

# TrackInFactory: A Tight Coupling Particle Filter for Industrial Vehicle Tracking in Indoor Environments

Ivo Silva<sup>1</sup>, Cristiano Pendão<sup>1</sup>, Joaquín Torres-Sospedra<sup>2</sup>, and Adriano Moreira<sup>1</sup>, *Member, IEEE*

**Abstract**—Localization and tracking of industrial vehicles have a key role in increasing productivity and improving the logistics processes of factories. Due to the demanding requirements of industrial vehicle tracking and navigation, existing systems explore technologies, such as LiDAR or ultra wide-band to achieve low positioning errors. In this article we propose TrackInFactory, a system that combines Wi-Fi with motion sensors, achieving submeter accuracy and a low maximum error. A tight coupling approach is explored in sensor fusion with a particle filter (PF). Information regarding the vehicle's initial position and heading is not required. This approach uses the similarity of Wi-Fi samples to update the particles' weights as they move according to motion sensor data. The PF dynamically adjusts its parameters based on a metric for estimating the confidence in position estimates, allowing to improve positioning performance. A series of simulations were performed to tune the PF. Then the approach was validated in real-world experiments with an industrial tow tractor, achieving a mean error of 0.81 m. In comparison to a loose coupling approach, this method reduced the maximum error by more than 60% and improved the overall mean error by more than 20%.

**Index Terms**—Bayesian filtering, dead reckoning (DR), indoor positioning, indoor tracking, industrial vehicle, industry 4.0, particle filter (PF), sensor fusion, tight coupling (TC), Wi-Fi-based positioning.

## I. INTRODUCTION

VEHICLES have an important role in the distribution of materials in industrial environments, thus current and future generation factories (Industry 4.0) can benefit from tracking these vehicles to increase productivity, improve logistics processes and enable device intercommunication. Besides, materials can be tracked along the supply chain by continuously monitoring the position of the vehicles carrying them,

Manuscript received October 21, 2020; revised February 24, 2021 and April 30, 2021; accepted June 14, 2021. Date of publication July 6, 2021; date of current version June 16, 2022. This work was supported in part by the FCT—Fundação para a Ciência e Tecnologia within the Research and Development Units Project Scope under Grant UIDB/00319/2020; in part by the Ph.D. Fellowship under Grant PD/BD/137401/2018; and in part by the Ministerio de Ciencia, Innovación y Universidades under Grant PTQ2018-009981. This article was recommended by Associate Editor R. Cui. (*Corresponding author: Ivo Silva.*)

Ivo Silva, Cristiano Pendão, and Adriano Moreira are with the Algoritmi Research Centre, University of Minho, 4800-058 Guimarães, Portugal (e-mail: ivo@dsi.uminho.pt; cpendao@dsi.uminho.pt; adriano.moreira@algoritmi.uminho.pt).

Joaquín Torres-Sospedra is with the UBIK Geospatial Solutions S.L., 12006 Castellón, Spain (e-mail: torres@ubikgs.com).

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/TSMC.2021.3091987>.

Digital Object Identifier 10.1109/TSMC.2021.3091987

improving internal and external logistics. With the rapid development of research in this area [1], [2], there are already autonomous vehicles as well as hybrid vehicles (can be human-operated and also operate autonomously) in factories, which can benefit from positioning and tracking.

Positioning of industrial vehicles can have two distinct applications, namely, continuous tracking and monitoring of the vehicle's position as it operates (main focus of this work), and as support for navigation of autonomous vehicles [3], [4]. Both applications have similar requirements, including coverage of the entire operating area (factory plant), submeter accuracy (or better for navigation), low maximum error, availability and high reliability. The reliability of a positioning system is especially important for the localization of vehicles, as it defines if the provided estimates can be trusted. A positioning system with low reliability can provide very accurate estimates as well as estimates with large error, without providing any feedback as to whether those estimates are accurate or not. Conversely, a reliable positioning system is able to determine if a position estimate has a high/low error associated, i.e., if the position estimate can be trusted.

For many years, the localization of vehicles has been supported by GPS in outdoor environments, however, GPS does not properly operate indoors due to the attenuation and distortion caused by the building structure. Among the available solutions for indoor positioning, Wi-Fi fingerprinting is widely used. Wi-Fi fingerprinting works on a simple principle where the radio environment at each particular position is defined by a unique fingerprint, i.e., a set of signal strength measurements [5]. A previously collected radio map (RM), composed of Wi-Fi samples (WSs), is used to obtain the similarity between each RM sample and an online WS, then an estimator algorithm [e.g.,  $k$ -nearest-neighbor (kNN) or a weighted kNN variant] is used to find a position estimate. Although Wi-Fi networks are already present in most buildings, even factories, enabling Wi-Fi fingerprinting, it usually provides a mean positioning error between 2 and 4 m with a presence of punctual large errors higher than 10 m [6].

Even though Wi-Fi may provide an absolute position, it is not effective for tracking vehicles, due to significant variation between position estimates and due to low accuracy. Alternatively, vehicles can be tracked using dead reckoning (DR) by exploring sensors, such as inertial measurement units (IMUs) and wheel encoders. DR depends on knowledge about the initial pose (position and heading), which must be estimated using other methods, and it suffers from cumulative errors that affect IMU sensors, causing drift in the

estimated trajectory. Sensor fusion of Wi-Fi with motion sensors allows to benefit from both approaches while minimising their drawbacks.

To overcome the main challenges of indoor vehicle localization, we propose TrackInFactory, a solution for the localization and tracking of industrial vehicles based on a particle filter (PF) that performs the sensor fusion in a novel way. The PF comprises a set of particles which weights represent position probabilities and uses a set of methods to update particle's states and replace the ones with lower weights with new ones. Data fusion is applied using a tight coupling (TC) approach at the measurement level by combining raw measurements from each sensor (Wi-Fi, IMU, and encoder) to provide an estimated position. Entire coverage of the factory plant is provided by Wi-Fi, which allows to obtain an initial position. Then, Wi-Fi, heading and displacement samples allow the PF to update the particles states considering noise in the sensors. The PF is capable of estimating the position and heading of the vehicle, continuously. Besides, it can be easily adapted to various types of manually operated or autonomous industrial vehicles. We have recurred to simulation to validate our approach before conducting experiments in the real-world with a real industrial vehicle in a factory-like environment.

This article introduces two main novel contributions to the use of PFs for sensor fusion in Wi-Fi-based indoor positioning. First, Wi-Fi data is fused with motion data using a TC approach that maximizes all the available Wi-Fi information. Second, a new method is introduced to dynamically update the particles' weights using a new reliability metric that defines the confidence of each position estimate. Moreover, a significant additional contribution of this article is the real-world experiments conducted in an industrial scenario with an industrial tow tractor, that allowed to compare the performance between the proposed solution and a loose coupling (LC) solution.

## II. RELATED WORK

Different technologies may be used for the localization of industrial vehicles depending on the application requirements, being the cost, accuracy and reliability the most important factors when choosing the technology. The localization of industrial vehicles indoors may resort to camera vision, ultrasounds, magnetic, laser ranging (LiDAR), ultra wide-band (UWB), RFID, Wi-Fi, or Bluetooth. These technologies are often combined with motion sensors using sensor fusion to improve positioning performance. Additional details about real-time localization systems in industrial settings can be found in [7].

UWB systems are used for vehicle localization in [2] and [8]. In [2], UWB is combined with GPS to provide indoor/outdoor positioning of industrial vehicles. Sensor fusion is performed with a PF, achieving an average error of 0.16 m in simulations. A drawback of this solution is that it depends on line-of-sight propagation and it is expensive due to the use of UWB.

Wu *et al.* [9] proposed an indoor positioning system (IPS) for mobile robot localization that does not depend on additional infrastructure because positioning relies on the magnetic

field and LiDAR data. The initial position is obtained from magnetic fingerprinting while a finer position is obtained by laser scan matching. The empirical experiments performed in an area of  $57 \times 19$  m revealed an RMSE of 1.43 m considering that the initial pose is known.

RFID has been explored in [3] and [10] for mobile robot localization. In [3], passive tags placed on walls of the space are detected by a robot equipped with RFID antennas. The robot's pose is obtained from the sensory fusion of RFID with odometry, using a similarity sensor model. Experiments have revealed a mean error of 0.14 m in an area with around 230 m<sup>2</sup>. Both solutions are complex to scale for larger buildings due to the installation of RFID tags and since they can also suffer from interference from Wi-Fi or Bluetooth.

