



LUND UNIVERSITY

Rao-Blackwellized Out-of-Sequence Processing for Mixed Linear/Nonlinear State-Space Models

Berntorp, Karl; Robertsson, Anders; Årzén, Karl-Erik

Published in:

[Host publication title missing]

2013

[Link to publication](#)

Citation for published version (APA):

Berntorp, K., Robertsson, A., & Årzén, K.-E. (2013). Rao-Blackwellized Out-of-Sequence Processing for Mixed Linear/Nonlinear State-Space Models. In *[Host publication title missing]* (pp. 805-812). IEEE - Institute of Electrical and Electronics Engineers Inc..

Total number of authors:

3

General rights

Unless other specific re-use rights are stated the following general rights apply:

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

Read more about Creative commons licenses: <https://creativecommons.org/licenses/>

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

LUND UNIVERSITY

PO Box 117
221 00 Lund
+46 46-222 00 00

Rao-Blackwellized Out-of-Sequence Processing for Mixed Linear/Nonlinear State-Space Models

Karl Berntorp, Anders Robertsson, and Karl-Erik Årzén

Department of Automatic Control,
Lund University, SE-221 00 Lund, Sweden
firstname.lastname@control.lth.se

Abstract—We investigate the out-of-sequence measurements particle filtering problem for a set of conditionally linear Gaussian state-space models, known as mixed linear/nonlinear state-space models. Two different algorithms are proposed, which both exploit the conditionally linear substructure. The first approach is based on storing only a subset of the particles and their weights, which implies low memory and computational requirements. The second approach is based on a Rao-Blackwellized forward filter/backward simulator, adapted to the out-of-sequence filtering task with computational considerations for enabling online implementations. Simulation studies on two examples show that both approaches outperform recently reported particle filters, with the second approach being superior in terms of tracking performance.

I. INTRODUCTION

During the past decade there has been an increase in the number of sensors used in tracking systems, as well as an increase in the number of distributed, heterogeneous sensing systems. Hence, the ability to account for that some measurements arrive after that a more recent measurement from the same target has already been processed is becoming crucial in modern tracking systems. These delayed measurements are denoted out-of-sequence measurements (OOSMs) [1], and occur, for example, because of data preprocessing and communication delays. They arise in a variety of applications, one example being an automotive application where network links cause transmission delays of radar sensors [2]. Another example is tracking of autonomous vehicles, where cameras have become increasingly important for giving spatial information. Here, it is often the processing times of the vision algorithms that cause the OOSMs [3]. Discarding the OOSMs implies discarding information, and may thus lead to deficient tracking performance. However, incorporating the measurements into nonlinear tracking systems can be challenging.

In this paper we propose two novel algorithms for processing OOSMs in a Rao-Blackwellized particle filter setting considering mixed linear/nonlinear state-space models. This model class is common in, for example, tracking, positioning, and navigation [4]. Earlier work has provided the exact Bayesian inference solution and its particle filter implementation to the OOSM problem, see [5]. By exploring a conditionally linear Gaussian substructure in the model class considered, however, we are able to provide both improved performance and reduced

computational demands. The first algorithm is both storage efficient and computationally fast while improving performance compared to previous approaches. It can be seen as a generalization of the storage efficient particle filters reported in [6], with the derivations adapted to the mixed linear/nonlinear setting. It is developed under the assumption that only a subset of the particles and their associated weights, together with the measurements, are stored. Because of its simplicity it is well suited for real-time applications, even for systems of considerable size. The second algorithm is developed with focus on tracking performance. Naturally, this performance improvement comes with increased computational complexity. Using two simulation examples, we show that both approaches outperform recently reported particle filter algorithms, [5], [7], for processing OOSMs.

A. Related Work

There has been a large amount of research performed in trying to incorporate OOSMs for linear-Gaussian (LG) systems. Two suboptimal algorithms can be found in [8] and [9]. Examples of OOSM algorithms which are optimal in the mean-square sense for the LG case are found in [1], [10], [11].

For nonlinear systems, several algorithms using the particle filter framework have been derived. An example is found in [12], which, however, needs a linear state-transition model to form the proposal density of the state. Moreover, to be able to retrodict the state vector back to the OOSM time, [13] assumes an invertible state-transition matrix. To enable nonlinear state-space models, [6] presented the storage efficient particle filter (SEPF). SEPF is computationally fast and memory efficient, since it only stores and processes means and covariances through an extended Kalman smoother to update the estimates with the OOSMs. However, the performance of SEPF suffers when the OOSMs change the particle weights too much in the measurement update step. This problem was solved in [14], where the approach in [6] was extended with an algorithm for detecting the problematic OOSMs. Another approach for increased performance was given in [7], where the SEPF in [6] and the smoother in [11] were combined to enhance performance and decrease storage requirements. In [5], an exact Bayesian solution and its corresponding particle filter implementation for nonlinear models with Gaussian noise was derived, denoted A-PF. The drawback with this algorithm is that it is computationally expensive. For OOSMs that have larger delays than one sample, its complexity is roughly $\mathcal{O}((l-1)N^3 + N^2)$, where l is the delay in the number of time steps, and N is the number of particles.

The authors are members of the LCCC Linnaeus Center and the ELLIIT Excellence Center at Lund University. This work was supported by the Swedish Foundation for Strategic Research through the project ENGROSS.

B. Outline

The outline of the rest of the paper is as follows: In Sec. II we give the problem statement and the notation used. Section III reviews particle filters and smoothers, both for the non-Rao-Blackwellized and the Rao-Blackwellized approaches. The two proposed algorithms are presented in Sec. IV. In Sec. V, the performance of the proposed methods is assessed through comparison with the algorithms in [5] and [7]. Finally, we conclude the work in Sec. VI.

II. PROBLEM FORMULATION

The conditional distribution density of the variable x at an arbitrary time $t_k \in \mathbb{R}$ conditioned on the variable y from time t_m to time t_k is denoted $p(x_k|y_{m:k})$. For each time t_k , we assume that the state vector $x_k \in \mathbb{R}^{n_x}$ can be partitioned into a linear part, z_k , and a nonlinear part, η_k , as $x_k = (z_k \ \eta_k)^T$. We consider discrete-time mixed linear/nonlinear state-space models of the form

$$z_{k+1} = f(\eta_k) + A(\eta_k)z_k + F(\eta_k)v_k^z, \quad (1)$$

$$\eta_{k+1} = g(\eta_k) + B(\eta_k)z_k + G(\eta_k)v_k^\eta, \quad (2)$$

$$y_k = h(\eta_k) + C(\eta_k)z_k + e_k, \quad (3)$$

where $f(\cdot)$, $g(\cdot)$, and the measurement function $h(\cdot)$ are vector-valued, possibly nonlinear functions. Further, $A(\cdot)$, $F(\cdot)$, $B(\cdot)$, $G(\cdot)$, and $C(\cdot)$ are matrices of suitable dimensions, with a possibly nonlinear dependence on η . For brevity, we write f_k for $f(\eta_k)$, A_k for $A(\eta_k)$, et cetera. Note that given η_k , (1) is linear with the measurement relations (2) and (3). The process noise v_k^z is white Gaussian with zero mean and unit covariance matrix according to $v_k^z \sim \mathcal{N}(0, I)$. Similarly, the process noise v_k^η is distributed as $v_k^\eta \sim \mathcal{N}(0, I)$, and the measurement noise e_k is distributed according to $e_k \sim \mathcal{N}(0, R(\eta_k))$. The timestamp is referred to as t_k . For the OOSM filtering task, we denote the set of in-sequence measurements generated in the interval $[0, k]$ as \mathcal{Y}_k . Moreover, we refer to the set of OOSMs generated in the interval $[0, k]$ available at time index k as \mathcal{Z}_k . For simplicity, we express the set $\mathcal{Y}_k \cup \mathcal{Z}_{k-1}$ with $y_{0:k}$.

