International Journal of Applied Mathematics

Volume 34 No. 5 2021, 817-872

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version) **doi:** http://dx.doi.org/10.12732/ijam.v34i5.1

A PARTICLE FILTER FOR NONLINEAR FILTERING WITH LÉVY JUMPS

Erika Hausenblas¹§, Kistosil Fahim², Pani W. Fernando³

Chair of Applied Mathematics Montanuniversität Leobe n Leoben – 8700, AUSTRIA

² Department of Mathematics Institut Teknologi Sepuluh Nopember Surabaya - 60111, INDONESIA

 3 Department of Mathematics University of Sri Jayewardenepura Gangodawila Nugegoda — 10250, SRI LANKA

Abstract: Dynamical systems arise in engineering, physical sciences as well as in social sciences. If the state of a system is known, one also knows its properties, and may, e.g., stabilise the system and prevent it from blowing up, or predict its near future. However, the state of a system consists often on internal parameters which are not always accessible. Instead, often only an observation process Y, which is a transformation of the current state, is accessible. Furthermore, a system operates in real environments; hence, itself and its observation are affected by random noise and/or disturbances. So, in reality, the dynamics of the system and the observation are corrupted by noise. The problem of nonlinear filtering is estimating the state of the system X(t) at a given time t > 0 through the data of the observation Y until time t (i.e. $\{Y(s): 0 \le s \le t\}$).

Usually, one considers models where the state process and the observation

Received: April 8, 2021

© 2021 Academic Publications

 $[\]S{\rm Correspondence}$ author

process are perturbed by Gaussian noise. When these perturbations are known to exhibit extreme behaviour, as seen frequently in application from finance or environmental studies, a model relying on the Gaussian distribution is not appropriate. A suitable alternative could be a model based on a heavy-tailed distribution, as the stable distribution. In such a model, these perturbations are allowed to have extreme values with a probability which is significantly higher than in a Gaussian-based model.

In general, a stable process can not be simulated directly. In practice, we can approximate it by a Gaussian and a compound Poisson process. In particular, we replace the small jumps by a Gaussian process. Thus, we are interested in nonlinear filtering where the signal and observation processes are corrupted by a Gaussian and a compound Poisson process. To catch up the jumps, we use methods from control engineering and construct a so-called Luenberger observer. These methods are combined with particle filters to construct an estimator of the state process, respective, an estimator of the density process. We apply this method to a mathematical pendulum, a single-link flexible joint robot, and a Van der Pol oscillator.

AMS Subject Classification: 60G35, 93B52, 62M20, 60G51, 60J75, 60H35 Key Words: nonlinear filtering; feedback algorithm; particle systems; Lévy processes; Lévy copula

1. Introduction

Stochastic filtering is one of the main approaches to tackle many problems emerging from fields such as engineering sciences, physics, fluid dynamics, and others. The main target in stochastic filtering is to predict a process evolving by a given dynamical system, usually called the signal process. Such a signal process can be the position of a drone or the state of a balanced system like a segway personal transporter or an arm of a single-link flexible joint robot. The signal process $X = \{X(t) : t \geq 0, X(0) = x_0 \in \mathbb{R}^d\}$ is usually modelled by an ordinary differential equation given by

$$dX(t) = f(X(t)) dt, \ t > 0, \quad X(0) = x_0 \in \mathbb{R}^d,$$
 (1)

where $f: \mathbb{R}^d \to \mathbb{R}^d$ is a possible nonlinear function. In many problems arising from the disciplines mentioned above, it is not possible to estimate the signal process directly. Usually, the state of the process is given by a transformation

of X, i.e., by a process $Y = \{Y(t) : t \ge 0\}$ given by

$$Y(t) = \int_0^t h(X(s)) ds, \quad t > 0, \quad Y(0) = 0 \in \mathbb{R}^m.$$
 (2)

Here, $h: \mathbb{R}^d \to \mathbb{R}^m$ is supposed to be a continuously differentiable function whose properties will be specified later on.

In practice, systems are operating in real environments and are inevitably affected by random noise and/or disturbances. The signal process X is exposed to unpredictable external perturbations. A sudden wind gust may perturb a drone; the arm of a single-link flexible joint robot may be kicked by another object nearby, a driver of a segway may be startled and flinches. Secondly, the measurements are usually corrupted by errors, depending on the quality of the sensors. These unpredictable perturbations and imprecise measurements are modelled by adding a random forcing term to the signal process X and the transformation Y. This mathematical approach to such a more realistic model is through stochastic differential equations (SDE). In this way, we model the signal process X and the observed process Y by stochastic differential equations driven by some random noise. To model the measurements errors, we add a Wiener process, to model the jumps or abrupt changes, we add a Lévy process. The observation data at time t > 0 are given by the information which accumulates all data observing Y until time t.

To be more precise, let $f: \mathbb{R}^d \to \mathbb{R}^d$, $\sigma_0: \mathbb{R}^d \to L(\mathbb{R}^{m_0}, \mathbb{R}^d)$, $\sigma_1: \mathbb{R}^d \to L(\mathbb{R}^{m_1}, \mathbb{R}^d)$ be three Lipschitz continuous functions. Also, we assume that $f \in C^1(\mathbb{R}^d; \mathbb{R}^d)$. The stochastic process L_0 be either a \mathbb{R}^{m_0} -valued Wiener process with covariance Q_0 . Let $\mathbb{L} = (L_1, L_2)$ be a $\mathbb{R}^{m_1} \times \mathbb{R}^{m_2}$ -valued compound Poisson processes. We suppose that the signal process $X = \{X(t): t \geq 0\}$ solves the following SDE:

$$\begin{cases}
 dX(t) &= f(X(t)) dt + \sigma_0(X(t^-)) dL_0(t) \\
 &+ \sigma_1(X(t^-)) dL_1(t), \quad t > 0, \\
 X(0) &= x_0 \in \mathbb{R}^d.
\end{cases}$$
(3)

To describe the observation process, let σ_2 be a real constant and let $h: \mathbb{R}^d \to \mathbb{R}^k$ and $\sigma_3: \mathbb{R}^k \to L(\mathbb{R}^{m_2}, \mathbb{R}^k)$ be two Lipschitz continuous functions. Moreover, we assume that $h \in C^1(\mathbb{R}^d; \mathbb{R}^k)$. We suppose that the observation process Y solves the following SDE

$$\begin{cases}
dY(t) &= h(X(t)) dt + \sigma_2 dW_2(t) \\
 &+ \sigma_3(Y(t^-)) dL_2(t), \quad t > 0, \\
Y(0) &= y_0 \in \mathbb{R}^k,
\end{cases} \tag{4}$$

where W_2 be a \mathbb{R}^k -valued Wiener process with covariance Q_2 . The dependence structure of L_1 and L_2 is modelled by a copula H. The observation data at time t>0 are given by the information which accumulates all data observing Y until time t; or, mathematically, by the σ -field generated by Y until time t. To be more precise, let $\{\mathcal{X}_t: t\geq 0\}$ and $\{\mathcal{Y}_t: t\geq 0\}$ be the filtration defined by $\mathcal{X}_t = \sigma(\{X(s), s\leq t\})$ and $\mathcal{Y}_t = \sigma(\{Y(s), s\leq t\})$, respectively. In addition, let $\mathcal{X} = \sigma(\cup_{t\geq 0} \mathcal{X}_t)$ and $\mathcal{Y} = \sigma(\cup_{t\geq 0} \mathcal{Y}_t)$.

The filtering problem consists in determining at a fixed time t>0 an estimator of the process X(t) which is usually the conditional distribution of the signal X(t), given the information accumulated from observing Y in the time interval [0,t]. The estimation is required to satisfy three properties, which have then been commonly adhered to:

- \circ Causality, X_t being estimated using $\{Y_s\}_{s < t}$;
- Optimality, the estimate \hat{X}_t minimizing the mean square error $E[|X \hat{X}_t|^2]$ if the variance is finite;
- Online estimation, the estimate \hat{X}_t being available at any arbitrary time t.

This requirements are satisfied by the Bayes estimator given by the conditional expectation

$$\hat{X}(t) = \mathbb{E}[X(t) \mid \mathcal{Y}_t], \quad t > 0.$$
(5)

To compute also functionals of X(t), one is interested additionally in an estimator of the density of X(t), that is, for ϕ being twice differentiable, the Bayes estimator given by the conditional expectation

$$\hat{\pi}_t(\phi) = \mathbb{E}\left[\phi(X(t)) \mid \mathcal{Y}_t\right], \quad \phi \in C^2(\mathbb{R}^d; \mathbb{R}). \tag{6}$$

Kolmogorov and Wiener obtained the first results in linear filtering of random processes, [34, 35] for the discrete case and [63] for the continuous case. Wiener solved the problem using the spectral theory of stationary processes. [30] and [31] have extended the work of Wiener to the case where the signal process X is given by the following linear stochastic integral equation

$$X(t) = X(0) + \int_0^t f(X(s))ds + \int_0^t \sigma(X(s))dW_1(s),$$

where $W_1 = \{W_1(t) : t \ge 0\}$ is a Wiener process and observation process solve a linear stochastic equation. The motivation for linear filtering came mainly from applications in aerospace engineering.

While Bucy and Kalman were pioneered the theory of linear stochastic filtering, [58, 59] were initiating the development of the probabilistic approach to nonlinear filtering. [39, 40] and also [65] independently developed the continuous nonlinear theory.

Assuming that X and Y solve some nonlinear SDEs as above, [39] derived the equation for nonlinear filtering, that is, the equation for the conditional distribution of X knowing Y. This is a nonlinear Stochastic PDE and is also know as Kushner-Stratonovich equation (or the Fujisaki-Kallianpur-Kunita equation). [70] developed the equivalent linear equation by a separate technique. [29] derived a Bayes' formula which, also known as the Kallianpur-Striebel formula, and permits to derive the Kushner equation from the Zakai equation.

The stochastic partial differential equations associated with the filtering equations with observation having continuous paths were further investigated, in particular, by [36, 37, 38] using methods inherited from classical PDE theory and by [48, 49] using a functional analytic approach. They analysed the density process

$$\pi_t(\phi) := \mathbb{E}\left[\phi(X(t)) \mid Y(s); 0 \le s \le t\right], \quad \phi \in C_b^2(\mathbb{R}^d; \mathbb{R}).$$

Interpreting the filtering equation as a stochastic partial differential equation, Krylov and Rozovskii were able to weaken the condition $\phi \in C_b^2(\mathbb{R}^d; \mathbb{R})$.

Nowadays, nonlinear filtering is well developed and has been extensively studied, see, e.g. [71, 50, 52, 54]. The following monographs are devoted to this topic: [6, 7, 49, 67] and the Oxford Handbook of Nonlinear Filtering [15]. In the case where the signal process is driven by a Lévy process, fewer works can be found. Most of these works deal with the issue where the signal process is driven by a Wiener process, and the observation process is driven by a Lévy process with Gaussian part, see, e.g. [2, 12, 21, 22, 57]. Nonlinear filtering with a pure jump process is considered in fewer works, see, e.g. [51, 8, 24] and the references therein.

As already mentioned above, there are many different filtering methods introduced in the literature. One of the essential technique is Particle filters was initially developed in the 1940s by [44]. The author suggested studying systems by evaluating the properties of sets of particles rather than the properties of individual particles. Moreover, [64] proposed something much like particle

filtering. However, it wasn't until the 1980s that the computing power was sufficient for their implementation. The particle filter works well for problems that are difficult for the conventional Kalman filter (i.e., systems that are highly nonlinear). Particle filtering has many other names, i.e. [25], interacting particle approximations [45], sequential importance sampling [18][Chapter 11], the condensation algorithm [27, 43], sequential Monte Carlo (SMC) filtering [4, 14], Monte Carlo filtering [33], and survival of the fittest [32]. The origins of particle filtering can be found in [26]. Interesting book on the particle filter can be seen in [18, 53, 55, 6]. Recently, particle filtering has attracted the attention of many researchers, see [56, 62, 28, 66, 68, 41, 61, 20, 47, 11].

In this paper, we focus on particle filtering methods to provide an estimator \hat{X} of the state of the system. The focus of this work is how to treat the jump part of the random force. To deal with the jumps, we use techniques coming from engineering, i.e. we use a feedback algorithm to catch up the original process. In addition, we illustrate the applicability by three examples, the mathematical pendulum perturbed by random kicks, a single-link flexible joint robot coming from engineering, and the Van der Pol oscillator. For the particle filter, we simulate a sample of virtual twins catching up the jumps by a feedback algorithm. In a second step, we deal with the Gaussian noise, by implementing incorporating the observation by adding some weights. That means we To model the dependence structure of the different Lévy processes, we use the concepts of copulas, already used in [23].

The article is organised as follows. In Section 2, we provide a brief introduction to control theory and treat the case where X and Y are only perturbed by a jump process. In Section 3, we introduce the particle filtering mechanism with jump processes, and also we provide a small description of the branching mechanism as well as the simulation results. In Section 4, we discuss the error due to sampling on one step, weak error due to discretisation in time and accumulated error. In Appendix A we summarise the necessary facts about copulas.

2. How to catch up with the Lévy jumps

Usually, a virtual observer in the sense of Luenberger is used in engineering to reconstruct the state of a system given only the output of the system and not knowing the initial data. However, in this work, we use the virtual observer, called virtual twin to avoid confusion with the observation process, to reconstruct the state of the system being perturbed by a Lévy process. Firstly, we

consider the case where jump processes only perturb X and Y. To be more precise, we consider a signal process X perturbed by a compound Poisson process L_1 . The observation process Y will also be perturbed by a compound Poisson process, denoted by L_2 and depending on L_1 . Since the perturbation by a compound Poisson process can be seen as a splitting of in deterministic systems with random initial data running a random time, we first describe the construction of a virtual twin describing the dynamical behaviour of a deterministic system where the initial state is unknown. In a second step, we analyse the whole process.

In the following, we describe briefly the setting of this section. We suppose that the signal process $X = \{X(t) : t \ge 0\}$ solves the following SDE:

$$\begin{cases}
 dX(t) = f(X(t^{-})) dt + \sigma_1(X(t^{-})) dL_1(t), & t > 0, \\
 X(0) = x_0 \in \mathbb{R}^d,
\end{cases}$$
(7)

where L_1 is an m_1 -dimensional compound Poisson process. Suppose that the observable process Y solves the following SDE

$$\begin{cases}
 dY(t) = h(X(t^{-})) dt + \sigma_2(Y(t^{-})) dL_2(t), & t > 0, \\
 Y(0) = y_0,
\end{cases}$$
(8)

where L_2 is a m_2 -dimensional compound Poisson process. Let us note that there are two cases to distinguish: one, where L_1 and L_2 are independent, and the other, where L_1 and L_2 are depending on each other. If both processes are independent, the jump times and jump sizes of L_1 and L_2 are independent. Hence, the algorithm would be the same as in the second case; one only omit L_2 .

