An approximation of the Bayesian state observer with Markov chain Monte Carlo propagation stage

L. Ecker ∗ K. Schlacher ∗

Abstract: The state estimation problem for nonlinear systems with stochastic uncertainties can be formulated in the Bayesian framework, where the objective is to replace the state completely by its probability density function. Without the restriction to selected system classes and disturbance properties, the Bayesian estimator is particularly interesting for highly nonlinear systems with non-Gaussian noise. The main limitations of Bayesian filters are the significant computational costs and the implementation problems for higher dimensional systems. The present paper introduces a piecewise linear approximation of the Bayesian state observer with Markov chain Monte Carlo propagation stage and kernel density estimation. These methods are suitable for the prediction of multivariate probability density functions. The piecewise linear approximation and the proposed algorithms can increase the estimation performance at reasonable computational cost. The estimation performance is demonstrated in a benchmark comparing the Bayesian state observer with an extended Kalman filter and a particle filter.

Keywords: nonlinear state estimation, Bayesian state observer, Bayesian statistics, stochastic process, multithreading, Markov chain Monte Carlo method, kernel density estimation

1. INTRODUCTION

Many processes from a wide range of scientific fields are modelled as nonlinear stochastic dynamic systems. Among other things, a mathematical model offers the possibility of a system-theoretical investigation. In this way, for example, influences can be estimated, predictions can be made, and trajectories or control strategies for system operation can be designed. For these purposes, knowledge of the entire system state is often necessary. In some cases, however, it is not possible to measure all of the state variables associated with the model, whether for technical, environmental, or economic reasons. In addition, the available measurements may be superimposed by nondeterministic influences such as measurement noise or process disturbances. For the design and implementation of a state observer, sampled-data systems of continuous-time processes or inherent discrete-time systems are considered. A fundamental formulation of the nonlinear state estimation problem is given in the Bayesian framework. The idea of the Bayesian state observer (BSO) is to describe the estimation of the state completely by its probability density function (pdf). The algorithm consists of two stages, one propagating the density to the next time step and the other inferring additional state information through measurements. For its most general form, i.e., for the nonlinear and non-Gaussian noise scenario, the prediction step corresponds to the Chapman–Kolmogorov equation and the update step complies with Bayesian inference. An analytically computable solution in closed-form is generally hardly available. Therefore, several approaches are used to approximate the Bayesian state estimation problem towards a numerical implementation. The increased computational effort compared to the extended Kalman filter (EKF), for example, is a trade off for improved estimation performance. In this paper, a grid-based approximation of the BSO with multilinear interpolation of the pdfs on a finite set of grid nodes is presented. While in this scenario the inference step can be efficiently performed by a multi-threaded algorithm, the propagation step is accomplished by a Markov chain Monte Carlo (MCMC) method, where a large number of samples distributed according to the a posteriori density of the state are propagated through the dynamic equation and reconstructed into an a priori pdf using kernel density estimation (KDE).

Therefore, the paper is structured as follows. In the remaining part of the introduction a brief survey of the Bayesian filtering theory is given, where different known numerical approximations are referenced. In addition, the notation used from estimation and probability theory is briefly explained. In Section 2, the fundamental formulation of the Bayesian state observer is introduced and the two main stages of the filter are explained. In Section 3, a grid-based observer with Markov chain Monte Carlo propagation stage and kernel density estimation is presented as numerical approximation of the Bayesian state estimation problem. Finally, experimental results of the proposed Bayesian approximation for a benchmark example are presented in Section 4.

* This work has been supported by the COMET-K2 Center of the Linz Center of Mechatronics (LCM) funded by the Austrian federal government and the federal state of Upper Austria.
1.1 Bayesian filtering theory

A general approach to the state estimation problem for nonlinear stochastic dynamic systems was formulated more than four decades ago in the Bayesian framework, see, e.g., Ho and Lee (1964). The analytical equations of the Bayesian state observer, described e.g., in Ristic et al. (2003), form a basic unified framework for many of the well-known filters, such as the Kalman filter and the particle filter, see Simon (2006). However, in the most general case of the Bayesian state estimation problem, there is often no exact parametric solution. An extensive overview of Bayesian filtering theory can be found, e.g., in Chen (2003), where the author further discusses a wide range of different numerical approximations. The BSO approximations can be roughly divided into grid-based methods, point-mass methods, parametric methods and non-parametric methods, see Stano et al. (2013). While grid-based and point-mass methods typically involve approximation of the state space, see, e.g., Kramer and Sorenson (1988), parametric and non-parametric methods look for a solution to the estimation problem in a finite-dimensional parameter space or are based on the idea of Monte Carlo integration. Among the best-known non-parametric methods are the various variants of the particle filter, which also include bootstrap filters, see Gordon et al. (1993), and sequential Monte Carlo filters, see Andreu et al. (2001).