Suppose that we at time t_k in the estimation process have an estimate of the filtering posterior $p(z_k, \eta_k|y_{0:k})$, where z_k is conditioned on $y_{0:k}$. Assume that an OOSM $y_\tau \in \mathcal{Z}_k$ with timestamp $t_\tau \in [t_{k-l}, t_{k-l+1})$ arrives, see Fig. 1 for an illustration. The Rao-Blackwellized OOSM filtering task is to update the weights and linear estimates at time t_k with y_τ , that is, to obtain $p(x_k|y_{0:k}, y_\tau) = p(z_k, \eta_k|y_{0:k}, y_\tau)$.

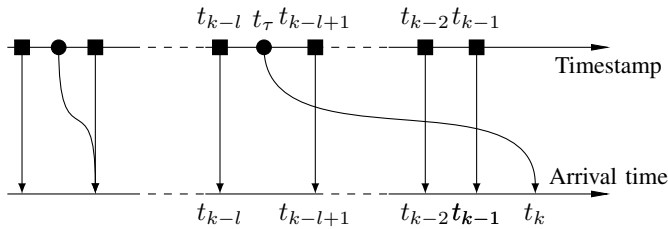


Fig. 1. An illustration of the l -step lag OOSM problem, where the circles denote OOSMs and squares refer to the in-sequence measurements.

III. BACKGROUND

Here, we briefly review particle filtering and smoothing, both for pure nonlinear and mixed linear/nonlinear state-space models.

A. Particle Filtering and Smoothing

Consider a standard Markov process with the dynamics and measurement equation as $x_{k+1} = f(x_k) + v_k$, $y_k = h(x_k) + e_k$, where $f(\cdot)$ and $h(\cdot)$ are nonlinear functions, and v_k and e_k have known densities. In general a closed form representation of the posterior density $p(x_{0:k}|y_{0:k})$ is impossible to obtain. Particle filters (PFs) are sequential Monte Carlo methods trying to represent the posterior density with a set of weighted samples, or particles [15]. Each particle represents a state trajectory $x_{0:k}$, which results in the approximation

$$p(x_{0:k}|y_{0:k}) \approx \sum_{i=1}^N w_k^i \delta_{x_{0:k}^i} (x_{0:k}). \quad (4)$$

In (4), $\delta(\cdot)$ is the Dirac delta function and w_k^i is the associated weight for the i th particle given the measurements $y_{0:k}$. The particle weights are typically updated as $w_k^i \propto p(y_k|x_k^i)w_{k-1}^i$. To propagate forward in time, new samples are drawn from an importance density, often, but not always, the dynamics itself. An approximation to the marginal (filtering) density is given by marginalization of (4), that is, discarding $x_{0:k-1}$, yielding $p(x_k|y_{0:k})$. To account for sample diversity, a resampling step is typically performed if $1/\sum(w_k^i)^2 < N_{\text{eff}}$, where $0 < N_{\text{eff}} \leq N$. In this step, particles that have negligible weights are replaced with particles with larger weights, meaning that probable particles are cloned and improbable particles are discarded.

In particle smoothing, $p(x_{0:T}|y_{0:T})$ is the density of interest. Note that the marginal and fixed-interval smoothing densities can be found by marginalization, similarly to the particle filter case described earlier. If all particle histories are stored in the particle filter, and if the resampling step resamples whole trajectories instead of only the latest time step, the particle filter approximation (4) is also an approximation to the smoothing density [16]. However, this estimate is often degenerate, which means that all trajectories collapse to the same for $k \ll T$. For this reason the smoothing problem is often approached with other algorithms. One example is the forward filter/backward simulator (FFBS) smoother [17]. The FFBS utilizes the sequential factorization

$$p(x_{0:T}|y_{0:T}) = p(x_T|y_{0:T}) \prod_{k=0}^{T-1} p(x_k|x_{k+1:T}, y_{0:T}), \quad (5)$$

where $p(x_k|x_{k+1:T}, y_{0:T}) = p(x_k|x_{k+1}, y_{0:k})$. Starting by sampling a state x'_T from the filtering approximation $p(x_T|y_{0:T})$ at time index T , the Markov property

$$p(x_k|x_{k+1}, y_{0:k}) \propto p(x_{k+1}|x_k)p(x_k|y_{0:k}) \quad (6)$$

is utilized to form the approximation

$$p(x_{T-1}|x'_T, y_{0:T}) \approx \sum_{i=1}^N w_{T-1|T}^i \delta_{x_{0:T-1}^i} (x_{0:T-1}),$$

where $w_{T-1|T}^i \propto w_{T-1}^i p(x_T^i | x_{T-1}^i)$. Iterating backward in time, x_{T-1} is formed by sampling from the filtering distribution at time index $T-1$ with probability $w_{T-1|T}^i$. This procedure is repeated until time index $k=0$ is reached, whereby an approximation can be formed as $p(x_{0:T} | y_{0:T}) \approx \delta_{x_{0:T}}(x_{0:T})$. For a better approximation of the smoothing density, the algorithm is repeated M times to yield

$$p(x_{0:T} | y_{0:T}) \approx \frac{1}{M} \sum_{j=1}^M \delta_{x_{0:T}^j}(x_{0:T}). \quad (7)$$

B. Rao-Blackwellized Particle Filtering and Smoothing

Particle filters and smoothers can be used for many different types of dynamic models. However, to decrease the number of samples needed in the filter it is often advantageous to exploit model structure. This is the idea behind Rao-Blackwellization, where the subset of the state space which allows for analytic expressions is marginalized out. Thus, the state space to be covered by samples is smaller, and it should therefore be possible to use significantly fewer samples [18].

The structure of the mixed linear/nonlinear model consisting of (1)–(3) can be exploited to give a Rao-Blackwellized particle filter (RBPF), see [4] for details. The idea is to use the factorization

$$p(z_k, \eta_{0:k} | y_{0:k}) = p(z_k | \eta_{0:k}, y_{0:k}) p(\eta_{0:k} | y_{0:k}), \quad (8)$$

where $p(\eta_{0:k} | y_{0:k})$ in (8) is approximated by the particle filter (4). Given the nonlinear state trajectory the first part in (8) is linear Gaussian, and can hence be estimated with constrained Kalman filters, one for each particle. The main difference compared to a standard Kalman filter consists of performing an extra measurement update using η_k . The first part in (8) equals

$$p(z_k | \eta_{0:k}, y_{0:k}) = \mathcal{N}(z_k; \hat{z}_{k|k}(\eta_{0:k}), P_{k|k}(\eta_{0:k})). \quad (9)$$

In (9), $\mathcal{N}(z_k; \mu, \Upsilon)$ is the Gaussian probability density function given mean μ and covariance matrix Υ , $\hat{z}_{k|k}(\eta_{0:k})$ is the linear state estimate given the trajectory $\eta_{0:k}$ and measurements $y_{0:k}$, and $P_{k|k}(\eta_{0:k})$ is its associated covariance. For brevity, we make the dependence on the trajectory implicit. By combining (4) and (9), an approximation to (8) is given by

$$p(z_k, \eta_{0:k} | y_{0:k}) \approx \sum_{i=1}^N w_k^i \mathcal{N}(z_k; \hat{z}_{k|k}^i, P_{k|k}^i) \delta_{\eta_{0:k}^i}(\eta_{0:k}). \quad (10)$$

The weights w_k^i are updated similarly to the standard particle filter, and the density $p(z_k, \eta_k | y_{0:k})$ is found by a marginalization of (10).

