Nonlinear Kalman Estimators for Low-Cost Bioprocess Monitoring with Unstructured Mechanistic Models

by

Cristovão Freitas Iglesias Junior

Thesis submitted to the University of Ottawa
in partial fulfillment of the requirement for the degree of
Doctor of Philosophy in
Computer Science

School of Electrical Engineering and Computer Science
Faculty of Engineering
University of Ottawa

Examining Committee

The following served on the Examining Committee for this thesis.

External Examiner:	Dr. Jesus Garcia Herrero Professor Department of Computer Science and Engineering Charles III University of Madrid
Internal Member:	Dr. Martin Bouchard Professor School of Electrical Engineering and Computer Science University of Ottawa
Internal Member:	Dr. Theodore Perkins Professor Department of Biochemistry, Microbiology and Immunology University of Ottawa
Carleton Member:	Dr. James R. Green Professor Department of Systems and Computer Engineering Carleton University
Supervisor:	Dr. Miodrag Bolic Professor School of Electrical Engineering and Computer Science University of Ottawa

Declaration of Authorship

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners. I understand that my thesis may be made electronically available to the public.

Abstract

Bioprocess monitoring is a critical component in the biopharmaceutical industry, essential for ensuring the consistent production of high-quality biopharmaceutical products. Despite significant advancements in bioprocessing technologies, monitoring techniques have not kept pace, leading to inefficiencies and increased production costs. Therefore, there is constant pressure for bioprocess monitoring strategies that are efficient and low-cost. The literature indicates that soft sensors based on nonlinear Kalman estimators (NKE) with unstructured mechanistic models (UMM) can enable fast and low-cost bioprocess monitoring. However, classic NKEs with UMM are limited to fast and low-cost bioprocess monitoring, and improvements are needed to handle different biomanufacturing conditions. Therefore, the main goal of this thesis is to enable the design and development of NKEs with UMM for fast and low-cost monitoring of highly nonlinear bioprocesses, such as viral vectors and monoclonal antibody productions in different biomanufacturing conditions. The thesis is structured around three main research questions. The first investigates how joint estimation of states and unshared parameters by NKE with specific UMM can be achieved under biomanufacturing conditions for efficient monitoring. The proposed solution, called SANTO (Specific initiAl coNdiTiOn), enhances the performance of joint NKE by preventing the Kalman gain from being zero throughout the process. The second research question addresses the automatic tuning of all NKE components for new biomanufacturing conditions. The Batch Bayesian Auto-Tuning (BAT) approach is introduced, leveraging all available measured data to define a posterior distribution of NKE components, thereby enhancing the consistency and accuracy of the estimators. The third research question explores increasing the adaptability (generalization), simplifying the tuning process, and improving real-time parameter estimation of NKEs with generic UMM for highly nonlinear bioprocesses. The Hybrid Nonlinear Kalman Estimator (HNKE) is proposed as a novel hybrid Gaussian filter integrating a hybrid dynamic model (containing uncertainty quantification) and unscented transformation or cubature rule. This novel method allows for auto-initialization of state error covariance and real-time estimation of parameters and significantly reduces the data and tuning requirements. Empirical evaluations using bioprocess datasets related to different real-world biomanufacturing conditions validate the effectiveness of these approaches, demonstrating significant improvements in monitoring accuracy and efficiency. The findings contribute to advancing bioprocess monitoring towards Biomanufacturing 4.0, optimizing operations, and reducing production costs. This research provides a foundation for future developments in bioprocess monitoring, with potential applications in various biomanufacturing scenarios.

Acknowledgements

Reflecting on this academic journey, I feel compelled to extend my sincere gratitude to those whose true support and insightful guidance have been instrumental in the successful completion of this thesis. First and foremost, I wish to express my sincere gratitude to my esteemed supervisor, Professor Miodrag Bolic. Your invaluable mentorship, expertise, and unwavering encouragement have not only shaped my academic journey but have also profoundly impacted my personal growth. I feel genuinely privileged to have had the opportunity to work under your guidance and am profoundly thankful for the knowledge and skills I have acquired through this enriching experience. I also extend my heartfelt thanks to my good friends who stood by me during this journey: Mohamad Hosein Davoodabadi Farahani, Shan He, Fardad Dadboud, Zixiong Han, Hamid Asad, Varunkumar Mehta, Rajitha Hathurusinghe, and Tianyu Zhang. Your unwavering support and companionship have been invaluable, and I truly appreciate everything you have done. In addtion, I would like to acknowledge the Dr. Amine Kamin and his group at McGill University for sharing the dataset and providing insights and advices regarding bioprocess monitoring. I thank my mother and Aunt Eva for trusting me and always setting good examples with me. I thank Thiago for always being a good brother and good companion and my Father for supporting me in this part of my life. I would like to thank my partner Krystyna Wojnarowicz, for helping me and encouraging me in moments when reality shows its hard and cold face. To all of you, thank you for being my source of strength. Your love, support, and steadfast belief in me have played an essential role in my academic success. I am truly blessed to have each of you in my life. Finally, I would like to acknowledge the generous financial support from the National Research Council Canada (this research was funded by the National Research Council through AI for Design Challenge Program - operating grants AI4D-103-1) and the University of Ottawa, which has been instrumental in enabling me to pursue my research goals.

Dedication

I dedicate this work to God. God, thanks for everything!

Table of Contents

List of Tables

List of Figures

Glossary

Biomanufacturing conditions It refers to the various operational settings and variables that can influence the bioprocess, such as temperature, pH, metabolites and nutrient concentrations, and agitation speed. Table 5.1 presents five different biomanufacturing conditions for monoclonal antibody (mAb) production. 5

Bioprocessing A series of processes that use living cells or their components to produce pharmaceuticals. 1

Critical process parameters (CPP) Key variables in a bioprocess that need to be monitored to ensure process control and product quality. 1

Critical quality attributes (CQA) Attributes that are essential to ensuring the quality of the final biopharmaceutical product. 1

Data-Driven Models (DDM) Machine learning models that are developed purely based on empirical data rather than theoretical principles. 44, 122

Fast and low-cost bioprocess monitoring It can be defined as a set of methods designed to track and analyze the parameters and states (critical process parameters and quality attributes) of biomanufacturing in real time to minimize both capital and operational expenses. An example illustrating an "extreme case" of fast and low-cost bioprocess monitoring involves the real-time estimation of nutrients (glucose, glutamine), metabolites (lactate, ammonium), and production formation (titer) based on online viable cell density (Xv) measurements and initial conditions. This strategy is low-cost because it requires only one device to measure Xv instead of multiple assays/devices to perform offline and online measurements of all state variables, as in traditional approaches for bioprocess monitoring. In addition, it is fast because nutrients, metabolites, and production formation are estimated in real-time, while the offline methods take hours or days to deliver results. 2

Generic UMM It has less process knowledge than specific UMM, but it offers a generalization that can be adapted to various bioprocesses with the help of additional computational tools. The generic UMMs are developed on the limiting assumption of only proportionality with the viable cell density for metabolites while it is known that it is possible to have kinetic consumption (or production) of metabolites via pathways that are not dependent on viable cell density (e.g., direct degradation of glutamine to ammonia). The ODE system in Equations 2.1 represent a generic UMM. 4

Hybrid dynamic models A common type of HM that integrates data-driven and knowledge-based approaches, thereby enabling greater accuracy and flexibility [109,125,139,145]. 44, 121

Hybrid model (HM) It is a combination of two or more models blended together to enhance the accuracy. It aims to leverage the strengths of different methods to achieve better performance, flexibility, and interpretability. A common type of HM is hybrid dynamic models with the the integration of data-driven and knowledge-based approaches, thereby enabling greater accuracy and flexibility [109, 125, 139, 145]. 121

Monoclonal Antibodies (mAb) Laboratory-made molecules that are designed to act like natural antibodies to fight infections or diseases. 1

Nonlinear Kalman Estimator (NKE) A recursive algorithm used for estimating states and parameters of nonlinear systems in real-time. It is a state estimator based on the Kalman filter framework that is designed to handle nonlinear systems. NKEs extend the traditional Kalman filter, which is limited to linear systems, by incorporating methods that allow for the prediction and correction of states in nonlinear models. The most commonly used NKEs include: Extended Kalman Filter, Unscented Kalman Filter and Cubature Kalman Filter. 15

Offline measurements They are not obtained in real-time from sensors but are usually gathered at a lower frequency. It involves collecting samples from the biomanufacturing process and analyzing them in a separate laboratory setting. It often takes several hours or days for the result to arrive. 2

Online measurements These are measurements refer to data obtained directly from sensors during the bioprocess production. Online measurements provide immediate feedback and are crucial for dynamic state estimation where current state information is critical. Sensors in the bioreactor provide real-time measurements of parameters like pH, temperature, oxygen levels, and viable cell density (Xv). 2

Quality by Design (QbD) A concept in pharmaceutical development where product quality is ensured through well-understood processes and controls. 3

Real-time estimation It refers to the capability of the monitoring system to provide continuous and immediate feedback on the state and parameters of the bioprocess in similar frequency of online measurements, without significant delay such as in offline measurements. This can be demonstrated by the system's ability to update parameters and provide feedback within a time frame that is effectively simultaneous with the process dynamics. This often involves using high-frequency measurements and low-latency data processing techniques. In this thesis, real-time estimation is specifically mentioned to involve online viable cell density (Xv) measurements and other variables which can be updated instantaneously using soft sensors based on NKE with UMM. 3

Soft sensors Computational tools that estimate unmeasured bioprocess variables based on mathematical models and available measurements. They are models that use data from easily detectable variables to estimate variables that are difficult to quantify [62]. 3

Specific UMM It provides a detailed and customized description of a particular bioprocess with specific parameters and empirical equations such as Monod type kinetics with inhibition are used for the specific cell growth and death rates. The ODE system in Equations 2.2 represent a specific UMM. 4

Unshared parameter It is a parameter that is used only in one term of an Ordinary Differential Equation (ODE) within a system and is not shared across different ODEs of the same Unstructured Mechanistic Model (UMM). In the context of biomanufacturing, unshared parameters are often employed to model product formation dynamics in bioprocesses. 5

Viable Cell Density (Xv) A key variable in bioprocesses that indicates the concentration of living cells within a bioreactor. 14

Abbreviations

X_t Total cell density 22

AMM Ammonium 2

BAT Batch Bayesian Auto-Tuning 9

CKF Cubature Kalman Filter 4

DDM Data-driven models 3

DNKE Dual nonlinear kalman estimator 5

EKF Extended Kalman Filter 4

GLC Glucose 2

GLN Glutamine 2

HMuq Hybrid dynamic model with uncertainty quantification 9

HNKE Hybrid nonlinear Kalman estimator 9

JNKE Joint nonlinear kalman estimator 5

LAC Lactate 2

mAb Monoclonal antibody 9

MCMC Markov Chain Monte Carlo 9

MRDE Matrix Ricatti Differential Equation 8

MSV Measured state variable 8

NKE Nonlinear Kalman Estimator 3

rAAV Recombinant adeno-associated virus 6

SANTO Specific initiAl coNdiTiOn 8

UKF Unscented Kalman Filter 4

UMM Unstructured mechanistic model 3

UP Unshared parameter 8

Xv Viable cell density 3

Chapter 1

Introduction

Bioprocessing in the biopharmaceutical industry involves a series of processes that utilize living cells or their components (e.g., bacteria, enzymes, or cellular organelles) to produce pharmaceuticals, including viral vectors, and Monoclonal Antibodies (mAb) [113,126]. The biopharmaceutical sector has undergone substantial changes over the last decade, driven by developing industry patterns and the rapid advancement of new technologies. Nonetheless, the progress of bioprocess monitoring technology development has not kept pace [50, 134]. Bioprocess monitoring is crucial for developing biopharmaceutical processes because it ensures the consistent production of high-quality biopharmaceutical products [110, 131]. Bioprocess monitoring involves real-time tracking of Critical process parameters (CPP) or Critical quality attributes (CQA). Advanced monitoring tools and techniques enable early detection of deviations from optimal conditions, allowing immediate corrective actions. It reduces batch failures, enhances process efficiency, and ensures compliance with regulatory standards.

