mathematics

MDPI

Article

On Filtering and Smoothing Algorithms for Linear State-Space Models Having Quantized Output Data

Angel L. Cedeño [1,2,*], Rodrigo A. González [3], Boris I. Godoy [4], Rodrigo Carvajal [5] and Juan C. Agüero [1,2]

[1] Electronics Engineering Department, Universidad Técnica Federico Santa María, Av. España 1680, Valparaíso 2390123, Chile
[2] Advanced Center for Electrical and Electronic Engineering, AC3E, Gral. Bari 699, Valparaíso 2390136, Chile
[3] Department of Mechanical Engineering, Eindhoven University of Technology, 5612 AZ Eindhoven, The Netherlands
[4] Department of Automatic Control, Lund University, 221 00 Lund, Sweden
[5] School of Electrical Engineering, Pontificia Universidad Católica de Valparaíso, Av. Brasil 2147, Valparaíso 2374631, Chile
* Correspondence: angel.cedeno@sansano.usm.cl

Abstract: The problem of state estimation of a linear, dynamical state-space system where the output is subject to quantization is challenging and important in different areas of research, such as control systems, communications, and power systems. There are a number of methods and algorithms to deal with this state estimation problem. However, there is no consensus in the control and estimation community on (1) which methods are more suitable for a particular application and why, and (2) how these methods compare in terms of accuracy, computational cost, and user friendliness. In this paper, we provide a comprehensive overview of the state-of-the-art algorithms to deal with state estimation subject to quantized measurements, and an exhaustive comparison among them. The comparison analysis is performed in terms of the accuracy of the state estimation, dimensionality issues, hyperparameter selection, user friendliness, and computational cost. We consider classical approaches and a new development in the literature to obtain the filtering and smoothing distributions of the state conditioned to quantized data. The classical approaches include the extended Kalman filter/smoother, the quantized Kalman filter/smoother, the unscented Kalman filter/smoother, and the sequential Monte Carlo sampling method, also called particle filter/smoother, with its most relevant variants. We also consider a new approach based on the Gaussian sum filter/smoother. Extensive numerical simulations—including a practical application—are presented in order to analyze the accuracy of the state estimation and the computational cost.

Keywords: extended Kalman filter/smoother; unscented Kalman filter/smoother; Gaussian sum filter/smoother; particle filter/smoother; state estimation; quantized data

MSC: 93E11

Citation: Cedeño, A.L.; González, R.A.; Godoy, B.I.; Carvajal, R.; Agüero, J.C. On Filtering and Smoothing Algorithms for Linear State-Space Models Having Quantized Output Data. *Mathematics* **2023**, *11*, 1327. https://doi.org/10.3390/math11061327

Academic Editor: Ivo Petráš

Received: 30 January 2023
Revised: 6 March 2023
Accepted: 7 March 2023
Published: 9 March 2023

1. Introduction

In the last two decades, there has been a growing number of applications for sensors, networks, and sensor networks, where common problems include dealing with the loss of information in signals measured with low-resolution sensors, or storing and/or transmitting a reduced representation of such signals—in order to minimize the resource consumption in a communication channel [1]. This kind of problem encompasses a nonlinear process called *quantization*, which divides the input signal space into a finite (or infinite but countable) number of intervals and each of them being represented by a single output value [2]. Applications using quantized data include networked control [3,4], fault detection [3,5–7], cyber-physical systems [8,9], multi-target tracking [10], and system identification [11–14], just to mention a few. In these applications, a key element is the state

estimation of a dynamical system conditioned upon the available quantized observations. For instance, ref [15] deals with the problem of state estimation and control of a microgrid incorporating multiple distributed energy resources, where the state estimation and control are based on uniform quantized observations that are transmitted through a wireless channel.

For linear systems subject to additive Gaussian noise, expressions for the filtering and smoothing probability density functions (filtering and smoothing PDFs) that give estimators with optimal mean-square-error properties can be obtained from the Kalman Filter and the Rauch–Tung–Striebel smoother (KS), respectively, [16]. However, for general nonlinear systems subject to non-Gaussian additive noise, it is difficult (or even impossible in most cases) to obtain closed-form expressions for these PDFs and, therefore, expressions for a state estimator, given input/output data. Many sub-optimal methods have been developed in order to obtain an approximation of the desired PDFs and an estimate of the state vector; see, e.g., [17–19]. For example, the extended Kalman filter (EKF) was studied in [20] to deal with quantized data. This approach is difficult to implement since the quantizer is a non-differentiable nonlinear function and requires the computation of a Jacobian matrix. The authors in [20] proposed to approximate the quantizer by a smooth function to compute an approximation of the Jacobian matrix. However, since the quantization function is highly nonlinear, the EKF/EKS approach typically produces inaccurate estimates of the system state. The Kalman filter was modified to include the quantization effect in the computation of the filtering state, which has been referred to as the quantized Kalman filter (QKF) [21,22]. The unscented Kalman filter (UKF) was applied by [23] to deal with quantized innovation systems in a wireless sensor network environment. The UKF is based on the Unscented transformation [18] that represents the mean and covariance of a Gaussian random variable through a reduced number of points. Then, these points are propagated through the nonlinear function to accurately capture the mean and covariance of the propagated Gaussian random variable. An advantage of the UKF lies in its high estimation accuracy and convergence rate, together with its simplicity of implementation compared with EKF, since it avoids the computation of the Jacobian matrix required in the EKF method.

One of the most used methods in nonlinear filtering is the Sequential Monte Carlo sampling approach called particle filtering (PF)[24]. The PF uses a set of weights and samples to create an approximation of the filtering and smoothing PDFs. The main advantages of the combined PF and particle smoothing (PS) are the implementation and ability to deal with nonlinear and non-Gaussian systems. Nevertheless, the PF also has drawbacks. One of them is the degeneracy problem, where most weights—calculated at one iteration of the PF approach—go to zero [25]. To get around this issue, [24] proposed an approach called resampling. Here, the heavily weighted particles are replicated sometimes, and the rest of the particles are discarded. A number of resampling techniques have been developed in the literature, such as systematic (SYS), multinomial (ML), Metropolis (MT), and local selection (LS) resampling methods [26]. These resampling techniques have advantages such as unbiasedness and parallelism capacity. Naturally, the performance of the particle filter depends upon the implemented resampling method [27]. Unfortunately, the resampling process produces a loss of diversity in the particle set since the particles with high weights are replicated. This problem is called sample impoverishment [25], and to mitigate it, a Markov chain Monte Carlo (MCMC) move is usually introduced after the resampling step to provide diversity to the samples so that the new particle set is still distributed according to the posterior PDF [28]. There are mainly two MCMC methods that we can use to deal with the impoverishment problem: the Gibbs and the Metropolis–Hasting (MH) sampling. Here, we discuss the MH algorithm and a special MH algorithm called random walk Metropolis (RWM [29,30]) in conjunction with the aforementioned resampling methods.

In addition to the methods detailed above, a new algorithm to deal with quantized output data is proposed in [31,32]. In these works, the authors defined the probability mass function of the quantized data conditioned upon the system state as an integral

whose limits depend on the quantizer regions. This integral is approximated by using Gauss–Legendre quadrature [33], yielding a model with a Gaussian sum structure. Such a model is later used to develop a Gaussian sum—filter GSF and smoother GSS—to obtain closed-form filtering and smoothing distributions for systems with quantized output data.

Despite the wide availability of the commonly used and also novel state-estimation methods described above, there is no consensus among the control and estimation community on (1) which methods are more suitable for a particular application and why, and (2) how these methods compare in terms of accuracy, computational cost, and user friendliness. This paper provides a comprehensive overview of the state-of-the-art algorithms to deal with the problem of state estimation of linear dynamical systems using quantized observations. This work aims to serve both as an introduction to the EKF/EKS, QKF/QKS, UKF/UKS, GSF/GSS, and PF/PS algorithms for estimation using quantized output data, and to provide clear guidelines on the advantages and shortcomings of each algorithm based on the accuracy of the state estimation, dimensionality issues, hyperparameter selection, user friendliness, and computational cost.

The organization of this paper is as follows: In Section 2, we define the problem of state estimation with quantized output data. In Section 3, we present a comprehensive review of the most effective filtering and smoothing methods that are available in the literature. In Section 4, a numerical example to show the traits of each method is presented, and in Section 5 a practical application is used for testing the algorithms. In Section 6, general user guidelines are provided. Finally, concluding remarks are given in Section 7.

2. Statement of the Problem

This paper considers the filtering and smoothing problem for the following discrete-time, LTI state-space system with quantized output (see Figure 1):

$$\mathbf{x}_{t+1} = \mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{u}_t + \mathbf{w}_t, \tag{1}$$

$$z_t = \mathbf{C}\mathbf{x}_t + \mathbf{D}\mathbf{u}_t + v_t, \tag{2}$$

$$y_t = \mathfrak{q}\{z_t\}, \tag{3}$$

where $\mathbf{x}_t \in \mathbb{R}^n$ is the state vector, $\mathbf{u}_t \in \mathbb{R}^m$ is the input of the system, $z_t \in \mathbb{R}$ is the non-quantized output, and $y_t \in \mathbb{R}$ is the quantized output. The matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\mathbf{B} \in \mathbb{R}^{n\times m}$, $\mathbf{C} \in \mathbb{R}^{1\times n}$, and $\mathbf{D} \in \mathbb{R}^{1\times m}$. The nonlinear map $\mathfrak{q}\{\cdot\}$ is the quantizer. The state noise $\mathbf{w}_t \in \mathbb{R}^n$ and the output noise $v_t \in \mathbb{R}$ are zero-mean white Gaussian noises with covariance matrix $\mathbf{Q}$ and R, respectively. Due to the random components (i.e., the noises $\mathbf{w}_t$ and v_t) in (1) and (2), the state-space model can be described using the state transition PDF $p(\mathbf{x}_{t+1}|\mathbf{x}_t) \sim \mathcal{N}(\mathbf{x}_{t+1}; \mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{u}_t, \mathbf{Q})$ and the non-quantized output PDF $p(z_t|\mathbf{x}_t) \sim \mathcal{N}(z_t; \mathbf{C}\mathbf{x}_t + \mathbf{D}\mathbf{u}_t, R)$ with $\mathbf{x}_1 \sim \mathcal{N}(\mathbf{x}_1; \boldsymbol{\mu}_1, \mathbf{P}_1)$, where $\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \mathbf{P})$ represents a PDF corresponding to a Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\mathbf{P}$ of the random variable $\mathbf{x}$. The initial condition $\mathbf{x}_1$, the model noise $\mathbf{w}_t$, and the measurement noise v_t are statistically independent random variables.

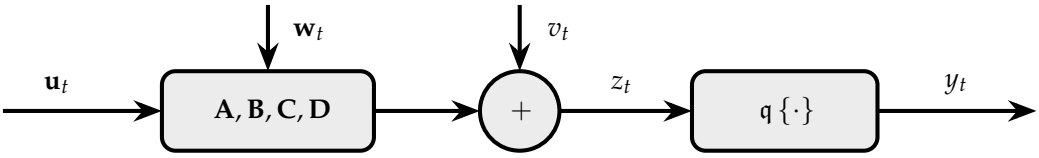

Figure 1. State-space model with quantized output.

On the other hand, the nonlinear map $\mathfrak{q}\{\cdot\} : \mathbb{R} \to \Psi$ is the quantizer, where $\Psi \subset \mathbb{R}$ is the output set. More precisely, $\mathfrak{q}\{\cdot\}$ is given by [1]:

$$y_t = \mathfrak{q}\{z_t\} = \psi_k \quad \text{if} \quad z_t \in \mathcal{R}_k, \quad k \in \mathcal{K}, \tag{4}$$

where ψ_k is the kth output value in the output set Ψ, $\mathcal{R}_k$ is the kth interval mapped to the value ψ_k, and the indices set $\mathcal{K}$ defines the number of quantization levels of the output set Ψ. Here we consider two types of quantizers: (i) an infinite-level quantizer (ILQ), in which its output has infinite (but countable) levels of quantization with

$$\mathcal{K} = \{\dots, 1, 2, \dots, L, \dots\}. \tag{5}$$

Here, $\mathcal{R}_k = \{z_t : q_{k-1} \leq z_t < q_k\}$ are disjoint intervals, and each ψ_k is the value that the quantizer takes in the region $\mathcal{R}_k$, and (ii) a finite-level quantizer (FLQ), in which the output of the quantizer is limited to minimum and maximum values (saturated quantizer) similar to (4) with

$$\mathcal{K} = \{1, 2, \dots, L-1, L\}. \tag{6}$$

Notice that the FLQ quantizer is comprised of finite and semi-infinite intervals, given by $\mathcal{R}_1 = \{z_t : z_t < q_1\}$, $\mathcal{R}_L = \{z_t : q_{L-1} \leq z_t\}$, and $\mathcal{R}_k = \{z_t : q_{k-1} \leq z_t < q_k\}$, with $k = 2 \dots, L-1$. Usually, the sets $\mathcal{R}_k$ and the output values ψ_k are defined in terms of the quantization step Δ, for instance, see, e.g., [12,34].

Thus, the problem of interest can be defined as follows: given the available data $\mathbf{u}_{1:N} = \{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_N\}$ and $y_{1:N} = \{y_1, y_2, \dots, y_N\}$, where N is the data length, we can obtain the filtering and smoothing PDFs of the state given the quantized measurements, $p(\mathbf{x}_t|y_{1:t})$ and $p(\mathbf{x}_t|y_{1:N})$, respectively, the state estimators

$$\hat{\mathbf{x}}_{t|t} = \mathbb{E}\{\mathbf{x}_t|y_{1:t}\} = \int \mathbf{x}_t p(\mathbf{x}_t|y_{1:t}) d\mathbf{x}_t, \tag{7}$$

$$\hat{\mathbf{x}}_{t|N} = \mathbb{E}\{\mathbf{x}_t|y_{1:N}\} = \int \mathbf{x}_t p(\mathbf{x}_t|y_{1:N}) d\mathbf{x}_t, \tag{8}$$

and the corresponding covariance matrices of the estimation error:

$$\Sigma_{t|t} = \mathbb{E}\Big\{(\mathbf{x}_t - \hat{\mathbf{x}}_{t|t})(\mathbf{x}_t - \hat{\mathbf{x}}_{t|t})^\top | y_{1:t}\Big\} = \int (\mathbf{x}_t - \hat{\mathbf{x}}_{t|t})(\mathbf{x}_t - \hat{\mathbf{x}}_{t|t})^\top p(\mathbf{x}_t|y_{1:t}) d\mathbf{x}_t, \tag{9}$$

$$\Sigma_{t|N} = \mathbb{E}\Big\{(\mathbf{x}_t - \hat{\mathbf{x}}_{t|N})(\mathbf{x}_t - \hat{\mathbf{x}}_{t|N})^\top | y_{1:N}\Big\} = \int (\mathbf{x}_t - \hat{\mathbf{x}}_{t|t})(\mathbf{x}_t - \hat{\mathbf{x}}_{t|t})^\top p(\mathbf{x}_t|y_{1:N}) d\mathbf{x}_t, \tag{10}$$

where $t \leq N$ and $\mathbb{E}\{\mathbf{x}|y\}$ denotes the conditional expectation of $\mathbf{x}$ given y.

