Rao-Blackwellised particle smoothers for mixed linear/nonlinear state-space models

Fredrik Lindsten, Thomas B. Schön

Division of Automatic Control

E-mail: lindsten@isy.liu.se, schon@isy.liu.se

30th May 2011

Report no.: LiTH-ISY-R-3018
Submitted to IEEE Transactions on Signal Processing

Address:
Department of Electrical Engineering
Linköpings universitet
SE-581 83 Linköping, Sweden

WWW: http://www.control.isy.liu.se

AUTOMATIC CONTROL
REGLERTEKNIK
LINKÖPINGS UNIVERSITET

Abstract

We consider the smoothing problem for a class of mixed linear/nonlinear state-space models. This type of models contain a certain tractable substructure. When addressing the filtering problem using sequential Monte Carlo methods, it is well known that this structure can be exploited in a Rao-Blackwellised particle filter. However, to what extent the same property can be used when dealing with the smoothing problem is still a question of central interest. In this paper, we propose different particle based methods for addressing the smoothing problem, based on the forward filtering/backward simulation approach to particle smoothing. This leads to a group of Rao-Blackwellised particle smoothers, designed to exploit the tractable substructure present in the model.

Keywords: Nonlinear estimation, smoothing, particle methods, Rao-Blackwellisation

Rao-Blackwellised particle smoothers for mixed linear/nonlinear state-space models

Fredrik Lindsten, *Student Member, IEEE,* and Thomas B. Schön, *Senior Member, IEEE,*

***Abstract*—We consider the smoothing problem for a class of mixed linear/nonlinear state-space models. This type of models contain a certain tractable substructure. When addressing the filtering problem using sequential Monte Carlo methods, it is well known that this structure can be exploited in a Rao-Blackwellised particle filter. However, to what extent the same property can be used when dealing with the smoothing problem is still a question of central interest. In this paper, we propose different particle based methods for addressing the smoothing problem, based on the forward filtering/backward simulation approach to particle smoothing. This leads to a group of Rao-Blackwellised particle smoothers, designed to exploit the tractable substructure present in the model.**

***Index Terms*—Nonlinear estimation, smoothing, particle methods, Rao-Blackwellisation.**

I. Introduction

Sequential Monte Carlo (SMC) methods, or particle filters (PFs), have shown to be powerful tools for solving nonlinear and/or non-Gaussian filtering problems; see e.g. [1]–[3] for an introduction. Since the introduction of the PF by [4], we have experienced a vast amount of research in the area. For instance, many improvements and extensions have been introduced to increase the accuracy of the filter, see e.g. [2] for an overview of recent developments. One natural idea is to exploit any tractable substructure in the model [5]–[7]. More precisely, if the model, conditioned on one partition of the state, behaves like e.g. a linear Gaussian state-space (LGSS) model it is sufficient to employ particles for the intractable part and make use of the analytic tractability for the remaining part. Inspired by the Rao-Blackwell theorem, this has become known as the Rao-Blackwellised particle filter (RBPF).

With a foundation in SMC, various particle methods for addressing other types of state inference problems have emerged as well. When dealing with marginal, fixed-interval and joint smoothing, a few different approaches have been considered, most notably those based on forward/backward smoothing [5], [8]–[10] and two-filter smoothing [11], [12].

For the filtering problem, the Rao-Blackwellisation idea can be applied rather straightforwardly. The reason is that the specific types of models that are considered, are "conditionally tractable in the forward direction". This property can thus be exploited in the (forward in time) filtering recursions. However, to what extent and in which ways, the same property can be exploited when addressing the smoothing problem, is still a question of central interest. In [13], a Rao-Blackwellised particle smoother (RBPS) based on the forward/backward approach, has been proposed for a certain type of hierarchical state-space models. The same model class is considered in [11], where a two-filter RBPS i proposed. In this paper, we continue this line of work and consider the problem of Rao-Blackwellised particle smoothing in a type of mixed linear/nonlinear Gaussian state-space models.

The remaining of this paper is outlined as follows. In Section II we introduce the smoothing problem and the class of models which we will be dealing with. We then provide a preview of the contributions of this paper in Section III. In Section IV we review some standard methods for particle filtering and smoothing, before we turn to the derivation of two Rao-Blackwellised particle smoothers in Section V and Section VI, respectively. In Section VII we discuss some of the properties of these smoothers, and in Section VIII they are evaluated in numerical examples. Finally, in Section IX we draw conclusions.

II. Problem formulation

To simplify the presentation, all distributions are assumed to have densities w.r.t. Lebesgue measure. The conditional distribution of any variable x conditioned on some other variable y will be denoted $p(dx \mid y) \triangleq \mathrm{P}(x \in dx \mid y)$ and the density of this distribution w.r.t. Lebesgue measure is written $p(x \mid y)$.

Let $\{x_t\}_{t\geq 1}$ be the state process in a state-space model (SSM). That is, $\{x_t\}_{t\geq 1}$ is a discrete-time Markov process evolving according to a transition density $p(x_{t+1} \mid x_t)$. The process is hidden, but observed through the measurements y_t. Given x_t, the measurements are conditionally independent and also independent of the state process x_s, $s \neq t$. Hence, the SSM is described by,

$$x_{t+1} \sim p(x_{t+1} \mid x_t), \tag{1a}$$

$$y_t \sim p(y_t \mid x_t). \tag{1b}$$

The *fixed-interval smoothing distribution*,

$$p(dx_{s:t} \mid y_{1:T}), \tag{2}$$

for some $s \leq t \leq T$ is the posterior distribution of the states $x_s, \ldots, x_t$ given a sequence of measurements $y_{1:T} \triangleq \{y_1, \ldots, y_T\}$ (a similar notation is used for other sequences as well). If we set $s = 1$ and $t = T$, (2) coincides with the *joint smoothing distribution*, i.e. the distribution of the full state sequence $x_{1:T}$ conditioned on the measurements $y_{1:T}$. This is the "richest" smoothing distribution, since it can be marginalised

F. Lindsten and T. B. Schön are with the Division of Automatic Control, Linköping University, SE-581 83, Linköping, Sweden, e-mail: {lindsten,schon}@isy.liu.se

This work was supported by CADICS, a Linnaeus center funded by the Swedish Research Council and the project Calibrating Nonlinear Dynamical Models (621-2010-5876) funded by the Swedish Research Council.

Manuscript received May 31, 2011.

to obtain any fixed-interval or marginal smoothing distribution. In this paper we will in particular seek the marginal smoothing distribution $p(dx_t \mid y_{1:T})$ for $t = 1, \ldots, T$. The reason for this is for notational convenience, since it would be cumbersome to address the (more general) joint smoothing distribution in all cases. However, the methods presented here can all be extended to fixed-interval or joint smoothing.

In this paper, we shall study the smoothing problem for a special case of (1). More precisely, we assume that the state can be partitioned according to $x_t = \begin{pmatrix} \xi_t^\mathsf{T} & z_t^\mathsf{T} \end{pmatrix}^\mathsf{T}$, where the z-process is conditionally linear Gaussian. Hence, conditioned on the *trajectory* $\xi_{1:t}$, the z-process follows a linear Gaussian state-space (LGSS) model. Models with this property will be denoted conditionally linear Gaussian state-space (CLGSS) models. We shall call ξ_t the *nonlinear* state and z_t the *linear* state.

In particular, we will consider a specific type of CLGSS models, denoted mixed linear/nonlinear Gaussian. A mixed linear/nonlinear Gaussian state-space model can be expressed on functional form as,

$$\xi_{t+1} = f_\xi(\xi_t) + A_z(\xi_t) z_t + v_{\xi,t}, \tag{3a}$$

$$z_{t+1} = f_z(\xi_t) + A_z(\xi_t) z_t + v_{z,t}, \tag{3b}$$

$$y_t = h(\xi_t) + C(\xi_t) z_t + e_t, \tag{3c}$$

where the process noise is white and Gaussian according to

$$v_t = \begin{bmatrix} v_{\xi,t} \\ v_{z,t} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} Q_\xi(\xi_t) & Q_{\xi z}(\xi_t) \\ (Q_{\xi z}(\xi_t))^\mathsf{T} & Q_z(\xi_t) \end{bmatrix} \right), \tag{4}$$

and the measurement noise is white and Gaussian according to $e_t \sim \mathcal{N}(0, R(\xi_t))$. For each time $t \geq 0$, the model can be seen as an LGSS model if we fix the nonlinear state trajectory $\xi_{1:t}$ up to that time. Note that there is a connection between the ξ-state and the z-state through the dynamic equation for the nonlinear state (3a). Hence, if we fix (i.e. condition on) the nonlinear state process, (3a) can be seen as an "extra measurement", containing information about the linear z-state. Note that it is necessary to condition on the entire trajectory $\xi_{1:t}$ to attain the linear Gaussian substructure, i.e. to condition on just ξ_t is not sufficient.

We will also make use of a more compact reformulation of (3) according to,

$$x_{t+1} = f(\xi_t) + A(\xi_t) z_t + v_t, \tag{5a}$$

$$y_t = h(\xi_t) + C(\xi_t) z_t + e_t, \tag{5b}$$

with

$$x_t = \begin{bmatrix} \xi_t \\ z_t \end{bmatrix}, \quad f(\xi_t) = \begin{bmatrix} f_\xi(\xi_t) \\ f_z(\xi_t) \end{bmatrix}, \quad A(\xi_t) = \begin{bmatrix} A_\xi(\xi_t) \\ A_z(\xi_t) \end{bmatrix}. \tag{5c}$$

Remark 1. In this paper, we focus the derivation of the proposed smoothers on mixed linear/nonlinear Gaussian state-space models, as defined above. Note, however, that this specific model is only a special case of the general class of CLGSS models. The results presented in this paper can straightforwardly be modified to any other CLGSS model. The reasons for why we choose to work with mixed linear/nonlinear models are; **i)** to obtain explicit algorithms, which would not be possible if we were to address the most general CLGSS model **ii)** mixed linear/nonlinear models highlight all the challenging parts of the derivations **iii)** mixed linear/nonlinear models constitutes an often encountered and important special case. □

It is a well known fact that the conditionally linear Gaussian substructure in a CLGSS model can be exploited when addressing the filtering problem using SMC methods, i.e. particle filters. This leads to the RBPF; see Section IV-C and [5]–[7]. However, to what extent and in which ways, the same property can be exploited when addressing the smoothing problem, is still a question of central interest. It is the purpose of this paper to propose functioning, SMC based algorithms for this problem, and also to highlight some of the difficulties that arise when dealing with it.

