Rao–Blackwellized Gaussian Smoothing

Roland Hostettler and Simo Särkkä

This is a pre-print of a paper accepted for publication in *IEEE Transactions on Automatic Control*. When citing this work, you must always cite the original article:

R. Hostettler and S. Särkkä, "Rao–Blackwellized Gaussian smoothing," *IEEE Transactions on Automatic Control*, 2019

DOI:

10.1109/TAC.2018.2828087

Rao–Blackwellized Gaussian Smoothing

Roland Hostettler, *Member, IEEE,* and Simo Särkkä, *Senior Member, IEEE*

***Abstract*—In this paper, we consider Rao–Blackwellization of linear substructures in sigma-point-based Gaussian assumed density smoothers. We derive marginalized prediction, smoothing, and update steps for the mixed linear/nonlinear Gaussian state-space model as well as for a hierarchical model for both conventional and iterated posterior linearization Gaussian smoothers. The proposed method is evaluated in a numerical example and it is shown that the computational complexity is reduced considerably compared to non-Rao–Blackwellized Gaussian smoothers for systems with high-dimensional linear subspaces.**

***Index Terms*—Gaussian assumed density smoothing, Rao–Blackwellization, nonlinear smoothing, nonlinear state estimation**

I. Introduction

Gaussian assumed density filtering and smoothing are popular approaches to approximate Bayesian filtering and smoothing in nonlinear state-space systems. In these methods, the filtering density $p(x_t \mid y_{1:t})$ (where x_t is the state vector and $y_{1:t} = \{y_1, y_2, \ldots, y_t\}$ are the measurements) and the smoothing density $p(x_t \mid y_{1:T})$ (for $1 \leq t \leq T$) are approximated as Gaussian densities (see, e.g., [1]–[3]). These approaches have the advantage that only estimating the first two moments of the posterior densities is required, similarly to the Kalman filter and Rauch–Tung–Striebel smoother [3]. The resulting moment matching integrals can be solved numerically using different types of sigma-point methods such as the third and higher order unscented transforms [4]–[9], Gauss–Hermite quadratures [1], [2], or spherical cubatures [4], [10], [11]. In many applications, these approaches are superior to simple Taylor series-based linearizations [12], [13].

A drawback of sigma-point methods is that even when the state-space model exhibits a linear substructure, the numerical integration is performed over the whole state-space. This wastes computational resources by numerically integrating the analytically tractable subspace. The aim of this article is to use Rao–Blackwellization to reduce this computational burden of sigma-point-based Gaussian smoothers. The use of Rao–Blackwellization is common in the context of sequential Monte Carlo (SMC) methods (particle filters and smoothers) that approximate the posterior densities using a set of weighted samples rather than an assumed density [14]–[17]. In SMC methods Rao–Blackwellization is used to reduce the number of samples required for approximating the posterior density.

Rao–Blackwellization has also been considered in Gaussian filtering. For example, Rao–Blackwellized unscented Kalman filters for non-mixing and completely mixing conditional linear Gaussian state-space models were introduced in [18] and [19]. Furthermore, [20] discussed models with directly and indirectly observed subsets of state variables as well as linear and nonlinear observations. Similarly, [21] considered models where only part of the state is observed nonlinearly and proposes a truncated unscented Kalman filter. A more generalized approach was presented in [22], where it is assumed that the problem exhibits a generic conditionally analytically tractable (with respect to the assumed Gaussian density) substructure. A similar approach was introduced in [23] in the context of Gauss–Hermite filtering. A marginalized UKF for correlated process and measurement noises was developed in [24]. Finally, a unified view on marginalized Gaussian filtering from a subspace projection perspective was introduced in [25].

The contribution of this paper is to derive novel Rao–Blackwellized Gaussian smoothing algorithms for two general classes of models recently considered in the context of Rao–Blackwellized sequential Monte Carlo smoothers [17]. As the model classes are more general than what has previously been considered in the context of Rao–Blackwellized Gaussian filtering algorithms, the results also extend the existing results for the Rao–Blackwellized Gaussian filters. Specifically, the technical contributions of this paper are as follows. First, we derive statistical linear regression for conditionally affine transformations of Gaussian random variables. Second, we develop Rao–Blackwellized Gaussian smoothers and Rao–Blackwellized posterior linearization smoothers [26] for two commonly encountered models, the fully mixing linear/nonlinear state-space model and a hierarchical model [17]. Third, we analytically analyze the reduction of the computational complexity in the proposed method. Fourth, the proposed methods are evaluated and compared to their non-Rao–Blackwellized counterparts as well as sequential Monte Carlo methods.

II. Problem Formulation

Consider the general nonlinear state-space system

$$x_t = f(x_{t-1}) + q_t, \tag{1a}$$

$$y_t = g(x_t) + r_t, \tag{1b}$$

where $x_t \in \mathbb{R}^{N_x}$ is the latent state vector with initial density $x_0 \sim \mathcal{N}(\hat{x}_0, P_0)$, $q_t \sim \mathcal{N}(0, Q_t)$ the Gaussian process noise, $f(\cdot)$ the dynamic model function, $y_t \in \mathbb{R}^{N_y}$ the measurement, $r_t \sim \mathcal{N}(0, R_t)$ the measurement noise, $g(\cdot)$ the measurement function, and t denotes the discrete time index. Then, the objective of Gaussian smoothing is to find a Gaussian approximation to the smoothing density

$$p(x_t \mid y_{1:T}) \approx \mathcal{N}(x_t; \hat{x}_{t|T}, P_{t|T}). \tag{2}$$

In this work, two particular subclasses of the general model (1) are considered. In both of these classes, the state space can be split into nonlinear and linear subspaces $x_t^n \in \mathbb{R}^{N_x^n}$ and $x_t^l \in \mathbb{R}^{N_x^l}$ such that $x_t = \begin{bmatrix}(x_t^n)^\mathsf{T} & (x_t^l)^\mathsf{T}\end{bmatrix}^\mathsf{T}$, which yields an analytically tractable substructure. The first model, Model 1, is a commonly encountered (see, e.g., [27]–[29]) mixed linear/nonlinear Gaussian state-space model defined as follows.

Model 1 (Mixed linear/nonlinear Gaussian State-Space Model)**.** *The mixed linear/nonlinear Gaussian state-space model is defined as*

$$x_t^n = f_t^n(x_{t-1}^n) + A_t^n(x_{t-1}^n)x_{t-1}^l + q_t^n, \tag{3a}$$

$$x_t^l = f_t^l(x_{t-1}^n) + A_t^l(x_{t-1}^n)x_{t-1}^l + q_t^l, \tag{3b}$$

$$y_t = g_t(x_t^n) + B_t(x_t^n)x_t^l + r_t. \tag{3c}$$

The noise terms q_t^n, q_t^l, and r_t are independent zero-mean, Gaussian random variables with $\mathrm{Cov}\{q_t^n\} = Q^n(x_{t-1}^n)$, $\mathrm{Cov}\{q_t^l\} = Q^l(x_{t-1}^n)$, $\mathrm{Cov}\{q_t^n, q_t^l\} = Q^{nl}(x_{t-1}^n)$, *and* $\mathrm{Cov}\{r_t\} = R(x_t^n)$*, respectively, and the initial state is Gaussian according to* $p(x_0) = \mathcal{N}(x_0; \hat{x}_{0|0}, P_{0|0})$.

In this model, the dynamics are mixing, that is, both the nonlinear and linear states may affect each other (see (3a)–(3b)). Furthermore, the nonlinear states are observed through some nonlinear function $g_t(\cdot)$

Financial support by the Academy of Finland under grants no. #266940 and #295080 is hereby gratefully acknowledged.

R. Hostettler and S. Särkkä are with the Department of Electrical Engineering and Automation, Aalto University, Finland. E-Mail: {roland.hostettler, simo.sarkka}@aalto.fi

while the linear states are observed through the measurement matrix $B_t(\cdot)$. This type of model is frequently encountered in applications such as target tracking, where, for example, the nonlinear states are the target's position that are observed nonlinearly (e.g. range and bearing measurements) and the linear states are the target's velocity [28].

The second model is the hierarchical model given below [30]–[32].

Model 2 (Hierarchical Model)**.** *The hierarchical model is defined as*

$$x_t^n = f_t^n(x_{t-1}^n) + q_t^n, \tag{4a}$$
$$x_t^l = f_t^l(x_t^n) + A_t^l(x_t^n)x_{t-1}^l + q_t^l, \tag{4b}$$
$$y_t = g_t(x_t^n) + B_t(x_t^n)x_t^l + r_t, \tag{4c}$$

where $q_t^n \sim p(q_t^n)$ *with* $\mathrm{E}\{q_t^n\} = 0$ *and* $\mathrm{Cov}\{q_t^n\} = Q^n(x_{t-1}^n)$, $q_t^l \sim \mathcal{N}(0, Q^l(x_t^n))$, $\mathrm{Cov}\{q_t^n, q_t^l\} = 0$, $r_t \sim \mathcal{N}(0, R(x_t^n))$, *and* $p(x_0) = \mathcal{N}(x_0; \hat{x}_{0|0}, P_{0|0})$.

Model 2 can be thought of as a generalized jump Markov model. Here, the linear and nonlinear states do not mix fully: the nonlinear state affects the linear state but not vice versa. Note that Model 2 is not a special case of Model 1: The difference lies in the linear dynamics (4b) that depend on x_t^n rather than x_{t-1}^n, while y_t depends on the current state x_t in both models. Also note that for Model 2, the state dynamics for x_t^n are sometimes given in terms of the transition density $x_t^n \sim p(x_t^n \mid x_{t-1}^n)$ (see [32]). Here, the functional form is chosen since it will simplify the derivations later on.

Given these two models, the aim is then to find Gaussian approximations of the form (2), taking the analytically tractable substructure into account. Note that the explicit dependence of f, A, Q, g, B, and R on t and x_t^n is omitted for the remainder of this paper.