2.1. Practical Application: Liquid-Level System

To give context to the problem stated above, we included a practical application. Consider the liquid-level system shown in Figure 2; see, e.g., [35]. The goal is to estimate the liquid level in a tank using the measurements obtained by a low-cost sensor based on a variable resistor that is attached to an arm with a floater. This sensor varies its resistance in discrete steps providing the quantized measurements $y_t \in \{0, 1, \dots, 9, 10\}$. F_1 and F_2 are the total inflow and outflow rates, respectively, and f_1 with f_2 are small deviations of the inflow and outflow rate from the steady-state value F_0. Additionally, h is a small deviation of the liquid level of the tank from the steady-state value H_0. The total liquid level is $H = H_0 + h$.

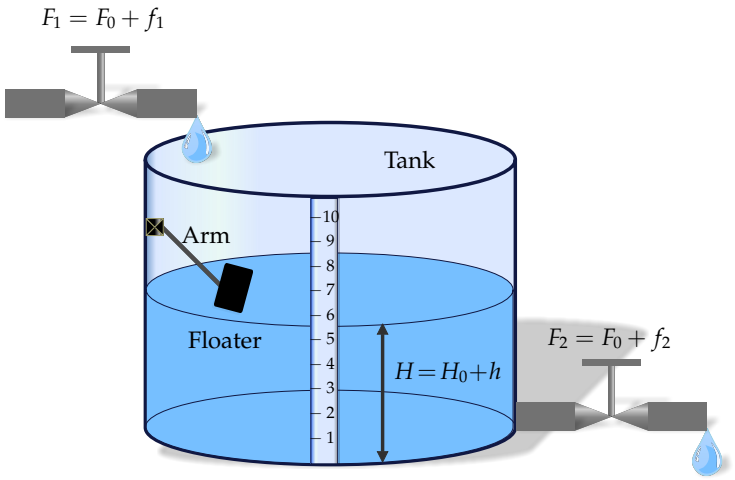

Figure 2. Liquid-level system.

The linearized model (around the operating points F_0 and H_0) that relates the input f_1 to the output h is given by the differential equation $a_1 \frac{dh}{dt} + h = a_2 f_1$. Setting $a_1 = 0.1$, $a_2 = 1$, and a sampling period $T = 0.1$ leads to the following state-space model:

$$x_{t+1} = 0.3678 x_t + u_t + w_t, \tag{11}$$

$$z_t = 0.6321 x_t + v_t, \tag{12}$$

where x_t is the level of the liquid present in the tank and z_t is a nonavailable signal which is transformed into quantized measurements by the sensor with the following model:

$$\mathfrak{q}\{z_t\} = \begin{cases} 0 & \text{if} \quad 0 \le z_t < 1, \\ 1 & \text{if} \quad 1 \le z_t < 2, \\ \vdots & \vdots \quad \vdots \\ 9 & \text{if} \quad 9 \le z_t < 10, \\ 10 & \text{if} \quad z_t \ge 10. \end{cases} \tag{13}$$

Using only input data and the quantized sensor measurements, the goal is to estimate the liquid level at every time instant $t = nT$, where $n \in \mathbb{Z}$.

3. Recursive Filtering and Smoothing Methods for Quantized Output Data

3.1. Bayesian Filtering and Smoothing

Under a Bayesian framework, the filtering distributions admit the following recursions; see, e.g., [16]:

$$p(\mathbf{x}_t | y_{1:t-1}) = \int p(\mathbf{x}_t | \mathbf{x}_{t-1}) p(\mathbf{x}_{t-1} | y_{1:t-1}) d\mathbf{x}_{t-1}, \tag{14}$$

$$p(\mathbf{x}_t | y_{1:t}) = \frac{p(y_t | \mathbf{x}_t) p(\mathbf{x}_t | y_{1:t-1})}{p(y_t | y_{1:t-1})}, \tag{15}$$

where (14) and (15) are the measurement- and time-update equations. The PDF $p(\mathbf{x}_t | \mathbf{x}_{t-1})$ is directly obtained from the model in (1), and $p(y_t | y_{1:t-1})$ is a normalization constant. On the other hand, the Bayesian smoothing equation, see, e.g., [16], is defined by the following:

$$p(\mathbf{x}_t | y_{1:N}) = p(\mathbf{x}_t | y_{1:t}) \int \frac{p(\mathbf{x}_{t+1} | y_{1:N}) p(\mathbf{x}_{t+1} | \mathbf{x}_t)}{p(\mathbf{x}_{t+1} | y_{1:t})} d\mathbf{x}_{t+1}. \tag{16}$$

Notice that to obtain $p(\mathbf{x}_t|y_{1:t})$ in (15), we need the probability density function (PDF) $p(y_t|\mathbf{x}_t)$. Since y_t is a discrete random variable, the probabilistic model of $p(y_t|\mathbf{x}_t)$ is a probability mass function (PMF). Then, the measurement-update equation in (15) combines both PDFs and a PMF. Here, we use *generalized probability density functions*; see, e.g., [36]. They combine both discrete and absolutely continuous distributions. In [32], an integral equation for $p(y_t|\mathbf{x}_t)$ is defined in order to solve the filtering recursion as follows

$$p(y_t|\mathbf{x}_t) = \int_{a_t}^{b_t} \mathcal{N}(v_t; 0, R) dv_t. \tag{17}$$

Here, a_t and b_t are functions of the boundary values of each region of the quantizers defined in Table 1.

Table 1. Integral limits of Equation (17).

	y_t	a_t	b_t
FLQ	ψ_1	$-\infty$	$q_1 - \mathbf{C}\mathbf{x}_t - \mathbf{D}\mathbf{u}_t$
	ψ_k, $k = 2, \ldots, L-1$	$q_{k-1} - \mathbf{C}\mathbf{x}_t - \mathbf{D}\mathbf{u}_t$	$q_k - \mathbf{C}\mathbf{x}_t - \mathbf{D}\mathbf{u}_t$
	ψ_L	$q_{L-1} - \mathbf{C}\mathbf{x}_t - \mathbf{D}\mathbf{u}_t$	∞
ILQ	ψ_k, $k = \ldots, 1, \ldots, L, \ldots$	$q_{k-1} - \mathbf{C}\mathbf{x}_t - \mathbf{D}\mathbf{u}_t$	$q_k - \mathbf{C}\mathbf{x}_t - \mathbf{D}\mathbf{u}_t$

Notice that $y_t|\mathbf{x}_t$ in (17) is a non-Gaussian random variable. This leads to obtaining non-Gaussian measurement- and time-update distributions. However, the EKF, QKF, and UKF filters are developed under the assumption that the measurement- and time-update distributions are Gaussian, which yields a loss of accuracy in the estimation. On the other hand, (17) is used in GSF/GSS and PF/PS, where the Gaussian assumption is not needed.

3.2. Extended State-Space System

To implement filtering and smoothing algorithms such as the EKF/EKS and the UKF/UKS, the state-space model in (1)–(3) is rewritten in an extended form as follows

$$\mathbf{x}_{t+1}^{\mathrm{e}} = \mathcal{A}\mathbf{x}_t^{\mathrm{e}} + \mathcal{B}\mathbf{u}_t^{\mathrm{e}} + \mathbf{w}_t^{\mathrm{e}}, \tag{18}$$

$$y_t = \mathfrak{q}\{\mathcal{C}\mathbf{x}_t^{\mathrm{e}}\} + \xi_t. \tag{19}$$

The extended system matrices are given by

$$\mathcal{A} = \begin{bmatrix} \mathbf{A} & 0 \\ \mathbf{C}\mathbf{A} & 0 \end{bmatrix}, \qquad \mathcal{B} = \begin{bmatrix} \mathbf{B} & 0 \\ \mathbf{C}\mathbf{B} & \mathbf{D} \end{bmatrix}, \qquad \mathcal{C} = [0 \quad 1], \tag{20}$$

where $\mathbf{x}_t^{\mathrm{e}} = [\mathbf{x}_t^\top \quad z_t]^\top$ is the extended state, $\mathbf{u}_t^{\mathrm{e}} = [\mathbf{u}_t^\top \quad \mathbf{u}_{t+1}^\top]$ is the extended input with $\mathbf{u}_{N+1} = 0$. Notice that the extended system transforms the algebraic equation of the linear output z_t into a recursive equation. It is then necessary to define an initial condition for z_t at $t = 1$. Considering $z_1 \sim \mathcal{N}(z_1; 0, \sigma_z)$, the initial condition of the extended vector become $\mathbf{x}_1^{\mathrm{e}} \sim \mathcal{N}(\mathbf{x}_1^{\mathrm{e}}; \boldsymbol{\mu}_1^{\mathrm{e}}, \mathbf{P}_1^{\mathrm{e}})$ where $\boldsymbol{\mu}_1^{\mathrm{e}} = [\boldsymbol{\mu}_1^\top \; 0]^\top$ and $\mathbf{P}_1^{\mathrm{e}} = \mathrm{diag}\{\mathbf{P}_1, \sigma_z\}$. The noise $\mathbf{w}_t^{\mathrm{e}} = [\mathbf{w}_t^\top \quad (\mathbf{C}\mathbf{w}_t + v_{t+1})^\top]^\top$ satisfies $\mathbf{w}_t^{\mathrm{e}} \sim \mathcal{N}(\mathbf{w}_t^{\mathrm{e}}; 0, \mathcal{Q})$ with

$$\mathcal{Q} = \begin{bmatrix} \mathbf{Q} & \mathbf{Q}\mathbf{C}^\top \\ \mathbf{C}\mathbf{Q}^\top & \mathbf{C}\mathbf{Q}\mathbf{C}^\top + R \end{bmatrix}. \tag{21}$$

The noise ξ_t is added in order to obtain the adequate structure of the system to implement the EKF/EKS and the UKF/UKS. However, this does not imply that the measurements are corrupted by the noise ξ_t. The idea of including an extra noise in the model

is proposed to ensure a full-rank covariance matrix in the EM algorithm; see, e.g., [37]. We assume that $\xi_t \sim \mathcal{N}(\xi_t; 0, \varepsilon)$, where the variance ε is small.

3.3. Extended Kalman Filtering and Smoothing

The idea of EKF [38,39] is to have a linear approximation around a state estimate, using a Taylor series expansion. The EKF is not directly applied to the problem of interest in this paper because the quantizer is a non-differentiable nonlinear function. In [20], it was suggested that it is possible to compute the Kalman gain using a smooth *arctan*-based approximation of the quantizer. Thus, following the idea in [20] and the representation of the arctan function found in [40], the following approximation for the quantizer is proposed:

$$\mathfrak{q}\{z_t\} \approx h(z_t) = \begin{cases} \vdots & & \vdots \\ (\Delta/\pi)\,\mathrm{atan}\{(z_t - 1.5\Delta)/\rho\} + 1.5\Delta & \text{if} & \Delta \le z_t < 2\Delta, \\ (\Delta/\pi)\,\mathrm{atan}\{(z_t - 0.5\Delta)/\rho\} + 0.5\Delta & \text{if} & 0 \le z_t < \Delta, \\ (\Delta/\pi)\,\mathrm{atan}\{(z_t + 0.5\Delta)/\rho\} - 0.5\Delta & \text{if} & -\Delta \le z_t < 0, \\ (\Delta/\pi)\,\mathrm{atan}\{(z_t + 1.5\Delta)/\rho\} - 1.5\Delta & \text{if} & -2\Delta \le z_t < -\Delta, \\ \vdots & & \vdots \end{cases} \tag{22}$$

Here, ρ is a user parameter that defines how well the approximation fits the quantizer function in the switch point, as shown in Figure 3.

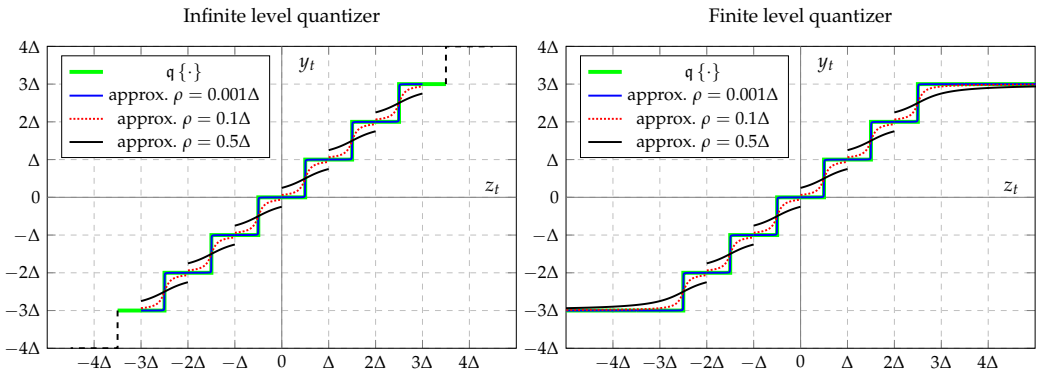

Figure 3. Quantizer approximation by using the arctan function.

On the other hand, by using the smooth approximation of the quantizer, it is possible to approximate the nonlinear system as a linear time-varying system as follows:

$$\mathbf{x}^{\mathrm{e}}_{t+1} = \mathcal{A}\mathbf{x}^{\mathrm{e}}_t + \mathcal{B}\mathbf{u}^{\mathrm{e}}_t + \mathbf{w}^{\mathrm{e}}_t, \tag{23}$$

$$y_t = \mathbf{H}_t\mathbf{x}^{\mathrm{e}}_t + \mathbf{F}_t + \xi_t, \tag{24}$$

where $\mathbf{H}_t$ is the Jacobian matrix of $h(\mathcal{C}\mathbf{x}^{\mathrm{e}}_t)$ with respect to $\mathbf{x}^{\mathrm{e}}_t$, and evaluated at $\hat{\mathbf{x}}^{\mathrm{e}}_{t|t-1}$ and $\mathbf{F}_t = h(\mathcal{C}\hat{\mathbf{x}}^{\mathrm{e}}_{t|t-1}) - \mathbf{H}_t\hat{\mathbf{x}}^{\mathrm{e}}_{t|t-1}$. Then, the equations of the EKF and EKS are summarized in Algorithms 1 and 2, respectively. One of the difficulties in applying the EKF to deal with quantized data is the computation of the Jacobian matrix $\mathbf{H}_t$. Despite the approximation of the quantizer, the Jacobian is nearly zero for all values of $\hat{\mathbf{x}}^{\mathrm{e}}_{t|t-1}$, except for the exact switch points, as shown in Figure 4 (left), where $\rho = 0.001\Delta$. Additionally, Figure 4 (center and right) shows that the quantizer and Jacobian approximations worsen as ρ increases, which reduces the accuracy of the estimation.