III. A PREVIEW OF THE CONTRIBUTIONS

As mentioned in the previous section, we will consider the problem of Rao-Blackwellised particle smoothing for CLGSS models and in particular for mixed linear/nonlinear Gaussian state-space models. We will focus on "backward simulation" type of smoothers [9], [10]. Basically, a backward simulator is a forward filtering/backward smoothing particle method. A PF (see Section IV-A) is run forward in time on a fix data sequence $y_{1:T}$. The output from the PF is then used to approximate a *backward kernel*, used to simulate approximate realisations from the joint smoothing distribution. The output from the backward simulator is thus (generally) a collection of *backward trajectories*, $\{\tilde{x}_{1:T}^j\}_{j=1}^M$ which are approximate realisation from the joint smoothing distribution. We will discuss this further in Section IV-B. Based on these backward trajectories, we can construct an empirical distribution approximating the joint smoothing distribution according to,

$$p(dx_{1:T} \mid y_{1:T}) \approx \frac{1}{M} \sum_{j=1}^{M} \delta_{\tilde{x}_{1:T}^j}(dx_{1:T}). \tag{6}$$

Now, as previously mentioned, we wish to do this in a way which exploits the CLGSS structure of the model, leading to a Rao-Blackwellised particle smoother (RBPS). However, we will see that, as opposed to the RBPF, there is no single "natural" way to construct an RBPS. Due to this, we will in this paper propose and discuss two different RBPS, based on the backward simulation idea. In a numerical evaluation (see Section VIII), it is shown that the methods have similar accuracy and that they both improve the performance over standard particle smoothing techniques.

The first is an extension of the RBPS previously proposed by [13]. This smoother simulates backward trajectories jointly for both the nonlinear state *and* the linear state, i.e. the smoother generates a collection of joint backward trajectories $\{\tilde{x}_{1:T}\}_{i=1}^M = \{\tilde{\xi}_{1:T}^i, \tilde{z}_{1:T}^i\}_{i=1}^M$, targeting the joint smoothing distribution. This results in a "joint point-mass representation" of the joint smoothing distribution in complete analogy with (6). Note that this is quite different from the representation of the filtering distribution generated by the RBPF. This method will be denoted joint backward simulation RBPS (JBS-RBPS). The difference between the JBS-RBPS and a "non-Rao-Blackwellised" backward simulator is that the former uses an

RBPF to approximate the backward kernel, whereas the latter uses a PF. In [13], a class of hierarchical CLGSS models is considered, in which the transition kernel of the nonlinear state process $\{\xi_t\}_{t\geq 1}$ is independent of the linear state process $\{z_t\}_{t\geq 1}$, i.e. it is only applicable to mixed linear/nonlinear models (3) in case $A_\xi \equiv Q_{\xi z} \equiv 0$. In Section V we extend this smoother to fully interconnected mixed linear/nonlinear models (3).

Furthermore, in Section V-C we propose an extension to the JBS-RBPS. Here, we complement the backward simulation with a constrained smoothing of the linear states. This replaces the point-mass representation of the linear states with a continuous representation, by evaluating the conditional smoothing densities for the linear states. Hence, instead of using an approximation of the form (6), we can then approximate the marginal smoothing distribution according to,

$$p(d\xi_t, dz_t \mid y_{1:T}) \approx \frac{1}{M}\sum_{j=1}^{M} \mathcal{N}\left(dz_t; \tilde{z}^j_{t|T}, \widetilde{P}^j_{t|T}\right) \delta_{\tilde{\xi}^j_t}(d\xi_t), \tag{7}$$

for some means and covariances, $\{\tilde{z}^j_{t|T}\}_{j=1}^M$ and $\{\widetilde{P}^j_{t|T}\}_{j=1}^M$, respectively. This representation more closely resembles the RBPF representation of the filtering distribution.

One obvious drawback with this "add-on" to the JBS-RBPS, is that we need to process the data twice, i.e. we make two consecutive forward filtering/backward smoothing passes. A natural question is then; can we make a single forward/backward smoothing pass and obtain a representation of the marginal smoothing distribution similarly to (7)? This will be the topic of Section VI. Here we propose an RBPS which aims at sampling backward trajectories only for the nonlinear state (just as the RBPF samples forward trajectories only for the nonlinear state). The mean and covariance functions for the linear state are updated simultaneously with the backward simulation, conditioned on the nonlinear backward trajectories. However, to enable this we are forced to make certain approximations, which will be discussed in Section VI and in Section VII. This smoother will be referred to as marginal backward simulation RBPS (MBS-RBPS). We have previously used a preliminary form of the MBS-RBPS for parameter estimation in mixed linear/nonlinear models in [14].

IV. Particle Filtering and Smoothing

Before we continue with the derivation of the RBPS mentioned in the previous section, we review some standard particle methods for filtering and smoothing. This is done to give a self-contained presentation and to introduce all the required notation. Readers familiar with this material may consider to skip this section.

A. Particle filter

Let $p(x_1)$ be a given prior density of the state process. The filtering density and the joint smoothing density can then be expressed recursively using the Bayesian filtering and smoothing recursions, respectively. Using the convention $p(x_1 \mid y_{1:0}) \triangleq p(x_1)$, the latter recursion is given by the two-step updating formulas,

$$p(x_{1:t} \mid y_{1:t}) \propto p(y_t \mid x_t) p(x_{1:t} \mid y_{1:t-1}), \tag{8a}$$

$$p(x_{1:t+1} \mid y_{1:t}) = p(x_{t+1} \mid x_t) p(x_{1:t} \mid y_{1:t}), \tag{8b}$$

for any $t \geq 1$. Despite the simplicity of these expressions, they are known to be intractable for basically all cases, except for LGSS models and models with finite state-spaces. In the general case, approximate methods for computing the filtering or smoothing densities are required. One popular approach is to employ SMC methods, commonly referred to as particle filters (PFs); see e.g. [1]–[4].

The essence of these methods is to approximate a sequence of probability distributions with empirical point-mass distributions. In the PF, a sequence of weighted particle systems $\{x^i_{1:t}, w^i_t\}_{i=1}^N$ for $t = 1, 2, \ldots$ is generated, each defining an empirical distribution approximating the joint smoothing distribution at time t according to,

$$p(dx_{1:t} \mid y_{1:t}) \approx \widehat{p}(dx_{1:t} \mid y_{1:t}) \triangleq \sum_{i=1}^{N} w^i_t \delta_{x^i_{1:t}}(dx_{1:t}). \tag{9}$$

We have, without loss of generality, assumed that the importance weights $\{w^i_t\}_{i=1}^N$ are normalised to sum to one.

The basic procedure for generating these particle systems is as follows. Assume that we have obtained a weighted particle system $\{x^i_{1:t-1}, w^i_{t-1}\}_{i=1}^N$ targeting the joint smoothing distribution at time $t-1$. We then proceed to time t by proposing new particles from a (quite arbitrary) proposal density r_t,

$$x^i_t \sim r_t(x_t \mid x^i_{1:t-1}, y_{1:t}), \tag{10}$$

for $i = 1, \ldots, N$. These samples are appended to the existing particle trajectories, i.e. $x^i_{1:t} := \{x^i_{1:t-1}, x^i_t\}$. The particles are then assigned importance weights according to,

$$w^i_t \propto w^i_{t-1} \frac{p(y_t \mid x^i_t) p(x^i_t \mid x^i_{t-1})}{r_t(x^i_t \mid x^i_{1:t-1}, y_{1:t})}, \tag{11a}$$

where the weights are normalised to sum to one. If the sampling procedure outlined above is iterated over time, we end up with the sequential importance sampling method [15]. However, it is well known that this approach will suffer from depletion of the particle weights, meaning that as we proceed through time, all weights except one will tend to zero [1]. To remedy this, a selection or resampling step is added to the filter. This has the effect of discarding particles with low weights and duplicating particles with high weights. This is a crucial step, needed to make the PF practically applicable.

As indicated by (9), the PF does in fact generate weighted particle trajectories targeting the joint smoothing distribution. However, as an effect of the consecutive resampling steps, the particle trajectories will suffer from degeneracy; see e.g. [1]. This means that the PF in general only can provide good approximations of the filtering distribution, or a fixed-lag smoothing distribution with a short enough lag. For instance, an approximation of the filtering distribution is obtained from (9) by simply discarding the history of the particle trajectories, resulting in a point-mass approximation according to,

$$p(dx_t \mid y_{1:t}) \approx \widehat{p}(dx_t \mid y_{1:t}) = \sum_{i=1}^{N} w_t^i \delta_{x_t^i}(dx_t). \tag{12}$$

B. Forward filter/backward simulator

As pointed out in the previous section, due to particle degeneracy, the PF is in general insufficient when it comes to approximating the joint smoothing distribution or a marginal smoothing distribution $p(dx_t \mid y_{1:T})$ for $t \ll T$. This problem can be circumvented by complementing the PF with a backward recursion. In [5], a forward filter/backward smoother algorithm is proposed, designed to target the marginal smoothing densities $p(x_t \mid y_{1:T})$ for $t = 1, \ldots, T$. Here, T is some fixed, final time point. It is possible to extend this approach to fixed-interval or joint smoothing, but the computational complexity of this would be prohibitive.

