Sequential Bayesian Inference for Uncertain Nonlinear Dynamic Systems: A Tutorial

Journal Article

Author(s):
Tatsis, Konstantinos E.; Dertimanis, Vasilis K.; Chatzi, Eleni (iD)

Journal of
Structural
Dynamics

Sequential Bayesian Inference for Uncertain Nonlinear Dynamic Systems: A Tutorial

Konstantinos E. Tatsis *[1], Vasilis K. Dertimanis †[1], and Eleni N. Chatzi ‡[1]

[1] *Institute of Structural Engineering, Department of Civil, Environmental and Geomatic Engineering, ETH Zürich, Stefano-Franscini-Platz 5, 8093, Zürich, Switzerland*

Abstract

In this article, an overview of Bayesian methods for sequential simulation from posterior distributions of nonlinear and non-Gaussian dynamic systems is presented. The focus is mainly laid on sequential Monte Carlo methods, which are based on particle representations of probability densities and can be seamlessly generalized to any state-space representation. Within this context, a unified framework of the various Particle Filter (PF) alternatives is presented for the solution of state, state-parameter and input-state-parameter estimation problems on the basis of sparse measurements. The algorithmic steps of each filter are thoroughly presented and a simple illustrative example is utilized for the inference of i) unobserved states, ii) unknown system parameters and iii) unmeasured driving inputs.

Keywords: Bayesian filtering; nonlinear non-Gaussian estimation; sequential Monte Carlo; importance sampling; Rao-Blackwellised Particle filtering

1 Introduction

The presence of diverse energy dissipation mechanisms in structural systems is directly connected to nonlinear phenomena, which are typically triggered upon exceedance of the material capacity or small displacement regime. Therefore, nonlinear behavior is oftentimes observed in dynamic systems, which are subjected to significantly high levels of excitation, such as earthquakes and wind fields. As a result, the nonlinearities are associated, for the vast majority of structural and mechanical systems, with the restoring force components, in which case the governing equations of motion are classified as nonlinear.

In general, however, the treatment of dynamics forms a challenging task, even when linear, particularly under the constraint of real-time performance, which is typically the case in Structural Health Monitoring (SHM) and control applications. Within this context, numerous time-domain methodologies have been proposed for system identification including the more simplified constructs that find application to linear systems, such as autoregressive moving average models (ARMA), the least squares estimation (LSE) method [1, 2], the Eigensystem Realization Algorithm (ERA) [3], as well as a large class of subspace identification techniques [4, 5, 6]. A particular class of identification schemes, relies on adoption of an observer setup, often in the form of Bayesian filtering methods, such as the Observer/Kalman filter Identification (OKID) [7, 8], which aim at identifying the system dynamics by recursively, or even in batch mode [9], assimilating data into the underlying model structure of the system at hand. These filtering formulations are particularly fitting within the context of virtual sensing, i.e., the task of inferring response quantities at unmeasured locations [10, 11], or even unknown system properties. Virtual sensing is essential for tasks such as digital twinning, diagnostics of condition in critical yet unreachable locations and control.

The estimation task increases in complexity for the case of nonlinear systems. A nonlinear formulation can arise even for the case of linear dynamics, when a joint state and parameter estimation task is pursued, as will be

---

*[*]tatsis@ibk.baug.ethz.ch
[†]v.derti@ibk.baug.ethz.ch
[‡]chatzi@ibk.baug.ethz.ch

elaborated in this work. Perhaps the simplest approach to tackle nonlinearities consists in successive linearizations of the system equations, leading to the so-called extended Kalman filter [12, 13]. A number of contributions based on these filters has been established for estimating not only the state and parameters (e.g. stiffness, damping or hysteretic properties) of a dynamic system, but also the unknown inputs [14, 15, 16]. Alternatively, particle-based methods [17, 18], with the Unscented Kalman filter (UKF) constituting an efficient and widely used option [19, 20] among them, can operate directly on the nonlinear equations of motion and propose a sample- or particle-based representation of the sequence of probability densities [21, 22]. These approaches are also known as sequential Monte Carlo samplers [23, 24], such as the standard Particle Filter (PF), which have been successfully applied to both structural and complex mechanical systems [25, 26]. One of the major drawbacks related to these techniques is the so-called sample impoverishment, which has been addressed in the literature by different means, such as evolutionary algorithms [27, 28, 29], extensions to more well-conditioned formulations [30] and marginalized samplers [31, 32].

The implementation of these filters is based on the approximation of the posterior state distribution by generating a number of samples, the so called particles, using Monte Carlo methods [33, 34]. Particle filters constitute essentially the extension of point-mass filters with the difference that the particles are concentrated in regions of high probability, instead of being uniformly distributed. These filters though, do not only suffer from the sample impoverishment problem, which pertains to the loss of particle diversity in time, but are also greedy in terms of particles. This implies that their performance might quickly become computationally inefficient due to the large pool of particles required for the distribution approximations. This issue can become even more pronounced in the case of joint state and parameter estimation problems [29], where the state vector is typically augmented in order to include the sought-after parameters.

At this point, it is worth mentioning a further identification task that is common within the SHM context, namely estimation under unknown input. Several Bayesian approaches have been proposed for jointly estimating the unmeasured input and the partially observed state of both linear [35, 36] and nonlinear systems [37, 38]. In the case of linear systems, an unbiased minimum-variance recursive filter without direct transmission has been initially proposed in [39] and later reformulated in [40]. To address the lack of optimality of these estimators in terms of the mean squared error, a new more effective filter has been developed in [41, 42] for joint input and state estimation. The numerical instabilities occurring therein, which are mainly observed when the number of measurements is smaller than the order of the system, have been also recently addressed in [43]. The state augmentation technique, which is mainly adopted for the joint identification of state and parameters, has been also employed in structural systems for state and input estimation [44, 45]. In a similar context, a dual Kalman filter (DKF) was derived in [46, 47], which seems to resolve the numerical issues related to the AKF. The required invertibility and observability conditions for all these joint estimators have been explored and reported in [48]. The same authors have also recently proposed the use of a smoothing algorithm [49], which aims at reducing the input-state estimation uncertainty by introducing a small time delay in the output.

A known issue associated with use of the above-mentioned Bayesian-type filters lies in the tuning of the process and measurement noise covariance matrices [50], which can severely affect performance when parameter estimation is pursued. This primarily affects estimation regimes, where the parameters are simultaneously estimated with the state, within the filtering setting, as opposed to schemes that employ a separate step for parameter estimation (e.g. Expectation-Maximization [51]). In joint state-parameter estimation, the quality of inference is strongly related to the initial parameter values and cross-covariance terms, which are responsible for translating the information of observed states into corrections of the hidden parameters. It has been demonstrated that a constant value for the coupling terms of the process noise covariance might prove insufficient for these type of problems [50], since a flow-dependent tuning [52] is required in order to avoid instabilities [53, 54]. Therefore, due to the fact that the actual error statistics of these predictors are unknown, the robust and reliable estimation might prove to be a significant challenge. For this purpose, a number of contributions has been published on the adaptation of the disturbance terms [55, 56], which can be classified into Bayesian, maximum-likelihood and covariance matching approaches [57, 58].

On the other hand, the main advantage of all aforementioned Bayesian filters lies in the recursive implementation, which allows for online and real-time performance. To accomplish the latter, the system dynamics ought to be typically represented in a reduced-order space, in order to deliver computationally efficient evaluations. Such models have been successfully used with Kalman-type filters [43, 48] however, the need for computationally efficient models becomes even more critical and also more challenging in the case of state-parameter and input-state parameter problems, for which reduced-order representations and system parametrizations have been proposed only for simple educational examples, such as beam models [14], spring-mass-dampers [59] and shear frames [15]. In this sense, the real-time implementation of such schemes with applications to more realistic and challenging systems calls for more sophisticated reduced-order representations [26, 60, 61], which can account for more complex system dynamics and parameter dependencies.

The structure of this paper, whose aim is to provide an overview of particle-based Bayesian filters for nonlinear non-Gaussian systems, is organized as follows: the problem of Bayesian filtering for stochastic state estimation in nonlinear system is presented in Section 2 in terms of the Chapman-Kolmogorov and Bayes equations. In Section 3, the problem is further generalized to joint state and parameter estimation problems and the sampling-based algorithms for optimal Bayesian filtering, namely the Unscented Kalman Filter (UKF), the Particle Filter (PF), the Particle Filter with Mutation (MPF), the Sigma-Point Particle Filter (SPPF), the Gaussian-Mixture Sigma-Point Particle Filter and the Rao-Blackwelised Particle Filter (RBPF) are presented. The problem is further extended to input-state and parameter estimation problems in Section 4, where the recently propose Dual Kalman Filter - Unscented Kalman Filter is summarized. Lastly, the performance of the algorithms is presented in Section 5 by means of an academic example, which we deem necessary for comprehensive illustration of state, state-parameter and input-state-parameter estimation cases.

2 Problem Formulation

Let us represent generic dynamic systems via use of the following general and nonlinear process equation in the continuous-time domain

$$\dot{\mathbf{x}}(t) = \mathbf{F}(\mathbf{x}(t), \mathbf{p}(t)) + \mathbf{v}(t) \tag{1}$$

where $\mathbf{x}(t) \in \mathbb{R}^n$ is the state vector, $\mathbf{p}(t) \in \mathbb{R}^{n_p}$ is the input force vector, $\mathbf{F} : \mathbb{R}^{n+n_p} \to \mathbb{R}^n$ denotes the vector-valued function that describes the system dynamics, and which in the structural context may encompass geometric (due to large deformations or instabilities) or material (e.g. due to plasticity) nonlinearities, and $\mathbf{v}(t) \in \mathbb{R}^n$ is the zero mean process noise vector with covariance matrix $\mathbf{Q}(t) \in \mathbb{R}^{n \times n}$. The corresponding observation equation, which is herein formed directly in the discrete-time domain reads

$$\mathbf{y}_k = \mathbf{H}(\mathbf{x}_k, \mathbf{p}_k) + \mathbf{w}_k \tag{2}$$

where $\mathbf{y}_k \in \mathbb{R}^{n_y}$ is the observation vector at $t = k\Delta t$, $\mathbf{H} : \mathbb{R}^n \to \mathbb{R}^{n_y}$ denotes the observation function, $\mathbf{x}_k \in \mathbb{R}^n$ and $\mathbf{p}_k \in \mathbb{R}^{n_p}$ are the discrete-time equivalents of the state $\mathbf{x}(t)$ and input $\mathbf{p}(t)$ respectively at $t = k\Delta t$, and $\mathbf{w}_k \in \mathbb{R}^{n_y}$ is the zero-mean observation noise vector, whose covariance matrix is denoted by $\mathbf{R}_k \in \mathbb{R}^{n_y \times n_y}$. Eq. (1) can further be expressed in the discrete-time domain, thus yielding the following nonlinear discrete-time state-space model

$$\mathbf{x}_{k+1} = \mathbf{G}(\mathbf{x}_k, \mathbf{p}_k) + \mathbf{v}_k \tag{3}$$

$$\mathbf{y}_k = \mathbf{H}(\mathbf{x}_k, \mathbf{p}_k) + \mathbf{w}_k \tag{4}$$

where $\mathbf{v}(t)$ and $\mathbf{Q}(t)$ are also mapped to the discrete-time counterparts $\mathbf{v}_k$ and $\mathbf{Q}_k$, respectively. In the SHM context, the observations pertain to typically measurable quantities, including accelerations, velocities, tilts or strains. The covariance of the process, $\mathbf{Q}_k$, and measurement noise, $\mathbf{R}_k$, are tunable filter parameters, which reflect the confidence that may be attributed to the fidelity of the model and the precision of acquired measurements, respectively. The relative selection of these parameters affects both the convergence and persistent error bounds of the estimation process, as elaborated in [62] for the case of the linear KF. Similarly, the system dynamics function $\mathbf{G} : \mathbb{R}^{n+n_p} \to \mathbb{R}^n$ is obtained upon integration of $\mathbf{F}$, as follows

$$\mathbf{G}(\mathbf{x}_k, \mathbf{p}_k) = \mathbf{x}_k + \int_{k\Delta t}^{(k+1)\Delta t} \mathbf{F}(\mathbf{x}(t), \mathbf{p}(t)) \, dt \tag{5}$$

From a Bayesian point of view, the problem of determining posterior estimates of the state $\mathbf{x}_{k|k}$, i.e., estimates at time k given the sequence of all available measurements $\mathbf{Y}_k = [\mathbf{y}_0 \ \mathbf{y}_1 \ \dots \ \mathbf{y}_k]$ up to time k, may be tailored to a recursive scheme, whereby the prior $p(\mathbf{x}_k|\mathbf{Y}_{k-1})$ and posterior $p(\mathbf{x}_k|\mathbf{Y}_k)$ distributions are obtained sequentially at each time instant k. Assuming the prior distribution $p(\mathbf{x}_0)$ is known and the required PDF $p(\mathbf{x}_{k-1}|\mathbf{Y}_{k-1})$ at time $k-1$ is also available, the prior probability $p(\mathbf{x}_k|\mathbf{Y}_{k-1})$ can be obtained through the prediction step, which is essentially derived from the Chapman-Kolmogorov equation

$$p(\mathbf{x}_k|\mathbf{Y}_{k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1}|\mathbf{Y}_{k-1}) \, d\mathbf{x}_{k-1} \tag{6}$$

Konstantinos E. Tatsis et. al

The probabilistic model of the state evolution $p(\mathbf{x}_k|\mathbf{x}_{k-1})$, which is also referred to as transitional density, is defined by the process equation Eq. (3), namely the system dynamics function $\mathbf{G}(\mathbf{x}_k)$ and the distribution $p(v_k)$ of the process noise. Subsequently, the prior may be conditioned on the measurement $\mathbf{y}_k$ at time k, using Bayes Theorem as follows

$$p(\mathbf{x}_k|\mathbf{y}_k) = p(\mathbf{x}_k|\mathbf{y}_k, \mathbf{Y}_{k-1}) = \frac{p(\mathbf{y}_k|\mathbf{x}_k)\,p(\mathbf{x}_k|\mathbf{Y}_{k-1})}{p(\mathbf{y}_k|\mathbf{Y}_{k-1})} \tag{7}$$

where the denominator $p(\mathbf{y}_k|\mathbf{Y}_{k-1})$, which acts as a normalizing factor, depends on the likelihood function $p(\mathbf{y}_k|\mathbf{x}_k)$, which is linked to the measurement process given by Eq. (4). The latter is in turn specified by the observation function $\mathbf{H}(\mathbf{x}_k)$ and the distribution of the measurement noise $\mathbf{w}_k$.

The relevance of Bayesian inference in structural dynamics is primarily applicable to model-based Structural Health Monitoring (SHM) applications. Within such a context, the aim is to infer the dynamic state of structural systems by fusing a numerical system representation, which specifies the state process as per Eq. (3), with sparse sensory information, whose type and location on the physical system define the measurement process expressed by Eq. (4). This inference step is oftentimes required to perform in real time, so as to allow for online assessment of the system dynamics, thus calling for continuous and sequential conditioning of the model prediction on the vibration-based measurement data.

The recursive prediction and correction formulas postulated by Eqs. (6) and (7) form the basis of the optimal Bayesian solution. Once the posterior PDF is computed, the optimal state estimate can be obtained by means of different statistical metrics. In engineering applications, this is typically the conditional mean of $\mathbf{g}: \mathbb{R}^n \to \mathbb{R}^m$, which can be any arbitrary function with respect to $\mathbf{x}_k$, and represents the minimum square error (MMSE) estimate

$$\mathbb{E}\left[\mathbf{g}(\mathbf{x}_k)\big|\mathbf{Y}_k\right] = \int \mathbf{g}(\mathbf{x}_k)\,p\left(\mathbf{x}_k\big|\mathbf{Y}_k\right)d\mathbf{x}_k \tag{8}$$

The function $\mathbf{g}$ is a user-defined parameter, linked to the requirements of the problem at hand, where often the quantities of interest that are to be inferred are specified as functions of the state vector $\mathbf{x}_k$. In control applications, the target engineering quantity is typically the state itself, either in terms of displacements or in terms of velocities. This implies that $\mathbf{g}(\mathbf{x}_k) = \mathbf{x}_k$ and as such the expectation of Eq. (8) is essentially the first moment of the posterior distribution $p\left(\mathbf{x}_k\big|\mathbf{Y}_k\right)$. The same applies to those applications which aim at inferring the entire response field using a limited number of observations. In SHM applications, where the aim might be for instance the inference of fatigue damage accumulation at unmeasured system locations [63], the quantity of interest is the stress field and as such, the function $\mathbf{g}$ is responsible for transforming the state variable $\mathbf{x}_k$ to stress values at fatigue-critical locations. It should be noted that the quantities of interest may be alternatively inferred by means of the maximum *a posteriori* (MAP) estimate, which corresponds to the value of $\mathbf{g}(\mathbf{x}_k)$ that maximizes $p(\mathbf{x}_k|\mathbf{Y}_k)$ or even higher order statistics.

The prediction and filtering steps described by Eqs. (6) and (7) are amenable to an analytical solution only in the particular case of linear system equations and additive Gaussian noise. In the case of nonlinear system dynamics, which fall within the scope of this paper, the solution is approximated using the sampling algorithms described in the following sections. It should be mentioned for the sake of generality that the Extended Kalman Filter is another alternative for dealing with nonlinear Gaussian problems, requiring though successive linearizations at each time step, around the posterior estimates. This might prove to be computationally inefficient for real-time or near real-time performance and as such, the focus of this paper is laid on the sample-based approximations of the filtering equations.

