Online EM algorithm for jump Markov systems

Carsten Fritsche, Emre Özkan and Fredrik Gustafsson

Original Publication:

Carsten Fritsche, Emre Özkan and Fredrik Gustafsson, Online EM algorithm for jump Markov systems, 2012, 15th International Conference on Information Fusion (FUSION), 2012, 1941-1946.

Online EM Algorithm for Jump Markov Systems

Carsten Fritsche, Emre Özkan, and Fredrik Gustafsson
Department of Electrical Engineering
Linköping University
SE-581 83 Linköping, Sweden

***Abstract*—The Expectation-Maximization (EM) algorithm in combination with particle filters is a powerful tool that can solve very complex problems, such as parameter estimation in general nonlinear non-Gaussian state space models. We here apply the recently proposed online EM algorithm to parameter estimation in jump Markov models, that contain both continuous and discrete states. In particular, we focus on estimating process and measurement noise distributions being modeled as mixtures of members from the exponential family.**

I. Introduction

The Expectation-Maximization (EM) algorithm is one of the most popular methods for Maximum Likelihood (ML) estimation [1]. It has been applied to a wide range of practical problems in different fields such as statistics, biology and signal processing, and it is often preferred over other numerical optimization methods, due to its numerical stability and ease of implementation, see for instance [2], [3], [4]. The classic EM algorithm given in [1] is formulated in batch (or offline) form, i.e. it uses a (possibly large) number of observations to iteratively estimate the unknown parameters. A comprehensive treatment of the general estimation problem in state space models using the offline EM algorithm is given in [5]. However, in many real-time applications, restrictive memory requirements and/or limiting processing power do not allow to store and process large datasets.

Recent developments focused on online EM algorithms, in which the observations are processed only once and never stored. An online version of the EM algorithm for hidden Markov models (HMMs) with a finite number of states and observations has been proposed in [6]. This idea has been extended to generalized HMMs with possibly continuous observations in [7]. The problem of jointly estimating the state and fixed model parameters in general (possibly nonlinear non-Gaussian) state-space models using the online EM algorithm has been addressed in [8]. Here, sequential Monte Carlo (SMC) filtering approximations is proposed to numerically approximate the EM recursions. The algorithm has been further modified in [9] to solve the simultaneous localization and mapping problem. Three different online EM type algorithms that aim at maximizing split-data likelihoods are proposed in [10], to solve the problem of fixed model parameter estimation in general state-space models. The joint estimation of continuous- and discrete-valued states together with fixed model parameters in jump Markov systems is rather unexplored. In [11], an ML estimator is derived using the reference probability method, in order to estimate the transition probabilities in jump Markov linear systems. Here, an EM procedure is utilized for maximizing the corresponding likelihood function. In order to avoid the exponential increase in the number of statistics of the optimal EM algorithm, interacting multiple model-type approximations are introduced.

In this paper, the online EM algorithm of [8] is further extended to apply to general state space models with Markovian switching structure. Besides of estimating the state of the system, we are interested in estimating the transition probabilities of the Markov chain as well as unknown model parameters. The proposed online EM algorithm is implemented using a particle filter-based method. In an accompanying paper, we present the online EM algorithm to a nonlinear state space model with noise processes being mixtures in the exponential family, where a discrete state selects the mixture mode. This discrete state is memoryless, and can be seen as a special case of jump Markov systems. However, the memoryless property enables a powerful marginalization procedure, that significantly improves the performance.

The rest of this paper is organized as follows: In Section II, the general jump Markov system model is introduced. In Section III, the EM algorithm basics are briefly reviewed and the proposed approach is introduced. An SMC implementation of the proposed online EM algorithm is provided in Section IV. In Section V, the important special case of estimating noise parameters in jump Markov Gaussian systems is investigated. The performance of the proposed approach is illustrated by means of simulations in Section VI. Finally, conclusions are drawn in Section VII.

II. System Model

Consider the following discrete-time jump Markov nonlinear system

$$x_t = f_\theta(x_{t-1}, r_t, v_t) \quad (1a)$$
$$y_t = h_\theta(x_t, r_t, w_t) \quad (1b)$$

where $y_t \in {}^{n_y}$ is the measurement vector at discrete time t, $x_t \in {}^{n_x}$ is the state vector and f and h are arbitrary nonlinear mapping functions. The jump Markov system is generally parametrized by unknown model parameters θ which is denoted by f_θ and h_θ, respectively. The mode variable r_t denotes a discrete-time Markov chain with M states. The Markov chain is assumed to be time-homogeneous with transition probability matrix Π_θ, whose elements are denoted as $\pi_\theta(r_t = j, r_{t-1} = i) \stackrel{\Delta}{=} \pi_{ij},\ i,j = 1,\ldots,M$. The noise vectors $v_t \in {}^{n_v}$ and $w_t \in {}^{n_w}$ are assumed mutually independent white processes with known probability density functions that may depend on r_t and/or θ.

