



Iterated Posterior Linearization Smoother

Downloaded from: <https://research.chalmers.se>, 2024-12-04 09:40 UTC

Citation for the original published paper (version of record):

Garcia, A., Svensson, L., Särkkä, S. (2017). Iterated Posterior Linearization Smoother. IEEE Transactions on Automatic Control, 62(4): 2056-2063. <http://dx.doi.org/10.1109/TAC.2016.2592681>

N.B. When citing this work, cite the original published paper.

© 2017 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, or reuse of any copyrighted component of this work in other works.

(article starts on next page)

Iterated posterior linearisation smoother

Ángel F. García-Fernández, Lennart Svensson, Simo Särkkä

Abstract—This paper considers the problem of Bayesian smoothing in nonlinear state-space models with additive noise using Gaussian approximations. Sigma-point approximations to the general Gaussian Rauch–Tung–Striebel smoother are widely used methods to tackle this problem. These algorithms perform statistical linear regression (SLR) of the nonlinear functions considering only the previous measurements. We argue that SLR should be done taking all measurements into account. We propose the iterated posterior linearisation smoother (IPLS), which is an iterated algorithm that performs SLR of the nonlinear functions with respect to the current posterior approximation. The algorithm is demonstrated to outperform conventional Gaussian nonlinear smoothers in two numerical examples.

Index Terms—Bayesian smoothing, Rauch–Tung–Striebel smoothing, iterated smoothing, sigma-points, statistical linear regression.

I. INTRODUCTION

Smoothing consists of estimating past states of a process that evolves with time from a sequence of noisy measurements [1]. It has applications in different fields, such as target tracking, audio signal processing or navigation [2]–[4]. In the Bayesian framework, all information of interest about the process is given by the probability density function (PDF) of the sequence of states given all measurements. We refer to this PDF as the posterior PDF. From the posterior PDF, we can also obtain the marginal PDF of the state at a certain time step, which contains all information about the process at this particular time step.

If the system is linear and Gaussian, the posterior PDF admits a closed-form Gaussian expression and marginals can be computed by the Rauch–Tung–Striebel (RTS) or the two-filter smoothers [5], [6]. In contrast, if the system is nonlinear/non-Gaussian, there is no closed-form equation and we need approximations. In a general system, we can use sequential Monte Carlo methods for this purpose [7]. If the posterior PDF is unimodal, Gaussian PDFs can instead be used to obtain a relatively accurate approximation to the posterior with a much lower computational complexity than sequential Monte Carlo methods. This is the focus of this paper in which we propose an iterative Gaussian smoother for systems with additive noise.

General Gaussian RTS smoothers are a family of algorithms that obtain Gaussian approximations of the marginals by first performing filtering and then smoothing [1], [8]. While

filtering, these algorithms calculate/approximate the same moments as the (non-linear) Kalman filter (KF) [9] plus the cross-covariance between the states at the current and next time steps. Based on these moments and the filtering PDFs, we can obtain the marginal posterior PDFs by a backward recursion. Depending on the method for approximating the moments, we obtain different smoothers, for instance, extended RTS smoother (analytical linearisation) [1], unscented RTS smoother (unscented transform) [10] or cubature RTS smoother (cubature rule) [11].

As we indicate in this paper, Gaussian RTS smoothers implicitly perform an enabling approximation: all nonlinear functions are approximated by an affine function plus Gaussian noise. Once this enabling approximation is made, we obtain the PDFs of interest in closed-form. Sigma-point Gaussian RTS smoothers perform the enabling approximation while filtering using statistical linear regression (SLR). Given a function and a PDF, SLR provides the affine function that minimises the mean square error and the covariance matrix of the linearisation error [12]. However, the above-mentioned smoothers perform SLR with respect to PDFs that do not take into account future measurements. The fact that all available measurements are not employed in the enabling approximation indicates that performance can be improved.

One algorithm that takes into account all measurements in the enabling approximation is the iterated extended Kalman smoother (IEKS) [13]. It uses analytical linearisation of the nonlinear functions at the maximum a posteriori (MAP) estimate of all states of the process, obtained by a Gauss–Newton method. Consequently, it does not use SLR and sets the Gaussian noise of the enabling approximation to zero, which is not necessary. Moreover, IEKS requires the gradients of the functions and makes use of the extended KF (EKF) or iterated EKF [14], which are usually outperformed by sigma-point Kalman filters [9]. Another iterated algorithm that can be used for smoothing is expectation-propagation [15]. However, the integrals needed in expectation-propagation for nonlinear systems cannot be accurately approximated by sigma-points and Monte Carlo methods should be used, which we aim to avoid due to their higher computational burden.

In this paper, we develop a Gaussian smoother that selects the enabling approximation by performing SLR of the nonlinear functions with respect to the posterior PDF, which depends on all measurements. By doing so, we combine the strengths of the approximation techniques used in sigma-point Gaussian filters with the ability to incorporate all the available information, as in iterated algorithms. The proposed algorithm can be seen as an extension of the ideas of the posterior linearisation filter (PLF) [16] to a smoothing context. The resulting posterior linearisation smoother (PLS) is intractable so, we approximate

Ángel F. García-Fernández is with the Department of Electrical and Computer Engineering, Curtin University, Perth, WA 6102, Australia (email: angel.f.garcia.fernandez@gmail.com). Lennart Svensson is with the Department of Signals and Systems, Chalmers University of Technology, SE-412 96 Gothenburg, Sweden (email: lennart.svensson@chalmers.se). Simo Särkkä is with the Department of Electrical Engineering and Automation, Aalto University, 02150 Espoo, Finland (email: simo.sarkka@aalto.fi).

it by the iterated PLS (IPLS), which performs iterated SLR with respect to the latest posterior approximation. This idea is similar to the iterated approximation of the PLF: the iterated PLF (IPLF) [16]. An important difference between the IPLF and IPLS is that the IPLS relinearises both the dynamic and measurement functions whereas the IPLF only relinearises the measurement functions. As the IPLF, the IPLS can be seen, under certain approximations, as a minimisation of the Kullback–Leibler divergence (KLD) of the joint density of the states and the variables obtained by passing the states through the dynamic and measurement functions.

Interestingly, the IPLS also differs from RTS smoothers because the approximation of the filtering posterior can improve with the iterations. Therefore, we can use the ideas behind the IPLS to design a new filter: the L -scan IPLF. It consists of relinearising the dynamic and measurement functions in the last L time steps with the objective of improving the filtering posterior approximation.

