ONLINE EXPECTATION MAXIMIZATION ALGORITHM TO SOLVE THE SLAM PROBLEM

S. Le Corff, G. Fort, E. Moulines

LTCI, TELECOM ParisTech-CNRS, 46 rue Barrault, 75634 Paris Cedex 13. France

ABSTRACT

In this paper, a new algorithm - namely the *onlineEM-SLAM* - is proposed to solve the simultaneous localization and mapping problem (SLAM). The mapping problem is seen as an instance of inference in latent models, and the localization part is dealt with a particle approximation method. This new technique relies on an online version of the Expectation Maximization (EM) algorithm: the algorithm includes a stochastic approximation version of the E-step to incorporate the information brought by the newly available observation. By linearizing the observation model, the stochastic approximation part is reduced to the computation of the expectation of additive functionals of the robot pose. Therefore, each iteration of the *onlineEM-SLAM* both provides a particle approximation of the distribution of the pose, and a point estimate of the map. This online variant of EM does not require the whole data set to be available at each iteration. The performance of this algorithm is illustrated through simulations using sampled observations and experimental data.

Index Terms— SLAM, Sequential Monte Carlo methods, additive functionals, Expectation Maximization.

1. INTRODUCTION

The SLAM problem arises when a robot evolves in an unknown environment without knowing its pose. The first solutions of this problem involved the estimation of the joint distribution of the robot pose (or of the full path) and of the map given all the controls and observations up to the current time; see [2]. This problem can be solved analytically assuming the underlying transition and observation models are linear and gaussian. It is based on the Kalman filter when the models are linear, or the extended Kalman filter (EKF) when the models have to be linearized. In this case, the posterior distribution of the robot pose and of the map is gaussian and thus the covariance matrix contains information about the correlation between the robot pose and the landmarks. Unfortunately, the use of a Taylor expansion to linearize both the transition and the observation models is known to lead to filter divergence; see [1, 8].

This work was supported by the French National Research Agency, under the program ANR-07 ROBO 0002.

The fastSLAM algorithm [12] is a particular instance of particle filtering taking advantage of the conditional independence of the landmarks given the observations, the controls, and the robot trajectories. For each particle representing a robot path, the position of each observed landmark is updated using standard EKF steps. This algorithm allows to maintain several map hypothesis (one for each particle). However, the presence of a static parameter (the map) in the state space prevents the particle approximation from converging uniformly in time; see [5].

To overcome this difficulty, [11] introduced a marginal SLAM algorithm: the key difference with the previous algorithms relies on the nature of the map which is treated as a parameter used to drive a latent data model. This parameter is estimated by a recursive maximum likelihood procedure, solved in practice by a stochastic gradient algorithm; see also [10]. Consequently, this marginal algorithm provides a point estimate of the map and a particle approximation of the marginal posterior distribution of the robot pose at each time.

The new algorithm we present in this paper also treats the map as a parameter in a latent model. The parameter is estimated in the maximum likelihood sense and this estimation is dealt with an online EM-type algorithm.

The paper is organized as follows. In section 2 we present the robot dynamical model and its observation model. Section 3 is devoted to the *onlineEM-SLAM* algorithm. This new technique is illustrated in section 4 and compared to the marginal SLAM.

2. FRAMEWORK

Let $\mathbf{x_t} = \begin{pmatrix} x_{t,1} & x_{t,2} & x_{t,3} \end{pmatrix}^T$ be the robot pose, where $\begin{pmatrix} x_{t,1} & x_{t,2} \end{pmatrix}^T$ stands for robot's cartesian coordinates and $x_{t,3}$ is its angular coordinate. The robot evolves in a 2-dimensional unknown environment which will be represented by a feature based map formed by a set of landmarks. Controls are denoted by $\mathbf{u_t} = \begin{pmatrix} v_t & \psi_t \end{pmatrix}^T$ where ψ_t stands for the robot's heading direction and v_t its velocity. The state transition model gives the robot pose at time t given its previous pose at time $t-1$ and the noisy controls $(\hat{v}_t, \hat{\psi}_t)$:

$$\mathbf{x_t} = f(\mathbf{x_{t-1}}, \hat{v}_t, \hat{\psi}_t) \ , \tag{1}$$

where $\begin{pmatrix}\hat{v}_t & \hat{\psi}_t\end{pmatrix}^T$ is a 2-dimensional Gaussian distribution with mean $\mathbf{u_t}$ and known covariance matrix Q. We denote by $m(\mathbf{x_t}|\mathbf{x_{t-1}}, \mathbf{u_t})$ the density function of the state transition model which does not depend on the parameter (i.e the map).

Let $\boldsymbol{\theta} = \left(\theta_{i,j}\right)_{1\leq i\leq 2, 1\leq j\leq q}$ be all the landmarks in the map; the vector $\theta_{\cdot,j}$ represents the cartesian coordinates of the $j-$th landmark. The total number of landmarks q is assumed to be known and, in addition, the association between observations and landmarks is done without error (this hypothesis is relaxed in the experiment with true data, see Section 4). At time t, the set of observed landmarks is denoted by $\mathcal{A}_t$. For any $i \in \mathcal{A}_t$, the observation $\mathbf{y_{t,i}} \in \mathbb{R}^2$ of landmark i at time t is modeled by

$$\mathbf{y_{t,i}} = h(\mathbf{x_t}, \theta_{\cdot,i}) + \delta_{t,i} \tag{2}$$

where h is defined by

$$h(\mathbf{x}, \tau) = \begin{pmatrix} \sqrt{(\tau_1 - \mathbf{x}_1)^2 + (\tau_2 - \mathbf{x}_2)^2} \\ \arctan \frac{\tau_2 - \mathbf{x}_2}{\tau_1 - \mathbf{x}_1} - \mathbf{x}_3 \end{pmatrix} . \tag{3}$$

The noise vectors $\left(\delta_{t,i}\right)_{t, i\in\mathcal{A}_t}$ are i.i.d Gaussian mixtures with two components $\mathcal{N}\left(0, \sigma_0^2 R\right)$, $\mathcal{N}\left(0, \sigma_1^2 R\right)$ and weights (ω_0, ω_1). We denote by $\mathbf{I_t} = \left(I_{t,i}\right)_{i\in\mathcal{A}_t}$ the latent variables specifying the identity of the mixture component of each observation at time t. R, σ_0^2 and σ_1^2 are assumed to be known. In the sequel we denote by $\mathbf{Z_t} = (\mathbf{X_t}, \mathbf{I_t})$ the extended latent variable. When the map is equal to $\boldsymbol{\theta}$, the likelihood of $\mathbf{y_t}$ given $\mathbf{z_t}$ is denoted by $g_\theta(\mathbf{y_t}|\mathbf{z_t})$.

3. ONLINE EM SLAM

3.1. Online EM algorithm for curved exponential family

The EM algorithm, introduced in [7], is a popular iterative technique to perform maximum likelihood estimation in latent models, when the observations are known in batch. Given a set of observations $\mathbf{y_{1:T}} = (\mathbf{y_1}, \cdots, \mathbf{y_T})$, EM produces iteratively a sequence of parameter estimates such that the (normalized) log-likelihood of the observations $\theta \mapsto T^{-1}\ell(\mathbf{y_{1:T}}; \theta)$ is non-decreasing over iterations. Each iteration is decomposed into an Expectation step (E-step) and a Maximization step (M-step). The E-step involves the computation of the conditional expectation of the complete data log-likelihood p_θ

$$Q_T(\theta, \theta') = \frac{1}{T}\mathbb{E}_{\theta'}\left[\log p_\theta(\mathbf{Z_{1:T}}, \mathbf{y_{1:T}})|\mathbf{y_{1:T}}\right] ,$$

