

**PARTICLE FILTERS FOR SYSTEM IDENTIFICATION OF
STATE-SPACE MODELS LINEAR IN EITHER PARAMETERS
OR STATES¹**

Thomas Schön and Fredrik Gustafsson

*Division of Automatic Control and Communication Systems
Department of Electrical Engineering
Linköpings universitet, SE-581 83 Linköping, Sweden
Tel: +46 13 281373; fax: +46 13 282622
{schon, fredrik}@isy.liu.se*

Abstract: The potential use of the marginalized particle filter for nonlinear system identification is investigated. The particle filter itself offers a general tool for estimating unknown parameters in non-linear models of moderate complexity, and the basic trick is to model the parameters as a random walk (so called roughening noise) with decaying variance. We derive algorithms for systems which are non-linear in either the parameters or the states, but not both generally. In these cases, marginalization applies to the linear part, which firstly significantly widens the scope of the particle filter to more complex systems, and secondly decreases the variance in the linear parameters/states for fixed filter complexity. This second property is illustrated on an example of chaotic model. The particular case of freely parametrized linear state space models, common in subspace identification approaches, is bi-linear in states and parameters, and thus both cases above are satisfied. One can then choose which one to marginalize.

Keywords: System identification, Nonlinear estimation, Recursive estimation, Particle filters, Kalman filters, Bayesian estimation.

1. INTRODUCTION

In this contribution, the particle filter (Doucet *et al.*, 2001a) is applied to some classical system identification problems (Ljung, 1999) based on time-varying parametric state-space models

$$z_{t+1} = f(z_t; \theta_t) + v_t^z, \quad (1a)$$

$$y_t = h(z_t; \theta_t) + e_t, \quad (1b)$$

where $z \in \mathbb{R}^n$ is the state variable, $\theta_t \in \mathbb{R}^d$ is the parameter vector, and $y \in \mathbb{R}^m$ is the output variable. The additive noise terms are assumed to be independent and identically distributed (i.i.d.).

First, we briefly review the problem formulation in (Gustafsson and Hriljac, 2003). By augmenting the state vector with the parameters, $x_t^T = (z_t^T, \theta_t^T)$, and assuming a random walk parameter variation (of which constant parameters is a special case), we get

$$\begin{bmatrix} z_{t+1} \\ \theta_{t+1} \end{bmatrix} = \begin{bmatrix} f(z_t; \theta_t) \\ \theta_t \end{bmatrix} + \begin{bmatrix} v_t^z + w_t^z \\ v_t^\theta + w_t^\theta \end{bmatrix} \quad (2a)$$

$$y_t = h(z_t; \theta_t) + e_t, \quad (2b)$$

where the noises are physical state noise v_t^z , state roughening noise w_t^z , parameter random walk for time-varying parameters v_t^θ and parameter roughening noise w_t^θ . The roughening noise is instrumental in the particle filter to get good performance, and is a second level design parameter. For system identification, $v_t^\theta = 0$ and w_t^θ has a variance decaying to zero, which

¹ This work was supported by the Swedish Research Council.

yields converging parameter estimates. The particle filter recursively approximates the posterior density function $p(X_t|Y_t)$, where $X_t = \{x_0, \dots, x_t\}$, and the approximation converges to the true posterior when the number of particles tends to infinity. The only problem is that the practical limit for 'infinity' depends on the dimension of x_t , that is, the sum of number of parameters θ_t and states z_t . As a very coarse rule of thumb, do not try to use the particle filter for more than five unknowns.

Now, if there is linear substructure available in the model this can be exploited using *marginalization*. Conceptually, marginalization means that the linear states are marginalized out and then we can apply optimal filters for these states and the particle filter is only applied to the truly nonlinear states. In this way, the samples in the particle filter will live in a lower dimensional space. Hence, we will intuitively obtain *more accurate estimates* for a given number of samples, since we use the optimal filters for a part of the state vector. Alternatively, we can apply the particle filter on *more complex models*. These are the practical implications of our contribution.

