

# Floor Plan-free Particle Filter for Indoor Positioning of Industrial Vehicles

Ivo Silva<sup>a</sup>, Adriano Moreira<sup>a</sup>, Maria João Nicolau<sup>a</sup> and Cristiano Pendão<sup>a</sup>

<sup>a</sup>*Algoritmi Research Center, University of Minho, Portugal*

## Abstract

Industry 4.0 is triggering the rapid development of solutions for indoor localization of industrial vehicles in the factories of the future. Either to support indoor navigation or to improve the operations of the factory, the localization of industrial vehicles imposes demanding requirements such as high accuracy, coverage of the entire operating area, low convergence time and high reliability.

Industrial vehicles can be located using Wi-Fi fingerprinting, although with large positioning errors. In addition, these vehicles may be tracked with motion sensors, however an initial position is necessary and these sensors often suffer from cumulative errors (e.g. drift in the heading). To overcome these problems, we propose an indoor positioning system (IPS) based on a particle filter that combines Wi-Fi fingerprinting with data from motion sensors (displacement and heading). Wi-Fi position estimates are obtained using a novel approach, which explores signal strength measurements from multiple Wi-Fi interfaces. This IPS is capable of locating a vehicle prototype without prior knowledge of the starting position and heading, without depending on the building's floor plan. An average positioning error of 0.74 m was achieved in performed tests in a factory-like building.

## Keywords

indoor positioning, particle filter, Wi-Fi fingerprinting, sensor fusion, industrial vehicles

## 1. Introduction

Factories of the future rely on the automation of vehicles to increase productivity in processes such as delivering raw materials and dispatching finished products. Industrial environments are highly dynamic, with new production lines being deployed or reconfigured almost every week. Most factories rely on industrial vehicles such as stackers and tow tractors operated by humans, and some already rely on autonomous vehicles, mobile robots or automated guided vehicles. To optimize the factories logistics operations, as well as to support autonomous navigation, it becomes necessary to localize and monitor these vehicles in real-time as they move. By knowing the position of the vehicles as they transport raw materials to production lines or when they transport finished products from the production lines to the expedition area, it is possible to track these materials along the supply chain.

Wi-Fi fingerprinting has been widely used for indoor positioning applications [1, 2] allowing to cover large buildings without additional infrastructure costs, since it takes advantage of the existing WLAN. This technique comprises two phases, the calibration phase and the online phase. In the calibration phase, Wi-Fi fingerprints (set of signal strength measurements from

---

*ICL-GNSS 2020 WiP Proceedings, June 02-04, 2020, Tampere, Finland*

EMAIL: ivo@dsi.uminho.pt (I. Silva); adriano.moreira@algoritmi.uminho.pt (A. Moreira); joao@dsi.uminho.pt (M.J. Nicolau); cpendao@dsi.uminho.pt (C. Pendão)

ORCID: 0000-0002-5423-0624 (I. Silva); 0000-0002-8967-118X (A. Moreira); 0000-0001-6172-5855 (M.J. Nicolau); 0000-0002-4563-7414 (C. Pendão)

© 2020 Copyright for this paper by its authors.

Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

visible Access Points (APs)) are obtained in known positions (reference points) which compose the building's radio map. In the online phase, a Wi-Fi fingerprint is obtained, then it is compared against the radio map using a similarity function (e.g. Euclidean or Manhattan distance). Finally, a position is obtained using for example the k-Nearest-Neighbor (kNN) algorithm. Due to the nature of Wi-Fi signals, fingerprinting is characterized by large maximum errors and achieves mean errors between 2 m and 4 m [3]. The main disadvantage of this technique is related with the construction and maintenance of the radio map, which are time demanding tasks that take a significant effort to accomplish. The larger the building, the longer it takes to properly construct the radio map. Since the radio map tends to degrade over time, recalibration is required by collecting new Wi-Fi fingerprints. To overcome this issue, some solutions explore collaborative ways of updating the radio map [4].

The positioning and tracking of industrial vehicles have demanding requirements, namely, a fast convergence time (the time it takes to find the absolute position of the vehicle with an acceptable positioning error), a low maximum error, and coverage of the entire operating area. In this paper, we address the indoor positioning of industrial vehicles, including autonomous mobile robots, manned or even hybrid vehicles (can be both autonomously and manually operated). To solve this problem, we propose a particle filter (PF) solution for the fusion of motion data with Wi-Fi fingerprinting. It takes advantage of the Wi-Fi technology that provides coverage of the entire operating area, which makes it affordable to deploy in large industrial scenarios. In addition, vehicles are tracked using motion sensors, namely, an inertial measurement unit (IMU) (providing heading) and a rotary encoder (providing displacement) to improve accuracy. In our solution, no assumptions are made regarding the starting position/heading, i.e. there is no initial information regarding the position and heading of the vehicle. The absolute heading provided from the IMU sensor could be used, however it can be affected by magnetic perturbations from large machinery in industrial environments. Since the proposed solution will be deployed in an industrial scenario it should be capable of estimating the position as well as the heading.

The main contributions of this paper are threefold. First, the proposed solution achieves better performance than traditional Wi-Fi fingerprinting for indoor tracking. Second, indoor positioning and tracking of industrial vehicles is achieved without any assumption regarding the starting position/heading. Third, in contrast with most particle filters, which take advantage of the floor plan to remove particles that have hit walls or obstacles, the proposed solution does not depend on the building's floor plan.

## 2. Related Work

Different technologies may be used for localization and tracking in indoor environments, such as, Wi-Fi [5], Wireless Sensor Networks [6] and IEEE 802.15.4a [7]. Usually, filtering techniques such as the Kalman filter or particle filter (PF) are used to perform the fusion of sensor data for indoor positioning [8, 5, 9, 10]. Variations of these filtering techniques are often explored for indoor positioning because, when properly configured, they are capable of estimating noise and achieving an accurate estimated position. Kalman filters are suitable for linear problems where the noise tends to be Gaussian. Extended and Unscented Kalman filters can be used in non-linear problems. In addition, PFs are commonly used to perform sensor fusion in non-linear and non-Gaussian problems.