III. Gaussian Smoothing

In this section, Gaussian filtering, smoothing, and posterior linearization smoothing are briefly reviewed. First, note that the Rauch–Tung–Striebel (RTS) smoothing recursion [3], [33] is given by a recursion backward in time and found from the joint approximation

$$\begin{aligned} p(x_t, x_{t+1} \mid y_{1:T}) &= p(x_t \mid x_{t+1}, y_{1:t}) p(x_{t+1} \mid y_{1:T}) \\ &\approx \mathcal{N}\left(\begin{bmatrix} x_t \\ x_{t+1} \end{bmatrix}; \begin{bmatrix} \hat{x}_{t|T} \\ \hat{x}_{t+1|T} \end{bmatrix}, \begin{bmatrix} P_{t|T} & E_t \\ E_t^{\mathsf{T}} & P_{t+1|T} \end{bmatrix} \right) \end{aligned}$$

by marginalizing with respect to x_{t+1}. The density $p(x_t \mid x_{t+1}, y_{1:t})$ is obtained from the joint-approximation $p(x_t, x_{t+1} \mid y_{1:t})$ during the prediction step of a Gaussian filter. This is given by [3]

$$\begin{aligned} &p(x_t, x_{t+1} \mid y_{1:t}) \\ &\quad \approx \mathcal{N}\left(\begin{bmatrix} x_t \\ x_{t+1} \end{bmatrix}; \begin{bmatrix} \hat{x}_{t|t} \\ \hat{x}_{t+1|t} \end{bmatrix}, \begin{bmatrix} P_{t|t} & C_t \\ C_t^{\mathsf{T}} & P_{t+1|t} \end{bmatrix} \right). \end{aligned} \tag{5}$$

Furthermore, since the recursion is backwards in time, a Gaussian approximation of the form (2) is given for $p(x_{t+1} \mid y_{1:T})$. Then, the well-known RTS smoothing equations [3], [33] are found to be

$$G_t = C_t P_{t+1|t}^{-1}, \tag{6a}$$
$$\hat{x}_{t|T} = \hat{x}_{t|t} + G_t(\hat{x}_{t+1|T} - \hat{x}_{t+1|t}), \tag{6b}$$
$$P_{t|T} = P_{t|t} + G_t \left(P_{t+1|T} - P_{t+1|t}\right) G_t^{\mathsf{T}}, \tag{6c}$$
$$E_t = G_t P_{t+1|T}. \tag{6d}$$

In order to find the approximation (5), a Gaussian filter has to be run in forward direction which provides a Gaussian approximation of the one step ahead prediction density $p(x_t \mid y_{1:t-1})$. The approximation of the filtering density $p(x_t \mid y_{1:t})$ is found from the joint approximation

$$p(x_t, y_t \mid y_{1:t-1}) \approx \mathcal{N}\left(\begin{bmatrix} x_t \\ y_t \end{bmatrix}; \begin{bmatrix} \hat{x}_{t|t-1} \\ \hat{y}_{t|t-1} \end{bmatrix}, \begin{bmatrix} P_{t|t-1} & D_t \\ D_t^{\mathsf{T}} & S_t \end{bmatrix} \right) \tag{7}$$

and subsequent conditioning on y_t (see, e.g., [3] for details). This yields the well-known Kalman filter update given by

$$K_t = D_t S_t^{-1}, \tag{8a}$$
$$\hat{x}_{t|t} = \hat{x}_{t|t-1} + K_t(y_t - \hat{y}_{t|t-1}), \tag{8b}$$
$$P_{t|t} = P_{t|t-1} - K_t S_t K_t^{\mathsf{T}}. \tag{8c}$$

Note that the smoothing recursion (6) not only depends on the means $\hat{x}_{t|t}$, $\hat{x}_{t+1|t}$, $\hat{x}_{t+1|T}$ and their respective covariances, but also the cross-covariance C_t. The latter is not required for filtering alone but is calculated during the prediction step (see (5)).

The Gaussian approximations in (5) and (7) require us to calculate the unknown moments $\hat{x}_{t+1|t}$, $P_{t+1|t}$, and C_t as well as $\hat{y}_{t|t-1}$, D_t, and S_t, respectively. These can be found by approximating the nonlinear state transition and measurement functions ($f(\cdot)$ and $g(\cdot)$) by using statistical linear regression (SLR) [34]. In regular one-pass Gaussian smoothing (e.g. [2]), SLR is performed with respect to the prior densities $p(x_{t-1} \mid y_{1:t-1})$ for the prediction and $p(x_t \mid y_{1:t-1})$ for the measurement update. In contrast to regular Gaussian smoothing, the recently proposed posterior linearization smoothing approach [26] tries to linearize with respect to the posterior $p(x_t \mid y_{1:T})$ directly. This can lead to significant performance gain in terms of the error in cases where the prior and posterior overlap poorly. Since the posterior is unknown to start with, the following iterative scheme can be used to gradually obtain improved approximations of the posterior [26]. First, regular smoothing is used to obtain an initial approximation of the posterior $p^0(x_t \mid y_{1:T}) \approx \mathcal{N}(x_t; \hat{x}_{t|T}^0, P_{t|T}^0)$. Then, the nonlinear functions $f(\cdot)$ and $g(\cdot)$ are linearized using SLR with respect to $p^0(x_t \mid y_{1:T})$ and smoothing is done anew in order to obtain $p^1(x_t \mid y_{1:T})$. The process is repeated for a predefined number of iterations or until convergence is achieved.

IV. Rao–Blackwellized Gaussian Smoothing

In this section, the Rao–Blackwellized Gaussian smoothing algorithms for the two models discussed in Section II will be developed. First, Rao–Blackwellized statistical linear regression of conditionally affine transformations of Gaussian random variables is derived. This is then applied to Gaussian filtering and smoothing to obtain the Rao–Blackwellized prediction and measurement update steps. Note that similar to the state vector, the covariance matrices can be partitioned too, such that

$$P_{t|t} = \begin{bmatrix} P_{t|t}^n & P_{t|t}^{nl} \\ (P_{t|t}^{nl})^{\mathsf{T}} & P_{t|t}^l \end{bmatrix} \tag{9}$$

and similar for $P_{t|t-1}$, C_t, D_t, $P_{t|T}$, and E_t.

A. Rao–Blackwellized Statistical Linear Regression

As discussed in the previous section, statistical linear regression can be used to approximate the nonlinear state transition and measurement functions [34]. In case these functions exhibit a conditionally affine substructure, the resulting integrals to be solved can be reduced in dimensionality since the conditionally affine subspace is analytically tractable. This is reviewed in Lemma 1, followed by the special case when SLR is applied with respect to the prior density in Corollary 1. The latter is the case for traditional Gaussian filtering and smoothing [3], [13].

Lemma 1 (Rao–Blackwellized Statistical Linear Regression)**.** *Let* $z_2 = h(z_1^n) + H(z_1^n)z_1^l + v$,

$$p(z_1^n, z_1^l) = \mathcal{N}\left(\begin{bmatrix} z_1^n \\ z_1^l \end{bmatrix}; \begin{bmatrix} \mu_1^n \\ \mu_1^l \end{bmatrix}, \begin{bmatrix} \Sigma_1^n & \Sigma_1^{nl} \\ (\Sigma_1^{nl})^{\mathsf{T}} & \Sigma_1^l \end{bmatrix} \right),$$

and $p(v \mid z_1^n) = \mathcal{N}(v; 0, \Sigma_v(z_1^n))$. *Furthermore, let us fix the linearization density to*

$$\pi(z_1^n, z_1^l) = \mathcal{N}\left(\begin{bmatrix} z_1^n \\ z_1^l \end{bmatrix}; \begin{bmatrix} \mu_{1,\pi}^n \\ \mu_{1,\pi}^l \end{bmatrix}, \begin{bmatrix} \Sigma_{1,\pi}^n & \Sigma_{1,\pi}^{nl} \\ (\Sigma_{1,\pi}^{nl})^\mathsf{T} & \Sigma_{1,\pi}^l \end{bmatrix}\right). \tag{10}$$

Given these, let

$$\tilde{\mu}_{1,\pi}^l = \mu_{1,\pi}^l + (\Sigma_{1,\pi}^{nl})^\mathsf{T}(\Sigma_{1,\pi}^n)^{-1}(z_1^n - \mu_{1,\pi}^n), \tag{11a}$$
$$\tilde{\Sigma}_{1,\pi}^l = \Sigma_{1,\pi}^l - (\Sigma_{1,\pi}^{nl})^\mathsf{T}(\Sigma_{1,\pi}^n)^{-1}\Sigma_{1,\pi}^{nl}, \tag{11b}$$

and

$$\mu_{2,\pi} = \int (h + H\tilde{\mu}_{1,\pi}^l)\pi(z_1^n)\mathrm{d}z_1^n, \tag{12a}$$
$$\Sigma_{12,\pi} = \begin{bmatrix} \Sigma_{12,\pi}^n \\ \Sigma_{12,\pi}^l \end{bmatrix}, \tag{12b}$$
$$\Sigma_{12,\pi}^n = \int (z_1^n - \mu_{1,\pi}^n)(h + H\tilde{\mu}_{1,\pi}^l - \mu_{2,\pi})^\mathsf{T}\pi(z_1^n)\mathrm{d}z_1^n, \tag{12c}$$
$$\Sigma_{12,\pi}^l = (\Sigma_{1,\pi}^{nl})^\mathsf{T}(\Sigma_{1,\pi}^n)^{-1}\Sigma_{12,\pi}^n + \tilde{\Sigma}_{1,\pi}^l \int H^\mathsf{T}\pi(z_1^n)\mathrm{d}z_1, \tag{12d}$$
$$\Sigma_{2,\pi} = \int \Big[(h + H\tilde{\mu}_{1,\pi}^l - \mu_{2,\pi})(h + H\tilde{\mu}_{1,\pi}^l - \mu_{2,\pi})^\mathsf{T} + H\tilde{\Sigma}_{1,\pi}^l H^\mathsf{T} + \Sigma_v\Big]\pi(z_1^n)\mathrm{d}z_1^n. \tag{12e}$$