In the following, we introduce the augmented state vector $z_t = [x_t^T, r_t]^T$, and let $z_{0:k} = [z_0^T, \ldots, z_k^T]^T$ and $y_{0:k} = [y_0^T, \ldots, y_k^T]^T$ denote the collection of augmented states and measurement vectors up to time k. We further introduce the conditional likelihood $l_\theta(y_t|z_t)$, which can be determined from (1b) and the augmented state transition density $p_\theta(z_t|z_{t-1}) = g_\theta(x_t|x_{t-1}, r_t)\, \pi_\theta(r_t|r_{t-1})$, where $g_\theta(x_t|x_{t-1}, r_t)$ can be determined from (1a). With a slight abuse of notation we further define $\sum_{r_{0:k}} \int dx_{0:k} \triangleq \sum \int dz_{0:k}$, and we note that $\{\cdot\}$ stands for mathematical expectation.

III. Expectation-Maximization Algorithm

The EM algorithm is an iterative approach that computes ML estimates of unknown parameters θ in probabilistic models involving unobserved variables (a.k.a. latent variables). The idea of the EM algorithm is to separate the original ML estimation problem into two linked problems, denoted as the Expectation Step (E-Step) and Maximization step (M-Step), each of which is hopefully easier to solve than the original problem. Thus, it is especially useful in settings where ML estimates are hard to obtain.

A. Batch EM Algorithm

Suppose that a set of K observations $y_{0:K}$ is available. Then, the original batch ML estimation problem can be formulated as

$$\hat{\theta}^{\mathrm{ML}} = \arg\max_\theta \log p_\theta(y_{0:K}) \tag{2}$$

For jump Markov systems, direct evaluation of (2) is difficult, since the computation of the likelihood $p_\theta(y_{0:K})$ involves the computation of high-dimensional integrals, which are generally not tractable analytically.

The key idea of the EM algorithm is to treat $y_{0:K}$ as incomplete data and to introduce a latent variable for which the joint (or complete-data) likelihood is available. The EM algorithm then solves iteratively for θ that maximizes the expected log-likelihood of the complete-data. For jump Markov systems, the latent variables are chosen as $z_{0:K} = [x_{0:k}^T, r_{0:k}^T]^T$, so that the complete-data likelihood is given by

$$p_\theta(z_{0:K}, y_{0:K}) = \prod_{t=0}^{K} p_\theta(z_t, y_t|z_{t-1}) \tag{3}$$

with $p_\theta(z_0, y_0|z_{-1}) \triangleq p_\theta(z_0, y_0)$ and $p_\theta(z_t, y_t|z_{t-1}) = l_\theta(y_t|z_t)\, p_\theta(z_t|z_{t-1})$. In the following, we focus on the complete-data sufficient statistics formulation of the EM algorithm [3], [8].

It is assumed that the density $p_\theta(z_t, y_t|z_{t-1})$ belongs to the exponential family of distributions, given by

$$p(z_t, y_t|z_{t-1}) = C \cdot \exp\{\langle \psi(\theta), s(z_t, z_{t-1}, y_t)\rangle - A(\theta)\}, \tag{4}$$

where $C \triangleq b(z_t, y_t, z_{t-1})$ denotes a function independent of θ, $\langle \cdot, \cdot \rangle$ denotes the inner product, $\psi(\theta)$ is the natural parameter, $s(z_t, z_{t-1}, y_t)$ is the complete-data sufficient statistic and $A(\theta)$ denotes the log-partition function. We further assume that the M-Step yields a unique (closed-form) solution for θ, which is denoted by $\bar{\theta}(\cdot)$. Then, the $(m+1)$-th iteration of the batch EM algorithm applied to the data $y_{0:K}$ can be written in terms of complete-data sufficient statistics as shown in Algorithm 1. We note that the initial term $\frac{1}{K} \log p(z_0, y_0)$ from the normalized complete-data log-likelihood has been omitted in (5) for notational convenience. In online estimation, its contribution is vanishing with K and thus can be safely ignored.