1.2 Mathematical notation

The notation and main formulas from probability and estimation theory are briefly mentioned. A detailed description can be found, e.g., in Papoulis and Pillai (2002) and Simon (2006). The system state and the disturbances are interpreted as tuples of continuous real-valued random variables. Given a multivariate random variable X with domain \(\mathcal{D}\), the monotonically increasing probability distribution function, i.e., the probability that \(X \leq x\) is denoted by \(P_X(x) = \Pr(X \leq x) \in [0,1]\). Its corresponding multivariate probability density function is defined as the derivative \(p_X(x) = \frac{\partial}{\partial x} P_X(x) = \frac{\partial^n}{\partial x_1...\partial x_n} P_X(x_1,...,x_n) \geq 0\) with the property \(\int_{\mathcal{D}} p_X(x) dx = 1\). The subscripts are omitted when the associated random variables are evident from the context, i.e., the pdfs of the two multivariate random variables \(X\) and \(Y\) are denoted \(p(x)\) and \(p(y)\), respectively. Moreover, the conditional pdf of \(X\) given \(Y\) is interpreted as \(p(x|y) = \frac{p(x,y)}{p(y)} = \frac{\Pr(X \leq x|Y = y)}{p(y)}\). Given a diffeomorphic mapping \(g : \mathcal{D}_X \rightarrow \mathcal{D}_V, x \mapsto g(x) = v\), the statistics of the random variable \(V = g(X)\) can be expressed in terms of the statistics of \(X\), i.e.,

\[
p(v) = (p(x) \det(\partial g)^{-1}) \circ g^{-1}(v)
\]

where \(\partial g\) denotes the Jacobian matrix of the transformation \(g\), see, e.g., Papoulis and Pillai (2002). A connection between the joint statistics of \(X\) and \(V\), i.e., \(p(x,v)\), and its conditional pdf \(p(x|v)\) and \(p(v|x)\), respectively, is given by the total probability \(p(v) = \int_{\mathcal{D}_X} p(x|v)p(x) dx\) and Bayes’ theorem according to

\[
p(x|v) = \frac{p(x,v)}{p(v)} = \frac{p(v|x)p(x)}{\int_{\mathcal{D}_X} p(v|x)p(x) dx}.
\]

The estimates or approximations of stochastic variables are noted with the hat symbol, e.g., \(\hat{x}\). Moreover, considering sampled-data systems with \(k\) denoting a particular point in time, estimates can be distinguished into a priori and posteriori quantities, e.g., \(\hat{x}_k^+\) and \(\hat{x}_k^-\). The phrase a priori indicates that the estimate \(\hat{x}_k^+\) takes all measurements \(Y_{k-1} = \{y_0, \ldots, y_{k-1}\}\) up to time \(k-1\) into account. In contrast, the a posteriori term designates an estimate \(\hat{x}_k^-\) with measurements \(Y_k = \{y_k, \ldots, y_k\}\) up to time \(k\).

2. BAYESIAN STATE OBSERVER

The intention is to design a state observer for the nonlinear system described by the state and measurement equations

\[
x_{k+1} = f_k(x_k, u_k, w_k) \quad (3a)
\]
\[
y_k = h_k(x_k, v_k) \quad (3b)
\]

with index \(k\) denoting the current time, the system state \(x_k \in \mathbb{R}^{n_x}\), the system input \(u_k \in \mathbb{R}^{n_u}\) and output \(y_k \in \mathbb{R}^{n_y}\), the random process disturbance or noise \(w_k \in \mathbb{R}^{n_w}\), and the random measurement noise \(v_k \in \mathbb{R}^{n_v}\). The model (3) corresponds to a stochastic process with random variables \(x_0, w_k, v_k\) and deterministic input \(u_k\), which implies that the state \(x_0\) is also a random variable. The actual values of the input \(u_k\) and the output \(y_k\) at the current time \(k\) are known, while those of the non-deterministic noises \(w_k\) and \(v_k\) are not available. Instead, only knowledge of the probability density functions \(p(w_k)\) and \(p(v_k)\) is assumed. The same applies for the initial value of the state \(x_0\), which is represented by its density \(p(x_0)\).