1.1 Motivation

Biomanufacturing has significant economic importance [36]. The estimated value of the worldwide biotechnology market stood at USD 1.37 trillion in 2022. It is projected to expand at a compound annual growth rate of 13.96% from 2023 through 2030 [133]. To achieve this target, it is necessary to employ more cost-effective solutions and smart manufacturing while simultaneously not sacrificing the process robustness and product quality when aiming for higher process productivity, more affordable end-products, and shorter production times [74]. Therefore, there is constant pressure for bioprocess monitoring

strategies that are fast (efficient) and low-cost (cost-effective) [50, 74, 113], particularly for the upstream stages of viral vectors and monoclonal antibodies productions [74]. Some reasons for this include the following.

First, bioprocess monitoring in areas like cell and gene therapy relies more heavily on Offline measurements [50, 134]. Commercial manufacturing still heavily relies on time-consuming offline analytics and manual control strategies [74]. For example, nutrients (Glucose (GLC), Glutamine (GLN)) and metabolites (Lactate (LAC), Ammonium (AMM)) can be measured using inline near-infrared sensors. However, due to potential cross-interference with other molecular species, the current industrial standard method of measuring glucose, glutamine, lactate, and ammonium is still based on offline measurement [50, 134]. Online measurements are data obtained in high-frequency measurements from the sensors, and offline measurements involve collecting samples from the biomanufacturing process and analyzing them in a separate laboratory setting. It often takes several hours or days for the result to arrive [25, 59, 119]. Thus, offline sensing typically leads to a lower level of process control and a higher demand for labor [50]. Typically, these offline techniques, which involve extracting cells from the bioreactor, are often time-intensive, labor-intensive, and lead to waste due to the use of expensive and potentially harmful reagents and samples [74, 153]. Furthermore, offline measurements are performed infrequently (e.g., every 12–24 hours), resulting in low-resolution process monitoring and the risk of missing metabolic changes in cells that could indicate process alterations or operational problems, making control strategies based on offline analytical measurements often ineffective [74, 152]. Additionally, each sample taken from the bioreactor carries a risk of contamination or batch loss [153]. Second, factors such as labor costs, raw materials, equipment, facility expenses, the type and quantity of in-process control analyses, and overall output all influence the bioprocess economy [113]. Additionally, there are often overlooked costs associated with lifecycle management, a lack of process agility when deviations occur, and regulatory compliance updates. For instance, the production cost of monoclonal antibodies varies significantly due to the factors presented previously, and it is expensive. An annual mAb therapy in oncology can cost around $100,000. Producing one mAb can range from $6,000 to $15,000 [13, 46, 148]. Another example is the production of recombinant Adeno-Associated Virus. Currently, a standard production run of a recombinant Adeno-Associated Virus vector treatment costs around $100,000 per dose (considering approximately 1×10^{17} vg per batch) [20]. Therefore, any technological innovation that reduces the cost per dose would be immediately beneficial [20, 113].

Fast and low-cost bioprocess monitoring can be defined as a set of methods designed to track and analyze the parameters and states (critical process parameters and quality attributes) of biomanufacturing in real time to minimize both capital and operational ex-

penses [62, 63, 113, 172]. An example illustrating an "extreme case" of fast and low-cost bioprocess monitoring involves the Real-time estimation of nutrients (glucose, glutamine), metabolites (lactate, ammonium), and production formation (titer) based on online Viable cell density (Xv) measurements and initial conditions [167, 168]. This strategy is low-cost because it requires only one device to measure Xv instead of multiple assays/devices to perform offline and online measurements of all state variables, as in traditional approaches for bioprocess monitoring. In addition, it is fast because nutrients, metabolites, and production formation are estimated in real-time, while the offline methods take hours or days to deliver results, as discussed before [62, 63, 113]. It is important to note that the information about product formation, such as titer in viral vectors and monoclonal antibody productions, is typically available only after the end of the process, which takes several days through offline measurements.

The literature indicates that Soft sensors based on Nonlinear Kalman Estimator (NKE) with Unstructured mechanistic model (UMM) can enable fast and low-cost bioprocess monitoring and achieve the goals of several initiatives, such as Quality by Design (QbD), Process Analytical tools, and Pharma 4.0 [26, 96, 106, 134, 145]. Soft sensors are computational tools that estimate unmeasured bioprocess variables based on mathematical models and available measurements [62]. It is important to point out that the soft sensor development for bioprocess monitoring can also be performed solely with Data-driven models (DDM) that are known as machine learning black-box models. However, the main obstacles to widespread adoption solely of DDM in the biomanufacturing and healthcare domains are lack of interpretability and large data requirements [34, 74, 108, 110, 138, 165]. DDMs are built purely on empirical observations of a process [130]. They establish links between input and output variables (system state variables) without having explicit knowledge of the physical laws in the system [130, 146]. Therefore, DDM does not have interpretability and cannot provide process understanding. DDM does not have the ability to explain the reasons behind its predicted outcomes, i.e., showing the rules for prediction and generalization [74]. Interpretability in ML refers to understanding and explaining predictions of the model and decision-making processes comprehensibly to humans [101]. This is especially important in the biopharmaceutical industry, where understanding the relationships between CPP and CQA is essential for optimizing production, ensuring safety, and adapting to new conditions [74]. In contrast, explainability refers to understanding how the model learns during the training process [101]. For example, complex DDMs, such as deep learning models, perform predictions without consideration of the underlying mechanisms, resulting in parameters with no physical, chemical, or biological meaning. The variables are based on correlations and not necessarily on causation [151].

In addition, the development of soft sensors solely based on DDM requires considerable

quantities of good-quality data to be collected. Poor data quality and low data quantity are a major concern for DDMs. The variables can only be predicted with high accuracy in the space adequately represented by the training set. This means that generalization ability relies on training with large amounts of data [33, 151]. However, biopharma is a data-limited industry, specifically in terms of actively generated data, where each experiment and analytics is resource-intensive, thus limiting the number of experiments and the corresponding data that can be generated. In addition, though the biopharmaceutical industry has reserves of the "historical data," they were not collected for the purpose of training a DDM and do not have good quality [62, 110].

On the contrary of DMM, UMM are derived from physical, chemical, and biological principles. In process engineering, UMM are represented as systems of ordinary or partial differential equations, and the numerical values are assigned to the UMMs parameters via parameters estimation [30,32,83,124,164] Compared to DDM, the parameters in UMM have a distinct physical meaning and, therefore, can often be estimated a priori, independent of the specific process under consideration. As long as they capture the relevant phenomena, UMM are highly reliable and show good extrapolation capabilities [109]. Therefore, UMM (purely knowledge-driven) can surpass the drawbacks of DDM [109].

Soft sensors based on NKE with UMM can enable real-time monitoring of CPP or CQA that are difficult to measure directly, or that can only be measured at low sampling frequencies in a bioprocess [144,167]. UMMs allow modeling the macro-scale of the bioprocess phenomenon, and they can be divided into two groups. Specific UMM provides a detailed and customized description of a particular bioprocess, while Generic UMM offers a more generalized system that can be adapted to various bioprocesses with the help of additional computational tools. NKE are nonlinear state estimators based on the Kalman filter framework. NKEs use a two-step recursive algorithmic process: prediction and update [73, 137]. NKEs are widely applicable in various fields today, and the most popular NKEs are [3]: Extended Kalman Filter (EKF), Unscented Kalman Filter (UKF), and Cubature Kalman Filter (CKF). The most used NKE in bioprocess monitoring are EKF and UKF [167]. It is essential to note that other nonlinear state estimators exist, such as particle filtering. However, due to the simple structure (practical implementation) and low computational effort of NKEs, these methods have gained more interest. Many research studies have been dedicated to the implementation of such filters for state and parameter estimation in bioprocess technologies based on micro-controllers which requires numerically economical and robust algorithms, such as NKEs [10, 48, 49, 111, 134, 167, 171]. 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 UMM [73, 87, 137]. In general, to enhance the performance of NKE as soft sensors, they are performed with joint and dual strategies to estimate the states and parameters of a UMM. The Joint nonlinear kalman estimator (JNKE) considers a single joint state variable vector, which includes both the states and parameters $\boldsymbol{\theta}$ of the UMM [44, 77, 92], and the Dual nonlinear kalman estimator (DNKE) employs two consecutive NKE, separating the estimation of the system states and parameters [66]. JNKE and DNKE are motivated by the need to correct the prediction of a UMM regarding state variables and to update the UMM by evolving its parameters based on the corrections made [44].

However, although JNKE and DNKE have been used in many different bioprocess monitoring applications [48, 49, 59, 63, 66, 120, 167, 168], the adoption of NKE with UMM for fast and low-cost bioprocess monitoring has been limited and improvements are needed to handle different Biomanufacturing conditions [74, 110]. This led us to the following research questions.

1.2 Research Questions

RQ1 - What is the approach to effectively enable the joint estimation of states and unshared parameters by NKEs with specific UMM under biomanufacturing conditions while ensuring accurate and computationally efficient performance of NKEs?

This problem involves enabling the joint estimation of states and Unshared parameter by NKE (EKF, UKF, and CKF) with specific UMM for fast and low-cost monitoring of a nonlinear bioprocess. Unshared parameters are unique to specific terms of an ordinary differential equation (ODE) within a system and are not shared across different ODEs. In biomanufacturing, UMM consists of ODEs and unshared parameters. The unshared parameters are typically used to model product formation (titer) dynamics in the bioprocess. When such UMM is coupled with an initial state error covariance matrix $\mathbf{P}(0)$ and a process error covariance matrix $\mathbf{Q}$ with uncorrelated elements, along with just one measured state variable, such as Xv, the joint and dual NKE cannot estimate the unshared parameters and states simultaneously. This is because the Kalman gain corresponding to the unshared parameter remains constant and equals zero. The RQ1 is a new open problem not solved before, resulting from the biopharmaceutical industry requirements. Therefore, enabling the joint NKE with specific UMM to side-step the failure case described above helps the

industry perform fast and low-cost bioprocess monitoring within similar biomanufacturing conditions with an approach that is easy to implement and has low computational requirements.

RQ2 - What is the effective method to address the problem of auto-tuning all NKE components, resulting in highly accurate estimation for any new biomanufacturing conditions?

