Supplementary Information for Batch Bayesian Auto-Tuning for Nonlinear Kalman Estimators

Cristovao Freitas Iglesias Jr[1,*] and Miodrag Bolic[1]

[1]School of Electrical Engineering and Computer Science (EECS), University of Ottawa, Ottawa, ON K1N 6N5, Canada
[*]cfrei096@uottawa.ca

Contents

S1 Proof of Theorem 1 2

S2 Theoretical Application of BAT to Estimate all NKEs components 3
S2.1 BAT for UKF . . . 3
S2.1.1 Pendulum case . . . 4
S2.2 BAT for CKF . . . 6
S2.2.1 Pendulum case . . . 7

S3 Related work extension 7
S3.1 Unstructured Mechanistic Models . . . 7
S3.2 State Augmentation "Failure" Case: Biomanufacturing conditions . . . 7

S4 Experimental Details 8
S4.1 Unstructured Mechanistic Model for mAb Production . . . 8
S4.2 Task Motivation . . . 9
S4.3 Development of Synthetic dataset for mAb Production . . . 10
S4.4 Baseline methods to auto-tuning NKEs . . . 10
S4.4.1 Objective Function with several metrics . . . 10
S4.4.2 Minimizing the residual prediction error (RPE) . . . 11
S4.4.3 Method to sample Θ that optimizes the objective function . . . 12
S4.5 Setup of BAT and Baseline methods to auto-tuning the EKF for mAb Production . . . 12
S4.5.1 Defining Priors for full covariance matrices such as $\mathbf{Q}$ . . . 14
S4.6 Metrics . . . 15
S4.6.1 Filter consistency test . . . 15
S4.6.2 Auto-correlation function (ACF) plot . . . 15
S4.6.3 Ground Truth values of Θ . . . 15

S5 Additional Results 16

References 19

S1 Proof of Theorem 1

Theorem 1. *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, $\hat{\mathbf{x}}_k^{NKE_{PS}} = NKE_{PS}(\mathbf{f}, \theta, t_{k-1}, t_k, \hat{\mathbf{x}}_{0,k-1} = \hat{\mathbf{x}}_{k-1}^{NKE_{US}}, \mathbf{P}_{0,k-1} = \mathbf{P}_{k-1}^{NKE_{US}}, \mathbf{Q}_{k-1}, \mathbf{R}_{k-1})$.
- $\mathbf{y}_k$ is described by a measurement model (outside of NKE update step) composed of $\hat{\mathbf{x}}_k^{NKE_{PS}}$ and additive noise as

  $$\mathbf{y}_k = h(\hat{\mathbf{x}}_k^{NKE_{PS}}) + \delta_k, \tag{1}$$

  Where $\delta_k \sim N(0, \sigma^2)$, 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

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

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

$$\begin{aligned} p(\theta, \hat{\mathbf{x}}_{0,0}, \mathbf{P}_{0,0}, \mathbf{Q}, \mathbf{R}, \sigma^2|\mathbf{y}_{online}, \mathbf{y}_{offline}) \propto p(\hat{\mathbf{x}}_{0,0}) \times p(\mathbf{P}_{0,0}) \times \\ p(\sigma^2) \times \prod_{k=1}^{T} p(\mathbf{y}_{online,k}, \mathbf{y}_{offline,k}|\theta, \hat{\mathbf{x}}_{0,0}, \mathbf{P}_{0,0}, \mathbf{Q}_{k-1}, \mathbf{R}_{k-1}, \sigma^2) \\ \times p(\mathbf{Q}_{k-1}) \times p(\mathbf{R}_{k-1}). \end{aligned} \tag{3}$$

Since, samples can be extracted from this posterior by MCMC methods and the mean of resulted distributions of $\theta, \hat{\mathbf{x}}_{0,0}, \mathbf{P}_{0,0}, \mathbf{Q}, \mathbf{R}$ 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 $\theta, \hat{\mathbf{x}}_{0,0}, \mathbf{P}_{0,0}, \mathbf{Q}, \mathbf{R}$ of any NKE.

□

S2 Theoretical Application of BAT to Estimate all NKEs components

The Figure S1 gives an overview of the sampling process 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, or even approximate the BAT posterior with variational inference.

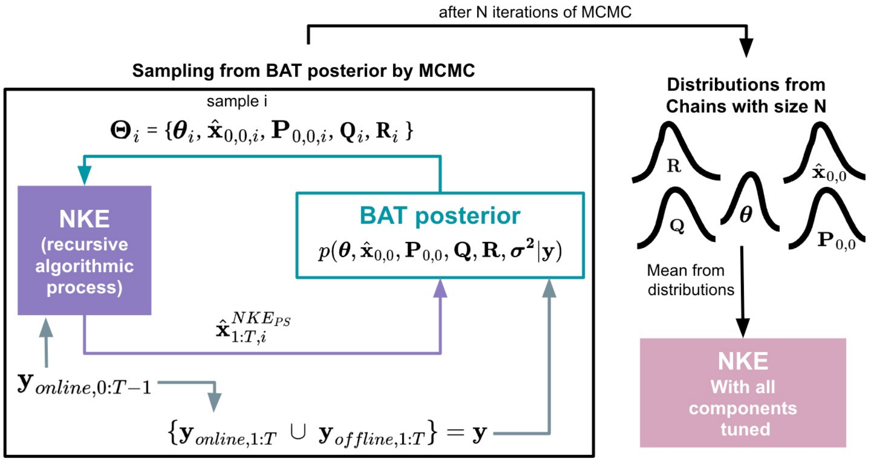
after N iterations of MCMC
Sampling from BAT posterior by MCMC
sample i
$\boldsymbol{\Theta}_i = \{\boldsymbol{\theta}_i, \hat{\mathbf{x}}_{0,0,i}, \mathbf{P}_{0,0,i}, \mathbf{Q}_i, \mathbf{R}_i\}$
NKE
(recursive algorithmic process)
BAT posterior
$p(\boldsymbol{\theta}, \hat{\mathbf{x}}_{0,0}, \mathbf{P}_{0,0}, \mathbf{Q}, \mathbf{R}, \boldsymbol{\sigma}^2|\mathbf{y})$
$\hat{\mathbf{x}}_{1:T,i}^{NKE_{PS}}$
$\mathbf{y}_{online,0:T-1}$
$\{\mathbf{y}_{online,1:T} \cup \mathbf{y}_{offline,1:T}\} = \mathbf{y}$
Distributions from Chains with size N
$\mathbf{R}$
$\hat{\mathbf{x}}_{0,0}$
$\mathbf{Q}$
$\boldsymbol{\theta}$
$\mathbf{P}_{0,0}$
Mean from distributions
NKE
With all components tuned