Let us now consider the case $\mathbb{L} = (L_1, L_2)$ be two Lévy processes where L_1 and L_2 are not independent and there exists a Lévy measure ν on $\mathbb{R}^{m_1} \times \mathbb{R}^{m_2}$ such that the margin of the first coordinate represents the Lévy measure of L_1 and its projection on the second coordinate represents the Lévy measure on L_2 . Furthermore, we assume that ν is absolutely continuous with respect to the Lebesgue measure on $(\mathbb{R}^{m_1} \times \mathbb{R}^{m_2})$ and put $\sigma = \nu(\mathbb{R}^{m_1} \times \mathbb{R}^{m_2})$. Using the representation of a compound Poisson processes in terms of the sum over their jumps (see [10, Chapter 3.2]), the processes L_1 and L_2 can be represented as a sum over finitely many jumps. However, both processes jump at the same time. Therefore, let $\{\tau_n : n \in \mathbb{N}\}$ be a family of independent exponential distributed random variables with parameter σ , let

$$T_n := \sum_{j=1}^n \tau_j, \quad n \in \mathbb{N}, \tag{9}$$

and let $\{N(t): t \geq 0\}$ be the counting process defined by

$$N(t) := \sum_{j=1}^{\infty} 1_{[T_j, \infty)}(t), \quad t \ge 0.$$
 (10)

Let $\{\Delta_j \mathbb{L} = (\Delta_j L_1, \Delta_j L_2) : j \in \mathbb{N}\}$ be a family of independent, distributed $\mathbb{R}^{m_1} \times \mathbb{R}^{m_2}$ -valued random variables. The multivariate distribution function is given by

$$F(x_1, x_2) := \mathbb{P}(\{\omega \in \Omega : \Delta_j L_1(\omega) \le x_1, \Delta_j L_2(\omega) \le x_2\})$$
$$:= \frac{\nu(\{X_1 : X_1 \le x_1\} \times \{X_2 : X_2 \le x_2\})}{\sigma}.$$

Here, we set $B \leq b$ for $B = (B_1, \dots, B_d)^T$, $b = (b_1, \dots, b_d)^T \in \mathbb{R}^d$ iff $B_i \leq b_i$ for all $i = 1, 2, \dots, d$. The marginal distribution functions F_j , j = 1, 2, are given by

$$F_1(x_1) = \lim_{x_2 \to \infty} F(x_1, x_2), \quad F_2(x_2) = \lim_{x_1 \to \infty} F(x_1, x_2),$$

where $x_1 \in \mathbb{R}^{m_1}$ and $x_2 \in \mathbb{R}^{m_2}$. The Lévy process L_j for j=1,2 can be represented as

$$L_{j}(t) = \begin{cases} 0 & \text{for } N(t) = 0, \\ \sum_{k=1}^{N(t)} \Delta_{k} L_{j} & \text{for } N(t) > 0. \end{cases}$$
 (11)

Let us fix the time $t \geq 0$. We assume that we know at time $t \geq 0$ the number of jumps N(t), the jumping time $T_1, \ldots, T_{N(t)}$, and the size of the jumps of L_j , i.e. $\{\Delta_1 L_j, \Delta_2 L_j, \ldots, \Delta_{N(t)} L_j\}$ for j = 1, 2. Since for any $k = 1, \ldots, N(t)$ the jumps $\Delta_k L_1$ and $\Delta_k L_2$ are not independent, $\{\Delta_1 L_2, \Delta_2 L_2, \ldots, \Delta_{N(t)} L_2\}$ has an impact of the distribution of $\{\Delta_1 L_1, \Delta_2 L_1, \ldots, \Delta_{N(t)} L_1\}$. To model the dependence structure we use the concept of copulas, for more details and the notation used here, we refer to Appendix A. The conditional distribution of $\Delta_k L_1$ given $\Delta_k L_2$ is given by

$$\nu_{x_2}^1(x_1) := \mathbb{P}\left(\Delta_k L_1 = x_1 \mid \Delta_k L_2 = x_2\right)
= \frac{c(F_1^{-1}(x_1), F_2^{-1}(x_2))}{c(1, F_2^{-1}(x_2))} f_1(x_1), \quad x_1, x_2 \in \mathbb{R}^d, \tag{12}$$

where f_1 is the density function of F_1 , and

$$c(u_1, u_2) := \frac{\partial H}{\partial u_1 \partial u_2}(u_1, u_2)$$

is the density of the copula. Here, H is called copula. To construct an estimator of X given Y we use the fact that between the jumps, the systems (7) and (8) are governed by a deterministic nonlinear system with a random initial data. Hence, we first simulate a sample starting at the given initial data until the first jump time T_1 . At T_1 we dice a sample of the jump $\Delta_1 L_1$ given $\Delta_1 L_2$ and start modelling X by a feedback algorithm using the information given by the observation process Y until T_2 . By the feedback algorithm, the simulated process tries to catch up the signal process X. At T_2 we dice again a sample of the jump $\Delta_2 L_1$ and proceed as before.

In the following, we motivate the feedback algorithm by firstly introducing some facts about controllability and observability for linear systems, for details we refer to the book about control engineering [69, 5, 19]. Also, we explain how to construct a virtual observer, a so-called virtual twin. Here, equation (13) corresponds to (3). However, the second is not of type (4). Hence, firstly, we extend the system such that y corresponds in the linear case to (4) and then construct a virtual observer. Finally, we apply the algorithm to simulate a sample of virtual observers mimicking x by a feedback control.

2.1. A virtual twin of a deterministic linear system

In this paragraph, we consider a linear system with signal process $x = \{x(t) : t \ge 0\}$, observation process $y = \{y(t) : t \ge 0\}$ and input $u = \{u(t) : t \ge 0\}$ of the following kind

$$\dot{x}(t) = Ax(t) + Bu(t), \ y(t) = Cx(t), \ t > 0, \ x(0) = x_0 \in \mathbb{R}^d.$$
 (13)

Here, A is a $d \times d$ matrix, B a $d \times m$ matrix, and C a $k \times d$ matrix. If we emphasise the dependence of x on u and x_0 we will write $x(u, x_0, \cdot)$ instead of $x(\cdot)$.

The system (13) is called controllable, if for any T > 0 and given points $x_0, x_T \in \mathbb{R}^d$ there exists a path $u : [0,T] \to \mathbb{R}^m$ such that $x(u,x_0,T) = x_T$. A criterium if a system is controllable or not is given by the so-called controllability matrix. The controllability matrix $[A \mid B]$ is the matrix generated by

$$(B, AB, A^2B, \dots, A^{d-1}B).$$

Now, if $\operatorname{rank}([A \mid B]) = d$ the system (13) is controllable. The system (13) is called observable, if it is possible to reconstruct x_0 for any T > 0 given the observation $\{y(t) : t \in [0,T]\}$ and the path $u : [0,T] \to \mathbb{R}^m$. Note, both concepts, controllability and observability, are in duality. In particular, if a

system described by (A, B) is controllable, then the system where A is replaced by its adjoint A^T and B is replaced by B^T is observable and vice versa. From this duality follows, if $\operatorname{rank}([A^T \mid C^T]) = d$ the system (13) is observable. We have seen, if a system of type (13) is observable, then we can construct a virtual observer in the sense of Lueneberger, see, e.g. [42]. Here, the first task is to find a $d \times k$ -matrix K such that A - KC has only eigenvalues with a negative real part. However, if the system $(A \mid C)$ is observable, Theorem 3.7 in [72] says given a set $\Lambda = \{\lambda_k : k = 1, \ldots, d\}$ of complex numbers being symmetric to the real axis, then there exists a matrix K such that A - KC has eigenvalues Λ . One technique that works for the single output case, i.e. k = 1, is Ackermann's formula (see [72])

$$K := p(A)U^{-1}[0, \dots, 0, 1]^T,$$

where $U := [A^T \mid C^T]^T$ and $p(r) := (r - \lambda_1)(r - \lambda_2) \cdots (r - \lambda_d)$. Let us assume that the eigenvalues $\lambda_1, \ldots, \lambda_d$ of A - KC has negative real part. Then the system given by

$$\dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t) + K(y(t) - C\hat{x}(t)), \ t > 0, \ \hat{x}(0) = \hat{x}_0, \tag{14}$$

is stable and the difference between \hat{x} and x tends to zero as t tends to infinity. If u=0 and $err(t):=x(t)-\hat{x}(t)$, then $|err(t)|\leq |x_0-\hat{x}_0|e^{-|\lambda|t}$, where $\lambda:=\lambda_k$ such that $\Re(\lambda_k)=\max_j\Re(\lambda_j)$. If the virtual observer approaches the system without overshooting depends on the imaginary part of the eigenvalues. Thus, choosing the eigenvalues properly we can design the quality of the virtual observer.

As mentioned before, y(t) = Cx(t) and (4) are of different type. To handle an observer which accumulates the observations and does not reflect the instantaneous state of the system we extend system (13) to describe the following system

$$\dot{x}(t) = Ax(t) + Bu(t), \ y(t) = \int_0^t Cx(s) \, ds, \ t > 0, \ x(0) = x_0, \tag{15}$$

where $C \in \mathbb{R}^{k \times d}$. To transform the system in the form of (13) let us define

$$\tilde{A} := \begin{pmatrix} A & 0 \\ C & 0 \end{pmatrix}, \quad \tilde{C} := (0, I), \quad \tilde{B} := \begin{pmatrix} B \\ 0 \end{pmatrix}, \quad \tilde{x}_0 := \begin{pmatrix} x_0 \\ 0 \end{pmatrix},$$

where 0 and I are zeros and identity matrices of appropriate dimensions, respectively. Now, the system

$$\dot{\tilde{x}}(t) = \tilde{A}\tilde{x}(t) + \tilde{B}u(t), \quad \tilde{y}(t) = \tilde{C}\tilde{x}(t), \quad t > 0, \quad \tilde{x}(0) = \tilde{x}_0, \tag{16}$$

is of the form (13) and the projection onto the first d variables, denoted later on by $p_{1,d}$, is the solution of (15). It can be shown by straightforward arguments that if (13) is observable, then the system (16) is observable. In fact, the controllability matrix $[\tilde{A}^T \mid \tilde{C}^T]$ is given by

$$\begin{pmatrix} \tilde{C}^T & \begin{pmatrix} C^T \\ 0 \end{pmatrix} & \begin{pmatrix} A^T C^T \\ 0 \end{pmatrix} & \cdots & \begin{pmatrix} (A^T)^{d-1} C^T \\ 0 \end{pmatrix} \end{pmatrix},$$

from which we can see that if $rank([A^T \mid C^T]) = d$, then

$$\operatorname{rank}([\tilde{A}^T \mid \tilde{C}^T]) = d + k.$$

Similarly to before, if system (16) is observable, then we can construct a virtual observer of system (16) and the virtual observer of x given by (15) is the projection onto the first d variables. The main task is to find a vector \tilde{K} such that

$$\tilde{A} - \tilde{K}\tilde{C}$$

has only eigenvalues with a negative real part. By this vector \tilde{K} , one can construct a feedback control such that the information is given by $\tilde{y} = \{\tilde{y}(t): t \geq 0\}$ is sufficient to catch up the state of the process \tilde{x} at time t > 0. In particular, let $\hat{x} = \{\hat{x}(t): t \geq 0\}$ be a solution to

$$\begin{cases}
\dot{\hat{x}}(t) = \tilde{A}\hat{x}(t) + \begin{pmatrix} Bu(t) \\ 0 \end{pmatrix} + \tilde{K}(y(t) - \hat{y}(t)), & t > 0, \\
\hat{y}(t) = \tilde{C}\hat{x}(t), & t > 0, \\
\hat{x}(0) = \hat{x}_{0}.
\end{cases} (17)$$

Then the virtual twin of the system (15) is given by $p_{1,d}\hat{x}$.

2.2. A virtual twin for nonlinear systems

Let us consider the nonlinear system given by

$$\dot{x}(t) = f(x(t)) + Bu(t), y(t) = \int_0^t h(x(s)) \, ds, t > 0, x(0) = x_0, \tag{18}$$

where x is the state process, y is the observation and u is the perturbation, B a given matrix independent on time, and, $f: \mathbb{R}^d \to \mathbb{R}^d$, $h: \mathbb{R}^d \to \mathbb{R}^m$ are be two Lipschitz continuous and differentiable functions. Now we transform the system (18) given by

$$\dot{\tilde{x}}(t) = \tilde{f}(\tilde{x}(t)) + \tilde{B}u(t), \ \tilde{y}(t) = \tilde{C}\tilde{x}(t), \ t > 0, \ \tilde{x}(0) = \tilde{x}_0,$$
 (19)

with
$$\tilde{x} := (x, y)^T$$
, $\tilde{C} := (0, I)$, $\tilde{B} := (B, 0)^T$, $\tilde{x}_0 := (x_0, 0)^T$, and

$$\tilde{f}(\tilde{x}) := \begin{pmatrix} f(p_{1,d}\tilde{x}) \\ h(p_{1,d}\tilde{x}) \end{pmatrix},$$

where 0 and I are zeros and identity matrices of appropriate dimensions, respectively.

The virtual observer is constructed first by linearizing system (19) which results in a system of type (16). To have observability of the linearized system we assume that $D_h(x)$ and $D_f(x)$ is an observable system for all $x \in \mathbb{R}^d$ where D_f and D_h denote the Jacobi matrix of f and h, respectively. The feedback \tilde{K} is constructed from the linearized system, however, the simulation is based on the original system (18). To construct the linearization we observes that in a neighbourhood of x_0 the system (19) behaves similar to the linearized system \bar{x} given by

$$\dot{\bar{x}}(t) = \tilde{A}(\bar{x}_0)\bar{x} + \tilde{B}u(t), \ \bar{y}(t) = \tilde{C}\bar{x}(t) \, ds, \ t > 0, \ \bar{x}(0) = x_0, \tag{20}$$

where

$$\tilde{A}(x) := D_{\tilde{f}}(x) := \begin{pmatrix} D_f(x) & 0 \\ D_h(x) & 0 \end{pmatrix}$$

is the Jacobi matrix of \tilde{f} . In this way, to calculate the feedback, we replace the nonlinear system (19) by its linearised extended version.