We will in this contribution consider the two following special cases of (1a):

- (1) The model is affine in the parameters and possibly nonlinear in the states, *i.e.*,

$$f(z_t; \theta_t) = A_t(z_t)\theta_t + f_t^z(z_t) \quad (3)$$

$$h(z_t; \theta_t) = C_t(z_t)\theta_t + h_t^z(z_t). \quad (4)$$

- (2) The model is affine in the states and possibly nonlinear in the parameters, *i.e.*,

$$f(z_t; \theta_t) = A_t(\theta)z_t + f_t^{\theta_t}(\theta_t) \quad (5)$$

$$h(z_t; \theta_t) = C_t(\theta)z_t + h_t^{\theta_t}(\theta_t). \quad (6)$$

2. THE PARTICLE FILTER

We here briefly present the theory and main algorithm. For a more intuitive presentation, see the accompanying paper (Gustafsson and Hriljac, 2003).

2.1 Recursive Bayesian estimation

Consider systems that are described by the generic state space model (2). The optimal Bayesian filter in this case is given below. For further details, consult (Doucet *et al.*, 2001b; Bergman, 1999).

Denote the observations at time t by $Y_t = \{y_0, \dots, y_t\}$. The Bayesian solution to compute the posterior distribution, $p(x_t|Y_t)$, of the state vector, given past observations, is given by (Bergman, 1999)

$$p(x_{t+1}|Y_t) = \int p(x_{t+1}|x_t)p(x_t|Y_t) dx_t, \quad (7a)$$

$$p(x_t|Y_t) = \frac{p(y_t|x_t)p(x_t|Y_{t-1})}{p(y_t|Y_{t-1})}. \quad (7b)$$

For expressions on $p(x_{t+1}|x_t)$ and $p(y_t|x_t)$ in (7) we use the known probability densities $p_{e_t}(x)$ and $p_{v_t+w_t}(x) = p_{v_t} * p_{w_t}(x)$, with all noises assumed independent,

$$p(x_{t+1}|x_t) = p_{v_t+w_t}(x_{t+1} - f(x_t)), \quad (8a)$$

$$p(y_t|x_t) = p_{e_t}(y_t - h(x_t)). \quad (8b)$$

2.2 Implementation

A numerical approximation to (7) is given by

$$p(x_t|Y_t) \approx \sum_{i=1}^N q_t^{(i)} \delta(x_t - x_t^{(i)}), \quad (9)$$

where $\delta(\cdot)$ is the Dirac delta function. The particles $x_t^{(i)}$ and the corresponding weights $q_t^{(i)}$ represent a sampled version of the posterior distribution $p(x_t|Y_t)$ (Doucet *et al.*, 2001b), and intuitively, the more samples the better approximation.

2.3 The algorithm

The discussion in the previous section is summarized in the algorithm below. This is the algorithm presented in (Gordon *et al.*, 1993) under the name, *Bayesian bootstrap filter*.

Algorithm 1. The Particle Filter

- (1) Generate N samples $\{x_{0|-1}^{(i)}\}_{i=1}^N$ from $p(x_0)$.
- (2) Calculate the weights $q_t^i = p(y_t|x_t^{(i)})$ and normalize, *i.e.*, $\bar{q}_t^{(i)} = \frac{q_t^i}{\sum_{j=1}^N q_t^{(j)}}$.
- (3) Resample with replacement,

$$P(x_{t|t}^{(i)} = x_{t|t-1}^{(i)}) = \bar{q}_t^{(i)}.$$

- (4) Predict (*i.e.*, simulate) new particles by

$$x_{t+1|t}^{(i)} = f(x_{t|t}^{(i)}) + v_t^{(i)} + w_t^{(i)}.$$

- (5) Iterate from step (2).

The particle filter can be interpreted as a simulation-based method, *i.e.*, N possible state trajectories, $x_t^{(i)}$, $i = 1, \dots, N$, are simulated. Based on the measurements each trajectory is assigned a weight, $\bar{q}_t^{(i)}$, representing the probability of that trajectory being the correct one.

