Garcia-Fernandez, Angel F.; Svensson, Lennart; Morelande, Mark R.; Särkkä, Simo

Posterior Linearization Filter: Principles and Implementation Using Sigma Points

Published in:
IEEE Transactions on Signal Processing

DOI:
10.1109/TSP.2015.2454485

Published: 01/01/2015

Document Version
Peer-reviewed accepted author manuscript, also known as Final accepted manuscript or Post-print

Please cite the original version:
Garcia-Fernandez, A. F., Svensson, L., Morelande, M. R., & Särkkä, S. (2015). Posterior Linearization Filter: Principles and Implementation Using Sigma Points. *IEEE Transactions on Signal Processing*, *63*(20), 5561-5573. https://doi.org/10.1109/TSP.2015.2454485

Posterior linearisation filter: principles and implementation using sigma points

Ángel F. García-Fernández, Lennart Svensson, Mark R. Morelande, Simo Särkkä

***Abstract*—This paper is concerned with Gaussian approximations to the posterior probability density function (PDF) in the update step of Bayesian filtering with nonlinear measurements. In this setting, sigma-point approximations to the Kalman filter (KF) recursion are widely used due to their ease of implementation and relatively good performance. In the update step, these sigma-point KFs are equivalent to linearising the nonlinear measurement function by statistical linear regression (SLR) with respect to the prior PDF. In this paper, we argue that the measurement function should be linearised using SLR with respect to the posterior rather than the prior to take into account the information provided by the measurement. The resulting filter is referred to as the posterior linearisation filter (PLF). In practice, the exact PLF update is intractable but can be approximated by the iterated PLF (IPLF), which carries out iterated SLRs with respect to the best available approximation to the posterior. The IPLF can be seen as an approximate recursive Kullback–Leibler divergence minimisation procedure. We demonstrate the high performance of the IPLF in relation to other Gaussian filters in two numerical examples.**

***Index Terms*—Kalman filter, Bayes' rule, nonlinear filtering, sigma-points, statistical linear regression**

I. Introduction

The estimation of the state of a dynamic Markov process from noisy observations is of interest in many problems in science and engineering, for example, target tracking, navigation, audio signal processing or finance [1], [2]. In the Bayesian framework, the process is characterised by the dynamic and measurement equations and, in order to estimate the current state in an optimal manner, we need the probability density function (PDF) of the current state given the available measurements. This PDF is referred to as the posterior PDF and can be calculated recursively in two phases: prediction and update.

If the process is non-linear/non-Gaussian, the posterior PDF cannot be calculated analytically so approximations must be used in practice. Particle filters are asymptotically optimal approximations although their computational burden can be high [3]. If the posterior is unimodal, a Gaussian approximation can be of sufficient accuracy. Therefore, it is of interest to develop computationally efficient Gaussian approximations. In this case, the prediction step typically consists of calculating/approximating the first two moments of a random variable that undergoes the possibly nonlinear transformation of the dynamic equation. In the update step, which is the focus of this paper, we use Bayes' rule along with the current measurement and the prior PDF, which is the output of the prediction step, to obtain the posterior. In the following, we briefly describe several algorithms to perform the update and our contributions. This article is an extended version of the paper [4].

In the Gaussian filtering update step, one possibility is to set the maximum a posteriori (MAP) estimator as the updated mean and obtain the updated covariance matrix by linearising the measurement function around the MAP estimate, as in the iterated extended Kalman filter (IEKF) [5] or the sigma-point Kalman optimisation algorithms in [6]. The previous filters do not guarantee convergence to the MAP estimate but if it is found, the posterior approximation is asymptotically optimal as the measurement noise tends to zero [7]. Nevertheless, the most widely used Gaussian filtering update step consists of approximating the updated mean by the linear minimum mean square error (LMMSE) estimator and the updated covariance by its mean square error matrix. The LMMSE-based approximation is usually more accurate than the MAP estimate one as the metric to assess the filter performance is usually the square error.

The LMMSE-based approximation is sometimes referred to as (nonlinear) Kalman filter (KF) update [8, Sec. II.A] [9] or Gaussian filter update [10]. This algorithm is equivalent to performing statistical linear regression (SLR) of the measurement function with respect to the prior PDF and calculating Bayes' rule with the resulting linearised measurement model [11]. As the KF is usually known in the literature as the solution to the linear/Gaussian filtering recursion and the term Gaussian filter can also refer to many other types of approximations [12], we find it useful to refer to this algorithm as prior linearisation filter (PrLF) in the context of this paper. The PrLF update requires the calculation of some moments: the mean and covariance matrix of the current measurement and cross-covariance between the current state and the current measurement [8]. In practice, these moments (PrLF moments) cannot be calculated in closed-form so we require approximations. In this paper, we refer to these approximations as approximations to the PrLF. For instance, the extended Kalman filter (EKF) approximates the PrLF moments using analytical linearisation

Ángel F. García-Fernández is with the Department of Electrical and Computer Engineering, Curtin University, Perth, WA 6102, Australia (email: angel.garciafernandez@curtin.edu.au). Lennart Svensson is with the Department of Signals and Systems, Chalmers University of Technology, SE-412 96 Gothenburg, Sweden (email: lennart.svensson@chalmers.se). Mark R. Morelande is with the School of Electrical and Computer Engineering, RMIT University, Melbourne, VIC 3000, Australia (email: m.morelande@gmail.com). S. Särkkä is with the Department of Biomedical Engineering and Computational Science, Aalto University, 02150 Espoo, Finland (email: simo.sarkka@aalto.fi).

This work was supported in part by the Australian Research Council under Discovery Project DP130104404.

while the unscented KF (UKF) [8], [13], cubature KF (CKF) [14] or linear regression KF [15] use sigma-points. The main drawback of the PrLF and all its approximations is that they often perform poorly with nonlinear measurement functions if the measurement noise is low enough [9]. Therefore, more accurate computationally efficient approximations must be sought.

As we indicate in this paper, the above-mentioned approaches perform closed-form Bayesian updates once we perform an enabling approximation in which the (nonlinear) measurement function is represented by a linear function plus Gaussian noise. We argue that, rather than selecting the enabling approximation parameters using SLR of the measurement function with respect to the prior as in the PrLF, SLR should be performed with respect to the posterior. The intuition behind this idea is that the approximation of the measurement function should be accurate in the region of interest, which is indicated by the posterior, not the prior [4]. The resulting filter is referred to as the posterior linearisation filter (PLF). However, the PLF is intractable as we would need to know the posterior to approximate the posterior. Nonetheless, we propose an approximation of the PLF by performing iterated SLRs with respect to the best available approximation to the posterior: the iterated PLF (IPLF). We show that the IPLF can be seen, under certain approximations, as a minimisation of the Kullback–Leilber divergence (KLD) of the joint density of the state and a variable obtained by passing the state through the measurement function. This criterion allows the approximated posterior density to be used in an iterative manner while ensuring that the approximation of the measurement function remains accurate. The benefits of using the IPLF in relation to other Gaussian filters are analysed in two numerical examples.

The rest of the paper is organised as follows. In Section II, we formulate the problem. The PLF is introduced in Section III. We derive the IPLF in Section IV. Numerical simulations for assessing the filter performance are given in Section V. Finally, conclusions are drawn in Section VI.

II. Problem statement

In this section, we explain the family of Gaussian approximations we consider in the update step of Bayesian filtering. As we focus on the update phase, the time index of the filtering recursion is removed for the sake of notational simplicity. The state $x \in \mathbb{R}^{n_x}$ has a Gaussian prior PDF $p_x(x) = \mathcal{N}(x; \overline{x}, P)$ with mean $\overline{x}$ and covariance matrix P. The measurement equation is

$$z = h(x) + \eta \tag{1}$$

where $z \in \mathbb{R}^{n_z}$ is the measurement, $h(\cdot)$ is the measurement function and η is a zero-mean Gaussian measurement noise with covariance matrix R.

The posterior PDF $p_{x|z}(\cdot)$ of the state after observing measurement z is obtained by Bayes' rule

$$p_{x|z}(x|z) \propto p_{z|x}(z|x)\, p_x(x) \tag{2}$$

where $\propto$ means "is proportional to" and $p_{z|x}(\cdot)$ is obtained using (1)

$$p_{z|x}(z|x) = \mathcal{N}(z; h(x), R). \tag{3}$$

In practice, the posterior does not admit a closed-form expression so it must be approximated. In this paper, we consider the enabling approximation

$$h(x) \approx \tilde{h}(x) = Ax + b + e \tag{4}$$

where $\tilde{h}(x)$ is the approximation of $h(x)$, $A \in \mathbb{R}^{n_z \times n_x}$, $b \in \mathbb{R}^{n_z}$ and $e \in \mathbb{R}^{n_z}$ is a zero-mean Gaussian distributed random variable with covariance matrix Ω. The variable e is uncorrelated with x and η. Equation (4) approximates how the random variable x is transformed through the deterministic function $h(\cdot)$. As will be seen in Section II-A, linear approximations with additive Gaussian noise have already been suggested in the literature. It should also be noted that affine measurement functions with additive Gaussian noise are the only functions for which the posterior is exactly Gaussian. The class of functions in (4) is therefore of particular interest since it allows us to represent all such measurement functions of importance.

Once the enabling approximation in (4) is employed, $p_{z|x}(z|x) \approx q_{z|x}^{A,b,\Omega}(z|x) = \mathcal{N}(z; Ax + b, \Omega + R)$ and the posterior becomes

$$q_{x|z}^{A,b,\Omega}(x|z) = \mathcal{N}(x; \overline{u}, W) \tag{5}$$

where

$$\overline{u} = \overline{x} + PA^T \left(APA^T + \Omega + R\right)^{-1} (z - A\overline{x} - b) \tag{6}$$

$$W = P - PA^T \left(APA^T + \Omega + R\right)^{-1} AP \tag{7}$$

and superscript T stands for transpose. Considering (6) and (7), it is clear that the accuracy of the posterior approximation depends on how we choose A, b, Ω. In practice, we are interested in computing posterior moments so, ideally, we would like that $\overline{u}$ and W match the first two moments of $p_{x|z}(\cdot)$. We can achieve this by selecting A, b, Ω such that they minimise the KLD [16]

$$\begin{aligned}
&\arg\min_{A,b,\Omega} D\left(p_{x|z} \middle\| q_{x|z}^{A,b,\Omega}\right) \\
&\quad = \arg\min_{A,b,\Omega} \int p_{x|z}(x|z) \log \frac{p_{x|z}(x|z)}{q_{x|z}^{A,b,\Omega}(x|z)} dx.
\end{aligned} \tag{8}$$

The solution to this problem can be approximately obtained by the Gaussian particle filter [12]. However, in the context of this paper, the solution to (8) is impractical as, in general, it is not possible to evaluate integrals w.r.t. $p_{x|z}(\cdot)$ without Monte Carlo methods, which we do not want to use due to their computational burden. In this paper, we propose a new way to select A, b, Ω. First, we proceed to explain previous work on the selection of the linearisation.