In Rao-Blackwellized particle smoothing (RBPS), an approximation to the density $p(z_k, \eta_k | y_{0:T})$ for $k < T$ is sought, in accordance with (7), as

$$p(z_k, \eta_k | y_{0:T}) \approx \frac{1}{M} \sum_{j=1}^M \mathcal{N}(z_k; \hat{z}_{k|T}^j, P_{k|T}^j) \delta_{\eta_k^j}(\eta_k), \quad (11)$$

where $\hat{z}_{k|T}^j$ and $P_{k|T}^j$ are conditioned on $\eta_{k:T}^j$. Since conditionally linear state-space models such as (1)–(3) do not have the Markov property (6), the FFBS-based smoothing approach is not straightforward for RBPS. The non-Markovian property

implies that the measurement likelihood depends on the whole nonlinear trajectory $\eta_{0:T}$. Therefore, whole trajectories must be sampled from the RBPF in order to preserve Gaussianity.

Recently, an RBPS considering the mixed linear/nonlinear model class was derived in [19]. The novelty lies in that the smoother only samples the nonlinear part of the state vector. This RBPS starts with drawing a sample η_T' from the RBPF particles at time index T . Then the task is to extend the trajectory to time index $k \leq T-1$. To this end, one of the RBPF particles $\{\eta_{0:k}^i\}_{i=1}^N$ are drawn with probability $w_{k|T}^i$, and by discarding $\eta_{0:k-1}^i$ the trajectory is appended, yielding $\{\eta_k, \eta_{k+1:T}'\}$. This procedure is repeated for each time step $k = T-1, \dots, 0$, resulting in a backward trajectory that can be used to approximate (5). To compute $w_{k|T}^i$ (i.e., $p(x_k | x_{k+1:T}, y_{0:T})$) in (5), Bayes' rule is utilized, yielding

$$p(\eta_{0:k} | \eta_{k+1:T}, y_{0:T}) \propto p(y_{k+1:T}, \eta_{k+1:T} | \eta_{0:k}, y_{0:k}) \times p(\eta_{0:k} | y_{0:k}). \quad (12)$$

The second factor in (12) is approximated by the RBPF. This results in

$$p(\eta_{0:k} | \eta_{k+1:T}, y_{0:T}) \approx \sum_{i=1}^N w_{k|T}^i \delta_{\eta_{0:k}^i}(\eta_{0:k}),$$

where

$$w_{k|T}^i \propto w_k^i p(y_{k+1:T}, \eta_{k+1:T}' | \eta_{0:k}^i, y_{0:k}). \quad (13)$$

The density in (13) is calculated by noting that

$$p(y_{k+1:T}, \eta_{k+1:T}' | \eta_{0:k}^i, y_{0:k}) = \int p(y_{k+1:T}, \eta_{k+1:T}' | z_k^i, \eta_k^i) p(z_k^i | \eta_{0:k}^i, y_{0:k}) dz_k^i, \quad (14)$$

where the second factor is given by (9). The density $p(y_{k+1:T}, \eta_{k+1:T}' | z_k^i, \eta_k^i)$ is found by propagating zero, first, and second order moments, namely $\{Z_k, \lambda_k, \Omega_k\}$, dependent on η_k but independent of z_k , backward in time as the trajectory η_{k+1}' is drawn. Given the set of statistics $\{Z_k, \lambda_k, \Omega_k\}$, $p(y_{k+1:T}, \eta_{k+1:T}' | z_k^i, \eta_k^i)$ equals, up to proportionality,

$$p(y_{k+1:T}, \eta_{k+1:T}' | z_k, \eta_k) \propto Z_k e^{-\frac{1}{2}(z_k^T \Omega_k z_k - 2\lambda_k^T z_k)}.$$

Marginalizing out z_k^i gives the sought density (14) as

$$p(y_{k+1:T}, \eta_{k+1:T}' | \eta_{0:k}^i, y_{0:k}) \propto Z_k |\Lambda_k|^{-1/2} e^{-\frac{1}{2}\zeta_k}, \quad (15)$$

where

$$\zeta_k = \|\hat{z}_{k|k}\|_{\Omega_k} - 2\lambda_k^T \hat{z}_{k|k} - \|\Gamma_k^T(\lambda_k - \Omega_k \hat{z}_{k|k})\|_{\Lambda^{-1}}, \quad (16)$$

$$P_{k|k} = \Gamma_k \Gamma_k^T, \quad (17)$$

$$\Lambda_k = \Gamma_k^T \Omega_k \Gamma_k + I, \text{ where } \|\mu\|_{\Omega} = \mu^T \Omega \mu. \quad (18)$$

When the full backward trajectory $\eta_{0:T}'$ has been found, the algorithm is typically repeated M times to give a set of backward trajectories $\{\eta_{0:T}^j\}_{j=1}^M$ similarly to (7). Note that (14) is calculated for all N particles, thus giving the complexity $\mathcal{O}(TMN)$. To find smoothed estimates of the linear states, constrained Rauch-Tung-Striebel (RTS) smoothers are used [20], for each trajectory. This finally gives the approximated smoothing density as in (11). For further details, see [19].

IV. RAO-BLACKWELLIZED OUT-OF-SEQUENCE PROCESSING

As described in Sec. II, in the Rao-Blackwellized OOSM filtering task the aim is to estimate the density

$$p(z_k, \eta_k | y_{0:k}, y_\tau), \quad (19)$$

where an estimate of $p(z_k, \eta_k | y_{0:k})$ already exists. To utilize the linear structure in (1)–(3), we factorize (19) as

$$p(z_k, \eta_k | y_{0:k}, y_\tau) = p(z_k | \eta_k, y_{0:k}, y_\tau) p(\eta_k | y_{0:k}, y_\tau). \quad (20)$$

By using Bayes' rule on the second factor of (20) we obtain

$$p(\eta_k | y_{0:k}, y_\tau) \propto p(y_\tau | \eta_k, y_{0:k}) p(\eta_k | y_{0:k}). \quad (21)$$

The factor $p(\eta_k | y_{0:k})$ in (21) can be approximated with the RBPF, which results in the approximation

$$p(\eta_k | y_{0:k}, y_\tau) \approx \sum_{i=1}^N w_{k|k,\tau}^i \delta_{\eta_k^i}(\eta_k), \quad (22)$$

where

$$w_{k|k,\tau}^i \propto w_k^i p(y_\tau | \eta_k^i, y_{0:k}). \quad (23)$$

Further,

$$p(y_\tau | \eta_k^i, y_{0:k}) = \int p(y_\tau | z_k^i, \eta_k^i, y_{0:k}) p(z_k^i | \eta_k^i, y_{0:k}) dz_k^i. \quad (24)$$

Thus, in order to update the posterior $p(\eta_k | y_{0:k})$ with y_τ to form (22), we need to estimate the likelihoods $\{p(y_\tau | z_k^i, \eta_k^i, y_{0:k})\}_{i=1}^N$.