3. MARGINALIZATION FOR VARIANCE REDUCTION

Consider the case where the model is linear in some of the states. Then the Kalman filter can be used to estimate the linear states, denoted x_t^l , and the particle filter can be used to estimate the nonlinear states, denoted x_t^n . To separate the problem of estimating $p(x_t^l, x_t^n | Y_t)$ into one linear and one nonlinear problem, Bayes' rule is used²

$$p(x_t^l, X_t^n | Y_t) = p(x_t^l | X_t^n, Y_t) p(X_t^n | Y_t). \quad (10)$$

Here the density $p(x_t^l | X_t^n, Y_t)$ is given by the Kalman filter and the particle filter is used to estimate $p(X_t^n | Y_t)$. This means that the particles live in a lower-dimensional space, and it can indeed be proven (Doucet *et al.*, 2001c; Nordlund, 2002) that the variance of any function of the state and parameter is decreased or remains constant when using marginalization for a given number of particles. This technique of marginalizing out the linear state is also referred to as Rao-Blackwellization (Doucet *et al.*, 2001c).

Before we state the theorem we have to introduce some notation. Let the estimate of any inference function of the state vector be given by

$$I(g(x_t)) = E_{p(x_t|Y_t)}[g(x_t)] = \int g(x_t) p(x_t | Y_t) dx_t, \quad (11)$$

and its estimate using N particles and the standard particle filter be denoted by $\hat{I}_N^s(g(x_t))$. When the marginalized particle filter is used the same estimate is denoted by $\hat{I}_N^m(g(x_t))$.

Theorem 1. Assume i.i.d. samples $\{x_t^{(i)}\}_{i=1}^N$ computed by Algorithm 1 and that the expected value and variance of the inference function $g(x_t)$ and the likelihood q_t exist and are finite. Then there is a central limit theorem stating that for large N ,

$$\begin{aligned} \hat{I}_N^s(g(x_t)) &\approx \mathcal{N}(I(g(x_t)), R_s(N)), \\ \hat{I}_N^m(g(x_t)) &\approx \mathcal{N}(I(g(x_t)), R_m(N)), \end{aligned}$$

where $R_s(N) \geq R_m(N)$.

See *e.g.*, (Doucet *et al.*, 2001c) for a proof.

Asymptotically as the number of particles tend to infinity there is nothing to gain in using marginalization, since then the particle filter will provide a perfect description of $p(x_t^l, x_t^n | Y_t)$. However, since we only can use a finite number of particles it is certainly useful to marginalize and use the optimal filter, *i.e.*, the Kalman filter, for the linear states. For details concerning the marginalized particle filter, the reader

² We have to use all the old nonlinear states, X_t^n , in order to make the Kalman filter work. The density $p(x_t^l, x_t^n)$ is then obtained by integrating $p(x_t^l, X_t^n)$ over the old nonlinear states, X_{t-1}

is referred to *e.g.*, (Chen and Liu, 2000), (Doucet *et al.*, 2001c), (Nordlund, 2002).

4. MODELS

In this section it will be shown how the particle filter can be used to estimate the nonlinear states and the Kalman filter to estimate the linear states, using the marginalization technique discussed above. All noise terms associated with the linear states are here assumed to be Gaussian, which means that the optimal estimator for the linear states/parameters is given by the Kalman filter. For the details concerning the Kalman filter equations, the state transition densities, and the likelihood functions in Algorithms 2 and 3 the reader is referred to (Nordlund, 2002). First there will be a discussion on models that are linear in the states and nonlinear in the parameters. This is followed by the reversed case, *i.e.*, linear in the parameters and nonlinear in the states.

