Particle Smoothing for Conditionally Linear Gaussian Models as Message Passing over Factor Graphs

Giorgio M. Vitetta, *Senior Member, IEEE,* Emilio Sirignano, *Student Member, IEEE,* and Francesco Montorsi, *Member, IEEE*.

***Abstract*—In this manuscript the fixed-lag smoothing problem for conditionally linear Gaussian state-space models is investigated from a factor graph perspective. More specifically, after formulating Bayesian smoothing for an arbitrary state-space model as forward-backward message passing over a factor graph, we focus on the above mentioned class of models and derive two novel particle smoothers for it. Both the proposed techniques are based on the well known two-filter smoothing approach and employ marginalized particle filtering in their forward pass. However, on the one hand, the first smoothing technique can only be employed to improve the accuracy of state estimates with respect to that achieved by forward filtering. On the other hand, the second method, that belongs to the class of Rao-Blackwellized particle smoothers, provides also a point mass approximation of the so called joint smoothing distribution. Finally, our smoothing algorithms are compared, in terms of estimation accuracy and computational requirements, with a Rao-Blackwellized particle smoother recently proposed by Lindsten et al. in [20].**

***Index terms*—** State Space Representation, Hidden Markov Model, Filtering, Smoothing, Marginalized Particle Filter, Belief Propagation.

I. Introduction

Bayesian filtering and *Bayesian smoothing* for *state space models* (SSMs) are two interrelated problems that have received significant attention for a number of years [1]. Bayesian filtering allows to recursively estimate, through a prediction/update mechanism, the *probability density function* (pdf) of the current state of any SSM, given the history of some observed data up to the current time. Unluckily, the general formulas describing the Bayesian filtering recursion (e.g., see [2, eqs. (4)-(5)]) admit closed form solutions for *linear Gaussian* and *linear Gaussian mixture* SSMs [1] only. On the contrary, approximate solutions are available for general *nonlinear* models; these are based on *sequential Monte Carlo* (SMC) techniques (also known as *particle filtering* methods) which represent a powerful tool for numerical approximations [3]-[5].

Bayesian smoothing, instead, exploits an entire batch of measurements to generate a significantly better estimate of the pdf (i.e., a *smoothed* or *smoothing* pdf) of a SSM state over a given observation interval. Two general methods are available in the literature for recursively calculating smoothing densities, namely the *forward filtering-backward smoothing recursion* [4], [7] and the method based on the *two-filter smoothing formula* [8]-[10]. In both cases the computation of smoothing densities requires combining the predicted and/or filtered densities generated by a standard Bayesian filtering method with those produced by a recursive backward technique (known as *backward information filtering*, BIF, in the case of two-filter smoothing). Similarly as filtering, closed form solutions for Bayesian smoothing are available for *linear Gaussian* and *linear Gaussian mixture* models [1], [11]. This has motivated the development of various SMC approximations (also known as *particle smoothers*) for the above mentioned two methods in the case of nonlinear SSMs (e.g., see [4], [6], [8], [9], [12]-[15] and references therein).

While SMC methods can be directly applied to an arbitrary nonlinear SSM for both filtering and smoothing, it has been recognized that their estimation accuracy can be improved in the case of *conditionally linear Gaussian* (CLG) SSMs. In fact, the linear substructure of such models can be marginalised, so reducing the dimension of their sample space [16], [17]. This idea has led to the development of important SMC techniques for filtering and smoothing, known as *Rao-Blackwellized particle filtering* (also dubbed *marginalized particle filtering*, MPF) [17], [18] and *Rao-Blackwellized particle smoothing* (RBPS) [13], [14], [20], respectively.

Recently, the filtering problem for CLG SSMs has been investigated from a *factor graph* (FG) perspective in [21], where a novel interpretation of MPF as a *forward only message passing algorithm* over a specific FG has been provided and a novel extension of it, dubbed *turbo filtering* (TF), has been derived. In this manuscript, the same conceptual approach is employed to provide new insights in the *fixed-interval smoothing problem* [13] and to develop novel solutions for it. The proposed solutions are represented by two novel particle smoothing methods, the first one dubbed *serial particle smoothing* (SPS), the second one *Rao-Blackwellized serial smoothing* (RBSS). These methods share the following relevant features: a) they are based on the two-filter smoothing formula and employ MPF in their forward pass; b) they can be derived applying the well known *sum-product algorithm* (SPA) [23], [24], together with a specific scheduling procedure, to the same FG developed in [21] and [22] for a CLG SSM; c) unlike the RBPS methods devised in [13] and [14], they can be employed for a SSM in which both the linear and

Manuscript received Month Day, 2016; revised Month Day, Year.

The authors are with the University of Modena and Reggio Emilia, Dept. of Engineering "Enzo Ferrari", Via P. Vivarelli 10/1, 41125 Modena (Italy), email: giorgio.vitetta@unimore.it, emilio.sirignano@unimore.it, francesco.montorsi@gmail.com.

nonlinear state components influence each other. Moreover, the computational requirements of the SPS technique are comparable with that of MPF, since a single estimate of the nonlinear state trajectory is generated in its backward pass; consequently, it is substantially simpler than the RBPS methods illustrated in [13], [14] and [20]. On the contrary, the computational load of the RBPS method is significantly higher, being close to that of the method developed in [20]; in fact, these methods employ similar procedures to generate a *joint smoothing distribution* over the entire observation interval.

It is worth mentioning that the application of FG methods to Bayesian smoothing is not new. However, as far as we know, the few results available in the technical literature about this topic refer to the case of *linear Gaussian* SSMs only [23], [25], [26], whereas we exclusively focus on the case in which the mathematical laws expressing state dynamics and/or available observations are *nonlinear*.

The remaining part of this manuscript is organized as follows. The model of the considered CLG SSM is briefly illustrated in Section II. A representation of the smoothing problem through Forney-style FGs for both an arbitrary SSM and a CLG SSM is provided in Section III. In Section IV the SPS and the RBSS techniques are developed for a CLG SSM, and are compared with that of other particle smoothing methods. Section V is devoted to comparing, in terms of accuracy and computational effort, our FG-based smoothing algorithms with the RBPS method developed in [20]. Finally, some conclusions are offered in Section VI.

Notations: The *probability density function* (pdf) of a random vector $\mathbf{R}$ evaluated at point $\mathbf{r}$ is denoted $f(\mathbf{r})$; $\mathcal{N}(\mathbf{r};\eta_{\mathbf{r}},\mathbf{C}_{\mathbf{r}})$ represents the pdf of a Gaussian random vector $\mathbf{R}$ characterized by the mean $\eta_{\mathbf{r}}$ and covariance matrix $\mathbf{C}_{\mathbf{r}}$ evaluated at point $\mathbf{r}$; the *precision* (or *weight*) *matrix* associated with the covariance matrix $\mathbf{C}_{\mathbf{r}}$ is denoted $\mathbf{W}_{\mathbf{r}}$, whereas the *transformed mean vector* $\mathbf{W}_{\mathbf{r}}\eta_{\mathbf{r}}$ is denoted $\mathbf{w}_{\mathbf{r}}$.

II. System Model

In the following we focus on the discrete-time CLG SSM described in [21], [22]. In brief, the SSM *hidden state* in the l-th interval is represented by the D-dimensional real vector $\mathbf{x}_l \triangleq [x_{0,l}, x_{1,l}, ..., x_{D-1,l}]^T$; this is partitioned in a) its D_L-dimensional *linear component* $\mathbf{x}_l^{(L)} \triangleq [x_{0,l}^{(L)}, x_{1,l}^{(L)}, ..., x_{D_L-1,l}^{(L)}]^T$ and b) its D_N-dimensional *nonlinear component* $\mathbf{x}_l^{(N)} \triangleq [x_{0,l}^{(N)}, x_{1,l}^{(N)}, ..., x_{D_N-1,l}^{(L)}]^T$ (with $D_L < D$ and $D_N = D - D_L$). The update equations of the linear and nonlinear components are given by

$$\mathbf{x}_{l+1}^{(L)} = \mathbf{A}_l^{(L)}\left(\mathbf{x}_l^{(N)}\right)\mathbf{x}_l^{(L)} + \mathbf{f}_l^{(L)}\left(\mathbf{x}_l^{(N)}\right) + \mathbf{w}_l^{(L)}, \quad (1)$$

and

$$\mathbf{x}_{l+1}^{(N)} = \mathbf{A}_l^{(N)}\left(\mathbf{x}_l^{(N)}\right)\mathbf{x}_l^{(L)} + \mathbf{f}_l^{(N)}\left(\mathbf{x}_l^{(N)}\right) + \mathbf{w}_l^{(N)}, \quad (2)$$

respectively; here, $\mathbf{f}_l^{(L)}(\mathbf{x})$ ($\mathbf{f}_l^{(N)}(\mathbf{x})$) is a time-varying D_L-dimensional (D_N-dimensional) real function, $\mathbf{A}_l^{(L)}(\mathbf{x}_l^{(N)})$ ($\mathbf{A}_l^{(N)}(\mathbf{x}_l^{(N)})$) is a time-varying $D_L \times D_L$ ($D_N \times D_L$) real matrix and $\mathbf{w}_l^{(L)}$ ($\mathbf{w}_l^{(N)}$) is the l-th element of the process noise sequence $\{\mathbf{w}_k^{(L)}\}$ ($\{\mathbf{w}_k^{(N)}\}$), which consists of D_L-dimensional (D_N-dimensional) *independent and identically distributed* (iid) noise vectors (statistical independence between $\{\mathbf{w}_k^{(L)}\}$ and $\{\mathbf{w}_k^{(N)}\}$ is also assumed for simplicity).

In the following Section we mainly focus on the so-called *fixed-interval smoothing problem* [13]; this consists of computing the sequence of *posterior densities* $\{f(\mathbf{x}_l|\mathbf{y}_{1:T}),\ l = 1, 2, ..., T\}$ (where T represents the length of the observation interval), given a) the *initial* pdf $f(\mathbf{x}_1)$ and b) the $T \cdot P$-dimensional *measurement* vector $\mathbf{y}_{1:T} = \left[\mathbf{y}_1^T, \mathbf{y}_2^T, ..., \mathbf{y}_T^T\right]^T$, where

$$\mathbf{y}_l \triangleq [y_{0,l}, y_{1,l}, ..., y_{P-1,l}]^T = \mathbf{h}_l\left(\mathbf{x}_l^{(N)}\right) + \mathbf{B}_l\left(\mathbf{x}_l^{(N)}\right)\mathbf{x}_l^{(L)} + \mathbf{e}_l, \quad (3)$$

with $l = 1, 2, ..., T$, is a P-dimensional vector collecting the noisy observations available about $\mathbf{x}_l$ in the l-th interval. Here, $\mathbf{B}_l(\mathbf{x}_l^{(N)})$ is a time-varying $P \times D_L$ real matrix, $\mathbf{h}_l(\mathbf{x}_l^{(N)})$ is a time-varying P-dimensional real function and $\mathbf{e}_l$ the l-th element of the measurement noise sequence $\{\mathbf{e}_k\}$ consisting of P-dimensional iid noise vectors (all independent of both $\{\mathbf{w}_k^{(N)}\}$ and $\{\mathbf{w}_k^{(L)}\}$).

III. A FG-Based Representation of Smoothing

In this Section we formulate the computation of the *marginal smoothed density* $f(\mathbf{x}_l|\mathbf{y}_{1:T})$ (with $l = 1, 2, ..., T$) as a message passing algorithm over a specific FG for the following two cases: C.1) a SSM whose statistical behavior is characterized by the *Markov model* $f(\mathbf{x}_{l+1}|\mathbf{x}_l)$ and the *observation model* $f(\mathbf{y}_l|\mathbf{x}_l)$; C.2) a SSM having the additional property of being CLG (see the previous Section).

In case C.1 we take into consideration the *joint* pdf $f(\mathbf{x}_l, \mathbf{y}_{1:T})$ in place of the *posterior* pdf $f(\mathbf{x}_l|\mathbf{y}_{1:T})$. This choice is motivated by the fact that: a) the computation of the former pdf can be easily formulated as a *recursive* message passing algorithm over a proper FG, since, as shown below, this involves only products and sums of products; b) the former pdf, being proportional to the latter one, is represented by the *same* FG (this issue is discussed in [23, Sec. II, p. 1297]). Note that the validity of statement a) relies on the following mathematical results: 1) the factorization (e.g., see [8, Sec. 3])

$$\begin{aligned} f(\mathbf{x}_l, \mathbf{y}_{1:T}) &= f\left(\mathbf{y}_{l:T}\,|\mathbf{x}_l, \mathbf{y}_{1:(l-1)}\right) f\left(\mathbf{x}_l, \mathbf{y}_{1:(l-1)}\right) \\ &= f\left(\mathbf{y}_{l:T}\,|\mathbf{x}_l\right) f\left(\mathbf{x}_l, \mathbf{y}_{1:(l-1)}\right) \end{aligned} \quad (4)$$

for the pdf of interest; 2) the availability of recursive methods, known as *Bayesian filtering* [2] (and called *forward filtering*, FF, in the following for clarity) and *backward information filtering* (BIF; e.g., see [8]) for computing the joint pdf $f(\mathbf{x}_l, \mathbf{y}_{1:(l-1)})$ and the conditional pdf $f(\mathbf{y}_{l:T}|\mathbf{x}_l)$, respectively, for any l.

As far as FF is concerned, the formulation illustrated in [21, Sec. 2] is adopted here; this consists of a *measurement update* (MU) step followed by a *time update* (TU) step and assumes the a priori knowledge of the pdf $f(\mathbf{x}_1)$ for its initialization. In the MU step of its l-th recursion (with $l = 1, 2, ..., T$) the joint pdf

$$f(\mathbf{x}_l, \mathbf{y}_{1:l}) = f\left(\mathbf{x}_l, \mathbf{y}_{1:(l-1)}\right) f\left(\mathbf{y}_l\,|\mathbf{x}_l\right) \quad (5)$$

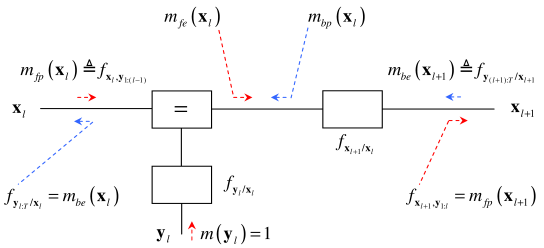


Fig. 1: Graphical representation of the message passing for the evaluation of the joint pdf $f(\mathbf{x}_{l+1}, \mathbf{y}_{1:l})$ and of the conditional pdf $f(\mathbf{y}_{l:T}|\mathbf{x}_l)$ on the basis of eqs. (5)-(6) and (7)-(8), respectively (the forward and backward message flows are indicated by red and blue arrows, respectively)

is computed on the basis of pdf $f(\mathbf{x}_l, \mathbf{y}_{1:(l-1)})$, and the new measurement vector $\mathbf{y}_l$. In the TU step, instead, the pdf $f(\mathbf{x}_l, \mathbf{y}_{1:l})$ (5) is exploited to compute the pdf

$$f(\mathbf{x}_{l+1}, \mathbf{y}_{1:l}) = \int f(\mathbf{x}_{l+1}|\mathbf{x}_l) f(\mathbf{x}_l, \mathbf{y}_{1:l}) d\mathbf{x}_l, \quad (6)$$

representing a *prediction* about the future state $\mathbf{x}_{l+1}$.