Algorithm 1: Extended Kalman filter algorithm for quantized output data

Input: The distribution of the initial condition $p(\mathbf{x}_1^{\mathrm{e}})$, e.g., $\hat{\mathbf{x}}_{0|1}^{\mathrm{e}} = \boldsymbol{\mu}_1^{\mathrm{e}}$ and $\boldsymbol{\Sigma}_{0|1}^{\mathrm{e}} = \mathbf{P}_1^{\mathrm{e}}$.
for $t = 1$ ***to*** N **do**
  Compute the Kalman gain using: $\mathbf{K}_t = \boldsymbol{\Sigma}_{t|t-1}^{\mathrm{e}} \mathbf{H}_t^{\top} \left(\varepsilon + \mathbf{H}_t \boldsymbol{\Sigma}_{t|t-1}^{\mathrm{e}} \mathbf{H}_t^{\top} \right)^{-1}$.
  Measurement Update:
  Compute the filtering state $\hat{\mathbf{x}}_{t|t}^{\mathrm{e}}$ according to $\hat{\mathbf{x}}_{t|t}^{\mathrm{e}} = \hat{\mathbf{x}}_{t|t-1}^{\mathrm{e}} + \mathbf{K}_t \left(y_t - h\left(\mathcal{C} \hat{\mathbf{x}}_{t|t-1}^{\mathrm{e}} \right) \right)$.
  Compute the covariance matrix $\boldsymbol{\Sigma}_{t|t}^{\mathrm{e}}$ according to $\boldsymbol{\Sigma}_{t|t}^{\mathrm{e}} = (\mathbf{I} - \mathbf{K}_t \mathbf{H}_t) \boldsymbol{\Sigma}_{t|t-1}^{\mathrm{e}}$.
  Time Update:
  Compute the predicted state $\hat{\mathbf{x}}_{t+1|t}^{\mathrm{e}}$ according to $\hat{\mathbf{x}}_{t+1|t}^{\mathrm{e}} = \mathcal{A} \hat{\mathbf{x}}_{t|t}^{\mathrm{e}} + \mathcal{B} \mathbf{u}_t^{\mathrm{e}}$.
  Compute the covariance matrix $\boldsymbol{\Sigma}_{t+1|t}^{\mathrm{e}}$ according to $\boldsymbol{\Sigma}_{t+1|t}^{\mathrm{e}} = \mathcal{Q} + \mathcal{A} \boldsymbol{\Sigma}_{t|t}^{\mathrm{e}} \mathcal{A}^{\top}$.
end
Output: The PDF $p(\mathbf{x}_t^{\mathrm{e}} | y_{1:t})$ and the PDF $p(\mathbf{x}_{t+1}^{\mathrm{e}} | y_{1:t})$ for $t = 1, \ldots, N$.

Algorithm 2: Extended Kalman smoother algorithm for quantized output data

Input: The PDF $p(\mathbf{x}_t^{\mathrm{e}} | y_{1:t}) \sim \mathcal{N}\left(\mathbf{x}_t^{\mathrm{e}}; \hat{\mathbf{x}}_{t|t}^{\mathrm{e}}, \boldsymbol{\Sigma}_{t|t}^{\mathrm{e}} \right)$ and the PDF $p(\mathbf{x}_{t+1}^{\mathrm{e}} | y_{1:t}) \sim \mathcal{N}\left(\mathbf{x}_{t+1}^{\mathrm{e}}; \hat{\mathbf{x}}_{t+1|t}^{\mathrm{e}}, \boldsymbol{\Sigma}_{t+1|t}^{\mathrm{e}} \right)$ for $t = 1, \ldots, N$ computed in Algorithm 1.
for $t = N$ ***to*** 1 **do**
  Compute the gain $\mathbf{G}_t = \boldsymbol{\Sigma}_{t|t}^{\mathrm{e}} \mathcal{A}_t^{\top} \left(\boldsymbol{\Sigma}_{t+1|t}^{\mathrm{e}} \right)^{-1}$.
  Compute the smoothing state $\hat{\mathbf{x}}_{t|N}^{\mathrm{e}}$ according to $\hat{\mathbf{x}}_{t|N}^{\mathrm{e}} = \hat{\mathbf{x}}_{t|t}^{\mathrm{e}} + \mathbf{G}_t \left(\hat{\mathbf{x}}_{t+1|T}^{\mathrm{e}} - \hat{\mathbf{x}}_{t+1|t}^{\mathrm{e}} \right)$.
  Compute the covariance matrix $\boldsymbol{\Sigma}_{t|N}^{\mathrm{e}}$ according to $\boldsymbol{\Sigma}_{t|N}^{\mathrm{e}} = \boldsymbol{\Sigma}_{t|t}^{\mathrm{e}} + \mathbf{G}_t \left(\boldsymbol{\Sigma}_{t+1|T}^{\mathrm{e}} - \boldsymbol{\Sigma}_{t+1|t}^{\mathrm{e}} \right) \mathbf{G}_t^{\top}$.
end
Output: The PDF $p(\mathbf{x}_t^{\mathrm{e}} | y_{1:N}) \sim \mathcal{N}\left(\mathbf{x}_t^{\mathrm{e}}; \hat{\mathbf{x}}_{t|N}^{\mathrm{e}}, \boldsymbol{\Sigma}_{t|N}^{\mathrm{e}} \right)$ for $t = 1, \ldots, N$.

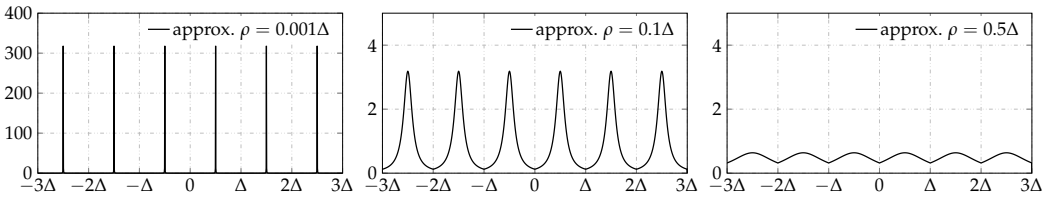

Figure 4. Jacobian matrix of the quantizer approximation.

3.4. Unscented Kalman Filtering and Smoothing

The unscented Kalman Filter [18] is a deterministic sampling-based approach that uses samples called sigma points to propagate the mean and covariance of the system state (assumed to be a Gaussian random variable) through the nonlinear functions of the system. These propagated points accurately capture the mean and covariance of the posterior state to the 3rd-order Taylor series expansion for any nonlinear function [41]. The key idea of UKF is to directly approximate the mean and covariance of the posterior distribution instead of approximating nonlinear functions [18]. The unscented Kalman Filter is based on the unscented transformation of the random variable $\mathbf{x} \in \mathbb{R}^n$ into the random variable $\mathbf{y} = \mathbf{g}(\mathbf{x}) + \mathbf{v}$, where $\mathbf{g}(\cdot)$ is a nonlinear function, $\mathbf{x} \sim \mathcal{N}(\mathbf{x}; \mathbf{m}, \boldsymbol{\Gamma})$, and $\mathbf{v} \sim \mathcal{N}(\mathbf{v}; \mathbf{0}, \mathbf{P})$. Thus, the sigma points are defined by

$$\mathcal{X}^0 = \mathbf{m}, \tag{25}$$

$$\mathcal{X}^{\tau} = \mathbf{m} + \sqrt{n + \lambda} \left[\boldsymbol{\Gamma}^{1/2} \right]_{\tau}, \tag{26}$$

$$\mathcal{X}^{\tau+n} = \mathbf{m} - \sqrt{n + \lambda} \left[\boldsymbol{\Gamma}^{1/2} \right]_{\tau}. \tag{27}$$

Here, $\tau = 1, \ldots, n$, the scaling parameter $\lambda = \alpha^2(n + \kappa) - n$, the parameters α and κ determine the propagation of the sigma points around the mean, and the notation $[\mathcal{P}]_\tau$ refers to the τth column of the matrix $\mathcal{P}$. The weights associated with the unscented transformation are the sets

$$\left\{\varphi^0, \varphi^1, \ldots, \varphi^{2n}\right\} = \{\lambda\zeta, 0.5\zeta, \ldots, 0.5\zeta\}, \tag{28}$$

$$\left\{\sigma^0, \sigma^1, \ldots, \sigma^{2n}\right\} = \{\lambda\zeta + \varrho, 0.5\zeta, \ldots, 0.5\zeta\}, \tag{29}$$

where $\zeta = (n + \lambda)^{-1}$ and $\varrho = 1 - \alpha^2 + \beta$. In this set of equations, β is an additional parameter that can be used to incorporate prior information on the $\mathbf{x}$ distribution. Then, the statistics of the transformed random variable are the mean $\boldsymbol{\mu} = \sum_{\tau=0}^{2n} \varphi^\tau \mathbf{g}(\mathcal{X}^\tau)$ and the covariance matrix $\boldsymbol{\Phi} = \sum_{\tau=0}^{2n} \sigma^\tau [\mathbf{g}(\mathcal{X}^\tau) - \boldsymbol{\mu}][\mathbf{g}(\mathcal{X}^\tau) - \boldsymbol{\mu}]^\top + \mathbf{P}$. Additionally, the cross-covariance matrix between $\mathbf{x}$ and $\mathbf{y}$ is given by $\boldsymbol{\Psi} = \sum_{\tau=0}^{2n} \sigma^\tau [\mathcal{X}^\tau - \mathbf{m}][\mathbf{g}(\mathcal{X}^\tau) - \boldsymbol{\mu}]^\top$. The steps to implement the UKF are summarized in Algorithm 3. Notice that for the problem of interest in this paper, the process equation is a linear function. Thus, the UKS algorithm is similar to EKS but uses the filtering and predictive distributions obtained from the UKF algorithm.

Algorithm 3: Unscented Kalman filter algorithm for quantized output data

Input: The distribution of the initial condition $p(\mathbf{x}_1^{\mathrm{e}})$, i.e., $\hat{\mathbf{x}}_{0|1}^{\mathrm{e}} = \boldsymbol{\mu}_1^{\mathrm{e}}$ and $\boldsymbol{\Sigma}_{0|1}^{\mathrm{e}} = \mathbf{P}_1^{\mathrm{e}}$, the constant α, κ, and β.
for $t = 1$ ***to*** N **do**
  Compute and store the sigma points $\mathcal{X}_{t|t-1}^\tau$, the weights $\varphi_{t|t-1}^\tau$ and $\sigma_{t|t-1}^\tau$ by using $\hat{\mathbf{x}}_{t|t-1}^{\mathrm{e}}$ and $\boldsymbol{\Sigma}_{t|t-1}^{\mathrm{e}}$ for $\tau = 0, \ldots, 2n$.
  Propagate the sigma points using the measurement model $\mathcal{Y}_t^\tau = \mathfrak{q}\left\{\mathcal{C}\mathcal{X}_{t|t-1}^\tau\right\}$ for $\tau = 0, \ldots, 2n$.
  Compute the gain $\mathbf{K}_t = \boldsymbol{\Xi}_t \mathbf{S}_t^{-1}$, where

$$\boldsymbol{\nu}_t = \sum_{\tau=1}^{2n} \varphi_{t|t-1}^\tau \mathcal{Y}_t^\tau,$$

$$\mathbf{S}_t = \sum_{\tau=1}^{2n} \sigma_{t|t-1}^\tau (\mathcal{Y}_t^\tau - \boldsymbol{\nu}_t)(\mathcal{Y}_t^\tau - \boldsymbol{\nu}_t)^\top + \varepsilon,$$

$$\boldsymbol{\Xi}_t = \sum_{\tau=1}^{2n} \sigma_{t|t-1}^\tau \left(\mathcal{X}_{t|t-1}^\tau - \hat{\mathbf{x}}_{t|t-1}^{\mathrm{e}}\right)(\mathcal{Y}_t^\tau - \boldsymbol{\nu}_t)^\top.$$

  Measurement Update:
  Compute the filtering state $\hat{\mathbf{x}}_{t|t}^{\mathrm{e}}$ according to $\hat{\mathbf{x}}_{t|t}^{\mathrm{e}} = \hat{\mathbf{x}}_{t|t-1}^{\mathrm{e}} + \mathbf{K}_t(y_t - \boldsymbol{\nu}_t)$.
  Compute the covariance matrix $\boldsymbol{\Sigma}_{t|t}^{\mathrm{e}}$ according to $\boldsymbol{\Sigma}_{t|t}^{\mathrm{e}} = \boldsymbol{\Sigma}_{t|t-1}^{\mathrm{e}} - \mathbf{K}_t \mathbf{S}_t \mathbf{K}_t^\top$.
  Time Update:
  Compute the predicted state $\hat{\mathbf{x}}_{t+1|t}^{\mathrm{e}}$ according to $\hat{\mathbf{x}}_{t+1|t}^{\mathrm{e}} = \mathcal{A}\hat{\mathbf{x}}_{t|t}^{\mathrm{e}} + \mathcal{B}\mathbf{u}_t$.
  Compute the covariance matrix $\boldsymbol{\Sigma}_{t+1|t}^{\mathrm{e}}$ according to $\boldsymbol{\Sigma}_{t+1|t}^{\mathrm{e}} = \mathcal{Q} + \mathcal{A}_t \boldsymbol{\Sigma}_{t|t}^{\mathrm{e}} \mathcal{A}_t^\top$.
end
Output: The PDF $p(\mathbf{x}_t^{\mathrm{e}}|\mathbf{y}_{1:t})$ and the PDF $p(\mathbf{x}_{t+1}^{\mathrm{e}}|\mathbf{y}_{1:t})$ for $t = 1, \ldots, N$.

3.5. Quantized Kalman Filtering and Smoothing

The quantized Kalman filter is an alternative version of the Kalman filter that modifies the measurement update equation to include the quantization effect in the computation of the filtering distributions. This modification can be performed in different ways [21,22]. In this work, we use the following modification of the KF:

$$\hat{\mathbf{x}}_{t|t} = \hat{\mathbf{x}}_{t|t-1} + \mathbf{K}_t \left(y_t - \mathfrak{q}\left\{\mathbf{C}\hat{\mathbf{x}}_{t|t-1} - \mathbf{D}\mathbf{u}_t\right\}\right), \tag{30}$$

where $\mathbf{K}_t$ is the Kalman gain. Notice that, with the modification in (30), the QKS algorithm is similar to the standard Kalman smoother (KS).

