sensors

MDPI

Article

Dynamic Indoor Localization Using Maximum Likelihood Particle Filtering

Wenxu Wang [1], Damián Marelli [1,2,*] and Minyue Fu [1,3]

[1] School of Automation, Guangdong University of Technology, Guangzhou 510006, China; wangwenxu0909@126.com (W.W.); minyue.fu@newcastle.edu.au (M.F.)
[2] National Scientific and French Argentine International Center for Information and Systems Sciences, Technical Research Council, Rosario S2000, Argentina
[3] School of Electrical Engineering and Computer Science, University of Newcastle, Newcastle, NSW 2308, Australia
* Correspondence: damian.marelli@newcastle.edu.au

Abstract: A popular approach for solving the indoor dynamic localization problem based on WiFi measurements consists of using particle filtering. However, a drawback of this approach is that a very large number of particles are needed to achieve accurate results in real environments. The reason for this drawback is that, in this particular application, classical particle filtering wastes many unnecessary particles. To remedy this, we propose a novel particle filtering method which we call maximum likelihood particle filter (MLPF). The essential idea consists of combining the particle prediction and update steps into a single one in which all particles are efficiently used. This drastically reduces the number of particles, leading to numerically feasible algorithms with high accuracy. We provide experimental results, using real data, confirming our claim.

Keywords: indoor tracking; particle filter; channel state information; WiFi fingerprinting

check for updates

Citation: Wang, W.; Marelli, D.; Fu, M. Dynamic Indoor Localization Using Maximum Likelihood Particle Filtering. *Sensors* **2021**, *21*, 1090. https://doi.org/10.3390/s21041090

Academic Editor: Paolo Spagnolo

Received: 17 December 2020
Accepted: 28 January 2021
Published: 5 February 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

1. Introduction

Wireless positioning is the most popular approach for indoor location-based services. It finds applications in emergency rescue, smart home, security monitoring and many other areas. In contrast to outdoor positioning, whose dominant approach is based on the global positioning system (GPS), indoor positioning is hindered by multiple sources of interference and multipath effects in time-varying environments. In view of this, multiple techniques for indoor positioning are currently being investigated. The most popular ones are based on radio frequency identification [1,2], Bluetooth signals [3,4], ultra-wideband signals [5,6] and WiFi signals [7,8]. Due to their large coverage, low implementation cost and availability in existing mobile devices, WiFi-based indoor positioning is the most promising technique.

A broad classification of indoor localization methods can be done in two categories, namely, geometric mapping and fingerprinting [9]. In geometric mapping, the target's position is estimated using geometric information like distances and angles from the target to a set of access points (APs) [10]. Typical techniques for obtaining geometric information are time of flight (ToF) [11] and angle of arrival (AoA) [12]. This approach typically requires obtaining information from more than three APs. the resulting methods are based on line of sight (LOS) signals, and therefore are easily influenced by obstacles. Strict synchronization is also needed between different devices. The above limitations are overcome by the fingerprinting method [13]. It consists of building a database containing the values of certain received signal features at a given set of known locations. Localization is then done by comparing the features obtained at the target's location with those of the database. Localization based on fingerprinting is more immune to distortions caused by the environment, since the latter are coded in the feature database. However, it suffers from poor positioning accuracy when the target is in a position far from anyone in the database.

The most commonly WiFi signal feature used for indoor localization is its received signal strength (RSS) [14]. It measures the signal power strength at the receiver side. Although RSS measurements are easy to obtain, only a very rough estimation of the receiver's position can be obtained from it. This is due to the fading and distortion suffered by signals propagating through multipaths. To tackle this problem, in recent years channel state information (CSI) has been used instead of RSS [9]. CSI contains information of the phase and amplitude value of each subcarrier of an orthogonal frequency division multiplexing (OFDM) channel. It can be readily obtained from any OFDM system based on the 802.11n protocol. Since multipath and fading information is represented in the CSI, its use for indoor localization leads to more accurate and stable results in comparison to RSS.

To improve the accuracy of any given localization method, extra information can be obtained from a motion model of the target. The common solution for fusing the information from the motion model with that from measurements consists of using some Bayesian tracking technique. We refer to resulting techniques as dynamic localization methods [15]. One such techniques is the Kalman filter [16–18]. While being optimal and having a closed-form, it only applies to linear Gaussian model. This assumption can be relaxed using the extended Kalman filter [19]. However, since this method is simply based on linearizing a non-linear model, it often leads to inaccurate results. This can be overcome by using particle filtering [20–23]. This essentially consists of approximating non-Gaussian probability distributions using a number of samples (particles) multiplied by their associated weights.

As it is known, a Bayesian tracking method consists of alternating two steps called prediction and update. In a particle filter, a set of new particles is drawn during each prediction step, whose weighs are determined during the subsequent update step. A drawback of this approach is that this weighting results in little use of those particles whose weights are very small. The number of wasted particles can be very large in the case where measurements already provide enough information to determine the target's location with a reasonable accuracy, without using the information provided by the motion model. This is often the case in localization setups, for otherwise, algorithms not using a motion model would not work with reasonable accuracy. Consequently, a very large number of particles are required to avoid particle depletion after the update step, which would result in an inaccurate result. The vast number of required particles renders the approach numerically intractable in many applications.

In order to tackle the above shortcoming, a number of methods are available to reduce the estimation error resulting from using particle filtering with a restricted number of particles. In [21], a machine learning classifier is added to decide whether particles are in the correct room. In [22], a context variable, with values in the set {'free space', 'constrained space', 'static space'}, is added to the state to account for the movement constraints at the target's current position. Also, the authors of [23] divide the environment into reachable and unreachable areas, so as to discard particles falling into an unreachable area.

Generally speaking, the above methods aim to reduce the particle depletion phenomenon described above by complementing the motion model with environment-dependent information to decide where to draw particles. The aim of this work is to make the most of this principle, i.e., ideally to totally avoid particle depletion. To this end, we propose a novel particle filtering scheme which combines the prediction and update steps into a single one. In other words, we avoid drawing unnecessary particles during the prediction step that will afterwards disappear during the subsequent update step. This drastically reduces the required number of particles, making the approach numerically feasible in virtually any application. In deriving our method, we exploit the principle underlying a Bayesian tracking method called maximum likelihood Kalman filtering (MLKF) [24]. We then call the proposed method the maximum likelihood particle filter (MLPF).

The authors used the MLKF in [25] to propose a dynamic localization method which only applies to the case in which the motion model is linear and Gaussian. In this sense, the method proposed in this work can be thought of as the generalization of the method we proposed in [25] to the case of non-linear motion models. This kind of models are typically used in robotics [26], and as we show with experiments using real data, leads to great accuracy improvements.

The rest of the paper is organized as follows: In Section 2, we state the dynamic localization problem. In Section 3, we describe the particle filtering approach for solving it. In Section 4, we derive the proposed MLPF method. In Section 5, we present experimental evidence, showing the numerical advantages offered by the proposed method. Concluding remarks are given in Section 6. For ease of readability, all proofs appear in Appendix C .

2. Problem Description

We assume there is a target moving in an indoor environment. The target's Cartesian coordinates at time t are $y_t = [a_t, b_t]^\top \in \mathbb{R}^2$ and $\theta_t \in (-\pi, \pi]$ is its orientation in radians, as shown in Figure 1. The target's motion model is described by the following non-linear state-transition equation

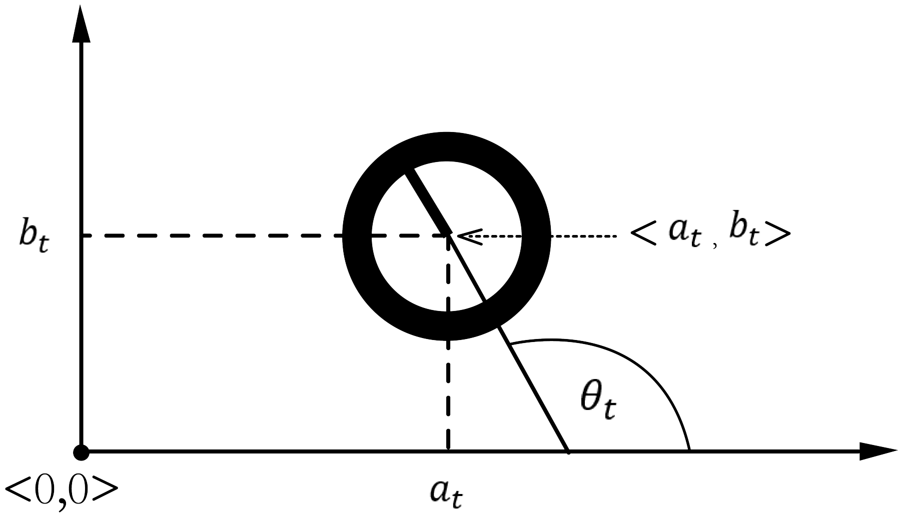

Figure 1. Target's Cartesian coordinates.

$$y_{t+1} = g(y_t, \theta_t, u_t, e_t), \tag{1}$$
$$\theta_{t+1} = h(y_t, u_t, e_t) + a(y_t, u_t, e_t)\theta_t + b(y_t, u_t, e_t)n_t, \tag{2}$$

where u_t is the target's control input and $e_t = \left[\tilde{a}_t, \tilde{b}_t\right]^\top$ and n_t are process noises accounting for the model's inaccuracy. To write the above in compact form we define the target's pose $x_t = \left[y_t^\top, \theta_t\right]^\top \in \mathbb{R}^3$ and the process noise $w_t = \left[e_t^\top, n_t\right]$. We then write

$$x_{t+1} = f(x_t, u_t, w_t), \tag{3}$$

with f defined according to (1) and (2). We assume that $x_1 = 0$ and the sequence $w_t \sim \mathcal{N}(0, Q(u_t))$ is i.i.d., with $Q(u)$ being a diagonal positive definite matrix for each u.