The RQ1 represents a problem related to the application of NKE for a set of similar biomanufacturing conditions. However, the biopharmaceutical industry is always looking for new conditions that can optimize bioprocess production. Consequently, this requires optimizing all NKE components for fast and low-cost bioprocess monitoring in a new condition. However, automatic tuning of all NKE components remains an open problem, although the problem of tuning $\mathbf{Q}$ and $\mathbf{R}$ has been addressed in several studies [14,75]. Most auto-tuning methods are based on Normalized Estimation Error Squared (NEES) [22] or Normalized Innovation Squared (NIS) [22], 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 unshared parameters ($\boldsymbol{\theta}_{UP}$) is an example where traditional methods, such as those based on NIS, are inefficient. Traditional auto-tuning approaches, such as those based on NIS, often result in multimodal distribution estimation for these parameters, which leads to less conclusive determinations of the true parameter values. In the biopharmaceutical industry, there is an increasing focus on monitoring new bioprocesses with no well-defined mechanism, such as the production of Recombinant adeno-associated virus (rAAV), for which there is a lack of extensive prior research [62]. Therefore, enabling proper tuning of all components of NKEs, including unshared parameters, is critical for the real-time monitoring of novel bioprocesses with unknown mechanisms. Furthermore, manual tuning of those components is hard, time-consuming, and highly susceptible to errors [14]. Thus, the ability to auto-tune all NKE components is essential for ensuring that the estimators are optimally configured. This leads to better performance, reduced errors, and increased reliability in various practical applications. It is important to note that the measurement criteria for addressing RQ2 are based on the estimation accuracy (e.g., root-mean-square error), and robustness of tuning algorithms (e.g., performance consistency across different scenarios).

RQ3 - How to design a new NKE algorithm for fast and low-cost monitoring of highly nonlinear bioprocesses using generic UMM, ensuring high adaptability and accuracy, reduced tuning requirements, and effective iterative parameter

estimations across different biomanufacturing conditions?

RQ1 and RQ2 represent problems related to applying NKE in similar biomanufacturing conditions. Specifically, the RQ2 represents the offline optimization problem of all NKE components with a specific UMM tailored to a new biomanufacturing condition, which could have a high computational requirement in high dimensions. Therefore, the RQ3 consists of automatically initializing $\mathbf{P}(0)$ and accurately determining the parameters of a generic UMM in real-time, enabling the effective and economical monitoring of highly nonlinear bioprocesses under various biomanufacturing conditions. A generic UMM is an ODE system in which all parameters are of unshared type, and the joint and dual NKEs are inefficient when a generic UMM is used for a highly nonlinear bioprocess [137]. In addition, joint and dual NKEs have low adaptability to new conditions, often requiring a new intensive tuning process for each new biomanufacturing condition. In general, a simple NKE requires the proper tuning of $\mathbf{P}(0)$, $\mathbf{Q}$, and $\mathbf{R}$ under a specific condition. However, a joint NKE requires much more attention because it also requires specific tuning of some off-diagonal elements of $\mathbf{P}(0)$, which increases the computational demand. Typically, there is a set of optimal biomanufacturing conditions that the pharmaceutical industry can utilize. Therefore, it is important for a monitoring strategy to be able to work across all these conditions, ensuring robust monitoring. For example, by addressing the RQ3, we can simplify the tuning process and increase the adaptability (generalization) of the NKE, facilitating its integration with advanced control systems, such as model predictive control (MPC) and digital twins, enhancing bioprocess monitoring and control. This is because the MPC and digital twins are systems that demand rapid adaptation to changing biomanufacturing conditions. For instance, in a bioreactor, MPC can utilize real-time state estimates of an NKE with high adaptability to dynamically adjust inputs such as nutrient feeds, optimizing conditions for cell growth and productivity. Similarly, integrating digital twin and highly adaptable NKE allows for real-time process simulations, supporting proactive adjustments and scenario analysis to prevent issues like culture contamination and optimize production outcomes. It is essential to note that the measurement criteria for addressing RQ3 are based on adaptability (e.g., performance across different bioprocess conditions), reduced tuning requirements (e.g., fewer hyperparameters to be tuned, and less user intervention required).

1.3 Objectives

Given the motivations discussed above, the main goal of this thesis is to support the design and development of NKEs with UMM for fast and low-cost monitoring of highly

nonlinear bioprocesses, such as viral vectors and monoclonal antibody productions in different biomanufacturing conditions. This kind of bioprocess monitoring approach (based on NKE and UMM) is essential to helping the biopharmaceutical industry advance towards Biomanufacturing 4.0, aiming to improve product quality, optimize operations, and cut costs [40, 48, 49, 93, 157]. Therefore, the main goal is broken down into three specific aims to address the research questions mentioned above. Figure 1.1 depicts the research problems and their effects on the design and development of NKEs with UMM (as soft sensors) for fast and low-cost monitoring of highly nonlinear bioprocess under challenges biomanufacturing conditions, as well as the contributions of the thesis.

Aim1 - It consists of addressing the RQ1 with the proposed approach called SANTO, which is a Specific initiAl coNdiTiOn (SANTO) for the Matrix Ricatti Differential Equation (MRDE) in joint EKF. This solution is inspired by the regularization technique to avoid singularity issues in EKF. However, instead of adding a small quantity to the diagonal elements of the state error covariance matrix $\mathbf{P}$ [163], we only add a quantity to the state error covariance between the Measured state variable (MSV) and an Unshared parameter (UP) in $\mathbf{P}(0)$ for the MRDE. The proposed approach can be applied also to joint CKF, and joint UKF. It avoids the joint NKE failure by preventing the Kalman gain from being zero throughout the entire process, an unrealistic situation that would mean that the predictions of the UMM (used as a process model) are perfect.

Aim2 - A possible solution to auto-tune all NKE components is to rely not only on the measured data of state variables selected to generate the innovation errors but also on 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 [4, 53, 159]. A real-world application of NKEs for bioprocess monitoring involves a state variables vector $\mathbf{x} \in \mathbb{R}^n$ and a 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) ($\mathbf{x}_{NIE}$ does not participate of process of innovation errors generation). The $\mathbf{x}_{NIE}$ are challenging to measure directly during the execution of a bioprocess [58, 59, 62, 167]. Generally, measurements about $\mathbf{x}_{NIE}$ are obtained from offline measurements $\mathbf{y}_{offline} \in \mathbb{R}^{n-m}$ (not measured in real-time) [25, 59, 119]. Although not available for immediate feedback in NKE, the offline measurements can provide 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 $\mathbf{x}_{NIE}$. Therefore, Aim2 addresses the RQ2

with an approach called Batch Bayesian Auto-Tuning (BAT) for NKEs. 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.

Aim3 - It addresses the RQ3 with a new Gaussian filter called the Hybrid nonlinear Kalman estimator (HNKE). This new method offers three key advantages: i) it allows auto-initialization of $\mathbf{P}(0)$, reducing the time and effort required in the tuning process; ii) it facilitates real-time estimation of parameters in a generic UMM for a wide range of conditions; and iii) it eliminates the need for large volumes of data (particularly from online measurements) during the training process. This is achieved by structuring HNKE as a hybrid Gaussian filter where the prediction step relies on a Hybrid dynamic model with uncertainty quantification (HMuq), while the update step uses an unscented transformation or cubature rule. Through empirical evaluations of the synthetic bioprocess dataset related to the Monoclonal antibody (mAb) production, we validated that the HNKE can enable an efficient estimation of state variables under five different biomanufacturing conditions that represent real cases of fast and low-cost bioprocess monitoring.

1.4 Contributions

Some studies of the state-of-the-art focus on bioprocess monitoring techniques and can be considered similar works. However, these studies are based on implementing soft sensors for monitoring specific bioprocesses. Andrea Tuveri [155] advances bioprocess monitoring by developing and applying Bayesian state estimators to address uncertainties and plant-model mismatches. She implements and compares constrained EKF, UKF, and Moving Horizon Estimators for fed-batch cultivation of Corynebacterium glutamicum. The research introduces methods to handle system dynamics uncertainties and integrates infrequent and delayed measurements, significantly enhancing the accuracy and reliability of bioprocess monitoring systems. Arnas Survyla [149] proposed the development of soft sensors based on bioreactor exhaust gas analysis that increased productivity through precise and reliable estimation algorithms for biomass concentration. Abdolrahim Yousefi-Darani [169] proposed the application of a soft sensor based on NKE for yeast fermentation supervision. Vincent Brunner [18] proposed the application of soft sensors for Pichia

pastoris bioprocesses, integrating process knowledge and sensor validation. The studies described above do not deal with the biomanufacturing conditions for fast and cost-effective bioprocess monitoring. Contrary to them, the present study stands out by focusing on the development of new approaches to support the implementation of NKEs with UMMs for fast and cost-effective bioprocess monitoring, emphasizing specific and generic UMM, integration with advanced control systems and adaptability to various biomanufacturing conditions.

The contributions of the present study are the following.

C1 - The contributions of addressing RQ1

- **Theoretical Proof**: We provide theoretical proof of joint and dual EKF limitations when acting as an unshared parameter estimator for fast and low-cost bioprocess monitoring. To our knowledge, this is the first study to formally report this limitation case, which includes a vocabulary describing UMM. This is also the first study to formally define unshared parameters and weak and strong terms/variables of an ODE of UMM.

- **SANTO Approach**: We propose the SANTO approach to enable joint NKE (EKF, UKF, and CKF) with a specific UMM for fast and low-cost bioprocess monitoring under a set of similar biomanufacturing conditions. SANTO approach preserves the advantages of joint NKE, such as real-time parameter estimator, easy implementation, and low computational requirements ideal for implementation in micro-controllers.

- **Empirical Validation**: Empirical results, using a synthetic dataset of mAb productions with different challenge sceneries, show the effectiveness of SANTO in improving the performance of joint NKE (EKF, UKF, and CKF). The code and data used in this work are available on GitHub repository [1] to facilitate reproducibility.

C2 - The contributions of addressing RQ2

- **BAT:** To the best of our knowledge, we are the first to introduce an approach (BAT posterior) 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, and CKF, by enabling the use all available measured data, not just those selected for generating innovation errors.

[1] https://github.com/cristovaoiglesias/NKEs-SANTO

- **Enhanced Performance:** The BAT approach, in addition to addressing the RQ2, also addresses the limitation of the SANTO approach. The empirical evaluation of SANTO shows that JEKF, JUKF and JCKF performed better only with the SANTO approach but with an unconventional behavior of Kalman gain values in one of the studied cases, starting from a negative value and converging to near zero. BAT solves this issue under the same case study by directly estimating the unshared parameter.

- **Accuracy:** Our empirical evaluation of a synthetic bioprocess dataset demonstrates that BAT outperforms the baselines in the auto-tuning process by significantly enhancing the accuracy of a tuned EKF with a specific UMM related to state variables strongly associated with unshared parameters. The code and data used in this work are available on GitHub repository [2] to facilitate reproducibility.

- **Reduced Data Requirements**: BAT is an offline approach with reduced data requirements applicable to any new biomanufacturing condition. The empirical evaluation across all testing sets shows that BAT performs well with limited data, addressing the limitation of the biopharma industry to generate a large amount of data to develop monitoring tools

- **Unimodal Posterior Distributions**: Unlike baseline methods, BAT consistently estimates unimodal posterior distributions, which closely align with ground truth values, reducing ambiguity in parameter estimation.

- **Instructional Application**: 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 for instructional purposes.

C3 - The contributions of addressing RQ3

- **Novel hybrid Gaussian filter:** HNKE is the first hybrid Gaussian filter that combines HMuq with an unscented transformation or cubature rule for fast and low-cost monitoring of highly nonlinear bioprocess within a set of different biomanufacturing conditions.

- **Innovative auto-initialization of P(0):** This is the first study to introduce this feature, which, along with the iterative estimation of generic UMM parameters and the ability to use initial noise state variable values, simplifies the setup process,

[2] https://github.com/cristovaoiglesias/BAT

requiring only the definition of measurement noise $\mathbf{R}$ and process noise $\mathbf{Q}$, thereby reducing the effort and expertise required for model tuning.