The remainder of the paper is organised as follows. The problem formulation and some background work are given in Section II. Section III describes the PLS, the IPLS and the L -scan IPLF. A derivation of the IPLS based on an approximate KLD minimisation is provided in Section IV. Simulation results are shown in Section V. Finally, conclusions are drawn in Section VI.

II. PROBLEM FORMULATION

We seek to estimate the state sequence $x_{0:N} = (x_0, \dots, x_N)$ of a process given a sequence of N measurements $z_{1:N}$, where $x_k \in \mathbb{R}^{n_x}$ and $z_k \in \mathbb{R}^{n_z}$. The relations between the states at different time steps and the measurements are given by the dynamic and measurement equations

$$x_{k+1} = f_k(x_k) + q_k \quad (1)$$

$$z_k = h_k(x_k) + r_k \quad (2)$$

where $f_k(\cdot)$ and $h_k(\cdot)$ are possibly nonlinear functions, q_k and r_k are independent zero-mean Gaussian noises with covariance matrices Q_k and R_k , respectively. The prior PDF of x_0 is $p(x_0) = \mathcal{N}(x_0; \bar{x}_0, P_0)$, which denotes a Gaussian PDF evaluated at x_0 with mean \bar{x}_0 and covariance matrix P_0 .

After observing $z_{1:N}$, all information of interest about the sequence $x_{0:N}$ of states is given by the posterior PDF

$$p(x_{0:N} | z_{1:N}) \propto \prod_{k=1}^N [p(z_k | x_k) p(x_k | x_{k-1})] p(x_0) \quad (3)$$

where \propto means “is proportional to” and, according to (1)-(2),

$$p(x_k | x_{k-1}) = \mathcal{N}(x_k; f_{k-1}(x_{k-1}), Q_{k-1})$$

$$p(z_k | x_k) = \mathcal{N}(z_k; h_k(x_k), R_k).$$

In practice, the posterior does not admit a closed-form expression so it must be approximated. In this paper, we seek the best possible enabling approximations

$$f_k(x_k) \approx F_k x_k + a_k + e_k \quad (4)$$

$$h_k(x_k) \approx H_k x_k + b_k + g_k \quad (5)$$

where $F_k \in \mathbb{R}^{n_x \times n_x}$, $a_k \in \mathbb{R}^{n_x}$, $H_k \in \mathbb{R}^{n_z \times n_x}$, $b_k \in \mathbb{R}^{n_z}$ and, $e_k \in \mathbb{R}^{n_x}$ and $g_k \in \mathbb{R}^{n_z}$, which are zero-mean independent Gaussian noises with covariance matrices Λ_k and Ω_k , respectively. That is, nonlinear dynamic and measurement functions are approximated as affine functions with additive Gaussian noise, where the noise can be viewed as a term that captures approximation errors. This kind of approximation is commonly used in the literature, as will be explained in Section II-B.

We denote the parameters of the enabling approximation by

$$\Theta = (F_{0:N-1}, a_{0:N-1}, \Lambda_{0:N-1}, H_{1:N}, b_{1:N}, \Omega_{1:N}). \quad (6)$$

Then, once the enabling approximation in (4)-(5) is performed, the posterior approximation becomes Gaussian with PDF

$$q^\Theta(x_{0:N} | z_{1:N}) \propto \prod_{k=1}^N [q^\Theta(z_k | x_k) q^\Theta(x_k | x_{k-1})] p(x_0) \quad (7)$$

where

$$q^\Theta(x_k | x_{k-1}) = \mathcal{N}(x_k; F_{k-1}x_{k-1} + a_{k-1}, \Lambda_{k-1} + Q_{k-1})$$

$$q^\Theta(z_k | x_k) = \mathcal{N}(z_k; H_k x_k + b_k, \Omega_k + R_k).$$

We can also compute the marginal PDFs

$$q^\Theta(x_k | z_{1:N}) = \mathcal{N}(x_k; \bar{u}_k, W_k) \quad (8)$$

for $k \in \{0, \dots, N\}$ in closed-form using the affine RTS smoother, which can be derived via the general Gaussian smoother equations in [1], or an affine two-filter smoother. The resulting steps for the forward-backward case are provided in Algorithm 1.

It is important to realise that the accuracy of the posterior approximation only depends on how we choose Θ . We are interested in computing posterior moments so, ideally, we want $q^\Theta(x_{0:N} | z_{1:N})$ to match the first two moments of $p(x_{0:N} | z_{1:N})$. The objective of this paper is to develop a computationally efficient technique to select Θ in a way that improves widely-used choices of Θ in the literature and compute marginal distributions based on it. First, we proceed to review statistical linear regression and previous work on the selection of Θ .

Algorithm 1 Affine RTS smoother

Input: Prior moments \bar{x}_0, P_0 and SLR parameter Θ .

Output: Posterior moments \bar{u}_k, W_k for $k = \{0, \dots, N\}$.

for $k = 1$ to N **do**

- $\bar{x}_k^- = F_{k-1}\bar{x}_{k-1} + a_{k-1}$. ▷ Filtering

- $P_k^- = F_{k-1}P_{k-1}F_{k-1}^T + \Lambda_{k-1} + Q_{k-1}$. ▷ KF prediction

- $\bar{z}_k = H_k\bar{x}_k^- + b_k$. ▷ KF update

- $S_k = H_kP_k^-H_k^T + \Omega_k + R_k$.

- $\Psi_k = P_k^-H_k^T$.

- $\bar{x}_k = \bar{x}_k^- + \Psi_k S_k^{-1}(z_k - \bar{z}_k)$.

- $P_k = P_k^- - \Psi_k S_k^{-1}\Psi_k^T$.

end for

- $\bar{u}_N = \bar{x}_N$ and $W_N = P_N$. ▷ Smoothing

for $k = N - 1$ to 0 **do**

- $G_k = P_k F_k^T (P_{k+1}^-)^{-1}$.

- $\bar{u}_k = \bar{x}_k + G_k (\bar{u}_{k+1} - \bar{x}_{k+1}^-)$.

- $W_k = P_k + G_k (W_{k+1} - P_{k+1}^-) G_k^T$.

end for

A. Statistical linear regression

In this section we describe SLR, which plays an important role in this paper.