Then, the statistical linear regression of z_2 *with respect to* $\pi(z_1^n, z_1^l)$ *is given by* $z_2 \approx \Phi z + \Gamma + \nu$, *where*

$$\Phi = \Sigma_{12,\pi}^\mathsf{T}\Sigma_{1,\pi}^{-1}, \tag{13a}$$
$$\Gamma = \mu_{2,\pi} - \Phi\mu_{1,\pi}, \tag{13b}$$
$$\Sigma_\nu = \Sigma_{2,\pi} - \Phi\Sigma_{1,\pi}\Phi^\mathsf{T}. \tag{13c}$$

Proof. The matrix Φ and vector Γ are chosen such that they minimize the error $e = h(z_1^n) + H(z_1^n)z_1^l + v - \Phi z_1 - \Gamma - \nu$ in the mean squared sense with respect to (10). This yields (13a)-(13b) (see [34] for details) where

$$\mu_{2,\pi} = \mathrm{E}_\pi\{h(z_1^n) + H(z_1^n)z_1^l + v\}, \tag{14a}$$
$$\Sigma_{2,\pi} = \mathrm{Cov}_\pi\{h(z_1^n) + H(z_1^n)z_1^l + v\}, \tag{14b}$$
$$\Sigma_{12,\pi} = \mathrm{Cov}_\pi\{z_1, h(z_1^n) + H(z_1^n)z_1^l + v\}. \tag{14c}$$

Furthermore, $\nu \sim \mathcal{N}(0, \Sigma_\nu)$ with

$$\Sigma_\nu = \mathrm{Cov}_\pi\{h(z_1^n) + H(z_1^n)z_1^l + v\} - \Phi\,\mathrm{Cov}_\pi\{z_1\}\Phi^\mathsf{T}. \tag{15}$$

In order to calculate the resulting expectations in (14)-(15), the linear substructure can now be exploited which yields the integrals

$$\begin{aligned}\mu_{2,\pi} &= \mathrm{E}_\pi\{\mathrm{E}_\pi\{h(z_1^n) + H(z_1^n)z_1^l + v \mid z_1^n\}\} \\ &= \int (h(z_1^n) + H(z_1^n)\tilde{\mu}_\pi^l)\pi(z_1^n)\mathrm{d}z_1^n,\end{aligned}$$

and similar for $\Sigma_{2,\pi}$, $\Sigma_{12,\pi}$, and Σ_ν, which leads to (12). □

Having calculated the linear approximation (12)–(13), the Gaussian approximation of the joint density $p(z_1, z_2)$ follows directly to be

$$p(z_1, z_2) \approx \mathcal{N}\left(\begin{bmatrix} z_1 \\ z_2 \end{bmatrix}; \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \begin{bmatrix} \Sigma_1 & \Sigma_{12} \\ \Sigma_{12}^\mathsf{T} & \Sigma_2 \end{bmatrix}\right) \tag{16}$$

with

$$\mu_2 = \Phi\mu_1 + \Gamma, \quad \Sigma_2 = \Phi\Sigma_1\Phi^\mathsf{T} + \Sigma_\nu, \quad \Sigma_{12} = \Sigma_1\Phi^\mathsf{T}. \tag{17}$$

Corollary 1. *If SLR is done with respect to the prior* $p(z_1)$ *(i.e.* $\pi(z_1) = p(z_1)$*), then* $\Sigma_{1,\pi} = \Sigma_1$ *and* $\mu_{1,\pi} = \mu_1$. *Thus* (17) *simplifies to* $\mu_2 = \mu_{2,\pi}$, $\Sigma_2 = \Sigma_{2,\pi}$, *and* $\Sigma_{12} = \Sigma_{12,\pi}$.

The expectations with respect to $\pi(z_1^n)$ in (12) are in general not analytically tractable. Instead, numerical integration schemes such as sigma-point methods can be used. Here, a set $\{z_1^{n,m}, w^m\}_{m=1}^M$ of M sigma-points $z_1^{n,m}$ and their weights w^m are calculated analytically based on the moments of z_1^n [3], [13]. Then, an arbitrary expectation with respect to the density $\pi(z_1^n)$ can be approximated as

$$\mathrm{E}_\pi\{h(z_1^n)\} = \int h(z_1^n)\pi(z_1^n)\mathrm{d}z_1 \approx \sum_{m=1}^M w^m h(z_1^{n,m}). \tag{18}$$

Choosing the sigma-points can be done according to different sigma-point rules such as the unscented transform, Gauss–Hermite quadratures, or spherical cubatures [3], [9].

B. Model 1 Prediction and Smoothing

For Model 1, given the density $\pi(x_{t-1}) = \mathcal{N}(x_{t-1}; \hat{x}_\pi, P_\pi)$, Lemma 1 is applied directly by choosing $z_1^n = x_{t-1}^n$, $z_1^l = x_{t-1}^l$, $z_2 = x_t$ and the covariance matrices accordingly. This yields

$$\bar{x}_{t|t-1} = \mathrm{E}_\pi\{f + A\tilde{x}_{t-1}^l\}, \tag{19a}$$
$$\bar{P}_{t|t-1} = \mathrm{E}_\pi\{(f + A\tilde{x}_{t-1}^l - \bar{x}_{t|t-1})(f + A\tilde{x}_{t-1}^l - \bar{x}_{t|t-1})^\mathsf{T}\} + \mathrm{E}_\pi\{A\tilde{P}_{t-1}^l A^\mathsf{T} + Q\}, \tag{19b}$$
$$\bar{C}_{t-1}^n = \mathrm{E}_\pi\{(x_{t-1}^n - \hat{x}_\pi^n)(f + A\tilde{x}_{t-1}^l - \bar{x}_{t|t-1}^n)^\mathsf{T}\}, \tag{19c}$$
$$\bar{C}_{t-1}^l = (P_\pi^{nl})^\mathsf{T}(P_\pi^n)^{-1}\bar{C}_{t-1}^n + \tilde{P}_{t-1}^l\,\mathrm{E}_\pi\{A\}^\mathsf{T}, \tag{19d}$$

for (12) where $\bar{C}_{t-1}^n = \begin{bmatrix}\bar{C}_{t-1}^{nn} & \bar{C}_{t-1}^{nl}\end{bmatrix}$ and similar for $\bar{C}_{t-1}^l$, see (9). The conditional mean $\tilde{x}_{t-1}^l$ and covariance $\tilde{P}_{t-1}$ are given by (11) with $\mu_{1,\pi} = \hat{x}_\pi$ and $\Sigma_{1,\pi} = P_\pi$. Furthermore, the regression coefficients obtained from SLR are

$$\Phi_{t-1} = (\bar{C}_{t-1})^\mathsf{T}(P_\pi)^{-1}, \tag{20a}$$
$$\Gamma_{t-1} = \bar{x}_{t|t-1} - \Phi_{t-1}\hat{x}_\pi, \tag{20b}$$
$$\Sigma_{\nu_t} = \bar{P}_{t|t-1} - \Phi_{t-1}P_\pi(\Phi_{t-1})^\mathsf{T}. \tag{20c}$$

1) Regular Smoothing: For regular smoothing SLR is performed with respect to $\pi(x_{t-1}^n) = p(x_{t-1}^n \mid y_{1:t-1})$. Hence, $\hat{x}_\pi = \hat{x}_{t-1|t-1}$ and $P_\pi = P_{t-1|t-1}$ in (19) and Corollary 1 applies. Thus

$$\hat{x}_{t|t-1} = \bar{x}_{t|t-1}, \quad P_{t|t-1} = \bar{P}_{t|t-1}, \quad \text{and} \quad C_{t-1} = \bar{C}_{t-1}. \tag{21}$$

2) Posterior Linearization Smoothing: For posterior linearization smoothing, $\pi(x_{t-1}^n) = p(x_{t-1}^n \mid y_{1:T})$. Hence, $\hat{x}_\pi = \hat{x}_{t-1|T}$ and $P_\pi = P_{t-1|T}$, while the predictive moments become

$$\hat{x}_{t|t-1} = \Phi_{t-1}\hat{x}_{t-1|t-1} + \Gamma_{t-1}, \tag{22a}$$
$$P_{t|t-1} = \Phi_{t-1}P_{t-1|t-1}(\Phi_{t-1})^\mathsf{T} + \Sigma_{\nu_t}, \tag{22b}$$
$$C_{t-1} = P_{t-1|t-1}(\Phi_{t-1})^\mathsf{T}. \tag{22c}$$

This leads to the prediction step summarized in Algorithm 1 where $\mathrm{SP}(\mu, \Sigma)$ denotes the generation of sigma-points with respect to the mean μ and covariance Σ.