Algorithm 1 Generic Batch EM Algorithm

E-Step:

$$S_{m+1} = \frac{1}{K} {}_{p_{\theta_m}(z_{0:K}|y_{0:K})}\left\{\sum_{t=1}^{K} s(z_t, z_{t-1}, y_t)\right\}. \tag{5}$$

M-Step:

$$\theta_{m+1} = \bar{\theta}(S_{m+1}) = \arg\max_\theta [\langle \psi(\theta), S_{m+1}\rangle - A(\theta)]. \tag{6}$$

B. Online EM Algorithm

The batch EM algorithm estimates the latent variable $z_{0:K}$ together with θ at each time step by processing all available observations $y_{0:K}$. In many applications, however, constraints on the memory do not permit to store and process the whole observation sequence. In these situations, a recursive procedure, that is based on processing only the current observation y_k and that never stores the whole sequence is highly desirable. In the following, it is shown how the batch EM algorithm can be reformulated into an online EM algorithm [8]. Instead of processing a batch of data $y_{0:K}$, it is assumed that observations $y_{0:k}$ up to the current time step k are available. We further assume that at each time step, only one EM iteration is performed. We define an auxiliary function

$$\rho_k(z_{0:k}) = \frac{1}{k} \sum_{t=1}^{k} s(z_t, z_{t-1}, y_t), \tag{7}$$

such that

$$\int \sum \rho_k(z_{0:k})\, p_{\theta_k}(z_{0:k}|y_{0:k})\, dz_{0:k} = S_k \tag{8}$$

holds. Then, it can be easily shown that $\rho_k(z_{0:k})$ can be updated recursively according to the following formula

$$\begin{aligned}\rho_{k+1}(z_{0:k+1}) \;&=\; \frac{1}{k+1}\, s(z_{k+1}, z_k, y_{k+1}) \\ &\quad + \left(1 - \frac{1}{k+1}\right) \rho_k(z_{0:k}).\end{aligned} \tag{9}$$

This weighted recursion is further modified by introducing a sequence $\{\gamma_k\}_{k \geq 1}$ of decreasing step-sizes which satisfies the stochastic approximation requirement, given by $\sum_{k\geq 1} \gamma_k = \infty$ and $\sum_{k \geq 1} \gamma_k^2 < \infty$. Then, the online EM algorithm at the

Algorithm 2 Generic Online EM Algorithm

Stochastic Approximation E-Step:

$$p_{\hat{\theta}_k}(z_{0:k+1}|y_{0:k+1}) = \frac{p_{\hat{\theta}_k}(z_{k+1}, y_{k+1}|z_k)\, p_{\hat{\theta}_k}(z_{0:k}|y_{0:k})}{\int \sum p_{\hat{\theta}_k}(z_{k+1}, y_{k+1}|z_k)\, p_{\hat{\theta}_k}(z_{0:k}|y_{0:k})\, dz_{0:k+1}}$$

$$\hat{\rho}_{k+1}(z_{0:k+1}) = (1-\gamma_{k+1})\,\hat{\rho}_k(z_{0:k}) + \gamma_{k+1}\, s(z_{k+1}, z_k, y_{k+1})$$

M-Step:

$$\hat{\theta}_{k+1} = \bar{\theta}\left(\int \sum \hat{\rho}_{k+1}(z_{0:k+1})\, p_{\hat{\theta}_k}(z_{0:k+1}|y_{0:k+1})\, dz_{0:k+1}\right)$$

$(k+1)$-th time step can be summarized as shown in Algorithm 2.

IV. SMC Implementation of Online EM Algorithm

A closed-form solution for Algorithm 2 is generally not available. We therefore resort to SMC methods of importance sampling resampling type (a.k.a. particle filters), to numerically approximate the online EM recursions [12], [4], [13]. More precisely, a multiple-model particle filter is proposed to approximate the density $p(z_{0:k+1}|y_{0:k+1})$ by a set of N particles and importance weights $\{z^n_{0:k+1}, w^n_{k+1}\}_{n=1}^N$, such that

$$p_\theta(z_{0:k+1}|y_{0:k+1}) \approx \sum_{n=1}^N w^n_{k+1}\, \delta(z_{0:k+1} - z^n_{0:k+1}) \qquad (10)$$

holds [14]. The importance weights satisfy $w^n_k \geq 0$ and $\sum_i w^n_k = 1$, and can be calculated recursively according to the following formula

$$w^n_{k+1} \propto w^n_k \frac{g_\theta(y_{k+1}|z^n_{k+1})\, p_\theta(z^n_{k+1}|z^n_k)}{q_\theta(z^n_{k+1}|z^n_k, y_{k+1})}, \qquad (11)$$

where $q_\theta(z_k|z_{k-1}, y_k)$ denotes the importance function that might depend on the unknown parameter θ. It is worth noting that $\hat{\rho}_{k+1}$, cf. Algorithm 2, is Monte Carlo approximated as well by a set of N samples $\{\rho^n_k\}_{n=1}^N$. This has the appealing advantage that the complicated integral and summation that appears in the M-Step, see also (8), can be replaced by the simple approximation

$$S_{k+1} \approx \sum_{n=1}^N \rho^n_{k+1} w^n_{k+1}. \qquad (12)$$

The proposed SMC approximation of the online EM algorithm is summarized in Algorithm 3.