To incorporate the OOSM y_τ into the first factor in (20), we recast (20) as

$$p(z_k | \eta_k, y_{0:k}, y_\tau) = \frac{p(y_\tau | z_k, \eta_k, y_{0:k}) p(z_k | \eta_k, y_{0:k})}{\int p(y_\tau | z_k, \eta_k, y_{0:k}) p(z_k | \eta_k, y_{0:k}) dz_k}. \quad (25)$$

We use the RBPF (9) to approximate the second factor in the numerator of (25). The first term in the numerator equals the first term on the right-hand side of (24). What is remaining, then, is to evaluate the densities $\{p(y_\tau | z_k^i, \eta_k^i, y_{0:k})\}_{i=1}^N$. In the following we present two different approaches for performing the particle filter and the Kalman filter measurement update, that is, to compute $p(y_\tau | x_k^i, y_{0:k}) = p(y_\tau | z_k^i, \eta_k^i, y_{0:k})$.

A. OOSM Processing with Supporting RBPF

For the first algorithm we focus on finding a computationally efficient method that computes $p(y_\tau | x_k^i, y_{0:k})$, while still improving performance over existing OOSM algorithms.

Utilizing the Chapman-Kolmogorov equation, the density $p(y_\tau | x_k^i, y_{0:k})$ is rewritten as

$$p(y_\tau | x_k^i, y_{0:k}) = \int p(y_\tau | z_\tau, \eta_{0:\tau}) p(z_\tau, \eta_{0:\tau} | x_k^i, y_{0:k}) dz_\tau d\eta_{0:\tau}. \quad (26)$$

The density $p(z_\tau, \eta_{0:\tau} | x_k^i, y_{0:k})$ in (26) can be regarded as a fixed-point (marginal) Rao-Blackwellized smoothing density, which we approximate by rewriting it as

$$p(z_\tau, \eta_{0:\tau} | x_k^i, y_{0:k}) \propto p(x_k^i, y_{k-l+1:k-1} | z_\tau, \eta_{0:\tau}) \times p(z_\tau, \eta_{0:\tau} | y_{0:k-l}). \quad (27)$$

In (27), we have dropped y_k since x_k^i is given. We see that $p(z_\tau, \eta_{0:\tau} | y_{0:k-l})$ is given by the forward RBPF (8). The first density on the right-hand side in (27) is a measurement update using both $y_{k-l+1:k-1}$ and x_k^i as measurements. Thus we need to propagate the past, $\{z_\tau, \eta_{0:\tau}\}$, to update with $\{x_k^i, y_{k-l+1:k-1}\}$.

As discussed in Sec. III-A, the PF (or RBPF) is also an approximate solution to the smoothing problem. Thus, an efficient way to approximate the smoothing densities is to run a supporting (additional) RBPF to find (27), which can then be inserted into (26) to update the weights in (23) and linear states in (25): At time index $k-l$ we start an additional, supporting RBPF. Since (27) is a smoothing density it can be represented by fewer particles than for the original RBPF. Therefore we start with sampling $M \leq N$ estimates $\{\hat{z}_{k-l|k-l}^j, \eta_{k-l}^j\}_{j=1}^M$ from the filtering density at time index $k-l$, where $\hat{z}_{k-l|k-l}$ is conditioned on $\eta_{0:k-l}$. Note that in practice we perform this sampling already in the original forward RBPF, at each time step, thus decreasing storage requirements. The density $p(z_\tau, \eta_{0:\tau} | y_{0:k-l})$ is given by the original RBPF (8) using a time update to time index τ . At time index τ we augment the linear state vector to

$$\zeta_m = \begin{pmatrix} \zeta_m^1 \\ \zeta_m^2 \end{pmatrix} = \begin{pmatrix} z_m \\ z_\tau \end{pmatrix}, \quad (28)$$

and initialize it with

$$\zeta_\tau = \begin{pmatrix} \hat{z}_{\tau|k-l} \\ \hat{z}_{\tau|k-l} \end{pmatrix}, \quad \bar{P}_\tau = \begin{pmatrix} P_{\tau|k-l} & P_{\tau|k-l} \\ P_{\tau|k-l} & P_{\tau|k-l} \end{pmatrix}. \quad (29)$$

Then, for each $m \in [k-l+1, k-1]$, we run the supporting augmented RBPF, where $\zeta_{k-1}^2 = \hat{z}_{\tau|k-1}$ yields the linear, smoothed estimate and $\bar{P}_{k-1}^{2,2}$ (the lower right block in \bar{P}_{k-1}) the smoothed covariance at time t_{k-1} . Note that the samples $\{\eta_{0:\tau}^j\}_{j=1}^M$ must be kept track of throughout the recursion. Finally, at time t_k we use the estimates $\{\hat{z}_{k|k}^i, \eta_k^i\}_{i=1}^N$ as measurements. Thus, at the end of the recursion an approximation to the smoothing density is given by

$$p(z_\tau, \eta_{0:\tau} | x_k^i, y_{0:k}) \approx \sum_{j=1}^M q_{\tau|k,i}^j \mathcal{N}(z_\tau; \hat{z}_{\tau|k,i}^j, P_{\tau|k,i}^j) \delta_{\eta_{0:\tau}^j}(\eta_{0:\tau}). \quad (30)$$

Here, $q_{\tau|k,i}^j$ are the smoothed weights given measurements up to time t_{k-1} and the estimates at time t_k , x_k^i . Given (30), we find an approximation to (26) as

$$p(y_\tau | x_k^i, y_{0:k}) \approx \sum_{j=1}^M q_{\tau|k,i}^j p(y_\tau | \hat{z}_{\tau|k,i}^j, \eta_{0:\tau}^j), \quad (31)$$

with the measurement likelihood given by:

$$p(y_\tau | z_{\tau|k,i}^j, \eta_{0:\tau}^j) = \mathcal{N}(y_\tau; \hat{y}_{\tau|k,i}^j, \Sigma_{\tau|k,i}^j) \quad (32)$$

where $\hat{y}_{\tau|k,i}^j = h_\tau^j + C_\tau^j \hat{z}_{\tau|k,i}^j$, with $h_\tau^j = h(\eta_\tau^j)$ and similarly for C_τ^j and R_τ^j , and where $\Sigma_{\tau|k,i}^j = C_\tau^j P_{\tau|k,i}^j (C_\tau^j)^T + R_\tau^j$. With (31) inserted into (24), we have that the weights after the OOSM update, (23), are

$$w_{k|k,\tau}^i \propto w_k^i \sum_{j=1}^M q_{\tau|k,i}^j p(y_\tau | \hat{z}_{\tau|k,i}^j, \eta_{0:\tau}^j). \quad (33)$$