A conceptually similar recursive procedure can be easily developed for the $(T-l)$-th recursion of BIF (with $l = T-1, T-2, ..., 1$). In fact, this can be formulated as a TU step followed by a MU step; these are expressed by

$$f(\mathbf{y}_{(l+1):T}|\mathbf{x}_l) = \int f(\mathbf{y}_{(l+1):T}|\mathbf{x}_{l+1}) f(\mathbf{x}_{l+1}|\mathbf{x}_l) d\mathbf{x}_l \quad (7)$$

and

$$f(\mathbf{y}_{l:T}|\mathbf{x}_l) = f(\mathbf{y}_{(l+1):T}|\mathbf{x}_l) f(\mathbf{y}_l|\mathbf{x}_l), \quad (8)$$

respectively. Note that this procedure requires the knowledge of the pdf $f(\mathbf{y}_T|\mathbf{x}_T)$ for its initialization (see (7)).

Eqs. (5)-(8) show that each of the FF (or BIF) recursions involves only *products of pdfs* and a *sum* (i.e., an integration) *of products*. For this reason, based on the general rules about graphical models illustrated in [23, Sect. II], such recursions can be interpreted as specific instances of the SPA[1] applied to the *cycle free* FG of Fig. 1 (where the simplified notation of [23] is employed).

More specifically, it is easy to show that eqs. (5) and (6) can be seen as a SPA-based algorithm for *forward* message passing over the FG shown in Fig. 1 (the flow of forward messages is indicated by *red* arrows in the considered figure). In fact, if the FG is fed by the message[2]

$$\vec{m}_{fp}(\mathbf{x}_l) \triangleq f(\mathbf{x}_l, \mathbf{y}_{1:(l-1)}), \quad (9)$$

the forward message emerging from the *equality node* and that passed along the edge associated with $\mathbf{x}_{l+1}$ are given by $\vec{m}_{fe}(\mathbf{x}_l) = f(\mathbf{x}_l, \mathbf{y}_{1:l})$ and $f(\mathbf{x}_{l+1}, \mathbf{y}_{1:l}) = \vec{m}_{fp}(\mathbf{x}_{l+1})$, respectively [21], [22]. A similar interpretation can be provided for eqs. (7) and (8), which, however, can be reformulated as a SPA-based algorithm for *backward* message passing over the considered FG. In fact, if the input message

$$\overleftarrow{m}_{be}(\mathbf{x}_{l+1}) \triangleq f(\mathbf{y}_{(l+1):T}|\mathbf{x}_{l+1}) \quad (10)$$

enters the FG along the half edge associated with $\mathbf{x}_{l+1}$ (the flow of backward messages is indicated by *blue* arrows in Fig. 1), the backward message $\overleftarrow{m}_{bp}(\mathbf{x}_l)$ (emerging from the node associated with the pdf $f(\mathbf{x}_{l+1}|\mathbf{x}_l)$) is given by (see (7))

$$\begin{aligned}\overleftarrow{m}_{bp}(\mathbf{x}_l) &= \int \overleftarrow{m}_{be}(\mathbf{x}_l) f(\mathbf{x}_{l+1}|\mathbf{x}_l) d\mathbf{x}_l \\ &= \int f(\mathbf{y}_{(l+1):T}|\mathbf{x}_{l+1}) f(\mathbf{x}_{l+1}|\mathbf{x}_l) d\mathbf{x}_l \\ &= f(\mathbf{y}_{(l+1):T}|\mathbf{x}_l). \end{aligned} \quad (11)$$

Therefore, the backward message emerging from the equality node can be evaluated as (see (8) and (10))

$$\begin{aligned} &f(\mathbf{y}_l|\mathbf{x}_l) \overleftarrow{m}_{bp}(\mathbf{x}_l) = f(\mathbf{y}_l|\mathbf{x}_l) f(\mathbf{y}_{(l+1):T}|\mathbf{x}_l) \\ &= f(\mathbf{y}_{l:T}|\mathbf{x}_l) = \overleftarrow{m}_{be}(\mathbf{x}_l) \end{aligned} \quad (12)$$

and this concludes our proof.

These results easily lead to the conclusion that, once the forward and backward message passing algorithms illustrated above have been carried out over the entire observation interval, the smoothed pdf $f(\mathbf{x}_l, \mathbf{y}_{1:T})$ can be evaluated as (see (4), (9) and (12))

$$f(\mathbf{x}_l, \mathbf{y}_{1:T}) = \vec{m}_{fp}(\mathbf{x}_l) \overleftarrow{m}_{be}(\mathbf{x}_l), \quad (13)$$

with $l = 1, 2, ..., T$ (note that $\overleftarrow{m}_{be}(\mathbf{x}_T) = 1$ and $\vec{m}_{fp}(\mathbf{x}_1) = f(\mathbf{x}_1)$) or, alternatively, as

$$f(\mathbf{x}_l, \mathbf{y}_{1:T}) = \vec{m}_{fe}(\mathbf{x}_l) \overleftarrow{m}_{bp}(\mathbf{x}_l). \quad (14)$$

It is worth noting that, despite the conceptual simplicity of the procedure illustrated above, its implementation can represent a formidable task, mainly because of the multidimensional integration required in (6) and (7). This has motivated the development of the so called *Rao-Blackwellization* approach, according to which the state vector $\mathbf{x}_l$ is partitioned in a nonlinear component $\mathbf{x}_l^{(N)}$ and a linear component $\mathbf{x}_l^{(L)}$; moreover, from a statistical viewpoint, $\mathbf{x}_l^{(N)}$ is represented through a set of weighted particles, whereas $\mathbf{x}_l^{(L)}$ through a set of a particle-dependent Gaussian pdfs (i.e., through a *Gaussian mixture*, GM). In [21] and [22] it has been shown that Rao-Blackwellized filtering can be seen as an instance of the SPA applied to the FG we develop for case C.2; the new FG is based not only on that analysed for case C.1, but also on the idea of representing a mixed linear/nonlinear SSM as the concatenation of two interacting sub-models, one referring to the linear component of system state, the other one to its nonlinear component [21]. For this reason, two distinct sub-graphs are drawn, one referring to smoothing for $\mathbf{x}_l^{(L)}$ (under the assumption that $\mathbf{x}_l^{(N)}$ is *known*), the other one to smoothing for $\mathbf{x}_l^{(N)}$ (under the dual assumption that $\mathbf{x}_l^{(L)}$ is *known*). The first (second) sub-graph can be easily obtained from the FG

[1] In a Forney-style FG, such a rule can be formulated as follows [23]: the message emerging from a node f along some edge x is formed as the product of f and all the incoming messages along all the edges that enter the node f except x, summed over all the involved variables except x.

[2] In the following the acronyms *bp*, *be*, *fp*, *fe* and *sm* are employed in the subscripts of various messages, so that readers can easily understand their meaning; in fact, the messages these acronyms refer to represent a form of *backward prediction*, *backward estimation*, *forward prediction, forward estimation* and *smoothing*, respectively.

shown in Fig. 1 by including the contribution of a *pseudo-measurement*, denoted $\mathbf{z}_l^{(L)}$ ($\mathbf{z}_l^{(N)}$), computed on the basis of the knowledge of $\mathbf{x}_l^{(N)}$ ($\mathbf{x}_l^{(L)}$). In fact, smoothing for the linear component $\mathbf{x}_l^{(L)}$ can benefit not only from the measurement $\mathbf{y}_l$, but also from the vector (see (2))

$$\mathbf{z}_l^{(L)} \triangleq \mathbf{x}_{l+1}^{(N)} - \mathbf{f}_l^{(N)}\left(\mathbf{x}_l^{(N)}\right) = \mathbf{A}_l^{(N)}\left(\mathbf{x}_l^{(N)}\right)\mathbf{x}_l^{(L)} + \mathbf{w}_l^{(N)}, \tag{15}$$

which, from a statistical viewpoint, is characterized by the pdf $f(\mathbf{z}_l^{(L)}|\mathbf{x}_l^{(L)}, \mathbf{x}_l^{(N)})$. Similarly, the vector (see (1))

$$\mathbf{z}_l^{(N)} \triangleq \mathbf{x}_{l+1}^{(L)} - \mathbf{A}_l^{(L)}\left(\mathbf{x}_l^{(N)}\right)\mathbf{x}_l^{(L)} = \mathbf{f}_l^{(L)}\left(\mathbf{x}_l^{(N)}\right) + \mathbf{w}_l^{(L)}, \tag{16}$$

characterized by the pdf $f(\mathbf{z}_l^{(N)}|\mathbf{x}_l^{(N)})$, can be exploited in smoothing for the nonlinear component $\mathbf{x}_l^{(N)}$. Then, the two sub-graphs are merged by adding five distinct equality nodes, associated with the shared variables (namely, $\mathbf{y}_l$, $\mathbf{x}_l^{(L)}$, $\mathbf{x}_l^{(N)}$, $\mathbf{x}_{l+1}^{(L)}$ and $\mathbf{x}_{l+1}^{(N)}$). This leads to the overall FG illustrated in Fig. 2, in which the sub-graph referring to the linear (nonlinear) state component is identified by red (blue) lines, whereas the equality nodes added to merge them are identified by black lines. This graphical model deserves the following important comments:

- No approximation is made in deriving it.
- Its upper (lower) sub-graph contains an additional node representing the pseudo-measurement pdf $f(\mathbf{z}_l^{(L)}|\mathbf{x}_l^{(L)}, \mathbf{x}_l^{(N)})$ ($f(\mathbf{z}_l^{(N)}|\mathbf{x}_l^{(N)})$) and a *specific node* not referring to a pdf factorization, but representing the *transformation* from the couple $(\mathbf{x}_l^{(N)}, \mathbf{x}_{l+1}^{(N)})$ to $\mathbf{z}_l^{(L)}$ ($(\mathbf{x}_l^{(L)}, \mathbf{x}_{l+1}^{(L)})$ to $\mathbf{z}_l^{(N)}$); the last peculiarity is evidenced by the presence of an arrow on all the edges connected to such a node.
- Unlike the FG represented in Fig. 1, it is *not cycle-free*. This property is due to the fact that, generally speaking, smoothing for $\mathbf{x}_l^{(N)}$ is not decouplable from that for $\mathbf{x}_l^{(L)}$; in other words, uncertainty in each component of system state needs to be accounted for in the distribution of the other state component.

Given the FG of Fig. 2, we would like to follow the same line of reasoning as that illustrated for the graphical model of Fig. 1. In particular, given the input backward (marginal) messages $\overleftarrow{m}_{be}(\mathbf{x}_{l+1}^{(L)}) \triangleq f(\mathbf{y}_{(l+1):T}, \mathbf{z}_{(l+1):T}^{(L)}, \mathbf{x}_{l+1}^{(L)})$ and $\overleftarrow{m}_{be}(\mathbf{x}_{l+1}^{(N)}) \triangleq f(\mathbf{y}_{(l+1):T}, \mathbf{z}_{(l+1):T}^{(N)}, \mathbf{x}_{l+1}^{(N)})$, we are interested in deriving a BIF algorithm[3] based on this FG and generating the output backward (marginal) messages $\overleftarrow{m}_{be}(\mathbf{x}_l^{(L)}) = f(\mathbf{y}_{l:T}, \mathbf{z}_{l:T}^{(L)}, \mathbf{x}_l^{(L)})$ and $\overleftarrow{m}_{be}(\mathbf{x}_l^{(N)}) = f(\mathbf{y}_{l:T}, \mathbf{z}_{l:T}^{(N)}, \mathbf{x}_l^{(N)})$ on the basis of the available a priori information and the noisy measurement $\mathbf{y}_l$. Note that, given the FG of of Fig. 2, the evaluation of the message $\overleftarrow{m}_{be}(\mathbf{x}_l^{(L)})$ ($\overleftarrow{m}_{be}(\mathbf{x}_l^{(N)})$) requires *marginalization* with respect to $\mathbf{x}_l^{(N)}$ ($\mathbf{x}_l^{(L)}$). It is well known that, on the one hand, in cycle-free graphical models *exact* marginalization can be accomplished by means of the SPA; on the other hand, the application of the SPA to a graphical model containing cycles unavoidably leads to *approximate solutions*

[3]FF based on this FG has been deeply investigated in [21] and [22] and, for this reason, is not analysed here.

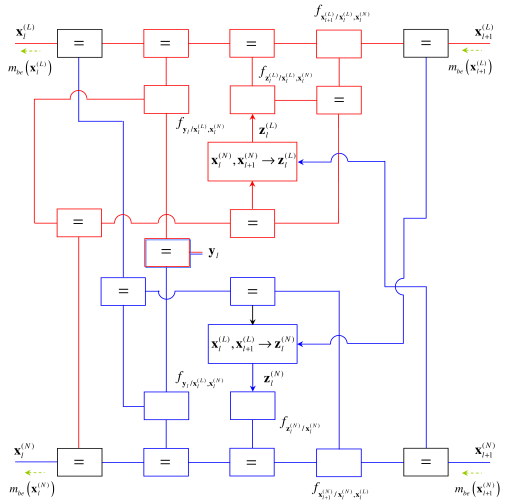


Fig. 2: Factor graph for case C.2. The sub-graph referring to the linear (nonlinear) state component is identified by red (blue) lines, whereas the equality nodes introduced to merge the two sub-graphs by black lines. The direction of the messages passed over the half edges $\mathbf{x}_l^{(L)}$ and $\mathbf{x}_l^{(N)}$ (inputs) and over the half edges $\mathbf{x}_{l+1}^{(L)}$ and $\mathbf{x}_{l+1}^{(N)}$(outputs) is indicated by green arrows.

[24], whatever *message scheduling* is adopted. Despite this, we believe that, like in the related problem of filtering for CLG SSMs [21], [22], this approach can lead to the development of accurate and computationally efficient smoothing algorithms, as shown in the following Section.

IV. Particle Smoothing as Message Passing

In this Section we first illustrate some assumptions about the statistical properties of the SSM described in Section II. Then, we develop the SPS and RBSS techniques. Finally, we compare the most relevant features of these techniques with those of the other RBPS algorithms available in the technical literature.

A. Statistical properties of the considered SSM

Even if the graphical model shown in Fig. 2 can be employed for any mixed linear/nonlinear system described by eqs. (1)-(3), the methods derived in this Section apply, like MPF [17] and TF [21], to the specific class of CLG SSMs. For this reason, following [21], [22] we assume that: a) the process noise $\{\mathbf{w}_k^{(L)}\}$ ($\{\mathbf{w}_k^{(N)}\}$) is Gaussian and all its elements have zero mean and covariance $\mathbf{C}_w^{(L)}$ ($\mathbf{C}_w^{(N)}$) for any l; b) the measurement noise $\{\mathbf{e}_k^{(L)}\}$ is Gaussian having zero mean and covariance matrix $\mathbf{C}_e$ for any l; c) all the above mentioned Gaussian processes are statistically independent. Under these assumptions, the pdfs $f(\mathbf{y}_l|\mathbf{x}_l^{(L)}, \mathbf{x}_l^{(N)})$, $f(\mathbf{z}_l^{(L)}|\mathbf{x}_l^{(L)})$ and $f(\mathbf{x}_{l+1}^{(L)}|\mathbf{x}_l^{(L)}, \mathbf{x}_l^{(N)})$ are Gaussian with mean (covariance matrix) $\mathbf{B}_l(\mathbf{x}_l^{(N)})\mathbf{x}_l^{(L)} + \mathbf{h}_l(\mathbf{x}_l^{(N)})$, $\mathbf{A}_l^{(N)}(\mathbf{x}_l^{(N)})\,\mathbf{x}_l^{(L)}$ and $\mathbf{f}_l^{(L)}(\mathbf{x}_l^{(N)}) + \mathbf{A}_l^{(L)}(\mathbf{x}_l^{(N)})\,\mathbf{x}_l^{(L)}$, respectively ($\mathbf{C}_e$, $\mathbf{C}_w^{(N)}$ and $\mathbf{C}_w^{(L)}$, respectively). Similarly, the pdfs $f(\mathbf{z}_l^{(N)}|\mathbf{x}_l^{(N)})$ and

$f(\mathbf{x}_{l+1}^{(N)}|\mathbf{x}_l^{(N)}, \mathbf{x}_l^{(L)})$ are Gaussian with mean (covariance matrix) $\mathbf{f}_l^{(L)}(\mathbf{x}_l^{(N)})$ and $\mathbf{f}_l^{(N)}(\mathbf{x}_l^{(N)}) + \mathbf{A}_l^{(N)}(\mathbf{x}_l^{(N)})\,\mathbf{x}_l^{(L)}$, respectively ($\mathbf{C}_w^{(L)}$ and $\mathbf{C}_w^{(N)}$, respectively).