A. Previous work

In the Gaussian filtering literature, there are three important kinds of linearisations that are used in (4): analytical linearisation at the prior mean, analytical linearisation at the MAP estimate and SLR with respect to the prior PDF. As the previous linearisations based on the MAP estimate or SLR are not always tractable, different approximations to these linearisations have been proposed. These give rise to different filters but their foundation is the use of one of these linearisations.

More specifically, if we select $\Omega = 0$, and A and b by analytical linearisation at the prior mean, the resulting algorithm is the EKF. If we select $\Omega = 0$, and A and b by analytical linearisation at the MAP estimate, obtained by a Gauss-Newton search, the resulting algorithm is the IEKF [5]. If we select A, b and Ω using SLR with respect to the prior, the resulting algorithm is the PrLF. If the PrLF moments are approximated using sigma-points drawn from the prior, the resulting algorithms are the widely used sigma-point KFs like the UKF or CKF. We think it is worth clarifying that even though the original UKF paper was motivated using the argument that approximating a probability distribution is simpler than a nonlinear function [13], the UKF performs SLR of the mesurement function implicitly. This was already pointed out in [15]. We also want to mention that another interesting linearisation, though less popular than the previous ones, is to set $\Omega = 0$ and select A and b by SLR with respect to the prior, which results in the statistically linearised filter [1], [17].

Remark 1. It should be noted that all the previously mentioned filters belong to the assumed density filtering (ADF) framework. In ADF, we make the approximation $p_{x|z}(\cdot) \approx q_{x|z}^{A,b,\Omega}(\cdot)$ to evaluate quantities of interest such as integrals or KLDs w.r.t. $p_{x|z}(\cdot)$ [1]. Otherwise, they are intractable and we would need to resort to Monte Carlo methods, which we want to avoid. For example, the mean of $p_{x|z}(\cdot)$ becomes

$$\int x p_{x|z}(x|z)\, dx \approx \int x q_{x|z}^{A,b,\Omega}(x|z)\, dx = \overline{u}.$$

1) Prior linearisation filter: Due to its relevance in this paper, we proceed to explain the PrLF from a KLD minimisation perspective. The parameters (A, b, Ω) of the PrLF, which are denoted as (A^+, b^+, Ω^+), are selected to minimise the KLD over the joint variable (x, z). This is indicated by the following proposition.

Proposition 2. *The solution to*

$$\begin{aligned}\left(A^+, b^+, \Omega^+\right) &= \underset{(A,b,\Omega)}{\arg\min}\, D\left(p_{x,z}\left\|q_{x,z}^{A,b,\Omega}\right.\right)\\ &= \underset{(A,b,\Omega)}{\arg\min} \int p_{x,z}(x,z)\log\frac{p_{x,z}(x,z)}{q_{x,z}^{A,b,\Omega}(x,z)}dxdz\end{aligned}$$

is given by SLR of $h(\cdot)$ *w.r.t.* $p_x(\cdot)$. *That is,*

$$A^+ = \Psi^T P^{-1} \tag{9}$$

$$b^+ = \overline{z} - A^+\overline{x} \tag{10}$$

$$\Omega^+ = \Phi - A^+ P \left(A^+\right)^T \tag{11}$$

where

$$\overline{z} = \int h(x)\, p_x(x)\, dx \tag{12}$$

$$\Psi = \int (x - \overline{x})(h(x) - \overline{z})^T p_x(x)\, dx \tag{13}$$

$$\Phi = \int (h(x) - \overline{z})(h(x) - \overline{z})^T p_x(x)\, dx. \tag{14}$$

Clearly, $q_{x,z}^{A^+,b^+,\Omega^+}(\cdot)$ matches the first two moments of $p_{x,z}(\cdot)$ as is required to minimise the KLD considered in Proposition 2 [16]. Integrals (12)-(14) can be approximated using sigma-points [1].

In order to gain more insight into how the PrLF selects (A, b, Ω), we also want to mention that [11]

$$\left(A^+, b^+\right) = \underset{(A,b)}{\arg\min}\, \mathrm{E}\left[(h(x) - Ax - b)^T (h(x) - Ax - b)\right] \tag{15}$$

$$\Omega^+ = \mathrm{E}\left[\left(h(x) - A^+x - b^+\right)\left(h(x) - A^+x - b^+\right)^T\right] \tag{16}$$

where the expectation is taken w.r.t. $p_x(\cdot)$. In other words, (A^+, b^+) represents the best linear approximation of $h(\cdot)$ in the sense of minimising its mean square error (MSE) and Ω^+ is the corresponding MSE matrix.

III. Posterior linearisation filter

As indicated in Section II-A, the PrLF selects the parameters of approximation (4) using SLR with respect to the prior. In Section III-A, we motivate why the SLR of the measurement function should be done with respect to the posterior, instead of the prior, to obtain a suitable approximation $\tilde{h}(x)$ in (4). In Section III-B, we analyse the PLF from a KLD minimisation perspective.

A. Motivation of the PLF

As we mentioned in the previous section, the SLR of a function with respect to a PDF provides us with the best linear approximation of the function in the region where the PDF lies. This fact is widely used in the update step of Bayesian filtering to get an enabling approximation of the form (4). The conventional way to apply SLR in the update step is given by the PrLF, see Proposition 2, in which we approximate the measurement function using SLR with respect to the prior. It should be highlighted that the PrLF does not make use of the measurement, which we know, to perform the enabling approximation. As the PrLF does not use all available information to perform the enabling approximation, we should expect that its performance deteriorates in some cases. It was proved in [9] that the PrLF and therefore all of its approximations do not work well with nonlinear measurement functions if the measurement noise is low enough. In the following, we provide one possible interpretation of this drawback of the PrLF in terms of the accuracy of the approximation (4). This interpretation motivates the introduction of the posterior linearisation filter (PLF).

If the measurement noise is low enough, the posterior PDF is considerably narrower than the prior PDF. Then, if the measurement function is nonlinear and we have performed SLR with respect to the prior, it is likely that the linear approximation of the measurement function is not accurate in the region where the posterior actually lies, which is our region of interest. That is, before we process the measurement, the PrLF provides us with the best linear approximation of the measurement function in our region of interest, which is indicated by the prior. However, when we receive the measurement, the region of interest changes according to the posterior, and the linear approximation given by SLR with respect to the prior is not necessarily accurate in the new region.

Intuition tells us that we should approximate the measurement function accurately in our region of interest. That is, the enabling approximation (4) should be chosen by the SLR w.r.t. the posterior PDF not the prior. The algorithm that uses the SLR of the measurement function with respect to the posterior in the enabling approximation (4) is referred to as PLF. It should also be noted that the PLF is intractable because it requires knowledge of the posterior to approximate the posterior. Nevertheless, we can design an iterative procedure to approximate the PLF as will be explained in Section IV.

1) Illustrative example: In order to clarify the concepts of the previous discussion, we find it convenient to use the following illustrative example. The prior PDF is Gaussian with mean $\overline{x} = 3$ and variance $P = 4$. The measurement equation is

$$z = ax^3 + \eta \tag{17}$$

where η is the measurement noise with variance $R = 0.1$ and $a = 0.01$. In this example, the required moments (12)-(14) can be calculated analytically [18] so we can use the exact PrLF instead of an approximation, such as the UKF or CKF.

We analyse the case where we measure $z = 1.5$. The prior and posterior PDFs are shown in Figure 1. In this figure, the posterior has been obtained by using a dense grid of points. This method is not generally practical because of its high computational burden. The PrLF approximates the posterior by the enabling approximation (4) using SLR of the measurement function with respect to the prior. The measurement function and its PrLF approximation are shown in Figure 2. The linearisation used in the PrLF would be the best linearisation of $h(\cdot)$ if our region of interest were given by the prior, which would be the case if we did not know the measurement. The fact is that we know that $z = 1.5$ and we therefore argue that the region of interest is now given by the posterior. The linearisation of $h(\cdot)$ with respect to the posterior, which is the enabling approximation of the PLF, is also plotted in Figure 2. It can be clearly seen that the PrLF linearisation is quite different from the linearisation we would like to use in (4). As the linearisation of the PrLF is not very accurate in our region of interest, it is not surprising that the resulting PrLF posterior approximation is poor, see Figure 3. On the contrary, if we use SLR of the measurement function with respect to the posterior (PLF), the resulting posterior approximation is rather accurate.

B. KLD considerations

In this section, we explain the PLF from a KLD perspective, which will be useful to motivate the iterated PLF in Section IV. The general idea is that we want to see how well we approximate the joint posterior density of the state and the state transformed by the measurement function using the KLD. This is of interest as the enabling approximation implicitly provides us with an approximation to $h(x)$. To do so, we consider the auxiliary variable

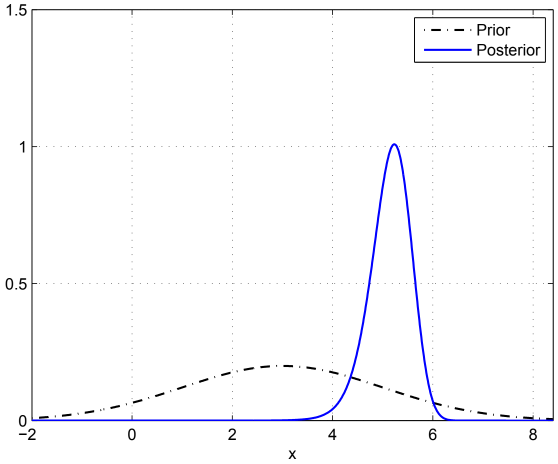

Figure 1: Prior and posterior for $z = 1.5$. The posterior is markedly narrower than the prior.