3.6. Gaussian Sum Filtering and Smoothing

The Gaussian Sum Filter [31,32] is a novel approach to deal with quantized output data. The key idea of GSF is to approximate the integral of $p(y_t|\mathbf{x}_t)$ given in (17) using the Gauss–Legendre quadrature rule. This approximation produces a model with a Gaussian sum structure as follows:

$$p(y_t|\mathbf{x}_t) \approx \sum_{\tau=1}^{K} \varsigma_t^\tau \mathcal{N}(\eta_t^\tau; \mathbf{C}\mathbf{x}_t + \mathbf{D}\mathbf{u}_t + \mu_t^\tau, \mathbf{R}). \tag{31}$$

Here, K is the number of points from the Gauss–Legendre quadrature rule, ς_t^τ, η_t^τ, and μ_t^τ are defined in Table 2, and ω_τ and ψ_τ are weights and points defined by the quadrature rule, given in, e.g., [33].

Table 2. Parameters of the $p(y_t|\mathbf{x}_t)$ approximation using the Gauss–Legendre quadrature.

	y_t	ς_t^τ	η_t^τ	μ_t^τ
FLQ	ψ_1	$2\omega_\tau/(1+\psi_\tau)^2$	$-(1-\psi_\tau)/(1+\psi_\tau)$	$-q_1$
	ψ_k, $k=2,\ldots,L-1$	$\omega_\tau(q_k-q_{k-1})/2$	$\psi_\tau(q_k-q_{k-1})/2$	$-(q_k+q_{i-1})/2$
	ψ_L	$2\omega_\tau/(1+\psi_\tau)^2$	$(1-\psi_\tau)/(1+\psi_\tau)$	$-q_{L-1}$
ILQ	ψ_k, $k=\ldots,1,\ldots,L,\ldots$	$\omega_\tau(q_k-q_{i-1})/2$	$\psi_\tau(q_k-q_{i-1})/2$	$-(q_k+q_{i-1})/2$

Using the approximation of $p(y_t|\mathbf{x}_t)$ given in (31), the Gaussian sum filter iterates between the following two steps:

$$p(\mathbf{x}_t|y_{1:t}) = \sum_{k=1}^{M_{t|t}} \gamma_{t|t}^k \mathcal{N}(\mathbf{x}_t; \hat{\mathbf{x}}_{t|t}^k, \Sigma_{t|t}^k), \tag{32}$$

$$p(\mathbf{x}_{t+1}|y_{1:t}) = \sum_{k=1}^{M_{t+1|t}} \gamma_{t+1|t}^k \mathcal{N}(\mathbf{x}_{t+1}; \hat{\mathbf{x}}_{t+1|t}^k, \Sigma_{t+1|t}^k). \tag{33}$$

All quantities in this recursion can be computed following the algorithm in Algorithm 4. On the other hand, to compute the smoothing distribution, (16) is separated into two formulas, to avoid the division by a non-Gaussian distribution [42]. The first formula is the backward recursion that is defined as follows (obtained by using the approximation of $p(y_t|\mathbf{x}_t)$ given in (31)):

$$p(y_{t+1:N}|\mathbf{x}_t) = \sum_{k=1}^{S_{t|t+1}} \epsilon_{t|t+1}^k \lambda_{t|t+1}^k \exp\left\{-\frac{1}{2}\left(\mathbf{x}_t^\top \mathbf{F}_{t|t+1}^k \mathbf{x}_t - 2\mathbf{G}_{t|t+1}^{kT} \mathbf{x}_t + H_{t|t+1}^k\right)\right\}, \tag{34}$$

$$p(y_{t:N}|\mathbf{x}_t) = \sum_{k=1}^{S_{t|t}} \epsilon_{t|t}^k \lambda_{t|t}^k \exp\left\{-\frac{1}{2}\left(\mathbf{x}_t^\top \mathbf{F}_{t|t}^k \mathbf{x}_t - 2\mathbf{G}_{t|t}^{kT} \mathbf{x}_t + H_{t|t}^k\right)\right\}. \tag{35}$$

All quantities in this recursion can be computed following the algorithm in Algorithm 5. The second formula computes the smoothing distribution as follows:

$$p(\mathbf{x}_t|y_{1:N}) = \sum_{k=1}^{S_{t|N}} \epsilon_{t|N}^k \mathcal{N}(\mathbf{x}_t; \hat{\mathbf{x}}_{t|N}^k, \Sigma_{t|N}^k). \tag{36}$$

In this equation, all quantities can be computed following Algorithm 6.

Algorithm 4: Gaussian sum filter algorithm for quantized output data.

Input: The PDF of the initial state $p(\mathbf{x}_1)$, e.g., $M_{1|0} = 1$, $\gamma_{1|0} = 1$, $\hat{\mathbf{x}}_{1|0} = \boldsymbol{\mu}_1$, $\boldsymbol{\Sigma}_{1|0} = \mathbf{P}_1$. The points of the Gauss–Legendre quadrature $\{\omega_\tau, \psi_\tau\}_{\tau=1}^{K}$.
for $t = 1$ ***to*** N **do**
  Compute and store ς_t^τ, η_t^τ, and μ_t^τ according to Table 2.
  Measurement Update:
  Set $M_{t|t} = KM_{t|t-1}$.
  for $\ell = 1$ ***to*** $M_{t|t-1}$ **do**
    for $\tau = 1$ ***to*** K **do**
      Set the index $k = (\ell - 1)K + \tau$.
      Compute the weights, means, and covariances matrices as follows:

$$\begin{aligned}
\gamma_{t|t}^k &= \bar{\gamma}_{t|t}^k \left(\textstyle\sum_{s=1}^{M_{t|t}} \bar{\gamma}_{t|t}^s \right)^{-1}, \\
\mathbf{K}_t^\ell &= \boldsymbol{\Sigma}_{t|t-1}^\ell \mathbf{C}^\top \left(R + \mathbf{C} \boldsymbol{\Sigma}_{t|t-1}^\ell \mathbf{C}^\top \right)^{-1}, \\
\hat{\mathbf{x}}_{t|t}^k &= \hat{\mathbf{x}}_{t|t-1}^\ell + \mathbf{K}_t^\ell (\eta_t^\tau - \mathbf{C} \hat{\mathbf{x}}_{t|t-1}^\ell - \mathbf{D}\mathbf{u}_t - \mu_t^\tau,), \\
\boldsymbol{\Sigma}_{t|t}^k &= (\mathbf{I} - \mathbf{K}_t^\ell \mathbf{C}) \boldsymbol{\Sigma}_{t|t-1}^\ell, \\
\bar{\gamma}_{t|t}^k &= \varsigma_t^\tau \gamma_{t|t-1}^\ell \mathcal{N} \left(\eta_t^\tau ; \mathbf{C} \hat{\mathbf{x}}_{t|t-1}^\ell + \mathbf{D}\mathbf{u}_t + \mu_t^\tau, R + \mathbf{C} \boldsymbol{\Sigma}_{t|t-1}^\ell \mathbf{C}^\top \right).
\end{aligned}$$

    end
  end
  Perform the Gaussian-sum-reduction algorithm according to [43] to obtain the reduced GMM of $p(\mathbf{x}_t | y_{1:t})$.
  Time Update
  Set $M_{t+1|t} = M_{t|t}$.
  for $k = 1$ ***to*** $M_{t+1|t}$ **do**
    Compute and store the weights, means, and covariance matrices as follows:

$$\begin{aligned}
\gamma_{t+1|t}^k &= \gamma_{t|t}^k, \\
\hat{\boldsymbol{x}}_{t+1|t}^k &= \mathbf{A} \hat{\mathbf{x}}_{t|t}^k + \mathbf{B}\mathbf{u}_t, \\
\boldsymbol{\Sigma}_{t+1|t}^k &= \mathbf{Q} + \mathbf{A} \boldsymbol{\Sigma}_{t|t}^k \mathbf{A}^\top.
\end{aligned}$$

  end
end
Output: The filtering PDFs $p(\mathbf{x}_t | y_{1:t})$, the predictive PDFs $p(\mathbf{x}_{t+1} | y_{1:t})$, and the set $\{\varsigma_t^\tau, \eta_t^\tau, \mu_t^\tau\}$, for $t = 1, \ldots, N$.

Algorithm 5: Backward-filtering algorithm for quantized output data.

Input: The initial backward measurement $p(y_N|\mathbf{x}_N)$ at $t = N$ with parameters: $S_{N|N} = K$ and

$$\epsilon^k_{N|N} = \varsigma^k_N, \qquad \theta^k_N = \eta^k_N - \mathbf{D}\mathbf{u}_N - \mu^k_N, \qquad \mathbf{G}^{kT}_{N|N} = \theta^{kT}_N R^{-1}\mathbf{C},$$
$$\lambda^k_{N|N} = (\det\{2\pi R\})^{-1/2}, \qquad \mathbf{F}^k_{N|N} = \mathbf{C}^\top R^{-1}\mathbf{C}, \qquad H^k_{N|N} = \theta^{kT}_N R^{-1}\theta^k_N,$$

with the set $\{\varsigma^\tau_t, \eta^\tau_t, \mu^\tau_t\}$ for $t = 1, \ldots, N$ being computed from Algorithm 4.

for $t = N - 1$ ***to*** 1 **do**

  Backward Prediction

  Set $S_{t|t+1} = S_{t+1|t+1}$.

  for $k = 1$ ***to*** $S_{t|t+1}$ **do**

    Compute and store the backward prediction update quantities as follows

$$\begin{aligned}
\epsilon^k_{t|t+1} &= \epsilon^k_{t+1|t+1},\\
\lambda^k_{t|t+1} &= \left(\det\{\mathbf{Q}\}\det\left\{\mathbf{F}_{qk}\right\}\right)^{-1/2}\lambda^k_{t+1|t+1},\\
\mathbf{F}^k_{t|t+1} &= \mathbf{A}^\top\mathbf{M}_{qk}\mathbf{A},\\
\mathbf{G}^{kT}_{t|t+1} &= \mathbf{G}^{kT}_{t+1|t+1}\mathbf{F}^{-1}_{qk}\mathbf{Q}^{-1}\mathbf{A} - \mathbf{u}^\top_t\mathbf{B}^\top\mathbf{M}_{qk}\mathbf{A},\\
H^k_{t|t+1} &= H^k_{t+1|t+1} - \mathbf{G}^{kT}_{t+1|t+1}\mathbf{F}^{-1}_{qk}\mathbf{G}^k_{t+1|t+1}\\
&\quad + \mathbf{u}^\top_t\mathbf{B}^\top\mathbf{M}_{qk}\mathbf{B}\mathbf{u}_t - 2\mathbf{u}^\top_t\mathbf{B}^\top\mathbf{Q}^{-1}\mathbf{F}^{-1}_{qk}\mathbf{G}^k_{t+1|t+1},
\end{aligned}$$

    where $\mathbf{F}_{qk} = \mathbf{F}^k_{t+1|t+1} + \mathbf{Q}^{-1}$ and $\mathbf{M}_{qk} = \mathbf{Q}^{-1} - \mathbf{Q}^{-1}\mathbf{F}^{-1}_{qk}\mathbf{Q}^{-1}$.

  end

  Backward Measurement Update:

  Set $S_{t|t} = KS_{t|t+1}$.

  for $\ell = 1$ ***to*** $S_{t|t+1}$ **do**

    for $\tau = 1$ ***to*** K **do**

      Set the index $k = (\ell - 1)K + \tau$.

      Compute the backward measurement update quantities as follows

$$\begin{aligned}
\epsilon^k_{t|t} &= \varsigma^\tau_t\epsilon^\ell_{t|t+1}, & \mathbf{F}^k_{t|t} &= \mathbf{F}^\ell_{t|t+1} + \mathbf{C}^\top R^{-1}\mathbf{C},\\
\theta^\tau_t &= \eta^\tau_t - \mathbf{D}\mathbf{u}_t - \mu^\tau_t, & \mathbf{G}^{kT}_{t|t} &= \mathbf{G}^{\ell T}_{t|t+1} + \theta^{\tau T}_t R^{-1}\mathbf{C},\\
\lambda^k_{t|t} &= (\det\{2\pi R\})^{-1/2}\lambda^\ell_{t|t+1}, & H^k_{t|t} &= H^\ell_{t|t+1} + \theta^{\tau T}_t R^{-1}\theta^\tau_t.
\end{aligned}$$

    end

  end

  Compute the GMM structure of $p(y_{t:N}|\mathbf{x}_t)$ using Lemma A.3 in [32].

  Perform the Gaussian sum reduction algorithm according to [43] to obtain the reduced GMM structure of $p(y_{t:N}|\mathbf{x}_t)$, see Equation (54) in [32]:

$$p(y_{t:N}|\mathbf{x}_t) = \sum_{k=1}^{S_{\text{red}}}\delta^k_{t|t}\mathcal{N}\left(\mathbf{x}_t; \mathbf{z}^k_{t|t}, \mathbf{U}^k_{t|t}\right),$$

  where S_{red}, $\delta^k_{t|t}$, $\mathbf{z}^k_{t|t}$, and $\mathbf{U}^k_{t|t}$ are the number of Gaussian components kept after the Gaussian reduction procedure, the weight, mean, and covariance matrix, respectively.

  Compute and store the backward filter form of the reduced version of $p(y_{t:N}|\mathbf{x}_t)$ using Lemma A.3 in [32].

end

Output: The backward prediction $p(y_{t+1:N}|\mathbf{x}_t)$ and the backward measurement update $p(y_{t:N}|\mathbf{x}_t)$ for $t = N, \ldots, 1$.

Algorithm 6: Gaussian sum smoothing algorithm for quantized output data.

Input: The PDFs $p(\mathbf{x}_t|y_{1:t-1})$ and $p(\mathbf{x}_N|y_{1:N})$ obtained from Algorithm 4 and the reduced version of $p(y_{t:N}|\mathbf{x}_t)$ obtained from Algorithm 5, see (54) in [32].
Save the PDF $p(\mathbf{x}_N|y_{1:N})$.
for $t = N-1$ ***to*** 1 **do**
  Set $S_{t|N} = M_{t|t-1}S_{\text{red}}$.
  for $\ell = 1$ ***to*** S_{red} **do**
    for $\tau = 1$ ***to*** $M_{t|t-1}$ **do**
      Set the index $k = (\ell-1)M_{t|t-1} + \tau$.
      Compute the weights, means, and covariance matrices as follows:

$$\epsilon^k_{t|N} = \bar{\epsilon}^k_{t|N}\left(\sum_{s=1}^{S_{t|N}} \bar{\epsilon}^s_{t|N}\right)^{-1},$$
$$\hat{\mathbf{x}}^k_{t|N} = \boldsymbol{\Sigma}^\tau_{t|t-1}\left(\mathbf{U}^\ell_{t|t} + \boldsymbol{\Sigma}^\tau_{t|t-1}\right)^{-1}\mathbf{z}^\ell_{t|t} + \mathbf{U}^\ell_{t|t}\left(\mathbf{U}^\ell_{t|t} + \boldsymbol{\Sigma}^\tau_{t|t-1}\right)^{-1}\hat{\mathbf{x}}^\tau_{t|t-1},$$
$$\boldsymbol{\Sigma}^k_{t|N} = \boldsymbol{\Sigma}^\tau_{t|t-1}\left(\mathbf{U}^\ell_{t|t} + \boldsymbol{\Sigma}^\tau_{t|t-1}\right)^{-1}\mathbf{U}^\ell_{t|t},$$

      where

$$\bar{\epsilon}^k_{t|N} = \frac{\gamma^\tau_{t|t-1}\delta^\ell_{t|t}\exp\left\{-\frac{1}{2}(\hat{\mathbf{x}}^\tau_{t|t-1} - \mathbf{z}^\ell_{t|t})^\top\left(\mathbf{U}^\ell_{t|t} + \boldsymbol{\Sigma}^\tau_{t|t-1}\right)^{-1}(\hat{\mathbf{x}}^\tau_{t|t-1} - \mathbf{z}^\ell_{t|t})\right\}}{(2\pi)^{\frac{n}{2}}\sqrt{\det\left\{\mathbf{U}^\ell_{t|t} + \boldsymbol{\Sigma}^\tau_{t|t-1}\right\}}}.$$

    end
  end
end
Output: The smoothing PDFs $p(\mathbf{x}_t|y_{1:N})$, for $t = 1, \ldots, N$.