Figure S1. BAT: Batch Bayesian Auto-Tuning for Nonlinear Kalman Estimators. BAT posterior of $\theta, \hat{\mathbf{x}}_{0,0}, \mathbf{P}_{0,0}, \mathbf{Q}, \mathbf{R})$ given $\mathbf{y}$ is defined outside of NKE recursive algorithm that after complete T steps produce a set of predicted mean of states variables $\hat{\mathbf{x}}_{1:T,i}^{NKE_{PS}} = [\hat{\mathbf{x}}_{1,i}^{NKE_{PS}}, \ldots, \hat{\mathbf{x}}_{T,i}^{NKE_{PS}}]$. $\mathbf{y}$ 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 $\mathbf{y}_k = h(\hat{\mathbf{x}}_k^{NKE_{PS}}) + \delta_k$ (Equation 1) outside of the NKE, it is possible to define the BAT likelihood and priors to obtain the BAT posterior . 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 $\theta, \hat{\mathbf{x}}_{0,0}, \mathbf{P}_{0,0}, \mathbf{Q}, \mathbf{R}$ 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 $\mathbf{y}_{online,1:T}$.

S2.1 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 an observation model as defined in Equation 1 with the specific prediction mechanism of UKF:

$$\mathbf{y}_k = h(\hat{\mathbf{x}}_k^{UKF_{PS}}) + \delta_k, \tag{4}$$

where $\delta_k \sim N(0, \sigma^2)$ 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:

$$\langle \hat{\mathbf{x}}_k^{UKF_{PS}}, \mathbf{P}_k^{UKF_{PS}} \rangle = UKF_{PS}(\hat{\mathbf{x}}_{0,k-1} = \hat{\mathbf{x}}_{k-1}^{UKF_{US}}, \mathbf{f}, \theta, t_{k-1}, t_k, \mathbf{Q}_{k-1}, \mathbf{R}_{k-1}, \mathbf{P}_{0,k-1} = \mathbf{P}_{k-1}^{UKF_{US}}), \tag{5}$$

where:

- $\mathbf{f}$ is the nonlinear process model function (unstructured mechanistic model).
- θ 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:

$$\begin{aligned} p(\theta, \sigma^2, \hat{\mathbf{x}}_{0,set}, \mathbf{Q}, \mathbf{R}, \mathbf{P}_{0,set} | \mathbf{y}) \propto p(\theta) \times p(\sigma^2) \times \prod_{k=1}^{T} MVN(\mathbf{y}_k; h(\hat{\mathbf{x}}_k^{UKF_{PS}}), \sum(\sigma^2)) \\ \times p(\hat{\mathbf{x}}_{0,k-1}) \times p(\mathbf{P}_{0,k-1}) \times p(\mathbf{Q}_{k-1}) \times p(\mathbf{R}_{k-1}). \end{aligned} \tag{6}$$

S2.1.1 Pendulum case

Aiming to illustrate the application of BAT for UKF, we explore the Pendulum tracking with UKF presented in[1], which is a classic problem. The main idea is to estimate the UKF parameters based on the simulated data of the pendulum. Table S1 shows the results obtained by sampling the BAT posterior of UKF parameters given the simulated data. All estimations are close to the ground truth. In addition, we also estimated the hyper-parameter of α of UKF. The result of applying the designed UKF with the estimated parameters (Table S1) to the pendulum model and simulated data is shown in S2. The details about the dynamic model of the pendulum and simulated data can be seen in[1]. Furthermore, the code for this example is available in our repository https://github.com/cristovaoiglesias/BAT.

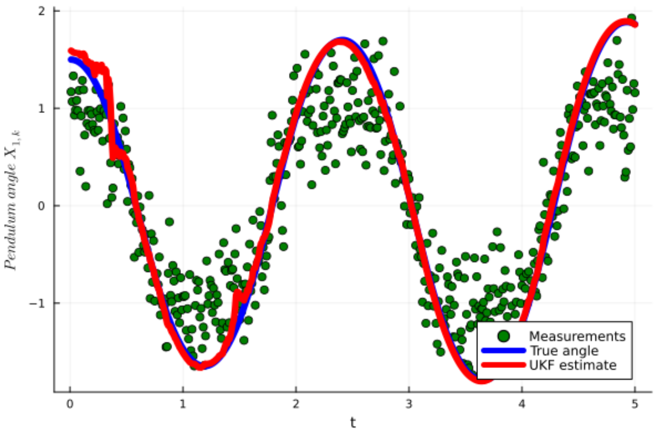

Figure S2. Simulated pendulum data and the result of tracking The pendulum using the UKF parameters estimated by BAT posterior.