The feedback map $\tilde{K}: \mathbb{R}^{d+k} \to \mathbb{R}^{(d+k) \times k}$ is constructed such that

$$\tilde{A}(x) - \tilde{K}(x)\tilde{C}, \quad x \in \mathbb{R}^{d+k}$$
 (21)

has only eigenvalues $\Lambda = \{\lambda_k : k+1, \ldots, d+k\}$ with negative real part. One technique that works for the single output case, i.e. k=1, is Ackermann's formula (see [72]). The virtual observer \hat{x} is the solution to

$$\begin{cases}
\dot{\hat{x}}(t) &= \begin{pmatrix} f(p_{1,d}\hat{x}(t)) \\ h(p_{1,d}\hat{x}(t)) \end{pmatrix} + \begin{pmatrix} Bu(t) \\ 0 \end{pmatrix} \\
&+ \tilde{K}(\hat{x}(t)) (y(t) - \hat{y}(t)), \quad t > 0, \\
\hat{y}(t) &= \tilde{C}\hat{x}(t), \quad t > 0, \\
\hat{x}(0) &= \hat{x}_{0},
\end{cases} (22)$$

where $y = \{y(t) : t \ge 0\}$ are the observation of the original system, the virtual twin of the system (18) is given by $p_{1,d}\hat{x}$.

2.3. A sample of virtual twins for a Lévy driven system

In this section, we construct a sample virtual copies $\{x^1, \dots, x^N\}$ catching up the state of a system driven by a compound process. In this section, we define the setting, then we present the algorithm and illustrate its applicability by some examples.

Let $\mathfrak{A} = (\Omega, \mathcal{F}, (\mathcal{F}_t)_{t \geq 0}, \mathbb{P})$ be a probability space, $\mathbb{L} = (L_1, L_2)$ be a $\mathbb{R}^{m_1} \times \mathbb{R}^{m_2}$ -valued compound Poisson process defined over \mathfrak{A} . The dependence structure of \mathbb{L} is given by the copula H. We suppose that the signal process $X = \{X(t) : t \geq 0\}$ solves

$$\begin{cases}
 dX(t) = f(X(t)) dt + \sigma_1(X(t^-)) dL_1(t), & t > 0, \\
 X(0) = x_0,
\end{cases}$$
(23)

where $f: \mathbb{R}^d \to \mathbb{R}^d$ and $\sigma_1: \mathbb{R}^d \to L(\mathbb{R}^{m_1}, \mathbb{R}^d)$ be two Lipschitz continuous functions. Also, we assume that $f \in C^1(\mathbb{R}^d; \mathbb{R}^d)$.

To describe the observable, let $h: \mathbb{R}^d \to \mathbb{R}$ and $\sigma_2: \mathbb{R} \to L(\mathbb{R}^{m_2}, \mathbb{R})$ be two Lipschitz continuous functions. Moreover, we assume that $h \in C^1(\mathbb{R}^d; \mathbb{R})$. The observation process Y solves

$$\begin{cases}
 dY(t) = h(X(t^{-})) dt + \sigma_2(Y(t^{-})) dL_2(t), & t > 0, \\
 Y(0) = y_0.
\end{cases}$$
(24)

We assume that we know in the time interval [0, T] the jump times $\{T_1, T_2, \ldots, T_K\}$ and the jumps $\{\Delta_1 L_2, \Delta_2 L_2, \ldots \Delta_K L_2\}$ of the processes Y. Let us denote the continuous part of Y by Y^c . In addition, we assume that the following assumption is valid.

Assumption 2.1. Let us suppose the controllability matrix $[D_f(x)^T \mid D_h(x)^T]$ is invertible for all $x \in \mathbb{R}^d$.

Algorithm 1. Let us denote the distribution of the initial data x_0 by χ . Let us assume that the state process X is a solution to (23) and the continuous part of the observation process Y driven by (24) is given at grid points $\{t_0 = 0 < t_1 = \tau < t_2 = 2\tau < \cdots < (K-1)\tau = t_{K-1} < K\tau = t_K = T\}$. In addition, the times $\{T_j: j \in \mathbb{N}\}$ and the jumps $\{\Delta L_j^2: j \in \mathbb{N}\}$ of the observation process are also given. To be more precise, at time T,

$$\{Y^{c}(0), Y^{c}(\tau), Y^{c}(2\tau), \dots, Y^{c}(T)\},\$$

the jump times

$$\{T_1, T_2, \ldots, T_N\}$$

and jump sizes

$$\left\{\Delta_1 L_2, \Delta_2 L_2, \dots, \Delta_N L_2\right\}$$

are given. In this algorithm, we will compute the estimator \hat{X}^{FB} of the process X. We follow the following steps.

• Step A: Find a mapping $\tilde{K}: \mathbb{R}^{d+1} \to \mathbb{R}^{d+1}$ such that the real part of the eigenvalues of (21) are strictly negative. As mentioned before, this can be done, e.g., by the Ackerman method. Here, one fix the characteristic polynomial p of the matrix $\tilde{A}(x) - \tilde{K}(x)\tilde{C}$. Then, the weights $\tilde{K}(x)$ can be computed by

$$\tilde{K}(x) := p(\tilde{A}(x)) \left([\tilde{A}^T(x) \mid \tilde{C}^T]^T \right)^{-1} [0, \dots, 0, 1]^T,$$

where

$$\tilde{A}(x) := \begin{pmatrix} D_f(x) & 0 \\ D_h(x) & 0 \end{pmatrix}, \ x \in \mathbb{R}^{d+1}, \text{ and } \tilde{C} := \underbrace{(0, \dots, 0, 1)}_{d \text{ times}}.$$
 (25)

• Step B: Fix $0 < \tau < T$, l = 1, $n_0 = 0$ and number of sample M. Simulate a sample of initial points

$$\{x_0^1, x_0^2, \dots, x_0^M\},\$$

where x_0^j , j = 1, ..., M are mutually independent and $x_0^j \stackrel{d}{=} \chi$, j = 1, ..., M. In addition, set $y_0^j := Y_0$ and $e_0^j := 0$ for j = 1, ..., N. The empirical measure of x_0 is now approximated by

$$\hat{\pi}_0^{app} = \frac{1}{M} \sum_{j=1}^{M} \delta_{x_0^j}.$$

- Step C: Find the smallest natural number n_l such that $n_l \tau \geq T_l$.
- Step D: Start from $k = n_{l-1}$. Iterate the following steps:

• Step (1): Simulate a new sample by

$$\begin{pmatrix} x_{k+1}^j \\ y_{k+1}^j \end{pmatrix} = \begin{pmatrix} x_k^j \\ y_k^j \end{pmatrix} + \tau \begin{pmatrix} f(x_k^j) \\ h(x_k^j) \end{pmatrix}$$

$$+ \tau \tilde{K} \begin{pmatrix} x_k^j \\ y_k^j \end{pmatrix} e_k^j,$$

$$e_{k+1}^j = Y^c((k+1)\tau) - y_{k+1}^j.$$

- Step (2): If $k < n_l$, replace k with k + 1 and go back to Step (1). Otherwise, continue with Step E.
- Step E: Dice a sample of $\{\Delta_l L_1^1, \Delta_l L_1^2, \Delta_l L_1^3, \dots, \Delta_l L_1^M\}$, knowing the copula and the jumps of L_2 at time T_l , i.e. $\Delta_l L_2$. Replace $x_{n_l}^j$ by $x_{n_l}^j + \sigma_1(x_{n_l}^j)\Delta_l L_1^j$ for $j = 1, \dots, M$.
- Step F: If $n_l \tau \geq T$, replace l with l+1 and go back to Step (C). Otherwise, continue to Step (G).
- Step G: Compute the feedback estimator at time $t = k\tau$ of the process X is given by

$$\hat{X}^{\text{FB}}(k\tau) := \frac{1}{M} \sum_{j=1}^{M} \hat{x}_k^j$$

and the estimator of the feedback density process $\hat{\pi}_k^{app}$ at time $k\tau$ is given by

$$\hat{\pi}_k^{app} := \frac{1}{M} \sum_{i=1}^M \delta_{\hat{x}_k^j}.$$

3. Particle filters for a general process

In case the perturbation is modelled by a Wiener process, particle filters are a well-established tool to estimate the state process, see [6, 16, 17]. In this section, we want to combine the feedback algorithm given in Algorithm 1 and the particle methods to tackle the Wiener perturbation. The problem which appears here is that the feedback depends on the driving noise W_2 , in particular, the feedback and the observation process will not be independent. Due to this

fact, one has to introduce a correction term to get convergence. The section is structured as follows. First, we give a short description of particle filters with branching; then, secondly, we show that the feedback estimator is an unbiased estimator, and in comparison to the estimator without feedback, the variance is reduced significantly.

Let us recall the setting shortly within this section. Let $f: \mathbb{R}^d \to \mathbb{R}^d$, $\sigma_0: \mathbb{R}^d \to L(\mathbb{R}^{m_0}, \mathbb{R}^d)$, $\sigma_1: \mathbb{R}^d \to L(\mathbb{R}^{m_1}, \mathbb{R}^d)$ be two Lipschitz continuous functions, and L_0 be a \mathbb{R}^{m_0} -valued Wiener process with covariance Q_0 . Also, we assume that $f \in C^1(\mathbb{R}^d; \mathbb{R}^d)$. Let $\mathbb{L} = (L_1, L_2)$ be a $\mathbb{R}^{m_1} \times \mathbb{R}^{m_2}$ -valued compound Poisson processes. We suppose that the signal process $X = \{X(t): t \geq 0\}$ solves

$$\begin{cases}
 dX(t) = f(X(t)) dt + \sigma_0(X(t^-)) dL_0(t) \\
 + \sigma_1(X(t^-)) dL_1(t), & t > 0, \\
 X(0) = x_0 \in \mathbb{R}^d,
\end{cases}$$
(26)

or, using the notation of (11)

To describe the observation process, let σ_2 be a real constant and let $h: \mathbb{R}^d \to \mathbb{R}$ and $\sigma_3: \mathbb{R} \to L(\mathbb{R}^{m_2}, \mathbb{R})$ be two Lipschitz continuous functions. Moreover, we assume that $h \in C^1(\mathbb{R}^d; \mathbb{R})$. We suppose that the observable process Y solves

$$\begin{cases}
dY(t) = h(X(t)) dt + \sigma_2 dW_2(t) \\
+\sigma_3(Y(t^-)) dL_2(t), & t > 0, \\
Y(0) = y_0 \in \mathbb{R}^k,
\end{cases}$$
(28)

where W_2 be a real-valued Wiener process with variance σ_2^2 . Again using the representation of (11) we can write

$$\begin{cases}
dY(t) = h(X(t)) dt + \sigma_2 dW_2(t) \\
+ \sum_{k=1}^{N(t)} \sigma_3(Y(T_k^-)) \Delta_k L_2, \quad t > 0, \\
Y(0) = y_0 \in \mathbb{R}^k,
\end{cases} (29)$$

As mentioned in the introduction, the filtering problem consists in determining at a fixed time t > 0 an estimator of the conditional distribution π_t of

the signal X given the information accumulated from observing Y in the time interval [0, t], i.e., to find representation to

$$\pi_t(\phi) := \mathbb{E}\left[\phi(X(t)) \mid \mathcal{Y}_t\right], \quad t \ge 0.$$

A trackable representation for π can be constructed via the Girsanov Transform. Here, one chooses new measure \mathbb{Q} such that the continuous part of Y is a martingale over the probability space $(\Omega, \mathcal{Y}, (\mathcal{Y}_t)_{t\geq 0}, \mathbb{Q})$. For this purpose let $Z = \{Z(t) : t \geq 0\}$ be given by

$$Z(t) := \exp\left(-\int_0^t h(X(s)) dW_2(s) - \frac{1}{2} \int_0^t h^2(X(s)) ds\right), \tag{30}$$

where $t \geq 0$. Setting $V(t) = Z(t)^{-1}$, we obtain as in [6, Eq. (3.30) page 56] that

$$\frac{d\mathbb{P}_t}{d\mathbb{O}}\Big|_{\mathcal{F}_t} = V(t), \quad t \ge 0. \tag{31}$$

Then the Kallianpur-Striebel formula gives (see [6, Proposition 3.16])

$$\mathbb{E}\phi(X(t)) = \int_{\mathbb{R}} \pi_t(x) \, \phi(x) \, dx = \frac{\mathbb{E}^{\mathbb{Q}} \left[\phi(X(t)) V(t) \mid \mathcal{Y} \right]}{\mathbb{E}^{\mathbb{Q}} \left[V(t) \mid \mathcal{Y} \right]}.$$

If the state process is perturbed by a Lévy process, one may also simulate a bunch of particles mimicking the virtual twin of the state process X by combining the feedback algorithm 1 and a particle filter. Due to the fact that the feedback system includes knowledge of Y, a correction term has to be added, and one has to mimic the following process

$$\begin{cases}
dX^{\text{FB}}(t) &= f(X^{\text{FB}}(t)) dt + \tilde{K}(X^{\text{FB}}(t)) \left[\frac{2}{2+\sigma_2^2} h(X(t)) - h(X^{\text{FB}}(t)) \right] dt - \tilde{K}(X^{\text{FB}}(t)) dW_2(t) \\
&+ \sigma_0(X^{\text{FB}}(t^-)) dL_0(t) \\
&+ \sum_{k=1}^{N(t)} \sigma_1(X^{\text{FB}}(T_k^-)) \Delta_k L_1, \quad t > 0, \\
X^{\text{FB}}(0) &= x_0.
\end{cases} (32)$$

The aim is to find at a given time t > 0 an estimator X(t). Later on in Section 4, we will approximate X(t) by an particle filter. Here, we will show that $X^{\text{FB}}(t)$ is indeed an estimator of X(t) and to verify the quality of the estimator $X^{\text{FB}}(t)$. Here, the order of convergence depends on several parameters: the number of particles, the size of the grid the observation process is sampled, and the gain function \tilde{K} . The advantage of the gain is that if the gain is sufficiently

large, the variance of the estimator will decrease, and one can choose a small sample size. Before starting this result, we introduce the following notation. Let $\mu(y) := \mathbb{E}[\Delta_1 L_1 \mid \Delta_1 L_2 = y]$ and

$$\varphi(y) := \mathbb{E}[(\Delta_1 L_1 - \mu(y))^2 \mid \Delta_1 L_2 = y], \quad y \in \mathbb{R}^d.$$

Assumption 3.1. For all $x, \xi \in \mathbb{R}^d$ there exists $K(x) \in \mathbb{R}^d$ such that

$$\langle (D_f(x) - K(x)D_h(x))\xi, \xi \rangle + |\sigma_1(x)|_{HS}^2 + \varphi(x) \le \lambda |\xi|^2, \tag{33}$$

where $\lambda < 0$ and $|\cdot|_{HS}^{1}$ is the Hilbert-Schmidt norm of an operator.

To combine successfully the Algorithm 1 with a particle filter usually used for the nonlinear filtering problem for SDEs with Gaussian noise, we need the following assumption on the functions f and h:

Assumption 3.2. For any $x, \xi \in \mathbb{R}^d$, $\langle D_f(x)\xi, \xi \rangle$ is positive. Moreover, we suppose the controllability matrix $[D_f(x)^T \mid D_h(x)^T]$ is invertible and its determinant is bounded away from zero for all $x \in \mathbb{R}^d$.