4.1 State-space models linear in the states

A state-space model linear in the states and possibly nonlinear in the parameters is written as

$$z_{t+1} = A_t(\theta_t)z_t + f^\theta(\theta_t) + v_t^z \quad (12a)$$

$$\theta_{t+1} = \theta_t + v_t^\theta + w_t^\theta \quad (12b)$$

$$y_t = C_t(\theta_t)z_t + h^\theta(\theta_t) + e_t, \quad (12c)$$

where $v_t^z \in \mathcal{N}(0, Q_t^{v,z})$, $v_{\theta,t} \in \mathcal{N}(0, Q_t^{v,\theta})$ and $w_{\theta,t} \in \mathcal{N}(0, Q_t^{w,\theta})$ ³. Note that we can let the roughening noise w_t^θ be zero when using marginalization. The posterior density will here be separated using Bayes' rule according to

$$p(z_t, \Theta_t | Y_t) = p(z_t | \Theta_t, Y_t) p(\Theta_t | Y_t). \quad (13)$$

Note that we here consider the posterior of the complete parameter trajectory Θ_t , but only the last state vector z_t . The first density on the right hand side in (13) is given by the Kalman filter, while the second one is approximated by the particle filter. That is, we randomize particles in the parameter space according to our prior, and then each particle trajectory will be associated with one Kalman filter. The exact algorithm is given below.

Algorithm 2. The particle filter for linear states

Below, we let $C_t^{(i)} = C_t(\theta_{t|t-1}^{(i)})$ and $A_t^{(i)} = A_t(\theta_{t|t}^{(i)})$ for ease of notation.

³ The noise on the non-linear part, here w_t^θ and v_t^θ , can in fact have an arbitrary distribution. Similarly, The PDF $p(\theta_0)$ does not have any restrictions, since it is only used in the particle filter, the same goes for $p(e_t)$ if $C = 0$ in (12c). However, we leave these generalizations as a remark and assume Gaussian distributions.

- (1) Sample $\theta_{0|-1}^{(i)} \sim p(\theta_0)$ and set, for $i = 1, \dots, N$,
 $\{z_{0|-1}^{(i)}, P_{0|-1}^{(i)}\}_{i=1}^N = \{z_0, \Pi_0\}$.
- (2) Calculate the weights

$$\begin{aligned} q_t^{(i)} &= p(y_t | \Theta_t^{(i)}, Y_{t-1}) \\ &= \mathcal{N}(h^\theta(\theta_{t|t-1}^{(i)}) + C_t^{(i)} \hat{z}_{t|t-1}^{(i)}, \\ &\quad R_t + C_t^{(i)} P_{t|t-1} (C_t^{(i)})^T) \end{aligned}$$

and normalize, *i.e.*, $\bar{q}_t^{(i)} = \frac{q_t^{(i)}}{\sum_{j=1}^N q_t^{(j)}}$.

- (3) Resample with replacement,

$$P(\theta_{t|t}^{(i)} = \theta_{t|t-1}^{(j)}) = \bar{q}_t^{(j)}$$

- (4) Kalman filter measurement update:

$$\begin{aligned} S_t^{(i)} &= R_t + C_t^{(i)} P_{t|t-1}^{(i)} (C_t^{(i)})^T \\ L_t^{(i)} &= P_{t|t-1}^{(i)} (C_t^{(i)})^T (S_t^{(i)})^{-1} \\ \hat{z}_{t|t}^{(i)} &= \hat{z}_{t|t-1}^{(i)} + L_t^{(i)} (y_t - h^\theta(\theta_{t|t}^{(i)}) - C_t^{(i)} \hat{z}_{t|t-1}^{(i)}) \\ P_{t|t}^{(i)} &= P_{t|t-1}^{(i)} - L_t^{(i)} S_t^{(i)} (L_t^{(i)})^T \end{aligned}$$

- (5) Predict (*i.e.*, simulate) new particles by $p(\theta_{t+1|t} | \Theta_t, Y_t)$, where

$$\theta_{t+1|t}^{(i)} \sim p(\theta_{t+1|t} | \Theta_t^{(i)}, Y_t) = \mathcal{N}(\theta_{t|t}^{(i)}, Q_t^{v,\theta} + Q_t^{w,\theta}).$$

