# BAT: Batch Bayesian Auto-Tuning for Nonlinear Kalman Estimators

**Anonymous Authors**[1]

## Abstract

We study how to enhance the auto-tuning process of all Nonlinear Kalman Estimators (NKEs) components by using additional information about all state variables. Traditional auto-tuning approaches based on Normalized Estimation Error Squared or Normalized Innovation Squared can not efficiently estimate all NKE components because they rely on ground truth state models (which are usually unavailable) or on a subset of measured data ($\mathbf{y}$) used to compute the innovation errors. In this work, we introduce an approach called Batch Bayesian Auto-Tuning (BAT) for NKEs that enables the use of all available measured data $\mathbf{y}$ (not just those selected for generating innovation errors) during the tuning process. This is done using a posterior distribution of all NKE components given $\mathbf{y}$, outside of the NKE recursive process, based on the equivalence between the posterior distributions used in batch and recursive Bayesian inference. We validate our theoretical studies with empirical experiments conducted on a synthetic dataset representing a bioprocess production, confirming the efficacy of the BAT over the baselines. The data and code used in this study are available on https://anonymous.4open.science/r/BAT.

## 1. Introduction

Nonlinear Kalman Estimators (NKEs) are nonlinear state estimators based on the Kalman Filter (KF) framework that uses a two-step recursive algorithmic process: Prediction and Update (Khodarahmi & Maihami, 2022; Särkkä & Svensson, 2023). NKEs are widely applicable in various fields today, and the most popular NKEs are (Akca & Efe, 2019): Extended Kalman Filter (EKF), Robust Kalman Filter (RKF), Unscented Kalman Filter (UKF), and Cubature Kalman Filter (CKF). The optimal performance of NKEs relies on the proper tuning of five components: i) the process noise covariance matrix ($\mathbf{Q}$), ii) the measurement noise covariance matrix ($\mathbf{R}$), iii) the initial state noise covariance matrix ($\mathbf{P}(0)$), iv) initial condition of state variables ($\mathbf{x}_0$) and v) parameters ($\boldsymbol{\theta}$) of the dynamic model (Li et al., 2021; Khodarahmi & Maihami, 2022; Särkkä & Svensson, 2023). However, the automatic tuning of all NKE components remains an open problem, despite the fact that the problem of tuning $\mathbf{Q}$ and $\mathbf{R}$ has been addressed in several studies (Boulkroune et al., 2023; Kim et al., 2022). Most auto-tuning methods are based on Normalized Estimation Error Squared (NEES) (Chen et al., 2023) or Normalized Innovation Squared (NIS) (Chen et al., 2023), and they cannot estimate all NKE components. Methods based on NEES require the true process model, which is impossible in most cases, and methods based on NIS work only with measured data related to the state variable used to produce the innovation errors. Since $\mathbf{x}_0$ and $\boldsymbol{\theta}$ might not have a direct, linear relationship with the estimation and innovation error, it is difficult to estimate them based on NEES and NIS. In addition, the problem of estimating the parameters $\boldsymbol{\theta}$ that are not shared with other components of the dynamic system is an example where methods based on NEES and NIS and state augmentation fail (Iglesias & Bolic, 2024).

A possible solution to auto-tune all NKE components is not relying only on the measured data of state variables selected to generate the innovation errors but also using measured data related to the other state variables of the system even if they are not used to generate innovation errors. This means including more information in the auto-tuning process related to all system state variables (Valderrama-Bahamóndez & Fröhlich, 2019; Huang et al., 2020; Alahmadi et al., 2020).

A real-world applications of NKEs is bioprocess monitoring in biomanufacturing with state variables vector $\mathbf{x} \in \mathbb{R}^n$ and set of measurements $\mathbf{y} \in \mathbb{R}^n$ , where $\mathbf{y} = \{\mathbf{y}_{online} \cup \mathbf{y}_{offline}\}$ and $\mathbf{x} = \{\mathbf{x}_{IE} \cup \mathbf{x}_{NIE}\}$. NKEs use the real-time measured data $\mathbf{y}_{online} \in \mathbb{R}^{m<n}$ (online measurement from the sensor) of state variables $\mathbf{x}_{IE} \in \mathbb{R}^{m<n}$ responsible by Innovation Errors (IE) to estimate the other state variables $\mathbf{x}_{NIE} \in \mathbb{R}^{n-m}$ that are Not responsible by Innovation Errors (NIE), and that are difficult to measure directly during the execution of a bioprocess (Iglesias & Bolic, 2022; Iglesias Jr et al., 2023; Iglesias & Bolic, 2024).

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

Generally, $\mathbf{x}_{NIE}$ (not measured in real time) are obtained from offline measurements $\mathbf{y}_{offline} \in \mathbb{R}^{n-m}$. This means collecting samples from the manufacturing process and analyzing them in a separate laboratory setting, often taking several days for the result to arrive (Chopda et al., 2022; Iglesias & Bolic, 2022; Papathanasiou et al., 2019). Therefore, if additional information about all state variables, such as offline measurements $\mathbf{y}_{offline}$ of $\mathbf{x}_{NIE}$ , is available, how can we include it in the auto-tuning process of all NKE components?

In this paper, we introduce an approach called Batch Bayesian Auto-Tuning (BAT) for NKEs to address the aforementioned problem. The proposed method enables the use of measured data available about all state variables $\mathbf{y}$ to auto-tune all NKE components by defining a posterior distribution of $\mathbf{x}_0$, $\boldsymbol{\theta}$, $\mathbf{P}(0)$, $\mathbf{Q}$ and $\mathbf{R}$ given $\mathbf{y}$ (called BAT posterior) outside of NKE recursive process. This is done based on the equivalence between the posterior distributions used in batch and recursive Bayesian inference. Through empirical evaluations on the bioprocess dataset, we validate that the BAT posterior can enable an efficient sampling from the distribution of $\mathbf{x}_0$, $\boldsymbol{\theta}$, $\mathbf{P}(0)$, $\mathbf{Q}$ and $\mathbf{R}$ by Markov Chain Monte Carlo (MCMC) to auto-tune any NKE. The code and data used in this work are available on anonymous repository to facilitate reproducibility. Our contributions can be summarized as follows: 1) To the best of our knowledge, we are the first to introduce an approach to automatically tune all components ($\mathbf{x}_0$ ,$\boldsymbol{\theta}$ , $\mathbf{P}(0)$, $\mathbf{Q}$ and $\mathbf{R}$) of any NKE, such as EKF, UKF, RKF and CKF. 2) We demonstrate that BAT empirically outperforms the baseline approaches for auto-tuning the EKF when executing two tasks that represent common real-world situations. 3) In addition to demonstrating the theoretical application of BAT in NKEs, we demonstrate the applicability of BAT with UKF and CKF in a classical problem as instructional material.

## 2. Related Work

The literature has reported several techniques to tune some NKE components; however, to the best of our knowledge, no offline or online approach to auto-tuning all NKE components has been reported yet. This section provides a short overview of these techniques, which can be grouped as online (real-time) and offline.

**Trial and error.** It is an offline approach that involves iteratively adjusting parameters based on observed outcomes and expert intuition (Sekarsari & Tata, 2021). While this method can be facilitated by an experienced individual, it is generally inefficient and labor-intensive, particularly for systems with high dimensionality. Furthermore, some authors concluded that the trial and error approach is not the right way to tune NKEs (Boulkroune et al., 2023).

**State augmentation.** It is an online approach that aims to auto-tune the parameters $\boldsymbol{\theta}$ of a process model in NKEs by expanding the state variable vector to include both the system states $\mathbf{x}_k$ and $\boldsymbol{\theta}$ (Yousefi-Darani et al., 2020; Aswal et al., 2022; Iglesias et al., 2021). The evolution of $\boldsymbol{\theta}$ based on the following equation $\boldsymbol{\theta}_k = \boldsymbol{\theta}_{k-1} + \delta_{k-1}$, where $\delta_{k-1}$ is a "small" noise process added at each time step. This approach is widely used due to the need to correct the prediction of a process model regarding state variables and to update the process model by evolving its parameters based on the corrections done (Haykin & Haykin, 2001). However, the increased dimensionality and complexity of state augmentation can make accurately estimating states and parameters challenging. Furthermore, it was proved recently that this approach fails to estimate the state and unshared parameters $\boldsymbol{\theta}_{up}$ (parameters not shared with other components of the dynamic system) simultaneously. The fail happens when a dynamic system with $\boldsymbol{\theta}_{up}$ is coupled with noise covariance matrix $\mathbf{P}(0)$ and $\mathbf{Q}$ with uncorrelated elements, along with just one measured state variable (Iglesias & Bolic, 2024). A solution to avoid this failure case is to estimate the $\boldsymbol{\theta}_{up}$ during the auto-tune process of NKE. Therefore, we explore this case in our empirical evaluation to show that BAT is a solution to this problem. A detailed description of this failure case can be seen in the Appendix D.

**Optimizing an objective function based on NEES and NIS.** The early offline methods to auto-tune NKEs are focused on optimizing objective functions based on metrics like NEES and /or NIS (Bar-Shalom et al., 2001; Chen et al., 2023; Lin & Kim, 2020; Burkhart et al., 2016; Sakai & Kuroda, 2010). Various optimization methods, such as Genetic Algorithms (Oshman & Shaviv, 2000), Particle Swarm Optimization (Tummala & Ramanarao, 2017), and particularly Bayesian Optimization (BO) (Chen et al., 2018; 2023; 2021), have been proposed to optimize objective functions. While BO is favored for its mathematical simplicity and effective learning and inference capabilities, which aid in creating efficient acquisition functions, it faces challenges in real-world applications. These challenges arise primarily from the high-dimensional, large-scale, and diverse nature of many optimization tasks. In Bayesian Optimization, Gaussian Processes (GPs) are commonly used as surrogate models for the unknown objective function. However, fitting GPs in high-dimensional settings can be problematic due to the "curse of dimensionality," where the complexity of the model increases significantly with the number of dimensions, complicating the optimization process (Zhang et al., 2023; Eriksson & Jankowiak, 2021).

In (Boulkroune et al., 2023), the authors present a method for tuning the EKF using chi-square tests on NIS samples. This novel approach combines several metrics, including RMSE, estimation error covariance, and consistency mea-

sures, into a weighted objective function to avoid local minima and ensure filter consistency. This method is detailed in the Appendix E.4.1, and serves as a baseline in our empirical evaluation due to its focus on innovation errors and avoidance of high-dimensional issues. In addition, the method proposed in (Abbeel et al., 2005) is also used as a baseline and detailed in Appendix E.4.3. The authors proposed minimizing the residual prediction error, which use the outputs of EKF update step and can enable the use of all measurement data available. However, this approach requires initializing an additional measurement noise covariance ($\mathbf{M}$). In cases where $\mathbf{M}$ is unknown or difficult to estimate, alternative methods or additional steps might be necessary to estimate or approximate $\mathbf{M}$ before applying this tuning approach. The novelty in BAT lies in a comprehensive posterior distribution that does not require anything beyond all measurement data available and the outputs from the NKE prediction step to estimate the distribution of all NKE components.

It is essential to point out that most auto-tuning approaches available were developed for EKF. However, there are some approaches tailored for UKF. See the extension of related work in Appendix D.

## 3. Background

### 3.1. Markov and Independence Assumptions in Probabilistic State Space Models

A general probabilistic state space models (SSM) with unknown parameters $\mathbf{\Theta}$ is defined as:

$$\begin{aligned} \mathbf{\Theta} &\sim p(\mathbf{\Theta}), \\ \mathbf{x}_0 &\sim p(\mathbf{x}_0|\mathbf{\Theta}), \\ \mathbf{x}_k &\sim p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{\Theta}), \\ \mathbf{y}_k &\sim p(\mathbf{y}_k|\mathbf{x}_k, \mathbf{\Theta}) \end{aligned} \tag{1}$$

for k= 0, 1, 2, $\cdots$ T, where $\mathbf{x}_k \in \mathbb{R}^n$ is the system state at k, and $\mathbf{y}_k \in \mathbb{R}^{m \leq n}$ is the measurement at k, where $\mathbf{y}_k = \{\mathbf{y}_{online,k} \cup \mathbf{y}_{offline,k}\}$ and $\mathbf{x}_k = \{\mathbf{x}_{IE,k} \cup \mathbf{x}_{NIE,k}\}$. More details about online and offline measurements ($\mathbf{y}_{online,k}$ and $\mathbf{y}_{offline,k}$) can be seen in Appendix A.3. $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ is the dynamic model that describes the stochastic dynamics of the system and is Markovian. Since, the future $\mathbf{x}_k$ is independent of the past given the present (here present is $\mathbf{x}_{k-1}$), $p(\mathbf{x}_k|\mathbf{x}_{1:k-1}, \mathbf{y}_{1:k-1}) = p(\mathbf{x}_k|\mathbf{x}_{k-1})$, and the past $\mathbf{x}_{k-1}$ is independent of the future given the present (here present is $\mathbf{x}_k$ ), $p(\mathbf{x}_{k-1}|\mathbf{x}_{k:T}, \mathbf{y}_{k:T}) = p(\mathbf{x}_{k-1}|\mathbf{x}_k)$. $p(\mathbf{y}_k|\mathbf{x}_k)$ is the measurement model, where the measurements $\mathbf{y}_k$ are conditionally independent given $\mathbf{x}_k$, then $p(\mathbf{y}_k|\mathbf{x}_{1:k}, \mathbf{y}_{1:k-1}) = p(\mathbf{y}_k|\mathbf{x}_k)$ (Inoue et al., 2022). In addition, the parameter $\mathbf{\Theta}$ was considered implicit in dynamic and measurement expressions.

### 3.2. Batch and Recursive Bayesian Inference

The batch Bayesian inference (solution) to a parameter estimation problem considering the Equation 1 can be formulated in three steps. First, specify the likelihood model of measurements $p(\mathbf{y}_k|\mathbf{x}_k, \mathbf{\Theta})$ given the parameter $\mathbf{\Theta}$. To focus on estimating the parameters $\mathbf{\Theta}$, it is useful to integrate out the state variable $\mathbf{x}_k$. Then, assuming the conditional independence of measurements $\mathbf{y}_k$, we have $p(\mathbf{y}_{1:T}|\mathbf{\Theta}) = \prod_{k=1}^{T} p(\mathbf{y}_k|\mathbf{\Theta})$. Second, define the prior distribution $p(\mathbf{\Theta})$. Lastly, compute the posterior distribution by the Bayes' rule:

$$p(\mathbf{\Theta}|\mathbf{y}_{1:T})_B = \frac{1}{Z} p(\mathbf{\Theta}) \prod_{k=1}^{T} p(\mathbf{y}_k|\mathbf{\Theta}), \tag{2}$$

where $Z$ is the normalization constant defined as $Z = \int p(\mathbf{\Theta}) \prod_{k=1}^{T} p(\mathbf{y}_k|\mathbf{\Theta}) d\mathbf{\Theta}$.

Now, the recursive Bayesian inference (solution) to the above problem can be also formulated in three steps (Pérez-Vieites & Míguez, 2021). Initially, assuming the conditional independence of measurements $\mathbf{y}_k$, specify the likelihood model of measurements as $p(\mathbf{y}_k|\mathbf{\Theta})$. Second, define prior distribution $p(\mathbf{\Theta})$ at the beginning of estimation (i.e., at step 0). Lastly, process the measurements $\mathbf{y}_1, \cdots, \mathbf{y}_T$, one at a time, and at each step we use the posterior distribution from the previous time step as the current prior distribution:

$$\begin{aligned} p(\mathbf{\Theta}|\mathbf{y}_1) &= \frac{1}{Z_1} p(\mathbf{y}_1|\mathbf{\Theta}) p(\mathbf{\Theta}), \\ p(\mathbf{\Theta}|\mathbf{y}_{1:2}) &= \frac{1}{Z_2} p(\mathbf{y}_2|\mathbf{\Theta}) p(\mathbf{\Theta}|\mathbf{y}_1), \\ p(\mathbf{\Theta}|\mathbf{y}_{1:3}) &= \frac{1}{Z_3} p(\mathbf{y}_3|\mathbf{\Theta}) p(\mathbf{\Theta}|\mathbf{y}_{1:2}), \\ &\vdots \\ p(\mathbf{\Theta}|\mathbf{y}_{1:T})_R &= \frac{1}{Z_T} p(\mathbf{y}_T|\mathbf{\Theta}) p(\mathbf{\Theta}|\mathbf{y}_{1:T-1}). \end{aligned} \tag{3}$$

Therefore, we have that the posterior distribution (Equation 3) obtained through recursive Bayesian inference is equivalent to the posterior distribution (Equation 2) obtained through batch Bayesian inference, $p(\mathbf{\Theta}|\mathbf{y}_{1:T})_B = p(\mathbf{\Theta}|\mathbf{y}_{1:T})_R$ (Pérez-Vieites & Míguez, 2021; Särkkä & Svensson, 2023). It is important to point out that the recursive formulation of Bayesian inference has two useful properties (Särkkä & Svensson, 2023): i) Because each step in the recursive estimation is a full Bayesian update step, batch Bayesian inference is a special case of recursive Bayesian inference; and ii) Considering the state $\mathbf{x}$ of a system beside parameters $\mathbf{\Theta}$, we can also model the effect of time on the state $\mathbf{x}$ of a system due to the sequential nature of estimation. This is actually the basis of Bayesian filtering theory (Appendix A.1) , where time behavior is modeled by assuming the state to be a time-dependent stochastic process $\mathbf{x}$(t).

### 3.3. Nonlinear Kalman Estimators

The Kalman filter is the closed-form solution to the Bayesian filtering equations (Appendix A.1) for a state space model, where the dynamic and measurement models are linear Gaussian without the need for numerical approximations due to its exact Gaussian posterior distribution. However, the Bayesian optimal filtering equations become computationally intractable when the dynamic and measurement models are nonlinear and non-Gaussian. To address this, nonlinear Kalman estimators have been developed. The two-step recursive algorithmic process used by Nonlinear Kalman Estimators (EKF, UKF, RKF, and CKF) is summarized as follows (Akca & Efe, 2019; Khodarahmi & Maihami, 2022): **NKE Prediction step)** This step is where the state and error are propagated forward in time. In this step, the predicted mean of state, $\hat{\mathbf{x}}_{k/k-1}$, and predicted error covariance matrix of state $\mathbf{P}_{k/k-1}$ are obtained using a process model (nonlinear dynamic model function $\mathbf{f}$ with parameters $\boldsymbol{\theta}$), initial conditions ($\hat{\mathbf{x}}_0$ and $\mathbf{P}_0$) and $\mathbf{Q}$. See the Figure 6 in Appendix A.1. Each Nonlinear Kalman Estimator performs the prediction step differently: EKF (Julier & Uhlmann, 1997) linearizes the system's dynamics around the current state estimate for prediction, RKF (Rocha & Terra, 2021) is similar to EKF but adapts to model uncertainties and outliers, UKF (Wan & Van Der Merwe, 2000) uses deterministically chosen sigma points to approximate nonlinear transformations, and CKF (Arasaratnam & Haykin, 2009) employs cubature rules to compute integrals of the state transition function over the state distribution without linearization. **NKE Update step)** This step is the same for all Nonlinear Kalman Estimators (EKF, UKF, RKF, and CKF). In this step the Predictions ( $\hat{\mathbf{x}}_{k/k-1}$ and $\mathbf{P}_{k/k-1}$ ) are combined with the measured values ($\mathbf{y}_k$) to provide updated states and errors ($\hat{\mathbf{x}}_{k/k}$ and $\mathbf{P}_{k/k}$) using $\mathbf{R}$. This involves calculating the Kalman gain, determining how much the state prediction should be corrected based on the new measurement, and updating the error covariance to reflect the reduced uncertainty after incorporating the measurement. In addition, the updated state becomes the initial condition for the next prediction performed by the prediction step, $\hat{\mathbf{x}}_0 = \hat{\mathbf{x}}_{k/k}$ and $\mathbf{P}_0 = \mathbf{P}_{k/k}$. See the Figure 6 in Appendix A.1.

## 4. BAT: Batch Bayesian Auto-Tuning for Nonlinear Kalman Estimators

In this Section, we describe likelihood, priors and posterior density of BAT to auto-tuning any NKE.

**BAT likelihood.** Let us recall that the NKE is a recursive algorithm that corrects the prediction of **NKE prediction step** about the states variables based on **NKE update step** with the available measurement, see Figure 6 in Appendix. Generally, the algorithm starts with the **NKE prediction step** because we need initial predictions of the state to compare against the measurement in the update step. However, in certain scenarios, we might start with the **NKE update step** if we have an initial measurement but no good initial prediction of the state (Kalman, 1960). Here, we start the recursive loop with the **NKE update step** at time $t_{k-1}$ to improve the performance of the **NKE prediction step** by estimating the best initial conditions used as input, and we have the following assumptions.

**Assumption 4.1.** **NKE prediction step** is still obtaining the solution for the state variables at time $t_k$ being executed after **NKE update step**.