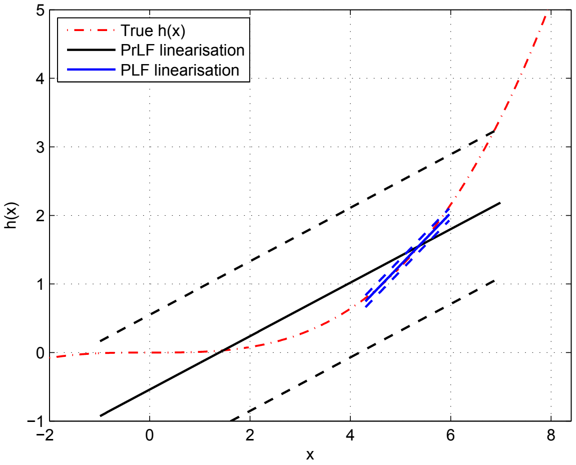

Figure 2: Measurement function and the PrLF and PLF approximations. We plot the linearisations in the 2σ-regions of the PDF that was used in SLR. The dashed lines indicate 2σ-regions of the error term in (4). The PrLF approximation is quite different from the PLF approximation, which gives us the best approximation of $h(\cdot)$ for the current measurement.

$$y = h(x) + \xi \tag{18}$$

where ξ is a zero-mean Gaussian noise independent of η with covariance matrix βI_{n_z}, I_{n_z} is the identity matrix of size n_z and $\beta > 0$ is a parameter to ensure that y has a density given x and we can therefore use the KLD. As $\beta \to 0$, $y \to h(x)$, so the joint posterior approximation of (x, y) given z in relation to the true joint posterior of (x, y) tells us how well we have approximated the distribution over the state transformed by the measurement function as well as the state.

The true joint PDF of the variables (x, y) given z factorises as

$$p_{x,y|z}(x, y|z) = p_{x|z}(x|z)\, p_{y|x}(y|x)\,. \tag{19}$$

We make the approximation

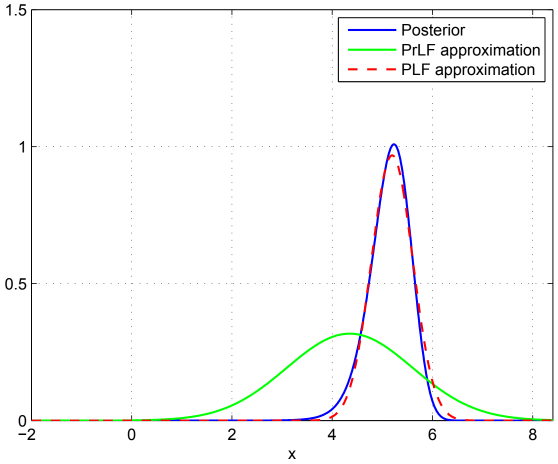

Figure 3: Posterior and the PrLF and PLF approximations. As the PrLF linearisation of $h(\cdot)$ is not accurate, the PrLF approximation is inaccurate. On the contrary, the PLF provides a quite accurate approximation.

- AP1:

$$q_{x,y|z}^{A,b,\Omega}(x,y|z) \approx q_{x|z}^{A,b,\Omega}(x|z)\, q_{y|x}^{A,b,\Omega}(y|x) \tag{20}$$

with $q_{x|z}^{A,b,\Omega}(x|z)$ given by (5) and

$$q_{y|x}^{A,b,\Omega}(y|x) = \mathcal{N}\left(y; Ax+b, \Omega+\beta I_{n_z}\right). \tag{21}$$

As shown in Appendix A, AP1 is accurate if $\|\Omega\|$ is small enough. Considering (x,y) jointly, the objective is to select (A,b,Ω) which minimises the joint KLD of $q_{x,y|z}^{A,b,\Omega}(\cdot)$ from $p_{x,y|z}(\cdot)$. That is, we would like to solve

$$\begin{aligned}&\underset{(A,b,\Omega)}{\arg\min}\, D\left(p_{x,y|z}\left\|q_{x,y|z}^{A,b,\Omega}\right.\right)\\ &= \underset{(A,b,\Omega)}{\arg\min}\left[D\left(p_{x|z}\left\|q_{x|z}^{A,b,\Omega}\right.\right)\right.\\ &\quad\left.+\int p_{x|z}(x|z) D\left(p_{y|x}\left\|q_{y|x}^{A,b,\Omega}\right.\right)dx\right]\end{aligned} \tag{22}$$

where we have used (20). Instead, the PLF selects (A,b,Ω) that minimises the second term in (22) as indicated by the following lemma.

Lemma 3. *The solution to*

$$\left(A^*,b^*,\Omega^*\right) = \underset{A,b,\Omega}{\arg\min}\int p_{x|z}(x|z) D\left(p_{y|x}\left\|q_{y|x}^{A,b,\Omega}\right.\right)dx$$

under AP1 is given by SLR w.r.t. the posterior $p_{x|z}(\cdot|z)$.

Lemma 3 can be easily proved by analogy with Proposition 2. We just have to simplify the KLDs in Proposition 2 and Lemma 3 such that we only consider the terms that depend on A,b,Ω.

The PLF therefore minimises a lower bound of $D\left(p_{x,y|z}\left\|q_{x,y|z}^{A,b,\Omega}\right.\right)$ under AP1 as

$$D\left(p_{x,y|z}\left\|q_{x,y|z}^{A,b,\Omega}\right.\right) \geq \int p_{x|z}(x|z) D\left(p_{y|x}\left\|q_{y|x}^{A,b,\Omega}\right.\right)dx.$$

As a result, selecting the best linear approximation of the measurement function given the measurement along with its mean square error matrix, which is what the PLF does, is equivalent to minimising a lower bound on the joint KLD. However, this approach is also impractical because we cannot evaluate integrals w.r.t. $p_{x|z}(x|z)$. Nevertheless, in the next section, we will see how useful criterion (22) is in the assumed density filtering framework.

IV. Iterated posterior linearisation filter

We derive the iterated PLF in Section IV-A and provide an illustrative example in Section IV-B. We discuss some aspects of the algorithm in Section IV-C. Finally, a stopping criterion for the iteration and a local convergence analysis are given in Sections IV-D and IV-E, respectively.

A. Derivation

Even though the PLF is intractable, it is the foundation of the IPLF, which iteratively minimises (22), under the usual approximation of the ADF framework explained in Remark 1. The idea is to build a sequence $\left(A^i,b^i,\Omega^i\right)$ $i\in\mathbb{N}$ such that at each iteration we improve the posterior approximation. More specifically, let us assume we have

$$q_{x|z}^{A^i,b^i,\Omega^i}(x|z) = \mathcal{N}\left(x;\overline{u}^i,W^i\right) \tag{23}$$

at step i. The aim is to obtain $\left(A^{i+1},b^{i+1},\Omega^{i+1}\right)$ that minimises (22) taking into account Remark 1. As a result, we approximate $p_{x|z}(x|z) \approx q_{x|z}^{A^i,b^i,\Omega^i}(x|z)$ in (22) to obtain

$$\begin{aligned}&\underset{\left(A^{i+1},b^{i+1},\Omega^{i+1}\right)}{\arg\min}\left[D\left(q_{x|z}^{A^i,b^i,\Omega^i}\left\|q_{x|z}^{A^{i+1},b^{i+1},\Omega^{i+1}}\right.\right)\right.\\ &\quad\left.+\int q_{x|z}^{A^i,b^i,\Omega^i}(x|z)\, D\left(p_{y|x}\left\|q_{y|x}^{A^{i+1},b^{i+1},\Omega^{i+1}}\right.\right)dx\right].\end{aligned} \tag{24}$$

We can in principle iterate this recursion until we reach a fixed point. Nonetheless, if we are sufficiently close to the fixed point

$$D\left(q_{x|z}^{A^i,b^i,\Omega^i}\left\|q_{x|z}^{A^{i+1},b^{i+1},\Omega^{i+1}}\right.\right) \approx 0.$$

As a result, the dominating term in (24) is the second one and we can make the approximation

- AP2

$$\begin{aligned}&D\left(q_{x|z}^{A^i,b^i,\Omega^i}\left\|q_{x|z}^{A^{i+1},b^{i+1},\Omega^{i+1}}\right.\right)\\ &\quad\ll \int q_{x|z}^{A^i,b^i,\Omega^i}(x|z)\, D\left(p_{y|x}\left\|q_{y|x}^{A^{i+1},b^{i+1},\Omega^{i+1}}\right.\right)dx.\end{aligned}$$

Under AP2 and applying Lemma 3, the solution to (24) is given by selecting $\left(A^{i+1},b^{i+1},\Omega^{i+1}\right)$ using SLR w.r.t. $q_{x|z}^{A^i,b^i,\Omega^i}(\cdot)$. This leads to an iterated algorithm, which is referred to as iterated PLF, in which we recursively perform SLR of $h(\cdot)$ w.r.t. the latest posterior approximation.

In practice, the required integrals of SLR, which are given by (12)-(14), can be approximated using any sigma-point method, e.g., the unscented transform (UT) [8]. We first select m sigma points $\mathcal{X}_1^i,...,\mathcal{X}_m^i$ and weights $\omega_1,...,\omega_m$ such that they match the moments $\overline{u}^i$ and W^i. There are multiple options to select these points and weights, which depend on the specific sigma point method [8], [14]. The transformed sigma points are

$$\mathcal{Z}_j^i = h\left(\mathcal{X}_j^i\right) \quad j = 1, ..., m \tag{25}$$

and

$$\overline{z}^i \approx \sum_{j=1}^{m} \omega_j \mathcal{Z}_j^i \tag{26}$$

$$\Psi^i \approx \sum_{j=1}^{m} \omega_j \left(\mathcal{X}_j^i - \overline{u}^i\right)\left(\mathcal{Z}_j^i - \overline{z}^i\right)^T \tag{27}$$

$$\Phi^i \approx \sum_{j=1}^{m} \omega_j \left(\mathcal{Z}_j^i - \overline{z}^i\right)\left(\mathcal{Z}_j^i - \overline{z}^i\right)^T \tag{28}$$

where $\overline{z}^i$, Ψ^i and Φ^i correspond to (12)-(14) where we use $\mathcal{N}\left(\cdot; \overline{u}^i, W^i\right)$ instead of $p_x(\cdot)$. The sigma point approximation to SLR w.r.t. $\mathcal{N}\left(\cdot; \overline{u}^i, W^i\right)$ corresponds to substituting (26)-(28) into (9)-(11). The recursion is initiated by performing SLR with respect to the prior as in a (nonlinear) Kalman filter. The steps of the IPLF are summarised in Algorithm 1. We want to remark that, in contrast with the PrLF (which uses the LMMSE estimator), the PLF and IPLF are not linear filters in z.