- **Solution for generic UMM**: HNKE also addresses the limitation of the BAT and SANTO approach with generic UMM modeling a highly nonlinear bioprocess.
- **Reduced large data requirements:** HNKE can be trained using smaller amounts of data, making it more efficient and bypassing data availability issues in biopharmaceutical monitoring tool development.
- **Empirical performance superiority:** Through empirical testing with real-world scenarios, the HNKE outperformed the baselines, confirming its effectiveness and offering a more reliable and accurate tool for fast and low-cost bioprocess monitoring showing that HNKE is a real-time approach with high adaptability. The code and data used in this work are available on GitHub repository [3] to facilitate reproducibility.

[3] https://github.com/cristovaoiglesias/HNKE

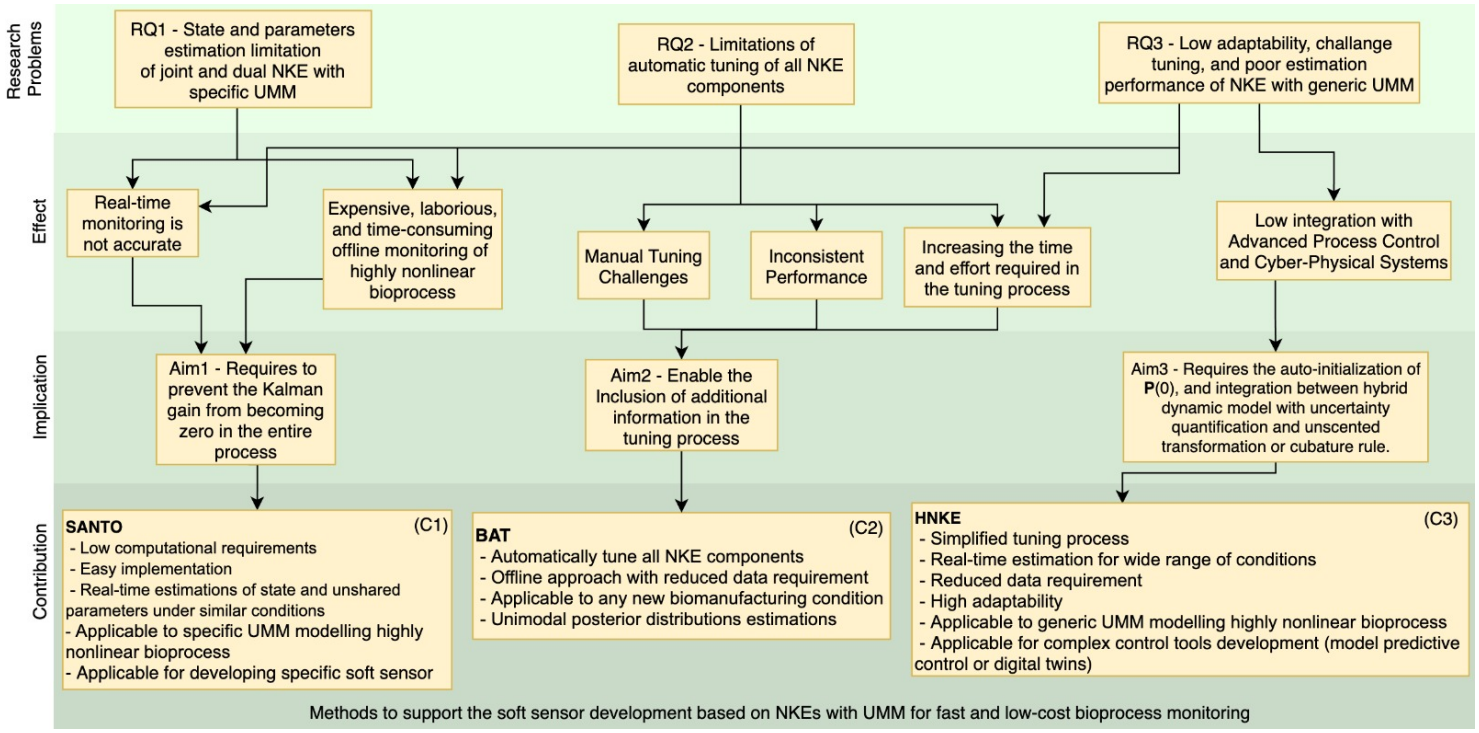

Figure 1.1: Research problems, and their effects on development of NKE with UMM for fast and low-cost bioprocess monitoring, as well as the contributions of the thesis.

1.5 Limitations and boundaries

Synthetic datasets. Obtaining data in biopharmaceutical studies is costly and time-consuming due to the need for conducting biological experiments that take a significant amount of time. Furthermore, these data are often associated with strategic products and confidential information of pharmaceutical companies, making it challenging to publicly share them with the research community. Several studies have used mechanistic models to generate synthetic data for empirical evaluation of proposed approaches for applications in biomanufacturing [74, 107, 109]. Therefore, the datasets used in this thesis are synthetic because of the scarcity of open upstream cell culture data. The Synthetic datasets have data regarding Monoclonal Antibody (mAb) productions that represent the biomanufacturing of a protein widely used as diagnostic reagents and for therapeutic purposes [70]. These

synthetic datasets are designed to mimic the complexities of real-world biomanufacturing processes but may not eliminate the need for validation with real dataset from bioreactors which comprises our future work .

Real-time estimation. Although this thesis aims to achieve real-time monitoring, the term "real-time" refers to near real-time estimations due to computational constraints [45], but represent estimation with higher frequency than offline measurements. These near real-time estimations are based on online measurements of Viable Cell Density (Xv) from sensors, which are taken at discrete intervals with a frequency of minutes or seconds. This approach ensures timely monitoring and control, but the slight delay inherent in near real-time processing must be acknowledged.

1.6 Publications

This PhD thesis is the result of my own original work, and the following publications (where I am the first author and responsible mainly for the conceptualization, investigation, implementation, validation and writing) were derived from this work:

Publications directly related to this thesis:

- Iglesias Jr, C. F., Ristovski, M., Bolic, M., & Cuperlovic-Culf, M. (2023). rAAV Manufacturing: The Challenges of Soft Sensing during Upstream Processing. *Bioengineering*, *10*(2), 229.
- Iglesias Jr, C. F., Pessoa, L., Miceli, C., Bolic M. "Limitations of Joint and Dual Nonlinear Kalman Estimators in Low-Cost Bioprocess Monitoring." *LatinX in AI Workshop at ICML 2024*.
- Iglesias Jr, C. F., Xu, X., Mehta, V., Akassou, M., Venereo-Sanchez, A., Belacel, N., ... & Bolic, M. (2022). Monitoring the Recombinant Adeno-Associated Virus Production using Extended Kalman Filter. *Processes*, *10*(11), 2180.
- Iglesias Jr, C. F., Bolic M. How Not to Make the Joint Extended Kalman Filter Fail with Unstructured Mechanistic Models. *Sensors*. 2024; 24(2):653.
- Iglesias Jr, C. F., Bolic M. "Batch Bayesian Inference for Nonlinear Kalman Estimators." Submitted.
- Iglesias Jr, C. F., Bolic M. "Hybrid Nonlinear Kalman Estimators for low-cost bioprocess." Submitted.

Publications based on this thesis (future research directions):

- Iglesias Jr, C. F., Nucci, P., Miceli, C., & Bolic, M. (2023, June). DEMDE: Decision Making Design based on Bayesian Network for Personalized Monitoring System. In *2023 26th International Conference on Information Fusion (FUSION)* (pp. 1-8). IEEE.
- Iglesias Jr, C. F., Guo, R., Nucci, P., Miceli, C., & Bolic, M. (2023, June). Automated Extraction of IoT Critical Objects from IoT Storylines, Requirements and User Stories via NLP. In *2023 10th IEEE Swiss Conference on Data Science (SDS)* (pp. 104-107). IEEE.
- Iglesias Jr, C. F., Mehta, V., Venereo-Sanchez, A., Xu, X., Robitaille, J., Voyer, R., ... & Bolic, M. (2021, November). Handling Massive Proportion of Missing Labels in Multivariate Long-Term Time Series Forecasting. In *10th International Conference on Mathematical Modeling in Physical Sciences (IC-MSQUARE 2021)* (Vol. 2090, No. 1, p. 012170). IOP Publishing.
- Iglesias Jr, C. F., Miceli, C., & Bolic, M. (2023). An Architectural Design Decision Model for Resilient IoT Application. *arXiv e-prints*, arXiv-2306.
- Iglesias Jr, C. SA de Outeiro, L., Miceli, C., Bolic M. "Two Students: Enabling Uncertainty Quantification in Federated Learning Clients." *NeurIPS 2024 Workshop on Bayesian Decision-making and Uncertainty.*

1.7 Thesis Organization

This thesis is organized as follows: In Chapter 2, we detail and review the background in relevant areas, namely UMM and NKE. This chapter provides foundational knowledge essential for understanding the subsequent chapters. Chapter 3 introduces our approach for joint Nonlinear Kalman Estimator (NKE) with SANTO for real-time parameter estimation in specific UMM. We illustrate the limitations of joint and dual NKEs for fast and low-cost bioprocess monitoring and present our proposed solution. The chapter includes empirical evaluations demonstrating the effectiveness of the SANTO approach compared to classic joint NKEs. In Chapter 4, we explore the auto-tuning of NKE components by presenting the Batch Bayesian Auto-Tuning method. This technique leverages all available measured data to automatically tune the components of any NKE. The chapter provides

detailed experimental comparisons with state-of-the-art approaches, highlighting the improved performance and consistency achieved with BAT. Chapter 5 examines the development of HNKE for enhanced adaptability and performance in monitoring highly nonlinear bioprocesses. This chapter discusses the integration of hybrid dynamic models with uncertainty quantification and presents empirical evaluations showcasing the superior accuracy and robustness of HNKE in various biomanufacturing conditions. Finally, in Chapter 6, we conclude the thesis by summarizing our contributions and discussing potential future work. This chapter outlines the broader implications of our research and suggests directions for further investigation to advance the field of bioprocess monitoring.

Chapter 2

Background

2.1 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 (considered the safest and most effective way to repair single-gene abnormalities in non-dividing cells) [62, 63, 83, 93]. 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 model predictive control, digital twin (DT) development, and soft sensors for bioprocess monitoring, the industrial application of UMMs is still nascent [100, 105, 121, 134]. The UMMs are important because they allow for the macro-scale modeling of the functionality of the bioreactor and can provide insight into the underlying macro-scale phenomena of upstream process. For example, this kind of model can be used to depict the dynamics of the cell density, viability, nutrient/metabolite concentrations, and product titer [100, 134, 151]. Therefore, UMMs are the most suitable option for explaining observed phenomena, predicting process behavior, and analyzing intrinsic bioprocess characteristics such as controllability [93]. The main difference between UMM and structured mechanistic model (SMM) is that, in general, SMM is more complex than UMM because it provides details about the intracellular environment of a homogenous cell population. Therefore, the development of SMM for a specific bioprocess requires extensive domain knowledge and substantial effort [93, 151]. SMM is unsuitable for dynamic control of bioprocess in bioreactors used

commonly in biomanufacturing because many of the variables used in SMM cannot be manipulated directly [93]. SMM is most suited for cell-line development, in which cells' genome-level properties are changed to produce desired process behavior [93].

In the following we have some cases of UMM and SMM to illustrate different approaches to modeling bioprocesses and to address various biomanufacturing conditions.