Lemma 1. Let us assume that $X^{FB}(t)$ is defined in (32) and X(t) is the original process given by (26). Then

(a) If Assumption 3.2 is satisfied, then X^{FB} is an unbiased estimator of X. In particular, for all $t \in [0,T]$ we have

$$\mathbb{E}[X^{FB}(t)] = \mathbb{E}[\mathbb{E}[X(t) \mid \mathcal{Y}_t]]$$

(b) If Assumption 3.1 is satisfied, then

$$\mathbb{E}[|X(t) - X^{FB}(t)|^2 \mid \mathcal{Y}_t] = \frac{1}{|\lambda|}.$$

Proof. To show Item (a), we will show that

$$\operatorname{Var}\left[X^{\text{\tiny FB}}(t) - X(t) \mid \mathcal{Y}_t\right] = 0.$$

Let $(H, \|\cdot\|)$ be a Hilbert space and $\{e_i : i \in I\}$ is an orthonormal basis of H. The Hilbert-Schmidt norm of a linear operator A on H is defined by $\|A\|_{HS}^2 = \sum_{i \in I} \|Ae_i\|^2$.

Note, that the process $V=\{V(t):t\geq 0\}$ defined in (31) solves on $(\Omega,\mathcal{F},\mathbb{P})$ the equation

 $(Y^c \text{ denotes the continuous part of } Y, \text{ i.e. the part of } Y \text{ without jumps}).$ Since the process $W_2(t) + \int_0^t h(X(s)) ds$ becomes a Brownian motion over $(\Omega, \mathcal{F}, \mathbb{Q})$, V is a $(\Omega, \mathcal{F}, \mathbb{Q})$ -martingale.

First, let us define the un-normalized estimator by

$$\hat{X}^{\scriptscriptstyle{\mathrm{FB}}}_{
ho} := \mathbb{E}^{\mathbb{Q}} \left[X^{\scriptscriptstyle{\mathrm{FB}}}(t) \mid \mathcal{Y}_t \right], \quad ext{and} \quad X_{
ho} := \mathbb{E}^{\mathbb{Q}} \left[X(t) \mid \mathcal{Y}_t \right],$$

and then the normalized estimator by

$$\hat{X}^{\scriptscriptstyle{\mathrm{FB}}}_{\varrho} := \frac{\hat{X}^{\scriptscriptstyle{\mathrm{FB}}}_{\rho}}{\mathbb{E}V(t)}, \quad \text{and} \quad X_{\varrho} := \frac{X_{\rho}}{\mathbb{E}V(t)}.$$

In the algorithm, we calculate for each small time interval first the density of the unnormalized estimator, and then, we normalize the estimator in a second step.

Observe, we know from the assumptions that $[0, \infty) \ni t \mapsto \mathbb{E}^{\mathbb{Q}}[X^{\text{FB}}(t) \mid \mathcal{Y}_t]$ and $[0, \infty) \ni t \mapsto \mathbb{E}^{\mathbb{Q}}[X(t) \mid \mathcal{Y}_t]$ are continuous. Next, we show that $\mathbb{E}^{\mathbb{Q}}[|X^{\text{FB}}(t) - X(t)|^2 \mid \mathcal{Y}_t] = 0$. Applying the Itô formula gives

$$\begin{split} &\frac{1}{2} \Big(|X(t) - X^{\text{\tiny FB}}(t)|^2 - |X(0) - X^{\text{\tiny FB}}(0)|^2 \Big) \\ &= \int_0^t \left\langle X(s) - X^{\text{\tiny FB}}(s), f(X(s)) - f(X^{\text{\tiny FB}}(s)) \right\rangle ds \\ &+ \int_0^t \left\langle X(s) - X(s), \left(\sigma_0(X(s)) - \sigma_0(X^{\text{\tiny FB}}(s)) \right) dL_0(s) \right\rangle \, + \\ &+ \int_0^t \left\langle \sigma(X(s)) - \sigma(X^{\text{\tiny FB}}(s)), \sigma(X(s)) - \sigma(X^{\text{\tiny FB}}(s)) \right\rangle ds \\ &- \int_0^t \left\langle X(s) - X^{\text{\tiny FB}}(s), K(X^{\text{\tiny FB}}(s)) \frac{2}{2 + \sigma_2^2} h(X(s)) \right\rangle ds \\ &+ \int_0^t \left\langle X(s) - X^{\text{\tiny FB}}(s), K(X^{\text{\tiny FB}}(s)) h(X^{\text{\tiny FB}}(s)) \right\rangle ds \\ &- \int_0^t \left\langle X(s) - X^{\text{\tiny FB}}(s), K(X^{\text{\tiny FB}}(s)) \frac{2}{2 + \sigma_2^2} dW_2(s) \right\rangle. \end{split}$$

Taking into account the change of measure we get

$$\begin{split} &\frac{1}{2}\mathbb{E}\Big[Z(t)|X(t)-X^{\mathrm{FB}}(t)|^2\mid\mathcal{Y}_t\Big] - \mathbb{E}|X(0)-X^{\mathrm{FB}}(0)|^2\\ &=\int_0^t\mathbb{E}\Big[Z(s)\langle X(s)-X^{\mathrm{FB}}(s),f(X(s))-f(X^{\mathrm{FB}}(s))\rangle\mid\mathcal{Y}_s\Big]\,ds\\ &-\int_0^t\mathbb{E}\Big[Z(s)\langle X(s)-X^{\mathrm{FB}}(s),K(X^{\mathrm{FB}}(s))\\ &\left[\frac{2}{2+\sigma_2^2}h(X(s))-h(X^{\mathrm{FB}}(s))\Big]\rangle\mid\mathcal{Y}_s\Big]\,ds\\ &+\frac{\sigma_2^2}{2+\sigma_2^2}\int_0^t\mathbb{E}\Big[Z(s)\langle X^{\mathrm{FB}}(s),K(X^{\mathrm{FB}}(s))h(X(s))\rangle\mid\mathcal{Y}_s\Big]\,ds\\ &-\frac{\sigma_2^2}{2+\sigma_2^2}\int_0^t\mathbb{E}\Big[Z(s)\langle X(s),K(X^{\mathrm{FB}}(s))h(X(s))\rangle\mid\mathcal{Y}_s\Big]\,ds\\ &-\int_0^tZ(s)\langle X(s)-X^{\mathrm{FB}}(s),K(X^{\mathrm{FB}}(s))\frac{2}{2+\sigma_2^2}dW_2(s)\rangle\\ &+\int_0^tZ(s)|X(s)-X^{\mathrm{FB}}(s)|^2h(X(s))dW_2(s)\\ &=\int_0^t\mathbb{E}\Big[Z(s)\langle X(s)-X^{\mathrm{FB}}(s),f(X(s))-f(X^{\mathrm{FB}}(s))\rangle\mid\mathcal{Y}_s\Big]\,ds\\ &-\int_0^t\mathbb{E}\Big[Z(s)\langle X(s)-X^{\mathrm{FB}}(s),K(X^{\mathrm{FB}}(s))\\ &\times(h(X(s))-h(X^{\mathrm{FB}}(s)))\rangle\mid\mathcal{Y}_s\Big]\,ds\\ &-\int_0^tZ(s)\langle X(s)-X^{\mathrm{FB}}(s),K(X^{\mathrm{FB}}(s))\frac{2}{2+\sigma_2^2}dW_2(s)\rangle\\ &+\int_0^tZ(s)\langle X(s)-X^{\mathrm{FB}}(s),K(X^{\mathrm{FB}}(s))\frac{2}{2+\sigma_2^2}dW_2(s)\rangle\\ &+\int_0^tZ(s)|X(s)-X^{\mathrm{FB}}(s)|^2h(X(s))dW_2(s). \end{split}$$

By taking into account the Lipschitz continuity of f and h and taking expectation we obtain

$$\frac{1}{2}\mathbb{E}\Big[\mathbb{E}^{\mathbb{Q}}\Big[|X(t) - X^{\text{\tiny FB}}(t)|^2 \mid \mathcal{Y}_t\Big]\Big] - \mathbb{E}\Big[\mathbb{E}[|X(0) - X^{\text{\tiny FB}}(0)|^2 \mid \mathcal{Y}_0]\Big] \\
\leq (K + L_h + L_b) \int_0^t \mathbb{E}\Big[\mathbb{E}^{\mathbb{Q}}\big[|X(s) - X^{\text{\tiny FB}}(s)|^2 \mid \mathcal{Y}_s\big]\Big] ds.$$

By the definition of the unormalized estimators $X_{\rho}(t)$ and $X_{\rho}^{\text{FB}}(t)$, taking into

account that $X_0^{\text{FB}} = X_0$, and applying a Grownwall type Lemma, we get

$$|X_{\rho}(t) - \hat{X}_{\rho}^{\text{FB}}(t)|^2 \le 0.$$

To show the second item we use Assumption 3.1 and get

$$\frac{1}{2}\mathbb{E}\Big[\mathbb{E}^{\mathbb{Q}}\Big[|X(t) - X^{\text{\tiny FB}}(t)|^2 \mid \mathcal{Y}_t\Big]\Big] - \mathbb{E}\Big[\mathbb{E}[|X(0) - X^{\text{\tiny FB}}(0)|^2 \mid \mathcal{Y}_0]\Big] \\
\leq -\lambda \int_0^t \mathbb{E}\Big[\mathbb{E}^{\mathbb{Q}}\big[|X(s) - X^{\text{\tiny FB}}(s)|^2 \mid \mathcal{Y}_s\big]\Big] ds.$$

The Grownwall Lemma gives the assertion.

4. The particle filter, its numerical implementation, and some examples

Calculating an particle filter of X^{FB} , one is mimicking the process X^{FB} by a sample of mutually independent particles

$${x^1(t), \dots, x^N(t) : t \ge 0}.$$

In order to realize this method on a computer, one has to discretize this algorithm in time. In particular, at a time grid $\Pi = \{t_1, t_2, \dots, t_K\}$ the observations are incorporated by associating to each particle a weight $\{v^1(t), \dots, v^N(t) : t \ge 0\}$. These weights are updated the time points $t \in \Pi$ following the Radon-Nikodym derivative $Z = \{Z(t) : t \ge 0\}$, given by equation (30). That means to each particle x^j corresponds a weight process v^j solving the differential equation

$$v^{j}(t) := 1 + \frac{\Delta_{i}Y^{c}}{t_{i+1} - t_{i}} \int_{0}^{t} v^{j}(s) h(x^{j}(s)) ds, \quad t \in (t_{i}, t_{i+1}].$$
 (35)

Here, $\Delta_i Y^c := Y^c_{t_{i+1}} - Y^c_{t_i}$, i = 1..., K and Y^c is the continuous part of Y. As one can observe v^j is a positive real valued process, determined by the observations $\{Y^c_{t_1}, Y^c_{t_2}, Y^c_{t_3}, \ldots, Y^c_{t_j}\}$. The measure-valued process at time t_k , i.e. π_{t_k} , is now approximated by the empirical measure

$$\sum_{j=1}^{N} v^{j}(t_k) \delta_{x^{j}(t_k)}.$$

To perform this at each time step, we will calculate an approximation of the density π by first calculating the so-called a priori density, and then taking into account the observation by calculating the posterior density.

However, one weak point of this method is that after a time the weights may differ, and one particle takes over nearly the whole weight. To avoid this, usually branching is introduced to get rid of the particles with low weights and multiplying particles with large weights. There are several methods to generate the random number of offspring by each particle at the end of each time step. Here one can apply different branching mechanism; we took as the number of the successors of a particle a Poisson distributed random variable. The method may be arbitrary; the expectation value of the successors must be equal to the weight.

Algorithm 2. Let be given the distribution of the initial data x_0 by χ . Let us assume that the state process X is a solution to (27) and the continuous part of the observation process Y driven by (29) is given at grid points $\{t_0 = 0 < t_1 = \tau < t_2 = 2\tau < \cdots < (K-1)\tau = t_{K-1} < K\tau = t_K = T\}$. In addition, the times $\{T_j : j \in \mathbb{N}\}$ and the jumps $\{\Delta L_j^2 : j \in \mathbb{N}\}$ of the observation process are also given. To be more precise, at time T,

$$\{Y^{c}(0), Y^{c}(\tau), Y^{c}(2\tau), \dots, Y^{c}(T)\},\$$

the jump times

$$\{T_1,T_2,\ldots,T_N\}$$

and jump sizes

$$\left\{\Delta_1 L_2, \Delta_2 L_2, \dots, \Delta_N L_2\right\}$$

are given. In this algorithm, we will compute the estimator \hat{X}^{FB} of the process X. We follow the following steps.

• Step A: Find a mapping $\tilde{K}: \mathbb{R}^{d+1} \to \mathbb{R}^{d+1}$ such that the real part of the eigenvalues of (21) are strictly negative. As mentioned before, this can be done, e.g., by the Ackerman method. Here, one fix the characteristic polynomial p of the matrix $\tilde{A}(x) - \tilde{K}(x)\tilde{C}$. Then, the weights $\tilde{K}(x)$ can be computed by

$$\tilde{K}(x) := p(\tilde{A}(x)) \left([\tilde{A}^T(x) \mid \tilde{C}^T]^T \right)^{-1} [0, \dots, 0, 1]^T,$$

where

$$\tilde{A}(x) := \begin{pmatrix} D_f(x) & 0 \\ D_h(x) & 0 \end{pmatrix}, \ x \in \mathbb{R}^{d+1}, \text{ and } \tilde{C} := \underbrace{(0, \dots, 0, 1)}_{d \text{ times}}.$$
 (36)

• Step B: Fix $0 < \tau < T$, l = 1, $n_0 = 0$ and number of sample M. Simulate a sample of initial points

$$\{x_0^1, x_0^2, \dots, x_0^M\},\$$

where x_0^j , $j=1,\ldots,M$ are mutually independent and $x_0^j \stackrel{d}{=} \chi$, $j=1,\ldots,M$. In addition, set $y_0^j := Y_0$ and $e_0^j := 0$ for $j=1,\ldots,N$. The empirical measure of x_0 is now approximated by

$$\hat{\pi}_0^{app} = \frac{1}{M} \sum_{j=1}^M \delta_{x_0^j}.$$

- Step C: Find the smallest natural number n_l such that $n_l \tau \geq T_l$.
- Step D: Start from $k = n_{l-1}$. Iterate the following steps:
- Step (1): Simulate a new sample by

$$\begin{cases} x_{k+1}^j &=& x_k^j + \tau f(x_k^j) \\ && + \tau p_{1,d} \left(\tilde{K}(x_k^j) \right) e_k^j + \xi_{k+1}^j, \\ y_{k+1}^j &=& y_k^j + \frac{2}{2 + \tau \sigma_2^2} h(x_k^j) \tau \\ && + \tau p_{d+1} \left(\tilde{K}(x_k^j) \right) e_k^j, \\ e_{k+1}^j &=& \frac{2}{2 + \tau \sigma_2^2} Y^c((k+1)\tau) - y_{k+1}^j, \end{cases}$$

where $\{\xi_{k+1}^j: j=1,\ldots,M^k\}$ is a family of independent normal distributed random variables with mean zero and variance $\tau\sigma_0$.

