# Learning dynamical systems from quantized observations: a Bayesian perspective

Dario Piga, Manas Mejari, and Marco Forgione *†

May 1, 2021

**Abstract**

Identification of dynamical systems from low-resolution quantized data presents several challenges because of the limited amount of information available in the data and since proper algorithms have to be designed to handle the error due to quantization. In this paper, we consider identification of *Infinite Impulse Response* (IIR) models from quantized outputs. Algorithms both for maximum-likelihood estimation and Bayesian inference are developed. Finally, a particle-filter approach is presented for recursive reconstruction of the latent non-quantized outputs from past quantized observations.

## 1 Introduction

Although significant improvements in hardware speed, precision, and computational power have been seen in the last decades, only low-resolution quantized data are often available in some engineering applications. Indeed, bandwidth limitations and transmission costs represent current bottlenecks of modern computer and sensor networks, where a large number of users share the network's resources. This requires to transmit low-resolution quantized data within the network in order to reduce the transmission load. Furthermore, cheap and low-resolution quantized sensors (e.g., binary-valued sensors) are often used in real-time applications. Examples includes contact switches in robotic hands to detect the presence of an object in the sensing region, linear and angular encoders, binary chemical sensors to detect the presence of chemical compounds, and general event detectors, just to cite a few.

*This work was partially supported by the European H2020-CS2 project ADMITTED, Grant agreement no. GA832003. Corresponding author D. Piga.

†D. Piga, M. Mejari and M. Forgione are with IDSIA Dalle Molle Institute for Artificial Intelligence, USI-SUPSI, Polo universitario Lugano, Via la Santa 1, 6962 Lugano, Switzerland. (e-mail: {dario.piga, manas.mejari, marco.forgione}@supsi.ch).

## 1.1 Contributions on system identification from quantized data

Because of the limited information contained in low-resolution data, identification of dynamical systems from quantized data has several challenges and attracted the attention of many researchers [27]. Indeed, even in the case of simple *Linear Time-Invariant* (LTI) systems, conventional algorithms like least-squares may lead to poor estimate of the system parameters because of the bias introduced by the quantization noise [14,16]. In order to improve the parameter estimate, an approach based on *instrumental variables* and multi-step prediction is presented in [26], while in [1] quantization noise shaping techniques are developed to properly choose the quantization mechanism.

System identification from binary sensors is addressed in [17] for *Finite Impulse Response* (FIR) models under the (strong) assumption that that noise distribution is exactly known. Algorithms for identification of FIR model from binary measurements inputs and outputs have been recently presented in [18,24]. Both of these contributions are based on the assumption that the (hidden) input is normally distributed. This assumption is relaxed in [28], where parametrized noise distributions are considered. A two-step identification procedure is proposed, where first the hidden non-quantized output is reconstructed and then the model parameters are estimated using periodic input signals. A weighted least-squares approach providing consistent estimate for FIR model parameters is proposed in [10]. In [15], identification of FIR models under structural uncertainties (including deterministic unmodelled dynamics, nonlinear model mismatch, and sensor observation bias) is discussed, and upper and lower bounds on identification errors are derived. Extension to the identification of Wiener systems (linear FIR block followed by a parametric nonlinear static function) from binary-valued observations is discussed in [29]. Two algorithms for recursive identification of *Infinite Impulse Response* IIR models via binary sensors with adjustable threshold is addressed in [11]. The first algorithm proposed in [11] is based on an FIR approximation of the IIR model and requires a matrix inversion to reconstruct the parameters of the original IIR, while the second approach directly estimates the parameters of the original IIR model along with the (latent) output variables.

Several contributions have been also proposed in the more general case of learning from multi-level quantized data. Results on identification of both FIR and (IIR) models are given in [5–8,23] in the set-membership setting, where the unknown-but-bounded description of the quantization error is adopted. Contributions are also available in the stochastic framework. Maximum-likelihood estimation of *multi-input multi-output* FIR model parameters is addressed in [13]. This work is extended by the same authors in [12] for sparse identification of FIR models by formulating the estimation problem in a *Maximum-A-Posteriori* (MAP) framework, with a Laplace prior on the model parameters used to promote sparsity. A generalised expectation-maximisation algorithm is employed to solve the MAP problem, and *Gibbs sampling* is used to approximate expected values of truncated multivariate Gaussian distributions required for the com-

putation of conditional expectations. Generalized quantization schemes with noise-shaping filters are also used, and it is shown that these filters, if properly designed, can improve the accuracy of the parameter estimates. Other approaches are available in the literature for identification of Wiener [19] and Hammerstein systems [30].

## 1.2 Paper contributions and related literature

This paper is focused on identification of *Infinite Impulse Response* (IIR) models from multi-level quantized data. A Bayesian framework is adopted and the following problems are addressed:

- maximum-likelihood and maximum-a-posteriori estimation of the IIR model parameters, initial conditions and noise variance. Analytical expressions of the gradients w.r.t. the unknown variables are derived and used to perform optimization through a gradient-ascent algorithm;
- approximation of the posterior distribution of the unknown parameters through Laplace's method and *Markov Chain Monte Carlo* (MCMC);
- probabilistic inference of an output sequence given a new (not used for training) input sequence;
- filtering problem, which aims at iteratively reconstructing the latent non-quantized output from past observations of quantized outputs and based on a Bayesian estimate of the model. To this aim, a particle filter algorithm is constructed.

To position our work in the literature on system identification from quantized data, it is worth mentioning the contributions [4, 9, 25] which address identification of FIR systems from a Bayesian perspective. Specifically, the unknown impulse response is modelled as a realization of a Gaussian Process, with covariance matrix described by *stable spline kernels* [22]. Gibbs sampler is used in [4] to approximate the posterior distribution of the impulse response. However, this method requires sampling the unknown non-quantized output observations, and thus its computational burden increases fast with the size of the dataset. On the other hand, in our approach we only sample from the space of the IIR model parameters, initial conditions and noise variance. Thus, the dimension of the sampling space does not increase with the size of the dataset. In [9], an expectation-maximization algorithm is proposed to compute the MAP estimate of the FIR model parameters. However, only single-point estimates are discussed, while our approach addresses complete approximation of the posterior distribution of the model parameters and of new output sequences. In [25] a variational approximation of the likelihood and of the posterior distribution of the impulse response is constructed and then used to find (an approximation of) the maximum-likelihood and maximum-a posteriori estimator. Like in [9], only a single-point estimate is computed.

Maximum-likelihood estimation of ARMA models is considered in [21]. An online algorithm based on the quasi-Newton method is proposed and shown to asymptotically achieves the minimum parameter estimation error covariance. As in the previously cited papers, a frequentist approach is followed and a single-point estimate of the model parameters is provided.

## 1.3 Paper outline

The manuscript is organized as follows. The system identification problem from quantized data is formalized in Section 2. Maximum-likelihood estimation is discussed in Section 3. The Bayesian framework is introduced in Section 4, where algorithms for: MAP estimate; approximation of the posterior distributions of the unknown parameters; and system's output inference are presented. The filtering problem is discussed in Section 5. The effectiveness of the approach is shown in Section 6 through numerical examples. Conclusions and directions for future research are discussed in Section 7.

Derivations of the second-order derivates used to construct the Hessian of the log-posterior distribution are reported in the Appendix.

## 1.4 Notation

**General definitions** The sets of real and integer numbers are denoted by $\mathbb{R}$ and $\mathbb{Z}$, respectively. All vectors are assumed to be column vectors, unless stated differently. A superscript $\top$ denotes the transpose of a vector or a matrix. Thus, if $x$ is a column vector, $x^\top$ is a row vector. The determinant of a square matrix $A$ is denoted by $|A|$. The natural exponential function is denoted by $\exp(\cdot)$ and the Dirac's delta function centered in the point $a$ is denoted by $\delta_a(\cdot)$.

**Univariate Normal distribution** $\mathcal{N}(\mu, \sigma^2)$ denotes the univariate Normal distribution with mean $\mu \in \mathbb{R}$ and positive variance $\sigma^2 \in \mathbb{R}, \sigma^2 > 0$. The reciprocal $\tau = \frac{1}{\sigma^2}$ of the variance is referred to as *precision*. For a scalar real-valued variable $x \in \mathbb{R}$, the probability density function (pdf) of the univariate Normal distribution is defined by:

$$\mathcal{N}(x; \mu, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{\frac{1}{2}}} \exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2\right). \tag{1}$$

**Multivariate Normal distribution** $\mathcal{N}(\mu, \Sigma)$ denotes a $D$-dimensional multivariate Normal distribution with mean $\mu \in \mathbb{R}^D$ and positive definite covariance matrix $\Sigma \in \mathbb{R}^{D\times D}$. For a $D$-dimensional real-valued vector $x \in \mathbb{R}^D$, the pdf of the multivariate Normal distribution is defined by:

$$\mathcal{N}(x; \mu, \Sigma) = \frac{1}{(2\pi)^{\frac{D}{2}}|\Sigma|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)\right). \tag{2}$$

# 2 Problem setting

## 2.1 System description

Let us consider an LTI system $\mathcal{S}$ described by the difference equation

$$x(k) = \sum_{i=1}^{n_\mathrm{a}} a_i x(k-i) + \sum_{j=1}^{n_\mathrm{b}} b_j u(k-j), \tag{3}$$

where $x(k)$ and $u(k)$ are the system's output and input, respectively, at time $k \in \mathbb{Z}$; $n_\mathrm{a}$ and $n_\mathrm{b}$ are fixed positive integers defining the dynamical order of the system; and $a_i \in \mathbb{R}$ (with $i = 1, \ldots, n_\mathrm{a}$) and $b_j \in \mathbb{R}$ (with $j = 1, \ldots, n_\mathrm{b}$) are unknown parameters to be estimated from data. To compact the notation, we collect the parameters $a_i$ and $b_i$ in the vector $\theta \in \mathbb{R}^{n_\theta}$, with $n_\theta = n_\mathrm{a} + n_\mathrm{b}$ and $\theta = [a_1 \ \ldots \ a_{n_\mathrm{a}} \ b_1 \ \ldots \ b_{n_\mathrm{b}}]^\top$. The initial conditions of the output $x_0 = [x(-n_\mathrm{a}+1) \ \ \ldots \ \ x(0)]^\top \in \mathbb{R}^{n_\mathrm{a}}$ are assumed to be unknown.

The output $x(k)$ is corrupted by an additive noise $e(k)$, *i.e.*, $y(k) = x(k) + e(k)$, where $y(k)$ denotes the noisy output. The noise $e(k)$ is supposed to be generated by a white random process, and $e(k)$ is normally distributed with zero mean and unknown variance $\sigma_e^2$, *i.e.*, $e(k) \sim \mathcal{N}(0, \sigma_e^2)$. Furthermore, $e(k)$ is supposed to be statistically independent of the latent output $x(k)$ and of the system input $u(k)$.

We assume that the (noisy) output $y(k)$ is not directly observed, but only a *quantized* measurement $v(k)$ of $y(k)$ is available, namely:

$$v(k) = Q(y(k)), \tag{4}$$

where $Q(\cdot)$ is the quantizing operator, defined by

$$Q(y) = v \quad \text{if} \ \ y \in (q_v, \ q_{v+1}], \tag{5}$$

where $v \in \{1, \ldots, K\}$ (with $K$ being the number of quantization intervals), and $(q_i, \ q_{i+1}]$ defines the range of the $i$-th quantization interval, which is assumed to be known. Note that $q_1$ and $q_{K+1}$ can be equal to $-\infty$ and $+\infty$, respectively. A particular case is the binary quantizer, with $v \in \{1, 2\}$, $q_1 = -\infty$, $q_3 = \infty$, and $q_2$ is the quantizer threshold, *i.e.*,

$$Q(y) = \begin{cases} 1 & \text{if} \ \ y < q_2 \\ 2 & \text{if} \ \ y \geq q_2. \end{cases} \tag{6}$$

## 2.2 Addressed problems

Let $\mathcal{V}$ and $\mathcal{U}$ be the available datasets containing quantized output observations and inputs up to time $N$, respectively, namely $\mathcal{V} = \{v(k)\}_{k=1}^{N}$ and $\mathcal{U} = \{u(k)\}_{k=-n_b+1}^{N}$. The following learning problems will be addressed in the paper:

- *Maximum likelihood estimation* of the unknown model parameters $\theta$, initial conditions $x_0$, and noise standard deviation $\sigma_e$;

- *Bayesian inference*, with: $(i)$ maximum-a-posteriori estimate of the unknown variables $\theta$, $x_0$ and $\sigma_e$; $(ii)$ approximation of the posterior distribution $p(\theta, x_0, \sigma_e|\mathcal{V}, \mathcal{U})$; $(iii)$ inference of a new (not used for training) output sequence;
- *Filtering*, for recursive reconstruction of the latent output $x(k)$ given quantized output observations $v(k)$ up to time $k$.