**Assumption 4.2.** The measurements $\mathbf{y}_k$, for $k = 1, \ldots, T$, used in a NKE, are described by an measurement model in the batch Bayesian inference (BBI) framework outside of the NKE (see Figure 1) as,

$$\mathbf{y}_k = h(\hat{\mathbf{x}}_k^{NKE_{PS}}) + \boldsymbol{\delta}_k, \tag{4}$$

given

$$\begin{aligned}&\langle \hat{\mathbf{x}}_k^{NKE_{PS}}, \mathbf{P}_k^{NKE_{PS}} \rangle = NKE_{PS}(\mathbf{f}, \boldsymbol{\theta}, t_{k-1}, t_k, \\ &\hat{\mathbf{x}}_{0,k-1} = \hat{\mathbf{x}}_{k-1}^{NKE_{US}}, \mathbf{P}_{0,k-1} = \mathbf{P}_{k-1}^{NKE_{US}}, \mathbf{Q}_{k-1}, \mathbf{R}_{k-1}).\end{aligned} \tag{5}$$

The $\hat{\mathbf{x}}_k^{NKE_{PS}}$ (predicted mean of state variables) is the solution obtained by the **NKE prediction step** at time $\mathrm{t}_k$, beside of predicted error covariance matrix $\mathbf{P}_k^{NKE_{PS}}$, see Figure 1. The additive noise $\boldsymbol{\delta}_k \sim N(0, \boldsymbol{\sigma}^2)$ is assumed as zero mean Gaussian with a given constant variance represented by $\boldsymbol{\sigma}^2$, and $h(\cdot)$ is a linear/non linear function that relates the predicted mean of state variables to the measurements. The $\hat{\mathbf{x}}_k^{NKE_{PS}}$ is obtained given $\mathbf{f}$ , $\boldsymbol{\theta}$, $\mathbf{Q}_{k-1}$ and the results of the **NKE update step** at time $\mathrm{t}_{k-1}$. These results are updated (estimated) mean $\hat{\mathbf{x}}_{k-1}^{NKE_{US}}$ and estimated error covariance matrix $\mathbf{P}_{k-1}^{NKE_{US}}$ of dependent states variables. They were generated by the **NKE update step** given $\mathbf{R}_{k-1}$, $\mathbf{y}_{k-1}$ and the previous predicted mean $\hat{\mathbf{x}}_{k-1}^{NKE_{PS}}$ and predicted error covariance matrix $\mathbf{P}_{k-1}^{NKE_{PS}}$, see Figure 1. It is important to point out that the $\hat{\mathbf{x}}_{k-1}^{NKE_{US}}$ and $\mathbf{P}_{k-1}^{NKE_{US}}$ are used as initial conditions ($\hat{\mathbf{x}}_{0,k-1} = \hat{\mathbf{x}}_{k-1}^{NKE_{US}}$ and $\mathbf{P}_{0,k-1}$=$\mathbf{P}_{k-1}^{NKE_{US}}$) in the **NKE prediction step** to obtain the $\hat{\mathbf{x}}_k^{NKE_{PS}}$ and $\mathbf{P}_k^{NKE_{PS}}$ from $\mathrm{t}_{k-1}$ to $\mathrm{t}_k$ .

Then, based on the previous assumptions, and assuming the conditional independence of measurement $\mathbf{y}_k$ in Equation 4, we can define the BAT likelihood of measurements such that

$$p(\mathbf{y}|h(\hat{\mathbf{x}}^{NKE_{PS}}), \boldsymbol{\sigma}^2) = \prod_{k=1}^{T} p(\mathbf{y}_k|h(\hat{\mathbf{x}}_k^{NKE_{PS}}), \boldsymbol{\sigma}^2) = \prod_{k=1}^{T} p(\mathbf{y}_k|h(NKE_{PS}(\mathbf{f}, \boldsymbol{\theta}, t_{k-1}, t_k, \hat{\mathbf{x}}_{0,k-1}, \mathbf{P}_{0,k-1}, \mathbf{Q}_{k-1}, \mathbf{R}_{k-1})), \boldsymbol{\sigma}^2). \quad (6)$$

**BAT Priors.** Given the Equation 6, we have the following sets for $k = 1, \ldots, T$: $\mathbf{R} = (\mathbf{R}_0, \mathbf{R}_1, ..., \mathbf{R}_{T-1})$, $\mathbf{Q} = (\mathbf{Q}_0, \mathbf{Q}_1, ..., \mathbf{Q}_{T-1})$, $\mathbf{P}_{0,set} = (\mathbf{P}_{0,0}, \mathbf{P}_{0,1}, ..., \mathbf{P}_{0,T-1})$ and $\hat{\mathbf{x}}_{0,set} = (\hat{\mathbf{x}}_{0,0}, \hat{\mathbf{x}}_{0,1}, ..., \hat{\mathbf{x}}_{0,T-1})$. However, since the NKE update step estimate the best initial condition for NKE prediction step at each k, in $\mathbf{P}_{0,set}$ and $\hat{\mathbf{x}}_{0,set}$, we can keep the focus to estimate first initial conditions $\mathbf{P}_{0,0}$ and $\hat{\mathbf{x}}_{0,0}$ of the sets $\mathbf{P}_{0,set}$ and $\hat{\mathbf{x}}_{0,set}$. Then, reorganizing the Equation 6 to have a likelihood of the measurement $\mathbf{y}_k$ given the all NKE components, we have

$$p(\mathbf{y}|h(\hat{\mathbf{x}}^{NKE_{PS}}), \boldsymbol{\sigma}^2) = \prod_{k=1}^{T} p(\mathbf{y}_k|\boldsymbol{\theta}, \hat{\mathbf{x}}_{0,0}, \mathbf{P}_{0,0}, \mathbf{Q}_{k-1}, \mathbf{R}_{k-1}, \boldsymbol{\sigma}^2), \quad (7)$$

and consequently the following BAT priors $p(\boldsymbol{\sigma}^2), p(\mathbf{Q}_{k-1}), p(\mathbf{R}_{k-1}), p(\boldsymbol{\theta}), p(\hat{\mathbf{x}}_{0,0})$, and $p(\mathbf{P}_{0,0})$.

**BAT posterior density.** Since, we have BAT likelihood and priors, we can apply the Bayes' rule. Then, the general BAT posterior distribution of all NKE components can be written as

$$p(\boldsymbol{\theta}, \hat{\mathbf{x}}_{0,0}, \mathbf{P}_{0,0}, \mathbf{Q}, \mathbf{R}, \boldsymbol{\sigma}^2|\mathbf{y}) \propto p(\hat{\mathbf{x}}_{0,0}) \times p(\mathbf{P}_{0,0}) \times p(\boldsymbol{\sigma}^2) \times \prod_{k=1}^{T} p(\mathbf{y}_k|\boldsymbol{\theta}, \hat{\mathbf{x}}_{0,0}, \mathbf{P}_{0,0}, \mathbf{Q}_{k-1}, \mathbf{R}_{k-1}, \boldsymbol{\sigma}^2) \times p(\mathbf{Q}_{k-1}) \times p(\mathbf{R}_{k-1}). \quad (8)$$

It is important to point out that the normalization constant, which is independent of all NKE components is left out, and the priors $p(\mathbf{Q}_{k-1})$ and $p(\mathbf{R}_{k-1})$ can be constant for $k = 1, \ldots, T$ in some cases.

**Assumption 4.3.** The density shown in Equation 8 can be sampled by MCMC methods.

Consequently, we have the following Theorem.

**Theorem 4.4.** *Assume $\mathbf{y}_k$ includes both online measurements $\mathbf{y}_{online,k}$ (for state variables computing innovation errors) and offline measurements $\mathbf{y}_{offline,k}$ (for other state variables). Then BAT posterior can incorporate $\mathbf{y}_{offline,k}$ as additional information to enhance the auto-tuning process of any NKE if NKE can predict all state variables.*

The proof of Theorem 4.4 can be seen in Appendix B. In addition, theoretical Application of BAT to Estimate $\boldsymbol{\Theta}$ of EKF, UKF and CKF can be seen in Appendix C.

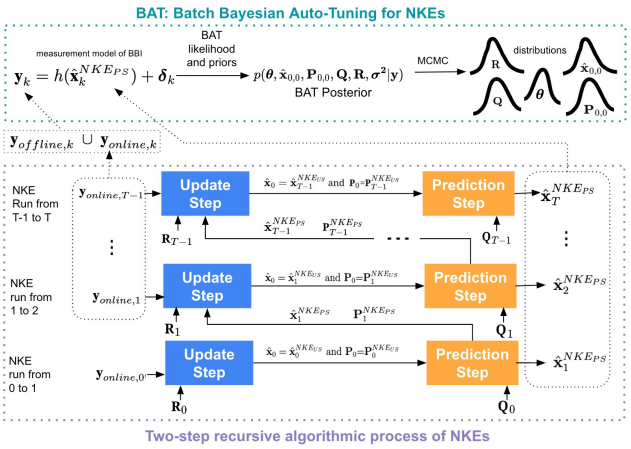


*Figure 1.* BAT: Batch Bayesian Auto-Tuning for Nonlinear Kalman Estimators. The online measurements $\mathbf{y}_{online,k}$ (used to generate innovation errors) and offline measurements $\mathbf{y}_{offline,k}$ of other state variables compose a full set $\mathbf{y}_k$ of measured data of all state variables of the system. Since $\mathbf{y}_k$ can be described by a measurement model outside of the NKE where the hidden state variables and additive noise process are represented by $\hat{\mathbf{x}}_k^{NKE_{PS}}$ (predicted mean of state variables) and $\boldsymbol{\delta}_k$, respectively. Using this measurement model outside of NKE, it is possible to define the BAT likelihood (Equation 7) and priors to obtain the BAT posterior (Equation 8). Then, samples can be extracted by MCMC methods to auto-tune NKEs by extracting the mean values from the obtained distribution of $\boldsymbol{\theta}, \hat{\mathbf{x}}_{0,0}, \mathbf{P}_{0,0}, \mathbf{Q}, \mathbf{R}$.

## 5. Experiments

We executed detailed experiments to answer the following three questions and understand the effectiveness of the proposed approach: **Q1**) How does the performance of BAT compare with the performance of baseline methods? **Q2**) Does the BAT posterior density (Equation 8) enable an efficient sampling with MCMC? **Q3**) What is the impact of prior distributions on the performance of BAT? Theses questions are answered by employing BAT and baseline methods to estimate the well-defined ground truth values of $\mathbf{x}_{0,0}, \boldsymbol{\theta}$, $\mathbf{Q}$, and $\mathbf{R}$ that enable the design of a consistent EKF in two different tasks. The EKF was chosen for this empirical evaluation due to its status as the most popular and widely used as NKE. However, in the Appendix C.2 and C.3, we show examples of the BAT application with UKF and CKF in a classical problem.

### 5.1. Experimental Setup

**Datasets.** The Synthetic dataset (SD) has data regarding Monoclonal Antibody (mAb) productions that represent the biomanufacturing of a protein widely used as diagnostic reagents and for therapeutic purposes (Jyothilekshmi & Jayaprakash, 2021). The SD is composed of runs (A and B) with different cell expansions and different maximum of mAb (titer) production. The runs have data related to 7 state variables (viable cell density (Xv), total cell (Xt), glu-

cose (GLC), glutamine (GLN), lactate (LAC), ammonium (AMM), and mAb), and each run is used in a different task of our empirical evaluation. Generally, the state variables GLC, GLN, LAC, AMM, and mAb are obtained through offline measurements by a researcher/engineer at a low sampling rate in the order of hours or days, and the state variable Xv is obtained by online measurements from sensors in the bioreactor at the high sampling rate in the order of minutes. The run A is composed of one training set A1 and two testing sets (A2 and A3) and all sets have only Xv noisy data. In addition, the training set A1 represents a case where we have only the measured data to generate innovation errors. The training set A1 and testing sets A2 and A3 has two regions of Xv noisy data generated with different white Gaussian noises, see Figure 2. The region 1 (blue) from 0h to 50h has the Gaussian white noise with a standard deviation of $10\times10^7$, and in the region 2 (orange) from 50h to 103h has the Gaussian white noise with a standard deviation of $20\times10^7$. The run B is also composed of one training B1 set and two testing sets (B2 and B3), however the training set B1 has noisy data related to all state variables representing the case where we have the measured data to generate innovation errors, and extract noisy data related to the other state variables but not used to generate innovation errors. In addition, the testing sets have only Xv noisy data, see Figure 2. In the training set B1, the Gaussian white noises with standard deviations (std) of $10\times10^7$, $10\times10^7$, 1, 0.5, 2.0, 0.1 and 40.5 were added to the main solution B of state variables Xv, Xt, GLC, GLN, LAC, AMM and mAb, respectively, from 0h to 103h. The testing set B2 and B3 have the Xv noisy data generated from white Gaussian noises with standard deviations (std) of $10\times10^7$ , see Figure 2. It is important to point out that the training and testing sets have different noisy data sample rates aiming to simulate real-world situations where we have to tune a NKE with a small (poor) dataset and apply the NKE in online mode with a high frequency of noisy data. The training set A1 has a sample rate of 1h for Xv noisy data, and the testing sets A2 and A3 have a sample rate of 1h and 7.5 min for Xv noisy data, respectively, see Figure 2. On the other hand, the training set B1 has a sample rate of 7h for all state variables and the testing sets B2 and B3 have sample rate of 7h and 7.5 min respectively, see Figure 2. The training set B1 and testing set B2 represent a very poor dataset with 15 data points for each state variable. This kind of small dataset commonly result from offline measurements for all state variables. More details about the development process of synthetic dataset can be seen in the Appendix E.3.

**Baselines.** The baseline methods used in the empirical tests aiming to optimize an objective function (OF) that combined several terms (Boulkroune et al., 2023) to help the optimizer to converge to the best sub-optimal $\boldsymbol{\Theta}$ and to avoid getting stuck in local minima. A detailed description of this

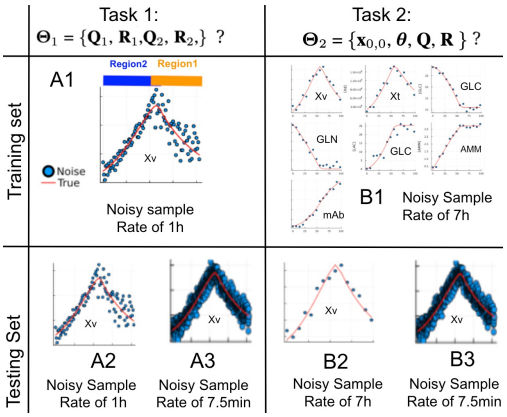


*Figure 2.* Datasets and tasks. Task 1 requires the estimation of $\boldsymbol{\Theta}_1 = \{\mathbf{Q}_1, \mathbf{R}_1, \mathbf{Q}_2, \mathbf{R}_2, \}$ using the training set A1 that has two regions (blue and orange) with different noises (different values of $\mathbf{R}$ in each region). The noisy Xv sample rate used in A1 is 1h, and the testing set used to assess the designed EKF with the estimated $\boldsymbol{\Theta}_1$ are A2 and A3. They have sample rates 1h and 7.5 min, respectively. Task 2 requires the estimation of $\boldsymbol{\Theta}_2 = \{\mathbf{x}_{0,0}, \boldsymbol{\theta}, \mathbf{Q}, \mathbf{R}\}$ using the training set B1 that has a sample rate of 7h related to noisy data of 7 state variables (blue) where each one has a specific noise. The testing sets used to assess the designed EKF with the estimated $\boldsymbol{\Theta}_2$ are B2 and B3. They have sample rates of 7h and 7.5 min, respectively.

objective function can be seen in Appendix E.4.1 . The baseline methods are divided in two groups. The first group comprises in the numerical optimization solvers to optimize the objective function. The methods used in empirical test are Nonlinear Optimization with the MADS algorithm (OF-NOMAD) (Montoison et al., 2020), Particle Swarm (OF-PS) (Wang et al., 2018), Gradient-based Covariance Matrix Adaptation Evolutionary Strategy (OF-GCMAES) (Dixit & Rackauckas, 2023), Covariance Matrix Adaptation Evolution Strategy (OF-CMAES) (Hansen, 2016), Evolutionary Computation Algorithm (OF-ECA) (Tupe & Kulkarni, 2015) and Simulated Annealing (OF-SA) (Delahaye et al., 2019), see the Appendix for more details. The last group is formed by the No-U-Turn Sampler (NUTS) approach (Hoffman et al., 2014). NUTS is the MCMC method used to sample the distribution of $\boldsymbol{\Theta}$ that minimize the objective function (Appendix E.4.1) and the residual prediction error (RPE) (Appendix E.4.3) through a probabilistic model. We refer to these approaches as OF-NUTS and RPE-NUTS, see the Appendix for more details about this approach.

It is important to point out that all baselines using the OF can only use the noisy data related to the state variable (Xv) that generate innovation errors to auto-tune the EKF during the execution of the tasks. As described before, this is the limitation of objective functions based on NIS (Boulkroune et al., 2023).

**Tasks.** The training and testing sets A and B are used in

tasks 1 and 2 respectively. It is important to point out that in all tasks only Xv (noisy data) is used as measured state variable to generate the innovation errors during the execution of the EKF during the process of auto-tuning. **Task 1:** It represents the challenge of auto-tuning the EKF by using only noisy data related to the state variable that generate innovation errors. It consists in two steps: i) using BAT and baselines with the Xv noisy data of training set A1 to tune a EKF with the optimal $\boldsymbol{\Theta}_1 = \{\mathbf{Q}_1, \mathbf{R}_1, \mathbf{Q}_2, \mathbf{R}_2, \}$ related to regions 1 and 2 of training set A1, and ii) testing the filter consistency of the designed EKF with the Xv noisy data of testing set A2 and A3. The task 1 is executed by BAT and all baselines, and this is a common task for all baselines. **Task 2:** It represents the challenge of auto-tuning the EKF by using noisy data related to the state variable that generate innovation errors, and the additional noisy data related to the others state variables that do not generate innovation errors. It comprises in three steps: i) using the BAT with all noisy data of training set B1 and the baseline (OF-NUTS) with only Xv noisy data of training set B1 to estimate the optimal $\boldsymbol{\Theta}_2 = \{\mathbf{x}_{0,0}, \boldsymbol{\theta}, \mathbf{Q}, \mathbf{R}\}$ for EKF, where $\mathbf{x}_{0,0}$=$\{u0_{LAC}, u0_{mAb}\}$ represents the initial concentrations of LAC and mAb, and $\boldsymbol{\theta}$ =$\{Y_{lac,glc}, Y_{amm,gln}, \lambda\}$ represents three unshared parameters of the unstructured mechanistic model used in the EKFs (details in Appendix E.1); ii) testing the filter consistency of the designed EKF with noisy Xv data of testing sets B2 and B3, and iii) assessing the accuracy between designed EKFs estimations and ground truth values of the testing set B2 and B3 through of the root-mean-square percentage error (RMSPE). It is important to point out that task 2 represents a challenging scenario where the stage augmentation approach fails, see Appendix D.

**Metrics.** First, to specifically answer Q1, we utilize the filter consistency test (outlined in Appendix E.6.1) as the metric for comparing the final estimations of $\boldsymbol{\Theta}_1$ and $\boldsymbol{\Theta}_2$ achieved by both BAT and baselines. In addition, we used RMSPE to asses the estimations done by the designed EKFs in task 2. Second, to specifically answer Q2, we utilize the auto-correlation function (ACF) plot as the metric for assessing the mixing quality chains obtained through MCMC simulations (Brooks et al., 2011), see the Appendix E.6.2 for details. Third, for addressing Q3, besides executing BAT and baselines with the same priors, we also execute BAT with wide Uniform priors (BAT-WP), OF-NUTS with wide Uniform priors (OF-NUTS-WP) and RPE-NUTS with wide Uniform priors (RP-NUTS-WP). This dual approach aims to demonstrate BAT's dependency on prior knowledge, see the Appendix E.5 for details about the selected priors. Lastly, the ground truth values of $\boldsymbol{\Theta}_1$ and $\boldsymbol{\Theta}_2$ were defined through of consistency test of EKF (as outlined in Appendix E.6.3). It is important to point out that BAT and the baseline used the same type of priors to guarantee a fair comparison. The setup information about the BAT and the baselines can be seen in Appendix E.5.

### 5.2. Results

**Answering Q1.** Table 1 presents the results of the filter consistency test of the designed EKF using the $\boldsymbol{\Theta}_1$ estimated by BAT and baselines during task 1. Only BAT was able to estimate the optimal $\boldsymbol{\Theta}_1$ that enabled the EKF to be consistent in all regions of the testing set A2 and A3, because no baseline was able to design an EKF acceptable in all consistency tests of task1. However, OF-NUTS and RPE-NUTS had the best performance among all baselines because they were the unique baselines to design an EKF that was consistent in region 2 of the testing set A3. All BAT and baselines estimations related to $\boldsymbol{\Theta}_1$ can be seen in Tables 15 and 16 of Appendix E.7. Table 2 presents the results of filter consistency test of the designed EKFs using the $\boldsymbol{\Theta}_2$ estimated by BAT , OF-NUTS and RPE-NUTS during task 2. The EKF designed by OF-NUTS and RPE-NUTS were not acceptable in consistency test with testing set B2 and B3. Furthermore, OF-NUTS and RPE-NUTS were not able to estimate the $\mathbf{x}_{0,0}$=$\{u0_{LAC}, u0_{mAb}\}$ , and $\boldsymbol{\theta}$ =$\{Y_{lac,glc}, Y_{amm,gln}, \lambda\}$ and the designed EKFs were not able to proper estimate LAC, AMM and mAb in both cases of testing set B2 and B3, see Figures 4 and 11 . In addition, this poor performance was confirmed by high RMSPE values, see Table 3. On the other hand, BAT has the best performance in the consistency tests (see Table 2) and presents the best estimations of LAC, AMM and mAB, see Figures 11 and 4, and Table 3. BAT and OF-NUTS estimation related to $\boldsymbol{\Theta}_2$ can be seen in Table 17 of Appendix E.7. It is important to point out that since we are considering a nonlinear model the posterior distribution estimated can become multimodal. However, BAT estimates the posterior distributions of parameters as unimodal, closely aligning with the ground truth values, see Figure 3 and Figure 9 in Appendix. This outcome contrasts with the baseline methods, which estimates the posterior distributions of parameters ($\mathbf{R}$, $\mathbf{x}_{0,0}$=$\{u0_{LAC}, u0_{mAb}\}$ , and $\boldsymbol{\theta}$ =$\{Y_{lac,glc}, Y_{amm,gln}, \lambda\}$ ) as multimodal distributions. The baseline methods result in multimodal distributions for the parameters indicating a spread across multiple plausible solutions. While this might suggest a broader exploration of the parameter space, it does not imply a more accurate estimation. The presence of multiple peaks could reflect uncertainties or ambiguities in parameter values, leading to less definitive conclusions about the true parameter values.