• Step (2): Compute the weights by

$$v_{k+1}^j := \exp\left(h(x_k^j) \, \Delta_k Y^c - \frac{1}{2} \tau \|h(x_k^j)\|^2\right),$$

where $\Delta_k Y^c = Y^c((k+1)\tau) - Y^c(k\tau)$.

• Step (3): Renormalize the weights by setting

$$\tilde{v}_{k+1}^j := \frac{v_{k+1}^j}{\sum_{j=1}^{M^k} v_{k+1}^j}.$$

 \circ Step (4): For each j generate a Poisson distributed random variable m_j^{k+1} with parameter $1/\tilde{v}_{k+1}^j$. Now, the number of offspring of the j^{th} particle is equal to m_j^{k+1} . Generate a new sample given by

$$\underbrace{\left\{\underbrace{x_{k+1}^1,\cdots,x_{k+1}^1}_{m_1^{k+1} \text{ times}},\underbrace{x_{k+1}^2,\cdots,x_{k+1}^2}_{m_2^{k+1} \text{ times}},\dots,\underbrace{x_{k+1}^{M^k},\cdots,x_{k+1}^{M^k}}_{m_{M^k}^{k+1} \text{ times}}\right\}}.$$

Compute

$$M^{k+1} := \sum_{j=1}^{M^k} m_j^{k+1}.$$

- Step (5): If $k < n_l$, replace k with k + 1 and go back to Step (1). Otherwise, continue to Step (E).
- Step E: Dice a sample of $\{\Delta_l L_1^1, \Delta_l L_1^2, \Delta_l L_1^3, \dots, \Delta_l L_1^{M^{n_l}}\}$, knowing the copula and the jumps of L_2 at time T_l , i.e. $\Delta_l L_2$. Replace $x_{n_l}^j$ by $x_{n_l}^j + \sigma_1(x_{n_l}^j)\Delta_l L_1^j$ for $j = 1, \dots, M^{n_l}$.
- Step F: If $n_l \tau \geq T$, replace l with l+1 and go back to Step (C). Otherwise, continue to Step (G).
- Step G: Compute the feedback estimator at time $t = k\tau$ of the process X is given by

$$\hat{X}_{\text{app}}^{\text{FB}}(k\tau) := \frac{1}{M^k} \sum_{j=1}^{M^k} \hat{x}_k^j,$$
 (37)

the estimator of the feedback density process π_k^{app} at time $t=\tau k$ is given by

$$\hat{\pi}_k^{app} := \frac{1}{M^k} \sum_{j=1}^{M^k} \delta_{\hat{x}_k^j}. \tag{38}$$

Under Assumption 3.1, the distribution of the sample

$$\{x_k^1, x_k^2, \cdots, x_k^{M_k}\}$$

in Step (4) of Algorithm 2 has smaller p moments.

Note that Algorithm 2 is constructed by closely following the branching particle system introduced in [13]. The main difference is first the feedback with the correction factor and the branching, i.e., the calculations of the offspring in Step (8). Here, we applied a different method. In particular, we generated the random number m_i^k as follows. Let

$$\lambda_i^k = v_i^k \times \sharp \{\text{number of particles at time } i\}.$$

Then m_i^k is Poisson distributed with parameter λ_i^k ; In particular,

$$\mathbb{P}\left(m_j^k = l\right) = \exp(-\lambda_j^k) \frac{(\lambda_j^k)^l}{l!}.$$

4.1. The numerical implementation and some examples

In the following, we give three examples to illustrate the applicability. In the example we modelled the dependence structure of the jumps by a Gumbel copula defined by

$$C(u_1, u_2) = \exp\left(-\left[(-\ln u_1)^{\theta} + (-\ln u_2)^{\theta}\right]^{\frac{1}{\theta}}\right),$$

where $u_1, u_2 \in [0, 1]$, $\theta \in (1, \infty)$. Note, the Gumbel copula is a parametrised copula; the parameter may change the dependence structure of the copula. Let the jumps be described by C_{θ} . Then, if θ is closed to one, the jumps of L_1 and L_2 are nearly independent, if θ is large, the jumps are closed. In fact, let us introduce the upper tail dependence λ_U for two random variable X_1 and X_2 with distributions F_1 and F_2 by

$$\lambda_U = \lim_{u \to 1} \mathbb{P}\left(X_1 > F_1^{\leftarrow}(u) \mid X_2 > F_2^{\leftarrow}(u)\right),\,$$

in case the limit exists. Then, in case the dependence structure of X_1 and X_2 is given by the Gumbel copula with parameter θ , the upper tail dependence λ_U is given by $2-2^{\frac{2}{\theta}}$. This possibility of considering different tail dependences was the motivation to take the Gumbel copula. For details on how to generate random variable conditioned by a copula, we refer to the appendix A and to the books [9, 60, 23].

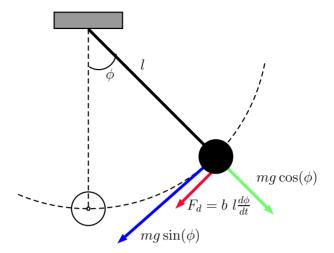


Figure 1: The force diagram of the simple damped pendulum due to air drag

4.1.1. A mathematical pendulum randomly kicked

A pendulum is a body suspended from a fixed support so that it swings freely back and forth under the influence of gravity. When a pendulum is displaced sideways from its resting, equilibrium position, it is subjected to the restoring force due to gravity that will accelerate it back toward the equilibrium position. Now, we introduce a pendulum being damped due to air drag. As depicted in Fig. 1 below, the model depends to a force $F_d = bl\frac{d\phi}{dt}$ where b is the drag coefficient for a laminar flows, m is mass of object and l is length of string. The governing equation of the simple damped pendulum due to air drag is $\ddot{\phi} + \frac{b}{m}\dot{\phi} + \frac{g}{l}\sin(\phi) = 0$.

In this study, we are interested in a damped pendulum with friction perturbed by $L_1 = \{L_1(t) : t \ge 0\}$. The governing equation is given by

$$\begin{cases} \ddot{\phi}(t) + b\dot{\phi}(t) + \frac{g}{l}\sin(\phi(t)) = \sigma_{L_1}L_1(t), \\ y(t) = \int_0^t h(\phi(s)) ds + \sigma_{L_2}L_2(t), \\ \phi(0) = 0, \ \dot{\phi}(0) = 0. \end{cases}$$

Let the state $X=(x_1,x_2)^T\in\mathbb{R}^2$ represent $(\phi,\dot{\phi})$ where ϕ is the angular displacement. The governing equation of a pendulum with friction perturbed

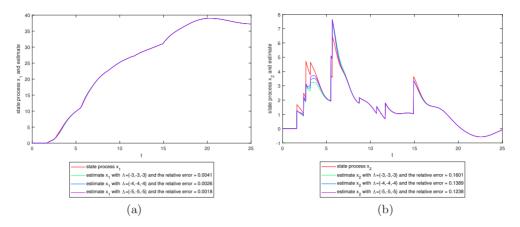


Figure 2: The state process ((A) x_1 and (B) x_2) of (39) and the estimates using Algorithm 1 with different eigenvalues (Λ).

by $L_1 = \{L_1(t) : t \ge 0\}$ is given by

$$\begin{cases}
dX(t) = \begin{pmatrix} dx_1(t) \\ dx_2(t) \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 0 & -\frac{b}{m} \end{pmatrix} \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} dt \\
+ \begin{pmatrix} 0 \\ -\frac{g}{l}\sin(x_1(t)) \end{pmatrix} dt \\
+ \begin{pmatrix} 0 \\ \sigma_{L_1} \end{pmatrix} dL_1(t), \\
y(t) = \int_0^t h(x_1(s)) ds + \sigma_{L_2} L_2(t), \\
x_1(0) = 0, x_2(0) = 0.
\end{cases} (39)$$

In our numerical experiments, we fixed the parameter $\sigma_{L_1} = \sigma_{L_2} = 1$, g = 10, l = 20, m = 10, b = 4 and for the function h we took h(x) = x. Moreover, we kicked the pendulum randomly by a compound Poisson process. The force of the kick was Lognormal distributed with mean 0.2 and standard deviation 0.9. The dependence structure of the size of the jumps is modelled by a Gumbel copula with parameter $\theta = 5$, the waiting time between the jumps was exponential distribution by the parameter $\lambda = 0.5$.

In Figure 2, we present an estimator of the state process x_1 and the state process x_2 . We simulate the model with the samples size N=70. The time step is $\tau=0.025$. To compare the impact of the feedback, we chose different eigenvalues ($\Lambda=\{\lambda_1,\lambda_2,\lambda_3\}$) of Eq. 21. Here, one can observe that if the eigenvalues are increased, then the error decreases.

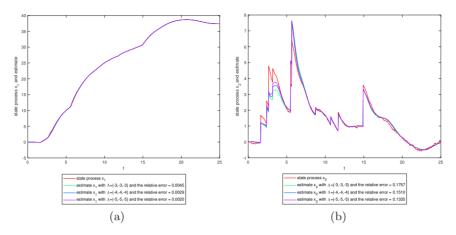


Figure 3: The state process ((A) x_1 and (B) x_2) of (39) and the estimates with some Gaussian noise to the state process using Algorithm 1 with different eigenvalues (Λ).

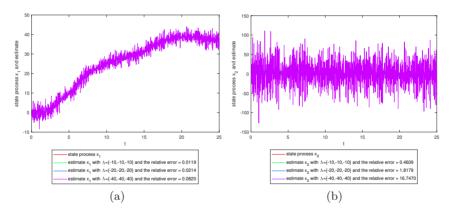


Figure 4: The state process ((A) x_1 and (B) x_2) of (39) and the estimates with some Gaussian noise to the observer using Algorithm 1 with different eigenvalues (Λ).

To observe how the process evolves with a Gaussian perturbation we added some Gaussian noise first to the observation and then to the process itself. In particular, in Figure 3 we had as governing equation $\ddot{\phi}(t) + b\dot{\phi}(t) + \frac{g}{l}\sin(\phi(t)) = \sigma_{L_1}L_1(t) + \sigma_{W_1}dW_1$ with $\sigma_{W_1} = 0.1$, in Figure 4 we had $y(t) = \int_0^t h(\phi(s)) ds + \sigma_{L_2}L_2(t) + \sigma_{W_2}dW_2$ with $\sigma_{W_2} = 0.1$. One can see, in Figure 3, Algorithm 1 works fine. If the Gaussian noise is part of the observation, the feedback estimator starts to overshoot and the quality decreases.

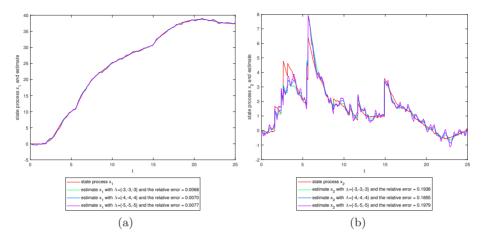


Figure 5: The state process ((A) x_1 and (B) x_2) of (40) and the estimates using Algorithm 2 with different eigenvalues (Λ).

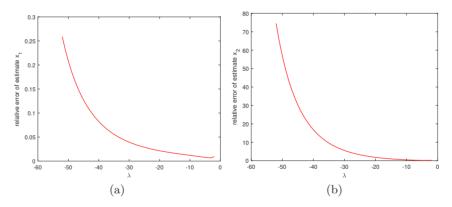


Figure 6: The estimation error of state process (40) ((A) x_1 and (B) x_2) using Algorithm 2 with the eigenvalues ($\Lambda = {\lambda, \lambda, \lambda}$).

To improve the quality of the feedback estimator by using the particle filter

described in Algorithm 2. In particular, we consider the following system

$$dX(t) = \begin{pmatrix} dx_1(t) \\ dx_2(t) \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 0 & -\frac{b}{m} \end{pmatrix} \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} dt + \begin{pmatrix} 0 \\ -\frac{g}{l} \sin(x_1(t)) \end{pmatrix} dt + \begin{pmatrix} 0 \\ \sigma_{L_1} \end{pmatrix} dL_1(t) + \begin{pmatrix} 0 \\ \sigma_{W_1} \end{pmatrix} dW_1(t),$$

$$y(t) = \int_0^t h(x_1(s)) ds + \sigma_{L_2} L_2(t) + \int_0^t \sigma_{W_2} dW_2(t),$$

$$x_1(0) = 0, x_2(0) = 0.$$

$$(40)$$

The parameter of the following simulation are given by $\sigma_{L_1} = \sigma_{L_2} = 1$, $\sigma_{W_1} = \sigma_{W_2} = 0.1$. The samples size N is 70 and a single time τ step is 0.025. To compare the impact of the feedback, we chose different eigenvalues ($\Lambda = \{\lambda_1, \lambda_2, \lambda_3\}$) of Eq. 21.

To verify the impact of the feedback on the quality of the estimator, we compare in Figure 5 the state process and its estimator with different eigenvalues (Λ). One can see that if for $\Lambda = \{-10, -10, -10\}$ the error has a minimum. If the eigenvalue is too high, the system starts to overshoot, and the estimator is unreliable. This can be observed in Figure 6.

In Lemma 1 we have seen that the variance of the feedback estimator decreases if the sample size increases. To verify this by numerical simulations, we calculated the feedback estimator varying the number of particles M. In Figure 7, we see that already with small sample size, the quality of the feedback estimator is high. Comparing the feedback estimator by an estimator without feedback, we see in Figure 8 that in order to get the same quality, the samples size has to be chosen quite high. Finally, in the last picture (Figure 9), we illustrate the impact of branching of the particles on the estimator.

4.1.2. A single-link flexible joint robot

Consider the model of a single link robot with a flexible joint rotating in a vertical plane [3]. As depicted in Fig. 10 below, the nonlinearities are due to joint flexibility, modelled as a stiffening torsional spring, and the gravitational force.

Let x_1 and x_2 are the link displacement and its velocity, respectively; x_3 and x_4 are the rotor displacement and its velocity, respectively. The governing

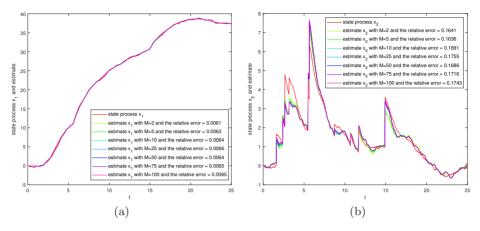


Figure 7: The state process ((A) x_1 and (B) x_2) of (40) and the estimates using Algorithm 2 with feedback variable and different number of particles (M).