B. Derivation of the particle serial smoother

In developing our first particle smoothing algorithm the following fundamental requirements have been set to limit its computational complexity as much as possible: 1) MPF is employed in its (single) forward pass[4]; 2) the statistical information generated by the BIF technique adopted in its (single) backward pass can be easily combined with that provided by MPF to generate the marginal smoothed densities of interest; c) the estimates of both the linear component and the nonlinear component are available at the end of the backward pass and each of them is represented by a *single trajectory*.

As far as the use of MPF is concerned, the following notation is adopted here:

- The j-th particle *predicted* for $\mathbf{x}_l^{(N)}$ in the $(l-1)$-th recursion (with $l = 2, 3, ..., T$) of this algorithm is denoted $\mathbf{x}_{l/(l-1),j}^{(N)}$ (with $j = 0, 1, ..., N_p - 1$, where N_p is the overall number of particles); moreover, the weight assigned to this particle is denoted $w_{l/(l-1),j}$ (this weight is equal to $1/N_p$ for any j, since the use of particle resampling in each recursion is assumed).
- The weight computed for the particle $\mathbf{x}_{l/(l-1),j}^{(N)}$ in the following (i.e., in the l-th) recursion on the basis of the new measurement $\mathbf{y}_l$ is denoted $w_{l/l,j}$ (with $j = 0, 1, ..., N_p - 1$).
- The j-th particle available after particle resampling based on the weights $\{w_{l/l,j}\}$ is denoted $\mathbf{x}_{l/l,j}^{(N)}$ (note that the set $\{\mathbf{x}_{l/l,j}^{(N)}\}$ usually contains multiple copies of the most likely particles of the set $\{\mathbf{x}_{l/(l-1),j}^{(N)}\}$).
- The Gaussian model *predicted* for $\mathbf{x}_l^{(L)}$ in the $(l-1)$-th recursion and associated with $\mathbf{x}_{l/(l-1),j}^{(N)}$ is denoted $\mathcal{N}(\mathbf{x}_l^{(L)}; \eta_{fp,l,j}^{(L)}, \mathbf{C}_{fp,l,j}^{(L)})$; note, however, that only a portion of these Gaussian models is usually updated in the next (i.e., l-th) recursion on the basis of $\mathbf{y}_l$; in fact, this task follows particle resampling, which typically leads to discarding a fraction of the particles collected in the set $\{\mathbf{x}_{l/(l-1),j}^{(N)}\}$.

In the *forward pass* of the proposed SPS method, the statistical information generated by MPF and saved for further processing is conveyed by the *forward* messages

$$\vec{m}_{fp,j}\left(\mathbf{x}_l^{(N)}\right) \triangleq \delta\left(\mathbf{x}_l^{(N)} - \mathbf{x}_{l/(l-1),j}^{(N)}\right) \tag{17}$$

and

$$\vec{m}_{fp,j}\left(\mathbf{x}_l^{(L)}\right) \triangleq \mathcal{N}\left(\mathbf{x}_l^{(L)}; \eta_{fp,l,j}^{(L)}, \mathbf{C}_{fp,l,j}^{(L)}\right), \tag{18}$$

with $j = 0, 1, ..., N_p - 1$. Let us now focus on the *backward pass* of the SPS algorithm and, in particular, tackle the problems of a) developing a *recursive* BIF algorithm based on the FG of Fig. 2 and b) merging the statistical information it generates with those computed in the forward pass (smoothing). The recursive algorithm we propose results from the application of the SPA to the FG shown in Fig. 2 and is based on the *message scheduling* illustrated in Figs. 3-(a) and 3-(b), both referring to the $(T-l)$-th recursion (with $l = T-1, T-2, ..., 1$) and the j-th particle[5] $\mathbf{x}_{l/(l-1),j}^{(N)}$ (with $j = 0, 1, ..., N_p - 1$). The processing accomplished by the SPS algorithm within the considered recursion can be divided in three parts, that are executed *serially* (and this motivates the presence of the adjective 'serial' in the name of the devised smoothing algorithms). The *first part*, that involves the computation of the messages illustrated in Fig. 3-(a), concerns the linear state component, since it generates the particle-dependent statistical model $\overleftarrow{m}_{be,j}(\mathbf{x}_l^{(L)})$ (conveying a particle-dependent backward estimate of $\mathbf{x}_l^{(L)}$) and combines it with the forward model $\vec{m}_{fp,j}(\mathbf{x}_l^{(L)})$ on the basis of (13); this results in the message $m_{sm,j}(\mathbf{x}_l^{(L)})$, conveying a particle-dependent statistical model for the smoothed pdf of $\mathbf{x}_l^{(L)}$. Note that the last message is expected to provide a more refined statistical representation of $\mathbf{x}_l^{(L)}$ than $\vec{m}_{fp,j}(\mathbf{x}_l^{(L)})$ or $\overleftarrow{m}_{be,j}(\mathbf{x}_l^{(L)})$ alone; consequently, it can be profitably exploited in the evaluation of new particle weights. This task is accomplished in the *second part* (see Fig.3-(b)), which mainly concerns the nonlinear state component; in fact, in this part the message[6] $m_{be,j}(\mathbf{x}_l^{(N)}) = m_{sm,j}(\mathbf{x}_l^{(N)})$, conveying a new weight for the particle $\mathbf{x}_{l/(l-1),j}^{(N)}$, is computed on the basis of the available measurements and pseudo-measurements, and of the message $m_{sm,j}(\mathbf{x}_l^{(L)})$. After running the first two parts for all the available particles, the sets of smoothed messages $\{m_{sm,j}(\mathbf{x}_l^{(L)})\}$ and $\{m_{sm,j}(\mathbf{x}_l^{(N)})\}$ are available; in the *third* (and last) *part* of the SPS algorithm, these messages are fused to generate an estimate of the marginal smoothed pdf of $\mathbf{x}_l$. From Figs. 3-(a) and 3-(b) it can be also inferred that the considered recursion is fed not only by the forward messages $\vec{m}_{fp,j}(\mathbf{x}_l^{(N)})$ (17) and $\vec{m}_{fp,j}(\mathbf{x}_l^{(L)})$ (18), but also by the *particle-independent backward* messages

$$\overleftarrow{m}_{be}\left(\mathbf{x}_{l+1}^{(N)}\right) \triangleq \delta\left(\mathbf{x}_{l+1}^{(N)} - \mathbf{x}_{be,l+1}^{(N)}\right) \tag{19}$$

and

$$\overleftarrow{m}_{be}\left(\mathbf{x}_{l+1}^{(L)}\right) \triangleq \mathcal{N}\left(\mathbf{x}_{l+1}^{(L)}; \eta_{be,l+1}^{(L)}, \mathbf{C}_{be,l+1}^{(L)}\right). \tag{20}$$

These messages have been generated in the previous (i.e., the $(T-l-1)$-th) recursion and provide statistical information about the backward estimates of $\mathbf{x}_{l+1}^{(N)}$ and $\mathbf{x}_{l+1}^{(L)}$, respectively. More specifically, as explained in detail below, the messages $\overleftarrow{m}_{be}(\mathbf{x}_{l+1}^{(N)})$ (19) and $\overleftarrow{m}_{be}(\mathbf{x}_{l+1}^{(L)})$ (20) convey the *backward estimate* $\mathbf{x}_{be,l+1}^{(N)}$ (i.e., a single particle representation) of $\mathbf{x}_{l+1}^{(N)}$ and a (simplified) *backward statistical representation* of $\mathbf{x}_{l+1}^{(L)}$,

[4]Note that TF can be employed in place of MPF in the forward pass of RBSS. However, our computer simulations have evidenced that, in the presence of strong measurement and/or process noise (like in the scenarios considered in Section V), this choice does not provide any performance improvement with respect to MPF.

[5]Note that, similarly to MPF, most of the processing tasks which SPS consists of can be formulated with reference to a single particle; this explains why the notation adopted for most of the messages appearing in Fig. 3 includes the subscript j, that represents the index of the particle (namely, the particle $\mathbf{x}_{l/(l-1),j}^{(N)}$) representing $\mathbf{x}_l^{(N)}$ within the considered recursion.

[6]The backward and the smoothed estimates coincide in this case since all the message $\{\vec{m}_{fp,j}(\mathbf{x}_l^{(N)})\}$ convey a unit weight (see (17)).

respectively. Consequently, in the $(T-l)$-th recursion of the SPS algorithm, the messages $\overleftarrow{m}_{be}(\mathbf{x}_l^{(N)})$ and $\overleftarrow{m}_{be}(\mathbf{x}_l^{(L)})$ must be also computed (see Fig. 3-(b)), so that they become available to the next backward recursion; in practice, they are generated merging the statistical information conveyed by the sets $\{m_{be,j}(\mathbf{x}_l^{(L)})\}$ and $\{m_{be,j}(\mathbf{x}_l^{(N)})\}$, respectively.

Let us analyse now the three processing steps accomplished in the *first part* of the SPS technique (see Fig. 3-(a)). The *first step* aims at evaluating the Gaussian message

$$\overleftarrow{m}_{1,j}(\mathbf{x}_l^{(L)}) = \mathcal{N}(\mathbf{x}_l^{(L)}; \eta_{1,l,j}^{(L)}, \mathbf{C}_{1,l,j}^{(L)}), \tag{21}$$

representing a *one-step backward prediction* of the pdf of $\mathbf{x}_l^{(L)}$, evaluated on the basis of $\vec{m}_{fp,j}(\mathbf{x}_l^{(N)})$ (17) and $\overleftarrow{m}_{be}(\mathbf{x}_{l+1}^{(L)})$ (20). Note that, given $\mathbf{x}_l^{(N)} = \mathbf{x}_{l/(l-1),j}^{(N)}$, the random variable $\mathbf{x}_{l+1}^{(L)}$ is generated by *adding* the output $\mathbf{A}_{l,j}^{(L)}\mathbf{x}_l^{(L)}$ of a *matrix multiplication* to the random vector $\mathbf{f}_{l,j}^{(L)} + \mathbf{w}_l^{(L)}$ (see (1)), where $\mathbf{A}_{l,j}^{(L)} \triangleq \mathbf{A}_l^{(L)}(\mathbf{x}_{l/(l-1),j}^{(N)})$ and $\mathbf{f}_{l,j}^{(L)} \triangleq \mathbf{f}_l^{(L)}(\mathbf{x}_{l/(l-1),j}^{(N)})$. Consequently, the precision matrix $\mathbf{W}_{1,l,j}^{(L)}$ and the transformed mean vector $\mathbf{w}_{1,l,j}^{(L)}$ associated with the covariance matrix $\mathbf{C}_{1,l,j}^{(L)}$ and the mean vector $\eta_{1,l,j}^{(L)}$, respectively, can be easily computed by applying eqs. (IV.6)-(IV.8) of [23, Table 4, p.1304] in their backward form (with $A \rightarrow \mathbf{I}_{D_L}$, $X \rightarrow \mathbf{A}_{l,j}^{(L)}\mathbf{x}_l^{(L)}$, $Z \rightarrow \mathbf{x}_{l+1}^{(L)}$ and $Y \rightarrow \mathbf{f}_{l,j}^{(L)} + \mathbf{w}_l^{(L)}$) and, then, eqs. (III.5)-(III.6) of [23, Table 3, p.1304] (with $A \rightarrow \mathbf{A}_{l,j}^{(L)}$, $X \rightarrow \mathbf{x}_l^{(L)}$ and $Y \rightarrow \mathbf{A}_{l,j}^{(L)}\mathbf{x}_l^{(L)}$); this results in

$$\mathbf{W}_{1,l,j}^{(L)} = \left(\mathbf{A}_{l,j}^{(L)}\right)^T \mathbf{P}_{l+1}\mathbf{W}_{be,l+1}^{(L)}\mathbf{A}_{l,j}^{(L)}, \tag{22}$$

and

$$\mathbf{w}_{1,l,j}^{(L)} = \left(\mathbf{A}_{l,j}^{(L)}\right)^T \left[\mathbf{P}_{l+1}\mathbf{w}_{be,l+1}^{(L)} - \mathbf{W}_{be,l+1}^{(L)}\mathbf{H}_{l+1}\mathbf{W}_w^{(L)}\mathbf{f}_{l,j}^{(L)}\right], \tag{23}$$

where $\mathbf{W}_{be,l+1}^{(L)} \triangleq (\mathbf{C}_{be,l+1}^{(L)})^{-1}$, $\mathbf{P}_{l+1} \triangleq \mathbf{I}_{D_L} - \mathbf{W}_{be,l+1}^{(L)}\mathbf{H}_{l+1}$, $\mathbf{H}_{l+1} \triangleq (\mathbf{W}_w^{(L)} + \mathbf{W}_{be,l+1}^{(L)})^{-1}$, $\mathbf{W}_w^{(L)} \triangleq (\mathbf{C}_w^{(L)})^{-1}$ and $\mathbf{w}_{be,l+1}^{(L)} \triangleq \mathbf{W}_{be,l+1}^{(L)}\eta_{be,l+1}^{(L)}$. The *second step* can be seen as a MU for the linear state component. In fact, it aims at generating the backward estimate $\overleftarrow{m}_{be,j}(\mathbf{x}_l^{(L)})$ by merging the backward prediction $\overleftarrow{m}_{1,j}(\mathbf{x}_l^{(L)})$ (21) with the Gaussian messages $\overleftarrow{m}_{2,j}(\mathbf{x}_l^{(L)})$ and $\overleftarrow{m}_{4,j}(\mathbf{x}_l^{(L)})$, that convey the available statistical information about the *measurement* $\mathbf{y}_l$ and the *pseudo-measurement* $\mathbf{z}_l^{(L)}$, respectively, given $\mathbf{x}_l^{(N)} = \mathbf{x}_{l/(l-1),j}^{(N)}$ and $\mathbf{x}_{l+1}^{(N)} = \mathbf{x}_{be,l+1}^{(N)}$. Since

$$\vec{m}_j\left(\mathbf{z}_l^{(L)}\right) = f\left(\mathbf{z}_l^{(L)} \left| \mathbf{x}_{l/(l-1),j}^{(N)}, \mathbf{x}_{be,l+1}^{(N)} \right.\right) = \delta\left(\mathbf{z}_l^{(L)} - \mathbf{z}_{l,j}^{(L)}\right), \tag{24}$$

where

$$\mathbf{z}_{l,j}^{(L)} \triangleq \mathbf{x}_{be,l+1}^{(N)} - \mathbf{f}_{l,j}^{(N)} \tag{25}$$

and $\mathbf{f}_{l,j}^{(N)} \triangleq \mathbf{f}_l^{(N)}(\mathbf{x}_{l/(l-1),j}^{(N)})$, the message $\overleftarrow{m}_{2,j}(\mathbf{x}_l^{(L)})$ can be easily evaluated as

$$\begin{aligned}\overleftarrow{m}_{2,j}\left(\mathbf{x}_l^{(L)}\right) &= \int\int f\left(\mathbf{z}_l^{(L)} \left| \mathbf{x}_l^{(L)}, \mathbf{x}_l^{(N)} \right.\right) \\ &\quad \cdot \overleftarrow{m}_j\left(\mathbf{z}_l^{(L)}\right) \vec{m}_{fp,j}\left(\mathbf{x}_l^{(N)}\right) d\mathbf{x}_l^{(N)} d\mathbf{z}_l^{(L)} \\ &= \mathcal{N}\left(\mathbf{z}_{l,j}^{(L)}; \mathbf{A}_{l,j}^{(N)}\mathbf{x}_l^{(L)}, \mathbf{C}_w^{(N)}\right).\end{aligned} \tag{26}$$