The ubiquity of WLAN networks has led many researchers to explore Wi-Fi for indoor

positioning and tracking. In [11], several positioning solutions are compared against each other in an off-site competition, in which teams implement their own algorithms using the same training and evaluation data, collected with a smartphone. The solution with the best results uses Wi-Fi fingerprinting based on the kNN algorithm, achieving a mean error of 3 m.

Improved positioning performance is achieved when Wi-Fi is combined with other sensors. In [5], odometry is combined with Wi-Fi fingerprinting and floor plan constraints in order to estimate the position of autonomous robots. In the real-time phase, the robot's location is estimated using a Monte Carlo algorithm (particle filter) with Bayesian filtering [8]. A mean location error of 1.2 m was achieved in an experiment where the robot covered a trajectory 818 m long.

Zou et al. [12], proposed a system based on pedestrian dead reckoning (PDR). Two approaches may be used to provide position estimates: one that uses a PF to fuse Wi-Fi fingerprinting with PDR, and another based on a PF that uses Wi-Fi fingerprinting with PDR but also adds iBeacon technology to improve position estimates. The proposed solution achieves a mean error of 1.48 m and 0.60 m in performed tests using the former and the latter approaches, respectively.

Liu et al. [13] proposed a network-based indoor tracking system, which uses a PF to combine inertial data with physical layer channel state information (CSI) from Wi-Fi signals. A mobile device (smartphone) broadcasts Wi-Fi packets (with IMU timestamped data) to anchor nodes that are distributed through the indoor environment. Anchor nodes send the packets to a central server that extracts the CSI information from WiFi cards in anchor nodes, and finally track the target with these two pieces of information. A mean error of 1.30 m was achieved in experiments carried out in a building with 16x18 m.

In [14], machine learning is used to merge Wi-Fi fingerprinting with PDR data. Initial position and heading are estimated with the proposed algorithm that takes advantage of PDR data to mitigate the weakness of Wi-Fi-based positioning. Performed experiments in a large building (167x27 m) have revealed a mean error of 2.71 m.

The main drawbacks of the above-mentioned solutions are that some use PDR which is not applicable to vehicles because the movement models of vehicles and pedestrians are different, and some solutions require additional infra-structure (iBeacons or anchor nodes) which represent additional deployment and maintenance costs.

### 3. Proposed Solution

Particle filters are a type of Monte Carlo algorithms suitable for solving estimation problems in non-linear non-Gaussian scenarios. In comparison to the Kalman filter, PFs have the advantage of not relying on any local linearization technique or any crude functional approximation. The disadvantage of PFs is that they are computationally expensive. However, with the increasing computational capabilities, it is now possible to have real-time applications for indoor positioning.

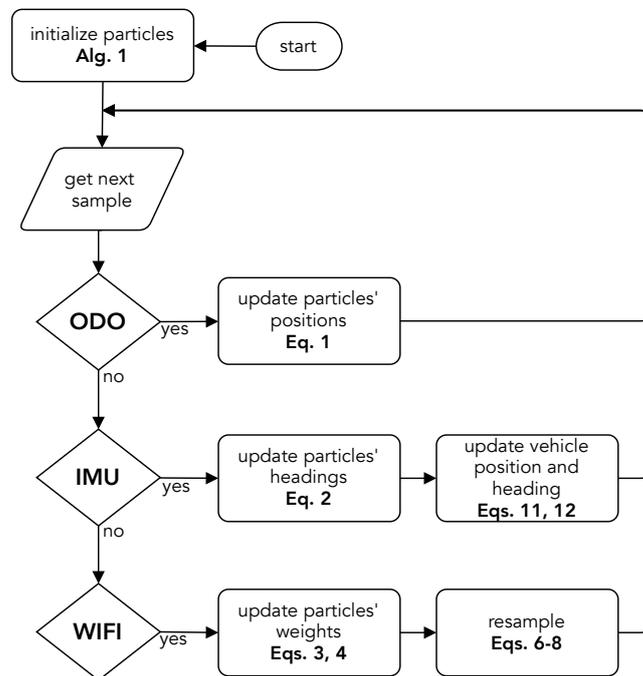
A particle filter uses many particles with different weights which represent the probability of the particle being in the real position. Initially, particles are dispersed throughout the space and, as the vehicle starts moving, particles also move and start converging to an area where it is more likely to be the vehicle's true position. Updating particles' positions and weights as the vehicle moves around allows to remove particles from places where it is unlikely or impossible for the vehicle to be at. The removal of particles is performed in a process called resampling.

In this process, particles with lower weights are removed and particles with higher weights are copied to create new particles. The weighted average of the particles' positions and headings represents the estimated position and heading of the vehicle.

### 3.1. Top level algorithm

As introduced previously, the particle filter depends on three processes: sampling (when data from sensors is received); updating particles' weights; and resampling. The flowchart, in Figure 1, depicts the top level algorithm of the proposed PF.

A particle is defined as  $p = (w, x, y, h, ho)$ , where  $w$  denotes the weight of the particle,  $(x, y)$  represent the position coordinates of the particle in a Euclidean space (since we are considering one single floor the  $z$  coordinate value remains fixed, hence it is not used),  $h$  represents the heading of the particle and  $ho$  represents the heading offset of the particle, which is used because the initial heading is unknown.



**Figure 1:** Particle filter top level algorithm.

After particles have been initialized (see Section 3.2), the algorithm enters a loop that waits for data coming from sensors. In this loop:

- Every time a new odometer sample is received, the positions of the particles are updated;
- Particles' headings are updated when a new IMU sample is obtained;
- Particles' weights are updated when a new Wi-Fi sample is obtained, then they are subject to resampling;
- Finally, an estimated position and heading is obtained by combining the particles' positions and headings. This process is performed after updating the particles' headings,

defining the PF sampling rate which is a reasonable update interval (20 Hz) that does not compromise computing performance.