3.7. Particle Filtering and Smoothing

Particle filtering [24,25] is a Monte Carlo method that approximately represents the filtering distributions $p(\mathbf{x}_t|y_{1:t})$ of the state variables conditioned to the observations $y_{1:t}$ by using a set of weighted random samples, called particles, so that

$$p(\mathbf{x}_t|y_{1:t}) \approx \sum_{i=1}^{M} w_t^{(i)}\delta\left(\mathbf{x}_t - \mathbf{x}_t^{(i)}\right), \tag{37}$$

where $\delta(\cdot)$ is the Dirac delta function, $w_t^{(i)}$ denotes the ith weight, $\mathbf{x}_t^{(i)}$ denotes the ith particle sampled from the filtering distribution $p(\mathbf{x}_t|y_{1:t})$, and M is the number of particles. Since the filtering distribution is unknown in the current iteration, it is difficult or impossible to sample directly from it. In this case, the particles are usually generated from a known density that is chosen (by the user) to facilitate the sampling process. This is called importance sampling, and the PDF is called importance density. Then, the importance weight computation can be carried out in a recursive fashion (sequential importance sampling, SIS) as follows:

$$w_t^{(i)} \propto w_{t-1}^{(i)}\frac{p(y_t|\mathbf{x}_t^{(i)})p(\mathbf{x}_t^{(i)}|\mathbf{x}_{t-1}^{(i)})}{h(\mathbf{x}_t|\mathbf{x}_{t-1}^{(i)}, y_t)}, \tag{38}$$

where $h(\mathbf{x}_t|\mathbf{x}_{t-1}^{(i)}, y_t)$ is the importance density and $w_{t-1}^{(i)}$ are the importance weights of the previous iteration. On the other hand, the choice of importance distribution is critical for performing particle filtering and smoothing. The particle filter literature shows that the importance density $p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(i)}, y_t)$ is optimal in the sense that it minimizes the variance in the importance weights $w_t^{(i)}$ [16,25]. However, in most cases, it is difficult or impossible to draw samples for this optimal importance density, except for particular cases such

as a state-space model with a nonlinear process and linear output Equation [25]. Many sub-optimal methods have been developed to approximate the importance density, such as Markov chain Monte Carlo [44], ensemble Kalman filter [28], local linearization of the state-space model, and local linearization of the optimal importance distribution [25], among others. One of the most commonly used importance densities in the literature is the state transition prior $p(\mathbf{x}_t|\mathbf{x}_{t-1})$; see, e.g., [25,31,45]. This choice yields an intuitive and simple-to-implement algorithm with $w_t^{(i)} \propto w_{t-1}^{(i)} p(y_t|\mathbf{x}_t^{(i)})$. This algorithm is called a *bootstrap* filter [24].

The particle filter suffers from a problem called the *degeneracy phenomenon*. As shown in [25], the variance in the importance weights can only increase over time. This implies that after a few iterations, most particles have negligible weights; see also [46]. A consequence of the degeneracy problem is that a large computational effort is devoted to updating particles whose contribution to the final estimate is nearly zero. To solve the degeneracy problem, the *resampling approach* was proposed in [24]. The resampling method eliminates the particles that have small weights, and the particles with large weights are replicated, generating a new set (with replacement) of equally weighted particles.

Additionally, the resampling method used to reduce the degeneracy effect on the particles produces another unwanted issue called *particle impoverishment*. This effect implies a loss of diversity in the sample set since the resampled particles will contain many repeated points that were generated from heavily weighted particles. In the worst-case scenario, all particles can be produced by a single particle with a large weight [29]. To solve the impoverishment problem, methods such as roughening and regularization have been suggested in the literature [26]. Markov chain Monte Carlo (MCMC) is another method used after the resampling step to add variability to the resampled particles [47]. The basic idea is to apply the MCMC algorithm to each resampled particle with $p(\mathbf{x}_t|y_{1:t})$ as the target distribution. That is, we need to build a Markov chain by sampling a proposal particle $\mathbf{x}_t^*$ from the proposal density. Then, $\mathbf{x}_t^*$ is accepted only if $u \leq \varpi(\mathbf{x}_t^*, \mathbf{x}_t)$, with $u \sim \mathcal{U}[0,1]$, where $\mathcal{U}[a_1, a_2]$ corresponds to the uniform distribution over the real numbers in the interval $[a_1, a_2]$, and $\varpi(\mathbf{x}_t^*, \mathbf{x}_t)$ is the acceptance ratio given by

$$\varpi(\mathbf{x}_t^*, \mathbf{x}_t) = \min\left\{1, \frac{p(y_t|\mathbf{x}_t^*)}{p(y_t|\mathbf{x}_t)}\right\}. \tag{39}$$

With this process, the diversity of the new particles is greater than the resampled ones, reducing the risk of particle impoverishment. Additionally, the new particles are distributed according to $p(\mathbf{x}_t|y_{1:t})$. In this paper, we use the MH and RWM algorithms to build on the MCMC step. In Algorithm 7, we summarize the steps to implement the particle filter with the MCMC step.

Algorithm 7: MCMC-based particle filter algorithm for quantized output data

Input: $p(\mathbf{x}_1)$, the number of particles M.
Draw the samples $\mathbf{x}_1^{(i)} \sim p(\mathbf{x}_1)$ and set $w_1^{(i)} = 1/M$ for $i = 1, \ldots, M$.
for $t = 1$ ***to*** N **do**
  From the importance distribution draw the samples $\mathbf{x}_t^{(i)} \sim h(\mathbf{x}_t|\mathbf{x}_{t-1}^{(i)}, y_t)$ for $i = 1, \ldots, M$.
  Calculate the weights $w_t^{(i)}$ using $p(y_t|\mathbf{x}_t)$ given in (17) according to (38) for $i = 1, \ldots, M$.
  Normalize the weights $w_t^{(i)}$ to sum one.
  Perform resampling and generates a new set of weights $w_t^{(i)}$ and particles $\mathbf{x}_t^{(i)}$ for $i = 1, \ldots, M$. Notice that in the resampling algorithms MR, SR, and MTR $w_t^{(i)} = 1/M$ for $i = 1, \ldots, M$. The LS algorithm produces a new set of weights; see, e.g., [26].
  Implement the MCMC move: for $\ell = 1, \ldots, M$.
    Pick the sample $\mathbf{x}_t^{(\ell)}$ from the set of the resampled particles.
      MH: Sample a proposal particle $\mathbf{x}_t^*$ from the proposal PDF.
      RWM: Generate $\mathbf{x}_t^+$ from $\mathcal{N}(0, \Lambda^2)$ (Λ is defined by the user) and compute $\mathbf{x}_t^* = \mathbf{x}_t^{(\ell)} + \mathbf{x}_t^+$.
    Evaluate $\varpi(\mathbf{x}_t^*, \mathbf{x}_t^{(\ell)})$ given in (39). If $u \leq \varpi(\mathbf{x}_t^*, \mathbf{x}_t^{(\ell)})$, then accept the move ($\mathbf{x}_t^{(\ell)} = \mathbf{x}_t^*$) else reject the move ($\mathbf{x}_t^{(\ell)} = \mathbf{x}_t^{(\ell)}$).
end
Output: $w_t^{(i)}$, and $\mathbf{x}_t^{(i)} \sim p(\mathbf{x}_t|y_{1:t})$, $i = 1, \ldots, M$.

Similar to particle filtering, particle smoothing is a Monte Carlo method that approximately represents the smoothing distributions $p(\mathbf{x}_t|y_{1:N})$ of the state variables conditioned upon the observations $y_{1:N}$, using random samples as follows:

$$p(\mathbf{x}_t|y_{1:N}) \approx \sum_{i=1}^{M} w_{t|N}^{(i)} \delta\left(\mathbf{x}_t - \tilde{\mathbf{x}}_t^{(i)}\right), \tag{40}$$

where $w_{t|N}^{(i)}$ denotes the ith weight, $\tilde{\mathbf{x}}_t^{(i)}$ denotes the ith particle sampled from the smoothing distribution $p(\mathbf{x}_t|y_{1:N})$, and M is the number of particles. Some smoothing algorithms are based on the particles provided by the particle filtering, i.e., $\mathbf{x}_t^{(i)}$, such as the backward-simulation particle smoother [48] and marginal particle smoother [25]. Particularly in the marginal particle smoother, the weights $w_{t|N}^{(i)}$ are updated in reverse time as follows:

$$w_{t|N}^{(i)} = \sum_{j=1}^{M} w_{t+1|N}^{(j)} \frac{w_t^{(i)} p(\mathbf{x}_{t+1}^{(j)}|\mathbf{x}_t^{(i)})}{\sum_{k=1}^{M} w_t^{(k)} p(\mathbf{x}_{t+1}^{(j)}|\mathbf{x}_t^{(k)})}, \tag{41}$$

where $w_{N|N}^{(i)} = w_N^{(i)}$ for $i = 1, \ldots, M$ and the approximation of (40) is performed using $\tilde{\mathbf{x}}_t^{(i)} = \mathbf{x}_t^{(i)}$ for $t = N, \ldots, 1$. In this paper, we use the smoothing method developed in [49]. The problem of interest in this work admits further simplifications; see also [50]. This smoothing method [49] requires the evaluation of the function $f(\mathbf{x}_{t+1}^{(i)}, \mathbf{x}_t^{(\tau)})$ given by

$$f(\mathbf{x}_{t+1}^{(i)}, \mathbf{x}_t^{(\tau)}) = \exp\left\{-\frac{1}{2}\boldsymbol{\eta}_t^\top \mathbf{Q}^{-1} \boldsymbol{\eta}_t\right\}, \tag{42}$$

where $\boldsymbol{\eta}_t = \mathbf{x}_{t+1}^{(i)} - \mathbf{A}\mathbf{x}_t^{(\tau)} - \mathbf{B}\mathbf{u}_t$. In Algorithm 8, we summarize the steps to implement the particle smoother.

Algorithm 8: Rejection-based particle smoother algorithm for quantized output data

Input: Weights $w_t^{(i)}$, and particles $\mathbf{x}_t^{(i)}$ provided by Algorithm 7, for $t = 1, \ldots, N$, and $i = 1, \ldots, M$.
Set $\tilde{\mathbf{x}}_N^{(i)} = \mathbf{x}_N^{(i)}$ and $w_{N|N}^{(i)} = 1/M$ for $i = 1, \ldots, M$.
for $t = N - 1$ ***to*** 1 **do**
  for $i = 1$ ***to*** M **do**
    do
      Take $\tau \sim \mathcal{U}([1, ..., M])$.
      Take $u \sim \mathcal{U}[0, 1]$.
    while $u > f(\mathbf{x}_{t+1}^{(i)}, \mathbf{x}_t^{(\tau)})$;
    Set $\tilde{\mathbf{x}}_t^{(i)} = \mathbf{x}_t^{(\tau)}$ and $w_{t|N}^{(i)} = 1/M$.
  end
end
Output: $w_{t|N}^{(i)}$, and $\tilde{\mathbf{x}}_t^{(i)} \sim p(\mathbf{x}_t|y_{1:N})$, $i = 1, \ldots, M$.

On the other hand, provided the weights $w^{(i)}$ and particles $\mathbf{x}^{(i)}$ (from particle filter or smoother), the state estimators in (7) and (8) and the covariance matrices of the estimation error in (9) and (10) can be computed from

$$\mathbb{E}\{g(\mathbf{x}_t)|s\} \approx \sum_{i=1}^{M} w^{(i)} g(\mathbf{x}^{(i)}), \tag{43}$$

where $g(\mathbf{x}_t)$ represents a function of $\mathbf{x}_t$. For example, the mean and covariance matrix of the filtering and smoothing distributions can be computed with $g(\mathbf{x}_t) = \mathbf{x}_t$ and $g(\mathbf{x}_t) = (\mathbf{x}_t - \mathbb{E}\{\mathbf{x}_t\})(\mathbf{x}_t - \mathbb{E}\{\mathbf{x}_t\})^\top$, respectively. The variable s represents the observation set that is used, which is $s = y_{1:t}$ for filtering and $s = y_{1:N}$ for smoothing.

4. Numerical Experiment

In this section, we present a numerical example to analyze the performance of KF/KS, EKF/EKS, QKF/QKS, UKF/UKS, GSF/GSS, and MCMC-based PF/PS having quantized observations. We use the discrete-time system in the state-space form given in (1)–(2) with

$$y_t = \Delta \text{round}(z_t/\Delta). \tag{44}$$

In (44), Δ is a quantization step, and *round* is the Matlab function that computes the nearest decimal or integer. The sets $\mathcal{R}_k$ are computed using $q_{k-1} = y_t - 0.5\Delta$ and $q_k = y_t + 0.5\Delta$. We compare the performance of all filtering and smoothing algorithms considering eight variations of the PF, where we use the Markov chain Monte Carlo method MH and RWM with the following resampling methods: SYS, ML, MT, and LS. For clarity of presentation, we use the bootstrap filter, and we solve the integral in (17) using the cumulative distribution function computed with the Matlab function *mvncdf*. We consider the state-space system given by (1)–(2) with $\mathbf{A} = 0.9$, $\mathbf{B} = 1.2$, $\mathbf{C} = 2.2$, and $\mathbf{D} = 0.75$. We also consider that $\mathbf{w}_t \sim \mathcal{N}(\mathbf{w}_t; 0, 1)$, $v_t \sim \mathcal{N}(v_t; 0, 0.5)$, the input signal is drawn from $\mathcal{N}(0, 1)$, and $\mathbf{x}_1 \sim \mathcal{N}(\mathbf{x}_1; 1, 0.01)$, and the quantization step $\Delta = 8$. To implement EKF/EKS $\rho = 0.1$, to implement UKF/UKS $\alpha = 0.001$, $\kappa = 0.001$, and $\beta = 1$. To implement the GSF/GSS $K = 10$, and to implement PF/PS we consider $\Lambda^2 = 100$, and $M = \{100, 500, 1000\}$.

