Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

SPLL: Simultaneous Probabilistic Localization and Learning

Bruno Betoni Parodi * **Andrei Szabo** * **Joachim Bamberger** *
Joachim Horn **

* *Siemens AG, Corporate Technology, Information and Communications, CT IC 4, Otto-Hahn-Ring 6, 81730 Munich, Germany*
** *Helmut-Schmidt-University / University of the Federal Armed Forces Hamburg, Department of Electrical Engineering, Institute for Control Engineering, Holstenhofweg 85, 22043 Hamburg, Germany*

Abstract: Indoor localisation systems based on existent radio communication networks often make use of received signal strength (RSS) as measured feature. In order to achieve a good accuracy such systems have a huge payload in the called calibration phase, where many labelled measurements are collected and used to build a representative feature map. The present paper introduces a new algorithm based on previous works from the same authors, where the calibration phase is avoided by unsupervised online learning, during the operational phase of the system. Using probabilistic localisation and non-parametric density estimation, the new approach uses unlabelled measurements to learn a feature map, having as start only a rough initial model. Simulations with artificial generated data and with real measurements validate the introduced algorithm.

Keywords: Learning theory; Bayesian methods; Stochastic system identification.

1. INTRODUCTION

Indoor localisation is still a defying subject among the localisation tasks. While for outdoor environments the Global Positioning System or GPS (and soon also Galileo) is the well established and popular solution, for indoor environments there are yet many different systems and solutions, each of them with promising features and some drawbacks. GPS is satellite based and uses difference of propagation time from radio signals to locate users. That is why it fails to operate with the same accuracy in indoor environments, where the complex setups of corridors, floors, walls, and doors attenuate and deflect the satellite signals beyond the system capabilities to adjust itself.

Many indoor systems achieve accuracy of a few metres, as the best results achieved with GPS for outdoor scenarios. However, these systems usually need many proprietary sensors populating the area where a user must be located, as in Priyantha et al. (2000) and Ni et al. (2003) or they need many received signal strength (RSS) measurements prior to system start to build a radio map, as in Bahl and Padmanabhan (2000), Roos et al. (2002) and Brunato and Battiti (2005). The major drawbacks in such cases are the low scalability and high implementation costs. Propagation time is also used in some indoor systems, where it achieves comparable performance as with systems using RSS, but using proprietary sensors and under assumptions, as for example line-of-sight (LOS), that limit generalisations, as in Oppermann et al. (2004) and Pahlavan et al. (2002).

The main advantage of using existent communication systems for localisation is that no extra hardware is needed to be installed (as is the case with special sensors and tag based systems), since the RSS measurement is a standard feature of these systems. However, this advantage is opposed by the costly calibration phase that must be accomplished before the system start. In Betoni Parodi et al. (2006) a localisation system was proposed, which addressed the cost reduction of this calibration effort. The Simultaneous Localisation and Learning (SLL) proposed the use of a rough initial model to start the system, with only a few information, like the base station (BS) positions. The system starts with accuracy equivalent to Cell-ID systems and, through successive localisation queries combined with an iterative learning algorithm, achieves accuracy equivalent as using calibration measurements, and that avoiding the effort of collecting these measurements. The system uses only unlabelled samples, that is, plain RSS measurements without information about the true location where the sample is obtained, contrasting with the required labelled samples for usual calibration. In Betoni Parodi et al. (2007) some important properties and conditions for successful use of SLL were proven.

The SLL is based on the Self Organising Maps or SOMs (see Kohonen (1990)) and records as feature the mean value of RSS in the feature map, performing localisation with a simple pattern matching approach called nearest neighbours (NN). As the natural evolution of SLL, the Simultaneous Probabilistic Localisation and Learning (SPLL) is presented here as the main contribution of this paper. As the SLL finds a parallel with SOMs, the same

could be said of SPLL and the Bayesian SOM (BSOM), as in Yin and Allinson (1997).

First a brief introduction about SOMs is presented, then the SPLL algorithm is described, with examples in one dimensional (1D) environments using simulated data and real measurements, which validate the algorithm.

