Abstract—The proliferation of wireless sensor networks (WSN) open a possibility for use in indoor localization. WSN is intended principally for monitoring and control applications that demand very low power consumption from a battery source.

This study investigates the use of WSN for indoor localization. The WSN is used to track and detect the location of a moving object inside a building. This system was developed in the context of WSN application for enabling and supporting improved facilities management. The overall goal of the facilities management is to improve energy efficiency of a building.

Bayesian probabilistic approach implemented as particle filter and map filtering algorithm is presented. The location system performance is also evaluated. It is found that the location system provide reliable prediction ($\mu = 2.17m$ with $\sigma = 1.27m$).

I. INTRODUCTION

Growing interest in pervasive computing has been fostered further by the ubiquity of mobile computing devices, such as portable media players, mobile phones, and notebooks. Automatically ascertaining the physical location of people or devices is regarded as a key pervasive computing application [1]. Knowledge of physical location would open up a wide range of possible future applications; for example, tracking people and assets, safety and security, and indoor navigation.

The global positioning system (GPS) is the mainstream technology for location and tracking in outdoor environments [2]. However, GPS provides insufficient reliability and accuracy for indoor environments, since its signal is heavily attenuated by building structures such as roofs and walls. Consequently, indoor localization systems have been proposed based on other technologies.

The proliferation of wireless sensor networks (WSN) open a possibility for use in indoor localization. WSN has recently attract many studies in research community as a fundamentally new tool for monitoring and data-gathering applications. WSN, which is based on the IEEE 802.15.4 standard [3], is intended principally for monitoring and control applications that demand very low power consumption from a battery source. Many applications with WSN are proposed, such as habitat monitoring [4][5][6], industrial monitoring [7][8], battlefield surveillance [9][10] and building management [11][12]. A general setup of WSN consists of a large number of sensors densely deployed in a certain area. Each sensor is capable of sensing, processing data at a small scale, and communicating through omni-directional radio signal [13].

Augmenting WSN application with location information is crucial for several reasons. Firstly, the measurement data collected by sensors are often necessary to have the location data stamped. For example, temperature and humidity data need the corresponding position data included to provide meaningful information. Secondly, many communication protocols of sensor networks are built on the knowledge of the geographic positions of sensors [14][15].

This paper investigates the use of WSN for indoor localization. The WSN is used to track and detect the location of a moving object inside a building. The inherent wireless communication infrastructure is utilized and hence the localization technology becomes a software-only solution.

II. RELATED WORKS AND CONTRIBUTIONS

Localization using WSN has received growing research interest [16][17][18]. One of the examples of an indoor localization system is Motetrack [19], a system that uses a decentralized approach that runs on programmable sensor nodes. This approach was used to minimize per-node storage overhead and to achieve high robustness to failure. It adopted an enhanced version of simple nearest neighbor algorithm found in wireless LAN indoor localization [20].

This system was developed in the context of WSN application for enabling and supporting improved facilities management [21]. The system was developed to help navigate the facility manager inside a building. The overall goal of the facilities management is to improve energy efficiency of a building.

The system uses centralized approach and utilizes Bayesian probabilistic algorithms to enhance the performance. The main contribution of this paper is two fold. Firstly, we describe the implementation and evaluation of an indoor localization system based on WSN. Secondly, we present an analysis of our algorithms using state of the art particle filter and map filtering.

III. SYSTEM OVERVIEW

Received signal strength (RSS) is one of the channel parameters of a wireless communication system that is relevant
for localization. The wireless communication infrastructure of WSN is utilized with fingerprinting technique to infer location.

The location system works in two phases (see Fig. 1): firstly, the calibration, building a database of received signal strength (RSS) or fingerprint. Secondly, the online tracking, the client’s mobile node will scan the RSS from available WSN beacons and input them to the particle filter-based fusion engine to estimate location.

![Block diagram of the localization system.](image)

Fingerprint is saved in a database as signatures tuples (consisting of the WSN beacon address, RSS values, and measurement’s position).

IV. ALGORITHM

This work uses a Bayesian probability approach implemented as particle filter algorithm. In this approach, the goal is to infer unknown object’s (referred as target) location (referred as state) based on RSS measurements.