As an alternative to using the building’s floor plan to remove particles that have hit walls or obstacles, we propose a different approach where the reference points used in Wi-Fi fingerprinting are used to define areas where vehicles can navigate, therefore only a limited set of points is necessary to define these areas. If particles move into areas where no reference points exist, they must be removed. Navigable areas are defined by a set of reference points, where each point covers the area of a circle with radius  $r$ .

### 3.2. Particles initialization

Usually, particles are uniformly distributed throughout the navigable space when they are initialized. This approach leads to large errors in initial iterations, as particles take some time before converging into a small area. We explore a different approach, where particles are created near reference points where it is more likely for the vehicle to be near. Algorithm 1 describes how particles are initialized.

Algorithm 1: Initialize particles.

```

1: procedure INITIALIZE PARTICLES( $WiFi_n, M$ )
2:    $c$ =centroid of first  $WiFi_n$  Wi-Fi position estimates
3:    $RPs$ =list of ref. points within a  $r_{ini}$  radius of  $c$ 
4:    $np = M/\#(RPs)$ 
5:   for  $rp$  in  $RPs$  do
6:     for  $i = 1$  until  $np$  do
7:        $w = 1/M$ 
8:        $x = rp.x + rand(0, r)$ 
9:        $y = rp.y + rand(0, r)$ 
10:       $h = 0$ 
11:       $ho = rand(-\pi, \pi)$ 
12:       $p = (w, x, y, h, ho)$ 
13:       $P = P \cup \{p\}$  set of all particles

```

In our experiments, the total number of particles ( $M$ ) considered is 3000, a distance of  $\sqrt{2}$  m and 4 m were defined as the  $r$  and  $r_{ini}$  values, respectively. The first value was chosen because the reference points used to create the radio map are distributed in a grid with 1 m between points. The second value (4 m) represents the typical mean error of a Wi-Fi fingerprinting system. The number of Wi-Fi position estimates obtained before the initialization was defined as  $WiFi_n=3$ . Each Wi-Fi position estimate was obtained using kNN with  $k=5$ .

### 3.3. Update particles’ positions

Particles need to be moved to a new position with each new odometer sample. The position of each particle is updated using a motion model, where the displacement is obtained from the latest odometer sample, and using its current heading value:

$$\begin{aligned}
 x &= x_{-1} - (l + n_l) * \sin(h) \\
 y &= y_{-1} + (l + n_l) * \cos(h)
 \end{aligned} \tag{1}$$

where  $x_{-1}$  and  $y_{-1}$  represent the previous position coordinates of the particle,  $l$  represents the displacement and  $n_l$  represents zero-mean Gaussian noise added to the displacement. In conducted tests,  $n_l$  is defined by  $N(0, 0.004 \text{ m})$ .

### 3.4. Update particles' headings

Once a new heading sample is received, particles' headings are updated. A particle's heading includes the heading obtained from the IMU sensor, a random value and the random heading offset (which was generated when the particle was created):

$$h = \theta + n_\theta + ho \quad (2)$$

where  $\theta$  represents the latest heading sample from the IMU sensor,  $n_\theta$  represents zero-mean Gaussian noise and  $ho$  is the heading offset of the particle. In our experiments,  $n_\theta$  is characterized by  $N(0, 0.05^\circ)$ .

### 3.5. Update particles' weights

Particles' weights are updated based on Wi-Fi position estimates. Whenever a particle is further than  $r$  meters of any reference point within the indoor space, its weight is set to zero, because it means that the particle has moved into an area where the vehicle cannot navigate through. The  $r$  parameter depends on the distance between adjacent reference points.

The current weight of a particle is updated based on the distance between the particle and the latest Wi-Fi position estimate. The larger the distance between the particle and the latest Wi-Fi position estimate, the lower its weight will be. The final weight of each particle depends on the current weight value and on its previous value:

$$w = (1 - \alpha) * w_{-1} + \alpha * (1 - d_n) \quad (3)$$

where  $w_{-1}$  represents the previous particle weight,  $d_n$  is the normalized distance between the particle and the latest Wi-Fi position estimate, and  $\alpha$  is a value between 0 and 1 which represents how much the current weight contributes to the particle weight, conversely,  $(1 - \alpha)$  represents how much the previous weight contributes to the particle's weight. The normalized distance,  $d_n$ , is given by:

$$d_n = \frac{d - \min(D)}{\max(D) - \min(D)} \quad (4)$$

where  $d$  represents the distance between the particle and the latest Wi-Fi position estimate, and  $D$  represents the set of distances between all particles and the latest Wi-Fi position estimate.

### Warm-up

Once particles are initialized, the PF starts estimating the vehicle's position. During the first iterations, the confidence on the vehicle's estimated position is low because the initial heading is unknown and because the particles take some time to converge into a concentrated cluster. Therefore, during the first iterations, the importance of Wi-Fi position estimates should be higher when updating the particles' weights. After this period, Wi-Fi position estimates can be given a lower importance in updating the particles' weights. This can be done by adjusting

the value of alpha in equation 3 over time, starting with alpha equal to one and reducing its value during the warm-up period up to a minimum value ( $\alpha_{min}$ ), as follows:

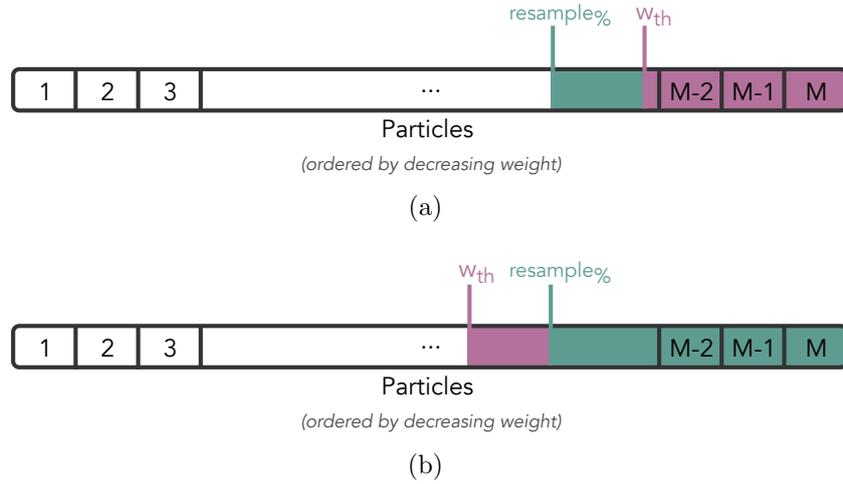
$$\alpha(t) = \begin{cases} (1 - \alpha_{min}) * (e^{-t/35}) + \alpha_{min} & , t < t_{warm-up} \\ \alpha_{min} & , t \geq t_{warm-up} \end{cases} \quad (5)$$

where  $t$  denotes the time since particles were created,  $\alpha_{min}$  denotes the value of  $\alpha$  after warming-up, and  $t_{warm-up}$  defines the warm-up time. We have set  $\alpha_{min} = 0.05$  and  $t_{warm-up} = 180$  s.

### 3.6. Resampling

Resampling particles has the purpose of removing particles with lower weights and replace them with new ones in order to allow a faster convergence and to maintain the particle diversity. Two parameters are used to define how particles are removed, the resample percentage and the weight threshold. The first ( $resample\%$ ) defines a percentage of particles with lower weights that are removed. The second ( $w_{th}$ ) defines a threshold where particles with weights lower than it are removed.

In the resampling process, two distinct cases (depicted in Figure 2) may occur depending on the  $resample\%$ , the  $w_{th}$  and the particles' weights. Figure 2 (a) shows the case in which percentage of samples removed from the  $w_{th}$  criteria is lower than  $resample\%$ , while Figure 2 (b) illustrates the opposite case.



**Figure 2:** Resampling scenarios.

In both scenarios, the particles removed due to the  $resample\%$  (represented in green in Figure 2) are replaced by new particles that are created near the latest Wi-Fi position estimate, following a similar process as the one described in Section 3.2. The weight of new particles near the Wi-Fi position estimate is defined by:

$$w = \begin{cases} w_{min} & , d_{vw} > d_{max} \\ f(d_{vw}) & , d_{vw} \leq d_{max} \end{cases} \quad (6)$$

where  $d_{vw}$  represents the distance between the vehicle's latest position estimate ( $\rho_{vehicle}$ ) and the Wi-Fi position estimate ( $\rho_{wifi}$ ),  $d_{max}$  defines the maximum allowed distance between  $\rho_{vehicle}$  and  $\rho_{wifi}$ , and,

$$f(d_{vw}) = \frac{d_{max} - d_{vw}}{d_{max}} \times (w_{max} - w_{min}) + w_{min} \quad (7)$$

where the distance  $d_{vw}$  is converted into a weight value between  $w_{min}$  and  $w_{max}$ , the lower the distance the higher the weight value. This gives more emphasis to  $\rho_{wifi}$  estimates when they are closer to  $\rho_{vehicle}$  and have a lower impact when  $\rho_{wifi}$  is further away from  $\rho_{vehicle}$ , which occurs when  $\rho_{wifi}$  is an outlier.

IMU sensors are susceptible to magnetic perturbations which affect magnetometer readings, consequently affecting the estimated heading, which usually leads to drift in the heading. Drift can be minimized by assigning different heading offsets to resampled particles because there are particles that follow the real trajectory whilst others will move away from the trajectory or move into non-navigable areas. Particles that follow the real trajectory are the ones that hold the most likely heading offset, therefore they will end-up having higher weights. To minimize drift in the heading and improve heading estimation, resampled particles created around  $\rho_{wifi}$  have a new heading offset, defined as:

$$ho = ho' + n_{ho} \quad (8)$$

where  $ho'$  is the heading offset of a randomly selected particle (from the subset of particles with weight higher than  $w_{th}$ ) and  $n_{ho}$  represents a zero-mean Gaussian distributed random angle. In our experiments,  $n_{ho}$  is defined by  $N(0, 8^\circ)$ .

When the  $resample\%$  of particles is lower than the number of removed particles due to the  $w_{th}$  (Figure 2 (b)), it is necessary to create new particles (represented in purple in Figure 2 (b)) to maintain the total number of particles constant. These new particles are created from the sub-set of particles that have weights higher than  $w_{th}$ , which are randomly selected and copied.

We have set the following parameters in the resampling process:  $d_{max} = 4$  m,  $w_{min} = 0.05$ ,  $w_{max} = 0.5$  and  $resample\% = 0.1$ .

### 3.7. Wi-Fi position estimation

We have devised a new technique for improving positioning performance with Wi-Fi fingerprinting. In this technique, described in [2], fingerprints from multiple Wi-Fi interfaces are merged into one Wi-Fi sample which is then compared with the samples from the radio map. Since received signal strength values from distinct interfaces are not correlated (see [2]), we can average them into a unique sample, which results in lower mean and maximum errors.

Let us assume that  $N$  interfaces are used and that all of them collect a sample at the same time instant. We define a Wi-Fi sample  $s_t^i$  as the list of RSSI (Received Signal Strength Indicator) values measured by interface  $i$ , from each of the  $R$  access points, in time instant  $t$ :

$$s_t^i = (RSSI_1^i, \dots, RSSI_R^i) \quad (9)$$

A combined sample can be obtained by computing the average of the RSSI values from each interface:

$$s_t = (\widehat{RSSI}_1, \dots, \widehat{RSSI}_R) \quad (10)$$

where

$$\widehat{RSSI}_j = \frac{(RSSI_j^1 + \dots + RSSI_j^N)}{N} \quad (11)$$

The calibration phase comprises the collection of several combined Wi-Fi samples at known reference points spread through the navigable space. In the real-time phase, the similarity of a Wi-Fi sample is computed for all samples of the radio map. Manhattan similarity was chosen as the similarity function, defined as:

$$sim = \sum_{i=1}^R |RSSI_i^{rm} - RSSI_i^t| \quad (12)$$

where  $RSSI_i$  refers to the signal strength of AP  $i$  from the radio map sample ( $rm$ ) and the online test sample ( $t$ ).