Although UWB, LiDAR and RFID technologies can meet the demanding requirements of indoor tracking of industrial vehicles, they are expensive. Alternatively, Wi-Fi is low-cost and ubiquitous since it is present in most buildings, including factories, thus it is an ideal technology for deployments that take advantage of already existing infrastructure. Wi-Fi has been explored for localization in industrial environments to improve productivity in [11] and [12], and explored for vehicle tracking in [13] and [14]. The IPS for vehicles based on Wi-Fi fingerprinting [14] uses a raw data smoothing technique to feed a classifier based on neural networks, which can deal with noisy Wi-Fi signal strength measurements. This solution achieved a mean error of 2.25 m in performed tests.

Traditionally, sensor fusion for positioning can be performed using LC or TC approaches. In LC, sensor fusion is performed at the position level, i.e., positions obtained from distinct estimators are combined into a new estimate. In TC, measurements from each sensor are combined in an algorithm to generate a unique estimate. Different Bayesian techniques have been explored in the literature about sensor fusion in indoor positioning, namely, the Kalman Filter (KF) [15], the extended KF (EKF) [16], and the PF [2], [3], [17]. KF models the uncertainty to be Gaussian distributed, hence it is suitable for problems where the noise in data tends to be Gaussian and uni-modal. EKF, a variation of the KF for nonlinear problems, is used in GPS and navigation as well as in the positioning of mobile robots [16]. Similarly to the EKF, the unscented KF (UKF), is used in nonlinear problems. The PF is a subset of Monte Carlo methods suitable for nonlinear and non-Gaussian problems because it does not assume linearity and Gaussian nature of noise in data. Some advantages of the PF are [17]: many particles represent several possible positions hypotheses simultaneously; it can work with different sensor types, motion dynamics and noise distributions; and, it is simple to implement. Even though the PF is computationally expensive, it is possible to adjust the number of particles, adapting it to the available computing resources.

In recent years, researchers have been working on novel approaches aiming to merge Wi-Fi with DR. Some systems adopt PFs to locate pedestrians using Wi-Fi+PDR [18], [19]. Other systems explore simultaneous localization and mapping (SLAM) techniques for indoor localization, e.g., in [20], a graph is used to combine Wi-Fi fingerprinting with DR information from a phone. Loop closure and turn detection

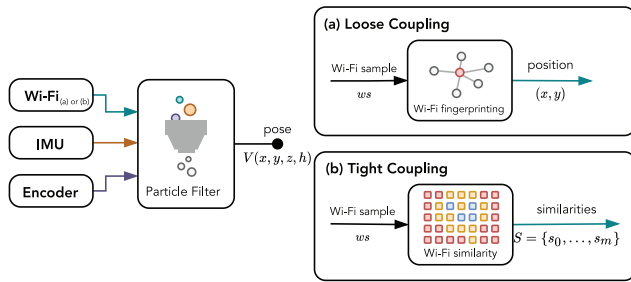


Fig. 1. TC and LC approaches.

are used to correct the estimated trajectory and eliminate cumulative errors from motion sensors. In experiments conducted in an area of  $130 \times 70$  m, this solution achieved an accuracy of 0.6 m and 4.76 m with Tango-based PDR (high motion tracking accuracy using visual-based odometry) and step-counter-based PDR, respectively.

In our approach, TC sensory fusion of Wi-Fi and motion data takes advantage of the fusion at the measurement level. It offers a good cost-benefit balance as it leverages existing WLAN infrastructure to provide coverage of the entire operational area and uses motion sensors to allow continuous tracking. In alternative to trajectory correction techniques, such as loop closure, the proposed solution takes advantage of the PF and a novel reliability metric that dynamically adjusts the particles' weights based on Wi-Fi similarity.

In summary, using accurate sensors (e.g., LiDAR) or technologies (e.g., UWB or RFID) for indoor localization leads to high positioning accuracy, but is expensive. On the other hand, low-cost solutions, such as Wi-Fi, are ubiquitous but have low accuracy. The wide adoption of PFs in previous works has shown they are suitable to combine Wi-Fi data with other sensors for indoor localization. In this article, we take advantage of previous knowledge and use a PF with a novel TC approach for industrial vehicle tracking, an area of indoor positioning that lacks significant contributions.

### III. TRACKINFACTORY: PARTICLE FILTER BASED ON TIGHT COUPLING

This section introduces TrackInFactory, an IPS based on a PF that combines Wi-Fi fingerprinting with DR for more accurate and reliable position estimates.

#### A. Tight Coupling Versus Loose Coupling

Since the Wi-Fi RM covers the entire area of the building and each particle represents the probability of the vehicle to be at a certain position, Wi-Fi has an important role in updating particles' weights. Wi-Fi fingerprinting can be used either to provide a position estimate [Fig. 1(a)] or to obtain a set of similarities between the operational WS and the reference samples of the RM [Fig. 1(b)]. Both outputs can serve as input for the PF, being the Wi-Fi position estimate related to a LC approach, whereas the set of similarities related to a TC approach.

The LC approach, explored in [21]–[23], combines a Wi-Fi position estimate with data from other sensors in the fusion

algorithm. For instance, a Wi-Fi fingerprinting method based on kNN first computes the similarity between the WS and the reference samples from the RM. Then, a position estimate is obtained based on the most similar WSs and is combined with data from other sensors to produce a more accurate position estimate. Large errors are usual in Wi-Fi-based positioning due to different signal propagation effects, hence this approach has the additional challenge of trying to eliminate the large positioning errors.

The TC approach has been explored in [18] (using Wi-Fi) and in [3] (using RFID). In a TC approach based on Wi-Fi, the similarities between a new WS and the RM are computed. Then, the similarities are used in updating the particles' weights. Each particle has a weight based on the similarity of its closest reference point (RP). In contrast with the LC approach, which only considers a small set of RM samples, this approach considers the entire RM, therefore it benefits from all available information. Moreover, since TC does not rely on a position estimate, it is less influenced by large positioning errors.

#### B. Particle Filter

PFs are one of the best tools to solve nonlinear, non-Gaussian problems, especially because noise is non-Gaussian in heading observations (due to magnetic disturbances that affect the heading provided by the IMU [24]) and in Wi-Fi signal strength measurements.

Another advantage of PFs is that floor plans can be easily integrated, which is important for indoor vehicle tracking, as it allows to define navigable areas (spaces where vehicles operate), consequently, position estimates are improved by removing particles that hit walls or obstacles.

PFs are based on the idea that a set of particles represents the belief of the vehicle being in a position and comprise three iterative steps named predict, update, and resampling. In the predict step, particles states are estimated based on the previous particle state and a motion model, taking into account the noise models of each sensor. In the update step, particles' weights are updated according to measurements from the sensors. After initialization, particles spread throughout the space before they start converging into a position where it is more likely for the vehicle to be at. This is achieved by assigning different weights to particles, being that particles that are closer to the vehicle's position have higher weights, conversely, particles further from the vehicle have lower weights. Lower weight particles are removed in the resampling step, and new particles are created by cloning higher weight particles. Finally, the position and heading of the vehicle are estimated based on the set of particles with higher weights.

Fig. 2 represents the main algorithm of the proposed PF. It starts with the particles' initialization, then waits for samples from sensors (displacement, heading or Wi-Fi). Particles move according to heading and displacement measurements considering noise. Upon receiving a new WS, the particles' weights are updated and then the particles are subject to resampling. The vehicle's position and heading are updated after this process.

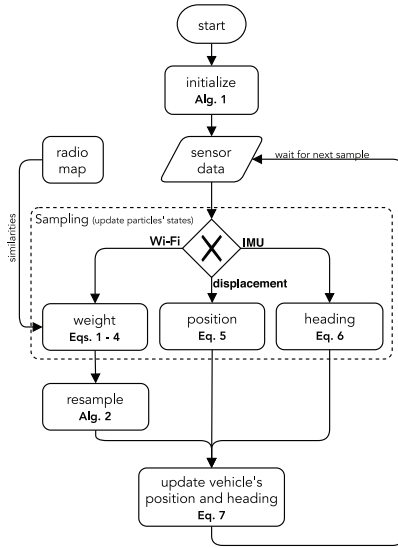


Fig. 2. PF main algorithm.

A particle is defined as  $p_i = (w, x, y, z, h, ho)$ , where  $w$  represents the weight of the particle,  $x, y,$  and  $z$ <sup>1</sup> represent the position coordinates of the particle,  $h$  represents the heading and  $ho$  represents the heading offset, necessary to estimate the heading of the vehicle (see Section III-C).

### C. Initialization

Particles are created around the positions where it is more likely for the vehicle to be at, to minimize large initial errors due to particles being uniformly dispersed throughout the operational area.