2. INTRODUCTION ON SELF ORGANISING MAPS

SOMs are a special class of neural networks, which are based on competitive learning. In a SOM, the neurons are placed at the nodes of a 1D or 2D lattice, known also as latent space. The neurons are selectively adapted to various input patterns in the course of a unsupervised competitive learning process. The locations of the winning neurons in latent space become ordered in such a way that a meaningful coordinate system for different input patterns is created over the lattice, Kohonen (1990). A SOM is therefore characterised by the formation of a topological map where the input patterns are mapped from the input space into the latent space, preserving the intrinsic statistical features contained in the input patterns, Haykin (1998).

The principal feature of Kohonen's SOM is the automatic and adaptive mapping of signals from input space into the latent space, using these signals to perform a parametric regression over the neurons, fitting them to the distribution of the input samples. This mapping forms clusters in the latent space, which preserve the topological relations of the data in the input space. This is the called self organisation.

The learning is achieved by performing iteratively three steps in addition to the initialisation: competition, cooperation and adaptation. During the initialisation the synaptic weights in the neural network are randomly set, if no other initialisation is specified.

In the competitive step a winning neuron c with the weight vector $m_c = [m_{c1}, \cdots, m_{cn}]$ in the n dimensional input space is selected such that it has the smallest cost with respect to a given input feature vector $\xi = [\xi_1, \cdots, \xi_n]^{\mathrm{T}}$. The cost is usually calculated using some distance measure in input space between ξ and all weights m_i, that is, $c = \arg\min\{d(\xi - m_i)\}$, with d as the distance measure (as the Euclidean distance for most practical applications) and the index i going through all neurons in the lattice. The winning neuron c will be the centre for the adaptation process.

The cooperation determines which neurons will be adapted together with the winning neuron c. A neighbourhood function $h_{ci}(k)$, dependent on the discrete time step k, is used to find the neuron i close to the winner c and to weigh it accordingly with the distance to the winner in the lattice. A typical choice for the neighbourhood function at 1D problems is the constant function, set to a constant $\alpha(k)$ for the winner and for an equal number of neighbours, forward and backward (usually just 2 neighbours are taken). For 2D or 3D maps the Gaussian function is usually chosen, so that:

$$h_{ci}(k) = \alpha(k) \cdot \exp\left(-\left(\frac{d_{ci}}{2 \cdot \sigma(k)}\right)^2\right), \qquad (1)$$

where $\alpha(k)$ is the learning rate, $\sigma(k)$ is the effective width of the topological neighbourhood, both dependent on k. d_{ci} is the distance in latent space from neuron i to neuron c at the centre. The adaptation law, given by

$$m_i(k+1) = m_i(k) + h_{ci}(k) \cdot \big(\xi(k) - m_i(k)\big), \qquad (2)$$

ensures that the response of the winning neuron to the subsequent application of a similar input pattern is enhanced, Haykin (1998).

The adaptation process consists of two phases: the self-organising or ordering phase and the convergence phase. In the ordering phase the topological ordering of the weight vectors takes place. During this phase the learning rate and the neighbourhood area should decrease. The neighbourhood area goes from complete coverage to a few neurons or even to the winning neuron itself. In the convergence phase the fine tuning of the feature map takes place in order to provide an accurate statistical quantification of the input space. The learning rate should stay constant or it could decay exponentially, Haykin (1998).

The Kohonen algorithm is very resilient to a complete mathematical study, according to Cottrell et al. (1994). Thorough analyses could be achieved only for the 1D case in a linear network. The results in higher dimensions are only partial.

3. DESCRIPTION OF THE ALGORITHM

The SPLL is an iterative algorithm based on the improvement of a feature map using unlabelled samples as its predecessor, the SLL. The feature recorded in the SPLL feature map is a probability density function (pdf), in contrast with only the mean RSS value at the SLL feature map. A new measurement is used to locate a user with probabilistic localisation, and this location estimate defines an area where the feature map is updated. The pdf is a natural choice for a description of a continuous random variable, however, as a discretisation step is invariably introduced when implementing the algorithm, the formally right description should be of a probability mass function (pmf) for discrete random variables. Throughout this paper, the description of the algorithm will use pdfs as base, and the implementation with pmfs.