- (6) Kalman filter time update:

$$\begin{aligned} \hat{z}_{t+1|t}^{(i)} &= A_t^{(i)} \hat{z}_{t|t}^{(i)} + f^\theta(\theta_{t|t}^{(i)}) \\ P_{t+1|t}^{(i)} &= A_t^{(i)} P_{t|t}^{(i)} A_t^{(i)T} + Q_t^{v,z} + Q_t^{w,z} \end{aligned}$$

- (7) Compute relevant estimates from $p(z_t, \theta_t | Y_t)$.

$$\hat{\theta}_t = \sum_{i=1}^N \bar{q}_t^{(i)} \hat{\theta}_{t|t}^{k,(i)} \quad (15a)$$

$$\hat{z}_t = \frac{1}{N} \sum_{i=1}^N z_{t|t}^{(i)} \quad (15b)$$

- (8) Iterate from step (2).

Comparing the algorithms 1 and 2 we see that the differences are in the prediction step, which now consists of a Kalman filter update stage (split into step 4 and 6) besides the prediction of the nonlinear states.

In some cases the same Riccati recursion can be used for all the particles, and hence a lot of computations can be saved. This occurs when the matrices A_t and C_t in (12) are independent of θ_t . In this case $P_{t|t}^{(i)} = P_{t|t}$ for all $i = 1, \dots, N$ and hence the covariance only has to be updated once for each t . More on this can be found in (Gustafsson *et al.*, 2002).

4.2 State-space models linear in the parameters

A state-space model that is linear in the parameters can be written as

$$z_{t+1} = A_t(z_t) \theta_t + f^z(z_t) + v_t^z + w_t^z \quad (16a)$$

$$\theta_{t+1} = \theta_t + v_t^\theta \quad (16b)$$

$$y_t = C_t(z_t) \theta_t + h^z(z_t) + e_t. \quad (16c)$$

In this case the posterior will be split the other way around, compared to the previous section, *i.e.*,

$$p(Z_t, \theta_t | Y_t) = p(\theta_t | Z_t, Y_t) p(Z_t | Y_t). \quad (17)$$

The last density is approximated by the particle filter, while the first one can be solved by a Kalman filter for a parameter estimation problem in a linear regression framework. The corresponding algorithm will thus be

Algorithm 3. The particle filter for linear parameters
Below, we let $C_t^{(i)} = C_t(z_{t|t-1}^{(i)})$ and $A_t^{(i)} = A_t(z_{t|t}^{(i)})$ for ease of notation.

- (1) Sample $z_{0|-1}^{(i)} \sim p(z_0)$ and set for $i = 1, \dots, N$

$$\{\theta_{0|-1}^{(i)}, P_{0|-1}^{(i)}\}_{i=1}^N = \{\theta_0, \Pi_0\}.$$

- (2) Calculate the weights

$$\begin{aligned} q_t^{(i)} &= p(y_t | Z_t^{(i)}, Y_{t-1}) \\ &= \mathcal{N}(h(z_{t|t-1}^{(i)}) + C_t^{(i)} \hat{\theta}_{t|t-1}^{(i)}, \\ &\quad R_t + C_t^{(i)} P_{t|t-1}^{(i)} (C_t^{(i)})^T) \end{aligned}$$

and normalize, *i.e.*, $\bar{q}_t^{(i)} = \frac{q_t^{(i)}}{\sum_{j=1}^N q_t^{(j)}}$.

- (3) Resample with replacement,

$$P(z_{t|t}^{(i)} = z_{t|t-1}^{(j)}) = \bar{q}_t^{(j)} \quad (18)$$

- (4) Kalman filter measurement update:

$$\begin{aligned} S_t^{(i)} &= (R_t + C_t^{(i)} P_{t|t-1}^{(i)} (C_t^{(i)})^T) \\ L_t^{(i)} &= P_{t|t-1}^{(i)} C_t^{(i)T} (S_t^{(i)})^{-1} \\ \hat{\theta}_{t|t}^{(i)} &= \hat{\theta}_{t|t-1}^{(i)} + L_t^{(i)} (y_t - h(z_{t|t}^{(i)}) - C_t^{(i)} \hat{\theta}_{t|t-1}^{(i)}) \\ P_{t|t}^{(i)} &= P_{t|t-1}^{(i)} - L_t^{(i)} S_t^{(i)} (L_t^{(i)})^T \end{aligned} \quad (19)$$