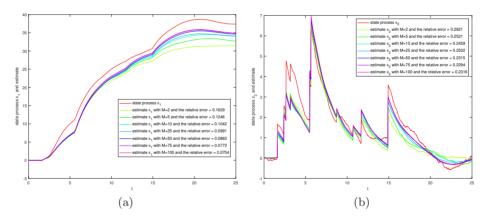


Figure 8: The state process ((A) x_1 and (B) x_2) of (40) and the estimates using Algorithm 2 without feedback variable and with different number of particles (M).

equation of a single-link flexible joint robot with friction perturbed by $L_1 = \{L_1(t) : t \geq 0\}$ is given by

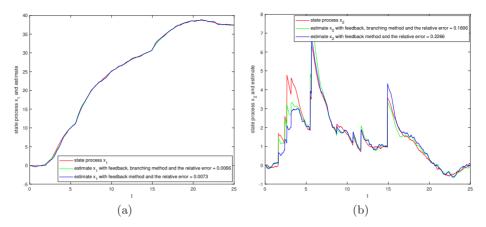


Figure 9: The state process ((A) x_1 and (B) x_2) of (40) and the estimates using Algorithm 2 with (and without) branching of particles steps.

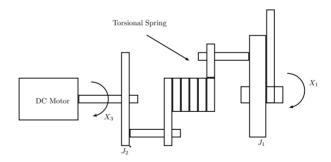


Figure 10: The flexible joint robot link

$$\begin{cases}
dX(t) = \begin{pmatrix} dx_1(t) \\ dx_2(t) \\ dx_3(t) \\ dx_4(t) \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ -\frac{k}{J_1} & 0 & \frac{k}{J_1} & 0 \\ 0 & 0 & 0 & 1 \\ \frac{k}{J_2} & 0 & -\frac{k}{J_2} & 0 \end{pmatrix} \begin{pmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \end{pmatrix} dt \\
+ \begin{pmatrix} 0 \\ -\frac{mgl}{J_1} \sin(x_1(t)) \\ 0 \\ 0 \end{pmatrix} dt \\
+ \begin{pmatrix} 0 \\ 0 \\ 0 \\ \sigma_{L_1} \end{pmatrix} dL_1(t), \\
y(t) = \int_0^t h(x_1(s)) ds + \sigma_{L_2} L_2(t), \\
X(0) = \begin{pmatrix} 1 & 1 & 1 & 1 \end{pmatrix}^T,
\end{cases}$$
(41)

where J_1 , J_2 , k, l, m and g are the link inertia, the rotor inertia, the elastic constant, the position of the center of mass, the mass, and the gravity acceleration, respectively.

In our numerical experiments, we fixed the parameter $\sigma_{L_1} = \sigma_{L_2} = 1$, g = 10, $J_1 = 30 \text{ Kg/m}^2$, $J_2 = 30 \text{ Kg/m}^2$, k = 1 N/m, l = 1 m, m = 0.5 Kg, $g = 10 \text{ m/s}^2$ and for the function h we took h(x) = x. Moreover, the dependence structure of the jumps size was modelled by a Gumbel copula with parameter $\theta = 5$, the waiting time between the jumps was exponential distributed by parameter $\lambda = 5$.

In Figure 11, we present an estimator of the state process. We simulate the model with the samples size is 70 and a single time step is 0.01. To compare the impact of the feedback, we chose different eigenvalues ($\Lambda = \{\lambda_1, \dots, \lambda_5\}$) of Eq. 21. Here, one can observe that if the eigenvalues are increased, then the error decreases.

To observe how the process evolves with a Gaussian perturbation, we added some Gaussian noise first to the observation and then to the process itself. In particular, in Figure 12 we had as governing equation

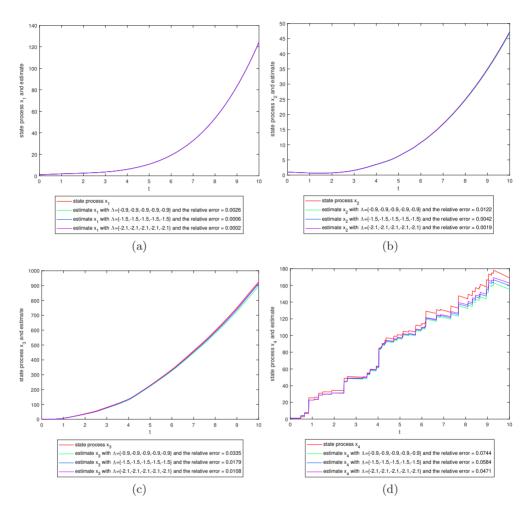


Figure 11: The state process ((A) x_1 , (B) x_2 , (C) x_3 and (D) x_4) of (41) and the estimates using Algorithm 1 with different eigenvalues (Λ).

$$dX(t) = \begin{pmatrix} dx_1(t) \\ dx_2(t) \\ dx_3(t) \\ dx_4(t) \end{pmatrix}$$

$$= \begin{pmatrix} 0 & 1 & 0 & 0 \\ -\frac{k}{J_1} & 0 & \frac{k}{J_1} & 0 \\ 0 & 0 & 0 & 1 \\ \frac{k}{J_2} & 0 & -\frac{k}{J_2} & 0 \end{pmatrix} \begin{pmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \end{pmatrix} dt$$

$$+ \begin{pmatrix} 0 \\ 0 \\ 0 \\ \sigma_{W_1} \end{pmatrix} dW_1(t)$$

$$+ \begin{pmatrix} 0 \\ 0 \\ 0 \\ \sigma_{L_1} \end{pmatrix} dL_1(t)$$

$$+ \begin{pmatrix} -\frac{mgl}{J_1} \sin(x_1(t)) \\ 0 \\ 0 \end{pmatrix} dt,$$

$$0$$

where $\sigma_{W_1} = 1$. In Figure 13 we had $y(t) = \int_0^t x_1(s) ds + \sigma_{L_2} L_2(t) + \sigma_{W_2} dW_2$ with $\sigma_{W_2} = 1$. One can see, in Figure 12, Algorithm 1 works fine. If the Gaussian noise is part of the observation, the feedback estimator starts to overshoot and the quality decreases.

To improve the quality of the feedback estimator by using the particle filter

described in Algorithm 2. In particular, we consider the following system

$$\begin{cases}
dX(t) = \begin{pmatrix} dx_1(t) \\ dx_2(t) \\ dx_3(t) \\ dx_4(t) \end{pmatrix} \\
= \begin{pmatrix} 0 & 1 & 0 & 0 \\ -\frac{k}{J_1} & 0 & \frac{k}{J_1} & 0 \\ 0 & 0 & 0 & 1 \\ \frac{k}{J_2} & 0 & -\frac{k}{J_2} & 0 \end{pmatrix} \begin{pmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \end{pmatrix} dt \\
+ \begin{pmatrix} 0 \\ 0 \\ 0 \\ \sigma_{W_1} \end{pmatrix} dW_1(t) \\
-\frac{mgl}{J_1} \sin(x_1(t)) \\ 0 \\ 0 \\ 0 \end{pmatrix} dt \\
+ \begin{pmatrix} 0 \\ -\frac{mgl}{J_1} \sin(x_1(t)) \\ 0 \\ 0 \\ 0 \\ -\frac{mgl}{J_1} \sin(x_1(t)) \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} dt \\
+ \begin{pmatrix} 0 \\ 0 \\ 0 \\ \sigma_{L_1} \end{pmatrix} dL_1(t), \\
y(t) = \int_0^t x_1(s) ds + \sigma_{L_2} L_2(t) + \int_0^t \sigma_{W_2} dW_2(t), \\
X(0) = \begin{pmatrix} 1 & 1 & 1 & 1 \end{pmatrix}^T,
\end{cases}$$

The parameter of the following simulation are given by $\sigma_{L_1} = \sigma_{L_2} = 1$, $\sigma_{W_1} = \sigma_{W_2} = 1$. The samples size is 70 and a single time step is 0.01. To compare the impact of the feedback, we chose different eigenvalues $(\Lambda = \{\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5\})$ of Eq. 21.

To verify the impact of the feedback on the quality of the estimator, we compare in Figure 14 the state process and its estimator with different eigenvalues (Λ) . One can see that $\Lambda = \{\lambda, \lambda, \lambda, \lambda\}$ with $\lambda = -0.7$ is the value with the smallest error. Again, if the eigenvalue is too high, the system starts to overshoot. Furthermore, Figure 15 shows us that there is a point of overshooting such that the result is not reliable.

In Lemma 1 we have seen that the variance of the feedback estimator decreases if the absolute value of increases. To verify this by numerical simu-

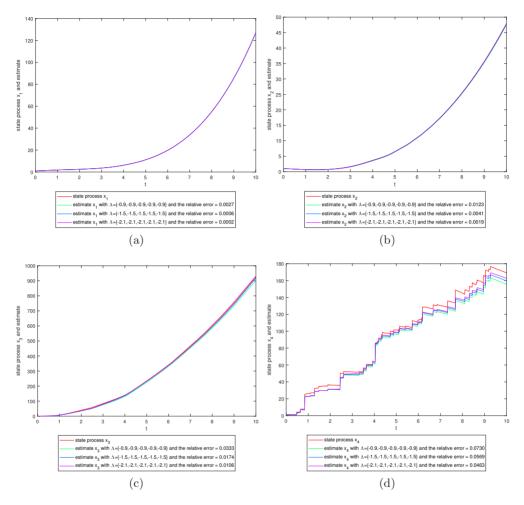


Figure 12: The state process ((A) x_1 , (B) x_2 , (C) x_3 and (D) x_4) of (41) and the estimates with some Gaussian noise to the state process using Algorithm 1 with different eigenvalues (Λ).

lations, we calculated the feedback estimator varying the sample size of the particles (M). In Figure 16, we see that already with small sample size, the quality of the feedback estimator is high. Comparing the feedback estimator by an estimator without feedback, we see in Figure 17 that in order to get the same quality, the samples size has to be chosen quite high. Finally, in the last picture (Figure 18), we illustrate how branching effects the estimator.

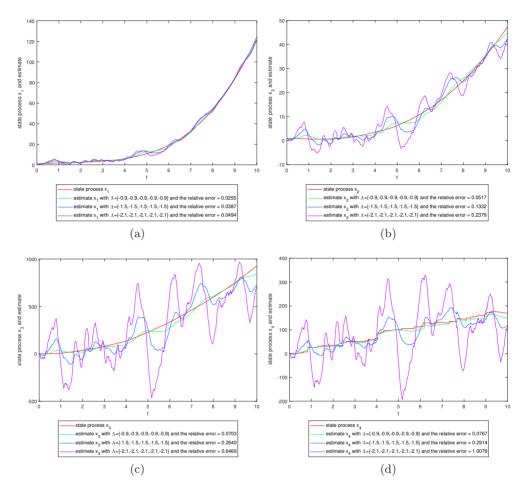


Figure 13: The state process ((A) x_1 , (B) x_2 , (C) x_3 and (D) x_4) of (41) and the estimates with some Gaussian noise to the observer using Algorithm 1 with different eigenvalues (Λ).

4.1.3. Van der Pol oscillator

The Van der Pol is a non-conservative oscillator with non-linear damping. The governing equation of the Van der Pol oscillation is $\ddot{\phi} - \mu(1 - \phi^2)\dot{\phi} + \phi = 0$, where ϕ is the voltage across the capacitor and μ is a damping indicator. Here, we are interested in the simple Van der Pol model with friction perturbed by

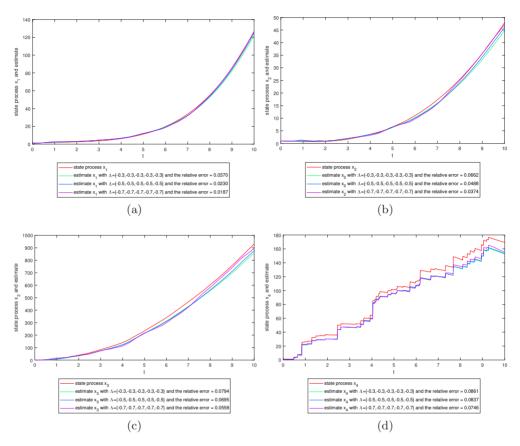


Figure 14: The state process ((A) x_1 , (B) x_2 , (C) x_3 and (D) x_4) of (42) and the estimates using Algorithm 2 with different eigenvalues (Λ).

the Levy process denoted by $L_1 = \{L_1(t) : t \ge 0\}$. The equation is given by

$$\begin{cases}
\ddot{\phi} - \mu(1 - \phi^2)\dot{\phi} + \phi &= \sigma_{L_1} L_1(t), \\
y(t) &= \int_0^t h(\phi(s)) ds + \sigma_{L_2} L_2(t), \\
\phi(0) &= \phi_0, \ \dot{\phi}(0) = \dot{\phi}_0.
\end{cases} (43)$$

In order to change the system (43) to a system of first order, let us introduce the following variables. Let us denote $(\phi, \dot{\phi})$ by $X = (x_1, x_2)^T \in \mathbb{R}^2$. The

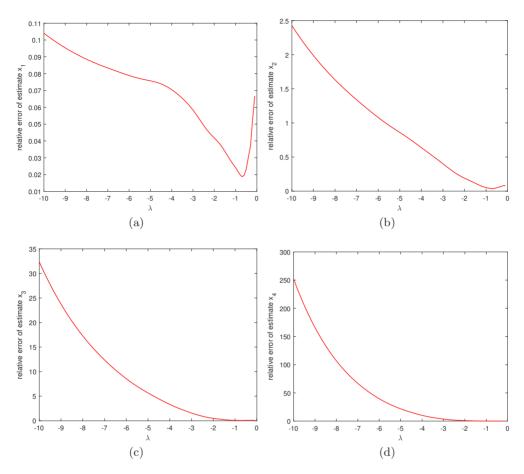


Figure 15: The estimation error of state process (42) ((A) x_1 , (B) x_2 , (C) x_3 and (D) x_4) using Algorithm 2 with the eigenvalues ($\Lambda = \{\lambda, \lambda, \lambda, \lambda, \lambda\}$).

governing equation with friction perturbed by $L_1 = \{L_1(t) : t \ge 0\}$ is given by

$$\begin{cases}
dX(t) = \begin{pmatrix} dx_1(t) \\ dx_2(t) \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} dt \\
+ \begin{pmatrix} 0 \\ -\mu(x_1(t)^2 - 1)x_2(t) \end{pmatrix} dt \\
+ \begin{pmatrix} 0 \\ \sigma_{L_1} \end{pmatrix} dL_1(t),
\end{cases}$$

$$y(t) = \int_0^t h(x_1(s)) ds + \sigma_{L_2} L_2(t),$$

$$x_1(0) = \phi_0, x_2(0) = \dot{\phi}_0.$$
(44)

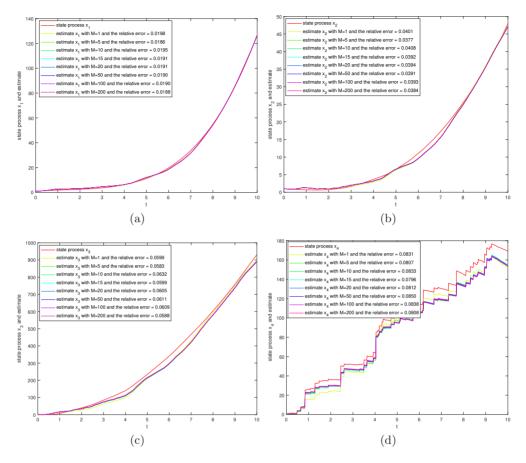


Figure 16: The state process ((A) x_1 , (B) x_2 , (C) x_3 and (D) x_4) of (42) and the estimates using Algorithm 2 with feedback variable and different number of particles M.

