An EM Algorithm for Lebesgue-sampled State-space Continuous-time System Identification ⋆

Rodrigo A. González * **Angel L. Cedeño** **,***
María Coronel *** **Juan C. Agüero** **,*** **Cristian R. Rojas** ****

* *Department of Mechanical Engineering, Eindhoven University of Technology, Eindhoven, The Netherlands*
** *Electronic Engineering Department, Universidad Técnica Federico Santa María, Valparaíso, Chile*
*** *Advanced Center for Electrical and Electronic Engineering, AC3E, Valparaíso, Chile*
**** *Division of Decision and Control Systems, KTH Royal Institute of Technology, Stockholm, Sweden*

Abstract: This paper concerns the identification of continuous-time systems in state-space form that are subject to Lebesgue sampling. Contrary to equidistant (Riemann) sampling, Lebesgue sampling consists of taking measurements of a continuous-time signal whenever it crosses fixed and regularly partitioned thresholds. The knowledge of the intersample behavior of the output data is exploited in this work to derive an expectation-maximization (EM) algorithm for parameter estimation of the state-space and noise covariance matrices. For this purpose, we use the incremental discrete-time equivalent of the system, which leads to EM iterations of the continuous-time state-space matrices that can be computed by standard filtering and smoothing procedures. The effectiveness of the identification method is tested via Monte Carlo simulations.

Keywords: System identification; continuous-time systems; event-based sampling; expectation-maximization.

1. INTRODUCTION

In digital control system design, signals are usually sampled equidistantly in time. This approach has led to a well established system theory and a vast number of successful applications due to its analytical tractability (Åström and Wittenmark, 1984). An alternative sampling scheme, called event-based sampling, consists in retrieving measurements based on the occurrence of an event rather than the passing of time instants. One of the most popular event-based sampling schemes is *Lebesgue sampling*, and it consists of sampling a signal whenever it crosses fixed and regularly-partitioned thresholds.

First steps in modern Lebesgue sampling theory can be found in Åström and Bernhardsson (1999), where a comparison between first-order systems using periodic and event-based sampling was made. The take-away message is that the Lebesgue sampling scheme requires fewer measurements on average than the equidistant (Riemann) sampling scheme, which is natural since the sampling is done at arguably the most relevant instants. Another advantage of Lebesgue sampling is that, in principle, only

1 bit is needed to indicate that the signal has crossed a threshold, which can be of interest when dealing with communication network systems.

One of the issues of standard Lebesgue sampling is that incorporating continuous-time noise in the model description leads to theoretical difficulties when defining the sampling time instants. This issue is overcome in this paper by considering a send-on-delta sampling strategy (Miskowicz, 2006) that includes a quantizer with hysteresis. Such type of sampling has been explored by, e.g., Kofman and Braslavsky (2006), and in addition to its theoretical advantages, it is convenient for implementing 1-bit coding communication and minimizing spurious sampling.

In this work, we study how to identify continuous-time systems in state-space using continuous-time input data and Lebesgue-sampled output data. Although this framework resembles the discrete-time identification problem with quantized data, key differences can be observed in the continuous-time treatment of the noise, and the hysteresis effect of the quantization step. Some contributions in such framework can be found in, e.g., Gustafsson and Karlsson (2009) and Bottegal et al. (2017). In Kawaguchi et al. (2016), transfer function identification under Lebesgue sampling was studied. This work considered an *approximate* Lebesgue sampling scheme, which has the shortcoming that the output data are not threshold values, but real numbers. Thus, it is not suitable for scenarios with

very limited communication bandwidth or computational resources. Other contributions (Sánchez et al., 2019) has focused on the design of experiments for transfer function identification in closed-loop with an event-based sampling scheme. These methods tune a controller so that the system enters into a limit cycle, facilitating the estimation. In contrast to these procedures, our approach admits any order for the continuous-time system, does not require controller tuning, is suited for open-loop identification, and can be performed on any input and output data set.

This work overcomes the shortcomings detailed above by providing an EM-based algorithm for identifying systems subject to Lebesgue sampling. In summary,

- We present closed-form expressions for the E and M-steps of the EM algorithm tailored for the identification of Lebesgue-sampled linear continuous-time systems described in state-space form. This algorithm delivers maximum likelihood estimates at convergence under mild conditions. Two forms are discussed, which use discrete-time equivalents of the system in shift operator and delta operator forms. The discrete-time equivalents rely on a *user-defined* sampling period Δ that may improve the estimation accuracy (for small Δ) at the cost of a higher computational cost;
- We provide two alternatives for computing the filtering and smoothing procedures required for the EM iterations; and
- We show the effectiveness of the proposed method via extensive Monte Carlo simulations.

The rest of the paper is organized as follows. In Section 2 the problem statement is described, and the EM algorithm for Lebesgue-sampled system identification is derived in Section 3. Section 4 illustrates the method with a numerical example, and conclusions are drawn in Section 5. Proofs of the main results can be found in the Appendix.

Notation: All matrices and vectors are written in bold, and column vectors are utilized, unless transposed. We employ the notation $\{f(t_l)\}_{l=1}^M$ to denote the set of evaluations $\{f(t_1), f(t_2), \ldots, f(t_M)\}$, and $\{x(t)\}_{t\in[t_1, t_M]}$ to denote the continuous-time signal defined on the closed interval $[t_1, t_M]$. A discrete-time signal is also written as $\{f_k\}_{k=1}^M$, depending on the context. The notation $\mathbf{x}_{1:N}$ describes $\{\mathbf{x}_k\}_{k=1}^N$. The Kronecker delta function is denoted as δ_k^K, and the Dirac delta distribution is written as $\delta(t)$.

2. PROBLEM FORMULATION

We consider the following linear time-invariant (LTI), single-input, single-output, continuous-time model:

$$\dot{\mathbf{x}}(t) = \mathbf{A}(\boldsymbol{\theta})\mathbf{x}(t) + \mathbf{B}(\boldsymbol{\theta})u(t) + \dot{\mathbf{w}}(t), \quad (1a)$$
$$z(t) = \mathbf{C}(\boldsymbol{\theta})\mathbf{x}(t) + D(\boldsymbol{\theta})u(t), \quad (1b)$$

where $\mathbf{A}(\boldsymbol{\theta}), \mathbf{B}(\boldsymbol{\theta}), \mathbf{C}(\boldsymbol{\theta})$, and $D(\boldsymbol{\theta})$ are $\boldsymbol{\theta}$-dependent matrices of suitable dimensions, and $\dot{\mathbf{w}}(t)$ is the formal derivative of a Wiener process of finite incremental covariance $\mathbf{Q}(\boldsymbol{\theta})$. The initial condition $\mathbf{x}(0)$ is assumed to be Gaussian-distributed with mean $\boldsymbol{\mu}_1$ and covariance $\mathbf{P}_1$, and the continuous-time white noise $\dot{\mathbf{w}}(t)$ is also assumed Gaussian with zero mean.
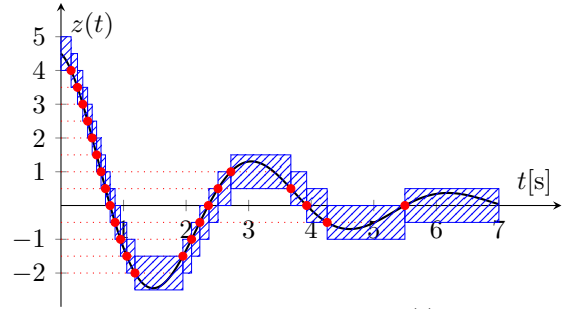