Moreover, this message can be put in the equivalent Gaussian form

$$\overleftarrow{m}_{2,j}\left(\mathbf{x}_l^{(L)}\right) = \mathcal{N}\left(\mathbf{z}_{l,j}^{(L)}; \eta_{2,l,j}^{(L)}, \mathbf{C}_{2,l,j}^{(L)}\right), \tag{27}$$

where the precision matrix $\mathbf{W}_{2,l,j}^{(L)}$ and the transformed mean vector $\mathbf{w}_{2,l,j}^{(L)}$ associated with $\mathbf{C}_{2,l,j}^{(L)}$ and $\eta_{2,l,j}^{(L)}$, respectively, are given by

$$\mathbf{W}_{2,l,j}^{(L)} = \left(\mathbf{A}_{l,j}^{(N)}\right)^T \mathbf{W}_w^{(N)}\mathbf{A}_{l,j}^{(N)}, \tag{28}$$

and

$$\mathbf{w}_{2,l,j}^{(L)} = \left(\mathbf{A}_{l,j}^{(N)}\right)^T \mathbf{W}_w^{(N)}\mathbf{z}_{l,j}^{(L)}, \tag{29}$$

respectively; here, $\mathbf{A}_{l,j}^{(N)} \triangleq \mathbf{A}_l^{(N)}(\mathbf{x}_{l/(l-1),j}^{(N)})$ and $\mathbf{W}_w^{(N)} \triangleq (\mathbf{C}_w^{(N)})^{-1}$. Similarly, the message $\overleftarrow{m}_{4,j}(\mathbf{x}_l^{(L)})$ is given by

$$\begin{aligned}\overleftarrow{m}_{4,j}\left(\mathbf{x}_l^{(L)}\right) &= \int f\left(\mathbf{y}_l \left| \mathbf{x}_l^{(N)}, \mathbf{x}_l^{(L)} \right.\right) \vec{m}_{fp,j}\left(\mathbf{x}_l^{(N)}\right) d\mathbf{x}_l^{(N)} \\ &= \mathcal{N}\left(\mathbf{y}_l; \mathbf{B}_{l,j}\,\mathbf{x}_l^{(L)} + \mathbf{h}_{l,j}, \mathbf{C}_e\right),\end{aligned} \tag{30}$$

where $\mathbf{B}_{l,j} \triangleq \mathbf{B}_l(\mathbf{x}_{l/(l-1),j}^{(N)})$ and $\mathbf{h}_{l,j} \triangleq \mathbf{h}_l(\mathbf{x}_{l/(l-1),j}^{(N)})$; this message can be also put in the equivalent Gaussian form

$$\overleftarrow{m}_{4,j}\left(\mathbf{x}_l^{(L)}\right) = \mathcal{N}\left(\mathbf{x}_l^{(L)}; \eta_{4,l,j}^{(L)}, \mathbf{C}_{4,l,j}^{(L)}\right), \tag{31}$$

where the precision matrix $\mathbf{W}_{4,l,j}^{(L)}$ and the transformed mean vector $\mathbf{w}_{4,l,j}^{(L)}$ associated with $\mathbf{C}_{4,l,j}^{(L)}$ and $\eta_{4,l,j}^{(L)}$, respectively, are evaluated as

$$\mathbf{W}_{4,l,j}^{(L)} = (\mathbf{B}_{l,j})^T \mathbf{W}_e \mathbf{B}_{l,j} \tag{32}$$

and

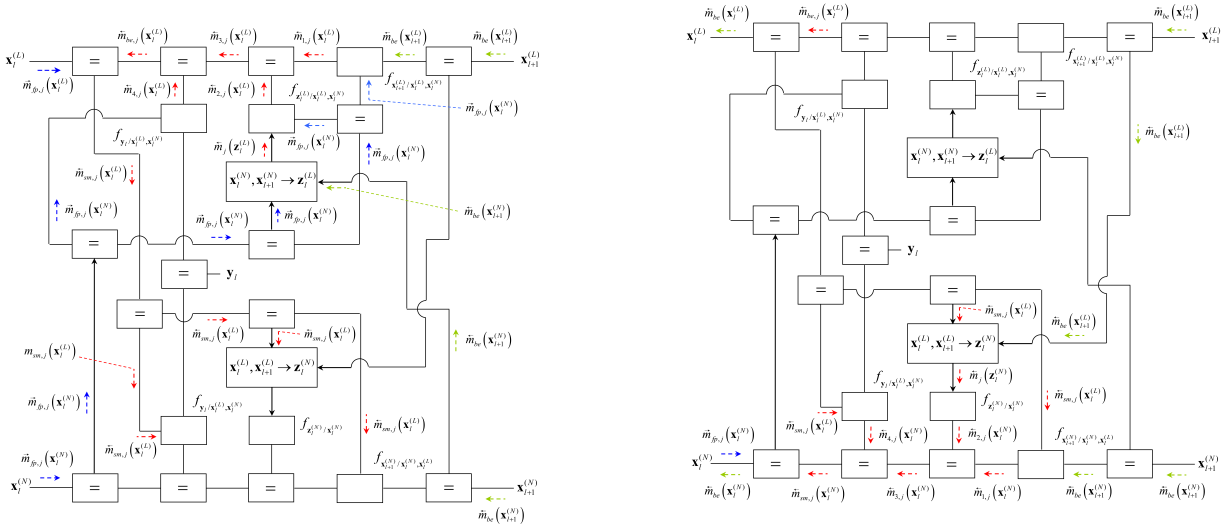
$$\mathbf{w}_{4,l,j}^{(L)} = (\mathbf{B}_{l,j})^T \mathbf{W}_e\,(\mathbf{y}_l - \mathbf{h}_{l,j}), \tag{33}$$

respectively, and $\mathbf{W}_e \triangleq \mathbf{C}_e^{-1}$. Given $\overleftarrow{m}_{1,j}(\mathbf{x}_l^{(L)})$ (21), $\overleftarrow{m}_{2,j}(\mathbf{x}_l^{(L)})$ (27) and $\overleftarrow{m}_{4,j}(\mathbf{x}_l^{(L)})$ (31), the message

$$\overleftarrow{m}_{be,j}\left(\mathbf{x}_l^{(L)}\right) = \mathcal{N}\left(\mathbf{x}_l^{(L)}; \eta_{be,l,j}^{(L)}, \mathbf{C}_{be,l,j}^{(L)}\right) \tag{34}$$

can be computed by applying twice eqs. (II.2) and (II.4) of [23, Table 2, p.1303] (referring to the computation of the backward messages emerging from an equality node). This produces the precision matrix

$$\begin{aligned}\mathbf{W}_{be,l,j}^{(L)} &\triangleq \mathbf{W}_{1,l,j}^{(L)} + \mathbf{W}_{2,l,j}^{(L)} + \mathbf{W}_{4,l,j}^{(L)} \\ &= \left(\mathbf{A}_{l,j}^{(L)}\right)^T \mathbf{P}_{l+1}\mathbf{W}_{be,l+1}^{(L)}\mathbf{A}_{l,j}^{(L)} \\ &\quad + (\mathbf{B}_{l,j})^T \mathbf{W}_e \mathbf{B}_{l,j} \\ &\quad + \left(\mathbf{A}_{l,j}^{(N)}\right)^T \mathbf{W}_w^{(N)}\mathbf{A}_{l,j}^{(N)}\end{aligned} \tag{35}$$



(a) (b)

Fig. 3: Representation of the message scheduling employed in the first part (a) and in the second part (b) of the $(T-l)$-th recursion of SPS backward processing. Blue, green and red arrows are employed to identify the input forward messages, the input/output backward messages and the remaining messages, respectively.

and the transformed mean vector

$$\begin{aligned}\mathbf{w}_{be,l,j}^{(L)} &\triangleq \mathbf{w}_{1,l,j}^{(L)} + \mathbf{w}_{2,l,j}^{(L)} + \mathbf{w}_{4,l,j}^{(L)} \\ &= \left(\mathbf{A}_{l,j}^{(L)}\right)^T \left[\mathbf{P}_{l+1}\mathbf{w}_{be,l+1}^{(L)} \right. \\ &\quad \left. -\mathbf{W}_{be,l+1}^{(L)}\mathbf{H}_{l+1}\mathbf{W}_w^{(L)}\mathbf{f}_{l,j}^{(L)}\right] \\ &\quad + \left(\mathbf{A}_{l,j}^{(N)}\right)^T \mathbf{W}_w^{(N)}\mathbf{z}_{l,j}^{(L)} \\ &\quad + \left(\mathbf{B}_{l,j}\right)^T \mathbf{W}_e\left(\mathbf{y}_l - \mathbf{h}_{l,j}\right), \end{aligned} \tag{36}$$

associated with $\mathbf{C}_{be,l,j}^{(L)}$ and $\eta_{be,l,j}^{(L)}$, respectively. In the *third* (and last) *step*, the messages $\vec{m}_{fp,j}(\mathbf{x}_l^{(L)})$ and $\overleftarrow{m}_{be,j}(\mathbf{x}_l^{(L)})$ are merged (see eq. (13)) to generate the Gaussian message

$$m_{sm,j}\left(\mathbf{x}_l^{(L)}\right) = \mathcal{N}\left(\mathbf{x}_l^{(L)}; \eta_{sm,l,j}^{(L)}, \mathbf{C}_{sm,l,j}^{(L)}\right), \tag{37}$$

that conveys the *smoothed* information for the linear component associated with the j-th particle. In this case, applying again eqs. (II.2) and (II.4) of [23, Table 2, p.1303] produces the precision matrix[7]

$$\mathbf{W}_{sm,l,j}^{(L)} \triangleq \left(\mathbf{C}_{sm,l,j}^{(L)}\right)^{-1} = \mathbf{W}_{fp,l,j}^{(L)} + \mathbf{W}_{be,l,j}^{(L)}, \tag{38}$$

and the transformed mean vector

$$\mathbf{w}_{sm,l,j}^{(L)} \triangleq \mathbf{W}_{sm,l,j}^{(L)}\eta_{sm,l,j}^{(L)} = \mathbf{w}_{fp,l,j}^{(L)} + \mathbf{w}_{be,l,j}^{(L)}, \tag{39}$$

associated with $\mathbf{C}_{sm,l,j}^{(L)}$ and $\eta_{sm,l,j}^{(L)}$, respectively (here, $\mathbf{W}_{fp,l,j}^{(L)} \triangleq (\mathbf{C}_{fp,l,j}^{(L)})^{-1}$ and $\mathbf{w}_{fp,l,j}^{(L)} \triangleq \mathbf{W}_{fp,l,j}^{(L)}\eta_{fp,l,j}^{(L)}$). Finally, the message $m_{sm,j}(\mathbf{x}_l^{(L)})$ (37) is passed to the second part of the SPS algorithm, since it represents the final estimate of the pdf of $\mathbf{x}_l^{(L)}$ associated with the j-th particle.

[7]Note that $m_{sm,j}(\mathbf{x}_1^{(L)}) = \overleftarrow{m}_{be,j}(\mathbf{x}_1^{(L)})$ and $m_{sm,j}(\mathbf{x}_T^{(L)}) = \vec{m}_{fp,j}(\mathbf{x}_T^{(L)})$ should be assumed, since at the instant $l=1$ ($l=T$) only a backward estimate (a forward prediction) is available for $\mathbf{x}_l^{(L)}$.

The processing accomplished in the *second part* of the SPS technique can be divided in two steps (see Fig. 3-(b)). In the *first step*, a new weight is computed for the j-th particle on the basis of the following information: a) the backward estimate $\mathbf{x}_{be,l+1}^{(N)}$ of $\mathbf{x}_{l+1}^{(N)}$ (see $\overleftarrow{m}_{be}(\mathbf{x}_{l+1}^{(N)})$ (19); b) the pseudo-measurement $\mathbf{z}_l^{(N)}$; c) the measurement $\mathbf{y}_l$. For this reason, the new overall weight $W_{l,j}$ for the j-th particle is expressed as a product of three factors (see the lower part of the FG shown in Fig. 3-(b)), i.e. as

$$W_{l,j} = w_{1,l,j} \cdot w_{2,l,j} \cdot w_{4,l,j}, \tag{40}$$

where the weights $w_{1,l,j}$, $w_{2,l,j}$ and $w_{4,l,j}$ are related to $\mathbf{x}_{be,l+1}^{(N)}$, $\mathbf{z}_l^{(N)}$ and $\mathbf{y}_l$, respectively, and represent the messages $\overleftarrow{m}_{1,j}(\mathbf{x}_l^{(N)})$, $\overleftarrow{m}_{2,j}(\mathbf{x}_l^{(N)})$ and $\overleftarrow{m}_{4,j}(\mathbf{x}_l^{(N)})$, respectively. In practice, the weight $w_{1,l,j}$ is evaluated as

$$\begin{aligned} w_{1,l,j} &= \int\int f\left(\mathbf{x}_{l+1}^{(N)} \middle| \mathbf{x}_l^{(L)}, \mathbf{x}_{l/(l-1),j}^{(N)}\right) \\ &\quad \cdot \overleftarrow{m}_{be}\left(\mathbf{x}_{l+1}^{(N)}\right) m_{sm,j}\left(\mathbf{x}_l^{(L)}\right) d\mathbf{x}_l^{(L)} d\mathbf{x}_{l+1}^{(N)} \\ &= \mathcal{N}\left(\mathbf{x}_{be,l+1}^{(N)}; \eta_{1,l,j}^{(N)}, \mathbf{C}_{1,l,j}^{(N)}\right) = \\ &= K_{1,l,j} \exp\left(-\frac{1}{2}Z_{1,l,j}\right) \end{aligned} \tag{41}$$

where $K_{1,l,j} = (2\pi \det(\mathbf{C}_{1,l,j}^{(N)}))^{-D_N/2}$, $Z_{1,l,j} = \left\|\mathbf{x}_{be,l+1}^{(N)} - \eta_{1,l,j}^{(N)}\right\|_{\mathbf{W}_{1,l,j}^{(N)}}^2$,

$$\eta_{1,l,j}^{(N)} \triangleq \mathbf{A}_{l,j}^{(N)}\eta_{sm,l,j}^{(N)} + \mathbf{f}_{l,j}^{(N)}, \tag{42}$$

$$\mathbf{C}_{1,l,j}^{(N)} \triangleq \mathbf{A}_{l,j}^{(N)}\mathbf{C}_{sm,l,j}^{(N)}\left(\mathbf{A}_{l,j}^{(N)}\right)^T + \mathbf{C}_w^{(N)} \tag{43}$$

and $\|\mathbf{x}\|_{\mathbf{W}}^2 \triangleq \mathbf{x}^T\mathbf{W}\mathbf{x}$ denotes the square of the norm of the vector $\mathbf{x}$ with respect to the positive definite matrix $\mathbf{W}$.