C. Model 2 Prediction and Smoothing

For the second model, first note that the prediction for x_t^n is purely nonlinear and independent of x_{t-1}^l. Thus, $\bar{x}_{t|t-1}^n$, $\bar{P}_{t|t-1}^n$, and $\bar{C}_{t-1}^{nn}$ are found through SLR with respect to $\pi(x_{t-1}^n)$ and are given by

$$\bar{x}_{t|t-1}^n = \mathrm{E}_\pi\{f^n\}, \tag{23a}$$
$$\bar{P}_{t|t-1}^n = \mathrm{E}_\pi\left\{(f^n - \bar{x}_{t|t-1}^n)(f^n - \bar{x}_{t|t-1}^n)^\mathsf{T} + Q^n\right\}, \tag{23b}$$
$$\bar{C}_{t-1}^{nn} = \mathrm{E}_\pi\{(x_{t-1}^n - \tilde{x}_\pi^n)(f^n - \bar{x}_{t|t-1}^n)^\mathsf{T}\}. \tag{23c}$$

This yields the following approximation

$$\Phi_{t-1}^n = \bar{C}_{t-1}^{nn}(\tilde{P}_\pi^n)^{-1}, \tag{24a}$$
$$\Gamma_{t-1}^n = \bar{x}_{t|t-1}^n - \Phi_{t-1}^n\tilde{x}_\pi^n, \tag{24b}$$

Algorithm 1 Prediction Step for Model 1

1: Let $\{x_{t-1}^{n,m}, w^m\}_{m=1}^M \leftarrow \mathrm{SP}(\hat{x}_\pi^n, P_\pi^n)$
2: Calculate $\tilde{x}_{t-1}^{l,m}$ and $\tilde{P}_{t-1}^l$ according to (11) using $\hat{x}_\pi$ and P_π
3: Calculate

$$x_{t|t-1}^m = \begin{bmatrix} x_{t|t-1}^{n,m} \\ x_{t|t-1}^{l,m} \end{bmatrix} = \begin{bmatrix} f^n(x_{t-1}^{n,m}) \\ f^l(x_{t-1}^{n,m}) \end{bmatrix} + \begin{bmatrix} A^n(x_{t-1}^{n,m}) \\ A^l(x_{t-1}^{n,m}) \end{bmatrix} \tilde{x}_{t-1}^{l,m}$$

$$\bar{x}_{t|t-1} = \sum_{m=1}^M w^m x_{t|t-1}^m$$

$$\bar{P}_{t|t-1} = \sum_{m=1}^M w^m \Big[(x_{t|t-1}^m - \bar{x}_{t|t-1})(x_{t|t-1}^m - \bar{x}_{t|t-1})^\mathsf{T} + A(x_{t-1}^{n,m}) \tilde{P}_{t-1}^l A(x_{t-1}^{n,m})^\mathsf{T} + Q(x_{t-1}^{n,m}) \Big]$$

$$\bar{C}_{t-1}^n = \sum_{m=1}^M w^m (x_{t-1}^{n,m} - \hat{x}_\pi^n)(x_{t|t-1}^m - \bar{x}_{t|t-1})^\mathsf{T}$$

$$\bar{C}_{t-1}^l = (P_\pi^{nl})^\mathsf{T} (P_\pi^n)^{-1} \bar{C}_{t-1}^n + \tilde{P}_{t-1}^l \sum_{m=1}^M w^m A(x_{t-1}^{n,m})^\mathsf{T}$$

4: Calculate $\hat{x}_{t|t-1}$, $P_{t|t-1}$, and C_{t-1}:
- Regular prediction: (21)
- Posterior linearization prediction: (20) and (22)

$$\Sigma_{\nu_t}^n = \bar{P}_{t|t-1}^n - \Phi_{t-1}^n \tilde{P}_\pi^n (\Phi_{t-1}^n)^\mathsf{T}, \tag{24c}$$

where $\tilde{x}_\pi^n$ and $\tilde{P}_\pi^n$ denote the mean and covariance used in (23).

The prediction for x_t^l is found from Lemma 1 with $z_1^n = x_t^n$, $z_1^l = x_{t-1}^l$, $z_2 = x_t^l$. This gives

$$\bar{x}_{t|t-1}^l = \mathrm{E}_\pi\{f^l + A^l \tilde{x}_{t-1}^l\}, \tag{25a}$$

$$\bar{P}_{t|t-1}^l = \mathrm{E}_\pi\{(f^l + A^l \tilde{x}_{t-1}^l - \bar{x}_{t|t-1}^l)(f^l + A^l \tilde{x}_{t-1}^l - \bar{x}_{t|t-1}^l)^\mathsf{T}\} + \mathrm{E}_\pi\{A^l \tilde{P}_{t-1}^l (A^l)^\mathsf{T} + Q^l\}, \tag{25b}$$

$$\bar{P}_{t|t-1}^{nl} = \mathrm{E}_\pi\{(x_t^n - \hat{x}_\pi^n)(f^l + A^l \tilde{x}_{t-1}^l - \bar{x}_{t|t-1}^l)^\mathsf{T}\}, \tag{25c}$$

$$\bar{C}_{t-1}^{ll} = (P_\pi^{nl})^\mathsf{T} (P_\pi^n)^{-1} \bar{P}_{t|t-1}^{nl} + \tilde{P}_{t-1}^l \mathrm{E}_\pi\{A^l\}^\mathsf{T}. \tag{25d}$$

The coefficients of the linear approximation are then

$$\Phi_{t-1}^l = \begin{bmatrix} \bar{P}_{t|t-1}^{nl} \\ \bar{C}_{t-1}^{ll} \end{bmatrix}^\mathsf{T} (P_\pi)^{-1}, \tag{26a}$$

$$\Gamma_{t-1}^l = \bar{x}_{t|t-1} - \Phi_{t-1}^l \hat{x}_\pi, \tag{26b}$$

$$\Sigma_{\nu_t}^l = \bar{P}_{n|n-1} - \Phi_{t-1}^l P_\pi (\Phi_{t-1})^\mathsf{T}, \tag{26c}$$

and $\hat{x}_\pi$ and P_π are the moments used for SLR with respect to the dynamics of the linear state.

The two moments C_{t-1}^{ln} and C_{t-1}^{nl} do not arise directly from (23)–(26). However, given the approximation $x_t^n \approx \Phi_{t-1}^n x_{t-1}^n + \Gamma_{t-1}^n + \nu_t^n$, C_{t-1}^{ln} is found as follows

$$\begin{aligned} C_{t-1}^{ln} &\approx \mathrm{E}\{(x_{t-1}^l - \hat{x}_{t-1|t-1}^l)(x_{t-1}^n - \hat{x}_{t-1|t-1}^n)^\mathsf{T} (\Phi_{t-1}^n)^\mathsf{T}\} \\ &= (\Phi_{t-1}^n P_{t-1|t-1}^{nl})^\mathsf{T}. \end{aligned} \tag{27}$$

Similarly we have the approximation $x_t^l \approx \Phi_{t-1}^l x_{t,t-1} + \Gamma_{t-1}^l + \nu_t^l$ with $x_{t,t-1} = \begin{bmatrix} (x_t^n)^\mathsf{T} & (x_{t-1}^l)^\mathsf{T} \end{bmatrix}^\mathsf{T}$. Thus, C_{t-1}^{nl} becomes

$$\begin{aligned} C_{t-1}^{nl} &\approx \mathrm{E}\left\{ (x_{t-1}^n - \hat{x}_{t-1|t-1}^n) \left(\begin{bmatrix} x_t^n \\ x_{t-1}^l \end{bmatrix} - \begin{bmatrix} \hat{x}_{t|t-1}^n \\ \hat{x}_{t-1|t-1}^l \end{bmatrix} \right)^\mathsf{T} \right\} (\Phi_{t-1}^l)^\mathsf{T} \\ &= \begin{bmatrix} C_{t-1}^{nn} & P_{t-1|t-1}^{nl} \end{bmatrix} (\Phi_{t-1}^l)^\mathsf{T}. \end{aligned} \tag{28}$$

The resulting integrals are with respect to either $\pi(x_{t-1}^n)$ or $\pi(x_t^n)$ (c.f. (23)-(25)) and thus, they need to be calculated in two steps. In each step, a new set of sigma-points with respect to each of these densities has to be calculated. Fortunately, the sigma-points with respect to the density $\pi(x_t^n)$ will be re-used in the measurement update and thus, in practice, no additional sigma-points are needed.

1) Regular Smoothing: For predicting the nonlinear states x_t^n, the expectations in (23) are evaluated with respect to $\pi(x_{t-1}^n) = p(x_{t-1}^n \mid y_{1:t-1})$. Thus, $\tilde{x}_\pi^n = \hat{x}_{t-1|t-1}^n$ and $\tilde{P}_\pi^n = P_{t-1|t-1}^n$. Hence, Corollary 1 applies, and

$$\begin{aligned} &\hat{x}_{t|t-1}^n = \bar{x}_{t|t-1}^n, \quad P_{t|t-1}^n = \bar{P}_{t|t-1}^n, \quad C_{t-1}^{nn} = \bar{C}_{t-1}^{nn}, \\ &C_{t-1}^{ln} = (P_{t-1|t-1}^{nl})^\mathsf{T} (P_{t-1|t-1}^n)^{-1} (C_{t-1}^{nn})^\mathsf{T}. \end{aligned} \tag{29}$$

In the second step, the integrals (25) are calculated with respect to $\pi(x_t^n, x_{t-1}^l) = p(x_t^n, x_{t-1}^l \mid y_{1:t-1})$. Thus,

$$\hat{x}_\pi = \begin{bmatrix} \hat{x}_{t|t-1}^n \\ \hat{x}_{t-1|t-1}^l \end{bmatrix}, \quad \text{and} \quad P_\pi = \begin{bmatrix} P_{t|t-1}^n & (C_{t-1}^{ln})^\mathsf{T} \\ C_{t-1}^{ln} & P_{t-1|t-1}^l \end{bmatrix}, \tag{30}$$

and the de-correlation is as in (11) with $\mu_{1,\pi} = \hat{x}_\pi$ and $\Sigma_{1,\pi} = P_\pi$ from (30). Again, Corollary 1 applies and thus,

$$\begin{aligned} &\hat{x}_{t|t-1}^l = \bar{x}_{t|t-1}^l, \quad P_{t|t-1}^l = \bar{P}_{t|t-1}^l, \\ &P_{t|t-1}^{nl} = \bar{P}_{t|t-1}^{nl}, \quad C_{t-1}^{ll} = \bar{C}_{t-1}^{ll}, \\ &C_{t-1}^{nl} = \begin{bmatrix} C_{t-1}^{nn} & P_{t-1|t-1}^{nl} \end{bmatrix} \left(\begin{bmatrix} P_{t|t-1}^{nl} \\ C_{t-1}^{ll} \end{bmatrix} (P_{t|t-1}^n)^{-1} \right)^\mathsf{T}. \end{aligned} \tag{31}$$