3   State and parameter estimation

As a first instance of nonlinear problems, we here present the class of state-parameter estimation problems. This is a typical pursuit in the context of SHM, when the monitored system features a model of known structure, albeit of uncertain or unknown parameters (e.g. stiffness, damping, or hysteretic parameters). The original Bayesian problem is formulated in Section 2 and essentially assumes that the state-space model is *a priori* fully specified. However, as aforementioned, certain model parameters, which will be henceforth denoted by $\boldsymbol{\theta} \in \mathbb{R}^{n_\theta}$, might be unknown as well. As such, their estimation within the sequential Bayesian context can be performed by treating them as random variables, for which a prior distribution $p(\boldsymbol{\theta})$ need be specified. The initial problem is therefore modified as follows

$$\boldsymbol{\theta}_k \sim \mathcal{N}\left(\boldsymbol{\theta}_k; \boldsymbol{\theta}_{k-1}, \boldsymbol{\Sigma}_{k-1}^{\theta}\right) \tag{9}$$

$$\mathbf{x}_{k+1} = \mathbf{G}(\mathbf{x}_k, \mathbf{p}_k, \boldsymbol{\theta}_k) + \mathbf{v}_k \tag{10}$$

$$\mathbf{y}_k = \mathbf{H}(\mathbf{x}_k, \mathbf{p}_k, \boldsymbol{\theta}_k) + \mathbf{w}_k \tag{11}$$

Konstantinos E. Tatsis et. al

where the transition model of the parameters is assumed to be a Gaussian random walk, derived from the density $\mathcal{N}(\mathbf{x}; \mathbf{m}, \mathbf{C})$ with argument $\mathbf{x} \in \mathbb{R}^n$, mean $\mathbf{m} \in \mathbb{R}^n$ and covariance $\mathbf{C} \in \mathbb{R}^{n \times n}$. A straightforward way of integrating the parameter estimation problem into the Bayesian framework is to use the so-called state augmentation approach, whereby the parameter vector is appended to the system state. This implies that the state vector $\mathbf{x}_k$ is redefined as $\tilde{\mathbf{x}}_k = \mathrm{vec}([\mathbf{x}_k, \boldsymbol{\theta}_k])$ and the state-space model is now written in the augmented form

$$\tilde{\mathbf{x}}_{k+1} = \tilde{\mathbf{G}}(\tilde{\mathbf{x}}_k, \mathbf{p}_k) + \tilde{\mathbf{v}}_k \tag{12}$$

$$\mathbf{y}_k = \tilde{\mathbf{H}}(\tilde{\mathbf{x}}_k, \mathbf{p}_k) + \mathbf{w}_k \tag{13}$$

where the initial system and observation functions $\mathbf{G}(\bullet)$ and $\mathbf{H}(\bullet)$ are accordingly transformed to their augmented counterparts $\tilde{\mathbf{G}}(\bullet)$ and $\tilde{\mathbf{H}}(\bullet)$ and respectively. It should be noted that the sampling algorithms presented in the following sections may refer either to the initial problem described by Eqs. (3) and (4), or to the augmented one represented by Eqs. (12) and (13). For the sake of simplicity though, all the steps will be formulated using the notation of the initial problem, as described by Eqs. (3) and (4).

3.1 The Unscented Kalman Filter (UKF)

The Unscented Kalman Filter (UKF) is a filtering algorithm, which draws its formulation from the Unscented Transform (UT) [64, 65] and boils down to the use of a structured set of points (sigma points) in order to approximate the mean and covariance of the target distribution. This is also the main difference between the UKF and the different variants of Particle Filters (PFs), which will be elaborated in the following sections; the former is based on deterministic sample points while the latter relies on Monte Carlo (MC) sampling for the evaluation of the expectation integrals. In the context of the UKF, it is postulated that the target distribution is represented by a Gaussian density, whose hyperparameters are estimated using a number of deterministically selected sample points, also referred to as the sigma points. These are propagated through the nonlinear state-space equations, enabling thus reconstruction of the prior $p(\mathbf{x}_k|\mathbf{Y}_{k-1})$ and posterior $p(\mathbf{x}_k|\mathbf{Y}_k)$ densities in terms of the sample-based mean and covariance. The UKF is related to the Bayesian formulation presented in the previous section through the following recursive formulas

$$p(\mathbf{x}_{k-1}|\mathbf{Y}_{k-1}) = \mathcal{N}(\mathbf{x}_{k-1}; \hat{\mathbf{x}}_{k-1|k-1}, \mathbf{P}_{k-1|k-1}) \tag{14}$$

$$p(\mathbf{x}_k|\mathbf{Y}_{k-1}) = \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1}) \tag{15}$$

$$p(\mathbf{x}_k|\mathbf{Y}_k) = \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k}, \mathbf{P}_{k|k}) \tag{16}$$

in which $\hat{\mathbf{x}}_{k-1|k-1}$ and $\mathbf{P}_{k-1|k-1}$ denote the state and error covariance estimates at step $k-1$ given measurements up to and including step $k-1$. It should be underlined that the approximation postulated by Eqs. (14) to (16) is essentially identical to the one characterizing a general Gaussian filter. However, the inference integrals of a general Gaussian filter can be calculated in a closed form only for specific model structures. Instead, they are practically calculated using numerical approximations, like the one proposed by the UKF. Similar approximations can be obtained using the Gauss-Hermite Kalman Filter (GHKF) or the Cubature Kalman Filter (CKF), which are based on Gauss-Hermite quadrature and spherical cubature rules respectively for the numerical integration of the inference equations.

Assuming that the posterior state estimate $\hat{\mathbf{x}}_{k-1|k-1}$ at step $k-1$ and the corresponding covariance $\mathbf{P}_{k-1|k-1}$ are available, the *a priori* state statistics at step k can be calculated by propagating the sigma points $\mathcal{X}_{k-1|k-1}^j \in \mathbb{R}^n$ for $j = 0, \ldots, 2n$, through the state equation. These points are deterministically placed with respect to the posterior density hyperparameters as follows

$$\mathcal{X}_{k-1|k-1}^j = \begin{cases} \hat{\mathbf{x}}_{k-1|k-1}, & j = 0 \\ \hat{\mathbf{x}}_{k-1|k-1} + \sqrt{n+\lambda}\left[\sqrt{\mathbf{P}_{k-1|k-1}}\right]_j, & j = 1, \ldots, n \\ \hat{\mathbf{x}}_{k-1|k-1} - \sqrt{n+\lambda}\left[\sqrt{\mathbf{P}_{k-1|k-1}}\right]_{j-n}, & j = n+1, \ldots, 2n \end{cases} \tag{17}$$

where n is the state size and $\lambda = \alpha^2(n+\kappa) - n$ is a scaling parameter which depends on the constants α and κ. The former determines the spread of sigma points around the mean value and typically ranges between 10^{-4} and 1, while the latter acts as a secondary scaling parameter which is usually equal to $3 - n$. The reader is referred to [66] for further details in this respect. The term $\left[\sqrt{\mathbf{P}_{k-1|k-1}}\right]_j$ denotes the jth column of the covariance matrix square root.

Each one of the posterior sigma points at step $k-1$ is propagated through the state equation, thus yielding a prior estimate of the sigma points $\mathcal{X}_{k|k-1}^j$ for $j = 0, \ldots, 2n$ at step k. These points represent the predicted density $p(\mathbf{x}_k|\mathbf{Y}_{k-1})$,

Konstantinos E. Tatsis et. al

Algorithm 1: Unscented Kalman Filter (UKF)

---

1   Specify the parameters α, β and κ

2   Initialize the search distribution parameters $\hat{\mathbf{x}}_{0|0}$ and $\mathbf{P}_{0|0}$

3   **for** $k = 1, 2, \ldots$ **do**

4   $\quad$ Calculate sigma points $\mathcal{X}^j_{k-1|k-1}$ using Eq. (17)

5   $\quad$ (a) Time update step

6   $\quad$ Calculate prior sigma points $\mathcal{X}^j_{k|k-1}$ by propagating $\mathcal{X}^j_{k-1|k-1}$ through Eq. (3)

7   $\quad$ Obtain the prior state estimates $\hat{\mathbf{x}}_{k|k-1}$ and $\mathbf{P}_{k|k-1}$ using Eqs. (18) and (19)

8   $\quad$ (b) Measurement update step

9   $\quad$ Propagate the sigma points $\mathcal{X}^j_{k|k-1}$ through Eq. (4) to obtain the predicted output $\hat{\mathbf{y}}_{k|k-1}$

10 $\quad$ Calculate the predicted output covariance $\mathbf{P}^{yy}_k$ and Kalman gain using Eqs. (24) to (26)

11 $\quad$ Obtain the posterior state estimates $\hat{\mathbf{x}}_{k|k}$ and $\mathbf{P}_{k|k}$ using Eqs. (22) and (23)

12 **end**

---

whose mean and covariance are approximated using a weighted sample mean and covariance

$$\hat{\mathbf{x}}_{k|k-1} = \sum_{j=0}^{2n} W^{\mathrm{m}}_j \mathcal{X}^j_{k|k-1} \tag{18}$$

$$\mathbf{P}_{k|k-1} = \sum_{j=0}^{2n} W^{\mathrm{c}}_j \left[\mathcal{X}^j_{k|k-1} - \hat{\mathbf{x}}_{k|k-1}\right]\left[\mathcal{X}^j_{k|k-1} - \hat{\mathbf{x}}_{k|k-1}\right]^{\mathrm{T}} + \mathbf{Q}_{k-1} \tag{19}$$

The mean and covariance weights, denoted by W^{m}_j and W^{c}_j respectively, are constant and given by the following formulas

$$W^{\mathrm{m}}_0 = \frac{\lambda}{\lambda + n}, \quad W^{\mathrm{c}}_0 = \frac{\lambda}{\lambda + n} + 1 - \alpha^2 + \beta, \quad W^{\mathrm{m}}_j = W^{\mathrm{c}}_j = \frac{1}{2(n + \lambda)}, \quad \text{for} \quad j = 1, \ldots, 2n \tag{20}$$

where β is a scalar parameter, which is related to the state distribution. For Gaussian distributions β is set to 2, while the reader is referred to [66] for further details. The predicted measurement is similarly obtained as the weighted average of the measurement equation evaluated for each one of the sigma points

$$\hat{\mathbf{y}}_{k|k-1} = \sum_{j=0}^{2n} W^{\mathrm{m}}_j \mathbf{H}\left(\mathcal{X}^j_{k|k-1}, \mathbf{p}_k\right) \tag{21}$$

Thereafter, the measurement update equations are as follows:

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\left(\mathbf{y}_k - \hat{\mathbf{y}}_{k|k-1}\right) \tag{22}$$

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k\,\mathbf{P}^{yy}_k\,\mathbf{K}^T_k \tag{23}$$

where

$$\mathbf{K}_k = \mathbf{P}^{xy}_k\left(\mathbf{P}^{yy}_k - \mathbf{R}_k\right)^{-1} \tag{24}$$

$$\mathbf{P}^{yy}_k = \sum_{j=0}^{2n} W^{\mathrm{c}}_j \left[\mathbf{H}\left(\mathcal{X}^j_{k|k-1}, \mathbf{p}_k\right) - \hat{\mathbf{y}}_{k|k-1}\right]\left[\mathbf{H}\left(\mathcal{X}^j_{k|k-1}, \mathbf{p}_k\right) - \hat{\mathbf{y}}_{k|k-1}\right]^{\mathrm{T}} + \mathbf{R}_k \tag{25}$$

$$\mathbf{P}^{xy}_k = \sum_{j=0}^{2n} W^{\mathrm{c}}_j \left[\mathcal{X}^j_{k|k-1} - \hat{\mathbf{x}}_{k|k-1}\right]\left[\mathbf{H}\left(\mathcal{X}^j_{k|k-1}, \mathbf{p}_k\right) - \hat{\mathbf{y}}_{k|k-1}\right]^{\mathrm{T}} \tag{26}$$

and $\mathbf{K}_k$ is the Kalman gain at step k. It should be noted that in the UKF implementation presented in this paper the effect of noise is simply additive, which is oftentimes encountered in engineering problems and provides a significant

reduction of the computational complexity. In the case of non-additive noise, which implies that $\mathbf{G}(\bullet)$ and $\mathbf{H}(\bullet)$ are functions of the process and measurement noise terms respectively, the state vector need be augmented with the noise variables $\mathbf{v}_k$ and $\mathbf{w}_k$ and the sigma points need be calculated using the augmented state vector.

3.2 The Particle Filter (PF)

In problems where Gaussian approximations are not suited for representing the target distribution, which is for instance the case when the latter is a multi-modal distribution, the filtering equations are solved with sequential Monte Carlo approximations, which are encapsulated in Particle Filters (PFs) [67]. Similar to the UKF, PFs approximate the target PDF $p(\mathbf{x}_k|\mathbf{Y}_k)$ using a set of sample points $\mathbf{x}_k^j$ for $j = 1,\ldots,N$ with associated weights w_k^j. The major difference consists in the fact that a PF is based on samples drawn from a Monte Carlo scheme, while the UKF makes use of deterministically placed samples. However, due to practical difficulties in sampling directly from $p(\mathbf{x}_k|\mathbf{Y}_k)$, the expectation of Eq. (8) when using a PF is approximated by drawing samples from an importance distribution, which is herein denoted by $q(\mathbf{x}_k|\mathbf{Y}_k)$, per the so called importance sampling (IS) strategy. Within this context, the expectation over the probability density function $p(\mathbf{x}_k|\mathbf{Y}_k)$ can be decomposed as follows

$$\mathbb{E}\left[\mathbf{g}(\mathbf{x}_k)\,\middle|\,\mathbf{Y}_k\right] = \int \mathbf{g}(\mathbf{x}_k)\,p\left(\mathbf{x}_k\middle|\mathbf{Y}_k\right)d\mathbf{x}_k = \int \left[\mathbf{g}(\mathbf{x}_k)\frac{p(\mathbf{x}_k|\mathbf{Y}_k)}{q(\mathbf{x}_k|\mathbf{Y}_k)}\right]q(\mathbf{x}_k|\mathbf{Y}_k)\,d\mathbf{x}_k \tag{27}$$

on the condition that the importance distribution $q(\mathbf{x}_k|\mathbf{Y}_k)$ receives non-zero values at the non-zero points of $p(\mathbf{x}_k|\mathbf{Y}_k)$. As such, Eq. (27) can be seen as the expectation of the terms contained within the brackets over the importance distribution, which can be sampled in order to obtain a Monte Carlo estimate of the integral.

Within this context, a Monte Carlo approximation of the expectation described by Eq. (27) can be obtained from the following weighted sum

$$\mathbb{E}\left[\mathbf{g}(\mathbf{x}_k)\,\middle|\,\mathbf{Y}_k\right] \approx \frac{1}{N}\sum_{j=1}^{N}\frac{p\left(\mathbf{x}_k^j\middle|\mathbf{Y}_k\right)}{q\left(\mathbf{x}_k^j\middle|\mathbf{Y}_k\right)}\mathbf{g}\left(\mathbf{x}_k^j\right) = \sum_{j=1}^{N}w_k^j\,\mathbf{g}\left(\mathbf{x}_k^j\right) \tag{28}$$

where w_k^j denotes the j-th sample weight. A similar expression can be derived for the approximation of the probability density $p(\mathbf{x}_k|\mathbf{Y}_k)$, in which case $\mathbf{g}(\mathbf{x}_k^j)$ would be substituted by $\delta(\mathbf{x}_k - \mathbf{x}_k^j)$. The importance weights can be generated by a recursive formula, which is derived [33, 34] using the Markov properties of the importance distribution and the system model described by Eq. (3) and postulates the following proportionality

$$w_k^j \propto w_{k-1}^j \frac{p\left(\mathbf{y}_k\middle|\mathbf{x}_k^j\right)p\left(\mathbf{x}_k^j\middle|\mathbf{x}_{k-1}^j\right)}{q\left(\mathbf{x}_k^j\middle|\mathbf{x}_{k-1}^j,\mathbf{Y}_k\right)} \tag{29}$$

where $p\left(\mathbf{x}_k^j\middle|\mathbf{x}_{k-1}^j\right)$ is the transitional density, defined by Eq. (3), and $p\left(\mathbf{y}_k\middle|\mathbf{x}_k^j\right)$ is the likelihood function, which is fully specified by Eq. (4).