Remark 1. *We state the motion model in the general form (1) in order to make our presentation valid for the different models used in the literature. In Appendix A we briefly describe the most popular models falling into this general form.*

At each time $t \in \mathbb{N}$, we obtain the following measurement

$$z_t = m(y_t) + v_t. \tag{4}$$

where the map $m : \mathbb{R}^2 \to \mathbb{R}^N$ is known and the measurement noise $v_t \sim \mathcal{N}(0, R)$ is i.i.d. and independent of w_t.

Problem 1. *Let $Z_T^\top = \left[z_1^\top, \dots, z_T^\top\right]$ denote the set of all measurements available up to time T. The dynamic localization problem consists of using, at time T, the measurements Z_T to obtain an estimate $\hat{x}_T$ of x_T.*

3. Available Solution Using Particle Filtering

As described in Section 1, the dynamic localization problem can be solved using the well-known particle filtering method [27]. In this section, we describe this approach.

The Bayesian filtering equations are initialized by $p(x_1|Z_1) = \delta(x_1)$, and proceed as follows

$$p(x_t|Z_{t-1}) = \int p(x_t|x_{t-1})p(x_{t-1}|Z_{t-1})dx_t, \tag{5}$$

$$p(x_t|Z_t) = \frac{p(z_t|x_t)}{p(z_t|Z_{t-1})}p(x_t|Z_{t-1}). \tag{6}$$

Equations (5) and (6) are called the prediction and update steps, respectively. These equations do not have close form expressions, in general. Hence, a numeric method is needed to approximately compute them. This can be done using particle filtering. More precisely, we start with

$$\begin{aligned} p(x_1|Z_1; m) &\simeq \sum_{i=1}^{I} \omega_{1|1}^i \delta\left(x_1 - x_{1|1}^i\right), \\ x_{1|1}^i &= 0, \\ \omega_{1|1}^i &= \frac{1}{I}. \end{aligned} \tag{7}$$

Then, at time $t > 1$, the prediction step is computed by

$$p(x_t|Z_{t-1}; m) \simeq \frac{1}{I}\sum_{i=1}^{I} \delta\left(x_t - x_{t|t-1}^i\right), \tag{8}$$

with $x_{t|t-1}^i$ obtained by drawing it from the following distribution

$$x_{t|t-1}^i \sim \sum_{i=1}^{I} \omega_{t-1|t-1}^i p\left(x_t|x_{t-1|t-1}^i\right)\delta\left(x_t - x_{t-1|t-1}^i\right).$$

Also, the update step is computed by

$$\begin{aligned} p(x_t|Z_t; m) &\simeq \sum_{i=1}^{I} \omega_{t|t}^i \delta\left(x_t - x_{t|t}^i\right), \\ x_{t|t}^i &= x_{t|t-1}^i, \\ \omega_{t|t}^i &\propto p\left(y_t|x_t^i\right). \end{aligned} \tag{9}$$

with $\omega_{t|t}^i$ normalized so that $\sum_{i=1}^I \omega_{t|t}^i = 1$.

The use of particle filtering for solving the localization problem permits accommodating the accuracy by increasing the number of particles. As mentioned in Section 1, a problem that often occurs is that the desired accuracy requires a very large number of particles. This can result in a numerically very expensive algorithm. To address this issue, in the next section we propose a variant of the particle filtering algorithm, which drastically reduces the number of particles required to achieve a given accuracy.

4. Proposed Algorithm

In the particle filtering algorithm described in Section 3, at time t, we use the approximation (9) of the updated distribution $p(x_{t-1}|Z_{t-1})$, together with the state-transition Equation (1), to generate the approximation (8) of the predicted distribution $p(x_t|Z_{t-1})$. The latter is then used in combination with the likelihood function $p(z_t|x_t)$ to approximate the new updated distribution $p(x_t|Z_t)$. This is done by weighting the i-th particle $x^i_{t|t}$ by a weight $\varpi^i_{t|t}$ proportional to the particle's likelihood $p\left(y_t|x^i_t\right)$. As mentioned in Section 1, a drawback of this approach is that this weighting wastes many particles. In order to avoid this, we combine the prediction and update steps into a single one. More precisely, particles for approximating the updated distribution $p(x_t|Z_t)$ are directly drawn using the knowledge of the particle approximation of the previous updated distribution $p(x_{t-1}|Z_{t-1})$, without drawing particles to represent the predicted distribution $p(x_t|Z_{t-1})$. We describe below how this is done.

Notation 1. *In order to simplify the notation we remove the dependence of Q on u_t, i.e., we use Q instead of $Q(u_t)$.*

Notation 2. *Let*

$$f^i_{t-1}(w) = f\left(x^i_{t-1|t-1}, u_{t-1}, w\right),$$
$$g^i_{t-1}(e) = g\left(y^i_{t-1|t-1}, \theta^i_{t-1|t-1}, u_{t-1}, e\right),$$

and, for $\xi = h, a, b$,

$$\xi^i_{t-1}(e) = \xi\left(y^i_{t-1|t-1}, u_{t-1}, e\right).$$

We assume that at time t we know the following approximation

$$p(x_{t-1}|Z_{t-1}) \simeq \sum_{i=1}^{I} \varpi^i_{t-1|t-1}\delta\left(x_{t-1} - x^i_{t-1|t-1}\right). \tag{10}$$

Our goal is to use the above to build an approximation of the form

$$p(x_t|Z_t) \simeq \sum_{i=1}^{I} \varpi^i_{t|t}\delta\left(x_t - x^i_{t|t}\right). \tag{11}$$

Our first step is to derive a convenient decomposition of $p(x_t|Z_t)$. This is done in the following lemma.

Lemma 1. *The following equality holds in the system given by (1), (2) and (4)*

$$p(x_t|Z_t) = p(\theta_t|y_t, Z_t)p(y_t|Z_t), \tag{12}$$

where

$$p(y_t|Z_t) \propto p(z_t|y_t)p(y_t|Z_{t-1}), \tag{13}$$
$$p(\theta_t|y_t, Z_t) \propto p(\theta_t|y_t, Z_{t-1}). \tag{14}$$

It follows from Lemma 1 that, in order to find an expression of $p(x_t|Z_t)$ we need expressions of $p(y_t|Z_{t-1})$ and $p(\theta_t|y_t, Z_t)$. Under the approximation (10), the desired expressions are given in the following lemma.