Before creating particles, an absolute position is necessary to determine where to place them. Wi-Fi fingerprinting allows to obtain an absolute position, however, there is a certain probability to have a high error in the initial position. To minimize the initial error, access points (APs) signal strengths of the first few WSs are averaged into a merged WS (only received signal strength indicator (RSSI) values of detected APs are considered), to increase the probability of placing the particles closer to the real position [25]. The similarity between the averaged sample and all RM samples is obtained, resulting in a list of samples ordered by similarity. Particles are created around the positions (RPs) where the most similar RM samples were collected.

As the initial heading of the vehicle is unknown, each particle has an initial heading offset ( $ho$ ) randomly set. Once particles start moving, the ones with the correct heading offset will remain alive because they have higher weights, hence after a few iterations particles converge to the correct heading.

Algorithm 1 describes how particles are created around RPs where the merged WS is more similar. See Section III-D1 for details about the initial particles' weights.

<sup>1</sup>For simplicity, the vehicle is assumed to move in a plane, therefore with constant  $z$ .

### Algorithm 1 Particles Initialization

#### Input

- $rm$  - radio map, set of WSs
- $ws$  - sample of averaged RSSIs of first  $M$  WSs
- $k$  - number of RPs where particles will be created around
- $N$  - total number of particles ( $\{N \in \mathbb{Z} \mid N > k\}$ )

#### Output

- $P$  - set of all particles.

#### 1: procedure PARTICLES INITIALIZATION

- 2:  $P = \{\}$
- 3:  $S = dsim(rm, ws)$  //compute Manhattan distance (dissimilarity) between samples of  $rm$  and  $ws$
- 4:  $RPs =$  set of  $k$  RPs from  $S$  that are more similar with  $ws$
- 5:  $np = N/\#(RPs)$  //number of particles to be created around each RP
- 6: **for**  $rp \in RPs$  **do**
- 7:  $s_n =$  normalized similarity between  $ws$  and  $rp$ , defined in Eq. 2
- 8: **for**  $i = 1 \in \{1 \dots np\}$  **do**
- 9:  $w \leftarrow s_n$
- 10:  $x \leftarrow rp.x + rand(0, r)$
- 11:  $y \leftarrow rp.y + rand(0, r)$
- 12:  $z \leftarrow 1$
- 13:  $h \leftarrow 0$
- 14:  $ho \leftarrow rand(-\pi, \pi)$
- 15:  $p \leftarrow (w, x, y, z, h, ho)$
- 16:  $P \leftarrow P \cup \{p\}$

### D. Sampling

The prediction and update steps are performed simultaneously when new sensor samples are obtained. The prediction is based on the propagation of particles according to heading and displacement samples, which consider noise modelled after the IMU and encoder, respectively. Therefore, whenever a new heading or displacement sample is collected, the sampling process includes the predict (added noise) and the update (new observations) in the same step (Section III-D2). The update step also includes the process to update the particles' weights, which is based on the particle's current state and the latest WS.

1) *Update Particles' Weights:* Wi-Fi ensures absolute positioning coverage of the entire operational area, therefore it is used to update particles weights as it allows to assign higher weights to particles that are closer to the most probable positions, and lower weights to particles that are further away from the most probable positions.

In this context, the RM is defined by  $RM = \{(\rho_1, ws_1), \dots, (\rho_m, ws_m)\}$ , which is the set of  $m$  WSs, each associated to the position  $\rho$  where it was collected. Each WS, defined as  $ws = \{RSSI_1, \dots, RSSI_n\}$  represents the set of received RSSI values detected by a Wi-Fi interface.

In Wi-Fi fingerprinting, a position estimate is usually obtained by computing the centroid of the most similar RM samples. While this estimated position can be used to update the weights of the particles (LC), this approach does not use all the available information (set of all the similarities between the new WS and all the RM samples, defined as  $S = \{s_1, \dots, s_m\}$ ).

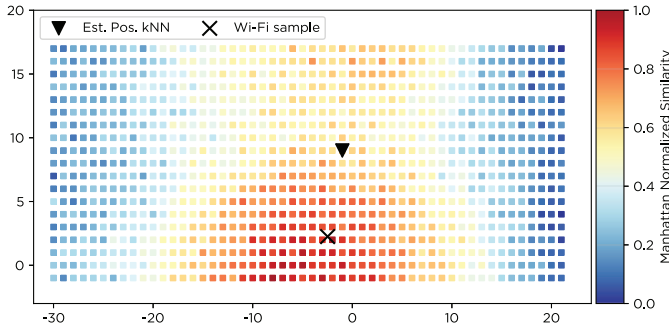


Fig. 3. Normalized similarity between a WS and RM samples.

Statistically, RM samples that are more similar to the WS are the ones closer to the true position of the WS, as depicted in Fig. 3.

In our approach, the set of all similarities is used when updating particles weights (TC). Two fingerprints are compared using a distance (dissimilarity) or similarity function, e.g., Manhattan, Euclidean or Minkowski distances, or the cosine similarity. We have opted for the Manhattan distance since it showed good performance in Wi-Fi fingerprinting [26]–[28]. The expression that defines the distance between a WS and a RM sample, obtained with the Manhattan distance, is presented next

$$s = \sum_{i=1}^n |\text{RSSI}_i^{rm} - \text{RSSI}_i^{ws}| \quad (1)$$

where  $\text{RSSI}_i$  refers to the signal strength of  $\text{AP}_i$  from the RM sample ( $rm$ ) and the Wi-Fi sample ( $ws$ ). If an AP is missing either in the RM sample or the WS, a default RSSI value of  $-90$  dBm is used. Once this is performed for all samples of the RM, the Manhattan distance is normalized and converted into the Manhattan normalized similarity, defined as

$$s_n = \frac{\max(S) - s}{\max(S) - \min(s)} \quad (2)$$

where  $S$  represents the set of all distances between the WS and all RM samples.

Fig. 3 shows an example of the normalized similarity between a WS and the RM samples. Areas with higher similarity are depicted at the top and bottom of the figure. If the traditional LC approach were used, the centroid could fall in an area with low similarity (black triangle), which is displaced with respect to the real location of the WS. This issue does not exist in a TC approach because it considers all available information and it does not depend on the estimated position.

The particles' weights have an important role in the PF as they define which particles are removed in the resampling process, and they are used when estimating the vehicle's position and heading. The weight of a particle is defined as

$$w_t = w_{t-1} * (1 - \alpha) + s_n * \alpha \quad (3)$$

where  $w_{t-1}$  represents the previous particle weight,  $s_n$  represents the normalized similarity, and  $\alpha$ , a factor used to define how much the latest available Wi-Fi information affects the particles' weights.

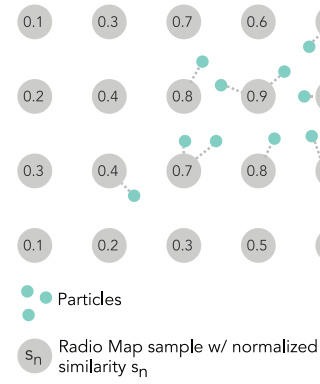


Fig. 4. Assigning a similarity value to each particle.

The normalized similarity ( $s_n$ ) component of the particle's weight is obtained from the nearest RP as shown in Fig. 4. As the particle's weight depends on its closest RP, a dense RM with a low distance between neighbouring RPs (e.g.,  $1 \times 1$  m grid), will improve the system performance because it ensures that the particle's closest RP is always at a distance lower than the grid size. A RM with a larger distance between RPs can also be adopted as long as its density is increased using log-distance path loss (LDPL) [5], or other techniques to interpolate RSSI values.

2) *Displacement and Heading*: Continuous tracking of the vehicle's position is achieved by updating particles' positions and headings using the data provided by the encoder and IMU sensors, which measure the displacement and heading (orientation), respectively.

For each new displacement sample, particles' positions are updated as follows:

$$\begin{aligned} x_t &= x_{t-1} + (d_t + n_d) \times \sin(h_t) \\ y_t &= y_{t-1} + (d_t + n_d) \times \cos(h_t) \end{aligned} \quad (4)$$

where  $x_{t-1}$ ,  $y_{t-1}$  represent the previous position of the particle,  $d_t$  represents the displacement (in meters) with respect to the previous reading,  $h_t$  represents the heading, and  $n_d$  is a zero-mean Gaussian distributed random variable added to the displacement measurement. When a particle hits a wall or obstacle in this step, its weight is set to zero.

The selected AHRS IMU (see Section V-B) estimates AHRS orientation data using its algorithms, which take advantage of reference vectors (i.e., gravity and the earth's magnetic field) to compensate for gyroscopes integration drift, resulting in more reliable orientation data (lower drift), provided in Euler angles (roll, pitch, and yaw). Particles' headings are updated when a new IMU sample is received

$$h_t = \theta_t + n_\theta + h_o \quad (5)$$

where  $\theta_t$  represents the heading (yaw) measurement from the sensor at instant  $t$ ,  $n_\theta$  is a zero-mean Gaussian distributed random variable and  $h_o$  represents the heading offset (necessary to estimate the initial heading, see Section III-C) of the particle.