Fig. 1. Lebesgue sampling of a signal $z(t)$ with threshold $\tau = 0.5$. Red dots indicate the sampling instants and thresholds being crossed, and dashed blue rectangles show the regions where $z(t)$ is known to be located.

We now introduce the sampling scheme of interest in this paper, which is also carefully explained by Kofman and Braslavsky (2006). Given $\tau > 0$ and the continuous-time output $z(t)\colon \mathbb{R} \to \mathbb{R}$, we define the sampled sequence $\{y(t_l)\}_{l=0}^\infty$ with quantization interval h by the piecewise constant function $y(t)\colon \mathbb{R} \to \mathbb{R}$ that satisfies

$$y(t) = \mathcal{Q}_\tau\{z\}(t) := \begin{cases} \lfloor z(t_0)/\tau \rfloor \tau, & \text{if } t_0 \le t < t_1, \\ z(t_l), & \text{if } t_l \le t < t_{l+1}, l \in \mathbb{N}, \end{cases} \quad (2)$$

where $\lfloor \cdot \rfloor$ denotes the floor function, and where the sampling points $\{t_l\}_{l=1}^\infty$ are defined by

$$t_l = \inf\{T \in (t_{l-1}, \infty)\colon |z(T) - z(t_{l-1})| > \tau\}. \quad (3)$$

Unlike level-crossing sampling, this setup implies that an output that consecutively crosses the same threshold more than once only triggers one event. This property helps avoid situations where many samples would be sent in a short time-span due to noise. The drawback of this hysteresis feature in the sampling procedure is that it introduces a dynamic nonlinearity; in other words, the event of sending a sample will depend on the previous threshold crossed.

An example of the Lebesgue sampling we consider is shown in Fig. 1. Our goal is to estimate the system parameters that describe the matrices $(\mathbf{A}, \mathbf{B}, \mathbf{C}, D, \mathbf{Q})$ using the input $\{u(t)\}_{t\in[t_1, t_M]}$ and the Lebesgue-sampled output $\{y(t_l)\}_{l=1}^M$.

3. LEBESGUE-SAMPLED SYSTEM IDENTIFICATION

In this work, we derive the maximum likelihood estimate of the system matrices using an equivalent discrete-time description of (1) for fast sampling rates. The intuition is that the output data $\{y(t_l)\}_{l=1}^M$ provide knowledge of what amplitude band $z(t)$ is located in at any instant of time; thus, the framework is asymptotically equivalent, as the sampling period tends to zero, to computing the maximum likelihood estimate of a system with a fastly-sampled output that is quantized in a special manner.

We begin describing our approach by recalling a shift-operator equivalent and an incremental equivalent description of the system (1) in Lemma 1. In the sequel, we denote any sampled signal $\rho(k\Delta)$ as ρ_k, where $k \in \mathbb{N}$. Note that in our application, k ranges between $\lfloor \frac{t_1}{\Delta} \rfloor$ and $\lfloor \frac{t_M}{\Delta} \rfloor$. For simplicity we relabel the discrete-time signal so that k ranges between $k = 1$ and $k = N := \lfloor \frac{t_M}{\Delta} \rfloor - \lfloor \frac{t_1}{\Delta} \rfloor + 1$.

Lemma 1. Consider the continuous-time state-space model in (1), where $u(t)$ is generated by a zero-order-hold with a sampling period Δ. This system has the same second-order output properties at the sampling instants $t = k\Delta$ as the following discrete-time model in shift-operator form:

$$q\mathbf{x}_k := \mathbf{x}_{k+1} = \mathbf{A}_\mathrm{d}\mathbf{x}_k + \mathbf{B}_\mathrm{d}u_k + \tilde{\mathbf{w}}_k, \quad (4a)$$

$$z_k = \mathbf{C}\mathbf{x}_k + Du_k, \quad (4b)$$

where the matrices $\mathbf{A}_\mathrm{d}$ and $\mathbf{B}_\mathrm{d}$ are given by

$$\mathbf{A}_\mathrm{d} = e^{\mathbf{A}\Delta}, \quad \mathbf{B}_\mathrm{d} = \int_0^\Delta e^{\mathbf{A}s}\mathrm{d}s\mathbf{B}, \quad (5)$$

and the covariance of the noise vector is given by $\mathbb{E}\left\{\tilde{\mathbf{w}}_k\tilde{\mathbf{w}}_l^\top\right\} = \mathbf{Q}_\mathrm{d}\delta_{k-l}^K$, with

$$\mathbf{Q}_\mathrm{d} = \int_0^\Delta e^{\mathbf{A}s}\mathbf{Q}e^{\mathbf{A}^\top s}\mathrm{d}s. \quad (6)$$

Alternatively, the discrete-time model derived above can also be written in its incremental or delta-operator form:

$$\mathrm{d}\mathbf{x}_k^+ = \Delta\mathbf{A}_\mathrm{in}\mathbf{x}_k + \Delta\mathbf{B}_\mathrm{in}u_k + \mathrm{d}\mathbf{w}_k^+, \quad (7a)$$

$$z_k = \mathbf{C}\mathbf{x}_k + Du_k, \quad (7b)$$

where the increments are defined as $\mathrm{d}f_k^+ := f_{k+1} - f_k$, the matrices of the state equation are given by $\mathbf{A}_\mathrm{in} = (\mathbf{A}_\mathrm{d} - \mathbf{I})/\Delta$, $\mathbf{B}_\mathrm{in} = \mathbf{B}_\mathrm{d}/\Delta$, and the covariance of the noise vector is $\mathbb{E}\{\mathrm{d}\mathbf{w}_k^+\mathrm{d}\mathbf{w}_l^{+\top}\} = \Delta\mathbf{Q}_\mathrm{in}\delta_{k-l}^K$, where $\mathbf{Q}_\mathrm{in} = \mathbf{Q}_\mathrm{d}/\Delta$.