Table S1. Results of sampling the BAT for UKF with simulated data of pendulum.

		BAT	
UKF COMPONENTS	GROUND TRUTH	MEAN	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
α	1	0.999	0.00983

S2.2 BAT for CKF

Again, let us consider an observation model as defined in Equation 1 with the specific prediction mechanism of CKF:

$$\mathbf{y}_k = h(\hat{\mathbf{x}}_k^{CKF_{PS}}) + \delta_k, \tag{7}$$

where $\delta_k \sim N(0,\sigma^2)$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:

$$\langle \hat{\mathbf{x}}_k^{CKF_{PS}}, \mathbf{P}_k^{CKF_{PS}} \rangle = CKF_{PS}(\hat{\mathbf{x}}_{0,k-1} = \hat{\mathbf{x}}_{k-1}^{CKF_{US}}, \mathbf{f}, \theta, t_{k-1}, t_k, \mathbf{Q}_{k-1}, \mathbf{R}_{k-1}, \mathbf{P}_{0,k-1} = \mathbf{P}_{k-1}^{CKF_{US}}), \tag{8}$$

where:

- $\mathbf{f}$ is the nonlinear process model function (unstructured mechanistic model).
- θ 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:

$$\mathbf{s}_i = \sqrt{\mathbf{P}_{k-1}}\,\mathbf{e}_i, \quad i = 1, \ldots, 2n \tag{9}$$

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:

$$\mathbf{s}'_i = \mathbf{f}(\mathbf{s}_i, \theta, t_k), \quad i = 1, \ldots, 2n \tag{10}$$

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

$$\hat{\mathbf{x}}_k^{CKF_{PS}} = \frac{1}{2n}\sum_{i=1}^{2n} \mathbf{s}'_i \tag{11}$$

$$\mathbf{P}_k^{CKF_{PS}} = \frac{1}{2n}\sum_{i=1}^{2n} (\mathbf{s}'_i - \hat{\mathbf{x}}_k^{CKF_{PS}})(\mathbf{s}'_i - \hat{\mathbf{x}}_k^{CKF_{PS}})^T + \mathbf{Q} \tag{12}$$

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:

$$\begin{aligned} p(\theta, \sigma^2, \hat{\mathbf{x}}_{0,set}, \mathbf{Q}, \mathbf{R}, \mathbf{P}_{0,set}|\mathbf{y}) \propto p(\theta) \times p(\sigma^2) \times \prod_{k=1}^{T} MVN(\mathbf{y}_k; h(\hat{\mathbf{x}}_k^{CKF_{PS}}), \textstyle\sum(\sigma^2)) \\ \times p(\hat{\mathbf{x}}_{0,k-1}) \times p(\mathbf{P}_{0,k-1}) \times p(\mathbf{Q}_{k-1}) \times p(\mathbf{R}_{k-1}). \end{aligned} \tag{13}$$

Table S2. Results of sampling the BAT for CKF with simulated data of pendulum.

		BAT	
CKF COMPONENTS	GROUND TRUTH	MEAN	STD
R	0.1	0.1009	0.0101
g	9.81	9.8097	0.0204

S2.2.1 Pendulum case

Aiming to illustrate the application of BAT for CKF, we explore the same classical problem of subsection S2.1.1. The main idea is also to estimate the CKF parameters based on the simulated data of the pendulum. Table S2 shows the results obtained by sampling the BAT posterior of CKF parameters given the simulated data. All estimation are closed to the ground truth. The details about the dynamic model of pendulum and simulated data can be seen in[1]. Furthermore, the code of this example is available in our repository `https://github.com/cristovaoiglesias/BAT`.

S3 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 S1. Contrary to the traditional auto-tuning approaches (such as[2]), we used only the output from the NKE prediction step, and we also estimated the covariance represented by σ^2 that is responsible by the additive noise $\delta_k \sim N(0,\sigma^2)$ of measurement model outside of NKE recursive loop (Equation 1).

S3.1 Unstructured Mechanistic Models

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[3–6]. 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[7–10]. 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[4, 11]. 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[12]. 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.

S3.2 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[13].

The following conditions are prevalent in biomanufacturing and should be taken into consideration while developing NKEs with state augmentation for this area because NKEs with state augmentation cannot estimate states and unshared parameters simultaneously with the following conditions[13]:

- **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 **P** (process error covariance) and **Q** (measurement error covariance) with uncorrelated elements, meaning they are diagonal with nonzero diagonal elements (noise variances) and zero off-diagonal elements.

- **ODEs of UMM with Weak Terms:** A *weak term* is a term of an ODE with a low percentage of variables of the state variable vector, and a *"strong term"* is one with a high percentage of variables of the state variable vector.
- **ODEs of UMM with Weak Variables:** A *weak variable* is a variable used only in the first member of an ODE in UMM, and a *strong variable* is a variable used in the first member and different terms of the second member of an ODE. In the case of EKF, 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 Joint EKF 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[13].