An alternative, very much related, approach is the forward filter/backward simulator (FFBSi) by [8], [9]. In this method, the joint smoothing distribution is targeted by sampling *backward trajectories* from an empirical smoothing distribution defined by the PF. Consider the following factorisation of the joint smoothing density,

$$p(x_{1:T} \mid y_{1:T}) = p(x_T \mid y_{1:T}) \prod_{t=1}^{T-1} \underbrace{p(x_t \mid x_{t+1:T}, y_{1:T})}_{=p(x_t \mid x_{t+1}, y_{1:t})}. \tag{13}$$

Here, the *backward kernel* density $p(x_t \mid x_{t+1}, y_{1:t})$ can be written as,

$$p(x_t \mid x_{t+1}, y_{1:t}) = \frac{p(x_{t+1} \mid x_t) p(x_t \mid y_{1:t})}{p(x_{t+1} \mid y_{1:t})}. \tag{14}$$

We note that the backward kernel depends on the filtering density $p(x_t \mid y_{1:t})$. The key enabler of the FFBSi (or any forward/backward based particle smoother) is that the filtering density (in many cases) can be readily approximated by a PF, without suffering from degeneracy. Hence, assume that we have filtered the data record $y_{1:T}$ using a PF. For each $t = 1, \ldots, T$ we have thus generated a weighted particle system $\{x_t^i, w_t^i\}_{i=1}^N$ targeting the filtering distribution at time t, according to (12). These particles can then be used to approximate the backward kernel (14) with,

$$\widehat{p}(dx_t \mid x_{t+1}, y_{1:t}) \triangleq \sum_{i=1}^{N} \frac{w_t^i p(x_{t+1} \mid x_t^i)}{\sum_k w_t^k p(x_{t+1} \mid x_t^k)} \delta_{x_t^i}(dx_t). \tag{15}$$

Based on this approximation, we may sample particle trajectories, backward in time, approximately distributed according to the joint smoothing density (13). The backward trajectories are initiated by sampling from the empirical filtering distribution at time T, defined by (12), i.e.

$$\tilde{x}_T^j \sim \widehat{p}(dx_T \mid y_{1:T}), \tag{16a}$$

for $j = 1, \ldots, M$. Note that the number of backward trajectories M is arbitrary, and need not equal the number of forward filter particles N. At time t, the backward trajectories are augmented by sampling from the empirical backward kernel (15),

$$\tilde{x}_t^j \sim \widehat{p}(dx_t \mid \tilde{x}_{t+1}^j, y_{1:t}), \tag{16b}$$

$$\tilde{x}_{t:T}^j := \{\tilde{x}_t^j, \tilde{x}_{t+1:T}^j\}, \tag{16c}$$

for $j = 1, \ldots, M$. When the backward recursion is complete, i.e. at time $t = 1$, the collection of backward trajectories $\{\tilde{x}_{1:T}^j\}_{j=1}^M$ are approximately distributed according to the joint smoothing distribution. Sampling from the empirical backward kernel (15) is straightforward, since it is discrete and has support in N points. Hence, for a fixed $\tilde{x}_{t+1}^j$, (15) reduces to,

$$\widehat{p}(dx_t \mid \tilde{x}_{t+1}^j, y_{1:t}) = \sum_{i=1}^{N} \tilde{w}_{t|T}^{i,j} \delta_{x_t^i}(dx_t), \tag{17a}$$

where we have defined the smoothing weights,

$$\tilde{w}_{t|T}^{i,j} \triangleq \frac{w_t^i p(\tilde{x}_{t+1}^j \mid x_t^i)}{\sum_k w_t^k p(\tilde{x}_{t+1}^j \mid x_t^k)}. \tag{17b}$$

The FFBSi is summarised in Algorithm 1.

Algorithm 1 Standard FFBSi [9]

Initialise the backward trajectories. Set $\tilde{x}_T^j = x_T^i$ with probability w_t^i for $j = 1, \ldots, M$.
for $t = T - 1$ **to** 1 **do**
  for $j = 1$ **to** M **do**
    Compute $\tilde{w}_{t|T}^{i,j} \propto w_t^i p(\tilde{x}_{t+1}^j \mid x_t^i)$, for $i = 1, \ldots, N$.
    Sample from the empirical backward kernel, i.e. set $\tilde{x}_t^j = x_t^i$ with probability $\tilde{w}_{t|T}^{i,j}$.
  end for
end for

The computational complexity of the standard FFBSi grows like MN, i.e. with $M = N$ it is quadratic in the number of particles/backward trajectories. However, [10] have recently proposed a reformulation of the FFBSi, which under certain assumptions can be shown to reach linear complexity in the number of particles. The key enabler of this approach is to perform the backward simulation by rejection sampling, which means that we do not need to compute all the MN smoothing weights (17b). This approach will be discussed further in Section V-B, where we show how it can be applied to the Rao-Blackwellised particle smoothers proposed in this paper.

C. Rao-Blackwellised particle filter

In the preceding sections we reviewed some "non-Rao-Blackwellised" particle methods for filtering and smoothing, designed for general SSMs according to (1). Let us now return to the filtering problem and instead consider the special class of CLGSS models. The task is to design a PF which exploits the tractable substructure in the model; the resulting filter is the RBPF [5]–[7]. Informally, the incentive for this is to obtain more accurate estimates than what is given by a standard PF. For a formal discussion on the difference in asymptotic variance between the PF and the RBPF, see [16].

The RBPF targets the density $p(\xi_{1:t}, z_t \mid y_{1:t})$, by utilising the factorisation

$$p(\xi_{1:t}, z_t \mid y_{1:t}) = p(z_t \mid \xi_{1:t}, y_{1:t})p(\xi_{1:t} \mid y_{1:t}). \tag{18}$$

The key observation is that, for a CLGSS model, the first factor in this expression is Gaussian and analytically tractable using a Kalman filter (KF), i.e.

$$p(z_t \mid \xi_{1:t}, y_{1:t}) = \mathcal{N}(z_t; \bar{z}_{t|t}(\xi_{1:t}), P_{t|t}(\xi_{1:t})), \tag{19}$$

for some (tractable) sequence of mean and covariance functions, $\bar{z}_{t|t}$ and $P_{t|t}$, of the nonlinear state trajectory $\xi_{1:t}$. Clearly, $\bar{z}_{t|t}$ and $P_{t|t}$ also depend on the measurement sequence, but we refrain from making that dependence explicit.

The second factor in (18), referred to as the state-marginal smoothing density[1], is targeted by a weighted particle system $\{\xi^i_{1:t}, \omega^i_t\}_{i=1}^N$ generated by an SMC method. Analogously to (9), the state-marginal smoothing distribution is approximated by an empirical distribution defined by the particles,

$$p(d\xi_{1:t} \mid y_{1:t}) \approx \widehat{p}(d\xi_{1:t} \mid y_{1:t}) \triangleq \sum_{i=1}^N \omega^i_t \delta_{\xi^i_{1:t}}(d\xi_{1:t}). \tag{20}$$

For each nonlinear particle trajectory, we can evaluate the mean and covariance functions for the conditional filtering density (19). Hence, from an implementation point of view, we typically compute and store quadruples of the form $\{\xi^i_{1:t}, \omega^i_t, \bar{z}^i_{t|t}, P^i_{t|t}\}_{i=1}^N$ for $t = 1, \ldots, T$ where $\bar{z}^i_{t|t} \triangleq \bar{z}_{t|t}(\xi^i_{1:t})$ and $P^i_{t|t} \triangleq P_{t|t}(\xi^i_{1:t})$. However, it is important to remember that for a CLGSS model, the conditional filtering density (19) is Gaussian, with mean and covariance as functions of the nonlinear state trajectory. Hence, if we are given some nonlinear state trajectory $\xi'_{1:t}$ (not necessarily generated by a PF), we may employ a KF to find the sufficient statistics of the density (19) conditioned on $\xi'_{1:t}$. This property is utilised in the RBPS presented in Section V-C.

Furthermore, by combining (18), (19) and (20) we obtain an approximation of the filtering distribution,

$$p(d\xi_t, dz_t \mid y_{1:t}) \approx \sum_{i=1}^N \omega^i_t \mathcal{N}(dz_t; \bar{z}_{t|t}(\xi^i_{1:t}), P_{t|t}(\xi^i_{1:t}))\delta_{\xi^i_t}(d\xi_t). \tag{21}$$

This also provides an approximation of the conditional of the filtering density,

$$p(z_t \mid \xi^i_t, y_{1:t}) \approx \mathcal{N}(z_t; \bar{z}_{t|t}(\xi^i_{1:t}), P_{t|t}(\xi^i_{1:t})), \tag{22}$$

for ξ^i_t belonging to the set of RBPF particles as time t. It is worth to note that both (21) and (22) are approximations, as opposed to (19) which is exact.

V. 'Joint backward simulation'-RBPS

We now turn to the problem of Rao-Blackwellised particle smoothing and derive the first of the two RBPS that we will present in this paper. This smoother is referred to as joint backward simulation RBPS (JBS-RBPS).

[1]The state-marginal smoothing density is a marginal of the joint smoothing density. The prefix "state" is used to distinguish it from what we normally mean by the marginal smoothing density, i.e. $p(x_t \mid y_{1:T})$.

A. *JBS-RBPS derivation*

The JBS-RBPS is similar to the FFBSi discussed in Section IV-B, in the sense that we wish to sample from the joint smoothing distribution by exploiting the factorisation (13). The difference is that the JBS-RBPS makes use of an RBPF to approximate the backward kernel, whereas the FFBSi uses a "standard" PF. The smoother is initialised by sampling from the empirical filtering distribution at time T, generated by the RBPF. Hence, we sample nonlinear forward trajectories $\{\xi'^j_{1:T}\}_{j=1}^M$ from (20) and thereafter we sample "linear states" from the Gaussian distribution (19), i.e.

$$\xi'^j_{1:T} \sim \widehat{p}(d\xi_{1:T} \mid y_{1:T}), \tag{23a}$$

$$\tilde{z}^j_T \sim \mathcal{N}(\bar{z}_{T|T}(\xi'^j_{1:T}), P_{T|T}(\xi'^j_{1:T})), \tag{23b}$$

for $j = 1, \ldots, M$. The pair $\{\xi'^j_{1:T}, \tilde{z}^j_T\}$ is an approximate realisation from $p(\xi_{1:T}, z_T \mid y_{1:T})$. The word *approximate* here refers to the fact that we approximate the target distribution with an RBPF, before sampling from it. The same type of approximation is used also in the standard FFBSi; see (16).

To obtain approximate realisations from the filtering density at time T, $p(\xi_T, z_T \mid y_{1:T})$, we simply discard $\xi'^j_{1:T-1}$ and set

$$\tilde{x}^j_T := \{\tilde{\xi}^j_T, \tilde{z}^j_T\}, \tag{24a}$$

$$\tilde{\xi}^j_T := \xi'^j_T, \tag{24b}$$

for $j = 1, \ldots, M$.

Now, assume that we have sampled joint backward trajectories $\{\tilde{x}^j_{t+1:T}\}_{j=1}^M$ from time T down to time $t+1$. We then wish to augment these trajectories with samples from the backward kernel, approximated by the forward RBPF. Using the partitioning of the state variable into nonlinear and linear states, the backward kernel density (14) can be expressed as,

$$p(\xi_t, z_t \mid \xi_{t+1}, z_{t+1}, y_{1:t}) = \int p(z_t \mid \xi_{1:t+1}, z_{t+1}, y_{1:t})p(\xi_{1:t} \mid \xi_{t+1}, z_{t+1}, y_{1:t})\, d\xi_{1:t-1}. \tag{25}$$

Sampling from this density is done similarly to (23) and (24). The outline of the procedure is as follows. We start by drawing a nonlinear trajectory $\xi'^j_{1:t}$ from the second factor of the integrand above. Given this sample, we draw a linear sample $\tilde{z}^j_t$ from the first factor of the integrand. We then discard $\xi'^j_{1:t-1}$ and set $\tilde{x}^j_t := \{\tilde{\xi}^j_t, \tilde{z}^j_t\}$ with $\tilde{\xi}^j_t := \xi'^j_t$. This is an example of a basic sampling technique, known as ancestral sampling, cf. with how sampling from e.g. a Gaussian mixture is done.