In Figures 5 and 6, we show the filtering and smoothing distributions, i.e., $p(\mathbf{x}_t|y_{1:t})$ and $p(\mathbf{x}_t|y_{1:N})$, for a time instant. We freeze the results of KF, QKF, EKF, UKF, and GSF to observe the behavior of the PF when varying the number of used samples, and when different MCMC methods are used with different resampling algorithms. These figures show that the PDFs obtained using GSF/GSS are the ones that best fit the ground truth,

followed by PF/PS. Furthermore, in Figures 7 and 8, we show the boxplot of the mean square error (MSE) between the estimated and the true state after running 1000 Monte Carlo experiments. These figures show the loss of accuracy of the state estimates obtained with KF/KS, EKF/EKS, QKF/QKS, and UKF/UKS, and better performance for GSF/GSS and PF/PS (except for the PF version that uses LS resampling). Additionally, we can observe that in terms of accuracy, the PF/PS implementation that gives the lower MSE is the one that uses the MCMC move RWM with SYS, ML, and MT resampling methods. On the other hand, in terms of the computational load, PS in almost all its versions exhibits the highest execution time, followed by the GSS, EKS, UKS, and KS (see Figure 9). In Table 3, we ranked all the algorithms studied in the present manuscript in terms of the mean of the MSE and the execution time. This table suggests that there is a trade-off between the accuracy of the estimates and the execution time in the case of PF/PS. The GSF/GSS, on the other hand, exhibits high accuracy in the estimation compared with all its analogs and exhibits a relatively short execution time compared with (i) the PF/PS using 500 and 1000 particles, (ii) the PF/PS using the MT resampling method with 100 particles and (iii) the EKF/EKS algorithms.

Table 3. Rank of the filtering and smoothing recursive algorithms for quantized data. References: KF/KS, EKF/EKS [16], QKF/QKS [21,22], PF/PS [24], UKF/UKS [16,18], GSF/GSS [31,32]. The notation XX-YY-ZZ(M) denotes the following: XX stands for the filtering or smoothing algorithm (PF or PS), YY stands for the MCMC algorithm (RWM or MH), ZZ stands for the resampling method (SYS, ML, MT or LS), and (M) stands for the number of particles used (100, 500, or 1000).

Rank	Filtering		Smoothing		Smoothing Execution Time	
	MSE	Algorithm	MSE	Algorithm	Execution Time	Algorithm
1	0.6724	GSF	0.5207	GSS	0.0026	KS
2	0.6740	PF-RWM-SYS(1000)	0.5212	PS-RWM-SYS(1000)	0.0031	QKS
3	0.6744	PF-RWM-ML(1000)	0.5220	PS-RWM-ML(1000)	0.0111	UKS
4	0.6754	PF-RWM-SYS(500)	0.5231	PS-RWM-SYS(500)	0.1453	PS-RWM-SYS(100)
5	0.6765	PF-RWM-ML(500)	0.5247	PS-RWM-ML(500)	0.1644	PS-RWM-LS(100)
6	0.6880	PF-RWM-SYS(100)	0.5393	PS-RWM-MT(1000)	0.1718	PS-RWM-ML(100)
7	0.6948	PF-RWM-ML(100)	0.5415	PS-RWM-SYS(100)	0.2077	PS-MH-LS(100)
8	0.7588	PF-RWM-MT(1000)	0.5420	PS-RWM-MT(500)	0.2109	PS-MH-SYS(100)
9	0.7830	PF-RWM-MT(500)	0.5470	PS-RWM-ML(100)	0.2354	PS-MH-ML(100)
10	0.9590	PF-MH-SYS(1000)	0.5689	PS-RWM-MT(100)	0.3931	GSS
11	0.9593	PF-MH-MT(1000)	0.6708	PS-MH-SYS(1000)	0.3984	PS-RWM-MT(100)
12	0.9595	PF-MH-ML(1000)	0.6711	PS-MH-ML(1000)	0.4579	PS-MH-MT(100)
13	0.9608	PF-MH-SYS(500)	0.6737	PS-MH-SYS(500)	0.4676	EKS
14	0.9612	PF-MH-MT(500)	0.6746	PS-MH-ML(500)	0.6048	PS-RWM-SYS(500)
15	0.9612	PF-MH-ML(500)	0.6752	PS-MH-MT(1000)	0.6348	PS-RWM-LS(500)
16	0.9686	PF-MH-SYS(100)	0.6781	PS-MH-MT(500)	0.7469	PS-RWM-ML(500)
17	0.9697	PF-MH-ML(100)	0.6927	PS-MH-SYS(100)	0.9772	PS-MH-LS(500)
18	0.9715	PF-MH-MT(100)	0.6927	PS-MH-ML(100)	1.2054	PS-RWM-SYS(1000)
19	1.0138	KF	0.6974	PS-MH-MT(100)	1.2274	PS-RWM-LS(1000)
20	1.6731	PF-RWM-MT(100)	0.7469	PS-MH-LS(1000)	1.2709	PS-MH-SYS(500)

Table 3. *Cont.*

Rank	Filtering		Smoothing		Smoothing Execution Time	
	MSE	**Algorithm**	**MSE**	**Algorithm**	**Execution Time**	**Algorithm**
21	1.8616	QKF	0.7497	PS-MH-LS(500)	1.4192	PS-MH-ML(500)
22	5.0549	UKF	0.7667	PS-MH-LS(100)	1.5197	PS-RWM-ML(1000)
23	7.3381	PF-RWM-LS(1000)	0.9100	KS	1.8362	PS-RWM-MT(500)
24	7.3602	PF-RWM-LS(500)	0.9393	PS-RWM-LS(1000)	2.0277	PS-MH-LS(1000)
25	7.6912	PF-RWM-LS(100)	1.2900	PS-RWM-LS(500)	2.3945	PS-MH-MT(500)
26	8.3846	PF-MH-LS(1000)	1.6693	QKS	3.0254	PS-MH-SYS(1000)
27	8.4079	PF-MH-LS(500)	5.0545	UKS	3.3505	PS-MH-ML(1000)
28	8.6717	PF-MH-LS(100)	6.4904	PS-RWM-LS(100)	3.6364	PS-RWM-MT(1000)
29	47.7827	EKF	33.8842	EKS	5.0651	PS-MH-MT(1000)

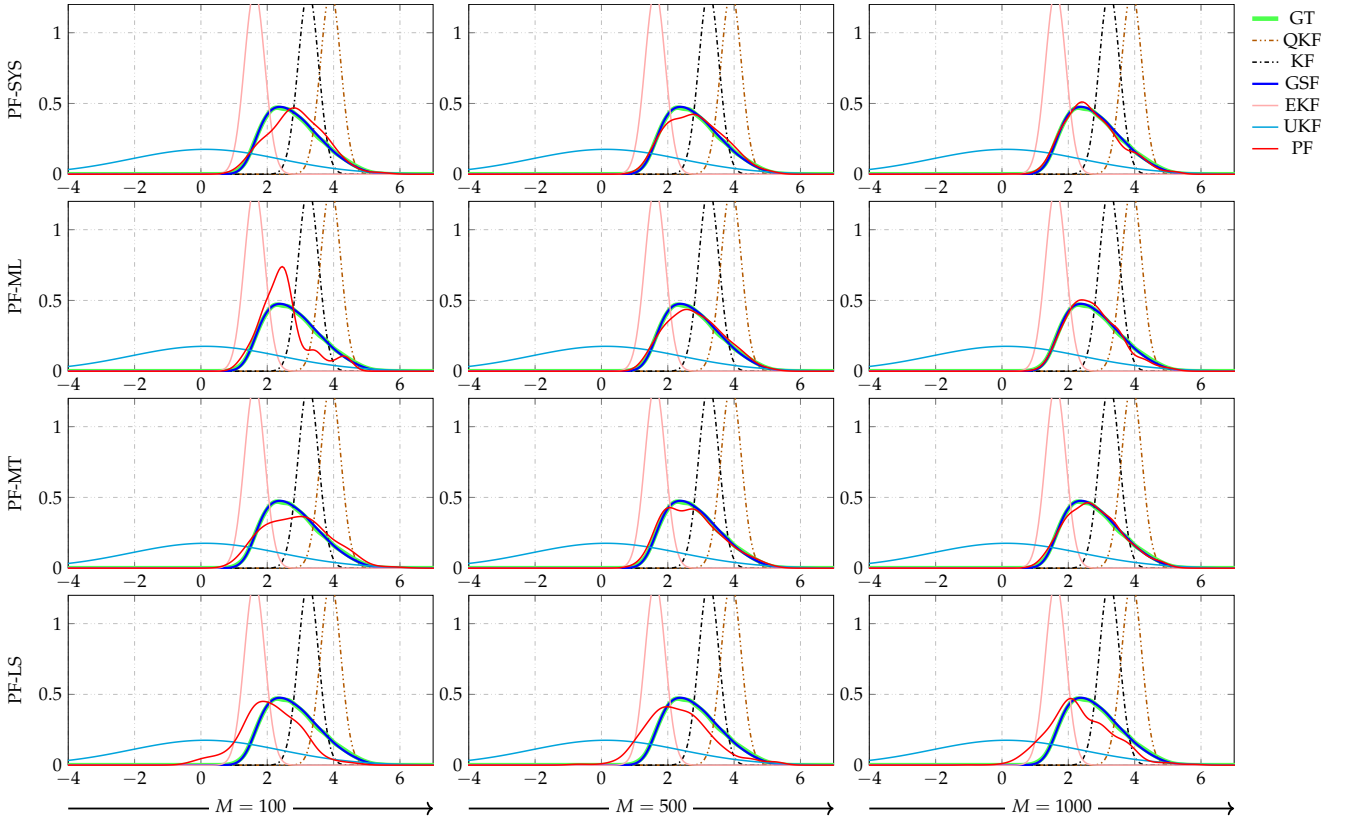

Figure 5. Filtering PDFs for a time instant. GT stands for the ground truth. KF, EKF, QKF, UKF, GSF, and PF stand for Kalman filter, extended Kalman filter, quantized Kalman filter, unscented Kalman filter, Gaussian sum filter, and particle filter, respectively. The PDFs given by the KF, EKF, QKF, UKF were frozen in all plots to observe the behavior of the PF (with RWM moves) when the number of particles increased. SYS, ML, MT, LS stand for systematic, multinomial, metropolis, and local selection resampling algorithms, respectively.

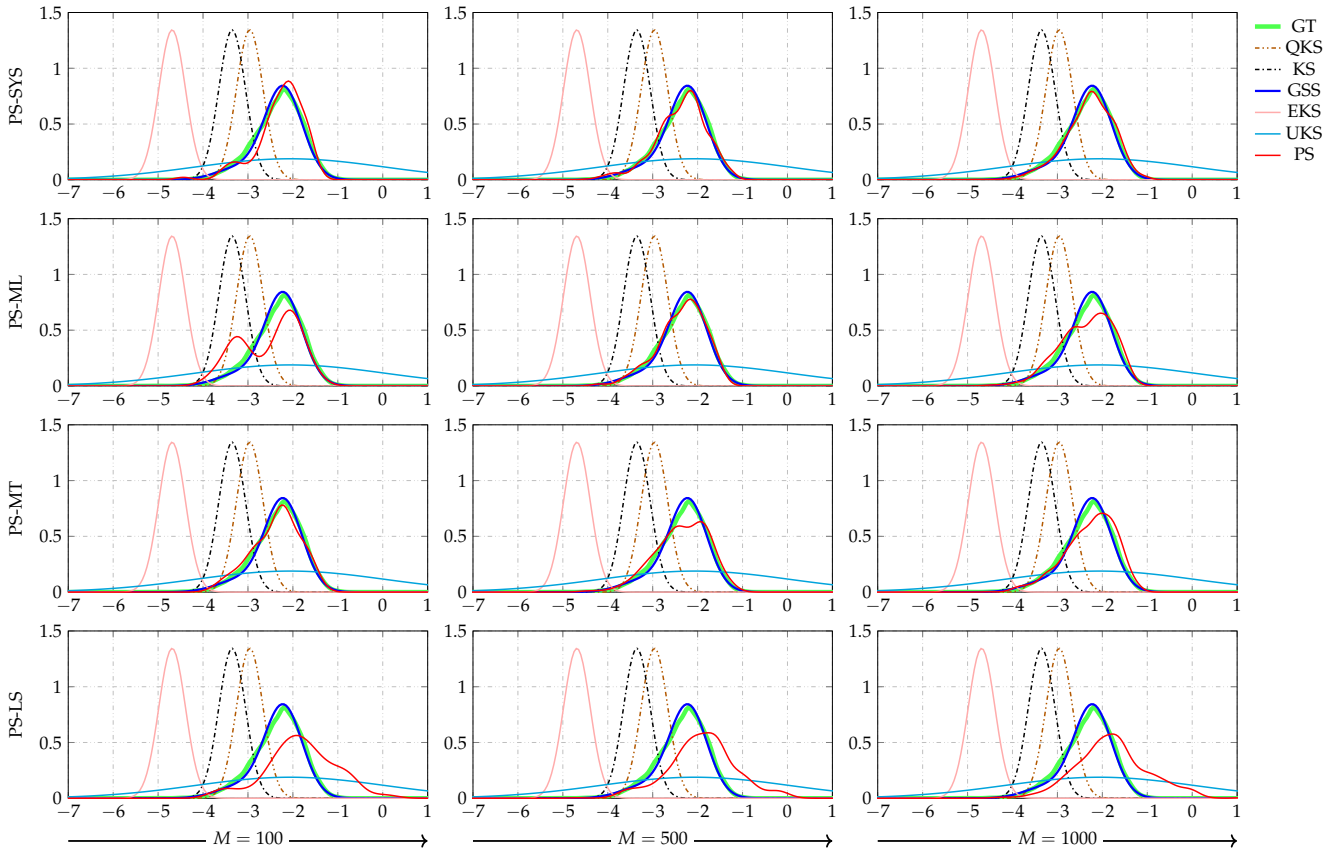

Figure 6. Smoothing PDFs for a time instant. GT stands for the ground truth. KS, EKS, QKS, UKS, GSS, and PS stand for Kalman smoother, extended Kalman smoother, quantized Kalman smoother, unscented Kalman smoother, Gaussian sum smoother, and particle smoother, respectively. The PDFs given by the KS, EKS, QKS, and UKS were frozen in all plots to observe the behavior of the PS (with RWM moves) when the number of particles increased. SYS, ML, MT, LS stand for systematic, multinomial, metropolis, and local selection resampling algorithms, respectively.