where $\mathbb{E}_{\theta'}\left[\cdot|\mathbf{y_{1:T}}\right]$ denotes the conditional expectation of the latent data given the observations and the current value θ' of the parameter. The M-step updates the parameter as a maximum of the function $\theta \mapsto Q_T(\theta, \theta')$. When processing large data sets or data streams, EM becomes impractical due to the requirement that the whole data be available at each iteration of the algorithm. Hence, EM algorithm is not designed for the SLAM framework since the data sets are too large. In [3] (resp. in [13]), the authors present an online version of EM for hidden Markov models (HMM) with finite state space (resp. with general state space). In these extensions, since the number of observations increases at each iteration, it is expected that the limiting points of these algorithms are the stationary points of the limiting normalized log-likelihood $\theta \mapsto \lim_{T\to+\infty} T^{-1}\ell(\mathbf{y_{1:T}}; \theta)$. Insights for such an asymptotic result are given in [3] for HMM with finite state-space.

These extensions (and more generally any EM-type procedures) are mostly useful in case where the complete data likelihood belongs to the exponential family. In that case, for any t, there exist functions $\mathsf{S_t}$ and Ξ such that

$$Q_t(\theta, \theta') = \left\langle \mathbb{E}_{\theta'}\left[\mathsf{S_t}(\mathbf{Z_{1:t}}, \mathbf{y_{1:t}})|\mathbf{y_{1:t}}\right], \Xi(\theta)\right\rangle .$$

The E-step thus reduces to the computation of a single expectation. In HMM, the sufficient statistics S_t is of the form

$$\mathsf{S_t}(\mathbf{Z_{1:t}}, \mathbf{y_{1:t}}) = t^{-1} \sum_{s=1}^{t} \mathsf{S}(\mathbf{Z_{s-1}}, \mathbf{Z_s}, \mathbf{y_s}) . \tag{4}$$

By standard properties of the conditional expectation, we have

$$\mathbb{E}_{\theta'}\left[\mathsf{S_t}(\mathbf{Z_{1:t}}, \mathbf{y_{1:t}})|\mathbf{y_{1:t}}\right] = \mathbb{E}_{\theta'}\left[\mathcal{S}_{t,\theta'}(\mathbf{Z_t})|\mathbf{y_{1:t}}\right] \tag{5}$$

where, by (4) an properties of HMM

$$\begin{aligned}\mathcal{S}_{t,\theta'}(\mathbf{Z_t}) = \frac{1}{t}\mathbb{E}_{\theta'}\left[\mathsf{S}(\mathbf{Z_{t-1}}, \mathbf{Z_t}, \mathbf{y}_t)|\mathbf{Z_t}, \mathbf{y_{1:t-1}}\right] \\ + \left(1 - \frac{1}{t}\right)\mathbb{E}_{\theta'}\left[\mathcal{S}_{t-1,\theta'}(\mathbf{Z_{t-1}})|\mathbf{Z_t}, \mathbf{y_{1:t-1}}\right] .\end{aligned} \tag{6}$$

Equations (5) and (6) show that the E-step necessitates *(a)* the filtering and backward retrospective (i.e. the conditional distribution of $\mathbf{Z_{t-1}}$ given $(\mathbf{Z_t}, \mathbf{y_{1:t-1}})$) distributions at time t and, *(b)* a recursive computation of $\mathcal{S}_{t,\theta'}$. Except in trivial models, neither the filtering distribution nor the recursive formula (6) are available explicitly and these computations have to be approximated. The filtering and the backward retrospective distributions at time t are replaced by particle-type approximations; see e.g. [4] and [6]. In (6), the quantities $\mathcal{S}_{t,\theta'}(x)$ are updated using a stochastic approximation step. As a conclusion, an iteration of the online EM is defined by an *online E-step* which is divided into a sequential Monte Carlo step and a stochastic approximation step; and a M-step. The main feature of this online algorithm is that there is no need to store the data since they are used sequentially.