Algorithm 1 The update step of the IPLF

```
Input: Prior moments $\overline{u}^1 = \overline{x}$, $W^1 = P$.
Output: Posterior moments $\overline{u}^i$, $W^i$.
  - Initialise by setting $i = 1$.
  repeat
    • Approximate the SLR $A^i, b^i, \Omega^i$:
      – Select $m$ sigma points $\mathcal{X}_1^i, ..., \mathcal{X}_m^i$ and weights $\omega_1, ..., \omega_m$
        that match $\overline{u}^i$ and $W^i$ using a sigma-point method.
      – Calculate the transformed sigma points $\mathcal{Z}_j^i \quad j = 1, ..., m$
        using (25).
      – Approximate $\overline{z}^i$, $\Psi^i$ and $\Phi^i$ using (26)-(28).
      – Obtain $A^i, b^i, \Omega^i$ from (9)-(11) using $\mathcal{N}\left(\cdot; \overline{u}^i, W^i\right)$ instead
        of $p_x(\cdot)$.
    • Compute the posterior approximation moments $\overline{u}^{i+1}$ and
      $W^{i+1}$:
      – Use $A^i, b^i, \Omega^i$ in (6) and (7).
    • $i \leftarrow i + 1$
  until convergence, see Section IV-D.
```

B. Illustrative example

In this section, we analyse how the IPLF works in the illustrative example of Section III-A1. The integrals required for the SLRs are calculated analytically and the results of the IPLF recursion for the illustrative example are shown in Figure 4. The PrLF corresponds to the first step of the IPLF and is not a good approximation of the posterior. As we continue with the IPLF iteration, the posterior approximation becomes closer and closer to the true posterior. It is appealing that the improvement in performance of the IPLF with respect to the PrLF is quite significant with just a few more iterations.

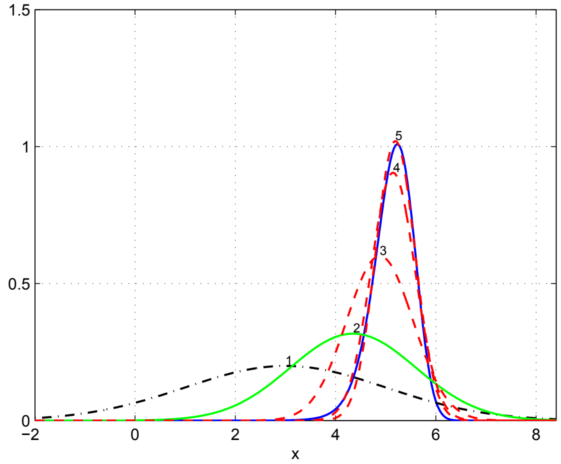

Figure 4: Illustration of IPLF. The true posterior is shown in blue, the prior in black, the PrLF posterior approximation in green and 3 more iterations of the IPLF in red. A number on the maximum of each PDF represents variable i in the IPLF recursion, see Algorithm 1. The PrLF is simply the first step of the recursion and is not a good approximation of the posterior. If we continue with the IPLF iteration, we attain an accurate approximation.

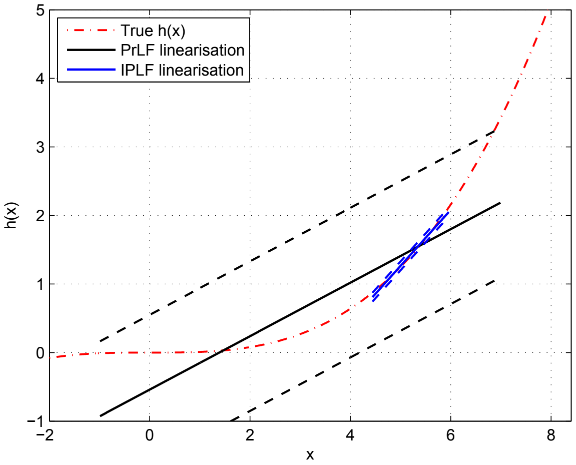

Figure 5: Comparison between the true measurement function and the linearisations used by the PrLF and the IPLF. We plot the linearisations in the 2σ-regions of the PDF that was used in SLR. The dashed lines indicate 2σ-regions of the error term in (4). The SLR of the IPLF is plotted with respect to the last posterior approximation, see Figure 4.

We also show the measurement function and the SLR used at the final step of the IPLF in Figure 5. Comparison of this figure with Figure 2 reveals remarkable similarity between the measurement function approximations of the PLF and the IPLF. Once we obtain the measurement, the PrLF linearisation is not a good approximation of $h(\cdot)$ because the state is expected to lie in a smaller area, which is given by the posterior, and the SLR in this area is quite different from the SLR with respect to the prior. On the contrary, IPLF seeks to linearise the measurement function with respect to the posterior. This is done by performing repeated SLRs with respect to the best available approximation to the posterior, which is given by the PDF at the last iteration. The resulting posterior approximation is much more accurate than for the PrLF as can be seen in Figure 4.

Finally, we want to show that the improvement indicated by Figures 4 and 5 results in a lower error in state estimation.

We estimate the root mean square error (RMSE) of several estimators averaged over the state and the measurement [19, Chap. 12], which is given by

$$\sqrt{\int\int \left(x-\overline{u}\left(z\right)\right)^{T}\left(x-\overline{u}\left(z\right)\right)p_{x,z}\left(x,z\right)dxdz} \tag{29}$$

where $\overline{u}\left(z\right)$ is the updated mean for the algorithms and we have written explicitly that it depends on z. We use Monte Carlo simulation to estimate (29) using 10^5 samples from $p_{x,z}\left(\cdot\right)$ and the results are: posterior mean 1.12, IPLF 1.18, PrLF 1.80. As expected, the IPLF outperforms the PrLF. The posterior mean produces the lowest error as it is the minimum MSE estimator.

C. Discussion

We proceed to discuss several characteristics of the IPLF as well as a possible modification of the algorithm. In order to derive the IPLF we have defined a new objective function, which is given by (22), instead of the ideal one, which is given by (8), and made several approximations. As indicated in Remark 1, in ADF, we use approximation $p_{x|z}\left(\cdot\right)\approx q_{x|z}^{A,b,\Omega}\left(\cdot\right)$. If we used the objective function (8) and the ADF approximation $p_{x|z}\left(\cdot\right)\approx q_{x|z}^{A,b,\Omega}\left(\cdot\right)$ to improve the selection of A,b,Ω as in (24), we would get that the KLD is minimised for the same values of A,b,Ω. Therefore, the objective function (8) is not useful for iterated improvements. In contrast, we realise that by making the enabling approximation (4), we are implicitly approximating the distribution over the state and the state transformed by measurement function. Consequently, we use auxiliary variable (18) to see how well the posterior density of the state and the measurement function is approximated. Approximation AP1 implies that the true posterior and its approximation factorise in the same form and allows us to perform iterated optimisations. As analysed in Appendix A, AP1 is accurate if $\left\|\Omega\right\|$ is small. In principle, if $\left\|\Omega\right\|$ is large, which implies that the measurement function nonlinearities are severe, the algorithm is making a big approximation so it does not necessarily work well. On the contrary, Approximation AP2 is always accurate if we are close to the fixed point. If we are far from the fixed point, it might be useful to relax Approximation AP2 and develop damped versions of the algorithm as the first term in (24) pulls the values of $A^{i+1},b^{i+1},\Omega^{i+1}$ closer to A^{i},b^{i},Ω^{i}. However, the development of this idea is beyond the scope of this paper.

Iterated sigma-point filters have previously been proposed in [20] and [21] but with ad-hoc approaches. In [20], only one set of sigma-points is generated and used to approximate the prior moments. Using these prior moment approximations the iteration proceeds similarly to the IEKF [5]. However, the analytical linearisation of the IEKF is replaced by an ad-hoc linearisation that mixes SLR with respect to the prior and analytical linearisation at the current MAP estimate. In [21], the iteration requires several ad-hoc parameters and conditions and the overall effect of the iteration is that several corrections are performed with the same measurement even though we observe it only once. Furthermore, the underlying philosophy of the filters in [20] and [21] is also different from ours as they attempt to find the MAP estimate while our objective is to minimise (22).

D. Stopping criterion

In this section, we propose a stopping criterion to determine when we should stop the IPLF iteration. The idea is that the IPLF recursion should finish if the change in the posterior approximation at a given iteration is small enough. In this section, we denote the ith Gaussian approximation as $\mathcal{N}_i\left(x\right)=\mathcal{N}\left(x;\overline{u}^{i},W^{i}\right)$. A usual method to evaluate the similarity between PDFs is the KLD. Therefore, we stop the recursion if

$$D\left(\mathcal{N}_i\left\|\mathcal{N}_{i+1}\right.\right)<\gamma \tag{30}$$

where γ is a threshold and [22]

$$\begin{aligned}D\left(\mathcal{N}_i\left\|\mathcal{N}_{i+1}\right.\right)&=\int\mathcal{N}_i\left(x\right)\log\frac{\mathcal{N}_i\left(x\right)}{\mathcal{N}_{i+1}\left(x\right)}dx\\&=\left[\operatorname{tr}\left(\left(W^{i+1}\right)^{-1}W^{i}\right)-\log\left(\frac{\left|W^{i}\right|}{\left|W^{i+1}\right|}\right)-n_x\right.\\&\left.\quad+\left(\overline{u}^{i+1}-\overline{u}^{i}\right)^{T}\left(W^{i+1}\right)^{-1}\left(\overline{u}^{i+1}-\overline{u}^{i}\right)\right]/2.\end{aligned}$$

In principle, we could have also chosen the KLD $D\left(\mathcal{N}_{i+1}\left\|\mathcal{N}_{i}\right.\right)$ rather than (30). However, in the examples of Section V, (30) works better. The main reason why this happens is that the KLD in (30) is large if $\mathcal{N}_{i+1}\left(\cdot\right)$ is small in the region where $\mathcal{N}_{i}\left(\cdot\right)$ is large [16]. As exemplified in Figure 4, $\mathcal{N}_{i+1}\left(\cdot\right)$ is expected to be more concentrated than $\mathcal{N}_{i}\left(\cdot\right)$ until the algorithm converges. Therefore, in order to increment the value of the KLD before the algorithm converges, it is convenient to use the KLD (30) rather than $D\left(\mathcal{N}_{i+1}\left\|\mathcal{N}_{i}\right.\right)$.