After the initialization process, the Bayesian state observer algorithm is divided into two steps that are executed alternately. As with the Kalman filter, these steps are referred to as prediction and update. The prediction or propagation step intends to estimate the a priori state \(\hat{x}_k^-\) at the time \(k\) given the best estimation of the a posteriori state \(\hat{x}_{k-1}^+\) of the previous time step \(k-1\). On the other hand, the update or inference step incorporates a received measurement \(y_k\) into its estimate, i.e., it takes the a priori state \(\hat{x}_k^-\) and infers the current measurement \(y_k\) to obtain the a posteriori state \(\hat{x}_k^+\). The Bayesian state observer does not restrict its estimations to a few stochastic moments. Rather, the descriptive pdf of the state is propagated from one time step to the next, also inferring measurements obtained. At each time step, various methods can be used to approximate an optimal estimate \(\hat{x}_k^+\) from an a posteriori pdf.

2.1 Bayesian Prediction

Given the probability density of the state at time \(k-1\), i.e., \(p(x_{k-1}|Y_{k-1})\), the objective of the prediction or propagation step within the Bayesian state observer algorithm is to predict the conditional pdf of \(x_k\) given all the measurements prior to time \(k\), i.e., \(p(x_k|Y_{k-1})\). An a priori estimate \(\hat{x}_k^-\) can be abstracted from the conditional pdf \(p(x_k|Y_{k-1})\) according to various optimality criteria. For instance, the estimate of the maximum a posteriori probability (MAP) is given by

\[
\hat{x}_k^- = \arg\max_{x_k \in \mathcal{D}} p(x_k|Y_{k-1}).
\]

Nevertheless, the expectation value as in the Kalman filter would also be conceivable, although caution is re-
quired with bimodal probability densities. However, even the MAP estimate of a bimodal distribution in which the highest mode is uncharacteristic of most of the distribution can have dramatic effects on subsequent estimates. The analytical propagation equation is obtained as marginal probability density function of \( p(x_k, x_{k-1}) \), see (5), where Bayes’ Rule allows to rewrite the density as \( p(x_k, x_{k-1}) = p(x_k | x_{k-1}) p(x_{k-1}) \). The Markov process characteristics of (3a), i.e., the probability of the current state given the immediately previous one is conditionally independent of earlier states as well as outputs, yields \( p(x_k | x_{k-1} , Y_{k-1}) = p(x_k | x_{k-1}) \). Therefore, a recursive assignment is given by

\[
p(x_k | Y_{k-1}) = \int_D p(x_k, x_{k-1} | Y_{k-1}) \, dx_{k-1} \\
= \int_D p(x_k | x_{k-1}) p(x_{k-1} | Y_{k-1}) \, dx_{k-1},
\]

where \( p(x_{k-1} | Y_{k-1}) \) is the known previous posterior pdf and \( p(x_k | x_{k-1}) \) is the pdf of \( x_k \) given a state \( x_{k-1} \). This density is known, since it is derived from the given probability density \( p(u_k) \) and the state mapping (3a). For this purpose, we assume that the mapping (3a) with fixed state \( x_k \) and input \( u_k \) is diffeomorphic and denote the associated inverting map by \( w_{k+1} = f_k^* (x_k, u_k, x_{k+1}) \). The pdf \( p(x_k | x_{k-1}) \), see also (1), is determined by

\[
p(x_k | x_{k-1}) = \left( p(u_k) \det(\partial f_k^{-1}) \right) \delta_{x_k}^{f_k^{-1} (x_{k-1}, u_k, x_k)},
\]

where \( \partial f_k \) corresponds to the Jacobian matrix with entries \( \partial f_{i,j,k} = \frac{\partial f_{i,k}}{\partial x_j} \). Therefore, the numerator \( p(y_k | x_k) \) weights the likelihood of the current measurement \( y_k \). The diffeomorphic inverting map of (3b) with fixed state \( x_k \) and input \( u_k \) is denoted by \( v_k = h_k^* (x_k, u_k, y_k) \). The pdf \( p(y_k | x_k) \) is thus given by

\[
p(y_k | x_k) = \left( p(v_k) \det(\partial h_k^{-1}) \right) \delta_{v_k}^{h_k^* (x_k, u_k, y_k)},
\]

where \( \partial h_k \) denotes the Jacobian matrix, whose \((i,j)\)-th entry is given by \( \partial h_{i,j,k} = \frac{\partial h_{i,k}}{\partial u_j} \).

