2012 International Conference on Indoor Positioning and Indoor Navigation, 13-15th November 2012

# Statistical Path Loss Parameter Estimation and Positioning Using RSS Measurements in Indoor Wireless Networks

Henri NURMINEN, Jukka TALVITIE, Simo ALI-LÖYTTY, Philipp MÜLLER, Elena-Simona LOHAN,
Robert PICHÉ and Markku RENFORS
Tampere University of Technology, Tampere, Finland
Emails: {henri.nurminen, jukka.talvitie, simo.ali-loytty, philipp.muller, elena-simona.lohan,
robert.piche, markku.renfors}@tut.fi

*Abstract*—**A Bayesian method for dynamical off-line estimation of the position and path loss model parameters of a WLAN access point is presented. Two versions of three different on-line positioning methods are tested using real data. The tests show that the methods that use the estimated path loss parameter distributions with finite precisions outperform the methods that only use point estimates for the path loss parameters. They also outperform the coverage area based positioning method and are comparable in accuracy with the fingerprinting method. Taking the uncertainties into account is computationally demanding, but the Gauss–Newton optimization method is shown to provide a good approximation with computational load that is reasonable for many real-time solutions.**

***indoor positioning; received signal strength; WLAN; path loss model; statistical estimation***

## I. INTRODUCTION

This article focuses on navigation in indoor spaces, where satellite-based navigation information is typically unavailable. Wireless local area networks (WLAN) have been shown to be a platform for efficient low-cost indoor positioning, which is accessible by an increasing portion of all the mobile devices. Indoor positioning performance of WLAN-based methods can be improved by introducing more refined technologies such as various inertial navigation systems or Bluetooth [1]. However, the need for reliable WLAN-based positioning remains clear.

There are several ways to exploit WLAN signals for indoor navigation. Due to the multiplicity of the positioning environment, signal propagation modeling tends to be challenging: for instance, walls, floors, furniture and the shape of the building cause shadowing and multipath effects that cannot be modeled accurately without the knowledge of the floor plan of the building. The effect may also be varying in time. However, there is statistical correlation between the position and some signal properties. In the literature, most statistical methods are based on the Signal-to-Noise-Ratio (SNR) or the Received Signal Strength (RSS) measurements [2]. Article [3] uses statistical coverage area estimates. This paper uses RSSs as distance measurements.

A path loss (PL) model is a model for signal attenuation in space. In the literature, for example in [4] and [5], there are many different path loss modeling methods from deterministic and computationally heavy ray-tracing algorithms to empirical and semi-empirical channel models based on extensive measurement campaigns. Each model contains a set of tunable parameters which attempt to capture the nature of the investigated radio propagation environment.

The main novelty of this article comes from introducing a method for dynamic estimation of the model parameters for each access point (AP) using learning data collected at known positions. The underlying model is the simplified statistical path loss model. Estimation is based on the concepts of Bayesian statistics, which is a flexible and theoretically principled framework. The number of required path loss parameters is kept small in order to keep down the computational complexity and the amount of information required in the positioning phase. As a very important built-in property, the presented Bayesian method returns also a statistical description of the uncertainty for estimated parameter values.

Furthermore, this article shows the influence of finite PL parameter precisions on the positioning results. The proposed positioning algorithms are Monte Carlo -based Metropolis–Hastings (MH) sampler and computationally lighter Gauss–Newton method (GN). Grid positioning is used as a reference method. For each of the methods, two versions are compared. The first one uses point estimates for the path loss parameters and assumes them to be accurate. The second version assumes the parameters to follow specified probability distributions. Using collected sets of real data, the latter is shown to be superior in consistency and similar or slightly better in accuracy. The latter also outperforms the coverage area solution, which does not use RSS measurements. Its accuracy is also comparable with the fingerprinting method, in which the RSS report is pattern-matched with the entire learning set.

In practice WLAN positioning is typically complemented by additional, more refined techniques such as map information, inertial navigation systems and Bluetooth. In this paper, we do not use additional information or other measurements. However, additional information and other measurements are easy to add to presented positioning algorithms.

The paper is organized as follows: First, in Section II the

978-1-4673-1954-6/12/$31.00 

path loss model is introduced and the method for estimating the model parameters is presented. Then, in Section III the path loss model is formulated as a statistical measurement model for the positioning phase, and the positioning algorithms are presented in Section IV. The results are shown in Section V. Finally, Section VI shows the conclusions.

**Notations:** Matrices are denoted with unitalicised uppercase letters. Vectors and scalars are not distinguished. $\mathrm{N}(m, \mathrm{P})$ refers to the (multivariate) normal distribution with mean $m$ and covariance matrix P, and $\mathrm{N}_{\mathrm{P}}^{m}(x)$ refers to its probability density function (pdf) evaluated at $x$.

## II. Path loss model

### A. Path loss model input

This sections presents a method for estimating the path loss parameters and location for a single AP. This procedure is then applied to each AP in the learning data. The assumption that the parameters of separate APs are statistically independent may result in some information losses, but it will simplify the form the AP database that is created and reduce the number of recorded statistics.

The input for the parameter estimation procedure is a set of RSS measurements of signals transmitted by the AP. The measurement set $\Omega$ includes $N_{\text{fp}}$ measurements given as

$$\Omega \in \{(x_i, P_i) \mid i = 1, 2, \ldots, N_{\text{fp}}\}, \tag{1}$$

where $x_i \in \mathbb{R}^2$ includes the easting and northing of the $i$:th measurement point, and $P_i$ is the received signal power of the $i$:th measurement point in dBm. We assume that the transmitter power and antenna gains are fixed during the measurements, so the RSS is only dependent on the measurement coordinates $x_i$. In this paper, we use only 2-dimensional coordinates, since in many positioning applications this is enough to fulfill the use case expectations. Extensions to 3D cases are under research.