### E. Resampling

After updating particles' weights, particles are subject to resampling. Particles with weights lower than a given threshold are removed and new particles are created from the subset of particles with higher weights. In order to improve heading estimation and cope with drift in the heading, which is cumulative over time, new particles have a new heading offset based on the higher weight particle that was selected to be copied. This process is described in Algorithm 2.

Removing particles with lower weights has the purpose of reducing the time it takes the particles to cluster around the most likely position. In contrast, the removal of these particles is also a disadvantage of this resampling method because it may lead to particle impoverishment, which occurs because of the reduction of truly distinct particles.

Once the particles with lower weights are removed (line 2, Algorithm 2), the algorithm enters a loop (line 3, Algorithm 2) to create new particles until a total of  $N$  particles exists in  $P'$ . We adopted the multinomial resampling method [29], where particles are randomly selected according to their weight, using the weight as the probability of being selected (line 5, Algorithm 2). This allows higher weight particles to be selected with higher frequency than the ones with lower weights. Since we consider the particle's previous weights when updating the particle's weight, our PF does not follow a standard implementation in resampling where weights are normalized, to sum up to one, and all particles have the same weight of  $1/N$ . Instead, newly created particles that are copies of particles with higher weights inherit the original particle's weight. In the unlikely case that all particles have weight lower than  $w_{th}$ , only 70% of particles are replaced to ensure that there is particle diversity and avoid particle impoverishment.

In the initialization process, a random heading offset is assigned to each particle and, as the PF starts converging, particles that were initialized with the correct heading offset will have a higher probability of being selected and copied in this process, which will allow all particles to converge into similar heading offset values.

In the resampling process, new particles are created with a new heading offset to compensate for heading drift (line 6, Algorithm 2).

### F. Vehicle's Position and Heading

The vehicle's pose is estimated from the weighted average of particle's positions and headings

$$V(x, y, z, h) = \left( \frac{\sum_{i=1}^N p_i(x, y, z) \times p_i.w}{\sum_{i=1}^N p_i.w}, \tan^{-1} \left( \frac{\sum_{i=1}^N \sin(p_i.h) \times p_i.w}{\sum_{i=1}^N \cos(p_i.h) \times p_i.w} \right) \right). \quad (6)$$

### G. Confidence in Position Estimates and Dynamic $\alpha$

Larger positioning errors may occur during the first iterations after initialization because the initial position and heading are unknown. During this period, Wi-Fi should have a higher influence on particles' weights because it will result

---

### Algorithm 2 Resample Particles

---

#### Input

$w_{th}$  - weight threshold, particles with weight lower than it are removed

$n_{ho}$  - zero-mean Gaussian distributed random variable, to minimize drift and improve heading estimation

#### Output

$P'$  - set of resampled particles

```

1: procedure RESAMPLE PARTICLES
2:    $P'$  = list of particles with weights higher than  $w_{th}$ 
3:   if  $\#P' == 0$  then
4:     //If all particles have weights lower than  $w_{th}$ 
5:      $P'$  = list of 30% of particles with higher weights
6:   while  $\#(P') < N$  do (multinomial resampling)
7:     //create copies of particles from  $P'$ 
8:      $p$  = particle selected from  $P'$  using its weight as
        probability
9:      $ho \leftarrow p.ho + n_{ho}$  //particle's heading offset com-
        posed of the original particle offset + noise component
10:     $p' \leftarrow (p.w, p.x, p.y, p.z, p.h, ho)$ 
11:     $P' \leftarrow P' \cup \{p'\}$  //add copied particle to the set of
        resampled particles

```

---

in removing particles in areas where it is less likely for the vehicle to be at. As particles are dispersed throughout the space during the first iterations of the PF, higher weights will be attributed to particles in areas where it is more likely for the vehicle to be, and lower weights to particles that are in unlikely positions. Since Wi-Fi defines how particles' weights are updated, it also has an important role to minimize drift in the heading. The  $\alpha$  parameter, introduced in Section III-D1, defines how much the particle's weight is influenced by the Wi-Fi similarity.

We devised a mechanism in which a confidence measure is used to define the value of  $\alpha$ . When the confidence is lower (meaning that a higher positioning error might exist), a larger value is assigned to  $\alpha$  which gives a higher influence to Wi-Fi. This mechanism is continuously performed since the initialization step.

The dispersion of particles  $\vartheta$  (in meters), defined as the average distance between the particles' positions and the PF estimated position, is given by

$$\vartheta = \frac{1}{N} \sum_{i=1}^N d(V, p_i) \times w_i \quad (7)$$

where  $V$  is the PF estimated position,  $p_i$  represents the  $i$ th particle position and  $w_i$  represents the particle's weight.

The dispersion is used to define a metric of confidence as

$$C(\vartheta) = \begin{cases} -\frac{1}{r_{\max}} \vartheta + 1, & 0 \leq \vartheta < r_{\max} \\ 0, & \vartheta \geq r_{\max} \end{cases} \quad (8)$$

where  $r_{\max}$  defines the maximum allowed dispersion (in meters) between particles. We defined  $r_{\max} = 4$  m, since, even for tracking, more than 4 m radius is a significant area to have confidence about the position of the vehicle.

The confidence was modelled after an empirical analysis of the particles' dispersion in simulations, as described in

Section IV. Upon our analysis, we found that when the cloud of particles is smaller (lower dispersion), there is usually higher confidence associated. Moreover, when the dispersion of particles is larger (e.g., during first iterations) the positioning error tends to be higher.

The confidence measure has to be a reliable indicator of whether there is a large error in position estimates, hence we have conducted experiments with synthetic data to define the appropriate function that defines the confidence. In Section IV-E we show the correlation between the confidence and the positioning error, demonstrating that the confidence function is reliable.

Even though  $\alpha$  could be a direct translation of the confidence value, i.e.,  $\alpha = 1 - C$ , it would greatly impact the PF estimated position, as Wi-Fi would have too much influence on particles' weights resulting in large errors. Since large positioning errors occur when  $\alpha \geq 0.6$  (see Section IV-C), was decided to use the next expression to ensure that when confidence is low, Wi-Fi does not have too much importance

$$\alpha = 0.6 - 0.6 \times C. \quad (9)$$

We have used simple (linear) models for  $C$  and  $\alpha$  in our implementation, because they have been effective in simulations, but we intend to explore other models in future works.

#### IV. SIMULATION

Simulations were performed to tune the PF parameters before running tests in a real environment. Initially, we created a simulated environment and generated synthetic data of displacement, heading and Wi-Fi measurements, affected by noise and drift in the heading. Finally, the proposed PF was tested and its parameters tuned to achieve the best results.

##### A. Simulation Scenarios

Two simulation scenarios were considered, one representing an empty building ( $50 \times 20$  m), without obstacles, and another representing a building with the same dimensions but with space restrictions (the vehicle cannot navigate in some areas). Six trajectories were considered in these experiments: three random trajectories (RTs) and three loop trajectories (LTs). The purpose of RT is to show that the PF is capable of converging, even without any space restrictions, and the purpose of LT is to show how the PF performs in a realistic scenario, with obstacles and space restrictions. Each RT is defined by a random movement with a travelled distance of approximate 500 m. Each LT is defined by a cyclic movement, where the vehicle performs several laps in a predefined trajectory and has an average travelled distance of about 1750 m.

##### B. Synthetic Data

As TrackInFactory depends on data from Wi-Fi and motion sensors, it is necessary to emulate them by generating synthetic data (heading, displacement and Wi-Fi) synchronized in time. To achieve that, it was assumed that the vehicle follows a trajectory from point A to point B at a constant speed of 1 m/s, then it stops for 1 s at B, before proceeding to the next point. A trajectory is composed of several points. Based on

the output of sensors to be used, sampling rates of heading, displacement and Wi-Fi were defined as 20, 50, and 1/2 Hz, respectively. Wi-Fi scanning in the 2.4-GHz frequency range usually takes more than 1.5 s to collect the signal strength values, hence it was decided that WSs were generated every 2 s. A displacement sample is defined as

$$ds = l + n_{ds} \quad (10)$$

where  $l$  represents the true traveled distance and  $n_{ds}$  represents a zero-mean Gaussian random distribution that we defined as  $N(0, 0.004$  m). Heading samples are generated as

$$hs = \beta + n_{hs} + \phi \quad (11)$$

where  $\beta$  represents the true absolute heading,  $n_{hs}$  represents a zero-mean Gaussian distributed random variable, and  $\phi$  represents the cumulative drift defined in degrees per hour, meaning that drift increases over time, hence after one hour drift has a value of  $\phi$  degrees. We defined  $n_{hs} = N(0, 10^\circ)$  and  $\phi = 20^\circ/\text{h}$ .