As evidenced from the presented formulation, the performance of a PF is strongly dependent on the quality of the proposal distribution $q(\mathbf{x}_k|\mathbf{Y}_k)$, which should further allow to easily draw samples therefrom. It has been proved [23] that the optimal importance distribution, in terms of variance, is $p(\mathbf{x}_k|\mathbf{x}_{k-1},\mathbf{y}_k)$. However, a convenient selection that is oftentimes used is the transitional density $p\left(\mathbf{x}_k\middle|\mathbf{x}_{k-1}^j\right)$, which results to the so called bootstrap particle filter. This is
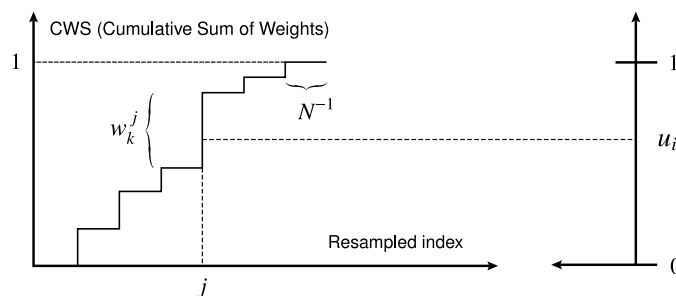

Fig. 1: Schematic representation of the re-sampling step; the uniformly distributed random variable u_i is mapped to index j and the corresponding particle $\mathbf{x}_k^j$ is likely to be selected due to its considerable weight w_k^j
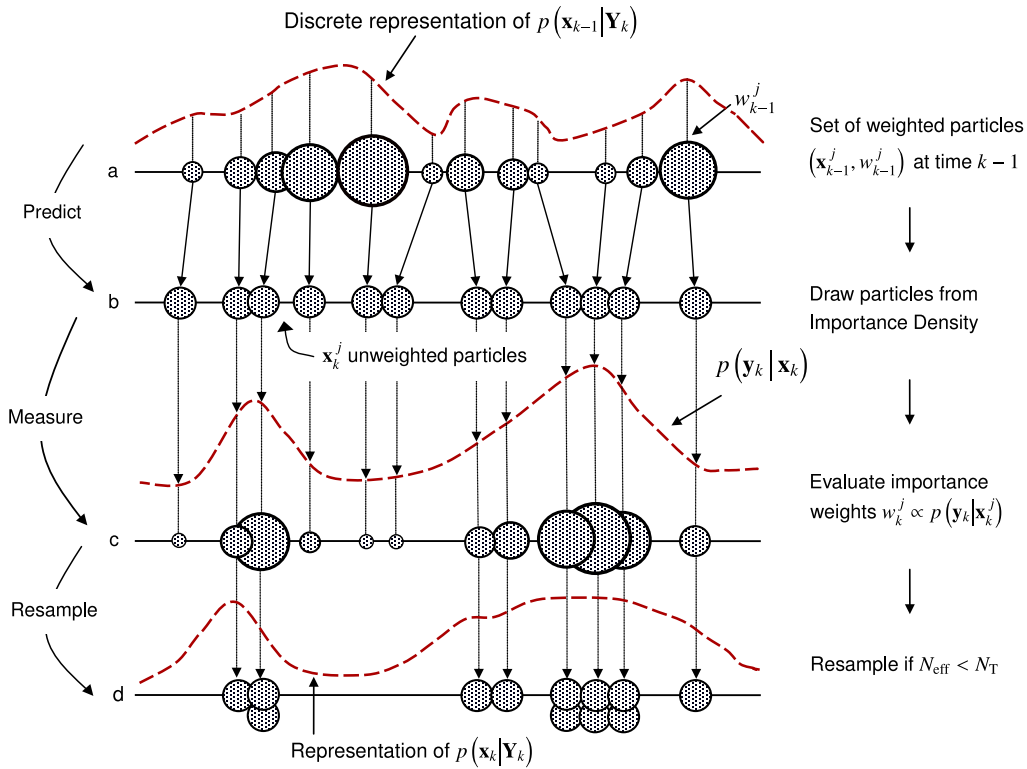
Fig. 2: Schematic representation of the standard Particle Filter (PF) algorithm: a) Predict by drawing samples from the importance density; b) Evaluate the importance weights based on the likelihood function; c) Resample if the effective number of particles is below N_{T} and normalize weights; d) Approximate the posterior density through the set of weighted particles.

not necessarily a very informative function and not always an optimal one since the space of $\mathbf{x}_k$ is explored without using the information contained in the measurements. On the other hand, the transitional density can be more easily sampled and upon substitution in Eq. (29) yields the following recursive formula for the calculation of weights

$$w_k^j = w_{k-1}^j \, p\left(\mathbf{y}_k \big| \mathbf{x}_k^j\right) \tag{30}$$

which implies that the selection of weights w_k^j at time step k depends on the likelihood $p\left(\mathbf{y}_k \big| \mathbf{x}_k^j\right)$, which is fully specified by the measurement process. Since the weights calculated by sampling a density function, be it the one of Eq. (30) or in the more general case the one of Eq. (29), represent the likelihood of each particle, they should be normalized so that the sum of all likelihoods is equal to one.

The convergence of all Monte Carlo methods, including PFs, is guaranteed by the central limit theorem as the number of particles approaches infinity however, an increased number of particles might lead to a significant computational cost, which can be a major disadvantage of the method. On the other hand, although the UKF is characterized by more limited applicability, since it contains the assumption of Gaussian approximations, it is in general considerably faster than the PFs, which is owed to the limited and predefined number of sample points.

3.2.1 Resampling and Sample Impoverishment

A standard issue associated with PFs is the problem of degeneracy, which refers to the tendency of a single particle to dominate the weights as the evaluation steps progress. This is a known pathology, which is owed to the inherently increasing variance of the importance weights over time [23]. However, it implies that the bulk of computational effort is wasted for the propagation of non-contributing particles. The effective sample size [68] is a representative metric of the degeneracy level, which can be estimated as

$$N_{\mathrm{eff}} = \frac{1}{\sum\limits_{j=1}^{N} \left(w_k^j\right)^2} \tag{31}$$

Algorithm 2: Standard Particle Filter (PF)

---

1   Initialize by drawing samples $\mathbf{x}_0^j \sim p(\mathbf{x}_0)$ for $j = 1, 2, \ldots, N$

2   **for** $k = 1, 2, \ldots$ **do**

3   (a) Importance sampling

4   **for** $j = 1, 2, \ldots, N$ **do**

5   Draw samples $\mathbf{x}_k^j \sim q\left(\mathbf{x}_k \middle| \mathbf{x}_{k-1}^j, \mathbf{Y}_k\right)$

6   Evaluate the importance weights $w_k^j = w_{k-1}^j \, p\left(\mathbf{y}_k \middle| \mathbf{x}_k^j\right)$

7   **end**

8   For $j = 1, 2, \ldots, N$ normalize weights $w_k^j \leftarrow w_k^j \left[\sum_{j=1}^{N} w_k^j\right]^{-1}$

9   (b) Resampling

10   Multiply or supress particles $\mathbf{x}_k^j$ to obtain samples distributed according to $p\left(\mathbf{x}_k \middle| \mathbf{Y}_k\right)$

11   For $j = 1, 2, \ldots, N$ reset weights $w_k^j = 1/N$

12   (c) Inference

13    $\mathbb{E}\left[\mathbf{g}(\mathbf{x}_k) \middle| \mathbf{Y}_k\right] \approx \sum_{j=1}^{N} w_k^j \mathbf{g}\left(\mathbf{x}_k^j\right)$

14 **end**

Sample degeneracy may be tackled either by increasing the number of employed particles or, more effectively, by resampling the particles upon exceedance of a certain degeneracy threshold N_T, which is a user-specified variable. The latter technique aims at discarding the particles with negligible weight and enhancing the ones with larger weights. As such, a new set of particles $\left\{\mathbf{x}_k^{j*}, \quad j = 1, .., N\right\}$ is generated, which occurs by replacement from the original set, so that $p\left(\mathbf{x}_k^{j*} = \mathbf{x}_k^j\right) = w_k^j$. The process is schematically shown in Fig. 1, where it should be further noted that all weights are reset to $w_k^j = 1/N$ upon resampling.

    Despite the efficient treatment of degeneracy, resampling may lead to the so-called sample impoverishment [67]. This issue may typically arise in problems with small process noise and subsequently result to large-weight particles being selected multiple times, destroying thus the diversity among particles. This problem has been addressed in
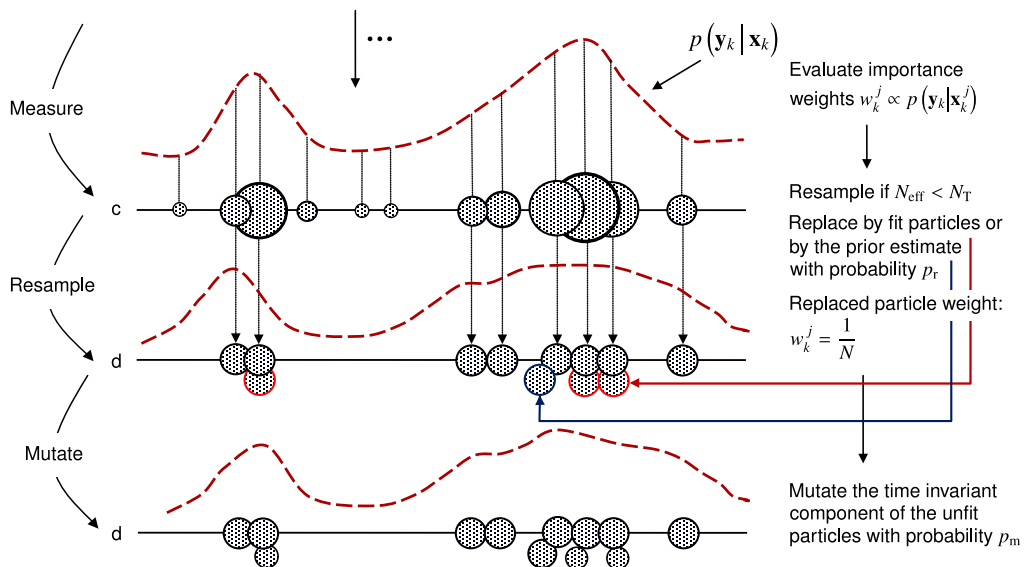

Fig. 3: Schematic representation of the Particle Filter with Mutation (MPF): b) Evaluate the importance weights based on the likelihood function; c) Resample if the effective number of particles is below N_T (replace by the fit particles or the prior estimate of the state) and normalize weights; d) Mutate the resampled particles; e) Approximate the posterior density through the set of weighted particles.

the literature using evolutionary operations [28] and Support Vector Regression (SVR) [69], as well as combination of PFs with the Kalman filter for the time update step and importance density generation [30]. The standard PF implementation, which is described in this section, is schematically presented in Fig. 2, while the detailed steps of the algorithm are listed in Alg. 2.

3.2.2   The Particle Filter with Mutation (MPF)

An improvement to the standard PF, in terms of the sample impoverishment problem, is presented in [29] where a mutation operation is introduced for the time invariant components of the state vector, which is used to modify the resampling step. In the standard PF implementation, the unfit particles are initially replaced by fitter ones that are already contained in the pool and then uniform weights are assigned to all particles. Such a strategy may lead to a fast convergence of the time invariant parameters to one of the initial particles, especially in the case of low process noise, as also underlined in [70]. In this sense, the search space of the state parameters in the standard PF is largely limited by the initially proposed density and particles. This implies that for a successful estimation, the particles should be drawn from the vicinity of the actual parameter values. It should be noted that the importance sampling scheme of the MPF is identical to the one used for the PF and the mutation operation is intended to alleviate the sample impoverishment problem by maintaining diversity in the particles, similarly to other resampling alternatives, such as the Markov Chain Monte Carlo (MCMC) scheme [71].

Algorithm 3: Particle Filter with Mutation (MPF)

1   Initialize by drawing samples $\mathbf{x}_0^j \sim p\left(\mathbf{x}_0\right)$ for $j = 1, 2, \ldots, N$

2   **for** $k = 1, 2, \ldots$ **do**

3   (a) Importance sampling

4   **for** $j = 1, 2, \ldots, N$ **do**

5   Draw samples $\mathbf{x}_k^j \sim q\left(\mathbf{x}_k | \mathbf{x}_{k-1}^j, \mathbf{Y}_k\right)$

6   Evaluate the importance weights $w_k^j = w_{k-1}^j \, p\left(\mathbf{y}_k | \mathbf{x}_k^j\right)$

7   **end**

8   For $j = 1, 2, \ldots, N$ normalize weights $w_k^j \leftarrow w_k^j \left[\sum_{i=1}^N w_k^i\right]^{-1}$

9   (b) Resampling

10   Multiply or supress particles $\mathbf{x}_k^j$ to obtain samples distributed according to $p\left(\mathbf{x}_k | \mathbf{Y}_k\right)$

11   **for** $j = 1, 2, \ldots, N_{\mathrm{u}}$ **do**

12   Replace particle $\mathbf{x}_k^j$ with probabilty p_{r} by the prior estimate $\mathbf{x}_{k|k-1} = \mathbf{G}\left(\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{p}_k\right)$

13   Mutate the time-invariant components of $\mathbf{x}_k^j$ with probability p_{m} using Eq. (32)

14   Adjust the weights of mutated particles using Eq. (33)

15   **end**

16   For $j = 1, 2, \ldots, N$ normalize weights $w_k^j \leftarrow w_k^j \left[\sum_{i=1}^N w_k^i\right]^{-1}$

17   (c) Inference

18     $\mathbb{E}\left[\mathbf{g}\left(\mathbf{x}_k\right) | \mathbf{Y}_k\right] \approx \sum_{j=1}^N w_k^j \mathbf{g}\left(\mathbf{x}_k^j\right)$

19   **end**

In the MPF algorithm, the N_{u} unfit particles are replaced, with a user-defined probability p_{r}, by the prior state estimate $\hat{\mathbf{x}}_{k|k-1} = \mathbf{G}\left(\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{p}_{k-1}\right)$ and the time invariant components of the state vector are mutated in order to allow the exploration of the parameter space. Concretely, the mutation is applied to each i-th parameter component of the j-th particle with a probability p_m, using the following perturbation rule

$$\mathbf{x}_k^{i,j} = \mathbf{x}_k^{i,j} \cdot \left(1 + d_i \cdot \left(m_i - 0.5\right)\right) \tag{32}$$

in which d_i denotes the perturbation radius and $m_i \sim \mathcal{U}\left(0, 1\right)$ is a random number drawn from the standard uniform distribution. This operation resembles the creep mutation of the traditional Genetic Algorithm (GA), which takes place in the phenotype, i.e. the real representation of the parameter strings [72]. Thereafter, the mutated particles are

Konstantinos E. Tatsis et. al

assigned a weight which is inversely proportional to the relative difference $\Delta \mathbf{x}_k^j$ between the mutated and the parent particle, calculated as follows

$$w_k^j = \frac{1}{N} \frac{1}{\frac{\left\| \Delta \mathbf{x}_k^j \right\|}{\left\| \mathbf{x}_k \right\|} + 1} \tag{33}$$

where $\| \bullet \|$ denotes the L_2 norm, although an alternate penalty weighting scheme of similar logic may be readily adopted. On the other hand, the weights of the non-mutated re-sampled particles remain unchanged, equal to $w_k^j = 1/N$ and the entire set of weights in subsequently normalized. Eq. (33) offers a heuristic approach to specifying a fast computed term that penalizes heavily mutated particles; this could alternatively be replaced by an another distance metric, which includes consideration of the likelihood. A graphical representation of the MPF is shown in Fig. 3, while the detailed steps of the algorithm are listed in Alg. 3.

Algorithm 4: Sigma-Point Particle Filter (SPPF)

---

1   Initialize by drawing samples $\mathbf{x}_0^j \sim p(\mathbf{x}_0)$ for $j = 1, 2, \dots, N$

2   **for** $k = 1, 2, \dots$ **do**

3   (a) Importance sampling

4   **for** $j = 1, 2, \dots, N$ **do**

5   Calculate sigma points $\mathcal{X}_{k-1|k-1}^j$ for particle $\mathbf{x}_{k-1}^j$ using Eq. (17)

6   Perform a time update step to obtain $\hat{\mathbf{x}}_{k|k-1}^j$, $\mathbf{P}_{k|k-1}^j$ using Eqs. (18) and (19)

7   Perform a measurement update step to obtain $\hat{\mathbf{x}}_{k|k}^j$, $\mathbf{P}_{k|k}^j$ using Eqs. (22) and (23)

8   Draw samples $\mathbf{x}_k^j \sim q\left(\mathbf{x}_k | \mathbf{x}_{k-1}^j, \mathbf{Y}_k\right) \approx \mathcal{N}\left(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k}, \mathbf{P}_{k|k}\right)$

9   Evaluate the importance weights $w_k^j = w_{k-1}^j \, p\left(\mathbf{y}_k | \mathbf{x}_k^j\right)$

10   **end**

11   For $j = 1, 2, \dots, N$ normalize weights $w_k^j \leftarrow w_k^j \left[\sum_{j=1}^N w_k^j\right]^{-1}$

12   (b) Resampling

13   Multiply or supress particles $\mathbf{x}_k^j$ to obtain samples distributed according to $p\left(\mathbf{x}_k | \mathbf{Y}_k\right)$

14   For $j = 1, 2, \dots, N$ reset weights $w_k^j = 1/N$

15   (c) Inference

16     $\mathbb{E}\left[\mathbf{g}(\mathbf{x}_k) | \mathbf{Y}_k\right] \approx \sum_{j=1}^N w_k^j \mathbf{g}\left(\mathbf{x}_k^j\right)$

17 **end**

3.3   Sigma-Point Particle Filter (SPPF)

An alternative improvement to the standard PF in terms of the proposal distribution is provided by the Sigma-Point Particle Filter (SPPF). This algorithm has been proposed in [30], on the basis of the work published in [73] on Gaussian filters, whereby the particles are moved towards the areas of high likelihood, as defined by the current observation $\mathbf{y}_k$. The idea of this algorithm, which is schematically presented in Fig. 4 and is intended to address the impoverishment issue, is materialized by assigning a Gaussian proposal distribution $\mathcal{N}\left(\mathbf{x}_k; \mathbf{x}_k^j, \mathbf{P}_k^j\right)$ to each particle $\mathbf{x}_k^j$ in order to calculate the mean and covariance of the importance distribution. These can be obtained using an EKF, or more efficiently a UKF, which avoids the linearization of system equations and additionally accounts for higher order statistics. Thereafter, each one of the Gaussian proposal distributions $\mathcal{N}\left(\mathbf{x}_k; \mathbf{x}_k^j, \mathbf{P}_k^j\right)$ for $j = 1, 2, \dots, N$ is used to draw the j-th particle at time k.