2.1.1 Case 1

The ODE system 2.1 represents a UMM of the dynamics of mammalian cell growth $Xv(t)$, nutrients $N(t)$ and metabolite/production formation $MP(t)$ in a general form [41,107]. In this system, $X_V(t)$ represents the viable cell concentration over time, $\frac{dX_V(t)}{dt}$ denotes the rate of change of viable cell concentration, and μ_{X_v} is the specific growth rate of the mammalian cells. The variable $N(t)$ stands for the nutrient concentration, with $\frac{dN(t)}{dt}$ indicating the rate of nutrient consumption, where μ_N is the specific nutrient uptake rate by the cells. The variable $MP(t)$ represents the concentration of metabolites or the production formation, and $\frac{dMP(t)}{dt}$ is the rate of metabolite or product formation, with μ_{mp} being the specific production rate linked to cell growth. Together, these equations capture the essential interactions between cell growth, nutrient consumption, and metabolite production in mammalian cell cultures.

$$\begin{aligned} \frac{dX_V(t)}{dt} &= \mu_{X_v} X_V(t) \\ \frac{dN(t)}{dt} &= -\mu_N X_V(t) \\ \frac{dMP(t)}{dt} &= \mu_{mp} X_V(t) \end{aligned} \tag{2.1}$$

2.1.2 Case 2

The ODE system 2.2 is a UMM used for mAb production [117]. This system represents the cell growth, cell dead, uptake of substrates, metabolism, and production process with 35 parameters. The provided system of ordinary differential equations (ODEs) models various aspects of the monoclonal antibody (mAb) production process, representing the dynamics of cell growth, nutrient uptake, metabolite production, and mAb synthesis. The equations account for the fractional growth rate (fgr), viable cell concentration (X_V), and

dead cell concentration (X_d), with specific terms for the growth and death rates influenced by glucose ([GLC]), glutamine ([GLN]), lactate ([LAC]), and ammonium ([AMM]). The glucose and glutamine consumption equations reflect their uptake by cells, while the lactate and ammonium equations account for their production as metabolic byproducts. Additionally, the dynamics of asparagine ([ASN]), aspartate ([ASP]), and alanine ([ALA]) are included, showing their interactions with other components. Finally, the mAb production equation ($\frac{d\ [mAb]}{dt}$) models the synthesis of monoclonal antibodies as a function of viable cell concentration and glucose availability. These comprehensive ODEs capture the intricate interactions within the bioprocess, essential for optimizing and controlling mAb production to enhance productivity and efficiency. More details can be found in [117].

$$
\begin{aligned}
\frac{d\ fgr}{dt} &= -k_{11}\frac{fgr}{1+([GLC]/K_{12})} \\
\frac{d\ X_V}{dt} &= \mu_{max}.fgr.X_V.\frac{[GLC]}{K_{21}+[GLC]}.\frac{[GLN]}{K_{22}+[GLN]}.\frac{1}{[LAC]/K_{24}+1}.\frac{1}{[AMM]/K_{23}+1} - C \\
C &= k_d.(1-fgr).X_V.(\frac{1}{1+([AMM]/k_{25})^n}+\frac{k_{26}}{[GLC]}) \\
\frac{d\ X_d}{dt} &= k_d.(1-fgr).X_V.(\frac{1}{1+([AMM]/k_{25})^n}+\frac{k_{26}}{[GLC]}) - k_{lys}.X_d \\
\frac{d\ [GLC]}{dt} &= -(\frac{K_{31}.[GLC].[GLN]}{(k_{32}+[GLC])(k_{33}+[GLN])}+\frac{K_{34}.[GLC]}{(k_{35}+[GLC])}).X_V - k_{36}X_V \\
\frac{d\ [GLN]}{dt} &= -k_{41}.X_V.(\frac{[GLC].[GLN]}{(k_{42}+[GLC])(k_{43}+[GLN])}) \\
\frac{d\ [LAC]}{dt} &= -k_{51}.X_V.\frac{d\ [GLC]}{dt} \\
\frac{d\ [ASN]}{dt} &= -k_{61}.X_V.\frac{[ASN]}{(k_{62}+[ASN])} \\
\frac{d\ [ASP]}{dt} &= X_V.(\frac{K_{61}[ASN]}{K_{62}+[ASN]}-\frac{K_{63}[ASP]}{K_{64}+[ASP]}+\frac{K_{65}.[GLC].[GLN]}{(K_{66}+[GLC])(K_{67}+[GLN])}) \\
\frac{d\ [ALA]}{dt} &= X_V.(\frac{K_{65}.[GLC].[GLN]}{(K_{66}+[GLC])(K_{67}+[GLN])}-\frac{K_{68}[ALA]}{K_{69}+[ALA]}) \\
\frac{d\ [AMM]}{dt} &= -k_{71}.\frac{d\ [GLN]}{dt}+k_{72}.X_V.(\frac{K_{61}[ASN]}{K_{62}+[ASN]}+\frac{K_{63}[ASP]}{K_{64}+[ASP]}+\frac{K_{68}.[ALA]}{(K_{69}+[ALA])}) \\
\frac{d\ [mAb]}{dt} &= X_V(k_{81}+k_{82}.[GLC])
\end{aligned}
\tag{2.2}
$$

The parameters are as follows:

- fgr: Fractional growth rate
- X_V: Viable cell concentration
- X_d: Dead cell concentration
- μ_{max}: Maximum specific growth rate

- $k_{11}, k_d, k_{lys}, k_{25}, k_{26}, k_{31}, k_{32}, k_{33}, k_{34}, k_{35}, k_{36}, k_{41}, k_{42}, k_{43}, k_{51}, k_{61}, k_{62}, k_{63}, k_{64}, k_{65}, k_{66}, k_{67}, k_{68}, k_{69}, k_{71}, k_{72}, k_{81}, k_{82}$: Rate constants
- $K_{12}, K_{21}, K_{22}, K_{23}, K_{24}, K_{61}, K_{62}, K_{63}, K_{64}, K_{65}, K_{66}, K_{67}, K_{68}, K_{69}$: Saturation constants
- n: Exponent for ammonium inhibition

2.1.3 Case 3

The ODE system 2.3 represents the Michaelis-Menten model for enzymatic reactions [66]. This model describes the dynamics of enzyme-substrate interactions and the formation of products in a biochemical reaction. The system consists of four differential equations that account for the concentrations of the enzyme (E), substrate (S), enzyme-substrate complex (ES), and product (P) over time. The parameters k_1, k_2, and k_3 are the rate constants for the respective reactions.

$$\begin{aligned}
\frac{dE(t)}{dt} &= -k_1 E(t)S(t) + k_2 ES(t) + k_3 ES(t) \\
\frac{dS(t)}{dt} &= -k_1 E(t)S(t) + k_2 ES(t) \\
\frac{dES(t)}{dt} &= k_1 E(t)S(t) - k_2 ES(t) - k_3 ES(t) \\
\frac{dP(t)}{dt} &= k_3 ES(t)
\end{aligned} \tag{2.3}$$

This structured mechanistic model provides a detailed view of the enzyme kinetics and is essential for understanding the biochemical reaction mechanisms at the molecular level. Other example of SSM can be seen in [112].

2.1.4 Case 4

The ODE system 2.4 is a UMM used for Mab production [90]. This system represents the cell growth, uptake of substrates, metabolism, and production process with 16 parameters described in the Table 3.1. 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 [90].

$$
\begin{aligned}
\frac{d\ X_V}{dt} &= (\mu - \mu_d)X_V \\
\frac{d\ X_t}{dt} &= \mu X_V - k_{lysis}(X_t - X_V) \\
\mu &= \mu_{max} . \frac{[GLC]}{K_{glc} + [GLC]} . \frac{[GLN]}{K_{gln} + [GLN]} . \frac{K_{Ilac}}{K_{Ilac} + [LAC]} . \frac{K_{Iamm}}{K_{Iamm} + [AMM]} \\
\mu_d &= \frac{\mu_{d,max}}{1 + (K_{d,amm} + [AMM])^2} \\
\frac{d\ [GLC]}{dt} &= -Q_{glc} X v \\
\frac{d\ [GLN]}{dt} &= -Q_{gln} X v - K_{d,gln}[GLN] \\
\frac{d\ [LAC]}{dt} &= Q_{lac} X v \\
\frac{d\ [AMM]}{dt} &= Q_{amm} X v + K_{d,gln}[GLN] \\
Q_{glc} X v &= \frac{\mu}{Y_{x,glc}} + m_{glc} \\
Q_{gln} X v &= \frac{\mu}{Y_{x,gln}} + m_{gln} = \frac{\mu}{Y_{x,gln}} + \frac{\alpha_2 [GLN]}{\alpha_2 + [GLN]} \\
Q_{lac} X v &= Y_{lac,glc} Q_{glc} \\
Q_{amm} X v &= Y_{amm,gln} Q_{gln} \\
\frac{d\ [mAb]}{dt} &= (2 - \gamma\mu) Q_{mAb} . X_V
\end{aligned}
\tag{2.4}
$$

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 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 strength of UMM 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 the biomanufacturing processes. The parameter Q_{mAb}, representing the specific mAb production rate, is particularly notable as it is an unshared parameter, meaning that its value is unique to this process and not shared with other models or components within this system.

2.1.5 Specific UMM

Specific UMMs provide a detailed and customized description of a particular bioprocess. The UMM of case 4 (Equations 2.4) is a UMM used for Mab production [90]. This system is

an example of a specific UMM because it provides a detailed mathematical representation to describe the temporal progression of the production of therapeutic monoclonal antibodies (mAbs). These models consist of a system of ordinary differential equations (ODEs) that explicitly capture the dynamics of cell growth, nutrient uptake, metabolite production, and product formation. Specific UMMs are characterized by their detailed parameters, which are tailored to specific processes. For example, the UMM (Equations 2.4) for mAb production includes 16 parameters, each defining different biological rates and interactions, such as specific growth rates, death rates, and substrate consumption rates.

2.1.6 Generic UMM

Generic UMMs are ODE systems that offer a more generalized framework that can be adapted to various bioprocesses with the help of additional model. A generic UMM is a more abstract, and general form of modeling bioprocesses, aimed at representing the cell culture dynamics based on mass balances without detailed specificity to any particular process. The generic UMM utilizes a simpler system of ODEs, which describe the overall behavior of cell growth, nutrient uptake, metabolism, and product formation using fewer parameters. These parameters typically include specific growth rates and specific rates of substrate uptake and metabolite production. The generic UMM is designed to be broadly applicable to various bioprocesses, providing a foundational framework that can be adapted or expanded as more specific information becomes available. This generality is often supplemented by machine learning techniques, such as ensembles of multilayer perceptrons (MLPs), which estimate specific rates based on experimental data, thereby compensating for the reduced detailed mechanistic knowledge [107, 108, 110].