S4 Experimental Details

S4.1 Unstructured Mechanistic Model for mAb Production

BAT and baselines used the same dynamic model, which is an ODE system (Equations 14) of mAb production proposed in[14]. The ODE system (Equations 14) is a Unstructured Mechanistic Model (UMM) (description in Section S3.1) used for mAb production[14]. This system represents the cell growth, uptake of substrates, metabolism, and production process with 18 parameters described in the Table S3. 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[14].

$$
\begin{aligned}
\frac{d\,X_V}{dt} &= (\mu - \mu_d)X_V \\
\frac{d\,X_t}{dt} &= \mu X_V - k_{lysis}(X_t - X_V) \\
\mu &= \mu_{max} \cdot \frac{[GLC]}{K_{glc} + [GLC]} \cdot \frac{[GLN]}{K_{gln} + [GLN]} \cdot \frac{K_{Ilac}}{K_{Ilac} + [LAC]} \cdot \frac{K_{Iamm}}{K_{Iamm} + [AMM]} \\
\mu_d &= \frac{\mu_{d,max}}{1 + (K_{d,amm} + [AMM])^2} \\
\frac{d\,[GLC]}{dt} &= -Q_{glc} Xv \\
\frac{d\,[GLN]}{dt} &= -Q_{gln} Xv - K_{d,gln}[GLN] \\
\frac{d\,[LAC]}{dt} &= Q_{lac} Xv \\
\frac{d\,[AMM]}{dt} &= Q_{amm} Xv + K_{d,gln}[GLN] \\
Q_{glc} Xv &= \frac{\mu}{Y_{x,glc}} + m_{glc} \\
Q_{gln} Xv &= \frac{\mu}{Y_{x,gln}} + m_{gln} = \frac{\mu}{Y_{x,gln}} + \frac{\alpha_2 [GLN]}{\alpha_2 + [GLN]} \\
Q_{lac} Xv &= Y_{lac,glc} Q_{glc} \\
Q_{amm} Xv &= Y_{amm,gln} Q_{gln} \\
\frac{d\,[mAb]}{dt} &= (2 - \gamma\mu) Q_{mAb} . X_V
\end{aligned}
\tag{14}
$$

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 (μ) minus the death rate (μ_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 (μ) and Death Rate (μ_d):

   - μ: 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.

   - μ_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 γ 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 the task.

S4.2 Task Motivation

This section describes the motivation for including the task in our Experimental Setup Section. 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 14). However, these characteristics of UMM in biomanufacturing, together with the use of $\mathbf{P}(t = 0)$ and $\mathbf{Q}$ 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 14) fails with the conditions described in Section S3.2.

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[6]. Enabling NKE with UMM, especially in handling the failure case detailed in Section S3.2, 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[15–18]. 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[4].

Therefore, this is the motivation for the task used in our experiments, which enables assessing whether BAT can enable the tuning of states LAC, AMM, mAb, and the parameters $QmAb, Y_{lac,glc}$ and $Y_{amm,gln}$ of UMM (Equations 14) with an EKF.

S4.3 Development of Synthetic dataset for mAb Production

The procedure to generate the synthetic dataset used in this work is the same as used in[13]. The training and testing sets of runs SD were obtained by adding white Gaussian noises with different standard deviations to the main solutions (ground truth) state variables of run D. This main solution was obtained by solving the unstructured mechanistic model (UMM) of mAb production presented in[14] from 0h to 103h with the parameters presented in Table S3, and with the same initial concentrations of states variables (viable cell density (Xv), glucose (GLC), glutamine (GLN), lactate (LAC), ammonium (AMM) and mAb) presented in Table S4. Then, white Gaussian noises with different standard deviations were added to these main solutions in different time ranges, generating the training (D1) and testing sets (D2 and D3) to enable the tuning and assessment of the designed EKF.

Table S3. Initial parameters used in UMM (Section S4.1) to generate the main solution (ground truth) run D of Synthetic Dataset (SD).

Parameter	Name	run D
$\mu_{max}(h^{-})$	Maximum growth rate	5×10^{-2}
$k_{glc}(mM)$	Monod constant glucose	7.5×10^{-1}
$k_{gln}(mM)$	Monod constant glutamine	7.5×10^{-2}
$k_{Ilac}(mM)$	Monod constant lactate for inhibition	1.72×10^{2}
$k_{Iamm}(mM)$	Monod constant ammonium for inhibition	2.85×10^{1}
$\mu_{d,max}(h^{-})$	Maximum death rate	3.0×10^{-2}
$K_{d,amm}(mM)$	Monod constant ammonium for death	1.76
$K_{lysis}(h^{-})$	Breakdown of cell membranes	5.51×10^{-2}
$Y_{X,glc}(cells\ mmol^{-})$	Yield coefficient cell conc./glucose	1.06×10^{8}
$m_{glc}(mmol/cells\ h)$	Glucose maintenance coefficient	4.85×10^{-14}
$Y_{X,gln}(cells/mmmol)$	Yield coefficient cell conc./glutamine	5.57×10^{8}
$\alpha_1(mmols\ cells^{-}\ h^{-})$	Coefficient for m_{gln}	3.40×10^{-13}
$\alpha_2(mM)$	Coefficient for m_{gln}	4.0
$k_{d,gln}(h^{-})$	Monod constant glutamine for death	9.6×10^{-3}
$Y_{lac/glc}(1)$	Yield coefficient lactate/glucose	1.4
$Y_{amm/gln}(1)$	Yield coefficient ammonium/glutamine	4.27×10^{-1}
γ	constant parameter	4.27×10^{-1}
$Q_{mAb}(mg\ cells^{-}\ h^{-})$	mAb specific production rate	4.21×10^{-9}

Table S4. Initial conditions of state variables of mAb production at time 0h (begin of process).