V. Jump Markov Gaussian Systems with Unknown Noise Parameters

In the following, we consider jump Markov systems with additive Gaussian noise structures. These models are used in

Algorithm 3 SMC Approximation Of Online EM Algorithm

Initialization:

- For $n = 1, \ldots, N$, initialize the particles $z^n_0 \sim p(z_0, y_0)$, the weights $w^n_0 = 1/N$, the vector of unknown parameters $\hat{\theta}_0$ and set ρ^n_0.

Iterations:

For $k = 0, 1, \ldots,$ do:

- Compute the effective number of samples according to $\hat{N}_{\text{eff}} = 1/\sum_n (w^n_k)^2$.
- If $\hat{N}_{\text{eff}} < N_{\text{th}}$, perform resampling. Take N samples with replacement from the set $\{z^n_k, \rho^n_k\}_{n=1}^N$, where the probability to take sample n is w^n_k. Set $w^n_k = 1/N$ for $n = 1, \ldots, N$.
- For $n = 1, \ldots, N$ draw samples from the importance density
$$z^n_{k+1} \sim q(z_{k+1}|z^n_k, y_{k+1}).$$
- For $n = 1, \ldots, N$ evaluate the importance weights according to
$$w^n_{k+1} \propto w^n_k \frac{g_{\hat{\theta}_k}(y_{k+1}|z^n_{k+1})\, p_{\hat{\theta}_k}(z^n_{k+1}|z^n_k)}{q_{\hat{\theta}_k}(z^n_{k+1}|z^n_k, y_{k+1})}.$$
- Normalize the weights such that $\sum_{n=1}^N w^n_{k+1} = 1$.
- For $n = 1, \ldots, N$ update the auxiliary quantity
$$\rho^n_{k+1} = (1-\gamma_{k+1})\,\rho^n_k + \gamma_{k+1}\, s(z^n_{k+1}, z^n_k, y_{k+1}) \qquad (13)$$
- Determine an estimate of the state vector
$$\hat{x}_{k+1} = \sum_{n=1}^N x^n_{k+1} w^n_{k+1}$$
- Perform parameter estimation
$$\hat{S}_{k+1} = \sum_{n=1}^N \rho^n_{k+1} w^n_{k+1}, \quad \hat{\theta}_{k+1} = \bar{\theta}\left(\hat{S}_{k+1}\right) \qquad (14)$$

a plethora of applications, such as change detection, sensor fault detection or tracking of maneuvering targets in air traffic control, see for instance [15]. The noise parameters as well as the parameters describing the Markov chain are design parameters and are typically chosen prior to deployment. In the following, however, these parameters are assumed to be unknown. The corresponding jump Markov Gaussian system can be described as

$$x_t = f(x_{t-1}, r_t) + v_{\theta,t}(r_t), \qquad (15a)$$
$$y_t = h(x_t, r_t) + w_{\theta,t}(r_t), \qquad (15b)$$

where the noise is distributed according to $v_{\theta,t}(r_t) \sim \mathcal{N}(\mu_{\text{v}}(r_t), \Sigma_{\text{v}}(r_t))$ and $w_{\theta,t}(r_t) \sim \mathcal{N}(\mu_{\text{w}}(r_t), \Sigma_{\text{w}}(r_t))$. Here, it is worth noting that the mapping functions f and h may depend on θ as well, but this is not further considered here. Thus, the unknowns can be collected in θ, which is given by $\theta = \{\theta_{\text{r}}(1,1), \ldots, \theta_{\text{r}}(M,M), \theta_{\text{v}}(1), \ldots, \theta_{\text{v}}(M), \theta_{\text{w}}(1), \ldots, \theta_{\text{w}}(M)\}$, where $\theta_\mathrm{r}(r_t = j, r_{t-1} = i) = \{\pi_{ij}\}$, $\theta_\mathrm{v}(r_t) = \{\mu_\mathrm{v}(r_t), \Sigma_\mathrm{v}(r_t)\}$ and $\theta_\mathrm{w}(r_t) = \{\mu_\mathrm{w}(r_t), \Sigma_\mathrm{w}(r_t)\}$. In order to apply Algorithm 3 to the system given by (15), the sufficient statistics $s(z_k, z_{k-1}, y_k)$ as well as closed-form relationships for the inverse mapping $\bar{\theta}(\cdot)$ have to be further specified. The inverse mapping functions can be found by explicitly evaluating the M-Step according to (6). Similarly to [7], for jump Markov Gaussian systems, the unknown parameters $\hat{\theta}_k$ can be updated according to