### B. Path loss model definition

Friis's law determines the received signal power as a function of distance in a free space as

$$p_{rx}(d) = p_{tx} g_{tx} g_{rx} \left(\frac{\lambda}{4\pi d}\right)^2, \tag{2}$$

where $p_{tx}$, $g_{tx}$, $g_{rx}$, and $\lambda$ are the transmitted signal power, transmitter antenna gain, receiver antenna gain, and signal wavelength, respectively. The distance between transmitter and receiver antenna is $d$. The square term is the actual channel dependent path loss term, while the other parameters are transmitter and receiver dependent. However, using the free-space model could be a practical approach only in line-of-sight scenarios, but not in real-life networks where there are obstacles on the radio signal path.

One of the most recognized outdoor path loss models is the classical log-distance model (or power law model) [6]. This model has also been applied in indoor positioning e.g. in [7] and in [8] that uses separate model parameters depending on the RSS value. In the log-distance model the received signal power is defined as

$$P_{rx}(d) = P_{rx}(d_0) - 10n \log_{10}\left(\frac{d}{d_0}\right) + w \tag{3}$$

where the power $P_{rx}(\cdot)$ is given in logarithmic scale, $d_0$ is a reference distance, $n$ is a path loss exponent, and $w \sim \mathrm{N}(0, \sigma^2)$ is a normally distributed random variable which models the slow fading (shadowing) phenomenon. Here the path loss exponent $n$ and the slow fading standard deviation are dependent on the local propagation environment. Notice that since the term $P_{rx}(d_0)$ indicates the received signal power at the reference distance $d_0$, it automatically takes account of the transmission power along with the antenna gains and wavelength shown in (2). Moreover, apart from the slow fading, $P_{rx}(d)$ is only affected by the path loss exponent $n$ whenever $d > d_0$. Now, by defining $d_0 = 1\,\mathrm{m}$, and denoting $P_{rx}(d_0) = A$, it is possible to write the final path loss model as

$$P_{rx}(d) = A - 10n \log_{10}(d) + w, \tag{4}$$

where the parameter $A$ is referred to as apparent transmission power. Note that each AP will have different path loss parameters.

### C. Estimation of AP position and path loss parameters

AP position and path loss parameters for a single AP are estimated using the Iterative Reweighted Least Squares method (Gauss–Newton method). The function to be minimized is

$$\phi(A, n, m) = \sum_{i=1}^{N_{\text{fp}}} \left|\left|P_i - h_{\text{est}}^i(A, n, m)\right|\right|^2, \tag{5}$$

where

$$h_{\text{est}}^i(A, n, m) = A - 10n \log_{10}(||m - x_i||).$$

$A$ is apparent transmission power, $n$ path loss exponent and $m$ BS position. The Jacobian matrix of the measurement model function $h_{\text{est}}^i$ is

$$\mathrm{J} = \begin{bmatrix} 1 & -10\log_{10}(||m - x_1||) & -\frac{10}{\ln(10)} n \frac{(m-x_1)^{\mathrm{T}}}{||m-x_1||^2} \\ \vdots & \vdots & \vdots \\ 1 & -10\log_{10}(||m - x_{N_{\text{fp}}}||) & -\frac{10}{\ln(10)} n \frac{(m-x_{N_{\text{fp}}})^{\mathrm{T}}}{||m-x_{N_{\text{fp}}}||^2} \end{bmatrix}. \tag{6}$$

The Bayesian Gauss–Newton algorithm is described in detail in Algorithm 3.

To improve convergence properties of the Gauss–Newton algorithm, all the quantities are given an almost uninformative Gaussian prior, i.e. a Gaussian distribution with so large variance that the influence on the optimum is negligible. A suitable initial value for the AP position is the position of the strongest observed measurement, because otherwise the algorithm might locate the AP position to the area of the weakest RSSs. Initial values for $A$ and $n$ can be chosen more arbitrarily from the valid ranges, since the distribution is typically unimodal, if the number of data points is large.

Note that the algorithm also returns an approximation for the covariance matrix of each quantity. Consequently, we are potentially able to distinguish between trustworthy and untrustworthy path loss models. In the Bayesian sense, the algorithm tries to estimate the MAP value (*maximum a posteriori*), and the covariance matrix is the covariance of the linearized model.

In an ideal case without any slow fading variations the AP position would be found at the coordinate point where the received signal power reaches its maximum value. However, in practice there might be several clear peaks in the received signal power map or there might not be enough measurements to find even a single peak. Besides, the AP might not even be located inside the measured power map.

The effects due to measurement error correlations are taken into account heuristically by increasing the covariance matrix of the AP position artificially with a small constant diagonal matrix. For this reason and for reducing the number of recorded parameters, the cross-covariances of APs and path loss parameters are ignored. AP positions and path loss parameters are thus assumed to be uncorrelated. Fig. 1 shows power maps (interpolated between the measurement points) of two separate APs, and the resulting AP position estimates along with the covariance ellipse.

As pointed out before, path loss exponent $n$ and the slow fading standard deviation are highly dependent on the radio propagation environment. For example, in a shadowed urban cellular radio network the typical values of $n$ and $\sigma$ are varying around $n = 0.1 - 4$ and $\sigma = 1 - 6\,\text{dBm}$, respectively [5, 9]. Examples of resulted path loss model curves can be found in Fig. 2, in which the path loss models are computed for the same APs that were previously show in Fig. 1.