     The detailed steps of the SPPF algorithm are listed in Alg. 4. These are herein derived on the basis of the UKF however, they can be substituted by an EKF or even by a Square-Root Central Difference Kalman filter (SR-CDKF), as proposed in [30]. It should be also noted that despite the fact that the SPPF is based on the Gaussian assumption, it has been shown to provide a better approximation to the target distribution for a number of applications [74].
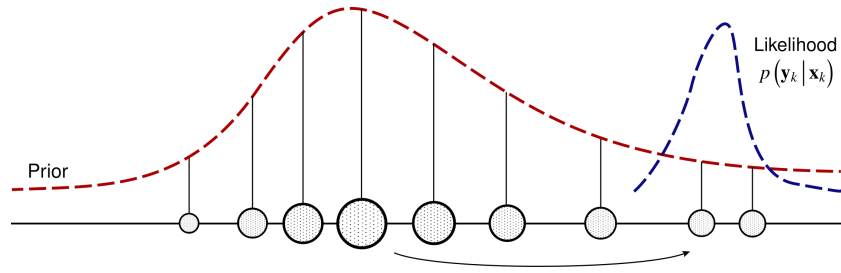
Fig. 4: Schematic representation of the particle movement towards the likelihood region

3.4 Gaussian Mixture Sigma-Point Particle Filter (GMSPPF)

In terms of performance, the SPPF has been shown to address the sample depletion problem and outperform the standard PF in terms of accuracy however, it is still characterized by a considerable computational cost. This is owed to the fact that each particle is associated with a UKF, or alternatively an EKF, which in turn requires a system linearization for each particle.

A further improvement to the shortcomings of the computational performance of the SPPF has been achieved by the Gaussian Mixture Sigma-Point Particle Filter (GMSPPF) , which has been proposed in [30] and is especially suited for simulation of problems with multi-modal probability density functions. This algorithm is based on i) an importance sampling step for the measurement update and ii) a UKF-based Gaussian sum filter, which is used for the time update

---

Algorithm 5: Gaussian Mixture Sigma-point Particle Filter (GMSPPF)

1   Specify the number of mixing components and initialize by drawing samples $\mathbf{x}_0^j \sim p(\mathbf{x}_0)$ for $j = 1, 2, \ldots, N$

2   **for** $k = 1, 2, \ldots$ **do**

3   (a) Time update

4   Set $p_q(\mathbf{v}_{k-1}) = \mathcal{N}\left(\mathbf{v}_{k-1}; \bar{\mathbf{v}}_{k-1}^q, \mathbf{Q}_{k-1}^q\right)$ for $q = 1, 2, \ldots, G_p$

5   Set $p_s\left(\mathbf{x}_{k-1}|\mathbf{Y}_{k-1}\right) = \mathcal{N}\left(\mathbf{x}_{k-1}; \hat{\mathbf{x}}_{k-1|k-1}^s, \mathbf{P}_{k-1|k-1}^s\right)$ for $s = 1, 2, \ldots, G_s$

6   Set $p(\mathbf{w}_k) = \mathcal{N}\left(\mathbf{w}_k; \bar{\mathbf{w}}_k^r, \mathbf{R}_k^r\right)$ for $r = 1, 2, \ldots, G_m$

7   **for** $s^- = (s, q)$ *with* $s \in \{1, 2, \ldots, G_s\}$ *and* $q \in \{1, 2, \ldots, G_p\}$ **do**

8   Calculate $p_{s^-}\left(\mathbf{x}_k|\mathbf{Y}_{k-1}\right) = \mathcal{N}\left(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k-1}^{s^-}, \mathbf{P}_{k|k-1}^{s^-}\right)$ and update the coefficients $\alpha_k^{s^-} = \alpha_{k-1}^s \beta_{k-1}^q \Big/ \left(\sum_{s=1}^{G_s} \sum_{q=1}^{G_p} \alpha_{k-1}^s \beta_{k-1}^q\right)$

9   **end**

10   **for** $s^+ = (s^-, r)$ *with* $s^- \in \{1, 2, \ldots, G_s^-\}$ *and* $r \in \{1, 2, \ldots, G_m\}$ **do**

11   Compute the likelihood $s_k^r = p_r\left(\mathbf{y}_k|\mathbf{x}_k = \hat{\mathbf{x}}_{k|k-1}^{s^-}\right)$

12   Calculate $p_{s^+}\left(\mathbf{x}_k|\mathbf{Y}_k\right) = \mathcal{N}\left(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k}^{s^+}, \mathbf{P}_{k|k}^{s^+}\right)$ and update $\alpha_k^{s^+} = \alpha_k^{s^-} \gamma_k^r s_k^r \Big/ \left(\sum_{s^-=1}^{G_s^-} \sum_{r=1}^{G_r} \alpha_k^{s^-} \gamma_k^r s_k^r\right)$

13   **end**

14   Calculate the prior state density $p\left(\mathbf{x}_k|\mathbf{Y}_{k-1}\right)$ using Eq. (35)

15   Calculate the posterior state density $p\left(\mathbf{x}_k|\mathbf{Y}_k\right)$ using Eq. (36)

16   (b) Measurement update

17   Draw particles $\mathbf{x}_k^j$ from the posterior density $p\left(\mathbf{x}_k|\mathbf{Y}_k\right)$ obtained in step 16

18   For $j = 1, 2, \ldots, N$ calculate the particle weights $w^j = p\left(\mathbf{y}_k|\mathbf{x}_k^j\right) p\left(\mathbf{x}_k^j|\mathbf{Y}_{k-1}\right) \Big/ p\left(\mathbf{x}_k^j|\mathbf{Y}_k\right)$

19   For $j = 1, 2, \ldots, N$ normalize weights $w_k^j \leftarrow w_k^j \left[\sum_{i=1}^{N} w_k^i\right]^{-1}$

20   Fit the G_s-component GMM of Eq. (34) to the particles $\mathbf{x}_k^j$ drawn in step 18

21   (c) Inference

22    $\mathbb{E}\left[\mathbf{g}(\mathbf{x}_k)|\mathbf{Y}_k\right] = \sum_{s=1}^{G_s} \alpha_k^s \mathbb{E}\left[\mathbf{g}\left(\mathbf{x}_k^s\right)|\mathbf{Y}_k\right]$

23 **end**

---

step and the generation of the proposal density. As such, a G_s-component Gaussian Mixture Model (GMM) is used at each time step for the approximation of the posterior state density $p\left(\mathbf{x}_{k-1}\middle|\mathbf{Y}_{k-1}\right)$, as follows

$$p\left(\mathbf{x}_{k-1}\middle|\mathbf{Y}_{k-1}\right) = \sum_{s=1}^{G_\text{s}} \alpha_{k-1}^s \mathcal{N}\left(\mathbf{x}_{k-1}; \hat{\mathbf{x}}_{k-1|k-1}^s, \mathbf{P}_{k-1|k-1}^s\right) \tag{34}$$

where α_{k-1}^s is the weight of the s-th component and $\mathcal{N}\left(\mathbf{x}; \mathbf{m}, \mathbf{P}\right)$ represents the normal distribution of the s-th component with argument $\mathbf{x}$, mean value $\mathbf{m}$ and covariance matrix $\mathbf{P}$. The density of Eq. (34) is obtained from the posterior particles extracted after the measurement update step. This calculation is performed with the use of the Expectation-Maximization (EM) algorithm and it further implies that not only the state densities but also the ones for the noise terms, $p(\mathbf{v}_{k-1})$ and $p(\mathbf{w}_k)$ are represented by GMMs, whose coefficients are β_{k-1}^q for $q = 1, 2, \ldots, G_\text{p}$ and γ_k^r for $r = 1, 2, \ldots, G_\text{m}$, respectively. As such, prior and posterior densities, which are obtained from the prediction and measurement steps, are also expressed by the following GMMs

$$p\left(\mathbf{x}_k\middle|\mathbf{Y}_{k-1}\right) = \sum_{s^-=1}^{G_\text{s}^-} \alpha_k^{s-} \mathcal{N}\left(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k-1}^{s-}, \mathbf{P}_{k|k-1}^{s-}\right) \tag{35}$$

$$p\left(\mathbf{x}_k\middle|\mathbf{Y}_k\right) = \sum_{s^+=1}^{G_\text{s}^+} \alpha_k^{s+} \mathcal{N}\left(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k}^{s-}, \mathbf{P}_{k|k}^{s-}\right) \tag{36}$$

where $G_\text{s}^- = G_\text{s} \cdot G_\text{p}$ and $G_\text{s}^+ = G_\text{s}^- \cdot G_\text{m}$. From the above equations it is suggested that the number of mixing components increases in each prediction and correction step. Namely, a G_s-component GMM is initially used for the target density, which increases to G_s^- in the prediction step and subsequently to G_s^+ in the correction step. This implies an exponential increase in the number of mixing terms, which can be avoided by fitting a G_s-component GMM to the N particles drawn from the G_s^+-component GMM of the posterior density, whose proposal is obtained from Eq. (36). Lastly, the inference step is carried out with the use of the fitted GMM, as documented in Alg. 5, which contains all the steps of the GMSPPF.

3.5 Rao-Blackwellised Particle Filter (RBPF)

In certain problems the model dynamics may contain a tractable subspace, which can be marginalised, as is the case for problems of joint state-parameter estimation, where the time invariant parameters evolve differently to the system's dynamical states. Marginalisation results in a more efficient PF implementation, the so called Rao-Blackwellised Particle Filter (RBPF) [75, 31], which allows the analytical computation of the filtering equations related to the marginalized state. To do so, the state vector is partitioned as follows

$$\mathbf{x}_k = \begin{bmatrix} \mathbf{x}_k^\text{a} \\ \mathbf{x}_k^\text{b} \end{bmatrix} \tag{37}$$

where $\mathbf{x}_k^\text{b}$ is the part of the state vector to be marginalized and $\mathbf{x}_k^\text{a}$ contains the remaining state vector. Then using Bayes rule, the posterior PDF can be rewritten as

$$p\left(\mathbf{x}_k\middle|\mathbf{Y}_k\right) = p\left(\mathbf{x}_k^\text{a}, \mathbf{x}_k^\text{b}\middle|\mathbf{Y}_k\right) = p\left(\mathbf{x}_k^\text{b}\middle|\mathbf{x}_k^\text{a}, \mathbf{Y}_k\right) p\left(\mathbf{x}_k^\text{a}\middle|\mathbf{Y}_k\right) \tag{38}$$

and the expectation over the density $p\left(\mathbf{x}_k\middle|\mathbf{Y}_k\right)$ can be also decomposed as follows

$$\mathbb{E}\left[\mathbf{g}\left(\mathbf{x}_k\right)\middle|\mathbf{Y}_k\right] = \int \mathbf{g}\left(\mathbf{x}_k\right) p\left(\mathbf{x}_k\middle|\mathbf{Y}_k\right) d\mathbf{x}_k \tag{39}$$

$$= \int \left[\int \mathbf{g}\left(\mathbf{x}_k^\text{a}, \mathbf{x}_k^\text{b}\right) p\left(\mathbf{x}_k^\text{b}\middle|\mathbf{x}_k^\text{a}, \mathbf{Y}_k\right) d\mathbf{x}_k^\text{b}\right] p\left(\mathbf{x}_k^\text{a}\middle|\mathbf{Y}_k\right) d\mathbf{x}_k^\text{a} \tag{40}$$

which implies that if the integral in the brackets can be evaluated analytically, the PF need be used only for the sample-based approximation of $p\left(\mathbf{x}_k^\text{a}\middle|\mathbf{Y}_k\right)$. Therefore, the above expectation can be rewritten as

$$\mathbb{E}\left[\mathbf{g}\left(\mathbf{x}_k\right)\middle|\mathbf{Y}_k\right] = \sum_{j=1}^{N} w_k^j \int \mathbf{g}\left(\mathbf{x}_k^\text{a}, \mathbf{x}_k^\text{b}\right) p\left(\mathbf{x}_k^\text{b}\middle|\mathbf{x}_k^\text{a}, \mathbf{Y}_k\right) d\mathbf{x}_k^\text{b} \tag{41}$$

where w_k^j for $j = 1, 2, \ldots, N$ are the importance weights associated with the particles of $\mathbf{x}_k^{\mathrm{a}}$. As such, the reduced in size $\mathbf{x}_k^{\mathrm{a}}$ will require less particles than the full size state $\mathbf{x}_k$, increasing thus the accuracy of estimates for a given amount of computational resources. In the context of filtering, such an analytical computation is carried using a Kalman filter, implying thus that $\mathbf{x}_k^{\mathrm{b}}$ is described by a linear transition model and Gaussian noise terms, which is typically the case in parameter estimation problems where a random walk model with Gaussian noise is assumed for the parameter evolution. Despite this assumption, the RBPF algorithm is herein presented for the general case, in which the transition model of $\mathbf{x}_k^{\mathrm{b}}$ may be described by any distribution $p\left(\mathbf{x}_k^{\mathrm{b}} \mid \mathbf{x}_k^{\mathrm{a}}, \mathbf{Y}_k\right)$.

Algorithm 6: Rao-Blackwellised Particle Filter (RBPF)

---

1   Initialize the state $\mathbf{x}_0^{\mathrm{a}}$ by drawing samples $\mathbf{x}_0^{\mathrm{a},j} \sim p\left(\mathbf{x}_0^{\mathrm{a}}\right)$ for $j = 1, 2, \ldots, N$

2   Initialize the j-th filter associated with $\mathbf{x}_0^{\mathrm{a},j}$ by specifying $\hat{\mathbf{x}}_{0|0}^{\mathrm{b},j}$ and $\mathbf{P}_{0|0}^{\mathrm{b},j}$ for $j = 1, 2, \ldots, N$

3   **for** $k = 1, 2, \ldots$ **do**

4   (a) Time update

5   **for** $j = 1, 2 \ldots, N$ **do**

6   Perform a time update step to obtain the prior state statistics $\hat{\mathbf{x}}_{k|k-1}^{\mathrm{b},j}$ and $\mathbf{P}_{k|k-1}^{\mathrm{b},j}$

7   Draw samples $\mathbf{x}_k^{\mathrm{a},j} \sim q\left(\mathbf{x}_k^{\mathrm{a}} \middle| \mathbf{x}_{k-1}^{\mathrm{a},j}, \mathbf{Y}_k\right)$

8   Calculate the particle weights using Eq. (42)

9   **end**

10   For $j = 1, 2, \ldots, N$ normalize weights $w_k^j \leftarrow w_k^j \left[\sum_{j=1}^N w_k^j\right]^{-1}$

11   (b) Measurement update

12   **for** $j = 1, 2, \ldots, N$ **do**

13   Perform a measurement update step to obtain the posterior state statistics $\hat{\mathbf{x}}_{k|k}^{\mathrm{b},j}$ and $\mathbf{P}_{k|k}^{\mathrm{b},j}$ conditional on $\mathbf{x}_k^{\mathrm{a},j}$

14   **end**

15   (c) Resampling

16   Multiply or supress particles $\mathbf{x}_k^{\mathrm{a},j}$ to obtain samples distributed according to $p\left(\mathbf{x}_k^{\mathrm{a}} \middle| \mathbf{Y}_k\right)$

17   For $j = 1, 2, \ldots N$ reset weights $w_k^j = 1/N$

18   (d) Inference

19     $\mathbb{E}\left[\mathbf{g}\left(\mathbf{x}_k\right) \middle| \mathbf{Y}_k\right] = \sum_{j=1}^{N} w_k^j \int \mathbf{g}\left(\mathbf{x}_k^{\mathrm{a}}, \mathbf{x}_k^{\mathrm{b}}\right) p\left(\mathbf{x}_k^{\mathrm{b}} \middle| \mathbf{x}_k^{\mathrm{a}}, \mathbf{Y}_k\right) d\mathbf{x}_k^{\mathrm{b}}$

20 **end**

Within this context, each particle $\mathbf{x}_k^{\mathrm{a},j}$ for $j = 1, 2, \ldots, N$ that is used for the approximation of $p\left(\mathbf{x}_k^{\mathrm{a}} \middle| \mathbf{Y}_k\right)$ is associated with a Bayesian filter for the integration of $p\left(\mathbf{x}_k^{\mathrm{b}} \middle| \mathbf{x}_k^{\mathrm{a}}, \mathbf{Y}_k\right)$, creating thus a hierarchical process in which the expectation over the density $p\left(\mathbf{x}_k \middle| \mathbf{Y}_k\right)$ is calculated in two steps, as postulated by Eq. (38). Thus, given the particles $\mathbf{x}_{k|k-1}^{\mathrm{a},j}$, which are drawn from $p\left(\mathbf{x}_k^{\mathrm{a}} \middle| \mathbf{Y}_k\right)$, and the associated prior estimates $\mathbf{x}_{k|k-1}^{\mathrm{b},j}$ along with the corresponding covariance matrices $\mathbf{P}_{k|k-1}^{\mathrm{b},j}$, which are extracted from the transition model of $\mathbf{x}_k^{\mathrm{b}}$, one can estimate the importance weights according to

$$w_k^j = w_{k-1}^j p\left(\mathbf{y}_k \middle| \mathbf{x}_{k|k-1}^{\mathrm{a},j}, \mathbf{Y}_{k-1}\right) \tag{42}$$

which are fully defined by the measurement equation. In this case, it is assumed that the importance density of $\mathbf{x}_k^{\mathrm{a}}$ is identical to the transitional density $p\left(\mathbf{x}_k^{\mathrm{a}} \middle| \mathbf{x}_{k-1}^{\mathrm{a}}\right)$, resulting thus to Eq. (42). In the more general case, in which a different importance density $q\left(\mathbf{x}_k^{\mathrm{a}} \middle| \mathbf{x}_{k-1}^{\mathrm{a},j}, \mathbf{Y}_k\right)$ is assumed, the weights can be calculated from Eq. (29).