Lemma 2. *If (10) holds, then*

$$p(y_t|Z_{t-1}) \overset{\sim}{\propto} \sum_{i=1}^{I} \psi_t^i(y_t), \tag{15}$$

$$p(\theta_t|y_t, Z_{t-1}) \overset{\sim}{\propto} \sum_{i=1}^{I} \psi_t^i(y_t) \mathcal{N}\Big(\theta_t; \mu_t^i(y_t), \rho_t^i(y_t)\Big). \tag{16}$$

where to simplify the notation we use $g_{t-1}^{-i} = \left(g_{t-1}^i\right)^{-1}$*, and* $\overset{\sim}{\propto}$ *denotes the approximately proportional sign. Also*

$$\psi_t^i(y_t) = \varpi_{t-1|t-1}^i \gamma_t^i(y_t) \left| \frac{\det \mathbf{J} g_{t-1}^{-i}(y_t)}{b_{t-1}^i \circ g_{t-1}^{-i}(y_t)} \right|,$$

where $\mathbf{J}$ *denoted the Jacobian operator,*

$$\begin{aligned}
\rho_t^i(y_t) &= \frac{1}{\varsigma_t^i(y_t)^\top Q^{-1} \varsigma_t^i(y_t)}, \\
\mu_t^i(y_t) &= -\nu_t^i(y_t)^\top Q^{-1} \varsigma_t^i(y_t) \rho_t^i(y_t) \\
\gamma_t^i(y_t) &= \sqrt{\frac{2\pi\rho_t^i(y_t)}{\det(2\pi Q)}} \exp\left(-\frac{1}{2}\nu_t^i(y_t)^\top Q^{-1} \zeta_t^i(y_t)\right), \\
\zeta_t^i(y_t) &= \nu_t^i(y_t) + \mu_t^i(y_t) \varsigma_t^i(y_t),
\end{aligned}$$

and

$$\begin{aligned}
\nu_t^i(y_t) &= \begin{bmatrix} g_{t-1}^{-i}(y_t) \\ \frac{-h_{t-1}^i \circ g_{t-1}^{-i}(y_t) - a_{t-1}^i \circ g_{t-1}^{-i}(y_t)\theta_{t-1}}{b_{t-1}^i \circ g_{t-1}^{-i}(y_t)} \end{bmatrix}, \\
\varsigma_t^i(y_t) &= \begin{bmatrix} 0 \\ 0 \\ \frac{1}{b_{t-1}^i \circ g_{t-1}^{-i}(y_t)} \end{bmatrix}.
\end{aligned}$$

In order to build an approximation of the form (11) we need a method for drawing samples from $p(x_t|Z_t)$. In view of (12), this can be done by drawing samples from $p(\theta_t|y_t, Z_t)$ and $p(y_t|Z_t)$. It follows from (16) that $p(\theta_t|y_t, Z_t)$ is a Gaussian mixture distribution. Hence, we can readily draw samples from it. On the other hand, from (13) and (15) we obtain

$$p(y_t|Z_t) \propto p(z_t|y_t) \sum_{i=1}^{I} \psi_t^i(y_t),$$

from where it is not clear how to draw samples. Our strategy to go around this consists finding a convenient Gaussian approximation $\tilde{p}_t(y_t)$ of $p(y_t|Z_t)$, from where we can draw samples, and then weight these samples to account for the difference between $\tilde{p}_t(y_t)$ and $p(y_t|Z_t)$.

To derive the Gaussian approximation $\tilde{p}_t(y_t)$ we use (13) and proceed in three steps. In Lemma 3 we do a Gaussian approximation of $p(y_t|Z_{t-1})$, in Lemma 4 we do the same with $p(z_t|y_t)$, and in Lemma 5 we combine these two approximations to obtain $\tilde{p}_t(y_t)$.

Lemma 3. *If (10) holds, then*

$$p(y_t|Z_{t-1}) \simeq \sum_{i=1}^{I} \varpi_{t-1|t-1}^i \mathcal{N}\Big(y_t; g_{t-1}^i(0), \mathbf{J} g_{t-1}^i(0) Q_e \mathbf{J} g_{t-1}^{i\top}(0)\Big),$$

where $Q_e = SQS^\top$ *with* $S = [\mathbf{I}_2, 0]$ *and* $\mathbf{I}_2$ *denoting the* 2×2 *identity matrix.*

Lemma 4. *The following approximation holds*

$$p(z_t|y_t) \overset{\sim}{\propto} \mathcal{N}(y_t; \lambda_t, \Lambda_t), \tag{17}$$

where

$$\lambda_t = \arg\max_{y \in \mathbb{R}^2} \Xi_t(y), \tag{18}$$

$$\Lambda_t = -\left[\nabla^2 \Xi_t(\lambda_t)\right]^{-1}. \tag{19}$$

with

$$\Xi_t(y) = \log \mathcal{N}(z_t; m(y), R).$$

Combining Lemmas 3 and 4 we obtain the Gaussian approximation of $p(y_t|Z_t)$ given in the following lemma.

Lemma 5. *The following approximation holds*

$$p(y_t|Z_t) \simeq \tilde{p}_t(y_t),$$

where

$$\tilde{p}_t(y_t) = \sum_{i=1}^{I} \omega^i_{t-1|t-1} \alpha^i_t \mathcal{N}\left(y_t; \xi^i_t, \Xi^i_t\right),$$

with

$$\alpha^i_t = \mathcal{N}\left(0; g^i_{t-1}(0) - \lambda_t, \Lambda_t + \mathbf{J}g^i_{t-1}(0) Q_e \mathbf{J}g^{i\top}_{t-1}(0)\right),$$

$$\Xi^i_t = \left[\Lambda_t^{-1} + \left(\mathbf{J}g^i_{t-1}(0) Q_e \mathbf{J}g^{i\top}_{t-1}(0)\right)^{-1}\right]^{-1},$$

$$\xi^i_t = \Xi^i_t \left[\Lambda_t^{-1}\lambda_t + \left(\mathbf{J}g^i_{t-1}(0) Q_e \mathbf{J}g^{i\top}_{t-1}(0)\right)^{-1} g^i_{t-1}(0)\right].$$

Combining the results in Lemmas 1, 2 and 5 we can build the desired approximation (11) following the procedure described in Algorithm 1.

Algorithm 1 Proposed localization algorithm.

```
At $t = 1$, initialize $p(x_1|Z_1)$ using (7). At each $t > 1$:
1.  Compute $\lambda_t$ and $\Lambda_t$ using (18) and (19).
2.  For each $j = 1, \cdots, I$, do:
    (a)  Pick an index $i \in \{1, \cdots, I\}$ by drawing it from the discrete distribution such
         that $i$ has probability proportional to $\omega^i_{t-1|t-1}\alpha^i_t$.
    (b)  Draw $y^j_{t|t}$ from $\mathcal{N}(y_t; \xi^i_t, \Xi^i_t)$.
    (c)  Compute the particle weight using $\omega^j_{t|t} = \frac{p\left(y^j_{t|t}|Z_t\right)}{\tilde{p}_t\left(y^j_{t|t}\right)}$.
    (d)  Pick an index $i \in \{1, \cdots, I\}$ by drawing it from the discrete distribution such
         that $i$ has probability proportional to $\psi^i_t\left(y^j_{t|t}\right)$.
    (e)  Draw $\theta^j_{t|t}$ from $\mathcal{N}\left(\mu^i_t\left(y^j_{t|t}\right), \rho^i_t\left(y^j_{t|t}\right)\right)$.
    (f)  Put $x^j_{t|t} = \left[y^{j\top}_{t|t}, \theta^j_{t|t}\right]^\top$.
3.  Normalize the weights so that $\sum_{j=1}^{I} \omega^j_{t|t} = 1$.
4.  Estimate the position using $\hat{x}_t = \sum_{i=1}^{I} \omega^i_{t|t} x^i_{t|t}$.
```

5. Experimental Validation

In this section, we evaluate the performance of our proposed indoor localization method. To this end, we compare it with that of two methods based on particle filtering (PF). The first one is a newly proposed RSS-based method described in [20], which we refer to as PF-RSS. In order to assess the advantage resulting from using CSI measurements instead of RSS ones, the second method is the CSI-based version the PF-RSS one. It is obtained by using the PF method, described in Section 3, instead of the proposed MLPF method. We refer to it as PF-CSI. We also consider the dynamic localization method from [25], which we call MLKF, and only applies to linear Gaussian motion models like the one given in Appendix B. We also consider the localization performance obtained via static localization. To this end, we use the static localization method proposed in [25], which, as shown in that work, outperforms other available static positioning methods like FILA [28], DeepFi [29], and PhaseFi [30].

As APs we used TP-Link TL-WDR4310 routers with the OpenWrt platform installed. To acquire CSI values we use the Atheros CSI tool [31]. It provides CSI values of 56 subcarriers, and for each one, two 10 bit values are used to represent its phase and amplitude. Motion capture cameras with millimeter-level accuracy are used to obtain the ground truth position.

For the target we use a two-wheel robot controlled by a NVIDIA Jetson Nano developer board, equipped with an Atheros AR9590 network interface controller. We consider a target whose motion is described by the velocity model given in Appendix A.1, with $\sigma_v^2 = 0.36$, $\sigma_\omega^2 = 0.072$ and $\sigma_\gamma^2 = 0.072$. To generate the measurements we pass the acquired CSI phase values through the linear calibration stage proposed in [30] (§ II.B), then unwrap the resulting phases and use the resulting phase differences as fingerprints. The whole procedure for generating fingerprints is described in [25]. The measurement model is given by the following Gaussian kernel expansion

$$m(y_t) = \sum_{n=1}^{N} \kappa_n \exp\left(-\chi \|y_t - p_n\|^2\right), \tag{20}$$

where $p_n \in \mathbb{R}^2$, $n \in [1, 2, \dots, N]$, denote the positions used to build the fingerprint database. Following [32] (Chapter 4.3.2) we choose $\chi = \frac{1}{2} I^{1/3}$. Also, $\kappa = [\kappa_1, \dots, \kappa_N]^\top$ is computed so that, for each $n = 1, \dots, N$, $m(p_n)$ matches the CSI measured at p_n.