State Variable	Name	Value
Xv	Viable cells density	2×10^{8} c/mL
Xt	total cells density	2×10^{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

S4.4 Baseline methods to auto-tuning NKEs

S4.4.1 Objective Function with several metrics

The objective function proposed in[19] is the most recent approach in the SOTA to Estimate **Q** and **R** of EKF with Objective Function, and is used in our empirical evaluation. In[19], the authors proposed a generic approach for tuning the EKF filter based on the well-known proprieties of the chi-square tests applied to NIS samples. The novelty of the proposed approach is in the combination of several metrics using a 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 for considering several metrics at the same time is to avoid local minima in the search for a suboptimal solution. Indeed, without the consistency terms, there is a risk that the optimization algorithm will converge 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 filter designer. The considered objective function to solve is defined as follows:

$$min_{Q,R} = \omega_1 \mid \frac{\mu(\varepsilon_{y,k})}{m} - 1 \mid + \omega_2 \mid \frac{\sigma^2(\varepsilon_{y,k})}{m} - 1 \mid + \omega_3 \parallel trace(P) \parallel + \omega_4 RMSE(e_y) + \omega_5 \mid \frac{N_{\chi^2_m \geq \alpha}}{0.05 \times \bar{N}} - 1 \mid \tag{15}$$

with $\omega_1 + \omega_2 + \omega_3 + \omega_4 + \omega_5 = 1$; $0 < \omega_i < 1$; i = 1:5. ω_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 abbreviation of root mean square error and is 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 fall outside the confidence region (concentration region) where α 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 $\omega_5 = 0.3$.

S4.4.2 Minimizing the residual prediction error (RPE)

The prediction error minimization technique proposed in[2] simply seeks the parameters **R** and **Q** that minimize the quadratic deviation of $\mathbf{y}_k$ and the expectation above, weighted by the inverse covariance **M**

$$\langle \mathbf{R}, \mathbf{Q} \rangle = \arg\min_{\mathbf{R},\,\mathbf{Q}} \sum_{k=0}^{T} (\mathbf{y}_k - h(\mu_t))\mathbf{M}^{-1}(\mathbf{y}_k - h(\mu_t)) \tag{16}$$

In this Equation 16:

- **R** and **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.
- $h(\mu_t)$ is the projection of the state estimate μ_t at time t, obtained from running the Extended Kalman Filter (EKF).
- **M** is the covariance matrix of the noise associated with the additional measurements $\mathbf{y}_k$. The inverse of **M** acts as a weighting factor for the squared residuals.

The objective of this approach is to adjust the parameters **R** and **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 **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 **M** is any multiple of the identity matrix, this simplifies the Equation 16 to

$$\langle \mathbf{R}, \mathbf{Q} \rangle = \arg\min_{\mathbf{R},\,\mathbf{Q}} \sum_{k=0}^{T} \|\mathbf{y}_k - h(\mu_t)\|_2^2 \tag{17}$$

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 17 can be considered also an objective function. Therefore, NUTS is the MCMC method used to sample the distribution of Θ that minimize the Equation 16 through a probabilistic model. We refer to these approaches as RPE-NUTS, see the next Section for more details about this probabilistic model.

S4.4.3 Method to sample Θ that optimizes the objective function

The No-U-Turn Sampler (NUTS)[20] is the MCMC method used to sample the distribution of optimal Θ that minimizes the objective function (Section S4.4.1) and RPE (Section S4.4.2) through a probabilistic model.

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

$$\begin{aligned}
&\Theta \sim p(\Theta) \\
&v = objective\_function(\Theta, \mathbf{y}) \\
&v \sim Exponential(1).
\end{aligned} \tag{18}$$

The $objective\_function(\Theta, \mathbf{y})$ computes a value (v) that represents a measure of how well the Θ is to design an EKF based on the measurements $\mathbf{y}$. Since an optimal value of Θ results in a value of v close to 0, this value (v) is then assumed to follow an Exponential(1) distribution, effectively using the output of the objective function 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 Θ parameter values from the following posterior,

$$P(\Theta|v) \propto P(v|\Theta) \times P(\Theta). \tag{19}$$

This is done in the following way[21,22]:

1. Sampling Parameters: The NUTS proposes values for Θ based on priors.

2. Evaluating the Objective Function: For each set of proposed values, the objective function is computed to check how well the Θ is 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 an Exponential(1) distribution. The Θ 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 Θ values, the NUTS updates its belief about the distribution of the parameters Θ. Parameters that yield a lower value v 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.

S4.5 Setup of BAT and Baseline methods to auto-tuning the EKF for mAb Production

BAT and baselines were executed with the same priors to guarantee a fair comparison between them. Table S6 shows the priors used during the execution of the empirical task. The priors in Table S6 are divided into two groups: more informative priors using normal distribution and less informative priors using uniform distribution. These two groups were used to assess the performances of BAT and baselines. The "more informative priors" were used to define the priors of $\mathbf{Q}$, and the "Less Informative Priors" were used to define the priors of $\mathbf{R}$, $\mathbf{x}_{0,0}$=$\{$u0$_{LAC}$, u0$_{mAb}\}$ that represent the initial concentrations of LAC and mAb, and θ =$\{$Y$_{lac,glc}$, Y$_{amm,gln}$, $\lambda\}$ that represent three unshared parameters of the UMM (see Equation S 14). Furthermore, the priors used to model the variances σ^2 of additive noise δ_k present in the measurement model outside of the recursive NKE process (in BAT approach) can be seen in Table S7. The initial conditions $\mathbf{x}_0$ and $\mathbf{P}(0)$ used during the execution of the task are in the Tables S4, and S5. It is important to point out that the empirical task is executed using the UMM (Equations 14), but the idea is to estimate the parameters θ =$\{$Y$_{lac,glc}$, Y$_{amm,gln}$, $\lambda\}$ used to generate the main solution (run D) based on the measured data of run D1, besides to estimate the other components of Θ_2, such as $\{\mathbf{x}_{0,0}$, $\mathbf{Q}$, $\mathbf{R}$. $\}$ Furthermore, we used $\mathbf{Q}$ and $\mathbf{P}$ with uncorrelated elements (off-diagonal elements defined as zero) due to the biomanufacturing conditions described in Section S3.2. However, more details about how to define priors for full covariance matrices such as $\mathbf{Q}$ can be seen in Section S4.5.1.