It should be noted that when the dynamics of $\mathbf{x}_k^{\mathrm{b}}$ are governed by a linear Gaussian model, the probability distribution of Eq. (42) is a Gaussian one and equal to the marginal measurement likelihood of the corresponding Kalman filter. Thereafter, the prior estimate of $\mathbf{x}_k^{\mathrm{a}}$ can be obtained as the following weighted sample mean

$$\hat{\mathbf{x}}_{k|k-1}^{\mathrm{a}} = \sum_{j=1}^{N} w_k^j \mathbf{x}_{k|k-1}^{\mathrm{a},j} \tag{43}$$

which should be evaluated upon normalization of the importance weights obtained from Eq. (42). A measurement update step can be performed for each filter associated with $\mathbf{x}_{k|k-1}^{\text{a},j}$ for $j = 1, 2, \ldots, N$ in order to obtain the posterior state estimate $\mathbf{x}_{k|k}^{\text{b},j}$ conditional on $\mathbf{x}_{k|k-1}^{\text{a},j}$ and subsequently perform the inference step using Eq. (41). The detailed steps for the implementation of the Bayesian inference problem using the RBPF algorithm are listed in Alg. 6.

These steps are formulated for the general case where the prior and posterior distributions of the marginalized state are obtained by any filtering algorithm. In the most simple and computationally efficient variant of the RBPF, which corresponds to linear Gaussian equations of the marginalized state, these distributions are extracted by means of a Kalman filter. However, any of the herein presented algorithms can be used instead, when either of the marginalized state or measurement equations is nonlinear or characterized by non-Gaussian noise. In such cases, only a slight computational improvement is offered by the RBPF due to the separation of the sampling space however, the major advantage of the RBPF is still maintained and this pertains, according to the Rao-Blackwell theorem [76], to the reduced variance of the estimated quantities.

4 State, input and parameter estimation

The filtering algorithms presented in the previous section have been formulated for the state and unknown system parameters estimation. This was accomplished by treating the parameters as random variables, which are appended to the state vector. However, a further typical problem met within the context of SHM is the need for estimating the response, and possibly the unknown system parameters, under additional absence of information in the loads (input). Similarly to what is previously described, the additional estimation of the unknown excitation can be performed by assuming that each input signal is a random variable, whose dynamics are governed by Gaussian random walk. In this sense, the state-parameter estimation problem, described by Eqs. (12) and (13), can be further modified as follows

$$\mathbf{p}_k \sim \mathcal{N}\left(\mathbf{p}_k; \mathbf{p}_{k-1}, \boldsymbol{\Sigma}_{k-1}^{\text{p}}\right) \tag{44}$$

$$\tilde{\mathbf{x}}_{k+1} = \tilde{\mathbf{G}}\left(\tilde{\mathbf{x}}_k, \mathbf{p}_k\right) + \tilde{\mathbf{v}}_k \tag{45}$$

$$\mathbf{y}_k = \tilde{\mathbf{H}}\left(\tilde{\mathbf{x}}_k, \mathbf{p}_k\right) + \mathbf{w}_k \tag{46}$$

where $\boldsymbol{\Sigma}_{k-1}^{\text{p}}$ denotes the covariance matrix of the random walk model assumed for the input. The latter can be also rewritten in the following form

$$\mathbf{p}_{k+1} = \mathbf{p}_k + \mathbf{w}_k^{\text{p}} \tag{47}$$

where $\mathbf{w}_k^{\text{p}}$ is the equivalent Gaussian zero-mean input process noise with covariance matrix $\boldsymbol{\Sigma}_{k-1}^{\text{p}}$.

One possibility of integrating the input estimation problem in the Bayesian framework is to simply augment $\tilde{\mathbf{x}}_k$ with $\mathbf{p}_k$, which would result to twice-augmented state, containing both parameters and inputs. An alternative to this, which was initially proposed for input-state estimation in linear systems [46] and subsequently extended to input-state-parameter problems in [59], is the combination of Eq. (47) with the observation equation, namely Eq. (13), in order to form a second state-space model, in which $\mathbf{p}_k$ is considered the state and $\mathbf{x}_k$ is treated as a constant. This results to a dual filtering problem, which is recursively solved, first for the input, using a standard Kalman filter, and subsequently for the augmented state, using a particle-based filter, at each time step. In this sense, the augmented problem, which refers to the state and parameter estimation, can be practically solved by any of the algorithms documented in Section 3 however, for the sake of simplicity and computational efficiency, the dual filter will be presented in this paper in terms of the UKF. This results to the so called Dual Kalman Filter - Unscented Kalman Filter (DKF -UKF) [59], whose implementation is in detail documented in Alg. 7.

It should be underlined that the problem of concurrent input, state and parameter estimation has been only recently addressed by the research community using recursive Bayesian techniques however, a number of alternatives to the DKF-UKF is already available. A straightforward approach for dealing with such a problem consists in augmenting the state with both input and parameter vectors [14] and subsequently performing the sequential inference by means of a linearized or particle-based filter. When a short time delay is allowed in the estimated quantities, the inference may be carried out using the smoothing algorithm presented in [15], which is essentially an extension of the input-state estimator proposed in [42]. Lastly, the problem can be solved again in a fully Bayesian context using a Gaussian Process Latent Force Model (GPLFM) for the modeling of the forcing term dynamics, in conjunction with Markov Chain Monte Carlo (MCMC) for the inference of unknown system parameters [16].

Algorithm 7: Dual Kalman Filter - Unscented Kalman Filter (DKF-UKF)

---

1 Specify the UKF parameters α, β and κ

2 Initialize the state and input distribution parameters $\hat{\mathbf{x}}_{0|0}$, $\mathbf{P}_{0|0}$ and $\hat{\mathbf{p}}_{0|0}$ $\mathbf{P}^{\mathrm{p}}_{0|0}$

3 **for** $k = 1, 2, \ldots$ **do**

4 $\quad$ Calculate sigma points $\boldsymbol{\mathcal{X}}^{j}_{k-1|k-1}$ using Eq. (17)

5 $\quad$ (a) Time update

6 $\quad$ Perform a KF time update step to obtain the prior input estimates $\hat{\mathbf{p}}_{k|k-1}$ and $\mathbf{P}^{\mathrm{p}}_{k|k-1}$

7 $\quad$ Calculate the prior sigma points $\boldsymbol{\mathcal{X}}^{j}_{k|k-1}$ by propagating $\boldsymbol{\mathcal{X}}^{j}_{k-1|k-1}$ through Eq. (3)

8 $\quad$ Obtain the prior state statistics $\hat{\mathbf{x}}_{k|k-1}$ and $\mathbf{P}_{k|k-1}$ using Eqs. (18) and (19)

9 $\quad$ (b) Measurement update step

10 $\quad$ Perform a KF measurement update step to obtain the posterior input estimates $\hat{\mathbf{p}}_{k|k}$ and $\mathbf{P}^{\mathrm{p}}_{k|k}$

11 $\quad$ Propagate the sigma points $\boldsymbol{\mathcal{X}}^{j}_{k|k-1}$ through Eq. (4) to obtain the predicted output $\hat{\mathbf{y}}_{k|k-1}$

12 $\quad$ Calculate the predicted output covariance $\mathbf{P}^{yy}_{k}$ and Kalman gain using Eqs. (24) to (26)

13 $\quad$ Obtain the posterior state estimates $\hat{\mathbf{x}}_{k|k}$ and $\mathbf{P}_{k|k}$ using Eqs. (22) and (23)

14 **end**

The main difference among these alternatives is associated with the input estimation step, which is based on different assumptions and formulations for each filter. In the dual schemes, such as DKF-UKF, the transition model of the input dynamics is used to form a second state space, which results in two filtering steps; one for the state and one for the input. The main feature of such a formulation is the decoupling of the error covariance matrices related to the input and state processes, which in turn implies that input corrections are not dependent on the state updates. On the contrary, a state augmentation approach would integrate the input process to a single augmented state-space model, thus resulting to a coupled error covariance matrix and interdependence between input and state corrections. In both of these formulations, the input noise covariance matrices, which essentially regulate the range of allowable input fluctuations, need be specified. This step is substituted by the kernel related parameters when using the Gaussian Process Latent Force Model for the input dynamics, while it is completely bypassed when using the smoothing algorithm proposed in [15], which is characterized by a more robust performance.

5   Applications

In this section, the presented algorithms for recursive Bayesian estimation are tested and compared using an academic example, which consists of a 3-degree-of-freedom spring-mass-damper system, as shown in Fig. 5. The three masses m_1, m_2 and m_3, which are all equal to 1000 kg, are connected to dampers c_1, c_2 and c_3, equal to 250 Ns/m, and linear springs k_1, k_2 and k_3, whose stiffness is equal to 10.000 N/m. The first mass m_1 is further connected to a nonlinear spring with $k_c = 10^{12}$ N/m, whose restoring force is a cubic function of the corresponding displacement $u_1(t)$. The system is lastly subjected to a white noise excitation force which is applied to m_2, as shown in Fig. 5.
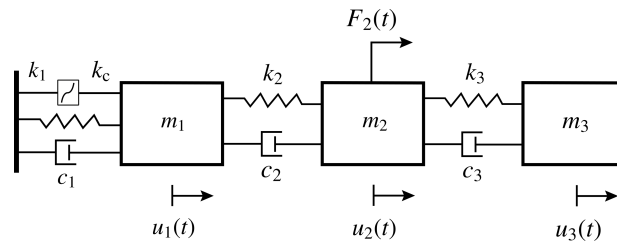

Fig. 5: Model of the 3-degree-of-freedom spring-mass-damper system in which the fist mass is connected with a nonlinear spring

The dynamic motion of the system in the continuous-time domain is described by the following nonlinear second
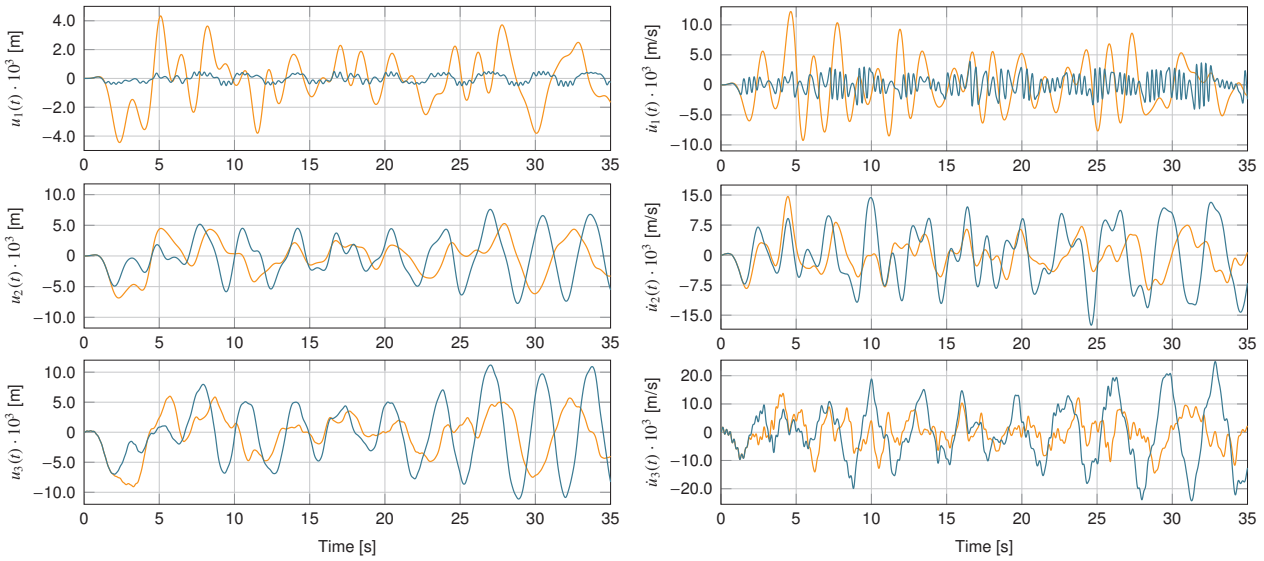
Fig. 6: Comparison of the linear (orange) and nonlinear (blue) state response

order differential equation

$$\mathbf{M\ddot{u}}(t) + \mathbf{r}(\mathbf{u}(t), \dot{\mathbf{u}}(t)) = \mathbf{p}(t) \tag{48}$$

where $\mathbf{u}(t) = [u_1(t)\ u_2(t)\ u_3(t)]^{\mathrm{T}}$ is the displacement vector, $\mathbf{M} \in \mathbb{R}^{3 \times 3}$ is the mass matrix, $\mathbf{r}(\mathbf{u}(t), \dot{\mathbf{u}}(t)) \in \mathbb{R}^3$ is the vector of restoring forces that depends on the velocity and displacement terms and $\mathbf{p}(t) \in \mathbb{R}^3$ is the vector of external forces. Upon substitution of the system matrix expressions, Eq. (48) takes the following form

$$\underbrace{\begin{bmatrix} m_1 & 0 & 0 \\ 0 & m_2 & 0 \\ 0 & 0 & m_3 \end{bmatrix}}_{\mathbf{M}} \underbrace{\begin{bmatrix} \ddot{u}_1 \\ \ddot{u}_2 \\ \ddot{u}_3 \end{bmatrix}}_{\mathbf{\ddot{u}}(t)} + \underbrace{\begin{bmatrix} c_1 + c_2 & -c_2 & 0 \\ -c_2 & c_2 + c_3 & -c_3 \\ 0 & -c_3 & c_3 \end{bmatrix}}_{\mathbf{C}} \underbrace{\begin{bmatrix} \dot{u}_1 \\ \dot{u}_2 \\ \dot{u}_3 \end{bmatrix}}_{\mathbf{\dot{u}}(t)} + \underbrace{\begin{bmatrix} k_1 + k_2 & -k_2 & 0 \\ -k_2 & k_2 + k_3 & -k_3 \\ 0 & -k_3 & k_3 \end{bmatrix}}_{\mathbf{K}} \underbrace{\begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}}_{\mathbf{u}(t)} + \underbrace{\begin{bmatrix} k_c \cdot u_1^3 \\ 0 \\ 0 \end{bmatrix}}_{\mathbf{r}_{\mathrm{nl}}(\mathbf{u}(t))} = \underbrace{\begin{bmatrix} 0 \\ F_2 \\ 0 \end{bmatrix}}_{\mathbf{p}(t)} \tag{49}$$

in which the explicit dependency on time is omitted for the sake of simplicity and the restoring force term is expanded and written as the summation of the damping-related, linear stiffness and nonlinear stiffness contributions.

In order to highlight the effect of nonlinearity, the system is initially simulated as a linear one, in which all three masses are connected among them with springs k_1, k_2 and k_3, and subsequently as a nonlinear one, in which the cubic spring k_c is further included, using the same excitation. The difference in the simulated states is plotted in Fig. 6, where the vibration amplitude of the first mass is significantly reduced, in terms of both velocity and displacement, and the majority of energy is mainly transferred to the third mass, whose amplitude is seen to increase. The difference in the dynamics of the two systems is also evidenced by the phase space trajectories of the first mass which are presented in Fig. 7.
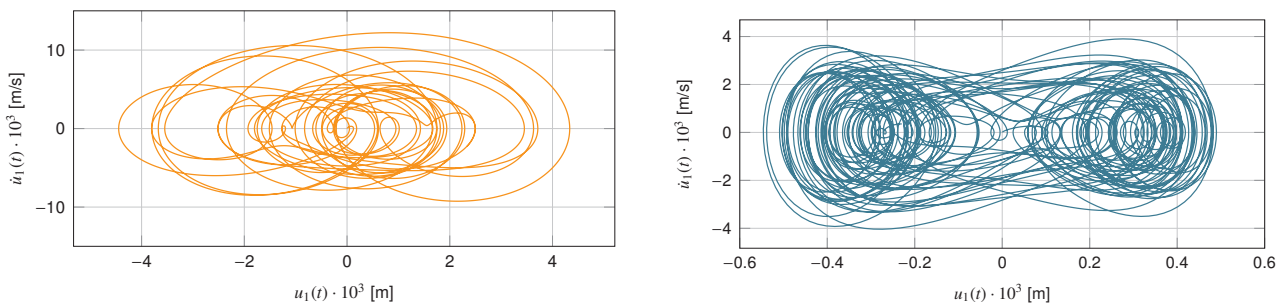

Fig. 7: Phase portrait of the first degree of freedom for the linear (left) and nonlinear (right) system

The problem of Bayesian inference in structural dynamics is presented in the following sections in the form of state, state-parameter and state-input-parameter estimation, with the aiming of calculating the posterior expectation

described by Eq. (8), when $\mathbf{g}(\mathbf{x}_k) = \mathbf{x}_k$. These quantities are oftentimes of interest, not only in structural dynamics, but also in SHM applications. Apart from the evident case of retrieving the dynamic response at unmeasured locations, the inference of these quantities may serve for identifying unknown system parameters [20, 77], as well as material constitutive models [12]. In the context of real-time vibration mitigation for structural systems, the filtering algorithms can be further combined with control schemes [78], in which case the aim is to deliver the inferred quantities of interest to the controller. Lastly, the problem of damage detection in structural and mechanical systems can be also tailored to a sequential Bayesian inference scheme. Not only in terms of damage in the form of cracks [61], but also in terms of sensor-faults [79] and accumulated fatigue damage [80].