**Answering Q2.** Figure 5 shows the ACF plots for the BAT, OF-NUTS and RPE-NUTS applied during the task 2. In both plots, the ACF values quickly dropped to zero, indicating the good quality of the mixing obtained with BAT, OF-NUTS and RPE-NUTS. It suggests that the samples in the chains become uncorrelated quickly. It important to point out that similar results were obtained during the execution of task 1, see Figure 10 in Appendix E.7. These

*Table 1.* Results of filter consistency test of the designed EKF with the Xv noisy data of testing set A2 (sample rate 1h) and A3 (sample rate 7.5 min) during task 1. × means unacceptable, √ means acceptable, and Reg means region.

| | Testing Set | | | |
|---|---|---|---|---|
| | A2 | | A3 | |
| Methods | Reg1 | Reg2 | Reg1 | Reg2 |
| OF-NOMAD | × | √ | × | × |
| OF-PS | × | √ | × | × |
| OF-GCMAES | × | × | × | × |
| OF-CMAES | × | × | × | × |
| OF-ECA | × | √ | × | × |
| OF-SA | × | √ | × | × |
| OF-NUTS | √ | √ | × | √ |
| OF-NUTS-WP | × | × | × | × |
| RPE-NUTS | √ | √ | × | √ |
| RPE-NUTS-WP | × | × | × | × |
| BAT | √ | √ | √ | √ |
| BAT-WP | × | × | × | × |

*Table 2.* Results of filter consistency test of the designed EKF with testing set B2 and B3 during the execution of task 2. × means unacceptable, √ means acceptable and SR means sample rate.

| | Testing Set | |
|---|---|---|
| Methods | B2 (SR=7h) | B3 (SR=7.5min) |
| OF-NUTS | × | × |
| RPE-NUTS | × | × |
| BAT | √ | √ |

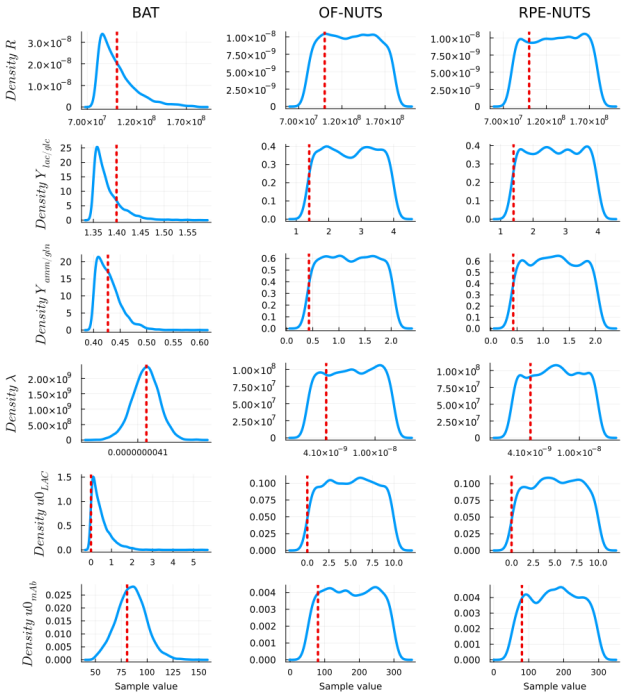


*Figure 3.* Distribution of the sampled values (for **R**, $u0_{LAC}$, $u0_{mAb}$, $Y_{lac,glc}$, $Y_{amm,gln}$, and $\lambda$) by BAT, OF-NUTS and RPE-NUTS with training set B1. OF-NUTS and RPE-NUTS estimated multimodal distributions, but BAT estimated unimodal distributions with peaks close to the ground truth values (red vertical dot line).

results have two implications for the sampling process and the nature of the posterior distribution obtained with BAT. i) Efficient sampling: A rapidly declining ACF indicates efficient sampling from the posterior distribution. In practical terms, this means that each new sample in the chain provides new information about the posterior. The chain does not 'waste time' revisiting similar states repeatedly, which is common when samples are highly correlated. ii) Good exploration of the posterior: Efficient sampling, as indicated by a rapidly declining ACF, suggests that the MCMC chain is exploring the posterior distribution effectively. This is particularly important in high-dimensional and complex posterior landscapes (13 dimensions in the case of task 2), where ensuring adequate exploration is challenging. It is important to point out that we made additional experiments to check how the performance of the MCMC method is affected by the time-length (number of data points) and the size of the state, see Apendix E.7.1. We concluded that increasing the state variable vector can cause more impact on the computation duration than increasing the number of data points

**Answering Q3.** The result obtained during the execution of the task 1 (Table 1) shows that prior distribution has a significant impact on the mean of the distribution obtained for $\boldsymbol{\Theta}_1$. Because the use of BAT with wide interval of values for prior (BAT-WP) resulted in an EKF that failed in all consistency tests performed with the testing sets A2 and A3 due to a distribution with a mean far from the ground truth of $\boldsymbol{\Theta}_1$, see Tables 15 and 16 in Appendix E.7. Furthermore, OF-NUTS-WP and RPE-NUTS-WP also had a worst performance than BAT. They failed in filter consistency test, see the Tables 1 and 18. It is important to point out that the size of training sets A1 and B1 did not affect the performance of BAT because it was able to estimate the optimal $\boldsymbol{\Theta}_1$ and $\boldsymbol{\Theta}_2$ to design stable EKFs in both tasks.

## 6. Discussion

The ACF plots (Figures 5 and 10 ) showed that BAT posterior can be efficiently sampled with NUTS since the obtained results shown: a fast convergence (the chain has reached a stationary distribution that adequately represents the posterior), reduced need for thinning the chains and potential for shorter chains. It is essential to point out that the results addressing the Q3 showed that the main limitation of BAT is the need of good selection of priors. However, that can be easily achieved by applying BAT with the baselines when we do not have any information about possible priors. So, the baseline results can be used as priors in BAT

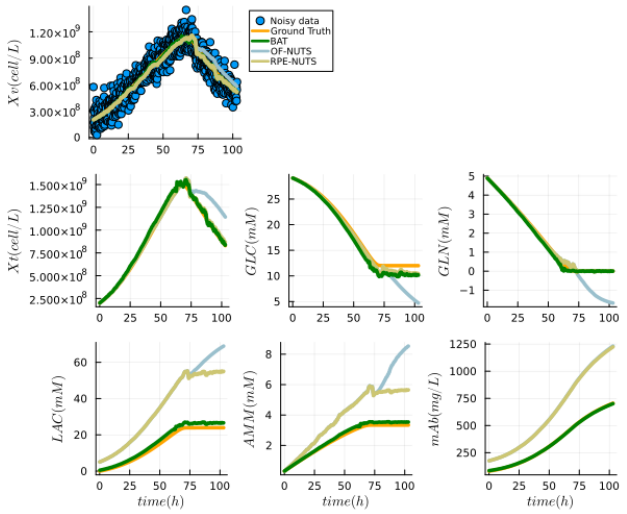


*Figure 4.* State variables estimations from EKF designed by BAT, OF-NUTS and RPE-NUTS during task 2 using the noisy Xv data of testing set B3.

*Table 3.* RMSPE between designed EKF (during task2) and ground truth values of the testing set B3 (sample rate of 7.5 min).

| STATE | BAT | OF-NUTS | RPE-NUTS |
|---|---|---|---|
| XV | 2.41 % | 6.39% | 2.89% |
| XT | 3.29 % | 13.32% | 2.46% |
| GLC | 9.86 % | 19.83% | 6.19% |
| GLN | 31.48 % | 47.82% | 45.22% |
| LAC | 39.20% | 144.18% | 124.61% |
| AMM | 4.77% | 71.69% | 53.58% |
| $mAb$ | 1.7% | 84.49% | 86.03% |

posterior density. Furthermore, other BAT limitation is that BAT needs observational data related to all state variable beyond ones used to generate the innovation errors. Without that the BAT may have similar performance to the baseline approaches. It is important to point out that BAT is an offline approach, but if it is possible to get a closed-form of BAT posterior Equation 8 (for example, using conjugate priors), BAT can be used as an online application. However, the online application of closed form of BAT posterior may require use the results obtained with the offline application of BAT posterior as priors. This approach helps to reduce the lack of information in online mode when only specific observed data is used for generating innovation errors in Nonlinear Kalman Estimators.

## 7. Conclusion

This study introduces Batch Bayesian Auto-Tuning (BAT), a novel approach for auto-tuning all NKE components, including the **Q**, **R**, **P**(0), $\mathbf{x}_{0,0}$, and ($\theta$). BAT enables utilizing all available measured data, not just those selected for generating innovation errors in NKEs. Our empirical results

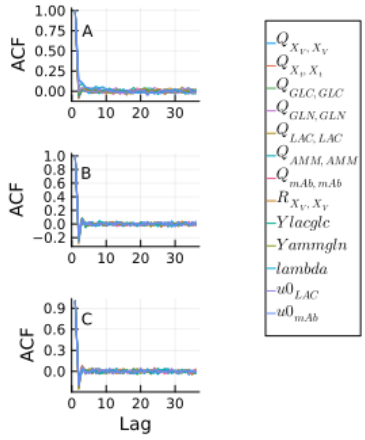


*Figure 5.* ACF plots of BAT (A), OF-NUTS (B) and RPE-NUTS (C) samples computed up to a lag of 30. This samples were obtained with training set B1 during the task 2.

demonstrate that BAT outperforms the baselines in tuning NKEs. BAT consistently achieved better performance in filter consistency tests and low RMSPE values besides exhibiting efficient sampling capabilities with MCMC. It is important to point out that BAT's effectiveness in estimating unimodal posterior distributions of all NKE components can be attributed to the well-defined BAT posterior distribution outside of NKE recursive loop. This enables the integration of all available measured data with the outputs from NKE prediction steps. This aspects collectively contribute to its ability to produce more precise and less uncertain estimates compared to baselines. While BAT currently is an offline approach, future research could explore the development of an online version, potentially by deriving a closed-form expression for the BAT posterior. Such an adaptation would enhance the applicability of BAT in dynamic environments where real-time auto-tuning of NKEs is required.

## References


Abbeel, P., Coates, A., Montemerlo, M., Ng, A. Y., Thrun, S., et al. Discriminative training of kalman filters. In *Robotics: Science and systems*, volume 2, pp. 1, 2005.

Akca, A. and Efe, M. Ö. Multiple model kalman and particle filters and applications: A survey. *IFAC-PapersOnLine*, 52(3):73–78, 2019.

Alahmadi, A. A., Flegg, J. A., Cochrane, D. G., Drovandi, C. C., and Keith, J. M. A comparison of approximate versus exact techniques for bayesian parameter inference in nonlinear ordinary differential equation models. *Royal Society open science*, 7(3):191315, 2020.

Arasaratnam, I. and Haykin, S. Cubature kalman filters. *IEEE Transactions on automatic control*, 54(6):1254–1269, 2009.

Aswal, N., Bhattacharya, B., and Sen, S. Joint and dual estimation of states and parameters with extended and unscented kalman filters. In *Recent Developments in Structural Health Monitoring and Assessment–Opportunities and Challenges: Bridges, Buildings and Other Infrastructures*, pp. 223–252. World Scientific, 2022.

Bar-Shalom, Y., Li, X. R., and Kirubarajan, T. *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons, 2001.

Bertipaglia, A., Shyrokau, B., Alirezaei, M., and Happee, R. A two-stage bayesian optimisation for automatic tuning of an unscented kalman filter for vehicle sideslip angle estimation. In *2022 IEEE Intelligent Vehicles Symposium (IV)*, pp. 670–677. IEEE, 2022.

Boulkroune, B., Geebelen, K., Wan, J., and van Nunen, E. Auto-tuning extended kalman filters to improve state estimation. In *2023 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1–6. IEEE, 2023.

Box, G. E. and Tiao, G. C. *Bayesian inference in statistical analysis*. John Wiley & Sons, 2011.

Brooks, S., Gelman, A., Jones, G., and Meng, X.-L. *Handbook of markov chain monte carlo*. CRC press, 2011.

Burkhart, M. C., Brandman, D. M., Vargas-Irwin, C. E., and Harrison, M. T. The discriminative kalman filter for nonlinear and non-gaussian sequential bayesian filtering. *arXiv preprint arXiv:1608.06622*, 2016.

Chen, Z., Heckman, C., Julier, S., and Ahmed, N. Weak in the nees?: Auto-tuning kalman filters with bayesian optimization. In *2018 21st International Conference on Information Fusion (FUSION)*, pp. 1072–1079. IEEE, 2018.

Chen, Z., Heckman, C., Julier, S., and Ahmed, N. Time dependence in kalman filter tuning. In *2021 IEEE 24th International Conference on Information Fusion (FUSION)*, pp. 1–8. IEEE, 2021.

Chen, Z., Biggie, H., Ahmed, N., Julier, S., and Heckman, C. Kalman filter auto-tuning through enforcing chi-squared normalized error distributions with bayesian optimization. *arXiv preprint arXiv:2306.07225*, 2023.

Chopda, V., Gyorgypal, A., Yang, O., Singh, R., Ramachandran, R., Zhang, H., Tsilomelekis, G., Chundawat, S. P., and Ierapetritou, M. G. Recent advances in integrated process analytical techniques, modeling, and control strategies to enable continuous biomanufacturing of monoclonal antibodies. *Journal of Chemical Technology & Biotechnology*, 97(9):2317–2335, 2022.

Delahaye, D., Chaimatanan, S., and Mongeau, M. Simulated annealing: From basics to applications. *Handbook of metaheuristics*, pp. 1–35, 2019.

Dixit, V. K. and Rackauckas, C. Optimization.jl: A unified optimization package, March 2023. URL https://doi.org/10.5281/zenodo.7738525.

Eriksson, D. and Jankowiak, M. High-dimensional bayesian optimization with sparse axis-aligned subspaces. In *Uncertainty in Artificial Intelligence*, pp. 493–503. PMLR, 2021.

Gargalo, C. L., Heras, S. C. d. l., Jones, M. N., Udugama, I., Mansouri, S. S., Krühne, U., and Gernaey, K. V. Towards the development of digital twins for the biomanufacturing industry. In *Digital Twins*, pp. 1–34. Springer, 2020.

Ge, H., Xu, K., and Ghahramani, Z. Turing: a language for flexible probabilistic inference. In *International conference on artificial intelligence and statistics*, pp. 1682–1690. PMLR, 2018.

Gu, J., Li, J., and Tei, K. A reinforcement learning approach for adaptive covariance tuning in the kalman filter. In *2022 IEEE 5th Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, volume 5, pp. 1569–1574. IEEE, 2022.

Hansen, N. The cma evolution strategy: A tutorial. *arXiv preprint arXiv:1604.00772*, 2016.

Haykin, S. S. and Haykin, S. S. *Kalman filtering and neural networks*, volume 284. Wiley Online Library, 2001.

Herwig, C., Pörtner, R., and Möller, J. *Digital Twins: tools and concepts for smart biomanufacturing*. Springer, 2021a.

Herwig, C., Pörtner, R., and Möller, J. *Digital Twins: Applications to the Design and Optimization of Bioprocesses*. Springer, 2021b.

Hoffman, M. D., Gelman, A., et al. The no-u-turn sampler: adaptively setting path lengths in hamiltonian monte carlo. *J. Mach. Learn. Res.*, 15(1):1593–1623, 2014.

Huang, H., Handel, A., and Song, X. A bayesian approach to estimate parameters of ordinary differential equation. *Computational statistics*, 35:1481–1499, 2020.

Hussain, M. K., Alshadeedi, B. M., and Hejeejo, R. Optimized kalman filters for sensorless vector control induction motor drives. *International Journal of Electrical and Computer Engineering*, 13(1):17, 2023.

Iglesias, C., Mehta, V., and Bolic, M. Handling massive proportion of missing labels in multivariate long-term time series forecasting. In *Journal of Physics: Conference Series*, volume 2090. IOP Publishing, 2021.

Iglesias, C. F. and Bolic, M. How not to make the joint extended kalman filter fail with unstructured mechanistic models. *Sensors*, 24(2), 2024. ISSN 1424-8220. doi: 10.3390/s24020653. URL https://www.mdpi.com/1424-8220/24/2/653.

Iglesias, Cristovão Freitas, J. X. X. V. M. M. A. A. V.-S. N. B. A. K. and Bolic, M. Monitoring the recombinant adeno-associated virus production using extended kalman filter. *Processes*, 10(11):2180, 2022.

Iglesias Jr, C. F., Xu, X., Mehta, V., Akassou, M., Venereo-Sanchez, A., Belacel, N., Kamen, A., and Bolic, M. Monitoring the recombinant adeno-associated virus production using extended kalman filter. *Processes*, 10(11):2180, 2022.

Iglesias Jr, C. F., Ristovski, M., Bolic, M., and Cuperlovic-Culf, M. raav manufacturing: The challenges of soft sensing during upstream processing. *Bioengineering*, 10 (2):229, 2023.

Inoue, H., Hukushima, K., and Omori, T. Estimating distributions of parameters in nonlinear state space models with replica exchange particle marginal metropolis–hastings method. *Entropy*, 24(1):115, 2022.

Julier, S. J. and Uhlmann, J. K. New extension of the kalman filter to nonlinear systems. In *Signal processing, sensor fusion, and target recognition VI*, volume 3068, pp. 182–193. Spie, 1997.

Jyothilekshmi, I. and Jayaprakash, N. Trends in monoclonal antibody production using various bioreactor systems. 2021.

Kalman, R. E. A new approach to linear filtering and prediction problems. 1960.

Khodarahmi, M. and Maihami, V. A review on kalman filter models. *Archives of Computational Methods in Engineering*, pp. 1–21, 2022.

Kim, S., Petrunin, I., and Shin, H.-S. A review of kalman filter with artificial intelligence techniques. In *2022 Integrated Communication, Navigation and Surveillance Conference (ICNS)*, pp. 1–12. IEEE, 2022.

Kyriakopoulos, S., Ang, K. S., Lakshmanan, M., Huang, Z., Yoon, S., Gunawan, R., and Lee, D.-Y. Kinetic modeling of mammalian cell culture bioprocessing: the quest to advance biomanufacturing. *Biotechnology Journal*, 13 (3):1700229, 2018.

Li, A. H., Wu, P., and Kennedy, M. Replay overshooting: Learning stochastic latent dynamics with the extended kalman filter. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 852–858. IEEE, 2021.

Lin, M. and Kim, B. Discriminative parameter training of the extended particle-aided unscented kalman filter for vehicle localization. *Applied Sciences*, 10(18):6260, 2020.

Liu, Y. and Gunawan, R. Bioprocess optimization under uncertainty using ensemble modeling. *Journal of biotechnology*, 244:34–44, 2017.

Luo, Y., Kurian, V., and Ogunnaike, B. A. Bioprocess systems analysis, modeling, estimation, and control. *Current Opinion in Chemical Engineering*, 33:100705, 2021.

Mears, L., Stocks, S. M., Albaek, M. O., Sin, G., and Gernaey, K. V. Mechanistic fermentation models for process design, monitoring, and control. *Trends in biotechnology*, 35(10):914–924, 2017.

Montoison, A., Pascal, P., and Salomon, L. NOMAD.jl: A Julia interface for the constrained blackbox solver NOMAD. https://github.com/bbopt/NOMAD.jl, July 2020.

Moser, A., Appl, C., Brüning, S., and Hass, V. C. Mechanistic mathematical models as a basis for digital twins. In *Digital Twins*, pp. 133–180. Springer, 2020.

Nitsch, M., Stenger, D., and Abel, D. Automated tuning of nonlinear kalman filters for optimal trajectory tracking performance of auvs. *arXiv preprint arXiv:2304.03565*, 2023.

Oshman, Y. and Shaviv, I. Optimal tuning of a kalman filter using genetic algorithms. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, pp. 4558, 2000.

Papathanasiou, M. M., Burnak, B., Katz, J., Shah, N., and Pistikopoulos, E. N. Assisting continuous biomanufacturing through advanced control in downstream purification. *Computers & Chemical Engineering*, 125:232–248, 2019.

Park, S.-Y., Park, C.-H., Choi, D.-H., Hong, J. K., and Lee, D.-Y. Bioprocess digital twins of mammalian cell culture for advanced biomanufacturing. *Current Opinion in Chemical Engineering*, 33:100702, 2021.

Pérez-Vieites, S. and Míguez, J. Nested gaussian filters for recursive bayesian inference and nonlinear tracking in state space models. *Signal Processing*, 189:108295, 2021.

Reid, I. and Term, H. Estimation ii discrete-time kalman filter. *Hilary Term*, pp. 1–44, 2001.

Reyes, S. J., Durocher, Y., Pham, P. L., and Henry, O. Modern sensor tools and techniques for monitoring, controlling, and improving cell culture processes. *Processes*, 10 (2):189, 2022.

Rocha, K. D. and Terra, M. H. Robust kalman filter for systems subject to parametric uncertainties. *Systems & Control Letters*, 157:105034, 2021.

Sakai, A. and Kuroda, Y. Discriminative parameter training of unscented kalman filter. *IFAC Proceedings Volumes*, 43(18):677–682, 2010.

Särkkä, S. and Svensson, L. *Bayesian filtering and smoothing*, volume 17. Cambridge university press, 2023.

Sekarsari, K. and Tata, T. Performance analysis of pid control in dc brushless motor using trial and error method. In *IOP Conference Series: Materials Science and Engineering*, volume 1098, pp. 042027. IOP Publishing, 2021.

Tang, Y., Hu, L., Zhang, Q., and Pan, W. Reinforcement learning compensated extended kalman filter for attitude estimation. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6854–6859. IEEE, 2021.

Tsopanoglou, A. and del Val, I. J. Moving towards an era of hybrid modelling: advantages and challenges of coupling mechanistic and data-driven models for upstream pharmaceutical bioprocesses. *Current Opinion in Chemical Engineering*, 32:100691, 2021.

Tummala, A. S. and Ramanarao, P. Tuning of extended kalman filter for power systems using two lbest particle swarm optimization, 2017.

Tupe, S. and Kulkarni, N. eca: Evolutionary computing algorithm for e-healthcare information system. *iPGCON-2015*, pp. 24–25, 2015.

Udugama, I. A., Öner, M., Lopez, P. C., Beenfeldt, C., Bayer, C., Huusom, J. K., Gernaey, K. V., and Sin, G. Towards digitalization in bio-manufacturing operations: A survey on application of big data and digital twin concepts in denmark. *Frontiers in Chemical Engineering*, 3:727152, 2021.

Valderrama-Bahamóndez, G. I. and Fröhlich, H. Mcmc techniques for parameter estimation of ode based models in systems biology. *Frontiers in Applied Mathematics and Statistics*, 5:55, 2019.

Wan, E. A. and Van Der Merwe, R. The unscented kalman filter for nonlinear estimation. In *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No. 00EX373)*, pp. 153–158. Ieee, 2000.

Wang, D., Tan, D., and Liu, L. Particle swarm optimization algorithm: an overview. *Soft computing*, 22:387–408, 2018.

Yousefi-Darani, A., Paquet-Durand, O., and Hitzmann, B. The kalman filter for the supervision of cultivation processes. *Digital Twins*, pp. 95–125, 2020.

Zhang, D., Del Rio-Chanona, E. A., Petsagkourakis, P., and Wagner, J. Hybrid physics-based and data-driven modeling for bioprocess online simulation and optimization. *Biotechnology and bioengineering*, 116(11):2919–2930, 2019.