3. APPROXIMATION AND IMPLEMENTATION

An exact solution to the Bayesian observation problem is available only for a few exceptions. Therefore, several approximation approaches have been proposed to obtain capable algorithms with improved estimation performance and acceptable computational cost, see, e.g., Chen (2003). In particular, the multidimensional integration terms of the propagation stage, see (6), are a thorn in the flesh for the implementation of the state observer. In this section, a grid-based piecewise linear approximation of the Bayesian state observer is presented using a Markov chain Monte Carlo method along with a kernel density estimation algorithm that replaces the usual propagation stage. The advantage of this approach is that the algorithms can be easily extended to multidimensional systems and the increasing computational effort is kept within limits.

3.1 Approximation of the State Space

In the proposed implementation of the Bayesian state observer, the \( n \)-dimensional state space \( D \) is approximated on a finite set of grid nodes, where the number of elements \( N_{G,i} \) for each dimension \( i \in [1, \ldots, n] \) is \( N_G \), serves as hyperparameter. The simple objective of this approach is to construct a grid on \( D \) in which the pdf of the state can be linear interpolated piecewise. The approximated one-dimensional state space for the \( i \)-th element of the \( n \)-dimensional state \( x_k = [x_{1,k}, \ldots, x_{n,k}]^T \) in time \( k \) is denoted by \( S_{i,k} \) and corresponds to a regular ordered set with \( N_{G,i} \) elements of real-valued grid positions between a defined upper and lower bound. The grid points in the set \( S_{i,k} = \{ x_{1,k}^i, \ldots, x_{N_{G,i}}^i \} \) do not need to be uniformly spaced and can be rasterised in each time step. Rasterization is necessary when the main feature of the pdf drifts closer to an edge of the set, the density already extends across the set, or the resolution of the pdf needs to be increased in a certain area. However, it must be ensured that the probability densities remain normalised by the approximation and the final grid at each time step. Nevertheless, the problem of grid design for point-mass approaches is addressed in Simandl et al. (2002), where also an algorithm for determining the minimum sufficient number of grid points is developed. The approximated state space \( D_k \) at time \( k \) is composed of the finite sets \( S_{i,k} \) as

\[
\prod_{i=1}^n S_{i,k} = \{ (x_{1,k}, \ldots, x_{n,k}) \} | x_{i,k} \in S_{i,k} \} \subset D.
\]

A state in the approximated state space \( D_k \) can be interpreted as an \( n \)-dimensional array that is uniquely indicated by an \( n \)-dimensional tuple \( \alpha \in A = \{ (\alpha_1, \ldots, \alpha_n) \} \) with \( 1 \leq \alpha_i \leq N_{G,i} \) with

\[
x_{\alpha}^k = (x_{\alpha_1}^{1,k}, \ldots, x_{\alpha_n}^{n,k}) \in \prod_{i=1}^n S_{i,k}.
\]
A finite pdf approximation of, for instance, \( p(x_k|Y_k) \) is symbolized with the tilde accent by \( \tilde{p}(x_k|Y_k) \). The density \( \tilde{p}(x_k|Y_k) \) is completely characterized by the set of values \( m^\alpha_{jk} = \tilde{p}(x'_k|Y_k) = p(x'_k|Y_k) \) for all \( \alpha \in \mathcal{A} \). On the domain \( x_k \in \mathcal{D} \setminus \mathcal{D}_k \), the density \( \tilde{p}(x_k|Y_k) \) is evaluated as the \( n \)-linear interpolation of the coefficients \( m^\alpha_{jk} \) with \( \alpha \in \partial \mathcal{A}(x_k) \) corresponding to the immediate grid node neighbours of \( x_k \), see, e.g., Press et al. (2007). The estimated densities of the a posteriori and a priori state are approximated in this way. The interpolation of the process noise \( p(w_k) \) and the measurement noise \( p(v_k) \) is not necessary, as they are assumed to be analytically given. The grid of the approximated state space \( \mathcal{D}_k \) consists of \( N_c = \prod_{i=1}^c N_{c,i} \) cells, where the center of the \( i \)-th cell is denoted as \( x'_k \). Therefore, the integration of a pdf can be numerically approximated by the trapezoidal rule, which is equal to the sum of the volume \( \Delta x'_k \) of the \( i \)-th cell times the interpolated center density \( \tilde{p}(x_k|Y_k) \) at \( x'_k \) over all \( N_c \) cells, \[
\int_{\mathcal{D}} p(x_k|Y_k)dx_k \approx \sum_{i=1}^{N_c} \tilde{p}(x'_k|Y_k)\Delta x'_k. \tag{10}
\]