Hence, we start by considering the second factor of the integrand in (25). From Bayes' rule we have,

$$p(\xi_{1:t} \mid \xi_{t+1}, z_{t+1}, y_{1:t}) \propto p(\xi_{t+1}, z_{t+1} \mid \xi_{1:t}, y_{1:t})p(\xi_{1:t} \mid y_{1:t}). \tag{26}$$

We thus arrive at a distribution that can be readily approximated by the forward filter (i.e. the RBPF) particles. From (20) and (26) we get

$$p(d\xi_{1:t} \mid \tilde{\xi}^j_{t+1}, \tilde{z}^j_{t+1}, y_{1:t}) \approx \sum_{i=1}^N \tilde{\omega}^{i,j}_{t|T} \delta_{\xi^i_{1:t}}(d\xi_{1:t}), \tag{27a}$$

with

$$\tilde{\omega}_{t|T}^{i,j} \triangleq \frac{\omega_t^i p(\tilde{\xi}_{t+1}^j, \tilde{z}_{t+1}^j \mid \xi_{1:t}^i, y_{1:t})}{\sum_k \omega_t^k p(\tilde{\xi}_{t+1}^j, \tilde{z}_{t+1}^j \mid \xi_{1:t}^k, y_{1:t})}. \tag{27b}$$

The density involved in the above weight expression is available from the RBPF for any CLGSS model. For the mixed linear/nonlinear Gaussian state-space models studied here, using the compact notation (5), it is given by,

$$\begin{aligned} &p(\xi_{t+1}, z_{t+1} \mid \xi_{1:t}^i, y_{1:t}) \\ &\quad = \mathcal{N}\left(x_{t+1}; f^i + A^i \bar{z}_{t|t}^i, Q^i + A^i P_{t|t}^i (A^i)^\mathsf{T}\right). \end{aligned} \tag{28}$$

Here we have employed the shorthand notation, $f^i = f(\xi_t^i)$ etc. For each backward trajectory, i.e. for $j = 1, \ldots, M$, we can now sample an index

$$I(j) \sim \text{Cat}\left(\{\tilde{\omega}_{t|T}^{i,j}\}_{i=1}^N\right), \tag{29}$$

corresponding to the forward filter particle that is to be appended to the j:th backward trajectory, i.e. we set $\xi_{1:t}^{\prime j} = \xi_{1:t}^{I(j)}$ and $\tilde{\xi}_t^j := \xi_t^{\prime j}$. Here, $\text{Cat}(\{p_i\}_{i=1}^N)$ denote the categorical (i.e. discrete) distribution over the finite set $\{1, \ldots, N\}$ with probabilities $\{p_i\}_{i=1}^N$.

With that, we have completed the backward simulation for the nonlinear state at time t, and it remains to sample the linear state. However, before we proceed with this, we summarise the sampling of the nonlinear state in Algorithm 2. This algorithm will be used as one component of the "full" JBS-RBPS method presented in Algorithm 3. The reason for this decomposition of the algorithm will become clear in the sequel.

Algorithm 2 Nonlinear trajectory backward simulation

1: **for** $j = 1$ **to** M **do**
2: Compute the smoothing weights $\{\tilde{\omega}_{t|T}^{i,j}\}_{i=1}^N$ according to (27b) and (28).
3: Sample $I(j) \sim \text{Cat}\left(\{\tilde{\omega}_{t|T}^{i,j}\}_{i=1}^N\right)$.
4: **end for**
5: **return** the indices $\{I(j)\}_{j=1}^M$.

We now turn our attention to the first factor of the integrand in (25). By the conditional independence properties of the model and Bayes' rule we have,

$$\begin{aligned} &p(z_t \mid \xi_{1:t+1}, z_{t+1}, y_{1:t}) \\ &\qquad \propto p(\xi_{t+1}, z_{t+1} \mid z_t, \xi_{1:t}, y_{1:t}) p(z_t \mid \xi_{1:t}, y_{1:t}) \\ &\qquad = p(\xi_{t+1}, z_{t+1} \mid \xi_t, z_t) p(z_t \mid \xi_{1:t}, y_{1:t}). \end{aligned} \tag{30}$$

We recognise the first factor of (30) as the transition density, which according to (5) is Gaussian and affine in z_t. The second factor of (30) is the conditional filtering density for the linear state, given the nonlinear state trajectory. For a CLGSS model, this density is also Gaussian according to (19). Hence, we arrive at an affine transformation of a Gaussian variable, which itself is Gaussian. For $\xi_{1:t}^i$ belonging to the set of RBPF particles, we get,

$$p(z_t \mid \xi_{1:t}^i, \xi_{t+1}, z_{t+1}, y_{1:t}) = \mathcal{N}\Big(z_t; \mu_{t|t}^i(\xi_{t+1}, z_{t+1}), \Pi_{t|t}^i\Big), \tag{31}$$

where we have defined

$$\mu_{t|t}^i \triangleq \bar{z}_{t|t}^i + H_t^i \left(\begin{bmatrix}\xi_{t+1}^\mathsf{T} & z_{t+1}^\mathsf{T}\end{bmatrix}^\mathsf{T} - f^i - A^i \bar{z}_{t|t}^i\right), \tag{32a}$$

$$\Pi_{t|t}^i \triangleq P_{t|t}^i - H_t^i A^i P_{t|t}^i, \tag{32b}$$

with

$$H_t^i \triangleq P_{t|t}^i (A^i)^\mathsf{T} \left(Q^i + A^i P_{t|t}^i (A^i)^\mathsf{T}\right)^{-1}. \tag{32c}$$

Remark 2. It is straightforward to rewrite (32a) according to,

$$\mu_{t|t}^i = \Pi_{t|t}^i W_z^i z_{t+1} + c_{t|t}^i(\xi_{t+1}) \tag{33a}$$

with

$$c_{t|t}^i \triangleq \Pi_{t|t}^i \left(W_\xi^i(\xi_{t+1} - f_\xi^i) - W_z^i f_z^i + (P_{t|t}^i)^{-1} \bar{z}_{t|t}^i\right), \tag{33b}$$

$$\begin{bmatrix}W_\xi(\xi_t) & W_z(\xi_t)\end{bmatrix} \triangleq A(\xi_t)^\mathsf{T} Q(\xi_t)^{-1}. \tag{33c}$$

This highlights the fact that (32a) is affine in z_{t+1}; a property that will be used in Section VI. □

Let $\{\tilde{x}_{t+1:T}^j\}_{j=1}^M = \{\tilde{\xi}_{t+1:T}^j, \tilde{z}_{t+1:T}^j\}_{j=1}^M$ be the joint backward trajectories available at time $t+1$. As in (29), let $I(j)$ be the index of the RBPF particle which is appended to the j:th backward trajectory. Then, for each $j = 1, \ldots, M$ we can evaluate the mean and covariance of the Gaussian distribution (31), given by $\mu_{t|t}^{I(j)}(\tilde{\xi}_{t+1}^j, \tilde{z}_{t+1}^j)$ and $\Pi_{t|t}^{I(j)}$, respectively. It is then straightforward to sample $\tilde{z}_t^j$ from (31), completing the joint backward simulation at time t. The backward trajectories are thus given by,

$$\tilde{x}_{t:T}^j := \{\tilde{x}_t^j, \tilde{x}_{t+1:T}^j\}, \tag{34a}$$

$$\tilde{x}_t^j := \{\xi_t^{I(j)}, \tilde{z}_t^j\}, \tag{34b}$$

for $j = 1, \ldots, M$.

At time $t = 1$, we have obtained a collection of joint backward trajectories, approximating the joint smoothing distribution according to,

$$p(dx_{1:T} \mid y_{1:T}) \approx \frac{1}{M} \sum_{j=1}^M \delta_{\tilde{x}_{1:T}^j}(dx_{1:T}). \tag{35}$$

We summarise the JBS-RBPS in Algorithm 3.

Algorithm 3 JBS-RBPS

1: Initialise the backward trajectories according to (23) and (24); $\{\tilde{x}_T^j\}_{j=1}^M = \{\tilde{\xi}_T^j, \tilde{z}_T^j\}_{j=1}^M$.
2: **for** $t = T-1$ **to** 1 **do**
3: Sample indices $\{I(j)\}_{j=1}^M$ according to Algorithm 2 or (preferably) according to Algorithm 4.
4: **for** $j = 1$ **to** M **do**
5: Set $\tilde{\xi}_t^j = \xi_t^{I(j)}$.
6: Sample $\tilde{z}_t^j \sim \mathcal{N}\left(\mu_{t|t}^{I(j)}(\tilde{\xi}_{t+1}^j, \tilde{z}_{t+1}^j), \Pi_{t|t}^{I(j)}\right)$ using (32).
7: Set $\tilde{x}_t^j = \{\tilde{\xi}_t^j, \tilde{z}_t^j\}$ and $\tilde{x}_{t:T}^j = \{\tilde{x}_t^j, \tilde{x}_{t+1:T}^j\}$.
8: **end for**
9: **end for**
10: **return** the backward trajectories $\{\tilde{x}_{1:T}^j\}_{j=1}^M$.

B. Fast sampling of the nonlinear backward trajectories

Algorithm 2 provides a straightforward way to sample the indices used to augment the nonlinear backward trajectories. However, this method has a computational complexity which grows like MN. This is easily seen from the fact that index j ranges from 1 to M and index i ranges from 1 to N. Hence, if we would use $M = N$, the JBS-RBPS using Algorithm 2 is quadratic in the number of particles. This is a major drawback, common to many particle smoothers presented in the literature. However, recently a new particle smoother has been presented, which allows us to sample backward trajectories with a cost that grows only linearly with the number of particles [10]. Below we explain how the same idea can be used also for the mixed linear/nonlinear models studied in this work.