Zhang, F., Song, J., Bowden, J. C., Ladd, A., Yue, Y., Desautels, T., and Chen, Y. Learning regions of interest for bayesian optimization with adaptive level-set estimation. In *International Conference on Machine Learning*, pp. 41579–41595. PMLR, 2023.

# A. Background Extension

## A.1. Bayesian Filtering Equations

The purpose of Bayesian filtering is to compute the marginal posterior distribution of the state $\mathbf{x}_k$ at each time step $k$ given the history of the measurements up to the time step $k$ as $p(\mathbf{x}_k|\mathbf{y}_{1:k})$. Then, the fundamental recursive equations of Bayesian filtering theory (Bayesian filtering equations) for computing the predicted distribution $p(\mathbf{x}_k|\mathbf{y}_{1:k-1})$ and the filtering distribution $p(\mathbf{x}_k|\mathbf{y}_{1:k})$ at time step k are given by:

i) An initialization where the recursion starts from the prior distribution $p(\mathbf{x}_0)$;

ii) an **Prediction step** where the predictive distribution of the state $\mathbf{x}_k$ at time step k, given the dynamic model, can be computed by the Chapman–Kolmogorov equation $p(\mathbf{x}_k|\mathbf{y}_{1:k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})d\mathbf{x}_{k-1}$; and

iii) An **Update step** where given the measurement $\mathbf{y}_k$ at time step k, the posterior distribution of the state $\mathbf{x}_k$ can be computed by Bayes' rule $p(\mathbf{x}_k|\mathbf{y}_{1:k}) = \frac{1}{Z_k}p(\mathbf{y}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{y}_{1:k-1})$ where the normalization constant $Z_k$ is given as $Z_k = \int p(\mathbf{y}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{y}_{1:k-1})d\mathbf{x}_k$. See Figure 6.

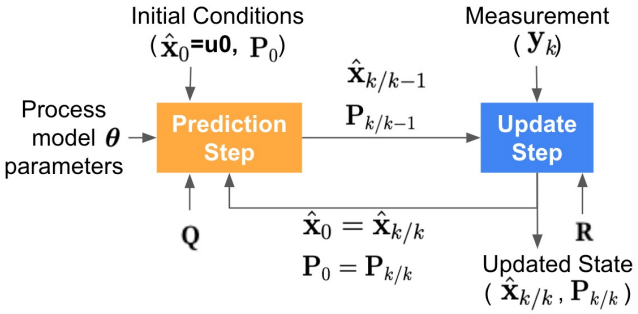


*Figure 6.* Two-step recursive algorithmic process used by Nonlinear Kalman Estimators such as EKF, UKF, RKF and CKF. The state predictions are updated with new measurements, and this updated state becomes the basis for the next prediction.

## A.2. Unstructured Mechanistic Models

UMMs also known as Unstructured Mechanistic Kinetic Models, are pivotal in modeling the temporal progression of bioprocesses like the production of therapeutic monoclonal antibodies (mAbs), projected to generate USD 300 billion by 2025, and rAAV production, a leading viral vector technology for gene therapy (Kyriakopoulos et al., 2018; Luo et al., 2021; Iglesias Jr et al., 2022; 2023). These models, grounded in fundamental principles, are key to understanding and simulating bioprocess dynamics at the macro-scale, such as cell density, viability, and nutrient/metabolite concentrations. Despite their critical role in digital twin (DT) development and soft sensors, the industrial application of UMMs is still nascent (Moser et al., 2020; Park et al., 2021; Mears et al., 2017; Reyes et al., 2022). In contrast to Structured Mechanistic Models (SMMs), which delve into the intracellular details of a homogeneous cell population and are more complex, requiring extensive expertise for development, UMMs are less detailed but more practical for dynamic control in common biomanufacturing bioreactors (Luo et al., 2021; Tsopanoglou & del Val, 2021). SMMs are better suited for cell-line development, focusing on genomic-level alterations for desired process behaviors. However, the predictive capability of simple UMMs is limited, often failing to accurately estimate process states across different operating conditions (Zhang et al., 2019). To enhance their predictive accuracy, UMMs are frequently integrated with the Kalman filter and its nonlinear variants like the extended Kalman filter, effectively predicting unobserved states.

## A.3. Online and Offline measurements

**Online Measurements ($\mathbf{y}_{online}$):** These are measurements taken in real-time. In the context of Nonlinear Kalman Estimators (NKEs), online measurements typically refer to data obtained directly from sensors during the operation of the system. These measurements are used for computing innovation errors, which are essential for the update step in Kalman filtering. Online measurements provide immediate feedback and are crucial for dynamic state estimation where current state information is critical. All of the traditional approaches to auto-tune NKE use only online measurements. Example: Imagine a bioreactor

cultivating a monoclonal antibody (mAb). Sensors in the bioreactor provide real-time measurements of parameters like pH, temperature, oxygen levels, and viable cell density (Xv). These real-time data points ($\mathbf{y}_{online}$) are fed directly into the Nonlinear Kalman Estimators for immediate state estimation and process adjustments. For instance, if the oxygen level drops below a certain threshold, the NKE uses this real-time data to update its state estimates and might signal to increase oxygen supply.

**Offline Measurements ($\mathbf{y}_{offline}$):** Offline measurements, in contrast, are not obtained in real-time but are usually gathered at a lower frequency and can involve more detailed analysis. In many practical applications, these measurements might be taken from the system and analyzed retrospectively, often in a laboratory setting. For example, in a bioprocess, certain biochemical parameters might be measured in this way. Offline measurements provide additional information about the state variables that are not directly involved in the generation of innovation errors. Example: The same bioprocess as above. Samples are periodically taken from the bioreactor and analyzed in a lab to measure concentrations of glucose, lactate, glutamine, and the monoclonal antibody product. These measurements ($\mathbf{y}_{offline}$) are not available in real-time. They might be taken every few hours or once a day, and it takes additional time to process and analyze these samples. Although not available for immediate feedback, this offline data provides a richer, more comprehensive understanding of the bioprocess. It can be used retrospectively to adjust the model parameters in the NKE for future runs or refine the process model, enhancing the accuracy of predictions and estimations of unmeasured state variables.

## B. Proof of Theorem 4.4

**Theorem 4.4** *Assume $\mathbf{y}_k$ includes both online measurements $\mathbf{y}_{online,k}$ (for state variables computing innovation errors) and offline measurements $\mathbf{y}_{offline,k}$ (for other state variables). Then BAT posterior can incorporate $\mathbf{y}_{offline,k}$ as additional information to enhance the auto-tuning process of any NKE, if NKE can predict all state variables.*

*Proof.* Let us consider the following:

- A system state $\mathbf{x}_k \in \mathbb{R}^n$ for k= 0, 1, 2, $\cdots$ T,
- A set of measurements $\mathbf{y}_k \in \mathbb{R}^n$ at k, where $\mathbf{y}_k = \mathbf{y}_{online,k} \cup \mathbf{y}_{offline,k}$,
- A subset of measurements $\mathbf{y}_{online,k} \in \mathbb{R}^{m<n}$ that represents the online measurements of state variables $\mathbf{x}_{IE,k} \in \mathbb{R}^{m<n}$ responsible by Innovation Errors (IE).
- A subset of measurements $\mathbf{y}_{offline,k} \in \mathbb{R}^{n-m}$ that represents the offline measurements of state variables $\mathbf{x}_{NIE,k} \in \mathbb{R}^{n-m}$ that are Not responsible by Innovation Errors (NIE).
- A NKE that uses $\mathbf{y}_{online,k}$ in the update step to predict the mean of all state variables $\hat{\mathbf{x}}_k^{NKE_{PS}} \in \mathbb{R}^n$ during the prediction step, $\hat{\mathbf{x}}_k^{NKE_{PS}} = NKE_{PS}(\mathbf{f}, \boldsymbol{\theta}, t_{k-1}, t_k, \hat{\mathbf{x}}_{0,k-1} = \hat{\mathbf{x}}_{k-1}^{NKE_{US}}, \mathbf{P}_{0,k-1} = \mathbf{P}_{k-1}^{NKE_{US}}, \mathbf{Q}_{k-1}, \mathbf{R}_{k-1})$.
- $\mathbf{y}_k$ is described by a measurement model (outside of NKE update step) composed of $\hat{\mathbf{x}}_k^{NKE_{PS}}$ and additive noise as $\mathbf{y}_k = h(\hat{\mathbf{x}}_k^{NKE_{PS}}) + \boldsymbol{\delta}_k$. Where $\boldsymbol{\delta}_k \sim N(0, \boldsymbol{\sigma}^2)$ , and $h(\cdot)$ is a linear/non linear function that relates the predicted mean of state variables to the measurements.

Given these consideration, we can define the BAT likelihood as

$$
\begin{aligned}
p(\mathbf{y}|h(\hat{\mathbf{x}}^{NKE_{PS}}), \boldsymbol{\sigma}^2) = \prod_{k=1}^{T} p(\mathbf{y}_k|h(NKE_{PS}(\mathbf{f}, \boldsymbol{\theta}, t_{k-1}, t_k, \hat{\mathbf{x}}_{0,k-1}, \mathbf{P}_{0,k-1}, \mathbf{Q}_{k-1}, \mathbf{R}_{k-1}))) = \\
\prod_{k=1}^{T} p(\mathbf{y}_k|\boldsymbol{\theta}, \hat{\mathbf{x}}_{0,0}, \mathbf{P}_{0,0}, \mathbf{Q}_{k-1}, \mathbf{R}_{k-1}, \boldsymbol{\sigma}^2) = \\
\prod_{k=1}^{T} p(\mathbf{y}_{online,k}, \mathbf{y}_{offline,k}|\boldsymbol{\theta}, \hat{\mathbf{x}}_{0,0}, \mathbf{P}_{0,0}, \mathbf{Q}_{k-1}, \mathbf{R}_{k-1}, \boldsymbol{\sigma}^2),
\end{aligned}
\tag{9}
$$

and consequently to define the following BAT priors $p(\boldsymbol{\sigma}^2), p(\mathbf{Q}_{k-1}), p(\mathbf{R}_{k-1}), p(\boldsymbol{\theta}), p(\hat{\mathbf{x}}_{0,0})$, and $p(\mathbf{P}_{0,0})$, where $\hat{\mathbf{x}}_{0,0}$ and $\mathbf{P}_{0,0}$ are the first initial conditions.

Then, by applying Bayes' rule, we have the following posterior density

$$
\begin{aligned}
p(\boldsymbol{\theta}, \hat{\mathbf{x}}_{0,0}, \mathbf{P}_{0,0}, \mathbf{Q}, \mathbf{R}, \boldsymbol{\sigma}^2|\mathbf{y}_{online}, \mathbf{y}_{offline}) \propto p(\hat{\mathbf{x}}_{0,0}) \times p(\mathbf{P}_{0,0}) \times \\
p(\boldsymbol{\sigma}^2) \times \prod_{k=1}^{T} p(\mathbf{y}_{online,k}, \mathbf{y}_{offline,k}|\boldsymbol{\theta}, \hat{\mathbf{x}}_{0,0}, \mathbf{P}_{0,0}, \mathbf{Q}_{k-1}, \mathbf{R}_{k-1}, \boldsymbol{\sigma}^2) \\
\times p(\mathbf{Q}_{k-1}) \times p(\mathbf{R}_{k-1}).
\end{aligned}
\tag{10}
$$

Since, samples can be extracted from this posterior by MCMC methods and the mean of resulted distributions of $\boldsymbol{\theta}, \hat{\mathbf{x}}_{0,0}, \mathbf{P}_{0,0}, \mathbf{Q}, \mathbf{R}$ can be computed. Then we have that BAT posterior can use all available measurement data, including $\mathbf{y}_{offline,k}$ that are not employed in the NKE update step, to auto-tune the $\boldsymbol{\theta}, \hat{\mathbf{x}}_{0,0}, \mathbf{P}_{0,0}, \mathbf{Q}, \mathbf{R}$ of any NKE.

□

## C. Theoretical Application of BAT to Estimate all NKEs components

The Figure 7 gives an overview about how is the process of sampling from the BAT posterior distribution using MCMC. In this work, we used NUTS, but it is also possible to use other methods such as HMC.

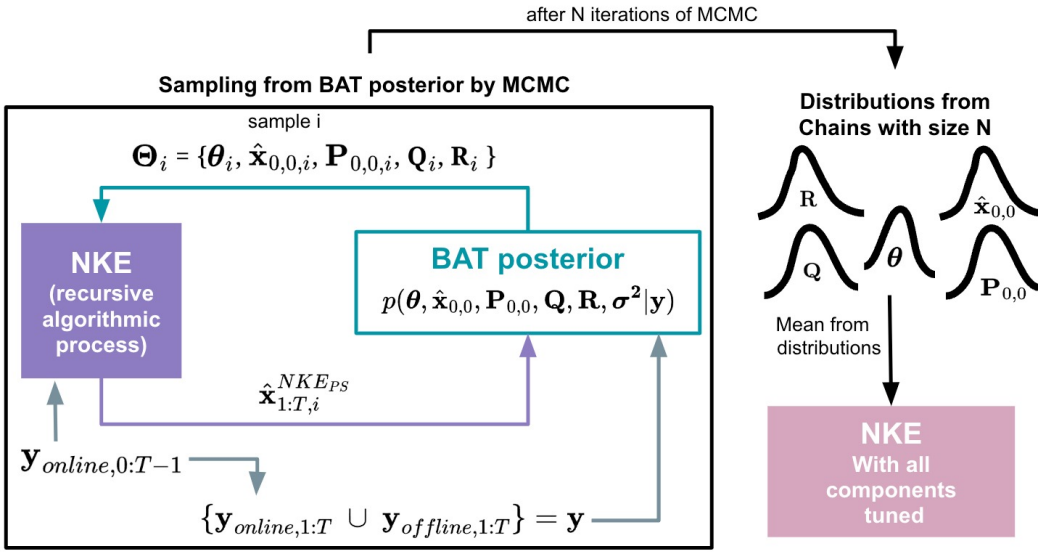


*Figure 7.* BAT: Batch Bayesian Auto-Tuning for Nonlinear Kalman Estimators. BAT posterior of $\boldsymbol{\theta}, \hat{\mathbf{x}}_{0,0}, \mathbf{P}_{0,0}, \mathbf{Q}, \mathbf{R}$) given $\mathbf{y}$ is defined outside of NKE recursive algorithm that after complete T steps produce a set of predicted mean of states variables $\hat{\mathbf{x}}_{1:T,i}^{NKE_{PS}} = [\hat{\mathbf{x}}_{1,i}^{NKE_{PS}}, ..., \hat{\mathbf{x}}_{T,i}^{NKE_{PS}}]$. $\mathbf{y}$ represent a full set of measured data of all state variables of the system composed of online measurements $\mathbf{y}_{online,1:T}$ (used to generate innovation errors) and offline measurements $\mathbf{y}_{offline,1:T}$ of other state variables. Since $\mathbf{y}$ can be described by a measurement model (Equation 4) outside of the NKE, it is possible to define the BAT likelihood (Equation 7) and priors to obtain the BAT posterior (Equation 8). Then, samples can be extracted by MCMC methods, and after N iterations of MCMC, the mean/mode values can be extracted from the obtained distribution of $\boldsymbol{\theta}, \hat{\mathbf{x}}_{0,0}, \mathbf{P}_{0,0}, \mathbf{Q}, \mathbf{R}$ to tune all NKE components. It is important to that the NKE loop start from update step then the first measurement is included in $\mathbf{y}_{online,0:T-1}$, but not in $\mathbf{y}_{online,1:T}$.

### C.1. BAT for EKF

Here, we describe how BAT can be applied to EKF. However, a similar approach can be used for UKF and CKF.

So, let us assume that each observation, $\mathbf{y}_k$ for $k = 1, \ldots, T$, has an associated additive noise process, $\boldsymbol{\delta}_k \sim N(0, \boldsymbol{\sigma}^2)$ , such that

$$\mathbf{y}_k = h(\hat{\mathbf{x}}_k^{EKF_{PS}}) + \boldsymbol{\delta}_k, \tag{11}$$

$$\langle \hat{\mathbf{x}}_k^{NKE_{PS}}, \mathbf{P}_k^{NKE_{PS}} \rangle = EKF_{PS}(\hat{\mathbf{x}}_{0,k-1} = \hat{\mathbf{x}}_{k-1}^{EKF_{US}}, \mathbf{f}, \boldsymbol{\theta}, t_{k-1}, t_k, \mathbf{Q}_{k-1}, \mathbf{R}_{k-1}, \mathbf{P}_{0,k-1} = \mathbf{P}_{k-1}^{EKF_{US}}) \tag{12}$$

where $\boldsymbol{\delta}_k$ is a D × 1 vector and $\hat{\mathbf{x}}_k^{EKF_{PS}}$ (predicted mean) is the solution obtained by the **EKF prediction step** at time $t_k$ for the state variables vector, beside of predicted error covariance matrix $\mathbf{P}_k^{EKF_{PS}}$. This solution is obtained given $\boldsymbol{\theta}$, $\mathbf{Q}_{k-1}$ and the results of the **EKF update step** at time $t_{k-1}$. These results are the estimated mean $\hat{\mathbf{x}}_{k-1}^{EKF_{US}}$ and estimated error covariance matrix $\mathbf{P}_{k-1}^{EKF_{US}}$ of states variables, and they were generated by the **EKF update step** given $\mathbf{R}_{k-1}$, $\mathbf{y}_{k-1}$ and the previous predicted mean $\hat{\mathbf{x}}_{k-1}^{EKF_{PS}}$ and predicted error covariance matrix $\mathbf{P}_{k-1}^{EKF_{PS}}$. It is important to point out that the $\hat{\mathbf{x}}_{k-1}^{EKF_{US}}$ is used as initial conditions ($\hat{\mathbf{x}}_{0,k-1}$) in the **EKF prediction step** to solve the dynamic system (ODE system) $\dot{\mathbf{x}}(t)$ from $t_{k-1}$ to $t_k$ through a black-box numerical ODE solver. The same is done with $\mathbf{P}_{k-1}^{EKF_{US}}$ where it is initial condition ( $\mathbf{P}_{0,k-1}$) to solve the matrix Ricatti differential equation (MRDE) from $t_{k-1}$ to $t_k$ through a black-box numerical ODE solver. This enables us to rewrite the Equation 12 as following

$$\langle \hat{\mathbf{x}}_k^{EKF_{PS}}, \mathbf{P}_k^{EKF_{PS}} \rangle = ODESolve^{EKF_{PS}}(\hat{\mathbf{x}}_{0,k-1} = \hat{\mathbf{x}}_{k-1}^{EKF_{US}}, \mathbf{f}, \boldsymbol{\theta}, t_{k-1}, t_k, \mathbf{Q}_{k-1}, \mathbf{R}_{k-1}, \mathbf{P}_{0,k-1} = \mathbf{P}_{k-1}^{EKF_{US}}). \tag{13}$$

Then, under a Gaussian error model (we assumed a Gaussian model for simplicity and illustration purposes, but we can assume any kind of error model), and assuming the $\boldsymbol{\delta}_k$ are independent of each other, $\mathbf{y}_k$ follows a multivariate normal distribution:

$$\mathbf{y}_k \approx MVN(h(\hat{\mathbf{x}}_k^{EKF_{PS}}), \sum(\boldsymbol{\sigma}^{\mathbf{2}})), \tag{14}$$

$$\mathbf{y}_k \approx MVN(h(ODESolve^{EKF_{PS}}(\hat{\mathbf{x}}_{0,k-1}, \mathbf{f}, \boldsymbol{\theta}, t_{k-1}, t_k, \mathbf{Q}_{k-1}, \mathbf{R}_{k-1}, \mathbf{P}_{0,k-1})), \sum(\boldsymbol{\sigma}^{\mathbf{2}})), \tag{15}$$

where $\sum(\boldsymbol{\sigma}^{\mathbf{2}})$ is a diagonal matrix with diagonal elements. Hence, the likelihood function is given by

$$\begin{aligned}\mathcal{L}(\mathbf{y}|h(\hat{\mathbf{x}}_k^{EKF_{PS}}), \boldsymbol{\sigma}^{\mathbf{2}}) = \prod_{k=1}^{T} MVN(\mathbf{y}_k; h(\hat{\mathbf{x}}_k^{EKF_{PS}}), \sum(\boldsymbol{\sigma}^{\mathbf{2}})) = \\ \prod_{k=1}^{T} MVN(\mathbf{y}_k; h(ODESolve^{EKF_{PS}}(\hat{\mathbf{x}}_{0,k-1}, \mathbf{f}, \boldsymbol{\theta}, t_{k-1}, t_k, \mathbf{Q}_{k-1}, \mathbf{R}_{k-1}, \mathbf{P}_{0,k-1})), \sum(\boldsymbol{\sigma}^{\mathbf{2}})),\end{aligned} \tag{16}$$

and the posterior density is

$$\begin{aligned}p(\boldsymbol{\theta}, \boldsymbol{\sigma}^2, \hat{\mathbf{x}}_{0,set}, \mathbf{Q}, \mathbf{R}, \mathbf{P}_{0,set}|\mathbf{y}) \propto p(\boldsymbol{\theta}) \times p(\boldsymbol{\sigma}^2) \times \prod_{k=1}^{T} MVN(\mathbf{y}_k; h(\hat{\mathbf{x}}_k^{EKF_{PS}}), \sum(\boldsymbol{\sigma}^{\mathbf{2}})) \\ \times p(\hat{\mathbf{x}}_{0,k-1}) \times p(\mathbf{P}_{0,k-1}) \times p(\mathbf{Q}_{k-1}) \times p(\mathbf{R}_{k-1}).\end{aligned} \tag{17}$$

where the $\mathbf{R} = (\mathbf{R}_0, \mathbf{R}_1, ..., \mathbf{R}_{T-1})$, $\mathbf{Q} = (\mathbf{Q}_0, \mathbf{Q}_1, ..., \mathbf{Q}_{T-1})$, $\mathbf{P}_{0,set} = (\mathbf{P}_{0,0}, \mathbf{P}_{0,1}, ..., \mathbf{P}_{0,T-1})$ and $\hat{\mathbf{x}}_{0,set} = (\hat{\mathbf{x}}_{0,0}, \hat{\mathbf{x}}_{0,1}, ..., \hat{\mathbf{x}}_{0,T-1})$. Applying BAT to the EKF involves using MCMC techniques to sample from the posterior distribution defined in Equation 17. This process iteratively refines the estimates of all NKE components leading to an auto-tuned EKF that is better adapted to the specific characteristics of the observed data.