Proof. The proof for the incremental model equivalence can be found in, e.g., (Åström, 1970, Sec. 3.10), while the equivalent shift-operator model follows from expanding the increment notation in (7) and rearranging terms. $\square$

The shift-operator model in (4) allows for the direct implementation of many filtering and smoothing algorithms. On the other hand, the model in (7) provides a natural way to describe a continuous-time system whose output is sampled at a fast rate, since the continuous-time matrices are recovered when $\Delta \to 0$. That is,

$$(\mathbf{A}_\mathrm{in}, \mathbf{B}_\mathrm{in}, \mathbf{Q}_\mathrm{in}) \xrightarrow{\Delta \to 0} (\mathbf{A}, \mathbf{B}, \mathbf{Q}). \quad (8)$$

Note that in our approach there is no need to physically include an additional sampling step, and we do not assume that there is a fast-sampling mechanism prior to the event sampler. However, this might be the case in some applications related to incremental encoders, in which Δ may represent the sampling rate of the high-resolution clock (Merry et al., 2013). Also, keep in mind that the sampling points t_l need not be multiples of the fast-sampling period Δ, although this is usually assumed in encoder setups (Strijbosch and Oomen, 2022).

The next step is to derive an algorithm that computes an estimate for the system matrices using the discrete-time equivalent of (1). We use the EM algorithm (Dempster et al., 1977) for this purpose, which will be applied taking into consideration the shift-operator model (4), as well as the incremental model (7), with Lebesgue-sampled output data. To this end, we require formulas for the expectation and maximization steps to implement the EM method. This derivation must include an output perturbation to ensure that the noise covariance matrix in the EM algorithm is full rank (Solo, 2003). Thus, we add a zero-mean, independent, Gaussian perturbation v_k with a user-defined variance $\epsilon^2 \ll 1$ to z_k:

$$z_k = \mathbf{C}\mathbf{x}_k + Du_k + v_k. \quad (9)$$

We stack the parameters describing the state-space matrices in a vector $\boldsymbol{\theta}$, and fix $\{\mathbf{x}_{1:N+1}, z_{1:N}\}$ as the set of unobserved latent data. Based on the estimate at the ith iteration $\hat{\boldsymbol{\theta}}_i$, we need to compute

$$Q(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}_i) = \mathbb{E}\left\{\log \mathrm{p}(\mathbf{x}_{1:N+1}, z_{1:N}|\boldsymbol{\theta})|y_{1:N}, \hat{\boldsymbol{\theta}}_i\right\}, \quad (10)$$

where $\mathrm{p}(\mathbf{x}_{1:N+1}, z_{1:N}|\boldsymbol{\theta})$ is the probability density function (PDF) of $\{\mathbf{x}_{1:N+1}, z_{1:N}\}$ given the model parameters $\boldsymbol{\theta}$. The function in (10) is later maximized with respect to $\boldsymbol{\theta}$, yielding the new parameter estimate:

$$\hat{\boldsymbol{\theta}}_{i+1} = \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}_i).$$

The iterations presented above are performed until the estimate has converged within a predefined tolerance value. In the following subsections, we derive the EM algorithm for the shift and delta operator models and discuss the required filtering and smoothing procedures.

3.1 E-step for the shift-operator model

Our goal is to estimate the matrices $\mathbf{A}_\mathrm{d}, \mathbf{B}_\mathrm{d}, \mathbf{C}, D$ and $\mathbf{Q}_\mathrm{d}$, which ultimately must be transformed into their continuous-time equivalents. We assume for this derivation that $\boldsymbol{\theta}$ is constituted by the parameters of the shift-operator model matrices. In order to compute the Q function in (10), we write the logarithm of the joint probability density function of the extended state vector data $\mathcal{L}(\boldsymbol{\theta}) = \mathrm{p}(\mathbf{x}_{1:N+1}, z_{1:N}|\boldsymbol{\theta})$ as

$$-2\log\mathcal{L}(\boldsymbol{\theta}) = -2\log\mathrm{p}(\mathbf{x}_1|\boldsymbol{\theta}) - 2\sum_{k=1}^N \log\mathrm{p}(\mathbf{x}_{k+1}, z_k|\mathbf{x}_k, \boldsymbol{\theta})$$

$$= -2\log\left(\frac{\mathrm{p}(\mathbf{x}_1|\boldsymbol{\theta})}{\sqrt[N]{2\pi\epsilon^2}}\right) + N\log\det(2\pi\mathbf{Q}_\mathrm{d})$$

$$+ \frac{1}{\epsilon^2}\sum_{k=1}^N \varphi_k^2 + \sum_{k=1}^N \mathbf{V}_k^\top\mathbf{Q}_\mathrm{d}^{-1}\mathbf{V}_k,$$

where $\varphi_k = z_k - \mathbf{C}\mathbf{x}_k - Du_k$ and $\mathbf{V}_k = \mathbf{x}_{k+1} - \mathbf{A}_\mathrm{d}\mathbf{x}_k - \mathbf{B}_\mathrm{d}u_k$. After some embellishments, $Q(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}_i)$ can be expressed as

$$-2Q(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}_i) = L_0(\hat{\boldsymbol{\theta}}_i) + N\log\det(\mathbf{Q}_\mathrm{d})$$

$$+ \frac{1}{\epsilon^2}\left(D^2\Gamma_{uu} - 2\mathbf{C}\Gamma_{xz} - 2D\Gamma_{uz} + 2D\Gamma_{ux}\mathbf{C}^\top + \mathbf{C}\Gamma_{xx}\mathbf{C}^\top\right)$$

$$+ \mathrm{tr}\left\{\mathbf{Q}_\mathrm{d}^{-1}\left(\Gamma_{qq} + \mathbf{A}_\mathrm{d}\Gamma_{xx}\mathbf{A}_\mathrm{d}^\top + \mathbf{B}_\mathrm{d}\Gamma_{uu}\mathbf{B}_\mathrm{d}^\top - \mathbf{A}_\mathrm{d}\Gamma_{xq} - \Gamma_{xq}^\top\mathbf{A}_\mathrm{d}^\top\right.\right.$$

$$\left.\left. - \mathbf{B}_\mathrm{d}\Gamma_{uq} - \Gamma_{uq}^\top\mathbf{B}_\mathrm{d}^\top + \mathbf{A}_\mathrm{d}\Gamma_{ux}^\top\mathbf{B}_\mathrm{d}^\top + \mathbf{B}_\mathrm{d}\Gamma_{ux}\mathbf{A}_\mathrm{d}^\top\right)\right\},$$

where $L_0(\hat{\boldsymbol{\theta}}_i)$ accounts for all terms solely depending on $\hat{\boldsymbol{\theta}}_i$ or constants, and