$$\hat{\pi}_{ij,k} \;=\; \bar{\theta}(\hat{S}_{\mathrm{r},k}) = \frac{\hat{S}_{\mathrm{r},k}(i,j)}{\sum_{j=1}^{M} \hat{S}_{\mathrm{r},k}(i,j)}, \tag{16a}$$

$$\hat{\mu}_{\mathrm{v},k}(j) \;=\; \bar{\theta}(\hat{S}_{\mathrm{v},k}, \hat{S}_k) = \frac{\hat{S}^{(1)}_{\mathrm{v},k}(j)}{\hat{S}_k(j)}, \tag{16b}$$

$$\hat{\Sigma}_{\mathrm{v},k}(j) \;=\; \bar{\theta}(\hat{S}_{\mathrm{v},k}, \hat{S}_k) = \frac{\hat{S}^{(2)}_{\mathrm{v},k}(j)}{\hat{S}_k(j)} - \hat{\mu}_{\mathrm{v},k}(j)\hat{\mu}^T_{\mathrm{v},k}(j), \tag{16c}$$

$$\hat{\mu}_{\mathrm{w},k}(j) \;=\; \bar{\theta}(\hat{S}_{\mathrm{v},k}, \hat{S}_k) = \frac{\hat{S}^{(1)}_{\mathrm{w},k}(j)}{\hat{S}_k(j)}, \tag{16d}$$

$$\hat{\Sigma}_{\mathrm{w},k}(j) \;=\; \bar{\theta}(\hat{S}_{\mathrm{w},k}, \hat{S}_k) = \frac{\hat{S}^{(2)}_{\mathrm{w},k}(j)}{\hat{S}_k(j)} - \hat{\mu}_{\mathrm{w},k}(j)\hat{\mu}^T_{\mathrm{w},k}(j), \tag{16e}$$

where $\hat{S}_{\mathrm{r},k}$, $\hat{S}_k$, $\hat{S}_{\mathrm{v},k}$ and $\hat{S}_{\mathrm{w},k}$ denote the components of the approximated EM extended sufficient statistics, cf. (14). The corresponding sufficient statistics, necessary to evaluate (13), are given by

$$s^{(1)}(z_k, z_{k-1}, y_k) \;=\; s_\mathrm{r}(r_k = j, r_{k-1} = i), \tag{17a}$$

$$s^{(2)}(z_k, z_{k-1}, y_k) \;=\; s(r_k = j), \tag{17b}$$

$$s^{(3)}(z_k, z_{k-1}, y_k) \;=\; \quad \{r_k = j\} s^{(1)}_\mathrm{v}(x_k, r_k = j, x_{k-1}), \tag{17c}$$

$$s^{(4)}(z_k, z_{k-1}, y_k) \;=\; \quad \{r_k = j\} s^{(2)}_\mathrm{v}(x_k, r_k = j, x_{k-1}), \tag{17d}$$

$$s^{(5)}(z_k, z_{k-1}, y_k) \;=\; \quad \{r_k = j\} s^{(1)}_\mathrm{w}(x_k, r_k = j, y_k), \tag{17e}$$

$$s^{(6)}(z_k, z_{k-1}, y_k) \;=\; \quad \{r_k = j\} s^{(2)}_\mathrm{w}(x_k, r_k = j, y_k), \tag{17f}$$

with

$$s_\mathrm{r}(r_k = j, r_{k-1} = i) \;=\; \quad \{r_k = j, r_{k-1} = i\}, \tag{18a}$$

$$s(r_k = j) \;=\; \quad \{r_k = j\}, \tag{18b}$$

$$s^{(1)}_\mathrm{v}(x_k, r_k = j, x_{k-1}) \;=\; [x_k - f(x_{k-1}, r_k = j)], \tag{18c}$$

$$s^{(2)}_\mathrm{v}(x_k, r_k = j, x_{k-1}) \;=\; [x_k - f(x_{k-1}, r_k = j)][\cdot]^T, \tag{18d}$$

$$s^{(1)}_\mathrm{w}(x_k, r_k = j, y_k) \;=\; [y_k - h(x_k, r_k = j)], \tag{18e}$$

$$s^{(2)}_\mathrm{w}(x_k, r_k = j, y_k) \;=\; [y_k - h(x_k, r_k = j)][\cdot]^T, \tag{18f}$$

where $\{\cdot\}$ stands for an indicator random variable [7]. We note that (17b) is a counter that has been introduced to ensure that the correpsonding auxiliary function, cf. (7), is properly normalized. Furthermore, the indicator random variables guarantee, that the contribution from the particles are assigned to the correct mode of the sufficient statistics.