3.2. Application to the SLAM problem

We now describe more precisely the steps of the algorithm. For the stochastic approximation step, we choose for each of the landmark i a stepsize sequence $(\gamma_{t,i})_{t\geq 1}$ such that $\sum_t \gamma_{t,i} = +\infty$ and $\sum_t \gamma_{t,i}^2 < +\infty$ [9]; for example,

$\gamma_{t,i} = c_i/t^\alpha$ for $\alpha \in (1/2, 1]$. In the sequel, we denote by $\widehat{\theta}$ the current estimate of the parameter.

Linearization step Under the model assumptions described in Section 2, the density of the state transition does not depend upon θ so that the complete data log-likelihood at time t is equal to

$$-\sum_{i=1}^{q} \frac{1}{2t} \sum_{s=1}^{t} \mathbf{1}_{i\in\mathcal{A}_s} \left[\mathbf{y_{s,i}} - h(\mathbf{X_s}, \theta_{.,i})\right]^T \times \sigma_{I_{s,i}}^{-2} R^{-1} \left[\mathbf{y_{s,i}} - h(\mathbf{X_s}, \theta_{.,i})\right] \quad (7)$$

up to an additive term that does not depend upon θ. Due to the expression of h (see (3)), the complete data likelihood does not belong to the exponential family. At each iteration, the E-step is thus preceded by a *linearization step* of the observation model. For any $i \in \mathcal{A}_s$, $h(\mathbf{x_s}, \theta_{.,i})$ is replaced by a first order Taylor expansion:

$$h(\mathbf{x_s}, \widehat{\theta}_{.,i}) + \nabla_\theta h(\mathbf{x_s}, \widehat{\theta}_{.,i})\, (\theta_{.,i} - \widehat{\theta}_{.,i}). \quad (8)$$

Plugging (8) in (7) implies that (7) is replaced by a quadratic function in θ so that *(i)* the complete data likelihood is in the exponential family and *(ii)* the update of the parameter consists in computing the unique maximum of a quadratic function.

Online E-step The linearization step leads to a particular form of the quantity $Q_t(\theta, \hat{\theta}) \stackrel{\text{def}}{=} \sum_{i=1}^{q} Q_{t,i}(\theta_{.,i}, \hat{\theta})$, where:

$$Q_{t,i}(\theta_{.,i}, \hat{\theta}) \stackrel{\text{def}}{=} \left\langle \mathbb{E}_{\hat{\theta}}\left[\mathsf{S}_{\mathrm{t,i}}(\mathbf{Z_{1:t}}, \mathbf{y_{1:t}}) | \mathbf{y_{1:t}}\right], \Xi_i(\theta_{.,i}) \right\rangle \quad (9)$$

and, for the model given in Section 2:

$$\mathsf{S}_{\mathrm{t,i}}(\mathbf{Z_{1:t}}, \mathbf{y_{1:t}}) = t^{-1} \sum_{s=1}^{t} 1_{i\in\mathcal{A}_s} \mathsf{S}_i(\mathbf{Z_s}, \mathbf{y_{s,i}})\ .$$

Denote by $\mathcal{S}_{t,i,\hat{\theta}}(\mathbf{Z_t}) \stackrel{\text{def}}{=} \mathbb{E}_{\hat{\theta}}\left[\mathsf{S}_{\mathrm{t,i}}(\mathbf{Z_{1:t}}, \mathbf{y_{1:t}}) | \mathbf{Z_t}, \mathbf{y_{1:t-1}}\right]$. As described in Section 3.1, the online E-step for exponential family combines a Sequential Monte Carlo step and a stochastic approximation step; see algorithm 1. New poses are sampled using the kernel π and $\rho_{t,i}^k$ is the particle approximation of $\mathcal{S}_{t,i,\hat{\theta}}(x)$ when $x = \xi_{\mathbf{t}}^{\mathbf{k}}$; see [13] for a $\mathcal{O}(N^2)$ implementation of this approximation. Finally, the coefficients $1/t$ and $1 - 1/t$ are replaced by $\gamma_{t,i}$ and $1 - \gamma_{t,i}$, additional flexibility improves the stochastic approximation scheme; see [9].