The quadratic complexity of Algorithm 2 arises from the evaluation of the weights (27b). However, by taking a rejection sampling approach, it is possible to sample from (29) without evaluating all the weights. The target distribution is, from (29) and (27), categorical, with probabilities given by $\{\tilde{\omega}_{t|T}^{i,j}\}_{i=1}^{N}$. As proposal distribution, we take the categorical distribution over the same index set $\{1, \ldots, N\}$, but with probabilities $\{\omega_t^i\}_{i=1}^N$, i.e. given by the filter weights. Let

$$\rho_t \triangleq (2\pi)^{-\frac{n_x}{2}} \max_{i=1,\ldots,N} \left[\det\left(Q^i + A^i P_{t|t}^i (A^i)^{\mathsf{T}}\right)^{-\frac{1}{2}} \right], \tag{36}$$

which implies that $\rho_t \geq p(\tilde{\xi}_{t+1}^j, \tilde{z}_{t+1}^j \mid \xi_{1:t}^i, y_{1:t})$ for all i and j. We can thus apply rejection sampling to sample the indices $\{I(j)\}_{j=1}^M$, as described in Algorithm 4. In terms of input and output, this algorithm is equivalent to Algorithm 2.

Algorithm 4 Fast nonlinear trajectory backward simulation

```
1:  L ← {1, …, M}.
2:  while L is not empty do
3:     n ← card(L).
4:     δ ← ∅.
5:     Sample independently {C(k)}_{k=1}^n ~ Cat({ω_t^i}_{i=1}^N).
6:     Sample independently {U(k)}_{k=1}^n ~ U(0, 1).
7:     for k = 1 to n do
8:        if U(k) ≤ p( z̃_{t+1}^{L(k)}, ξ̃_{t+1}^{L(k)} | ξ_{1:t}^{C(k)}, y_{1:t} ) / ρ_t then
9:           I(L(k)) ← C(k).
10:          δ ← δ ∪ {L(k)}.
11:       end if
12:    end for
13:    L ← L \ δ.
14: end while
15: return the indices {I(j)}_{j=1}^M.
```

For $M = N$ and under some additional assumptions, it can be shown that the rejection sampling approach used by Algorithm 4 reaches linear complexity [10]. However, it is worth to note that there is no upper bound on the number of times that the while-loop may be executed. Empirical studies indicate that most of the time required by Algorithm 4, is spent on just a few particles. In other words, the cardinality of L decreases fast in the beginning (we get a lot of accepted samples), but can linger for a long time close to zero. This can for instance occur when just a single backward trajectory remains, for which all RBPF particles gets low acceptance probabilities. To circumvent this, a "timeout check" can be added to Algorithm 4. Hence, let M_{max} be the maximum allowed number of executions of the while-loop at row 2. If L is not empty after M_{max} iterations, we make an exhaustive evaluation of the smoothing weights for the remaining elements in L, i.e. as in Algorithm 2, but with j ranging only over the remaining indices in L. By empirical studies, such a timeout check can drastically reduce the execution time of Algorithm 4, and seems to be crucial for its applicability for certain problems. A sensible value for M_{max} seems to be in the range $M/3$ to $M/2$.

As we will argue in Section VIII, it is generally a good idea to use $N \gg M$. Still, by empirical studies, we have found that Algorithm 4 provides a substantial speed-up over Algorithm 2 for many problems.

C. Constrained smoothing of the linear states

After a complete pass of the JBS-RBPS algorithm, we have obtained a collection of backward trajectories $\{\tilde{x}_{1:T}^j\}_{j=1}^M = \{\tilde{\xi}_{1:T}^j, \tilde{z}_{1:T}^j\}_{j=1}^M$, approximating the joint smoothing distribution with a point-mass distribution according to (35). However, since the model under study is CLGSS, it holds that for fixed *nonlinear* state trajectories, the smoothing problem is analytically tractable, since the model then reduces to an LGSS. Hence, if we keep the nonlinear backward trajectories, but discard the linear ones, we may perform a constrained forward/backward smoothing for the linear states.

Hence, for each $j = 1, \ldots, M$ we fix $\tilde{\xi}_{1:T}^j$ and run a KF and an RTS smoother [17], [18] on the model (3). The conditional marginal smoothing density[2] for the linear state is then obtained as,

$$p(z_t \mid \tilde{\xi}_{1:T}^j, y_{1:T}) = \mathcal{N}\left(z_t; \tilde{z}_{t|T}^j, \widetilde{P}_{t|T}^j\right), \tag{37}$$

for some means and covariances, $\{\tilde{z}_{t|T}^j\}_{t=1}^T$ and $\{\widetilde{P}_{t|T}^j\}_{t=1}^T$, respectively. In contrast to the "joint point-mass representation" (35) produced by the JBS-RBPS, we thus obtain a mixed representation of the marginal smoothing distribution (similarly to the RBPF representation of the filtering distribution),

$$p(d\xi_t, dz_t \mid y_{1:T}) \approx \frac{1}{M} \sum_{j=1}^{M} \mathcal{N}\left(dz_t; \tilde{z}_{t|T}^j, \widetilde{P}_{t|T}^j\right) \delta_{\tilde{\xi}_t^j}(d\xi_t). \tag{38}$$

Hence, we use the JBS-RBPS given in Algorithm 3 to sample the nonlinear backward trajectories. Note, however, that we still need to sample the linear backward trajectories, since the linear samples are used in the computation of the weights in (27). Hence, the linear backward trajectories can be seen as auxiliary variables in this method, needed to generate the nonlinear backward trajectories. Once this is done, the linear samples are replaced by an analytical evaluation of the conditional smoothing densities (37).

[2] Recall that we, for notational convenience, focus on the marginal smoothing distribution.

One obvious drawback with this method is that we need to process the data twice, i.e. we make two consecutive forward filtering/backward smoothing passes.

VI. 'MARGINAL BACKWARD SIMULATION'-RBPS

One obvious question to ask is whether it is possible to sample the nonlinear backward trajectories and at the same time, sequentially backward in time, update the sufficient statistics for the conditional smoothing densities for the linear states. In other words; can we run a single forward filtering/backward smoothing pass and obtain an approximation of the marginal smoothing distribution of the same form as (38)? As we shall see, this is not that easily achievable and we will require some approximations to enable such a backward recursion.

Before we engage in the derivation of the marginal backward simulation RBPF (MBS-RBPS), let us pause to think about the cause for this problem. The basis for both the RBPF and any RBPS is the CLGSS property of the model under study, which more or less boils down to the conditional filtering density given by (19). This states that, as long as we traverse along (and condition on) a nonlinear state trajectory $\xi'_{1:t}$, the conditional distribution is Gaussian. However, the purpose of smoothing through backward simulation is clearly to "update" the trajectories generated by the forward RBPF; if we do not allow for any change of the trajectories, we will not gain anything from smoothing. The problem is that when we no longer have fixed nonlinear state trajectories, the Gaussianity implied by (19) is lost.

We can also understand this by thinking of the nonlinear state trajectories as "extra measurements" in the RBPF. Since we, during the backward smoothing, sample new nonlinear state trajectories, we will in fact change these "extra measurements". Clearly, for a forward/backward smoother for an LGSS model to be applicable, we may not change the measurement sequence between the forward and the backward passes.

To get around this problem we will, as mentioned above, need some approximation. Naturally, we wish to compute the conditional, marginal smoothing densities $p(z_t \mid \xi_{1:T}, y_{1:T})$. Furthermore, we wish to compute these densities sequentially, backward in time. Hence, we require that this conditional smoothing density should be available for computation at time t, which highlights the problem that we face; at time t we have generated nonlinear backward trajectories $\{\tilde{\xi}^j_{t:T}\}_{j=1}^M$, but we do not know how these trajectories will extended to time $t-1$. This insight suggests the following approximation.

Approximation 1. *For each $t = 2, \ldots, T$, conditioned on $\xi_{t:T}$ and $y_{1:T}$, the linear state z_t is approximately independent of $\xi_{1:t-1}$, i.e. $p(z_t \mid \xi_{t:T}, y_{1:T}) \approx p(z_t \mid \xi_{1:T}, y_{1:T})$.*

We continue the discussion on this approximation in Section VII, but first we turn to the derivation of the MBS-RBPS.

A. Initialisation

We start the derivation of the MBS-RBPS by considering the initialisation at time T. This will also provide some insight into the nature of Approximation 1. The nonlinear backward trajectories are initialised by sampling from the empirical state-marginal smoothing distribution (20), defined by the RBPF,

$$\{I(j)\}_{j=1}^M \sim \mathrm{Cat}\left(\{\omega_T^i\}_{i=1}^N\right), \tag{39a}$$

$$\tilde{\xi}_T^j := \xi_T^{I(j)}, \qquad j = 1, \ldots, M. \tag{39b}$$

Furthermore, by Approximation 1 we conjecture that

$$p(z_T \mid \tilde{\xi}_T^j, y_{1:T}) \approx p(z_T \mid \underbrace{\xi_{1:T-1}^{I(j)}, \tilde{\xi}_T^j}_{=\xi_{1:T}^{I(j)}}, y_{1:T}). \tag{40}$$

The density on the right hand side is the conditional filtering density at time T, which is provided by the RBPF according to (19). Hence, we can approximate the density on the left hand side with,

$$\widehat{p}(z_T \mid \tilde{\xi}_T^j, y_{1:T}) \triangleq \mathcal{N}\left(z_T; \tilde{z}_{T|T}^j, \widetilde{P}_{T|T}^j\right), \tag{41a}$$

where,

$$\tilde{z}_{T|T}^j := \bar{z}_{T|T}\left(\xi_{1:T}^{I(j)}\right), \qquad j = 1, \ldots, M, \tag{41b}$$

$$\widetilde{P}_{T|T}^j := P_{T|T}\left(\xi_{1:T}^{I(j)}\right), \qquad j = 1, \ldots, M. \tag{41c}$$

This is in fact exactly the same approximation of the density $p(z_T \mid \xi_T, y_{1:T})$ as given by the RBPF in (22).