As performance metric we use the mean squared localization error ϵ, i.e., if the estimated pose at time t is $\hat{x}_t = \left[\hat{y}_t^\top, \hat{\theta}_t\right]^\top$ and the ground truth is y_t, then

$$\epsilon = \frac{1}{T} \sum_{t=1}^{T} \|\hat{y}_t - y_t\|^2.$$

For the PF method, following [23] we use $I = 1000$ particles. Also, for the MLKF method, we use the linear Gaussian motion model described in Appendix B, with $\sigma_p^2 = 0.12$ and $\sigma_\iota^2 = 0.25$.

In the first experiment, we aim to evaluate the performance in large environments. To this end we use an empty room of 7 × 16 m^2. Only one AP is placed at the corner of the room as shown in Figure 2. To build the fingerprinting database, we measured CSI's at $N = 70$ positions, one meter apart from each other, on a grid as also shown in the same figure. As the target moves within the room, we estimate its position every $\tau = 0.5$ seconds.

As we can see from Table 1, the proposed MLPF method, with only 10 particles, clearly outperforms the outcomes of static positioning and particle filtering. Its performance is comparable to that of the MLKF. However, it outperforms the latter if we increase the number of particles to 50. This is due to the use of a non-linear motion model which yields more accurate statistical knowledge of the target's position at each sample time. Also, although particle filtering should yield the theoretically optimal result if the number of particles is large enough, the very large number of required particles result in that, even with 1000 particles, its accuracy is significantly inferior than that of the proposed MLPF method with 10 particles. Figure 3 shows how the positioning accuracy of the proposed method improves with the number of particles. In Figure 4 we show the error cumulative distribution (ECD) of each method, i.e., for each error (in meters), we show the proportion of positions whose localization error is smaller than that error. The estimated trajectory and ground truth of the experiment are shown in Figure 5.

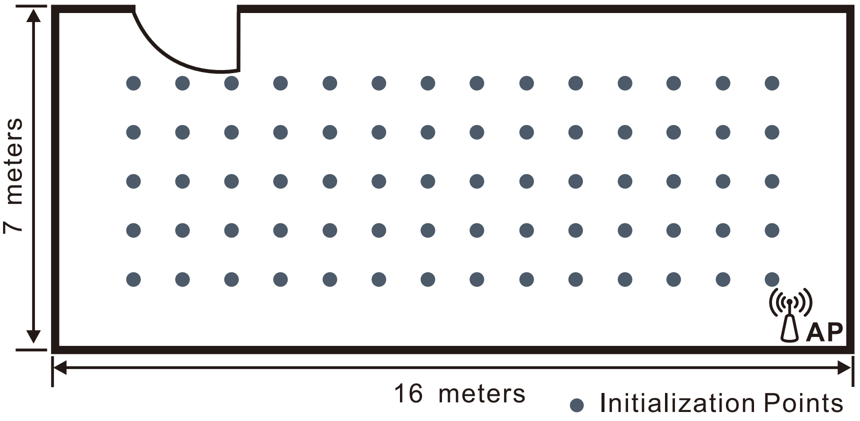

Figure 2. Layout and database points of the empty room. The access point (AP) is placed at the lower-right corner.

Table 1. Positioning Error (empty room).

Methods	Mean Squared Error [m^2]	90% Acc. [m^2]
Static positioning	1.4153	3.0437
MLKF	0.6506	1.2848
PF-RSS (1000 particles)	12.1310	28.3569
PF-CSI (1000 particles)	2.0834	5.8151
MLPF (10 particles)	0.5607	1.3724
MLPF (50 particles)	0.3370	1.0891

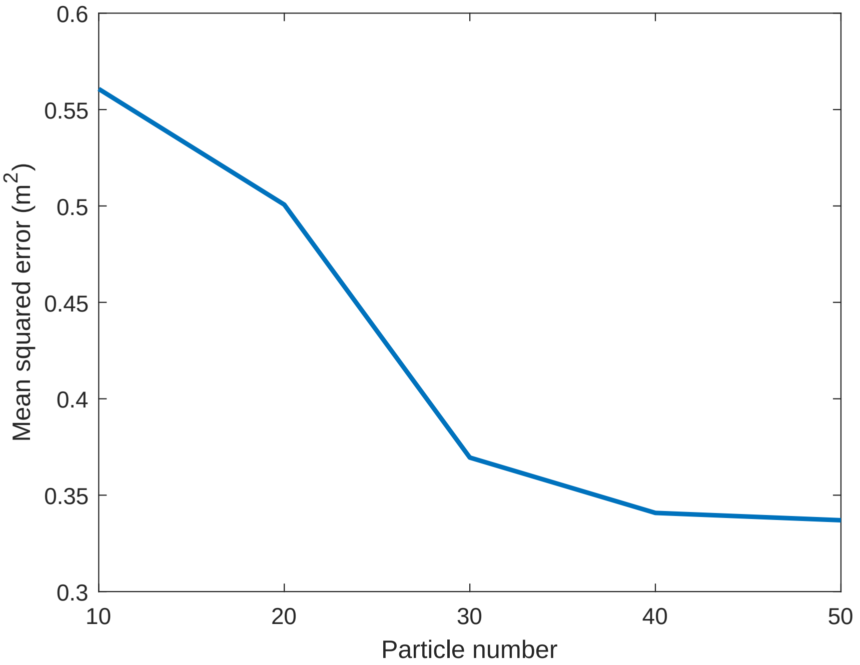

Figure 3. Impact of particle numbers on positioning accuracy.

The second experiment aims to show the performance of our method in living and working environments. To this end we use two adjacent offices of 7.1 × 11.3 m^2 connected by a corridor. We build the fingerprint database using $N = 72$ positions, as shown in Figure 6, together with the AP's location.

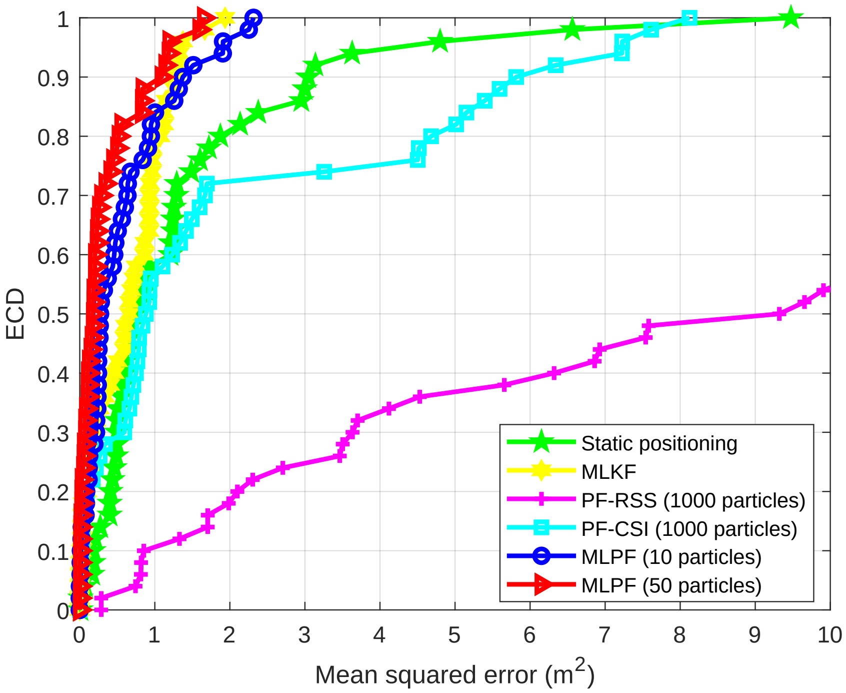

Figure 4. Positioning error cumulative distribution (ECD) (empty room).

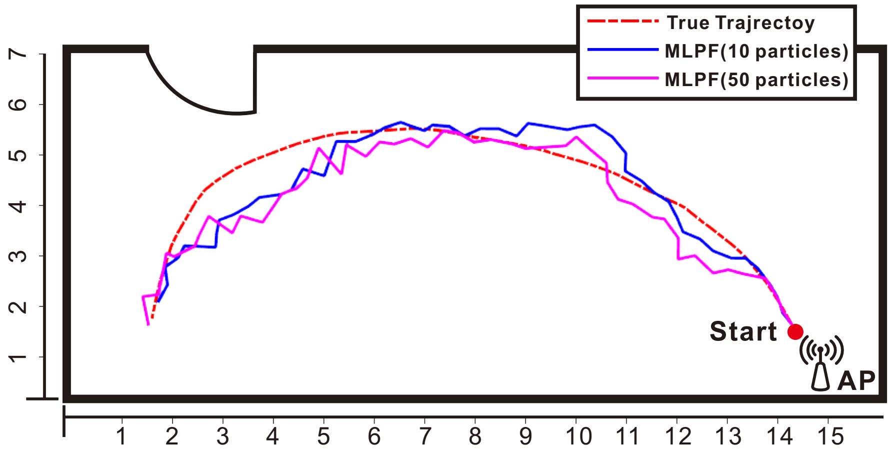

Figure 5. Tracking result of maximum likelihood particle filter (MLPF) with 10 and 50 particles in an empty room.