The most similar radio map Wi-Fi samples are the ones used to find a Wi-Fi position estimate. We use the kNN algorithm to find the position estimate based on the  $k$  most similar radio map Wi-Fi samples:

$$\rho_{wifi}(x, y) = \frac{\sum_{i=1}^k s_i(x, y)}{k} \quad (13)$$

where  $\rho_{wifi}(x, y)$  represents the Wi-Fi position estimate and  $s$  represents each of the  $k$  most similar Wi-Fi radio map samples. In our experiments we set  $k = 5$ .

### 3.8. Particle filter position estimation

The weighted average of particles' positions represents the vehicle's estimated position, defined as:

$$\rho_{vehicle}(x, y) = \frac{\sum_{i=1}^M p_i(x, y) \times p_i.w}{\sum_{i=1}^M p_i.w} \quad (14)$$

where  $\rho_{vehicle}(x, y)$  represents the vehicle's position,  $p_i(x, y)$  and  $p_i.w$  refer to the position and weight of the  $i$ th particle.

Similarly, we obtain the estimated heading of the vehicle:

$$h_{vehicle} = \tan^{-1} \left( \frac{\sum_{i=1}^M \sin(p_i.h) \times p_i.w}{\sum_{i=1}^M \cos(p_i.h) \times p_i.w} \right) \quad (15)$$

where  $h_{vehicle}$  represents vehicle's heading,  $p_i.h$  and  $p_i.w$  refer to the heading and weight of the  $i$ th particle.

## 4. Experiments

We conducted experiments using a prototype of a vehicle that we refer to as mobile unit. Before starting any experiments it was necessary to map and affix tags to the floor in the testing scenario so that we could create the radio map and obtain ground truth information when running experiments.

### 4.1. Testing Scenario

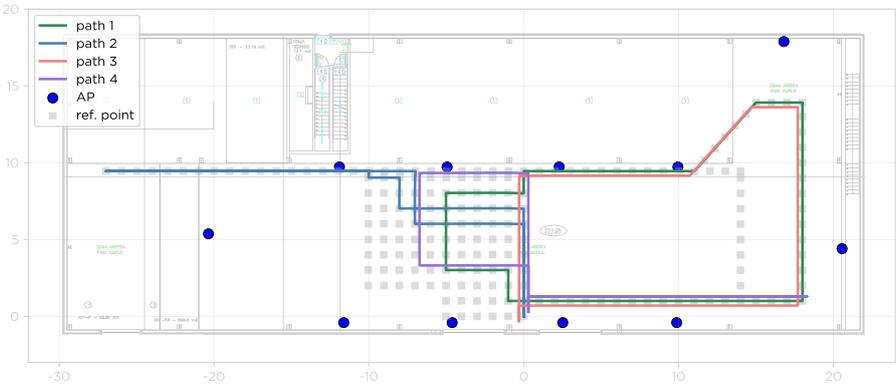
The PIEP building, shown in Figure 3, is located at the University of Minho and measures 50 by 20 m and is more than 8 m high. In many aspects, it is similar to a factory plant with large

machinery, plenty of metal structures and tools, and some large open spaces, therefore it is an ideal scenario to conduct experiments.



**Figure 3:** PIEP building at University of Minho.

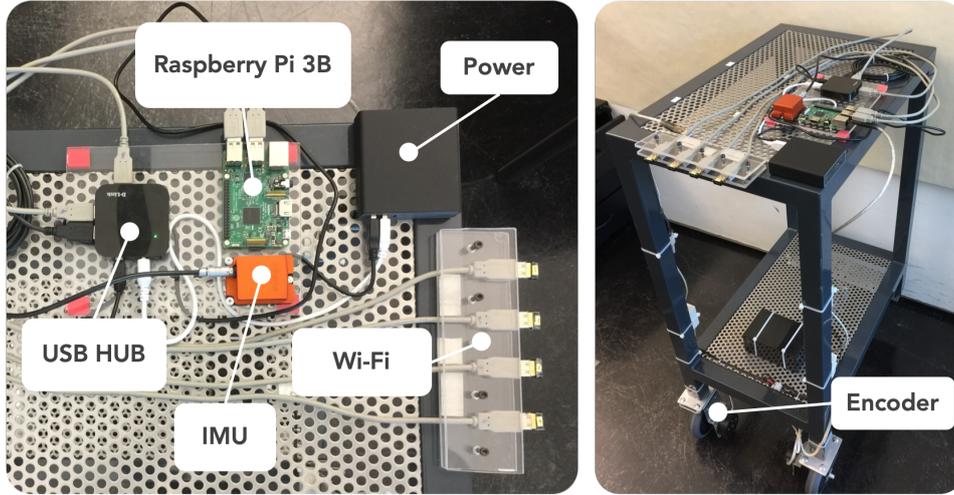
Eleven Wi-Fi APs operating in the 2.4 GHz frequency band, represented by the blue circles in Figure 4, are present in the building. The set of reference points considered in the Wi-Fi radio map are represented by the grey squares in Figure 4. The distance between adjacent reference points is one meter, in most of the cases.



**Figure 4:** Test trajectories at the PIEP building.

Before performing tests, it was necessary to create a radio map by collecting 20 samples at each reference point (around 4000 total samples). Each one of the samples includes merged RSS values from five fingerprints, each one of them collected through a different Wi-Fi interface.

We collected test data in four distinct trajectories, depicted in Figure 4. The duration of each trajectory was 10, 6, 6 and 4 minutes respectively, and the travelled distance over all



**Figure 5:** Mobile unit.

trajectories sums up to almost 300 m. Data was collected by moving the mobile unit over the trajectories at normal pedestrian speed (around 1 m/s). Ground truth data was manually collected, with the help of a video camera pointed towards the floor where it recorded the reference tags as the mobile unit moves over them.