$$\Gamma_{xx} = \bar{\mathbb{E}}\{\mathbf{x}_k\mathbf{x}_k^\top\}, \qquad \Gamma_{qq} = \bar{\mathbb{E}}\{\mathbf{x}_{k+1}\mathbf{x}_{k+1}^\top\}, \quad (11a)$$

$$\Gamma_{xq} = \bar{\mathbb{E}}\{\mathbf{x}_k\mathbf{x}_{k+1}^\top\}, \qquad \Gamma_{ux} = \bar{\mathbb{E}}\{u_k\mathbf{x}_k^\top\}, \quad (11b)$$

$$\Gamma_{uq} = \bar{\mathbb{E}}\{u_k\mathbf{x}_{k+1}^\top\}, \qquad \Gamma_{xz} = \bar{\mathbb{E}}\{\mathbf{x}_kz_k\}, \quad (11c)$$

$$\Gamma_{uz} = \bar{\mathbb{E}}\{u_kz_k\}, \qquad \Gamma_{uu} = \bar{\mathbb{E}}\{u_k^2\}, \quad (11d)$$

where $\bar{\mathbb{E}}\{\cdot\} := \sum_{k=1}^N \mathbb{E}\{\cdot|y_{1:N}, \hat{\boldsymbol{\theta}}_i\}$.

3.2 Computation of the state moments

In order to compute the quantities in (11) that determine the auxiliary function Q, it is necessary to compute some moments related to the system state $\mathbf{x}_k$, the output z_k,

and their corresponding cross moments. This computation requires the evaluation of filtering and smoothing distributions of the extended vector $\mathbf{x}_k^{\mathrm{e}} := [\mathbf{x}_k^\top\ z_k]^\top$ conditioned on the measured data, i.e., the PDFs $\mathrm{p}(\mathbf{x}_k^{\mathrm{e}}|y_{1:k})$ and $\mathrm{p}(\mathbf{x}_k^{\mathrm{e}}|y_{1:N})$. Several methods allow the computation of these PDFs; one recent and promising approach is the Gaussian Sum Filter and Smoother developed in Cedeño et al. (2021a,b), where the desired PDFs are represented by a Gaussian Mixture Model. An alternative approach to obtain the moments of $\mathbf{x}_k$ and z_k is the Sequential Monte Carlo sampling approach, also called particle filter/smoother (PF/PS) (Gordon et al., 1993; Doucet et al., 2000). In this approach, filtering and smoothing distributions are represented by using a set of weighted random samples called particles so that

$$\mathrm{p}(\mathbf{x}_k|y_{1:k}) \approx \sum_{i=1}^{\mathcal{M}} w_k^{(i)} \delta\left(\mathbf{x}_k - \mathbf{x}_k^{(i)}\right), \qquad (12)$$

$$\mathrm{p}(\mathbf{x}_k|y_{1:N}) \approx \sum_{i=1}^{\mathcal{M}} w_{k|N}^{(i)} \delta\left(\mathbf{x}_k - \tilde{\mathbf{x}}_k^{(i)}\right), \qquad (13)$$

where $w_k^{(i)}$ and $w_{k|N}^{(i)}$ denote the ith weights, and $\mathbf{x}_k^{(i)}$ with $\tilde{\mathbf{x}}_k^{(i)}$ denote the ith particles sampled from the filtering and smoothing PDFs $\mathrm{p}(\mathbf{x}_k|y_{1:k})$ and $\mathrm{p}(\mathbf{x}_k|y_{1:N})$, respectively. The quantity $\mathcal{M}$ is the number of particles. The importance weight computation can be carried out recursively (Sequential Importance Sampling) as follows:

$$w_k^{(i)} \propto w_{k-1}^{(i)} \frac{\mathrm{p}(y_k|\mathbf{x}_t^{(i)})\mathrm{p}(\mathbf{x}_k^{(i)}|\mathbf{x}_{k-1}^{(i)})}{\mathrm{h}(\mathbf{x}_k|\mathbf{x}_{k-1}^{(i)}, y_k)},$$

$$w_{k|N}^{(i)} = \sum_{j=1}^{\mathcal{M}} w_{k+1|N}^{(j)} \frac{w_k^{(i)}\mathrm{p}(\mathbf{x}_{k+1}^{(j)}|\mathbf{x}_k^{(i)})}{\sum_{k=1}^{\mathcal{M}} w_k^{(k)}\mathrm{p}(\mathbf{x}_{k+1}^{(j)}|\mathbf{x}_k^{(k)})},$$

where $\mathrm{h}(\mathbf{x}_k|\mathbf{x}_{k-1}^{(i)}, y_k)$ is the importance density, $w_{N|N}^{(i)} = w_N^{(i)}$ for $i = 1, \dots, \mathcal{M}$, and $w_{k-1}^{(i)}$ are the importance weights of the previous iteration. Details of the implementation and comparison of particle filters/smoothers to deal with quantized data can be found in Cedeño et al. (2023). Once the PF and PS are implemented, the particles of the smoothing distributions can be used to approximate the moments and cross moments of $\mathbf{x}_k$ and z_k as follows:

$$\mathbb{E}\{g(\mathbf{x}_k)|y_{1:N}\} \approx \sum_{i=1}^{\mathcal{M}} w_{k|N}^{(i)} g(\tilde{\mathbf{x}}_k^{(i)}),$$

where $w_{k|N}^{(i)}$ and $\tilde{\mathbf{x}}_k^{(i)}$ are the weights and particles (from the PS), and $g(\mathbf{x}_k)$ is a function of $\mathbf{x}_k$, for instance $g(\mathbf{x}_k) = \mathbf{x}_k$ or $g(\mathbf{x}_k) = (\mathbf{x}_k - \mathbb{E}\{\mathbf{x}_k\})(\mathbf{x}_k - \mathbb{E}\{\mathbf{x}_k\})^\top$. The corresponding moments of z_k are approximated by

$$\mathbb{E}\{z_k|y_{1:N}\} \approx \sum_{i=1}^{\mathcal{M}} w_{k|N}^{(i)} \hat{z}_k, \quad \mathbb{E}\{\mathbf{x}_k z_k|y_{1:N}\} \approx \sum_{i=1}^{\mathcal{M}} w_{k|N}^{(i)} \tilde{\mathbf{x}}_k^{(i)} \hat{z}_k,$$

with $\hat{z}_k = \mathbb{E}\{z_k|\tilde{\mathbf{x}}_k^{(i)}, y_k\}$ being the mean of the truncated Gaussian distribution given by

$$\hat{z}_k = \mathbf{C}\tilde{\mathbf{x}}_k^{(i)} + Du_k + \epsilon\frac{\psi(a_k, b_k)}{\Psi(a_k, b_k)},$$

$$\psi(a_k, b_k) = \phi\left[\frac{a_k - \mathbf{C}\tilde{\mathbf{x}}_k^{(i)} - Du_k}{\epsilon}\right] - \phi\left[\frac{b_k - \mathbf{C}\tilde{\mathbf{x}}_k^{(i)} - Du_k}{\epsilon}\right],$$

$$\Psi(a_k, b_k) = \Phi\left[\frac{a_k - \mathbf{C}\tilde{\mathbf{x}}_k^{(i)} - Du_k}{\epsilon}\right] - \Phi\left[\frac{b_k - \mathbf{C}\tilde{\mathbf{x}}_k^{(i)} - Du_k}{\epsilon}\right],$$

where $[a_k, b_k]$ is an interval defined by the threshold regions and the output y_k, $\phi[x]$ is the standard normal density, and $\Phi[x]$ is its cumulative distribution function.