We see from Table 2 that, while in the empty room the proposed method with 10 particles performed similarly to the MLKF method, in this second experiment it largely outperforms the latter. The table also shows the clear advantage of the proposed method over its rivals. The reason for this advantage is as follows. Due to the signal distortion and attenuation caused by multipaths, walls and furnitures, the localization information provided by measurements is not enough to yield accurate estimates at certain locations. This can be seen in the ECD shown in Figure 7. At those locations, dynamic localization methods resort to motion model information to produce an estimate. Figure 7 shows that only the proposed MLPF method can produce accurate estimates at those locations.

This is because it is the only method that is able to make full use of the rich information provided by a non-linear motion model. While methods based on PF potentially have the same possibility, the extremely large number of required particles make this unfeasible in practice. To conclude the experiment, we show in Figure 8 the ground truth and estimated trajectories.

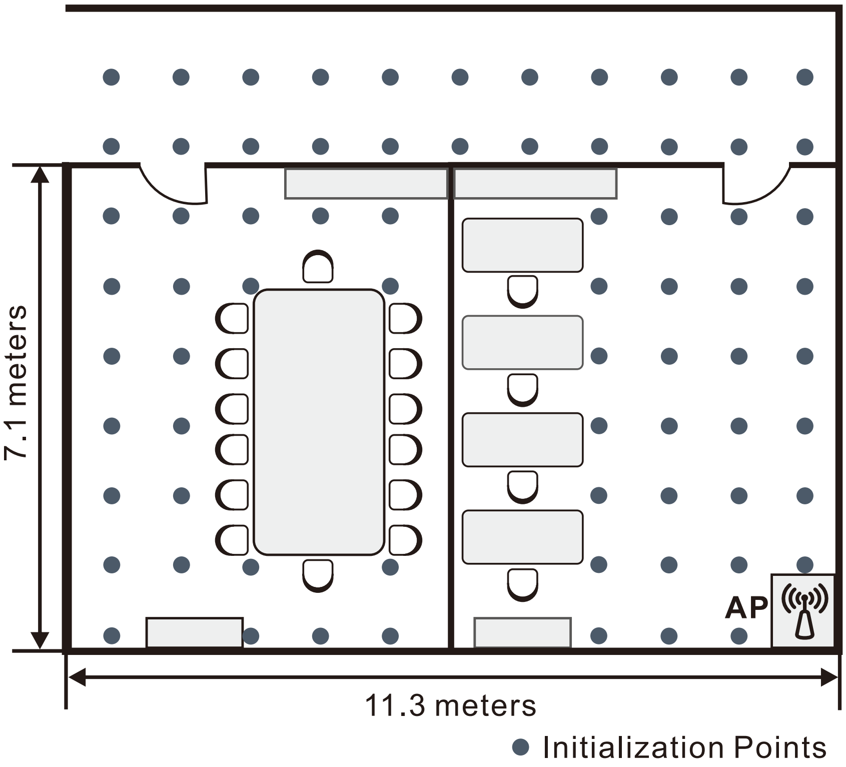

Figure 6. Layout and database points for the two office rooms. The AP is placed at the lower-right corner of the right room.

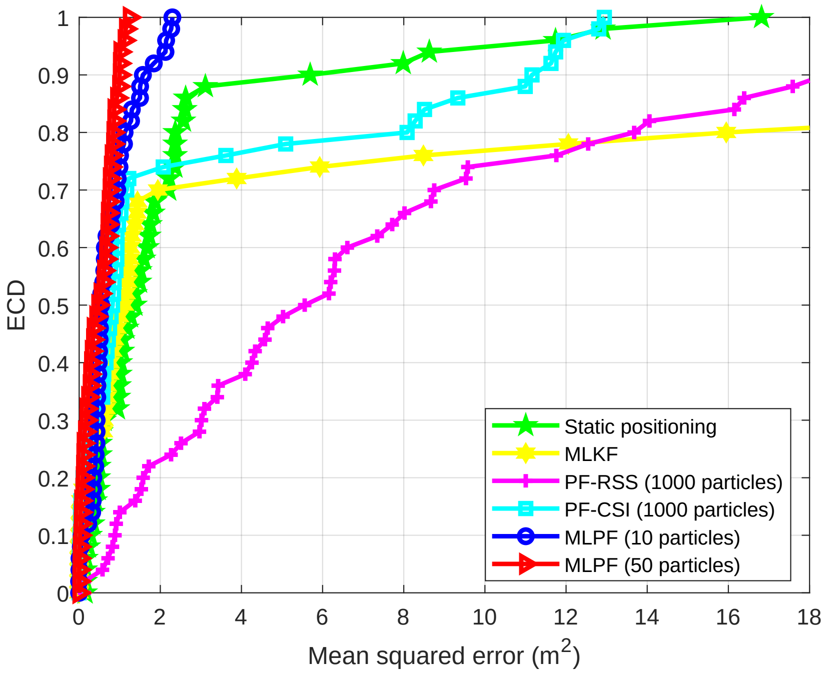

Figure 7. Positioning ECD (office rooms).

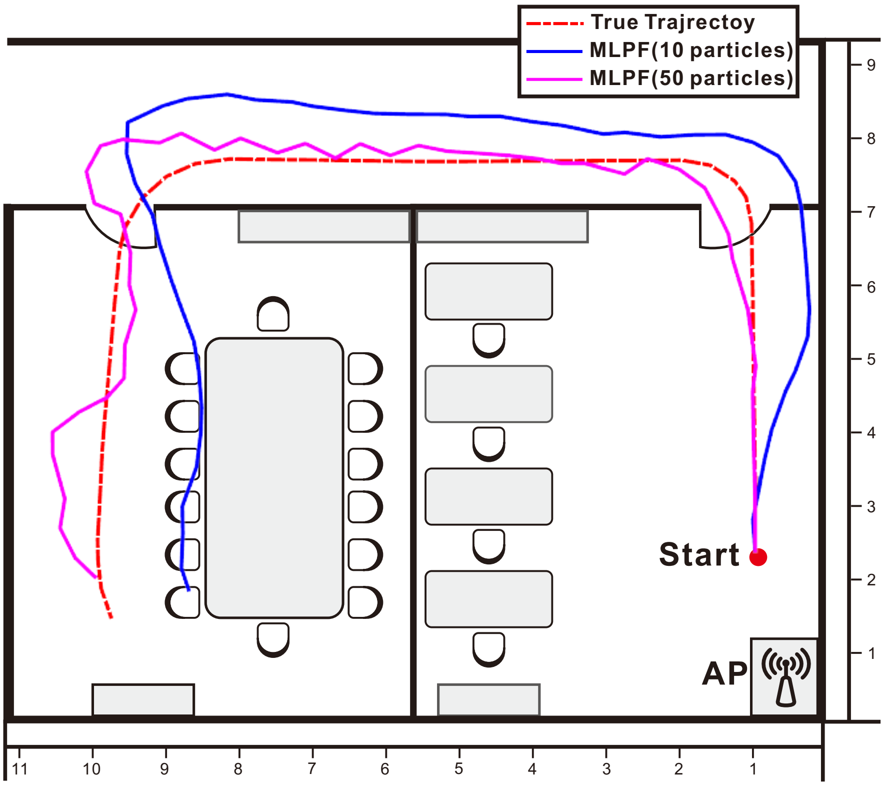

Figure 8. Tracking result of MLPF with 10 and 50 particles in the office rooms.

Table 2. Positioning Error (office rooms).

Methods	Mean Squared Error [m^2]	90% Acc. [m^2]
Static positioning	2.3962	5.6935
MLKF	11.7255	50.8279
PF-RSS (1000 particles)	8.2366	18.3872
PF-CSI (1000 particles)	2.9418	11.1606
MLPF (10 particles)	0.7755	1.5712
MLPF (50 particles)	0.4939	0.9970

6. Conclusions

We proposed a new indoor positioning method, which we called maximum likelihood particle filtering. Its essential idea consists of combing the prediction and updates steps of a traditional particle filter into a single step, which requires solving a maximum likelihood estimation problem. This results in a better utilization of particles. The method so obtained achieves high accuracy indoor positioning with a drastically smaller number of particles in comparison with particle filtering. This makes it numerically feasible for real-time applications. We validated our claims in a WiFi indoor positioning experiment using real data. Our experiment show that our method leads to great accuracy improvements achieving sub-meter level indoor localization accuracy, which exceeds the requirements of the 5G NR Release 16 standard from 3GPP [33].