TABLE I
MODEL PARAMETERS

Parameter	Value	Parameter	Value
π_{11}	0.6	π_{22}	0.8
$\mu_\mathrm{w}(1)$	0	$\mu_\mathrm{w}(2)$	8
$\Sigma_\mathrm{w}(1)$	1	$\Sigma_\mathrm{w}(2)$	2

VI. NUMERICAL EXPERIMENTS

We consider the following modified benchmark model

$$x_t \;=\; \frac{1}{2}x_{t-1} + 25\,\frac{x_{t-1}}{1 + x^2_{t-1}} + 8\,\cos(1.2\,t) + v_t, \tag{19a}$$

$$y_t \;=\; \frac{x_t^2}{20} + w_{\theta,t}(r_t), \tag{19b}$$

where the measurement noise is governed by a 2-state Markov chain. The mode-dependent measurement noise is assumed to be Gaussian distributed, $w_{\theta,t}(r_t = j) \sim \mathcal{N}(\mu_\mathrm{w}(j), \Sigma_\mathrm{w}(j))$, $j = 1, 2$. The mode-dependent mean and variance as well as the transition probabilities of the Markov chain are assumed unknown. Hence, we have $\theta = \{\theta_\mathrm{r}(1,1), \ldots, \theta_\mathrm{r}(2,2), \theta_\mathrm{w}(1), \theta_\mathrm{w}(2)\}$, with $\theta_\mathrm{r}(r_t = j, r_{t-1} = i) = \{\pi_{ij}\}$ and $\theta_\mathrm{w}(r_t = j) = \{\mu_\mathrm{w}(j), \Sigma_\mathrm{w}(j)\}$. The model parameter values used in the simulations are summarized in Table I. The remaining parameters are chosen as $x_0 \sim \mathcal{N}(0,1)$, $r_0 \sim p(r_0) = 0.5$ and $v_k \sim \mathcal{N}(0,1)$.

The online EM algorithm is implemented according to Algorithm 3, where a multiple-model bootstrap particle filter has been used, i.e. $q_{\hat{\theta}_k}(z_k | z_{k-1}, y_k) = p_{\hat{\theta}_k}(z_k | z_{k-1})$. Since the model described in (19) is a special case of the model given by (15), the sufficient statistics and inverse mappings presented in the previous section can be used. The multiple-model particle filter used $N = 1000$ particles and the resampling threshold was set to $N_\mathrm{th} = N/2$. The algorithm was systematically started from the initial parameter values $\hat{\theta}_0 = \{0.5, 0.5, 0.5, 0.5, -2, 2, 6, 4\}$, and the sequences of step-sizes for the stochastic approximation were set to $\gamma_k = (k - k_0)^{-\alpha}$, with $k_0 = 50$ and $\alpha = 0.6$. All simulation results have been obtained from 100 Monte Carlo runs.

In Figs. 1-3, the parameter estimation results are summarized as box and whiskers plots for different time steps. It can be observed that the proposed algorithm is able to efficiently estimate the unknown parameters. As expected, the parameter estimates are biased shortly after the initialization stage and the estimation variance is quite large, which is due to the relative small number of observations that are available to compute ML estimates. However, as the number of time steps increases, this bias disappears and the estimation variance is decreasing as well.

In Figs. 4-6, the corresponding Monte Carlo averaged estimation results for the unknown parameters vs. the time step k are shown. It can be seen that the proposed algorithm quickly converges to the true parameter values and seems not to be affected by any drifts or abrupt changes that may affect the estimation performance. Even though results are not shown here, the proposed approach has been also tested

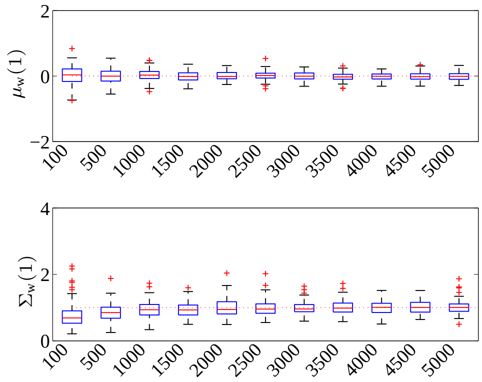

Fig. 1. Box plots for the estimates of $\mu_\mathrm{w}(1)$ and $\Sigma_\mathrm{w}(1)$ over 100 Monte Carlo runs at different time steps.