2) Posterior Linearization Smoothing: In posterior linearization smoothing, predicting the nonlinear state is with respect to $\pi(x_{t-1}^n) = p(x_{t-1}^n \mid y_{1:T})$. Thus, $\tilde{x}_\pi^n = \hat{x}_{t-1|T}^n$, $\tilde{P}_\pi^n = P_{t-1|T}^n$, and

$$\hat{x}_{t|t-1}^n = \Phi_{t-1}^n \hat{x}_{t-1|t-1}^n + \Gamma_{t-1}^n, \tag{32a}$$

$$P_{t|t-1}^n = \Phi_{t-1}^n P_{t-1|t-1}^n (\Phi_{t-1}^n)^\mathsf{T} - \Sigma_{\nu_t}^n, \tag{32b}$$

$$C_{t-1}^{nn} = P_{t|t-1}^n (\Phi_{t-1}^n)^\mathsf{T}, \tag{32c}$$

with Φ_{t-1}^n, Γ_{t-1}^n, and $\Sigma_{\nu_t}^n$ as in (24), and C_{t-1}^{ln} as given in (27).

When predicting x_t^l, linearization is performed with respect to $\pi(x_t^n, x_{t-1}^l) = p(x_t^n, x_{t-1}^l \mid y_{1:T})$ and

$$\hat{x}_\pi = \begin{bmatrix} \hat{x}_{t|T}^n \\ \hat{x}_{t-1|T}^l \end{bmatrix}, \quad \text{and} \quad P_\pi = \begin{bmatrix} P_{t|T}^n & (E_{t-1}^{ln})^\mathsf{T} \\ E_{t-1}^{ln} & P_{t-1|T}^l \end{bmatrix}, \tag{33}$$

from which the $\tilde{x}_{t-1}^l$ and $\tilde{P}_{t-1}^l$ are calculated as in (11) with (33).

Having calculated Φ_{t-1}^l, Γ_{t-1}^l and $\Sigma_{\nu_t}^l$ according to (26), the predicted moments thus become

$$\hat{x}_{t|t-1}^l = \Phi_{t-1}^l \begin{bmatrix} \hat{x}_{t|t-1}^n \\ \hat{x}_{t-1|t-1}^l \end{bmatrix} + \Gamma_{t-1}^l, \tag{34a}$$

$$P_{t|t-1}^l = \Phi_{t-1}^l \begin{bmatrix} P_{t|t-1}^n & (C_{t-1}^{ln})^\mathsf{T} \\ C_{t-1}^{ln} & P_{t-1|t-1}^l \end{bmatrix} (\Phi_{t-1}^l)^\mathsf{T} + \Sigma_{\nu_t}^l, \tag{34b}$$

$$\begin{bmatrix} P_{t|t-1}^{nl} \\ C_{t-1}^{ll} \end{bmatrix} = \begin{bmatrix} P_{t|t-1}^n & (C_{t-1}^{ln})^\mathsf{T} \\ C_{t-1}^{ln} & P_{t-1|t-1}^l \end{bmatrix} (\Phi_{t-1}^l)^\mathsf{T}, \tag{34c}$$

and C_{t-1}^{nl} as in (28).

This yields the prediction step for Model 2 given in Algorithm 2.

D. Measurement Update

Since the structure of the measurement model is the same for both models (c.f. Eq. (3c) vs. (4c)), $\bar{y}_{t|t-1}$, $\bar{D}_t$, and $\bar{S}_t$ follow directly by applying Lemma 1 with $z_1^n = x_t^n$, $z_1^l = x_t^l$, and $z_2 = y_t$. This gives

$$\bar{y}_{t|t-1} = \mathrm{E}_\pi\{g + B\tilde{x}_t^l\}, \tag{35a}$$

$$\bar{D}_t^n = \mathrm{E}_\pi\{(x_t^n - \hat{x}_\pi^n)(g + B\tilde{x}_t^l - \bar{y}_{t|t-1})^\mathsf{T}\}, \tag{35b}$$

$$\bar{D}_t^l = (P_\pi^{nl})^\mathsf{T} (P_\pi^n)^{-1} \bar{D}_t^n + \tilde{P}_t^l \mathrm{E}_\pi\{B\}^\mathsf{T}, \tag{35c}$$

$$\bar{S}_t = \mathrm{E}_\pi\{(g + B\tilde{x}_t^l - \bar{y}_{t|t-1})(g + B\tilde{x}_t^l - \bar{y}_{t|t-1})^\mathsf{T}\} + \mathrm{E}_\pi\{B\tilde{P}_t^l B^\mathsf{T} + R\}. \tag{35d}$$

Algorithm 2 Prediction Step for Model 2

1: Let $\{x_{t-1}^{n,m}, w^m\}_{m=1}^M \leftarrow \mathrm{SP}(\tilde{x}_\pi^n, \tilde{P}_\pi^n)$
2: Calculate

$$x_{t|t-1}^{n,m} = f^n(x_{t-1}^{n,m})$$
$$\bar{x}_{t|t-1}^n = \sum_{m=1}^M w^m x_{t|t-1}^{n,m}$$
$$\bar{P}_{t|t-1}^n = \sum_{m=1}^M w^m \Big[(x_{t|t-1}^{n,m} - \bar{x}_{t|t-1}^n)(x_{t|t-1}^{n,m} - \bar{x}_{t|t-1}^n)^\mathsf{T} + Q^n(x_{t-1}^{n,m})\Big]$$
$$\bar{C}_{t-1}^{nn} = \sum_{m=1}^M w^m (x_{t-1}^{n,m} - \hat{x}_\pi^n)(x_{t|t-1}^{n,m} - \bar{x}_{t|t-1}^n)^\mathsf{T}$$

3: Calculate $\hat{x}_{t|t-1}^n$, $P_{t|t-1}^n$, C_{t-1}^{nn}, and C_{t-1}^{ln}
- Regular prediction: (29)
- Posterior linearization prediction: (24) and (32)

4: Let $\{x_t^{n,m}, w^m\}_{m=1}^M \leftarrow \mathrm{SP}(\hat{x}_\pi^n, P_\pi^n)$
5: Calculate $\tilde{x}_{t-1}^{l,m}$ and $\tilde{P}_{t-1}^l$ according to (11) using $\hat{x}_\pi$ and P_π
6: Calculate

$$x_{t|t-1}^{l,m} = f^l(x_t^{n,m}) + A^l(x_t^{n,m})\tilde{x}_{t-1}^{l,m}$$
$$\bar{x}_{t|t-1}^l = \sum_{m=1}^M w^m x_{t|t-1}^{l,m}$$
$$\bar{P}_{t|t-1}^l = \sum_{m=1}^M w^m \Big[(x_{t|t-1}^{l,m} - \bar{x}_{t|t-1}^l)(x_{t|t-1}^{l,m} - \bar{x}_{t|t-1}^l)^\mathsf{T} + A^l(x_t^{n,m})\tilde{P}_{t-1}^l A^l(x_t^{n,m})^\mathsf{T} + Q^l(x_t^{n,m})\Big]$$
$$\bar{P}_{t|t-1}^{nl} = \sum_{m=1}^M w^m (x_t^{n,m} - \hat{x}_\pi^n)(x_{t|t-1}^{l,m} - \bar{x}_{t|t-1}^l)^\mathsf{T}$$
$$\bar{C}_{t-1}^{ll} = (P_\pi^{nl})^\mathsf{T}(P_\pi^n)^{-1}\bar{P}_{t|t-1}^{nl} + \tilde{P}_{t-1}^l \sum_{m=1} w^m A^l(x_t^{n,m})^\mathsf{T}$$

7: Calculate $\hat{x}_{t|t-1}^l$, $P_{t|t-1}^l$, $P_{t|t-1}^{nl}$, C_{t-1}^{ll}, and C_{t-1}^{nl}
- Regular prediction: (31)
- Posterior linearization prediction: (26) and (32)

Furthermore, the conditional variables $\tilde{x}_t^l$ and $\tilde{P}_t^l$ are again calculated as in (11). From SLR, Φ_t, Γ_t, and Σ_{ν_t} are given by

$$\Phi_t = (\bar{D}_t)^\mathsf{T} P_\pi^{-1}, \tag{36a}$$
$$\Gamma_t = \bar{y}_{t|t-1} - \Phi_t \hat{x}_\pi, \tag{36b}$$
$$\Sigma_{\nu_t} = \bar{S}_t - \Phi_t P_\pi (\Phi_t)^\mathsf{T}. \tag{36c}$$

1) Regular Smoothing: In regular smoothing, $\pi(z_1) = p(x_t \mid y_{1:t-1})$, that is, $\hat{x}_\pi = \hat{x}_{t|t-1}$ and $P_\pi = P_{t|t-1}$. It follows that

$$\hat{y}_{t|t-1} = \bar{y}_{t|t-1}, \quad S_t = \bar{S}_t, \quad \text{and} \quad D_t = \bar{D}_t. \tag{37}$$

2) Posterior Linearization Smoothing: For posterior linearization smoothing, $\pi(x_t^n) = p(x_t^n \mid y_{1:T})$ with mean $\hat{x}_\pi = \hat{x}_{t|T}$ and covariance $P_\pi = P_{t|T}$. The moments $\hat{y}_{t|t-1}$, S_t, and D_t are thus

$$\hat{y}_{t|t-1} = \Phi_t \hat{x}_{t|t-1} + \Gamma_t, \tag{38a}$$
$$S_t = \Phi_t P_{t|t-1} (\Phi_t)^\mathsf{T} + \Sigma_{\nu_t}, \tag{38b}$$
$$D_t = P_{t|t-1} (\Phi_t)^\mathsf{T}, \tag{38c}$$

with Φ_t, Γ_t, and Σ_{ν_t} as in (36).