3.2 Approximation of the Propagation Stage

At the beginning of the propagation state at time \( k \), the a posteriori density \( p(x_{k-1}|Y_{k-1}) \) is given. The purpose of this step is to propagate the density with the dynamics of the considered model to predict the a priori pdf \( p(x_k|Y_{k-1}) \). A direct approximation of (6) by computing the probability density and numerical integration on the constructed grid, as proposed in Section 3.1, involves an enormous computational effort. Therefore, the idea is to represent the a posteriori density by a large number of samples that are individually propagated forward through the state equation and from which the a priori density can be estimated. The proposed algorithms are called Markov chain Monte Carlo and Kernel Density Estimation.

3.3 Markov chain Monte Carlo Propagation

The objective is to generate two sequences of random vectors distributed according to the pdfs of the a posteriori state \( \tilde{p}(x_{k-1}|Y_{k-1}) \) and the process noise \( p(w_{k-1}) \). These samples, i.e., \( x_{k-1}^{(i)} \) and \( w_{k-1}^{(i)} \), are propagated pairwise by the state equation (3a) to obtain a third sequence \( x_k^{(i)} \) representing the a priori state of the current time step. The density of the predicted a priori state \( \tilde{p}(x_0|Y_{k-1}) \) is reconstructed from this sequence by kernel density estimation, see Section 3.4. For the purpose of generating a sampled sequence of in general \( n \)-dimensional random variables of a multivariate probability density function Markov chain Monte Carlo methods are considered. Starting from an arbitrary value \( x_{k-1}^{(0)} \), the elements of a Markov chain \( \{X_k^{(t)}\} = \{x_{k-1}^{(0)}, x_{k-1}^{(1)}, x_{k-1}^{(2)}, \ldots\} \) describe a stochastic process, where the probability of the element \( x_{k-1}^{(t)} \) solely depends on the state attained in the previous step \( x_{k-1}^{(t-1)} \), i.e., \( \Pr(x_{k-1}^{(t)} = x_{k-1}^{(t-1)}|X_{k-1}^{(t-1)}) = \Pr(x_{k-1}^{(t)} = x_{k-1}^{(t-1)}|x_{k-1}^{(t-1)}) \). Therefore, a Markov process is defined by a transition kernel \( Q(x_{k-1}^{(t)}|x_{k-1}^{(t-1)}) \) that proposes the next sample \( x_{k-1}^{(t+1)} \). The distribution of the Markov chain converges to a stationary density \( \pi(x_{k-1}) \) when the existence and the uniqueness of stationary distribution is given, see Robert and Casella (2005). These two conditions are sufficiently satisfied if the transition is detailed balanced and the Markov process is ergodic. Algorithms that construct this type of ergodic Markov chain with stationary distribution \( \pi(x_{k-1}) \) are called Markov chain Monte Carlo methods. This includes the well-known Metropolis-Hastings algorithm, which is often used for the sampling of high-dimensional distributions. Detailed balanced, i.e., the reversibility of the transition \( Q(x_{k-1}^{(t)}|x_{k-1}^{(t-1)}) \pi(x_{k-1}^{(t-1)}) = Q(x_{k-1}^{(t-1)}|x_{k-1}^{(t)}) \pi(x_{k-1}^{(t)}) \) is ensured by seperating the transition kernel \( Q(x_{k-1}^{(t)}|x_{k-1}^{(t-1)}) = q(x_{k-1}^{(t)}|x_{k-1}^{(t-1)}) \pi(x_{k-1}^{(t-1)}) \) in a proposal distribution \( q(x_{k-1}^{(t)}|x_{k-1}^{(t-1)}) \) and an acceptance distribution \( A(x_{k-1}^{(t-1)}, x_{k-1}^{(t)}) \) with Metropolis choice
\[
A(x_{k-1}^{(t-1)}, x_{k-1}^{(t)}) = \min \left( 1, \frac{\pi(x_{k-1}^{(t)}) q(x_{k-1}^{(t)}|x_{k-1}^{(t-1)})}{\pi(x_{k-1}^{(t-1)}) q(x_{k-1}^{(t-1)}|x_{k-1}^{(t)})} \right). \tag{11}
\]
The proposal distribution \( q(x_{k-1}^{(t)}|x_{k-1}^{(t-1)}) \), like the transition kernel \( Q(x_{k-1}^{(t-1)}|x_{k-1}^{(t)}) \), proposes a next sample \( x_{k-1}^{(t)} \), which, depending on the acceptance probability \( A(x_{k-1}^{(t-1)}, x_{k-1}^{(t)}) \) will be accepted or rejected. The proposal density can be chosen arbitrarily. Assuming that the density is symmetric and therefore the second fraction in (11) is omitted, the algorithm is also simply called Metropolis. A sketch of the proposed Metropolis-Hastings method for the sampling from \( \tilde{p}(x_{k-1}|Y_{k-1}) \) is given in Algorithm 1.