Table S5. Initial state error covariance matrix (**P**(t=0)) for BAT and baselines during the empirical task.

Parameter	**Name**	value
$\mathrm{P}_{\mathrm{X_v,X_v}}$ $(\mathrm{c^2/mL^2})$	Viable cells	0.0
$\mathrm{P}_{\mathrm{X_t,X_t}}$ $(\mathrm{c^2/mL^2})$	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
$\mathrm{P}_{\mathrm{QmAb,QmAb}}$ $(\mathrm{g\ cells^{-1}h^{-1}})^2$	Specific production rate of mAb	0.0

Table S6. Priors of BAT, OF-NUTS, and RPE-NUTS used in the first step of the empirical task with the training set D1.

Prior	Distribution
P(R_{X_v})	Uniform(9.5e7,18e7)
P(Q_{X_v})	truncated(Normal(10e6, 10e4),0,10e10)
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($\mathrm{Y}_{lac,glc}$)	Uniform(1.35 , 4.0)
P($\mathrm{Y}_{amm,gln}$)	Uniform(4.0e-1 , 20.5e-1)
P(λ)	Uniform(2.e-9, 12.21e-9)
P($\mathrm{u0}_{LAC}$)	Uniform(0, 10)
P($\mathrm{u0}_{mAb}$)	Uniform(50, 300)

Table S7. Priors of σ^2 related to each state variable used by BAT in the first step of the empirical task with the training set D1.

Prior	Distribution
P($\sigma^2_{X_v}$)	InverseGamma(2, 3)
P($\sigma^2_{X_t}$)	InverseGamma(2, 3)
P(σ^2_{GLC})	InverseGamma(2, 3)
P(σ^2_{GLN})	InverseGamma(2, 3)
P(σ^2_{LAC})	InverseGamma(2, 3)
P(σ^2_{AMM})	InverseGamma(2, 3)
P(σ^2_{mAb})	InverseGamma(2, 3)

S4.5.1 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 $\mathbf{Q} = \mathbf{L}\mathbf{L}^T$, 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:

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

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

$L_{2,2}$ ~ 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 $\mathbf{Q} = \mathbf{L}\mathbf{L}^T$. 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:

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

  $\vdots$

  $L_{N,N}$ ~ 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 $\mathbf{Q} = \mathbf{L}\mathbf{L}^T$. By construction, this **Q** will always be positive semi-definite. This choice of α results in a σ_α 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 n x n matrix with ones on the diagonal and zeros elsewhere.

S4.6 Metrics

In this section, we discuss the metrics used to answer the key questions of Section **??**.

S4.6.1 Filter consistency test

To evaluate the performance of designed EKF by BAT and baselines, we can use performance measures based on the innovation process[1,23,24]. For a properly functioning filter, the innovation sequence $e_{y,k}$ should have a zero mean and be white, with a covariance denoted as $\mathbf{S}_k$[1,23,24]. 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 χ^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

$$\mu(NIS) = \frac{1}{N}\sum_{k=1}^{N} e_{y,k}\mathbf{S}_k^{-}e_{y,k} \tag{20}$$

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 χ^2_{Nm} distributed with probability 1-α. Thus we need to find [r1, r2] such that

$$P(N \times \mu(NIS) \in [r1, r2] | H_0) = 1 - \alpha \tag{21}$$

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[1,23,24]

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:

- Empirical task
  - **FCT1**) Training set D1 with N=14: $P(N \times \mu(NIS) \in [5.629, 26.1] | H_0) = 1 - \alpha$?
  - **FCT2**) Testing set D2 with N=14: $P(N \times \mu(NIS) \in [5.629, 26.1] | H_0) = 1 - \alpha$?
  - **FCT3**) Testing set D3 with N=824: $P(N \times \mu(NIS) \in [745.3, 904.39] | H_0) = 1 - \alpha$?

S4.6.2 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[25]. For a series of samples $x_1, x_2, x_3, \ldots, x_N$, 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 ψ_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.

S4.6.3 Ground Truth values of Θ

The ground truth values of Θ are defined in Table S9. They allow the design of a consistent EKF with all testing sets. In Table S11, we can see that ground truth values were acceptable in FCT2 and FCT3 defined in Section S4.6.1. In addition, the N $\times \mu(NIS)$ obtained by the designed EKF with ground truth values of Θ for FCT1 was 12.35 (acceptable).