E. Convergence analysis

In this section we provide a discussion about the convergence properties of the IPLF. Clearly, if the measurement function is linear, the algorithm converges in one iteration to the solution, which is the PrLF update. If the moments (12)-(14) are approximated by the EKF instead of a sigma-point method, the IPLF is equivalent to the IEKF, which is Gauss–Newton method [5] and is not guaranteed to converge. The same happens if at some point $W^{i}\rightarrow 0$ (e.g. this could happen if $R\rightarrow 0$) as in this case the SLR w.r.t. the ith posterior approximation is in fact equal to the analytical linearisation at $\overline{u}^{i}$, which sets $\Omega^{i}=0$, so we obtain again a Gauss–Newton method.

For the general case, we provide a local convergence analysis in Appendix B. The result is that the algorithm converges if it is initiated sufficiently close to the fixed point and the eigenvalues of matrix Ξ, which is given by (78), are lower than one. Matrix Ξ has a very complex form and depends on the derivatives of (12)-(14) with respect to the components of the mean and covariance matrix at the ith iteration and at the optimal point. Clearly, convergence is also affected by the accuracy of the sigma-point method that is used to approximate integrals (12)-(14).

Despite the interesting properties of the IPLF, the convergence analysis also shows some drawbacks that should be mentioned. If the algorithm is not sufficiently close to the fixed point, the algorithm could diverge although a damped version of the algorithm, which was mentioned in Section IV-C, could help. In addition, the algorithm could converge to a bad fixed point, i.e., one PDF that does not represent the posterior accurately.

V. Numerical examples

In this section, we analyse the performance of the IPLF in two tracking scenarios. Specifically, we compare it with the following approximations of the PrLF: EKF, UKF and CKF. The UT of the UKF and IPLF has been implemented with $N_s = 2n_x + 1$ sigma-points and the weight of the sigma-point located on the mean is 1/3. The threshold $\gamma = 10^{-1}$ for IPLF1 and 10^{-5} for IPLF2. The prediction step of the IPLF is performed as in the UKF. In addition, we have also tested the IEKF (with 30 iterations) [5] and the Levenberg–Marquardt Kalman optimisation filter (LMKOF) [6], which are based on the MAP estimate.

A. Maneuvering target tracking

We consider an air-traffic control scenario, where an aircraft executes a maneuvering turn in a horizontal plane at a height h with respect to the radar. The state vector at time k is $x^k = \left[p_x^k, \dot{p}_x^k, p_y^k, \dot{p}_y^k, \Delta^k\right]^T$ where Δ^k is the turn rate at time k and, $\left[p_x^k, p_y^k\right]^T$ and $\left[\dot{p}_x^k, \dot{p}_y^k\right]^T$ are the position and velocity vector in the x and y coordinates at time k respectively.

The prior at time 0 is

$$p(x^0) = \mathcal{N}\left(x^0; \bar{x}^0, \Sigma^0\right) \tag{31}$$

where $\bar{x}^0 =$

$$\left[130\ (\mathrm{m}), 25\ (\mathrm{m/s}), -20\ (\mathrm{m}), 1\ (\mathrm{m/s}), -4\pi/180\ (\mathrm{rad/s})\right]^T$$

and $\Sigma^0 = \mathrm{diag}\left(\left[\sigma^2_{p_x}, \sigma^2_{\dot{p}_x}, \sigma^2_{p_y}, \sigma^2_{\dot{p}_y}, \sigma^2_{\Delta}\right]\right)$, with $\sigma^2_{p_x} = 5\,\mathrm{m}^2$, $\sigma^2_{\dot{p}_x} = 5\,\mathrm{m}^2/\mathrm{s}^2$, $\sigma^2_{p_y} = 2 \cdot 10^4\,\mathrm{m}^2$, $\sigma^2_{\dot{p}_y} = 10\,\mathrm{m}^2/\mathrm{s}^2$, $\sigma^2_{\Delta} = 10^{-7}\,\mathrm{rad}^2/\mathrm{s}^2$.

The kinematics of the turning motion are modeled by

$$x^{k+1} = F\left(\Delta^k\right) x^k + v^k \tag{32}$$

where

$$F(\Delta) = \begin{bmatrix} 1 & \frac{\sin\Delta\tau}{\Delta} & 0 & -\left(\frac{1-\cos\Delta\tau}{\Omega}\right) & 0 \\ 0 & \cos\Delta\tau & 0 & -\sin\Delta\tau & 0 \\ 0 & \left(\frac{1-\cos\Delta\tau}{\Delta}\right) & 1 & \frac{\sin\Delta\tau}{\Delta} & 0 \\ 0 & \sin\Delta\tau & 0 & \cos\Delta\tau & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{33}$$

and τ is the sampling period and v^k is the zero-mean Gaussian process noise at time k with covariance matrix

$$\mathbf{Q} = \begin{bmatrix} q_1\frac{\tau^3}{3} & q_1\frac{\tau^2}{2} & 0 & 0 & 0 \\ q_1\frac{\tau^2}{2} & q_1\tau & 0 & 0 & 0 \\ 0 & 0 & q_1\frac{\tau^3}{3} & q_1\frac{\tau^2}{2} & 0 \\ 0 & 0 & q_1\frac{\tau^2}{2} & q_1\tau & 0 \\ 0 & 0 & 0 & 0 & q_2\tau \end{bmatrix} \tag{34}$$

Table I: Measurement noise parameters

	$\sigma^2_{r,i}\,(\mathrm{m}^2)$	$\sigma^2_{\varphi,i}\,(\mathrm{rad}^2)$	$\sigma^2_{\theta,i}\,(\mathrm{rad}^2)$	$\sigma^2_{\dot{r},i}\,(\mathrm{m}^2/\mathrm{s}^2)$
$i = 1$	1000	$(30\pi/180)^2$	$(30\pi/180)^2$	100
$i = 2$	1000	$(10^{-3}\pi/180)^2$	$(30\pi/180)^2$	10^{-4}

where q_1 and q_2 are parameters of the motion model. As is usually assumed in tracking, v^k is independent of v^m if $m \neq k$. The dynamic parameters we use are $\tau = 0.2$ s, $q_1 = 0.5$ $\mathrm{m}^2/\mathrm{s}^3$, $q_2 = 10^{-6}\mathrm{rad}^2/\mathrm{s}^3$ and $h = 50\mathrm{m}$. The total number of time steps in the simulation is 130.

The sensor produces range, bearings, elevation and range rate measurements modelled by

$$z^k = \begin{bmatrix} \sqrt{\left(p_x^k\right)^2 + \left(p_y^k\right)^2 + h^2} \\ \mathrm{atan2}\left(p_y^k, p_x^k\right) \\ \mathrm{atan}\left(\frac{h}{\sqrt{\left(p_x^k\right)^2 + \left(p_y^k\right)^2}}\right) \\ \frac{p_x^k \dot{p}_x^k + p_y^k \dot{p}_y^k}{\sqrt{\left(p_x^k\right)^2 + \left(p_y^k\right)^2 + h^2}} \end{bmatrix} + \eta^k \tag{35}$$

where $\mathrm{atan2}(\cdot,\cdot)$ is the four-quadrant inverse tangent and η^k is the zero-mean Gaussian measurement noise at time k such that η^k is independent of η^m if $m \neq k$.

In order to illustrate how the filter performances vary with the accuracy of the measurements, we consider a scenario where the tracking system has two measurement modes with different accuracies. Sensors with different accuracies are usually used in sensor management applications. In the first type of measurement, the covariance matrix of η^k is $R_1 = \mathrm{diag}\left(\left[\sigma^2_{r,1}, \sigma^2_{\varphi,1}, \sigma^2_{\theta,1}, \sigma^2_{\dot{r},1}\right]\right)$ and, in the second, $R_2 = \mathrm{diag}\left(\left[\sigma^2_{r,2}, \sigma^2_{\varphi,2}, \sigma^2_{\theta,2}, \sigma^2_{\dot{r},2}\right]\right)$. In this example, measurements of the second type are performed every M time steps. In the simulations, we use the parameters shown in Table I and $M = 60$ and the first accurate measurement is taken at $k = 2$.

In order to evaluate the filter performance, we use 1000 Monte Carlo runs with 20 different trajectories, which have been drawn from (31) and (32). The trajectories are shown in Figure 6. EKF and IEKF diverge in all Monte Carlo runs so they are not further considered. The rest of the filters do not diverge. The root mean square (RMS) error for the position is provided in Figure 7. The IPLF and LMKOF have similar performance and clearly outperform the UKF and CKF.

We show the average number of iterations of the IPLF against time in Figure 8. Although the two versions of IPLF with different γ roughly provide the same error, the required number of iteration differs. IPLF2 has a lower threshold so it takes more iterations to converge. As expected, the number of iterations increases when the accurate measurements are taken. It should be noted that at most time steps IPLF1 corresponds to the UKF update as only one iteration is performed. Therefore, in this scenario, simply carrying out few more iterations at the right time steps can provide a significant increase in performance with a low increase in computational burden.

We also plot the normalised estimation error squared (NEES) of the position in Figure 9. While the UKF and CKF are optimistic filters in this example, the IPLF with

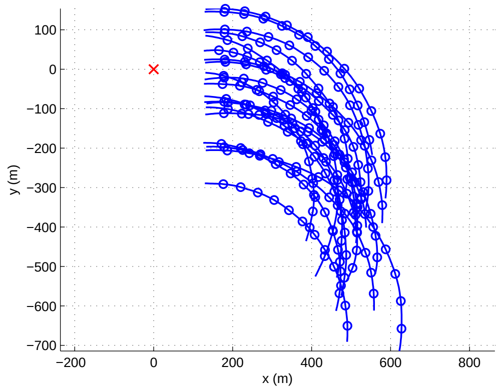

Figure 6: Turning target tracking scenario. The trajectories of the target are represented in blue. The target position every ten time steps is represented by a blue circle and its initial position by a filled blue circle. The sensor position is represented by a red cross.

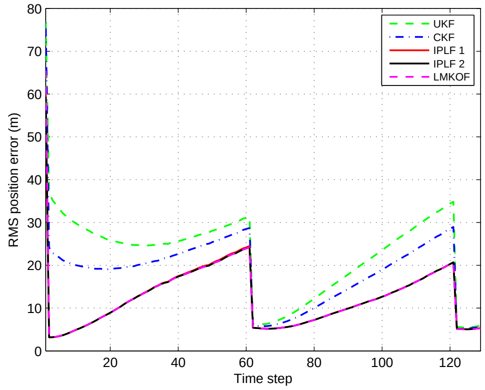

Figure 7: RMS position error against time for turning target example. IPLF and LMKOF provide a much lower error.