A. Recursive Bayesian Estimation

To define the problem during location estimation, the target state evolves according to the following discrete-time stochastic model:

\[ x_t = f_{t-1}(x_{t-1}, n_{t-1}) \]  

(1)

Where \( x_t \) denotes the state of the target being estimated; \( f_{t-1} \) is a known, possibly non-linear function of the state \( x_{t-1} \); \( n_{t-1} \) is an independent and identically-distributed noise. The measurement is related to the target state with the following model:

\[ z_t = h_t(x_t, e_t) \]  

(2)

Where \( h_t \) is a known, possibly non-linear function; \( e_t \) denotes an independent and identically-distributed noise. In the case of indoor localization, it aims to filter the object state \( x_t \) based on the sequence of all available RSS measurements \( z_t \) up to time \( t \).

From a Bayesian perspective, the problem is to recursively construct the posterior probability density function (PDF) \( p(x_t | z_t) \) given the previous posterior \( p(x_{t-1} | z_{t-1}) \). In principle pdf \( p(x_t | z_t) \) can be calculated in two stages: prediction and updates [22]. The general algorithm for calculating the posterior distribution of \( p(x_t | z_t) \) is given by Bayes’ filter. From a Bayesian perspective, construction of posterior probability is achieved in two steps: prediction and update. The prediction stage is performed to obtain the prior probability \( p(x_t | z_{t-1}) \), whereas the update stage corrects the prediction with a new measurement to obtain the posterior \( p(x_t | z_t) \).

The prediction is achieved through the Chapman-Kolmogorov equation [23]:

\[ p(x_t | z_{t-1}) = \int p(x_t | x_{t-1}) p(x_{t-1} | z_{t-1}) dx_{t-1} \]  

(3)

When the measurement \( z_t \) becomes available, Bayes’ rule is utilised for the update stage:

\[ p(x_t | z_t) = k^{-1} p(z_t | x_t) p(x_t | z_{t-1}) \]  

(4)

with normalizing constant:

\[ k = \int p(z_t | x_t) p(x_t | z_{t-1}) dx_t \]  

(5)

\( p(z_t | x_t) \) represents the measurement probability, \( p(x_t | z_{t-1}) \) is the transition probability and \( p(x_t | z_{t-1}) \) is the previous posterior probability.

**Algorithm 1 BayesFilter** \( p(x_{t-1} | z_{t-1}), z_t \)

1: for all \( x_t \) do  
2: \( p(x_t | z_{t-1}) = \int p(x_t | x_{t-1}) p(x_{t-1} | z_{t-1}) dx_{t-1} \)  
3: \( p(x_t | z_t) = k^{-1} p(z_t | x_t) p(x_t | z_{t-1}) \)  
4: end for  
5: return \( p(x_t | z_t) \)

Algorithm 1 describes the prediction and update stages of Bayes’ filter. The algorithm input is the previous posterior and a current measurement. In line 2, prediction is obtained by integration of two probability distributions: previous posterior and transition probability. The measurement update is described in line 3, where Bayes’ algorithm updates the prior probability with the measurement probability. The result is the posterior probability which is returned in line 5 of the algorithm. The recursive nature of the Bayesian filter is due to the fact that the posterior probability \( p(x_t | z_t) \) is calculated form the previous posterior \( p(x_{t-1} | z_{t-1}) \).

Knowledge of the posterior probability enables an estimation of the state to be made, for instance to obtain the mean of \( x_t \) [24].

\[ x_t^{ME} = \int x_t p(x_t | z_t) dx_t \]  

(6)

The aforementioned recursive propagation of the posterior density is only a conceptual solution in that, in general, it cannot be determined analytically. Only in a highly restrictive case when the system is linear and Gaussian, does an analytical
solution exist in the form of the Kalman filter and its numerous variants [25] [26].

Since the analytical solution is intractable for most practical situations, an approximate solution to the Bayesian filter is required. The current state of the art of such approximate solutions is the particle filter [27] [28].

B. Particle Filter Implementation

The particle filter is a non-parametric implementation of the Bayes’ filter. It approximates the posterior probability by a finite number of discrete samples with associated weights, called particles. The approximation of the posterior density is non-parametric. Therefore, it can represent a wider distribution than the parametric one, such as Gaussian. The particle filter is also known as the bootstrap filter [29], condensation algorithm [30] and survival of the fittest [31].