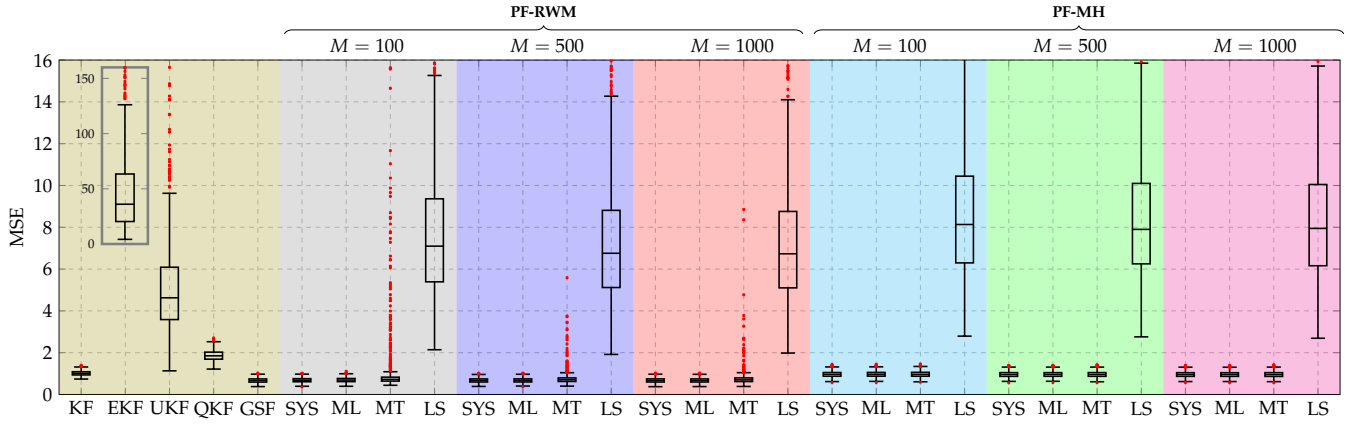

Figure 7. Boxplot of the MSE between the estimated and true state for 1000 Monte Carlo experiments. KF, EKF, QKF, UKF, GSF, and PF stand for Kalman filter, extended Kalman filter, quantized Kalman filter, unscented Kalman filter, Gaussian sum filter, and particle filter, respectively. Additionally, SYS, ML, MT, LS stand for systematic, multinomial, metropolis, and local selection resampling algorithms, respectively. RWM and MH denote random walk Metropolis and Metropolis–Hasting moves.

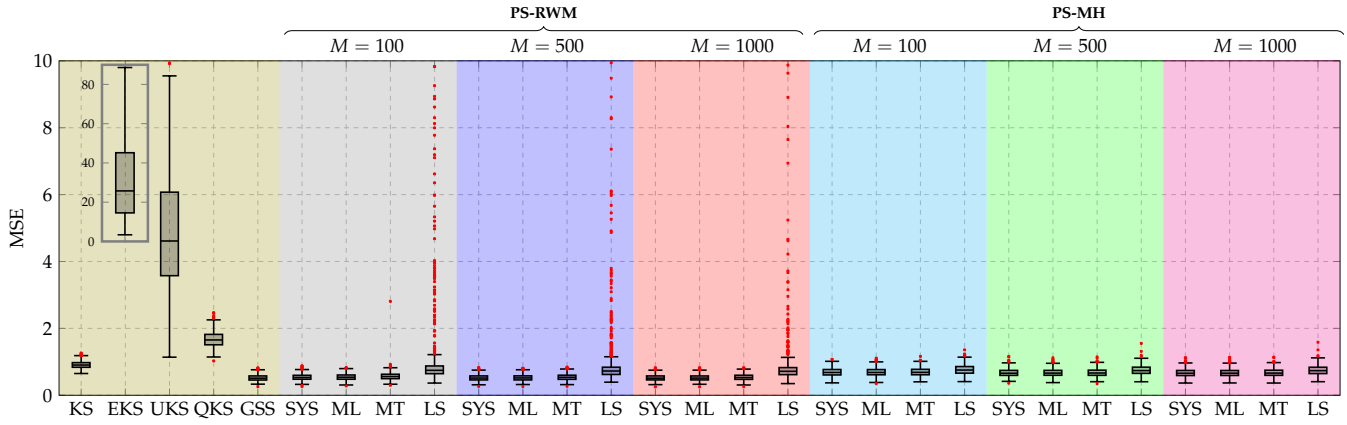

Figure 8. Boxplot of the MSE between the estimated and true state for 1000 Monte Carlo experiments. KS, EKS, QKS, UKS, GSS, and PS stand for Kalman smoother, extended Kalman smoother, quantized Kalman smoother, unscented Kalman smoother, Gaussian sum smoother, and particle smoother, respectively. Additionally, SYS, ML, MT, and LS stand for systematic, multinomial, metropolis, and local selection resampling algorithms, respectively. RWM and MH denote random walk Metropolis and Metropolis–Hasting moves.

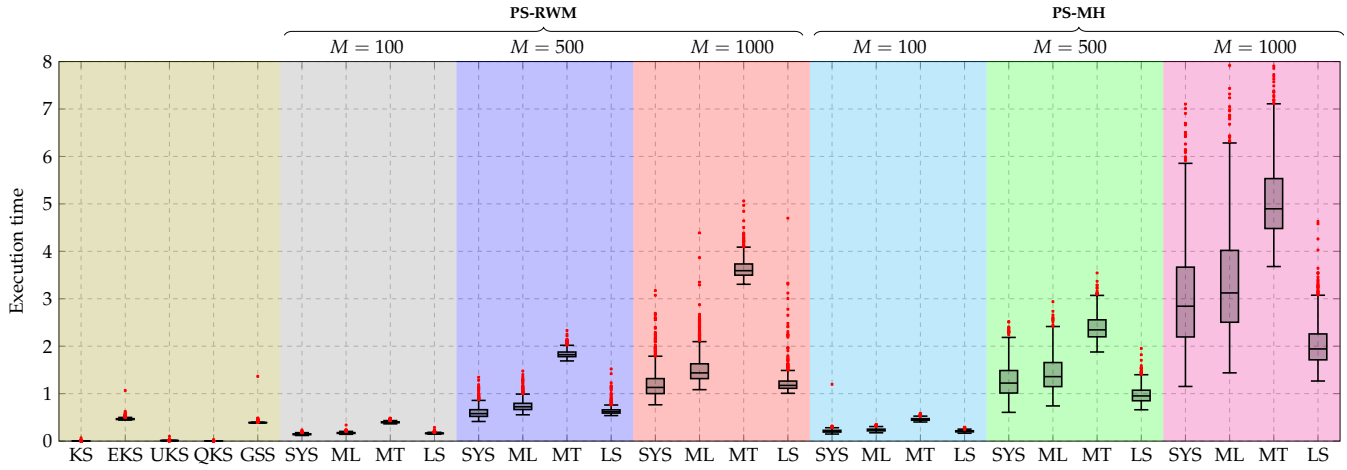

Figure 9. Boxplot of the execution time for 1000 Monte Carlo experiments. KS, EKS, QKS, UKS, GSS, and PS stand for Kalman smoother, extended Kalman smoother, quantized Kalman smoother, unscented Kalman smoother, Gaussian sum smoother, and particle smoother, respectively. Additionally, SYS, ML, MT, and LS stand for systematic, multinomial, metropolis, and local selection resampling algorithms, respectively. RWM and MH denote random walk Metropolis and Metropolis–Hasting moves.

5. Practical Application Revisited: Liquid-Level System

We now consider the liquid-level system detailed in Section 2.1. To simulate the system in (11) and (12), we consider $w_t \sim \mathcal{N}(w_t; 0, 0.1)$ and $v_t \sim \mathcal{N}(v_t; 0, 0.05)$. The input u_t is drawn from $u_t \sim \mathcal{N}(u_t; 8, 25)$, and the initial condition satisfies $x_1 \sim \mathcal{N}(x_1; 1, 0.01)$. In this example, we implement the following filtering and smoothing algorithms: EKF/EKS, UKF/UKS, QKF/QKS, GSF/GSS, and PF/PS. The latter filtering algorithm was implemented using the systematic resampling and MH methods with 500 particles. Additionally, to implement the Gaussian Sum algorithms, we consider $K = 10$ points from the Gauss–Legendre quadrature rule. We simulate 100 Monte Carlo experiments for both the filtering and smoothing algorithms with $N = 100$.

Figure 10a shows one realization of the non-quantized signal z_t and the quantized output y_t. Figure 10b shows the execution time of all smoothing algorithms, where we see that PS has the highest computational cost by a considerable margin, followed by the

GSS. In terms of estimation accuracy, the MSE between the real and estimated tank liquid level corresponding to the filtered and smoothed states is documented in Figure 10c,d, respectively. These boxplots show that GSF and GSS exhibit the lowest MSE, followed by the UKF/UKS, QKF/QKS, PF/PS, and EKF/EKS. Taking into consideration these results, this practical setup also illustrates that the Gaussian sum filter and smoother provides the best trade-off between estimation accuracy and computational cost.

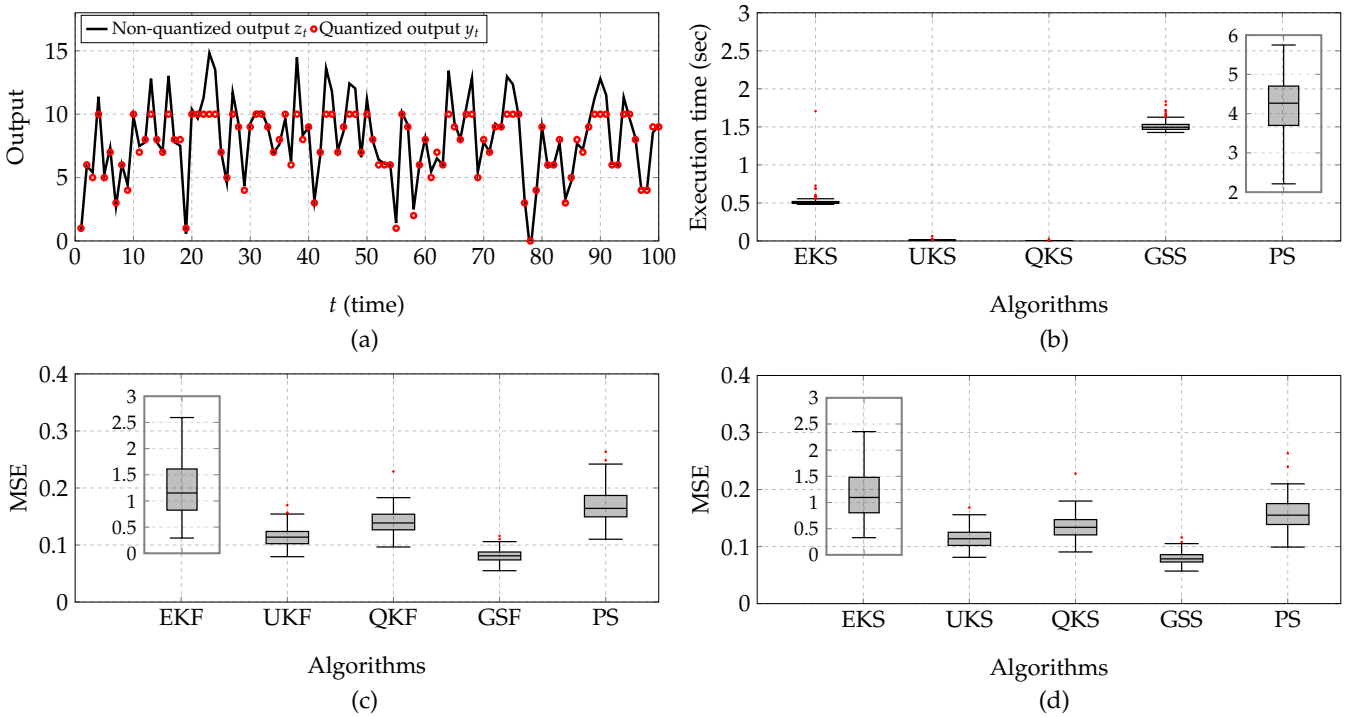

Figure 10. Practical application: (**a**) A realization of the non-quantized signal z_t and the quantized output y_t. (**b**) Boxplot of the execution time of the smoothing algorithms. (**c**) Boxplot of the MSE between the real and estimated (filtered) tank liquid level. (**d**) Boxplot of the MSE between the real and estimated (smoothed) tank liquid level. EKF/EKS, QKF/QKS, UKF/UKS, GSF/GSS, and PF/PS stand for extended Kalman filter/smoother, quantized Kalman filter/smoother, unscented Kalman filter/smoother, Gaussian sum filter/smoother, and particle filter/smoother, respectively.

6. User Guidelines and Comments

All the filtering and smoothing algorithms studied in this paper have a number of hyperparameters that need to be chosen based on their purpose. Hence, some guidelines are provided based on both the numerical analysis and also on the authors' practical experience.

- To implement the EKF/EKS, the user parameter ρ defines the arctan-approximation accuracy of the quantizer, which impacts the accuracy of the approximation for $\mathbf{H}_t$ in (24). Choose a small value of ρ to obtain a high accuracy for the quantizer. The disadvantage of these algorithms is that despite an accurate approximation of the quantizer, the estimation of the state of the system is not accurate for a coarse quantization scheme;
- To implement UKF/UKS, the parameter α is usually set to a small positive value, for instance, $\alpha = \{0.01, 0.001, 0.0001\}$. The parameter κ is typically set to zero or a very small positive value, for instance, $\kappa = \{0, 0.001, 1 \times 10^{-10}\}$. For the extra parameter β, if the random variable to transform is Gaussian distributed, it is known that $\beta = 2$ is optimal [41]. In the problem of interest in this work, the random variables—after the unscented transformation—are non-Gaussian, and the parameter β can be chosen heuristically so that the estimation error is acceptable;

- Based on the authors' practical experience, the QKF/QKS produces an accurate estimation of the system states (under the assumption that filtering and smoothing distributions are Gaussian) if the quantization step is small compared to the amplitude of the output signal. However, the accuracy of the estimates decreases as the quantization step increases. The advantage of this algorithm is that it is easy to implement, and it is faster compared to more sophisticated implementations such as the PF/PS and the GSF/GSS;
- To implement the GSF/GSS, choose the number of Gauss–Legendre quadrature points as $K = \{4, 6, 10\}$. These values of K produce highly accurate estimates for the system states and the filtering/smoothing PDFs with a low computational cost. Additionally, these algorithms produce directly an explicit model for the filtering and smoothing PDFs without extra algorithms. The disadvantage of the GSF/GSS algorithms is that they are difficult to implement since they require the backward filter recursion and the Gaussian sum reduction algorithms;
- The PF/PS produces accurate estimations of the system state with a relatively low amount of particles. For instance, $M = \{100, 200, 500\}$ are good choices for low-order models. These algorithms are easy to implement, and there are many resampling methods that can be replicated. The disadvantage of the PF/PS algorithms is that the computational cost increases rapidly as the number of particles and the system order increases. Additionally, the PF/PS does not directly produce the filtering and smoothing PDFs unless a PDF-fitting algorithm is implemented. This introduces an extra computational cost if filtering and smoothing PDFs are required;
- In some situations, as is the case shown in Figures 7 and 8, the QKF/QKS performs worse than the standard KF/KS in terms of estimation accuracy. This suggests that there are cases with fine quantization, where if the accuracy of the estimation is not critical, but the execution time is, then the user can choose to neglect the quantization block and pick the standard KF/KS algorithms for state estimation.