Definition 1. Given a function $h(\cdot)$ and a PDF $p(\cdot)$, whose first two moments are \bar{x} and P , the SLR of $h(\cdot)$ with respect to $p(\cdot)$ is given by [12]

$$H^+ = \Psi^T P^{-1} \quad (9)$$

$$b^+ = \bar{z} - H^+ \bar{x} \quad (10)$$

$$\Omega^+ = \Phi - H^+ P (H^+)^T \quad (11)$$

where

$$\bar{z} = \int h(x) p(x) dx \quad (12)$$

$$\Psi = \int (x - \bar{x}) (h(x) - \bar{z})^T p(x) dx \quad (13)$$

$$\Phi = \int (h(x) - \bar{z}) (h(x) - \bar{z})^T p(x) dx. \quad (14)$$

The approximation $h(x) \approx H^+ x + b^+$ represents the best affine approximation of $h(\cdot)$ in the sense of minimising its mean square error (MSE) with respect to $p(\cdot)$ and Ω^+ is its MSE matrix [12]:

$$(H^+, b^+) = \arg \min_{(H, b)} \mathbb{E} \left[(h(x) - Hx - b)^T (h(x) - Hx - b) \right] \quad (15)$$

$$\Omega^+ = \mathbb{E} \left[(h(x) - H^+ x - b^+) (h(x) - H^+ x - b^+)^T \right] \quad (16)$$

where the expected values are taken with respect to $p(\cdot)$.

In practice, the required moments of SLR (12)-(14) cannot be computed in closed-form in most non-linear cases. Nevertheless, we can approximate them using sigma-point methods such as the unscented transform [9]. We first select m sigma-points $\mathcal{X}_1, \dots, \mathcal{X}_m$ and weights $\omega_1, \dots, \omega_m$, which match the moments \bar{x} and P , according to a suitable sigma-point method [1], [9], [17]. Then, the SLR is performed as indicated in Algorithm 2.

Algorithm 2 Statistical linear regression using sigma-points

Input: Function $h(\cdot)$ and first two moments \bar{x} , P of a PDF $p(\cdot)$.

Output: SLR parameters (H^+, b^+, Ω^+) .

- Select m sigma-points $\mathcal{X}_1, \dots, \mathcal{X}_m$ and weights $\omega_1, \dots, \omega_m$ according to \bar{x} and P .
- Transform the sigma-points $\mathcal{Z}_j = h(\mathcal{X}_j) \quad j = 1, \dots, m$.
- Approximate the moments (12)-(14) as

$$\bar{z} \approx \sum_{j=1}^m \omega_j \mathcal{Z}_j$$

$$\Psi \approx \sum_{j=1}^m \omega_j (\mathcal{X}_j - \bar{x}) (\mathcal{Z}_j - \bar{z})^T$$

$$\Phi \approx \sum_{j=1}^m \omega_j (\mathcal{Z}_j - \bar{z}) (\mathcal{Z}_j - \bar{z})^T$$

- Obtain H^+, b^+, Ω^+ from (9)-(11).
-

B. Previous work

In this section, we briefly describe how popular Gaussian smoothers from the literature can be viewed as different strategies to select the enabling approximation in (4)-(5). Let

$$q^{\Theta_{k-1}}(x_k | z_{1:k-1}) = \mathcal{N}(x_k; \bar{x}_k^-, P_k^-)$$

$$q^{\Theta_k}(x_k | z_{1:k}) = \mathcal{N}(x_k; \bar{x}_k, P_k)$$

represent the predicted and filtering densities at time k where Θ_k represents the parameters of the enabling approximations up to time step k .

If we select $\Omega_k = 0$, $\Lambda_k = 0$, and (H_k, b_k) and (F_k, a_k) by analytical linearisation at the predicted mean \bar{x}_k^- and filtered mean \bar{x}_k , respectively, the resulting algorithm is the extended RTS smoother [1]. If we select (H_k, b_k, Ω_k) and (F_k, a_k, Λ_k) by SLR with respect to the predicted density and the filtering density, respectively, the resulting algorithm is the (nonlinear) RTS smoother. If these SLRs are approximated using sigma-points, we obtain the sigma-point RTS smoothers such as unscented, cubature or Gauss–Hermite RTS smoothers [1], [8], [10], [18]. It should be noted that the previous linearisations are performed while filtering without taking into account current and future measurements. If we select $\Omega_k = 0$, $\Lambda_k = 0$, and (H_k, b_k) and (F_k, a_k) by analytical linearisation at the MAP estimate of $x_{0:N}$ calculated by a Gauss–Newton method, we obtain the IEKS [13].

III. ITERATED POSTERIOR LINEARISATION SMOOTHER

In this section we explain the posterior linearisation smoother (PLS). Then, we present the iterated PLS (IPLS), which will be derived using KLD minimisation in Section IV, and analyse its convergence. As the IPLS can also be used to improve filtering, we also provide an efficient adaptation of the smoother to this case.

A. Posterior linearisation smoother

We have seen in the previous section that many smoothers use SLR in order to select the enabling approximation (4)-(5). However, these algorithms select the SLR without taking into account all measurements. This is clearly a suboptimal approach as we are not considering all available information to select the enabling approximation.

In this paper, we extend the ideas behind the PLF [16] to smoothing. In the PLF, the SLR of the measurement function is performed with respect to the filtering posterior so that we take into account the current measurement in the enabling approximation. The generalisation of this idea to smoothing is straightforward and consists of selecting the enabling approximation (4)-(5) by SLR of the functions with respect to the posterior PDF, which is given by (3). The resulting algorithm, the posterior linearisation smoother (PLS), is intractable because it requires knowledge of the posterior to approximate the posterior. Nevertheless, we can design an iterative procedure to approximate the PLS as will be explained next.

B. Forward-backward IPLS

The IPLS is based on carrying out iterated SLRs. That is, since we do not have access to the posterior to perform the SLRs of the PLS, we perform SLR of the nonlinear functions with respect to the best available approximation of the posterior. At the end of each iteration, we expect to obtain an improved approximation of the posterior, which means that can later be used to obtain an even better approximation to the posterior.

The IPLS creates a sequence Θ^i $i \in \mathbb{N}$ of enabling approximation parameters such that the objective is to improve the posterior approximation at each iteration. As the nonlinear functions only depend on the state at a given time, the SLRs are performed with respect to the marginals of $q^{\Theta^i}(x_{0:N}|z_{1:N})$, which are denoted as

$$q^{\Theta^i}(x_k|z_{1:N}) = \mathcal{N}(x_k; \bar{u}_k^i, W_k^i), \quad k \in \{0, \dots, N\}.$$

Given Θ^i , these marginals can be computed via forward-backward smoothing, see Algorithm 1, or two-filter smoothing. The steps of the forward-backward IPLS are summarised in Algorithm 3, where we use J to denote the number of iterations and SLR($f(\cdot), \bar{u}_k^i, W_k^i$) stands for SLR of $f(\cdot)$ with respect to $\mathcal{N}(\cdot; \bar{u}_k^i, W_k^i)$.