The following sections introduce the main aspects that together form the SPLL, concluding with the algorithm itself:

3.1 Probabilistic Localisation

The following scenario exemplifies a typical example found in localisation systems: An area is covered by N BSs. A feature map is constructed at Q selected positions x_q, and at each of these positions some sufficient amount of measurements are taken in order to build some feature, for example, the mean RSS value, the mean and variance, or the whole distribution. At some unknown position x_{M} a measurement vector $\boldsymbol{p}_{\mathbf{M}}$, with dimension N, one for each BS, is obtained. The localisation task concerns in how to combine the information contained in the measurement $\boldsymbol{p}_{\mathbf{M}}$ with the information recorded in the feature map so that some location estimate x is retrieved, as close as possible to the true location x_{M}.

The probabilistic localisation deals with the task of determining $\Pr(x|\boldsymbol{p}_\mathbf{M})$ the probability that a user is at the position x given the feature $\boldsymbol{p}_\mathbf{M}$, here a RSS measurement. The solution to this problem lies on calculating the posterior probability over all possible Q locations on a discrete feature map. That is usually accomplished using Bayes rule:

$$\Pr(x|\boldsymbol{p}_\mathbf{M}) = \frac{\Pr(\boldsymbol{p}_\mathbf{M}|x)\Pr(x)}{\sum_{q=1}^{Q}\Pr(\boldsymbol{p}_\mathbf{M}|x_q)\Pr(x_q)}, \tag{3}$$

where the conditional probability $\Pr(\boldsymbol{p}_\mathbf{M}|x)$ is retrieved from labelled samples, that is, measurements at known locations, which are used to build the pdf $g(p,x)$ as the information recorded in the feature map; and $\Pr(x)$ is usually set as a constant, assuming it is uniform distributed if there is no prior information from where a user can be. The denominator at (3) acts as a normaliser.

If the measured feature $\boldsymbol{p}_\mathbf{M}$ has dimension $N \neq 1$, then $\Pr(\boldsymbol{p}_\mathbf{M}|x)$ can be calculated as:

$$\Pr(\boldsymbol{p}_\mathbf{M}|x) = \prod_{n=1}^{N}\Pr(p_{\mathrm{M,n}}|x), \tag{4}$$

if measurements from distinct BSs are assumed independent. In this case a unidimensional pdf $g_n(p,x)$ is needed for each BS.

A position estimate is usually retrieved from the probability densities either using the Maximum Likelihood (ML):

$$x = \arg\max_{x_q}\Pr(\boldsymbol{p}_\mathbf{M}|x_q), \tag{5}$$

where, as self explained by its name, the most likely location is determined; or using the Minimum Mean Squared Error (MMSE), where

$$x = \mathrm{E}[x_\mathrm{M}|\boldsymbol{p}_\mathbf{M}] = \sum_{q=1}^{Q} x_q \cdot \Pr(x_q|\boldsymbol{p}_\mathbf{M}), \tag{6}$$

is the best estimation for x, which minimises $\mathrm{E}[(x_\mathrm{M}-x)^2]$, being $\mathrm{E}[\cdot]$ the expected value. For its intrinsic robustness, the latter is the method chosen for this work.

3.2 Non-parametric Density Estimation

In order to obtain $\Pr(p_\mathrm{M}|x)$ from the measurements there are many possibilities to consider: parametric approaches are often found in the literature under the assumption that the real distribution can be approximated by some known model, usually describing the pdfs in terms of Gaussian distributions or even Gaussian mixtures. Non-parametric approaches, as the histogram and the kernel based, are then used when the pdf cannot be described in terms of a set of parameters. The SPLL uses a variant of the kernel based approach, known also as Parzen method and described thoroughly in Webb (2002).

According to the kernel method, a pdf for a set of k observations can be estimated using:

$$g_k(p) = \frac{1}{k}\cdot\sum_{i=1}^{k} f_i(p), \tag{7}$$

being $f_i(p)$ the kernel function, assumed to be a pdf by its own, i.e., $f_i \geq 0$ and $\int f_i \,\mathrm{d}p = 1$.