A simple model based on LDPL was used to calculate signal strength values of APs. A signal strength value is defined as

$$\text{RSSI} = \text{RSSI}_0 - 10 \times \eta \times \log_{10}(\xi) + n_{\text{rssi}} \quad (12)$$

where  $\text{RSSI}_0$  represents the RSSI value at 1 m distance of the AP,  $\eta$  represents the path loss exponent,  $\xi$  represents the distance between the AP and the point where the signal strength is estimated, and  $n_{\text{rssi}}$  is a zero-mean Gaussian distributed random variable. We defined  $\text{RSSI}_0 = -40$  dBm,  $n_{\text{rssi}} = N(0, 4$  dBm), and  $\eta = 2$ .

Both RM samples, as well as WSs of trajectories, were created using this model. The RM of the simulation scenario comprises 20 WSs at each RP for a 1x1 m grid of RPs.

In the definition of noise components ( $n_{hs}$  and  $n_{ds}$ ) and drift in the heading ( $\phi$ ), we assumed that the sensors produced more noise than normal (industrial-grade) sensors, to test the proposed solution under challenging conditions. The Wi-Fi noise component was defined considering that Wi-Fi signals have high variability in indoor environments, due to multipath and other propagation effects.

##### C. Analysis of Results Using Constant $\alpha$

Before modeling the confidence function, described in Section III-G, we studied how  $\alpha$  affects results, using a constant  $\alpha$  in all trajectories.

The PF was run three times for each trajectory, using different random seeds. Table I (top part) shows the overall positioning error that includes position estimates of all runs of LT and RT trajectories. The Euclidean distance between the estimated and true positions was used as the error metric.

A clear deterioration of results was observed for  $\alpha \geq 0.6$ . This suggests that in the long run, the latest WS should have a low influence over the particles' weights, therefore we defined the value of  $\alpha$  (9) varying between 0 and 0.6 as a function of the confidence. When the confidence is high,  $\alpha$  is low, and vice-versa. This approach is important in the warm-up period during which Wi-Fi has more importance while the particles are still converging into a position and heading where it is

TABLE I  
POSITIONING RESULTS (IN METERS) OF SIMULATED TRAJECTORIES FOR  
DIFFERENT  $\alpha$  (TOP) AND  $w_{th}$  (BOTTOM) VALUES

$\alpha$	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Mean	0.70	0.71	0.82	1.47	1.63	3.81	4.36	6.51	4.07	4.63
Median	0.49	0.50	0.56	0.64	0.69	0.80	0.94	3.19	1.12	1.24
$P_{75th}$	0.86	0.90	1.07	1.29	1.44	2.70	4.04	5.36	3.27	4.06
$P_{95th}$	1.98	2.09	2.43	4.38	5.02	20.56	21.60	34.09	21.45	24.60
$P_{99th}$	3.69	3.28	3.54	19.45	18.24	31.24	29.75	46.94	43.83	41.26
Max	8.46	6.93	6.51	27.91	36.78	38.10	35.30	52.19	52.43	50.08
$w_{th}$	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Mean	0.88	0.87	0.87	0.81	0.77	0.73	0.71	12.30	16.59	N.A.
Median	0.49	0.49	0.49	0.49	0.48	0.49	0.50	11.61	16.01	
$P_{75th}$	0.94	0.92	0.93	0.92	0.90	0.91	0.90	18.02	23.14	
$P_{95th}$	2.78	2.87	2.61	2.15	2.00	1.93	2.09	25.93	33.81	
$P_{99th}$	8.28	8.03	7.99	6.81	5.65	4.49	3.28	31.96	44.86	
Max	13.11	10.95	11.07	10.79	10.29	8.58	6.93	34.93	47.22	

more likely to be the true vehicle's pose. After the warm-up period, confidence tends to be high, hence  $\alpha$  will be lower.

#### D. Analysis of Weight Threshold $w_{th}$

We also analyzed the PF performance for different values of the weight threshold ( $w_{th}$ ) used in the resampling process. In this process, if the weight threshold is too low, particles will move freely in RT trajectories or will move freely until they hit walls or obstacles in LT trajectories. If it is too high, too many particles are removed in the resampling process.

Based on the previous results, we conducted the same experiment for LT and RT trajectories using  $\alpha = 0.2$  and tested different values of  $w_{th}$  to understand how this parameter affects results. Table I (bottom part) shows the overall positioning error that includes position estimates of RT and LT trajectories, obtained for different  $w_{th}$  values. When  $w_{th} = 0.7$  it achieves the best mean,  $P_{75th}$ ,  $P_{99th}$  and maximum error, therefore it is the selected value for  $w_{th}$  in experiments, including the ones of Section IV-C. As expected, the PF performance degrades when  $w_{th}$  is too high, i.e., for  $w_{th} \geq 0.8$  the PF removes an excessive amount of particles in resampling, hence the PF is not able to converge.

#### E. Simulation Results

The following results consider  $\alpha$  as a function of  $C$ , as described in Section III-G. Each trajectory was run three times, as previously. Positioning error statistics include all position estimates since  $t = 0$ , including the initial warm-up period.

In Table II, we show the positioning results of LTs, RTs and all trajectories. In comparison with the constant  $\alpha$  results, the major improvement was observed in the overall maximum error from 6.93 m (for  $\alpha = 0.2$ ) to 5.95 m, besides, all other performance metrics are also improved. Also, better results are achieved in LT, in comparison to RT, because many particles are removed when they hit walls or obstacles. This filters out those particles that are in a less-likely situation where the vehicle suddenly jumps to an area that is not linked to the previous positions and, therefore, improving positioning. Conversely, the positioning error is larger in RT because particles are more dispersed, and they are removed in the resampling process or when they hit the limits of the open-space building.

TABLE II  
POSITIONING RESULTS OF RT, LT, AND ALL TRAJECTORIES USING A  
DYNAMIC  $\alpha$  (IN METERS)

	LT	RT	All
Mean	0.39	1.07	0.66
Median	0.34	0.86	0.48
$P_{75th}$	0.50	1.35	0.80
$P_{95th}$	0.82	2.46	1.84
$P_{99th}$	1.51	3.93	3.15
Max	3.55	5.95	5.95

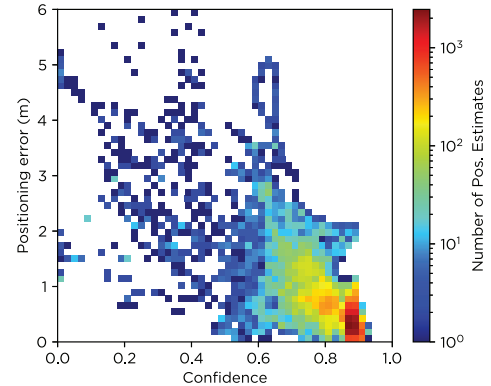


Fig. 5. Density plot of position error versus confidence.

#### F. Positioning Error Versus Confidence

Fig. 5 depicts a density plot of the positioning error versus confidence, considering all position estimates of all simulations. It shows that there is a relationship between the confidence (given by the dispersion of particles) and the positioning error, where the vast majority of position estimates with high confidence ( $>0.8$ ) have positioning errors lower than 1 m. There is a strong negative correlation between the positioning error and confidence, which is statistically significant,  $r_{\text{Pearson}} = -0.70$ ,  $p = 0.0$ . A logarithmic scale is used in the color scale of Fig. 5 to better represent these results.

#### G. Parameters

Based on the simulation process and preliminary experiments with the vehicle prototype (see Section V-B), we have defined the parameters of the PF used in simulations and real-world experiments. In the initialization process we have set  $N = 3000$ ,  $M = 3$ , and  $k = 6$ . The random variables characterizing noise in the displacement and heading were defined as  $n_d = N(0, 0.01 \text{ m})$  and  $n_\theta = N(0, 1^\circ)$ , respectively, based on sensor's datasheet information. In the resampling process,  $w_{th} = 0.7$  and  $n_{ho} = N(0, 2^\circ)$  were used.

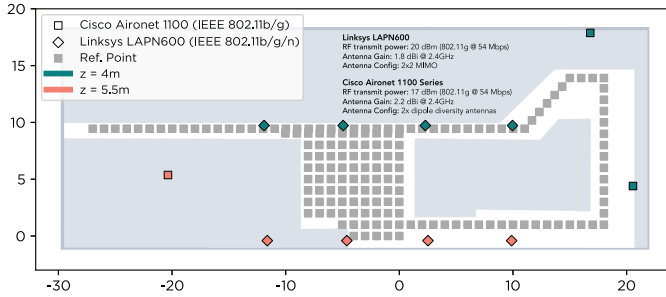
## V. REAL-WORLD EXPERIMENTS

We performed experiments with an industrial vehicle in a factory-like building. First, we mapped and affixed tags to the floor to be used as RPs in the testing scenario. Second, we have equipped the vehicle with a positioning module. Finally, we performed experiments to evaluate our solution in a real environment.