The ODE system 2.5 is an example of generic UMM (used in the empirical evaluation of HNKE) representing the cell culture based on mass balances in Mab production in a general form [41, 107, 109]. It is defined as

$$\mathbf{g}(\mathbf{x}_t; \boldsymbol{\theta}_t) = \begin{array}{r} \frac{dX_V(t)}{dt} = \mu_{X_v}(t_k)X_V(t), \\ \frac{d[Glc(t)]}{dt} = -\mu_{Glc}(t_k)X_V(t), \\ \frac{d[Gln(t)]}{dt} = -\mu_{Gln}(t_k)X_V(t), \\ \frac{d[Lac(t)]}{dt} = \mu_{Lac}(t_k)X_V(t), \\ \frac{d[Amm(t)]}{dt} = \mu_{Amm}(t_k)X_V(t), \\ \frac{d[mAb(t)]}{dt} = \mu_{mAb}(t_k)X_V(t), \end{array} \tag{2.5}$$

where the state variables $\mathbf{x}_t$ are represented by [Xv(t), Glc(t), Gln(t), Lac(t), Amm(t), mAb(t)], and the parameters $\boldsymbol{\theta}_{k-1}$ by [$\mu_{Xv}(t_k)$, $\mu_{Glc}(t_k)$, $\mu_{Gln}(t)$, $\mu_{Lac}(t_k)$, $\mu_{Amm}(t_k)$, $\mu_{mAb}(t_k)$]. This system represents the cell growth, uptake of substrates, metabolism, and production process with six parameters: the specific cell growth rate (μ_{X_v}), the specific rates of uptake (consumption) of the main nutrients, glucose (μ_{Glc}) and glutamine (μ_{Gln}), the specific rates of production of the metabolite waste, lactate (μ_{Lac}) and ammonium (μ_{Amm}), and the specific rate of production of mAb(μ_{mAb}). Furthermore, the reduced knowledge in this generic UMM is compensated by the ensemble of MLP, as shown in the schematics in Figure 5.1, which estimates specific rates at $k-1$ based on the information from the culture experiments at $k-1$.

2.2 Importance of parameter and state variable estimation

System identification enables us to obtain a UMM and its parameters related to a bioprocess based on observed data [31, 72, 123]. But it is difficult to use the same parameters to perform prediction in different conditions of the same bioprocess. Given that, an UMM is commonly implemented with the NKE to improve the prediction accuracy by estimating state variables and parameters simultaneously based on the initial parameters [31, 62, 63, 72, 74, 110, 123, 145, 170], see Figure 2.1.

Estimation of Parameters. Depending of the UMM type, it is not always necessary to estimate all parameters of a UMM [62, 63, 145, 170]. For example, specific UMM such as described in Equations 2.4 has knowledge about the process. Then, in some case is not need to estimate all parameters. Therefore, only the estimation of Q_{mAb} may be enough. This because Q_{mAb} (metabolite production rate) is a unshared parameter, and they are crucial for monitoring and optimization. On the other hand, generic UMM has low knowledge about the process and in the most of the cases it is need to estimate all parameters.

Important State Variables. There are critical state variables that need to be estimated accurately [31, 62, 63, 72, 74, 110, 123, 145, 170]. Variables such as viable cell density (Xv), glucose and glutamine concentrations, lactate and ammonium levels, and product titer (in processes like monoclonal antibody production) are vital for bioprocess monitoring. These variables are directly related to the CQA and CPP of the bioprocess, which influence product quality and process efficiency. Therefore, not all parameters need to be estimated, but the key state variables (those affecting product formation and process control) are crucial and should be monitored accurately for effective bioprocess management.

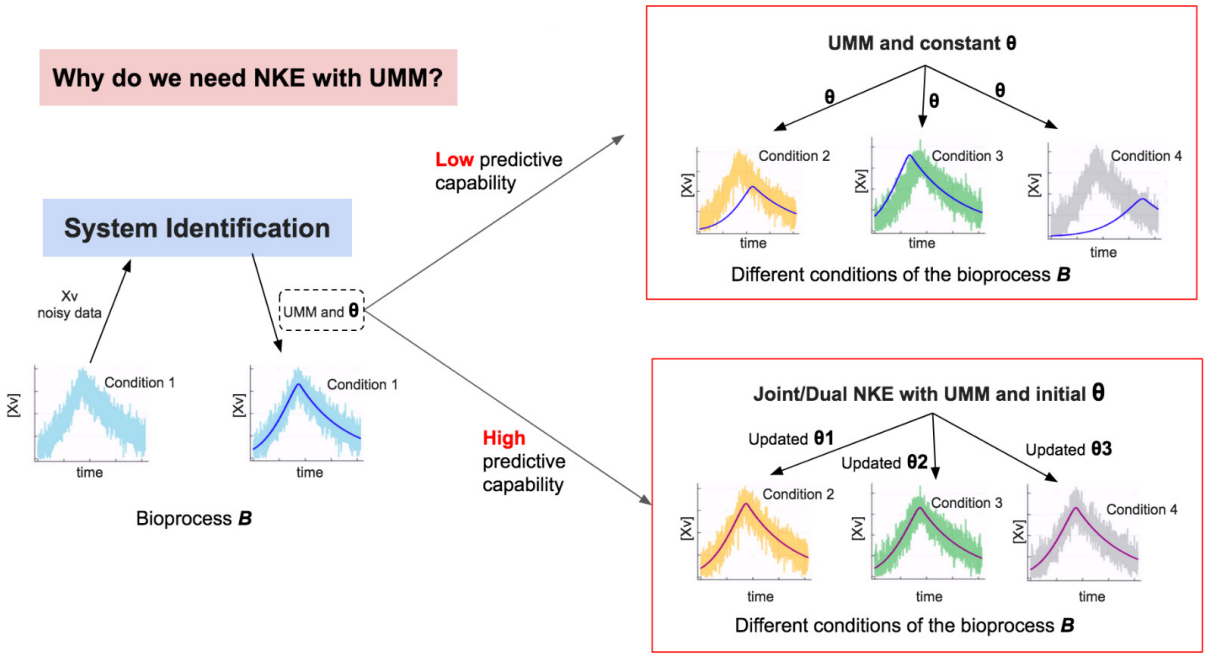

Figure 2.1: System identification enables us to obtain a UMM and its parameters related to a bioprocess based on observed data. But it is difficult to use the same parameters to perform prediction in different conditions of the same bioprocess. Given that, an UMM is commonly implemented with the NKE to improve the prediction accuracy by estimating states variables and parameters simultaneously based on the initial parameters.

2.3 Nonlinear Kalman Estimators

2.3.1 Markov and Independence Assumptions in Probabilistic State Space Models

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

$$
\begin{aligned}
&\boldsymbol{\Theta} \sim p(\boldsymbol{\Theta}),\\
&\mathbf{x}_0 \sim p(\mathbf{x}_0|\boldsymbol{\Theta}),\\
&\mathbf{x}_k \sim p(\mathbf{x}_k|\mathbf{x}_{k-1}, \boldsymbol{\Theta}),\\
&\mathbf{y}_k \sim p(\mathbf{y}_k|\mathbf{x}_k, \boldsymbol{\Theta})
\end{aligned}
\tag{2.6}
$$

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 $\mathbf{x}_k = \{\mathbf{x}_{IE,k} \cup \mathbf{x}_{NIE,k}\}$. $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ 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 $p(\mathbf{y}_k|\mathbf{x}_{1:k}, \mathbf{y}_{1:k-1}) = p(\mathbf{y}_k|\mathbf{x}_k)$ [64]. It is important to point out that online measurements $\mathbf{y}_{online,k}$ are data collected in real time from sensors during the operation of a system, and offline measurements $\mathbf{y}_{offline,k}$, on the other hand, are collected at a lower frequency, typically involving more detailed analysis. In addition, the parameter $\boldsymbol{\Theta}$ was considered implicit in dynamic and measurement expressions.

2.3.2 State space model

Consider the state space model

$$
\begin{aligned}
\mathbf{x}_k &= \mathbf{g}(\mathbf{x}_{k-1}) + \boldsymbol{q}_{k-1}\\
\mathbf{y}_k &= \mathbf{h}(\mathbf{x}_k) + \boldsymbol{r}_k
\end{aligned}
\tag{2.7}
$$

where $\mathbf{x}_k \in \mathbb{R}^n$ is the state, $\mathbf{y}_k \in \mathbb{R}^{m \leq n}$ is the measurement, for k=1, 2, $\cdots$ K, $\boldsymbol{q}_{k-1} \sim \mathcal{N}(0, \mathbf{Q})$ is the white Gaussian process noise, $\boldsymbol{r}_k \sim \mathcal{N}(0, \mathbf{R})$ is the white Gaussian measurement noise, $\mathbf{g}(\cdot)$ is the unstructured mechanistic model (UMM) composed of a dynamic ODE system (dynamic model function), and $\mathbf{h}(\cdot)$ is the measurement model function. It is important to note that $\mathbf{g}(\mathbf{x}_{k-1}; \boldsymbol{\theta}_{k-1}) = \mathbf{x}'_k$ where $\mathbf{x}'_k \in \mathbb{R}^n$ is the solution of $\mathbf{g}(\mathbf{x}_{k-1}; \boldsymbol{\theta}_{k-1})$ that is obtained, for example, by an ODE solver [21, 127], from t_{k-1} to t_k given the initial state variables $\mathbf{x}_0 = \mathbf{x}_{k-1}$ and a vector of parameters $\boldsymbol{\theta}_{k-1}$ of the UMM.

2.3.3 Batch and Recursive Bayesian Inference

The batch Bayesian inference (solution) to a parameter estimation problem considering the Equation 2.6 can be formulated in three steps. First, specify the likelihood model of measurements $p(\mathbf{y}_k|\mathbf{x}_k, \boldsymbol{\Theta})$ given the parameter $\boldsymbol{\Theta}$. To focus on estimating the parameters $\boldsymbol{\Theta}$, it is useful to integrate out the state variable $\mathbf{x}_k$. Then, assuming the conditional independence of measurements $\mathbf{y}_k$, we have $p(\mathbf{y}_{1:T}|\boldsymbol{\Theta}) = \prod_{k=1}^{T} p(\mathbf{y}_k|\boldsymbol{\Theta})$. Second, define the prior distribution $p(\boldsymbol{\Theta})$. Lastly, compute the posterior distribution by the Bayes' rule:

$$p(\boldsymbol{\Theta}|\mathbf{y}_{1:T})_B = \frac{1}{Z} p(\boldsymbol{\Theta}) \prod_{k=1}^{T} p(\mathbf{y}_k|\boldsymbol{\Theta}), \tag{2.8}$$

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

The recursive Bayesian inference (solution) to the above problem can also be formulated in three steps [122]. Initially, assuming the conditional independence of measurements $\mathbf{y}_k$, specify the likelihood model of measurements as $p(\mathbf{y}_k|\boldsymbol{\Theta})$. Second, define prior distribution $p(\boldsymbol{\Theta})$ 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:

$$\begin{aligned}
p(\boldsymbol{\Theta}|\mathbf{y}_1) &= \frac{1}{Z_1} p(\mathbf{y}_1|\boldsymbol{\Theta}) p(\boldsymbol{\Theta}), \\
p(\boldsymbol{\Theta}|\mathbf{y}_{1:2}) &= \frac{1}{Z_2} p(\mathbf{y}_2|\boldsymbol{\Theta}) p(\boldsymbol{\Theta}|\mathbf{y}_1), \\
p(\boldsymbol{\Theta}|\mathbf{y}_{1:3}) &= \frac{1}{Z_3} p(\mathbf{y}_3|\boldsymbol{\Theta}) p(\boldsymbol{\Theta}|\mathbf{y}_{1:2}), \\
&\vdots \\
p(\boldsymbol{\Theta}|\mathbf{y}_{1:T})_R &= \frac{1}{Z_T} p(\mathbf{y}_T|\boldsymbol{\Theta}) p(\boldsymbol{\Theta}|\mathbf{y}_{1:T-1}).
\end{aligned} \tag{2.9}$$

Therefore, we have that the posterior distribution (Equation 2.9) obtained through recursive Bayesian inference is equivalent to the posterior distribution (Equation 2.8) obtained through batch Bayesian inference, $p(\boldsymbol{\Theta}|\mathbf{y}_{1:T})_B = p(\boldsymbol{\Theta}|\mathbf{y}_{1:T})_R$ [122, 137]. It is important to point out that the recursive formulation of Bayesian inference has two useful properties [137]: 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 $\boldsymbol{\Theta}$, 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 the basis of Bayesian filtering theory, where time behavior is modeled by assuming the state to be a time-dependent stochastic process $\mathbf{x}(\mathrm{t})$.