3.3 M-step for the shift-operator model

Now we need to maximize $Q(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}_i)$ with respect to $\boldsymbol{\theta}$. The following result provides the EM iterations that are proposed for estimating the discrete-time equivalent of the state-space matrices of interest.

Theorem 2. The matrices that maximize $Q(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}_i)$ for the shift-operator model are given by

$$\begin{bmatrix} \mathbf{A}_{\mathrm{d},i+1} & \mathbf{B}_{\mathrm{d},i+1} \\ \mathbf{C}_{i+1} & D_{i+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Gamma}_{xq} & \boldsymbol{\Gamma}_{xz} \\ \boldsymbol{\Gamma}_{uq} & \Gamma_{uz} \end{bmatrix}^\top \begin{bmatrix} \boldsymbol{\Gamma}_{xx} & \boldsymbol{\Gamma}_{ux}^\top \\ \boldsymbol{\Gamma}_{ux} & \boldsymbol{\Gamma}_{uu} \end{bmatrix}^{-1}, \qquad (14)$$

$$\mathbf{Q}_{\mathrm{d},i+1} = \frac{1}{N}\left(\boldsymbol{\Gamma}_{qq} - \begin{bmatrix} \boldsymbol{\Gamma}_{xq} \\ \boldsymbol{\Gamma}_{uq} \end{bmatrix}^\top \begin{bmatrix} \boldsymbol{\Gamma}_{xx} & \boldsymbol{\Gamma}_{ux}^\top \\ \boldsymbol{\Gamma}_{ux} & \boldsymbol{\Gamma}_{uu} \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{\Gamma}_{xq} \\ \boldsymbol{\Gamma}_{uq} \end{bmatrix}\right), \quad (15)$$

where all the $\boldsymbol{\Gamma}$ matrices are given in (11).

Proof. See Appendix A. □

3.4 EM algorithm for the incremental model

Theorem 2 provides the EM iterations for the shift-operator model (4). However, since our interest is in the continuous-time state-space matrices, we must revert the discretization process of Lemma 1 in some way. One alternative is to estimate the discrete-time matrices and later transform them to continuous-time, although this step is known to be ill-conditioned for small sampling periods (Garnier and Young, 2014). Another option for small Δ is to derive the EM iterations for the incremental model instead, and let the incremental state-space matrices represent the continuous-time ones according to (8), i.e., $(\mathbf{A}_{\mathrm{in}}, \mathbf{B}_{\mathrm{in}}, \mathbf{Q}_{\mathrm{in}}) \approx (\mathbf{A}, \mathbf{B}, \mathbf{Q})$. This is the approach suggested by Yuz et al. (2011) for the identification of stochastic differential equations, and it is convenient since it leads to explicit iterations of the EM algorithm for the continuous-time system parameters, as seen next.

Theorem 3. The EM iterations for the incremental model (7) with Lebesgue-sampled output data are given by

$$\begin{bmatrix} \mathbf{A}_{\mathrm{in},i+1} & \mathbf{B}_{\mathrm{in},i+1} \\ \mathbf{C}_{i+1} & D_{i+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Gamma}_{x\delta} & \boldsymbol{\Gamma}_{xz} \\ \boldsymbol{\Gamma}_{u\delta} & \Gamma_{uz} \end{bmatrix}^\top \begin{bmatrix} \boldsymbol{\Gamma}_{xx} & \boldsymbol{\Gamma}_{ux}^\top \\ \boldsymbol{\Gamma}_{ux} & \boldsymbol{\Gamma}_{uu} \end{bmatrix}^{-1}, \qquad (16)$$

$$\mathbf{Q}_{\mathrm{in},i+1} = \frac{\Delta}{N}\left(\boldsymbol{\Gamma}_{\delta\delta} - \begin{bmatrix} \boldsymbol{\Gamma}_{x\delta} \\ \boldsymbol{\Gamma}_{u\delta} \end{bmatrix}^\top \begin{bmatrix} \boldsymbol{\Gamma}_{xx} & \boldsymbol{\Gamma}_{ux}^\top \\ \boldsymbol{\Gamma}_{ux} & \boldsymbol{\Gamma}_{uu} \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{\Gamma}_{x\delta} \\ \boldsymbol{\Gamma}_{u\delta} \end{bmatrix}\right), \quad (17)$$

where $\boldsymbol{\Gamma}_{xx}, \boldsymbol{\Gamma}_{ux}, \boldsymbol{\Gamma}_{xz}, \Gamma_{uz}$ and Γ_{uu} are given in (11), and

$$\boldsymbol{\Gamma}_{x\delta} = \Delta^{-1}\bar{\mathbb{E}}\left\{\mathbf{x}_k(\mathbf{x}_{k+1} - \mathbf{x}_k)^\top\right\}, \qquad (18a)$$

$$\boldsymbol{\Gamma}_{u\delta} = \Delta^{-1}\bar{\mathbb{E}}\left\{u_k(\mathbf{x}_{k+1} - \mathbf{x}_k)^\top\right\}, \qquad (18b)$$

$$\boldsymbol{\Gamma}_{\delta\delta} = \Delta^{-2}\bar{\mathbb{E}}\left\{(\mathbf{x}_{k+1} - \mathbf{x}_k)(\mathbf{x}_{k+1} - \mathbf{x}_k)^\top\right\}. \qquad (18c)$$

Proof. See Appendix B. □

It is well known that the converging point of EM iterations belongs to the set of stationary points of the likelihood function. Thus, if adequately initialized and under mild

conditions regarding the model structure and input excitation, the iterations in Theorem 3 provide the maximum likelihood estimate of the system parameters in its incremental form. These estimates are biased estimates of the continuous-time parameters due to the approximation $(\mathbf{A}_{\text{in}}, \mathbf{B}_{\text{in}}, \mathbf{Q}_{\text{in}}) \approx (\mathbf{A}, \mathbf{B}, \mathbf{Q})$. This bias can be shown to be proportional to the fast-sampling period Δ; explicit bounds will be published elsewhere.