B. Marginal backward simulation

To enable a marginal backward simulation for the nonlinear state, we consider a factorisation of the state-marginal smoothing density, similar to (13),

$$p(\xi_{1:T} \mid y_{1:T}) = p(\xi_T \mid y_{1:T}) \prod_{t=1}^{T-1} p(\xi_t \mid \xi_{t+1:T}, y_{1:T}). \tag{42}$$

Now, assume that the backward smoothing has been completed for time T down to time $t+1$. Hence, we assume that we have obtained a collection of nonlinear backward trajectories $\{\tilde{\xi}^j_{t+1:T}\}_{j=1}^M$; at time T these are given by (39). Furthermore, analogously to (41), we assume that we have approximated the conditional smoothing densities for the linear state with,

$$\widehat{p}(z_{t+1} \mid \tilde{\xi}^j_{t+1:T}, y_{1:T}) = \mathcal{N}\left(z_{t+1}; \tilde{z}^j_{t+1|T}, \widetilde{P}^j_{t+1|T}\right), \tag{43}$$

for some mean $\tilde{z}^j_{t+1|T}$ and covariance $\widetilde{P}^j_{t+1|T}$ and for $j = 1, \ldots, M$. How to compute these densities, sequentially backward in time, will be the topic of Section VI-C.

Based on the factorisation (42), we see that we wish to augment the backward trajectories with samples from

$$p(\xi_t \mid \xi_{t+1:T}, y_{1:T}). \tag{44}$$

To enable this, we write the target density as a marginal, similarly to (25),

$$p(\xi_t \mid \xi_{t+1:T}, y_{1:T}) = \int p(\xi_{1:t}, z_{t+1} \mid \xi_{t+1:T}, y_{1:T})\, d\xi_{1:t-1} dz_{t+1}, \tag{45}$$

which implies that we instead may sample from the joint density,

$$p(\xi_{1:t}, z_{t+1} \mid \tilde{\xi}^j_{t+1:T}, y_{1:T}) = p(\xi_{1:t} \mid z_{t+1}, \tilde{\xi}^j_{t+1:T}, y_{1:T}) p(z_{t+1} \mid \tilde{\xi}^j_{t+1:T}, y_{1:T}). \tag{46}$$

Using (43), the second factor in (46) is approximately Gaussian, and we can easily sample,

$$\widetilde{Z}_{t+1}^{j} \sim \mathcal{N}\left(z_{t+1}; \tilde{z}_{t+1|T}^{j}, \widetilde{P}_{t+1|T}^{j}\right). \tag{47}$$

For the first factor of (46), using the conditional independence properties of the model, we get

$$p(\xi_{1:t} \mid z_{t+1}, \xi_{t+1:T}, y_{1:T}) = p(\xi_{1:t} \mid z_{t+1}, \xi_{t+1}, y_{1:t}), \tag{48}$$

which coincides with (26). Hence, marginal backward simulation in the MBS-RBPS is done analogously to the nonlinear backward simulation in the JBS-RBPS, given by (27) – (29); only with $\tilde{z}_{t+1}^{j}$ replaced with the auxiliary variable $\widetilde{Z}_{t+1}^{j}$ generated by (47).

That is, for each backward trajectory $j = 1, \ldots, M$, we sample an index

$$I(j) \sim \mathrm{Cat}\left(\{\tilde{\omega}_{t|T}^{i,j}\}_{i=1}^{N}\right), \tag{49}$$

corresponding to the forward filter particle that is to be appended to the j:th backward trajectory. The smoothing weights $\{\tilde{\omega}_{t|T}^{i,j}\}_{i=1}^{N}$ are computed as in (27b) and (28). As before, since (27) defines a distribution over RBPF particle *trajectories*, we will in fact obtain a sample $\xi_{1:t}^{I(j)}$ from the space of trajectories. However, by simply discarding $\xi_{1:t-1}^{I(j)}$ and also the auxiliary variable $\widetilde{Z}_{t+1}^{j}$, we end up with an approximate realisation from (44). This sample can then be appended to the j:th backward trajectory,

$$\tilde{\xi}_{t:T}^{j} := \{\xi_{t}^{I(j)}, \tilde{\xi}_{t+1:T}^{j}\}. \tag{50}$$

Finally, we note that the fast backward simulation described in Section V-B can be used also for the MBS-RBPS.

C. Updating the linear states

When traversing backward in time, we also need to update the sufficient statistics for the linear states. As previously pointed out, the aim in the MBS-RBPS is to do this sequentially. The requirement for this is also indicated by (47), from which we note that the conditional smoothing densities for the linear states are needed during the backward simulation of the nonlinear state trajectories. In accordance with (43), we seek a Gaussian approximation of the conditional smoothing density,

$$\widehat{p}(z_t \mid \tilde{\xi}_{t:T}^{j}, y_{1:T}) = \mathcal{N}\left(z_t; \tilde{z}_{t|T}^{j}, \widetilde{P}_{t|T}^{j}\right); \tag{51}$$

at time $t = T$, the approximation is given by (41).

The mean and covariance of this distribution will be determined by fusing the information available in this RBPF at time t, with the (existing) smoothing distribution for the linear states at time $t+1$. We start by noting that, by the conditional independence properties of the model, we have

$$p(z_t \mid \xi_{1:T}, z_{t+1}, y_{1:T}) = p(z_t \mid \xi_{1:t+1}, z_{t+1}, y_{1:t}). \tag{52}$$

The density on the right hand side coincides with (31). Now, as before, let $I(j)$ be the index of the forward RBPF particle which is appended to the j:th backward trajectory at time t, so that $\tilde{\xi}_{t:T}^{j} = \{\xi_{t}^{I(j)}, \tilde{\xi}_{t+1:T}^{j}\}$. By (52), (31) and (33a) we then have,

$$\begin{aligned} &p(z_t \mid \xi_{1:t}^{I(j)}, \tilde{\xi}_{t+1:T}^{j}, z_{t+1}, y_{1:T}) \\ &\quad = \mathcal{N}\left(z_t; \Pi_{t|t}^{I(j)} W_z^{I(j)} z_{t+1} + c_{t|t}^{I(j)}(\tilde{\xi}_{t+1}^{j}), \Pi_{t|t}^{I(j)}\right), \end{aligned} \tag{53}$$

which is Gaussian and affine in z_{t+1}. Furthermore, by making use of Approximation 1 (for time $t+1$) and (43) we have.

$$p(z_{t+1} \mid \xi_{1:t}^{I(j)}, \tilde{\xi}_{t+1:T}^{j}, y_{1:T}) \approx \mathcal{N}\left(z_{t+1}; \tilde{z}_{t+1|T}^{j}, \widetilde{P}_{t+1|T}^{j}\right). \tag{54}$$

If we accept the approximation above, (53) and (54) describe an affine transformation of a Gaussian variable, which itself is Gaussian. Hence, with an additional application of Approximation 1 (for time t) we obtain,

$$\begin{aligned} &p(z_t \mid \xi_{1:t}^{I(j)}, \tilde{\xi}_{t+1:T}^{j}, y_{1:T}) \\ &\quad \approx \widehat{p}(z_t \mid \underbrace{\xi_{t}^{I(j)}, \tilde{\xi}_{t+1:T}^{j}}_{=\tilde{\xi}_{t:T}^{j}}, y_{1:T}) \triangleq \mathcal{N}\left(z_t; \tilde{z}_{t|T}^{j}, \widetilde{P}_{t|T}^{j}\right), \end{aligned} \tag{55}$$

with

$$\tilde{z}_{t|T}^{j} = \Pi_{t|t}^{I(j)} W_z^{I(j)} \tilde{z}_{t+1|T}^{j} + c_{t|t}^{I(j)}(\tilde{\xi}_{t+1}^{j}), \tag{56a}$$

$$\widetilde{P}_{t|T}^{j} = \Pi_{t|t}^{I(j)} + M_{t|T}^{j} (W_z^{I(j)})^{\mathsf{T}} \Pi_{t|t}^{I(j)}, \tag{56b}$$

$$M_{t|T}^{j} = \Pi_{t|t}^{I(j)} W_z^{I(j)} \widetilde{P}_{t+1|T}^{j}. \tag{56c}$$

The expression above provides the sought density (51).

Remark 3. Considering the relationship between (32a) and (33a) we may alternatively express (56a) as,

$$\tilde{z}_{t|T}^{j} = \bar{z}_{t|t}^{I(j)} + H_t^{I(j)}\left(\begin{bmatrix} \tilde{\xi}_{t+1}^{j} \\ \tilde{z}_{t+1|T}^{j} \end{bmatrix} - f^{I(j)} - A^{I(j)} \bar{z}_{t|t}^{I(j)}\right), \tag{57}$$

which may be more natural to use in an implementation. □

Remark 4. In many cases, the 2-step, fixed interval smoothing distribution $p(d\xi_{t:t+1}, dz_{t:t+1} \mid y_{1:T})$ is needed; see e.g. [14]. If this is the case, the variable defined in (56c) provides the conditional covariance between z_t and z_{t+1}, i.e.

$$M_{t|T}^{j} \approx \mathrm{Cov}\left(z_t z_{t+1}^{\mathsf{T}} \,\middle|\, \tilde{\xi}_{t:T}^{j}, y_{1:T}\right). \tag{58}$$

The approximate equality in the expression above is due to the fact that we made use of Approximation 1 when deriving (55) and (56c). □

We summarise the MBS-RBPS in Algorithm 5.

VII. Discussion

Before we continue with a numerical evaluation of the proposed smoothers, we provide some additional discussion regarding Approximation 1 used during the derivation of the MBS-RBPS. As pointed out at the beginning of Section VI, the need for this approximation arises since we wish to traverse backward in time, along nonlinear backward trajectories which are different from the RBPF forward trajectories.

In Figure 1 we illustrate the steps used to update the linear states when moving from time $t+1$ to time t in the MBS-RBPS. The boxes illustrate one nonlinear forward trajectory

Algorithm 5 MBS-RBPS

Initialise the marginal backward trajectories $\{\tilde{\xi}_T^j\}_{j=1}^M$ according to (39).
Initialise the means and covariances for the linear state $\{\tilde{z}_{T|T}^j, \widetilde{P}_{T|T}^j\}_{j=1}^M$ according to (41).
for $t = T - 1$ **to** 1 **do**
  Sample $\widetilde{Z}_{t+1}^j$ for $j = 1, \ldots, M$ according to (47).
  Sample indices $\{I(j)\}_{j=1}^M$ according to Algorithm 2 or (preferably) according to Algorithm 4 (with $\tilde{z}_{t+1}^j$ replaced by $\widetilde{Z}_{t+1}^j$).
  Augment the backward trajectories according to (50).
  Update the means and covariances according to (56).
end for
return the marginal backward trajectories $\{\tilde{\xi}_{1:T}^j\}_{j=1}^M$, and the means and the covariances for the linear state, $\{\tilde{z}_{t|T}^j, \widetilde{P}_{t|T}^j\}_{j=1}^M$ for $t = 1, \ldots, T$.