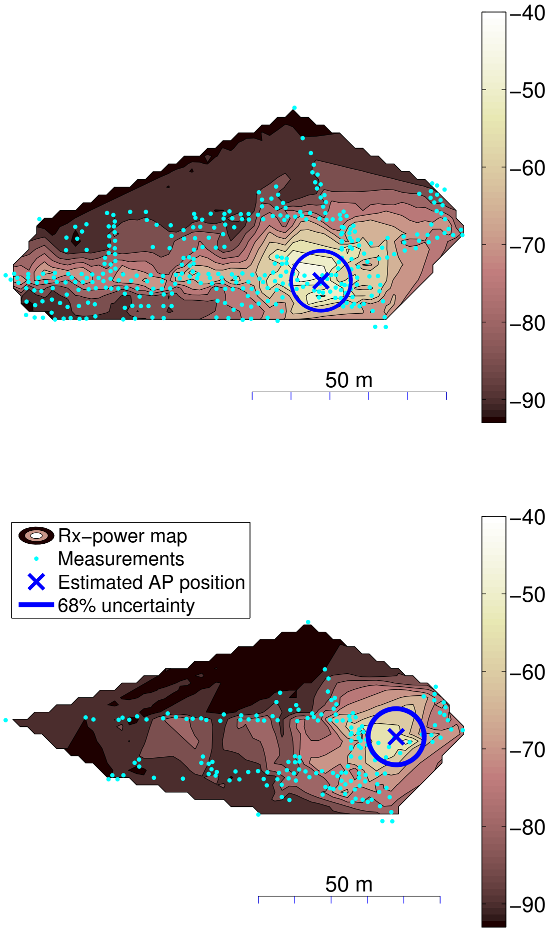


Figure 1. Power maps of two separate APs and the AP position estimates along with covariance ellipses. The color indicates the interpolated RSS power map in dBm.

## III. Estimation theory

### A. Bayesian filtering equations

Consider the Gaussian system

$$\begin{aligned} x_{k+1} &= \Phi x_k + w_k \\ y_k &= h(x_k, a) + v_k, \end{aligned} \tag{7}$$

where $y_k \in \mathbb{R}^{N_{y_k}}$ is the vector of observations at time instant $t_k$, $x_k \in \mathbb{R}^{N_x}$ represents the state of the system at $t_k$ and $a \in \mathbb{R}^{N_a}$ represents nuisance parameters that have prior distribution $p(a)$. The motion model is linear and independent of the nuisance parameters. The random noise terms $w_k \sim \mathrm{N}(0, \mathrm{Q}_k)$ and $v_k \sim \mathrm{N}(0, \mathrm{R}_k)$ are assumed to be mutually independent and independent of state $x$ and parameter vector $a$. Matrix $\Phi \in \mathbb{R}^{N_x \times N_x}$ is state transition matrix.

By the Chapman–Kolmogorov equation, the prior distribution of the state at time instant $k$ is

$$p(x_k|y_{1:k-1}) = \int p(x_k|x_{k-1}) p(x_{k-1}|y_{1:k-1})\,\mathrm{d}x_{k-1}. \tag{8}$$

This is the prediction step of a Bayesian filter.

We assume that the estimate of parameter vector $a$ is not modified online, so it is approximated that the distribution of $a$ remains unaffected by the data that is received in the positioning phase, i.e. $p(a|x_{1:k}, y_{1:k}) \approx p(a)$. By Bayes' rule, the posterior pdf of the state is thus

$$\begin{aligned} p(x_k|y_{1:k}) &= \int p(x_k, a|y_{1:k})\,\mathrm{d}a \\ &= \frac{\int p(y_k|a, x_k) p(x_k, a|y_{1:k-1})\,\mathrm{d}a}{\int\int p(y_k|a, x_k) p(x_k, a|y_{1:k-1})\,\mathrm{d}a\,\mathrm{d}x_k} \\ &\approx \frac{\int p(y_k|a, x_k) p(a) p(x_k|y_{1:k-1})\,\mathrm{d}a}{\int\int p(y_k|a, x_k) p(a) p(x_k|y_{1:k-1})\,\mathrm{d}a\,\mathrm{d}x_k}. \end{aligned} \tag{9}$$

This is the update step of the Bayesian filter with unknown static nuisance parameters in the measurement model.

The filtering technique used in this paper is an approximation of the general Bayesian filtering procedures. The presented positioning algorithms are formulated so that they return the posterior mean $\hat{x}_k^+$ and covariance matrix $\hat{\Sigma}_k^+$ of the user position. The posterior distribution is then approximated

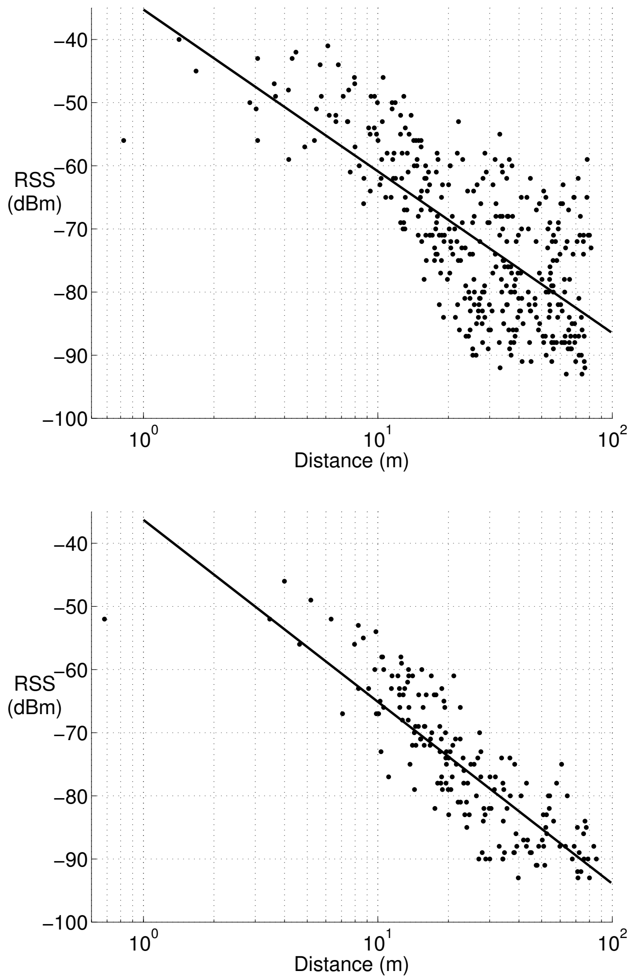


Figure 2. Path loss curves for two separate APs.

by a normal distribution with the estimated parameter values. Using this simplification and linear motion model with additive Gaussian noise, the filter prediction step (8) becomes

$$p(x_k|y_{1:k-1}) = \mathrm{N}^{\Phi\hat{x}^+_{k-1}}_{\Phi\hat{\Sigma}^+_{k-1}\Phi^{\mathrm{T}}+\mathrm{Q}}(x_k), \tag{10}$$

where

$$\Phi = \mathrm{I}_{2\times 2}, \quad \mathrm{Q} \ \propto (\Delta t)^2 \cdot \mathrm{I}_{2\times 2}.$$

This approximation is done in order to simplify the prediction step of the filter, which is now the conventional Kalman filter prediction. Note that this procedure may result in information losses especially in case of multimodal posterior.