#### 4.2. Mobile Unit

We have equipped a trolley with several sensors. It can be easily pushed to emulate the movement of a vehicle in indoor environments. As shown in Figure 5, the mobile unit is equipped with a Raspberry Pi Model 3B with integrated Wi-Fi interface as well as the following sensors: four external Wi-Fi interfaces (Edimax EW-7811un) connected through a USB HUB; an IMU (Xsens MTi-300 AHRS) to measure the heading; and an absolute encoder (US Digital A2) to measure the displacement.

### 5. Results

The euclidean distance between the position estimate and the ground truth was used as the positioning error metric. The PF solution was executed with data from each testing trajectory five times and obtained similar results.

Table 1 shows the PF positioning results of each trajectory which include position estimates of all runs. The last two columns of the table (“All” and “After warm-up”) consider all position estimates of all runs. For comparison, Table 2 shows Wi-Fi fingerprinting results, obtained with  $k = 5$ .

The proposed PF is capable of minimizing the main problems of Wi-Fi fingerprinting, reducing overall the maximum error (from 18.12 m to 9.48 m) and achieving a great improvement in the mean error (from 3.68 m to 1.47 m). The maximum error with the PF is still considerable due to the warm-up period, in which particles are still converging.

**Table 1**

Positioning results (in meters) of trajectories with mobile unit.

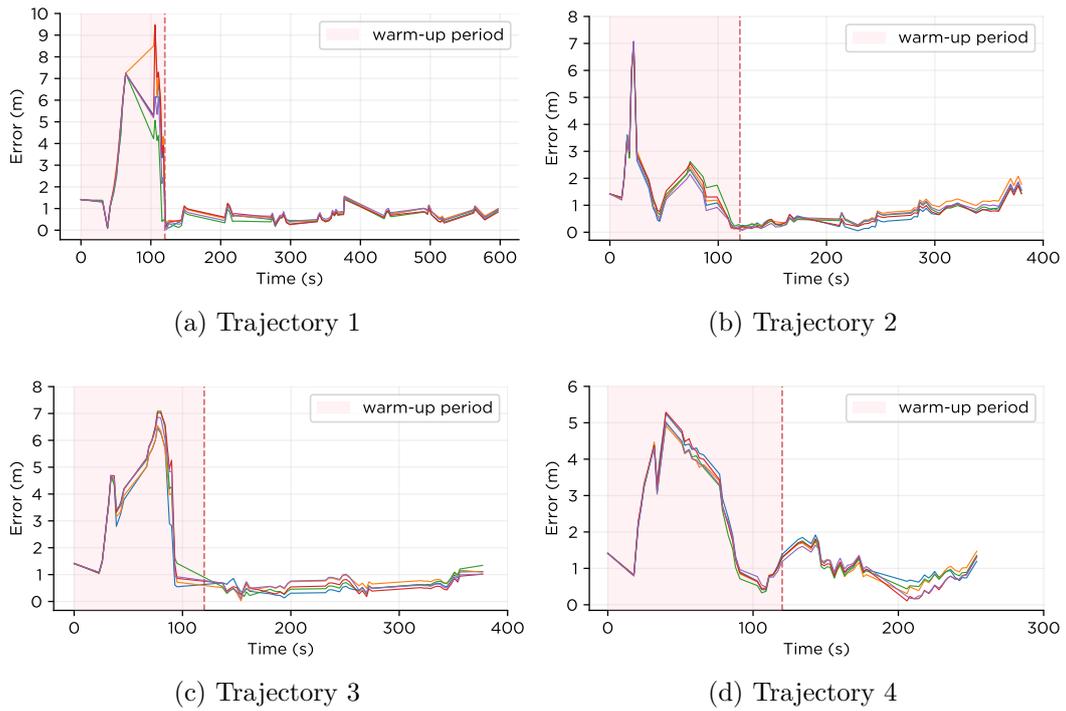
	Trajectory (all)				All	After warm-up
	1	2	3	4		
mean	1.42	1.10	1.76	1.67	1.47	0.74
median	0.77	0.83	0.74	1.08	0.85	0.70
P75th	1.10	1.32	2.81	2.20	1.50	0.94
P99th	6.08	2.91	6.04	4.37	5.27	1.54
max	9.48	7.08	7.09	5.30	9.48	2.08

**Table 2**

Positioning results (in meters) with Wi-Fi fingerprinting.

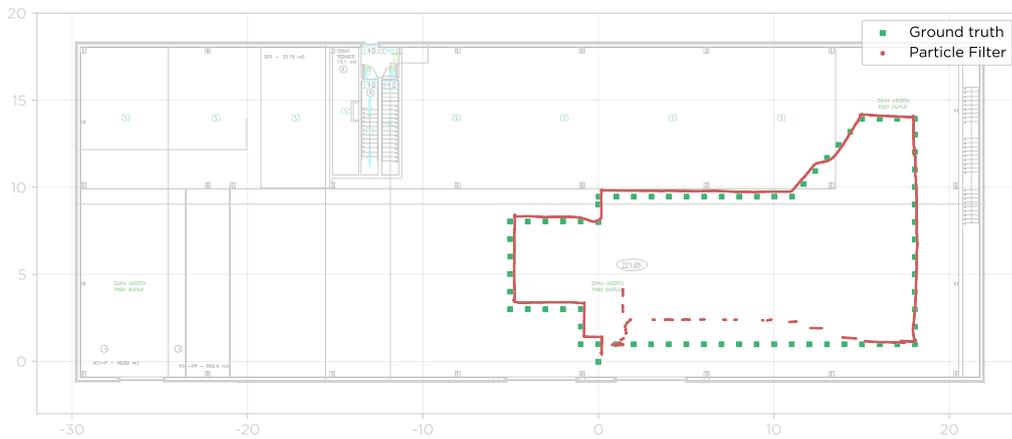
	Trajectory				All
	1	2	3	4	
mean	3.48	3.83	4.27	3.20	3.68
median	2.47	3.95	3.60	3.04	3.16
P75th	4.84	4.81	5.98	4.10	4.82
P99th	13.02	10.69	13.06	9.60	13.00
max	13.25	18.12	13.25	10.58	18.12

Regarding the results achieved with the PF, a significant improvement is observed (Table 1) in the overall mean error after the warm-up time. The mean error is reduced from 1.47 m to 0.74 m because larger positioning errors, mostly observed during the warm-up period, are not considered. This can be observed in Figure 6, that depicts the error variation over time in each trajectory (includes all five runs). As expected, larger positioning errors are observed during the initial phase when the PF is still converging. Once the PF converges, the positioning error remains low, suggesting that better overall results would be achieved in longer experiments. Industrial vehicles are expected to be in operation during long periods, therefore this last metric (after warm-up) is the one that better represents the expected accuracy of our solution in a real environment.