Finally, the measurement update is as summarized in Algorithm 3. For Model 2, the sigma-points from Step 4) in Algorithm 2 can be re-used and thus, Step 1) in Algorithm 3 can be omitted.

Algorithm 3 Measurement Update

1: Let $\{x_t^{n,m}, w^m\}_{m=1}^M \leftarrow \mathrm{SP}(\hat{x}_\pi^n, P_\pi^n)$
2: Calculate $\tilde{x}_t^{l,m}$ and $\tilde{P}_t^l$ according to (11) using $\hat{x}_\pi$ and P_π
3: Calculate

$$y_{t|t-1}^m = g(x_t^{n,m}) + B(x_t^{n,m})\tilde{x}_t^{l,m}$$
$$\bar{y}_{t|t-1} = \sum_{m=1}^M w^m y_{t|t-1}^m$$
$$\bar{D}_t^n = \sum_{m=1}^M w^m (x_t^{n,m} - \hat{x}_\pi^n)(y_{t|t-1}^m - \bar{y}_{t|t-1})^\mathsf{T}$$
$$\bar{D}_t^l = (P_\pi^{nl})^\mathsf{T}(P_\pi^n)^{-1}\bar{D}_t^n + \tilde{P}_t^l \sum_{m=1}^M w^m B(x_t^{n,m})^\mathsf{T}$$
$$\bar{S}_t = \sum_{m=1}^M w^m \Big[(y_{t|t-1}^m - \bar{y}_{t|t-1})(y_{t|t-1}^m - \bar{y}_{t|t-1})^\mathsf{T} + B(x_t^{n,m})\tilde{P}_t^l B(x_t^{n,m})^\mathsf{T} + R(x_t^{n,m})\Big]$$

4: Calculate $\hat{y}_{t|t-1}$, D_t, and S_t:
- Regular smoothing: (37)
- Posterior linearization smoothing: (36) and (38)

5: Calculate $\hat{x}_{t|t}$ and $P_{t|t}$ according to (8)

Algorithm 4 Rao–Blackwellized Gaussian Smoother

for $t = 1, \ldots, T$ **do** ▷ Filtering
  Calculate and store $\hat{x}_{t|t-1}$, $P_{t|t-1}$, and C_{t-1}:
  - Model 1: Algorithm 1
  - Model 2: Algorithm 2

  Calculate and store $\hat{x}_{t|t}$, $P_{t|t}$, and D_t (Algorithm 3)
end for
for $t = T-1, \ldots, 1$ **do** ▷ Smoothing
  Calculate $\hat{x}_{t|T}$, $P_{t|T}$, and E_t according to (6)
end for

E. Filtering, Smoothing, and Posterior Linearization Smoothing

Having developed the prediction and update steps for both models, the complete algorithm can now be formalized. First, filtering is achieved by simply alternating between prediction and measurement update at each time step over the whole dataset. After filtering, smoothing is performed by a backward sweep over the filtered data, implementing the Rauch–Tung–Striebel smoothing equations. This yields the Rao–Blackwellized Gaussian smoother in Algorithm 4.

Finally, posterior linearization smoothing is achieved by first running Algorithm 4 to obtain $p^0(x_t \mid y_{1:T})$, followed by re-linearization and iteratively improving the posterior approximation (Algorithm 5).

F. Computational Complexity

One of the main reasons for considering Rao–Blackwellization in the context of Gaussian filtering and smoothing is the reduction of the computational complexity when the dimension of the conditionally linear subspace is large. Since the proposed methods are based on Lemma 1, we start by analyzing the computational complexity of regular SLR and Rao–Blackwellized SLR with respect to the linear subspace dimension N_1^l, which is provided in Lemma 2 below.

Lemma 2 (Computational Complexity of Rao–Blackwellized Statistical Linear Regression)**.** *Let $z_2 = h(z_1^n) + H(z_1^n)z_1^l + v$ be a conditionally affine transformation of the Gaussian random variable $z_1 = \begin{bmatrix}(z_1^n)^\mathsf{T} & (z_1^l)^\mathsf{T}\end{bmatrix}^\mathsf{T}$, where $z_1^n \in \mathbb{R}^{N_1^n}$, $z_1^l \in \mathbb{R}^{N_1^l}$, $z_2 \in \mathbb{R}^{N_2}$, and $N_1 = N_1^n + N_1^l$. Then the asymptotic (in N_1^l) computational*

Algorithm 5 Rao–Blackwellized Posterior Linearization Smoother

1: Run Algorithm 4 to obtain $p^0(x_t \mid y_{1:T})$ and set $i \leftarrow 0$
2: **do**
3: **for** $t = 1, \ldots, T$ **do** ▷ Filtering
4: Calculate and store $\hat{x}_{t|t-1}^{i+1}$, $P_{t|t-1}^{i+1}$, and C_{t-1}^{i+1}:
• Model 1: Algorithm 1
• Model 2: Algorithm 2
5: Calculate and store $\hat{x}_{t|t}^{i+1}$, $P_{t|t}^{i+1}$, and D_t^{i+1} (Algorithm 3)
6: **end for**
7: **for** $t = T-1, \ldots, 1$ **do** ▷ Smoothing
8: Calculate and store $\hat{x}_{t|T}^{i+1}$, $P_{t|T}^{i+1}$, and E_t^{i+1} according to (6)
9: **end for**
10: Set $i \leftarrow i + 1$
11: **while** Not converged

Table I: Comparison of the asymptotic computational complexity for SLR and Rao–Blackwellized SLR (RB-SLR).

Calculation	SLR $\mathcal{O}(\cdot)$	RB-SLR $\mathcal{O}(\cdot)$
Sigma-points	N_1^3	$(N_1^n)^3$
Orthogonalization	n/a	$MN_1^l N_1^n + (N_1^l)^2 N_1^n$
z_2^m, Σ_v^m	$M(\mathcal{C}_h + \mathcal{C}_H + \mathcal{C}_\Sigma)$ $+MN_2N_1^l$	$M(\mathcal{C}_h + \mathcal{C}_H + \mathcal{C}_\Sigma)$ $+MN_2N_1^l$
$\mu_{2,\pi}$	MN_2	MN_2
$\Sigma_{2,\pi}$	MN_2^2	$M(N_1^l)^2N_2 + MN_1^lN_2^2$
$\Sigma_{12,\pi}$	MN_1N_2	$MN_1^lN_2 + (N_1^l)^2N_2$
Φ	N_1^3	N_1^3
Γ	$N_1^2N_2$	$N_1^2N_2$
Σ_ν	$N_1^2N_2 + N_1N_2^2$	$N_1^2N_2 + N_1N_2^2$
μ_2	$N_1^2N_2$	$N_1^2N_2$
Σ_2	$N_1^2N_2 + N_1N_2^2$	$N_1^2N_2 + N_1N_2^2$
Σ_{12}	$N_1^2N_2$	$N_1^2N_2$

complexity of calculating the transformed moments μ_2, Σ_{12}, and Σ_2 using M sigma-points scales according to $\mathcal{O}(MN_1^lN_2 + (N_1^l)^3)$ for regular SLR and $\mathcal{O}((N_1^l)^2N_2 + (N_1^l)^3)$ for Rao–Blackwellized SLR.

Proof. Regular SLR has the following main steps: 1) Calculating the sigma-points, 2) evaluating the functions $h(z_1^n) + H(z_1^n)z_1^l$ and $\Sigma_v(z_1^n)$ (M times each), 3) calculating sums over M terms to obtain $\mu_{2,\pi}$, $\Sigma_{12,\pi}$, and $\Sigma_{2,\pi}$, 4) calculating the linear approximation Φ, Γ, Σ_ν, and 5) calculating the moments μ_2, Σ_2, Σ_{12}. Rao–Blackwellized SLR has the additional step of calculating the Gram–Schmidt orthogonalization (11). Furthermore, the sums for calculating $\Sigma_{2,\pi}$ and $\Sigma_{12,\pi}$ include more terms.

The asymptotic complexity for each of these steps is shown in Table I, where $\mathcal{C}_h$, $\mathcal{C}_H$, and $\mathcal{C}_\Sigma$ denote the complexity of evaluating the functions $h(\cdot)$, $H(\cdot)$, and $\Sigma_v(\cdot)$. First, note that calculating the sigma-points is independent of N_1^l for Rao–Blackwellized SLR. Second, the complexity of the sums for calculating $\mu_{2,\pi}$, $\Sigma_{2,\pi}$, and $\Sigma_{12,\pi}$ is dominated by the calculation of the covariance matrices. For regular SLR, this scales according to $MN_2^2 + MN_1N_2$ where the first term comes from the outer product involved in calculating $\Sigma_{2,\pi}$ and the second term from calculating $\Sigma_{12,\pi}$. Noting that M scales at least linearly in N_1^l and that N_2 may also scale linearly in N_1^l (for the prediction step), it follows that the overall complexity of implementing the sums scales according to $MN_1^lN_2$.

For Rao–Blackwellized SLR (Lemma 1), the most expensive operations in implementing the integrals (12) using sigma-points are again computing $\Sigma_{2,\pi}$ and $\Sigma_{12,\pi}$. Here, the complexity is dominated by the product $H(z_1^{n,m})\tilde{\Sigma}_{1,\pi}^l H(z_1^{n,m})^\mathsf{T}$ which requires $M(N_1^l)^2N_2 + MN_1^lN_2^2$ operations. Since M is independent of N_1^l in this case, this scales according to $(N_1^l)^2N_2$ at most.