# 3 Maximum Likelihood estimation

## 3.1 Likelihood definition

Because of the conditional independence of the observed quantized outputs $v(k)$ given the latent output $x(k)$, the likelihood function of the system parameters $\theta$, the initial conditions $x_0$ and the noise standard deviation $\sigma_e$ is given by

$$L(\theta, x_0, \sigma_e) = p(\mathcal{V}|\mathcal{U}; \theta, x_0, \sigma_e) \tag{7a}$$

$$= \prod_{k=1}^{N} p(v(k)|x(k), \sigma_e) = \prod_{k=1}^{N} p\left(Q\left(x(k) + e(k)\right) = v(k)\right) \tag{7b}$$

$$= \prod_{k=1}^{N} p\left(q_{v(k)} \leq x(k) + e(k) \leq q_{v(k)+1}\right) = \tag{7c}$$

$$= \prod_{k=1}^{N} \left(\Phi\left(\frac{q_{v(k)+1} - x(k)}{\sigma_e}\right) - \Phi\left(\frac{q_{v(k)} - x(k)}{\sigma_e}\right)\right), \tag{7d}$$

where $\Phi$ is the cumulative density function of the Normal distribution, *i.e.*, $\Phi(x) = \int_{w=-\infty}^{x} \mathcal{N}(w; 0, 1)dw$. Note that (7b) holds since the noise free output $x(k)$ is a determinist function of the model parameters $\theta$, the initial conditions $x_0$ and the input sequence $\mathcal{U}$.

## 3.2 Likelihood maximization

For maximum likelihood estimation, let us consider the *logarithm* of $L(\theta, x_0, \sigma_e)$:

$$\mathcal{L}(\theta, x_0, \sigma_e) = \log L(\theta, x_0, \sigma_e) =$$
$$\sum_{k=1}^{N} \log\left(\Phi\left(\frac{q_{v(k)+1} - x(k)}{\sigma_e}\right) - \Phi\left(\frac{q_{v(k)} - x(k)}{\sigma_e}\right)\right). \tag{8}$$

Maximization of the log-likelihood $\mathcal{L}(\theta, x_0, \sigma_e)$ is performed through a gradient-ascent approach, which requires to compute the gradients of $\mathcal{L}(\theta, x_0, \sigma_e)$ (at each iteration of the gradient-ascent algorithm) with respect to the unknown variables $\theta$, $x_0$, and $\sigma_e$.

### 3.2.1 Gradient with respect to the model parameters $\theta$

The gradient of the log-likelihood $\mathcal{L}$ w.r.t. the model parameters $\theta$ can be derived by simple rules of calculus and noticing that $\frac{d\Phi(x)}{dx} = \mathcal{N}(x; 0, 1)$:

$$\frac{\partial \mathcal{L}}{\partial \theta} = \sum_{k=1}^{N} \frac{\mathcal{N}\left(\frac{q_{v(k)+1} - x(k)}{\sigma_e}; 0, 1\right) - \mathcal{N}\left(\frac{q_{v(k)} - x(k)}{\sigma_e}; 0, 1\right)}{\Phi\left(\frac{q_{v(k)+1} - x(k)}{\sigma_e}\right) - \Phi\left(\frac{q_{v(k)} - x(k)}{\sigma_e}\right)} \times \frac{1}{\sigma_e}\left(-\frac{\partial x(k)}{\partial \theta}\right). \tag{9}$$

Note that evaluation of $\frac{\partial \mathcal{L}}{\partial \theta}$ requires to compute $\frac{\partial x(k)}{\partial \theta}$ for all $k = 1, \ldots, N$. Following the same approach commonly used in *prediction error methods* (PEM) [20, Ch. 10], the partial derivatives $\frac{\partial x(k)}{\partial a_i}$ (with $i = 1, \ldots, n_\mathrm{a}$) and $\frac{\partial x(k)}{\partial b_j}$ (with $j = 1, \ldots, n_\mathrm{b}$) can be computed by taking the derivative of left and right side of eq. (3) w.r.t. $a_i$ and $b_j$, which yields

$$\frac{\partial x(k)}{\partial a_i} = x(k-i) + \sum_{t=1}^{n_\mathrm{a}} a_t \frac{\partial x(k-t)}{\partial a_i}, \tag{10a}$$

and

$$\frac{\partial x(k)}{\partial b_j} = \sum_{i=1}^{n_\mathrm{a}} a_i \frac{\partial x(k-i)}{\partial b_j} + u(k-j). \tag{10b}$$

Thus, $\frac{\partial x(k)}{\partial a_i}$ and $\frac{\partial x(k)}{\partial b_j}$ can be computed by simulating the difference equation (10a) and (10b), respectively. Note that the term $x(k)$ appearing in (9) and (10a) has to be computed at each iteration of the gradient-ascent algorithm by simulating the dynamical model (3) using the current value of the model parameters $\theta$ and of the initial conditions $x_0$.

### 3.2.2 Gradient with respect to the initial conditions $x_0$

The gradient of the log-likelihood $\mathcal{L}$ w.r.t. the initial conditions $x_0$ is given by

$$\frac{\partial \mathcal{L}}{\partial x_0} = \sum_{k=1}^{N} \frac{\mathcal{N}\left(\frac{q_{v(k)+1} - x(k)}{\sigma_e}; 0, 1\right) - \mathcal{N}\left(\frac{q_{v(k)} - x(k)}{\sigma_e}; 0, 1\right)}{\Phi\left(\frac{q_{v(k)+1} - x(k)}{\sigma_e}\right) - \Phi\left(\frac{q_{v(k)} - x(k)}{\sigma_e}\right)} \times \frac{1}{\sigma_e}\left(-\frac{\partial x(k)}{\partial x_0}\right) \tag{11}$$

The gradient $\frac{\partial x(k)}{\partial x_0}$ appearing in (11) is derived by differentiation the left and right hand side of (3):

$$\frac{\partial x(k)}{\partial x_0} = \sum_{i=1}^{n_\mathrm{a}} a_i \frac{\partial x(k-i)}{\partial x_0}. \tag{12}$$

Thus, $\frac{\partial x(k)}{\partial x_0}$ can be computed by simulating the autonomous dynamical system (12), with initial conditions

$$\frac{\partial x(1-j)}{\partial x_0} = \mathbf{e}_{n_\mathrm{a}+1-j}, \quad j = 1, \ldots, n_\mathrm{a} \tag{13}$$

where $\mathbf{e}_j$ is the canonical vector, which has value zero for all elements except for index $j$ which takes value 1. It is worth pointing out that (13) holds if we do not consider the inter-dependence within the elements of the initial conditions' vector $x_0$.

### 3.2.3 Derivative with respect to the noise standard deviation $\sigma_e$

The partial derivative of the log-likelihood w.r.t. the noise standard deviation $\sigma_e$ is given by:

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \sigma_e} = & \sum_{k=1}^{N} \left( \frac{-\mathcal{N}\left(\frac{q_{v(k)+1}-x(k)}{\sigma_e}; 0, 1\right)\left(q_{v(k)+1} - x(k)\right)}{\sigma_e^2} \right. \\ & \left. - \frac{-\mathcal{N}\left(\frac{q_{v(k)}-x(k)}{\sigma_e}; 0, 1\right)\left(q_{v(k)} - x(k)\right)}{\sigma_e^2} \right) \times \\ & \frac{1}{\Phi\left(\frac{q_{v(k)+1}-x(k)}{\sigma_e}\right) - \Phi\left(\frac{q_{v(k)}-x(k)}{\sigma_e}\right)}. \end{aligned} \tag{14}$$

Note that the terms $\frac{\partial \mathcal{L}}{\partial \theta}$ (eq. (9)), $\frac{\partial \mathcal{L}}{\partial x_0}$ (eq. (11)) and $\frac{\partial \mathcal{L}}{\partial \sigma_e}$ (eq. (14)) have several factors in common that can be evaluated only once at each iteration to reduce the computational burden of the gradient-ascent algorithm.

# 4 Bayesian inference

In this section, we address the estimation of $\theta$, $x_0$ and $\sigma_e$ in a Bayesian setting, and thus we aim at computing the posterior probability distribution $p(\theta, x_0, \sigma_e|\mathcal{V},\mathcal{U})$. In order to simplify the derivations and to take into account the constraint $\sigma_e > 0$, the distribution of the noise $e(k)$ is parametrized in terms of the logarithm $\bar{\tau}$ of the noise precision, *i.e.*, $\bar{\tau} = \log \sigma_e^{-2}$. Thus, in the following, our distribution of interest is $p(\theta, x_0, \bar{\tau}|\mathcal{V},\mathcal{U})$.

As a prior distribution for $\theta$, $x_0$ and $\bar{\tau}$, we assume an isotropic Gaussian with zero mean and covariance matrix $\lambda^2 I$, *i.e.*,

$$p(\theta, x_0, \bar{\tau}) = \mathcal{N}([\theta^\top \;\; x_0^\top \;\; \bar{\tau}]^\top; 0, \lambda^2 I) \tag{15}$$

where $\lambda$ is a positive hyper-parameter and $I$ is the identity matrix of size $(n_\theta + n_a + 1)$.

From Bayes' rule, the posterior probability distribution of the unknown parameters $\theta$, $x_0$ and $\bar{\tau}$ is given by:

$$p(\theta, x_0, \bar{\tau}|\mathcal{V},\mathcal{U}) = \frac{p(\mathcal{V}|\mathcal{U},\theta,x_0,\bar{\tau})p(\theta,x_0,\bar{\tau})}{p(\mathcal{V}|\mathcal{U})}. \tag{16}$$

## 4.1 Maximum-a-posteriori estimation

We first focus on the computation of the *maximum-a-posteriori* (MAP) estimate of $\theta$, $x_0$ and $\bar{\tau}$, which is by definition the mode of the posterior distribution $p(\theta, x_0, \bar{\tau}|\mathcal{V},\mathcal{U})$. As in the case of maximum likelihood estimation discussed in Section 3, a gradient-ascent approach is used to maximize (w.r.t. $\theta$, $x_0$ and $\bar{\tau}$) the logarithm of the posterior (16), which is given by:

$$\begin{aligned}\log p(\theta, x_0, \bar{\tau}|\mathcal{V},\mathcal{U}) = \mathcal{L}(\theta, x_0, \bar{\tau}) + \log p(\theta, x_0, \bar{\tau}) \\ - \log p(\mathcal{V}|U).\end{aligned} \tag{17}$$

Note that the term $\log p(\mathcal{V}|U)$ in the previous equation does not depend on $\theta, x_0, \bar{\tau}$ and thus it is ignored in the optimization. The gradient of the log-likelihood $\mathcal{L}(\theta, x_0, \bar{\tau})$ is provided in Section 3. More precisely, eq. (14) actually provides the derivative of $\mathcal{L}(\theta, x_0, \bar{\tau})$ w.r.t. the noise standard deviation $\sigma_e$. Nevertheless, the derivative of $\mathcal{L}(\theta, x_0, \bar{\tau})$ w.r.t. $\bar{\tau}$ can be easily computed from (14), and it is equal to

$$\frac{\partial \mathcal{L}}{\partial \bar{\tau}} = \frac{\partial \mathcal{L}}{\partial \sigma_e}\frac{\partial \sigma_e}{\partial \bar{\tau}} = -\frac{1}{2}\frac{\partial \mathcal{L}}{\partial \sigma_e}\frac{1}{\sqrt{\exp(\bar{\tau})}} \tag{18}$$

As for the log-prior $\log p(\theta, x_0, \bar{\tau})$, it is equal to

$$\log p(\theta, x_0, \bar{\tau}) = -\frac{\lambda^{-2}}{2}[\theta^\top \;\; x_0^\top \;\; \bar{\tau}] \begin{bmatrix}\theta \\ x_0 \\ \bar{\tau}\end{bmatrix} + \text{const}, \tag{19}$$

where "const" is a quantity independent of $\theta, x_0, \bar{\tau}$. Then, the gradient of the log-prior can be easily computed and is given by:

$$\frac{\partial \log p(\theta, x_0, \bar{\tau})}{\partial [\theta^\top \;\; x_0^\top \;\; \bar{\tau}]} = -\lambda^{-2} \begin{bmatrix}\theta \\ x_0 \\ \bar{\tau}\end{bmatrix}. \tag{20}$$

Since the gradients of all the terms in (17) are available, the MAP estimate of the variables $\theta$, $x_0$ and $\bar{\tau}$ can be computed using a gradient-ascent approach. This estimate will be denoted as $\theta^{\text{MAP}}, x_0^{\text{MAP}}, \bar{\tau}^{\text{MAP}}$.