It is important to point out that in our experiments with EKF we used the Bogacki-Shampine 3/2 method (BS3) as ODE solver. The BS3 ODE solver is recommended for efficiently solving non-stiff problems at higher tolerances, particularly when the focus is on achieving fast solutions. This solver is part of the OrdinaryDiffEq.jl suite and is noted for its effectiveness in scenarios where quick computation is a priority while still maintaining acceptable accuracy.

### C.2. BAT for UKF

This section presents a detailed theoretical application of BAT for the Unscented Kalman Filter (UKF). The UKF, known for its ability to handle highly nonlinear systems, uses a deterministic sampling technique (sigma points) to approximate the state distribution. Let us consider a observation model as defined in Equation 4 with the specific prediction mechanism of UKF:

$$\mathbf{y}_k = h(\hat{\mathbf{x}}_k^{UKF_{PS}}) + \boldsymbol{\delta}_k, \tag{18}$$

where $\boldsymbol{\delta}_k \sim N(0, \boldsymbol{\sigma}^2)$ is the D-dimensional additive noise vector at time step $k$, and $\hat{\mathbf{x}}_k^{UKF_{PS}}$ is the predicted state from the UKF prediction step. The UKF prediction step is detailed as follows:

$$\langle \hat{\mathbf{x}}_k^{UKF_{PS}}, \mathbf{P}_k^{UKF_{PS}} \rangle = UKF_{PS}(\hat{\mathbf{x}}_{0,k-1} = \hat{\mathbf{x}}_{k-1}^{UKF_{US}}, \mathbf{f}, \boldsymbol{\theta}, t_{k-1}, t_k, \mathbf{Q}_{k-1}, \mathbf{R}_{k-1}, \mathbf{P}_{0,k-1} = \mathbf{P}_{k-1}^{UKF_{US}}), \tag{19}$$

where:

- $\mathbf{f}$ is the nonlinear process model function.
- $\boldsymbol{\theta}$ are the parameters of the process model.
- $\mathbf{Q}_{k-1}$ and $\mathbf{R}_{k-1}$ are the process and measurement noise covariance matrices, respectively.
- $\hat{\mathbf{x}}_{0,k-1}$ and $\mathbf{P}_{0,k-1}$ are the updated state $\hat{\mathbf{x}}_{k-1}^{UKF_{US}}$ and the error covariance matrix $\mathbf{P}_{k-1}^{UKF_{US}}$ from the previous update step given $\mathbf{R}_{k-1}$ , $\mathbf{y}_{k-1}$ and the previous predicted mean $\hat{\mathbf{x}}_{k-1}^{UKF_{PS}}$ and predicted error covariance matrix $\mathbf{P}_{k-1}^{UKF_{PS}}$.

The UKF utilizes a set of deterministically chosen sigma points to capture the mean and covariance of the Gaussian-distributed state estimate. These sigma points are propagated through the nonlinear model $\mathbf{f}$, and their statistics are used to estimate the new state mean and covariance. For an $n$-dimensional state space, the UKF selects $2n + 1$ sigma points. These points, denoted as $\mathbf{s}_i$, are strategically placed around the state estimate's mean, with their distribution designed to match the original state's mean and covariance. Mathematically, this is represented as: Sigma Points $= \mathbf{s}_0, \mathbf{s}_1, \ldots, \mathbf{s}_{2n}$ where $\mathbf{s}_0$ is the mean of the state estimate, and the remaining points are spread along the dimensions of the covariance matrix, $\mathbf{P}$, of the state. Each sigma point, $\mathbf{s}_i$, undergoes propagation through the nonlinear system model $\mathbf{f}$, capturing the influence of non-linear dynamics on the state distribution. This process is described as: $\mathbf{s}'_i = \mathbf{f}(\mathbf{s}_i), \quad i = 0, 1, \ldots, 2n$ The propagated sigma points, $\mathbf{s}'_i$, form a new set that approximates the state distribution after transformation. The updated state mean, $\hat{\mathbf{x}}_k^{UKF_{PS}}$, and covariance, $\mathbf{P}_k^{UKF_{PS}}$, are computed as weighted averages of these points: $\hat{\mathbf{x}}_k^{UKF_{PS}} = \sum_{i=0}^{2n} W_i^m \mathbf{s}'_i, \quad \mathbf{P}_k^{UKF_{PS}} = \sum_{i=0}^{2n} W_i^c (\mathbf{s}'_i - \hat{\mathbf{x}}_k^{UKF_{PS}})(\mathbf{s}'_i - \hat{\mathbf{x}}_k^{UKF_{PS}})^T + \mathbf{Q}$, where $W_i^m$ and $W_i^c$ are the weights for the mean and covariance, respectively. This approach is particularly advantageous in scenarios with non-linear state dynamics, as it avoids the linearization errors inherent in the EKF. Despite its enhanced accuracy in non-linear contexts, the UKF requires more computational resources due to the propagation of multiple sigma points through the non-linear model $\mathbf{f}$. This increased computational demand is often a worthwhile trade-off for the improved fidelity in representing the state's probability distribution under non-linear transformations.

Then, the BAT posterior density for the UKF can be formulated as:

$$\begin{aligned} p(\boldsymbol{\theta}, \boldsymbol{\sigma}^2, \hat{\mathbf{x}}_{0,set}, \mathbf{Q}, \mathbf{R}, \mathbf{P}_{0,set} | \mathbf{y}) \propto p(\boldsymbol{\theta}) \times p(\boldsymbol{\sigma}^2) \times \prod_{k=1}^{T} MVN(\mathbf{y}_k; h(\hat{\mathbf{x}}_k^{UKF_{PS}}), \sum(\boldsymbol{\sigma}^2)) \\ \times p(\hat{\mathbf{x}}_{0,k-1}) \times p(\mathbf{P}_{0,k-1}) \times p(\mathbf{Q}_{k-1}) \times p(\mathbf{R}_{k-1}). \end{aligned} \tag{20}$$

#### C.2.1. PENDULUM CASE

Aiming to illustrate the application of BAT for UKF, we explore the Pendulum tracking with UKF presented in (Särkkä & Svensson, 2023) that is a classic problem.

*Table 4.* Results of sampling the BAT for UKF with simulated data of pendulum.

| UKF COMPONENTS | GROUND TRUTH | BAT MEAN | BAT STD |
|---|---|---|---|
| $Q_{1,1}$ | 3.3E-9 | 3.3345E-9 | 9.826E-11 |
| $Q_{2,2}$ | 5.0E-7 | 5.0011E-7 | 1.030E-8 |
| $Q_{3,3}$ | 5.0E-7 | 5.0006E-7 | 1.001E-8 |
| $Q_{4,4}$ | 0.0001 | 9.9708E-5 | 9.5978E-6 |
| $R$ | 0.1 | 0.101 | 0.00967 |
| $g$ | 9.81 | 9.811 | 0.0192 |
| $P11$ | 1 | 1.029 | 0.0779 |
| $\alpha$ | 1 | 0.999 | 0.00983 |

The main idea is estimate the UKF parameters based on the simulated data of pendulum. The Table 4 shows the results obtained by sample the BAT posterior of UKF parameters give the simulated data. All estimation are closed to the ground truth. In addition, we also estimated the hyper-parameter of $\alpha$ of UKF. The result of applying the designed UKF with the estimated parameters (Table 4) to the pendulum model and simulated data is shown in 8. The details about the dynamic model of pendulum and simulated data can be seen in (Särkkä & Svensson, 2023). Furthermore, the code of this example is available in our repository `https://anonymous.4open.science/r/BAT`.

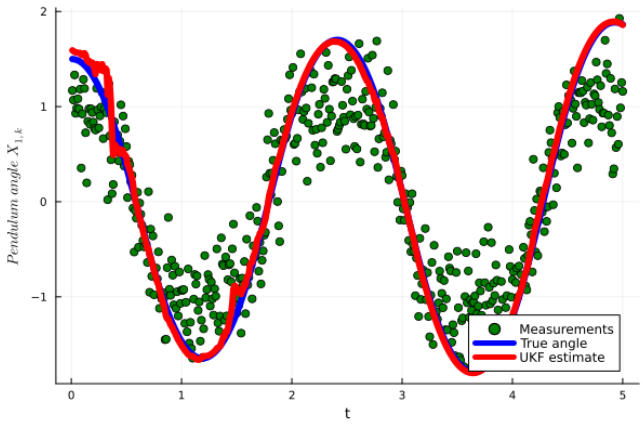


*Figure 8.* Simulated pendulum data and the result of tracking the pendulum using the UKF parameters estimated with BAT posterior.

### C.3. BAT for CKF

Again, let us consider a observation model as defined in Equation 4 with the specific prediction mechanism of CKF:

$$\mathbf{y}_k = h(\hat{\mathbf{x}}_k^{CKF_{PS}}) + \boldsymbol{\delta}_k, \tag{21}$$

where $\boldsymbol{\delta}_k \sim N(0, \boldsymbol{\sigma}^2)$is the D-dimensional additive noise vector at time step $k$, and $\hat{\mathbf{x}}_k^{CKF_{PS}}$ is the predicted state from the CKF prediction step. It is essential for propagating the state and error covariance through the nonlinear system dynamics. The CKF prediction step is detailed as follows:

$$\langle \hat{\mathbf{x}}_k^{CKF_{PS}}, \mathbf{P}_k^{CKF_{PS}} \rangle = CKF_{PS}(\hat{\mathbf{x}}_{0,k-1} = \hat{\mathbf{x}}_{k-1}^{CKF_{US}}, \mathbf{f}, \boldsymbol{\theta}, t_{k-1}, t_k, \mathbf{Q}_{k-1}, \mathbf{R}_{k-1}, \mathbf{P}_{0,k-1} = \mathbf{P}_{k-1}^{CKF_{US}}), \tag{22}$$

where:

- $\mathbf{f}$ represents the nonlinear process model function.
- $\boldsymbol{\theta}$ denotes the parameters of the process model.
- $\mathbf{Q}_{k-1}$ and $\mathbf{R}_{k-1}$ are the process and measurement noise covariance matrices, respectively.
- $\hat{\mathbf{x}}_{0,k-1}$ and $\mathbf{P}_{0,k-1}$ are the updated state $\hat{\mathbf{x}}_{k-1}^{CKF_{US}}$ and the error covariance matrix $\mathbf{P}_{k-1}^{CKF_{US}}$ from the previous update step given $\mathbf{R}_{k-1}$ , $\mathbf{y}_{k-1}$ and the previous predicted mean $\hat{\mathbf{x}}_{k-1}^{CKF_{PS}}$ and predicted error covariance matrix $\mathbf{P}_{k-1}^{CKF_{PS}}$.

The CKF employs cubature points, which are deterministically selected to approximate the integral of the nonlinear transformation of the state. These points are designed to capture the mean and covariance of the Gaussian-distributed state estimate accurately. The CKF propagates these cubature points through the nonlinear model $\mathbf{f}$, and then uses the results to estimate the new state mean and covariance. The cubature points are calculated as:

$$\mathbf{s}_i = \sqrt{\mathbf{P}_{k-1}}\,\mathbf{e}_i, \quad i = 1, \ldots, 2n \tag{23}$$

where $\mathbf{e}_i$ are the unit vectors in the state space, and $n$ is the dimension of the state space. The square root of the covariance matrix, $\sqrt{\mathbf{P}_{k-1}}$, can be computed using methods such as the Cholesky decomposition. Each cubature point is then propagated through the nonlinear model:

$$\mathbf{s}'_i = \mathbf{f}(\mathbf{s}_i, \boldsymbol{\theta}, t_k), \quad i = 1, \ldots, 2n \tag{24}$$

The new state mean and covariance are estimated from these propagated points:

$$\hat{\mathbf{x}}_k^{CKF_{PS}} = \frac{1}{2n} \sum_{i=1}^{2n} \mathbf{s}'_i \tag{25}$$

$$\mathbf{P}_k^{CKF_{PS}} = \frac{1}{2n} \sum_{i=1}^{2n} (\mathbf{s}'_i - \hat{\mathbf{x}}_k^{CKF_{PS}})(\mathbf{s}'_i - \hat{\mathbf{x}}_k^{CKF_{PS}})^T + \mathbf{Q} \tag{26}$$

This approach allows the CKF to effectively handle the non-linearities in the system model, providing a more accurate state estimation compared to linear approximation methods.

Then, the BAT posterior density for the CKF, within the Bayesian framework, can be formulated as:

$$\begin{aligned} p(\boldsymbol{\theta}, \boldsymbol{\sigma}^2, \hat{\mathbf{x}}_{0,set}, \mathbf{Q}, \mathbf{R}, \mathbf{P}_{0,set} | \mathbf{y}) \propto p(\boldsymbol{\theta}) \times p(\boldsymbol{\sigma}^2) \times \prod_{k=1}^{T} MVN(\mathbf{y}_k; h(\hat{\mathbf{x}}_k^{CKF_{PS}}), \sum(\boldsymbol{\sigma}^2)) \\ \times p(\hat{\mathbf{x}}_{0,k-1}) \times p(\mathbf{P}_{0,k-1}) \times p(\mathbf{Q}_{k-1}) \times p(\mathbf{R}_{k-1}). \end{aligned} \tag{27}$$

*Table 5.* Results of sampling the BAT for CKF with simulated data of pendulum.

| CKF COMPONENTS | GROUND TRUTH | BAT MEAN | BAT STD |
|---|---|---|---|
| $\Delta t$ | 0.01 | 0.0091 | 0.0006 |
| $R$ | 0.1 | 0.1009 | 0.0101 |
| $g$ | 9.81 | 9.8097 | 0.0204 |

#### C.3.1. PENDULUM CASE

Aiming to illustrate the application of BAT for CKF, we explore the same classical problem of subsection C.2.1.

The main idea is also to estimate the CKF parameters based on the simulated data of pendulum. The Table 5 shows the results obtained by sample the BAT posterior of CKF parameters give the simulated data. All estimation are closed to the ground truth. Here, as an illustrative application, instead to try to estimate a hyper-parameter, we estimated $\Delta t$ used to define the process noise covariance matrix $\mathbf{Q}$. The details about the dynamic model of pendulum and simulated data can be seen in (Särkkä & Svensson, 2023). Furthermore, the code of this example is available in our repository `https://anonymous.4open.science/r/BAT`.

## D. Related work extension

The novelty in BAT lies in how it handles the outputs from the NKE prediction step, integrating these predictions with all measurement data available to form a comprehensive posterior distribution, which is then sampled using MCMC, see Figure 7. Contrary to the traditional auto-tuning approaches (such as (Abbeel et al., 2005)), we used only the output from the NKE prediction step and we also estimated the covariance represented by $\boldsymbol{\sigma}^2$ that is responsable by the additive noise $\boldsymbol{\delta}_k \sim N(0, \boldsymbol{\sigma}^2)$ of measurement model outside of NKE recursive loop (Equation 4).

### D.1. State Augmentation Failure Case: Biomanufacturing conditions

The concept of state augmentation is driven by the necessity to refine the predictions of a process model about state variables and to dynamically adapt the model's parameters based on these refinements. This approach suggests that a process model must be continuously adjusted (evolved) under varying conditions within the same application. Take biomanufacturing as an example: the parameters of a process model used for overseeing a cell culture should be modified for each unique set of conditions. Initially, a general set of parameters may be employed at the beginning of the process, but to enhance the accuracy of state predictions of the cell culture, these parameters must be progressively updated. In this method, state augmentation leverages every new measurement to simultaneously correct the process model's predictions and update its parameters (Iglesias & Bolic, 2024).

The following conditions are prevalent in biomanufacturing and should be taken into consideration while developing of NKE with state augmentation for this area, because NKE with state augmentation fail to estimate state and unshared parameters simultaneously with this conditions (Iglesias & Bolic, 2024):

- **ODEs of Unstructured Mechanistic Model (UMM) with Unshared Parameters:** Parameters unique to one term of an ODE and not shared with other ODEs in the UMM are typical for modeling product formation dynamics in biomanufacturing.
- **P and Q with Uncorrelated Elements:** Often, limited data leads to assuming error covariance matrices $\mathbf{P}$ (process error covariance) and $\mathbf{Q}$ (measurement error covariance) with uncorrelated elements, meaning they are diagonal with nonzero diagonal elements (noise variances) and zero off-diagonal elements. This results in two scenarios: 1) *Using **P** with uncorrelated elements for MRDE construction and as **P**(t=0) initial condition:* Here, the MRDE ODEs depend only on noise variances of $\mathrm{P}_{i,i}$ and $\mathrm{Q}_{i,i}$, and elements of Jacobian $\mathbf{J}_t^{\phi}$. 2) *Using **P** with correlated elements for MRDE construction and **P**(t=0) with uncorrelated elements as initial condition:* This can reduce time-invariant ODEs predicting state error covariance between two state variables.
- **ODEs of UMM with Weak Terms:** Weak terms in an ODE have less impact on the predicted state error covariance $\mathbf{P}(t_{k|k-1})$ compared to strong terms. The Jacobian $\mathbf{J}_t^{\phi}$ is more influenced by strong terms.

- **ODEs of UMM with Weak Variables:** Weak variables, present only in the first member of an ODE, do not contribute to the computation of predicted error covariance $\mathbf{P}(t_{k|k-1})$, as their first-order partial derivatives in Jacobian $\mathbf{J}_t^{\phi}$ are zero. In contrast, strong variables significantly influence $\mathbf{P}(t_{k|k-1})$ computation.

- **Only One Measured State Variable:** In some JEKF applications, only a single state variable is measured. This variable determines the column of predicted state error covariance $\mathbf{P}(t_{k|k-1})$ used for Kalman gain computation. If a row in this column is zero (no covariance between the measured and state variable), the Kalman gain for the state variable represented by that row cannot be computed.

These conditions emphasize the complexities and limitations in applying NKE with state augmentation in biomanufacturing, where unique parameter characteristics and measurement constraints can impact its effectiveness (Iglesias & Bolic, 2024).

### D.2. Hybrid NKE

In recent years, the integration of neural networks with Kalman filter estimation has gained significant attention,(Kim et al., 2022). These online approaches were developed to deal with cases where $\mathbf{Q}$ and $\mathbf{R}$ are not constant. In dynamic environments, the performance of the KF can be degraded by using constant noise covariance matrices. However, these filters are often adaptive, with covariances computed and adjusted based on measurements and environmental conditions. A notable limitation of these methods is the lack of consideration for consistency conditions and the challenge in proving the stability of AI-designed filters. Similar concerns are raised in other studies combining reinforcement learning with Kalman filter estimation, such as (Gu et al., 2022) and (Tang et al., 2021). It is important to point out that we consider the case of estimate non-constant $\mathbf{R}$ in task 1 of our empirical evaluation. However, we do not include hybrid NKEs as baseline models because they are online approaches. In addition, they were not validated properly. They do not address the filter consistency assessment, and therefore, it is difficult to prove the stability of the designed filters.

### D.3. Auto-tuning approaches for UKF and CKF

**UKF.** To optimize the tuning of an UKF, a two-stage Bayesian optimization (TSBO) methodology can be used. This involves using a t-Student Process to optimize the process noise parameters of the UKF (Bertipaglia et al., 2022). In (Hussain et al., 2023) was proposed the a multi-objective function genetic algorithm to find the optimal values of state and noise covariance matrices for optimizing the unscented Kalman filter tuning. Another approach is to perform the optimization of unscented Kalman filter tuning using Bayesian optimization (BO) for open and closed-loop performance in autonomous underwater vehicle (Nitsch et al., 2023).

**CKF.** There is a lack of works providing a specific discuss about the optimization of cubature Kalman filter. However, the optimization of Cubature Kalman Filter (CKF) tuning can be achieved using BO techniques. BO is a principled approach to optimization-based estimator tuning that can handle the nonlinear and stochastic relationship between noise values and estimator performance.

# E. Experimental Details

## E.1. Unstructured Mechanistic Model for mAb Production

BAT and baselines used the same dynamic model, which is an ODE system (Equations 28) of mAb production proposed in (Liu & Gunawan, 2017). The ODE system (Equations 28) is a Unstructured Mechanistic Model (UMM) (description in Section A.2) used for mAb production (Liu & Gunawan, 2017). This system represents the cell growth, uptake of substrates, metabolism, and production process with 18 parameters described in the Table 6. It is important to point out that $QmAb$ denotes the specific mAb production rate, and is an example of unshared parameter. More details can be found in (Liu & Gunawan, 2017).

$$
\begin{aligned}
\frac{d\,X_V}{dt} &= (\mu - \mu_d)X_V \\
\frac{d\,X_t}{dt} &= \mu X_V - k_{lysis}(X_t - X_V) \\
\mu &= \mu_{max}.\frac{[GLC]}{K_{glc}+[GLC]}.\frac{[GLN]}{K_{gln}+[GLN]}.\frac{K_{Ilac}}{K_{Ilac}+[LAC]}.\frac{K_{Iamm}}{K_{Iamm}+[AMM]} \\
\mu_d &= \frac{\mu_{d,max}}{1+(K_{d,amm}+[AMM])^2} \\
\frac{d\,[GLC]}{dt} &= -Q_{glc}Xv \\
\frac{d\,[GLN]}{dt} &= -Q_{gln}Xv - K_{d,gln}[GLN] \\
\frac{d\,[LAC]}{dt} &= Q_{lac}Xv \\
\frac{d\,[AMM]}{dt} &= Q_{amm}Xv + K_{d,gln}[GLN] \\
Q_{glc}Xv &= \frac{\mu}{Y_{x,glc}} + m_{glc} \\
Q_{gln}Xv &= \frac{\mu}{Y_{x,gln}} + m_{gln} = \frac{\mu}{Y_{x,gln}} + \frac{\alpha_2[GLN]}{\alpha_2+[GLN]} \\
Q_{lac}Xv &= Y_{lac,glc}Q_{glc} \\
Q_{amm}Xv &= Y_{amm,gln}Q_{gln} \\
\frac{d\,[mAb]}{dt} &= (2-\gamma\mu)Q_{mAb}.X_V
\end{aligned}
\tag{28}
$$

Let's break down the components of this ODE system:

1. **Cell Growth and Death Dynamics:**

   • $\frac{dX_V}{dt} = (\mu - \mu_d)X_V$: This equation models the rate of change of viable cell density ($X_V$) over time. The growth rate ($\mu$) minus the death rate ($\mu_d$) is multiplied by the current viable cell density.