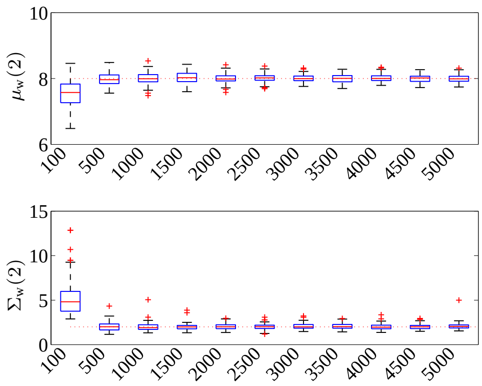

Fig. 2. Box plots for the estimates of $\mu_\mathrm{w}(2)$ and $\Sigma_\mathrm{w}(2)$ over 100 Monte Carlo runs at different time steps.

on data records with 100 000 time steps, where no long-term instabilities or degeneracies have been observed, which affect many other online Monte Carlo-based fixed parameter estimators.

In Fig. 7 the state estimation performance of the proposed online EM algorithm is compared to a clairvoyant multiple-model bootstrap particle filter that knows the model parameters, cf. Table I. Here, the time-averaged RMSE performance in estimating x_t for the two different estimators has been computed for different selected numbers of particles. It can be observed that the clairvoyant estimator always outperforms the online EM-based algorithm as expected. However, as the number of particles increases the performance difference becomes smaller. The figure promises the posibility of achieving the performance of the clairvoyant estimator by increasing the number of particles in the online EM algorithm.

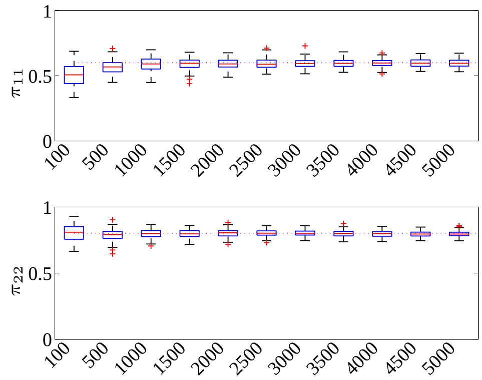

Fig. 3. Box plots for the estimates of π_{11} and π_{22} over 100 Monte Carlo runs at different time steps.

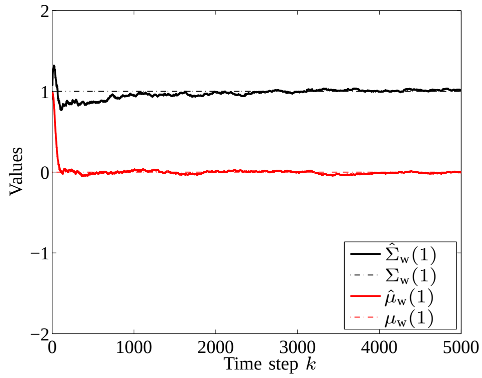

Fig. 4. Monte Carlo averaged estimation results for $\mu_\mathrm{w}(1)$ and $\Sigma_\mathrm{w}(1)$ vs. time step k

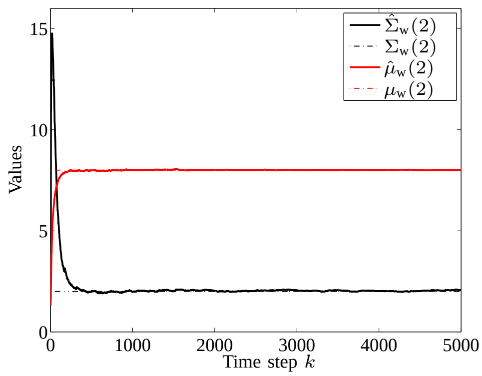

Fig. 5. Monte Carlo averaged estimation results for $\mu_\mathrm{w}(2)$ and $\Sigma_\mathrm{w}(2)$ vs. time step k

VII. Conclusion

We have presented an online EM algorithm for fixed model parameter estimation in jump Markov nonlinear systems. The proposed approach is based on Monte Carlo approximations where the EM recursions are embedded into a multiple-model particle filter to jointly estimate the state and fixed model parameters. The EM recursions are formulated in terms of sufficient statistics, which are updated by a procedure resembling stochastic approximation. The resulting algorithm is simple and its formulation is general such that it can be applied to a wide range of examples. Simulation results show, that the newly proposed algorithm is capable of estimating fixed parameters with good accuracy.