To find the linear estimates and their covariances after the update with the OOSM, and hence to find (25), we utilize results in [4] and [11] of how to calculate expected means and their covariances in RBPFs and update linear states with the OOSM, respectively. Then, for each i , the means and covariances are calculated using

$$\begin{aligned}\hat{z}_{k|k,\tau}^i &= \hat{z}_{k|k}^i + E^i \\ P_{k|k,\tau}^i &= P_{k|k}^i + \sum_{j=1}^M q_{\tau|k,i}^j \left((E^{j,i} - E^i)(E^{j,i} - E^i)^T \right. \\ &\quad \left. - W_{k,\tau}^{j,i} \Sigma_{\tau|k,i}^j (W_{k,\tau}^{j,i})^T \right) \\ E^i &= \sum_{j=1}^M q_{\tau|k,i}^j W_{k,\tau}^{j,i} e_{\tau}^{j,i}, \quad E^{j,i} = W_{k,\tau}^{j,i} e_{\tau}^{j,i} \\ e_{\tau}^{j,i} &= y_{\tau} - h_{\tau}^j - C_{\tau}^j \hat{z}_{\tau|k,i}^j \\ W_{k,\tau}^{j,i} &= P_{k,\tau|k,i}^j (C_{\tau}^j)^T (\Sigma_{\tau|k,i}^j)^{-1}.\end{aligned}\tag{34}$$

Here the crosscovariance between the state estimates at time index k and τ , $P_{k,\tau|k,i}^j$, is calculated in the smoother recursions as $\bar{P}_{k|k}^{2,1}$.

A summary of the algorithm, denoted SERBPF, is given in Algorithm 1. The storage requirements are $\{\{\hat{z}_{m|m}^j, P_{m|m}^j, \eta_m^j\}_{m=k-l_{\max}}^k, \{y_m\}_{m=k-l_{\max}+1}^k\}_{j=1}^M$, where l_{\max} is the predetermined maximum delay in the number of time steps. Further, the algorithm needs $\mathcal{O}(lM+N)$ operations.

Algorithm 1: SERBPF

- 1: **Input:** $\{\hat{z}_{k-l|k-l}^j, P_{k-l|k-l}^j, \eta_{k-l}^j, y_{k-l+1:k-1}\}_{j=1}^M$
 - 2: Predict up to time t_{τ} , yielding $p(z_{\tau}^j, \eta_{0:\tau}^j | y_{0:k-l})$.
 - 3: Set new weights as $q_{\tau|k-l}^j = w_{k-l}^j$.
 - 4: Augment the state vector as in (28) and initialize with (29).
 - 5: **for** $m = k - l + 1$ **to** $k - 1$ **do**
 - 6: Predict the particles to time t_m and perform particle filter measurement update according to measurement likelihood
 - 7: **if** $\frac{1}{\sum (q_m^j)^2} < N_{\text{eff}}$ **then**
 - 8: Resample M new particles, with replacement. Keep track of $\{\eta_{0:\tau}^j\}_{j=1}^M$, and equalize weights, $q_{\tau|m}^j = 1/M$.
 - 9: **end if**
 - 10: Perform augmented Kalman filter measurement update.
 - 11: **end for**
 - 12: At time t_k , use $\{\hat{z}_{k|k}^i, \eta_k^i\}_{i=1}^N$ as measurements, using (1) and (2) as measurement likelihoods, yielding $q_{\tau|k,i}^j$, $\hat{z}_{\tau|k,i}^j$, $P_{\tau|k,i}^j$, and $P_{k,\tau|k,i}^j$. Optionally, resample between the measurement updates.
 - 13: Update weights as in (33).
 - 14: Update linear estimates by applying (34).
-

Remark 1: This algorithm is a generalization and expansion of the storage efficient particle filter in [6], thus the name SERBPF. Note that if $\tau < 1$, steps 5–11 in the algorithm may be skipped.

B. Rao-Blackwellized OOSM Update Using Backward Simulation

Algorithm 1, SERBPF, uses a supporting RBPF while only saving a subset of the particles in order to keep the complexity low. Here we instead focus on estimation performance, thus taking a backward-simulation approach to the smoothing problem. This time, we rewrite $p(y_{\tau} | x_k^i, y_{0:k})$ as

$$\begin{aligned}p(y_{\tau} | x_k^i, y_{0:k}) &= \int p(y_{\tau} | z_{\tau}, \eta_{0:\tau}) \\ &\quad \times p(z_{\tau}, \eta_{\tau}, \eta_{0:k-1} | x_k^i, y_{0:k}) dz_{\tau} d\eta_{\tau} d\eta_{0:k-1}\end{aligned}\tag{35}$$

for later use in (24), where

$$\begin{aligned}p(z_{\tau}, \eta_{\tau}, \eta_{0:k-1} | x_k^i, y_{0:k}) \\ = p(z_{\tau}, \eta_{0:\tau} | \eta_{k-l:k-1}, x_k^i, y_{0:k}) p(z_{\tau} | \eta_{0:\tau}, \eta_{k-l:k-1}, x_k^i, y_{0:k}) \\ \times p(\eta_{0:\tau} | \eta_{k-l:k-1}, x_k^i, y_{0:k}) p(\eta_{k-l:k-1} | x_k^i, y_{0:k}),\end{aligned}$$

and as

$$\begin{aligned}p(y_{\tau} | x_k^i, y_{0:k}) &= \int p(y_{\tau} | z_{\tau}, \eta_{\tau}) \\ &\quad \times p(z_{\tau}, \eta_{\tau}, \eta_{k-l:k-1} | x_k^i, y_{0:k}) dz_{\tau} d\eta_{\tau} d\eta_{k-l:k-1}\end{aligned}\tag{36}$$

for later use in (25), where the second term on the right-hand side can be factorized similarly to the second term in (35). In both (35) and (36), we factorize $p(\eta_{k-l:k-1} | x_k^i, y_{0:k})$ in the same manner as (5):

$$\begin{aligned}p(\eta_{k-l:k-1} | x_k^i, y_{0:k}) &= p(\eta_{k-l} | \eta_{k-l+1:k-1}, x_k^i, y_{0:k}) \\ &\quad \times \prod_{m=k-l+1}^{k-1} p(\eta_m | \eta_{m+1:k-1}, x_k^i, y_{0:k}).\end{aligned}\tag{37}$$

This smoothing density can be solved for in a similar way as (12) is solved for in [19], however, with adaptations to the OOSM scenario as follows: At time t_k , instead of choosing $\eta_k' = \eta_k^j$ with probability w_k^j for $j = 1, \dots, M$, we choose $\eta_k' = \eta_k^i$ for each forward particle. Then, at time t_{k-1} , we draw D forward particles $\{\eta_{k-1}^d\}_{d=1}^D$ with probability w_{k-1}^d . For these D particles, we calculate the right-hand side of (15) using (16)–(18). The smoothing weights $w_{k-1|k,i}^d$ are then found by applying (13), which implies that we at this point have found (37) as

$$p(\eta_{k-1} | \eta_k^i, y_{0:k}) \approx \sum_{d=1}^D w_{k-1|k,i}^d \delta_{\eta_{k-1}^d}(\eta_{k-1}),$$

To proceed in the algorithm, we set $J = d$ with probability $w_{k-1|k,i}^d$ and set $\eta_{k-1:k}' = \{\eta_{k-1}^J, \eta_k^i\}$. This recursion is performed down to time t_{k-l} , for M times. At time t_{k-l} (37) approximates to

$$\begin{aligned}p(\eta_{k-l:k-1} | x_k^i, y_{0:k}) \\ \approx \frac{1}{M} \sum_{j=1}^M \sum_{d=1}^D w_{k-l|k,i}^{j,d} \delta_{\eta_{k-l}^d}(\eta_{k-l}).\end{aligned}\tag{38}$$