Author Contributions: Conceptualization, D.M. and M.F.; methodology, D.M.; software, W.W.; validation, W.W.; formal analysis, D.M.; investigation, W.W. and D.M.; resources, M.F.; data curation, W.W.; writing—original draft preparation, W.W.; writing—review and editing, D.M. and M.F.; visualization, W.W. and M.F.; supervision, D.M. and M.F.; project administration, M.F.; funding acquisition, M.F. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China grant numbers 61633014, 61803101 and U1701264, Guangdong Basic and Applied Basic Research Foundation grant number 2020A1515011505, and the Argentinean Agency for Scientific and Technological Promotion grant number PICT- 201-0985.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Non-Linear Robot Motion Models

Appendix A.1. Velocity Model

The input is $u_t = [v_t, \omega_t]$, with v_t and ω_t being the robot's measured linear and angular speeds, respectively. Also $e_t = [\tilde{v}_t, \tilde{\omega}_t]$, $n_t = \tilde{\gamma}_t$, $Q(u_t) = \mathrm{diag}\left(\sigma_v^2, \sigma_\omega^2, \sigma_\gamma^2\right)$ with

$$\begin{aligned}
\sigma_v^2 &= \alpha_1 |v_t| + \alpha_2 |\omega_t|, \\
\sigma_\omega^2 &= \alpha_3 |v_t| + \alpha_4 |\omega_t|, \\
\sigma_\gamma^2 &= \alpha_5 |v_t| + \alpha_6 |\omega_t|,
\end{aligned}$$

and $\alpha_1, \ldots, \alpha_6$ being system parameters. The motion model is given by

$$\begin{aligned}
a_{t+1} &= a_t - \frac{\hat{v}_t}{\hat{\omega}_t} \sin\theta_t + \frac{\hat{v}_t}{\hat{\omega}_t} \sin(\theta_t + \hat{\omega}_t \tau), \\
b_{t+1} &= b_t + \frac{\hat{v}_t}{\hat{\omega}_t} \cos\theta_t - \frac{\hat{v}_t}{\hat{\omega}_t} \cos(\theta_t + \hat{\omega}_t \tau), \\
\theta_{t+1} &= \theta_t + (\hat{\omega}_t + \tilde{\gamma}_t)\tau,
\end{aligned}$$

where τ denotes the sampling time and

$$\begin{aligned}
\hat{v}_t &= v_t + \tilde{v}_t, \\
\hat{\omega}_t &= \omega_t + \tilde{\omega}_t.
\end{aligned}$$

Appendix A.2. Odometry Model

In this case $u_t = \left[\bar{x}_{t+1}^\top, \bar{x}_t^\top\right]^\top$, where $\bar{x}_t = \left[\bar{a}_t, \bar{b}_t, \bar{\theta}_t\right]^\top$ denotes the odometry measurement at time t. Using u_t we compute

$$\begin{aligned}
\delta_{\text{rot1}} &= \arctan \frac{\bar{b}_{t+1} - \bar{b}_t}{\bar{a}_{t+1} - \bar{a}_t} - \bar{\theta}_t, \\
\delta_{\text{trans}} &= \sqrt{(\bar{a}_{t+1} - \bar{a}_t)^2 + \left(\bar{b}_{t+1} - \bar{b}_t\right)^2}, \\
\delta_{\text{rot1}} &= \bar{\theta}_{t+1} - \bar{\theta}_t - \delta_{\text{rot1}}.
\end{aligned}$$

Then, $w_t = \left[\tilde{\delta}_{\text{rot1}}, \tilde{\delta}_{\text{trans}}, \tilde{\delta}_{\text{rot2}}\right]$ and $Q(u_t) = \mathrm{diag}\left(\sigma_{\text{rot1}}^2, \sigma_{\text{trans}}^2, \sigma_{\text{rot2}}^2\right)$ with

$$\begin{aligned}
\sigma_{\text{rot1}}^2 &= \alpha_1 \left|\hat{\delta}_{\text{rot1}}\right| + \alpha_2 \left|\hat{\delta}_{\text{trans}}\right|, \\
\sigma_{\text{trans}}^2 &= \alpha_3 \left|\hat{\delta}_{\text{trans}}\right| + \alpha_4 \left|\hat{\delta}_{\text{rot1}} + \hat{\delta}_{\text{rot2}}\right|, \\
\sigma_{\text{rot2}}^2 &= \alpha_5 \left|\hat{\delta}_{\text{rot2}}\right| + \alpha_6 \left|\hat{\delta}_{\text{trans}}\right|,
\end{aligned}$$

and $\alpha_1, \ldots, \alpha_6$ being system parameters. The motion model is given by

$$\begin{aligned}
a_{t+1} &= a_t + \hat{\delta}_{\text{trans}} \cos\big(\theta_t + \hat{\delta}_{\text{rot1}}\big), \\
b_{t+1} &= b_t + \hat{\delta}_{\text{trans}} \sin\big(\theta_t + \hat{\delta}_{\text{rot1}}\big), \\
\theta_{t+1} &= \theta_t + \hat{\delta}_{\text{rot1}} + \hat{\delta}_{\text{rot2}},
\end{aligned}$$

where

$$\begin{aligned}
\hat{\delta}_{\text{rot1}} &= \delta_{\text{rot1}} + \tilde{\delta}_{\text{rot1}}, \\
\hat{\delta}_{\text{trans}} &= \delta_{\text{trans}} + \tilde{\delta}_{\text{trans}}, \\
\delta_{\text{rot2}} &= \delta_{\text{rot2}} + \tilde{\delta}_{\text{rot2}}.
\end{aligned}$$

Appendix A.3. Differential Drive Model

As in the velocity model, $u_t = [v_t, \omega_t]$. Also $w_t = [\tilde{v}_t, \tilde{\eta}_t, \tilde{\omega}_t]$, where $Q(u_t) = \operatorname{diag}\Big(\sigma_v^2, \sigma_\eta^2, \sigma_\omega^2\Big)$ with σ_v^2, σ_η^2, and σ_ω^2 being system parameters. The motion model is given by

$$\begin{aligned}
a_{t+1} &= a_t + \tau \cos(\theta_t)(v_t + \tilde{v}_t) - \tau \sin(\theta_t)\tilde{\eta}_t, \\
b_{t+1} &= b_t + \tau \sin(\theta_t)(v_t + \tilde{v}_t) + \tau \cos(\theta_t)\tilde{\eta}_t, \\
\theta_{t+1} &= \theta_t + \tau\tilde{\omega}_t,
\end{aligned}$$

where τ denotes the sampling time.

Appendix B. Linear Motion Model

In the linear motion model, p_ζ and v_ζ, $\zeta \in \{x, y\}$, denote the position and velocity, respectively, in the ζ axis. The model is given by

$$\begin{aligned}
p_x(t+1) &= p_x(t) + \tau v_x(t) + w_x(t), \\
p_y(t+1) &= p_y(t) + \tau v_y(t) + w_y(t), \\
v_x(t+1) &= v_x(t) + \iota_x(t), \\
v_y(t+1) &= v_y(t) + \iota_y(t),
\end{aligned}$$

where w_x, w_y, ι_x and ι_y are mutually independent, $w_x \sim w_y \sim \mathcal{N}\Big(0, \sigma_p^2\Big)$, and $\iota_x \sim \iota_y \sim \mathcal{N}\big(0, \sigma_\iota^2\big)$.

Appendix C. Proofs

Proof of Lemma 1. We have

$$p(x_t | Z_t) = p(\theta_t | y_t, Z_t) p(y_t | Z_t).$$

Now

$$\begin{aligned}
p(y_t | Z_t) &= \frac{1}{p(z_t | Z_{t-1})} p(y_t, z_t | Z_{t-1}) \\
&\propto p(z_t | y_t) p(y_t | Z_{t-1}),
\end{aligned}$$

and

$$\begin{aligned} p(\theta_t|y_t, Z_t) &\propto p(\theta_t, y_t|Z_t) \\ &\propto p(\theta_t, y_t, z_t|Z_{t-1}) \\ &\propto p(z_t|\theta_t, y_t)p(\theta_t, y_t|Z_{t-1}) \\ &\propto p(z_t|y_t)p(\theta_t, y_t|Z_{t-1}) \\ &\propto p(\theta_t, y_t|Z_{t-1}) \\ &\propto p(\theta_t|y_t, Z_{t-1}). \end{aligned}$$

□

Proof of Lemma 2. We split the proof in steps:

1. It follows straightforwardly from [34] (§ 5.4) that

$$p\left(x_t|x^i_{t-1|t-1}\right) = \sum_{w:x_t=f_{t-1}(w)} \left|\det \mathbf{J} f^i_{t-1}(w)\right|^{-1} p(w_{t-1} = w).$$

Since the variance of n_{t-1} is small, we can do the following approximation,

$$\begin{aligned} p\left(x_t|x^i_{t-1|t-1}\right) &\simeq \left|\det \mathbf{J} f^i_{t-1}\left(f^{-i}_{t-1}(x_t)\right)\right|^{-1} \mathcal{N}\left(f^{-i}_{t-1}(x_t); 0, Q\right) \\ &\simeq \left|\det \mathbf{J} f^{-i}_{t-1}(x_t)\right| \mathcal{N}\left(f^{-i}_{t-1}(x_t); 0, Q\right) \end{aligned} \tag{A1}$$

where to simplify the notation we used $f^{-i}_{t-1} = \left(f^i_{t-1}\right)^{-1}$.