7. Conclusions

In this paper, we investigated the performance of the extended Kalman filter/smoother, quantized Kalman filter/smoother, unscented Kalman filter/smoother, Gaussian sum filter/smoother, and particle filter/smoother for state-space models with quantized observations. The analysis was carried out in terms of the accuracy of the estimation, using the MSE and the computational cost as performance indexes. Simulations show that the PDFs of Gaussian sum filter/smoother and particle filter/smoother with a high number of particles are the ones best fitting the ground-truth PDFs. However, contrary to the particle filter/smoother, the Gaussian sum filter/smoother does not require a high computational load to achieve accurate results. The extended Kalman filter/smoother, quantized Kalman filter/smoother, and unscented Kalman filter/smoother produce results with low accuracy, although their execution time is minor. From simulations, we observed that the performance of the particle filter is closely related to the number of samples, the choice of the resampling method, and the MCMC algorithms, which address the degeneracy problem and mitigate the sample impoverishment. We used four different resampling schemes combined with two MCMC algorithms. We found out that the implementation of the MCMC-based particle filter and smoothing that produces the lower MSE is the one using random walk Metropolis combined with the systematic resampling technique.

Author Contributions: Conceptualization, A.L.C., R.A.G., and J.C.A.; methodology, A.L.C. and J.C.A.; software, A.L.C.; validation, R.C. and B.I.G.; formal analysis, A.L.C., R.A.G., R.C., B.I.G. and J.C.A.; investigation, A.L.C., R.A.G. and J.C.A.; resources, J.C.A.; writing—original draft preparation, A.L.C., R.A.G., R.C., B.I.G. and J.C.A.; writing—review and editing, A.L.C., R.C., B.I.G. and J.C.A.; visualization, R.C. and B.I.G.; supervision, J.C.A. All authors have read and agreed to the published version of the manuscript

Funding: Grants ANID-Fondecyt 1211630 and 11201187, and ANID-Basal Project FB0008 (AC3E). Chilean National Agency for Research and Development (ANID) Scholarship Program/Doctorado Nacional/2020-21202410. VIDI Grant 15698, which is (partly) financed by the Netherlands Organization for Scientific Research (NWO). Excellence Center at Linköping, Lund, in Information Technology, ELLIIT.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

KF/KS	Kalman Filter/Rauch–Tung–Striebel smoother
EKF/EFS	Extended Kalman filter/smoother
UKF/UKS	Unscented Kalman filter/smoother
QKF/QKS	Quantized Kalman filter/smoother
PF/PS	Particle filter/smoother
GSF/GSS	Gaussian sum filter/smoother
MCMC	Markov chain Monte Carlo
MH	Metropolis–Hasting
RWM	Random walk Metropolis
PDF	Probability density function
PMF	Probability mass function
FLQ	Finite level quantizer
ILQ	Infinite level quantizer
SYS	Systematic (resampling)
ML	Multinomial (resampling)
MT	Metropolis (resampling)
LS	Local selection (resampling)
GT	Ground truth
MSE	Mean square error

References

1. Gersho, A.; Gray, R.M. *Vector Quantization and Signal Compression*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2012; Voume 159,
2. Widrow, B.; Kollár, I. *Quantization Noise: Roundoff Error in Digital Computation, Signal Processing, Control, and Communications*; Cambridge University Press: Cambridge, UK, 2008.
3. Li, S.; Sauter, D.; Xu, B. Fault isolation filter for networked control system with event-triggered sampling scheme. *Sensors* **2011**, *11*, 557–572. [CrossRef]
4. Zhang, X.; Han, Q.; Ge, X.; Ding, D.; Ding, L.; Yue, D.; Peng, C. Networked control systems: A survey of trends and techniques. *IEEE/CAA J. Autom. Sin.* **2020**, *7*, 1–17. [CrossRef]
5. Zhang, L.; Liang, H.; Sun, Y.; Ahn, C.K. Adaptive Event-Triggered Fault Detection Scheme for Semi-Markovian Jump Systems With Output Quantization. *IEEE Trans. Syst. Man, Cybern. Syst.* **2021**, *51*, 2370–2381. [CrossRef]
6. Noshad, Z.; Javaid, N.; Saba, T.; Wadud, Z.; Saleem, M.Q.; Alzahrani, M.E.; Sheta, O.E. Fault Detection in Wireless Sensor Networks through the Random Forest Classifier. *Sensors* **2019**, *19*, 1568.
7. Huang, C.; Shen, B.; Zou, L.; Shen, Y. Event-Triggering State and Fault Estimation for a Class of Nonlinear Systems Subject to Sensor Saturations. *Sensors* **2021**, *21*, 1242. [CrossRef]
8. Liu, S.; Wang, Z.; Hu, J.; Wei, G. Protocol-based extended Kalman filtering with quantization effects: The Round-Robin case. *Int. J. Robust Nonlinear Control* **2020**, *30*, 7927–7946. [CrossRef]
9. Ding, D.; Han, Q.L.; Ge, X.; Wang, J. Secure State Estimation and Control of Cyber-Physical Systems: A Survey. *IEEE Trans. Syst. Man Cybern. Syst.* **2021**, *51*, 176–190. [CrossRef]
10. Wang, X.; Li, T.; Sun, S.; Corchado, J.M. A Survey of Recent Advances in Particle Filters and Remaining Challenges for Multitarget Tracking. *Sensors* **2017**, *17*, 2707. [CrossRef]
11. Curry, R.E. *Estimation and control with Quantized Measurements*; MIT Press: Cambridge, MA, USA, 1970.

12. Gustafsson, F.; Karlsson, R. Statistical results for system identification based on quantized observations. *Automatica* **2009**, *45*, 2794–2801. [CrossRef]
13. Wang, L.Y.; Yin, G.G.; Zhang, J.; Zhao, Y. *System identification with Quantized Observations*; Springer: Berlin/Heidelberg, Germany, 2010.
14. Marelli, D.E.; Godoy, B.I.; Goodwin, G.C. A scenario-based approach to parameter estimation in state-space models having quantized output data. In Proceedings of the 49th IEEE Conference on Decision and Control (CDC), Atlanta, GE, USA, 15–17 December 2010; pp. 2011–2016.
15. Rana, M.M.; Li, L. An Overview of Distributed Microgrid State Estimation and Control for Smart Grids. *Sensors* **2015**, *15*, 4302–4325. [CrossRef]
16. Särkkä, S. *Bayesian Filtering and Smoothing*; Cambridge University Press: Cambridge, UK, 2013; Vloume 3,
17. Anderson, B.D.O.; Moore, J.B. *Optimal Control: Linear Quadratic Methods*; Courier Corporation: Mineola, NY, USA, 2007.
18. Julier, S.J.; Uhlmann, J.K. New extension of the Kalman filter to nonlinear systems. In Proceedings of the Signal Processing, Sensor Fusion, and Target Recognition VI, Orlando, FL, USA, 21–24 April 1997; International Society for Optics and Photonics: Bellingham, WA, USA, 1997; Volume 3068, pp. 182–193.
19. Arasaratnam, I.; Haykin, S.; Elliott, R.J. Discrete-time nonlinear filtering algorithms using Gauss–Hermite quadrature. *Proc. IEEE* **2007**, *95*, 953–977. [CrossRef]
20. Sviestins, E.; Wigren, T. Nonlinear techniques for Mode C climb/descent rate estimation in ATC systems. *IEEE Trans. Control. Syst. Technol.* **2001**, *9*, 163–174. [CrossRef]
21. Gómez, J.C.; Sad, G.D. A State Observer from Multilevel Quantized Outputs. In Proceedings of the 2020 Argentine Conference on Automatic Control (AADECA), Buenos Aires, Argentin, 28–30 October 2020a; pp. 1–6.
22. Leong, A.S.; Dey, S.; Nair, G.N. Quantized Filtering Schemes for Multi-Sensor Linear State Estimation: Stability and Performance Under High Rate Quantization. *IEEE Trans. Signal Process.* **2013**, *61*, 3852–3865. [CrossRef]
23. Zhou, Y.; Li, J.; Wang, D. Unscented Kalman Filtering based quantized innovation fusion for target tracking in WSN with feedback. In Proceedings of the 2009 International Conference on Machine Learning and Cybernetics, Baoding, China, 12–15 July 2009; Volume 3, pp. 1457–1463.
24. Gordon, N.J.; Salmond, D.J.; Smith, A.F.M. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. In P*roceedings of the IEE Proceedings F (Radar and Signal Processing)*; IET: London, UK, 1993; Volume 140, pp. 107–113.
25. Doucet, A.; Godsill, S.; Andrieu, C. On sequential Monte Carlo sampling methods for Bayesian filtering. *Stat. Comput.* **2000**, *10*, 197–208. [CrossRef]
26. Li, T.; Bolic, M.; Djuric, P.M. Resampling Methods for Particle Filtering: Classification, implementation, and strategies. *IEEE Signal Process. Mag.* **2015**, *32*, 70–86. [CrossRef]
27. Douc, R.; Cappe, O. Comparison of resampling schemes for particle filtering. In Proceedings of the ISPA 2005, Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis, Zagreb, Croatia, 15–17 September, 2005; pp. 64–69.
28. Bi, H.; Ma, J.; Wang, F. An Improved Particle Filter Algorithm Based on Ensemble Kalman Filter and Markov Chain Monte Carlo Method. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2015**, *8*, 447–459. [CrossRef]
29. Zhai, Y.; Yeary, M. Implementing particle filters with Metropolis-Hastings algorithms. In Proceedings of the Region 5 Conference: Annual Technical and Leadership Workshop, Norman, OK, USA, 24 May 2004; pp. 149–152.
30. Sherlock, C.; Thiery, A.H.; Roberts, G.O.; Rosenthal, J.S. On the efficiency of pseudo-marginal random walk Metropolis algorithms. *Ann. Stat.* **2015**, *43*, 238–275. [CrossRef]
31. Cedeño, A.L.; Albornoz, R.; Carvajal, R.; Godoy, B.I.; Agüero, J.C. On Filtering Methods for State-Space Systems having Binary Output Measurements. *IFAC-PapersOnLine* **2021**, *54*, 815–820. [CrossRef]
32. Cedeño, A.L.; Albornoz, R.; Carvajal, R.; Godoy, B.I.; Agüero, J.C. A Two-Filter Approach for State Estimation Utilizing Quantized Output Data. *Sensors* **2021**, *21*, 7675. [CrossRef]
33. Cohen, H. *Numerical Approximation Methods*; Springer: Berlin/Heidelberg, Germany, 2011.
34. Agüero, J.C.; González, K.; Carvajal, R. EM-based identification of ARX systems having quantized output data. *IFAC-PapersOnLine* **2017**, *50*, 8367–8372. [CrossRef]
35. Ogata, K. *Modern Control Engineering*; Prentice Hall: Upper Saddle River, NJ, USA, 2010; Volume 5.
36. DeGroot, M.H. *Optimal Statistical Decisions*; Wiley Classics Library, Wiley: Hoboken, NJ, USA, 2005.
37. Solo, V. An EM algorithm for singular state space models. In Proceedings of the 42nd IEEE International Conference on Decision and Control (IEEE Cat. No.03CH37475), Maui, HI, USA, 9–12 December 2003; Volume 4, pp. 3457–3460.
38. Gelb, A.; Kasper, J.; Nash, R.; Price, C.; Sutherland, A. *Applied Optimal Estimation*; MIT Press: Cambridge, MA, USA, 1974.
39. Grewal, M.S.; Andrews, A.P. *Kalman Filtering: Theory and Practice with MATLAB*; John Wiley & Sons: Hoboken, NJ, USA, 2014.
40. Traoré, N.; Le Pourhiet, L.; Frelat, J.; Rolandone, F.; Meyer, B. Does interseismic strain localization near strike-slip faults result from boundary conditions or rheological structure? *Geophys. J. Int.* **2014**, *197*, 50–62. [CrossRef]
41. Wan, E.A.; Merwe, R.V.D. The unscented Kalman filter for nonlinear estimation. In Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No.00EX373), Lake Louise, AB, Canada, 4 October 2000; pp. 153–158.
42. Kitagawa, G. The two-filter formula for smoothing and an implementation of the Gaussian-sum smoother. *Ann. Inst. Stat. Math.* **1994**, *46*, 605–623. [CrossRef]

43. Balenzuela, M.P.; Dahlin, J.; Bartlett, N.; Wills, A.G.; Renton, C.; Ninness, B. Accurate Gaussian Mixture Model Smoothing using a Two-Filter Approach. In Proceedings of the 2018 IEEE Conference on Decision and Control (CDC), Miami Beach, FL, USA, 17–19 December 2018; pp. 694–699.
44. Liu, J.S.; Chen, R. Sequential Monte Carlo Methods for Dynamic Systems. *J. Am. Stat. Assoc.* **1998**, *93*, 1032–1044. [CrossRef]
45. Hostettler, R. A two filter particle smoother for Wiener state-space systems. In Proceedings of the 2015 IEEE Conference on Control and Applications, CCA 2015 - Proceedings, Dubrovnik, Croatia, 3–5 October 2012; pp. 412–417.
46. Arulampalam, M.S.; Maskell, S.; Gordon, N.; Clapp, T. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Trans. Signal Process.* **2002**, *50*, 174–188. [CrossRef]
47. Doucet, A.; de Freitas, N.; Gordon, N. *Sequential Monte Carlo Methods in Practice*; Springer: Berlin/Heidelberg, Germany, 2001.
48. Godsill, S.J.; Doucet, A.; West, M. Monte Carlo Smoothing for Nonlinear Time Series. *J. Am. Stat. Assoc.* **2004**, *99*, 156–168. [CrossRef]
49. Douc, R.; Garivier, A.; Moulines, E.; Olsson, J. Sequential Monte Carlo smoothing for general state space hidden Markov models. *Ann. Appl. Probab.* **2011**, *21*, 2109–2145. [CrossRef]
50. Wills, A.; Schön, T.B.; Ljung, L.; Ninness, B. Identification of Hammerstein–Wiener models. *Automatica* **2013**, *49*, 70–81. [CrossRef]