Fig. 6. PIEP building at University of Minho.

Fig. 7. Floor plan of testing scenario—RPs (gray squares) and the installed APs according to their model, network standard and height ( $z$ ).

### A. Testing Scenario

Experiments were conducted at the PIEP building (at the University of Minho) which measures approximately 50 by 20 m and is more than 8 m high. The PIEP building is dedicated to research on plastic polymers, hence it is similar to a factory plant with large open spaces, large machinery and metal objects (injection molds), as shown in Fig. 6.

Fig. 7 shows RPs where RM samples were collected (gray squares), the positions and specifications of the APs present in the building (colored squares and diamonds), and the navigable areas (places where the vehicle can operate). The distance between adjacent RPs is 1 m in most cases. Also, we affixed tags to the floor marking RP locations which allowed collecting ground truth using a video camera (pointed to the floor) to register the time when the vehicle has passed over a tag. It took about 180 min. to build the RM with 178 RPs, considering 40 s to collect 20 WSs at each RP and 30 s to move the vehicle to a new RP.

### B. Vehicle Prototype

We have installed a positioning module in a tow truck (Toyota BT Movit TSE300) used to transport materials in factories. Fig. 8 shows the integrated positioning module composed of a Raspberry Pi Model 3B as well as the following sensors: four Edimax EW-7811un external Wi-Fi interfaces; an Xsens MTi-300 AHRS IMU to measure the heading; and, a U.S. Digital A2 absolute encoder to measure the displacement. The encoder sensor is attached to the back of the vehicle with a custom mount. We use multiple Wi-Fi interfaces since signals of distinct Wi-Fi interfaces are poorly correlated [25], which allows to average signal strength values into a merged WS.

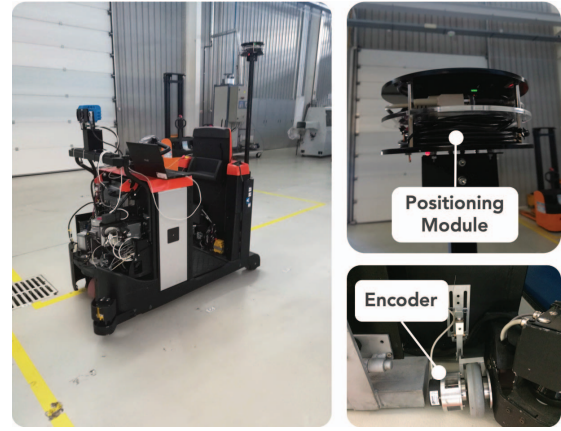


Fig. 8. Industrial vehicle (tow tractor) used in tests.

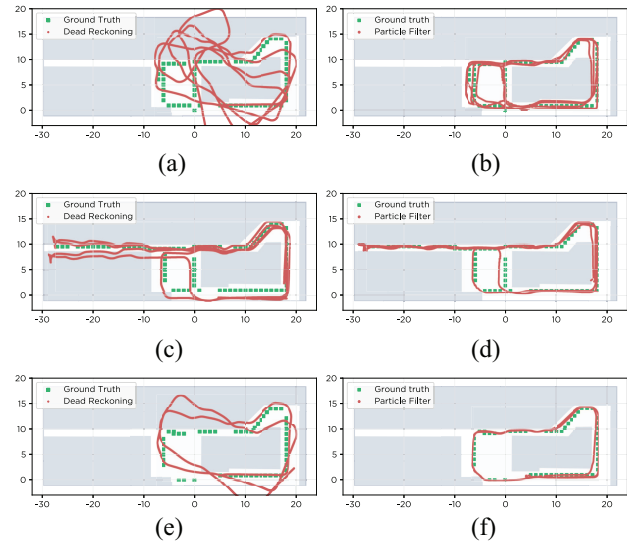


Fig. 9. Comparison between DR and PF estimated trajectories. (a) DR trajectory 1. (b) PF trajectory 1. (c) DR trajectory 2. (d) PF trajectory 2. (e) DR trajectory 3. (f) PF trajectory 3.

The positioning module can be easily installed in other types of vehicles because it operates independently of the vehicle, but, it still depends on coupling the encoder to the vehicle.

### C. Results

Carried tests with the industrial tow tractor included three trajectories with a duration of 15, 15, and 7 min, respectively. The total distance covered over all trajectories sums up to approximately 775 m.

Fig. 9 displays the DR and PF trajectories. The DR trajectories were obtained using an initial position and heading of the vehicle, while in the PF trajectories the initial pose was unknown. Drift in the heading is present in all DR trajectories. Conversely, the PF can compensate for the drift as shown by the PF estimated trajectory. This is possible due to the removal of particles that hit walls or obstacles and also because the heading offset of particles is updated in the resampling process.

In order to compare TrackInFactory to the LC approach in [21], Fig. 10 shows how the positioning error varies over

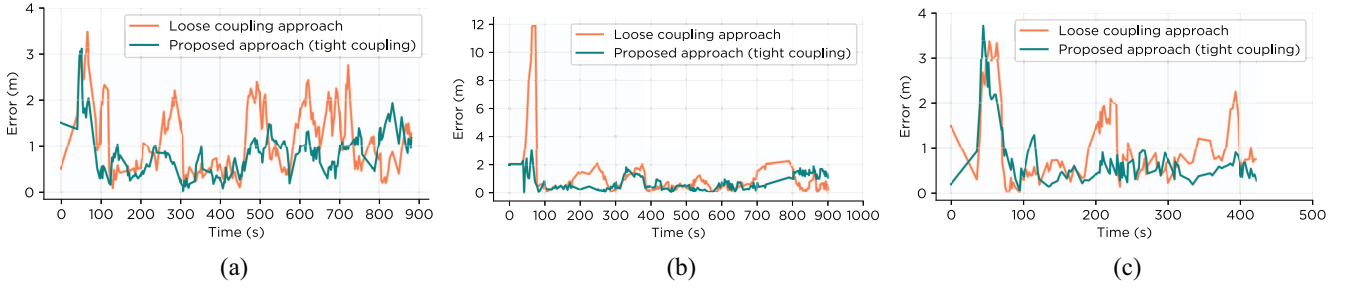


Fig. 10. Error over time of each trajectory using this PF and the LC approach [21]. (a) Trajectory 1. (b) Trajectory 2. (c) Trajectory 3.

TABLE III  
OVERALL POSITIONING RESULTS (IN METERS) OF EXPERIMENTS WITH INDUSTRIAL VEHICLE

	Wi-Fi	Loose coupling approach		Tight coupling approach	
	FP	Overall	AW	Overall	AW
<b>Mean</b>	2.15	1.03	0.85	<b>0.81</b>	<b>0.71</b>
<b>Median</b>	1.85	0.68	0.66	0.67	0.64
<b><math>P_{75th}</math></b>	2.97	1.40	1.23	<b>1.08</b>	<b>1.00</b>
<b><math>P_{95th}</math></b>	5.39	2.46	2.00	1.72	1.48
<b><math>P_{99th}</math></b>	7.48	6.87	2.33	<b>3.01</b>	<b>1.70</b>
<b>Max</b>	12.24	12.26	2.88	<b>4.00</b>	<b>2.06</b>

FP – Fingerprinting; AW – After warm-up.

time for both solutions, in each testing trajectory. As expected, errors are higher during the initial phase, while the PF is converging. We refer to this phase as the warm-up period and it comprises the position estimates in the interval  $t \in [0 \text{ s}, 100 \text{ s}]$ .

TrackInFactory converges faster, has a lower maximum error and a better mean error after converging. This is achieved due to the process to update particles weights and resampling which rapidly removes particles from improbable areas. In [21], when a particle is more than  $\sqrt{2}$  m of its nearest RP, it is removed. This method is equivalent to using the floor plan to remove particles.

Results of all trajectories are aggregated and presented in Table III. It shows a comparison between the proposed TC approach, the LC approach [21] and pure Wi-Fi fingerprinting (kNN with  $k = 5$ ). The proposed TC approach has significant improvements especially in the last quartile in comparison to Wi-Fi fingerprinting and the LC approach. The maximum error with the TC approach after the warm-up time (2.06 m) is lower than the mean error achieved with Wi-Fi fingerprinting (2.15 m), showing that the TC approach fuses information efficiently taking advantage of all information from the set of similarities, instead of relying on a previously estimated position based on Wi-Fi, which is the case in fingerprinting and LC.