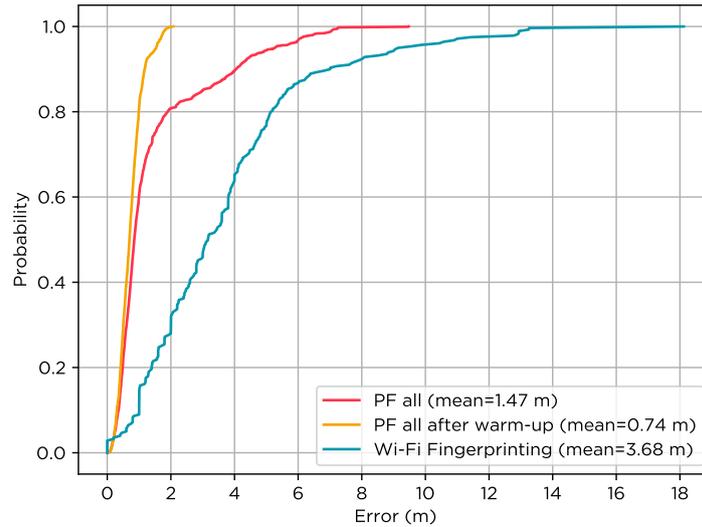
**Figure 6:** Error in meters of each trajectory.

The PF estimated positions of trajectory 1 are depicted in Figure 7. As can be seen, initially there is a higher error during the warm-up period, then the PF converges and the estimated path follows the real trajectory closely.



**Figure 7:** Estimated path of trajectory 1.

Figure 8, shows the CDF of the positioning error. The overall CDF curve of the PF has a much higher maximum error (over 9 m) when comparing to the PF CDF after warm-up where the maximum error is approximately 2 m.



**Figure 8:** Overall CDF of proposed solution and Wi-Fi fingerprinting.

In the performed tests, we have considered that the initial heading is unknown, because magnetic perturbations, if present, might affect the absolute heading estimated by the IMU sensor. However, in scenarios where magnetic perturbations are not a concern, there are applications where the initial heading is provided and reliable, for instance, electric industrial vehicles usually charge in a docking station where they maintain their heading, hence knowing the initial heading can speed-up the convergence time of the PF. In this context, we decided to run the previous experiments with a known initial heading and achieved an overall mean error of 0.85 m and 0.80 m (after warm-up). The main difference in these results is that the larger errors are minimized, reducing the mean and maximum errors. When the initial heading is known, the maximum error of trajectories 1, 2 3, and 4 is 1.58 m, 4.39 m, 1.42 m, and 1.87 m, respectively.

Table 3 shows the proposed solution compared with similar indoor positioning systems. The best performance is achieved with the proposed PF (after warm-up). The solutions in [5] and [13] achieve 1.20 m and 1.30 m mean error, followed by the proposed PF solution (when considering also the warm-up period). A mean error of 2.71 m is achieved with the positioning system [14], which uses PDR with Wi-Fi fingerprinting, hence, it cannot be applied to vehicles. Finally, our fingerprinting implementation has achieved slightly worse results than the UMinho Team, the winning team of Track 3 (smartphone-based off-site) IPIN 2017 Competition.

**Table 3**

Comparison between proposed solution and similar indoor positioning systems.

<b>Solution</b>	<b>Sensor(s)/ Technology</b>	<b>Pos. Technique</b>	<b>Mean Error</b>
Proposed solution PF (after warm-up)	WiFi + motion sensors	Particle filter	0.74 m
[5]	WiFi + encoder	Fingerprinting + Odometry	1.20 m
FW-PF [13]	WiFi + IMU	Particle Filter	1.30 m
Proposed solution PF	WiFi + motion sensors	Particle filter	1.46 m
[14]	WiFi + PDR	Fingerprinting + Machine Learning	2.71 m
UMinho Team [11]	WiFi	Fingerprinting	3.00 m
Proposed solution (fingerprinting)	WiFi	Fingerprinting	3.68 m

## 6. Conclusions

We have presented a solution for the indoor positioning and tracking of industrial vehicles, consisting of a particle filter that performs the sensory fusion of Wi-Fi and motion sensors' data without depending on the floor plan of the building. In addition, the proposed solution is capable of estimating the position and heading without knowledge of the initial position and heading. Wi-Fi fingerprinting takes advantage of existing WLAN infra-structure and allows to obtain an absolute position which is used to provide an initial position and to update particles' weights whenever a new Wi-Fi sample is obtained. Motion sensors (IMU and encoder) allow an accurate tracking of the vehicle during the navigation through the industrial space. We conducted experiments in a factory-like environment where we collected test data with a vehicle-like prototype. An overall mean error of 0.74 m and maximum error of 2 m are achieved when considering that the particle filter has already converged. The performance achieved by our solution makes it suitable for the localization and tracking of vehicles in industrial environments, allowing to improve day-to-day tasks performed by these vehicles. It is also appropriate for supporting indoor navigation, however it does not have sufficient accuracy for tasks such as docking, which require higher accuracy. In the future, we intend to develop a solution to improve the particle filter convergence time and to conduct further experiments to validate this solution using real industrial vehicles.