4. SIMULATIONS

In this section, we present a numerical example to analyze the performance of the proposed method (PS-EM), in which we utilize the particle smoother to compute the EM iterations. We compare PS-EM to the standard method to identify state-space models with the Kalman Smoother (KS-EM, Gibson and Ninness (2005)), which does not consider any type of quantization. The continuous-time system we consider is given by

$$\dot{x}(t) = -x(t) + 0.7u(t) + \dot{w}(t), \quad z(t) = x(t),$$

where the measured output is computed from (2) with $\tau = 0.3$, $\dot{w}(t) \sim \mathcal{N}(\dot{w}(t); 0, 0.5)$, and the input is sampled from $\mathcal{N}(u(t); 0, \sigma^2)$ with $\sigma = 10$. We use $N = 2000$ and $\Delta = 0.01$, and to compute the moments of x_k and z_k, we implement the particle filter with $\mathcal{M} = 1000$ particles. The frequency response of the true system and 100 Monte Carlo runs are shown in Fig. 2. The blue-shaded region represents the area where all Monte Carlo frequency responses lie, and the left and right plots show the response obtained with KS-EM and PS-EM, respectively. The results show that our approach produces more accurate estimates of the system than the KS-EM method. In addition, Fig. 3 presents boxplots of the parameter estimates that are invariant under similarity transformations. The proposed method PS-EM outperforms KS-EM in the system parameter estimation and is competitive against KS-EM in estimating the noise covariance matrix. Systems of higher order have also been tested and show similar results to this case study. In terms of computation time, the burden of our approach using 1000 particles and 2000 data points was approximately 36 minutes per Monte Carlo run. This agrees with the execution time reported for particle smoothing in, e.g., Cedeño et al. (2021a, 2023) multiplied by the total number of the EM iterations. The computer used has an Intel(R) Core(TM) i5-8300H CPU @ 2.30 GHz processor, and a RAM of 8.00 GB, with Windows 11 and MATLAB 2021b.

5. CONCLUSIONS

This paper introduces an identification method for continuous-time LTI systems with Lebesgue-sampled observations based on the maximum likelihood principle. We have used the EM algorithm to derive an iterative procedure that obtains an estimate of the system parameters and the noise covariance matrix. For this, we implemented a particle filter that evaluates the filtering and smoothing distributions that are required to compute the auxiliary function of the EM algorithm. The proposed method was compared to the standard approach that does not take into account the Lebesgue sampling of the output data, verifying that the former approach yields more accurate estimates of the system than the latter approach.

APPENDIX

5.1 Proof of Theorem 2

We first derive (14). If we consider only the terms that are dependent on $\mathbf{A}_{\text{d}}$ and $\mathbf{B}_{\text{d}}$, $Q(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}_i)$ is proportional to

$$-Q(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}_i) \propto \text{tr}\bigg\{ \mathbf{Q}_{\text{d}}^{-1}\bigg([\mathbf{A}_{\text{d}}\ \mathbf{B}_{\text{d}}] \begin{bmatrix} \boldsymbol{\Gamma}_{xx} & \boldsymbol{\Gamma}_{ux}^{\top} \\ \boldsymbol{\Gamma}_{ux} & \Gamma_{uu} \end{bmatrix} \begin{bmatrix} \mathbf{A}_{\text{d}}^{\top} \\ \mathbf{B}_{\text{d}}^{\top} \end{bmatrix}$$
$$-[\mathbf{A}_{\text{d}}\ \mathbf{B}_{\text{d}}] \begin{bmatrix} \boldsymbol{\Gamma}_{xq} \\ \boldsymbol{\Gamma}_{uq} \end{bmatrix} - [\boldsymbol{\Gamma}_{xq}^{\top}\ \boldsymbol{\Gamma}_{uq}^{\top}] \begin{bmatrix} \mathbf{A}_{\text{d}}^{\top} \\ \mathbf{B}_{\text{d}}^{\top} \end{bmatrix} \bigg) \bigg\}. \quad (19)$$

Using the fact that for any positive definite matrix $\mathbf{A}$ and any matrices $\mathbf{B}$ and $\mathbf{X}$ of suitable dimension we have
$$\mathbf{XAX}^{\top} - \mathbf{XB} - \mathbf{B}^{\top}\mathbf{X}^{\top} = (\mathbf{X} - \mathbf{B}^{\top}\mathbf{A}^{-1})\mathbf{A}(\mathbf{X} - \mathbf{B}^{\top}\mathbf{A}^{-1})^{\top} - \mathbf{B}^{\top}\mathbf{A}^{-1}\mathbf{B}$$
$$\succeq -\mathbf{B}^{\top}\mathbf{A}^{-1}\mathbf{B},$$

we exploit the partial ordering of the trace (Horn and Johnson, 2012, Corollary 7.7.4) to conclude that the minimum of (19) is achieved when $\mathbf{A}_{\text{d}}$ and $\mathbf{B}_{\text{d}}$ are as in (14). Proceeding similarly with $\mathbf{C}$ and D, we find that

$$-Q(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}_i) \propto \begin{bmatrix} \mathbf{C}^{\top} \\ D \end{bmatrix}^{\top} \begin{bmatrix} \boldsymbol{\Gamma}_{xx} & \boldsymbol{\Gamma}_{ux}^{\top} \\ \boldsymbol{\Gamma}_{ux} & \Gamma_{uu} \end{bmatrix} \begin{bmatrix} \mathbf{C}^{\top} \\ D \end{bmatrix} - \begin{bmatrix} \mathbf{C}^{\top} \\ D \end{bmatrix}^{\top} \begin{bmatrix} \boldsymbol{\Gamma}_{xz} \\ \Gamma_{uz} \end{bmatrix} - \begin{bmatrix} \boldsymbol{\Gamma}_{xz} \\ \Gamma_{uz} \end{bmatrix}^{\top} \begin{bmatrix} \mathbf{C}^{\top} \\ D \end{bmatrix}.$$

By the same reasoning as above, the minimum is achieved when $\mathbf{C}$ and D are as in (14). To obtain (15), we note that when $\mathbf{A}_{\text{d},i+1}$ and $\mathbf{B}_{\text{d},i+1}$ are inserted in $Q(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}_i)$ we have

$$-Q(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}_i) \propto N \log \det(\mathbf{Q}_{\text{d}})$$
$$+ \text{tr}\bigg\{ \mathbf{Q}_{\text{d}}^{-1}\bigg(\boldsymbol{\Gamma}_{qq} - \begin{bmatrix} \boldsymbol{\Gamma}_{xq} \\ \boldsymbol{\Gamma}_{uq} \end{bmatrix}^{\top} \begin{bmatrix} \boldsymbol{\Gamma}_{xx} & \boldsymbol{\Gamma}_{ux}^{\top} \\ \boldsymbol{\Gamma}_{ux} & \Gamma_{uu} \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{\Gamma}_{xq} \\ \boldsymbol{\Gamma}_{uq} \end{bmatrix} \bigg) \bigg\}. \quad (20)$$