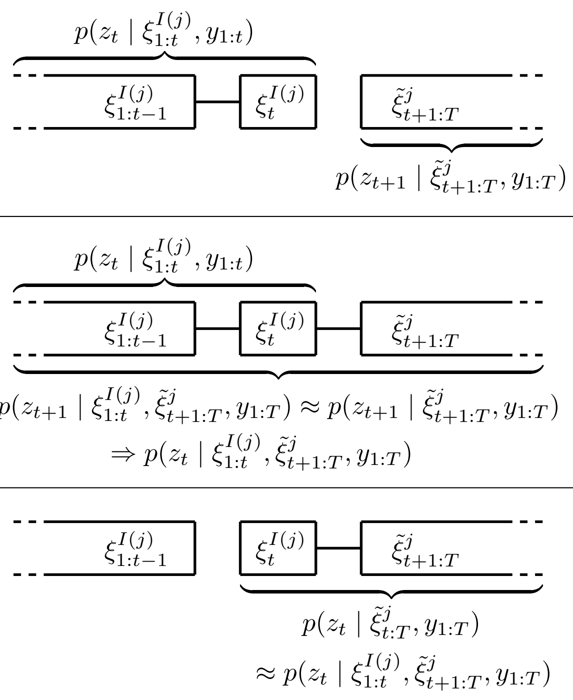

Fig. 1. Illustration of one time step in the MBS-RBPS. See the text for details.

$\xi_{1:t}^{I(j)}$ (generated by the RBPF) reaching up to time t, and one nonlinear backward trajectory $\tilde{\xi}_{t+1:T}^j$ reaching down to time $t+1$. In the upper plot of the figure, the backward trajectory is not yet connected to the forward trajectory. Two of the densities for the linear state are shown, provided by the RBPF and the MBS-RBPS, respectively.

In the middle plot, we extend the backward trajectory by sampling among the forward filter particles. Hence, the backward trajectory is connected to one of the RBPF forward trajectories. Here, we make use of Approximation 1, as in (54). In words, the meaning of this is that we assume that the conditional smoothing density for the linear state at time $t+1$ is unaffected by the concatenation of the forward trajectory. This enables us to fuse the conditional filtering density at time t, provided by the RBPF, with the conditional smoothing density at time $t + 1$. The result is given by (55) and (56).

Finally, in the bottom plot we discard the forward trajectory up to time $t - 1$, $\xi_{1:t-1}^{I(j)}$. The reason for this is that we, in general, wish to take a different path from time t to time $t-1$, than what is given by the current forward trajectory. To enable this, we again make use of Approximation 1 to "cut the link" with the forward trajectory. This is the approximation utilised in (55).

The basic meaning of Approximation 1 is that, conditioned on the present nonlinear state (and the future nonlinear states) as well as the measurements, the linear state is independent of "previous" nonlinear states. The accuracy of the approximation should thus be related to the mixing properties of the system. If the system under study is slowly mixing, we expect the approximation to be poor. Consequently, the MBS-RBPS should be used with care for this type of systems.

In the derivation of the JBS-RBPS in Section V, we did not encounter Approximation 1. However, it can be realised that a similar procedure to that outlined in Figure 1 is used in the JBS-RBPS as well. Recall that the backward kernel here is expressed as (25). As described in Section V, to sample from this kernel we first draw one of the RBPF forward trajectories $\xi_{1:t}^{I(j)}$. At this stage, we are in a similar state as illustrated by the middle plot in Figure 1. We then draw a linear sample from (31), which is appended to the joint backward trajectory. Finally, we discard the forward trajectory up to time $t - 1$, which corresponds to the procedure illustrated by the bottom plot in Figure 1.

Based on this similarity, it is believed that not only the MBS-RBPS, but also the JBS-RBPS, will encounter problems for slowly mixing systems. However, this is not that surprising, since they are both based on the RBPF, which is known to degenerate if the system is too slowly mixing.

VIII. Numerical illustrations

In this section we will evaluate the proposed smoothers on numerical data. Two different examples will be presented; first we consider a linear Gaussian system and thereafter a mixed linear/nonlinear system. The purpose of including a linear Gaussian example is to gain confidence in the presented smoothers. This is possible since, for this case, the smoothing densities can be computed analytically.

For both the linear and the mixed linear/nonlinear examples, we can also address the state inference problems using standard particle methods. To solve the filtering problem, we then use the bootstrap PF [4]. The smoothing problem is thereafter addressed using the fast FFBSi [10] (recall that this smoother is equivalent to the FFBSi by [9], given in Algorithm 1). For the Rao-Blackwellised particle smoothers, a bootstrap RBPF [7] is used as forward filter.

A. A linear system

Consider the second order linear system,

$$\begin{pmatrix} \xi_{t+1} \\ z_{t+1} \end{pmatrix} = \begin{pmatrix} 1 & 0.1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \xi_t \\ z_t \end{pmatrix} + v_t, \quad v_t \sim \mathcal{N}(0, Q), \tag{59a}$$

$$y_t = \xi_t + e_t, \qquad e_t \sim \mathcal{N}(0, R), \tag{59b}$$

with $Q = 0.1 I_{2\times 2}$ and $R = 0.1$. The initial state of the system is Gaussian with mean $\begin{pmatrix} 0 & 1 \end{pmatrix}^\mathsf{T}$ and covariance $0.1 I_{2\times 2}$. In the Rao-Blackwellised particle methods, the first state ξ_t is treated as if it is nonlinear, whereas the second state z_t is treated as linear.

The comparison was made by a Monte Carlo study over 100 realisations of data $y_{1:T}$ from the system (59), each consisting of $T = 100$ measurements. The three filters, KF, PF and RBPF, and thereafter the four smoothers, RTS, (fast) FFBSi, (fast) JBS-RBPS and (fast) MBS-RBPS, were run in parallel. Furthermore, we performed a constrained RTS smoothing of the linear state, based on the nonlinear backward trajectories generated by JBS-RBPS, as described in Section V-C. The PF and the RBPF both employed $N = 50$ particles and the particle smoothers all used $M = 50$ backward trajectories.

In Table I we present the time-averaged root mean squared errors (RMSEs) for the smoothers. As can be seen, all the Rao-Blackwellised smoothers performs similarly and close to the RTS. The FFBSi, which is a "pure" particle smoother, performs worse at estimating the z_t-state. Clearly, this is a simple state inference problem, but it still provides some confidence for the adequacy of the proposed smoothers.

TABLE I
RMSE VALUES FOR SMOOTHERS ($\times 10^{-1}$)

Smoother	ξ_t	z_t
FFBSi	2.33	11.26
JBS-RBPS	2.22	7.34
JBS-RBPS w. constr. RTS	2.22	7.25
MBS-RBPS	2.22	7.25
RTS	2.11	7.23

B. A mixed linear/nonlinear system

We continue with a more challenging mixed linear/nonlinear example. Consider the following first order nonlinear system,

$$\xi_{t+1} = 0.5\xi_t + \theta_t \frac{\xi_t}{1+\xi_t^2} + 8\cos(1.2t) + v_{\xi,t}, \tag{60a}$$

$$y_t = 0.05\xi_t^2 + e_t, \tag{60b}$$

for some process $\{\theta_t\}_{t\geq 1}$. The case with a static $\theta_t \equiv 25$, has been studied in e.g. [4], [19] and has become something of a benchmark example for nonlinear filtering. Here, we assume instead that θ_t is a time varying parameter with known dynamics, given by the output from a fourth order linear system,

$$z_{t+1} = \begin{pmatrix} 3 & -1.691 & 0.849 & -0.3201 \\ 2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0.5 & 0 \end{pmatrix} z_t + v_{z,t} \tag{61a}$$

$$\theta_t = 25 + \begin{pmatrix} 0 & 0.04 & 0.044 & 0.008 \end{pmatrix} z_t, \tag{61b}$$

with poles in $0.8 \pm 0.1i$ and $0.7 \pm 0.05i$. Combined, (60) and (61) is a mixed linear/nonlinear system. The noises are assumed to be white, Gaussian and mutually independent; $v_{\xi,t} \sim \mathcal{N}(0, 0.005)$, $v_{z,t} \sim \mathcal{N}(0, 0.01 I_{4\times 4})$ and $e_t \sim \mathcal{N}(0, 0.1)$.

We evaluate the proposed smoothers in a Monte Carlo study, where we simulate 1000 realisations of data $y_{1:T}$ from the system, each consisting of $T = 100$ measurements. As forward filters, we employ a PF and an RBPF, both using $N = 300$ particles. We then run the different smoothers; FFBSi, JBS-RBPS, JBS-RBPS with constrained RTS smoothing and finally MBS-RBPS. This is done for three different number of backward trajectories, $M = 10$, $M = 50$ and $M = 100$. Table II summarises the results, in terms of the time averaged RMSE values for the nonlinear state ξ_t and for the time varying parameter θ_t (note that θ_t is a linear combination of the four linear states z_t).

TABLE II
RMSE VALUES FOR SMOOTHERS ($\times 10^{-1}$)

	$M = 10$		$M = 50$		$M = 100$	
Smoother	ξ_t	θ_t	ξ_t	θ_t	ξ_t	θ_t
FFBSi	4.28	7.89	4.27	7.87	4.27	7.86
JBS-RBPS	3.17	5.89	3.13	5.74	3.12	5.72
JBS-RBPS w. constr. RTS	3.17	5.85	3.13	5.71	3.12	5.69
MBS-RBPS	3.16	5.81	3.13	5.73	3.13	5.72

As comparison, the RMSE values for the PF with $N = 300$ particles were 5.10×10^{-1} for ξ_t and 9.32×10^{-1} for θ_t, respectively. The corresponding numbers for the RBPF were 4.40×10^{-1} and 8.49×10^{-1}. From this, we note that smoothing clearly improves the performance over filtering, even with as few as 10 backward trajectories. This is the case, both for the "standard" particle methods, and for the Rao-Blackwellised ones. This provides some insight into how one should proceed when designing a forward/backward type of particle smoother. Most important is to obtain an accurate approximation of the backward kernel, and this depends only on the filter! Once this approximation is fixed, the backward simulators will generate conditionally i.i.d. samples from the empirical smoothing distribution (see further [10]). Hence, the computational effort should to a large extent be spent on the forward filter. If the filter provides an accurate approximation of the backward kernel, it will in many applications be sufficient to generate only a few backward trajectories.