### *B. Path loss model*

The path loss model with uncertain parameters presented in Section II is a special case of model (7). In this case, $y_k$ is the vector of RSS measurements $y_k = \begin{bmatrix} P_1 & \dots & P_{N_P} \end{bmatrix}^{\mathrm{T}}$ at time instant $t_k$ and $x_k$ is the user position. Parameter vector $a$ contains the path loss model parameters of all the possible APs, but the measurement contains information only on the observed APs, so the parameter vector can be denoted by

$$a = \begin{bmatrix} A_1 & n_1 & m_1^{\mathrm{T}} & \cdots & A_{N_P} & n_{N_P} & m_{N_P}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}.$$

The measurement model function is

$$h(x,a) = \begin{bmatrix} A_1 - 10 n_1 \log_{10}(||m_1 - x||) \\ \vdots \\ A_{N_P} - 10 n_{N_P} \log_{10}(||m_{N_P} - x||) \end{bmatrix},$$

where $N_P$ is the number of observed APs. Measurement noise covariance matrix is $\mathrm{R} = \sigma^2 \cdot \mathrm{I}_{N_P \times N_P}$. Note that because the path loss parameters of an AP are isotropic, $p(a|x_k) = p(a)$. For simplicity, this paper assumes that the path loss parameters' prior distributions are Gaussian and that AP position and PL parameters are independent *a priori*. Thus,

$$\begin{aligned} p(a) &= p(A_{1:N_P}, n_{1:N_P}, m_{1:N_P}) \\ &= \prod_{i=1}^{N_P} \mathrm{N}^{\hat{\mu}_{A_i,n_i}}_{\hat{\Sigma}_{A_i,n_i}}\left(\begin{bmatrix} A_i \\ n_i \end{bmatrix}\right) \cdot \mathrm{N}^{\hat{m}_i}_{\hat{\Sigma}_{m_i}}(m_i), \end{aligned} \tag{11}$$

where the parameters $\hat{\mu}_{A_i,n_i} = \begin{bmatrix} \hat{A}_i & \hat{n}_i \end{bmatrix}^{\mathrm{T}}$, $\hat{\Sigma}_{A_i,n_i}$, $\hat{m}_i$ and $\hat{\Sigma}_{m_i}$ are estimated from the learning data using the Gauss–Newton algorithm.

The priors are modeled to be normal, since the Gauss–Newton algorithm requires this in its basic form and the normal pdf of $A$ and $n$ is the conjugate prior of the likelihood. However, other prior distribution families such as Student's $t$-distribution could also be studied.

## IV. Positioning algorithms

In this section, a Gaussian prior distribution for the user position $p(x) = p_{x_k|y_{1:k-1}}(x) = \mathrm{N}^{\hat{x}}_{\hat{\Sigma}}(x)$ is assumed.

### *A. Grid method*

By (9), the posterior pdf value at point $x$ is

$$\begin{aligned} p(x|P_{1:N_P}) \propto \prod_{i=1}^{N_P} \iiint & p(P_i|x, A_i, n_i, m_i) p(A_i, n_i) p(m_i) \\ & \mathrm{d}A_i \, \mathrm{d}n_i \, \mathrm{d}m_i \cdot p(x), \end{aligned} \tag{12}$$

which can be approximated using standard Monte Carlo integration. The grid method is presented in Algorithm 1.

The most crucial implementation issues are the Monte Carlo sample size parameter $N$ as well as grid size and density. Note that combining of the likelihood of each AP is done in logarithmic space to avoid numerical underflows.

### *B. Metropolis–Hastings method*

The Metropolis–Hastings (MH) sampler generates Monte Carlo samples from an arbitrary posterior distribution of a multivariate random variable. It is an iterative algorithm that can be proved to converge towards the target distribution. The posterior mean and covariance can then be approximated

**Algorithm 1** Grid with Monte Carlo Integration

1) Set a grid $\{x_m \in \mathbb{R}^2 \mid m \in \{1,\dots,N_m\}\}$ that covers most of the prior probability mass.
2) For each heard base station $i = 1,\dots,N_P$, draw
$$\begin{bmatrix} A_i^{(k)} \\ n_i^{(k)} \end{bmatrix} \leftarrow \mathrm{N}\left(\begin{bmatrix}\hat{A}_i \\ \hat{n}_i\end{bmatrix}, \hat{\Sigma}_{A_i,n_i}\right)$$
$$m_i^{(k)} \leftarrow \mathrm{N}\left(\hat{m}_i, \hat{\Sigma}_{m_i}\right)$$
for $k = 1,\dots,N$.
3) At each grid point $x_m$ compute for each AP $i = 1,\dots,N_P$ and for each each sample $k = 1,\dots,N$
$$I_{i,m}^{(k)} := \mathrm{N}_{\sigma^2}^{P_i}\left(A_i^{(k)} - 10 n_i^{(k)} \log_{10}\left(\left|\left|m_i^{(k)} - x_m\right|\right|\right)\right),$$
and $I_{i,m} := \frac{1}{N}\sum_{k=1}^{N} I_{i,m}^{(k)}$. Then set
$$\ell_m := \ln\left(\mathrm{N}_{\hat{\Sigma}}^{\hat{x}}(x_m)\right) + \sum_{i=1}^{N_P} \ln(I_{i,m}), \quad L_m := \exp(\ell_m).$$
4) Normalize the grid to get a set of weights $w_m = \frac{L_m}{\sum_{m=1}^{n_m} L_m}$ and compute mean and covariance estimates
$$\hat{x}^+ := \sum_{m=1}^{N_m} w_m x_m$$
$$\hat{\Sigma}^+ := \sum_{m=1}^{N_m} w_m (x_m - \hat{x}^+)(x_m - \hat{x}^+)^{\mathrm{T}}.$$

by the sample mean and covariance of the sampled set. The algorithm is presented in Algorithm 2.