The computation of the weight $w_{2,l,j}$ requires the knowledge of the Gaussian message

$$\overleftarrow{m}_j\left(\mathbf{z}_l^{(N)}\right) = \mathcal{N}\left(\mathbf{x}_l^{(N)}; \eta_{\mathbf{z},l,j}^{(N)}, \mathbf{C}_{\mathbf{z},l,j}^{(N)}\right), \tag{44}$$

that expresses the statistical knowledge acquired about $\mathbf{z}_l^{(N)}$ (16) on the basis of the messages $\overleftarrow{m}_{be}(\mathbf{x}_{l+1}^{(L)})$ (20) and $m_{sm,j}(\mathbf{x}_l^{(L)})$ (37); the mean vector and the covariance matrix of the message $\overleftarrow{m}_j(\mathbf{z}_l^{(N)})$ (44) are evaluated as (see [21, p. 4, eqs. (35)-(36)])

$$\eta_{\mathbf{z},l,j}^{(N)} \triangleq \eta_{be,l+1}^{(L)} - \mathbf{A}_{l,j}^{(L)} \eta_{sm,l,j}^{(L)}, \tag{45}$$

and

$$\mathbf{C}_{\mathbf{z},l,j}^{(N)} \triangleq \mathbf{C}_{be,l+1}^{(L)} - \mathbf{A}_{l,j}^{(L)} \mathbf{C}_{sm,l,j}^{(L)} \left(\mathbf{A}_{l,j}^{(L)}\right)^T, \tag{46}$$

respectively. Given the message $\overleftarrow{m}_j(\mathbf{z}_l^{(N)})$ (44), $w_{2,l,j}$ is computed by *correlating* it with the pdf $f(\mathbf{z}_l^{(N)}|\mathbf{x}_{l/(l-1),j}^{(N)})$ (expressing our prior knowledge about $\mathbf{z}_l^{(N)}$ for the considered particle), i.e. as

$$w_{2,l,j} = \int \overleftarrow{m}_j\left(\mathbf{z}_l^{(N)}\right) f\left(\mathbf{z}_l^{(N)} \left| \mathbf{x}_{l/(l-1),j}^{(N)} \right.\right) d\mathbf{z}_l^{(N)}. \tag{47}$$

Then, substituting $\overleftarrow{m}_j(\mathbf{z}_l^{(N)})$ (44) in (47) yields, after some manipulation,

$$w_{2,l,j} = K_{2,l,j} \exp\left(-\frac{1}{2} Z_{2,l,j}\right), \tag{48}$$

where

$$Z_{2,l,j} \triangleq \left\|\eta_{\mathbf{z},l,j}^{(N)}\right\|^2_{\mathbf{W}_{2,l,j}^{(N)}} + \left\|\mathbf{f}_{l,j}^{(L)}\right\|^2_{\mathbf{W}_w^{(L)}} - \left\|\eta_{2,l,j}^{(N)}\right\|^2_{\mathbf{W}_{2,l,j}^{(N)}} \tag{49}$$

$$K_{2,l,j} = (2\pi \det(\mathbf{C}_{\mathbf{z},l,j}^{(N)} + \mathbf{C}_w^{(L)}))^{-D_L/2},$$

$$\mathbf{W}_{2,l,j}^{(N)} = \mathbf{W}_{\mathbf{z},l,j}^{(N)} + \mathbf{W}_w^{(L)}, \tag{50}$$

$$\mathbf{w}_{2,l,j}^{(N)} = \mathbf{w}_{\mathbf{z},l,j}^{(N)} + \mathbf{W}_w^{(L)} \mathbf{f}_{l,j}^{(L)}, \tag{51}$$

$\mathbf{W}_{\mathbf{z},l,j}^{(N)} \triangleq (\mathbf{C}_{\mathbf{z},l,j}^{(N)})^{-1}$ and $\mathbf{w}_{\mathbf{z},l,j}^{(N)} \triangleq \mathbf{W}_{\mathbf{z},l,j}^{(N)} \eta_{\mathbf{z},l,j}^{(N)}$.

Finally, the weight $w_{4,l,j}$ is evaluated as

$$\begin{aligned} w_{4,l,j} &= \int f\left(\mathbf{y}_l \left| \mathbf{x}_{l/(l-1),j}^{(N)}, \mathbf{x}_l^{(L)} \right.\right) m_{sm,j}\left(\mathbf{x}_l^{(L)}\right) d\mathbf{x}_l^{(L)} \\ &= \mathcal{N}\left(\mathbf{y}_l; \eta_{4,l,j}^{(N)}, \mathbf{C}_{4,l,j}^{(N)}\right) = K_{4,l,j} \exp\left(-\frac{1}{2} Z_{4,l,j}\right), \end{aligned} \tag{52}$$

where $K_{4,l,j} \triangleq (2\pi \det(\mathbf{C}_{4,l,j}^{(N)}))^{-P/2}$, $Z_{4,l,j} \triangleq \|\mathbf{y}_l - \eta_{4,l,j}^{(N)}\|^2_{\mathbf{W}_{4,l,j}^{(N)}}$,

$$\eta_{4,l,j}^{(N)} \triangleq \mathbf{B}_{l,j} \eta_{sm,l,j}^{(L)} + \mathbf{h}_{l,j}, \tag{53}$$

$$\mathbf{C}_{4,l,j}^{(N)} \triangleq \mathbf{B}_{l,j} \mathbf{C}_{sm,l,j}^{(L)} (\mathbf{B}_{l,j})^T + \mathbf{C}_e \tag{54}$$

and $\mathbf{W}_{4,l,j}^{(N)} \triangleq (\mathbf{C}_{4,l,j}^{(N)})^{-1}$. Substituting the *right-hand side* (RHS) of (41), (48) and (52) in (40) yields[8]

$$W_{l,j} = K_{l,j} \exp\left(-\frac{1}{2} Z_{l,j}\right), \tag{55}$$

where

$$K_{l,j} \triangleq K_{1,l,j} \cdot K_{2,l,j} \cdot K_{4,l,j} \tag{56}$$

and

$$\begin{aligned} Z_{l,j} &\triangleq Z_{1,l,j} + Z_{2,l,j} + Z_{4,l,j} = \\ &= \left\|\mathbf{x}_{be,l+1}^{(N)} - \eta_{1,l,j}^{(N)}\right\|^2_{\mathbf{W}_{1,l,j}^{(N)}} + \left\|\mathbf{y}_l - \eta_{4,l,j}^{(N)}\right\|^2_{\mathbf{W}_{4,l,j}^{(N)}} \\ &\quad + \left\|\eta_{\mathbf{z},l,j}^{(N)}\right\|^2_{\mathbf{W}_{2,l,j}^{(N)}} + \left\|\mathbf{f}_{l,j}^{(L)}\right\|^2_{\mathbf{W}_w^{(L)}} - \left\|\eta_{2,l,j}^{(N)}\right\|^2_{\mathbf{W}_{2,l,j}^{(N)}}. \end{aligned} \tag{57}$$

It worth noting that: a) for a given SSM and under specific operating conditions, the three weights $w_{1,l,j}$, $w_{2,l,j}$ and $w_{4,l,j}$ may not have the same importance; b) the weight $w_{4,l,j}$ can be seen as the backward counterpart of the weight $w_{l/l,j}$ evaluated in FF. For these reasons, in some cases, the computational load due to the evaluation of the overall weight $W_{l,j}$ (40) might be substantially reduced by discarding one of the weigths $\{w_{p,l,j}, p = 1, 2, 4\}$ and/or adopting the approximation $w_{4,l,j} \simeq w_{l/l,j}$ with a limited impact on estimation accuracy.

Once the whole set of weights $\{W_{l,j}\}$ is available, its normalization must be accomplished; this produces the final (normalised) weight

$$W_{sm,l,j} \triangleq W_{l,j} / \sum_{j=0}^{N_p-1} W_{l,j}, \tag{58}$$

which is conveyed by the messages (see Fig. 3-(b))

$$m_{be,j}\left(\mathbf{x}_l^{(N)}\right) = m_{sm,j}\left(\mathbf{x}_l^{(N)}\right) = W_{sm,l,j}\, \delta\left(\mathbf{x}_l^{(N)} - \mathbf{x}_{l/(l-1),j}^{(N)}\right) \tag{59}$$

for $j = 0, 1, , ..., N_p - 1$.

In the *second step* the input messages

$$\overleftarrow{m}_{be}\left(\mathbf{x}_l^{(N)}\right) = \delta\left(\mathbf{x}_l^{(N)} - \mathbf{x}_{be,l}^{(N)}\right) \tag{60}$$

and

$$\overleftarrow{m}_{be}\left(\mathbf{x}_l^{(L)}\right) = \mathcal{N}\left(\mathbf{x}_l^{(L)}; \eta_{be,l}^{(L)}, \mathbf{C}_{be,l}^{(L)}\right) \tag{61}$$

for the next recursion are generated on the basis of the particle weights $\{W_{sm,l,j}\}$ (see (58)) and the particle-dependent Gaussian messages $\{\overleftarrow{m}_{be,j}(\mathbf{x}_l^{(L)})\}$ (see (34)-(36)). The vector $\mathbf{x}_{be,l}^{(N)}$, appearing in the RHS of (60), represents the l-th element of the *estimated nonlinear state trajectory* and is evaluated as a weighted sum of the particles $\{\mathbf{x}_{l/(l-1),j}^{(N)}\}$, i.e. as

$$\mathbf{x}_{be,l}^{(N)} \triangleq \sum_{j=0}^{N_p-1} W_{sm,l,j} \mathbf{x}_{l/(l-1),j}^{(N)}. \tag{62}$$

[8]Our numerical results have evidenced that ignoring the factor $K_{l,j}$ in the evaluation of the weights $\{W_{l,j}\}$ has a negligible impact on the performance of the SPS algorithm for the SSMs considered in Section V. For, this reason, this factor has been always set to one in our computer simulations.

The evaluation of the mean vector $\eta_{be,l}^{(L)}$ and $\mathbf{C}_{be,l}^{(L)}$, appearing in the RHS of (61), is based on the observation that: a) the statistical presentation of $\mathbf{x}_l^{(L)}$ based on the sets $\{W_{sm,l,j}\}$ (see (58)) and $\{\overleftarrow{m}_{be,j}(\mathbf{x}_l^{(L)})\}$ (see (34)) is expressed by a N_p-component order GM; b) any GM can be easily condensed in a single Gaussian pdf by means of a simple transformation preserving both the mean and the covariance matrix of the GM itself (e.g., see [28, Sect. 4]). Adopting this transformation leads us to evaluating $\eta_{be,l}^{(L)}$ (which represents the l-the element of the *estimated linear state trajectory*) and $\mathbf{C}_{be,l}^{(L)}$ as

$$\eta_{be,l}^{(L)} \triangleq \sum_{j=0}^{N_p-1} W_{sm,l,j}\, \eta_{be,l,j}^{(L)} \tag{63}$$

and

$$\begin{aligned}\mathbf{C}_{be,l}^{(L)} \triangleq & \sum_{j=0}^{N_p-1} W_{sm,l,j}\, \mathbf{C}_{be,l,j}^{(L)} \\ & + \sum_{j=0}^{N_p-1} W_{sm,l,j} \left(\eta_{be,l,j}^{(L)} - \eta_{be,l}^{(L)}\right)\left(\eta_{be,l,j}^{(L)} - \eta_{be,l}^{(L)}\right)^T,\end{aligned} \tag{64}$$

respectively.

The *third* (and final) *part* of the SPS algorithm aims at generating its output, i.e. at computing an approximation of the marginal smoothed pdfs of $\mathbf{x}_l$; this approximation is expressed by

$$\hat{f}\left(\mathbf{x}_l|\mathbf{y}_{1:N}\right) \triangleq \sum_{j=0}^{N_p-1} m_{sm,j}\left(\mathbf{x}_l^{(N)}\right)\, m_{sm,j}\left(\mathbf{x}_l^{(L)}\right), \tag{65}$$

where $m_{sm,j}(\mathbf{x}_l^{(L)})$ and $m_{sm,j}(\mathbf{x}_l^{(N)})$ are given by (37) and (59), respectively; note that this formula has the typical structure of Rao-Blackwellized approximations (e.g., see eq. (11) in [20, p. 355, second column]).

After completing the evaluation of $\hat{f}\left(\mathbf{x}_l|\mathbf{y}_{1:N}\right)$ (65), the $(T-l)$-th recursion of the SPS technique is over. Then, the recursion index l is decreased by one; if it equals zero, the backward pass is over, otherwise a new recursion starts. It is important to point out that the first recursion of the backward pass requires the knowledge of the input messages $\overleftarrow{m}_{be}(\mathbf{x}_T^{(N)})$ and $\overleftarrow{m}_{be}(\mathbf{x}_T^{(L)})$. Like in any particle smoothing method employing BIF, the evaluation of these messages is based on the statistical information generated in the last recursion of FF. In particular, the above mentioned messages are still expressed by (60) and (61) (with $l=T$). However, in this case, we have that $\mathbf{x}_{be,T}^{(N)}$ is computed according to (62) (with $l=T$), but using the FF weights $\{w_{T/T,j}\}$ in place of the smoothed weights $\{W_{sm,l,j}\}$; similarly, the parameters $\eta_{be,T}^{(L)}$ and $\mathbf{C}_{be,T}^{(L)}$ appearing in (61) are evaluated on the basis of formulas (63) and (64), but employing, in place of the Gaussian messages $\{\overleftarrow{m}_{be,j}(\mathbf{x}_l^{(N)})\}$ (see (40)), the messages $\{\mathcal{N}(\mathbf{x}_T^{(L)}; \eta_{fe,T,j}^{(L)}, \mathbf{C}_{fe,T,j}^{(L)})\}$ generated by the MU for the linear state component in the last (i.e., in the T-th) recursion of FF.

The complete SPS technique is given in Algorithm 1.

Algorithm 1: Serial particle smoothing

1 **Forward filtering**: Run the MPF algorithm for time $l=1,2,...,T$. For each l, store $\{\mathbf{x}_{l/(l-1),j}^{(N)}, \eta_{fp,l,j}^{(L)}, \mathbf{C}_{fp,l,j}^{(L)}\}_{j=1}^{N_p}$; for $l=T$ store $\{w_{T/T,j}, \eta_{fe,T,j}^{(L)}, \mathbf{C}_{fe,T,j}^{(L)}\}_{j=1}^{N_p}$.

2 **Initialisation of backward filtering**: compute $\mathbf{x}_{be,T}^{(N)}$, $\eta_{be,T}^{(L)}$ and $\mathbf{C}_{be,T}^{(L)}$ according to (62), (63) and (64) (with $W_{sm,l,j}=w_{T/T,j}$, $\eta_{be,T,j}^{(L)}=\eta_{fe,T,j}^{(L)}$ and $\mathbf{C}_{be,T,j}^{(L)}=\mathbf{C}_{fe,T,j}^{(L)}$ for any j), respectively.

3 **Backward filtering and smoothing**: For $l=T-1$ to 1:

a- For $j=1$ to N_p:

- *Backward filter prediction* (linear component only): compute $\mathbf{W}_{1,l,j}^{(L)}$ and $\mathbf{w}_{1,l,j}^{(L)}$ according to (22) and (23), respectively;
- *Backward measurement update* (linear component only): compute $\mathbf{W}_{be,l,j}^{(L)}$ and $\mathbf{w}_{be,l,j}^{(L)}$ according to (35) and (36), respectively;
- *Smoothing* (linear state component only): compute $\mathbf{W}_{sm,l,j}^{(L)}$ and $\mathbf{w}_{sm,l,j}^{(L)}$ according to (38) and (39), respectively;
- *Computation of particle weights* (MU for the nonlinear component only):
  - i) compute $\eta_{1,l,j}^{(N)}$ and $\mathbf{C}_{1,l,j}^{(N)}$ according to (42) and (43), respectively;
  - ii) compute $\mathbf{W}_{2,l,j}^{(N)}$ and $\mathbf{w}_{2,l,j}^{(N)}$ according to (50) and (51), respectively;
  - iii) compute $\eta_{4,l,j}^{(N)}$ and $\mathbf{C}_{4,l,j}^{(N)}$ according to (53) and (54), respectively;
  - iv) compute the j-th weight $W_{l,j}$ according to (55)-(57) and store it.

b- *Normalisation of particle weights*: compute the normalised weights $\{W_{sm,l,j}\}$ according to (58) and store them;

c- *Condensation of backward estimation messages*: compute $\mathbf{x}_{be,l}^{(N)}$, $\eta_{be,l}^{(L)}$ and $\mathbf{C}_{be,l}^{(L)}$ according to (62), (63) and (64).

d- *Generation of marginal smoothed densities*: generate an estimate of $f\left(\mathbf{x}_l, |\mathbf{y}_{1:N}\right)$ according to (65).