The kernel method takes all contributions from 1 to k equally, placing one kernel for every observation on the set. Permutations on the sequence of observations have no effect at the estimated density, which is not adequate for SPLL learning yet.

Thus, a small modification on the kernel method is proposed. It is presented in a recursive form, which is more appropriate for a recursive process as the SPLL:

$$g_{k+1}(p,x) = \frac{g_k(p,x) + f_{k+1}(p)}{\int \left(g_k(p,x) + f_{k+1}(p)\right)\,\mathrm{d}p}, \tag{8}$$

where $f_{k+1}(p)$ is the kernel function, which does not need to be constrained to $\int f_k \,\mathrm{d}p = 1$, as the integral at the denominator normalises the kernel contribution at $k+1$. This, combined with the fact that $g(p,x) \geq 0$ assures that (8) describes in fact a pdf.

The kernel function f_k may have almost any shape, but the choice of the kernel reflects directly at the estimated pdf. For example, a rectangular kernel, with discontinuity at its borders, will generate also a discontinuous pdf. The kernel at time k is centred at $p_{\mathrm{M},k}$ and for the SPLL a natural shape choice was the Gaussian function:

$$f_k = \kappa \cdot \exp\left(-\left(\frac{p - p_{\mathrm{M},k}}{\phi}\right)^2\right), \tag{9}$$

where κ controls the amplitude and ϕ the width of the kernel. As defined in Betoni Parodi et al. (2007) for the SLL, κ is the learn rate and ϕ is often mentioned as smoothing parameter or bandwidth in the literature about kernel method. In fact, ϕ controls how smooth the learned pdf will be: if ϕ is too wide, then the pdf will be very smooth and fine details as peaks in the pdf will never be learned. On the other hand, if ϕ is too narrow the pdf will be very spiky and rough. The condition imposed for the kernel in Webb (2002), forcing it to have unity area, is here relaxed since (8) has already a normalising factor. Noteworthy is that f_k at SPLL is a function of power p and not anymore of position x as $f_{\mathrm{c},k}$ with SLL.

For further analysis the recursive form of (8) is not very helpful. The following theorem introduces a closed formulation for (8):

Theorem 1. Calling the denominator of (8) as A_{k+1} for the sake of readability, it is possible to write the recursive formula at (8) in closed form as:

$$g_k(p,x) = \frac{g_0(p,x)}{\prod_{j=1}^{k} A_j} + \sum_{i=1}^{k}\frac{f_i(p)}{\prod_{j=i}^{k} A_j}, \tag{10}$$

where $g_0(p,x)$ is the pdf chosen for the system initialisation, at time $k=0$.

Proof. For the sake of readability the dependencies of p and x from g and f were suppressed. The first and second iterations of SPLL can be directly written as:

$$g_1 = \frac{g_0 + f_1}{A_1},$$
$$g_2 = \frac{g_0 + f_1}{A_1 \cdot A_2} + \frac{f_2}{A_2},$$

which are conform with (8).

Assuming that the closed form of the update law (10) at step k is true, an inductive proof is constructed applying (10) on the recursive form (8) at step $k+1$:

$$g_{k+1} = \frac{g_k + f_{k+1}}{A_{k+1}} = \frac{\frac{g_0}{\prod_{j=1}^{k} A_j} + \sum_{i=1}^{k} \frac{f_i}{\prod_{j=i}^{k} A_j} + f_{k+1}}{A_{k+1}}$$
$$= \frac{g_0}{A_{k+1} \cdot \prod_{j=1}^{k} A_j} + \frac{1}{A_{k+1}} \cdot \sum_{i=1}^{k} \frac{f_i}{\prod_{j=i}^{k} A_j} + \frac{f_{k+1}}{A_{k+1}}$$

Moving A_{k+1} to inside the product on the first and second fractions gives:

$$g_{k+1} = \frac{g_0}{\prod_{j=1}^{k+1} A_j} + \sum_{i=1}^{k} \frac{f_i}{\prod_{j=i}^{k+1} A_j} + \frac{f_{k+1}}{A_{k+1}},$$

and finally the last fraction can be moved into the sum, so that the expression for step $k+1$ is obtained in the same form as the expression for step k:

$$g_{k+1} = \frac{g_0}{\prod_{j=1}^{k+1} A_j} + \sum_{i=1}^{k+1} \frac{f_i}{\prod_{j=i}^{k+1} A_j} \quad (11)$$

□

The area defined by A_{k+1} is composed by two terms, as seen in (8): the area under the pdf $g_k(p, x)$, which was normalised at the previous step $k-1$ consequently having value 1, and the area of the actual $f_{k+1}(p)$. Hence, $A_k > 1 \forall k$.