Results of experiments in a real environment are similar to the ones achieved with the simulations (see Section IV-E) which demonstrate that the PF was properly configured and the simulation model is reliable.

Fig. 11 shows the CDF curves of the different positioning methods. CDF curves of TC and LC are similar for probabilities below 0.6, but after that point, we see the clear improvement of the proposed solution over the LC one. The

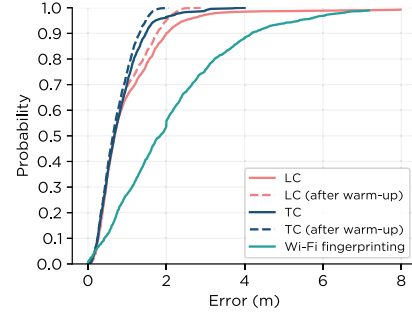


Fig. 11. CDF of positioning error.

TABLE IV  
OVERVIEW OF THE PROPOSED AND OTHER SYSTEMS

Solution	Technology	Localization Technique	Mean Error (m)
[2]	UWB + GPS	PF	0.16
<b>TrackInFactory (tight coupling)</b>	Wi-Fi + MS	PF w/ tight coupling	<b>0.81</b>
[21] (loose coupling)	Wi-Fi + MS	PF + W-FP	1.03
[13]	Wi-Fi + odometry + LiDAR	PF	1.2
[18]	Wi-Fi + PDR	PF	1.2
[19]	Wi-Fi + IMU + FM Transmitters	PF w/ W-FP	2.11
[14]	Wi-Fi	W-FP	2.25

PF – Particle Filter; MS – Motion Sensors (IMU and encoder); W-FP – Wi-Fi fingerprinting.

curves after the warm-up are the ones that better represent the positioning performance in a realistic scenario because it is expected to be running in vehicles for long periods, in which the larger errors in the beginning (while the PF is converging) are less relevant. In positioning applications with demanding requirements, such as high accuracy and low maximum error, the performance in the last quartile is very important. Our solution achieved 1.0 and 1.70 m in the  $P_{75th}$  and  $P_{99th}$ , respectively, showing that it is capable of locating industrial vehicles accurately.

Table IV shows an overview of the proposed solution and several positioning systems for the localization of industrial vehicles, mobile robots, and pedestrians. This overview presents each system according to its technology, localization technique, and reported accuracy (mean error).

An overall mean error of 0.81 m is achieved in real-world experiments, with the proposed TC approach. It performs better than the LC approach [21] (with 1.03-m mean error) that uses the same sensors as the proposed solution. Regarding the other positioning systems, the results of the solution presented in this article are in line with systems with similar technologies based on PFs.

## VI. COMPUTATIONAL COMPLEXITY

PFs are known for having demanding computational requirements due to the high sample rate of sensors, which demands that the main operation is executed fast enough so that there is no delay between position estimates provided by the PF. The computational burden of PFs depends mostly on the number of particles, as most operations affect all particles and their states over time. When new heading or displacement measurements are obtained, the sampling algorithm has low complexity of  $O(N)$ , with  $N$  representing the number of particles. In the process to update particles weights, the number of WSs  $N_{ws}$  in the RM, as well as, the number of detected APs  $N_{AP}$  in the latest WS, lead to an asymptotic complexity of  $O(N_{ws} \times N_{AP})$ . A similar complexity is achieved in the process to find the closest RP to each particle, where a comparison is made between the positions of WSs  $N_{ws}$  and particles  $N$ , thus achieving an asymptotic complexity of  $O(N_{ws} \times N)$ . The resampling algorithm, which implements the multinomial resampling approach [29], has a complexity of  $O(N \log_2(N))$ , lower than the complexity in updating particles weights, but also presents asymptotic complexity. The system handles part of the complexity problem by fixing the number of particles being used, and the achieved performance reported is for that specific number of particles. Moreover, the number of particles used can be the same for larger spaces, not increasing the computational cost of the system. This leaves the RM as the only input that can increase the computational cost of the system. For larger spaces, the number of samples in the RM may increase, however, this problem can be minimized by applying a clustering or filtering technique [30] which allows the reduction of RM search in large areas without compromising the accuracy. As reported in [30] there are techniques that effectively reduced the RM size while improving the accuracy in some cases. Adopting clustering or filtering techniques to optimize the RM will be part of future work.

In order to evaluate whether the proposed solution can be deployed in a real factory, the processing time was measured. Using a computer with a Quad-core@3.1-GHz processor and 20 GB of RAM and estimating the pose every 50 ms (IMU sampling period), we observed that all the steps of the process are completed before a new sample of the IMU is available for processing. This means that TrackInFactory can be used in a real scenario.

## VII. CONCLUSION AND FUTURE WORK

In this article, we presented TrackInFactory, a solution for indoor positioning and tracking of industrial vehicles, based on a TC approach for the fusion of data from Wi-Fi, IMU (heading), and encoder (displacement) sensors, to estimate the

position and heading of the vehicle. Wi-Fi is used when updating particles' weights since it provides full coverage of the space where vehicles navigate. The RM is considered when obtaining the similarity between a WS and all RM samples to update the particles' weights. In addition, we proposed a reliability metric that measures the confidence in position estimates, allowing to increase the system's reliability and to dynamically update the weights of particles.

Synthetic data was used while designing the PF, allowing analysis and fine-tuning of different parameters. After defining the PF parameters, the proposed system was evaluated in real-world experiments with an industrial tow tractor.

Achieved results have revealed an improved mean error over similar solutions for vehicle localization and good performance in the last quartile. The best applications of the proposed solution are the continuous monitoring and tracking of the vehicle's position, as well as assisting in autonomous navigation. Even though it can be used to assist in autonomous navigation, in order to support full autonomous navigation, the proposed solution would require sensors to detect obstacles and allow the vehicle to navigate when larger errors exist, mainly during the warm-up period.

Future work will focus on further exploring the confidence metric, to automatically construct or update the RM by saving annotated fingerprints, in order to reduce or even eliminate the deployment effort. Our positioning solution was validated using industrial-grade motion sensors, which are the most expensive part of the hardware used in the positioning module. In the future, we intend to test the proposed solution with low-cost sensors to evaluate if it can be a cost-effective solution.

## REFERENCES

- [1] G. Vasiljević, D. Miklič, I. Draganjac, Z. Kovačić, and P. Lista, "High-accuracy vehicle localization for autonomous warehousing," *Robot. Comput. Integr. Manuf.*, vol. 42, pp. 1–16, Dec. 2016. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0736584515300314>
- [2] J. Fernandez-Madriral, E. Cruz-Martin, J. Gonzalez, C. Galindo, and J. Blanco, "Application of UWB and GPS technologies for vehicle localization in combined indoor–outdoor environments," in *Proc. IEEE 9th Int. Symp. Signal Process. Appl.*, Feb. 2007, pp. 1–4. [Online]. Available: <http://ieeexplore.ieee.org/document/4555416/>
- [3] A. Koch and A. Zell, "RFID-enabled location fingerprinting based on similarity models from probabilistic similarity measures," in *Proc. IEEE Int. Conf. Robot. Autom.*, Jun. 2016, pp. 4557–4563.
- [4] S. Fu, Z.-G. Hou, and G. Yang, "An indoor navigation system for autonomous mobile robot using wireless sensor network," in *Proc. Int. Conf. Netw. Sens. Control*, 2009, pp. 227–232. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4919277>
- [5] P. Bahl and V. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, vol. 2, 2000, pp. 775–784. [Online]. Available: <http://ieeexplore.ieee.org/document/832252/>
- [6] J. Torres-Sospedra *et al.*, "Off-line evaluation of mobile-centric indoor positioning systems: The experiences from the 2017 IPIN competition," *Sensors*, vol. 18, no. 2, p. 487, Feb. 2018. [Online]. Available: <http://www.mdpi.com/1424-8220/18/2/487>
- [7] A. Rácz-Szabó, T. Ruppert, L. Bántay, A. Löcklin, L. Jakab, and J. Abonyi, "Real-time locating system in production management," *Sensors*, vol. 20, no. 23, p. 6766, Nov. 2020. [Online]. Available: <https://www.mdpi.com/1424-8220/20/23/6766>