2.3.4 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) a **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 2.2.

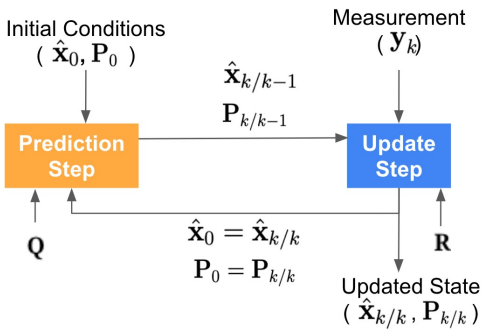

Figure 2.2: Two-step recursive algorithmic process used by NKEs such as EKF, UKF, and CKF. The state predictions are updated with new measurements, and this updated state becomes the basis for the next prediction.

The Kalman filter is the closed-form solution to the Bayesian filtering equations 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, such as in biomanufacturing where the bioprocesses are highly nonlinear. Therefore, NKEs have been developed to address this. The two-step recursive algorithmic process used by NKEs (EKF, UKF, and CKF) is summarized as follows [3, 73]: **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 dynamic model (UMM described by $\mathbf{g}(\hat{\mathbf{x}}_0, \boldsymbol{\theta})$), initial conditions ($\hat{\mathbf{x}}_0$ and $\mathbf{P}_0$) and $\mathbf{Q}$. See the Figure 2.2. Each Nonlinear Kalman Estimator performs the prediction step differently: EKF [68] linearizes the system's dynamics around the current state estimate for prediction, UKF [161] uses deterministically chosen sigma points to approximate nonlinear transformations, and CKF [6] 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 NKE (EKF, UKF, 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 2.2.

2.3.5 Extended Kalman Filter

The EKF can be implemented in Discrete-Discrete, and Continuous-Discrete versions [38, 65, 81]. The most common version used in biomanufacturing for modeling nonlinear biochemical dynamic pathways is Continuous-Discrete EKF (CD-EKF) [5, 117, 120, 166] and we will describe it here. The EKF requires a state–space model to perform estimation on the state variables of a process (nonlinear system) present in a state variable vector $\mathbf{x}(t)$ [66, 67, 170]. A state-space model consists of process and measurement (observation) models [16]. EKF linearizes the nonlinear system (state–space model) by calculating the Jacobians of the nonlinear process and measurement models based on the first-order Taylor series expansion in order to analytically propagate the Gaussian random-variable representation [44, 166, 170].

Effectively, the nonlinear dynamics are approximated by a time-varying linear system, and the linear Kalman filters equations are applied. Essentially, the mean $\hat{\mathbf{x}}(t)$ and error covariance matrix $\mathbf{P}(t)$ of the state variables in $\mathbf{x}(t)$ are recursively corrected. The EKF

recursively estimates the (posterior) mean $\hat{\mathbf{x}}(t_{k/k})$ and error covariance matrix $\mathbf{P}(t_{k/k})$ of the state variables by combining the predicted (a priori) mean $\hat{\mathbf{x}}(t_{k/k-1})$ and error covariance matrix $\mathbf{P}(t_{k/k-1})$ with the current noisy measurement $\mathbf{y}_k$ [44].

Process Model: An UMM as described in Section 2.1 can be used as the process model of EKF. The state variables vector to be used by the EKF is composed of the state variables of the UMM (observed and unobserved) and the state variables vector is defined as:

$$\mathbf{x}(t) = [x_1, x_2, ..., x_n]^T. \tag{2.10}$$

Subsequently, the process model is represented as

$$\hat{\mathbf{x}}(t_{k/k-1}) = \mathbf{g}(\hat{\mathbf{x}}(t_{k-1})) + \boldsymbol{q}(t), \tag{2.11}$$

where $\mathbf{g}(\cdot)$ denotes non-linear functions of the state variables in $\mathbf{x}(t)$, which corresponds to an UMM. The process model is formulated in a continuous time t and the process noise vector (additive white Gaussian noise) is represented by $\boldsymbol{q} \sim \mathcal{N}(0, \mathbf{Q})$, with zero mean and error covariance matrix of process model represented by $\mathbf{Q}$.

Measurement Model: The measurement model is treated as a discrete system and defined as

$$\mathbf{y}_k = \mathbf{h}(\hat{\mathbf{x}}(t_{k/k-1})) + \boldsymbol{r}_k. \tag{2.12}$$

The non-linear function $\mathbf{h}$ in the measurement model relates the current state variables to the measurements $\mathbf{y}_k$. The measurement noise vector (additive white Gaussian noise) is represented by $\boldsymbol{r}_k \sim \mathcal{N}(0, \mathbf{R})$, with zero mean and measurement noise variance represented by $\mathbf{R}$. When the some state variables can be measured directly, we have a simple case and $\mathbf{h}$ can be a linear model. If $\mathbf{h}$ is linear, we have $\mathbf{h}(\hat{\mathbf{x}}(t_{k/k-1})) = \mathbf{H}\hat{\mathbf{x}}(t_{k/k-1})$ [66, 117, 166]. Where the matrix $\mathbf{H}$ is a linear operator (row vector) that matches the states variables of $\hat{\mathbf{x}}(t_{k/k-1})$ to the measured variables $\mathbf{y}_k$ that are obtained at a discrete instance k [117, 166]. Consequently, the measurement model (2.12) can be rewritten as

$$\mathbf{y}_k = \mathbf{H}\hat{\mathbf{x}}(t_{k/k-1}) + \boldsymbol{r}_k. \tag{2.13}$$

EKF algorithm: The EKF algorithm is implemented through the initial condition, prediction step (time update) and correction step (measurement update) [66, 67, 117, 120, 166].

Initialization step: The initial conditions are composed of the initial mean $\hat{\mathbf{x}}_0 = E[\mathbf{x}_0]$, and initial error covariance matrix $\mathbf{P}_0 = \mathbf{P}_{i,i}(t = 0) = E[(\mathbf{x}_0 - \hat{\mathbf{x}}_0)(\mathbf{x}_0 - \hat{\mathbf{x}}_0)^T]$ of state variables vector [44].

Prediction step: In this step, the *a priori* predictions represented by the predicted mean $\hat{\mathbf{x}}(t_{k/k-1})$ and predicted error covariance matrix $\mathbf{P}(t_{k|k-1})$ of state variables vector $\mathbf{x}(t)$ are obtained respectively by numerically integrating $\mathbf{g}(\mathbf{x}(t))$ from discrete time t_{k-1} to t_k the following equation

$$\hat{\mathbf{x}}(t_{k/k-1}) = \hat{\mathbf{x}}(t_{k-1}) + \int_{t_{k-1}}^{t_k} \mathbf{g}(\hat{\mathbf{x}}(t))dt \Big|_{\hat{\mathbf{x}}(t_{k-1})} \tag{2.14}$$

and solving the matrix Riccati Differential equation (MRDE) for predict the state error covariance matrix [8, 65]

$$\frac{d\mathbf{P}(t)}{dt} = \mathbf{J}_t^{\mathbf{g}}\mathbf{P}(t) + \mathbf{P}(t)(\mathbf{J}_t^{\mathbf{g}})^T + \mathbf{Q} \tag{2.15}$$

from t_{k-1} to t_k, where a new measurement is obtained at time k [82], [65] and [37]. The Equation 2.15 is basically a matrix of ODEs, and the matrix of ODEs solutions obtained from t_{k-1} to t_k represent each state error covariance of the system.

Correction step: In this step, the results of the prediction step ($\hat{\mathbf{x}}(t_{k/k-1})$ and $\mathbf{P}(t_{k|k-1})$) are combined with the measured value $\mathbf{y}_k$ and Kalman gain ($\mathbf{K}_k$) to provide the estimated mean $\hat{\mathbf{x}}(t_{k/k})$ and estimated error covariance matrix $\mathbf{P}(t_{k|k})$ of state variables vector using the following equations:

$$\mathbf{K}_k = \mathbf{P}(t_{k|k-1})\mathbf{H}^T(\mathbf{H}\mathbf{P}(t_{k|k-1})\mathbf{H}^T + \mathbf{R})^{-1} \tag{2.16}$$

$$\hat{\mathbf{x}}(t_{k/k}) = \hat{\mathbf{x}}(t_{k/k-1}) + \mathbf{K}_k(\mathbf{y}_k - \mathbf{H}\hat{\mathbf{x}}(t_{k/k-1})) \tag{2.17}$$

$$\mathbf{P}(t_{k|k}) = (\mathbf{I} - \mathbf{K}_k\mathbf{H})\mathbf{P}(t_{k|k-1}) \tag{2.18}$$

The Kalman gain is a scaling factor (ratio) to estimate the state variables by setting a value between the predicted state and measured state [65, 84]. The $\mathbf{K}_k$ chooses a value along the residual range ($\mathbf{y}_k$ - $\mathbf{H}\hat{\mathbf{x}}(t_{k/k-1})$) [44, 84]. $\mathbf{K}_k$ enables to set a value for $\hat{\mathbf{x}}(t_{k/k})$ between the $\hat{\mathbf{x}}(t_{k/k-1})$ (prediction) and $\mathbf{y}_k$ (measurement) using Equation 2.17, and update the belief regards the state variables based on how certain we are regards the measurement using the Equation 2.18, [84]. The Kalman gain is computed as a ratio of prior and measurement uncertainty available; see Equation 2.16. It is important to point out that

linear operator $\mathbf{H}$ matches the states variables of $\hat{\mathbf{x}}(t_{k/k-1})$to the measured variables $\mathbf{y}_k$ that are obtained at a discrete instance. It is linear operator with zeros and ones. Where the elements ones represent elements of the state variable vector that are measured. For example, if the state variables vector has 3 variables and only the first one is measured, we have $\mathbf{H} = [1\ 0\ 0]$.

Using the estimated mean $\hat{\mathbf{x}}(t_{k/k})$ and estimated error covariance matrix $\mathbf{P}(t_{k|k})$ state variables vector as an initial condition, we can return to the prediction step until the next measurement be obtained and everything repeated again.