   • $\frac{dX_t}{dt} = \mu X_V - k_{lysis}(X_t - X_V)$: This equation describes the total cell density ($X_t$), considering both viable and non-viable cells. The rate of total cell density change is determined by the growth of viable cells and the lysis (breakdown) of cells, where $k_{lysis}$ is the lysis rate constant.

2. Growth Rate ($\mu$) and Death Rate ($\mu_d$):

   • $\mu$: Defined as a function of substrate concentrations ([GLC] for glucose and [GLN] for glutamine) and inhibitors ([LAC] for lactate and [AMM] for ammonium). This function reflects how cell growth rate is influenced by the availability of nutrients and the presence of metabolic byproducts.

   • $\mu_d$: The death rate, modeled as a function of the ammonium concentration, with $\mu_{d,max}$ representing the maximum death rate and $K_{d,amm}$ as a constant.

3. Substrate Consumption and Metabolite Production:

   • The following set of equations ($\frac{d[GLC]}{dt}$, $\frac{d[GLN]}{dt}$, $\frac{d[LAC]}{dt}$, $\frac{d[AMM]}{dt}$) represent the rates of change in concentrations of glucose, glutamine, lactate, and ammonium, respectively. These are key substrates and metabolites in the cell culture. The terms $Q_{glc}, Q_{gln}, Q_{lac}, Q_{amm}$ denote specific consumption/production rates of these components, and $K_{d,gln}$ is the degradation constant for glutamine.

4. Balancing Equations for Substrate Consumption and Product Formation:

   • The equations relating $Q_{glc}Xv, Q_{gln}Xv, Q_{lac}Xv, Q_{amm}Xv$ establish relationships between growth rate, substrate consumption, and metabolite production rates. These are based on yield coefficients ($Y_{x,glc}, Y_{x,gln}, Y_{lac,glc}, Y_{amm,gln}$) and maintenance coefficients ($m_{glc}, m_{gln}, \alpha_2$).

5. Monoclonal Antibody (mAb) Production:

   • $\frac{d[mAb]}{dt} = (2 - \gamma\mu)Q_{mAb}.X_V$: This equation models the rate of mAb production. The specific mAb production rate ($Q_{mAb}$) is multiplied by the viable cell density and a factor considering the growth rate, where $\gamma$ is a constant.

The model's strength lies in its ability to capture the interplay between cell growth, nutrient consumption, metabolite accumulation, and product formation, which are crucial for optimizing and monitoring biomanufacturing processes. The parameter $Q_{mAb}$, representing the specific mAb production rate, is particularly notable as it's an unshared parameter, meaning its value is unique to this process and not shared with other models or components within this system. The parameters $Y_{lac,glc}$ and $Y_{amm,gln}$ are also unshared parameters that should be estimated during task 2.

### E.2. Task 2: Motivation

This section describes the motivation for including task 2 in our Experiments (Section 5). The state augmentation approach uses each measurement as soon as it becomes available to correct both the predictions and parameters of a UMM (such as Equations 28). However, these characteristics of UMM in biomanufacturing, together with the use of **P**(t = 0) and **Q** with uncorrelated elements and the presence of a single measured state variable, represent a failure case that occurs when the state augmentation approach cannot estimate the unshared parameters and the state simultaneously.

For example, the application of state augmentation with EKF to estimate simultaneously the following states LAC, AMM, mAb, and the following parameters $QmAb, Y_{lac,glc}$ and $Y_{amm,gln}$ of UMM (Equations 28) fails with the conditions described in Section D.1.

The biopharmaceutical sector is increasingly focusing on emerging bioprocesses, such as the production of recombinant adeno-associated virus (rAAV), for which there is a lack of extensive prior research (Iglesias Jr et al., 2023). Enabling NKE with UMM, especially in handling the failure case detailed in Section D.1, could be vital for real-time monitoring of these novel bioprocesses. This advancement is essential in propelling the industry towards Biomanufacturing 4.0, fostering greater agility and intelligence, and thereby enhancing the quality of products, streamlining operations, and reducing expenses (Gargalo et al., 2020; Udugama et al., 2021; Herwig et al., 2021a;b). Despite its substantial market valuation of USD 239.8 billion in 2019 and an expected annual growth rate above 13%, the biopharmaceutical industry continues to face challenges in maintaining consistent productivity and quality (Luo et al., 2021).

Therefore, this is the motivation for task 2, which enables to assess if BAT can enable the tuning of states LAC, AMM, mAb, and the parameters $QmAb, Y_{lac,glc}$ and $Y_{amm,gln}$ of UMM (Equations 28) with an EKF.

### E.3. Development of Synthetic dataset for mAb Production

The procedure to generate the synthetic dataset used in this work is the same used in (Iglesias & Bolic, 2024). The training and testing sets of runs A and B were obtained by adding white Gaussian noises with different standard deviations to the main solutions (ground truth) states variables of run A and B. These main solutions were obtained by solving the unstructured mechanistic model (UMM) of mAb production presented in (Liu & Gunawan, 2017) from 0h to 103h with small variations in parameters $\mu_{max}$ (Maximum growth rate) and QmAb (mAb specific production rate) (see Table 6), but with the same initial concentrations of states variables (viable cell density (Xv), glucose (GLC), glutamine (GLN), lactate (LAC), ammonium (AMM) and mAb), see Table 7. The run A was generated using the original parameters proposed by (Liu & Gunawan, 2017), that are the parameters $\mu_{max}$ = $5.8 \times 10^{-9}(h^{-})$ and QmAb = 7.21 ($\times 10^{-9}\ mg\ cells^{-1}h^{-1}$). On the other hand, the run B has the minimum cell expansions and minimum mAb (titer) production, and they were obtained with the parameters $\mu_{max}$ = $5 \times 10^{-9}(h^{-})$ and QmAb = 4.21 ($\times 10^{-9}\ mg\ cells^{-1}h^{-1}$). Then, white Gaussian noise with different standard deviations were added to these main solutions in different ranges of time generating the training and testing sets to enable the tuning and the assessment of designed EKF, see Figure 2.

*Table 6.* Initial parameters used in UMM (Section E.1) to generate the runs A and B of Synthetic Dataset (SD).

| **Parameter** | **Name** | **run A** | **run B** |
|---|---|---|---|
| $\mu_{max}(h^{-})$ | Maximum growth rate | $5.8\times10^{-2}$ | $5\times10^{-2}$ |
| $k_{glc}(mM)$ | Monod constant glucose | $7.5\times10^{-1}$ | $7.5\times10^{-1}$ |
| $k_{gln}(mM)$ | Monod constant glutamine | $7.5\times10^{-2}$ | $7.5\times10^{-2}$ |
| $k_{Ilac}(mM)$ | Monod constant lactate for inhibition | $1.72\times10^{2}$ | $1.72\times10^{2}$ |
| $k_{Iamm}(mM)$ | Monod constant ammonium for inhibition | $2.85\times10^{1}$ | $2.85\times10^{1}$ |
| $\mu_{d,max}(h^{-})$ | Maximum death rate | $3.0\times10^{-2}$ | $3.0\times10^{-2}$ |
| $K_{d,amm}(mM)$ | Monod constant ammonium for death | 1.76 | 1.76 |
| $K_{lysis}(h^{-})$ | Breakdown of cell membranes | $5.51\times10^{-2}$ | $5.51\times10^{-2}$ |
| $Y_{X,glc}(cells\ mmol^{-})$ | Yield coefficient cell conc./glucose | $1.06\times10^{8}$ | $1.06\times10^{8}$ |
| $m_{glc}(mmol/cells\ h)$ | Glucose maintenance coefficient | $4.85\times10^{-14}$ | $4.85\times10^{-14}$ |
| $Y_{X,gln}(cells/mmmol)$ | Yield coefficient cell conc./glutamine | $5.57\times10^{8}$ | $5.57\times10^{8}$ |
| $\alpha_1(mmols\ cells^{-}\ h^{-})$ | Coefficient for $m_{gln}$ | $3.40\times10^{-13}$ | $3.40\times10^{-13}$ |
| $\alpha_2(mM)$ | Coefficient for $m_{gln}$ | 4.0 | 4.0 |
| $k_{d,gln}(h^{-})$ | Monod constant glutamine for death | $9.6\times10^{-3}$ | $9.6\times10^{-3}$ |
| $Y_{lac/glc}(1)$ | Yield coefficient lactate/glucose | 1.4 | 1.4 |
| $Y_{amm/gln}(1)$ | Yield coefficient ammonium/glutamine | $4.27\times10^{-1}$ | $4.27\times10^{-1}$ |
| $\gamma$ | constant parameter | $4.27\times10^{-1}$ | $4.27\times10^{-1}$ |
| $Q_{mAb}(mg\ cells^{-}\ h^{-})$ | mAb specific production rate | $7.21\times10^{-9}$ | $4.21\times10^{-9}$ |

*Table 7.* Initial conditions of state variables of mAb production at time 0h (begin of process).

| **State Variable** | **Name** | **Value** |
|---|---|---|
| Xv | Viable cells density | $2 \times 10^{8}$ c/mL |
| Xt | total cells density | $2 \times 10^{8}$ c/mL |
| GLC | Glucose | 29.1 mM |
| GLN | Glutamine | 4.9 mM |
| LAC | Lactate | 0 mM |
| AMM | Ammonium | 0.31 mM |
| mAb | Monoclonal Antibody (titer) | 80.6 mg/L |

### E.4. Baseline methods to auto-tuning NKEs

#### E.4.1. Objective Function with several metrics

Now lets describe the most recent Point-Estimation Approach (PEA) in the SOTA to Estimate **Q** and **R** of EKF with Objective Function, and that is used in our empirical evaluation. In (Boulkroune et al., 2023), the authors proposed a generic

approach for tuning the EKF filter based on the well known proprieties of the chi-square tests applied on NIS samples. The novelty of proposed approach is in the combination of several metrics using weighted cost function. The considered metrics are: the root mean square error (RMSE), estimation error covariance and also the mean, variance and the total number of samples that fall outside the confidence region. According to the authors, the reason of considering several metrics at the same time is to avoid local minima in the search of a suboptimal solution. Indeed, without the consistency terms there is a big risk that the optimization algorithm converges to solutions that give inconsistent filters.

Therefore, the weighted average of the combination of several metrics was chosen to formulate the objective function instead of the multi-optimization formulation. The objective is to keep the problem easy to solve and does not need any extra effort from the the filter designer. The considered objective function to solve is defined as follows:

$$min_{Q,R} = \omega_1 \mid \frac{\mu(\varepsilon_{y,k})}{m} - 1 \mid + \omega_2 \mid \frac{\sigma^2(\varepsilon_{y,k})}{m} - 1 \mid + \omega_3 \parallel trace(P) \parallel + \omega_4 RMSE(e_y) + \omega_5 \mid \frac{N_{\chi^2_m \geq \alpha}}{0.05 \times \bar{N}} - 1 \mid \quad (29)$$

with $\omega_1$ + $\omega_2$ + $\omega_3$ + $\omega_4$ + $\omega_5$ = 1; $0 < \omega_i < 1$; i = 1:5. $\omega_i$ are user defined parameters and can be changed by the user depending on the result and the expectations/preferences of the filter designer. $\mu(\varepsilon_{y,k}) \approx m$ and $\sigma^2(\varepsilon_{y,k}) \approx 2 \times m$ are respectively the mean and variance of the NIS and $m$ is the number of measured state variable. $\varepsilon_{y,k}$ represents the innovation at time k. RMSE is the abreviation of root mean square error and expressed as $RMSE(e_y) = \sqrt{\frac{\sum_{k=1}^{\bar{N}} \|e_{y,k}\|^2}{\bar{N}}}$, where $\bar{N}$ is the total number of NIS samples. $N_{\chi^2_m \geq \alpha}$ represents the number of samples that falls outside the confidence region (concentration region) where $\alpha$ defines the critical value of $\chi^2_{m,0.05}$ (can be obtained from a Chi-Square Probabilities table). This term is introduced to keep the total NIS samples falling outside the confidence region less than or around 5%.

In our experiments all baselines used the following weights: $\omega_1 = 0.3$, $\omega_2 = 0.399998$, $\omega_3 = 0.000001$ , $\omega_4 = 0.000001$ , and $\omega_5$ = 0.3.

### E.4.2. Baseline methods to optimize the objective function

**NOMAD (Nonlinear Optimization with the MADS algorithm):** NOMAD is a solver designed for blackbox optimization. It is based on the Mesh Adaptive Direct Search (MADS) algorithm, which is particularly effective for problems where the objective function is difficult to express analytically or derivatives are unavailable. NOMAD is known for its robustness in handling real-world applications with noisy and costly functions (Montoison et al., 2020).

**PS (Particle Swarm):** Particle Swarm Optimization (PSO) is a computational method that optimizes a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality. It solves problems by having a population of candidate solutions, called particles, and moving these particles around in the search-space according to simple mathematical formulae. The movements of the particles are guided by their own best known position and the best known positions in the search-space (Wang et al., 2018).

**GCMAES (Gradient-based Covariance Matrix Adaptation Evolutionary Strategy):** GCMAES is an advanced optimization algorithm that extends the traditional Covariance Matrix Adaptation Evolution Strategy (CMAES) by incorporating gradient information. This hybrid approach combines the strengths of evolutionary strategies with gradient-based optimization techniques (Dixit & Rackauckas, 2023).

**CMAES (Covariance Matrix Adaptation Evolution Strategy)** CMA-ES is a stochastic, derivative-free method for numerical optimization of non-linear or non-convex continuous optimization problems. It is well-suited for problems with noisy and discontinuous objective functions. CMA-ES adjusts the covariance matrix of a multivariate normal distribution, which is used to sample new candidate solutions. This adjustment is based on the historical performance of the optimization (Hansen, 2016).

**ECA (Evolutionary Computation Algorithm)** ECA encompasses a range of algorithms based on the principles of biological evolution, like reproduction, mutation, recombination, and selection. These algorithms are often used for optimization and search problems. They are particularly useful in domains where the search space is large and complex, and the global optimum is difficult to find using traditional methods (Tupe & Kulkarni, 2015).

**SA (Simulated Annealing)** Simulated Annealing is an optimization technique that mimics the process of annealing in metallurgy. It is known for its ability to avoid being trapped in local minima, making it effective in finding global minima in complex optimization problems. SA works by gradually reducing the ”temperature” of the system, allowing it to explore

various states and eventually settle in a state with a minimum energy configuration (Delahaye et al., 2019).

#### E.4.3. MINIMIZING THE RESIDUAL PREDICTION ERROR (RPE)

The prediction error minimization technique proposed in (Abbeel et al., 2005) simply seeks the parameters $\mathbf{R}$ and $\mathbf{Q}$ that minimize the quadratic deviation of $\mathbf{y}_k$ and the expectation above, weighted by the inverse covariance $\mathbf{M}$

$$\langle \mathbf{R}, \mathbf{Q} \rangle = \arg\min_{\mathbf{R},\,\mathbf{Q}} \sum_{k=0}^{T} (\mathbf{y}_k - h(\boldsymbol{\mu}_t))\mathbf{M}^{-1}(\mathbf{y}_k - h(\boldsymbol{\mu}_t)) \tag{30}$$

In this Equation 30:

- $\mathbf{R}$ and $\mathbf{Q}$ are the noise covariance matrices of the process and measurement respectively, which are being optimized.
- The summation runs over the time steps from $k = 0$ to $T$, where $T$ is the total number of time steps.
- $\mathbf{y}_k$ represents the additional measurements obtained during the tuning phase. These are considered as high-accuracy estimates for the state variables.
- $h(\boldsymbol{\mu}_t)$ is the projection of the state estimate $\mu_t$ at time $t$, obtained from running the Extended Kalman Filter (EKF).
- $\mathbf{M}$ is the covariance matrix of the noise associated with the additional measurements $\mathbf{y}_k$. The inverse of $\mathbf{M}$ acts as a weighting factor for the squared residuals.

The objective of this approach is to adjust the parameters $\mathbf{R}$ and $\mathbf{Q}$ so that the EKF's output (state estimates) minimizes the squared differences between the EKF estimates and the additional high-accuracy measurements $\mathbf{y}_k$ , weighted by $\mathbf{M}$. This approach directly evaluates the performance of the EKF in terms of its ability to replicate these high-accuracy measurements, therefore focusing on improving the EKF's predictive accuracy for the state variables.

If $\mathbf{M}$ is any multiple of the identity matrix, this simplifies the Equation 30 to

$$\langle \mathbf{R}, \mathbf{Q} \rangle = \arg\min_{\mathbf{R},\,\mathbf{Q}} \sum_{k=0}^{T} \|\mathbf{y}_k - h(\mu_t)\|_2^2 \tag{31}$$

Therefore, we are simply choosing the parameters R and Q that cause the filter to output the state estimates that minimize the squared differences to the measured values. The Equation 31 can be considered also an objective function. Therefore, NUTS is the MCMC method used to sample the distribution of $\boldsymbol{\Theta}$ that minimize the Equation 30 through a probabilistic model. We refer to these approaches as RPE-NUTS, see the next Section  for more details about this probabilistic model.

#### E.4.4. METHOD TO SAMPLE $\boldsymbol{\Theta}$ THAT OPTIMIZE THE OBJECTIVE FUNCTION

The No-U-Turn Sampler (NUTS) (Hoffman et al., 2014) is the MCMC method used to sample the distribution of optimal $\boldsymbol{\Theta}$ that minimize the objective function (Appendix E.4.1) through a probabilistic model.

This approach involves defining the probabilistic model with the $\boldsymbol{\Theta}$ parameters assigned as prior distributions. The objective function, computed based on $\boldsymbol{\Theta}$ and the measurements $\mathbf{y}$, is then interpreted as a surrogate for likelihood. The probabilistic model is defined as

$$\begin{aligned} &\boldsymbol{\Theta} \sim p(\boldsymbol{\Theta}) \\ &\nu = objective\_function(\boldsymbol{\Theta}, \mathbf{y}) \\ &\nu \sim Exponential(1). \end{aligned} \tag{32}$$

The $objective\_function(\boldsymbol{\Theta}, \mathbf{y})$ computes a value ($\nu$) that represents a measure of how well the $\boldsymbol{\Theta}$ are to design an EKF based on the measurements $\mathbf{y}$. Since, a optimal value of $\boldsymbol{\Theta}$ results in a value of $\nu$ close to 0, this value ($\nu$) is then assumed

to follow a Exponential(1) distribution, effectively using the objective function's output as a likelihood measure. It is important to point out that it is possible to use Truncated(Normal(0,1),0,5) distribution instead of Exponential(1). The standard deviation of 1 suggests a certain spread or variation in the objective function values around the mean 0. In the context of Bayesian inference, this probabilistic model allows the NUTS to sample (explore) different $\mathbf{\Theta}$ parameter values from the following posterior

$$P(\mathbf{\Theta}|\nu) \propto P(\nu|\mathbf{\Theta}) \times P(\mathbf{\Theta}). \tag{33}$$

This is done in the following way (Ge et al., 2018; Box & Tiao, 2011):

1. Sampling Parameters: The NUTS proposes values for $\mathbf{\Theta}$ based on priors.
2. Evaluating the Objective Function: For each set of proposed values, the objective function is computed to check how well the $\mathbf{\Theta}$ are to design an EKF based on the measurements $\mathbf{y}$.
3. Likelihood Calculation: The result of the objective function is then interpreted as a likelihood value, based on the assumption that it follows a Exponential(1) distribution. The $\mathbf{\Theta}$ values that result in an objective function value closer to zero are seen as more likely because they fit this assumed Exponential(1) distribution better.
4. Updating Beliefs: Based on the likelihood calculated for each set of $\mathbf{\Theta}$ values, the NUTS updates its belief about the distribution of the parameters $\mathbf{\Theta}$. Parameters that yield a lower value $\nu$ of the objective function (i.e., closer to the mean of the assumed Exponential(1) distribution) are considered more probable. This is because they are more consistent with the assumed likelihood model. Therefore, the NUTS sampling process will tend to favor those parameter values that minimize the objective function, reflecting the underlying assumption of the Exponential(1) distribution used in the likelihood.

### E.5. Setup of BAT and Baseline methods to auto tuning the EKF for mAb Production

BAT and baseline were executed with the same type of priors to guarantee a fair comparison between them. The Tables 10, 11, 12 show the priors used during the execution of task 1. It is important to point out that BAT-WP used a wide interval in Uniform distribution to enable us to assess the sensibility of BAT for the selected priors. Table 13 shows the priors used during the execution of task 2. The initial conditions $\mathbf{x}_0$ used during the execution of task 1 and 2 are in the Tables 7, 8, and 9. All tasks were executed with the same P(0) defined in the Table 9.

It is important to point out that the task 2 is executed using the UMM (Equations 28) with the parameters (second column in Table 6) used to generate the run A. The idea is estimate the real parameters used to generate the run B based on the measured data of run B. Therefore, we used the parameters of run A as an initial set of parameters that should be evolved.

More details about how to define priors for full covariance matrices such as $\mathbf{Q}$ can be seen in Section E.5.1.