## 4.2 Posterior approximation

In the following, we present two possible approximations for the whole posterior $p(\theta, x_0, \bar{\tau}|\mathcal{V},\mathcal{U})$, which is in general a complex and intractable probability distribution.

The first approach is based on Laplace's method [3, Ch. 4.4] and leads to a Gaussian approximation of the intractable distribution $p(\theta, x_0, \bar{\tau}|\mathcal{V},\mathcal{U})$. The second approach is based on Monte Carlo sampling techniques and leads to a point-mass approximation of the same distribution.

#### 4.2.1 Version 1: Laplace approximation

Laplace's method is used to approximate the posterior $p(\theta, x_0, \bar{\tau}|\mathcal{V},\mathcal{U})$ as a Gaussian distribution. To this aim, the following second-order Taylor expansion of the logarithm of the numerator in (16) centred around its peak $(\theta^{\text{MAP}}, x_0^{\text{MAP}}, \bar{\tau}^{\text{MAP}})$ is considered

$$\begin{aligned}
&\log\left(p(\mathcal{V}|\mathcal{U},\theta,x_0,\bar{\tau})p(\theta,x_0,\bar{\tau})\right) \approx \\
&\log\left(p(\mathcal{V}|\mathcal{U},\theta^{\text{MAP}},x_0^{\text{MAP}},\bar{\tau}^{\text{MAP}})p(\theta^{\text{MAP}},x_0^{\text{MAP}},\bar{\tau}^{\text{MAP}})\right)+ \\
&-\frac{1}{2}[(\theta-\theta^{\text{MAP}})^\top\ (x_0-x_0^{\text{MAP}})^\top\ (\bar{\tau}-\bar{\tau}^{\text{MAP}})]\Lambda\begin{bmatrix}\theta-\theta^{\text{MAP}}\\ x_0-x_0^{\text{MAP}}\\ \bar{\tau}-\bar{\tau}^{\text{MAP}}\end{bmatrix}
\end{aligned} \tag{21}$$

where $\Lambda$ is the Hessian matrix of (minus) the log-posterior $-\log p(\theta, x_0, \bar{\tau}|\mathcal{V},\mathcal{U})$ evaluated at $\theta^{\text{MAP}}, x_0^{\text{MAP}}, \bar{\tau}^{\text{MAP}}$, whose entries contain the second derivatives of (minus) the log-likelihood $-\mathcal{L}(\theta, x_0, \bar{\tau})$ and of (minus) the log-prior $-\log(p(\theta, x_0, \bar{\tau}))$ w.r.t. $\theta$, $x_0$, and $\bar{\tau}$. Their analytical expressions are provided in Appendix A. We remark that first-order term in the Taylor expansion (21) does not appear since $\theta^{\text{MAP}}, x_0^{\text{MAP}}, \bar{\tau}^{\text{MAP}}$ is a stationary point of $p(\mathcal{V}|\mathcal{U},\theta,x_0,\bar{\tau})p(\theta,x_0,\bar{\tau})$.

Laplace's approximation of the posterior $p(\theta, x_0, \bar{\tau}|\mathcal{V},\mathcal{U})$ is obtained by taking the exponent of the Taylor expansion in (21) and normalizing it. This leads to the Gaussian approximation

$$\begin{aligned}
&p(\theta, x_0, \bar{\tau}|\mathcal{V},\mathcal{U}) \approx p_{LAP}(\theta, x_0, \bar{\tau}|\mathcal{V},\mathcal{U}) = \\
&=\frac{|\Lambda|^{\frac{1}{2}}}{(2\pi)^{\frac{n_\theta+n_a+1}{2}}}\exp\left(-\frac{1}{2}\begin{bmatrix}\theta-\theta^{\text{MAP}}\\ x_0-x_0^{\text{MAP}}\\ \bar{\tau}-\bar{\tau}^{\text{MAP}}\end{bmatrix}^\top\Lambda\begin{bmatrix}\theta-\theta^{\text{MAP}}\\ x_0-x_0^{\text{MAP}}\\ \bar{\tau}-\bar{\tau}^{\text{MAP}}\end{bmatrix}\right).
\end{aligned} \tag{22}$$

#### 4.2.2 Version 2: MCMC approximation

In this paragraph, we discuss approximations of the entire posterior probability distribution $p(\theta, x_0, \bar{\tau}|\mathcal{V},\mathcal{U})$ using *Markov Chain Monte-Carlo* (MCMC) algorithms, which attempt to simulate draws from a complex distribution of interest [2].

The main idea behind application of MCMC to the considered inference problem consists in generating a sequence of $H$ random samples $\theta[h], x_0[h], \bar{\tau}[h]$, $h = 1, 2, \ldots, H$, from an irreducible and aperiodic Markov chain whose stationary distribution is the target posterior $p(\theta, x_0, \bar{\tau}|\mathcal{V},\mathcal{U})$. Once samples are

generated, the posterior is approximated by the empirical point-mass distribution

$$p(\theta, x_0, \bar{\tau}|\mathcal{V},\mathcal{U}) \approx \frac{1}{H}\sum_{h=1}^{H} \delta_{(\theta[h],x_0[h],\bar{\tau}[h])}(\theta, x_0, \bar{\tau}). \tag{23}$$

The *independent sampler*, an instance of the well known Metropolis-Hastings MCMC method [2], is used to draw samples from $p(\theta, x_0, \bar{\tau}|\mathcal{V},\mathcal{U})$, as outlined in Algorithm 1. The algorithm simulates a Markov Chain with stationary distribution $p(\theta, x_0, \bar{\tau}|\mathcal{V},\mathcal{U})$, and it requires to specify: an initial sample $\theta[0], x_0[0], \bar{\tau}[0]$; a proposal distribution $q(\theta, x_0, \bar{\tau})$; the length $H$ of the Markov Chain (namely, the number of iterations).

At each iteration $h$, a candidate sample $\theta^*, x_0^*, \bar{\tau}^*$ is drawn from the distribution $q(\theta, x_0, \bar{\tau})$ (Step 2) and accepted with probability $\mathcal{A}(\theta^*, x_0^*, \bar{\tau}^*, \theta[h], x_0[h], \bar{\tau}[h])$ (Steps 3 and 4). It is worth noticing that acceptance probability $\mathcal{A}(\theta^*, x_0^*, \bar{\tau}^*, \theta[h], x_0[h], \bar{\tau}[h])$ (eq. (24)) depends on the ratio $\frac{p(\theta^*,x_0^*,\bar{\tau}^*|\mathcal{V},\mathcal{U})}{p(\theta[h],x_0[h],\bar{\tau}[h]|\mathcal{V},\mathcal{U})}$ and thus its evaluation does not require to compute the normalization constant $p(\mathcal{V}|\mathcal{U})$ in (16). Furthermore, for numerical reasons, it is often more convenient to compute the logarithm of this ratio. If the candidate sample $\theta^*, x_0^*, \bar{\tau}^*$ is accepted, then $\theta[h+1], x_0[h+1], \bar{\tau}[h+1]$ is set to $\theta^*, x_0^*, \bar{\tau}^*$ (Step 6), otherwise $\theta[h+1], x_0[h+1], \bar{\tau}[h+1]$ is set to the previous sample $\theta[h], x_0[h], \bar{\tau}[h]$ (Step 8). The output is the sequence of samples $\{\theta[h], x_0[h], \bar{\tau}[h]\}_{h=1}^{H}$ generated during the execution of the algorithm.

It is well known that the choice of the proposal distribution $q(\theta, x_0, \bar{\tau})$ is crucial in any Metropolis-Hastings MCMC algorithm to rapid convergence to the target distribution. The proposal density leading to fastest convergence is obviously $q(\theta, x_0, \bar{\tau}) = p(\theta, x_0, \bar{\tau}|\mathcal{V},\mathcal{U})$. Unfortunately, $p(\theta, x_0, \bar{\tau}|\mathcal{V},\mathcal{U})$ cannot be directly sampled, as it represents itself the target distribution to be approximated. Based on the above considerations, a reasonable proposal distribution is the Laplace approximation $p_{LAP}(\theta, x_0, \bar{\tau}|\mathcal{V},\mathcal{U})$ derived in Section 4.2.1.

**Remark 1** *Since in general it is not possible to know a-priori the number of iterations needed by the Markov chain to reach its equilibrium distribution, it is a common practice to discard an initial set of samples in order to reduce the effect of the transient in the simulation of the Markov chain. This practice is known as* burn-in *strategy [2].*

## 4.3 Inference of system's output sequence

In this section, we discuss the estimation of the system's output $x^\star(k)$ given a new (not used for training) sequence of inputs $\mathcal{U}_{k-1}^\star = \{u^\star(t)\}_{t=-n_b+1}^{k-1}$ up to time $k-1$. Formally, we would like to compute the probability distribution

**Algorithm 1** Approximation of the posterior distribution $p(\theta, x_0, \bar{\tau}|\mathcal{V},\mathcal{U})$ through independent sampler MCMC.

Inputs: initial samples $\theta[0], x_0[0], \bar{\tau}[0]$; proposal distribution $q(\theta, x_0, \bar{\tau})$; data samples $\mathcal{V}$ and $\mathcal{U}$.

1: **for** $h = 0, \ldots, H-1$ **do**
2: draw candidate sample $\theta^*, x_0^*, \bar{\tau}^*$ from proposal $q(\theta, x_0, \bar{\tau})$;
3: **set** acceptance probability

$$\begin{aligned}&\mathcal{A}(\theta^*, x_0^*, \bar{\tau}^*, \theta[h], x_0[h], \bar{\tau}[h]) \leftarrow \\ &\min\left\{1, \tfrac{p(\theta^*, x_0^*, \bar{\tau}^*|\mathcal{V},\mathcal{U}) q(\theta[h], x_0[h], \bar{\tau}[h])}{p(\theta[h], x_0[h], \bar{\tau}[h]|\mathcal{V},\mathcal{U}) q(\theta^*, x_0^*, \bar{\tau}^*)}\right\};\end{aligned} \tag{24}$$

4: **accept** candidate $\theta^*, x_0^*, \bar{\tau}^*$ with probability $\mathcal{A}(\theta^*, x_0^*, \bar{\tau}^*, \theta[h], x_0[h], \bar{\tau}[h])$;
5: **if** the $\theta^*, x_0^*, \bar{\tau}^*$ is accepted **then**
6: $\theta[h+1], x_0[h+1], \bar{\tau}[h+1] \leftarrow \theta^*, x_0^*, \bar{\tau}^*$;
7: **else**
8: $\theta[h+1], x_0[h+1], \bar{\tau}[h+1] \leftarrow \theta[h], x_0[h], \bar{\tau}[h]$;
9: **end if**
10: **end for**

Output: Samples $\{\theta[h], x_0[h], \bar{\tau}[h]\}_{h=1}^{H}$.

$p(x^\star(k)|\mathcal{V},\mathcal{U},\mathcal{U}_{k-1}^\star)$, which is factorized as

$$\begin{aligned}&p(x^\star(k)|\mathcal{V},\mathcal{U},\mathcal{U}_{k-1}^\star) \\ &= \int \Big(\underbrace{\int p(x^*(k)|\theta,\mathcal{U}_{k-1}^\star, x_0^\star) p(x_0^\star) dx_0^\star}_{p(x^\star(k)|\theta,\mathcal{U}_{k-1}^\star)}\Big) p(\theta|\mathcal{V},\mathcal{U}) d\theta,\end{aligned} \tag{25}$$

where $p(x_0^\star)$ represents the prior distribution of the initial conditions $x_0^\star = [x^\star(0), \ldots, x^\star(-n_\mathrm{a}+1)]^\top$ of the new sequence, which is assumed to be Gaussian with mean $\bar{x}_0^\star$ and covariance matrix $\mathrm{Cov}(x_0^\star)$.