- [8] R. G. Yudanto and F. Petre, "Sensor fusion for indoor navigation and tracking of automated guided vehicles," in *Proc. IEEE Int. Conf. Indoor Position. Indoor Navig. (IPIN)*, Oct. 2015, pp. 1–8. [Online]. Available: <http://ieeexplore.ieee.org/document/7346941/>
- [9] Z. Wu, M. Wen, G. Peng, X. Tang, and D. Wang, "Magnetic-assisted initialization for infrastructure-free mobile robot localization," in *Proc. 9th IEEE Int. Conf. Cybern. Intell. Syst. (CIS) Robot. Autom. Mechatron. (RAM)*, Nov. 2019, pp. 518–523. [Online]. Available: <http://arxiv.org/abs/1911.09313>.
- [10] A. Motroni, A. Buffi, P. Nepa, and B. Tellini, "Sensor-fusion and tracking method for indoor vehicles with low-density UHF-RFID tags," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–14, 2021, doi: [10.1109/TIM.2020.3027926](https://doi.org/10.1109/TIM.2020.3027926).
- [11] P. Mpeis *et al.*, "The anyplace 4.0 IoT localization architecture," in *Proc. 21st IEEE Int. Conf. Mobile Data Manag. (MDM)*, Jun. 2020, pp. 218–225. [Online]. Available: <https://ieeexplore.ieee.org/document/9162205/>
- [12] X. Ma and T. Liu, "The application of Wi-Fi RTLS in automatic warehouse management system," in *Proc. IEEE Int. Conf. Autom. Logist. (ICAL)*, Aug. 2011, pp. 64–69. [Online]. Available: <http://ieeexplore.ieee.org/document/6024685/>
- [13] J. Biswas and M. Veloso, "Wi-Fi localization and navigation for autonomous indoor mobile robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2010, pp. 4379–4384.
- [14] N. Dinh-Van, F. Nashashibi, N. Thanh-Huong, and E. Castelli, "Indoor intelligent vehicle localization using WiFi received signal strength indicator," in *Proc. IEEE MTT-S Int. Conf. Microw. Intell. Mobility (ICMIM)*, 2017, pp. 33–36.
- [15] Z. Chen, H. Zou, H. Jiang, Q. Zhu, Y. Soh, and L. Xie, "Fusion of WiFi, Smartphone sensors and landmarks using the Kalman filter for indoor localization," *Sensors*, vol. 15, no. 1, pp. 715–732, Jan. 2015. [Online]. Available: <http://arxiv.org/abs/1007.0085>.
- [16] M. B. Alatise and G. P. Hancke, "Pose estimation of a mobile robot based on fusion of IMU data and vision data using an extended Kalman filter," *Sensors*, vol. 17, no. 10, p. 2164, 2017.
- [17] S. Thrun, D. Fox, W. Burgard, and F. Dellaert, "Robust Monte Carlo localization for mobile robots," *Artif. Intell.*, vol. 128, nos. 1–2, pp. 99–141, May 2001. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0004370201000698>
- [18] Z. Wu, E. Jedari, R. Muscedere, and R. Rashidzadeh, "Improved particle filter based on WLAN RSSI fingerprinting and smart sensors for indoor localization," *Comput. Commun.*, vol. 83, pp. 64–71, Jun. 2016. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0140366416300597>
- [19] R. Liu, C. Yuen, T. N. Do, Y. Jiang, X. Liu, and U. X. Tan, "Indoor positioning using similarity-based sequence and dead reckoning without training," in *Proc. IEEE Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Jul. 2017, pp. 1–5.
- [20] R. Liu *et al.*, "Collaborative SLAM based on WiFi fingerprint similarity and motion information," *IEEE Internet Things J.*, vol. 7, no. 3, pp. 1826–1840, Mar. 2020.
- [21] I. Silva, A. Moreira, M. J. Nicolau, and C. Pendão, "Floor plan-free particle filter for indoor positioning of industrial vehicles," in *Proc. ICL-GNSS Int. Conf. Localization GNSS*, Tampere, Finland, 2020, p. 6. [Online]. Available: <http://ceur-ws.org/Vol-2626/paper2.pdf>
- [22] L. H. Chen, E. H. K. Wu, M. H. Jin, and G. H. Chen, "Intelligent fusion of Wi-Fi and inertial sensor-based positioning systems for indoor pedestrian navigation," *IEEE Sensors J.*, vol. 14, no. 11, pp. 4034–4042, Nov. 2014.
- [23] A. Panyov, A. A. Golovan, A. S. Smirnov, and V. V. Kosyanchuk, "Indoor positioning using Wi-Fi fingerprinting, magnetometer and pedestrian dead reckoning," in *Proc. 21st Saint Petersburg Int. Conf. Integr. Navig. Syst. (ICINS)*, 2014, pp. 129–134.
- [24] V. Renaudin, M. H. Afzal, and G. Lachapelle, "Magnetic perturbations detection and heading estimation using magnetometers," *J. Location Services*, vol. 6, no. 3, pp. 161–185, 2012.
- [25] A. Moreira, I. Silva, F. Meneses, M. J. Nicolau, C. Pendão, and J. Torres-Sospedra, "Multiple simultaneous Wi-Fi measurements in fingerprinting indoor positioning," in *Proc. Int. Conf. Indoor Position. Indoor Navig. (IPIN)*, Nov. 2017, pp. 1–8. [Online]. Available: <http://ieeexplore.ieee.org/document/8115914/>
- [26] J. Torres-Sospedra and A. Moreira, "Analysis of sources of large positioning errors in deterministic fingerprinting," *Sensors*, vol. 17, no. 12, pp. 1–48, 2017.
- [27] N. Marques, F. Meneses, and A. Moreira, "Combining similarity functions and majority rules for multi-building, multi-floor, WiFi positioning," in *Proc. Int. Conf. Indoor Position. Indoor Navig. (IPIN)*, 2012, pp. 1–9.
- [28] G. Retscher and J. Joksch, "Comparison of different vector distance measure calculation variants for indoor location fingerprinting," in *Proc. 13th Int. Conf. Location Based Services*, Nov. 2016, pp. 14–16.
- [29] J.-A. Fernández-Madriral and J. L. Blanco Claraco, *Simultaneous Localization and Mapping for Mobile Robots* (Advances in Computational Intelligence and Robotics). London, U.K.: IGI Global, 2013. [Online]. Available: <http://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/978-1-4666-2104-6>
- [30] J. Torres-Sospedra *et al.*, "A comprehensive and reproducible comparison of clustering and optimization rules in Wi-Fi fingerprinting," *IEEE Trans. Mobile Comput.*, early access, Aug. 17, 2020, doi: [10.1109/TMC.2020.3017176](https://doi.org/10.1109/TMC.2020.3017176).



**Ivo Silva** received the M.Sc. degree in telecommunications and informatics engineering from the University of Minho, Guimaraes, Portugal, in 2016, where he is currently pursuing the Ph.D. degree in telecommunications engineering, with a focus on the indoor positioning of industrial vehicles based on Wi-Fi.

He is a Researcher with the Algoritmi Research Centre and an Invited Assistant Professor with the University of Minho. His research interests are in indoor positioning, mobile computing, and smart devices.



**Cristiano Pendão** received the M.Sc. degree in telecommunications and informatics engineering from the University of Minho, Guimaraes, Portugal, in 2012, and the Ph.D. degree in telecommunications/computer science engineering from the Universities of Minho, Aveiro and Porto, Portugal, in 2019.

He is a Professor with the School of Engineering, University of Minho, where he is a Researcher with the Algoritmi Research Centre. His research interests are in positioning and navigation systems, computer vision, and mobile computing.



**Joaquín Torres-Sospedra** received the Ph.D. degree in ensembles of neural networks and machine learning from Universitat Jaume I, Castellón de la Plana, Spain, in 2011.

Since January 2020, he has been the Manager of UBK Geospatial Solutions and Collaborates with the Institute of New Imaging Technologies, Castellón, Spain. He has authored more than 120 articles in journals and conferences. His current research interests include indoor positioning solutions based on Wi-Fi & BLE, machine learning, and evaluation.

Dr. Torres-Sospedra is the Chair of the IPIN International Standards Committee and IPIN Off-Site Competition.



**Adriano Moreira** (Member, IEEE) received the Licenciatura degree in electronics and telecommunications engineering and the Ph.D. degree in electrical engineering from the University of Aveiro, Aveiro, Portugal, in 1989 and 1997, respectively.

He is an Associate Professor, with Habilitation, with the University of Minho, Guimaraes, Portugal, where he is a Researcher with the Algoritmi Research Centre. He co-founded the Computer Communications and Pervasive Media Research Group, and is the Director of the MAP-Tele Doctoral Program in Telecommunications. He participated in many research projects funded by national and EU programs. He has authored several scientific publications in conferences and journals, and one patent in the area of computational geometry. His research activities have been taking place within the ubicomp@uminho research subgroup, which has been focusing in the creation of technologies for smart places.

Dr. Moreira won the First Prize on the Off-Site Track of the EvAAL-ETRI Indoor Localization Competition (IPIN 2015 and 2017) together with his colleagues, and the Second Prize of the corresponding competition in 2016.