## Acknowledgments

This work has been supported by FCT – Fundação para a Ciência e Tecnologia within the R&D Units Project Scope: UIDB/00319/2020, the PhD fellowship PD/BD/137401/2018 and the Technological Development in the scope of the projects in co-promotion n<sup>o</sup> 002814/2015 (iFACTORY 2015-2018).

## References

- [1] P. Bahl, V. Padmanabhan, RADAR: an in-building RF-based user location and tracking system, in: Proc. IEEE INFOCOM 2000. Conf. Comput. Commun. Ninet. Annu. Jt. Conf. IEEE Comput. Commun. Soc. (Cat. No.00CH37064), volume 2, 2000, pp. 775–784. URL: <http://ieeexplore.ieee.org/document/832252/>. doi:10.1109/INFCOM.2000.832252. arXiv:1106.0222.
- [2] A. Moreira, I. Silva, F. Meneses, M. J. Nicolau, C. Pendao, J. Torres-Sospedra, Multiple simultaneous Wi-Fi measurements in fingerprinting indoor positioning, in: 2017 Int. Conf. Indoor Position. Indoor Navig., IEEE, Sapporo, Japan, 2017, pp. 1–8. URL: <http://ieeexplore.ieee.org/document/8115914/>. doi:10.1109/IPIN.2017.8115914.
- [3] J. Torres-Sospedra, A. Jiménez, S. Knauth, A. Moreira, Y. Beer, T. Fetzter, V.-C. Ta, R. Montoliu, F. Seco, G. Mendoza-Silva, O. Belmonte, A. Koukofikis, M. Nicolau, A. Costa, F. Meneses, F. Ebner, F. Deinzer, D. Vaufreydaz, T.-K. Dao, E. Castelli, The Smartphone-Based Offline Indoor Location Competition at IPIN 2016: Analysis and Future Work, *Sensors* 17 (2017) 557. URL: <http://www.mdpi.com/1424-8220/17/3/557>. doi:10.3390/s17030557.
- [4] D. Matos, A. Moreira, F. Meneses, Wi-Fi fingerprint similarity in collaborative radio maps for indoor positioning, in: Inforum 2014, 2014, pp. 184–194.
- [5] J. Biswas, M. Veloso, Wi-Fi localization and navigation for autonomous indoor mobile robots, in: Proc. - IEEE Int. Conf. Robot. Autom., 2010, pp. 4379–4384. doi:10.1109/ROBOT.2010.5509842.
- [6] S. Fu, Z.-g. Hou, G. Yang, An indoor navigation system for autonomous mobile robot using wireless sensor network, in: 2009 Int. Conf. Networking, Sens. Control, 2009, pp. 227–232. URL: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4919277>. doi:10.1109/ICNSC.2009.4919277.
- [7] C. Rohrig, D. Hess, C. Kirsch, F. Kunemund, Localization of an omnidirectional transport robot using IEEE 802.15.4a ranging and laser range finder, in: 2010 IEEE/RSJ Int. Conf. Intell. Robot. Syst., IEEE, 2010, pp. 3798–3803. URL: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5651981><http://ieeexplore.ieee.org/xpls/abs/all.jsp?arnumber=5651981><http://ieeexplore.ieee.org/document/5651981/>. doi:10.1109/IRoS.2010.5651981.
- [8] S. Thrun, D. Fox, W. Burgard, F. Dellaert, Robust Monte Carlo localization for mobile robots, *Artif. Intell.* 128 (2001) 99–141. URL: <https://linkinghub.elsevier.com/retrieve/pii/S0004370201000698>. doi:10.1016/S0004-3702(01)00069-8.
- [9] Z. Chen, H. Zou, H. Jiang, Q. Zhu, Y. Soh, L. Xie, Fusion of WiFi, Smartphone Sensors and Landmarks Using the Kalman Filter for Indoor Localization, *Sensors* 15 (2015) 715–732. URL: <http://arxiv.org/abs/1007.0085><http://www.mdpi.com/1424-8220/15/1/715>. doi:10.3390/s150100715. arXiv:1007.0085.

- [10] E. Ivanjko, I. Petrović, Extended Kalman filter based mobile robot pose tracking using occupancy grid maps, Proc. Mediterr. Electrotech. Conf. - MELECON 1 (2004) 311–314. doi:10.1109/melcon.2004.1346851.
- [11] J. Torres-Sospedra, A. Jiménez, A. Moreira, T. Lungenstrass, W.-C. Lu, S. Knauth, G. Mendoza-Silva, F. Seco, A. Pérez-Navarro, M. Nicolau, A. Costa, F. Meneses, J. Farina, J. Morales, W.-C. Lu, H.-T. Cheng, S.-S. Yang, S.-H. Fang, Y.-R. Chien, Y. Tsao, Off-Line Evaluation of Mobile-Centric Indoor Positioning Systems: The Experiences from the 2017 IPIN Competition, Sensors 18 (2018) 487. URL: <http://www.mdpi.com/1424-8220/18/2/487>. doi:10.3390/s18020487.
- [12] H. Zou, Z. Chen, H. Jiang, L. Xie, C. Spanos, Accurate indoor localization and tracking using mobile phone inertial sensors, WiFi and iBeacon, 4th IEEE Int. Symp. Inert. Sensors Syst. Inert. 2017 - Proc. (2017) 1–4. doi:10.1109/ISISS.2017.7935650.
- [13] Z. Li, D. B. Acuna, Z. Zhao, J. L. Carrera, T. Braun, Fine-grained indoor tracking by fusing inertial sensor and physical layer information in WLANs, 2016 IEEE Int. Conf. Commun. ICC 2016 (2016). doi:10.1109/ICC.2016.7511162.
- [14] L. H. Chen, E. H. K. Wu, M. H. Jin, G. H. Chen, Intelligent fusion of wi-fi and inertial sensor-based positioning systems for indoor pedestrian navigation, IEEE Sens. J. 14 (2014) 4034–4042. doi:10.1109/JSEN.2014.2330573.