The SPS algorithm illustrated above deserves some additional comments, that are listed below:

1) The condensation of the sets of messages $\{\overleftarrow{m}_{be,j}(\mathbf{x}_l^{(L)})\}$ (see (34)) and $\{\overleftarrow{m}_{be,j}(\mathbf{x}_l^{(N)})\}$ (see (59)) in the messages $\overleftarrow{m}_{be}(\mathbf{x}_l^{(N)})$ (60) and $\overleftarrow{m}_{be}(\mathbf{x}_l^{(L)})$ (61), respectively, is necessary to significantly mitigate the computational requirements of BIF. In fact, merging the complete statistical information provided by sets $\{\overleftarrow{m}_{be,j}(\mathbf{x}_l^{(L)})\}$ and $\{\overleftarrow{m}_{be,j}(\mathbf{x}_l^{(N)})\}$ with the forward messages representing $\mathbf{x}_{l-1}$ (namely, the messages $\{\overrightarrow{m}_{fp,j}(\mathbf{x}_{l-1}^{(N)})\}$ and $\{\overrightarrow{m}_{fp,j}(\mathbf{x}_{l-1}^{(L)})\}$, respectively) would entail a computational load proportional to N_p^2.
2) Given (65), the approximations for the marginal smoothed pdfs of $\mathbf{x}_l^{(N)}$ and $\mathbf{x}_l^{(L)}$ are expressed by the

particle-based model

$$\hat{f}\left(\mathbf{x}_l^{(N)}|\mathbf{y}_{1:N}\right) \triangleq \sum_{j=0}^{N_p-1} m_{sm,j}\left(\mathbf{x}_l^{(N)}\right) \quad (66)$$

and the N_p-component GM

$$\hat{f}\left(\mathbf{x}_l^{(L)}|\mathbf{y}_{1:N}\right) \triangleq \sum_{j=0}^{N_p-1} W_{sm,l,j}\, m_{sm,j}\left(\mathbf{x}_l^{(L)}\right), \quad (67)$$

respectively, and are both suitable to represent *multimodal* pdfs. From these formulas it is easily inferred that the smoothed estimate of $\mathbf{x}_l^{(N)}$ is given by $\mathbf{x}_{be,l}^{(N)}$ (62), whereas the smoothed estimate $\mathbf{x}_{sm,l}^{(L)}$ of $\mathbf{x}_l^{(L)}$ has to be evaluated as

$$\mathbf{x}_{sm,l}^{(L)} = \sum_{j=0}^{N_p-1} W_{sm,l,j}\, \eta_{sm,l,j}^{(L)}, \quad (68)$$

where the mean vector $\eta_{sm,l,j}^{(L)}$ is computed on the basis of (38) and (39).

Our final comment concerns the smoothing of the linear state component and has been inspired by the considerations illustrated in [20, Par. IV-D], where it is stressed that in Rao-Blackwellized methods the statistics for the linear state component need to be computed *conditionally* on the considered nonlinear state trajectory. As a matter of fact, our SPS algorithm generates a single estimate of the *nonlinear state trajectory* in its backward pass; however, the statistical models for the linear state components associated with this trajectory (see $\overleftarrow{m}_{be}(\mathbf{x}_l^{(L)})$ (61) and $\hat{f}(\mathbf{x}_l^{(L)}|\mathbf{y}_{1:N})$ (67)) do not satisfy the above mentioned condition (since they do not really refer to a specific nonlinear state trajectory). This suggests that, once the SPS algorithm has been carried out, more refined statistics for the linear state component can be computed by accomplishing *Kalman smoothing* (KS) on the linear state component, under the assumption that the nonlinear state component is *known* over the entire observation interval. In practice, this requires:

1) Carrying out a new forward pass followed by a new backward pass; the first (second) pass produces a *single* Gaussian message $\vec{m}_{fp}(\mathbf{x}_l^{(L)}) \triangleq \mathcal{N}(\mathbf{x}_l^{(L)}; \eta_{fp,l}^{(L)}, \mathbf{C}_{fp,l}^{(L)})$ ($\overleftarrow{m}_{be}(\mathbf{x}_l^{(L)}) \triangleq \mathcal{N}(\mathbf{x}_l^{(L)}; \eta_{be,l}^{(L)}, \mathbf{C}_{be,l}^{(L)})$) for $l = 2, .., T$ ($l = T-1, T-2, .., 1$).
2) Merging $\vec{m}_{fp}(\mathbf{x}_l^{(L)})$ and $\overleftarrow{m}_{be}(\mathbf{x}_l^{(L)})$ in the message $m_{sm}(\mathbf{x}_l^{(L)}) = \mathcal{N}(\mathbf{x}_l^{(L)}; \eta_{sm,l}^{(L)}, \mathbf{C}_{sm,l}^{(L)})$, with $l = 2, 1, .., T-1$ ($m_{sm}(\mathbf{x}_1^{(L)}) = \overleftarrow{m}_{be}(\mathbf{x}_1^{(L)})$ and $m_{sm}(\mathbf{x}_T^{(L)}) = \vec{m}_{fp}(\mathbf{x}_T^{(L)})$ are assumed) on the basis of (38)-(39), so that a new final estimate $\eta_{sm,l}^{(L)}$ is available for $\mathbf{x}_l^{(L)}$.

We believe that, even if this procedure is conceptually appealing, the improvement it may provide in the estimation accuracy for the linear state component is influenced by a) the *number of modes* of the density of $\mathbf{x}_l^{(L)}$ (since the adopted *unimodal* model for this state component might provide a poor statistical representation of it) and b) the presence of *large errors*, at specific instants, in the estimated nonlinear state trajectory. As a matter of fact, in our computer simulations no improvement has been found in the estimation accuracy for the specific SSMs considered in Section V.

C. Derivation of the Rao-Blackwellized serial smoother

The backward processing in the SPS algorithm has been explicitly devised for estimating the marginal smoothing densities $\{f(\mathbf{x}_l|\mathbf{y}_{1:T})\}$; however, the structure of its recursion can be easily modified to generate, like in the RBPS method proposed in [20], M (equally likely) *nonlinear* state trajectories, that jointly provide a point mass approximation of the joint smoothing pdf $f(\mathbf{x}_{1:T}^{(N)}|\mathbf{y}_{1:T})$. In practice, this result can be achieved accomplishing: 1) a *single* forward pass (MPF); 2) M distinct backward passes, each generating a new trajectory for the *nonlinear* state component; 3) KS for each of the M nonlinear state trajectories, in order to estimate the associated linear state trajectories. As far as the second point is concerned, the recursive filtering algorithm we adopt in each backward pass is obtained by incorporating a *particle sampling mechanism* in the BIF of the SPS technique; this modification is due to the fact that the nonlinear state trajectory $\{\mathbf{x}_{be,l}^{(N)}, l = 1, 2, ..., T\}$ generated in the backward pass of the new particle smoother (called RBSS in the following) consists entirely of particles generated in the forward pass and not of a linear combination of them, like in the SPS algorithm (see (62)). Consequently, the first change we make in the SPS algorithm concerns the algorithm initialization (see step 2) of Algorithm 1); in fact, in the RBSS technique the input messages feeding the first backward recursion are expressed by $\overleftarrow{m}_{be}(\mathbf{x}_T^{(N)}) \triangleq \delta(\mathbf{x}_T^{(N)} - \mathbf{x}_{T/(T-1),j_T}^{(N)})$ and $\overleftarrow{m}_{be}(\mathbf{x}_T^{(L)}) \triangleq \mathcal{N}(\mathbf{x}_T^{(L)}; \eta_{fe,T,j_T}^{(L)}, \mathbf{C}_{fe,T,j_T}^{(L)})$, where $\mathbf{x}_{T/(T-1),j_T}^{(N)}$ denotes the particle drawn from the particle set $\{\mathbf{x}_{T/(T-1),j}^{(N)}\}$ on the basis of the associated weights $\{w_{T/T,j}\}$ (generated by the MPF MU for the nonlinear state component in its final recursion), whereas $\overleftarrow{m}_{be}(\mathbf{x}_T^{(L)})$ represents the Gaussian model associated with this particle (and generated by the MPF MU for the linear state component in its final recursion). The other modifications refer to step 3)-c) of Algorithm 1, which is replaced by the following two tasks: 1) drawing a new sample, denoted $\mathbf{x}_{l/(l-1),j_l}^{(N)}$, from the particle set $\{\mathbf{x}_{l/(l-1),j}^{(N)}\}$ on the basis of the particle weights $\{W_{sm,l,j}\}$ (see (58)); 2) setting $\mathbf{x}_{be,l}^{(N)} = \mathbf{x}_{l/(l-1),j_l}^{(N)}$ and $(\eta_{be,l}^{(L)}, \mathbf{C}_{be,l}^{(L)}) = (\eta_{be,l,,j_l}^{(L)}, \mathbf{C}_{be,l,,j_l}^{(L)})$.

The backward pass is followed by KS; this aims at computing the statistics for the linear state component only and involves a new forward pass, which is carried out under the assumption that the nonlinear state component is *known*; this generates a new couple $(\eta_{fp,l}^{(L)}, \mathbf{C}_{fp,l}^{(L)})$ for any l. Finally, similarly to [20], the smoothed covariance $\mathbf{C}_{sm,l}^{(L)}$ and the smoothed mean $\eta_{sm,l}^{(L)}$ are evaluated on the basis of (38) and (39), respectively (note also that the final estimate of $\mathbf{x}_l^{(L)}$ is expressed by $\eta_{sm,l}^{(L)}$).

The complete RBSS technique is given in Algorithm 2, where, for simplicity, the generation of a single trajectory is considered.

D. Comparison of the SPS and RBSS algorithms with other RBPS methods

In this Paragraph the SPS and the RBSS techniques devised in the previous two Paragraphs are compared with the other

Algorithm 2: Rao-Blackwellized serial smoothing

1 **Forward filtering**: Run step 1) of Algorithm 1.
2 **Initialisation of backward filtering**: draw a particle, denoted $\mathbf{x}_{T/(T-1),j_T}^{(N)}$, from the particle set $\{\mathbf{x}_{T/(T-1),j}^{(N)}\}$ on the basis of the particle weights $\{w_{T/T,j}\}$; then, set $\eta_{be,T}^{(L)} = \eta_{fe,T,j_T}^{(L)}$ and $\mathbf{C}_{be,T}^{(L)} = \mathbf{C}_{fe,T,j_T}^{(L)}$ and store them.
3 **Backward filtering and smoothing**: For $l = T-1$ to 1:
a- For $j = 1$ to N_p:
- *Backward information filtering and smoothing*: Run all the steps listed at points 3)-a) and 3)-b) of Algorithm 1.
b- *Computation of backward estimation messages*: draw a particle, denoted $\mathbf{x}_{l/(l-1),j_l}^{(N)}$, from the particle set $\{\mathbf{x}_{l/(l-1),j}^{(N)}\}$ on the basis of the particle weights $\{W_{sm,l,j}\}$; then, set $\eta_{be,l}^{(L)} = \eta_{be,l,,j_l}^{(L)}$ and $\mathbf{C}_{be,l}^{(L)} = \mathbf{C}_{be,l,,j_l}^{(L)}$.
4 **Linear state smoothing**: For $l = 1$ to T
a- Run a Kalman filter for the linear state component, conditionally on the nonlinear trajectory selected in step 3) and store the predicted mean and covariance, i.e. the couple $(\eta_{fp,l}^{(L)}, \mathbf{C}_{fp,l}^{(L)})$.
b- Compute the smoothed covariance $\mathbf{C}_{sm,l}^{(L)}$ and the smoothed mean $\eta_{sm,l}^{(L)}$ on the basis of (38) and (39), respectively.

RBPS algorithms available in the literature, namely with the particle smoothers proposed by Briers *et al.* in [13, Sec. 4, p. 75], by Fong *et al.* [14] and by Lindsten *et al.* [20] (these algorithms are denoted Alg-B, Alg-F and Alg-L, respectively, to ease reading of the remaining part of this manuscript). Despite their substantially different structures, the last three algorithms share the following relevant features: 1) the computation of an estimate of the *joint* smoothing density $f(\mathbf{x}_{1:T}|\mathbf{y}_{1:T})$; 2) the *reuse* of the FF particles and weights; 3) the use of *resampling* in the generation of backward trajectories; 4) the exploitation of *Kalman techniques* for the linear state component. In the following we provide some details about these features, so that some important differences between such techniques and the SPS/RBSS algorithms can be easily identified.

The *first feature* refers to the fact that these techniques aim at generating *realizations* from the complete *joint* smoothing pdf $f(\mathbf{x}_{1:T}|\mathbf{y}_{1:T})$. Each realization consists of a) a trajectory (i.e., a set of T particles, one for each observation instant) for the nonlinear state component and a set of T Gaussian pdfs (one for each observation instant) in Alg-B and Alg-L, or b) a trajectory for the entire state in Alg-F (since a particle-based representation is adopted for the linear state component too). This approach provides the following relevant advantage: any marginal smoothing density (like those we are interested in) can be easily obtained from the joint density by marginalization (i.e., by discarding the particle sets and the associated Gaussian densities that refer to the instants we are not interested in). This benefit, however, is obtained at the price of a substantial computational complexity in all cases. In fact, backward filtering and smoothing in Alg-F and Alg-L require to be re-run M times, if M realizations of $f(\mathbf{x}_{1:T}|\mathbf{y}_{1:T})$ are needed; luckily, the processing accomplished in each run reuses all the particles and the weights computed in the forward pass. On the contrary, a *single backward pass* is accomplished in the algorithm derived in Alg-B ; in this pass, however, the generation of a new set of weighted particles and Gaussian densities (representing the nonlinear state component and the linear state component, respectively) is carried out.

The *second feature* concerns the fact that the particles and the associated weights generated in the forward pass are reused in the backward pass, even if in different ways. More specifically, in the backward pass of the RPBS techniques of Alg-F and Alg-L, particles are re-weighted; moreover, each new weight is evaluated as the product of the weight computed in the forward pass for the considered particle with a new weight generated on the basis of backward statistics (see, in particular, the *particle smoothing* task of Algorithm 4 in [14, p. 443] and step 3)-b)-ii) of Algorithm 1 in [20, p. 357]). On the one hand, the reuse, in the backward pass, of the particles generated in the forward pass greatly simplifies BIF. On the other hand, it places a strong constraint on the *support* of each of the pdfs computed for nonlinear state component; in fact, such a support is restricted to that identified for the predicted/filtered pdfs in the forward pass[9]. This is the reason why Alg-B includes an algorithm for generating, in its backward pass, new particles, which are independent of those computed in the forward pass. The price to be paid for this, however, is represented by the additional computational load due to 1) particle generation in the backward pass and 2) the complexity of the processing required for fusing forward and backward particles (and their associated weights) in the computation of smoothed densities (see, in particular, [13, Par. 4.1.2, p. 80]).