For later use, we note that by drawing a last sample, $\eta_{k-l}' = \eta_{k-l}^d$, we have also found an approximation of (37) based on the mean of the full backward trajectories as

$$p(\eta_{k-l:k-1} | x_k^i, y_{0:k}) \approx \frac{1}{M} \sum_{j=1}^M \delta_{\eta_{k-l:k-1}^j}(\eta_{k-l:k-1}).\tag{39}$$

Note that $\eta_{k-l:k-1}^j$ depends on x_k^i . To update the weights, we insert (38) into (35), yielding

$$p(y_\tau | x_k^i, y_{0:k}) \approx \frac{1}{M} \sum_{j=1}^M \sum_{d=1}^D w_{\tau|k,i}^{j,d} p(y_\tau | \hat{z}_{\tau|k-l}^d, \eta_{0:\tau}^d), \quad (40)$$

where $w_{\tau|k,i}^{j,d} = w_{k-l|k,i}^{j,d}$ since only a time update differs between t_τ and t_{k-l} . By insertion of (40) into (24) and applying (23), we may calculate the weights after processing the l -step lag OOSM as

$$w_{k|k,\tau}^i \propto w_k^i \sum_{j=1}^M \sum_{d=1}^D w_{\tau|k,i}^{j,d} p(y_\tau | \hat{z}_{\tau|k-l}^d, \eta_\tau^d), \quad (41)$$

where $p(y_\tau | \hat{z}_{\tau|k-l}^d, \eta_\tau^d)$ is calculated similarly to (32).

To update the linear density $p(z_k | \eta_k, y_{0:k}, y_\tau)$ (i.e., (25)), we need the smoothing density $p(z_\tau | \eta_\tau, \eta_{k-l:k-1}^j, x_k^i, y_{0:k})$ in the measurement update step (36). For this we can resort to different linear smoothers (conditioned on the generated backward trajectories and the measurements), iterating back to time t_{k-l} and then performing a time update to t_τ . Here, we choose an RTS-smoother, which for the mixed linear/nonlinear model class is given as follows:

$$\begin{aligned} \hat{z}_{m|k} &= \hat{z}_{m|m} + H_m (\hat{z}_{m+1|k} - \hat{z}_{m+1|m}) \\ P_{m|k} &= P_{m|m} + H_m (P_{m+1|k} - P_{m+1|m}) \\ P_{m,k|m} &= P_{m+1,k|m+1} H_m^T \\ H_m &= P_{m|m}^* A_m^T P_{m+1|m}^{-1} \\ \hat{z}_{m+1|m} &= f_m + A_m \hat{z}_{m|m} \\ P_{m+1|m} &= A_m P_{m|m}^* A_m^T + F_m F_m^T \\ \hat{z}_{m|m}^* &= \hat{z}_{m|m} + L_m (\eta_{m+1} - g_m - A_m \hat{z}_{m|m}) \\ P_{m|m}^* &= P_{m|m} - L_m N_m L_m^T \\ L_m &= P_{m|m} B_m N_m^{-1} \\ N_m &= B_m P_{m|m} B_m^T + G_m G_m^T. \end{aligned} \quad (42)$$

Equations (42) follow from a derivation using a Bayesian setting similar to that of the Kalman filter derivations for the RBPF [4], but it is omitted due to lack of space. After the RTS-backward recursions we conclude that

$$p(z_\tau | \eta_\tau, \eta_{k-l:k-1}^j, x_k^i, y_{0:k}) = \mathcal{N}(z_\tau; \hat{z}_{\tau|k,i}^j, P_{\tau|k,i}^j). \quad (43)$$

With (39) and (43) inserted in (36), we get

$$p(y_\tau | x_k^i, y_{0:k}) \approx \frac{1}{M} \sum_{j=1}^M p(y_\tau | \eta_\tau^j, \hat{z}_{\tau|k,i}^j), \quad (44)$$

where, again, the measurement likelihood is calculated similarly to (32). Finally, we update the linear estimates and the associated covariances, and thereby find (25) using the

measurement update (44), as

$$\begin{aligned} \hat{z}_{k|k,\tau}^i &= \hat{z}_{k|k}^i + \frac{1}{M} \sum_{j=1}^M W_{k,\tau}^{j,i} e_\tau^{j,i} \\ P_{k|k,\tau}^i &= P_{k|k}^i - \frac{1}{M} \sum_{j=1}^M W_{k,\tau}^{j,i} \Sigma_{\tau|k,i}^j (W_{k,\tau}^{j,i})^T \\ W_{k,\tau}^{j,i} &= P_{k,\tau|k,i}^j C_\tau^T (\Sigma_{\tau|k,i}^j)^{-1} \\ \Sigma_{\tau|k,i}^j &= C_\tau^{j,i} P_{\tau|k,i}^j (C_\tau^{j,i})^T + R_\tau \\ e_\tau^{j,i} &= y_\tau - h_\tau^{j,i} - C_\tau^{j,i} \hat{z}_{\tau|k,i}^j. \end{aligned} \quad (45)$$

A summary is given in Algorithm 2, denoted RBOOSMBS. The storage requirements are $\{\hat{z}_{m|m}^j, P_{m|m}^j, \eta_m^j, y_m\}_{j=1}^{\max(M,D)}$ for $m = k - l_{\max}, \dots, k$, and requires roughly $\mathcal{O}(lNM D)$ operations, as compared with $\mathcal{O}(lM + N)$ for SERBPF and $\mathcal{O}((l-1)N^3 + N^2)$ for A-PF in [5].

Remark 2: By sampling $D \leq N$ particles in Algorithm 2 we can trade tracking performance against complexity. For example, for certain densities it may suffice with $D \ll N$ particles, thus saving processing time. We can also use it as a tradeoff with the number of smoothing iterations M .

Remark 3: If the measurements are stored it is theoretically possible to reorder and reprocess the measurements once an OOSM arrives. However, reprocessing includes redoing the data association, which in itself can be an overwhelming task [5]. Further, SERBPF is still faster than reordering and reprocessing.

Algorithm 2: RBOOSMBS

- 1: **Input:** $\{\hat{z}_{m|m}^i, P_{m|m}^i, \eta_m^i, y_m\}_{i=1}^N$ where $k-l \leq m \leq k$.
 - 2: **for** $i = 1$ **to** N **do**
 - 3: **for** $j = 1$ **to** M **do**
 - 4: Set $\eta_k^j = \eta_k^i$.
 - 5: **for** $m = k-1$ **to** $k-l$ **do**
 - 6: **for** $d = 1$ **to** D **do**
 - 7: Sample forward particle η_{k-1}^d with probability w_{k-1}^d .
 - 8: Calculate (15) using (16)–(18).
 - 9: Find unnormalized smoothing weight $w_{m|k,i}^d$ using (13).
 - 10: **end for**
 - 11: Normalize weights, $w_{m|k,i}^d = w_{m|k,i}^d / (\sum_{d=1}^D w_{m|k,i}^d)$.
 - 12: Set $J = d$ with probability $w_{m|k,i}^d$, and set $\eta_{m:k}^j = \{\eta_m^J, \eta_{m+1:k-1}^j, \eta_k^i\}$.
 - 13: Perform a backward RTS step using (42) conditioned on $\eta_{m:k}^j$.
 - 14: **end for**
 - 15: **end for**
 - 16: Update weight w_k^i using (41), yielding $w_{k|k,\tau}^i$.
 - 17: Update mean and covariance using (45).
 - 18: **end for**
-

V. NUMERICAL RESULTS

We evaluated the proposed algorithms on two examples by comparing their performance against two recently reported

particle filter algorithms. For a fair comparison of the algorithms' abilities to process the OOSMs, all filters used identical bootstrap RBPFs in the forward direction [4]. We used the root-mean square error (RMSE) at each time step and the time average of it as performance measures. The time-averaged RMSE is found by taking the mean of the RMSE.