Setting to zero the partial derivative of (20) with respect to $\mathbf{Q}_{\text{d}}$ indicates that $\mathbf{Q}_{\text{d},i+1}$ must satisfy

$$\mathbf{Q}_{\text{d},i+1}^{-1}\bigg(N\mathbf{Q}_{\text{d},i+1} - \boldsymbol{\Gamma}_{xx} + \begin{bmatrix} \boldsymbol{\Gamma}_{xq} \\ \boldsymbol{\Gamma}_{uq} \end{bmatrix}^{\top} \begin{bmatrix} \boldsymbol{\Gamma}_{xx} & \boldsymbol{\Gamma}_{ux}^{\top} \\ \boldsymbol{\Gamma}_{ux} & \Gamma_{uu} \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{\Gamma}_{xq} \\ \boldsymbol{\Gamma}_{uq} \end{bmatrix} \bigg)\mathbf{Q}_{\text{d},i+1}^{-1} = \mathbf{0},$$

which directly leads to (15), concluding the proof. $\quad\square$

5.2 Proof of Theorem 3

By leveraging the fact that $\mathbf{A}_{\text{d}} = \mathbf{I} + \Delta\mathbf{A}_{\text{in}}$ and $\mathbf{B}_{\text{d}} = \Delta\mathbf{B}_{\text{in}}$, the first block row of (14) can be expressed as

$$\Delta[\mathbf{A}_{\text{in},i+1}\ \mathbf{B}_{\text{in},i+1}] + [\mathbf{I}\ \mathbf{0}] = \begin{bmatrix} \boldsymbol{\Gamma}_{xq} \\ \boldsymbol{\Gamma}_{uq} \end{bmatrix}^{\top} \begin{bmatrix} \boldsymbol{\Gamma}_{xx} & \boldsymbol{\Gamma}_{ux}^{\top} \\ \boldsymbol{\Gamma}_{ux} & \Gamma_{uu} \end{bmatrix}^{-1}$$
$$\implies [\mathbf{A}_{\text{in},i+1}\ \mathbf{B}_{\text{in},i+1}] = \frac{1}{\Delta}\bigg(\begin{bmatrix} \boldsymbol{\Gamma}_{xq} \\ \boldsymbol{\Gamma}_{uq} \end{bmatrix} - \begin{bmatrix} \boldsymbol{\Gamma}_{xx} \\ \boldsymbol{\Gamma}_{ux} \end{bmatrix} \bigg)^{\top} \begin{bmatrix} \boldsymbol{\Gamma}_{xx} & \boldsymbol{\Gamma}_{ux}^{\top} \\ \boldsymbol{\Gamma}_{ux} & \Gamma_{uu} \end{bmatrix}^{-1}$$
$$= \begin{bmatrix} \boldsymbol{\Gamma}_{x\delta} \\ \boldsymbol{\Gamma}_{u\delta} \end{bmatrix}^{\top} \begin{bmatrix} \boldsymbol{\Gamma}_{xx} & \boldsymbol{\Gamma}_{ux}^{\top} \\ \boldsymbol{\Gamma}_{ux} & \Gamma_{uu} \end{bmatrix}^{-1},$$

where $\boldsymbol{\Gamma}_{x\delta}$ and $\boldsymbol{\Gamma}_{u\delta}$ are as in (18). Similarly, we can write $\mathbf{Q}_{\text{d},i+1}$ in terms of $\boldsymbol{\Gamma}_{\delta\delta}, \boldsymbol{\Gamma}_{x\delta}, \boldsymbol{\Gamma}_{u\delta}, \boldsymbol{\Gamma}_{xx}, \boldsymbol{\Gamma}_{ux}$ and Γ_{uu} as

$$\mathbf{Q}_{\text{d},i+1} = \frac{1}{N}\bigg(\Delta^2\boldsymbol{\Gamma}_{\delta\delta} + \Delta\boldsymbol{\Gamma}_{x\delta} + \Delta\boldsymbol{\Gamma}_{x\delta}^{\top} + \boldsymbol{\Gamma}_{xx}$$
$$- \begin{bmatrix} \Delta\boldsymbol{\Gamma}_{x\delta} + \boldsymbol{\Gamma}_{xx} \\ \Delta\boldsymbol{\Gamma}_{u\delta} + \boldsymbol{\Gamma}_{ux} \end{bmatrix}^{\top} \begin{bmatrix} \boldsymbol{\Gamma}_{xx} & \boldsymbol{\Gamma}_{ux}^{\top} \\ \boldsymbol{\Gamma}_{ux} & \Gamma_{uu} \end{bmatrix}^{-1} \begin{bmatrix} \Delta\boldsymbol{\Gamma}_{x\delta} + \boldsymbol{\Gamma}_{xx} \\ \Delta\boldsymbol{\Gamma}_{u\delta} + \boldsymbol{\Gamma}_{ux} \end{bmatrix} \bigg).$$