Making $k \to \infty$ it is possible to observe what happens with (10) when the number of iterations grows. As $A_k > 1$ it follows that $\lim \prod_{j=1}^{k} A_j = \infty$, which cancels the first term of (10). As a consequence of this, the initial model $g_0(p, x)$ is entirely replaced by the second term in (10), which depends only on the measurements. An important property observed is that older terms of f_i have less weight on the sum than the newer ones. This capability to forget old values while learning new ones is a necessary condition so tat changes in the radio environment can be tracked. This is the major difference between the non-parametric density estimation developed here and the methods found at the literature, where all measurements are equally weighted to compose the pdf.

3.3 Algorithm Main Steps

Now that the main aspects of SPLL were introduced, they are here grouped to compose the main algorithm.

Initialisation Although the initial model $g_0(p, x)$ will eventually vanish, as explained in section 3.2, the choice of it should reflect some plausibility. For example, pdfs with higher mean RSS value at BS positions and decay with distance. Since the algorithm operates blindly over the actual feature map, a bad choice, as uniform distributions overall, can make the algorithm fail to learn.

Localisation As justified at section 3.1, the localisation technique chosen is the MMSE. With a new measurement p_{M}, an estimate for location x is retrieved using (6). This estimate lays inside the cartesian boundaries defined by the feature map positions x_q, with $q = 1, \cdots, Q$.

Neighbour Selection Once a location estimate x is retrieved, then it must be selected which positions in the feature map will effectively be updated.

Defining the Euclidean distance between the location estimate x and the q^{th} feature map position x_q as:

$$d_q = \sqrt{(x - x_q)^{\mathrm{T}}(x - x_q)}, \quad (12)$$

and defining the maximum distance from x where a update is made as ψ, then every position x_q for which $d_q < \psi$ will be updated.

Feature Map Update At the x_q positions inside the ψ radius the feature map is updated. The update uses the non-parametric density estimation as explained in section 3.2, and considering the discrete time as $k+1$, then the update law follows (8) applied at x_q.

Since the update is made at several positions using the same measurement p_{M}, a small modification is made at f so that the positions x_q closer to the centre x are privileged, with a higher update amplitude: now κ in (9) is made dependent on d_q, so that

$$\kappa = \kappa(d_q) = \frac{\psi - d_q}{\psi} \kappa_{\max}, \quad (13)$$

where $\kappa_{\max}$ is the maximum value for κ.

The control parameters of SPLL κ, ϕ, and ψ can and should be made variable with time. Right after initialisation they should start with great magnitude such that the model is brought from its false initialisation condition to the vicinity of the true model, and then the parameters should be made smaller such that the fine details of the model can also be learned.

4. SIMULATED 1D EXAMPLE

In order to show the learning capabilities of the SPLL the following 1D simulation example was set:

$x_q = q - 1$ with $q = \{1, \cdots, 21\}$ in metres. Only one BS is considered and placed at $x_1 = 0$m. The propagation profile is Gaussian with standard deviation $\sigma = 5$ and the mean μ decays linearly with distance, such that $\mu_q = p_0 - \gamma x_q$, where the output power $p_0 = -20$dBm and the attenuation factor $\gamma = 2$dB/m. Additionally, at $x_{11} = 10m$ a discontinuity of -20dBm was placed, which could be the effect of a thick wall. In this way, the pdf at x_q is $\mathcal{N}(-20 - 2x_q, 5)$ if $q < 11$ and $\mathcal{N}(-40 - 2x_q, 5)$ if $q \geq 11$. The pdfs were recorded as pmfs in a discrete space with field strength range from 0 to $-100dBm$ with step of 1dB. For each position 1000 samples were generated from the correspondent distributions and used as measurements.