The particle filter directly estimates the posterior probability of the state expressed in the following equation [28]:

\[ p(x_t|z_t) \approx \sum_{i=1}^{N} w_i \delta(x_t - x^i_t) \]  

(7)

where \( x^i_t \) is the i-th sampling point or particle of the posterior probability with \( 1 < i < N \) and \( w_i \) is the weight of the particle. \( N \) represents the number of particles in the particle set, denoted by \( \mathcal{X}_t \).

\[ \mathcal{X}_t := x^1_t, x^2_t, \ldots, x^N_t \]  

(8)

Each particle is a concrete instantiation of the state at time \( t \), or put differently, each particle is a hypothesis of what the true state \( x_t \) might be, with a probability given by its weight.

The property of equation (7) holds for \( N \uparrow \infty \). In the case of finite \( N \), the particles are sampled from a slightly different distribution. However, the difference is negligible as long as the number of particles is not too small [32]. Based on recent finding [28], \( N \) value above 500 is suggested. In this work, \( N \) value of 1000 is used and remain static over time.

Algorithm 2: Particle Filter (\( \mathcal{X}_{t-1}, z_t \))

1. \( \mathcal{X}_t = \mathcal{X}_t = \emptyset \)
2. for \( i = 1 \) to \( N \) do
3. \( \) sample \( x^i_t \sim p(x_t|\mathcal{X}_{t-1}) \)
4. \( \) assign particle weight \( w^i_t = p(z_t|x^i_t) \)
5. end for
6. \( \) calculate total weight \( k = \sum_{i=1}^{N} w^i_t \)
7. for \( i = 1 \) to \( N \) do
8. \( \) normalise \( w^i_t = k^{-1} w^i_t \)
9. \( \) \( \mathcal{X}_t = \mathcal{X}_t + \{x^i_t, w^i_t\} \)
10. end for
11. \( \mathcal{X}_t = \) Resample (\( \mathcal{X}_t \))
12. return \( \mathcal{X}_t \)

The algorithm 2 describes a generic particle filter algorithm. The input of the algorithm is the previous set of the particle \( \mathcal{X}_{t-1} \), and the current measurement \( z_t \), whereas the output is the recent particle set \( \mathcal{X}_t \).

C. Motion Model

During the prediction stage each particle will have dynamics according to a motion model that represents the estimated object. Let \( x^i_t \) denote the state vector that describes the particle position in local Cartesian coordinate. Particles motion can be modeled with:

\[ x^i_t = \begin{bmatrix} x^i_t \\ y^i_t \end{bmatrix} = \begin{bmatrix} x^i_{t-1} + v^i_t \cos(\alpha^i_t) \Delta t + n^i_t \\ y^i_{t-1} + v^i_t \sin(\alpha^i_t) \Delta t + n^i_t \end{bmatrix} \]  

(9)

where \( v^i_t \) denotes velocity, \( \alpha^i_t \) describes particle direction at the time \( t \), \( n^i_t \) is a noise with Gaussian distribution.

Both particles velocity and direction can be obtained directly from an inertial sensor measurement. In this work, the approach for the motion model in indoor localization is taken from both Brownian movement [33] and the first-order motion model [34]. The motion model assumes that the kinematics of a target has random values in it. However, this randomness is constrained by the previous state.

Figure 2 illustrates the evolution of particle distribution determined only by the motion model. The particle distribution is started from a known state in 2D space. The initial velocity \( v_0 = 3 \) m/s and number of particles are 2000. Since there is no RSS measurement update performed, the distribution will spread wider over time.

![Evolution of particles with motion model.](image)

D. Likelihood Function

The likelihood function \( p(z_t|x^i_t) \) describes the probability of receiving a set of signal level tuples (signature) in a specific location. To obtain the likelihood function, statistical inference is performed between true signatures value stored in the fingerprint and recent RSS measurement during online tracking (see [35] for the detail explanation of the likelihood function).