Algorithm 3 Forward-backward IPLS

Input: Prior moments $\bar{u}_0^1 = \bar{x}_0, W_0^1 = P_0$.

Output: Posterior moments \bar{u}_k^J, W_k^J for $k \in \{0, \dots, N\}$.

- Initialisation: Compute \bar{u}_k^1, W_k^1 for $k \in \{0, \dots, N\}$ by a sigma-point smoother.

for $i = 1$ to $J - 1$ **do**

for $k = 0$ to N **do**

 - $(F_k^{i+1}, a_k^{i+1}, \Lambda_k^{i+1}) = \text{SLR}(f_k(\cdot), \bar{u}_k^i, W_k^i)$. ▷ We use Algorithm 2

 - $(H_k^{i+1}, b_k^{i+1}, \Omega_k^{i+1}) = \text{SLR}(h_k(\cdot), \bar{u}_k^i, W_k^i)$.

end for

 - Obtain $\bar{u}_k^{i+1}, W_k^{i+1}$ for $k \in \{0, \dots, N\}$ using Algorithm 1 with parameters $(F_k^{i+1}, a_k^{i+1}, \Lambda_k^{i+1})$ and $(H_k^{i+1}, b_k^{i+1}, \Omega_k^{i+1})$.

end for

C. Convergence

If the moments (12)-(14) are approximated by analytical linearisation rather than SLR, the IPLS becomes the IEKS, which is a Gauss–Newton method. In this case, the algorithm converges locally, but in general it is not guaranteed to converge [13]. In general, we provide the following theorem:

Theorem 2 (Local convergence of the IPLS). *The IPLS converges if it is initiated sufficiently close to the fixed point and the eigenvalues of matrix Ξ , which is given by (27), are lower than one.*

Theorem 2 is proved in the Appendix. The algorithm converges if the initial point of the iteration is sufficiently close to the fixed point and the nonlinearities of the system are sufficiently mild, as indicated by the eigenvalues of Ξ . If the initial point of the iteration is not sufficiently close to the fixed point, the algorithm could diverge or converge to a bad fixed point, which corresponds to a PDF that does not represent the posterior accurately. We also want to remark that quite similar conditions are obtained for the convergence of the

Gauss–Newton method [19, Eq. (6.1.11)], which corresponds to the IEKS.

D. Use in filtering

In this section, we present a modification of the IPLS to tackle filtering problems, which we call the L -scan IPLF. Popular sigma-point smoothers such as the ones mentioned in Section II-B do not improve filtering performance. In contrast, the IPLS can also improve the filtering due to the iterated SLRs of all nonlinear functions. As a result, even if we are only interested in the filtering solution, we can also use the IPLS.

In filtering, N increases and the objective is to approximate the filtering density $p(x_N | z_{1:N})$ per each value of N [1]. We can run the IPLS every time we receive a new measurement. However, this is a computationally inefficient option as the marginals posteriors that correspond to a long time ago probably remain unaltered by a measurement at the current time. Instead, we can set a parameter $L \in \mathbb{N}$ so that we only relinearise the nonlinear dynamic and measurement functions in the window that comprises the last L time steps. Assuming that, at time step N , we have the filtering posterior approximations

$$q^{\Theta}(x_k | z_{1:k}) = \mathcal{N}(x_k; \bar{x}_k, P_k)$$

and the SLRs $(F_{k-1}, a_{k-1}, \Lambda_{k-1})$ and (H_k, b_k, Ω_k) for $k \in \{N^-, \dots, N\}$, where $N^- = \max(1, N - L + 1)$, the objective is to approximate the filtering posterior at time $N + 1$ using the IPLS in the last L time steps. In order to be able to continue with the filtering recursion, we also need to provide the filtering posterior approximations and SLRs for $k \in \{N^- + 1, \dots, N + 1\}$.

One step of the filtering recursion using the L -scan IPLF is described in Algorithm 4. It should be noted that the 1-scan IPLF corresponds to the IPLF [16]. In this case, the recursion simplifies as we do not need to iterate on the prediction step and relinearise dynamic functions.

Algorithm 4 Prediction and update step for the L -scan IPLF

Input: Measurement z_{k+1} , filtering moments \bar{x}_k, P_k and SLRs $(F_{j-1}, a_{j-1}, \Lambda_{j-1})$ and (H_j, b_j, Ω_j) for $j \in \{N^-, \dots, N\}$.

Output: Filtering moments \bar{x}_k, P_k and SLRs $(F_{j-1}, a_{j-1}, \Lambda_{j-1})$ and (H_j, b_j, Ω_j) for $j \in \{N^- + 1, \dots, N + 1\}$.

- Let $K = \{N^- + 1, \dots, N + 1\}$.

- Use a sigma-point filter prediction to obtain $\bar{x}_{N+1}^-, P_{N+1}^-$ and (F_N, a_N, Λ_N) .

- Use a sigma-point filter update to obtain \bar{x}_{N+1}, P_{N+1} and $(H_{N+1}, b_{N+1}, \Omega_{N+1})$.

- Run smoother to obtain \bar{u}_k, W_k for $k \in K$. ▷ First iteration

- Apply KF prediction to get $\bar{x}_{N+1}^-, P_{N+1}^-$ based on $(F_{N-}, a_{N-}, \Lambda_{N-})$ and \bar{x}_{N-}, P_{N-} . ▷ Prior at the beginning of the window

for $i = 2$ to J **do**

for each $k \in K$ **do**

 - $(F_k, a_k, \Lambda_k) = \text{SLR}(f_k(\cdot), \bar{u}_k, W_k)$.

 - $(H_k, b_k, \Omega_k) = \text{SLR}(h_k(\cdot), \bar{u}_k, W_k)$.

end for

- Run KF with parameters (F_k, a_k, Λ_k) and (H_k, b_k, Ω_k) and prior $\bar{x}_{N+1}^-, P_{N+1}^-$ to recalculate \bar{x}_k, P_k for $k \in K$.