Algorithm 1 Metropolis-Hastings
1: Set initial state \( x_{0}^{(0)} \)
2: Set index variable \( i = 0 \)
3: while \( i < N_c \) do
4: Generate random variable:
5: \( x_{k-1}^{(i)} \leftarrow q(x_{k-1}^{(i)}|x_{k-1}^{(i-1)}) \)
6: Evaluate acceptance probability:
7: \( A(x_{k-1}^{(i-1)}, x_{k-1}^{(i)}) = \min \left( 1, \frac{\tilde{p}(x_{k-1}^{(i)}|Y_{k-1}) q(x_{k-1}^{(i)}|x_{k-1}^{(i-1)})}{\tilde{p}(x_{k-1}^{(i-1)}|Y_{k-1}) q(x_{k-1}^{(i-1)}|x_{k-1}^{(i)})} \right) \)
8: Generate uniform random variable:
9: \( u \leftarrow U[0,1] \)
10: Accept or reject:
11: if \( u \leq A(x_{k-1}^{(i-1)}, x_{k-1}^{(i)}) \) then
12: \( x_{k-1}^{(i+1)} \leftarrow x_{k-1}^{(i)} \)
13: else
14: \( x_{k-1}^{(i+1)} \leftarrow x_{k-1}^{(i)} \)
15: Increment index variable:
16: \( i \leftarrow i + 1 \)

3.4 Kernel Density Estimation

After the propagation of the random samples drawn from the a posteriori distribution, the probability density function of the propagated state can be obtained by kernel density estimation. This non-parametric method takes the previous obtained finite data set of random sample vectors for a approximation of the a posteriori density \( p(x_k|Y_k) \). The kernel density estimator of \( p(x_k|Y_k) \) is given as
from a computational point of view, and note that in the specific grid nodes aims to distribute the workload across multiple threads. Since there are no dependencies between the coefficients interpolated density ensures the normalization property. The denominator \( p \) factor that does not affect the proportion of the distribution

\[
m \in \mathbb{R}^{n \times n}
\]

where \( H \in \mathbb{R}^{n \times n} \) corresponds to the positive definite and symmetric bandwidth matrix and \( K(\cdot) \) to the multivariate kernel function. In common, the Epanechnikov and normal functions are used as multivariate kernel densities. The bandwidth matrix \( H \) has a huge impact on the accuracy of the estimator, as it is the most important hyperparameter for density smoothing. In the multidimensional scenario, both the width and the orientation are adjustable. The bandwidth has to be adapted to the considered state space. However, there are optimal as well as rule-of-thumb criteria for the choice of \( H \), see, e.g., Wand and Jones (1995). A sketch of the proposed kernel density estimator of \( \tilde{p}(x_{k-1}|Y_{k-1}) \) in a multi-threaded environment is given in Algorithm 2. Note, however, that there are a number of machine learning packages that contain very efficient software implementations of the kernel density estimator, such as mlpack (C++ / Python) and sklearn (Python).