Furthermore, it will be used for updating particle weight \( w^i_t \) (as stated in algorithm 2, point 4). The figures below show how likelihood function was used for updating the particle weight and the posterior distribution subsequently. Figure 3 shows the posterior distribution at \( t = 0 \), the weights of the particles \( w^i_0 = \frac{1}{N}, N = 3000 \).

Figures 4 show when particles weight is updated with likelihood function (blue circle), resample to obtain posterior distribution and then state estimation is calculated at \( t = 1s \), \( t = 20s \) and \( t = 30s \).
E. Map Filtering

The particle movement is also taking into account environment description, i.e.: wall, room and corridor. Map filtering is implemented in a way that particles, which act as a people representation, can not move across a wall or another solid object. Particles are only permitted to move in corridors or within rooms.

Map filtering is implemented in a fairly straightforward way. The new particle position, determined by the motion model, should fulfill the requirements mentioned above. If an attempt to find a new position fails (when moving particle path is obstructed), the algorithm will try to find a new particle position according to the motion model (equation 9). If several attempts within predetermined threshold still fail, the particle will die.

Figure 5 illustrates evolution of particles with and without map filtering from a known state in 2D space ($v_0 = 3$ m/s, $N = 2000$). Map filtering constrains particle movement within the walls, as can be seen in Figure 5(b).

V. EXPERIMENT

Power electronic laboratory at Eidgenössische Technische Hochschule Zürich (ETH) Zurich, Switzerland was used as a test-bed to analyze the proposed system. The test-bed dimension is 21m x 15m.

WSN infrastructure, which consists of 10 tmote nodes, was installed at the test-bed. Figure 6 shows the node used for the experiment. We also installed indoor localization using WLAN technology. It was used for performance comparison with the WSN. Four WLAN APs were installed in the test-bed.

The first step was to draw floor-plan of the power electronic building. We have developed a tool for drawing the floor-plan in scalable vector graphic (SVG) format. It is also possible to export the floor-plan from an AutoCAD file. Several tools have also been developed using the C++ language. They include a site survey tool for the calibration phase (building the RSS fingerprint), a fusion engine and TinyOS-based WSN software.

The RSS fingerprint was built on top of the floor-plan that was divided into 0.75m square uniform-grid. Signal levels of WSN were scanned every 0.5 second. Only a single floor problem was considered. Figure 7 shows the fingerprint measured from a WSN node.

A walk around the laboratory along the ground truth was performed for the off-line analysis. Some real measurements were collected along this path and then reused to measure the performances of each technique. Figure 8 shows the WSN infrastructure and ground truth in the test-bed.
VI. RESULT

Widely-used gauges of tracking accuracy are the mean error $\mu$ and standard deviation $\sigma$ of the difference between the ground-truth position and the prediction. The location accuracy in the test-bed is summarized in Table I.

<table>
<thead>
<tr>
<th>Localization Error (meters)</th>
<th>WLAN</th>
<th>WSN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Particle Filter with Map Filtering</td>
<td>$\mu = 5.12$</td>
<td>$\mu = 2.17$</td>
</tr>
<tr>
<td></td>
<td>$\sigma = 2.5$</td>
<td>$\sigma = 1.27$</td>
</tr>
</tbody>
</table>

Table I shows that the Particle Filter algorithm provides reasonably good location accuracy. For comparison, we also tried localization based on WLAN technology. However, it does not produce sufficient accuracy.

It was caused by WLAN fingerprint that relatively flat in an open room (see Figure 9). Whereas, WSN fingerprint has more RSS variation between grids. In the particle filter algorithm, the variation causes more distinctive likelihood function $p(z_t|x_i^t)$ between the grids. Eventually, it leads to a better localization accuracy.

WSN indoor localization provide the best accuracy ($\mu = 2.17$ m with $\sigma = 1.27$ m). It is considered sufficient for object localization in indoor environment.

VII. CONCLUSION

In this paper, an indoor localization based on WSN technology is summarized. A Bayesian approach for estimating the position is described. The Particle Filter algorithm with map filtering also evaluated.

It is found that the location system provide reliable prediction. Future works include the development of the navigation system based on the location information.

ACKNOWLEDGMENT

The authors acknowledge funding from WearIT@Work, a European Union funded project (EC IST IP2003 0004216).

REFERENCES