Note that the integral $\int p(x^*(k)|\theta,\mathcal{U}_{k-1}^\star, x_0^\star) p(x_0^\star) dx_0^\star = p(x^*(k)|\theta,\mathcal{U}_{k-1}^\star)$ represents the probability distribution of the output of the LTI system (3) for a given input sequence $\mathcal{U}_{k-1}^\star$, given parameters $\theta$ and Gaussian-distributed initial conditions $x_0^\star$. In order to compute this integral, it is convenient to split the system's response $x^\star(k)$ into a *forced* response term $x_F^\star(k)$ driven by the (deterministic) input $\mathcal{U}_{k-1}^\star$ and a *natural* response term $x_N^\star(k)$ due to the (stochastic)

initial condition $x_0^\star$:

$$x_F^\star(k) = \sum_{i=1}^{n_\mathrm{a}} a_i x_F^\star(k-i) + \sum_{j=1}^{n_\mathrm{b}} b_j u^\star(k-j)$$
$$\text{with } x_F^\star(k) = 0 \text{ for } k \leq 0 \tag{26a}$$
$$x_N^\star(k) = \sum_{i=1}^{n_\mathrm{a}} a_i x_N^\star(k-i)$$
$$\text{with } x_N^\star(-k) = \bar{x}_0^\star[k+1] \text{ for } k \leq 0 \tag{26b}$$
$$x^\star(k) = x_F^\star(k) + x_N^\star(k). \tag{26c}$$

For a given value of $\theta$, the forced response term $x_F^\star(k)$ in (26a) is deterministic, since the input sequence $\mathcal{U}_{k-1}^\star$ is known. Conversely, the natural response term $x_N^\star(k)$ in (26b) is stochastic since the initial condition $x_0^\star$ is a random variable. Specifically, $x_N^\star(k)$ may be written as

$$x_N^\star(k) = EA^k x_0^\star, \tag{27}$$

where

$$A = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 \\ 0 & 0 & 0 & \ddots & \vdots & \vdots \\ \vdots & \vdots & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ a_{n_a} & a_{n_a-1} & a_{n_a-2} & \dots & a_2 & a_1 \end{bmatrix}, \tag{28a}$$

$$E = \begin{bmatrix} 0 & 0 & \dots & 0 & 1 \end{bmatrix}. \tag{28b}$$

Therefore, the probability distribution of $x^\star(k)$ given $\theta$ is also Gaussian:

$$p(x^\star(k)|\theta, \mathcal{U}_{k-1}^\star) =$$
$$\mathcal{N}\left(x^\star(k); x_F^\star(k;\theta) + EA^k\bar{x}_0^\star, EA^k\mathrm{Cov}(x_0^\star)\left(A^k\right)^\top E^\top\right). \tag{29}$$

Given the explicit expression of $p(x^\star(k)|\mathcal{U}_k^\star, \theta)$ in (29), the integral $\int p(x^\star(k)|\mathcal{U}_k^\star, \theta)p(\theta|\mathcal{V}, \mathcal{U})d\theta$ in (25) is approximated via Monte-Carlo sampling, either generating samples $\{\theta[h]\}_{h=1}^H$ from the marginal (over $x_0$ and $\bar{\tau}$) of the Laplace approximation $p_{LAP}(\theta, x_0, \bar{\tau}|\mathcal{V}, \mathcal{U})$ (eq. (22)) or using the samples already drawn through the MCMC scheme (Algorithm 1). The final distribution $p(x^\star(k)|\mathcal{V}, \mathcal{U}, \mathcal{U}_{k-1}^\star)$ is then

approximated by a mixture of Gaussian distributions, *i.e.*,

$$p(x^\star(k)|\mathcal{V},\mathcal{U},\mathcal{U}^\star_{k-1}) = \frac{1}{H}\sum_{h=1}^{H} p(x^*(k)|\theta[h],\mathcal{U}^\star_{k-1}) = \frac{1}{H}\sum_{h=1}^{H}\mathcal{N}\Big(x^\star(k); x^\star_F(k;\theta[h]) + EA^k[h]\bar{x}^\star_0, EA^k[h]\mathrm{Cov}(x^\star_0)\big(A^k[h]\big)^\top E^\top\Big), \tag{30}$$

where $A^k[h]$ denotes the value of the matrix $A^k$ computed for $\theta = \theta[h]$.

# 5 Filtering

In this section, we consider the problem of iteratively reconstructing the non-quantized output $x^\star(k)$ at time $k$ given:

- past estimate $x^\star(k-1)$ at time $k-1$ inferred from new sequences (not used for training) of quantized outputs $\mathcal{V}^\star_{k-1}$ and inputs $\mathcal{U}^\star_{k-1}$ up to time $k-1$;
- new input $u^\star(k)$ and quantized output $v^\star(k)$;
- uncertain model parameters $\theta$ and the log of the noise precision $\bar{\tau}$ described by the posterior distribution $p(\theta,\bar{\tau}|\mathcal{V},\mathcal{U})$, obtained by marginalizing either the Laplace in (22) or the MCMC approximation in (23) w.r.t. the initial condition $x_0$.

Specifically, we aim at recursively approximating the probability distribution $p(x^\star(k)|\mathcal{V}^\star_k,\mathcal{U}^\star_k,\mathcal{V},\mathcal{U})$ from $p(x^\star(k-1)|\mathcal{V}^\star_{k-1},\mathcal{U}^\star_{k-1},\mathcal{V},\mathcal{U})$. In this work, we do not exploit the new sequences $\mathcal{V}^\star_k$ and $\mathcal{U}^\star_k$ to update the probability distribution of $\theta$ and $\bar{\tau}$. Formally, we approximate $p(x^\star(k)|\mathcal{V}^\star_k,\mathcal{U}^\star_k,\mathcal{V},\mathcal{U})$ as follows:

$$\begin{aligned} & p(x^\star(k)|\mathcal{V}^\star_k,\mathcal{U}^\star_k,\mathcal{V},\mathcal{U}) \\ &= \int p(x^\star(k)|\theta,\bar{\tau},\mathcal{V}^\star_k,\mathcal{U}^\star_k,\mathcal{V},\mathcal{U})p(\theta,\bar{\tau}|\mathcal{V}^\star_k,\mathcal{U}^\star_k,\mathcal{V},\mathcal{U})d\theta d\bar{\tau} \\ &= \int p(x^\star(k)|\theta,\bar{\tau},\mathcal{V}^\star_k,\mathcal{U}^\star_k)p(\theta,\bar{\tau}|\mathcal{V}^\star_k,\mathcal{U}^\star_k,\mathcal{V},\mathcal{U})d\theta d\bar{\tau} \\ &\approx \int p(x^\star(k)|\theta,\bar{\tau},\mathcal{V}^\star_k,\mathcal{U}^\star_k)p(\theta,\bar{\tau}|\mathcal{V},\mathcal{U})d\theta d\bar{\tau}. \end{aligned} \tag{31}$$

The above integral is then approximated through Monte-Carlo sampling as

follows:

$$\int p(x^\star(k)|\theta,\bar{\tau},\mathcal{V}_k^\star,\mathcal{U}_k^\star)p(\theta,\bar{\tau}|\mathcal{V},\mathcal{U})d\theta d\bar{\tau}$$
$$\approx\frac{1}{H}\sum_{h=1}^{H}p(x^\star(k)|\theta[h],\bar{\tau}[h],\mathcal{V}_k^\star,\mathcal{U}_k^\star) \tag{32}$$

where $\theta[h]$, $\bar{\tau}[h]$ are either drawn from the Laplace approximation of $p(\theta,\bar{\tau}|\mathcal{V},\mathcal{U})$ or already available from the MCMC approximation (23).

The probability distribution (32) is approximated using a bank of particle filters. To this aim, we represent the system (3) in the controllable canonical state-space form:

$$\tilde{x}(k) =A\tilde{x}(k-1)+Bu(k-1) \tag{33a}$$
$$x(k) =C\tilde{x}(k) \tag{33b}$$

with $B=\begin{bmatrix}0 & \cdots & 0 & 1\end{bmatrix}^\top$, $C=\begin{bmatrix}b_{n_\mathrm{b}} & \cdots & b_2 & b_1\end{bmatrix}$, and $A$ in (28a).[1]

Based on the above state-space representation, the evolutionary equation of $\tilde{x}^\star(k)$ for a given model parameter $\theta[h]$ and the related non-linear output equation are given by:

$$\tilde{x}^\star(k) =A[h]\tilde{x}^\star(k-1)+Bu^\star(k)+\eta(k) \tag{34a}$$
$$v^\star(k) =Q(C[h]\tilde{x}^\star(k)+e^\star(k)) \tag{34b}$$

where $A[h]$ and $C[h]$ are the matrices in (33) defined for the sample $\theta[h]$, and $\eta(k)\in\mathbb{R}^{n_\mathrm{a}}$ is a fictitious white process noise (described by a probability distribution $p(\eta)$) which is introduced to take into account a possible model mismatch between (34a) and the underlying dynamics (33a).

Using particle filter algorithms, the probability distribution $p(\tilde{x}^\star(k)|\theta[h],\bar{\tau}[h],\mathcal{V}_k^\star,\mathcal{U}_k^\star)$ in (32) is iteratively approximated with the empirical point-mass distributions

$$p(\tilde{x}^\star(k)|\theta[h],\bar{\tau}[h],\mathcal{V}_k^\star,\mathcal{U}_k^\star)\approx\sum_{l=1}^{N_p}w_{h,l}(k)\delta_{\tilde{x}_{h,l}^\star(k)}(\tilde{x}^\star(k)), \tag{35}$$

based on $p(\tilde{x}^\star(k-1)|\theta[h],\bar{\tau}[h],\mathcal{V}_{k-1}^\star,\mathcal{U}_{k-1}^\star)$ and current values of $v^\star(k)$ and $u^\star(k)$. In (35), $N_p$ is the number of particles; $\tilde{x}_{h,l}^\star(k)$ is the position of the $l$-th particle at time $k$ associated to the sample $\theta[h]$ and $\bar{\tau}[h]$; and $w_{h,l}$ are non-negative weights associated to the particle, with $\sum_{l=1}^{N_p}w_{h,l}=1$. The distribution of interest $p(x^\star(k)|\theta[h],\bar{\tau}[h],\mathcal{V}_k^\star,\mathcal{U}_k^\star)$ can be easily obtained from (35) as

$$p(x^\star(k)|\theta[h],\bar{\tau}[h],\mathcal{V}_k^\star,\mathcal{U}_k^\star)\approx\sum_{l=1}^{N_p}w_{h,l}(k)\delta_{x_{h,l}^\star(k)}(x^\star(k)), \tag{36}$$

with $x_{h,l}^\star(k)=C[h]\tilde{x}_{h,l}^\star(k)$. To avoid notational clutter, the index $h$ is omitted from $\tilde{x}_{h,l}^\star(k)$ and $w_{h,l}$ in the following.

[1] The considered controllable canonical state-space form holds for $n_\mathrm{a}=n_\mathrm{b}$.

## 5.1 Particle filter algorithm

For every sample $\theta[h]$ and $\bar{\tau}[h]$ (or equivalently $\sigma_e[h]$), the particles $\{\tilde{x}_l^\star(k)\}_{l=1}^{N_p}$ and the corresponding weights $\{w_l\}_{l=1}^{N_p}$ are computed recursively at each time step $k$ using the particle filter algorithm outlined in Algorithm 2. From line 2 to line 6, $N_p$ particles are sampled either from the prior distribution of the initial conditions $\tilde{x}^\star(0)$ (if $k = 1$) or from the approximated probability distribution of $\tilde{x}_l^\star(k-1)$ (if $k > 1$) obtained from the previous iteration of the algorithm.

At line 8, the particle's position $\tilde{x}_l^\star(k)$ of the $l$-th particle at time $k$ is updated based on the previous position of the same particle by sampling the stochastic model dynamics (34a). Then, the particle's weights are updated (line 9) according to the conditional likelihood $p(v^\star(k)|\tilde{x}_l^\star(k))$ and normalized (line 11). Given particles' position and corresponding weights, the probability distributions $p(\tilde{x}^\star(k)|\theta[h], \bar{\tau}[h], \mathcal{V}_k^\star, \mathcal{U}_k^\star)$ and $p(x^\star(k)|\theta[h], \bar{\tau}[h], \mathcal{V}_k^\star, \mathcal{U}_k^\star)$ are approximated as in (35) and (36), respectively.