Implementing (13) and (17) requires the same computations, irrespective of whether Rao–Blackwellization is used or not. Here, the most complex operations are calculating the inverse (or Cholesky factorization) of an $N_1 \times N_1$ matrix when calculating Φ (N_1^3 operations) and calculating $\Phi\Sigma_1\Phi^\mathsf{T}$ when calculating Σ_2 ($N_1^2N_2 + N_1N_2^2$). This is in addition to the cost for calculating the sums. Thus, the overall computational complexity becomes $\mathcal{O}(MN_1^lN_2 + (N_1^l)^3)$ for regular SLR and $\mathcal{O}((N_1^l)^2N_2 + (N_1^l)^3)$ for Rao–Blackwellized SLR. □

Given the asymptotic computational complexity for regular and Rao–Blackwellized SLR in Lemma 2, Corollary 2 is readily found.

Corollary 2. *The asymptotic computational complexity for sigma-point-based SLR where the number of sigma-points is an affine function $M = \alpha N_1^l + \beta$ is $\mathcal{O}((N_1^l)^2N_2 + (N_1^l)^3)$ for both regular and Rao–Blackwellized SLR. Then, the ratio between nonlinear and linear states as well as the cost for evaluating the functions $h(\cdot)$, $H(\cdot)$, and $\Sigma_v(\cdot)$ determine which method is faster. Conversely, if the number of sigma-points increases faster than linear, the asymptotic complexity of Rao–Blackwellized SLR is always lower.*

It follows that for the third order unscented transform ($M = 2N + 1$), the constants determine which method is faster, whereas for higher order methods, such as the fifth order unscented transform ($M = 2N^2 + 1$) or the Gauss–Hermite quadrature ($M = p^N$), Rao–Blackwellization is beneficial for systems with high-dimensional linear subspaces.

Finally, the overall complexity of the proposed methods is given in Lemma 3.

Lemma 3 (Computational Complexity of Rao–Blackwellized Gaussian Smoothing)**.** *For Algorithm 4 and Algorithm 5, the asymptotic computational complexity with respect to N_x^l is $\mathcal{O}((N_x^l)^3)$ and the computational complexity for the corresponding non-Rao–Blackwellized smoothers is $\mathcal{O}(M(N_x^l)^2 + (N_x^l)^3)$*

Proof. First, note that the prediction for Model 1 is a direct application of Lemma 1 with $N_1 = N_2 = N_x$ with no additional operations. Hence, the total computational complexity of Algorithm 1 readily follows to be $\mathcal{O}((N_1^l)^3)$. For Algorithm 2 (prediction for Model 2), the nonlinear state is predicted in a regular prediction step with $N_1 = N_2 = N_1^n$ which is independent of N_1^l and thus has constant cost. For predicting the linear state, Lemma 1 is applied with $N_1 = N_x$ and $N_2 = N_x^l$ and thus, the complexity is again $\mathcal{O}((N_x^l)^3)$, The measurement update (Algorithm 3) is again a direct application of Lemma 1 with $N_1 = N_x$ and $N_2 = N_y$. Hence, SLR scales according to $\mathcal{O}((N_x^l)^2)$ for regular filtering while it scales according to $\mathcal{O}((N_x^l)^3)$ for posterior linearization filtering/smoothing (due to (13) and (17) which scale according to $\mathcal{O}((N_x^l)^3)$). Furthermore, the measurement update (8) is done, which has the complexity $\mathcal{O}(N_x^2)$ due to the covariance update. The backward pass for smoothing just adds the Rauch–Tung–Striebel smoothing calculations in (6). Here, the most expensive operations are calculating the gain G_t and the covariance matrices $P_{t|T}$ and E_t which all scale according to $\mathcal{O}(N_x^3)$. Thus, it follows that the overall asymptotic computational complexity in N_x^l for Rao–Blackwellized Gaussian smoothing for Model 1 as well as Model 2 is $\mathcal{O}((N_x^l)^3)$.

Similarly, for non-Rao–Blackwellized smoothing, it follows from Lemma 2 that the prediction step for both Model 1 and Model 2 scale according to $\mathcal{O}(M(N_x^l)^2 + (N_x^l)^3)$. Furthermore, the regular measurement update scales as $\mathcal{O}(MN_x^l + (N_x^l)^3)$, while the smoothing recursion is the same as for Rao–Blackwellized smoothing (i.e. $\mathcal{O}(N_x^3)$). Thus, the overall computational complexity for non-Rao–Blackwellized smoothing is $\mathcal{O}(M(N_x^l)^2 + (N_x^l)^3)$. □

V. Numerical Illustrations

A. Setup

In order to illustrate the proposed method, a system of harmonic oscillators with unknown, time-varying frequency is considered. This system allows for simple extension to higher order linear state-spaces by just considering higher order harmonics. The model is given by

$$x_t^n = x_{t-1}^n + q_t^n, \tag{39a}$$
$$x_t^l = A^l(x_{t-1}^n)x_{t-1}^l + q_t^l, \tag{39b}$$
$$y_t = Bx_{t-1}^l + r_t. \tag{39c}$$

The matrices A^l and B in (39) are

$$F(\omega) = \begin{bmatrix} \cos(\omega T_s) & -\sin(\omega T_s) \\ \sin(\omega T_s) & \cos(\omega T_s) \end{bmatrix},$$
$$A^l(x_t^n) = \operatorname{blkdiag}(F(x_t^n), F(2x_t^n), \ldots, F(N_x^l/2x_t^n)),$$
$$B = \begin{bmatrix} 1 & 0 & 1 & 0 & \ldots & 1 & 0 \end{bmatrix},$$

and $T_s = 0.05\,\mathrm{s}$ is the sampling time.

The process and measurement noise covariances are chosen as $Q = 0.01 I_{N_x^l+1}$ and $R = 0.1$, respectively. The number of harmonics is varied from 1 to 5, which yields $N_x^l = 2, 4, \ldots, 10$. For each case, $T = 100$ samples are generated and 100 Monte Carlo runs are performed. As performance measures, the time-averaged root mean squared error (RMSE) is calculated over all Monte Carlo simulations. Additionally, the computational time per sample is measured. The algorithms are implemented in m-code in Matlab and the simulations are run on a 3.4 GHz 3rd generation Intel Xeon E3 CPU with 16 GB RAM. We compare the following five filtering methods:

- An unscented Kalman filter (UKF) [35],
- a Rao–Blackwellized unscented Kalman filter (RB-UKF),
- a Gauss–Hermite filter (GHF) of order $p = 3$,
- a Rao–Blackwellized Gauss–Hermite filter (RB-GHF),
- a Rao–Blackwellized particle filter [16] with $M_f = 250$ particles (RB-PF),

as well as the corresponding smoothers, that is,

- an unscented Rauch–Tung–Striebel smoother (URTSS) [6],
- a Rao–Blackwellized unscented Rauch–Tung–Striebel smoother (RB-URTSS),
- a Gauss–Hermite smoother (GHS),
- a Rao–Blackwellized Gauss–Hermite smoother (RB-GHS),
- a Rao–Blackwellized forward-filtering backward-simulation particle smoother [17] with $M_f = 250$ filter and $M_s = 100$ smoother particles (RB-FFBSi).

B. Results and Discussion

Fig. 1 (top) illustrates the RMSE of the compared methods as a function of N_x^l. As it can be seen, all filters perform roughly equally well, with a slightly lower RMSE for the UKF and RB-PF for higher dimensions compared to the remaining filters. However, there is no difference between the GHF and the RB-GHF. Furthermore, Fig. 1 (bottom) depicts the time required for one complete filter update (time and measurement update). Here, the differences are obvious and striking. The computational load increases exponentially for the GHF and cubically for the RB-GHF, while it increases cubically for both the UKF as well as the RB-UKF but at a slower rate for the latter.

The results for smoothing are shown in Fig. 2. Similar to filtering, there is no significant difference in the RMSE for the different smoothers (Fig. 2, top) with the Gaussian smoothers performing slightly better than the particle smoother. The significant difference is again in the computational time (Fig. 2, bottom). For the Gaussian

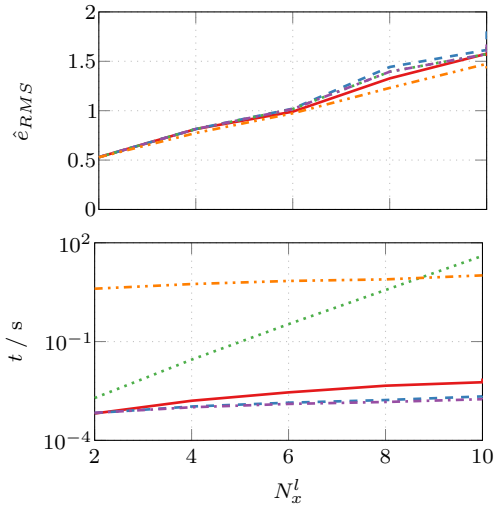

Figure 1: Time-averaged RMSE (top) and average filter update time per sample (bottom) for the UKF (——), RB-UKF (– – –), GHF (·····), RB-GHF (–·–·–), and the RB-PF (–··–··–).

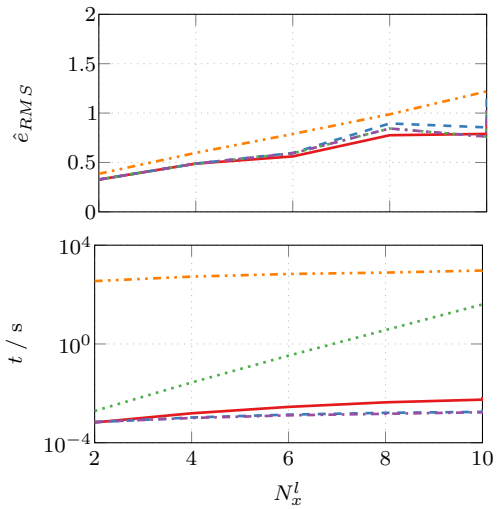

Figure 2: Time-averaged RMSE (top) and average smoother update time per sample (bottom) for the URTSS (——), RB-URTSS (– – –), GHS (·····), RB-GHS (–·–·–), and the RB-FFBSi (–··–··–).

smoothers, the results from filtering carry over since only little overhead is added by the backward recursion (see Algorithm 4).