2. We have

$$f^{-i}_{t-1}(x_t) = \left[\begin{array}{c} g^{-i}_{t-1}(y_t) \\ \frac{\theta_t - h^i_{t-1} \circ g^{-i}_{t-1}(y_t) - a^i_{t-1} \circ g^{-i}_{t-1}(y_t)\theta_{t-1}}{b^i_{t-1} \circ g^{-i}_{t-1}(y_t)} \end{array}\right] \tag{A2}$$

$$= \nu^i_t(y_t) + \varsigma^i_t(y_t)\theta_t. \tag{A3}$$

Then, from (A2),

$$\mathbf{J} f^{-i}_{t-1}(x_t) = \left[\begin{array}{cc} \mathbf{J} g^{-i}_{t-1}(y_t) & 0 \\ * & \frac{1}{b^i_{t-1} \circ g^{-i}_{t-1}(y_t)} \end{array}\right],$$

where $*$ denotes a non-zero term whose value is irrelevant. Then

$$\left|\det \mathbf{J} f^{-i}_{t-1}(x_t)\right| = \left|\frac{\det \mathbf{J} g^{-i}_{t-1}(y_t)}{b^i_{t-1} \circ g^{-i}_{t-1}(y_t)}\right|. \tag{A4}$$

3. Now, using (10) and (A1), we get

$$\begin{aligned} p(y_t, \theta_t|Z_{t-1}) =& p(x_t|Z_{t-1}) \\ =& \int p(x_t|x_{t-1})p(x_{t-1}|Z_{t-1}) dy_{t-1}\theta_{t-1} \\ \simeq& \sum_{i=1}^{I} \varpi^i_{t-1|t-1} p\left(x_t|x^i_{t-1|t-1}\right) \\ \simeq& \sum_{i=1}^{I} \varpi^i_{t-1|t-1} \left|\det \mathbf{J} f^{-i}_{t-1}(x_t)\right| \mathcal{N}\left(f^{-i}_{t-1}(x_t); 0, Q\right). \end{aligned} \tag{A5}$$

But, from (A3),

$$\mathcal{N}\left(f^{-i}_{t-1}(x_t); 0, Q\right) = \mathcal{N}\left(\nu^i_t(y_t) + \varsigma^i_t(y_t)\theta_t; 0, Q\right).$$

Now

$$\begin{aligned}
&\left(\nu_t^i(y_t)+\varsigma_t^i(y_t)\theta_t\right)^\top Q^{-1}\left(\nu_t^i(y_t)+\varsigma_t^i(y_t)\theta_t\right)\\
=&\varsigma_t^i(y_t)^\top Q^{-1}\varsigma_t^i(y_t)\theta_t^2+2\nu_t^i(y_t)^\top Q^{-1}\varsigma_t^i(y_t)\theta_t+\nu_t^i(y_t)^\top Q^{-1}\nu_t^i(y_t)\\
=&\varsigma_t^i(y_t)^\top Q^{-1}\varsigma_t^i(y_t)\theta_t^2-2\varsigma_t^i(y_t)^\top Q^{-1}\varsigma_t^i(y_t)\mu_t^i(y_t)\theta_t\\
&+\varsigma_t^i(y_t)^\top Q^{-1}\varsigma_t^i(y_t)\mu_t^i(y_t)^2+\nu_t^i(y_t)^\top Q^{-1}\nu_t^i(y_t)-\varsigma_t^i(y_t)^\top Q^{-1}\varsigma_t^i(y_t)\mu_t^i(y_t)^2\\
=&\frac{\left(\theta_t-\mu_t^i(y_t)\right)^2}{\rho_t^i(y_t)}+\nu_t^i(y_t)^\top Q^{-1}\nu_t^i(y_t)+\nu_t^i(y_t)^\top Q^{-1}\varsigma_t^i(y_t)\mu_t^i(y_t)\\
=&\frac{\left(\theta_t-\mu_t^i(y_t)\right)^2}{\rho_t^i(y_t)}+\nu_t^i(y_t)^\top Q^{-1}\left(\nu_t^i(y_t)+\varsigma_t^i(y_t)\mu_t^i(y_t)\right)
\end{aligned}$$

Hence

$$\begin{aligned}
\mathcal{N}\left(f_{t-1}^{-i}(x_t);0,Q\right)=&\frac{1}{\sqrt{\det(2\pi Q)}}\exp\left(-\frac{\left(\theta_t-\mu_t^i(y_t)\right)^2}{2\rho_t^i(y_t)}\right)\times\exp\left(-\frac{1}{2}\nu_t^i(y_t)^\top Q^{-1}\zeta_t^i(y_t)\right)\\
=&\gamma_t^i(y_t)\mathcal{N}\left(\theta_t;\mu_t^i(y_t),\rho_t^i(y_t)\right),
\end{aligned} \tag{A6}$$

4. Putting (A4) and (A6) into (A5) we obtain

$$p(y_t,\theta_t|Z_{t-1})\ \tilde{\propto}\ \sum_{i=1}^{I}\psi_t^i(y_t)\mathcal{N}\left(\theta_t;\mu_t^i(y_t),\rho_t^i\left(y_t^i\right)\right),$$

and the result follows.

□

Proof of Lemma 3. If $x_{t-1}=x_{t-1|t-1}^i$, doing a first order Taylor expansion we obtain

$$\begin{aligned}
y_t&=g_{t-1}^i(e_t)\\
&\simeq g_{t-1}^i(0)+\mathbf{J}g_{t-1}^i(0)e_t.
\end{aligned}$$

So

$$p\left(y_t|x_{t-1|t-1}^i\right)\simeq\mathcal{N}\left(y_t;g_{t-1}^i(0),\mathbf{J}g_{t-1}^i(0)Q_e\mathbf{J}g_{t-1}^{i\top}(0)\right).$$

The result then follows since

$$\begin{aligned}
p(y_t|Z_{t-1})&=\sum_{i=1}^{I}\varpi_{t-1|t-1}^i p\left(y_t|x_{t-1|t-1}^i\right)\\
&=\sum_{i=1}^{I}\varpi_{t-1|t-1}^i\mathcal{N}\left(y_t;g_{t-1}^i(0),\mathbf{J}g_{t-1}^i(0)Q_e\mathbf{J}g_{t-1}^{i\top}(0)\right).
\end{aligned}$$

□

Proof of Lemma 4. Let

$$L_t(y)=p(z_t|y_t=y).$$

Recall that N is the dimension of z_t. It follows from ([24], Theorem 6) that, for all $\epsilon\in\mathbb{R}^2$, we can do the following approximation

$$\lim_{N\to\infty}\frac{1}{L_t(\lambda_t)}L_t\left((\Lambda_t)^{-1/2}\epsilon+\lambda_t\right)\overset{\text{w.p.1}}{=}\exp\left(-\frac{\epsilon^T\epsilon}{2}\right), \tag{A7}$$

with

$$\begin{aligned}\Xi_t(y) &= \log L_t(y) \\ &= \log \mathcal{N}(z_t; m(y), R).\end{aligned}$$

In view of the above, for large N we have

$$L_t(y) \simeq L_t\left(\lambda_t^i\right) \exp\left(-\frac{1}{2}(y-\lambda_t)^T \Lambda_t^{-1}(y-\lambda_t)\right), \tag{A8}$$

Hence

$$\begin{aligned}p(z_t|y_t) &= L_t(y_t) \\ &\mathrel{\tilde{\propto}} \mathcal{N}(y_t; \lambda_t, \Lambda_t).\end{aligned}$$