The MH sampler uses a so-called proposal distribution, from which it is straightforward to generate random numbers. At each iteration of the algorithm, proposal values for the estimated variables are drawn from the proposal distribution. The proposal values are then accepted with the probability that is proportional to the ratio of the pdf values of the proposal value and the latest accepted value. [10, Ch. 5]

Note that
$$\begin{aligned} &p(x, m_{1:N_P} | P_{1:N_P}) \\ &\overset{(9)}{\propto} \iint p(P_{1:N_P} | x, A_{1:N_P}, n_{1:N_P}, m_{1:N_P}) \\ &\quad \cdot p(A_{1:N_P}, n_{1:N_P}) \,\mathrm{d}A_{1:N_P}\, \mathrm{d}n_{1:N_P} \cdot p(x) \cdot p(m_{1:N_P}) \\ &\propto p(x) \prod_{i=1}^{N_P} p(m_i) \det(\check{\Sigma}_{A_i,n_i})^{\frac{1}{2}} \exp\left(\frac{1}{2}\check{\mu}_{A_i,n_i}^{\mathrm{T}} \check{\Sigma}_{A_i,n_i}^{-1} \check{\mu}_{A_i,n_i}\right) \end{aligned} \tag{13}$$

where
$$\check{\Sigma}_{A_i,n_i} := \left(\frac{1}{\sigma^2}\mathrm{B}_i{}^{\mathrm{T}}\mathrm{B}_i + \hat{\Sigma}_{A_i,n_i}^{-1}\right)^{-1}$$
$$\check{\mu}_{A_i,n_i} := \check{\Sigma}_{A_i,n_i}\left(\frac{1}{\sigma^2}\mathrm{B}_i{}^{\mathrm{T}} P_i + \hat{\Sigma}_{A_i,n_i}^{-1}\begin{bmatrix}\hat{A}_i \\ \hat{n}_i\end{bmatrix}\right),$$

where $\mathrm{B}_i = \begin{bmatrix} 1 & -10\log_{10}(||m_i - x||)\end{bmatrix}$. The simple form of this formula enables analytical integration over PL parameters $A$ and $n$.

In the implementation phase, great care must be taken when setting the proposal distributions to make the algorithm converge in a computationally feasible number of iterations. For convenience, the proposal distributions are chosen to be multivariate normal with the latest accepted value as the mean and the covariance matrices $\mathrm{P}_x$ and $\mathrm{P}_{m_i}$ tuned from prior covariances of $x$ and $m_{1:N_P}$.

**Algorithm 2** Metropolis–Hastings algorithm

1) Set $x^{(0)} := \hat{x}$, $A_i^{(0)} := \hat{A}_i$, $n_i^{(0)} := \hat{n}_i$ and $m_i^{(0)} := \hat{m}_i^{(0)}$ for $i = 1,\dots,N_P$. Set $p^{(0)}$ using the formulae in step 3. Set $k = 1$.
2) Generate $x'^{(k)} \leftarrow \mathrm{N}(x^{(k-1)}, \mathrm{P}_x)$ , and for each AP $i = 1,\dots,N_P$, generate $m_i'^{(k)} \leftarrow \mathrm{N}(m_i^{(k)}, \mathrm{P}_{m_i})$.
3) For each $i = 1,\dots,N_P$, compute $\mathrm{B}_i^{(k)} = \left[1 \quad -10\log_{10}\left(\left|\left|m_i'^{(k)} - x'^{(k)}\right|\right|\right)\right]$ and
$$\check{\Sigma}_{A_i,n_i}^{(k)} := \left(\frac{1}{\sigma^2}\mathrm{B}_i^{(k)\mathrm{T}}\mathrm{B}_i^{(k)} + \hat{\Sigma}_{A_i,n_i}^{-1}\right)^{-1}$$
$$\check{\mu}_{A_i,n_i}^{(k)} := \check{\Sigma}_{A_i,n_i}^{(k)}\left(\frac{1}{\sigma^2}\mathrm{B}_i^{(k)\mathrm{T}} P_i + \hat{\Sigma}_{A_i,n_i}^{-1}\begin{bmatrix}\hat{A}_i \\ \hat{n}_i\end{bmatrix}\right)$$
$$\begin{aligned} p'^{(k)} &:= \ln\left(\mathrm{N}_{\hat{\Sigma}}^{\hat{x}}(x'^{(k)})\right) + \sum_{i=1}^{N_P}\left[\frac{1}{2}\ln\left(\det(\check{\Sigma}_{A_i,n_i}^{(k)})\right)\right. \\ &\left. + \frac{1}{2}\,\check{\mu}_{A_i,n_i}^{(k)}{}^{\mathrm{T}}\,\check{\Sigma}_{A_i,n_i}^{(k)}{}^{-1}\,\check{\mu}_{A_i,n_i}^{(k)} + \ln\left(\mathrm{N}_{\hat{\Sigma}_{m_i}}^{\hat{m}_i}(m_i'^{(k)})\right)\right] \end{aligned}$$
4) Set $r := \exp(p'^{(k)} - p^{(k-1)})$. Generate $u \leftarrow \mathrm{Uni}(0,1)$. Compute