The steps of Sequential Bayesian Inference are performed using the associated open access python library, which is available via GitHub. The structure of the object-oriented library is displayed in the UML (Unified Modeling Language) class diagram shown in Fig. 8, while each class is individually documented in the repository. The algorithms presented in this paper inherit the attributes and methods of the abstract parent class named `Filter`, which encapsulates all common attributes and methods, contained in the blue and green boxes of the class diagram, respectively. Apart from the UKF and DKF-UKF algorithms, which are based on the Unscented Transform and therefore make use of deterministically placed samples, all algorithms inherit their attributes and methods from the PF, which is illustrated in Fig. 8 by the extension arrows of the class diagram. The initialization of the algorithms is performed by the `setInitialState()` method, while the process and measurement noise covariances are specified by the `setProcessNoise` and `setMeasurementNoise` methods. The main inference steps of prediction and correction are carried out using the methods `correct()` and `predict()`, which are implemented by all filters accordingly.
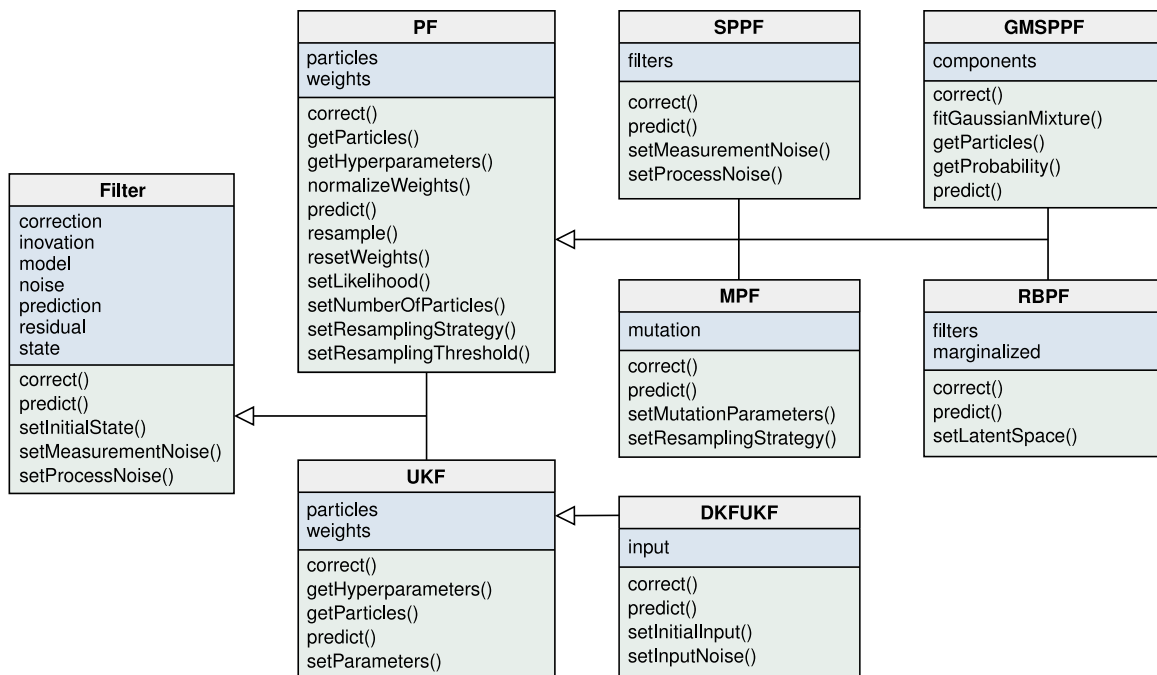

Fig. 8: Class diagram of the python library for Sequential Bayesian Inference

The results reported in the following sections are obtained by means of simulated data, which are generated according to the above-described nonlinear set-up. In all cases, the vibration response of the system is measured in terms of displacements in m_1 and accelerations in m_1, m_2 and m_3. The simulated response in terms of these quantities is obtained by means of a 4th order Runge-Kutta integration scheme, subsequently polluted with a 3% which Gaussian noise, and lastly combined with the filtering algorithms for the solution of state, state-parameter and input-state-parameter estimation problems. It should be underlined, that the results presented below are obtained upon proper tuning of the noise covariance matrices, which constitutes a decisive step for optimal estimation performance.

In this paper, the noise terms are manually tuned by minimizing the difference between measured signals, which are not used in the filtering equations, and the corresponding posterior signals. This is a standard practice in the literature for selecting the covariance values [15, 59], which are used to account for the uncertainties related to the state, measurement, input and parameter equations. These uncertainties are propagated through the filtering equations and finally reflected on the estimated probability densities. This implies that an increased uncertainty in the

process noise $\mathbf{v}_k$, which can be specified by the corresponding covariance matrix $\mathbf{Q}_k$, would allow strong fluctuations of the state vector $\mathbf{x}_k$ around the model predicted values, while a reduced uncertainty would force the state dynamics to be strictly governed by the underlying equations. Under this perspective, the covariance matrix of the input process is typically tuned to high values, so as to allow for strong variations of the external loads, such as sudden impacts. A formal way of tuning this noise term is the L-curve [81], which is extracted though from an offline step. On the other hand, a low value is assigned to the covariance matrix of the unknown parameters, thus allowing only slow variations.

Apart from the agreement between measurements and predictions, the tuning point can be selected on the basis of different optimality metrics [82], such as the whiteness of the residual sequence. The numerous contributions in this regard can be divided into two categories, the ones operating online, thus delivering an adaptive noise estimation, and the ones extracting the estimates on the basis of data batches. A straightforward approach falling into the first category, which can be also tailored to the algorithms presented in this paper, consists in augmenting the state vector with the unknown noise parameters and solving a joint state-parameter problem [83]. An alternative online estimate can be obtained using a covariance matching approach [84], which aims at estimating the noise covariance matrices using the residuals of the inference step. When relaxing the constraint of online performance, noise estimates can be obtained by means of the Autocovariance Least-Squares technique [58, 85] or using an Expectation Maximization (EM) approach [86]. It should be underlined that the topic of optimal noise identification is out of the scope of this paper however, the reader is referred to [87] for a more thorough review and comparison of noise estimation methods.

5.1 State estimation

In this first case study, which deals with the simplest form of inference in dynamic systems, the unobserved state of the structure shown in Fig. 5 is estimated using the UKF, PF and SPPF. To do so, the system dynamics described by Eq. (49) must be written in a state-space form in accordance with Eqs. (1) and (2). This requires the construction of the state and measurement functions $\mathbf{F}$ and $\mathbf{H}$, which are obtained upon definition of the state vector $\mathbf{x}(t) = \mathrm{vec}\left(\left[\mathbf{u}(t)\ \dot{\mathbf{u}}(t)\right]\right)$, as follows

$$\mathbf{F}(\mathbf{x}(t), \mathbf{p}(t)) = \begin{bmatrix} \dot{\mathbf{u}}(t) \\ -\mathbf{M}^{-1}\left(\mathbf{p}(t) - \mathbf{C}\dot{\mathbf{u}}(t) - \mathbf{K}\mathbf{u}(t) - \mathbf{r}_{\mathrm{nl}}(\mathbf{u(t)})\right) \end{bmatrix}, \quad \mathbf{H}(\mathbf{x}_k, \mathbf{p}_k) = \begin{bmatrix} \mathbf{S}_{\mathrm{d}}\,\mathbf{u}_k \\ -\mathbf{M}^{-1}\left(\mathbf{p}_k - \mathbf{C}\dot{\mathbf{u}}_k - \mathbf{K}\mathbf{u}_k - \mathbf{r}_{\mathrm{nl}}(\mathbf{u}_k)\right) \end{bmatrix}$$

where $\mathbf{S}_{\mathrm{d}}$ is a Boolean matrix selecting the observed displacements, which for this case study reads $\mathbf{S}_{\mathrm{d}} = [0\ 1\ 0]$. It should be noted that in order to perform the inference steps of each algorithm in the discrete-time domain, the state function should be further discretized, according to Eq. (5). This step is herein carried out using a 4th order Runge-Kutta scheme and yields function $\mathbf{G}(\mathbf{x}_k, \mathbf{p}_k)$, which is the discrete-time equivalent of $\mathbf{F}(\mathbf{x}(t), \mathbf{p}(t))$.
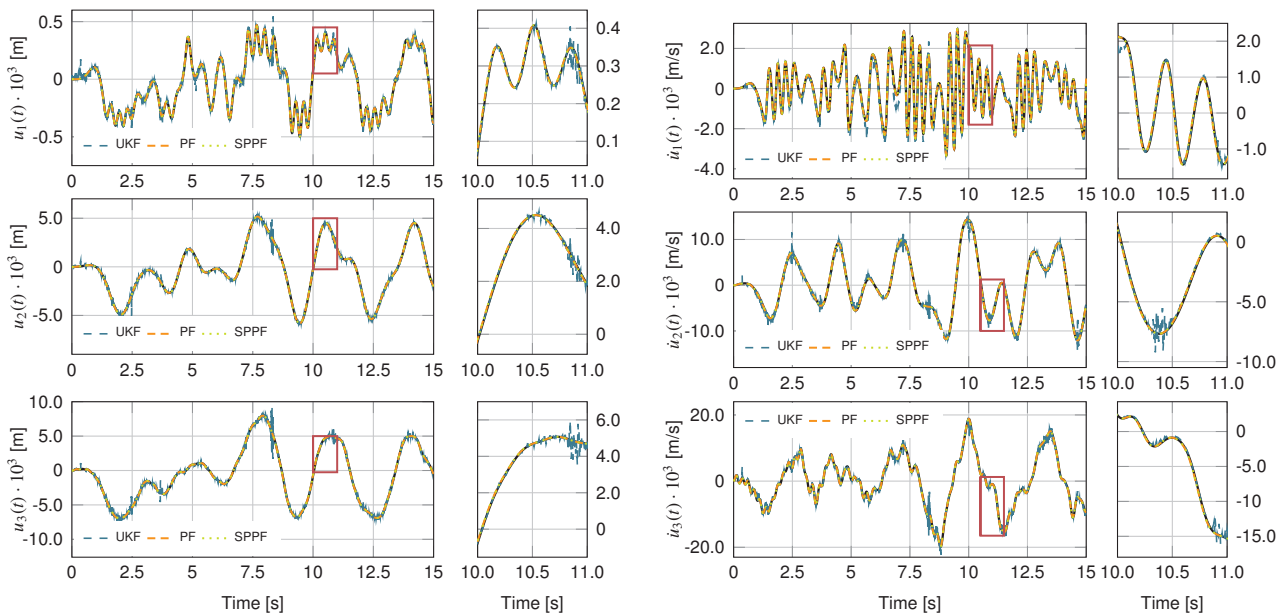

Fig. 9: Comparison of the UKF (blue), PF (orange) and SPPF (green) state estimates against the ground truth (black)

The UKF is based on a deterministic sampling of the sought-after distributions, using $2n + 1$ sigma points, where n designates the size of the state vector. For the considered example, this results to a total of 13 particles, which are symmetrically placed with respect to the prior state statistics. The PF is herein based on $N = 30$ particles, which is the required number of samples for achieving a sufficiently good performance, comparable to the one delivered by the UKF, while the estimates of the SPPF are based on 25 particles. The estimates obtained from the three algorithms are plotted in Fig. 9 in terms of displacements and velocities at all three degrees of freedom of the system, with blue, orange and green lines. The target response is represented in all figures by a continuous black line, which is only partially visible due to the good agreement between the actual and predicted time histories.

In all three runs, namely UKF, PF and SPPF, the state is initialized with a zero mean $\hat{\mathbf{x}}_{0|0} = \mathbf{0}_n$ and covariance matrix $\mathbf{P}_{0|0} = 10^{-20} \cdot \mathbf{I}_n$, where $\mathbf{0}_n$ and $\mathbf{I}_n$ denote the null and identity matrices of size n, respectively. Similarly, identical measurement and process noise covariance matrices are used for all three algorithms, equal to $\mathbf{R}_k = \mathrm{diag}\left(\left[10^{-5},\ 10^{-8},\ 10^{-8},\ 10^{-8}\right]\right)$ and $\mathbf{Q}_k = \mathrm{diag}\left(\left[10^{-9},\ 10^{-9},\ 10^{-9},\ 10^{-14},\ 10^{-14},\ 10^{-14}\right]\right)$ respectively. The UKF distribution-related parameters are set to the default values, namely $\alpha = 1$ and $\beta = 2$, while the effective sample size of the PF and SPPF, which essentially represents the resampling limit, is $N_{\mathrm{eff}} = 0.2 \cdot N$, with N denoting the number of particles.

5.2   State and parameter estimation

In the previous section, it was assumed that the parameters of the state-space model were known, which is not always the case in practical applications. As underlined in the beginning of Section 5, a typical application of sequential Bayesian inference in structural dynamics systems pertains to the identification of uncertain or unknown system parameters. Within this context, it is further assumed in this section that the nonlinear spring stiffness is an unknown system parameter and will be estimated alongside the unobserved state, using measurements of the vibration response. It should be clarified that it is only the value of k_c that is to be estimated and not the entire model class, namely the form of nonlinearity.

To do so, the state augmentation technique is used, whereby the system dynamics are written in an augmented state-space form, which is obtained by appending the unknown parameters to the state vector. In this case study, the parameter to be estimated is the value of the nonlinear stiffness term, so that $\theta = k_c$ and $\tilde{\mathbf{x}}(t) = \mathrm{vec}\left(\left[\mathbf{u}(t), \dot{\mathbf{u}}(t), \theta\right]\right)$. As pointed out in Section 3, the transition model of the parameter vector is assumed to be a Gaussian random walk, which in the continuous-time domain is translated into the following equation: $\dot{\theta} = \mathbf{0}$. This model enables the construction of the continuous-time state function, which reads

$$\mathbf{F}(\tilde{\mathbf{x}}(t), \mathbf{p}(t)) = \begin{bmatrix} \dot{\mathbf{u}}(t) \\ -\mathbf{M}^{-1}\left(\mathbf{p}(t) - \mathbf{C}\dot{\mathbf{u}}(t) - \mathbf{K}\mathbf{u}(t) - \mathbf{r}_{\mathrm{nl}}(\mathbf{u}(t))\right) \\ 0 \end{bmatrix}$$

and is transformed to the discrete-time equivalent $\tilde{\mathbf{G}}(\tilde{\mathbf{x}}_k, \mathbf{p}_k)$ by means of a 4th order Runge-Kutta scheme. The observation function $\mathbf{H}(\mathbf{x}_k, \mathbf{p}_k)$ is identical to the one used in the previous case study since the same response quantities are observed.

In this case study, the use of UKF results to the most computationally efficient solution of the Bayesian problem using only 15 sigma points, while the remaining algorithms, namely PF, MPF, GMSPPF and RBPF are based on a significantly larger amount of particles. In the latter algorithm, the states related to the dynamics of the system are marginalized, so that $\mathbf{x}_k^{\mathrm{b}} = \mathrm{vec}\left([\mathbf{u}_k,\ \dot{\mathbf{u}}_k]\right)$, and their posterior is obtained by means of a UKF, while the PF sampling takes place only for the sought-after parameter, $\mathbf{x}_k^{\mathrm{a}} = k_c$. It should be noted that major advantage of the RBPF is obtained when the filtering equations of the marginalized state can be analytically computed. When this condition is not met, which might be due to nonlinearities existing in the state and/or observation equations, the inference over the marginalized state can be performed by replacing the analytical evaluation with any filter capable of accounting for nonlinearities, such as the EKF or the UKF [88]. Consequently, the use of UKF for $\mathbf{x}_k^{\mathrm{b}}$ is herein owed to the fact that both state and observation equations are characterized by nonlinear terms.