The methods compared were: An RBPF discarding all OOSMs (RBPFDISC); an idealized RBPF collecting all measurements from both sensors with zero delay (RBPF); the particle filter derived from exact Bayesian inference in [5] (A-PF); the modified storage efficient particle filter described in [7] (SEPFPPS), which uses an augmented-state fixed-point Extended Kalman smoother for processing the OOSMs; the first method proposed in this paper, described in Sec. IV and summarized in Algorithm 1 (SERBPF); and the proposed backward-simulation based method described in Sec. IV and summarized in Algorithm 2 (RBOOSMBS).

A. Example 1

We consider the fourth order mixed linear/nonlinear system

$$z_{k+1} = \begin{pmatrix} 1 & 0.3 & 0 \\ 0 & 0.92 & -0.3 \\ 0 & 0.3 & 0.92 \end{pmatrix} z_k + 0.1v_k^z, \quad (46)$$

$$\eta_{k+1} = \arctan(\eta_k) + (1 \ 0 \ 0)z_k + 0.1v_k^\eta, \quad (47)$$

$$y_k = \begin{pmatrix} 0.1\eta_k^2 \text{sign}(\eta_k) \\ 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ 1 & -1 & 1 \end{pmatrix} z_k + e_k, \quad (48)$$

where $\text{sign}(\cdot)$ is the signum function. The process noise is considered white, Gaussian, and mutually independent, and $e_k \sim \mathcal{N}(0, 0.1I_{2 \times 2})$. This model has previously been used in, for example, [21]. Measurement two (i.e., the second element in y_k) is assumed to have communication issues: A measurement arrives with probability $p_{\text{OOSM}} = 0.5$. Further, when the measurement arrives, it is delayed according to a discrete uniform distribution in the interval $[1, 4]$. Note that since only 50 % of the measurements arrive, the performance of RBPF is impossible to achieve with either of the proposed methods.

In Fig. 2 we show the RMSE using $N = 200$ particles in the forward filters for all four states, numbered row-wise from the top left. Further, SERBPF uses $M = 200$ particles in the supporting RBPF and RBOOSMBS uses $M = 1, D = N$. The number of time steps are $T = 50$. We used 10000 Monte Carlo simulations to obtain the results. Clearly, RBOOSMBS gives the most accurate results in terms of RMSE, followed by SERBPF. For the delays in this example, the performance difference between RBOOSMBS and SERBPF for the nonlinear state is small. However, it can be expected to increase when the delay increases. Note that there is a rather significant performance difference for the linear states, caused by that we condition on the nonlinear, backward simulated states in RBOOSMBS. The low performance of A-PF is due to that 200 particles is not enough to reliably estimate the smoothing weights in all four dimensions when the model structure in (46)–(48) is unexploited. Further, SEPFPPS hardly outperforms RBPFDISC, caused by too few particles in the RBPF.

For this particular example, we found that SERBPF executed the simulations ten times faster than RBOOSMBS, which in its turn was a factor five faster than A-PF.

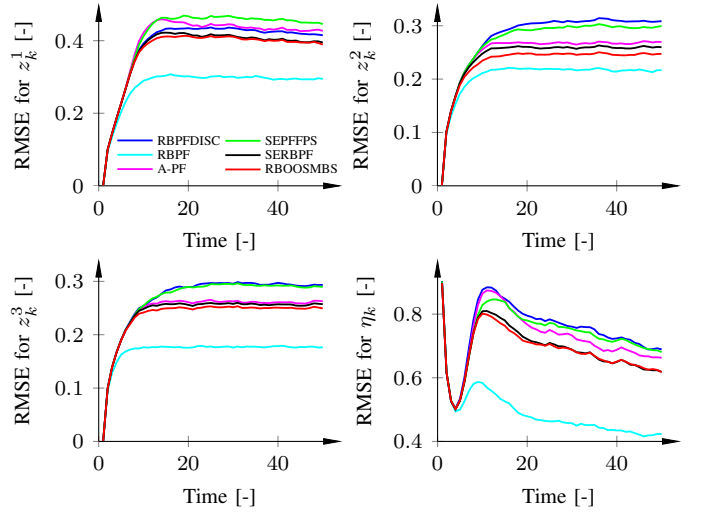


Fig. 2. RMSE of the four states in Sec. V-A numbered row-wise from the top left with the nonlinear state in the second row, second column. We executed 10000 Monte Carlo simulations. The number of particles is set to $N = 200$. SERBPF uses $M = 200$ particles in the supporting RBPF. RBOOSMBS uses $M = 1$ backward trajectory, something we noticed is enough in most cases for the lags considered. RBOOSMBS performs best, followed by SERBPF. Note that SERBPF consistently outperforms A-PF, despite its computational simplicity.

B. Example 2

In this evaluation we use a fifth order mixed linear/nonlinear system. The nonlinear part is given by

$$\eta_{k+1} = 0.5\eta_k + \theta_k \frac{\eta_k}{1 + \eta_k^2} + 8 \cos(1.2k) + 0.07v_k^\eta, \quad (49)$$

$$y_k = 0.05\eta_k^2 + e_k, \quad (50)$$

where $e_k \sim \mathcal{N}(0, 1)$. The case with $\theta_k = 25$ has been used in several papers, among them [5]. Here, θ_k is the output from a linear system with dynamics given by

$$z_{k+1} = \begin{pmatrix} 3 & -1.691 & 0.849 & -0.3201 \\ 2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0.5 & 0 \end{pmatrix} z_k + 0.1v_k^z, \quad (51)$$

$$\theta_k = 25 + (0 \ 0.04 \ 0.044 \ 0.008)z_k. \quad (52)$$

Again, the process noise is white, Gaussian, and mutually independent. Combined, (49)–(52) is a mixed linear/nonlinear system, and as such it has previously been used in, for example, [19]. We ran the simulations with 20 time steps in each simulation. We generated two data sets by executing 10000 Monte Carlo simulations twice. In both data sets we let $N = 400$. In the first data set we let every second measurement be delayed one time step (i.e., $l = 1$), whereas in the second data set we let every third measurement be delayed two time steps (i.e., $l = 2$). Note that in this example all measurements arrive. Hence, at the OOSM arrival times the performance of RBPF should, at least theoretically for a sufficiently large number of particles, be possible to achieve. In this example, SERBPF uses $M = N$ particles in the supporting RBPF, and RBOOSMBS uses $M = 1, D = N$ for the smoothing.