- Recalculate smoothing moments \bar{u}_k, W_k for $k \in K$.

end for

IV. DERIVATION OF THE IPLS USING THE KLD

In this section, we derive the IPLS as an iterated approximate KLD minimisation. To put the derivation into context, we first present the ideal smoother solution, the nonlinear RTS smoother and the PLS from a KLD perspective.

A. Ideal smoother

As mentioned in Section II, ideally, we want to select Θ such that $q^\Theta(x_{0:N} | z_{1:N})$ matches the first two moments of $p(x_{0:N} | z_{1:N})$. This selection can be seen as the KLD minimisation [20]

$$\begin{aligned} & \arg \min_{\Theta} D(p(x_{0:N} | z_{1:N}) \| q^\Theta(x_{0:N} | z_{1:N})) \\ &= \arg \min_{\Theta} \int p(x_{0:N} | z_{1:N}) \log \frac{p(x_{0:N} | z_{1:N})}{q^\Theta(x_{0:N} | z_{1:N})} dx_{0:N}. \end{aligned} \quad (17)$$

This minimisation could be solved approximately via Monte Carlo methods but they are computationally expensive in relation to the IPLS.

B. Nonlinear RTS smoother

In the nonlinear RTS smoother, the nonlinear functions are linearised once while filtering without considering current and future measurements in the enabling approximations [8]. At time step k , the enabling approximation (5) for the measurement function is selected by minimising the following joint KLD of PDFs on the state and the measurement at the current time step:

$$\begin{aligned} (H_k^+, b_k^+, \Omega_k^+) = & \arg \min_{(H_k, b_k, \Omega_k)} D(q^{\Theta_{k-1}}(x_k | z_{1:k-1}) p(z_k | x_k) \\ & \| q^{\Theta_{k-1}}(x_k | z_{1:k-1}) q^{H_k, b_k, \Omega_k}(z_k | x_k)). \end{aligned} \quad (18)$$

The solution is given by SLR of $h_k(\cdot)$ with respect to $q^{\Theta_{k-1}}(x_k | z_{1:k-1})$. The linearisation for the dynamic measurement is obtained by minimising (18) but substituting z_k by x_{k+1} , Θ_{k-1} by Θ_k and $z_{1:k-1}$ by $z_{1:k}$. The solution is given by SLR of $f_k(\cdot)$ with respect to $q^{\Theta_k}(x_k | z_{1:k})$. We want to highlight that, in contrast to the ideal solution, which is given by (17), the nonlinear RTS smoother does not consider all states and measurements in the KLD minimisation.

C. PLS

In this section, we explain the PLS from a KLD point of view, which will be useful to derive the iterated PLF in Section IV-D. The general idea is that we want to see how well we approximate the joint posterior density of the states and the states transformed by the nonlinear functions using the KLD. This is of interest as the enabling approximation (4)-(5) implicitly provides us with an approximation to these functions. To do so, we consider the auxiliary variables

$$\begin{aligned} v_k &= f_k(x_k) + \varepsilon_k \\ y_k &= h_k(x_k) + \xi_k \end{aligned}$$

where ε_k and ξ_k are independent zero-mean Gaussian noises with covariance matrix βI_{n_x} and βI_{n_z} , respectively, I_{n_z} is the identity matrix of size n_z and $\beta > 0$ is a parameter to ensure that v_k and y_k have a density given x_k and we can therefore use the KLD. As $\beta \rightarrow 0$, then, $v_k \rightarrow f_k(x_k)$ and $y_k \rightarrow h_k(x_k)$. Therefore, the joint posterior approximation $q^\Theta(x_{0:T}, v_{0:T-1}, y_{1:T} | z_{1:T})$, which can be obtained analytically once the enabling approximation (4)-(5) is performed, in relation to the true joint posterior $p(x_{0:N}, v_{0:N-1}, y_{1:N} | z_{1:N})$ tells us how well we have approximated the distribution over the state transformed by the functions as well as the state.

The true joint PDF of the variables $(x_{0:N}, v_{0:N-1}, y_{1:N})$ given $z_{1:N}$ factorises as

$$\begin{aligned} & p(x_{0:N}, v_{0:N-1}, y_{1:N} | z_{1:N}) \\ &= p(x_{0:N} | z_{1:N}) \prod_{k=0}^{N-1} p(v_k | x_k) \prod_{k=1}^N p(y_k | x_k). \end{aligned} \quad (19)$$

We make the approximation

- AP1

$$\begin{aligned} & q^\Theta(x_{0:T}, v_{0:T-1}, y_{1:T} | z_{1:T}) \\ & \approx q^\Theta(x_{0:T} | z_{1:T}) \prod_{k=0}^{N-1} q^\Theta(v_k | x_k) \prod_{k=1}^N q^\Theta(y_k | x_k) \end{aligned} \quad (20)$$

with

$$\begin{aligned} q^\Theta(v_k | x_k) &= \mathcal{N}(v_k; F_k x_k + a_k, \Lambda_k + \beta I_{n_x}) \\ q^\Theta(y_k | x_k) &= \mathcal{N}(y_k; H_k x_k + b_k, \Omega_k + \beta I_{n_z}). \end{aligned}$$

If $\Lambda_k = 0$ and $\Omega_k = 0$ for all k then we have an equality in (20) so Approximation AP1 is accurate for low nonlinearities. This kind of approximation is analysed more thoroughly for the filtering case in [16].

We would like to select Θ that minimises the KLD from the true posterior (19) to its approximation (20)

$$\begin{aligned} & \arg \min_{\Theta} [D(p(x_{0:N} | z_{1:N}) \| q^\Theta(x_{0:N} | z_{1:N})) \\ & + \int p(x_{0:N} | z_{1:N}) D(p(v_{0:N-1}, y_{1:N} | x_{0:N}) \\ & \| q^\Theta(v_{0:N-1}, y_{1:N} | x_{0:N})) dx_{0:N}]. \end{aligned} \quad (21)$$

Instead, the PLS selects Θ that minimises the second term in (21) as indicated by the following lemma.

Lemma 3. *The solution to*

$$\begin{aligned} \Theta^* = & \arg \min_{\Theta} \int p(x_{0:N} | z_{1:N}) D(p(v_{0:N-1}, y_{1:N} | x_{0:N}) \\ & \| q^\Theta(v_{0:N-1}, y_{1:N} | x_{0:N})) dx_{0:N} \end{aligned} \quad (22)$$

under Approximation AP1 is given by SLR of the measurement and dynamic functions with respect to the posterior $p(x_{0:N} | z_{1:N})$.