- (5) Predict (*i.e.*, simulate) new particles by $p(z_{t+1|t} | Z_t, Y_t)$, where

$$\begin{aligned} z_{t+1|t}^{(i)} &\sim p(z_{t+1|t} | Z_t^{(i)}, Y_t) \\ &= \mathcal{N}(A_t^{(i)} \hat{\theta}_{t|t}^{(i)} + f^z(z_{t|t}^{(i)}) + v_t^{z,v} + Q_t^{\theta,w} + \\ &\quad A_t^{(i)} P_{t|t}^{(i)} (A_t^{(i)})^T). \end{aligned} \quad (20)$$

- (6) Kalman filter time and state update:

$$\begin{aligned} \hat{\theta}_{t+1|t}^{(i)} &= \hat{\theta}_{t|t}^{(i)} + K_t^{(i)} (z_{t+1|t}^{(i)} - A_t^{(i)} \hat{\theta}_{t|t}^{(i)}) \\ P_{t+1|t}^{(i)} &= P_{t|t}^{(i)} + Q_t^{v,z} + Q_t^{w,z} - \\ &\quad K_t^{(i)} (Q_t^n + A_t^{(i)}) P_{t|t}^{(i)} (A_t^{(i)})^T (K_t^{(i)})^T \\ K_t^{(i)} &= P_{t|t}^{(i)} (A_t^{(i)})^T (Q_t^n + A_t^{(i)} P_{t|t}^{(i)} (A_t^{(i)})^T)^{-1} \end{aligned}$$

- (7) Compute relevant estimates from $p(z_t, \theta_t | Y_t)$.

$$\hat{z}_t = \frac{1}{N} \sum_{i=1}^N z_{t|t}^{(i)} \quad (22a)$$

$$\hat{\theta}_t = \sum_{i=1}^N \bar{q}_t^{(i)} \hat{\theta}_{t|t}^{k,(i)} \quad (22b)$$

(8) Iterate from step (2).

The measurements used in the Kalman filter are thus the “normal” measurements y_t and the predicted state trajectory $z_{t+1|t}$, *i.e.*, the samples from the particle filter. Step 6 in the current algorithm contains a measurement update, using the prediction (since this contains information about θ_t) from the particle filter, and a time update.

5. PARAMETRIC INNOVATION MODELS

An interesting special case of the two different model types discussed above is when we consider “the intersection” of the two types, *i.e.*, a model that is bilinear in the states, z_t , and in the parameters, θ_t .

A particular case of interest is a general state-space model in innovation form

$$z_{t+1} = A(\theta_t)z_t + K(\theta_t)e_t \quad (23a)$$

$$y_t = C(\theta_t)z_t + e_t, \quad (23b)$$

where the parameters enter linearly in A , K , and C . The posterior will here be according to (17). One popular approach here is so called subspace identification, see (?). This class of algorithms usually perform very well and gives consistent estimates. One limitation is that it is hard to give the *a posteriori* distribution of the parameters, even in the Gaussian case, and this is perhaps where the particle filter can help. However, the particle filter has perhaps not so much to offer to this bi-linear model, and this case is mentioned more to show the relation to classical system identification problems.

Assume, to avoid ambiguities in the state coordinates, an observer canonical form and scalar output, where $C = (1, 0, \dots, 0)$ and that all parameters in A and K are unknown. Then, given the state trajectory and measurement, we have from (16) the linear regression $y_t = Az_t + K(y_t - z_t^{(1)})$. This regression problem has to be solved for each particle $z_t^{(i)}$.

In the case where there are more states to be estimated than parameters, *i.e.*, $\dim z > \dim \theta$ it is better to split the density $p(Z_t, \theta_t | Y_t)$ in (17) the other way around, *i.e.*, as in (13). This time, a Kalman filter estimating the states z_t for each particle $\theta_t^{(i)}$ is needed. In this way the dimension of the state estimated by the particle filter is kept as low as possible. An example where this situation typically occurs is in gray-box identification (Ljung, 1999).