The initial model for the feature map was set such that the profile is also Gaussian but following $\mathcal{N}(-60 - 2x_q, 5), \forall q$. This ensures that the initial model totaly mismatches the true measurements, but still attending to physical plausibility, since the highest μ is placed where the BS is. At Figs. 1 and 2 both the true model that generated the measurements and the initialisation can be observed.

The control parameters were left fixed so that $\kappa_{\max} = 0.05$, $\phi = 3$, and $\psi = 3$. This makes it possible to observe the convergence potential of SPLL without composition

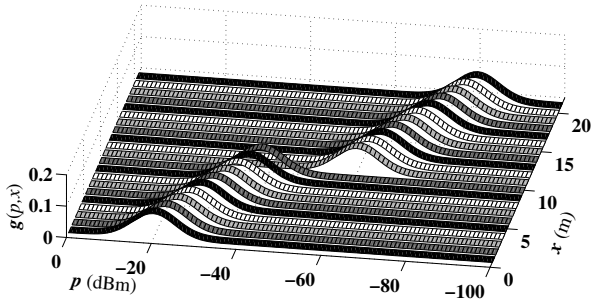

Fig. 1. Simulated example: true model

of many effects. A faster and better convergence could be obtained with variable parameters but this was not the aim of this example. A global mean error between the true model pdf that generated the measurements $g_{\text{true}}(p,x)$, discretised as pmf, and the actual pmf $g_k(p,x)$ recorded in the feature map at time k was defined as:

$$e_{\text{pdf},k} = \frac{\sum_{q=1}^{Q}\sqrt{\sum_{r=1}^{R}\left(g_k(p_r,x_q)-g_{\text{true}}(p_r,x_q)\right)^2}}{Q}, \quad (14)$$

where R is the number of discrete field strength sampling points for the pmfs.

The experiment was run going from x_1 to x_{21} and back to x_1 100 times, resulting in 4200 iterations. At each position x_q one measurement was randomly selected from the 1000 possible. Figure 3 shows the evolution of $e_{\text{pdf},k}$.

$e_{\text{pdf},k}$ starts falling immediately and achieves its minimum after 1500 iterations. The final feature map after 4200 iterations is displayed in Fig. 4.

It is noteworthy to compare Figs. 1 and 4 and verify that the learned pmf was brought by the SPLL to the position of the true model and that even the discontinuity was learned at the right position.

5. REAL WORLD 1D EXAMPLE

For this experiment WLAN measurements were taken in a corridor with length of 31.2m, with a BS at $x_1 = 0m$ and step of 1.2m between the feature map positions $x_q = (q-1)\cdot 1.2$ with $q = \{1,\cdots,27\}$. Each position was measured 100 times and the corridor has a thick metal door at $x_{14} = 15.6m$, which was kept shut during all experiment.

The feature map was initialised using $\mathcal{N}(-60-x_q,5), \forall q$. As there is no a priori knowledge of the true model, a

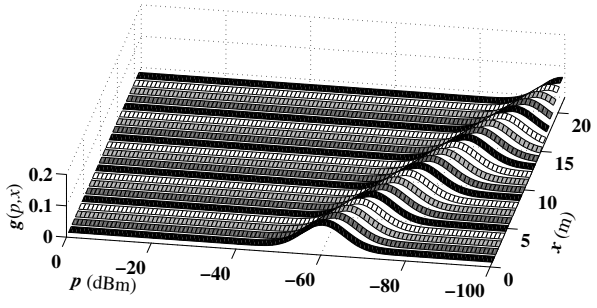

Fig. 2. Simulated example: initialisation

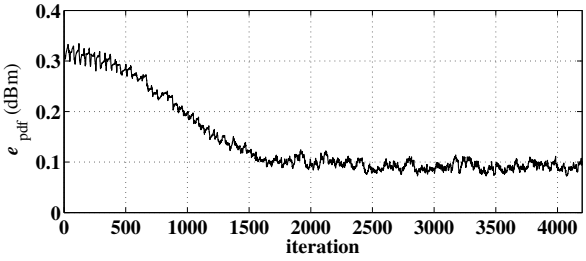

Fig. 3. Simulated example: pdf error

representation of its pdf is achieved using common non-parametric density estimation, but only for comparison with the learned feature map. The mismatch between the measurements and the initial model are presented in Figs. 5 and 6, respectively. Noticeable are the discontinuity due to the door and also the difference in the wideness of the true and the initial feature map pmfs.