Algorithm 2 Kernel Density Estimation

1: Set number of threads \( N_{\text{threads}} \)
2: Split \( A \) in \( N_{\text{threads}} \) subsets \( A_1, \ldots, A_{N_{\text{threads}}} \)
3: for all \( A_i \in A_1, \ldots, A_{N_{\text{threads}}} \) do
4: 
5: for all \( \alpha \in A_i \) do
6: Evaluate coefficients (9):
7: \[
m_{a|k-1} = \sum_{i=1}^{N_s} K \left( H^{-1/2} \left( x_k - x_i \right) \right)
\]

3.5 Approximation of the Inference Stage

At the stage of inference at time \( k \), the a priori density \( \tilde{p}(x_k|Y_{k-1}) \) is given as the current best estimate and a newly obtained measurement \( y_k \) is available. These data are used in order to determine the a posteriori probability density \( \tilde{p}(x_k|Y_k) \). Since in the approximated state space, the pdfs are described by the coefficients \( m_{a|k} \) and \( m_{a|k-1} \), only their transition is considered. Since the coefficients correspond more or less only to the evaluation of the densities at the grid nodes, the relationship is immediately given with (7) as

\[
m_{a|k} = \frac{p(y_k|x_k^a)}{p(y_k|Y_{k-1})} m_{a|k-1}
\]

where \( m_{a|k} = p(y_k|x_k^a|Y_k) \) and \( m_{a|k-1} = p(y_k|x_k^a|Y_{k-1}) \). The denominator \( p(y_k|Y_{k-1}) \) corresponds to a normalization factor that does not affect the proportion of the distribution and the MAP estimation \( \hat{x}_k \). Therefore, the normalization of the individual coefficients is omitted for the time being, since a subsequent scaling with the integral of the interpolated density ensures the normalization property. Since there are no dependencies between the coefficients \( m_{a|k} \) and \( m_{b|k} \) for all \( \alpha, \beta \in A \) and \( \alpha \neq \beta \), the set \( A \) can be split into several subset, which allows the coefficients to be calculated in parallel. The divide-and-conquer strategy aims to distribute the workload across multiple threads on a multi-core system to enable fast parallel execution. The numerator can be computed by evaluating (8) at the specific grid nodes \( x_i^k \). The calculation is not very complex from a computational point of view, and note that in the typical scenario of additive noise, even the Jacobian matrix becomes the identity matrix. A sketch of the proposed procedure is given in Algorithm 3.

Algorithm 3 Inference

1: Set number of threads \( N_{\text{threads}} \)
2: Split \( A \) in \( N_{\text{threads}} \) subsets \( A_1, \ldots, A_{N_{\text{threads}}} \)
3: for all \( A_i \in A_1, \ldots, A_{N_{\text{threads}}} \) do
4: 
5: for all \( \alpha \in A_i \) do
6: 
7: Evaluate numerator of (8):
8: Evaluate coefficients (9):
9: \[
m_{a|k} = \psi_k m_{a|k-1}
\]
10: Approximate density integral, see (10):
11: \[
\hat{x}_k = \sum_{i=1}^{N_s} \tilde{p}(x_i^k|Y_k) \Delta x_i^k
\]
12: Normalize density:
13: for all \( \alpha \in A \) do
14: \[
m_{a|k} = \frac{1}{V_k} m_{a|k-1}
\]

4. EXPERIMENTAL RESULTS

The previously presented algorithm is demonstrated on an autonomous scalar time-varying system with nonlinear state and measurement equation. The proposed test scenario is a benchmark in the nonlinear estimation literature, since the nonlinearity together with the time-dependent nature of the system turned out to be hardly capable by standard state observer. The considered system with state \( x_k \in \mathbb{R} \) and measured output \( y_k \in \mathbb{R} \) is given by

\[
x_{k+1} = \frac{1}{2} x_k^2 + 25 v_k 
\]

\[
y_k = \frac{1}{20} x_k^2 + v_k
\]