6. CHAOS EXAMPLE

The ideas presented in this article will be illustrated using the following chaotic model

$$z_{t+1} = (1 - z_t)z_t\theta + v_t, \quad (24a)$$

$$y_t = z_t + e_t, \quad (24b)$$

where, z_t , is the state variable, y_t , is the measurement, θ is the unknown parameter, v_t is the process noise, and e_t is the measurement noise. Both these noise densities are Gaussian distributed. The aim is to recursively estimate both the state, z_t , and the parameter, θ . This model is linear in the time-invariant parameter θ and nonlinear in the state z_t . This fits our framework, according to Section 4.2 and hence Algorithm 3 can be applied. This problem has also been studied in (Gustafsson and Hriljac, 2003), where the particle filter was directly applied to the augmented state $x_t = (z_t, \theta_t)$. The model (24) can be written on the form (16), *i.e.*,

$$z_{t+1} = A_t(z_t)\theta_t + v_t + w_t^z, \quad (25a)$$

$$\theta_{t+1} = \theta_t + w_t^\theta, \quad (25b)$$

$$y_t = h_t(z_t) + e_t, \quad (25c)$$

where $A_t(z_t) = z_t(1 - z_t)$ and $h_t(z_t) = z_t$. The two noises $w_t^z \sim \mathcal{N}(0, Q_t^{w,z})$ and $w_t^\theta \sim \mathcal{N}(0, Q_t^{w,\theta})$ are roughening noises. Furthermore $e_t \sim \mathcal{N}(0, R_t)$.

In the simulations, two different particle filters were used, the standard particle filter, Algorithm 1, applied to the augmented state vector, x_t , and the marginalized particle filter according to Algorithm 3. The true value of θ is 3.92, and the initial guess is $\theta_{0|-1} \sim \mathcal{N}(3.83, 0.04)$. The initial state is $z_0 \sim \mathcal{N}(0, 1)$. We do not use any process noise, however we have roughening noises $Q_0^{w,z} = Q_0^{w,\theta} = 10^{-2}$, which is decreased at each time step, according to (Gustafsson and Hriljac, 2003). The measurement noise has variance $R_t = 10^{-5}$, and we have used 200 Monte Carlo simulations. In Fig. 1 the filtered estimates of θ are shown using these two algorithms for 150, 1000, and 10000 particles respectively. In order to make the dif-

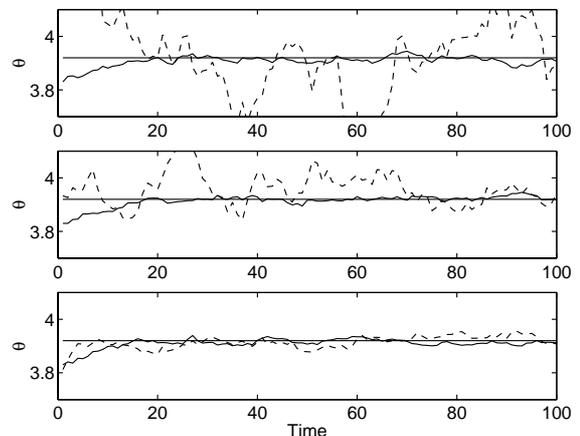


Fig. 1. Estimates of θ using the standard (dotted) and the marginalized (solid) particle filters. The true θ is shown using a dashed line. Top plot - 150 particles, middle - 1000 particles, bottom - 10000 particles.

ference more apparent the Root Mean Square Error (RMSE) is plotted in Fig. 2 as a function of the number of particles used in the simulations. Note that the RMSE values are calculated from time 50. In that way the transient effects are not included in the RMSE values. According to Theorem 1 the estimates should