The state estimates of each algorithm are compared against the ground truth in Fig. 10, while the corresponding parameter estimates are displayed in Fig. 11. These are obtained upon initialization of all three filters with a zero state $\hat{\mathbf{x}}_{0|0}$ and a diagonal covariance matrix $\mathbf{P}_{0|0}$, whose entries are equal to 10^{-11}, except for the one corresponding to the parameter k_c which is set to 10^{-2}. The measurement noise covariance for the UKF is $\mathbf{R} = \mathrm{diag}\left(\left[10^{-9},\ 10^{2},\ 10^{-3},\ 10^{-3}\right]\right)$, for PF: $\mathbf{R} = \mathrm{diag}\left(\left[10^{-9},\ 10^{1},\ 10^{-3},\ 10^{-3}\right]\right)$, for the MPF: $\mathbf{R} = \mathrm{diag}\left(\left[10^{-9},\ 10^{2},\ 10^{-2},\ 10^{-2}\right]\right)$, for the RBPF: $\mathbf{R} = \mathrm{diag}\left(\left[10^{-14},\ 10^{-4},\ 10^{-6},\ 10^{-4}\right]\right)$ and for the GMSPPF: $\mathbf{R} = \mathrm{diag}\left(\left[10^{-9},\ 10^{1},\ 10^{-3},\ 10^{-2}\right]\right)$, while an identical process
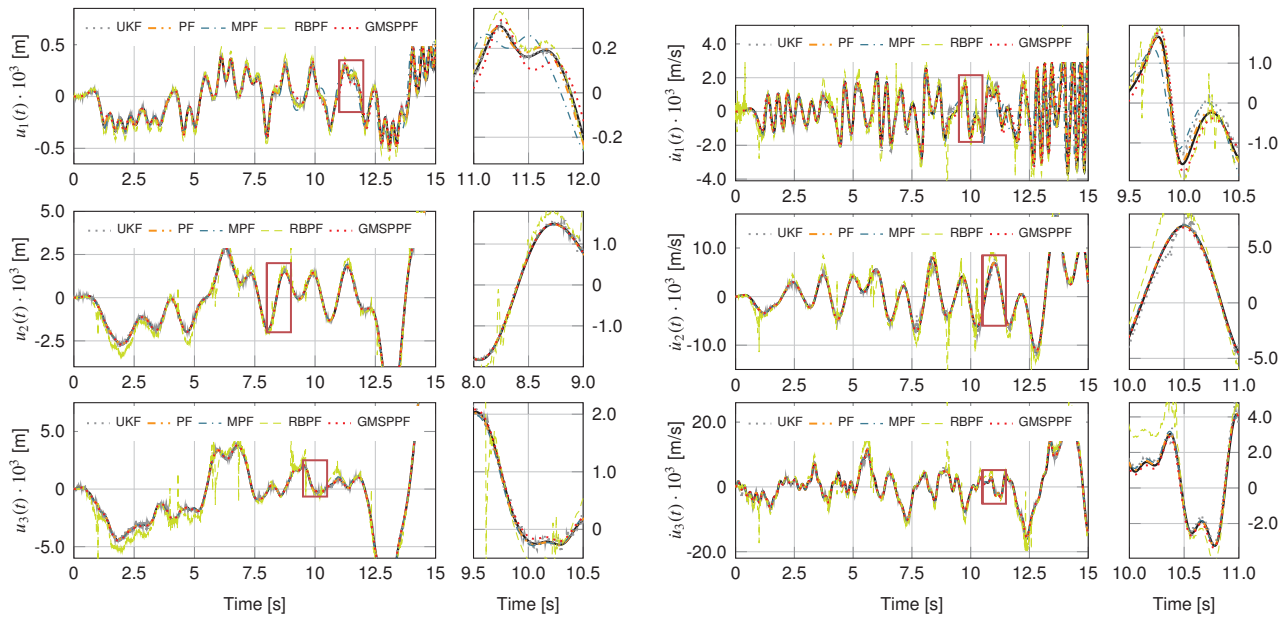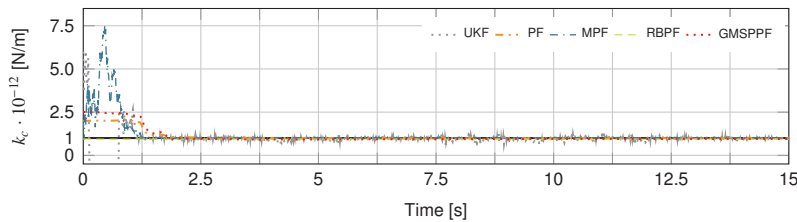
Fig. 10: Comparison of the UKF (gray), PF (orange), MPF (blue), RBPF (green) and GMSPPF (red) state estimates against the ground truth (black)

noise covariance is used for all five algorithms: $\mathbf{Q} = \mathrm{diag}\left(\left[10^{-11}, 10^{-9}, 10^{-9}, 10^{-11}, 10^{-9}, 10^{-9}, 10^{-25}\right]\right)$. The last entry of the process noise is assigned to the unknown parameter, which as discussed in Section 5 is chosen such that only slow variations of the sought-after parameters are allowed.

Fig. 11: Comparison of the UKF (gray), PF (orange), MPF (blue), RBPF (green) and GMSPPF (red) parameter estimates with the ground truth (black)

In what concerns the filtering parameters, the PF and GMSPPF in this example run with $N = 1000$ particles, whose resampling threshold is set to $N_{\mathrm{eff}} = 0.2 \cdot N$, while the MPF is seen to achieve a similar accuracy in terms of both state and parameter estimates using $N = 400$ particles with $N_{\mathrm{eff}} = 0.3 \cdot N$, $p_{\mathrm{r}} = 0.05$ and $p_{\mathrm{m}} = 0.25$. This difference in the required number of particles is owed to the mutation scheme, which introduces a more efficient exploration of the parameter space and creates a significant computational advantage for the MPF when it comes to constant parameter identification, as has been also observed in [29]. This effect is also evidenced in Fig. 11, which depicts the estimated parameter time histories. Although all three algorithms converge to the target parameter value, the MPF is seen to have a faster convergence with respect to the PF, which is also combined with a more extensive exploration of the parameter space. On the other hand, the RBPF achieves a quite fast convergence to the target parameter value, using only 70 particles for the non-marginalized part of the state vector and a resampling threshold $N_{\mathrm{eff}} = 0.2 \cdot N$.

5.3   Input, state and parameter estimation

A further relaxation to the optimal Bayesian estimation problem is introduced by assuming that the system dynamics are driven by an unmeasured and unknown input excitation, which can be also estimated along with the state and system parameters. The DKF-UKF scheme, which was initially proposed in [59] for linear systems and presented herein in Alg. 7, is therefore used for the input, state and parameter estimation of the nonlinear system illustrated in Fig. 5. It should be noted that the filter proposed in [59] has been presented for linear systems, which upon

augmentation become nonlinear. However, in this example, apart from the nonlinearity occurring from the state augmentation, the system contains an additional nonlinearity, which is owed to the nonlinear spring force acting on the first degree of freedom.
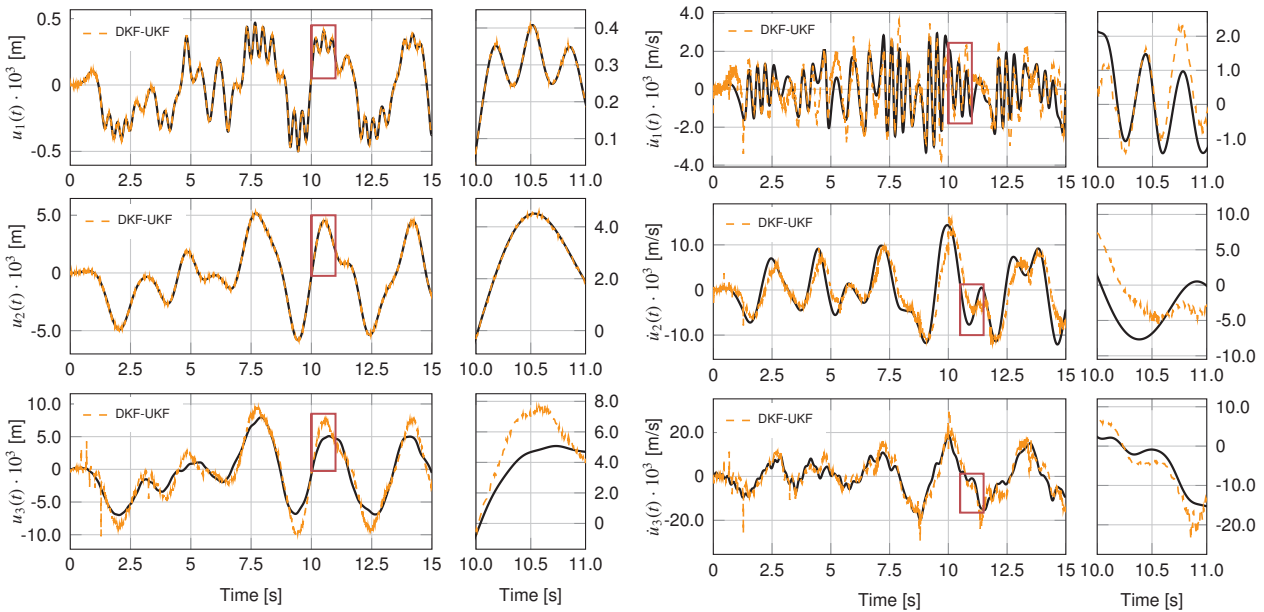

Fig. 12: Comparison of the Dual Kalman Filter - Unscented Kalman Filter (DKF-UKF) state estimates with the ground truth (black)

Similarly to the state and parameter estimation problem presented in Section 5.2, the dynamic equations of the system are initially augmented so as to accommodate the sought-after parameter as unknown state. Thereafter, a second stochastic process is assumed for the input evolution, as described in Section 4, and the input-state-parameter estimation problem is solved using Alg. 7. The state estimates, along with the actual vibration response of the system are displayed in Fig. 12, in which the latter is represented by a continuous black line and the DKF-UKF estimates are plotted with an orange dashed line. The corresponding input and parameter estimates, which are also delivered by the DKF-UKF, are presented in Fig. 13.
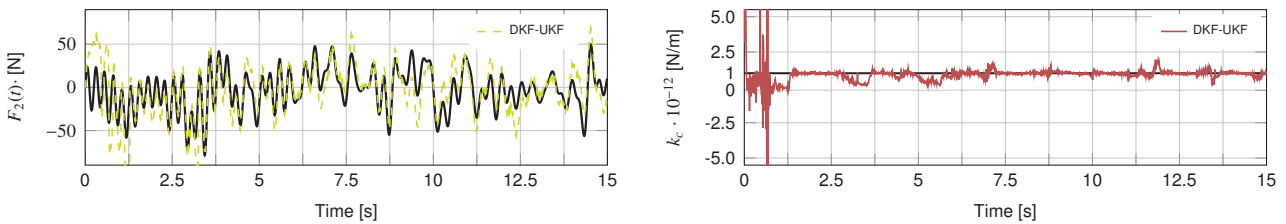

Fig. 13: Comparison of the Dual Kalman Filter - Unscented Kalman Filter (DKF-UKF) input (left) and parameter (right) estimates with the ground truth (black)

In terms of the algorithm parameters, a zero initial state $\hat{\mathbf{x}}_{0|0}$ is assumed, with covariance matrix $\mathbf{P}_{0|0} = \text{diag}\left(\left[10^{-5},\ 10^{-5},\ 10^{-5},\ 10^{-5},\ 10^{-5},\ 10^{-5},\ 10^{-2}\right]\right)$ and a zero $\hat{\mathbf{p}}_{0|0}$ is also assumed for the input signal, with covariance $\mathbf{P}_{0|0}^{\text{p}} = 10^{10} \cdot \mathbf{I}_{n_{\text{p}}}$. The measurement noise covariance is equal to $\mathbf{R}_k = \text{diag}\left(\left[10^{-11},\ 10^{-9},\ 10^{-3},\ 10^{-10}\right]\right)$ and the state process covariance matrix is tuned to the following values $\mathbf{Q}_k = \text{diag}\left(\left[10^{-10},\ 10^{-9},\ 10^{-9},\ 10^{-8},\ 10^{-8},\ 10^{-8},\ 10^{-25}\right]\right)$. The last parameter of the DKF-UKF algorithm is the noise covariance matrix of the input process, which is equal to $\mathbf{Q}_k^{\text{p}} = 2 \cdot 10^2 \cdot \mathbf{I}_{n_{\text{p}}}$.

Konstantinos E. Tatsis et. al

6   Conclusions

This contribution provides an overview of particle-based methods for sequential Bayesian inference of nonlinear and non-Gaussian dynamic systems for Structural Health Monitoring applications. The presented class of filtering approaches is intended to tackle the difficulties associated with the nonlinearities and uncertainties encountered in typical engineering applications. As such, the formulation and specifics for a number of particle-based filtering algorithms are described along with the detailed implementation steps. The performance of the most representative recursive filters, whose implementation is available at https://github.com/ETH-WindMil/JSD-SBI, is also compared and discussed on the basis of a simple illustrative example, which is used for state, state-parameter and input-state parameter estimation.

Authors' Contributions

Eleni Chatzi and Vasilis Dertimanis conceived and established an initiating structure for this review paper. All three authors elaborated on the initial structure with a thorough articulation of suited algorithms within the SHM context. Konstantinos Tatsis took the lead in writing the manuscript, performed the necessary computations and further developed a Python library, which is to be made available to the community upon publication of this work. Eleni Chatzi and Vasilis Dertimanis offered parts of a preliminary code structure in MATLAB, together with preliminary drafts of the figures presented for the Particle filter algorithm. All authors provided critical feedback and helped shape the research, analysis and manuscript.

Acknowledgements

References

[1]   A. W. Smyth, S. F. Masri, E. B. Kosmatopoulos, A. G. Chassiakos, and T. K. Caughey. Development of adaptive modeling techniques for non-linear hysteretic systems. *International Journal of Non-Linear Mechanics*, 37(8):1435–1451, 2002. ISSN 00207462. doi:10.1016/S0020-7462(02)00031-8.

[2]   S. Nagarajaiah and Z. Li. Time segmented least squares identification of base isolated buildings. *Soil Dynamics and Earthquake Engineering*, 24(8):577–586, 2004. ISSN 02677261. doi:10.1016/j.soildyn.2004.04.004.

[3]   J.-N. Juang and R. S. Pappa. An eigensystem realization algorithm for modal parameter identification and model reduction. *Journal of Guidance, Control, and Dynamics*, 8(5):620–627, 1985. ISSN 0731-5090. doi:10.2514/3.20031.

[4]   P. van Overschee and B. De Moor. N4SID: Subspace Algorithms for the Identification of Combined Deterministic-Stochastic Systems. *Automatica*, 30(1):75–93, 1994.

[5]   B. Peeters and G. De Roeck. Reference-based stochastic subspace identification for output-only modal analysis. *Mechanical Systems and Signal Processing*, 13(6):855–878, 1999. ISSN 08883270. doi:10.1006/mssp.1999.1249.

[6]   T. Katayama. *Subspace methods for system identification*. Springer-Verlag, 2005. ISBN 9781852339814.

[7]   J. N. Juang, M. Phan, L. G. Horta, and R. W. Longman. Identification of observer/kalman filter markov parameters - theory and experiments. *Journal of Guidance, Control, and Dynamics*, 16(2):320–329, 1993. ISSN 07315090. doi:10.2514/3.21006.

[8]   G. Fraraccio, A. Brügger, and R. Betti. Identification and damage detection in structures subjected to base excitation. *Experimental Mechanics*, 48(4):521–528, 2008. ISSN 00144851. doi:10.1007/s11340-008-9124-6.

[9]   H. Ebrahimian, R. Astroza, J. P. Conte, and R. A. de Callafon. Nonlinear finite element model updating for damage identification of civil structures using batch Bayesian estimation. *Mechanical Systems and Signal Processing*, 84:194–222, 2017. ISSN 10961216. doi:10.1016/j.ymssp.2016.02.002.

[10] S. P. Tchemodanova, M. Sanayei, B. Moaveni, K. Tatsis, and E. Chatzi. Strain predictions at unmeasured locations of a substructure using sparse response-only vibration measurements. *Journal of Civil Structural Health Monitoring*, 11(4): 1113–1136, 2021. ISSN 21905479. doi:10.1007/s13349-021-00476-x.

[11] S. Vettori, E. Di Lorenzo, B. Peeters, and E. Chatzi. On the influence of physical sensor placement on the virtualization of wind turbine blades testing. In *Proceedings of the 10th International Conference on Structural Health of Intelligent Infrastructure*, 2021.

[12] H. Ebrahimian, R. Astroza, and J. P. Conte. Extended Kalman filter for material parameter estimation in nonlinear structural finite element models using direct differentiation method. *Earthquake Engineering & Structural Dynamics*, 44:1459–1522, 2015.

[13] Z. Wan, T. Wang, L. Li, and Z. Xu. A Novel Coupled State/Input/Parameter Identification Method for Linear Structural Systems. *Shock and Vibration*, 2018, 2018. ISSN 10709622. doi:10.1155/2018/7691721.

[14] F. Naets, J. Croes, and W. Desmet. An online coupled state/input/parameter estimation approach for structural dynamics. *Computer Methods in Applied Mechanics and Engineering*, 283:1167–1188, 2015. ISSN 00457825. doi:10.1016/j.cma.2014.08.010.

[15] K. Maes, F. Karlsson, and G. Lombaert. Tracking of inputs, states and parameters of linear structural dynamic systems. *Mechanical Systems and Signal Processing*, 130:755–775, 2019. ISSN 10961216. doi:10.1016/j.ymssp.2019.04.048.

[16] T. J. Rogers, K. Worden, and E. J. Cross. On the application of Gaussian process latent force models for joint input-state-parameter estimation: With a view to Bayesian operational identification. *Mechanical Systems and Signal Processing*, 140: 106580, 2020. ISSN 10961216. doi:10.1016/j.ymssp.2019.106580.

[17] M. Wu and A. W. Smyth. Application of the unscented Kalman filter for real-time nonlinear structural system identification. *Structural Control and Health Monitoring*, 14:971–990, 2007. ISSN 15452255. doi:10.1002/stc.

[18] E. N. Chatzi, A. W. Smyth, and S. F. Masri. Experimental application of on-line parametric identification for nonlinear hysteretic systems with model uncertainty. *Structural Safety*, 32(5):326–337, 2010. ISSN 0167-4730. doi:10.1016/j.strusafe.2010.03.008.

[19] R. Astroza, H. Ebrahimian, and J. P. Conte. Material Parameter Identification in Distributed Plasticity FE Models of Frame-Type Structures Using Nonlinear Stochastic Filtering. *Journal of Engineering Mechanics*, 141(5):04014149, 2015. ISSN 0733-9399. doi:10.1061/(asce)em.1943-7889.0000851.