S5 Additional Results

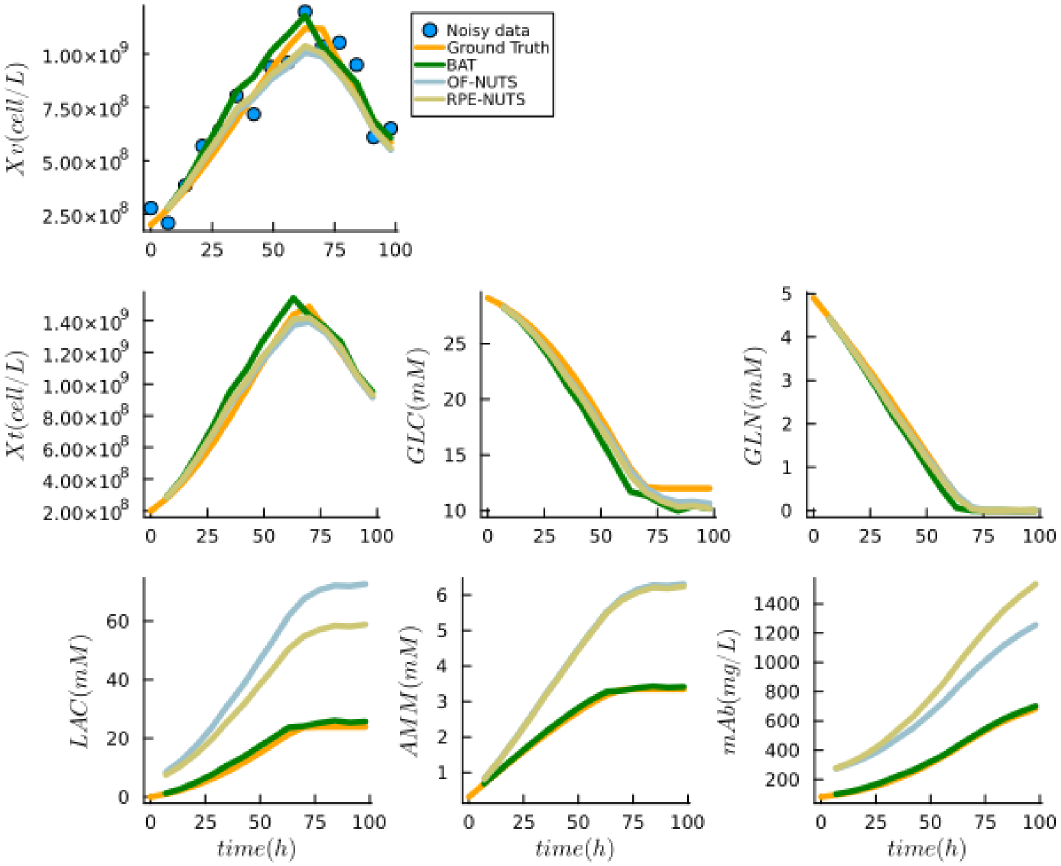

Figure S3. State variables estimations from the EKF designed by BAT, RPE-NUTS, and OF-NUTS during task using noisy Xv data of testing set D2. Table S8 shows the RMSPE between designed EKF (during empirical task) and ground truth values of the testing set D2 (sample rate of 7h). RPE-NUTS and OF-NUTS had similar performance with the worst estimation about LAC, AMM, and mAb.

Table S8. RMSPE between designed EKF (during empirical task) and ground truth values of the testing set D2 (sample rate of 7h).

State	BAT	OF-NUTS	RPE-NUTS
Xv	9.9 %	6.56%	5.96 %
Xt	10.57%	5.32%	5.89 %
GLC	10.37 %	5.58%	7.69 %
GLN	9.02%	2.52 %	3.13 %
LAC	12.03%	199.45%	152.39%
AMM	2.75 %	66.79%	65.18%
mAb	5.39 %	124.31%	151.2 %

Table S9. Mean and standard deviation of the distribution estimated by BAT, RPE-NUTS and OF-NUTS of Θ using training set D2 during the task.

	Ground	BAT		OF-NUTS		RPE-NUTS	
Θ	Truth	mean	std	mean	std	mean	std
R_{X_v}	10E7	1.14E8	1.68E7	1.37E8	2.46E7	1.377E8	2.464E7
Q_{X_v}	10E6	1.06E7	948902.5	9.99E7	1.01E6	9.9847E6	999325.3
Q_{X_t}	0.1	0.0773	0.0598	0.0816	0.0599	0.0798	0.0589
Q_{GLC}	0.1	0.0764	0.0595	0.0777	0.0609	0.0791	0.0618
Q_{GLN}	0.1	0.0707	0.0441	0.0805	0.0608	0.0801	0.0578
Q_{LAC}	0.1	0.0767	0.0594	0.0792	0.0608	0.0802	0.0607
Q_{AMM}	0.1	0.0214	0.0149	0.0793	0.0596	0.0802	0.0608
Q_{mAb}	0.1	0.0798	0.06	0.0803	0.0599	0.0795	0.0594
$Y_{lac,glc}$	1.399	1.3798	0.0286	2.68	0.7688	2.6899	0.7514
$Y_{amm,gln}$	0.427	0.43	0.0232	1.2131	0.4726	1.2321	0.4698
λ	4.21E-9	4.18E-9	1.68E-10	7.09E-9	2.91E-9	7.05E-9	2.971E-9
$u0_{LAC}$	0	0.4894	0.4664	5.0664	2.8513	4.9268	2.8585
$u0_{mAb}$	80.6	85.2374	13.6992	175.2764	73.0186	174.8894	71.4892

Table S10. Mean and variance of the sigma priors (related to each state variable) estimated by BAT in the first step of the empirical task with the training set D1. Here, the ground truth is the standard deviations (std) of 10×10^7, 10×10^7, 1, 0.5, 2.0, 0.1, and 40.5 used by the Gaussian white noises added to the main solution D of state variables Xv, Xt, GLC, GLN, LAC, AMM and mAb.