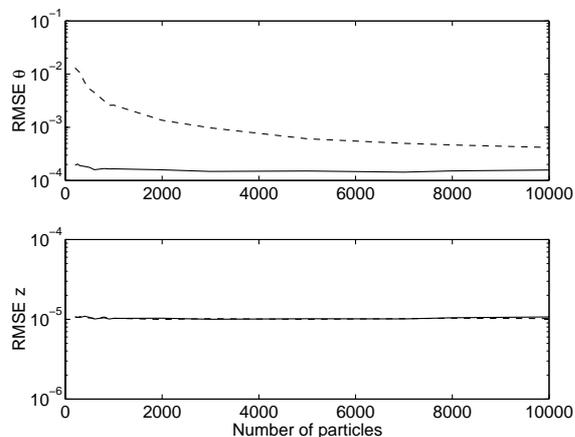


Fig. 2. RMSE values for $\hat{\theta}$ (top) and \hat{z} (bottom) as a function of the number of particles used. Notice that a log-scale has been used in the plots, and that a dashed line has been used for the standard particle filter and a solid line for the marginalized particle filter.

be better or the same when we use the marginalized particle filter. From Fig. 2 we can see that this is indeed the case. It is only the estimate of the linear part, θ , that is improved, this is also consistent with the theory, see e.g., (Nordlund, 2002) for the theoretical details. That this is true in the simulations is apparent by Fig. 2, from which it is clear that the estimate of the linear part (top) clearly is better using the marginalized particle filter. The estimate of the nonlinear part, z_t , has the same quality. Of course if we could use an infinite number of particles the results using the standard and the marginalized particle filter would have been the same, since then the particle filter would be able to provide an arbitrarily good estimate of $p(x_t|Y_t)$. We can see indications of this fact in the top plot in Fig. 2, since the more particles that are used the closer the estimates get.

7. CONCLUSIONS

The potential use of particle filtering for system identification of unknown parameters in non-linear systems was explained in the accompanying paper (Gustafsson and Hriljac, 2003). Here, we have proposed the use of marginalized particle filters. More specifically, we studied the cases where the model is either linear in the states and nonlinear in the parameters, or nonlinear in the states and linear in the parameters. The algorithms were given for these two cases. It is straightforward

to give the algorithm for an arbitrary mix of linear and nonlinear states and parameters. The advantage of marginalization is that one can apply the filter to larger problems with more states and parameters, or that fewer particles and thus less filter complexity is needed for a given performance. Finally an example was given, which illustrates the improvement in estimation performance compared to using a standard particle filter.

REFERENCES

- Bergman, N. (1999). Recursive Bayesian Estimation: Navigation and Tracking Applications. Dissertation nr. 579. Linköping University, Sweden.
- Chen, R. and J.S. Liu (2000). Mixture Kalman filters. *Journal of the Royal Statistical Society* **62**(3), 493–508.
- Doucet, A., de Freitas, N. and Gordon, N., Eds.) (2001a). *Sequential Monte Carlo Methods in Practice*. Springer Verlag.
- Doucet, A., de Freitas, N. and Gordon, N., Eds.) (2001b). *Sequential Monte Carlo Methods in Practice*. Springer Verlag.
- Doucet, A., N. Gordon and V. Krishnamurthy (2001c). Particle Filters for State Estimation of Jump Markov Linear Systems. *IEEE Transactions on Signal Processing* **49**(3), 613–624.
- Gordon, N.J., D.J. Salmond and A.F.M. Smith (1993). A novel approach to nonlinear/non-Gaussian Bayesian state estimation. In: *IEE Proceedings on Radar and Signal Processing*. Vol. 140. pp. 107–113.
- Gustafsson, F. and P. Hriljac (2003). Particle filters for prediction of chaos. Accepted for publication at the 13th IFAC Symposium on System Identification.
- Gustafsson, F., F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson and Nordlund P-J (2002). Particle filters for positioning, navigation and tracking. *IEEE Transactions on Signal Processing*.
- Ljung, L. (1999). *System identification, Theory for the user*. System sciences series. 2 ed.. Prentice Hall. Upper Saddle River, NJ.
- Nordlund, P.J. (2002). Sequential Monte Carlo Filters and Integrated Navigation. Licentiate thesis. Thesis No. 945.