We present the time-averaged RMSE values at the OOSM arrival times (i.e., $k = 1, 3, \dots, 19$ and $k = 1, 4, \dots, 19$, respectively) in Table I for η_k and θ_k . The results of SEPFPPS

TABLE I. TIME AVERAGED RMSE VALUES AT THE OOSM ARRIVAL TIMES USING 10000 MONTE CARLO EXECUTIONS FOR THE EXAMPLE IN SEC. V-B. THE DELAYS USED ARE INDICATED IN THE TOP ROW.

Algorithm	$l = 1$		$l = 2$	
	η_k	θ_k	η_k	θ_k
RBPFDISC	0.189	0.607	0.205	0.525
A-PF	0.180	0.598	-	-
SERBPF	0.162	0.537	0.194	0.516
RBOOSMBS	0.162	0.548	0.192	0.515
RBPF	0.156	0.531	0.189	0.513

have been omitted due to inadequate handling of multimodal distributions. Further, for the data set with two samples delay (columns three and four in Table I), A-PF was omitted because of its computational complexity. We observe that the tracking performances of RBOOSMBS and SERBPF are very close to RBPF for both delay settings, which also are the tracking performances achievable in theory. Moreover, there is only minor advantage in using A-PF compared to discarding the OOSMs. For $l = 1$ the smoothing in SERBPF yields a better approximation for the linear states than using $M = 1$ in RBOOSMBS, which is the reason why the RMSE for θ_k is smaller for SERBPF. The last two columns show that RBOOSMBS has a better relative tracking performance compared to SERBPF when we increase the OOSM delay, implying better robustness.

C. Discussion

During the extensive simulations we noticed that for the delays considered it was often enough to only use one backward trajectory in RBOOSMBS, yielding complexity $\mathcal{O}(lND)$ and thus possibly opening up for online applications. For the examples considered, a suitable number of particles in the supporting RBPF in SERBPF considering the tradeoff between performance and computation time, turned out to be $N/2 \leq M \leq N$.

We found that for small OOSM delays, the differences in tracking performance between the two proposed methods were minor. This is no surprise since for small delays both approaches compute good approximations of the smoothing density. For small delays, then, SERBPF is the best option; at least when computational complexity is taken into account. However, for larger delays RBOOSMBS clearly outperforms SERBPF, as seen in Sec. V-A.

VI. CONCLUSIONS

We derived and presented two new algorithms for OOSM processing considering the class of mixed linear/nonlinear state-space models, which is a model structure that is frequently used in positioning and navigation applications. The two approaches both use Rao-Blackwellization to exploit the conditionally linear Gaussian substructure in the model. They differ in the way they tackle the smoothing problem; one being focused on fast execution, and the other aiming at tracking performance. Simulation examples have shown that both approaches yield improvements in terms of RMSE when comparing to recent particle filter algorithms for OOSM processing. SERBPF is the most viable option when considering online applications. However, the results indicate that for small lags it is often sufficient to set $M = 1$ in RBOOSMBS, possibly enabling real-time applications also for this algorithm.

REFERENCES

- [1] Y. Bar-Shalom, "Update with out-of-sequence measurements in tracking: exact solution," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 38, no. 3, pp. 769–777, 2002.
- [2] M. M. Muntzinger, M. Aeberhard, S. Zuther, M. Maehlich, M. R. Schmid, and K. Dietmayer, "Reliable automotive pre-crash system with out-of-sequence measurement processing," in *Proc. of IEEE Intell. Vehicles Symp.*, San Diego, CA, June 2010, pp. 1022–1027.
- [3] K. Berntorp, K.-E. Årzén, and A. Robertsson, "Sensor fusion for motion estimation of mobile robots with compensation for out-of-sequence measurements," in *11th Intl. Conf. on Control, Automation, and Systems*, KINTEX, Gyeonggi-do, Korea, October 2011.
- [4] T. B. Schön, F. Gustafsson, and P.-J. Nordlund, "Marginalized particle filters for mixed linear nonlinear state-space models," *IEEE Trans. Signal Process.*, vol. 53, pp. 2279–2289, 2005.
- [5] S. Zhang and Y. Bar-Shalom, "Out-of-sequence measurement processing for particle filter: Exact Bayesian solution," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 48, no. 4, pp. 2818–2831, 2012.
- [6] U. Orguner and F. Gustafsson, "Storage efficient particle filters for the out of sequence measurement problem," in *11th Intl. Conf. Information Fusion*, Cologne, Germany, June 2008, pp. 1–8.
- [7] K. Berntorp, K.-E. Årzén, and A. Robertsson, "Storage efficient particle filters with multiple out-of-sequence measurements," in *15th Intl. Conf. on Information Fusion*, Singapore, July 2012, pp. 471–478.
- [8] Y. Bar-Shalom, H. Chen, and M. Mallick, "One-step solution for the multistep out-of-sequence-measurement problem in tracking," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 40, no. 1, pp. 27–37, 2004.
- [9] S. Zhang, Y. Bar-Shalom, and G. Watson, "Tracking with multisensor out-of-sequence measurements with residual biases," in *13th Intl. Conf. on Information Fusion*, Edinburgh, UK, July 2010, pp. 1–8.
- [10] X. Shen, Y. Zhu, E. Song, and Y. Luo, "Optimal centralized update with multiple local out-of-sequence measurements," *IEEE Trans. Signal Process.*, vol. 57, no. 4, pp. 1551–1562, 2009.
- [11] S. Zhang and Y. Bar-Shalom, "Optimal update with multiple out-of-sequence measurements with arbitrary arriving order," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 48, no. 4, pp. 3116–3132, 2012.
- [12] M. Orton and A. Marrs, "Particle filters for tracking with out-of-sequence measurements," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 41, no. 2, pp. 693–702, 2005.
- [13] M. Mallick, T. Kirubarajan, and S. Arulampalam, "Out-of-sequence measurement processing for tracking ground target using particle filters," in *IEEE Aerosp. Conf. Proc.*, vol. 4, 2002, pp. 1809–1818.
- [14] B. Oreshkin, X. Liu, and M. Coates, "Efficient delay-tolerant particle filtering," *IEEE Trans. Signal Process.*, vol. 59, no. 7, pp. 3369–3381, 2011.
- [15] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 174–188, 2002.
- [16] G. Kitagawa, "Monte Carlo filter and smoother for non-Gaussian nonlinear state space models," *Journal of Computational and Graphical Statistics*, vol. 5, no. 1, pp. 1–25, 1996.
- [17] S. J. Godsill, A. Doucet, and M. West, "Monte Carlo smoothing for non-linear time series," *Journal of the American Statistical Association*, vol. 99, pp. 156–168, 2004.
- [18] A. Doucet, S. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statistics and Computing*, vol. 10, no. 3, pp. 197–208, 2000.
- [19] F. Lindsten, P. Bunch, S. J. Godsill, and T. B. Schön, "Rao-Blackwellised particle smoothers for mixed linear/nonlinear state-space models," in *38th Intl. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, Vancouver, Canada, May 2013.
- [20] B. D. O. Anderson and J. B. Moore, *Optimal Filtering*. Prentice Hall, 1979.
- [21] F. Lindsten and T. B. Schön, "Identification of mixed linear/nonlinear state-space models," in *49th IEEE Conf. on Decision and Control (CDC)*, Grand Wailea, Maui, Hawaii, December 2010, pp. 6377–6382.