Prior	ground truth	mean	Variance
$P(\sigma^2_{X_v})$	$(10\times10^7)^2$	109117590.7028^2	21910340.6253^2
$P(\sigma^2_{X_t})$	$(10\times10^7)^2$	184181377.1960^2	35374028.5638^2
$P(\sigma^2_{GLC})$	1^2	1.8434^2	0.3820^2
$P(\sigma^2_{GLN})$	0.5^2	$0.3843\ ^2$	0.0816^2
$P(\sigma^2_{LAC})$	2.0^2	2.7837^2	0.7180^2
$P(\sigma^2_{AMM})$	0.1^2	0.2730^2	0.0760^2
$P(\sigma^2_{mAb})$	40.5^2	30.6076^2	6.4392^2

Table S11. The N $\times\mu(NIS)$ obtained by the designed EKF with BAT and OF-NUTS estimations during the execution of task. The filter consistency test used are FCT2 and FCT3 defined in Section S4.6.1. In blue, we have the acceptable values on the FCTs.

METHODS	N $\times\mu(NIS)$ FOR FCT2 (D2)	N $\times\mu(NIS)$ FOR FCT3 (D3)
RPE-NUTS	9.083	418.61
OF-NUTS	8.11	418.50
BAT	13.71	786.68
GROUND TRUTH	7.173	774.8

References

1. Särkkä, S. & Svensson, L. *Bayesian filtering and smoothing*, vol. 17 (Cambridge university press, 2023).
2. Abbeel, P. *et al.* Discriminative training of kalman filters. In *Robotics: Science and systems*, vol. 2, 1 (2005).
3. Kyriakopoulos, S. *et al.* Kinetic modeling of mammalian cell culture bioprocessing: the quest to advance biomanufacturing. *Biotechnol. J.* **13**, 1700229 (2018).
4. Luo, Y., Kurian, V. & Ogunnaike, B. A. Bioprocess systems analysis, modeling, estimation, and control. *Curr. Opin. Chem. Eng.* **33**, 100705 (2021).
5. Iglesias Jr, C. F. *et al.* Monitoring the recombinant adeno-associated virus production using extended kalman filter. *Processes* **10**, 2180 (2022).
6. Iglesias Jr, C. F., Ristovski, M., Bolic, M. & Cuperlovic-Culf, M. raav manufacturing: The challenges of soft sensing during upstream processing. *Bioengineering* **10**, 229 (2023).
7. Moser, A., Appl, C., Brüning, S. & Hass, V. C. Mechanistic mathematical models as a basis for digital twins. In *Digital Twins*, 133–180 (Springer, 2020).
8. Park, S.-Y., Park, C.-H., Choi, D.-H., Hong, J. K. & Lee, D.-Y. Bioprocess digital twins of mammalian cell culture for advanced biomanufacturing. *Curr. Opin. Chem. Eng.* **33**, 100702 (2021).
9. Mears, L., Stocks, S. M., Albaek, M. O., Sin, G. & Gernaey, K. V. Mechanistic fermentation models for process design, monitoring, and control. *Trends biotechnology* **35**, 914–924 (2017).
10. Reyes, S. J., Durocher, Y., Pham, P. L. & Henry, O. Modern sensor tools and techniques for monitoring, controlling, and improving cell culture processes. *Processes* **10**, 189 (2022).
11. Tsopanoglou, A. & 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. *Curr. Opin. Chem. Eng.* **32**, 100691 (2021).
12. Zhang, D., Del Rio-Chanona, E. A., Petsagkourakis, P. & Wagner, J. Hybrid physics-based and data-driven modeling for bioprocess online simulation and optimization. *Biotechnol. bioengineering* **116**, 2919–2930 (2019).
13. Iglesias, C. F. & Bolic, M. How not to make the joint extended kalman filter fail with unstructured mechanistic models. *Sensors* **24**, DOI: 10.3390/s24020653 (2024).
14. Liu, Y. & Gunawan, R. Bioprocess optimization under uncertainty using ensemble modeling. *J. biotechnology* **244**, 34–44 (2017).
15. Gargalo, C. L. *et al.* Towards the development of digital twins for the bio-manufacturing industry. In *Digital Twins*, 1–34 (Springer, 2020).
16. Udugama, I. A. *et al.* Towards digitalization in bio-manufacturing operations: A survey on application of big data and digital twin concepts in denmark. *Front. Chem. Eng.* **3**, 727152 (2021).
17. Herwig, C., Pörtner, R. & Möller, J. *Digital Twins: tools and concepts for smart biomanufacturing* (Springer, 2021).
18. Herwig, C., Pörtner, R. & Möller, J. *Digital Twins: Applications to the Design and Optimization of Bioprocesses* (Springer, 2021).
19. Boulkroune, B., Geebelen, K., Wan, J. & van Nunen, E. Auto-tuning extended kalman filters to improve state estimation. In *2023 IEEE Intelligent Vehicles Symposium (IV)*, 1–6 (IEEE, 2023).
20. 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**, 1593–1623 (2014).
21. Ge, H., Xu, K. & Ghahramani, Z. Turing: a language for flexible probabilistic inference. In *International conference on artificial intelligence and statistics*, 1682–1690 (PMLR, 2018).
22. Box, G. E. & Tiao, G. C. *Bayesian inference in statistical analysis* (John Wiley & Sons, 2011).
23. Bar-Shalom, Y., Li, X. R. & Kirubarajan, T. *Estimation with applications to tracking and navigation: theory algorithms and software* (John Wiley & Sons, 2001).
24. Reid, I. & Term, H. Estimation ii discrete-time kalman filter. *Hilar. Term* 1–44 (2001).
25. Brooks, S., Gelman, A., Jones, G. & Meng, X.-L. *Handbook of markov chain monte carlo* (CRC press, 2011).