**Algorithm 2** Iterative update of particles $\{\tilde{x}_l^\star(k)\}_{l=1}^{N_p}$.

1: **for** $k = 1, 2, \ldots$ **do**
2: **if** $k = 1$ **then**
3: **sample** $N_p$ particles of the initial conditions $\{\tilde{x}_l^\star(0)\}_{l=1}^{N_p}$ from a prior $p(\tilde{x}^\star(0))$
4: **else**
5: **resample** particles $\{\tilde{x}_l^\star(k-1)\}_{l=1}^{N_p}$ from probability distribution $\sum_{l=1}^{N_p} w_l \delta_{x_l^\star(k-1)}(\tilde{x}^\star(k-1))$;
6: **end if**
7: **for** $l = 1, \ldots, N_p$ **do**
8: **generate** new particle positions $\tilde{x}_l^\star(k)$ from the evolutionary equation (34a)
9: **set** weights

$$w_l(k) \leftarrow p(v^\star(k)|\tilde{x}_l^\star(k))$$

with

$$p(v^\star(k)|\tilde{x}_l^\star(k)) = \Phi\left(\frac{q_{v^\star(k+1)} - C[h]\tilde{x}_l^\star(k)}{\sigma_e[h]}\right) - \Phi\left(\frac{q_{v^\star(k)} - C[h]\tilde{x}_l^\star(k)}{\sigma_e[h]}\right)$$

10: **end for**
11: **normalize** weights $w_l(k) \leftarrow \frac{w_l(k)}{\sum_{j=1}^{N_p} w_j(k)}$
12: **end for**

Output: at each iteration $k$, return particles' position $\{\tilde{x}_l^\star(k)\}_{l=1}^{N_p}$ and weights $\{w_l\}_{l=1}^{N_p}$.

# 6 Numerical example

The effectiveness of the proposed identification algorithm is demonstrated via a numerical example. All computations are carried out on an i7 1.9-GHz Intel core processor with 32 GB of RAM running MATLAB R2019a. Results of the examples can be replicated by running the MATLAB codes that can be downloaded at `http://dariopiga.com/Software/QuBay.rar`.

The data are generated by a linear difference system described as in (3) with model orders $n_{\mathrm{a}} = 4$ and $n_{\mathrm{b}} = 4$. The true parameters of the system are $a = [0.3 \ -0.4 \ \ 0.5 \ -0.6]$ and $b = [0.8 \ \ 0.7 \ -0.2 \ 0.1]$. We consider a training data with $N = 2000$ samples, generated by exciting the system with a white input signal which is a uniformly distributed in the interval $[-1 \ 1]$. The output is corrupted by a zero-mean Gaussian noise $e(k)$ with variance $\sigma_e^2 = 0.36$.

In order to asses the statistical properties of the proposed method, a Monte-Carlo simulation of 100 runs is performed. At each run, a new realization of input and noise process is generated. The average *Signal-to-Noise Ratio* (SNR) over 100 runs is 9.8 dB, where the SNR for a single Monte-Carlo run is defined as

$$\mathrm{SNR} = 10 \log \frac{\sum_{k=1}^{N} x^2(k)}{\sum_{k=1}^{N} e^2(k)}. \tag{37}$$

For each Monte-Carlo run, the *maximum-likelihood* (ML) and *maximum-a-posteriori* (MAP) estimates of the parameters $\bar{\theta} = \begin{bmatrix} \theta^\top & x_0^\top & \bar{\tau} & \end{bmatrix}^\top$ are computed (as detailed in Section 3) using a gradient-ascent algorithm. For MAP estimates, an isotropic Gaussian prior with $\lambda = 1$ in (15) is chosen. In the gradient-ascent algorithm, the initial condition of each parameter value is chosen randomly from a uniform distribution in the interval $(0, 1)$. The algorithm is terminated either when the maximum number of iterations $h_{\max} = 350$ is reached or when the norm of the variation of the objective function between two consecutive iterations is smaller than $10^{-8}$.

The ML and MAP estimates of the model parameters and noise standard deviation $\sigma_e$ are reported in Table 1, for data generated using $K = 8$ quantization intervals, uniformly spaced in the range $[q_1 \ q_{K+1}] = [-9 \ 7]$. A close match between true and estimated parameters can be observed.

In order to asses the effect of the number of quantization intervals $K$ on the estimate of the model parameters, we compute the ML and MAP estimates for three different quantization intervals $K = 2, 4, 8$. For binary quantizer $K = 2$, the quantization level $q_2$ in (6) is set to $q_2 = 1$. For quantizers with $K = 4$ and $K = 8$, the quantization levels are uniformly spaced in the range $[q_1 \ q_{K+1}] = [-9 \ \ 8]$. The identified models with ML and MAP estimates are used to reconstruct the outputs of the model (3). The match between the estimated and the true output is quantified on a noise-free validation data of

Table 1: Monte-Carlo analysis with 100 runs: true *vs* estimated model parameters (mean ± std deviation) with $K = 8$ level quantizer.

| | True | ML estimate (mean ± std) | MAP estimate (mean ± std) |
|---|---|---|---|
| $a_1$ | 0.30 | $0.2998 \pm 0.0219$ | $0.2941 \pm 0.0222$ |
| $a_2$ | $-0.40$ | $-0.4037 \pm 0.0160$ | $-0.4035 \pm 0.0152$ |
| $a_3$ | 0.50 | $0.4961 \pm 0.0130$ | $0.4964 \pm 0.0121$ |
| $a_4$ | $-0.60$ | $-0.5969 \pm 0.0193$ | $-0.5959 \pm 0.0168$ |
| $b_1$ | 0.80 | $0.7990 \pm 0.0386$ | $0.8001 \pm 0.0379$ |
| $b_2$ | 0.70 | $0.7075 \pm 0.0318$ | $0.7112 \pm 0.0303$ |
| $b_3$ | $-0.20$ | $-0.1976 \pm 0.0427$ | $-0.1903 \pm 0.0416$ |
| $b_4$ | 0.1 | $0.1134 \pm 0.0460$ | $0.1108 \pm 0.0438$ |
| $\sigma_e$ | 0.6 | $0.6008 \pm 0.0167$ | $0.6296 \pm 0.0142$ |

length $N_{\text{val}} = 1000$, in terms of the *Best Fit Rate* (BFR) index defined by:

$$\text{BFR} = \max\left\{1 - \sqrt{\frac{\sum_{k=1}^{N_{\text{val}}} \left(x(k) - \hat{x}(k)\right)^2}{\sum_{k=1}^{N_{\text{val}}} \left(x(k) - \bar{x}(k)\right)^2}}, 0\right\} \times 100\%, \tag{38}$$

where $x(k)$ and $\hat{x}(k)$ are the (non-quantized) true and open-loop simulated model output at time $k$, respectively and $\bar{x}$ is the sample mean of the output over the validation set.

The box-plots of the BFR with three quantization levels $K = 2, 4, 8$ are plotted in Fig. 1. It can be seen from Fig. 1 that with ML and MAP estimates of the parameters, the output is accurately reconstructed with quantized data for all three quantization levels. As expected, the overall accuracy increases with the number of quantization intervals.

Next, we compute the ML and MAP estimates by changing the length $N$ of the training data set for a binary quantizer $K = 2$. In Fig. 2 we report the BFR and the norm of the parameter estimation error $\|\theta_{\text{o}} - \theta\|_2$ and the CPU time, for different values of $N$. The accuracy of the estimated parameters increases with increase in data size $N$, at the cost of a *linear* increase in computation time.

*Laplace approximation and MCMC*

We now compute the approximation of the conditional posterior $p(\theta, x_0, \bar{\tau}|\mathcal{V}, \mathcal{U})$ using both Laplace's method and MCMC algorithm, as described in Section 4.2.

Starting from the MAP estimates $\theta^{\text{MAP}}, x_0^{\text{MAP}}, \bar{\tau}^{\text{MAP}}$, the Hessian $\Lambda$ of (minus) the log posterior is computed and the expression of the Laplace approximation in (22) is obtained. The average execution time to compute the Hessian $\Lambda$ is 0.036 sec, with a training data of length $N = 2000$ samples, quantized with a binary quantizer $K = 2$. The MAP estimates and the estimated standard devia-

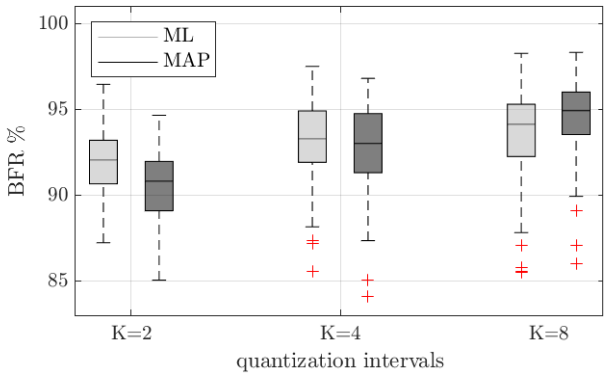



Figure 1: Monte-Carlo analysis: boxplots of achived BFR indices with ML and MAP estimates for data quantized with different quantization levels $K = 2, 4, 8$.

Table 2: True *vs* estimated model parameters, mean $\pm$ standard deviation computed via Laplace approximation and MCMC, with binary $K = 2$ quantized data, for a single realization of input and noise sequence.

| | True | Laplace approx (mean $\pm$ std) | MCMC estimate (mean $\pm$ std) |
|---|---|---|---|
| $a_1$ | 0.30 | $0.3193 \pm 0.0414$ | $0.3125 \pm 0.0362$ |
| $a_2$ | $-0.40$ | $-0.3887 \pm 0.0294$ | $-0.4025 \pm 0.0193$ |
| $a_3$ | 0.50 | $0.4822 \pm 0.0260$ | $0.4756 \pm 0.0205$ |
| $a_4$ | $-0.60$ | $-0.6142 \pm 0.0262$ | $-0.6129 \pm 0.0247$ |
| $b_1$ | 0.80 | $0.8315 \pm 0.0486$ | $0.8129 \pm 0.0437$ |
| $b_2$ | 0.70 | $0.7275 \pm 0.0637$ | $0.7005 \pm 0.0543$ |
| $b_3$ | $-0.20$ | $-0.2446 \pm 0.0792$ | $-0.2013 \pm 0.0589$ |
| $b_4$ | 0.1 | $0.1283 \pm 0.0767$ | $0.1508 \pm 0.0600$ |
| $\bar{\tau}$ | 1.02 | $0.7285 \pm 0.0641$ | $1.0276 \pm 0.0808$ |

tions of the parameters corresponding to the computed Laplace's approximation are reported in Table 2.

In order to approximate the posterior $p(\theta, x_0, \bar{\tau}|\mathcal{V}, \mathcal{U})$ via MCMC, Algorithm 1 is run for $M = 5000$ iterations. According to the *burn-in* strategy, the first 1000 samples of the Markov chain are discarded. The execution time to run Algorithm 1 is 12.1 seconds. Based on the samples generated by Algorithm 1, the estimated mean along with estimated standard deviations of the parameters are reported in Table 2 for a binary $K = 2$ quantizer.

From Table 2, it can be noted that both Laplace's method and the MCMC algorithm provide an approximation of the desired posterior distribution such

that the true parameters lie in the uncertainty intervals defined by the standard deviations.

The performance of the Laplace and MCMC approximations is further assessed for data generated with different quantization levels $K = 2, 4, 8$. The quantization levels $q_1, \ldots, q_K$ considered are $\{-5, 1, 6\}$ for $K = 2$, $\{-5, -3, 1, 2, 6\}$ for $K = 4$ and $\{-5, -3, -2, -1, 1, 2, 3, 4, 6\}$ for $K = 8$ level quantizers. For brevity, the estimate of parameter $a_1$ along with its 3-standard deviations intervals, computed from the Hessian $\Lambda$ of Laplace approximation and through the MCMC samples are plotted in Fig. 3. We observe a slight reduction in the uncertainty intervals with increase in the number of quantization intervals.