As far as the *third feature* is concerned, it is worth mentioning that the use of resampling in Alg-F and Alg-L is substantially different from that of Alg-B. In fact, in the first case, resampling is applied to the particle set produced in the TU of each recursion of the forward pass when generating a new trajectory in a backward pass; this is motivated by the fact that the mechanism of particle selection can benefit from more refined statistical information, since the new weights generated in the backward pass for the available particle set are expected to be more reliable than those computed in the forward pass. On the contrary, in the second case, resampling is applied to the new particle set generated in each recursion of the backward pass, exactly like in the forward pass.

Finally, the *fourth feature* concerns the exploitation of Kalman techniques and, in particular, of KS for the linear state component in the considered RBPS algorithms. Note, however, that a different use of these standard tools is made in the considered manuscripts. In fact, on the one hand, in Alg-B and Alg-F, smoothing for linear state component is accomplished within the backward pass and exploits the statistical information about the linear state component generated by Rao-Blackwellized filtering in the forward pass. On the other hand, in Alg-L the backward pass aims at generating

[9]Note that this occurs in the SPS and in the RBSS techniques too.

a trajectory for the *nonlinear state component only*. For this reason, in this case, an additional forward pass for the linear state component only is accomplished, under the assumption that the nonlinear state trajectory is known, after that the backward pass has been completed; finally, KS is carried out to merge forward and backward information, as illustrated at the end of the previous Paragraph.

Let us now focus on the similarity of the proposed SPS technique with Alg-B, Alg-F and Alg-L in terms of the four features illustrated above. The SPS algorithm shares feature 4) and part of feature 2) with the other RBPS techniques; in fact, it reuses the FF particles, but not their weights (unless the weight $w_{4,l,j}$ (52) is replaced by the MPF weight $w_{l/l,j}$, as suggested in Paragraph IV-B). It does not share, however, features 1) and 3); this makes it much faster than Alg-B, Alg-F and Alg-L in the computation of marginal smoothed densities, since both resampling and the generation of multiple trajectories or new particles in the backward pass are time consuming tasks. In fact, the computational complexity associated with the computation of marginal smoothed densities is of order $O(N_pTM)$ for Alg-F and Alg-L, of order $O(N_p^2T)$ for Alg-B[10]; on the contrary, the computational complexity for the SPS technique is of order $O(N_pT)$ only.

The RBSS technique deserves different comments, since it is structurally very similar to Alg-F and, consequently, its complexity is also of order $O(N_pTM)$. In fact, the main difference between these two algorithms is related to the computation of particle weights in the backward pass. More specifically, in Alg-F, the evaluation of the weight for the j-th particle is not based on (40) (and, consequently, on (55)-(57)), but on the formula (see step b)-ii) of Algorithm 1 in [20, p. 357])

$$W_{l,j} = w_l^j \, Z_t^j \det\left(\Lambda_t^j\right)^{-1/2} \exp\left(-\frac{1}{2}\eta_t^j\right). \qquad (69)$$

Here, w_l^j corresponds to $w_{l/l,j}$ (i.e., it represents the j-th particle weight computed in the MPF MU for the nonlinear state component and, consequently, depends on $\mathbf{y}_l$), whereas the remaining factors appearing in the RHS of the last equation are related to the statistical information about $\mathbf{x}_l^{(L)}$ generated in both the forward pass and the backward pass, and to the pseudo-measurements. More specifically, it can be shown that (analytical details are omitted here for space limitations):

- The product $\det(\Lambda_t^j)^{-1/2}\exp(-\eta_t^j/2)$ appearing in the RHS of (69) quantifies the *correlation* between the Gaussian pdfs representing $\mathbf{x}_l^{(L)}$, conditioned on $\mathbf{x}_l^{(N)} = \mathbf{x}_{l/(l-1),j}^{(N)}$, in the forward pass and in the backward pass (η_t^j is expressed by the sum of three terms, a scalar product and two squared non Euclidean norms; see eq. (21a) of [20, p. 357]). Note, however, that these Gaussian pdfs are not expressed by $\vec{m}_{fp,j}(\mathbf{x}_l^{(L)})$ (18) and $\overleftarrow{m}_{be}(\mathbf{x}_l^{(L)})$ (61). In fact, on the hand, the Gaussian pdf referring to the l-th recursion of the forward pass (represented by $\mathcal{N}(z_l; \bar{z}_{l/l}^j, P_{l/l}^j)$; see eq. (11) of [20, p. 355])) is not that predicted in the previous recursion for the j-th particle, but is that generated in the MU of the l-th recursion on the basis of $\vec{m}_{fp,j}(\mathbf{x}_l^{(L)})$ and $\mathbf{y}_l$; on the other end, the Gaussian pdf referring to the l-th recursion of the backward pass is not influenced, unlike $\overleftarrow{m}_{be}(\mathbf{x}_l^{(L)})$, by the measurement $\mathbf{y}_l$ (the precision matrix and transformed mean vector for this Gaussian pdf are denoted Ω_l and λ_l, respectively; see eqs. (33b) and (33c) of [20, p. 358]), but only by the pseudo-measurement $\mathbf{z}_l^{(L)}$.
- If the notation of the algorithms developed in this manuscript is adopted, the factor Z_t^j appearing in the RHS of (69) can be expressed as[11] (see eq. (33a) of [20, p. 358])

$$Z_l^j = \left(\det\left[\mathbf{W}_{be,l+1}^{(L)} + \mathbf{W}_w^{(L)}\right]\right)^{-1/2} \exp\left(-\frac{\tau_l^j}{2}\right), \qquad (70)$$

where

$$\begin{aligned}\tau_l^j \;=\; & \left\|\mathbf{z}_{l,j}^{(L)}\right\|^2_{\mathbf{W}_w^{(N)}} + \left\|\mathbf{f}_{l,j}^{(L)}\right\|^2_{\mathbf{W}_{be,l+1}^{(L)}} \\ & -2\left(\mathbf{w}_{be,l+1}^{(L)}\right)^T \mathbf{f}_{l,j}^{(L)} - \left\|\mathbf{w}_{be,l+1}^{(L)}\right. \\ & \left.-\mathbf{W}_{be,l+1}^{(L)}\mathbf{f}_{l,j}^{(L)}\right\|^2_{\left[\mathbf{W}_{be,l+1}^{(L)}+\mathbf{W}_w^{(L)}\right]^{-1}}. \qquad (71)\end{aligned}$$

Consequently, the quantity Z_t^j (70) depends on both the pseudo-measurement $\mathbf{z}_{l,j}^{(L)}$ (24) and on the statistics of the backward estimate for the linear state component $\mathbf{x}_{l+1}^{(L)}$ (namely, on $\mathbf{w}_{be,l+1}^{(L)}$and $\mathbf{W}_{be,l+1}^{(L)}$), but it is not influenced by the statistics computed in the forward pass for that component.

Finally, it is worth stressing that Alg-B and Alg-F apply to a mixed linear/nonlinear SSM whose state equation for the nonlinear component (see (1)) does contain the nonlinear term $\mathbf{f}_l^{(L)}(\mathbf{x}_l^{(N)})$ (see [13, Sec. 4, p. 75] and [14, Sec. 4, p. 441]); consequently, the only alternative method applicable to the SSM described by (1)-(3) *in its complete form* is represented by Alg-L.

V. Numerical Results

In this Section the SPS and RBSS algorithms are compared, in terms of accuracy and computational load, for two specific SSMs with a) the RBPS developed in [20] (i.e., Alg-L) and b) MPF. As far as our comparison with MPF is concerned, it is worth remembering that, in principle, SMC filtering algorithms should be able to generate a point mass approximation of the joint smoothing pdf $f(\mathbf{x}_{1:T}^{(N)}|\mathbf{y}_{1:T})$ (and, consequently, of all its marginals); this would make backward filtering useless. Unluckily, SMC filtering techniques suffer from the so called *degeneracy problem* (e.g., see [13, Par. 1.3] and [29, Sec. 4]),

[10]The dependence on N_p^2 is due to the fact that the particles generated in the backward pass are different from those of the forward pass; this makes the evaluation of the marginal densities, that requires merging all the statistical information emerging from both passes (see [13, Par. 4.1.2, p. 80]) for further details), computationally intensive.

[11]Particle-independent factors are omitted in the following formula for simplicity.

so that their approximation of the above mentioned pdf is extremely poor for large T.

The first SSM (denoted SSM #1 in the following) we consider in our simulations refers to an agent node (represented as a rigid body) that randomly moves on a plane and whose state $\mathbf{x}_l$ in the l-th observation interval is defined as

$$\mathbf{x}_l \triangleq [\mathbf{v}_l^T, \mathbf{p}_l^T]^T, \tag{72}$$

where $\mathbf{v}_l \triangleq [v_{x,l}, v_{y,l}]^T$ and $\mathbf{p}_l \triangleq [p_{x,l}, p_{y,l}]^T$ represent the target velocity and its position, respectively (their components are expressed in m/s and in m, respectively). As far as the state update equations are concerned, we assume that the target velocity is approximately constant within each sampling interval and its evolution is described by the first-order autoregressive model

$$\mathbf{v}_{l+1} = \rho \mathbf{v}_l + (1-\rho)\, \mathbf{n}_{v,l}, \tag{73}$$

where ρ is a real parameter (with $0 < \rho < 1$), $\mathbf{n}_{v,l}$ is an *additive Gaussian noise* (AGN) vector characterized by the covariance matrix $\mathbf{I}_2$. Consequently, the dynamic model

$$\mathbf{p}_{l+1} = \mathbf{p}_l + \mathbf{v}_l \cdot T_s + \mathbf{n}_{p,l} \tag{74}$$

can be employed for target position; here, T_s is the sampling interval, and $\mathbf{n}_{p,l}$ is an AGN vector characterized by the covariance matrix $\sigma_p^2 \mathbf{I}_2$ and accounting for model inaccuracy. We also assume that noisy and unbiased measurements are available for both velocity and position; therefore, the measurement vector $\mathbf{y}_l$ can be expressed as

$$\mathbf{y}_l = \mathbf{x}_l + \mathbf{e}_l, \tag{75}$$

where $\mathbf{e}_l \triangleq [\mathbf{e}_{v,l}^T, \mathbf{e}_{p,l}^T]^T$ and $\mathbf{e}_{v,l}$ ($\mathbf{e}_{p,l}$) is an AGN vector characterized by the covariance matrix $\sigma_{ev}^2 \mathbf{I}_2$ ($\sigma_{ep}^2 \mathbf{I}_2$). As it can be easily inferred from (73)-(75), the first model we are considering is *linear* and *Gaussian*. Our interest in it is motivated by the fact that an optimal algorithm, namely KS [1], can be employed in this case and that its *mean square error* (MSE) performance represents a lower bound for any particle smoother applied to SSM #1. In our work, the state vector $\mathbf{x}_l$ (72) has been artificially partitioned, setting $\mathbf{x}_l^{(L)} = \mathbf{v}_l$ and $\mathbf{x}_l^{(N)} = \mathbf{p}_l$ (this choice is made for our second SSM too). Consequently, the state equation (73) ((74)) can been interpreted as an instance of (1) ((2)), with $\mathbf{A}_l^{(L)}(\mathbf{x}_l^{(N)}) = \mathbf{I}_2$, $\mathbf{f}_l^{(L)}(\mathbf{x}_l^{(N)}) = \mathbf{0}_{2,1}$ and $\mathbf{w}_l^{(L)} = \mathbf{n}_{v,l}$ (with $\mathbf{A}_l^{(N)}(\mathbf{x}_l^{(N)}) = T_s \mathbf{I}_2$, $\mathbf{f}_l^{(N)}(\mathbf{x}_l^{(N)}) = \mathbf{x}_l^{(N)}$ and $\mathbf{w}_l^{(N)} = \mathbf{n}_{p,l}$); moreover, comparing the measurement equation (75) with (3) leads easily to the conclusion that $\mathbf{h}_l(\mathbf{x}_l^{(N)}) = \mathbf{0}_{4,1}$ and $\mathbf{B}_l(\mathbf{x}_l^{(N)}) = \mathbf{I}_4$ for the considered SSM.

The second SSM (denoted SSM #2 in the following) is obtained from the first one by simply introducing in its dynamic model the contribution of the acceleration

$$\mathbf{a}_l(\mathbf{p}_l) = -a_0 \frac{\mathbf{p}_l}{\|\mathbf{p}_l\|} \frac{1}{1 + (\|\mathbf{p}_l\|/d_0)^2}, \tag{76}$$

exhibiting a *nonlinear dependence* on $\mathbf{p}_l$, and representing the effect of a force applied to the target and pointing towards the origin of our reference system; here, a_0 is a scale factor (expressed in m/s^2), whereas d_0 is a (small) *reference distance* that prevents the magnitude of acceleration from diverging if the target approaches the origin of our reference system (note that the magnitude of $\mathbf{a}_l(\mathbf{p}_l)$ (76) is approximately proportional to $\|\mathbf{p}_l\|^{-2}$). Consequently, the new state update equations are obtained by adding the new nonlinear terms $T\mathbf{a}_l(\mathbf{p}_l)$ and $(T^2/2)\mathbf{a}_l(\mathbf{p}_l)$) to the RHSs of (73) and (74), respectively. This entails that, as far as eqs. (1) and (2) are concerned, the only changes with respect to SSM #1 are represented by the functions $\mathbf{f}_l^{(L)}(\mathbf{x}_l^{(N)})$ and $\mathbf{f}_l^{(N)}(\mathbf{x}_l^{(N)})$, respectively; in fact, now we have that $\mathbf{f}_l^{(L)}(\mathbf{x}_l^{(N)}) = T_s \mathbf{a}_l(\mathbf{x}_l^{(N)})$ and $\mathbf{f}_l^{(N)}(\mathbf{x}_l^{(N)}) = \mathbf{x}_l^{(N)} - (T_s^2/2)\mathbf{a}_l(\mathbf{x}_l^{(N)})$. On the contrary, no change is required for the measurement model.

In our computer simulations, our assessment of state estimation *accuracy* is based on the evaluation of two *root* MSEs (RMSEs), one (denoted $RMSE_N$(alg), where 'alg' denotes the algorithm this parameter refers to) referring to the nonlinear state component, the other one (denoted $RMSE_L$(alg)) to the linear state component; note that each of these parameters represents the square root of the average MSE evaluated for the elements of the state component it refers to. Our assessment of *computational requirements* is based, instead, on assessing the average *computation time* required for processing a single *block* of measurements (this quantity is denoted CTB in the following). In all the simulations the following choices have been always made: a) $T = 200$ has been selected for the length of the observation interval; b) unless differently stated, $M = 100$ has been chosen for the number of trajectories generated by the RBSS algorithm and Alg-L (note that $M \lesssim N_p$ is recommended in [15]).

In our computer simulations, unless differently stated, the following values have been selected for the parameters of the considered SSMs: SSM #1 - $T_s = 10^{-2}$ s, $\rho = 0.99$, $\sigma_p = 10^{-2}$ m, $\sigma_v = 10^{-2}$ m/s, $\sigma_{ev} = 0.1$ m/s, $\sigma_{ep} = 0.1$ m, $\mathbf{v}_0 = [10^{-2}$ m/s, 10^{-2} m/s], $\mathbf{p}_0 = [0, 0]$; SSM #2 - $T = 10^{-2}$ s, $\rho = 0.995$, $\sigma_p = 10^{-2}$ m, $\sigma_v = 5 \cdot 10^{-3}$ m/s, $\sigma_{ev} = 5 \cdot 10^{-2}$ m/s, $\sigma_{ep} = 5 \cdot 10^{-2}$ m, $\mathbf{v}_0 = [10^{-2}$ m/s, 10^{-2} m/s], $\mathbf{p}_0 = [10^{-2}$ m, 10^{-2}m], $a_0 = 0.5$ m/s^2 and $d_0 = 10^{-2}$ m.