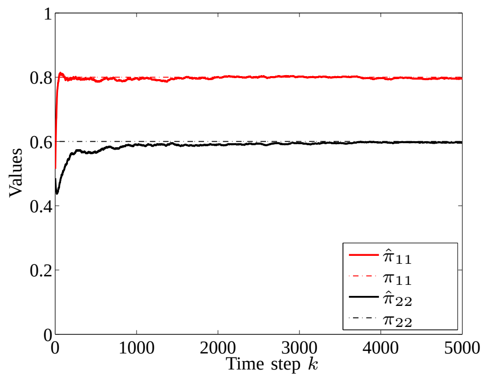

Fig. 6. Monte Carlo averaged estimation results for π_{11} and π_{22} vs. time step k

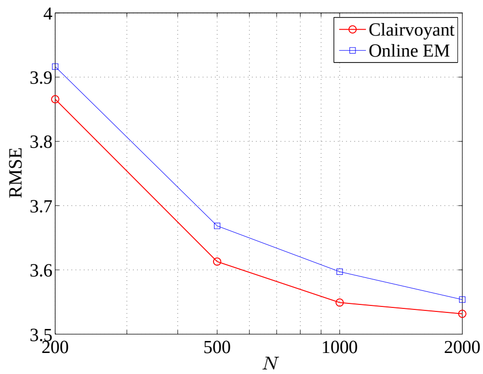

Fig. 7. RMSE of x_t vs. number of particles N for clairvoyant- and online EM-based multiple model particle filter

VIII. Acknowledgment

The authors would like to thank the Linnaeus research environment CADICS, funded by the Swedish Research Council.

References

[1] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977.

[2] T. K. Moon, "The Expectation-Maximization algorithm," *IEEE Signal Process. Mag.*, vol. 13, no. 6, pp. 47–60, 1996.

[3] G. J. McLachlan and T. Krishnan, *The EM algorithm and extensions*, ser. Wiley Series in Probability and Statistics. New York, NY, USA: John Wiley & Sons, Inc., 1997.

[4] O. Cappé, E. Moulines, and T. Rydén, *Inference in Hidden Markov Models*, ser. Springer Series in Statistics. New York, NY, USA: Springer Science + Business Media, LLC, 2005.

[5] T. B. Schön, A. Wills, and B. Ninness, "System identification of nonlinear state-space models," *Automatica*, vol. 47, no. 1, pp. 39–49, Jan 2011.

[6] G. Mongillo and S. Denve, "Online learning with hidden Markov models," *Neural Computation*, vol. 20, no. 7, pp. 1706–1716, 2008.

[7] O. Cappé, "Online EM algorithm for hidden Markov models," *Journal of Computational and Graphical Statistics*, vol. 20, no. 3, pp. 728–749, 2011.

[8] O. Cappé, "Online sequential Monte Carlo EM algorithm," in *Statistical Signal Processing, 2009. SSP '09. IEEE/SP 15th Workshop on*, Cardiff, Wales, Sept. 2009, pp. 37–40.

[9] S. Le Corff, G. Fort, and E. Moulines, "Online Expectation Maximization algorithm to solve the SLAM problem," in *Statistical Signal Processing Workshop (SSP), 2011 IEEE*, Nice, France, Jun., pp. 225 –228.

[10] C. Andrieu and A. Doucet, "Online Expectation-Maximization type algorithms for parameter estimation in general state space models," in *Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03). 2003 IEEE International Conference on*, vol. 6, Kowloon, Hong Kong, Apr. 2003, pp. 69–72.

[11] U. Orguner and M. Demirekler, "Maximum Likelihood estimation of transition probabilities of jump Markov linear systems," *IEEE Trans. Signal Process.*, vol. 56, no. 10, pp. 5093 –5108, Oct. 2008.

[12] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Boston, MA, USA: Artech-House, 2004.

[13] A. Doucet, N. de Freitas, and N. Gordon, Eds., *Sequential Monte Carlo Methods in Practice*. New York, NY, USA: Springer-Verlag, 2001.

[14] S. McGinnity and G. W. Irwin, "Multiple model bootstrap filter for maneuvering target tracking," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 36, no. 3, pp. 1006–1012, 2000.

[15] F. Gustafsson, *Adaptive Filtering and Change Detection*. New York, NY, USA: John Wiley & Sons, 2000.