*Table 8.* Initial conditions of state variables of mAb production at time 50h. This initial concentrations were used during the execution of task1.

| **State Variable** | **Name** | **Value** |
|---|---|---|
| Xv | Viable cells density | 1.20832e9 c/mL |
| Xt | total cells density | 1.47958e9 c/mL |
| GLC | Glucose | 14.904 mM |
| GLN | Glutamine | 0.785599 mM |
| LAC | Lactate | 19.8602 mM |
| AMM | Ammonium | 2.87289mM |
| mAb | Monoclonal Antibody (titer) | 534.628 mg/L |

*Table 9.* Initial state error covariance matrix ($\mathbf{P}$(t=0)) for BAT, BAT-WP and baselines during tasks 1 and task2.

| **Parameter** | **Name** | value |
|---|---|---|
| $\mathrm{P}_{\mathrm{X_v,X_v}}$ $(\mathrm{c}^2/\mathrm{mL}^2)$ | Viable cells | 0.0 |
| $\mathrm{P}_{\mathrm{X_t,X_t}}$ $(\mathrm{c}^2/\mathrm{mL}^2)$ | Viable cells | 0.0 |
| $\mathrm{P}_{\mathrm{GLC,GLC}}$ $(\mathrm{mM}^2)$ | Glucose | 0.0 |
| $\mathrm{P}_{\mathrm{GLN,GLN}}$ $(\mathrm{mM}^2)$ | Glutamine | 0.0 |
| $\mathrm{P}_{\mathrm{LAC,LAC}}$ $(\mathrm{mM}^2)$ | Lactate | 0.0 |
| $\mathrm{P}_{\mathrm{AMM,AMM}}$ $(\mathrm{mM}^2)$ | Ammonium | 0.0 |
| $\mathrm{P}_{\mathrm{mAb,mAb}}$ $(\mathrm{mg/L})^2$ | Monoclonal Antibody (titer) | 0.0 |
| $\mathrm{P}_{\mathrm{QmAb,QmAb}}$ $(\mathrm{g\ cells}^{-1}\mathrm{h}^{-1})^2$ | Specific production rate of mAb | 0.0 |

*Table 10.* Priors of BAT and OF-NUTS for region 1 and 2 of training set A1 for the task 1.

| PRIOR | DISTRIBUTION USED IN REG1 | DISTRIBUTION USED IN REG2 |
|---|---|---|
| $\mathrm{P}(\mathrm{R}_{X_v})$ | UNIFORM(1E7,18E7) | UNIFORM(15E7,25E7) |
| $\mathrm{P}(\mathrm{Q}_{X_v})$ | TRUNCATED(NORMAL(0., 0.01),0,1) | TRUNCATED(NORMAL(0., 0.01),0,1) |
| $\mathrm{P}(\mathrm{Q}_{X_t})$ | TRUNCATED(NORMAL(0., 0.01),0,1) | TRUNCATED(NORMAL(0., 0.01),0,1) |
| $\mathrm{P}(\mathrm{Q}_{GLC})$ | TRUNCATED(NORMAL(0., 0.01),0,1) | TRUNCATED(NORMAL(0., 0.01),0,1) |
| $\mathrm{P}(\mathrm{Q}_{GLN})$ | TRUNCATED(NORMAL(0., 0.01),0,1) | TRUNCATED(NORMAL(0., 0.01),0,1) |
| $\mathrm{P}(\mathrm{Q}_{LAC})$ | TRUNCATED(NORMAL(0., 0.01),0,1) | TRUNCATED(NORMAL(0., 0.01),0,1) |
| $\mathrm{P}(\mathrm{Q}_{AMM})$ | TRUNCATED(NORMAL(0., 0.01),0,1) | TRUNCATED(NORMAL(0., 0.01),0,1) |
| $\mathrm{P}(\mathrm{Q}_{mAb})$ | TRUNCATED(NORMAL(0., 0.01),0,1) | TRUNCATED(NORMAL(0., 0.01),0,1) |

*Table 11.* Priors of BAT-WP for region 1 and 2 of training set A1 for the task 1.

| PRIOR | DISTRIBUTION USED IN REG1 | DISTRIBUTION USED IN REG2 |
|---|---|---|
| P($R_{X_v}$) | UNIFORM(1E7,10E8) | UNIFORM(1E7,10E8) |
| P($Q_{X_v}$) | TRUNCATED(NORMAL(0.1, 0.01),0,1) | TRUNCATED(NORMAL(0.1, 0.01),0,1) |
| P($Q_{X_t}$) | TRUNCATED(NORMAL(0.1, 0.01),0,1) | TRUNCATED(NORMAL(0.1, 0.01),0,1) |
| P($Q_{GLC}$) | TRUNCATED(NORMAL(0.1, 0.01),0,1) | TRUNCATED(NORMAL(0.1, 0.01),0,1) |
| P($Q_{GLN}$) | TRUNCATED(NORMAL(0.1, 0.01),0,1) | TRUNCATED(NORMAL(0.1, 0.01),0,1) |
| P($Q_{LAC}$) | TRUNCATED(NORMAL(0.1, 0.01),0,1) | TRUNCATED(NORMAL(0.1, 0.01),0,1) |
| P($Q_{AMM}$) | TRUNCATED(NORMAL(0.1, 0.01),0,1) | TRUNCATED(NORMAL(0.1, 0.01),0,1) |
| P($Q_{mAb}$) | TRUNCATED(NORMAL(0.1, 0.01),0,1) | TRUNCATED(NORMAL(0.1, 0.01),0,1) |

*Table 12.* Priors of the numerical solvers baselines for region 1 and 2 of training set A1 for the task 1. $ub$ and $lb$ are the upper and lower bounds for box constraints in numerical solvers on the optimization variables. $ig$ is the initial guess for the optimization variables.

| | REGION 1 | | | REGION 2 | | |
|---|---|---|---|---|---|---|
| PRIOR | $ig$ | $lb$ | $ub$ | $ig$ | $lb$ | $ub$ |
| P($R_{X_v}$) | 17.5E7 | 1E7 | 1E7 | 24E7 | 15E7 | 25E7 |
| P($Q_{X_v}$) | 0.1 | 0 | 1 | 0.1 | 0 | 1 |
| P($Q_{X_t}$) | 0.1 | 0 | 1 | 0.1 | 0 | 1 |
| P($Q_{GLC}$) | 0.1 | 0 | 1 | 0.1 | 0 | 1 |
| P($Q_{GLN}$) | 0.1 | 0 | 1 | 0.1 | 0 | 1 |
| P($Q_{LAC}$) | 0.1 | 0 | 1 | 0.1 | 0 | 1 |
| P($Q_{AMM}$) | 0.1 | 0 | 1 | 0.1 | 0 | 1 |
| P($Q_{mAb}$) | 0.1 | 0 | 1 | 0.1 | 0 | 1 |

*Table 13.* Priors of BAT and OF-NUTS used in training set B1 for the task 2.

| PRIOR | DISTRIBUTION |
|---|---|
| P($R_{X_v}$) | UNIFORM(8E7,18E7) |
| P($Q_{X_v}$) | TRUNCATED(NORMAL(10E6, 10E5)) |
| P($Q_{X_t}$) | TRUNCATED(NORMAL(0., 0.01),0,1) |
| P($Q_{GLC}$) | TRUNCATED(NORMAL(0., 0.01),0,1) |
| P($Q_{GLN}$) | TRUNCATED(NORMAL(0., 0.01),0,1) |
| P($Q_{LAC}$) | TRUNCATED(NORMAL(0., 0.01),0,1) |
| P($Q_{AMM}$) | TRUNCATED(NORMAL(0., 0.01),0,1) |
| P($Q_{mAb}$) | TRUNCATED(NORMAL(0., 0.01),0,1) |
| P($Y_{lac,glc}$) | UNIFORM(1.35 , 4.0) |
| P($Y_{amm,gln}$) | UNIFORM(4.0E-1 , 20.5E-1) |
| P($\lambda$) | UNIFORM( 2.E-9, 12.21E-9) |
| P($U0_{LAC}$) | UNIFORM( 0, 10) |
| P($U0_{mAb}$) | UNIFORM( 50, 300) |

#### E.5.1. DEFINING PRIORS FOR FULL COVARIANCE MATRICES SUCH AS **Q**

There are two strategies that can be used. However the choice between them would depend on the specific requirements of the scenario, the nature of the process noise in the system, and the computational resources available. The strategies to guarantee a positive semi-definite matrix **Q** are:

- Strategy 1: **Q** with low dimensions.
- Strategy 2: **Q** with high dimensions.

**Strategy 1:**

Use the Cholesky decomposition to ensure **Q** is positive semi-definite. Represent **Q** as **Q** = $\mathbf{L}\mathbf{L}^T$, where L is a lower triangular matrix. The elements of **L** are parameterized as follows:

Suppose **Q** is a 2x2 matrix. We first define a lower triangular matrix **L**:

**L**= [$L_{1,1}$, 0 ; $L_{2,1}$, $L_{2,2}$]

The elements of **L** can be modeled as:

$L_{1,1} \sim$ truncated(Normal(0, variance), 0, upperBound)

$L_{2,1} \sim$ Normal(0, variance)

$L_{2,2} \sim$ truncated(Normal(0, variance), 0, upperBound)

**Strategy 2:**

Instead of directly modeling **Q**, we model a matrix **L** (as in Strategy 1) and obtain **Q** as **Q** = $\mathbf{L}\mathbf{L}^T$. For higher-dimensional **Q** matrices, this approach extends naturally. For a matrix **L** representing a 2x2 Q, the parameterization would remain as stated in Strategy 1. However, for higher dimensions, additional parameters are introduced following a similar pattern – diagonal elements from a truncated normal distribution and off-diagonal elements from a normal distribution.

For example,

$$L = \begin{bmatrix} L_{1,1} & 0 & 0 & \cdots & 0 \\ L_{2,1} & L_{2,2} & 0 & \cdots & 0 \\ L_{3,1} & L_{3,2} & L_{3,3} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ L_{N,1} & L_{N,2} & L_{N,3} & \cdots & L_{N,N} \end{bmatrix}$$

- The diagonal elements of L can be modeled as:

  $L_{1,1} \sim$ truncated(Normal(0, variance), 0, upperBound)

  $\vdots$

  $L_{N,N} \sim$ truncated(Normal(0, variance), 0, upperBound)

- The off-diagonal elements ($L_{offDiag}$) of L can be modeled as:

  $\alpha = 0.1$ and

  $\sigma_\alpha = sqrt(1.0/\alpha)$

  $number\_offDiag = (N^2 - N)/2$

  $L_{offDiag} \sim MvNormal(zeros(number\_offDiag), \sigma_\alpha * ones(number\_offDiag) * I)$

Both strategies involve Bayesian inference to estimate the parameters of L, after which **Q** is computed as **Q** = $\mathbf{L}\mathbf{L}^T$. By construction, this **Q** will always be positive semi-definite. This choice of $\alpha$ results in a $\sigma_\alpha$ that defines the spread of the distribution, indicating a moderate level of uncertainty about the initial values of the process noise components. Furthermore, **I** is the identity matrix, which in a n-dimensional space would be a nxn matrix with ones on the diagonal and zeros elsewhere.

## E.6. Metrics

In this section we discuss the metrics used to answer the key questions of Section 5.

### E.6.1. Filter consistency test

To evaluate the performance of designed EKF by BAT and baselines, we can use performance measures based on the innovation process (Bar-Shalom et al., 2001; Särkkä & Svensson, 2023; Reid & Term, 2001). For a properly functioning filter, the innovation sequence $e_{y,k}$ should have a zero mean and be white, with a covariance denoted as $\mathbf{S}_k$ (Bar-Shalom et al., 2001; Särkkä & Svensson, 2023; Reid & Term, 2001). The filter's consistency can be assessed by confirming that the innovation is unbiased and white, which can be done through hypothesis testing, such as the $\chi^2$ test. The Normalised Innovations Squared (NIS) is defined as $NIS_k = e_{y,k}\mathbf{S}_k^- e_{y,k}$. Then, to test unbiasedness, we need to compute the mean of Normalised Innovations Squared (NIS) as

$$\mu(NIS) = \frac{1}{N}\sum_{k=1}^{N} e_{y,k}\mathbf{S}_k^- e_{y,k} \tag{34}$$

from a single run of a NKE. Therefore, to test unbiasedness we need to verify that $\mu(NIS)$ lies in the confidence interval [r1, r2] defined by the hypothesis $H_0$ that $N \times \mu(NIS)$ is $\chi^2_{Nm}$ distributed with probability 1-$\alpha$. Thus we need to find [r1, r2] such that

$$P(N \times \mu(NIS) \in [r1, r2]|H_0) = 1 - \alpha \tag{35}$$

where $m$ is the number of measured state variables and $N$ is the number of samples from the measured state variables. Furthermore, the case of two-sided 95% confidence region, we have $[r1, r2] = [\chi^2_{Nm}(0.025), \chi^2_{Nm}(0.975)]$. More details in (Bar-Shalom et al., 2001; Särkkä & Svensson, 2023; Reid & Term, 2001)

In our experiments the unique measured state variable is Xv, but the training and testing sets have different sizes. Then, we have the following filter consistent test (FCT) based on the sets sizes and divided by tasks:

- Task 1
  - **FCT1**) Region 1 of Training set A1 with N=49: $P(N \times \mu(NIS) \in [31.555, 70.222]|H_0) = 1 - \alpha$?
  - **FCT2**) Region 2 of Training set A1 with N=54: $P(N \times \mu(NIS) \in [35.586, 76.192]|H_0) = 1 - \alpha$?
  - **FCT3**) Region 1 of Testing set A2 with N=49: $P(N \times \mu(NIS) \in [31.555, 70.222]|H_0) = 1 - \alpha$?
  - **FCT4**) Region 2 of Testing set A2 with N=54: $P(N \times \mu(NIS) \in [35.586, 76.192]|H_0) = 1 - \alpha$?
  - **FCT5**) Region 1 of Testing set A3 with N=399: $P(N \times \mu(NIS) \in [344.6, 455.1]|H_0) = 1 - \alpha$?
  - **FCT6**) Region 2 of Testing set A3 with N=425 : $P(N \times \mu(NIS) \in [368.8, 482.9]|H_0) = 1 - \alpha$?
- Task 2
  - **FCT7**) Training set B1 with N=14: $P(N \times \mu(NIS) \in [5.629, 26.1]|H_0) = 1 - \alpha$?
  - **FCT8**) Testing set B2 with N=14: $P(N \times \mu(NIS) \in [5.629, 26.1]|H_0) = 1 - \alpha$?
  - **FCT9**) Testing set B3 with N=824: $P(N \times \mu(NIS) \in [745.3, 904.39]|H_0) = 1 - \alpha$?

### E.6.2. Auto-correlation function (ACF) plot

The ACF plot is a valuable diagnostic tool in MCMC analysis, particularly for understanding the properties of the posterior distribution from which samples are drawn. It measures how well the MCMC sampler is performing by measuring the dependence of the adjacent samples (Brooks et al., 2011). For a series of samples $x_1$,$x_2$,$x_3$,...,$x_N$, the autocorrelation function can be estimated by the following equation: $\psi \approx \frac{1}{\hat{\sigma}^2_{acf}}\frac{1}{N}\sum_{i=1}^{N-s}(x_i - \hat{\mu})(x_{i+s} - \hat{\mu})$, where $\hat{\mu}$ is the average of the samples, and $\hat{\sigma}^2_{acf}$ is the sample variance. If the samples were completely independent, we would have $\psi_0$ = 1 and $\psi_s$ = 0 for s $> 0$. Generally, lower values of $\psi_s$, for s $> 0$ are indicative of better mixing within the Markov chain of the MCMC sampler, suggesting that the samples are less correlated and more representative of the target distribution.

### E.6.3. GROUND TRUTH VALUES OF $\mathbf{\Theta}_1$ AND $\mathbf{\Theta}_2$

The ground truth values of $\mathbf{\Theta}_1$ and $\mathbf{\Theta}_2$ are defined in Tables 15, 16 and 17, and they allow to design an consistent EKF with all testing sets. In Tables 18 and 19, we can see that ground truth were acceptable in all FCT defined in Section E.6.1. In addition, the N $\times\mu(NIS)$ obtained by the designed EKF with ground truth values of $\mathbf{\Theta}_1$ for FCT 1 and FCT2 were 40.5 and 49.30, respectively. The N $\times\mu(NIS)$ obtained by the designed EKF with ground truth values of $\mathbf{\Theta}_2$ for FCT 7 was 12.35.

*Table 14.* RMSPE between designed EKF (during task2) and ground truth values of the testing set B2 (sample rate of 7h).

| STATE | BAT | OF-NUTS | RPE-NUTS |
|---|---|---|---|
| XV | 9.81 % | 5.78% | 5.76 % |
| XT | 10.55 % | 7.6% | 7.57 % |
| GLC | 11.55 % | 13.03% | 12.97 % |
| GLN | 42.45% | 57.48 % | 55.12 % |
| LAC | 27.24% | 237.78% | 238.32% |
| AMM | 4.83 % | 69.31% | 68.7% |
| MAB | 4.57 % | 90.96% | 90.46 % |

## E.7. Additional Results

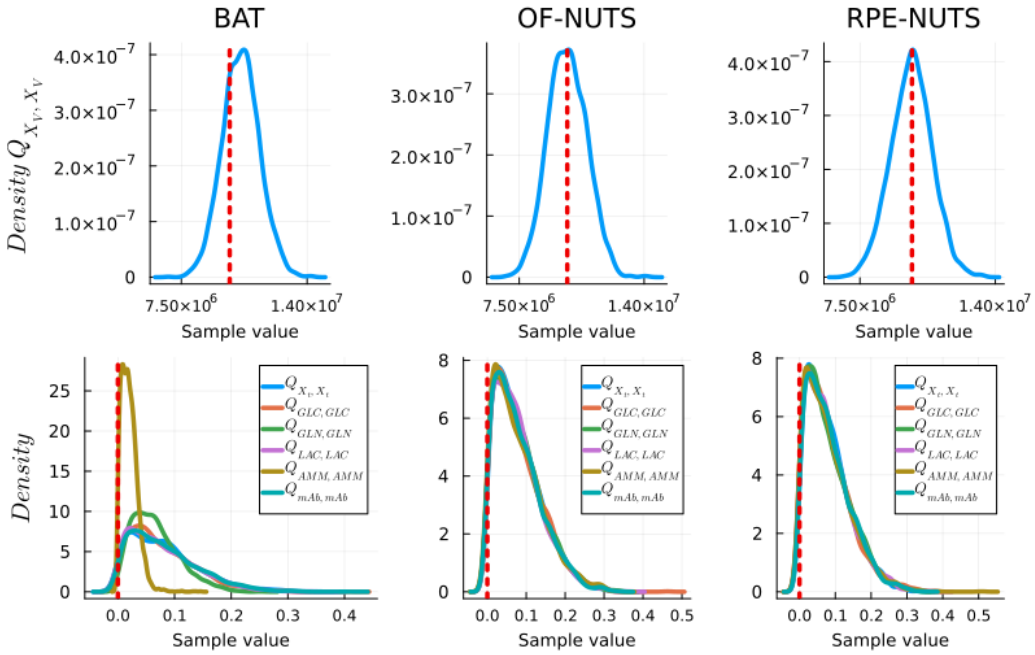


*Figure 9.* Distribution of the sampled values (for **Q** related to the seven state variables) by BAT, OF-NUTS and RPE-NUTS with training set B1. BAT, OF-NUTS and RPE-NUTS estimated unimodal distributions with peaks close to the ground truth values (red vertical line).

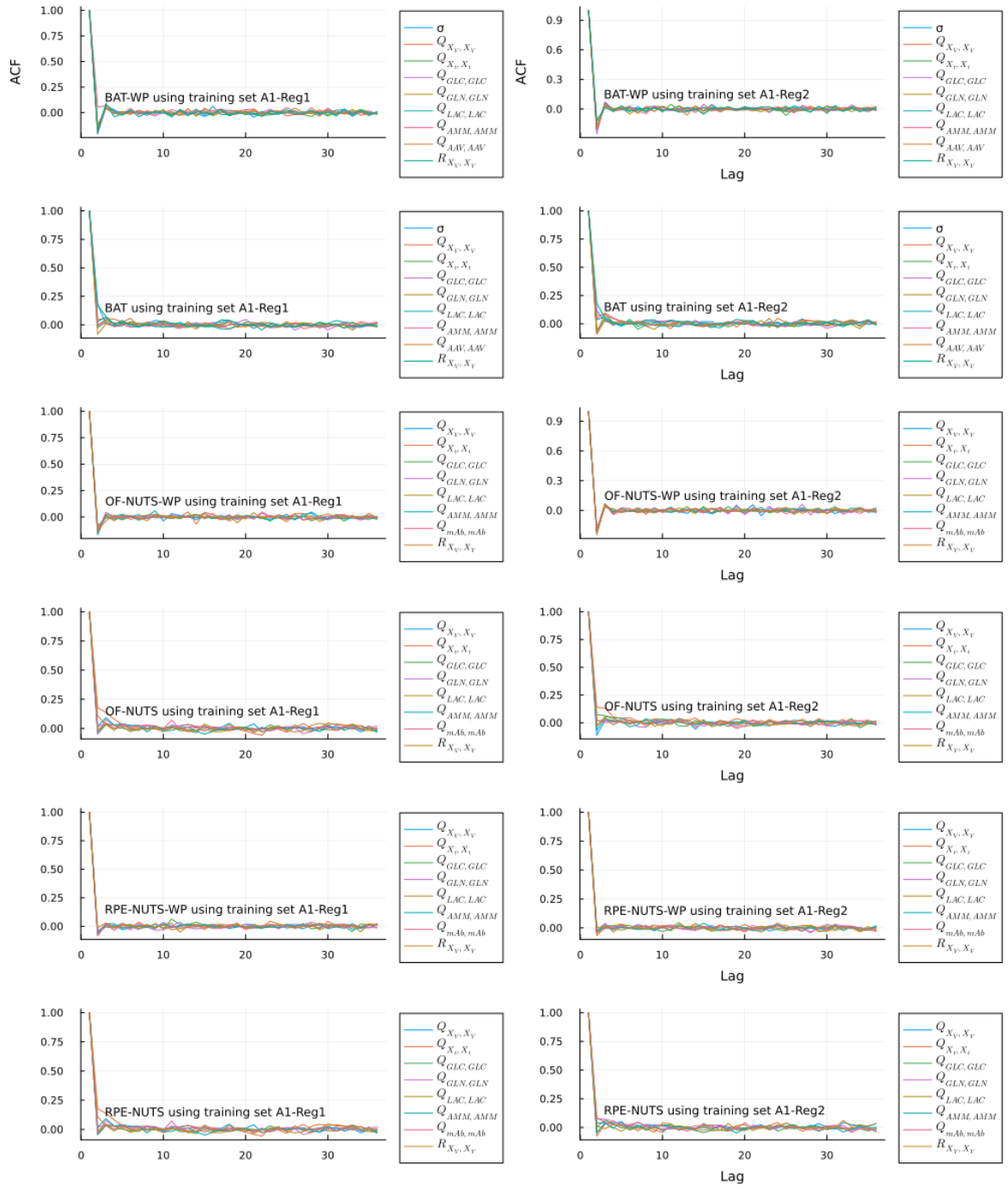


*Figure 10.* ACF plots of BAT-WP, BAT, OF-NUTS-WP, OF-NUTS, RPE-NUTS-WP and RPE-NUTS samples computed up to a lag of 30. This samples were obtained with training set A1 during the task 1. The first column shows the results related to the region 1 of training set A1, and the second column shows the results related to region 2 of training set A1. All approaches presented the same behaviour quickly declining autocorrelation. It important to point out the three points from the obtained results in ACP plots: i) Faster Convergence: A chain with quickly declining autocorrelation typically reaches convergence faster. Convergence in this context means that the chain has reached a stationary distribution that adequately represents the posterior. This is crucial for ensuring that the samples drawn from the chain can be used for reliable statistical inference. ii) Reduced Need for Thinning: When samples become uncorrelated quickly, there is less need for thinning the chain (i.e., discarding some samples to reduce autocorrelation). Thinning is often used to reduce the size of correlated data, but it can be unnecessary when the ACF declines rapidly. iii) Potential for Shorter Chains: Since each sample is more informative (less correlated with previous samples), it might be possible to achieve reliable estimates with shorter MCMC chains.