Algorithm 1 Online E-Step of the OnlineEM-SLAM

Require: Controls $\mathbf{u_t}$, observations $\mathbf{y_t}$, current map estimate $\widehat{\theta}$, weighted samples $\left\{\left(\xi_{t-1}^\ell, \omega_{t-1}^\ell\right)\right\}_{\ell=1}^N$ approximating the filtering distribution of the robot pose and $\{\rho_{t-1,i}^\ell\}_{\substack{1\le\ell\le N\\ 1\le i\le q}}$.

Online E-Step
for $\ell = 1$ to N **do**

Draw J^ℓ in $\{1, \cdots, N\}$ with probabilities $\{\omega_{t-1}^j\}_{j=1}^N$ and sample independently $\xi_{\mathbf{t}}^\ell \sim \pi(\cdot|\xi_{\mathbf{t-1}}^{\mathbf{J}^\ell}, \mathbf{u_t}, \mathbf{y_t})$ and for all $i \in \mathcal{A}_t$, $I_{t,i}^\ell \sim \mathcal{B}(r)$[1].

Set $\mathcal{I}_t^\ell \stackrel{\text{def}}{=} \sum_{i\in\mathcal{A}_t} I_{t,i}^\ell$ and

$$\omega_t^\ell \propto \frac{g_{\hat{\theta}}(\mathbf{y_t}|\xi_{\mathbf{t}}^\ell, I_t^\ell) m(\xi_{\mathbf{t}}^\ell|\xi_{\mathbf{t-1}}^{\mathbf{J}^\ell}, \mathbf{u_t})}{\pi(\xi_{\mathbf{t}}^\ell|\xi_{\mathbf{t-1}}^{\mathbf{J}^\ell}, \mathbf{u_t}, \mathbf{y_t})} \times \left(\frac{\omega_1}{r}\right)^{\mathcal{I}_t^\ell} \left(\frac{1-\omega_1}{1-r}\right)^{|\mathcal{A}_t| - \mathcal{I}_t^\ell}\ .$$

end for
Stochastic approximation step
For any $i \in \{1, \ldots, q\}$, for any $k \in \{1, \ldots, N\}$, set

$$\rho_{t,i}^k = \mathbf{1}_{i\in\mathcal{A}_t} \gamma_{t,i} \mathsf{S}_i(\xi_{\mathbf{t}}^{\mathbf{k}}, I_{t,i}^k, y_{t,i}) + (1-\gamma_{t,i}) \frac{\sum_{\ell=1}^N \omega_{t-1}^\ell m(\xi_{\mathbf{t}}^{\mathbf{k}}|\xi_{\mathbf{t-1}}^\ell, \mathbf{u_t}) \rho_{t-1,i}^\ell}{\sum_{\ell=1}^N \omega_{t-1}^\ell m(\xi_{\mathbf{t}}^{\mathbf{k}}|\xi_{\mathbf{t-1}}^\ell, \mathbf{u_t})}\ .$$

Approximate $\mathbb{E}_{\hat{\theta}}\left[\mathsf{S}_{\mathrm{t,i}}(\mathbf{Z_{1:t}}, \mathbf{y_{1:t}}) | \mathbf{y_{1:t}}\right]$ by $\sum_{k=1}^N \omega_t^k \rho_{t,i}^k$.

M-step Based on the equation (9), the maximum of $\theta \mapsto Q_t(\theta, \hat{\theta})$ on $\mathbb{R}^{2q}$ is obtained from the maximization of $\tau \mapsto Q_{t,i}(\tau, \hat{\theta})$ on $\mathbb{R}^2$ for any $i \in \{1, \cdots, q\}$ so that the update of each landmark can be done independently of the others. In addition, each function $\tau \mapsto Q_{t,i}(\tau, \hat{\theta})$ is quadratic and its maximization is explicit.