   **if** $r > u$ **then**
   **for** $i = 1 : N_P$ **do**
   $m_i^{(k)} := m_i'^{(k)}$
   **end for**
   $x^{(k)} := x'^{(k)},\ p^{(k)} := p'^{(k)}$
   **else**
   **for** $i = 1 : N_P$ **do**
   $m_i^{(k)} := m_i^{(k-1)}$
   **end for**
   $x^{(k)} := x^{(k-1)},\ p^{(k)} := p^{(k-1)}$
   **end if**
5) Set $k := k+1$. If $k < N$, go to step 2. Otherwise, set
$$\hat{x}^+ := \frac{1}{N}\sum_{k=1}^{N} x^{(k)}$$
$$\hat{\Sigma}^+ := \frac{1}{N}\sum_{k=1}^{N}(x^{(k)} - \hat{x}^+)(x^{(k)} - \hat{x}^+)^{\mathrm{T}}.$$

### C. Gauss–Newton method

With suitable measurement models, iterative state estimation methods can be as accurate as any closed form solution but simpler and easier to implement [11]. The Gauss–Newton method, also known as the Iterative Reweighted Least Squares method, is tested for positioning with the presented path loss model. A similar kind of method is studied for positioning in simulated cases in [12]. The detailed description is in Algorithm 3.

The iteration does not converge globally, but including good enough prior information and initial values usually prevents the method from diverging. To improve convergence properties further, the step length in the state-space is adaptive so that the objective function value decreases at every iteration. As an exception to this, the while loop is exited after a number of iterations to ensure stability. The global convergence results of the Gauss–Newton method with adaptive step length are discussed in [13, 14].

For formulating the Jacobian matrix J that is needed in the Gauss–Newton algorithm, the analytical partial derivatives of the measurement function $h$ are formed:

$$\frac{\partial h_i}{\partial x} = \frac{10}{\ln(10)} n_i \frac{(m_i - x)^{\mathrm{T}}}{||m_i - x||^2}, \ \frac{\partial h_i}{\partial A_i} = 1,$$

$$\frac{\partial h_i}{\partial n_i} = -10 \log_{10}(||m_i - x||), \frac{\partial h_i}{\partial m_i} = -\frac{10}{\ln(10)} n_i \frac{(m_i - x)^{\mathrm{T}}}{||m_i - x||^2}.$$

The remaining partial derivatives are zeros. Note that the prior covariance matrix is always full rank so the least-squares estimation can be performed. The measurement covariance matrix R is the diagonal matrix of the measurement variances. In Algorithm 3 the complete state is denoted with $z = \begin{bmatrix} x \\ a \end{bmatrix}$. As in the PL parameter estimation phase, the output of the algorithm contains estimates for the MAP and the covariance matrix of the posterior of the linearized model.

### D. Comparison methods

The presented methods are compared with two conventional purely WLAN-based indoor positioning methods: statistical coverage areas (CA) [3, 15] and the (weighted) $k$-nearest neighbor algorithm (WKNN) [16].

The statistical CAs are bivariate Gaussian distributions that are fitted to the fingerprint database. Since the product of Gaussian densities is a Gaussian density, the standard Kalman filte can be applied to filter these measurements.

In the WKNN method, the measurements are not compressed into parametric form, i.e. no statistical assumptions are made of the measurement model. Instead, the whole measurement database is stored in the memory. In the positioning phase, the difference of the measurement to each database point is computed using the Euclidean norm of RSS differences, and the location estimate is set to the mean value of three closest database points. The WKNN estimates are not filtered in this paper, and the algorithm only uses measurements of one time instant.

**Algorithm 3** Gauss–Newton algorithm

1) Choose the stopping tolerance $\delta$. Let

$$\hat{\Sigma}_z := \mathrm{blkdiag}(\hat{\Sigma}, \hat{\Sigma}_{A_1,n_1}, \hat{\Sigma}_{m_1}, \ldots, \hat{\Sigma}_{A_{N_P},n_{N_P}}, \hat{\Sigma}_{m_{N_P}})$$

and

$$\hat{z} := \begin{bmatrix} \hat{x}^{\mathrm{T}} & \hat{A}_1 & \hat{n}_1 & \hat{m}_1^{\mathrm{T}} & \ldots & \hat{A}_{N_P} & \hat{n}_{N_P} & \hat{m}_{N_P}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$$

be the prior covariance and mean. Let the initial guess be $z_0 := \hat{z}$. Additionally, measurement covariance matrix R is required. Set $k := 0$. Denote the objective function with

$$\theta(z) := (z - \hat{z})^{\mathrm{T}} \hat{\Sigma}_z^{-1} (z - \hat{z}) + \sum_{i=1}^{N_P} \frac{h_i(z) - P_i}{\sigma^2}.$$

2) Compute the Jacobian

$$\mathrm{J}_k := \begin{bmatrix} \frac{\partial h_1}{\partial x} & \frac{\partial h_1}{\partial A_1} & \frac{\partial h_1}{\partial n_1} & \frac{\partial h_1}{\partial m_1} & \mathbf{0}^{\mathrm{T}}_{4(N_P-1)} \\ \vdots & & \ddots & & \vdots \\ \frac{\partial h_{N_P}}{\partial x} & \mathbf{0}^{\mathrm{T}}_{4(N_P-1)} & \frac{\partial h_{N_P}}{\partial A_{N_P}} & \frac{\partial h_{N_P}}{\partial n_{N_P}} & \frac{\partial h_{N_P}}{\partial m_{N_P}} \end{bmatrix}.$$

3) Set

$$\Delta z_k := -\left( \hat{\Sigma}_z^{-1} + \frac{1}{\sigma^2} \mathrm{J}_k^{\mathrm{T}} \mathrm{J}_k \right)^{-1} \cdot \left( \hat{\Sigma}_z^{-1}(z_k - \hat{z}) + \frac{1}{\sigma^2} \mathrm{J}^{\mathrm{T}} (h(z_k) - P) \right).$$

4) Adapt step length:

$\alpha := 1$
**while** $||\theta(z_k + \alpha \Delta z_k)|| \geq ||\theta(z_k)||$ and $\alpha > \alpha_0$ **do**
 $\alpha := \frac{\alpha}{2}$
**end while**

where $\alpha_0$ is a configuration parameter, e.g. 0.05. Set $z_{k+1} := z_k + \alpha \Delta z_k$.