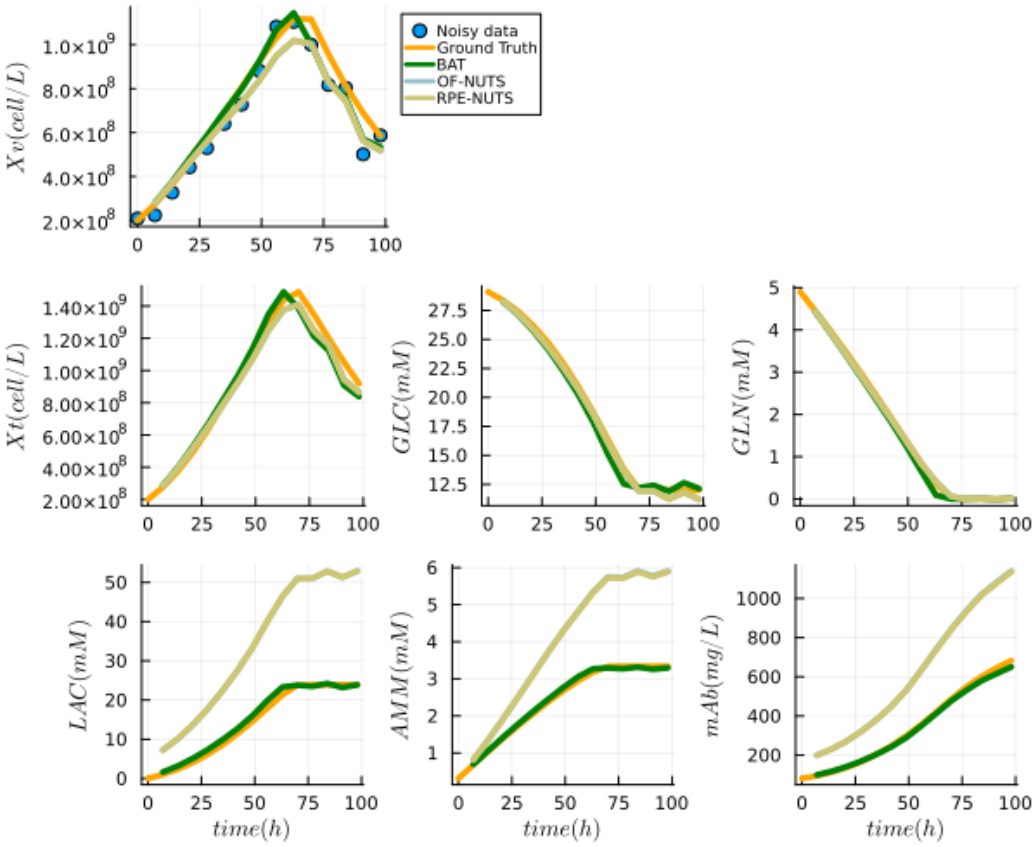


*Figure 11.* State variables estimations from the EKF designed by BAT and OF-NUTS during task 2 using noisy Xv data of testing set B2. The Table 14 shows the RMSPE between designed EKF (during task2) and ground truth values of the testing set B2 (sample rate of 7h).

*Table 15.* The $\mathbf{Q}_1$ and $\mathbf{R}_1 \in \mathbf{\Theta_1}$ estimated by BAT and baseline using training set A1 related to region 1 during the task 1. The BAT results are the mean from the obtained distribution of $\mathbf{Q}$ and $\mathbf{R}$.

| | $\mathrm{R}_{X_v}\ (c^2/mL^2)$ | $\mathrm{Q}_{X_v}(c^2/mL^2)$ | $\mathrm{Q}_{X_t}(c^2/mL^2)$ | $\mathrm{Q}_{GLC}(mM^2)$ | $\mathrm{Q}_{GLN}(mM^2)$ | $\mathrm{Q}_{LAC}(mM^2)$ | $\mathrm{Q}_{AMM}(mM^2)$ | $\mathrm{Q}_{mAb}(VG^2/mL^2)$ |
|---|---|---|---|---|---|---|---|---|
| GROUND TRUTH | $(10 \times 10^7)^2$ | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| OF-NOMAD | $(7.24 \times 10^7)^2$ | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | $(0.011)^2$ |
| OF-PS | $(7.24 \times 10^7)^2$ | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | $(0.011)^2$ |
| OF-GCMAES | $(17.5 \times 10^7)^2$ | $(-0.002)^2$ | $(-0.047)^2$ | $(0.085)^2$ | $(-0.003)^2$ | $(-0.099)^2$ | 0.0 | $(0.014)^2$ |
| OF-CMAES | $(17.5 \times 10^7)^2$ | $(-0.05)^2$ | $(0.001)^2$ | $(-0.05)^2$ | $(-0.05)^2$ | $(-0.05)^2$ | $(-0.05)^2$ | $(0.047)^2$ |
| OF-ECA | $(7.24 \times 10^7)^2$ | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| OF-SA | $(7.24 \times 10^7)^2$ | 0.0 | $(0.355)^2$ | 0.0 | 0.0 | 0.0 | 0.0 | $(0.001)^2$ |
| OF-NUTS | $(9.41 \times 10^7)^2$ | $(0.008)^2$ | $(0.008)^2$ | $(0.008)^2$ | $(0.008)^2$ | $(0.008)^2$ | $(0.008)^2$ | $(0.008)^2$ |
| BAT | $(10.2 \times 10^7)^2$ | $(0.008)^2$ | $(0.008)^2$ | $(0.008)^2$ | $(0.008)^2$ | $(0.008)^2$ | $(0.007)^2$ | $(0.008)^2$ |
| BAT-WP | $(6.92 \times 10^8)^2$ | $(0.463)^2$ | $(0.454)^2$ | $(0.455)^2$ | $(0.424)^2$ | $(0.459)^2$ | $(0.126)^2$ | $(0.454)^2$ |
| OF-NUTS-WP | $(5.07 \times 10^8)^2$ | $(0.458)^2$ | $(0.453)^2$ | $(0.457)^2$ | $(0.454)^2$ | $(0.459)^2$ | $(0.456)^2$ | $(0.462)^2$ |
| RPE-NUTS-WP | $(5.019 \times 10^8)^2$ | $(0.466)^2$ | $(0.464)^2$ | $(0.458)^2$ | $(0.459)^2$ | $(0.465)^2$ | $(0.468)^2$ | $(0.458)^2$ |

*Table 16.* The $\mathbf{Q}_2$ and $\mathbf{R}_2 \in \mathbf{\Theta_1}$ estimated by BAT and baseline using training set A1 related to region 2 during the task 1. The BAT results are the mean from the obtained distribution of $\mathbf{Q}$ and $\mathbf{R}$.

| | $\mathrm{R}_{X_v}\ (c^2/mL^2)$ | $\mathrm{Q}_{X_v}(c^2/mL^2)$ | $\mathrm{Q}_{X_t}(c^2/mL^2)$ | $\mathrm{Q}_{GLC}(mM^2)$ | $\mathrm{Q}_{GLN}(mM^2)$ | $\mathrm{Q}_{LAC}(mM^2)$ | $\mathrm{Q}_{AMM}(mM^2)$ | $\mathrm{Q}_{mAb}(VG^2/mL^2)$ |
|---|---|---|---|---|---|---|---|---|
| GROUND TRUTH | $(20 \times 10^7)^2$ | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| OF-NOMAD | $(16.8 \times 10^7)^2$ | $(0.012)^2$ | $(0.01)^2$ | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| OF-PS | $(16 \times 10^7)^2$ | $(0.012)^2$ | $(0.01)^2$ | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| OF-GCMAES | $(34 \times 10^7)^2$ | $(-0.051)^2$ | $(-0.072)^2$ | $(-0.076)^2$ | $(-0.0)^2$ | $(-0.072)^2$ | $(0.001)^2$ | $(0.006)^2$ |
| OF-CMAES | $(34 \times 10^7)^2$ | $(0.009)^2$ | $(-0.451)^2$ | $(-0.05)^2$ | $(-0.05)^2$ | $(-0.05)^2$ | $(-0.05)^2$ | $(-0.453)^2$ |
| OF-ECA | $(17.9 \times 10^7)^2$ | 0.0 | $(0.001)^2$ | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| OF-SA | $(17.9 \times 10^7)^2$ | $(0.533)^2$ | $(0.005)^2$ | 0.0 | 0.0 | 0.0 | 0.0 | $(0.737)^2$ |
| OF-NUTS | $(22.4 \times 10^7)^2$ | $(0.0078)^2$ | $(0.0081)^2$ | $(0.008)^2$ | $(0.0079)^2$ | $(0.0081)^2$ | $(0.008)^2$ | $(0.008)^2$ |
| BAT | $(22.7 \times 10^7)^2$ | $(0.008)^2$ | $(0.008)^2$ | $(0.0082)^2$ | $(0.0078)^2$ | $(0.008)^2$ | $(0.008)^2$ | $(0.0081)^2$ |
| BAT-WP | $(6.99 \times 10^8)^2$ | $(0.458)^2$ | $(0.461)^2$ | $(0.460)^2$ | $(0.105)^2$ | $(0.459)^2$ | $(0.420)^2$ | $(0.458)^2$ |
| OF-NUTS-WP | $(5.03 \times 10^8)^2$ | $(0.457)^2$ | $(0.460)^2$ | $(0.461)^2$ | $(0.465)^2$ | $(0.452)^2$ | $(0.459)^2$ | $(0.456)^2$ |
| RPE-NUTS-WP | $(4.97 \times 10^8)^2$ | $(0.462)^2$ | $(0.463)^2$ | $(0.456)^2$ | $(0.456)^2$ | $(0.462)^2$ | $(0.454)^2$ | $(0.461)^2$ |

*Table 17.* BAT and OF-NUTS estimation of $\Theta_2$ using training set B2 during the task 2.

| | GROUND | BAT | | OF-NUTS | | RPE-NUTS | |
|---|---|---|---|---|---|---|---|
| $\Theta_2$ | TRUTH | MEAN | STD | MEAN | STD | MEAN | STD |
| $R_{X_v}$ | 10E7 | 1.005E8 | 1.84E7 | 1.3E8 | 2.8E7 | 1.302E8 | 2.913E7 |
| $Q_{X_v}$ | 10E6 | 1.057E7 | 962931.9848 | 1.00E7 | 1.0E6 | 9.9740E6 | 990402.3615 |
| $Q_{X_t}$ | 0.1 | 0.0805 | 0.0605 | 0.0794 | 0.0586 | 0.0799 | 0.0588 |
| $Q_{GLC}$ | 0.1 | 0.0762 | 0.0567 | 0.0796 | 0.06 | 0.0788 | 0.0622 |
| $Q_{GLN}$ | 0.1 | 0.0657 | 0.0413 | 0.0804 | 0.0605 | 0.0797 | 0.0608 |
| $Q_{LAC}$ | 0.1 | 0.0777 | 0.0587 | 0.079 | 0.059 | 0.0799 | 0.0611 |
| $Q_{AMM}$ | 0.1 | 0.0201 | 0.0145 | 0.0793 | 0.0618 | 0.0794 | 0.0625 |
| $Q_{mAb}$ | 0.1 | 0.0807 | 0.06 | 0.0789 | 0.0593 | 0.0805 | 0.0599 |
| $Y_{lac,glc}$ | 1.399 | 1.3794 | 0.0287 | 2.6759 | 0.773 | 2.6743 | 0.775 |
| $Y_{amm,gln}$ | 0.427 | 0.4289 | 0.0227 | 1.2259 | 0.4778 | 1.2192 | 0.4759 |
| $\lambda$ | 4.21E-9 | 4.19E-9 | 1.7E-10 | 7.17E-9 | 2.9E-9 | 7.136E-9 | 2.939E-9 |
| $U0_{LAC}$ | 0 | 0.4738 | 0.4637 | 5.0168 | 2.8442 | 5.0415 | 2.8228 |
| $U0_{mAb}$ | 80.6 | 84.8377 | 14.1264 | 175.3337 | 71.2955 | 175.6891 | 69.0671 |

*Table 18.* The N $\times\mu(NIS)$ obtained by the designed EKF with BAT and baselines estimations during execution of task1. The filter consistency test (FCT) used are FCT3, FCT4, FCT5, and FCT6 defined in Section E.6.1. In blue, we have the acceptable values on the FCTs. This Table justifies the results presented in Table 1.

| | A2 | | A3 | |
|---|---|---|---|---|
| | REGION1 | REGION2 | REGION1 | REGION2 |
| METHODS | N $\times\mu(NIS)$ FOR FCT3 | N $\times\mu(NIS)$ FOR FCT4 | N $\times\mu(NIS)$ FOR FCT5 | N $\times\mu(NIS)$ FOR FCT6 |
| OF-NOMAD | 77.82 | 62.07 | 793.25 | 530.8 |
| OF-PS | 77.82 | 62.07 | 793.25 | 530.8 |
| OF-GCMAES | 13.33 | 34.64 | 135.92 | 296.20 |
| OF-CMAES | 13.14 | 29.79 | 135.18 | 428.18 |
| OF-ECA | 77.82 | 62.04 | 793.26 | 530.62 |
| OF-SA | 77.79 | 62.05 | 640.23 | 530.66 |
| OF-NUTS | 46.02 | 49.62 | 468.95 | 418.09 |
| RPE-NUTS | 45.99 | 49.81 | 468.63 | 420.01 |
| BAT | 37.96 | 49.63 | 386.77 | 418.33 |
| BAT-WP | 0.85 | 3.97 | 8.67 | 37.26 |
| OF-NUTS-WP | 1.57 | 6.76 | 16.08 | 93.30 |
| RPE-NUTS-WP | 1.601 | 6.95 | 16.45 | 95.86 |
| GROUND TRUTH | 40.85 | 49.88 | 416.39 | 426.6 |

*Table 19.* The N $\times\mu(NIS)$ obtained by the designed EKF with BAT and OF-NUTS estimations during the execution of task 2. The filter consistency test used are FCT8 and FCT9 defined in Section E.6.1. In blue, we have the acceptable values on the FCTs. This Table justifies the results presented in Table 2.

| METHODS | N $\times\mu(NIS)$ FOR FCT8 (B2) | N $\times\mu(NIS)$ FOR FCT9 (B3) |
|---|---|---|
| RPE-NUTS | 5.4127 | 467.141 |
| OF-NUTS | 5.4152 | 567.06 |
| BAT | 11.90 | 775.44 |
| GROUND TRUTH | 7.173 | 774.8 |

### E.7.1. PENDULUM CASE: NUTS (MCMC) PERFORMANCE

Aiming to check how the performance of the MCMC method is affected by the time-length (number of data points) and the size of the state, we explore the pendulum tracking with EKF presented in (Särkkä & Svensson, 2023).

The main task is estimate the EKF parameters based on the simulated data of pendulum using BAT approach and check the NUTS performance using the computation duration of NUT as metric. Therefore, we executed this task through three tests with different setups, see Table 21. The setups include: size of state variables, number of estimated parameters, number of data points (measured data used to generate innovation errors in EKF) and number of iterations for NUTS. It important to point out that to be able to check the impact of size of state variable vector, we compare the test 2 with test 4 that is the task 1 of Bioprocess monitoring problem described in Section 5.1. The Table 20 shows the results obtained by sample the BAT posterior of EKF parameters give the simulated data. All estimation are closed to the ground truth. However, in the Table 21, we can see the difference in the compute duration of NUTS obtained for each test. Comparing test 2 with test 4, we can see that increasing the size of state variables vector, cause a significant increasing in the computation duration of $\approx 3247$ sec. However, comparing test 2 with test 1, we can see that increasing the number of data points, cause a increasing in the computation duration of only $\approx 301$ sec. Therefore, we conclude that increasing in the state variable vector can cause more impact on the computation duration than increase the number of data points.

The details about the dynamic model of pendulum and simulated data can be seen in (Särkkä & Svensson, 2023). Furthermore, the code of this example is available in our anonymous repository.

It important to point out that the ACP plots of all testes presented faster convergence, see Figures 12, 13 and 14. Convergence in this context means that the chain has reached a stationary distribution that adequately represents the posterior. This is crucial for ensuring that the samples drawn from the chain can be used for reliable statistical inference.

*Table 20.* Results of sampling the BAT for EKF with simulated data of pendulum.

| | | TEST1 | | TEST 2 | | TEST 3 | |
|---|---|---|---|---|---|---|---|
| UKF COMPONENTS | GROUND TRUTH | MEAN | STD | MEAN | STD | MEAN | STD |
| $Q_{1,1}$ | 3.3E-9 | 3.2345E-9 | 9.726E-11 | 3.1345E-9 | 9.226E-11 | 3.3345E-9 | 9.2345E-11 |
| $Q_{1,2}$ | 5.0E-7 | 5.1011E-7 | 1.630E-8 | 5.0011E-7 | 1.330E-8 | 5.555E-7 | 1.243E-8 |
| $Q_{2,1}$ | 5.0E-7 | 5.0306E-7 | 1.501E-8 | 5.0006E-7 | 1.2201E-8 | 5.0456E-7 | 1.3425E-8 |
| $Q_{2,2}$ | 0.0001 | 9.9708E-5 | 9.6978E-6 | 9.0008E-5 | 9.08E-6 | 9.5656E-5 | 9.3244E-6 |
| $R$ | 0.1 | 0.0997 | 0.0102 | 0.1004 | 0.0101 | 0.10006 | 0.0102 |
| $g$ | 9.81 | 9.8111 | 0.0193 | 9.8103 | 0.0200 | 9.8101 | 0.0203 |
| $P_{1,1}$ | 1 | 1.029 | 0.5124 | 0.9962 | 0.4802 | 0.9963 | 0.4876 |
| $P_{1,2}$ | 0 | - | - | - | - | 0.9966 | 0.4903 |
| $P_{2,1}$ | 0 | - | - | - | - | 0.4036 | 0.3060 |
| $P_{2,2}$ | 1 | 0.9791 | 0.5046 | 0.9964 | 0.4991 | 0.9966 | 0.3010 |
| $\text{STATE}_1(0)$ | 0 | - | - | - | - | 0.3946 | 0.3066 |
| $\text{STATE}_2(0)$ | 0 | - | - | - | - | 0.3958 | 0.2953 |

*Table 21.* NUTS (MCMC) performance scale with time-length (number of data points) and the size of the state.

| | Pendulum problem with EKF | | | Bioprocess monitoring problem (Region 2 in Task1) |
|---|---|---|---|---|
| | Test 1 | Test 2 | Test 3 | Test 4 |
| Size of state variables | 2 | 2 | 2 | 7 |
| Number of estimated parameters | 8 | 8 | 12 | 8 |
| Number of data points | 500 | 53 | 53 | 53 |
| Number of iterations for NUTS | 3000 | 3000 | 3000 | 3000 |
| Compute Durations of NUTS | $\approx$354 sec | $\approx$53 sec | $\approx$123 sec | $\approx$ 3300 sec |

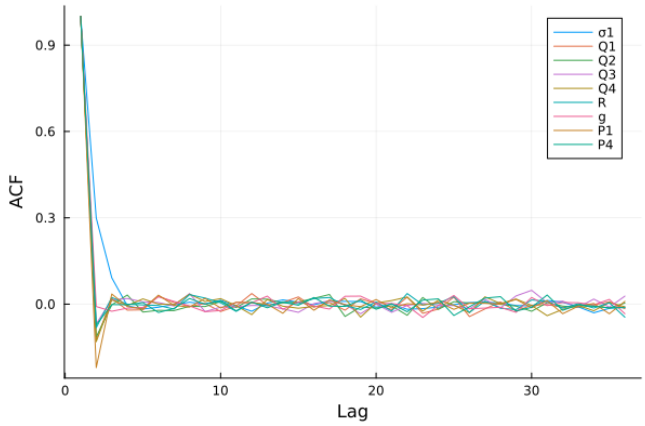


*Figure 12.* ACF plots of BAT samples computed up to a lag of 30 during test 1.

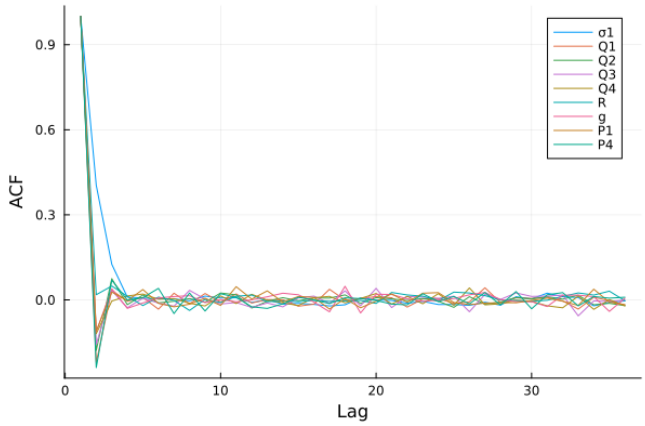


*Figure 13.* ACF plots of BAT samples computed up to a lag of 30 during test 2.

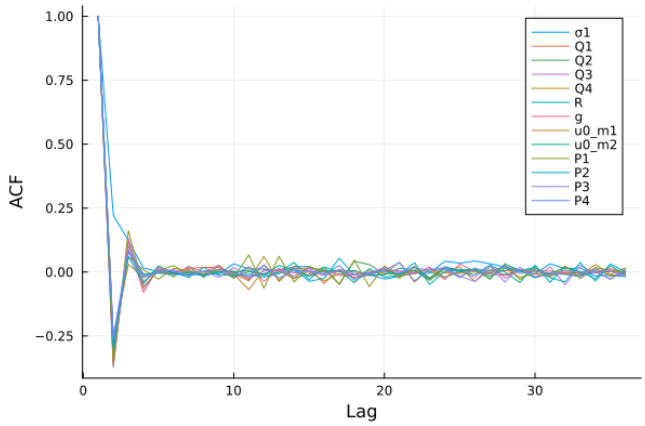


*Figure 14.* ACF plots of BAT samples computed up to a lag of 30 during test 3.