In our numerical experiments, we fixed the parameter $\mu = 0.4$. The initial data $\phi(0) = 0.35$, $\dot{\phi}(0) = 0$ and for the function h we took h(x) = x. Moreover, the dependence structure of the jumps size was modelled by a Gumbel copula with parameter $\theta = 5$, the waiting time between the jumps was exponential distributed by parameter $\lambda = 5$.

In Figure 19, we present an estimator of the state process. We simulate the model with the samples size is 70 and a single time step is 0.01. To compare the impact of the feedback, we chose different eigenvalues ($\Lambda = \{\lambda_1, \lambda_2, \lambda_3\}$) of Eq. 21. Here, one can observe that if $|\Lambda|$ is increased, the error decreases.

To observe how the process evolves with a Gaussian perturbation we added

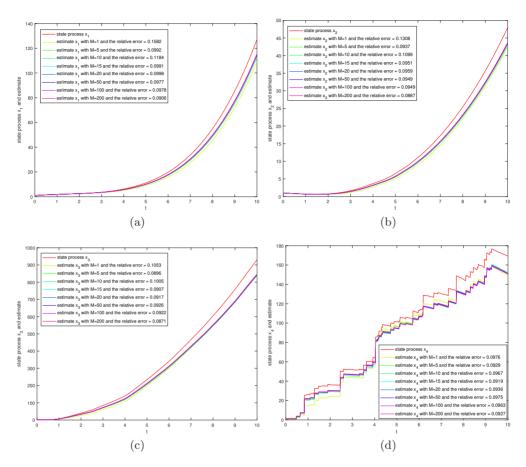


Figure 17: The state process ((A) x_1 , (B) x_2 , (C) x_3 and (D) x_4) of (42) and the estimates using Algorithm 2 without feedback variable and with different number of particles (M).

some Gaussian noise first to the observation and then to the process itself. In particular, in Figure 20 we had as governing equation $\ddot{\phi} - \mu(1 - \phi^2)\dot{\phi} + \phi = \sigma_{L_1}L_1(t) + \sigma_{W_1}dW_1$ with $\sigma_{W_1} = 0.1$, in Figure 21 we had $y(t) = \int_0^t h(\phi(s)) ds + \sigma_{L_2}L_2(t) + \sigma_{W_2}dW_2$ with $\sigma_{W_2} = 0.2$. One can see, in Figure 20, Algorithm 1 works fine. If the Gaussian noise is part of the observation, the feedback estimator starts to overshoot.

To improve the quality of the feedback estimator by using the particle filter

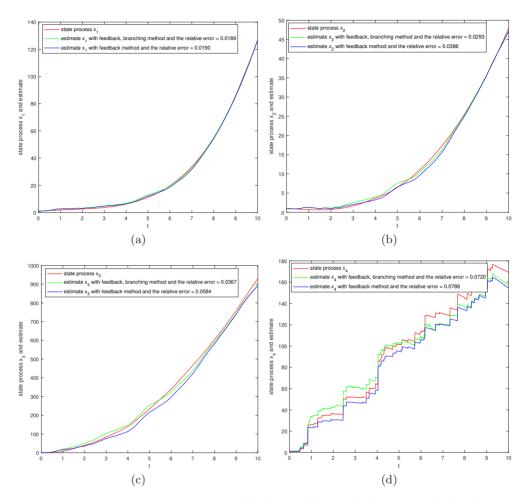


Figure 18: The state process ((A) x_1 , (B) x_2 , (C) x_3 and (D) x_4) of (42) and the estimates using Algorithm 2 with (and without) branching of particles steps.

described in Algorithm 2. In particular, we consider the following system

$$\begin{cases}
dX(t) = \begin{pmatrix} dx_1(t) \\ dx_2(t) \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} dt \\
+ \begin{pmatrix} 0 \\ -\mu(x_1(t)^2 - 1)x_2(t) \end{pmatrix} dt \\
+ \begin{pmatrix} 0 \\ \sigma_{L_1} \end{pmatrix} dL_1(t) + \begin{pmatrix} 0 \\ \sigma_{W_1} \end{pmatrix} dW_1(t), \\
y(t) = \int_0^t h(x_1(s)) ds + \sigma_{L_2} L_2(t) \\
+ \int_0^t \sigma_{W_2} dW_2(t), \\
x_1(0) = \phi_0, x_2(0) = \dot{\phi}_0.
\end{cases} (45)$$

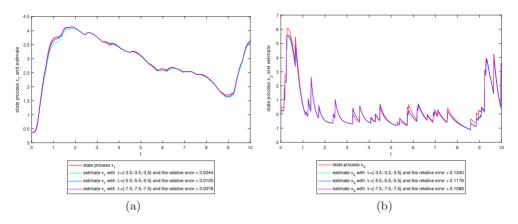


Figure 19: The state process ((A) x_1 and (B) x_2) of (44) and the estimates using Algorithm 1 with different eigenvalues (Λ).

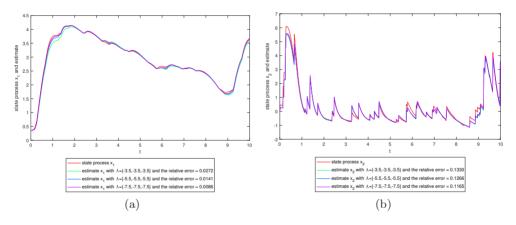


Figure 20: The state process ((A) x_1 and (B) x_2) of (44) and the estimates with some Gaussian noise to the state process using Algorithm 1 with different eigenvalues (Λ).

The parameter of the following simulation are given by $\sigma_{L_1} = \sigma_{L_2} = 0.25$, $\sigma_{W_1} = 0.1$, $\sigma_{W_2} = 0.2$, $\phi(0) = 0.35$, and $\dot{\phi}(0) = 0$. The samples size is 70, a single time step is 0.01 and for the function h we took h(x) = x. To compare the impact of the feedback, we chose different eigenvalues $(\Lambda = \{\lambda_1, \lambda_2, \lambda_3\})$ of Eq. 21.

To verify the impact of the feedback on the quality of the estimator, we compare in Figure 22 the state process and its estimator with different eigenvalues (Λ). One can see that $\Lambda = \{-3, -3, -3\}$ is the eigenvalue with the

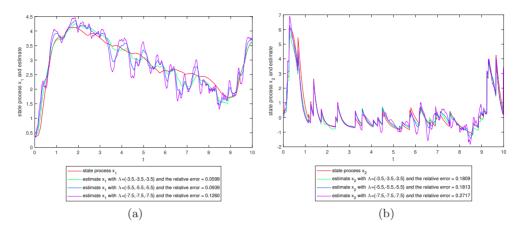


Figure 21: The state process ((A) x_1 and (B) x_2) of (44) and the estimates with some Gaussian noise to the observer using Algorithm 1 with different eigenvalues (Λ).

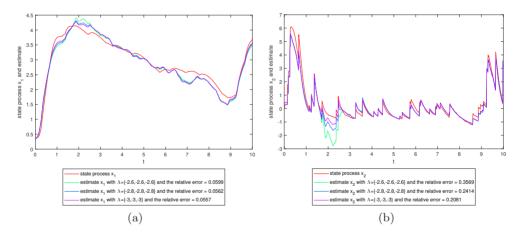


Figure 22: The state process ((A) x_1 and (B) x_2) of (44) and the estimates using Algorithm 2 with different eigenvalues (Λ).

smallest error. If the eigenvalue is too high, the system starts to overshoot, and the estimator is unreliable. Furthermore, Figure 23 shows us that there is a point of overshooting.

In Figure 24, we see that already with small sample size, the quality of the feedback estimator is high. Again, comparing the feedback estimator by an estimator without feedback, we see in Figure 25 that in order to get the same quality, the samples size has to be chosen quite high. Finally, in the last picture

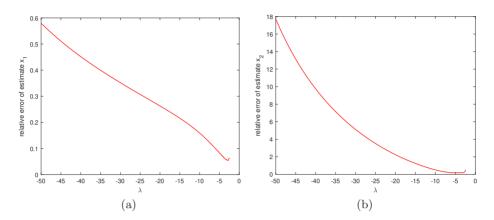


Figure 23: The estimation error of state process (44) ((A) x_1 and (B) x_2) using Algorithm 2 with the eigenvalues ($\Lambda = {\lambda, \lambda, \lambda}$).

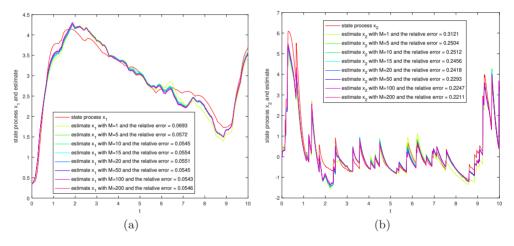


Figure 24: The state process ((A) x_1 and (B) x_2) of (44) and the estimates using Algorithm 2 with feedback variable and different number of particles (M).

(Figure 26), we illustrate how branching effects the estimator.

5. Conclusion

As one can see from our examples that the feedback estimator provides better performance than using only probabilistic methods by itself. First of all, one

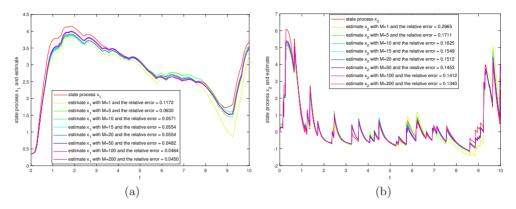


Figure 25: The state process ((A) x_1 and (B) x_2) of (44) and the estimates using Algorithm 2 without feedback variable and with different number of particles (M).

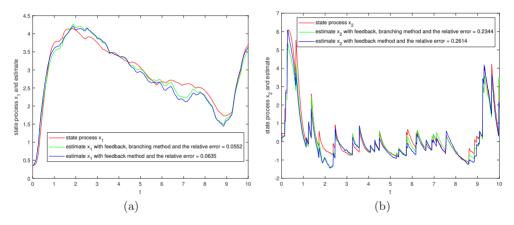


Figure 26: The state process ((A) x_1 and (B) x_2) of (44) and the estimates using Algorithm 2 with (and without) branching of particles steps.

can use a smaller sample to get the same quality. The reason is that the variance of the feedback estimator is smaller than the variance of the estimator without any feedback. However, one can see at the examples, that, if the feedback is too strong, the systems start to overshoots the quality of the estimator gets unreliable. Therefore the feedback has to be chosen carefully.

Acknowledgments

We would like to mention that Pani Fernando and Kistosil Fahim were supported partially by the Austrian Science Foundation (FWF), project number P23591-N12. Kistosil Fahim was also partially supported by the Austrian Agency for International Cooperation in Education and Research (OeAD), Centre for International Cooperation and Mobility (ICM) with reference number ICM-2019-13458.

A. Copulas - a short overview

Copulas are popular to describe the dependence structure of non–Gaussian random variables. One of the main advantages is that one can model and estimate the distribution of random vectors by estimating marginals and copulas separately. In this way, a copula can be seen as a multivariate cumulative distribution function for which the marginal probability distribution of each variable is uniform. For an introduction to copulas we refer to the books [9, 60, 23].

Definition 2. A 2-dimensional copula is a function

$$C: [0,1]^d \to [0,1]$$

such that

- $\forall u \in [0,1], C(u,1) = C(1,u) = u,$
- $\forall u_i \in [0,1], C(u_1,u_2) = 0$ if at least one of the u_i 's equals zero,
- C is grounded and n-increasing, i.e., the C-volume of every box whose vertices lie in $[0,1]^2$ is positive.

In order to explain shortly the concept of copula, let (X_1, X_2) be a couple of two d-dimensional random vectors. Let us denote the distribution of X_i by F_i , i = 1, 2. Assuming that F_1 and F_2 are absolutely continuous, the random variables $F_1(X_1)$ and $F_2(X_2)$ are uniformly distributed on [0, 1]. Besides,

$$(F(X_1), F(X_2))$$

has uniform distributed margins. Now, for any random vector (X_1, X_2) continuous marginales distribution functions F_i , the function

$$C:[0,1]^2\to [0,1]$$

such that

$$C(F(x_1), F(x_2)) = \mathbb{P}(X_1 \le x_1, X_2 \le x_2),$$

is called copula. Here, $x \leq y$ for two vectors $x = (x_1, \ldots, x_d)$ and $y = (y_1, \ldots, y_d)$ means $x_1 \leq y_1, \ldots, x_d \leq y_d$. Since we need here only copulas for two random variable, we restrict ourselves here to two dimensional copulas. By Sklar's Theorem one can find for any multivariate distribution a copula C with

$$F(x_1, x_2) = C(F_1(x_1), F_2(x_2)), \quad \forall (x_1, x_2) \in \mathbb{R}^2.$$

Remark 3. From this formula one can easily get the conditional density, given by

$$\mathbb{P}(X_1 = x_1 \mid X_2 = x_2) = \frac{\mathbb{P}(X_1 = x_1, X_2 = x_2)}{\mathbb{P}(X_2 = x_2)}$$
(46)

$$\frac{c(F_1^{-1}(x_1), F_2^{-1}(x_2))}{c(1, F_2^{-1}(x_2))} f_1(x_1), \quad x_1, x_2 \in \mathbb{R}^d.$$
(47)

where

$$c(u_1, u_2) := \frac{d^2}{du_1 du_2} C(u_1, u_2)$$

and f_1 denotes the density of F_1 .