The experiment was run going from x_1 to x_{27} and back to x_1 300 times, resulting in 16200 iterations. At each position x_q one measurement was randomly selected from the 100 possible. The control parameters $\kappa_{\max}$, ϕ, and ψ have the same value as in the simulated example. Figure 7 shows the evolution of $e_{\text{pdf},k}$.

$e_{\text{pdf},k}$ starts falling after a short rise and achieves its minimum after 7000 iterations. It takes somewhat longer to achieve the minimum as in the simulated example, as well as the minimum error itself is bigger. This can be explained as the measurement profile here is more complex, and the pdfs used for comparison are only a representation of the unknown true pdfs. Nevertheless, the feature map goes from its initially false start state to a much better representation of the measurements as Fig. 8 shows. In this plot the learned feature map after 16200 iterations is displayed.

Once again it is possible to verify that the learned pmf was brought by the SPLL to the position of the true model and that even the discontinuity was learned at the right position, as Figs. 5 and 8 show.

6. CONCLUSION

In this paper, the SPLL, a new algorithm for online learning of feature maps for localisation systems, was introduced. The core of the algorithm is based on probabilistic localisation using discretisation of pdfs as recorded feature in the feature map and on non-parametric density estima-

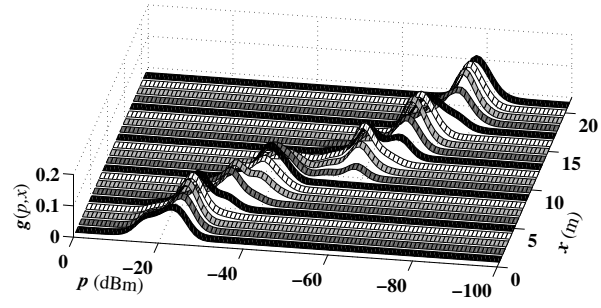

Fig. 4. Simulated example: learned pdf at $k = 4200$

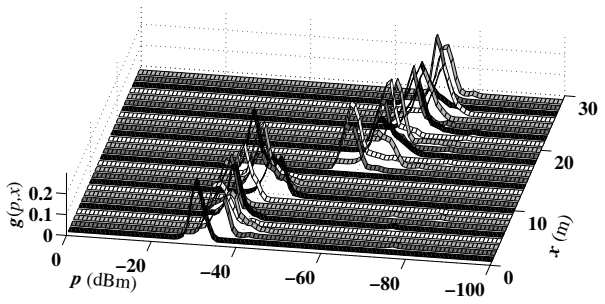

Fig. 5. Real world example: true model

tion as ways of recursive learning of these pdfs. The system starts with a rough initial model and unlabelled samples are used as information source for the learning algorithm, bringing the feature map to a state that better describes the measurements than the initialisation.

The algorithm was performed in an artificial 1D test environment and with a real world 1D scenario, using measurements taken in a corridor. On both cases the feature map was improved, going from its initial state to one that better describes the measurements. That was verified using a global error variable that was made smaller with increasing iterations of the algorithm. A discontinuity imposed on the measurements was also learned by the SPLL, indicating its versatility.

REFERENCES

Paramvir Bahl and Venkata N. Padmanabhan. RADAR: An in-building RF-based user location and tracking system. In *IEEE INFOCOM 2000*, pages 775–784, Tel Aviv, Israel, March 2000.

Bruno Betoni Parodi, Henning Lenz, Andrei Szabo, Hui Wang, Joachim Horn, Joachim Bamberger, and Dragan Obradovic. Initialization and online-learning of RSS maps for indoor / campus localization. In *PLANS 2006*, pages 164–172, San Diego - CA, USA, April 2006.