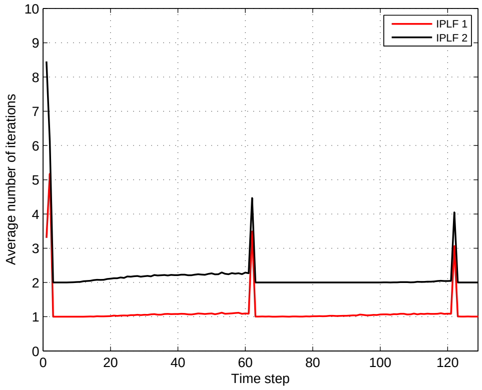

Figure 8: Average number of iterations of the IPLF against time for turning target example. For accurate measurements, more steps are carried out in the recursion.

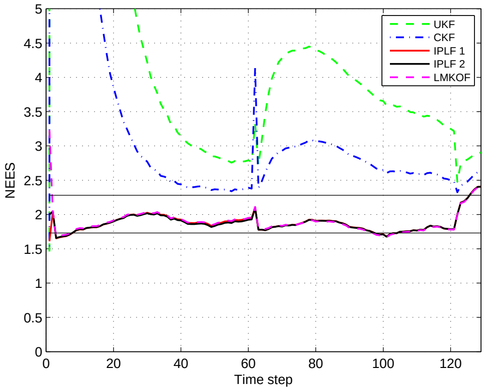

Figure 9: NEES of the position for turning target example. Two horizontal black lines indicate the 95% probability region.

the two considered threshold and LMKOF provide consistent approximations to the posterior PDF.

The execution times in milliseconds of a non-optimised Matlab implementation of the algorithms on a Intel Core i7 laptop are shown in Table II. Obviously, the UKF and CKF are the algorithms with lowest computational burden. The IPLF algorithms and LMKOF have the same performance in this scenario. Nevertheless, the computational burden of the IPLF is much lower than the LMKOF. In addition, the IPLF is straightforward to implement while the LMKOF is more laborious due to the need of computing more matrices and a square root sigma-point implementation [6].

B. Tracking with a sensor network

This example deals with target tracking using a sensor network. The state vector at time k is $x^k = \left[p_x^k, \dot{p}_x^k, p_y^k, \dot{p}_y^k\right]^T$.

Table II: Execution times for maneuvering target tracking scenario

	UKF	CKF	LMKOF	IPLF1	IPLF2
Time (ms)	38	38	104	49	73

The dynamic model of the target is the nearly-constant velocity model:

$$x^{k+1} = Fx^k + v^k \tag{36}$$

$$F = \begin{pmatrix} 1 & \tau \\ 0 & 1 \end{pmatrix} \otimes I_2 \tag{37}$$

where $\otimes$ is the Kronecker product and v^k is the process noise at time k. The process noise is zero-mean Gaussian distributed with covariance matrix

$$Q = q \begin{pmatrix} \tau^3/3 & \tau^2/2 \\ \tau^2/2 & \tau \end{pmatrix} \otimes I_2 \tag{38}$$

where q is a parameter of the model.

The deployment of the sensor network and the target trajectory used in this example are shown in Figure 10. There are $M = 25$ sensors and the measurement vector at time k is $z^k = \left[z_1^k, ..., z_M^k\right]^T$ where z_j^k is the measurement of the jth sensor at time k. Sensor j is located at $\left[\xi_{x,j}, \xi_{y,j}\right]^T$ and measures an acoustic signal emitted by the target and the measurement model is

$$z_j^k = h_j\left(x^k\right) + \eta_j^k \tag{39}$$

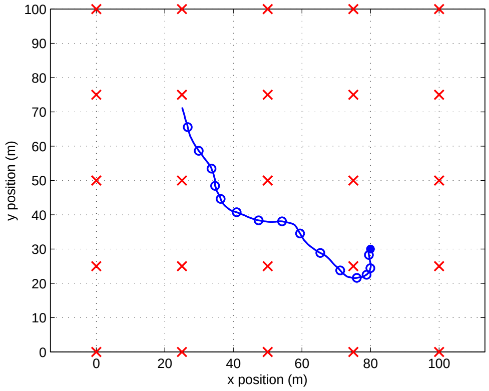

Figure 10: Tracking with a sensor network scenario. The trajectory of the target is represented in blue. The target position every ten time steps is represented by a blue circle and its initial position by a filled blue circle. The sensor positions are represented by red crosses.

Table III: Parameters for the turning target example

Parameter	Value
τ	0.5 s
q	0.04 $\mathrm{m}^2/\mathrm{s}^3$
σ_s^2	0.1 W
P_0	1000 W
d_0	1 m

where

$$h_j\left(x^k\right) = \begin{cases} \sqrt{\frac{P_0 d_0^2}{d_j^2\left(x^k\right)}} & d_j^2\left(x^k\right) > d_0^2 \\ \sqrt{P_0} & d_j^2\left(x^k\right) \leq d_0^2 \end{cases}$$

and η_j^k is an independent zero-mean Gaussian noise with variance σ_s^2, P_0 is the saturation power, d_0 is the distance at which this saturation power is produced and $d_j^2\left(x^k\right)$ is the square distance from the target x^k to the jth sensor

$$d_j^2\left(x^k\right) = \left(p_x^k - \xi_{x,j}\right)^2 + \left(p_y^k - \xi_{y,j}\right)^2. \tag{40}$$

The prior at time 0 is given by (31) with $\Sigma^0 = \mathrm{diag}\left(\left[\sigma_{p_x}^2, \sigma_{\dot{p}_x}^2, \sigma_{p_y}^2, \sigma_{\dot{p}_y}^2\right]\right)$, with $\sigma_{p_x}^2 = 49\,\mathrm{m}^2$, $\sigma_{\dot{p}_x}^2 = 4\,\mathrm{m}^2/\mathrm{s}^2$, $\sigma_{p_y}^2 = 1\,\mathrm{m}^2$, $\sigma_{\dot{p}_y}^2 = 2\,\mathrm{m}^2/\mathrm{s}^2$ and $\bar{x}^0$ is chosen randomly from a Gaussian PDF with mean identical to the true initial state and covariance matrix Σ^0. The parameters employed in this example are those given in Table III.

The RMS position errors against time for the filters are shown in Figure 11. On the whole, IPLF outperforms the rest of the filters. UKF and CKF attain the same error as IPLF after roughly 70 time steps. MAP-based algorithms, IEKF and LMKOF, are far from the performance of the IPLF. EKF is the filter that performs worst. In this scenario, the measurement model is quite nonlinear and, in general, the IPLF needs more iterations to converge as shown in Figure 12.

The execution times in milliseconds of our Matlab implementations of the algorithms are shown in Table IV. IPLF1 has the best trade-off between performance and execution time. EKF, UKF and CKF have lower computational complexity than IPLF1 but their performance is lower. IEKF and LMKOF have a higher execution time than IPLF1 and worse performance.

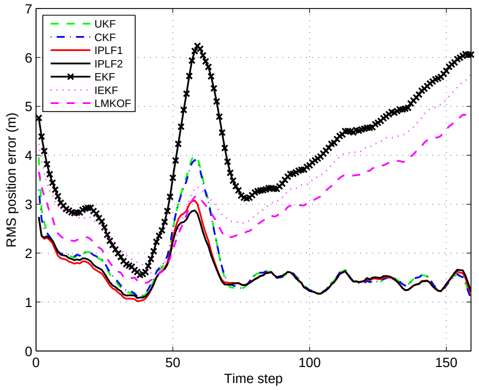

Figure 11: RMS position error in the sensor network scenario with $\sigma_s^2 = 0.1$. IPLF performs better than the rest of the algorithms.

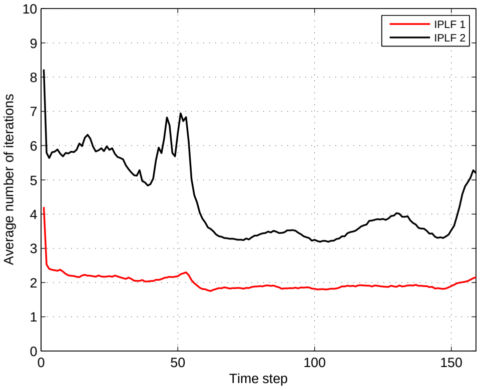

Figure 12: Average number of iterations of the IPLF against time for the sensor network scenario.

It is known that under some conditions, MAP filters tend to be optimal as the measurement noise tend to zero [7] but, otherwise, good performance is not ensured. This is probably the reason why IEKF and LMKOF do not perform as well as IPLF in this example. If we lower the noise variance to $\sigma_s^2 = 0.01$, the performance of the MAP filters should increase. This is what happens as shown in Figure 13. Now, IPLF, IEKF and LMKOF roughly provide the same error. Prior linearisation filters such as EKF, UKF and CKF have worse performance than the rest at the beginning but all the filters achieve the same error as time goes on.

VI. Conclusions

In this paper, we have developed an algorithm for the update step of Gaussian nonlinear filtering with additive Gaussian noise: the iterated posterior linearisation filter (IPLF). The IPLF iteratively improves the selection of the parameters in the enabling approximation by approximately minimising a Kullback–Leibler divergence. The result is that these parameters are chosen by statistical linear regression of the

Table IV: Execution times for tracking with a sensor network scenario

	UKF	CKF	IEKF	LMKOF	IPLF1	IPLF2
Time (ms)	57	57	280	480	95	213

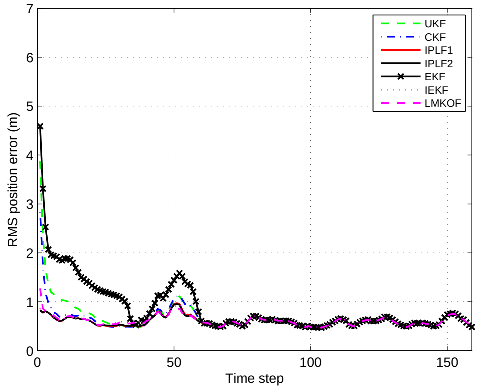

Figure 13: RMS position error in the sensor network scenario with $\sigma_s^2 = 0.01$. IPLF and MAP filters perform better than prior linearisation filters.

measurement function with respect to the current posterior approximation. Due to the iterated improvements in the enabling approximation, the IPLF should outperform the prior linearisation filter and its approximations, e.g., EKF, UKF and CKF. This is of special importance when the measurement noise is low enough and nonlinearities of the measurement function are non-negligible [9]. Moreover, the IPLF is straightforward to implement using sigma-points and the first step of the iteration corresponds to the usual sigma-point Kalman filters.