Finally, in order to asses the robustness of MCMC w.r.t different input and noise realizations, we perform Monte-Carlo analysis with 100 runs. At each run, the posterior is approximated via the Laplace and the MCMC approaches, and mean and standard deviations of parameters are recorded. Figure 4 shows the mean and 3x standard deviation uncertainty intervals of parameter value $b_1 = 0.8$, obtained over different Monte Carlo runs. It can be observed that the true parameter value belongs to the uncertainty intervals computed from the Laplace and MCMC approximations of the posterior. Furthermore, uncertainty intervals provided by MCMC are tighter than the ones obtained through Laplace's method, showing that MCMC approximation is more accurate than Laplace's method, at the price of a higher computational load.

*Inference*

Based on the samples generated by the MCMC algorithm, we infer the output $x^\star(k)$ for a new sequence of inputs $\mathcal{U}^\star_{k-1}$ as detailed in Section 4.3. To this end, we consider a new white input sequence of length 1000 samples generated from a uniform distribution in the interval $[-1 \;\; 1]$, which is not used for training. The distribution of the inferred output $x^\star(k)$ sequence is computed according to eq. (30), by choosing the prior distribution of the initial states $x^\star_0$ as an isotropic zero-mean Gaussian with covariance matrix $(0.01)I_{n_a}$. Note that the distribution $p(x^\star(k)|\mathcal{V}, \mathcal{U}, \mathcal{U}^\star_{k-1})$ in (30), is a mixture of Gaussians whose mean and variance can be computed analytically and are given by

$$\mu_{x^\star(k)} = \frac{1}{H}\sum_{h=1}^{H} \mu[h,k],$$

$$\sigma^2_{x^\star(k)} = \frac{1}{H}\sum_{h=1}^{H} \sigma^2[h,k] + \frac{1}{H}\sum_{h=1}^{H} (\mu[h,k] - \mu_{x^\star(k)})^2,$$

where $H = 4000$ is the number of MCMC samples and $\mu[h,k] = x^\star_F(k;\theta) + EA^k\bar{x}^\star_0$ and $\sigma^2[h,k] = EA^k \mathrm{Cov}(x^\star_0)\big(A^k\big)^\top E^\top$, are derived according to (26c) and (27). The mean $\mu_{x^\star(k)}$ and the uncertainty interval corresponding to 3x standard deviations $3\sigma_{x^\star(k)}$ of the inferred output distribution $p(x^\star(k)|\mathcal{V}, \mathcal{U}, \mathcal{U}^\star_{k-1})$ are plotted in Fig. 5, for $K = 2$ and $K = 8$ quantization intervals. For the sake of better visualization, only a subset of output sequence is plotted. It can be observed that the reconstructed output matches closely with the true one.

Moreover, the variance of the estimated output is reduced with the increase in the number of quantization levels.

*Filtering*

We now compute the distribution $p(x^\star(k)|\mathcal{V}_k^\star,\mathcal{U}_k^\star,\mathcal{V},\mathcal{U})$ (given in (32)) of the latent output at time $k$ given past observations, as described in Section 5. To this end, we first compute the conditional distribution $p(\tilde{x}^\star(k)|\theta[h],\bar{\tau}[h],\mathcal{V}_k^\star,\mathcal{U}_k^\star)$ given in (36) for each of the MCMC samples $\{\theta[h],\bar{\tau}[h]\}_{h=1}^H$ via the *particle filter* approach in Algorithm 2.

A dataset with 2000 samples is processed sequentially, simulating a scenario where data are acquired and processed in real time. At each time $k$, the posterior $p(\tilde{x}^\star(k)|\theta[h],\bar{\tau}[h],\mathcal{V}_k^\star,\mathcal{U}_k^\star)$ is updated based on Algorithm 2 with $N_p = 100$ particles. Initial conditions $\{\tilde{x}_l^\star(0)\}_{l=1}^{N_p}$ of the particles are sampled from a prior distribution $p(\tilde{x}^\star(0))$ which is chosen to be an isotropic Gaussian with zero-mean and covariance $0.01I_{n_\mathrm{a}}$. At each successive time steps, the particles $\{\tilde{x}_l^\star(k)\}_{l=1}^{N_p}$ are sampled according to the evolutionary equation (34a). In particular, the distribution of the process noise $\eta$ is chosen to be zero-mean isotropic Gaussian $p(\eta) = \mathcal{N}(\eta;0,\sigma_\eta^2 I))$. According to (34a), each particle $\{\tilde{x}_l^\star(k)\}$ is generated from a Gaussian distribution with mean $A[h]\tilde{x}_l^\star(k-1) + Bu^\star(k)$ and variance $\sigma_\eta^2 = (0.05)^2$ chosen via trial and error. Once the particles $\{\tilde{x}_l^\star(k)\}$ are computed, the output $\{x_l^\star(k)\}$ is obtained as $x_l^\star(k) = C[h]\tilde{x}_l^\star(k)$ and the distribution of interest $p(x^\star(k)|\theta[h],\bar{\tau}[h],\mathcal{V}_k^\star,\mathcal{U}_k^\star)$ is computed according to (36). The execution time required by Algorithm 2 to process 2000 samples for a single MCMC sample is 0.52 sec.

Finally, the filtering distribution $p(x^\star(k)|\mathcal{V}_k^\star,\mathcal{U}_k^\star,\mathcal{V},\mathcal{U})$ is approximated as given in (32), by running the particle filter algorithm for each MCMC sample $\{\theta[h],\bar{\tau}[h]\}_{h=1}^H$. The mean and variance of this distribution are computed according to

$$\mu_{x^\star(k)} = \frac{1}{H}\sum_{h=1}^{H}\mu[h,k],$$

$$\sigma^2_{x^\star(k)} = \frac{1}{H}\sum_{h=1}^{H}(\sigma^2[h,k] + (\mu[h,k])^2) - (\mu_{x^\star(k)})^2,$$

where $H = 4000$ is the length of MCMC samples and $\mu[h,k]$ and $\sigma^2[h,k]$ denote the mean and variance of discrete distribution $p(x^\star(k)|\theta[h],\bar{\tau}[h],\mathcal{V}_k^\star,\mathcal{U}_k^\star)$ (36), which are computed using the computed probability distributions (36). The expected value $\mu_{x^\star(k)}$ of the output $x^\star(k)$ is plotted in Fig 6, along with the uncertainty intervals $\pm 3\sigma_{x^\star(k)}$ for a binary quantizer $K = 2$. The plot shows that particle filtering algorithm is able to recursively reconstruct the non-quantized output of the true system accurately.

## 6.1 Analysis for multiple data-generating systems

In order to asses the robustness of the proposed method, we compute the ML and MAP estimates for different data generating systems. Using MATLAB's drss

command, 15 different 4-th order stable LTI state-space models are generated. The quantized measurements with three different quantization intervals $K = 2, 4, 8$, are gathered for each system with training and validation data of lengths 2000 and 1000 samples, respectively. The quantization intervals are spaced uniformly between the minimum and maximum value of the system output. The training output is corrupted with a Gaussian noise leading to average signal-to-noise ratio of 10 dB.

We compute the ML and MAP estimates for the model (3) with model orders set to $n_\mathrm{a} = 4$ and $n_\mathrm{b} = 4$. For MAP estimates, an isotropic Gaussian prior with $\lambda = 1$ in (15) is chosen. For each system, the gradient-ascent algorithm is run for 800 iterations, with initial condition of each parameter chosen randomly from a uniform distribution in the interval $(0, 0.1)$. The box-plots of the achieved BFR index with three quantization levels $K = 2, 4, 8$ are plotted in Fig. 7. It can be seen from Fig. 7 that the output is accurately reconstructed with quantized data for all 15 systems, with only a few outliers.

# 7 Conclusions

A Bayesian framework for transfer function estimation of linear dynamical systems from quantized output observations is discussed in the paper. A (log)-likelihood function of the system parameters, initial conditions and noise standard deviation is formulated. Its gradient is computed and used in a gradient-ascent algorithm for maximum likelihood and maximum-a-posteriori estimation. The Hessian of the posterior distribution is also computed, and used to approximate the posterior distribution of the unknown variables through Laplace's method. A Markov-Chain Monte-Carlo approximation of the posterior is also proposed, exploiting the Laplace approximation as a proposal distribution. Predictive inference is also addressed. Finally, a filtering problem is considered and particle filters are used to recursively reconstruct a latent (non-quantized) output from past (quantized) observations. This can be useful in control design, where the hidden non-quantized output can be used both to assess closed-loop performance and to make prediction of the next output sequence, for example in model-predictive control.

Several extensions of the proposed contribution are currently under investigation, which include: parametric Bayesian estimation of non-linear systems, experiment design and optimal choice of the quantization level for minimization of model uncertainty.

# References

[1] J. C. Agüero, G. C. Goodwin, and J. I. Yuz. System identification using quantized data. In *46th IEEE Conference on Decision and Control*, pages 4263–4268, New Orleans, LA, USA, 2007.

[2] C. Andrieu, N. De Freitas, A. Doucet, and M. I. Jordan. An introduction to MCMC for machine learning. *Machine learning*, 50(1-2):5–43, 2003.

[3] C. M. Bishop. *Pattern recognition and machine learning*. Springer, 2006.

[4] G. Bottegal, H. Hjalmarsson, and G. Pillonetto. A new kernel-based approach to system identification with quantized output data. *Automatica*, 85:145 – 152, 2017.

[5] M. Casini, A. Garulli, and A. Vicino. Set-membership identification of ARX models with quantized measurements. In *50th IEEE Conference on Decision and Control and European Control Conference*, pages 2806–2811, Orlando, FL, USA, 2011.

[6] M. Casini, A. Garulli, and A. Vicino. Bounding nonconvex feasible sets in set membership identification: OE and ARX models with quantized information. In *16th IFAC Symposium on System Identification*, pages 1191 – 1196, Brussels, Belgium, 2012.

[7] M. Casini, A. Garulli, and A. Vicino. Input design in worst-case system identification with quantized measurements. *Automatica*, 48(12):2997 – 3007, 2012.

[8] V. Cerone, D. Piga, and D. Regruto. Fixed-order FIR approximation of linear systems from quantized input and output data. *Systems and Control Letters*, 62(12):1136 – 1142, 2013.

[9] T. Chen, Y. Zhao, and L. Ljung. Impulse response estimation with binary measurements: A regularized FIR model approach. In *16th IFAC Symposium on System Identification*, pages 113–118, Brussels, Belgium, 2012.

[10] E. Colinet and J. Juillard. A weighted least-squares approach to parameter estimation problems based on binary measurements. *IEEE Transactions on Automatic Control*, 55(1):148–152, 2009.

[11] B. Csáji and E. Weyer. Recursive estimation of ARX systems using binary sensors with adjustable thresholds. *IFAC Proceedings Volumes*, 45(16):1185–1190, 2012.

[12] B. I. Godoy, J. C. Agüero, R. Carvajal, G. C. Goodwin, and J. I. Yuz. Identification of sparse FIR systems using a general quantisation scheme. *International Journal of Control*, 87(4):874–886, 2014.

[13] B. I. Godoy, G. C. Goodwin, J. C. Agüero, D. Marelli, and T. Wigren. On identification of FIR systems having quantized output data. *Automatica*, 47(9):1905 – 1915, 2011.

[14] F. Gustafsson and R. Karlsson. Statistical results for system identification based on quantized observations. *Automatica*, 45:2794–2801, 12 2009.

[15] S. Kan, G. Yin, and L. Y. Wang. Parameter estimation in systems with binary-valued observations and structural uncertainties. *International Journal of Control*, 87(5):1061–1075, 2014.

[16] I. Kollar. Bias of mean value and mean square value measurements based on quantized data. *IEEE Transactions on Instrumentation and Measurement*, 43(5):733–739, 1994.

[17] L. Y. Wang, J.F. Zhang, and G. G. Yin. System identification using binary sensors. *IEEE Transactions on Automatic Control*, 48(11):1892–1907, 2003.

[18] A. S. Leong, E. Weyer, and G. N. Nair. Identification of FIR systems with binary input and output observations. *IEEE Transactions on Automatic Control*, 66(3):1190–1198, 2021.

[19] G. Li and C. Wen. Identification of wiener systems with clipped observations. *IEEE Transactions on Signal Processing*, 60(7):3845–3852, 2012.

[20] L. Ljung. *System Identification: Theory for the User*. Prentice-Hall Englewood Cliffs, NJ, 1999.