The next thing to note is that the three RBPS all outperform the FFBSi, but compared to each other, they have very similar performance. For the nonlinear state, there is basically no difference at all. For the linear states (i.e. the time varying parameter), the JBS-RBPS with constrained RTS is, unsurprisingly, slightly better than just JBS-RBPS, regardless of the number of backward trajectories. This comes at the cost of an additional forward/backward sweep of the data. However, it should be noted that the time requirement for this constrained smoothing is much lower than for the JBS-RBPS and the MBS-RBPS. Hence, in terms of time consumption, it does not cost us very much to complement the JBS-RBPS with a constrained RTS, too achieve somewhat better estimates. Hence, the main drawback with this approach is perhaps not the increased computational complexity. Instead, it might be the fact that it requires the implementation of an additional smoothing procedure, which increases the volume and the complexity of the code.

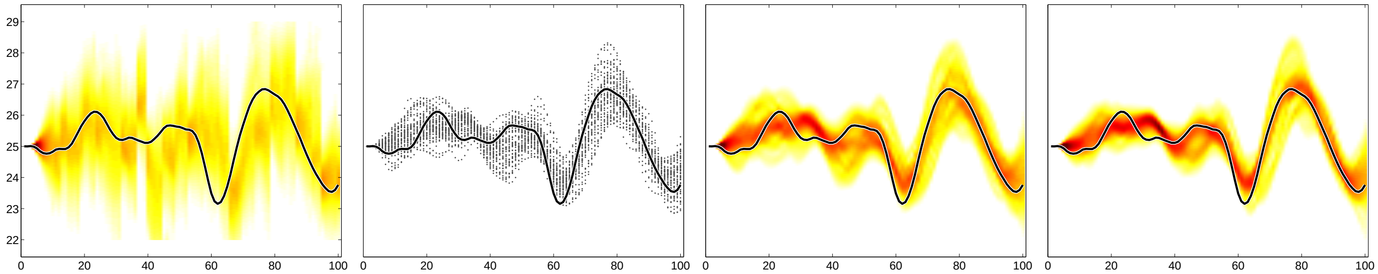

Fig. 2. Plots over distributions for θ_t over time, for one data realisation. The true value of θ_t is shown as a solid black line. From left to right; RBPF, JBS-RBPS, JBS-RBPS w. constrained RTS and MBS-RBPS.

As pointed out above, the JBS-RBPS and the MBS-RBPS have very similar performance, and it is not possible to say that one is better than the other. However, there is a slight indication that the MBS-RBPS has higher accuracy when we use very few backward trajectories ($M = 10$). For this specific example, the JBS-RBPS "catches up" as we increase the number of backward trajectories. Hence, if we do not have any particular time constraints, JBS-RBPS with a high number of backward trajectories and complemented with constrained RTS might be the better choice. On the contrary, if we wish to use only a few backward trajectories, MBS-RBPS might be preferable. However, we emphasise once again that the RBPS all have similar performance, and that they all solve the problem with increased accuracy when compared to standard FFBSi.

Finally, in Figure 2 we further illustrate the difference in the representation of the smoothing distribution, between the methods. The plots show (as thick black lines) the evolution of the parameter θ_t over time $t = 1, \dots, 100$, for one data realisation. The plots also show the estimated marginal smoothing distributions $p(\theta_t \mid y_{1:100})$ for the RBPF and the three RBPS (with $M = 50$). All methods, except JBS-RBPS, have continuous representations of the density function, which are color coded in the plots (the darker the color, the higher is the value of the density function). The JBS-RBPS uses an (unweighted) particle representation of the smoothing distribution, which is illustrated with dots in the figure.

IX. Conclusions

We have developed methods for Rao-Blackwellised particle smoothing, based on forward filter/backward simulator type of particle smoothers. We argued that, as opposed to the Rao-Blackwellised particle filter, there is no single "natural" way to construct an RBPS. Therefore, we have proposed two different approaches. The first, JBS-RBPS, uses a joint backward simulation in analogy with the "standard" FFBSi. The difference is that the JBS-RBPS approximates the backward kernel using an RBPF, whereas the FFBSi makes use of a PF. This method has previously been used in [13] for hierarchical models. Here we have extended the approach to fully interconnected mixed linear/nonlinear Gaussian state-space models. We also proposed to complement this approach with a constrained RTS smoothing for the linear states.

The second approach, MBS-RBPS, draws particles trajectories only for the nonlinear state process. This shows a stronger resemblance with the RBPF. However, due to the fact that we wish to update the nonlinear particle trajectories in the backward simulation, we were forced to make certain approximations.

In numerical studies, the different approaches gave similar results, all with improved performance over "standard" FFBSi. There is a slight indication that MBS-RBPS is preferable if we wish to use only a few backward trajectories, whereas JBS-RBPS with constrained RTS performs better when we increase the number of backward trajectories. However, which approach that is preferable over the other is likely to be problem dependent.

Finally, as a general message when designing a forward/backward type of particle smoother (Rao-Blackwellised or not) is to put effort in the forward filtering. For the smoothers to perform well, it is crucial that the backward kernel is accurately approximated, and this depends only on the filter.

References

[1] O. Cappé, S. J. Godsill, and E. Moulines, "An overview of existing methods and recent advances in sequential Monte Carlo," *Proceedings of the IEEE*, vol. 95, no. 5, pp. 899–924, 2007.

[2] A. Doucet and A. Johansen, "A tutorial on particle filtering and smoothing: Fifteen years later," in *The Oxford Handbook of Nonlinear Filtering*, D. Crisan and B. Rozovsky, Eds. Oxford University Press, 2011.

[3] F. Gustafsson, "Particle filter theory and practice with positioning applications," *IEEE Aerospace and Electronic Systems Magazine*, vol. 25, no. 7, pp. 53–82, 2010.

[4] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *Radar and Signal Processing, IEE Proceedings F*, vol. 140, no. 2, pp. 107 –113, Apr. 1993.

[5] A. Doucet, S. J. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statistics and Computing*, vol. 10, no. 3, pp. 197–208, 2000.

[6] A. Doucet, N. de Freitas, K. Murphy, and S. Russell, "Rao-Blackwellised particle filtering for dynamic Bayesian networks," in *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*, Stanford, USA, Jul. 2000, pp. 176–183.

[7] T. Schön, F. Gustafsson, and P.-J. Nordlund, "Marginalized particle filters for mixed linear/nonlinear state-space models," *IEEE Transactions on Signal Processing*, vol. 53, no. 7, pp. 2279–2289, Jul. 2005.

[8] A. Doucet, S. J. Godsill, and M. West, "Monte Carlo filtering and smoothing with application to time-varying spectral estimation," in *Proceedings of the 2000 IEEE International Conference on Computer Vision (ICCV)*, Istanbul , Turkey, Jun. 2000.

[9] S. J. Godsill, A. Doucet, and M. West, "Monte Carlo smoothing for nonlinear time series," *Journal of the American Statistical Association*, vol. 99, no. 465, pp. 156–168, Mar. 2004.

[10] R. Douc, A. Garivier, E. Moulines, and J. Olsson, "Sequential Monte Carlo smoothing for general state space hidden Markov models," *Submitted to Annals of Applied Probability*, 2010.

[11] M. Briers, A. Doucet, and S. Maskell, "Smoothing algorithms for state-space models," *Annals of the Institute of Statistical Mathematics*, vol. 62, no. 1, pp. 61–89, Feb. 2010.

[12] P. Fearnhead, D. Wyncoll, and J. Tawn, "A sequential smoothing algorithm with linear computational cost," *Biometrika*, vol. 97, no. 2, pp. 447–464, 2010.

[13] W. Fong, S. J. Godsill, A. Doucet, and M. West, "Monte Carlo smoothing with application to audio signal enhancement," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 438–449, Feb. 2002.

[14] F. Lindsten and T. B. Schön, "Identification of mixed linear/nonlinear state-space models," in *Proceedings of the 49th IEEE Conference on Decision and Control (CDC)*, Atlanta, USA, Dec. 2010.

[15] J. Handschin and D. Mayne, "Monte Carlo techniques to estimate the conditional expectation in multi-stage non-linear filtering," *International Journal of Control*, vol. 9, no. 5, pp. 547–559, May 1969.

[16] F. Lindsten, T. B. Schön, and J. Olsson, "An explicit variance reduction expression for the Rao-Blackwellised particle filter," in *Proceedings of the 18th World Congress of the International Federation of Automatic Control (IFAC) (accepted for publication)*, Milan, Italy, Aug. 2011.

[17] H. E. Rauch, F. Tung, and C. T. Striebel, "Maximum likelihood estimates of linear dynamic systems," *AIAA Journal*, vol. 3, no. 8, pp. 1445–1450, Aug. 1965.

[18] T. Kailath, A. H. Sayed, and B. Hassibi, *Linear Estimation*. Upper Saddle River, NJ, USA: Prentice Hall, 2000.

[19] M. L. Andrade Netto, L. Gimeno, and M. J. Mendes, "A new spline algorithm for non-linear filtering of discrete time systems," in *Proceedings of the 7th Triennial World Congress*, Helsinki, Finland, 1979, pp. 2123–2130.

Avdelning, Institution
Division, Department

Division of Automatic Control
Department of Electrical Engineering

Datum
Date

2011-05-30

Språk
Language

☐ Svenska/Swedish
☒ Engelska/English
☐ ________________

Rapporttyp
Report category

☐ Licentiatavhandling
☐ Examensarbete
☐ C-uppsats
☐ D-uppsats
☒ Övrig rapport
☐ ________________

ISBN
—

ISRN
—

Serietitel och serienummer
Title of series, numbering

ISSN
1400-3902

LiTH-ISY-R-3018

URL för elektronisk version

`http://www.control.isy.liu.se`

Titel
Title

Rao-Blackwellised particle smoothers for mixed linear/nonlinear state-space models

Författare
Author

Fredrik Lindsten, Thomas B. Schön

Sammanfattning
Abstract

We consider the smoothing problem for a class of mixed linear/nonlinear state-space models. This type of models contain a certain tractable substructure. When addressing the filtering problem using sequential Monte Carlo methods, it is well known that this structure can be exploited in a Rao-Blackwellised particle filter. However, to what extent the same property can be used when dealing with the smoothing problem is still a question of central interest. In this paper, we propose different particle based methods for addressing the smoothing problem, based on the forward filtering/backward simulation approach to particle smoothing. This leads to a group of Rao-Blackwellised particle smoothers, designed to exploit the tractable substructure present in the model.

Nyckelord
Keywords

Nonlinear estimation, smoothing, particle methods, Rao-Blackwellisation