by the Kalman filter, for all three types of trajectories.

TABLE II: Comparison of the estimation error (in meters) for the three tracking methods.

	Trajectory 1	Trajectory 2	Trajectory 3
Particle filter	1.42	6.25	8.17
WKNN+Kalman	1.30	3.93	4.23
Proposed method	1.27	1.91	2.38

As for the second method used for our comparison, it employs a particle filter along with RSSI measurements and a first-order state-space model [31]. The particle filter approximates the minimum mean-square error (MMSE) estimate of the emitter state given all present and past observations, i.e. RSSI measurements. It seeks to represent the posterior distribution of the hidden states by a properly weighted set of time-varying random samples such that, as the number of samples go to infinity, the weighted average of those samples converges at each time step, in some statistical sense, to the true global MMSE estimate of the current unknown states given all present and past network measurements [31]. For the first trajectory, we used the first-order state-space model as described in the authors work. The average estimation error obtained is close to the error obtained with our tracking method. However, for the second and third trajectories, we used the second-order state-space model, because the first-order model did not work well due to the abrupt trajectory variations. One can see from Table II that our tracking method still outperforms the well-known tracking technique based on particle filtering.

VI. Conclusion

In this paper, we proposed a new method for target tracking in wireless sensor networks by combining machine learning and Kalman filtering. For the learning process, we investigated the use of two kernel-based machine learning algorithms: the ridge regression and the vector-output regularized least squares. We also described three different orders for the state-space models to be used in the Kalman filtering, and highlighted the difference between them and how they can affect the performance of the tracking procedure. Simulation results showed that the proposed method outperforms two recently developed approaches. The method allows accurate tracking, and is proved to be robust in the case of noisy data, whether

the noise affects the acceleration information or the RSSI measures. Future works will handle further improvements of this method, such as introducing a model that estimates distances between sensors instead of positions. Solutions to cases where zones of the surveillance area are not covered by all stationary sensors could also be provided.

REFERENCES

[1] N. Suryadevara and S. Mukhopadhyay, "Wireless sensor network based home monitoring system for wellness determination of elderly," *IEEE Sensors Journal*, vol. 12, no. 6, pp. 1965–1972, 2012.

[2] H. Ramamurthy, B. S. Prabhu, R. Gadh, and A. Madni, "Wireless industrial monitoring and control using a smart sensor platform," *IEEE Sensors Journal*, vol. 7, no. 5, pp. 611–618, 2007.

[3] G. Cote, R. Lec, and M. Pishko, "Emerging biomedical sensing technologies and their applications," *IEEE Sensors Journal*, vol. 3, no. 3, pp. 251–266, 2003.

[4] D. Li, K. Wong, Y. H. Hu, and A. Sayeed, "Detection, classification, and tracking of targets," *IEEE Signal Processing Magazine*, vol. 19, no. 2, pp. 17–29, 2002.

[5] A. Dallil, M. Oussalah, and A. Ouldali, "Sensor fusion and target tracking using evidential data association," *IEEE Sensors Journal*, vol. 13, no. 1, pp. 285–293, 2013.

[6] 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.

[7] 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.

[8] 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 *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2012, pp. 1–10.

[9] 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.

[10] I. Jami, M. Ali, and R. Ormondroyd, "Comparison of methods of locating and tracking cellular mobiles," in *IEE Colloquium on Novel Methods of Location and Tracking of Cellular Mobiles and Their System Applications*, 1999, pp. 1/1–1/6.

[11] S. Dias and M. Bruno, "Cooperative target tracking using decentralized particle filtering and rss sensors," *IEEE Transactions on Signal Processing*, vol. 61, no. 14, pp. 3632–3646, 2013.

[12] F. Mourad, H. Snoussi, M. Kieffer, and C. Richard, "Robust bounded-error tracking in wireless sensor networks," *16th IFAC Symposium on System Identification, SYSID, 2012*.

[13] J. Teng, H. Snoussi, and C. Richard, "Decentralized variational filtering for simultaneous sensor localization and target tracking in binary sensor networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing,ICASSP.*, 2009, pp. 2233–2236.

[14] L. Zhang, J. Liu, H. Jiang, and Y. Guan, "Senstrack: Energy-efficient location tracking with smartphone sensors," *IEEE Sensors Journal*, vol. 13, no. 10, pp. 3775–3784, 2013.

[15] Y. Li and J. Li, "Robust adaptive kalman filtering for target tracking with unknown observation noise," in *24th Chinese Control and Decision Conference (CCDC)*, 2012, pp. 2075–2080.

[16] S. Jayamohan and M. Mathurakani, "Noise tolerance analysis of marginalized particle filter for target tracking," in *Annual International Conference on Emerging Research Areas and International Conference on Microelectronics, Communications and Renewable Energy (AICERA/ICMiCR)*, 2013, pp. 1–6.

[17] T.-N. Lin and P.-C. Lin, "Performance comparison of indoor positioning techniques based on location fingerprinting in wireless networks," in *Wireless Networks, Communications and Mobile Computing, International Conference on*, vol. 2, 2005, pp. 1569 – 1574.

[18] L. Chen, H. Wu, M. Jin, and G. Chen, "Homogeneous features utilization to address the device heterogeneity problem in fingerprint localization," *IEEE Sensors Journal*, vol. PP, no. 99, pp. 1–1, 2013.

[19] H. Koyuncu and S. H. Yang, "A 2D positioning system using WSNs in indoor environment," *International Journal of Electrical and Computer Sciences IJECS-IJENS*, vol. 11, no. 3, 2011.

[20] 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.

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

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

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

[24] G. Welch and G. Bishop, "An Introduction to the Kalman Filter: SIGGRAPH 2001 Course 8," in *Computer Graphics, Annual Conference on Computer Graphics & Interactive Techniques*, 2001, pp. 12–17.

[25] V. N. Vapnik, *Statistical Learning Theory*. John Wiley and Sons, 1998.

[26] 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, UK: Springer-Verlag, 2001, pp. 416–426.

[27] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and Computing*, vol. 14, no. 3, pp. 199–222, Aug. 2004. [Online]. Available: http://dx.doi.org/10.1023/B:STCO.0000035301.49549.88

[28] G. Kimeldorf and G. Wahba, "Some results on tchebycheffian spline functions," *Journal of Mathematical Analysis and Applications*, vol. 33, pp. 82 – 95, 1971.

[29] P. Honeine, Z. Noumir, and C. Richard, "Multiclass classification machines with the complexity of a single binary classifier," *Signal Processing*, vol. 93, no. 5, pp. 1013–1026, May 2013. [Online]. Available: http://dx.doi.org/10.1016/j.sigpro.2012.11.009

[30] 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.

[31] S. Dias and M. Bruno, "Cooperative particle filtering for emitter tracking with unknown noise variance," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012, pp. 2629–2632.

[32] 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.

[33] M. Stone, "Cross-validatory choice and assessment of statistical predictions," *J. Royal Stat. Soc.*, vol. 36, no. 2, pp. 111 – 147, 1974.