[20] K. Erazo and S. Nagarajaiah. Bayesian structural identification of a hysteretic negative stiffness earthquake protection system using unscented Kalman filtering. *Structural Control and Health Monitoring*, 25(9):1–18, 2018. ISSN 15452263. doi:10.1002/stc.2203.

[21] T. Chen, J. Morris, and E. Martin. Particle filters for state and parameter estimation in batch processes. *Journal of Process Control*, 15(6):665–673, 2005. ISSN 09591524. doi:10.1016/j.jprocont.2005.01.001.

[22] J. Ching, J. L. Beck, K. A. Porter, and R. Shaikhutdinov. Bayesian State Estimation Method for Nonlinear Systems and Its Application to Recorded Seismic Response. *Journal of Engineering Mechanics*, 132(4):396–410, 2006. doi:10.1061/(ASCE)0733-9399(2006)132.

[23] A. Doucet, S. Godsill, and C. Andrieu. On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10(3):197–208, 2000. ISSN 09603174. doi:10.1023/A:1008935410038.

[24] P. Del Moral, A. Doucet, and A. Jasra. Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68:411–436, 2006.

[25] K. Erazo and S. Nagarajaiah. An offline approach for output-only Bayesian identification of stochastic nonlinear systems using unscented Kalman filtering. *Journal of Sound and Vibration*, 397:222–240, 2017. ISSN 10958568. doi:10.1016/j.jsv.2017.03.001.

[26] K. Tatsis, L. Wu, P. Tiso, and E. Chatzi. State estimation of geometrically non-linear systems using reduced-order models. In *Proceedings of the 6th International Symposium on Life-Cycle Civil Engineering*, Ghent, Belgium, 2018.

[27] N. M. Kwok, G. Fang, and W. Zhou. Evolutionary particle filter: Re-sampling from the genetic algorithm perspective. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, pages 1053–1058, 2005. ISBN 0780389123. doi:10.1109/IROS.2005.1545119.

[28] S. Park, J. Hwang, K. Rou, and E. Kim. A new particle filter inspired by biological evolution: Genetic filter. In *Proceedings of World Academy of Science, Engineering and Technology*, 2007.

[29] E. N. Chatzi and A. W. Smyth. Particle filter scheme with mutation for the estimation of time-invariant parameters in structural health monitoring. *Structural Control and Health Monitoring*, 20:1081–1095, 2013. ISSN 09738916. doi:10.1002/stc.

[30] R. van der Merwe and E. Wan. Gaussian mixture sigma-point particle filters for sequential probabilistic inference in dynamic state-space models. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages VI–701–4, 2003. ISBN 0-7803-7663-3. doi:10.1109/icassp.2003.1201778.

[31] P. L. R. Goodall and V. Kadirkamanathan. Estimation of parameters in a linear state space model using a Rao-Blackwellised particle filter. *IEE Proceedings - Control Theory and Applications*, 151(6):727–738, 2004. ISSN 14746670. doi:10.1049/ip-cta.

[32] A. Olivier and A. W. Smyth. Particle filtering and marginalization for parameter identification in structural systems. *Structural Control and Health Monitoring*, 24(3):1–25, 2017. ISSN 15452263. doi:10.1002/stc.1874.

[33] A. Doucet, N. de Freitas, and N. Gordon. *Sequential Monte Carlo Methods in Practice*. Springer, 2001. ISBN 978-1-4419-2887-0. doi:10.1198/tech.2003.s23.

[34] S. Särkkä. *Bayesian filtering and smoothing.* Cambridge University Press, 2010. ISBN 9781139344203. doi:10.1017/CBO9781139344203.

[35] K. Tatsis and E. Lourens. A comparison of two Kalman-type filters for robust extrapolation of offshore wind turbine support structure response. In *Fifth International Symposium on Life-Cycle Civil Engineering (IALCCE 2016)*, pages 209–216, Delft, The Netherlands, 2016. ISBN 9781315375175.

[36] R. Nayek, S. Chakraborty, and S. Narasimhan. A Gaussian process latent force model for joint input-state estimation in linear structural systems. *Mechanical Systems and Signal Processing*, 128:497–530, 2019. ISSN 10961216. doi:10.1016/j.ymssp.2019.03.048.

[37] Y. Lei, D. Xia, K. Erazo, and S. Nagarajaiah. A novel unscented Kalman filter for recursive state-input-system identification of nonlinear systems. *Mechanical Systems and Signal Processing*, 127:120–135, 2019. ISSN 10961216. doi:10.1016/j.ymssp.2019.03.013.

[38] T. J. Rogers, K. Worden, and E. J. Cross. Bayesian Joint Input-State Estimation for Nonlinear Systems. *Vibration*, 3(3):281–303, 2020. doi:10.3390/vibration3030020.

[39] P. K. Kitanidis. Unbiased minimum-variance linear state estimation. *Automatica*, 23(6):775–778, 1987. ISSN 00051098. doi:10.1016/0005-1098(87)90037-9.

[40] C.-s. Hsieh. Extension of unbiased minimum-variance input and state estimation for systems with unknown inputs. *Automatica*, 45(9):2149–2153, 2009. ISSN 0005-1098. doi:10.1016/j.automatica.2009.05.004.

[41] S. Gillijns and B. De Moor. Unbiased minimum-variance input and state estimation for linear discrete-time systems. *Automatica*, 43(5):934–937, 2007. ISSN 00051098. doi:10.1016/j.automatica.2006.11.016.

[42] S. Gillijns and B. De Moor. Unbiased minimum-variance input and state estimation for linear discrete-time systems with direct feedthrough. *Automatica*, 43(5):934–937, 2007. ISSN 00051098. doi:10.1016/j.automatica.2006.11.016.

[43] E. Lourens, C. Papadimitriou, S. Gillijns, E. Reynders, G. De Roeck, et al. Joint input-response estimation for structural systems based on reduced-order models and vibration data from a limited number of sensors. *Mechanical Systems and Signal Processing*, 29:310–327, 2012. ISSN 08883270. doi:10.1016/j.ymssp.2012.01.011.

[44] E. Lourens, E. Reynders, G. De Roeck, G. Degrande, and G. Lombaert. An augmented Kalman filter for force identification in structural dynamics. *Mechanical Systems and Signal Processing*, 27(1):446–460, 2012. ISSN 08883270. doi:10.1016/j.ymssp.2011.09.025.

[45] F. Naets, J. Cuadrado, and W. Desmet. Stable force identification in structural dynamics using Kalman filtering and dummy-measurements. *Mechanical Systems and Signal Processing*, 50-51:235–248, 2015. ISSN 10961216. doi:10.1016/j.ymssp.2014.05.042.

[46] S. E. Azam, E. Chatzi, and C. Papadimitriou. A dual Kalman filter approach for state estimation via output-only acceleration measurements. *Mechanical Systems and Signal Processing*, 60:866–886, 2015. ISSN 10961216. doi:10.1016/j.ymssp.2015.02.001.

[47] S. E. Azam, E. Chatzi, C. Papadimitriou, and A. Smyth. Experimental validation of the Kalman-type filters for online and real-time state and input estimation. *Journal of Vibration and Control*, 23(15):1–26, 2015. doi:10.1177/1077546315617672.

[48] K. Maes, E. Lourens, K. Van Nimmen, E. Reynders, G. De Roeck, et al. Design of sensor networks for instantaneous inversion of modally reduced order models in structural dynamics. *Mechanical Systems and Signal Processing*, 52-53(1):628–644, 2015. ISSN 10961216. doi:10.1016/j.ymssp.2014.07.018.

[49] K. Maes, S. Gillijns, and G. Lombaert. A smoothing algorithm for joint input-state estimation in structural dynamics. *Mechanical Systems and Signal Processing*, 98:292–309, 2018. ISSN 10961216. doi:10.1016/j.ymssp.2017.04.047.

[50] P. J. Smith, S. L. Dance, M. J. Baines, N. K. Nichols, and T. R. Scott. Variational data assimilation for parameter estimation: Application to a simple morphodynamic model. *Ocean Dynamics*, 59(5):697–708, 2009. ISSN 16167341. doi:10.1007/s10236-009-0205-6.

[51] S. Chitralekha, J. Prakash, H. Raghavan, R. Gopaluni, and S. Shah. *Comparison of Expectation-Maximization based parameter estimation using Particle Filter, Unscented and Extended Kalman Filtering techniques*, volume 42. IFAC, 2009. ISBN 9783902661470. doi:10.3182/20090706-3-fr-2004.00133.

[52] A. Carrassi and S. Vannitsem. State and parameter estimation with the extended Kalman filter: An alternative formulation of the model error dynamics. *Quarterly Journal of the Royal Meteorological Society*, 137(655):435–451, 2011. ISSN 00359009. doi:10.1002/qj.762.

[53] X. Yang and T. Delsole. Using the ensemble Kalman filter to estimate multiplicative model parameters. *Tellus, Series A: Dynamic Meteorology and Oceanography*, 61(5):601–609, 2009. ISSN 02806495. doi:10.1111/j.1600-0870.2009.00407.x.

[54] H. Koyama and M. Watanabe. Reducing forecast errors due to model imperfections using ensemble kalman filtering. *Monthly Weather Review*, 138(8):3316–3332, 2010. ISSN 00270644. doi:10.1175/2010MWR3067.1.

[55] R. K. Mehra. On the Identification of Variances and Adaptive Kalman Filtering. *IEEE Transactions on Automatic Control*, 15(2):175–184, 1970. ISSN 15582523. doi:10.1109/TAC.1972.1100153.

[56] K. A. Myers and B. D. Tapley. Adaptive sequential estimation with unknown noise statistics. *IEEE Transactions on Automatic Control*, 21(4):520–523, 1976. ISSN 19457197. doi:10.1210/jcem-47-4-877.

[57] K. E. Tatsis, V. K. Dertimanis, and E. N. Chatzi. Adaptive process and measurement noise identification for recursive Bayesian estimation. In *Proceedings of the 38th IMAC*, Houston, USA, 2020.

[58] B. J. Odelson, M. R. Rajamani, and J. B. Rawlings. A new autocovariance least-squares method for estimating noise covariances. *Automatica*, 42(2):303–308, 2006. ISSN 00051098. doi:10.1016/j.automatica.2005.09.006.

[59] V. K. Dertimanis, E. N. Chatzi, S. Eftekhar Azam, and C. Papadimitriou. Input-state-parameter estimation of structural systems from limited output information. *Mechanical Systems and Signal Processing*, 126:711–746, 2019. ISSN 10961216. doi:10.1016/j.ymssp.2019.02.040.

[60] K. E. Tatsis, V. K. Dertimanis, C. Papadimitriou, E. Lourens, and E. N. Chatzi. A general substructure-based framework for input-state estimation using limited output measurements. *Mechanical Systems and Signal Processing*, 150:107223, 2021. ISSN 10961216. doi:10.1016/j.ymssp.2020.107223.

[61] K. E. Tatsis, K. Agathos, E. N. Chatzi, and V. K. Dertimanis. A hierarchical output-only Bayesian approach for online vibration-based crack detection using parametric reduced-order models. *Mechanical Systems and Signal Processing*, 167:108558, 2022. ISSN 0888-3270. doi:10.1016/j.ymssp.2021.108558.

[62] M. B. Rhudy and Y. Gu. Online stochastic convergence analysis of the Kalman filter. *International Journal of Stochastic Analysis*, 2013. ISSN 20903332. doi:10.1155/2013/240295.

[63] C. Papadimitriou, C.-P. Fritzen, P. Kraemer, and E. Ntotsios. Fatigue predictions in entire body of metallic structures from a limited number of vibration sensors using Kalman filtering. *Structural Control and Health Monitoring*, 18:554–573, 2011. doi:10.1002/stc.

[64] E. A. Wan and R. van der Merwe. The Unscented Kalman Filter for Nonlinear Estimation. In *Proceedings of the IEEE*, pages 153–158, 2000. ISBN 0780358007. doi:10.1109/ASSPCC.2000.882463.

[65] S. J. Julier and J. K. Uhlmann. Unscented Filtering and Nonlinear Estimation. *Proceedings of the IEEE*, 92(3):401–422, 2004. ISSN 0960-894X. doi:10.1016/j.bmcl.2006.10.047.

[66] S. J. Julier, J. K. Uhlmann, and H. F. Durrant-Whyte. A new approach for filtering nonlinear systems. *Proceedings of the American Control Conference*, 3(June):1628–1632, 1995. ISSN 07431619. doi:10.1109/acc.1995.529783.

[67] M. S. 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*, 50(2):174–188, 2002. ISSN 1053587X. doi:10.1109/78.978374.

[68] J. S. Liu. Metropolized independent sampling with comparisons to rejection sampling and importance sampling. *Statistics and Computing*, 6(2):113–119, 1996. ISSN 09603174. doi:10.1007/BF00162521.

[69] G. Zhu, D. Liang, Y. Liu, Q. Huang, and W. Gao. Improving particle filter with support vector regression for efficient visual tracking. In *Proceedings - International Conference on Image Processing, ICIP*, volume 2, pages 422–425, 2005. ISBN 0780391349. doi:10.1109/ICIP.2005.1530082.

[70] E. N. Chatzi and A. W. Smyth. The Unscented Kalman filter and particle filter methods for nonlinear structural system identification with non-collocated heterogeneous sensing. *Structural Control and Health Monitoring*, 16:99–123, 2009. ISSN 05529360. doi:10.1002/stc.

[71] W. R. Gilks and C. Berzuini. Following a moving target - Monte Carlo inference for dynamic Bayesian models. *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, 63(1):127–146, 2001. ISSN 13697412. doi:10.1111/1467-9868.00280.

[72] D. Whitley. A genetic algorithm tutorial. *Statistics and Computing*, 4:65–85, 1994.

[73] K. Ito and K. Xiong. Gaussian Filters for Nonlinear Filtering Problems. *IEEE Transactions on Automatic Control*, 45(5):910–927, 2000.

[74] F. G. D. Freitas. *Bayesian Methods for Neural Networks*. PhD thesis, Cambridge University, 1999.

[75] P. Li, R. Goodall, and V. Kadirkamanathan. Parameter estimation of railway vehicle dynamic model using rao-blackwellised particle filter. *European Control Conference, ECC 2003*, pages 2384–2389, 2003. doi:10.23919/ecc.2003.7085323.

[76] J. Berger. *Stastical Decision Theory and Bayesian Analysis.* Springer, 1985. ISBN 9781441930743. doi:10.2307/2288950.

[77] A. Calabrese, S. Strano, and M. Terzo. Adaptive constrained unscented Kalman filtering for real-time nonlinear structural system identification. *Structural Control and Health Monitoring*, 25(2):1–17, 2018. ISSN 15452263. doi:10.1002/stc.2084.

[78] M. S. Miah, E. N. Chatzi, and F. Weber. Semi-active control for vibration mitigation of structural systems incorporating uncertainties. *Smart Materials and Structures*, 24(5):55016, 2015. ISSN 1361665X. doi:10.1088/0964-1726/24/5/055016.

[79] T. Kobayashi and D. L. Simon. Application of a Bank of Kalman Filters for Aircraft Engine Fault Diagnostics. Technical Report August, NASA, 2003.

[80] F. Cadini, C. Sbarufatti, M. Corbetta, and M. Giglio. A particle filter-based model selection algorithm for fatigue damage identification on aeronautical structures. *Structural Control and Health Monitoring*, 24(11):1–17, 2017. ISSN 15452263. doi:10.1002/stc.2002.

[81] P. C. Hansen. Analysis of Discrete Ill-Posed Problems by Means of the L-Curve. *SIAM Review*, 34(4):561–580, 1992. ISSN 0036-1445. doi:10.1137/1034115.

[82] P. Matisko and V. Havlena. *Optimality tests and adaptive Kalman filter*, volume 16. IFAC, Brussels, Belgium, 2012. ISBN 9783902823069. doi:10.3182/20120711-3-BE-2027.00011.

[83] T. Kontoroupi and A. W. Smyth. Online Noise Identification for Joint State and Parameter Estimation of Nonlinear Systems. *Journal of Risk and Uncertainty in Engineering Systems*, 2(3), 2016. ISSN 2376-7642. doi:10.1061/AJRUA6.0000839.

[84] W. Gao, J. Li, G. Zhou, and Q. Li. Adaptive Kalman filtering with recursive noise estimator for integrated SINS/DVL systems. *Journal of Navigation*, 68(1):142–161, 2015. ISSN 14697785. doi:10.1017/S0373463314000484.

[85] M. R. Rajamani and J. B. Rawlings. Estimation of the disturbance structure from data using semidefinite programming and optimal weighting. *Automatica*, 45(1):142–148, 2009. ISSN 00051098. doi:10.1016/j.automatica.2008.05.032.

[86] M. Ge and E. C. Kerrigan. Noise covariance identification for nonlinear systems using expectation maximization and moving horizon estimation. *Automatica*, 77:336–343, 2017. ISSN 00051098. doi:10.1016/j.automatica.2016.11.011.

[87] J. Duník, O. Straka, O. Kost, and J. Havlík. Noise covariance matrices in state-space models: A survey and comparison of estimation methods - Part I. *International Journal of Adaptive Control and Signal Processing*, 31(11):1505–1543, 2017. doi:10.1002/acs.2783.

[88] S. Särkkä, A. Vehtari, and J. Lampinen. Rao-Blackwellized particle filter for multiple target tracking. *Information Fusion*, 8(1 SPEC. ISS.):2–15, 2007. ISSN 15662535. doi:10.1016/j.inffus.2005.09.009.