We want to remark that the IPLF is a new framework to perform Bayesian filtering and there are many possible ways to devise algorithms based on it, for example, design of sigma-point methods tailored to the IPLF problem or use of Gaussian processes, Fourier–Hermite series or Monte Carlo methods to approximate the required integrals. In addition, we have also indicated that, in principle, it is possible to develop damped versions of the IPLF.

Future work will address the extension of the IPLF to smoothing problems and Bayesian graphical models.

Appendix A

In this section, we see when Approximation AP1, see (20), is accurate. Under the enabling approximation (4), the posterior of (x, y) can be obtained from (1) and (18) using the linear KF equations.

$$q_{x,y|z}(x,y) = \mathcal{N}\left(\begin{bmatrix} x \\ y \end{bmatrix}; \begin{bmatrix} \overline{u} \\ \overline{u}_y \end{bmatrix}, \begin{bmatrix} W & W_{xy} \\ W_{xy}^T & W_y \end{bmatrix}\right) \quad (41)$$

where $\overline{u}$ and W are given by (6) and (7), respectively, and

$$\begin{aligned}\overline{u}_y = {} & A\overline{x} + b + \left(APA^T + \Omega\right)\left(APA^T + \Omega + R\right)^{-1} \\ & \times (z - A\overline{x} - b) \quad (42)\end{aligned}$$

$$W_{xy} = PA^T - PA^T\left(APA^T + \Omega + R\right)^{-1}\left(APA^T + \Omega\right) \quad (43)$$

$$\begin{aligned}W_y = {} & APA^T + \Omega + \beta I_{n_z} \\ & - \left(APA^T + \Omega\right)\left(APA^T + \Omega + R\right)^{-1}\left(APA^T + \Omega\right). \quad (44)\end{aligned}$$

We can also write $\overline{u}_y$, W_{xy} and W_y in terms of $\overline{u}$ and W as

$$\begin{aligned}\overline{u}_y = {} & A\overline{u} + b + \Omega\left(APA^T + \Omega + R\right)^{-1} \\ & \times (z - A\overline{x} - b). \quad (45)\end{aligned}$$

$$W_{xy} = WA^T - PA^T\left(APA^T + \Omega + R\right)^{-1}\Omega \quad (46)$$

$$\begin{aligned}W_y = {} & AWA^T + \Omega + \beta I_{n_z} \\ & - APA^T\left(APA^T + \Omega + R\right)^{-1}\Omega \\ & - \Omega\left(APA^T + \Omega + R\right)^{-1}APA^T \\ & - \Omega\left(APA^T + \Omega + R\right)^{-1}\Omega. \quad (47)\end{aligned}$$

If

$$\left\|\Omega\left(APA^T + \Omega + R\right)^{-1}\right\| \ll \frac{\|A\overline{u} + b\|}{\|(z - A\overline{x} - b)\|} \quad (48)$$

$$\left\|\left(APA^T + \Omega + R\right)^{-1}\Omega\right\| \ll \frac{\|WA^T\|}{\|PA^T\|} \quad (49)$$

$$\left\|\Omega\left(APA^T + \Omega + R\right)^{-1}\Omega\right\| \ll \left\|AWA^T + \Omega\right\|, \quad (50)$$

we get that

$$\overline{u}_y \approx A\overline{u} + b \quad (51)$$

$$W_{xy} \approx WA^T \quad (52)$$

$$W_y \approx AWA^T + \Omega + \beta I_{n_z}, \quad (53)$$

which results in (20) and (21). Conditions (48) and (49) are met if $\|\Omega\|$ is sufficiently small. For example, if we use the PLF to select A, b, Ω, these conditions imply that the nonlinear function $h(\cdot)$ should be mildly nonlinear (in the sense that its nonlinearities represented by $\|\Omega\|$ are small) in the region indicated by the posterior.

Appendix B

In this appendix, we provide a local convergence proof for the IPLF. This proof bears some resemblance to the convergence proofs for Newton's and Gauss–Newton methods explained in [23].

First, we define $e_j \in \mathbb{R}^{n_x \times 1}$ and $\tilde{e}_j \in \mathbb{R}^{n_x + n_z \times 1}$ as the vectors whose components are zero except component j, which is one. Then, we rearrange the elements of W^i into a vector as

$$w^i = \left[\ \left(W^i e_1\right)^T, \quad \ldots \quad, \left(W^i e_{n_x}\right)^T\ \right]^T \quad (54)$$

and define $y^i = \left[\left(\overline{u}^i\right)^T, \left(w^i\right)^T\right]^T$. We also define

$$r\left(y^i\right) = \begin{bmatrix} \Sigma^{-1/2}\left(y^i\right)\left(z - \overline{z}\left(y^i\right)\right) \\ P^{-1/2}\left(\overline{u}^i - \overline{x}\right) \end{bmatrix} \quad (55)$$

$$J\left(y^i\right) = \begin{bmatrix} -\Sigma^{-1/2}\left(y^i\right)A\left(y^i\right) \\ P^{-1/2} \end{bmatrix} \quad (56)$$

where

$$\Sigma\left(y^i\right) = \Omega\left(y^i\right) + R \quad (57)$$

and $\overline{z}\left(y^i\right)$, $\Omega\left(y^i\right)$, $A\left(y^i\right)$ are obtained using (9)-(14) with $\mathcal{N}\left(\cdot; \overline{u}^i, W^i\right)$ instead of $p_x(\cdot)$. We write

$$\widetilde{J}\left(y^i\right) = \left(J^T\left(y^i\right)J\left(y^i\right)\right)^{-1} \quad (58)$$

and the IPLF recursion becomes

$$y^{i+1}=\left[\begin{array}{c}\overline{u}^{i+1}\\ w^{i+1}\end{array}\right]=\left[\begin{array}{c}\overline{u}^{i}-\widetilde{J}\left(y^{i}\right)J^{T}\left(y^{i}\right)r\left(y^{i}\right)\\ \widetilde{J}\left(y^{i}\right)e_{1}\\ \vdots\\ \widetilde{J}\left(y^{i}\right)e_{n_{x}}\end{array}\right]. \quad (59)$$

We want to see when the recursion (59) converges.

A. Recursion for the mean

The fixed point of the recursion (59) is denoted as $y^{\star}=\left[\begin{array}{cc}\left(\overline{u}^{\star}\right)^{T} & \left(w^{\star}\right)^{T}\end{array}\right]^{T}$. We also denote $h_{i}=\left[\begin{array}{cc}\left(h_{i}^{u}\right)^{T} & \left(h_{i}^{w}\right)^{T}\end{array}\right]^{T}$ with $h_{i}^{u}=\overline{u}^{i}-\overline{u}^{\star}$ and $h_{i}^{w}=w^{i}-w^{\star}$. We denote the jth row of $J\left(\cdot\right)$ as $J_{j}\left(\cdot\right)$ and its Jacobian evaluated at y^{i} as

$$H_{j}\left(y^{i}\right)=\left(\nabla J_{j}\left(y\right)\big|_{y=y^{i}}\right)^{T}. \quad (60)$$

Then, the Taylor series expansion of $J_{j}^{T}\left(\cdot\right)$ around $y^{\star}$ becomes

$$J_{j}^{T}\left(y^{i}\right)=J_{j}^{T}\left(y^{\star}\right)+H_{j}\left(y^{\star}\right)h_{i}+O\left(\left\Vert h_{i}\right\Vert ^{2}\right). \quad (61)$$

Concatenating all the columns of $J^{T}(y^{i})$, we get

$$\begin{aligned}J^{T}(y^{i})=&\left[\begin{array}{ccc}H_{1}\left(y^{\star}\right)h_{i} & \ldots & H_{n_{z}+n_{x}}\left(y^{\star}\right)h_{i}\end{array}\right]\\ &+J^{T}(y^{\star})+O\left(\left\Vert h_{i}\right\Vert ^{2}\right)\end{aligned} \quad (62)$$

which can be written as

$$J^{T}(y^{i})=J^{T}(y^{\star})+\sum_{j=1}^{n_{z}+n_{x}}H_{j}\left(y^{\star}\right)h_{i}\tilde{e}_{j}^{T}+O\left(\left\Vert h_{i}\right\Vert ^{2}\right). \quad (63)$$

Multiplying by $r\left(y^{\star}\right)$ on both sides of (63), we get

$$J^{T}(y^{i})r\left(y^{\star}\right)=\sum_{j=1}^{n_{z}+n_{x}}\left(\tilde{e}_{j}^{T}r\left(y^{\star}\right)\right)H_{j}\left(y^{\star}\right)h_{i}+O\left(\left\Vert h_{i}\right\Vert ^{2}\right) \quad (64)$$

where we should note that $\tilde{e}_{j}^{T}r\left(y^{\star}\right)$ is a scalar and, due to (59),

$$J^{T}\left(y^{\star}\right)r\left(y^{\star}\right)=0. \quad (65)$$

Using the Taylor series of $r\left(\cdot\right)$ around y^{i}, we get

$$r\left(y^{\star}\right)=r\left(y^{i}\right)-M(y^{i})h_{i}+O\left(\left\Vert h_{i}\right\Vert ^{2}\right) \quad (66)$$

where the Jacobian $M(y^{i})=\left(\nabla r\left(y\right)\big|_{y=y^{i}}\right)^{T}$. We also get

$$J^{T}(y^{i})r\left(y^{\star}\right)=J^{T}(y^{i})r\left(y^{i}\right)-J^{T}(y^{i})M(y^{i})h_{i}+O\left(\left\Vert h_{i}\right\Vert ^{2}\right). \quad (67)$$

Substituting (64) into (67), we get

$$\begin{aligned}&\sum_{j=1}^{n_{z}+n_{x}}\left(\tilde{e}_{j}^{T}r\left(y^{\star}\right)\right)H_{j}\left(y^{\star}\right)h_{i}+O\left(\left\Vert h_{i}\right\Vert ^{2}\right)\\ &\quad=J^{T}(y^{i})r\left(y^{i}\right)-J^{T}(y^{i})M(y^{i})h_{i}.\end{aligned} \quad (68)$$