5) If stopping condition $||\Delta z_k|| < \delta$ is not satisfied and $k \leq k_{\max}$, increment $k$ and repeat from Step 2. Otherwise compute $\mathrm{P} := \left( \hat{\Sigma}_z^{-1} + \frac{1}{\sigma^2} \mathrm{J}_k^{\mathrm{T}} \mathrm{J}_k \right)^{-1}$ and set the state estimate

$$\hat{x}^+ := z_{k+1,1:2}, \quad \hat{\Sigma}^+ := \mathrm{P}_{1:2,1:2}$$

## V. Positioning Tests With Real Data

### A. Experiment setup

A measurement campaign was accomplished to evaluate the performance of different algorithms in a real use case. First, a large set of WLAN fingerprints was collected in public indoor spaces in the city of Tampere, Finland for learning the radiomap. The measured RSS values are based on the measured Received Signal Code Power (RSCP) indicator reported by the measurement device. Path loss model parameters were estimated using the method that was presented in Section II of this paper. Furthermore, statistical coverage areas (CA) were

TABLE I. Results for the real data tests.

| Solver | Mean (m) | Med (m) | 95% err (m) | Cons (%) | Time (s) |
|---|---|---|---|---|---|
| grid, N | 7.2 | 5.6 | 18.7 | 88 | 84.67 |
| grid, acc | 7.4 | 5.9 | 19.8 | 80 | 34.04 |
| MH, N | 7.6 | 6.0 | 20.6 | 83 | 43.60 |
| MH, acc | 7.6 | 6.2 | 20.0 | 80 | 9.02 |
| GN, N | 6.9 | 5.6 | 18.3 | 91 | 1.33 |
| GN, acc | 6.9 | 5.2 | 20.8 | 64 | 0.34 |
| CA | 8.8 | 7.6 | 17.0 | 13 | 0.18 |
| WKNN | 5.7 | 4.8 | 11.1 | | 29.12 |

fitted for each AP using the fingerprint database and Gaussian distribution [3].

The test case presented in this paper is located in a building at Tampere University of Technology campus area. The test track and most of the learning data have been collected indoors. The test track consists of several parts measured at different floors of the same building. The measurement device is a tablet computer, and the reference locations were set manually on the floor plan figure. The correct floor is assumed known in both learning and positioning phases.

### B. Results and discussion

Fig. 3 shows position solutions for a part of the test track given by both versions of the GN algorithm. The results of the true data tests are in Table I and Table II. In Table II only 20% of the grid points in the learning data have been used for every third AP. Abbreviation "N" stands for the algorithms that assume the path loss parameters to be normally distributed *a priori* whereas "acc" indicates that the parameter values are assumed known. "CA" refers to the product of coverage areas. "WKNN" is the k-Nearest Neighbor method with the Euclidean distance.

The positioning error at one time step is the Euclidean distance of the position estimate and the corresponding reference location. Columns "Mean", "Med" and "95% err." are mean error, median error and empirical 95% percentile of errors in meters. "Time" is the the average running time of our MATLAB implementation in seconds. Note that the codes are not highly optimized so the running time values have to be considered only roughly indicative. The times are also highly dependent on the chosen configuration parameters. Column "Cons." displays the 95% consistency that was determined using Gaussian consistency test [17, pp. 235] with risk level 5%. The solver is deemed to be consistent at a certain time step if the true position is within the 95%-ellipse of the posterior distribution, assuming normality of the posterior. The closer this number is to 95%, the more realistic the covariance matrix estimation is.

In terms of the presented error statistics, the proposed RSS methods seem to perform better than the coverage area solution but slightly worse than the WKNN solution in positioning accuracy. Note, however that both Gauss–Newton solutions are computationally much more efficient and the requirements for the database are much lower for the parametric algorithms, since only the PL parameter estimates and their variances have to be stored for each AP instead of all the measurement points. Moreover, pruning the database influences the WKNN solution much more than the statistical methods.

TABLE II. Results for the real data tests. For every second AP nine out of ten location reports have been removed artificially. These APs and the left-out points were chosen randomly.

| Solver | Mean (m) | Med (m) | 95% err (m) | Cons (%) | Time (s) |
|---|---|---|---|---|---|
| grid, N | 7.8 | 7.0 | 18.1 | 92 | 80.58 |
| grid, acc | 8.1 | 6.9 | 19.6 | 71 | 35.28 |
| MH, N | 7.7 | 6.9 | 19.8 | 81 | 41.65 |
| MH, acc | 8.2 | 7.6 | 19.2 | 70 | 8.77 |
| GN, N | 6.9 | 6.2 | 15.3 | 93 | 1.16 |
| GN, acc | 8.6 | 7.5 | 20.5 | 49 | 0.35 |
| CA | 9.6 | 8.3 | 18.1 | 11 | 0.18 |
| WKNN | 7.9 | 6.3 | 15.9 | | 16.89 |


Figure 3. Part of the test track with the GN method. In the upper figure parameter uncertainties have been taken into account.

As shown by Table I, the presented "N" algorithms seem to outperform "acc" algorithms slightly in the positioning accuracy. However, the differences are clearer in Table II, where the learning data sets of some the APs have been pruned. Thus, it seems that PL parameter uncertainties should be taken into account especially if some of the APs are likely to be badly mapped. This might be the case e.g. if there are newly added APs or if the area as a whole is inadequately covered by the database.

From the figures it can be seen that taking the parameter uncertainties into account improves the consistency remarkably and independent of the estimation method. When several types of measurements are combined, it is crucial to be knowledgeable of the accuracy of each measurement, and so the improvement in consistency is a good reason for taking the parameter uncertainties into account in indoor positioning.