4. EXPERIMENTS

4.1. Simulated data

This new algorithm is compared to the marginal SLAM algorithm of [11]. In this section, the function f in (1) is given by

$$f(\mathbf{x_{t-1}}, \hat{v}_t, \hat{\psi}_t) = \mathbf{x_{t-1}} + \begin{pmatrix} \hat{v}_t d_t \cos(x_{t-1,3} + \hat{\psi}_t) \\ \hat{v}_t d_t \sin(x_{t-1,3} + \hat{\psi}_t) \\ \hat{v}_t d_t \frac{\sin(\hat{\psi}_t)}{B} \end{pmatrix}\ ,$$

where d_t is the time period between two successive pose and $B = 1.5$m is the robot wheelbase. Starting with a *true* map

[1] $\mathcal{B}(r)$ denotes the Bernoulli distribution with parameter r.

θ^*, observations are sampled by setting: $\omega_0 = 0.8$, $\sigma_0 = 1$, $\sigma_1 = 5$ and $R = \begin{pmatrix} \sigma_r^2 & \rho \\ \rho & \sigma_b^2 \end{pmatrix}$, where $\sigma_r = 0.5$m, $\sigma_b = \frac{\pi}{60}$rad and $\rho = 0.01$. The robot path was sampled using $Q = \mathrm{diag}(\sigma_v^2, \sigma_\phi^2)$ where $\sigma_v = 0.5\mathrm{m.s}^{-1}$ and $\sigma_\psi = \frac{\pi}{60}$rad. All the simulations were performed using $N = 100$ particles. For the SMC step, $\xi_\mathbf{t}^\mathbf{k}$ were sampled from the prior kernel $m(\cdot|\mathbf{X}_{\mathbf{t-1}}^{\mathbf{J^k}}, \mathbf{u_t})$ and $I_{t,i}$ from a Bernoulli distribution with parameter ω_0. Finally, the sequences of steps for the stochastic approximation were set to $\gamma_{t,i} = 1/t^{0.8}$ for any $i \in \{1, \cdots, q\}$. For each run the estimated path (equal to the weighted mean of the particles) and the estimated map at the end of the loop ($T = 1626$) are stored. Figure 1 represents the mean estimated path and the mean map over 50 independent Monte Carlo runs. Using the same data, Figure 2 presents

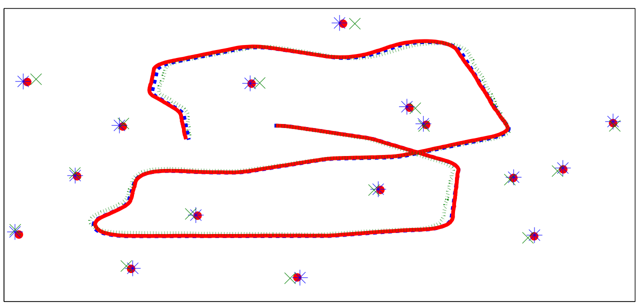

Fig. 1: Map and path estimates with simulated data given by the *OnlineEM-SLAM* algorithm (dashed line and stars) and the marginal SLAM (doted line and crosses). The true path (bold line) and the true landmark locations (dots) are also represented.

boxplots (over the 50 Monte Carlo runs) for the estimation of the error on the robot x-coordinate (difference between the true and the estimated pose) at different time steps.

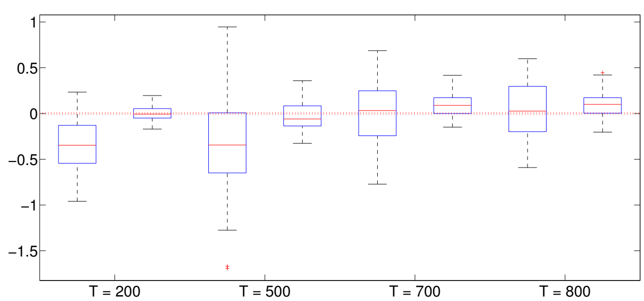

Fig. 2: Error on the robot x-coordinate estimation at different time step with the *OnlineEM-SLAM* algorithm [right] and the marginal SLAM [left].