Given the parameters listed above, $RMSE_L(KF) \cong 0.02898$ and $RMSE_L(KS) = 0.0222$ have been obtained for *Kalman filtering* (KF) and KS, respectively; this means that KS roughly provides a 25% improvement in state estimation accuracy with respect to KF. Moreover, it has been found that, on the one hand, even for small values of N_p (say, $N_p \geq 10$), the accuracy of considered particle-based filtering (smoothing) algorithms is very close to that of KF (KS) in terms of $RMSE_L$; on the other hand, a significant number of particles is required to closely approach the performance of Kalman algorithms in terms of $RMSE_N$. The last conclusion is easily inferred from Fig. 4, that shows[12] the dependence of $RMSE_N$ on the number of particles (N_p) for MPF and for the three considered smoothing algorithms (the two horizontal lines represent the RMSEs achieved by KF and KS); note that two $RMSE_N$ curves are shown for Alg-L, one referring to the case $M = 100$, the other one to the case $M = N_p$. These results also lead to the conclusion that:

[12]In these and in the following figures simulation results are identified by markers, whereas continuous lines are drawn to ease reading.

1) A small improvement in the estimation accuracy of all the considered algorithms is achieved for $N_p > 200$.
2) The MPF and the SPS algorithm closely approach KF and KS accuracy, respectively, as N_p increases.
3) Even if the RBSS algorithm and Alg-L provide by far richer statistical information than the SPS algorithm, they achieve slightly better accuracy in state estimation for $N_p < 300$, despite their substantially larger computational load. Moreover, for $N_p > 300$ the SPS algorithm performs slightly better than the RBSS algorithm and Alg-L if $M = 100$ is set; this is due to the fact the overall number of trajectories generated by the last two algorithms is not allowed to increase with N_p. In fact, a further improvement in their accuracy can be achieved by increasing M proportionally to N_p; this is evidenced by the trend of the $RMSE_N$ curve provided for Alg-L with $M = N_p$. Unluckily, this result is achieved at the price of a substantial increase in the computational load. For instance, the CTB of the Alg-L with $M = N_p$ is roughly 2, 3, 4 and 5 times larger than that of the Alg-L with $M = 100$ for $N_p = 200$, 300, 400 and 500, respectively.

Further numerical results (not shown here for space limitations) have also evidenced that: a) as N_p increases, the particle-based representation $\hat{f}(\mathbf{p}_l|\mathbf{y}_{1:N})$ (see (66)) of the pdf of $\mathbf{x}_l^{(N)}$ closely approaches the Gaussian pdf computed by KF; b) the RMSE improvement provided by the considered smoothing algorithms over MPF is mainly related to its 'peak shaving' effect in state estimation error (in other words, the amplitude of the spikes appearing in the state estimation error at the end of the forward pass are substantially reduced by smoothing; c) a minor degradation in the estimation accuracy of the SPS and RBSS algorithms is found if $w_{4,l,j} = w_{l/l,j}$ and $w_{2,l,j} = 1$ are set in formula (40) to reduce the computational effort[13] required by the evaluation of the particle weights $\{W_{l,j}\}$ (consequently, only the weight $w_{1,l,j}$ (41) has to be computed in BIF; note that this requires computing the inverse of the matrix $\mathbf{C}_{1,l,j}^{(N)}$ (43) only). As far the last point is concerned, it is also worth stressing that the little relevance of the weight $w_{2,l,j}$ is due to the fact that the correlation appearing in the RHS of (47) exhibits a weak dependence on the selected particle in the presence of strong process noise for the linear state component (i.e., when σ_v is large in this case).

All the conclusions illustrated above are also supported by the performance results in Fig. 5, that shows the dependence of $RMSE_N$ (blue lines) and $RMSE_L$ (red lines) on the number of particles (N_p) for SSM #2. In particular, it is interesting to note that, even in this case, all the $RMSE_L$ curves are flat, whereas the $RMSE_N$ curves achieve a floor as N_p increases. Moreover, the accuracy gap between the SPS algorithm and the RBSS algorithm (or Alg-L) is really small, despite the fact they have substantially different computational requirements. This can be easily inferred from Fig. 6, that shows the dependence of the CTB on the number of particles (N_p) for all the considered algorithms. In particular, these results show that:

[13]This approximation can be adopted in the RBSS algorithm too; the resulting technique is called *simplified* RBSS (SBRSS) in the following.

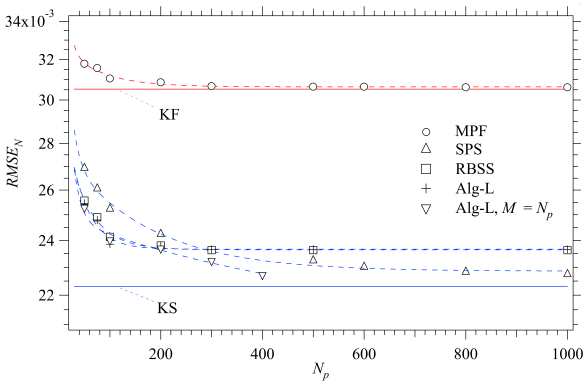


Fig. 4: $RMSE_N$ performance versus N_p for SSM #1; MPF, SPS, RBSS and Alg-L are considered.

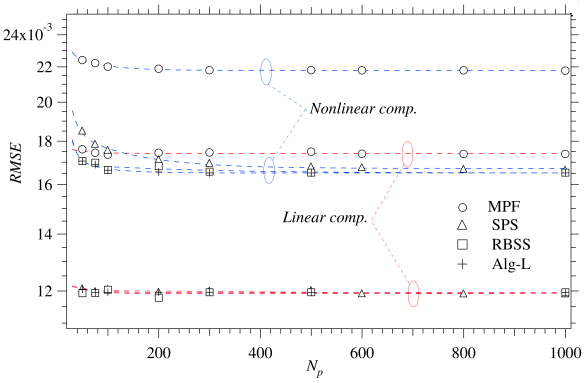


Fig. 5: RMSE performance versus N_p for the linear state component ($RMSE_L$) and the nonlinear state component ($RMSE_N$) for SSM #2; MPF, SPS, RBSS and Alg-L are considered.

1) The accuracy improvement of SPS over MPF is obtained at the price of a limited increase in computational complexity (the SPS and the MPF algorithms have similar computational requirements);
2) The CTB gap between Alg-L (or the RBSS algorithm) and the SPS algorithm is significant (for instance, the computation time of Alg-L is about 50 times larger than that of SPS for $N_p = 100$).
3) The RBSS and its simplified version (SRBSS) have larger computational requirements than Alg-L for the considered SSM, despite the fact that the evaluation of the weights for SRBSS is based on a substantially simpler formula than that employed for Alg-L. This gap can be related to the slightly different processing accomplished for the linear state component (see the last part of Paragraph IV-D).

In our work the dependence of $RMSE_L$ and $RMSE_N$ on the intensity of the measurement noise has been also analysed. Our results (not illustrated here for space limitations) show that the performance gap between MPF and all the considered smoothing gets larger as σ_e increases; this is due to the fact that a stronger measurement noise results in a poorer quality of the statistical information generated in the forward pass,

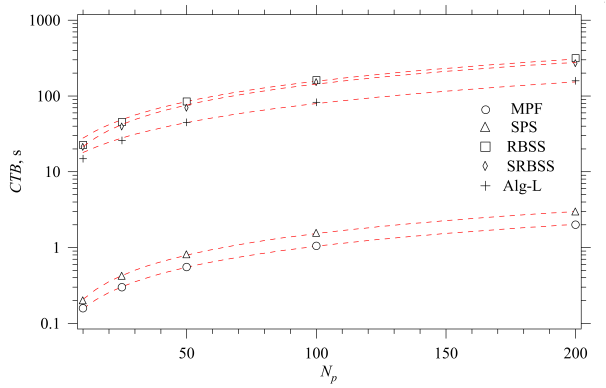


Fig. 6: CTB versus N_p for SSM #2; MPF, SPS, RBSS, SRBSS and Alg-L are considered.

and this impairs more and more the estimation process in the backward pass.

VI. Conclusions

In this manuscript the smoothing problem for SSMs has been analysed from a FG perspective. This has allowed us to devise two new particle smoothing algorithms for CLG SSMs. The first algorithm, dubbed SPS, generates estimates of the marginal smoothing densities at a limited computational cost. Moreover, as evidenced by our computer simulations for specific SSMs, its accuracy is similar to that achieved by the second algorithm, dubbed RBSS. The RBSS technique, however, generates an estimate of the joint smoothing pdf for the entire observation interval; in addition, its accuracy in state estimation and computational costs are comparable with those of the RBPS algorithm recently proposed by Lindsten et al. in [20].

Our future work concerns the application of FG methods to the problems of filtering and smoothing for other classes of SSMs.

Acknowledgment

We would like to thank Dr. Fredrik Lindsten (Uppsala University, Department of Information Technology) and the anonymous Reviewers for their constructive comments, that really helped us to improved the quality of this manuscript.

References

[1] B. Anderson and J. Moore, **Optimal Filtering**, Englewood Cliffs, NJ, Prentice-Hall, 1979.

[2] M. S. Arulampalam, S. Maskell, N. Gordon and T.Clapp, "A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking", *IEEE Trans. Sig. Proc.*, vol. 50, no. 2, pp. 174-188, Feb. 2002.

[3] A. Doucet, J. F. G. de Freitas and N. J. Gordon, "An Introduction to Sequential Monte Carlo methods," in **Sequential Monte Carlo Methods in Practice**, A. Doucet, J. F. G. de Freitas, and N. J. Gordon, Eds. New York: Springer-Verlag, 2001.

[4] A. Doucet, S. Godsill and C. Andrieu, "On Sequential Monte Carlo Sampling Methods for Bayesian Filtering", *Statist. Comput.*, vol. 10, no. 3, pp. 197-208, 2000.

[5] F. Gustafsson, "Particle Filter Theory and Practice with Positioning Applications", *IEEE Aerosp. and Electr. Syst. Mag.*, vol. 25, no. 7, pp. 53-82, July 2010.

[6] R. Douc, A. Garivier, E. Moulines and J. Olsson, "Sequential Monte Carlo smoothing for general state space hidden Markov models", *Ann. Appl. Probab.*, vol. 21, no. 6, pp. 2109–2145, 2011.

[7] G. Kitagawa, "Non-Gaussian state-space modeling of nonstationary time series", *Journal of the American Statistical Association*, vol. 82, pp. 1032-1063, 1987.

[8] G. Kitagawa, "The two-filter formula for smoothing and an implementation of the Gaussian-sum smoother", *Annals of the Institute of Statistical Mathematics*, vol. 46, pp. 605-623, 1994.

[9] G. Kitagawa, "Monte Carlo filter and smoother for non-Gaussian nonlinear state space models", *J. Comput. Graph. Statist.*, vol. 5, no. 1, pp. 1–25, 1996.

[10] Y. Bresler, "Two-filter formula for discrete-time non-linear Bayesian smoothing", *Int. Journal of Control*, vol. 43, no. 2, pp. 629-641, 1986.

[11] B. N. Vo, B. T. Vo and R. P. S. Mahler, "Closed-Form Solutions to Forward–Backward Smoothing", *IEEE Trans. Sig. Proc.*, vol. 60, no. 1, pp. 2-17, Jan. 2012.

[12] S. J. Godsill, A. Doucet, and M. West, "Monte Carlo smoothing for nonlinear time series", *J. Amer. Statist. Assoc.*, vol. 99, no. 465, pp. 156–168, Mar. 2004.

[13] M. Briers, A. Doucet and S. Maskell, "Smoothing algorithms for state-space models", *Ann. Inst. Statist. Math.*, vol. 62, no. 1, pp. 61–89, Feb. 2010.

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

[15] F. Lindsten and T. B. Schön, "Backward simulation methods for Monte Carlo statistical inference", *Foundat. Trends Mach. Learn.*, vol. 6, no. 1, pp. 1–143, 2013.

[16] R. Chen and J. S. Liu, "Mixture Kalman filters", *J. Roy. Statist. Soc.: Ser. B*, vol. 62, no. 3, pp. 493–508, 2000.

[17] T. Schön, F. Gustafsson, P.-J. Nordlund, "Marginalized Particle Filters for Mixed Linear/Nonlinear State-Space Models", *IEEE Trans. Sig. Proc.*, vol. 53, no. 7, pp. 2279-2289, July 2005.

[18] C. Andrieu and A. Doucet, "Particle filtering for partially observed Gaussian state space models", *J. Roy. Statist. Soc.: Ser. B*, vol. 64, no. 4, pp. 827–836, 2002.

[19] J. Olsson, R. Douc, O. Cappé, and E. Moulines, "Sequential Monte Carlo smoothing with application to parameter estimation in nonlinear state-space models", *Bernoulli*, vol. 14, no. 1, pp. 155–179, 2008.

[20] F. Lindsten, P. Bunch, S. Särkkä, T. B. Schön and S. J. Godsill, "Rao-Blackwellized Particle Smoothers for Conditionally Linear Gaussian Models", *IEEE J. Sel. Topics in Sig. Proc.*, vol. 10, no. 2, pp. 353-365, March 2016.

[21] G. M. Vitetta, E. Sirignano, F. Montorsi and M. Sola, "Marginalized Particle Filtering and Related Filtering Techniques as Message Passing", submitted to the *IEEE Trans. Inf. Theory*, july 2016 (available online at https://arxiv.org/abs/1605.03017).

[22] G. M. Vitetta, E. Sirignano and F. Montorsi, "A Novel Message Passing Algorithm for Online Bayesian Filtering: Turbo Filtering", Proc. of the *IEEE ICC 2017 Workshop on Advances in Network Localization and Navigation (ANLN)*, Paris, May 2017.

[23] H.-A. Loeliger, J. Dauwels, Junli Hu, S. Korl, Li Ping, F. R. Kschischang, "The Factor Graph Approach to Model-Based Signal Processing", *IEEE Proc.*, vol. 95, no. 6, pp. 1295-1322, June 2007.

[24] F. R. Kschischang, B. Frey, and H. Loeliger, "Factor Graphs and the Sum-Product Algorithm", *IEEE Trans. Inf. Theory*, vol. 41, no. 2, pp. 498-519, Feb. 2001.

[25] H.-A. Loeliger, L. Bruderer, H. Malmberg, F. Wadehn and N. Zalmai, "On Sparsity by NUV-EM, Gaussian Message Passing, and Kalman Smoothing", *Proc. of the 2016 Inf. Theory & Appl. Workshop* (ITA), La Jolla, CA (USA), Jan. 2016.

[26] F. Wadehn, J. Dauwels, H.-A. Loeliger and H. Yu, "Outlier-insensitive Kalman Smoothing and Marginal Message Passing", *Proc. of the 24th European Sig. Proc. Conf.* (EUSIPCO 2016), Budapest (Hungary), August 2016.

[27] T. Schön, "Example Used in Exemplifying the Marginalized (Rao-Blackwellized) Particle Filter", Nov. 2010 (available at users.isy.liu.se/en/rt/schon/Code/RBPF/Document/MPFexample.pdf).

[28] A. R. Runnalls, "Kullback-Leibler Approach to Gaussian Mixture Reduction", *IEEE Trans. on Aerosp. and Elec. Syst.*, vol. 43, no. 3, pp. 989-999, July 2007.

[29] N. Kantas, A. Doucet, S. S. Singh, J. Maciejowski and N. Chopin, "On Particle Methods for Parameter Estimation in State-Space Models", *Statistical Science*, vol. 30, no. 3, pp. 328–351, 2015.