□

Proof of Lemma 5. From the above we get

$$\begin{aligned}p(y_t|Z_t) &\propto p(z_t|y_t)p(y_t|Z_{t-1}) \\ &\simeq \sum_{i=1}^{I} \varpi_{t-1|t-1}^i \mathcal{N}(y_t; \lambda_t, \Lambda_t) \times \mathcal{N}\left(y_t; g_{t-1}^i(0), \mathbf{J}g_{t-1}^i(0) Q_e \mathbf{J}g_{t-1}^{i\top}(0)\right).\end{aligned}$$

The result then follows since, from [35] (§ 8.1.8),

$$\mathcal{N}(y_t; \lambda_t, \Lambda_t)\mathcal{N}\left(y_t; g_{t-1}^i(0), \mathbf{J}g_{t-1}^i(0) Q_e \mathbf{J}g_{t-1}^{i\top}(0)\right) = \alpha_t^i \mathcal{N}\left(y_t; \zeta_t^i, \Xi_t^i\right).$$

□

References

1. Saab, S.S.; Nakad, Z.S. A standalone RFID indoor positioning system using passive tags. *IEEE Trans. Ind. Electron.* **2010**, *58*, 1961–1970. [CrossRef]
2. Seco, F.; Jiménez, A.R. Smartphone-Based Cooperative Indoor Localization with RFID Technology. *Sensors* **2018**, *18*, 266. [CrossRef] [PubMed]
3. Hallberg, J.; Nilsson, M.; Synnes, K. Positioning with bluetooth. In Proceedings of the 10th International Conference on Telecommunications 2003 (ICT 2003), Papeete, Tahiti, French Polynesia, 23 Feburary–1 March 2003; Volume 2, pp. 954–958.
4. Yuan, Z.; Yang, J.; You, L.; Qi, L.; Naser, E.S. Smartphone-Based Indoor Localization with Bluetooth Low Energy Beacons. *Sensors* **2016**, *16*, 596.
5. Gigl, T.; Janssen, G.J.; Dizdarevic, V.; Witrisal, K.; Irahhauten, Z. Analysis of a UWB indoor positioning system based on received signal strength. In Proceedings of the 2007 4th Workshop on Positioning, Navigation and Communication, Hannover, Germany, 22 March 2007; pp. 97–101.
6. Mazhar, F.; Khan, M.G.; Sällberg, B. Precise indoor positioning using UWB: A review of methods, algorithms and implementations. *Wirel. Pers. Commun.* **2017**, *97*, 4467–4491. [CrossRef]
7. Yang, C.; Shao, H.R. WiFi-based indoor positioning. *IEEE Commun. Mag.* **2015**, *53*, 150–157. [CrossRef]
8. Jian, C.; Gang, O.; Ao, P.; Zheng, L.; Shi, J. An INS/WiFi Indoor Localization System Based on the Weighted Least Squares. *Sensors* **2018**, *18*, 1458.
9. Yang, Z.; Zhou, Z.; Liu, Y. From RSSI to CSI: Indoor localization via channel response. *ACM Comput. Surv. (CSUR)* **2013**, *46*, 1–32. [CrossRef]
10. Zafari, F.; Gkelias, A.; Leung, K.K. A survey of indoor localization systems and technologies. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 2568–2599. [CrossRef]
11. Schatzberg, U.; Banin, L.; Amizur, Y. Enhanced WiFi ToF indoor positioning system with MEMS-based INS and pedometric information. In Proceedings of the 2014 IEEE/ION Position, Location and Navigation Symposium-PLANS 2014, Monterey, CA, USA, 5–8 May 2014; pp. 185–192.
12. Kotaru, M.; Joshi, K.; Bharadia, D.; Katti, S. Spotfi: Decimeter level localization using wifi. In Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication, London, UK, 17–21 August 2015; pp. 269–282.
13. Xia, S.; Liu, Y.; Yuan, G.; Zhu, M.; Wang, Z. Indoor fingerprint positioning based on Wi-Fi: An overview. *ISPRS Int. J. Geo-Inf.* **2017**, *6*, 135. [CrossRef]

14. Wen, Y.; Tian, X.; Wang, X.; Lu, S. Fundamental limits of RSS fingerprinting based indoor localization. In Proceedings of the 2015 IEEE Conference on Computer Communications (INFOCOM), Kowloon, Hong Kong, 26 April–1 May 2015; pp. 2479–2487.
15. Dardari, D.; Closas, P.; Djurić, P.M. Indoor tracking: Theory, methods, and technologies. *IEEE Trans. Veh. Technol.* **2015**, *64*, 1263–1278. [CrossRef]
16. Belmonte-Hernández, A.; Hernández-Peñaloza, G.; Alvarez, F.; Conti, G. Adaptive fingerprinting in multi-sensor fusion for accurate indoor tracking. *IEEE Sens. J.* **2017**, *17*, 4983–4998. [CrossRef]
17. Du, Y.; Yang, D.; Yang, H.; Xiu, C. Flexible indoor localization and tracking system based on mobile phone. *J. Netw. Comput. Appl.* **2016**, *69*, 107–116.
18. Xu, W.; Liu, L.; Zlatanova, S.; Penard, W.; Xiong, Q. A pedestrian tracking algorithm using grid-based indoor model. *Autom. Constr.* **2018**, *92*, 173–187. [CrossRef]
19. Deng, Z.A.; Hu, Y.; Yu, J.; Na, Z. Extended Kalman filter for real time indoor localization by fusing WiFi and smartphone inertial sensors. *Micromachines* **2015**, *6*, 523–543. [CrossRef]
20. Belmonte-Hernández, A.; Hernández-Peñaloza, G.; Gutiérrez, D.M.; Álvarez, F. SWiBluX: Multi-Sensor Deep Learning Fingerprint for precise real-time indoor tracking. *IEEE Sens. J.* **2019**, *19*, 3473–3486. [CrossRef]
21. Zhao, Z.; Braun, T.; Li, Z.; Neto, A. A real-time robust indoor tracking system in smartphones. *Comput. Commun.* **2018**, *117*, 104–115.
22. Jia, R.; Jin, M.; Zou, H.; Yesilata, Y.; Xie, L.; Spanos, C. Mapsentinel: Can the knowledge of space use improve indoor tracking further? *Sensors* **2016**, *16*, 472. [CrossRef]
23. Wu, C.; Zhang, F.; Wang, B.; Liu, K.R. EasiTrack: Decimeter-Level Indoor Tracking With Graph-Based Particle Filtering. *IEEE Internet Things J.* **2019**, *7*, 2397–2411. [CrossRef]
24. Marelli, D.; Fu, M.; Ninness, B. Asymptotic Optimality of the Maximum-Likelihood Kalman Filter for Bayesian Tracking With Multiple Nonlinear Sensors. *IEEE Trans. Signal Process.* **2015**, *63*, 4502–4515. [CrossRef]
25. Wang, W.; Marelli, D.; Fu, M. Fingerprinting-Based Indoor Localization Using Interpolated Preprocessed CSI Phases and Bayesian Tracking. *Sensors* **2020**, *20*, 2854. [CrossRef]
26. Thrun, S. Probabilistic robotics. *Commun. ACM* **2002**, *45*, 52–57. [CrossRef]
27. Arulampalam, M.S.; Maskell, S.; Gordon, N.; Clapp, T. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Trans. Signal Process.* **2002**, *50*, 174–188. [CrossRef]
28. Wu, K.; Xiao, J.; Yi, Y.; Chen, D.; Luo, X.; Ni, L.M. CSI-based indoor localization. *IEEE Trans. Parallel Distrib. Syst.* **2012**, *24*, 1300–1309. [CrossRef]
29. Wang, X.; Gao, L.; Mao, S.; Pandey, S. DeepFi: Deep learning for indoor fingerprinting using channel state information. In Proceedings of the 2015 IEEE Wireless Communications and Networking Conference (WCNC), New Orleans, LA, USA, 9–12 March 2015; pp. 1666–1671.
30. Wang, X.; Gao, L.; Mao, S. CSI phase fingerprinting for indoor localization with a deep learning approach. *IEEE Internet Things J.* **2016**, *3*, 1113–1123. [CrossRef]
31. Xie, Y.; Li, Z.; Li, M. Precise power delay profiling with commodity Wi-Fi. *IEEE Trans. Mob. Comput.* **2018**, *18*, 1342–1355. [CrossRef]
32. Silverman, B.W. *Density Estimation for Statistics and Data Analysis*; CRC Press: Boca Raton, FL, USA, 1986; Volume 26.
33. 3GPP. TR38.855. In *Study on NR Positioning Support (Release 16)*; 3rd Generation Partnership Project (3GPP): Sophia Antipolis, France, 2019.
34. Devore, J.L.; Berk, K.N. *Modern Mathematical Statistics with Applications*; Springer: New York, NY, USA, 2012.
35. Petersen, K.; Pedersen, M. *The Matrix Cookbook*; Technical Manual; Technical University of Denmark: Lyngby, Denmark, 2008 .