4.2. Experimental data

Figure 3 shows the performance of the *OnlineEM-SLAM* algorithm with true experimental data. We tried the algorithm with the car park data set[2] which is a landmark based SLAM situation. In this context the association process is not assumed to be known. Then, each time a new observation is available, its likelihood is computed for each landmark in the current map and in a neighborhood of the observation (following the same procedure as in this data set). If all likelihoods are lower than a given threshold, a new landmark is created. Otherwise, this observation is associated to the landmark corresponding to the largest likelihood. The algorithm is started with an empty map and landmarks are created with the first set of observations. In Figure 3 the estimated path and the estimated map (with $N = 100$ particles) at the end of the run ($T = 5565$) is represented.

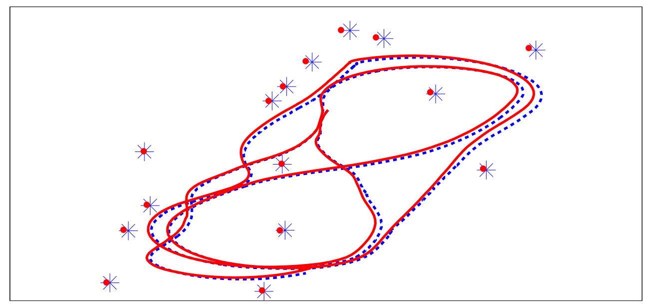

Fig. 3: Map and path estimates using the car park datasets. The estimated path (dashed line) and the estimated landmark positions (stars) given by the *OnlineEM-SLAM* algorithm are compared to the true data (bold line and dots).

5. REFERENCES

[1] T. Bailey, J. Nieto, J. Guivant, M. Stevens, and E. Nebot. Consistency of the EKF-Slam algorithm. In *IEEE International Conference on Intelligent Robots and Systems*, pages 3562–3568, 2006.

[2] W. Burgard, D. Fox, and S. Thrun. *Probabilistic robotics*. Cambridge, MA:MIT Press, 2005.

[3] O. Cappé. Online EM algorithm for hidden Markov models. *To appear in J. Comput. Graph. Statist*, 2011.

[4] O. Cappé, E. Moulines, and T. Rydén. *Inference in Hidden Markov Models*. Springer, 2005.

[5] D. Crisan and A. Doucet. A survey of convergence results on particle filtering methods for practitioners. *IEEE Trans. Signal Process.*, 50(3):736–746, 2002.

[6] P. Del Moral. *Feynman-Kac Formulae. Genealogical and Interacting Particle Systems with Applications*. Springer, 2004.

[7] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. Roy. Statist. Soc. B*, 39(1):1–38 (with discussion), 1977.

[8] S.J. Julier and J.K. Uhlmann. A counter example to the theory of simultaneous localization and map building. In *IEEE international conference on robotics and automation*, volume 4, pages 4238–4243. Citeseer, 2001.

[9] H. J. Kushner and G. G. Yin. *Stochastic Approximation and Recursive Algorithms and Applications*, volume 35. Springer, New York, 2nd edition, 2003.

[10] F. Le Gland and L. Mevel. Recursive estimation in HMMs. In *Proc. IEEE Conf. Decis. Control*, pages 3468–3473, 1997.

[2]Thanks to `http://www.cas.kth.se/SLAM`

[11] R. Martinez-Cantin. Active map learning for robots: insights into statistical consistency. PhD Thesis, 2008.

[12] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In *Proceedings of the Sixteenth IJCAI*, Mexico, 2003.

[13] P. Del Moral, A. Doucet, and S. S. Singh. Forward smoothing using sequential monte carlo. arXiv:1012.5390v1, 2010.