Lemma 3 can be proved by analogy with the filtering case [16]. It should be noted that the SLRs of the measurement and dynamic functions at time step k only depend on the marginal posterior $p(x_k | z_{1:N})$ and they are calculated according to Definition 1. The PLS therefore minimises a lower bound of the joint KLD (21) under Approximation AP1 [16].

D. IPLS

In this section, we derive the IPLS using the KLD. First, we remark that the previously mentioned smoothers belong to the assumed density framework, where the approximation $p(\cdot | z_{1:N}) \approx q^\Theta(\cdot | z_{1:N})$ is made to evaluate quantities of interest such as integrals or KLDs [16, Remark 1].

In the IPLS, there is a sequence of enabling approximations Θ^i , $i \in \mathbb{N}$. Given Θ^i , the aim is to obtain Θ^{i+1} that minimises (21) under the assumed density approximation $p(x_{0:N} | z_{1:N}) \approx q^{\Theta^i}(x_{0:N} | z_{1:N})$ such that

$$\begin{aligned} \arg \min_{\Theta^{i+1}} & \left[D \left(q^{\Theta^i}(x_{0:N} | z_{1:N}) \parallel q^{\Theta^{i+1}}(x_{0:N} | z_{1:N}) \right) \right. \\ & \left. + \int q^{\Theta^i}(x_{0:N} | z_{1:N}) D \left(p(v_{0:N-1}, y_{1:N} | x_{0:N}) \right. \right. \\ & \left. \left. \parallel q^{\Theta^{i+1}}(v_{0:N-1}, y_{1:N} | x_{0:N}) \right) dx_{0:N} \right]. \end{aligned} \quad (23)$$

At the fixed point of this recursion, the first term in (23) is zero. Therefore, as in the IPLF, if we are sufficiently close to the fixed point, we can make the approximation

- AP2 The first term in (23) is negligible compared to the second.

Under Approximations AP1 and AP2, we can use Lemma 3 so that the solution to (23) is given by selecting Θ^{i+1} using SLR with respect to $q^{\Theta^i}(x_{0:N} | z_{1:N})$. This leads to the IPLS, in which we recursively perform SLR of the nonlinear functions with respect to the latest posterior approximation.

V. SIMULATIONS

In this section we compare several Gaussian smoothers and filters via simulations. The SLRs have been implemented using the unscented transform with $N_s = 2n_x + 1$ sigma-points and the weight of the sigma-point located on the mean is 1/3. We use the following terminology: IPLS(i)- J denotes the IPLS in which we first run an IPLF with i iterations and J iterations of the smoother; IEKS(i)- J denotes the IEKS in which we first run an IEKF with i iterations and J iterations of the smoother [13]. We should note that IPLS(1)-0, IPLS(1)-1, IEKS(1)-0, IEKS(1)-1 and IPLS(i)-0 represent the UKF, the unscented RTS smoother [1], the EKF, the extended RTS smoother [1] and the IPLF with i iterations [16], respectively. In other words, popular sigma-point filters and smoothers correspond to the IPLS with certain selection of parameters. We have also implemented the L -scan IPLF to see how filtering can be improved by relinearising the dynamic and measurement functions.

We analyse the univariate nonstationary growth model whose dynamic function is

$$f_k(x_k) = 0.9x_k + \frac{10x_k}{1 + (x_k)^2} + 8 \cos(1.2 \cdot k)$$

with a quadratic and cubic measurement models. The quadratic measurement model is a usual model to compare how filters deal with multimodality and nonlinearities [7], [21]. However, as mentioned in the introduction, Gaussian filters only approximate the posterior accurately if it is unimodal, as with cubic measurements. Otherwise, more general filters, with a higher computational complexity, should be used [21].

Table I: RMS error with cubic measurement

	J			
	0	1	5	10
IEKS(1)	8.80	7.67	1.25	0.73
IEKS(5)	1.17	1.53	0.78	0.72
IEKS(10)	0.74	0.87	0.76	0.74
IPLS(1)	2.20	1.92	0.46	0.46
IPLS(5)	0.60	0.50	0.47	0.49
IPLS(10)	0.61	0.53	0.47	0.49

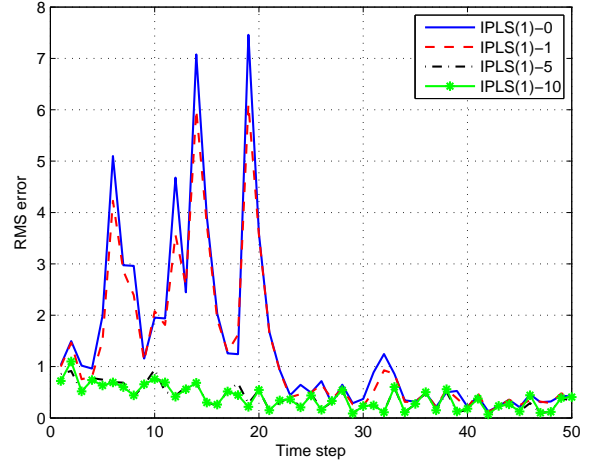


Figure 1: RMS error against time step for the IPLS. Iteration improves performance.

Cubic measurement: We first analyse the case with a cubic measurement

$$h(x_k) = \frac{(x_k)^3}{20}.$$

The prior PDF parameters are $\bar{x}_0 = 5$, $P_0 = 4$ and the noise variances are $Q = 1$, $R = 1$. In order to evaluate the algorithms, we generate 20 independent trajectories with 50 time steps and 50 independent measurement sequences for each trajectory, so we consider 1000 Monte Carlo runs in total.

The root mean square (RMS) errors of the state estimates averaged over all Monte Carlos runs for the IEKS and IPLS are shown in Table I. It can be seen that performing iterations in the filter and the smoother lowers the error for both the IEKS and the IPLS. The IPLS has consistently a lower error than the IEKS. The error of the unscented RTS smoother (IPLS(1)-1) can be greatly lowered by iterations using the IPLS. In this case, best results are given by IPLS(1)-10, which implies that we iterate in the smoother but not in the filter. This is achieved even though the best filtering results are obtained by performing iterations in the filter, compare IPLS(5)-0 and IPLS(10)-10 with IPLS(1)-0. We also show how the RMS error at each time step decreases with the iterations for the IPLS(1) in Figure 1. The RMS error is reduced significantly by iteration for time steps lower than 25.