where the additive process noise \( v_k \) and measurement noise \( w_k \) are modeled as zero-mean normal distributed random variables with variances equal to one, i.e., \( v_k, w_k \sim N(0, 1) \). The EKF and the PF are adopted from Simon (2006), whereas the Bayesian state observer corresponds to the previously presented grid-based implementation. Fig. 1 demonstrates the estimation performance of the three proposed observer, where the PF is initialized with 500 particles and the BSO is implemented on a set of 500 grid points uniformly spaced between the boundaries -40 and 40. As Fig. 1 shows, the EKF deviates far from the correct state in parts of the observation interval. What is remarkable, however, is that the correct state is sometimes even outside the observer’s 95 percent confidence interval (red area). Although the confidence interval (orange area) of the PF outweighs that of the EKF in some places, the PF shows a significantly better performance. The estimates were only exceeded by the BSO, which is also clearly reflected in the following figure. Note, that the estimates depend on randomly generated samples. Therefore, between two series of experiments with different random seeds, the result may vary slightly. The linearization error \( e_k = x_k - \hat{x}_k \) for the three proposed observer is illustrated in Fig. 2. The observer performance is assessed by the mean squared error \( \text{MSE} = \frac{1}{T} \sum_{k=1}^{T} (x_k - \hat{x}_k)^2 \) and CPU computing time for a estimation step. An overview of the performance with different numbers of cells, i.e.
particles and grid points, is given in Tab. 1, where the minimum and average performance values of one hundred simulations were recorded. The BSO outperformed the PF in both computation time and observer performance above a certain number of cells. Note that the PF is the standard implementation as proposed in Simon (2006) and there are also extensions to improve the estimation performance. However, the hyperparameter of the BSO, i.e., grid width, grid limit as well as the number of grid points, were not optimized either. A significant increase in performance could still be achieved by adaptive regridding.

**Table 1. Observer performance evaluated on a standard workstation (6 x Intel Core i5-85000 CPU @ 3.00 GHz and 8 GB RAM)**

| Filter | Cells | Time (ms) | MSE       |
|--------|-------|-----------|-----------|
|        |       | avg.      | min.      | avg.      | min.      |
| EKF    | -     | 0.04      | 0.03      | 51.93     | 20.77     |
| PF     | 100   | 11.16     | 10.09     | 25.16     | 8.16      |
|        | 200   | 22.64     | 21.24     | 19.32     | 7.10      |
|        | 500   | 75.46     | 73.46     | 14.92     | 5.78      |
| BSO    | 100   | 5.53      | 5.23      | 30.35     | 10.35     |
|        | 200   | 8.83      | 7.44      | 13.02     | 6.02      |
|        | 500   | 15.48     | 13.83     | 7.99      | 0.45      |

5. CONCLUSION

In this contribution, a piecewise linear approximation of the Bayesian state estimation problem for nonlinear systems has been presented. The approach proposes a Markov chain Monte Carlo propagation stage, where the Metropolis-Hastings algorithm is used for the sampling from the a posteriori density. The a priori density is reconstructed by means of a kernel density estimation of the propagated samples. The performance of the Bayesian state observer is verified on a benchmark example. The decisive advantage of this variant is its applicability for systems of higher order, which will be presented in a further paper.

REFERENCES

Andrieu, C., Doucet, A., and Punskaya, E. (2001). *Sequential Monte Carlo Methods for Optimal Filtering*, 79–95. Springer New York.

Chen, Z. (2003). Bayesian Filtering: From Kalman Filters to Particle Filters, and Beyond. *Statistics*, 182.

Gordon, N., Salmond, D., and Smith, A. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings F (Radar and Signal Processing)*, 140, 107–113(6).

Ho, Y. and Lee, R. (1964). A Bayesian approach to problems in stochastic estimation and control. *IEEE Transactions on Automatic Control*, 9, 382–387.

Kramer, S.C. and Sorenson, H.W. (1988). Recursive Bayesian estimation using piece-wise constant approximations. *Automatica*, 24(6), 789–801.

Papoulis, A. and Pillai, S.U. (2002). *Probability, Random Variables, and Stochastic Processes*. McGraw Hill.

Press, W.H., Teukolsky, S.A., Vetterling, W.T., and Flannery, B.P. (2007). *Numerical recipes : the art of scientific computing*. Cambridge Univ. Pr.

Ristic, B., Arulampalam, S., and Gordon, N. (2003). *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech House.

Robert, C.P. and Casella, G. (2005). *Monte Carlo Statistical Methods* (*Springer Texts in Statistics*). Springer-Verlag, Berlin, Heidelberg.

Simandl, M., Kralovec, J., and Soderstrom, T. (2002). Anticipative grid design in point-mass approach to nonlinear state estimation. *IEEE Transactions on Automatic Control*, 47(4), 699–702.

Simon, D. (2006). *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. Wiley-Interscience.

Stano, P., Lendek, Z., Braaksma, J., and den Dekker, A.J. (2013). Parametric Bayesian Filters for Nonlinear Stochastic Dynamical Systems: A Survey. *IEEE Transactions on Cybernetics*, 43(6), 1607–1624.

Wand, M. and Jones, M.C. (1995). *Kernel Smoothing*. Chapman & Hall/CRC Monographs on Statistics & Applied Probability. Chapman & Hall.