Among the three estimation methods, the grid and MH sampler approach the exact Bayesian model posterior distribution. The grid gives precise posterior values in the grid points assuming that the Monte Carlo integration's accuracy is adequate. The MH sampler converges theoretically to the true posterior as the sample size parameter $N$ approaches infinity. In practice, however, the rate of convergence in MH algorithms is highly dependent on the form and parameters of the proposal distributions. With the chosen configuration the MH method provides an equally performing estimate as the grid method with slightly lighter computation. Due to the non-Gaussianity of e.g. many inertia-based indoor navigation measurements, grid and Monte Carlo algorithms provide the most natural framework for incorporating these measurements into the filtering system.

The Gauss–Newton method lacks global convergence properties, and the covariance matrix estimate is based on iterative linearization procedure and has thus a less clear Bayesian interpretation. However, the presented results are comparable with those of the other methods, and the GN is clearly the computationally lightest one of the RSS algorithms and is applicable in many real-time solutions.

## VI. Conclusions

This article presented statistical methods for dynamic path loss parameter estimation and positioning using received signal strength measurements. According to the tests performed using wireless LAN in indoor spaces, RSS positioning based on dynamically estimated PL parameters outperforms cell-ID-based coverage area positioning and is comparable in accuracy with the $k$-Nearest Neighbor method. The database requirements of PL model methods are lighter that those of the $k$-Nearest Neighbor method, and the PL model methods are less sensitive to inadequate database coverage. It was also shown that taking the parameter uncertainties into account in the positioning phase improves positioning accuracy and especially consistency of error estimates compared to the methods in which the path loss parameters are assumed to be known accurately. The differences are emphasized if some of the APs have been estimated using pruned learning database. Furthermore, it was shown that Gauss–Newton optimization algorithm provides satisfactory accuracy and consistency compared with grid and Metropolis–Hastings methods, being also computationally feasible for many real-time applications. Adding other sources of navigation information such as maps or inertia-based information and showing the influence of the parameter uncertainties in a hybrid positioning system is a topic for future research.

## Acknowledgment

This research was partly funded by Nokia Inc. and the Academy of Finland. The authors are grateful to Dr. Tech. Lauri Wirola and Dr. Tech. Jari Syrjärinne for their support and advice. Shweta Shrestha, Toni Fadjukoff and Miila Martikainen are thanked for conducting the measurements.

## References

[1] W. Chai, C. Chen, E. Edwan, J. Zhang, and O. Loffeld, "INS/Wi-Fi based indoor navigation using adaptive Kalman filtering and vehicle constraints," in *Proceedings of 9th Workshop on Positioning, Navigation and Communication 2012 (WPNC'12)*, 2012.

[2] P. Addesso, L. Bruno, and R. Restaino, "Adaptive localization techniques in WiFi environments," in *2010 5th IEEE International Symposium on Wireless Pervasive Computing (ISWPC)*, May 2010, pp. 289–294.

[3] L. Koski, T. Perälä, and R. Piché, "Indoor positioning using WLAN coverage area estimates," in *2010 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, September 2010.

[4] A. F. Molisch, *Wireless Communications*, 2nd ed. Wiley - IEEE, January 2011.

[5] T. S. Rappaport, *Wireless communications: Principles and practice, 2nd edition*. Prentice Hall PTR, 2008, ch. Mobile Radio Propagation: Large-Scale Path Loss.

[6] M. Hata, "Empirical formula for propagation loss in land mobile radio services," *IEEE Transactions on Vehicular Technology*, vol. 29, no. 3, pp. 317–325, August 1980.

[7] A. Kotanen, M. Hännikäinen, H. Leppäkoski, and T. Hämäläinen, "Positioning with IEEE 802.11b wireless LAN," in *14th IEEE Proceedings on Personal, Indoor and Mobile Radio Communications, 2003. PIMRC 2003.*, vol. 3, September 2003, pp. 2218–2222.

[8] A. Bose and C. H. Foh, "A practical path loss model for indoor WiFi positioning enhancement," in *2007 6th International Conference on Information, Communications Signal Processing*, December 2007, pp. 1–5.

[9] S. Shrestha, E. Laitinen, J. Talvitie, and E. S. Lohan, "RSSI channel effects in cellular and WLAN positioning," in *Proceedings of the 9th Workshop on Positioning, Navigation and Communication 2012 (WPNC'12)*, March 2012.

[10] J. S. Liu, *Monte Carlo Strategies in Scientific Computing*. Springer, 2001.

[11] N. Sirola, "Closed-form algorithms in mobile positioning: Myths and misconceptions," in *Proceedings of the 7th Workshop on Positioning, Navigation and Communication 2010 (WPNC'10)*, March 2010, pp. 38–44.

[12] X. Li, "RSS-based location estimation with unknown pathloss model," *IEEE Transactions on Wireless Com-*

*munications*, vol. 5, no. 12, pp. 3626–3633, December 2006.

[13] A. Ruhe, "Accelerated Gauss-Newton algorithms for nonlinear least squares problems," *BIT Numerical Mathematics*, vol. 19, no. 3, pp. 356–367, 1979.

[14] Å. Björck, *Numerical methods for least squares problems*. Society for Industrial and Applied Mathematics, 1996.

[15] L. Koski, R. Piché, V. Kaseva, S. Ali-Löytty, and M. Hännikäinen, "Positioning with coverage area estimates generated from location fingerprints," in *Proceedings of the 7th Workshop on Positioning, Navigation and Communication 2010 (WPNC'10)*, March 2010, pp. 99–106.

[16] V. Honkavirta, T. Perälä, S. Ali-Löytty, and R. Piché, "A Comparative Survey of WLAN Location Fingerprinting Methods," in *6th Workshop on Positioning, Navigation and Communication, 2009. WPNC 2009.*, March 2009, pp. 243–251.

[17] Y. Bar-Shalom, R. X. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation, Theory Algorithms and Software*. John Wiley & Sons, 2001.