[21] D. Marelli, K. You, and M. Fu. Identification of ARMA models using intermittent and quantized output observations. *Automatica*, 49(2):360 – 369, 2013.

[22] G. Pillonetto and G. De Nicolao. A new kernel-based approach for linear system identification. *Automatica*, 46(1):81 – 93, 2010.

[23] M. Pouliquen, E. Pigeon, O. Gehan, and A. Goudjil. Identification using binary measurements for IIR systems. *IEEE Transactions on Automatic Control*, 65(2):786–793, 2019.

[24] M. Pouliquen, E. Pigeon, O. Gehan, A. Goudjil, and R. Auber. Impulse response identification from input/output binary measurements. *Automatica*, 123, 2021.

[25] R. S. Risuleo, G. Bottegal, and H. Hjalmarsson. Identification of linear models from quantized data: A midpoint-projection approach. *IEEE Transactions on Automatic Control*, 65(7):2801–2813, 2020.

[26] H. Suzuki and T. Sugie. System identification based on quantized i/o data corrupted with noises and its performance improvement. In *Proceedings of the 45th IEEE Conference on Decision and Control*, pages 3684–3689, San Diego, CA, USA, 2006.

[27] L. Y. Wang, G. G. Yin, J. F. Zhang, and Y. Zhao. *System identification with quantized observations*. 2010.

[28] L. Y. Wang, G. G. Yin, and J.F. Zhang. Joint identification of plant rational models and noise distribution functions using binary-valued observations. *Automatica*, 42(4):535 – 547, 2006.

[29] Y. Zhao, L. Y. Wang, G. G. Yin, and J.F. Zhang. Identification of Wiener systems with binary-valued output observations. *Automatica*, 43(10):1752 – 1765, 2007.

[30] Y. Zhao, J.F. Zhang, L. Y. Wang, and G. G. Yin. Identification of Hammerstein systems with quantized observations. *SIAM J. Control and Optimization*, 48:4352–4376, 05 2010.

# A Second-order derivatives of the log posterior

In this appendix, we derive the analytical expressions of the second-order derivatives of the log-posterior distribution $\log p(\theta, x_0, \bar{\tau}|\mathcal{V},\mathcal{U})$. These derivatives are required to construct the Hessian $\Lambda$ used to compute the Laplace approximation discussed in Section 4.2.1.

For the sake of exposition, we remind the expression of the log-posterior distribution $\log p(\theta, x_0, \bar{\tau}|\mathcal{V},\mathcal{U})$:

$$\log p(\theta, x_0, \bar{\tau}|\mathcal{V},\mathcal{U}) =$$

$$\underbrace{\sum_{k=1}^{N} \log\left(\Phi\left(\frac{q_{v(k)+1} - x(k)}{\sigma_e}\right) - \Phi\left(\frac{q_{v(k)} - x(k)}{\sigma_e}\right)\right)}_{\mathcal{L}(\theta,x_0,\bar{\tau})} \tag{39a}$$

$$\underbrace{-\frac{\lambda^{-2}}{2}[\theta^\top \;\; x_0^\top \;\; \bar{\tau}] \begin{bmatrix} \theta \\ x_0 \\ \bar{\tau} \end{bmatrix} + \text{const},}_{\log p(\theta,x_0,\bar{\tau}) - \log p(\mathcal{V}|U)} \tag{39b}$$

where we recall that $\sigma_e = \sqrt{\exp(-\bar{\tau})}$.

The Hessian of the quadratic form in eq. (39b) is given by:

$$\frac{\partial^2 \log p(\theta, x_0, \bar{\tau})}{\partial[\theta^\top \;\; x_0^\top \;\; \bar{\tau}]^2} = -\lambda^{-2} I,$$

where $I$ is the identity matrix of size $n_\theta + n_\text{a} + 1$.

As for the Hessian of the log-likelihood $\mathcal{L}$ in (39a), we first define the following

terms that will help to compact the notation:

$$
\begin{aligned}
N_g(k) =& \mathcal{N}\left(\frac{Q_{v(k)+1}-x(k)}{\sigma_e};0,1\right) - \mathcal{N}\left(\frac{Q_{v(k)}-x(k)}{\sigma_e};0,1\right),\\
D_g(k) =& \Phi\left(\frac{Q_{v(k)+1}-x(k)}{\sigma_e}\right) - \Phi\left(\frac{Q_{v(k)}-x(k)}{\sigma_e}\right),\\
N^d(k) =& \mathcal{N}'\left(\frac{Q_{v(k)+1}-x(k)}{\sigma_e};0,1\right) - \mathcal{N}'\left(\frac{Q_{v(k)}-x(k)}{\sigma_e};0,1\right),\\
F_1(k) =& \mathcal{N}\left(\frac{Q_{v(k)+1}-x(k)}{\sigma_e};0,1\right) \times (Q_{v(k)+1}-x(k)),\\
F_2(k) =& \mathcal{N}\left(\frac{Q_{v(k)}-x(k)}{\sigma_e};0,1\right) \times (Q_{v(k)}-x(k)),\\
G_1(k) =& \mathcal{N}'\left(\frac{Q_{v(k)+1}-x(k)}{\sigma_e};0,1\right) \times (Q_{v(k)+1}-x(k)),\\
G_2(k) =& \mathcal{N}'\left(\frac{Q_{v(k)}-x(k)}{\sigma_e};0,1\right) \times (Q_{v(k)}-x(k)),
\end{aligned}
$$

where $\mathcal{N}'$ is the derivative of the probability density function of the Normal distribution w.r.t. to its argument, *i.e.*,

$$\mathcal{N}'(x;0,1) = -x\mathcal{N}(x;0,1).$$

In Section 3.2.1 we have already computed the gradient of $\mathcal{L}$ w.r.t. $\theta$, which is reported in the following to facilitate reading:

$$\frac{\partial \mathcal{L}}{\partial \theta} = \sum_{k=1}^{N} \frac{N_g(k)}{D_g(k)} \times \frac{1}{\sigma_e}\left(-\frac{\partial x(k)}{\partial \theta}\right).$$

Then, we have

$$
\begin{aligned}
&\frac{\partial^2 \mathcal{L}}{\partial \theta^2} = \\
&-\frac{1}{\sigma_e^2}\sum_{k=1}^{N} \frac{N^d(k)\frac{\partial x(k)}{\partial \theta} \times D_g(k) - N_g^2(k)\frac{\partial x(k)}{\partial \theta}}{D_g^2(k)}\left(-\frac{\partial x(k)}{\partial \theta}\right)^{\top}\\
&+\frac{1}{\sigma_e}\sum_{k=1}^{N} \frac{N_g(k)}{D_g(k)}\left(-\frac{\partial^2 x(k)}{\partial \theta^2}\right). \qquad (40)
\end{aligned}
$$

The second-order derivatives $\frac{\partial^2 x(k)}{\partial^2 \theta}$ appearing in (40) are derived by taking

derivatives of left and right expression of (10) as follows:

$$\frac{\partial^2 x(k)}{\partial a_z \partial a_i} = \frac{\partial x(k-i)}{\partial a_z} + \frac{\partial x(k-z)}{\partial a_i} + \sum_{t=1}^{n_\mathrm{a}} a_t \frac{\partial^2 x(k-t)}{\partial a_z \partial a_i}$$

$$\frac{\partial^2 x(k)}{\partial b_z \partial a_i} = \frac{\partial x(k-i)}{\partial b_z} + \sum_{t=1}^{n_\mathrm{a}} a_t \frac{\partial^2 x(k-t)}{\partial b_z \partial a_i}$$

$$\frac{\partial^2 x(k)}{\partial b_z \partial b_j} = \sum_{i=1}^{n_\mathrm{a}} a_i \frac{\partial^2 x(k-i)}{\partial b_z \partial b_j}$$

$$\frac{\partial^2 x(k)}{\partial a_z \partial b_j} = \frac{\partial x(k-z)}{\partial b_j} + \sum_{t=1}^{n_\mathrm{a}} a_t \frac{\partial^2 x(k-t)}{\partial a_z \partial b_j}$$

As for the second-order derivatives w.r.t the initial conditions $\frac{\partial^2 \mathcal{L}}{\partial x_0^2}$, by differentiating the gradient $\frac{\partial \mathcal{L}}{\partial x_0}$ in (11) w.r.t. $x_0$ we obtain

$$\begin{aligned}
&\frac{\partial^2 \mathcal{L}}{\partial x_0^2} = \\
&-\frac{1}{\sigma_e^2} \sum_{k=1}^{N} \frac{N^d(k) \frac{\partial x(k)}{\partial x_0} \times D_g(k) - N_g^2(k) \frac{\partial x(k)}{\partial x_0}}{D_g^2(k)} \left( -\frac{\partial x(k)}{\partial x_0} \right)^\top \\
&+\frac{1}{\sigma_e} \sum_{k=1}^{N} \frac{N_g(k)}{D_g(k)} \left( -\frac{\partial^2 x(k)}{\partial x_0^2} \right).
\end{aligned} \tag{41}$$

Thus, in order to compute the second-order derivatives in (41), $\frac{\partial^2 x(k)}{\partial x_0^2}$ is required and derived by taking derivatives of left and right expression in (12):

$$\frac{\partial^2 x(k)}{\partial x_0^2} = \sum_{i=1}^{n_\mathrm{a}} a_i \frac{\partial^2 x(k-i)}{\partial x_0^2},$$

with $\frac{\partial^2 x(k)}{\partial x_0^2} = 0$ for $k \leq 0$ due to eq. (13).

As for the second-order derivative $\frac{\partial^2 \mathcal{L}}{\partial \bar{\tau}^2}$, we first compute $\frac{\partial^2 \mathcal{L}}{\partial \sigma_e^2}$ by differentiating the gradient $\frac{\partial \mathcal{L}}{\partial \sigma_e}$ in (14) w.r.t. $\sigma_e$, thus obtaining

$$\begin{aligned}
&\frac{\partial^2 \mathcal{L}}{\partial \sigma_e^2} = \\
&\sum_{k=1}^{N} \left( \left( \frac{(q_{v(k)+1} - x(k)) G_1(k) - (q_{v(k)} - x(k)) G_2(k)}{\sigma_e^4} \right) \right. \\
&\left. -2 \left( \frac{-F_1(k) + F_2(k)}{\sigma_e^3} \right) \right) \times \frac{1}{D_g(k)} + \\
&+ \sum_{k=1}^{N} \left( \frac{-F_1(k) + F_2(k)}{\sigma_e^2} \right)^2 \frac{-1}{D_g^2(k)}.
\end{aligned}$$

Then, in order to derive $\frac{\partial^2 \mathcal{L}}{\partial \bar{\tau}^2}$, we consider the relation

$$\begin{aligned}\frac{\partial^2 \mathcal{L}}{\partial \bar{\tau}^2} &= \frac{\partial}{\partial \bar{\tau}}\left(\frac{\partial \mathcal{L}}{\partial \bar{\tau}}\right) = \frac{\partial^2 \mathcal{L}}{\partial \sigma_e \partial \bar{\tau}} \frac{\partial \sigma_e}{\partial \bar{\tau}} \\ &= \left(\frac{\partial^2 \mathcal{L}}{\partial \sigma_e^2}\frac{\partial \sigma_e}{\partial \bar{\tau}} + \frac{\partial \mathcal{L}}{\partial \sigma_e}\frac{\partial^2 \sigma_e}{\partial \sigma_e \partial \bar{\tau}}\right)\frac{\partial \sigma_e}{\partial \bar{\tau}} \\ &= \frac{1}{4\exp(\bar{\tau})}\frac{\partial^2 \mathcal{L}}{\partial \sigma_e^2} + \frac{1}{4}\frac{\partial \mathcal{L}}{\partial \sigma_e}\frac{1}{\sqrt{\exp(\bar{\tau})}}.\end{aligned}$$

## Mixed derivatives

Let us now compute the mixed derivative

$$\begin{aligned}&\frac{\partial^2 \mathcal{L}}{\partial x_0 \partial \theta} = \\ &-\frac{1}{\sigma_e^2}\sum_{k=1}^{N}\frac{N_g(k)\frac{\partial x(k)}{\partial x_0} \times D_g(k) - N_g^2(k)\frac{\partial x(k)}{\partial x_0}}{D^2(k)}\left(-\frac{\partial x(k)}{\partial \theta}\right)^\top \\ &+\frac{1}{\sigma_e}\sum_{k=1}^{N}\frac{N_g(k)}{D_g(k)}\left(-\frac{\partial^2 x(k)}{\partial x_0 \partial \theta}\right).\end{aligned}$$