We also show the expected value of the negative log-likelihood (ENLL) [22] in Table II. The lower it is, the better we fit the posterior moments [22]. Iterations are also beneficial in general and, in this case, lowest ENLL is achieved by IPLS(10)-1. The execution times in milliseconds of the

Table II: ENLL with cubic measurement

	J			
	0	1	5	10
IEKS(1)	$3.22 \cdot 10^9$	$3.22 \cdot 10^9$	$4.17 \cdot 10^5$	31.21
IEKS(5)	$7.72 \cdot 10^4$	$7.72 \cdot 10^4$	190.73	23.02
IEKS(10)	12.17	14.17	32.41	58.61
IPLS(1)	$1.21 \cdot 10^3$	$1.21 \cdot 10^3$	4.82	-0.58
IPLS(5)	39.88	39.87	-0.55	-0.50
IPLS(10)	-0.63	-0.68	-0.45	-0.45

Table III: Execution times (ms) of the algorithms

	J			
	0	1	5	10
IEKS(1)	4	5	21	39
IEKS(5)	10	11	25	45
IEKS(10)	16	17	32	50
IPLS(1)	11	12	53	105
IPLS(5)	29	30	70	120
IPLS(10)	50	51	92	145

algorithms are shown in Table III. Obviously, the IEKS has a lower computational burden but requires the computation of gradients and does not attain the performance of the IPLS.

We also show the results of the L -scan IPLF in Table IV. The L -scan IPLF is a filter so it should be compared with the filters, which are given by column $J = 0$ in Table I. In this scenario, the L -scan IPLF attains roughly the same performance for different values of L . If $J = 2$, the L -scan IPLF does not improve the performance of the IPLS(10)-0, which only relinearises the measurement function at the current time step. However, if we perform more iterations the L -scan IPLF outperforms the IPLS(10)-0, which corresponds to the IPLF [16]. The execution times of the L -scan IPLFs are shown in Table V. The greater L and J are, the higher the computational burden is.

Quadratic measurement: We also provide simulation results with a quadratic measurement

$$h(x_k) = \frac{(x_k)^2}{20}.$$

Using the same parameters as in the previous section, the RMS errors of the algorithms are given in Table VI. As before, the IPLS outperforms the IEKS and performing more iterations in the IPLS decreases the value of the error. The most significant difference with the previous case is that now it is not convenient to run iterations within the filters. That is, IPLS(10)-0 has a considerably higher error than IPLS(1)-0, and the same applies to the IEKS. This is due to the fact

Table IV: RMS error for the L -scan IPLF with cubic measurement

L	J		
	2	5	10
2	1.39	0.56	0.57
5	1.43	0.56	0.57
10	1.43	0.56	0.57

Table V: Execution times (ms) for the L -scan IPLF

L	J		
	2	5	10
2	38	107	214
5	75	228	481
10	126	414	900

Table VI: RMS error with quadratic measurement

	J			
	0	1	5	10
IEKS(1)	6.24	6.06	6.14	6.10
IEKS(5)	7.99	8.12	7.98	7.96
IEKS(10)	8.33	8.49	8.30	8.30
IPLS(1)	1.80	1.46	1.04	1.01
IPLS(5)	5.64	5.67	5.57	5.56
IPLS(10)	6.92	7.00	6.89	6.84

that the posterior is not always unimodal so increasing the iterations at this point is not beneficial, but it is in smoothing as we have information from all measurements.

VI. CONCLUSIONS

We have proposed an iterated smoothing algorithm called the iterated posterior linearisation smoother (IPLS). At each step of the iteration, the IPLS performs statistical linear regression of the nonlinear functions with respect to the current posterior approximation. The IPLS is derivative-free, uses sigma-points to approximate the statistical linear regressions and outperforms other Gaussian smoothers in smoothers in the literature in two numerical examples.

In addition, popular sigma-point filters and smoothers are equivalent to the IPLS with a certain selection of parameters. Moreover, if we approximate the SLR by analytical linearisation rather than sigma-point methods, the IPLS becomes the IEKS. We can therefore see the IPLS as a nontrivial extension of popular Gaussian filters and smoothers. As the IPLS also changes the filtering solution, we have also proposed the L -scan IPLF. This filter is based on applying the IPLS in the last L time steps and can perform better than other filters due the relinearisation of the dynamic and measurement functions.

Future work includes the extension of this work to Bayesian graphical models or the development of damped versions of the algorithm to improve convergence.

APPENDIX

In this appendix, we prove Theorem 2. This local convergence proof is quite similar to the convergence proof of the IPLF [16, App. B]. In this section, we use \bar{u}^i and W^i to denote the mean and covariance matrix of the whole trajectory at the i th iteration. The vectorisation of W^i is denoted as w^i [16]. We also set $y^i = \left[(\bar{u}^i)^T, (w^i)^T \right]^T$, $\Sigma_k^i = (\Omega_k^i + R_k)^{-1/2}$, $\Gamma_k^i = (\Lambda_k^i + Q_k)^{-1/2}$ and

$$r(y^i) = \begin{bmatrix} P_0^{-1/2} (x_0 - \bar{x}_0) \\ \Gamma_0^i (x_1 - F_0^i x_0 - a_0^i) \\ \Sigma_1^i (z_1 - H_1^i x_1 - b_1^i) \\ \vdots \\ \Gamma_{N-1}^i (x_N - F_{N-1}^i x_{N-1} - a_{N-1}^i) \\ \Sigma_N^i (z_N - H_N^i x_N - b_N^i) \end{bmatrix}$$

$$J(y^i)$$

$$= \begin{bmatrix} P_0^{-1/2} & 0 & \dots & 0 & 0 \\ -\Gamma_0^i F_0^i & \Gamma_0^i & \dots & 0 & 0 \\ 0 & -\Sigma_1^i H_1^i & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & 0 & 0 \\ 0 & 0 & \dots & -\Gamma_{N-1}^i F_{N-1}^i & \Gamma_{N-1}^i \\ 0 & 0 & \dots & 0 & -\Sigma_N^i H_N^i \end{bmatrix}$$

where $r(y^i)$ and $J(y^i)$ have dimensions $n_x + N(n_x + n_z) \times 1$ and $n_x + N(n_x + n_z) \times n_x(N+1)$, respectively. From the solution of the linear case, we have that

$$\bar{u}^{i+1} = \bar{u}^i - \tilde{J}(y^i) J^T(y^i) r(y^i)$$

where the updated covariance matrix of the trajectory is

$$\tilde{J}(y^i) = (J^T(y^i) J(y^i))^{-1}. \quad (24)$$

Its vectorisation is

$$w^i = \begin{bmatrix} \tilde{J}(y^i) e_1 \\ \vdots \\ \tilde{J}(y^i) e_{(N+1)n_x} \end{bmatrix} \quad (25)$$

where $e_j \in \mathbb{R}^{n_x(N+1) \times 1}$ is a vector whose components are zero except component j , which is one.