As shown in Section IV-F, the significant increase in computational time observed for the GHF and UKF are directly related to the number of sigma-points. For the Gauss–Hermite quadrature used in the GHF, the number of sigma-points scales as $\mathcal{O}(p^{N_x})$ and for the third order unscented transform used here, it scales as $\mathcal{O}(2N_x+1)$. By using Rao–Blackwellization, the number of sigma-points only depends on the size of the nonlinear subspace N_x^n, which is constant in this example. This yields the reduction of computational complexity from exponential to cubic for the RB-GHF and RB-GHS. For the unscented filter and smoother, the scaling remains cubic, but the Rao–Blackwellized version is still faster. Closer inspection showed that evaluating A^l is one of the most expensive operations in this case (evaluating the sine and cosine functions). Thus, since the Rao–Blackwellized smoother requires less function evaluations, it is faster in this case.

The results also indicate that the RMSE is not affected significantly by the Rao–Blackwellization for the compared methods. Furthermore,

note that while the RB-PF is slightly superior in terms of RMSE, the RB-FFBSi is not. This is due to a particular combination of the considered model and the way the backward simulation smoother works: Frequency estimation schemes (e.g. phase-locked loops) require some time to converge to the correct frequency. For particle methods, this means that the individual state trajectories during forward filtering may be relatively far from the true trajectory. When performing backward simulation, the RB-FFBSi smoother is limited to sample from these trajectories and can not do better than the best trajectory. The Gaussian smoothers on the other hand re-estimate the smoothed state freely, which turns out to be better in this case.

The dimension of the nonlinear state might be different in the dynamic and observation models, as in the example considered in this section. In such cases, the algorithms may be further improved by only considering the respective nonlinear subspaces and do exact (conditional) updates for the remaining states. Finally, we have assumed that the system does not suffer from delayed or out-of-sequence measurements. If such delays are present, the method can be extended by, for example, using approaches similar to [36] or [37].

VI. Conclusions

In this paper, Gaussian smoothers with analytically tractable conditional linear substructures were developed and their computational complexity was analyzed. The framework is general in the sense that it can be used together with any sigma-point based Gaussian filtering/smoothing algorithm such as the unscented Rauch–Tung–Striebel smoother or Gauss–Hermite quadrature-based smoothers. The simulations showed that a significant performance gain in terms of computational efficiency can be achieved, even for low-dimensional linear subspaces. Hence, it is useful to use Rao–Blackwellization not only to reduce the computational burden but also to enable the usage of otherwise computationally prohibitive methods such as Gauss–Hermite quadratures for higher dimensional systems.

References

[1] K. Ito and K. Xiong, "Gaussian filters for nonlinear filtering problems," *IEEE Transactions on Automatic Control*, vol. 45, no. 5, pp. 910–927, May 2000.

[2] S. Särkkä and J. Hartikainen, "On Gaussian optimal smoothing of non-linear state space models," *IEEE Transactions on Automatic Control*, vol. 55, no. 8, pp. 1938–1941, August 2010.

[3] S. Särkkä, *Bayesian Filtering and Smoothing*. Cambridge University Press, 2013.

[4] J. McNamee and F. Stenger, "Construction of fully symmetric numerical integration formulas of fully symmetric numerical integration formulas," *Numerische Mathematik*, vol. 10, no. 4, pp. 327–344, 1967.

[5] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 401–422, 2004.

[6] S. Särkkä, "Unscented Rauch-Tung-Striebel smoother," *Automatic Control, IEEE Transactions on*, vol. 53, no. 3, pp. 845–849, April 2008.

[7] D. Tenne and T. Singh, "The higher order unscented filter," in *American Control Conference (ACC)*, vol. 3, June 2003, pp. 2441–2446.

[8] Y. Wu, D. Hu, M. Wu, and X. Hu, "A numerical-integration perspective on Gaussian filters," *IEEE Transactions on Signal Processing*, vol. 54, no. 8, pp. 2910–2921, August 2006.

[9] J. Kokkala, A. Solin, and S. Särkkä, "Sigma-point filtering and smoothing based parameter estimation in nonlinear dynamic systems," *Journal of Advances in Information Fusion*, vol. 11, no. 1, pp. 15–30, 2016.

[10] I. Arasaratnam and S. Haykin, "Cubature Kalman filters," *IEEE Transactions on Automatic Control*, vol. 54, no. 6, pp. 1254–1269, June 2009.

[11] ——, "Cubature Kalman smoothers," *Automatica*, vol. 47, no. 10, pp. 2245–2250, 2011.

[12] T. Lefebvre, H. Bruyninckx, and J. De Schutter, "Kalman filters for non-linear systems: a comparison of performance," *International Journal of Control*, vol. 77, no. 7, pp. 639–653, 2004.

[13] M. Roth, G. Hendeby, and F. Gustafsson, "Nonlinear Kalman filters explained: A tutorial on moment computations and sigma point methods," *Journal of Advances in Information Fusion*, vol. 11, no. 1, pp. 47–70, 2016.

[14] N. Shephard, "Partial non-Gaussian state space," *Biometrika*, vol. 81, no. 1, pp. 115–131, 1994.

[15] A. Doucet, N. De Freitas, K. Murphy, and S. Russell, "Rao–Blackwellised particle filtering for dynamic Bayesian networks," in *16th Conference on Uncertainty in Artificial Intelligence*, 2000, pp. 176–183.

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

[17] F. Lindsten, P. Bunch, S. Särkkä, T. B. Schön, and S. J. Godsill, "Rao–Blackwellized particle smoothers for conditionally linear Gaussian models," *IEEE Journal of Selected Topics in Signal Processing*, vol. 10, no. 2, pp. 353–365, March 2016.

[18] M. R. Morelande and B. Ristic, "Reduced sigma point filtering for partially linear models," in *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, May 2006.

[19] M. R. Morelande and B. Moran, "An unscented transformation for conditionally linear models," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, April 2007, pp. 1417–1420.

[20] F. Beutler, M. F. Huber, and U. D. Hanebeck, "Gaussian filtering using state decomposition methods," in *Information Fusion, 2009. FUSION '09. 12th International Conference on*, July 2009, pp. 579–586.

[21] A. F. García-Fernández, M. R. Morelande, and J. Grajal, "Truncated unscented Kalman filtering," *IEEE Transactions on Signal Processing*, vol. 60, no. 7, pp. 3372–3386, July 2012.

[22] M. F. Huber, F. Beutler, and U. D. Hanebeck, "Semi-analytic Gaussian assumed density filter," in *Proceedings of the 2011 American Control Conference*, June 2011, pp. 3006–3011.

[23] H. Singer, "Conditional Gauss-Hermite filtering with application to volatility estimation," *IEEE Transactions on Automatic Control*, vol. 60, no. 9, pp. 2476–2481, September 2015.

[24] G. Chang, "Marginal unscented Kalman filter for cross-correlated process and observation noise at the same epoch," *IET Radar, Sonar Navigation*, vol. 8, no. 1, pp. 54–64, January 2014.

[25] J.-O. Nilsson, "Marginalized Bayesian filtering with Gaussian priors and posteriors," *ArXiv e-prints*, July 2016, arXiv:1603.06462v2.

[26] A. F. García-Fernández, L. Svensson, and S. Särkkä, "Iterated posterior linearisation smoother," *IEEE Transactions on Automatic Control*, vol. 62, no. 4, pp. 2056–2063, April 2017.

[27] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P.-J. Nordlund, "Particle filters for positioning, navigation, and tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 425–437, February 2002.

[28] X. R. Li and V. P. Jilkov, "Survey of maneuvering target tracking: Part I. Dynamic models," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 39, no. 4, pp. 1333–1364, 2003.

[29] R. Hostettler, W. Birk, and M. Lundberg Nordenvaad, "Joint vehicle trajectory and model parameter estimation using road side sensors," *IEEE Sensors Journal*, vol. 15, no. 9, pp. 5075–5086, September 2015.

[30] R. Chen and J. S. Liu, "Mixture Kalman filters," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 62, no. 3, pp. 493–508, 2000.

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

[32] S. Saha, G. Hendeby, and F. Gustafsson, "Mixture Kalman filters and beyond," in *Current Trends in Bayesian Methodology with Applications*, S. K. Upadhyay, U. Singh, D. K. Dey, and A. Loganathan, Eds. Chapman and Hall/CRC, 2015, ch. 26, pp. 537–562.

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

[34] I. Arasaratnam, S. Haykin, and R. J. Elliott, "Discrete-time nonlinear filtering algorithms using Gauss-Hermite quadrature," *Proceedings of the IEEE*, vol. 95, no. 5, pp. 953–977, May 2007.

[35] E. A. Wan and R. Van Der Merwe, "The unscented Kalman filter for nonlinear estimation," in *Adaptive Systems for Signal Processing, Communications, and Control Symposium*, 2000, pp. 153–158.

[36] K. Berntorp, A. Robertsson, and K.-E. Årzén, "Rao–Blackwellized particle filters with out-of-sequence measurement processing," *IEEE Transactions on Signal Processing*, vol. 62, no. 24, pp. 6454–6467, December 2014.

[37] Q. Gao and N. Olgac, "Bounds of imaginary spectra of LTI systems in the domain of two of the multiple time delays," *Automatica*, vol. 72, pp. 235–241, 2016.