Multiplying by $\widetilde{J}\left(y^{i}\right)$, we get

$$\begin{aligned}&\widetilde{J}\left(y^{i}\right)\sum_{j=1}^{n_{z}+n_{x}}\left(\tilde{e}_{j}^{T}r\left(y^{\star}\right)\right)H_{j}\left(y^{\star}\right)h_{i}+O\left(\left\Vert h_{i}\right\Vert ^{2}\right)\\ &\quad=\overline{u}^{i}-\overline{u}^{i+1}-\widetilde{J}\left(y^{i}\right)J^{T}(y^{i})M(y^{i})h_{i}\end{aligned} \quad (69)$$

where we have used (59). We can write

$$M(y^{i})=\left[\begin{array}{cc}M^{x}(y^{i}) & M^{w}\left(y^{i}\right)\end{array}\right] \quad (70)$$

where $M^{x}(y^{i})$ corresponds to the first n_{x} columns and $M^{w}\left(y^{i}\right)$ denotes the rest. We also define $L(y^{i})=M^{x}(y^{i})-J(y^{i})$ and, using (58), we write (69) as

$$\begin{aligned}&\widetilde{J}\left(y^{i}\right)\sum_{j=1}^{n_{z}+n_{x}}\left(\tilde{e}_{j}^{T}r\left(y^{\star}\right)\right)H_{j}\left(y^{\star}\right)h_{i}+O\left(\left\Vert h_{i}\right\Vert ^{2}\right)\\ &\quad=-h_{i+1}^{u}-\widetilde{J}\left(y^{i}\right)J^{T}(y^{i})\left[\begin{array}{cc}L(y^{i}), & M^{w}\left(y^{i}\right)\end{array}\right]h_{i}.\end{aligned} \quad (71)$$

Then

$$h_{i+1}^{u}=N\left(y^{\star},y^{i}\right)h_{i}+O\left(\left\Vert h_{i}\right\Vert ^{2}\right) \quad (72)$$

where

$$\begin{aligned}N\left(y^{\star},y^{i}\right)=-\widetilde{J}\left(y^{i}\right)\Bigg(&\sum_{j=1}^{n_{z}+n_{x}}\left(\tilde{e}_{j}^{T}r\left(y^{\star}\right)\right)H_{j}\left(y^{\star}\right)\\ &+J^{T}(y^{i})\left[\begin{array}{cc}L(y^{i}), & M^{w}\left(y^{i}\right)\end{array}\right]\Bigg).\end{aligned} \quad (73)$$

B. Recursion for the covariance

The Taylor series of $\widetilde{J}\left(\cdot\right)$ around $y^{\star}$ is

$$\widetilde{J}\left(y^{i}\right)=\widetilde{J}\left(y^{\star}\right)+\sum_{j=1}^{n_{x}}T_{j}\left(y^{\star}\right)h_{i}e_{j}^{T}+O\left(\left\Vert h_{i}\right\Vert ^{2}\right) \quad (74)$$

where $\widetilde{J}\left(y^{i}\right)\in\mathbb{R}^{n_{x}\times n_{x}}$ and $T_{j}\left(y^{\star}\right)$ is the Jacobian of the jth column of $\widetilde{J}\left(\cdot\right)$ evaluated at $y^{\star}$. Multiplying by e_{j}, $j\in\{1,..,n_{x}\}$, and using (59), we get

$$w^{i+1}=\left[\begin{array}{c}T_{1}\left(y^{\star}\right)\\ \vdots\\ T_{n_{x}}\left(y^{\star}\right)\end{array}\right]h_{i}+w^{\star}+O\left(\left\Vert h_{i}\right\Vert ^{2}\right) \quad (75)$$

$$h_{i+1}^{w}=\left[\begin{array}{c}T_{1}\left(y^{\star}\right)\\ \vdots\\ T_{n_{x}}\left(y^{\star}\right)\end{array}\right]h_{i}+O\left(\left\Vert h_{i}\right\Vert ^{2}\right). \quad (76)$$

C. Final result

Concatenating (72) and (76), we get

$$h_{i+1}=\Xi\left(y^{\star},y^{i}\right)h_{i}+O\left(\left\Vert h_{i}\right\Vert ^{2}\right) \quad (77)$$

where

$$\Xi\left(y^{\star},y^{i}\right)=\left[\begin{array}{c}N\left(y^{\star},y^{i}\right)\\ T_{1}\left(y^{\star}\right)\\ \vdots\\ T_{n_{x}}\left(y^{\star}\right)\end{array}\right]. \quad (78)$$

Therefore,

$$\|h_{i+1}\| \leq \left\|\Xi\left(y^{\star},y^{i}\right)\right\| \|h_i\| + O\left(\|h_i\|^2\right). \qquad (79)$$

If the absolute values of the eigenvalues of $\Xi\left(y^{\star},y^{i}\right)$ are lower than one, $\left\|\Xi\left(y^{\star},y^{i}\right)\right\| < 1$ and we get linear convergence [24].

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] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, Feb. 2002.

[4] A. F. García-Fernández, L. Svensson, and M. R. Morelande, "Iterated statistical linear regression for Bayesian updates," in *17th International Conference on Information Fusion*, 2014.

[5] 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.

[6] A. F. García-Fernández and L. Svensson, "Gaussian MAP filtering using Kalman optimization," *IEEE Transactions on Automatic Control*, vol. 60, no. 5, pp. 1336–1349, May 2015.

[7] A. M. Walker, "On the asymptotic behaviour of posterior distributions," *Journal of the Royal Statistical Society. Series B (Methodological).*, vol. 31, no. 1, pp. 80–88, 1969.

[8] 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.

[9] M. R. Morelande and A. F. García-Fernández, "Analysis of Kalman filter approximations for nonlinear measurements," *IEEE Transactions on Signal Processing*, vol. 61, no. 22, pp. 5477–5484, Nov. 2013.

[10] 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.

[11] 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.

[12] J. H. Kotecha and P. M. Djuric, "Gaussian particle filtering," *IEEE Transactions on Signal Processing*, vol. 51, no. 10, pp. 2592–2601, Oct. 2003.

[13] S. Julier, J. Uhlmann, and H. F. Durrant-Whyte, "A new method for the nonlinear transformation of means and covariances in filters and estimators," *IEEE Transactions on Automatic Control*, vol. 45, no. 3, pp. 477–482, Mar. 2000.

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

[15] T. Lefebvre, H. Bruyninckx, and J. De Schuller, "Comment on "a new method for the nonlinear transformation of means and covariances in filters and estimators" [and authors' reply]," *IEEE Transactions on Automatic Control*, vol. 47, no. 8, pp. 1406–1409, Aug. 2002.

[16] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer Science + Business Media, 2006.

[17] A. Gelb, *Applied optimal estimation*. The MIT Press, 1974.

[18] A. F. García-Fernández, "Detection and tracking of multiple targets using wireless sensor networks," Ph.D. dissertation, Universidad Politécnica de Madrid, 2011. [Online]. Available: http://oa.upm.es/9823/

[19] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. Prentice-Hall, 1993.

[20] G. Sibley, G. Sukhatme, and L. Matthies, "The iterated sigma point Kalman filter with applications to long range stereo," in *Proceedings of Robotics: Science and Systems*, 2006.

[21] R. Zhan and J. Wan, "Iterated unscented Kalman filter for passive target tracking," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 43, no. 3, pp. 1155–1163, July 2007.

[22] J. Hershey and P. Olsen, "Approximating the Kullback Leibler divergence between Gaussian mixture models," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 4, April 2007, pp. 317–320.

[23] R. Fletcher, *Practical methods of optimization*. John Wiley & Sons, 2000.

[24] J. Nocedal and S. J. Wright, *Numerical Optimization*. Springer, 1999.

Ángel F. García-Fernández received the telecommunication engineering degree (with honours) and the Ph.D. degree from Universidad Politécnica de Madrid, Madrid, Spain, in 2007 and 2011, respectively.
He is currently a Research Associate in the Department of Electrical and Computer Engineering at Curtin University, Perth, Australia. His main research activities and interests are in the area of Bayesian nonlinear inference.

Lennart Svensson was born in Älvängen, Sweden in 1976. He received the M.S. degree in electrical engineering in 1999 and the Ph.D. degree in 2004, both from Chalmers University of Technology, Gothenburg, Sweden.
He is currently Associate Professor at the Signal Processing group, again at Chalmers University of Technology. His research interests include Bayesian inference in general, and nonlinear filtering and tracking in particular.

Mark R. Morelande received the B.Eng. degree in aerospace avionics from Queensland University of Technology and the Ph.D. in electrical engineering from Curtin University of Technology.
He has worked as a research fellow at the Centre for Eye Research, Queensland University of Technology and the School of Electrical and Electronic Engineering at the University of Melbourne. He is now a Senior Research Fellow in the School of Electrical and Computer Engineering at RMIT University. He has co-authored a book and over 100 research articles in the areas of non-stationary signal analysis and Bayesian estimation with particular emphasis on multiple target tracking and sequential Monte Carlo methods.

Simo Särkkä received his Master of Science (Tech.) degree (with distinction) in engineering physics and mathematics, and Doctor of Science (Tech.) degree (with distinction) in electrical and communications engineering from Helsinki University of Technology, Espoo, Finland, in 2000 and 2006, respectively.
From 2000 to 2010 he worked with Nokia Ltd., Indagon Ltd., and Nalco Company in various industrial research projects related to telecommunications, positioning systems, and industrial process control. From 2010 to 2013 he worked as a Senior Researcher with the Department of Biomedical Engineering and Computational Science (BECS) at Aalto University, Finland.Currently, Dr. Särkkä is an Associate Professor and Academy Research Fellow with Aalto University, Technical Advisor and Director of IndoorAtlas Ltd., and an Adjunct Professor with Tampere University of Technology and Lappeenranta University of Technology. In 2013 he was a Visiting Professor with the Department of Statistics of Oxford University and in 2011 he was a Visiting Scholar with the Department of Engineering at the University of Cambridge, UK. His research interests are in multi-sensor data processing systems with applications in location sensing, machine learning, inverse problems, and brain imaging. He has authored or coauthored 50 peer-reviewed scientific articles and has 3 granted patents. His first book "Bayesian Filtering and Smoothing" was recently published via the Cambridge University Press. He is a Senior Member of IEEE.