The proof is analogous to the IPLF convergence proof [16, App. B] so we only provide the result here. We use $y^* = \begin{bmatrix} (\bar{u}^*)^T & (w^*)^T \end{bmatrix}^T$ to denote the fixed point, $h_i = \begin{bmatrix} (h_i^u)^T & (h_i^w)^T \end{bmatrix}^T$ with $h_i^u = \bar{u}^i - \bar{u}^*$ and $h_i^w = w^i - w^*$. We also define

$$\begin{aligned} \tilde{H}_j(y^i) &= \left(\nabla J_j(y) \Big|_{y=y^i} \right)^T \\ M(y^i) &= \left(\nabla r(y) \Big|_{y=y^i} \right)^T \end{aligned}$$

and write

$$M(y^i) = \begin{bmatrix} M^x(y^i) & M^w(y^i) \end{bmatrix}$$

where $M^x(y^i)$ corresponds to the first $(N+1) \cdot n_x$ columns and $M^w(y^i)$ denotes the rest. We also define $L(y^i) = M^x(y^i) - J(y^i)$,

$$\begin{aligned} N(y^*, y^i) &= -\tilde{J}(y^i) \left(\sum_{j=1}^{N(n_z+n_x)+n_x} (\tilde{e}_j^T r(y^*)) \tilde{H}_j(y^*) \right. \\ &\quad \left. + J^T(y^i) \begin{bmatrix} L(y^i) & M^w(y^i) \end{bmatrix} \right) \end{aligned}$$

where $\tilde{e}_j \in \mathbb{R}^{N(n_z+n_x)+n_x}$ is a vector whose components are zero except component j , which is one. We also denote by $T_j(y^*)$ the Jacobian of the j th column of $\tilde{J}(\cdot)$ evaluated at y^* . Then, we get that [16, App. B]

$$\|h_{i+1}\| \leq \|\Xi(y^*, y^i)\| \|h_i\| + O(\|h_i\|^2) \quad (26)$$

where

$$\Xi(y^*, y^i) = \begin{bmatrix} N(y^*, y^i) \\ T_1(y^*) \\ \vdots \\ T_{N n_x}(y^*) \end{bmatrix}. \quad (27)$$

Therefore, if the absolute values of the eigenvalues of $\Xi(y^*, y^i)$ are lower than one, $\|\Xi(y^*, y^i)\| < 1$ and we get linear convergence.

REFERENCES

- [1] S. Särkkä, *Bayesian filtering and smoothing*. Cambridge University Press, 2013.
- [2] Y. Bar-Shalom, T. Kirubarajan, and X. R. Li, *Estimation with Applications to Tracking and Navigation*. John Wiley & Sons, Inc., 2001.
- [3] A. H. Jazwinski, *Stochastic Processes and Filtering Theory*. Academic Press, 1970.
- [4] S. J. Godsill and P. J. Rayner, *Digital Audio Restoration: A Statistical Model Based Approach*. Springer, 1998.
- [5] H. E. Rauch, F. Tung, and C. T. Striebel, "Maximum likelihood estimates of linear dynamic systems," *AIAA Journal*, vol. 3, no. 8, pp. 1445–1450, Aug. 1965.
- [6] D. Fraser and J. Potter, "The optimum linear smoother as a combination of two optimum linear filters," *IEEE Transactions on Automatic Control*, vol. 14, no. 4, pp. 387–390, Aug. 1969.
- [7] G. Kitagawa, "Monte Carlo filter and smoother for non-Gaussian nonlinear state space models," *Journal of Computational and Graphical Statistics*, vol. 5, no. 1, pp. 1–25, Mar. 1996.
- [8] S. Särkkä and J. Hartikainen, "On Gaussian optimal smoothing of nonlinear state space models," *IEEE Transactions on Automatic Control*, vol. 55, no. 8, pp. 1938–1941, Aug. 2010.
- [9] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 401–422, Mar. 2004.
- [10] S. Särkkä, "Unscented Rauch–Tung–Striebel smoother," *IEEE Transactions on Automatic Control*, vol. 53, no. 3, pp. 845–849, April 2008.
- [11] I. Arasaratnam and S. Haykin, "Cubature Kalman smoothers," *Automatica*, vol. 47, no. 10, pp. 2245 – 2250, 2011.
- [12] I. Arasaratnam, S. Haykin, and R. Elliott, "Discrete-time nonlinear filtering algorithms using Gauss–Hermite quadrature," *Proceedings of the IEEE*, vol. 95, no. 5, pp. 953–977, May 2007.
- [13] B. M. Bell, "The iterated Kalman smoother as a Gauss–Newton method," *SIAM Journal on Optimization*, vol. 4, no. 3, pp. 626–636, Aug. 1994.
- [14] B. Bell and F. Cathey, "The iterated Kalman filter update as a Gauss–Newton method," *IEEE Transactions on Automatic Control*, vol. 38, no. 2, pp. 294–297, Feb. 1993.
- [15] T. P. Minka, "Expectation propagation and approximate Bayesian inference," in *Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence*, 2001, pp. 362–369.
- [16] A. F. García-Fernández, L. Svensson, M. R. Morelande, and S. Särkkä, "Posterior linearization filter: principles and implementation using sigma points," *IEEE Transactions on Signal Processing*, vol. 63, no. 20, pp. 5561–5573, Oct. 2015.
- [17] I. Arasaratnam and S. Haykin, "Cubature Kalman filters," *IEEE Transactions on Automatic Control*, vol. 54, no. 6, pp. 1254–1269, June 2009.
- [18] M. Šimandl and J. Dunik, "Derivative-free estimation methods: New results and performance analysis," *Automatica*, vol. 45, no. 7, pp. 1749–1757, July 2009.
- [19] R. Fletcher, *Practical methods of optimization*. John Wiley & Sons, 2000.
- [20] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer Science + Business Media, 2006.
- [21] O. Cappé, E. Moulines, and T. Rydén, *Inference in Hidden Markov models*. Springer Science + Business Media, Inc., 2005.
- [22] M. Deisenroth, "Efficient reinforcement learning using Gaussian processes," Ph.D. dissertation, Karlsruhe Institute of Technology, 2010.