However, replacing the following matrix identities above

$$\boldsymbol{\Gamma}_{xx} = \begin{bmatrix} \boldsymbol{\Gamma}_{xx} \\ \boldsymbol{\Gamma}_{ux} \end{bmatrix}^{\top} \begin{bmatrix} \boldsymbol{\Gamma}_{xx} & \boldsymbol{\Gamma}_{ux}^{\top} \\ \boldsymbol{\Gamma}_{ux} & \Gamma_{uu} \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{\Gamma}_{xx} \\ \boldsymbol{\Gamma}_{ux} \end{bmatrix}, \quad \boldsymbol{\Gamma}_{x\delta} = \begin{bmatrix} \boldsymbol{\Gamma}_{xx} \\ \boldsymbol{\Gamma}_{ux} \end{bmatrix}^{\top} \begin{bmatrix} \boldsymbol{\Gamma}_{xx} & \boldsymbol{\Gamma}_{ux}^{\top} \\ \boldsymbol{\Gamma}_{ux} & \Gamma_{uu} \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{\Gamma}_{x\delta} \\ \boldsymbol{\Gamma}_{u\delta} \end{bmatrix},$$
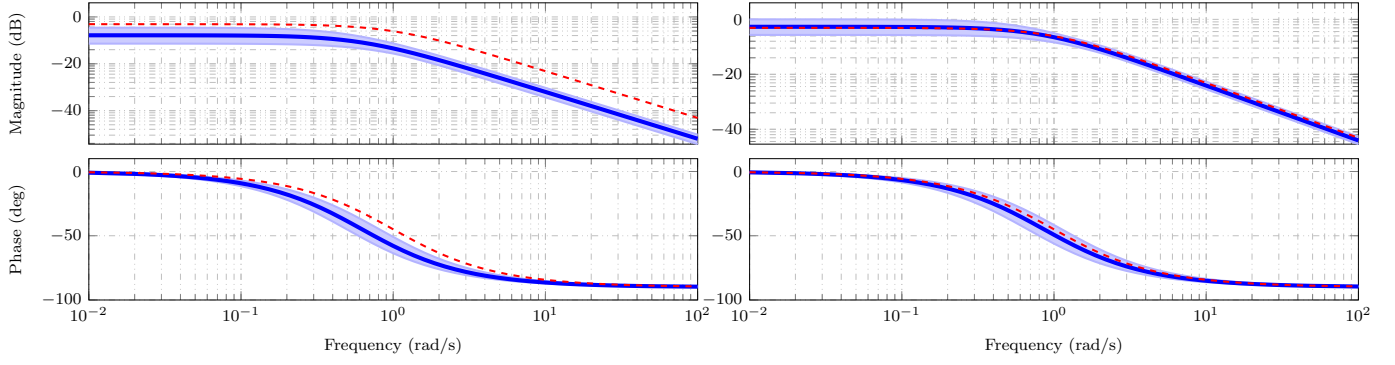
Fig. 2. Frequency response of the true and estimated system. Left: KS-EM; right: PS-EM (proposed method).
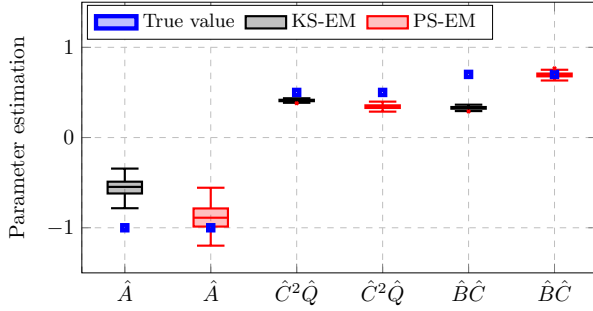

Fig. 3. Estimates of the system parameters that are invariant under similarity transformations.

we conclude that

$$\mathbf{Q}_{\text{in},i+1} = \frac{\mathbf{Q}_{\text{d},i+1}}{\Delta} = \frac{\Delta}{N} \left(\mathbf{\Gamma}_{\delta\delta} - \begin{bmatrix} \mathbf{\Gamma}_{x\delta} \\ \mathbf{\Gamma}_{u\delta} \end{bmatrix}^{\top} \begin{bmatrix} \mathbf{\Gamma}_{xx} & \mathbf{\Gamma}_{ux}^{\top} \\ \mathbf{\Gamma}_{ux} & \mathbf{\Gamma}_{uu} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{\Gamma}_{x\delta} \\ \mathbf{\Gamma}_{u\delta} \end{bmatrix} \right),$$

which is what we wanted to prove. □

REFERENCES

Åström, K.J. (1970). *Introduction to Stochastic Control Theory.* Academic Press.

Åström, K.J. and Bernhardsson, B.M. (1999). Comparison of periodic and event based sampling for first-order stochastic systems. In *Proceedings of the 14th IFAC World congress*, volume 11, 301–306.

Åström, K.J. and Wittenmark, B. (1984). *Computer Controlled Systems: Theory and Design.* Prentice-Hall.

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

Cedeño, A.L., Albornoz, R., Carvajal, R., Godoy, B.I., and Agüero, J.C. (2021a). A two-filter approach for state estimation utilizing quantized output data. *Sensors*, 21(22).

Cedeño, A.L., Albornoz, R., Carvajal, R., Godoy, B.I., and Agüero, J.C. (2021b). On filtering methods for state-space systems having binary output measurements. *IFAC-PapersOnLine*, 54(7), 815–820.

Cedeño, A.L., González, R.A., Godoy, B.I., Carvajal, R., and Agüero, J.C. (2023). On Filtering and Smoothing Algorithms for Linear State-Space Models Having Quantized Output Data. *Mathematics*, 11(6).

Dempster, A.P., Laird, N.M., and Rubin, D.B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B*, 39(1), 1–22.

Doucet, A., Godsill, S., and Andrieu, C. (2000). On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and computing*, 10(3), 197–208.

Garnier, H. and Young, P.C. (2014). The advantages of directly identifying continuous-time transfer function models in practical applications. *International Journal of Control*, 87(7), 1319–1338.

Gibson, S. and Ninness, B. (2005). Robust maximum-likelihood estimation of multivariable dynamic systems. *Automatica*, 41(10), 1667–1682.

Gordon, N.J., Salmond, D.J., and Smith, A.F.M. (1993). Novel approach to nonlinear and non-Gaussian Bayesian state estimation. In *IEE proceedings-F*, volume 140, 107–113. IET.

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

Horn, R.A. and Johnson, C.R. (2012). *Matrix Analysis*, 2nd ed. Cambridge University Press.

Kawaguchi, T., Hikono, S., Maruta, I., and Adachi, S. (2016). System identification under Lebesgue sampling and its asymptotic property. In *55th IEEE Conference on Decision and Control (CDC)*, 2079–2084.

Kofman, E. and Braslavsky, J.H. (2006). Level crossing sampling in feedback stabilization under data-rate constraints. In *45th IEEE Conference on Decision and Control*, 4423–4428.

Merry, R.J.E., Van de Molengraft, M.J.G., and Steinbuch, M. (2013). Optimal higher-order encoder time-stamping. *Mechatronics*, 23(5), 481–490.

Miskowicz, M. (2006). Send-on-Delta concept: An event-based data reporting strategy. *Sensors*, 6(1), 49–63.

Sánchez, J., Guinaldo, M., Visioli, A., and Dormido, S. (2019). Identification and tuning methods for PI control systems based on symmetric send-on-delta sampling. *International Journal of Control, Automation and Systems*, 17(11), 2784–2795.

Solo, V. (2003). An EM algorithm for singular state space models. In *42nd IEEE International Conference on Decision and Control*, volume 4, 3457–3460.

Strijbosch, N. and Oomen, T. (2022). Iterative learning control for intermittently sampled data: Monotonic convergence, design, and applications. *Automatica*, 139, Article 110171.

Yuz, J.I., Alfaro, J.A., Agüero, J.C., and Goodwin, G.C. (2011). Identification of continuous-time state-space models from non-uniform fast-sampled data. *IET Control Theory & Applications*, 5(7), 842–855.