Bruno Betoni Parodi, Henning Lenz, Andrei Szabo, Joachim Bamberger, and Joachim Horn. Algebraic and statistical conditions for the use of SLL. In *ECC 07*, Kos, Greece, July 2007.

Mauro Brunato and Roberto Battiti. Statistical learning theory for location fingerprinting in wireless LANs. *Computer Networks*, 47(6):825–845, 2005.

Marie Cottrell, Jean-Claude Fort, and Gilles Pagés. Two or three things that we know about the Kohonen algorithm. In *ESANN'94*, pages 235–244, Brussels, Belgium, April 1994.

Simon Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1998. ISBN 0132733501.

Teuvo Kohonen. The self-organizing map. *Proceedings of the IEEE*, 78:1464–1480, 1990.

Lionel M. Ni, Yunhao Liu, Yiu Cho Lau, and Abhishek P. Patil. LANDMARC: Indoor location sensing using active RFID. In *IEEE International Conference on Pervasive Computing and Communications2003*, volume 10, pages 407–415, Fort Worth - TX, USA, 2003.

I. Oppermann, A. Karlsson, and H. Linderbäck. Novel phase based, cross-correlation position estimation technique. In *IEEE ISSSTA 2004*, pages 340–345, September 2004.

K. Pahlavan, X. Li, and J. P. Makela. Indoor geolocation science and technology. *IEEE Communications Magazine*, 40(2):112–118, February 2002.

Nissanka B. Priyantha, Anit Chakraborty, and Hari Balakrishnan. The Cricket location-support system. In 6^{th} *ACM International Conference on Mobile Computing and Networking*, pages 32–43, Boston - MA, USA, August 2000.

Teemu Roos, Petri Myllymäki, Henry Tirri, Pauli Misikangas, and Juha Sievänen. A probabilistic approach to WLAN user location estimation. *International Journal of Wireless Information Networks*, 9(3):155–164, July 2002.

Andrew R. Webb. *Statistical Pattern Recognition*. Wiley, 2nd edition, 2002. ISBN 0470845147.

Hujun Yin and Nigel M. Allinson. Comparison of a Bayesian SOM with the EM algorithm for gaussian mixtures. In *WSOM97*, pages 118–123, Espoo, Finland, June 1997.

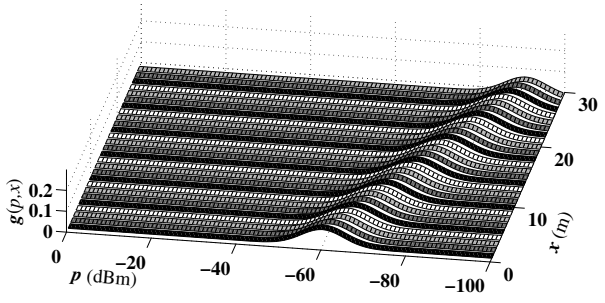

Fig. 6. Real world example: initialisation

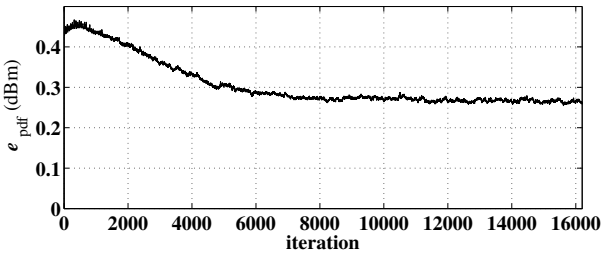

Fig. 7. Real world example: pdf error

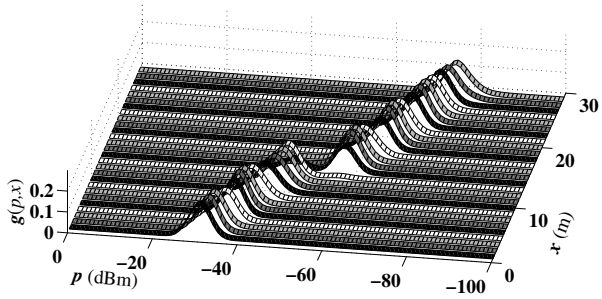

Fig. 8. Real world example: learned pdf at $k = 16200$ and initialisation