The mixed second order derivative $\frac{\partial^2 x(k)}{\partial x_0 \partial \theta}$ can be computed by differentiating (10a) and (10b) w.r.t. $x_0$ as follows:

$$\begin{aligned}\frac{\partial^2 x(k)}{\partial x_0 \partial a_i} &= \frac{\partial x(k-i)}{\partial x_0} + \sum_{t=1}^{n_\mathrm{a}} a_t \frac{\partial^2 x(k-t)}{\partial x_0 \partial a_i}, \\ \frac{\partial^2 x(k)}{\partial x_0 \partial b_j} &= \sum_{t=1}^{n_\mathrm{a}} a_t \frac{\partial^2 x(k-t)}{\partial x_0 \partial b_j}.\end{aligned}$$

Let us now consider the mixed derivative $\frac{\partial^2 \mathcal{L}}{\partial \theta \partial \bar{\tau}}$

$$\frac{\partial^2 \mathcal{L}}{\partial \theta \partial \bar{\tau}} = \left(-\frac{1}{2}\right)\frac{1}{\sqrt{\exp(\bar{\tau})}}\frac{\partial^2 \mathcal{L}}{\partial \theta \partial \sigma_e}.$$

We then compute

$$\frac{\partial \mathcal{L}}{\partial \sigma_e} = \sum_{k=1}^{N}\left(\frac{-F_1(k)}{\sigma_e^2} - \frac{-F_2(k)}{\sigma_e^2}\right) \times \frac{1}{D_g(k)}$$

and

$$
\begin{aligned}
&\frac{\partial^2 \mathcal{L}}{\partial\theta\partial\sigma_e} = \\
&\sum_{k=1}^{N}\left(\frac{1}{\sigma_e^3}\frac{\partial x(k)}{\partial\theta}G_1(k) + \frac{1}{\sigma_e^2}\frac{\partial x(k)}{\partial\theta}\mathcal{N}\left(\frac{q_{v(k)+1}-x(k)}{\sigma_e};0,1\right)\right) \\
&\quad\times\frac{1}{D_g(k)} \\
&-\sum_{k=1}^{N}\left(\frac{1}{\sigma_e^3}\frac{\partial x(k)}{\partial\theta}G_2(k) + \frac{1}{\sigma_e^2}\frac{\partial x(k)}{\partial\theta}\mathcal{N}\left(\frac{q_{v(k)}-x(k)}{\sigma_e};0,1\right)\right) \\
&\quad\times\frac{1}{D_g(k)} \\
&-\sum_{k=1}^{N}\left(\frac{-F_1(k)}{\sigma_e^2}-\frac{-F_2(k)}{\sigma_e^2}\right)\times\frac{-\frac{1}{\sigma_e}\frac{\partial x(k)}{\partial\theta}N_g(k)}{D_g^2(k)}.
\end{aligned}
$$

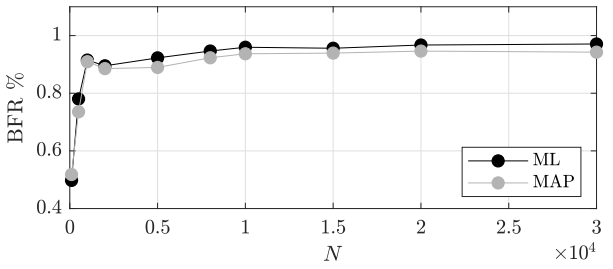
Let us now consider the mixed derivative

$$
\frac{\partial^2 \mathcal{L}}{\partial x_0\partial\bar{\tau}} = \left(-\frac{1}{2}\right)\frac{1}{\sqrt{\exp(\bar{\tau})}}\frac{\partial^2 \mathcal{L}}{\partial x_0\partial\sigma_e}.
$$

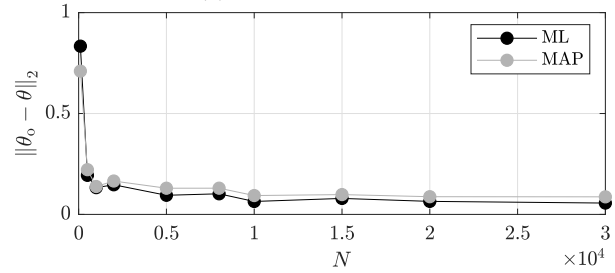
We then compute

$$
\begin{aligned}
&\frac{\partial^2 \mathcal{L}}{\partial x_0\partial\sigma_e} = \\
&\sum_{k=1}^{N}\left(\frac{1}{\sigma_e^3}\frac{\partial x(k)}{\partial x_0}G_1(k) + \frac{1}{\sigma_e^2}\frac{\partial x(k)}{\partial x_0}\mathcal{N}\left(\frac{q_{v(k)+1}-x(k)}{\sigma_e};0,1\right)\right) \\
&\quad\times\frac{1}{D_g(k)} + \\
&-\sum_{k=1}^{N}\left(\frac{1}{\sigma_e^3}\frac{\partial x(k)}{\partial x_0}G_2(k) + \frac{1}{\sigma_e^2}\frac{\partial x(k)}{\partial x_0}\mathcal{N}\left(\frac{q_{v(k)}-x(k)}{\sigma_e};0,1\right)\right) \\
&\quad\times\frac{1}{D_g(k)} \\
&-\sum_{k=1}^{N}\left(\frac{-F_1(k)}{\sigma_e^2}-\frac{-F_2(k)}{\sigma_e^2}\right)\times\frac{-\frac{1}{\sigma_e}\frac{\partial x(k)}{\partial x_0}N_g(k)}{D_g^2(k)}.
\end{aligned}
$$

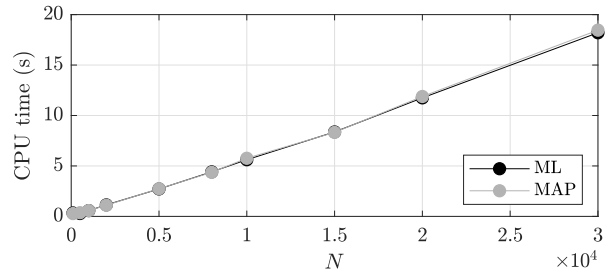
This complete the computation of all the second-order derivatives required to construct the Hessian $\Lambda$ of the log-posterior distribution $\log p(\theta, x_0, \bar{\tau}|\mathcal{V},\mathcal{U})$.




(a) BFR *vs* data size.




(b) parameter estimation error *vs* data size.




(c) CPU time (s) *vs* data size.

Figure 2: Performance of the identified model with ML and MAP estimates using binary quantizer $K = 2$ for different data sizes $N$.

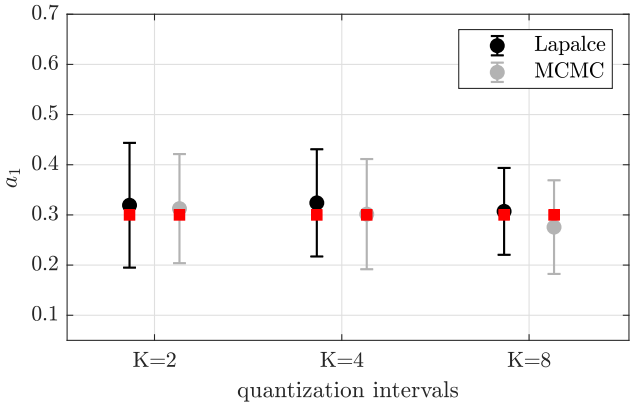



Figure 3: Laplace and MCMC approximation: (mean ± 3std) of the estimated parameter $a_1$ for different quantization levels $K = 2, 4, 8$. True values are marked with red color.

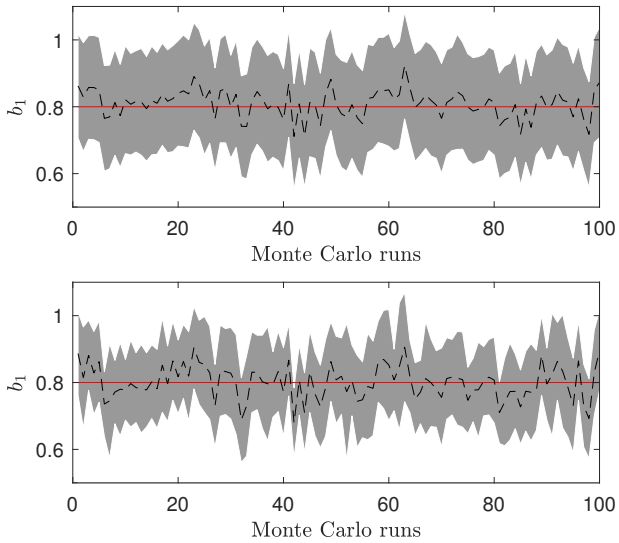



Figure 4: Monte-Carlo analysis: true value (red), mean (dashed black), mean ± 3std of the estimated parameter $b_1$ (gray shaded area) via Laplace (top plot) and MCMC (bottom plot) approximation for different Monte-Carlo simulations.

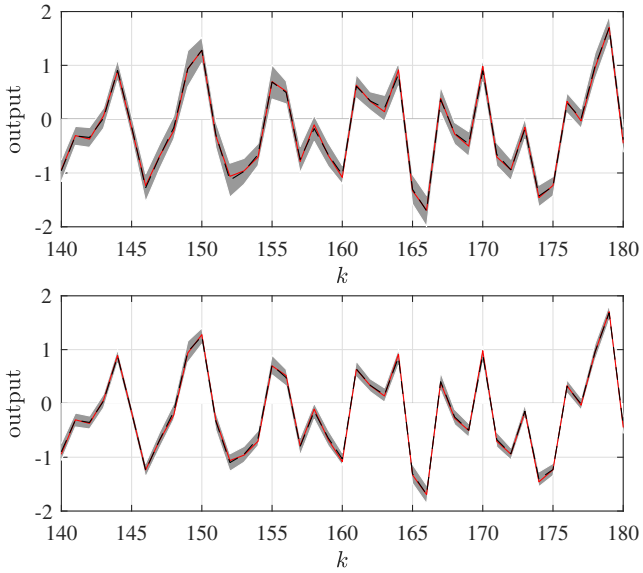



Figure 5: Inference: True output $x^\star(k)$; expected value of the output $\mu_{x^\star(k)}$ (dashed black); confidence intervals $\mu_{x^\star(k)} \pm 3\sigma_{x^\star(k)}$ (shaded grey region), computed with $H = 4000$ MCMC samples. Top panel: $K = 2$ quantization levels. Bottom panel: $K = 8$ quantization levels.

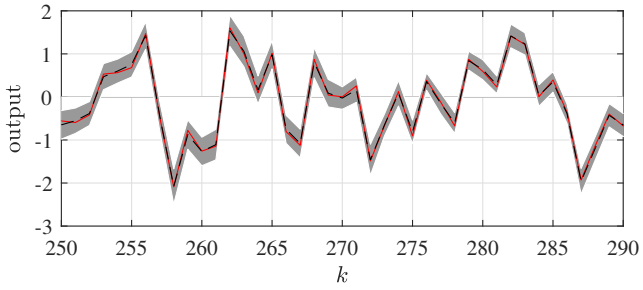



Figure 6: Particle filtering: true output $x^\star(k)$; expected value of the output $\mu_{x^\star(k)}$ (dashed black); confidence intervals $\mu_{x^\star(k)} \pm 3\sigma_{x^\star(k)}$ (shaded gray region), computed with $H = 4000$ MCMC samples and $K = 2$ quantization levels.

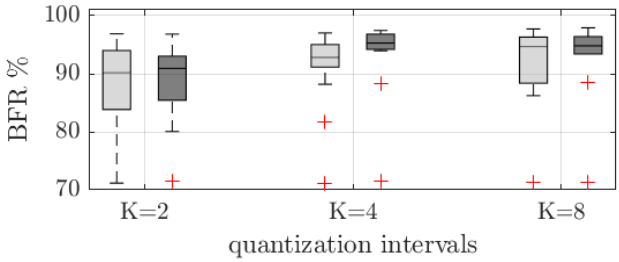



Figure 7: Boxplots of achived BFR indices with ML (light gray) and MAP (dark gray) estimates for 15 different data-generating systems.