Different kind of copulas are e.g. the independence copula for pairs defined by

$$C(u_1, u_2) = u_1 u_2, \quad u_1, u_2 \in [0, 1],$$

the comonotonicity copula defined by

$$C(u_1, u_2) = \min(u_1, u_2), \quad u_1, u_2 \in [0, 1],$$

the Gumbel copula defined for $\theta \in [1, \infty)$ by

$$C(u_1, u_2) = \exp\left(-\left[(-\ln u_1)^{\theta} + (-\ln u_2)^{\theta}\right]^{\frac{1}{\theta}}\right), \quad u_1, u_2 \in [0, 1],$$

and the Clayton copula defined for $\theta \in (\theta, \infty)$ by

$$C(u_1, u_2) = \max \left([u_1^{-\theta} + u_2^{-\theta} - 1]^{-\frac{1}{\theta}}, 0 \right), \quad u_1, u_2 \in [0, 1].$$

The parameter $\theta>0$ determines the dependence of the jump sizes. Larger values of θ indicate a stronger dependence, smaller values of θ indicate independence. In particular, if $\theta=1$ one obtains for the Gumbel copula the independence copula, and for $\theta\to\infty$ converges the Gumbel copula to the comonotonic copula. In the limit $\theta\to 0$ the Clayton copula converges to the independence copula, and for $\theta\to\infty$ the Clayton copula converges to the comonotonic copula. The Gumbel copula turns out to have upper tail dependence. This significate, roughly spoken, if X_1 and X_2 are defined by its marginal distribution and a Gumbel copula, there is a strong tendency for X_2 to be extreme, if X_1 is extreme and vice versa. The Clayton copula turns out to have a lower tail dependence.

A.1. Simulation of Copulas

Given a two-dimensional copula C and a uniformly distributed random variable U_1 , then one can easily simulate a random variable U_2 , such that

$$\mathbb{P}(U_2 \le u_1 \mid U_2 \le u_2) = C(u_1, u_2), \quad u_1, u_2 \in [0, 1],$$

by acceptation–rejection method (see e.g. book [46]). To illustrate another way, let us define the function G given by

$$G_{\tilde{u}}(u) := \frac{d}{du_2} C(u, u_2) \big|_{u_2 = \tilde{u}}, \quad u, \tilde{u} \in [0, 1].$$

If G is invertible, generate a on [0,1] uniform distributed random variable U_2 and put $Y := G_{U_2}^{-1}(U_1)$. For further details on simulating copulas, we refer to [23].

References

- [1] V. Agarwal and H. Parthasarathy, Disturbance estimator as a state observer with extended Kalman filter for robotic manipulator, *Nonlinear Dynamics*, **85** (2016), 2809-2825.
- [2] H. Ahn and R. Feldman, Optimal filtering of a Gaussian signal in the presence of Lévy noise, SIAM J. Appl. Math., **60** (2000), 359–369.
- [3] A. Alessandri, Design of observers for Lipschitz nonlinear systems using LMI, *IFAC Nonlinear Control Systems*, **37** (2004), 459–464.

- [4] C. Andrieu, A. Doucet, S. Singh, and V. Tadic, Particle methods for change detection, system identification, and control, In: *Proceedings of the IEEE*, **92** (2004), 42-438.
- [5] K. J. Åström and R. M. Murray, *Feedback Systems*, Princeton University Press, Princeton (2008).
- [6] A. Bain and D. Crisan, Fundamentals of Stochastic Filtering, Stochastic Modelling and Applied Probability, 60, Springer, New York (2009).
- [7] A. Bensoussan, Nonlinear Filtering Theory, in Recent Advances in Stochastic Calculus (College Park, MD, 1987), Progr. Automat. Info. Systems, Springer, New York (1990).
- [8] C. Ceci and K. Colaneri, Nonlinear Filtering for jump diffusion processes with a financial application, *Advances in Appl. Prob.*, 44 (2012), 603–906.
- [9] U. Cherubini, E. Luciano, and W. Vecchiato, *Copula Methods in Finance*, John Wiley & Sons Ltd (2004).
- [10] C. Cont and P. Tankov, Financial Modelling with Jump Processes, Chapmann & Hall (2004).
- [11] D. Coufal, On convergence of kernel density estimates in particle filtering, *Kybernetika*, **52** (2016), 735-756.
- [12] D. Creal, Analysis of filtering and smoothing algorithms for Lévy-driven stochastic volatility models, *Comput. Statist. Data Anal.*, **52** (2008), 2863–2876.
- [13] D. Crisan, Exact rates of convergence for branching particle approximation to the solution of the Zakai equation, *Anal. of Prob.*, **31** (2003), 693–718.
- [14] D. Crisan and A. Doucet, A survey of convergence results on particle filtering methods for practitioners, *IEEE Transactions on Signal Processing*, **50** (2002), 736-746.
- [15] D. Crisan and B. Rozovskii, *The Oxford Handbook of Nonlinear Filtering*, Oxford University Press, Oxford (2011).
- [16] P. Del Moral, Measure valued processes and interacting particle systems: Applications to nonlinear filtering problems, Anal. of Appl. Probab., 8 (1998), 438–495.

- [17] P. Del Moral, Nonlinear filtering: Interacting particle solution, *Markov Processes Relat. Fields*, **2** (1996), 555–580.
- [18] A. Doucet, N. de Freitas, and N. Gordon, Sequential Monte Carlo Methods in Practice, Springer-Verlag, New York (2001).
- [19] J. C. Doyle, B. A. Francis, and A. R. Tannenbaum, Feedback Control Theory, Macmillan Publishing Company, New York (1992).
- [20] K. Eilertson, J. Fricks, and M. Ferrari, Estimation and prediction for a mechanistic model of measles transmission using particle filtering and maximum likelihood estimation, Stat. Med., 38 (2019), 4146-4158.
- [21] R. J. Elliott and W. P. Malcolm, A general smoothing equation for Poisson observations, In: *IEEE Conf. Decision and Control*, 4 (1999), 4106–4110, Phoenix, AZ, USA.
- [22] R. J. Elliott and W. P. Malcolm, General smoothing formulas for Markov-modulated Poisson observations, *IEEE Trans. Automat. Control.*, **50** (2005), 1123–1134.
- [23] B. P. W. Fernando, and E. Hausenblas, Nonlinear filtering with correlated Lévy noise characterized by copulas, *Braz. J. Probab. Stat.*, 32 (2018), 374-421.
- [24] B. P. W. Fernando and S. S. Sritharan, Nonlinear filtering of stochastic Navier-Stokes equation with Itô-Lévy noise, Stoch. Anal. Appl., 31 (2013), 381–426.
- [25] N. Gordon, D. Salmond, and A. Smith, Novel approach to nonlinear/non-Gaussian Bayesian state estimation, In: *IEEE Proceedings-F*, 140 (1993), 107-113.
- [26] Y. Iba, Population Monte Carlo algorithms, Transactions of the Japanese Society for Artificial Intelligence, 16 (2001), 279-286.
- [27] M. Isaard and A. Blake, Contour tracking by stochastic propagation of conditional density, In: European Conference on Computer Vision, (1996), Cambridge, 343-356.
- [28] A. Jasra, F. Yu, and J. Heng, Multilevel particle filters for the non-linear filtering problem in continuous time, *Stat. Comput.*, **30** (2020), 1381-1402.

- [29] G. Kallianpur and C. Striebel, Estimation of stochastic systems, Arbitrary system process with additive white noise observation errors. Annals of Mathematical Statistics, 39 (1968), 785–801.
- [30] R. E. Kalman, A new approach to linear filtering and prediction problems, Transactions of the ASME-Journal of Basic Engineering, 82 (1960), 35–45.
- [31] R. E. Kalman and R. S. Bucy, New results in linear filtering and prediction theory, *Trans. ASME Ser. D. J. Basic Engrg.*, **83** (1961), 95–108.
- [32] K. Kanazawa, D. Koller, and S. Russell, Stochastic simulation algorithms for dynamics probabilistic networks, In: *Eleventh Annual Conference on Uncertainty in AI* (1995), 346-351.
- [33] G. Kitagawa, Monte Carlo filter and smoother for non-Gaussian nonlinear state space models, *Journal of Computational and Graphical Statistics*, **5** (1996), 1-25.
- [34] A. Kolmogoroff: Sur l'interpolation et extrapolation des suites stationnaires. Comptes Rendus de Academic des Science, 208 (1939), 2043–2045.
- [35] A. Kolmogoroff, Interpolation und Extrapolation von stationären zufälligen Folgen, Bull. Acad. Sci. URSS Sér. Math. [Izvestia Akad. Nauk. SSSR], 5 (1941), 3–14.
- [36] N. Krylov and B. Rozovskiĭ, The Cauchy problem for linear stochastic partial differential equations, *Izv. Akad. Nauk SSSR Ser. Mat.*, 41 (1977), 1329–1347.
- [37] N. Krylov and B. Rozovskiĭ, Conditional distributions of diffusion processes, Izv. Akad. Nauk SSSR Ser. Mat., 42 (1978), 356–378.
- [38] N. Krylov and B. Rozovskii, Stochastic partial differential equations and diffusion processes, *Uspekhi Mat. Nauk.*, **37** (1982), 75–95.
- [39] H. J. Kushner, On the differential equations satisfied by conditional probability densities of Markov processes, with applications, *J. Soc. Indust. Appl. Math. Ser. A Control.*, **2** (1964), 106–119.
- [40] H. J. Kushner, Dynamical equations for optimal nonlinear filtering, *J. Differential Equations*, **3** (1967), 179–190.

- [41] W. Li, Z. Wang, Y. Yuan, and L. Guo, Two-stage particle filtering for non-Gaussian state estimation with fading measurements, *Automatica J. IFAC*, 115 (2020), 108882.
- [42] D. G. Luenberger, Observing the state of a linear system, *IEEE Transaction on Military Electronics*, 8 (1964), 74-80.
- [43] J. MacCormick and A. Blake, A probabilistic exclusion principle for tracking multiple objects, In: *International Conference on Computer Vision* (1999), 572-578.
- [44] N. Metropolis and S. Ulam, The Monte Carlo method, *Journal of the American Statistical Association*, 44 (1949), 335-341.
- [45] P. Moral, Measure valued processes and interacting particle systems: Application to non-linear filtering problems, *Annals of Applied Probability*, 8 (1998), 438-495.
- [46] T. Müller-Gronbach, E. Novak and K. Ritter, *Monte Carlo algorithms*, Springer-Lehrbuch, Berlin: Springer (2012).
- [47] B. N. Oreshkin, X. Liu, and M. Coates, J. Efficient delay-tolerant particle filtering, *IEEE Trans. Signal Process*, **59** (2011), 3369-3381.
- [48] E. Pardoux, Stochastic partial differential equations and filtering of diffusion processes, *Stochastics*, **3** (1979), 127–167.
- [49] E. Pardoux: Filtrage non linéaire et équations aux dérivées partielles stochastiques associées. In: École d'Été de Probabilités de Saint-Flour XIX—1989, volume 1464 of Lecture Notes in Math. (1991), 67–163. Springer, Berlin.
- [50] L. Pavelková, Nonlinear Bayesian state filtering with missing measurements and bounded noise and its application to vehicle position estimation, *Kybernetika*, 47 (2011), 370-384.
- [51] S. Popa and S. S. Sritharan, Nonlinear filtering of Itô-Lévy stochastic differential equations with continuous observations, Commun. Stoch. Anal., 3 (2009), 313-330.
- [52] M. Prokešová and V. Beneš, Nonlinear filtering in spatio-temporal doubly stochastic point processes driven by OU processes, Kybernetika, 42 (2006), 539-556.

- [53] B. Ristic, S. Arulampalam, and N. Gordon, Beyond the Kalman Filter: Particle Filters for Tracking Applications, Artech House, Norwell, Massachusetts, 2004.
- [54] L. Sheng, Y. Niu, M. Gao, and D. Zhou, Polynomial filtering for nonlinear stochastic systems with state- and disturbance-dependent noises, *Internat.* J. Robust Nonlinear Control, 30 (2020), 4726-4743.
- [55] D. Simon, Optimal State Estimation: Kalman, H_{∞} and Nonlinear Approaches, Wiley, New York (2006).
- [56] W. Song, Z. Wang, J. Wang, and J. Shan, Particle filtering for a class of cyber-physical systems under Round-Robin protocol subject to randomly occurring deception attacks, *Inform. Sci.*, 544 (2021), 298-307.
- [57] D. Sornette and K. Ide, The Kalman-Lévy filter, Phys. D., 151 (2001), no. 2-4, 142-174.
- [58] R. L. Stratonovič, On the theory of optimal nonlinear filtering of random functions, *Teor. Verojatnost. i Primenen.*, 4 (1959), 239–242.
- [59] R. L. Stratonovič, Conditional Markov processes, *Teor. Verojatnost. i Primenen.*, **5** (1960), 172–195.
- [60] P. Trivedi and D. Zimmer, Copula Modeling: An Introduction for Practitioners, now Publishers Inc., Hanover, MA 02339 USA (2005).
- [61] X. Wang, L. Li, and W. Xie, A novel T-S fuzzy particle filtering algorithm based on fuzzy C-regression clustering, *Internat. J. Approx. Reason.*, 117 (2020), 81-95.
- [62] L. Wen, J. Wu, L. Lu, and J. Li, A Defensive Marginal Particle Filtering Method for Data Assimilation, SIAM/ASA J. Uncertain. Quantif., 8 (2020), 1215-1235.
- [63] N. Wiener, Extrapolation, Interpolation, and Smoothing of Stationary Time Series. With Engineering Applications, The Technology Press of the Massachusetts Institute of Technology, Cambridge, Mass (1949).
- [64] N. Wiener, I am a Mathematician, MIT Press, Cambridge, Massachusetts (1956).

- [65] W. M. Wonham, Some applications of stochastic differential equations to optimal nonlinear filtering, J. Soc. Indust. Appl. Math. Ser. A Control., 2 (1965), 347–369.
- [66] W. Wu, G. Wang, S. Ding, A. Song, and L. Yu, Repetitive accuracy degradation of numerical control rotary table based on hidden Markov model and improved particle filtering, J. Vib. Control, 26 (2020), 1399-1410.
- [67] J. Xiong, An Introduction to Stochastic Filtering Theory, Oxford Graduate Texts in Mathematics, 18. Oxford University Press, Oxford (2008).
- [68] W. Xia, M. Sun, and Q. Wang, Direct target tracking by distributed Gaussian particle filtering for heterogeneous networks, *IEEE Trans. Signal Process.*, **68** (2020), 1361-1373.
- [69] J. Zabczyk, Mathematical Control Theory, Modern Birkhäuser Classics. Birkhäuser Boston, Inc., Boston, MA. An Introduction, Reprint of the 1995 Ed. (2008).
- [70] M. Zakai, On the optimal filtering of diffusion processes, Z. Wahrscheinlichkeitstheorie und Verw. Gebiete., 11 (1969), 230–243.
- [71] P. Zhu, G. Wei, and J. Li, On hybrid consensus-based extended Kalman filtering with random link failures over sensor networks, *Kybernetika*, **56** (2020), 189-212.
- [72] S.H. Zak, *Systems and Control*, Oxford series in electrical and computer engineering, Oxford University Press (2003).