2.3.6 General Gaussian filters and Gaussian Moment Matching

First, we define the procedure for Gaussian approximations using moment matching (Gaussian moment matching). Here, we use the discrete-discrete conventional notation for easy reading. The moment-matching-based Gaussian approximation of the joint distribution of $\mathbf{x}_{k-1}$ and $\mathbf{x}_k = \mathbf{g}(\mathbf{x}_{k-1}) + \boldsymbol{q}_{k-1}$, where $\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1} \sim \mathcal{N}(\boldsymbol{m}_{k-1}, \boldsymbol{P}_{k-1})$, is given by

$$\begin{pmatrix} \mathbf{x}_{k-1} \\ \mathbf{x}_k \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \boldsymbol{m}_{k-1} \\ \boldsymbol{m}_k^- \end{pmatrix}, \begin{pmatrix} \boldsymbol{P}_{k-1} & \boldsymbol{A} \\ \boldsymbol{A}^T & \boldsymbol{P}_k^- \end{pmatrix} \right), \tag{2.19}$$

where

$$\begin{aligned} \mathbf{m}_k^- &= \mathbb{E}[\mathbf{x}_k] = \int \mathbf{g}(\mathbf{x}_{k-1}) \mathcal{N}(\boldsymbol{m}_{k-1}, \boldsymbol{P}_{k-1}) d\mathbf{x}_{k-1}, \\ \mathbf{P}_k^- &= Cov[\mathbf{x}_k] = \int (\mathbf{g}(\mathbf{x}_{k-1}) - \mathbf{m}_k^-)(\mathbf{g}(\mathbf{x}_{k-1}) - \mathbf{m}_k^-)^T \mathcal{N}(\boldsymbol{m}_{k-1}, \boldsymbol{P}_{k-1}) d\mathbf{x}_{k-1} + \mathbf{Q}, \\ \mathbf{A} &= \int (\mathbf{x}_{k-1} - \mathbf{m}_{k-1})(\mathbf{g}(\mathbf{x}_{k-1}) - \mathbf{m}_k^-)^T \mathcal{N}(\boldsymbol{m}_{k-1}, \boldsymbol{P}_{k-1}) d\mathbf{x}_{k-1}. \end{aligned} \tag{2.20}$$

Consequently $\mathbf{x}_k|\mathbf{y}_{1:k-1} \sim \mathcal{N}(\boldsymbol{m}_k^-, \boldsymbol{P}_k^-)$. In the context of Gaussian filters such as the Unscented Kalman Filter (UKF) and Cubature Kalman Filter (CKF), Gaussian Moment Matching is a pivotal concept. This is because the moment matching formulation enables the use of many well-known numerical integration methods to approximate integrals, such as $\mathbf{m}_k^- = \mathbb{E}[\mathbf{x}_k]$, which are part of general Gaussian filtering.

The general Gaussian filter for state space models of form 2.7 is composed of the following steps:

- Prediction steps:

$$\begin{aligned}\mathbf{m}_k^- &= \mathbb{E}[\mathbf{x}_k] = \int \mathbf{g}(\mathbf{x}_{k-1})\mathcal{N}(\boldsymbol{m}_{k-1}, \boldsymbol{P}_{k-1})d\mathbf{x}_{k-1},\\ \mathbf{P}_k^- &= Cov[\mathbf{x}_k] = \int (\mathbf{g}(\mathbf{x}_{k-1}) - \mathbf{m}_k^-)(\mathbf{g}(\mathbf{x}_{k-1}) - \mathbf{m}_k^-)^T\mathcal{N}(\boldsymbol{m}_{k-1}, \boldsymbol{P}_{k-1})d\mathbf{x}_{k-1} + \mathbf{Q},\end{aligned} \tag{2.21}$$

- Update step:

$$\begin{aligned}\boldsymbol{\mu}_k &= \int \mathbf{h}(\mathbf{x})\mathcal{N}(\boldsymbol{m}_k^-, \boldsymbol{P}_k^-)d\mathbf{x}_k,\\ \mathbf{S}_k &= \int (\mathbf{h}(\mathbf{x}_k) - \boldsymbol{\mu}_k)(\mathbf{h}(\mathbf{x}_k) - \boldsymbol{\mu}_k)^T\mathcal{N}(\boldsymbol{m}_k^-, \boldsymbol{P}_k^-)d\mathbf{x}_k + \mathbf{R},\\ \mathbf{C}_k &= \int (\mathbf{x}_k - \boldsymbol{m}_k^-)(\mathbf{h}(\mathbf{x}_k) - \boldsymbol{\mu}_k)^T\mathcal{N}(\boldsymbol{m}_k^-, \boldsymbol{P}_k^-)d\mathbf{x}_k,\\ \mathbf{K}_k &= \mathbf{C}_k\mathbf{S}_k^{-1}\\ \mathbf{m}_k &= \mathbf{m}_k^- + \mathbf{K}_k(\mathbf{y}_k - \boldsymbol{\mu}_k)\\ \mathbf{P}_k &= \mathbf{P}_k^- - \mathbf{K}_k\mathbf{S}_k\mathbf{K}_k^T\end{aligned} \tag{2.22}$$

The methods for approximating an integral such as $\mathbf{m}_k^- = \int \mathbf{g}(\mathbf{x}_{k-1})\mathcal{N}(\boldsymbol{m}_{k-1}, \boldsymbol{P}_{k-1})d\mathbf{x}_{k-1}$ are Monte Carlo integration and sigma-point methods such as unscented transformation and cubature rule [137]. Therefore, the idea behind the UKF and CKF is to form a Gaussian approximation of the filtering distribution:

$$p(\mathbf{x}_k|\mathbf{y}_{1:k}) \approx \mathcal{N}(\mathbf{x}_k|\boldsymbol{m}_k, \boldsymbol{P}_k). \tag{2.23}$$

Where $\mathbf{m}_k$ and $\mathbf{P}_k$ are the mean and covariance, respectively, computed using a recursive algorithm composed of prediction and update steps.

2.3.7 Unscented Kalman filter

The unscented Kalman filter uses the unscented transform to compute the approximate means and covariance in nonlinear filtering problems [69, 137].

UKF Prediction step. Assuming that the filtering distribution of the previous step is Gaussian

$$p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1}) \approx \mathcal{N}(\mathbf{x}_{k-1}|\boldsymbol{m}_{k-1}, \boldsymbol{P}_{k-1}), \tag{2.24}$$

the joint distribution of $\mathbf{x}_{k-1}$ and $\mathbf{x}_k = \mathbf{g}(\mathbf{x}_{k-1}) + \boldsymbol{q}_{k-1}$ can be approximated with the unscented transformation as a Gaussian by

$$p(\mathbf{x}_{k-1}, \mathbf{x}_k|\mathbf{y}_{1:k-1}) \approx \mathcal{N}\left(\begin{pmatrix}\mathbf{x}_{k-1}\\ \mathbf{x}_k\end{pmatrix} | \begin{pmatrix}\boldsymbol{m}_{k-1}\\ \boldsymbol{m}_k^-\end{pmatrix}, \begin{pmatrix}\boldsymbol{P}_{k-1} & \boldsymbol{A}\\ \boldsymbol{A}^T & \boldsymbol{P}_k^-\end{pmatrix}\right). \tag{2.25}$$

Where $\boldsymbol{A}$ and $\boldsymbol{A}^T$ represent the cross-covariance between $\mathbf{x}_{k-1}$ and $\mathbf{x}_k$, and the marginal distribution for $\mathbf{x}_k$ is $\mathcal{N}(\boldsymbol{m}_k^-, \boldsymbol{P}_k^-)$ (see the joint distribution of Gaussian variables in Appendix A.1), which provides the prediction equations. Then, the predicted mean $\boldsymbol{m}_k^-$ and covariance $\boldsymbol{P}_k^-$ of $\mathbf{x}_k$ can be computed by performing the operations described in Algorithm 2.1.

UKF Update step. The joint distribution of $\mathbf{x}_k$ and $\mathbf{y}_k = \mathbf{h}(\mathbf{x}_k) + \boldsymbol{r}_k$ can be approximated with the unscented transformation as Gaussian by

$$p(\mathbf{x}_k, \mathbf{y}_k|\mathbf{y}_{1:k-1}) \approx \mathcal{N}\left(\begin{pmatrix}\mathbf{x}_k\\ \mathbf{y}_k\end{pmatrix} | \begin{pmatrix}\boldsymbol{m}_k^-\\ \boldsymbol{\mu}_k\end{pmatrix}, \begin{pmatrix}\mathbf{P}_k^- & \mathbf{C}_k\\ \mathbf{C}_k^T & \mathbf{B}_k\end{pmatrix}\right), \tag{2.26}$$

Then, based on the conditional distribution of the Gaussian variables (A.1), we have:

$$p(\mathbf{x}_k|\mathbf{y}_k) \sim \mathcal{N}(\mathbf{m}_k^- + \mathbf{C}\mathbf{B}^{-1}(\mathbf{y}_k - \boldsymbol{\mu}_k), \mathbf{P}_k^- - \mathbf{C}\mathbf{B}^{-1}\mathbf{C}^T) = \mathcal{N}(\mathbf{m}_k, \mathbf{P}_k). \tag{2.27}$$

Therefore, for each measurement step k=1, ... ,K, the updated mean $\boldsymbol{m}_k$ and covariance $\boldsymbol{P}_k$ of $\mathbf{x}_k$ can be computed by performing the operations described in Algorithm 2.1.

Algorithm 2.1 UKF Algorithm

Initialization

$\mathbf{P}_{k-1} = \mathbf{P}(0)$

$\mathbf{m}_{k-1} = \mathbf{x}(0)$

Prediction step

1)Form the sigma points S^i_{k-1} of $\mathbf{x}_{k-1} \sim \mathcal{N}(\boldsymbol{m}_{k-1}, \boldsymbol{P}_{k-1})$:

$S^0_{k-1} = \boldsymbol{m}_{k-1}$

$S^i_{k-1} = \boldsymbol{m}_{k-1} + \sqrt{(n+\lambda)}[\sqrt{\mathbf{P}_{k-1}}]_i$

$S^{i+n}_{k-1} = \boldsymbol{m}_{k-1} - \sqrt{(n+\lambda)}[\sqrt{\mathbf{P}_{k-1}}]_i, i = 1, ..., n$

where the $\lambda = \alpha^2(n+\kappa) - n$.

2) Propagate the sigma points through $\chi^i_k = \mathbf{g}(S^i_{k-1})$, i=0, ..., 2n.

3)Compute the $\boldsymbol{m}^-_k$ and $\boldsymbol{P}^-_k$:

$\mathbf{m}^-_k = \sum_{i=0}^{2n} W^m_i \chi^i_k$

$\mathbf{P}^-_k = \sum_{i=0}^{2n} W^c_i (\chi^i_k - \mathbf{m}^-_k)(\chi^i_k - \mathbf{m}^-_k)^T + \mathbf{Q}$

where the weights and are defined by

$W^m_0 = \frac{\lambda}{n+\lambda}$

$W^c_0 = \frac{\lambda}{n+\lambda} + (1 - \alpha^2 + \beta)$

$W^m_i = \frac{1}{2(n+\lambda)}, i = 1, ..., 2n$

$W^c_i = \frac{1}{2(n+\lambda)}, i = 1, ..., 2n.$

Update step

4)Form the sigma points

$\bar{S}^0_k = \boldsymbol{m}^-_k$

$\bar{S}^i_k = \boldsymbol{m}^-_k + \sqrt{(n+\lambda)}[\sqrt{\mathbf{P}^-_k}]_i$

$\bar{S}^{i+n}_k = \boldsymbol{m}^-_k - \sqrt{(n+\lambda)}[\sqrt{\mathbf{P}^-_k}]_i, i = 1, ..., n$

5) Propagate the sigma points $\psi^i_k = \mathbf{h}(\bar{S}^i_k)$, i=0, ..., 2n.

6) Compute $\boldsymbol{\mu}_k$, $\mathbf{B}_k$, and $\mathbf{C}_k$:

$\boldsymbol{\mu}_k = \sum_{i=0}^{2n} W^m_i \psi^i_k$

$\mathbf{B}_k = \sum_{i=0}^{2n} W^c_i (\psi^i_k - \boldsymbol{\mu}_k)(\psi^i_k - \boldsymbol{\mu}_k)^T + \mathbf{R}$

$\mathbf{C}_k = \sum_{i=0}^{2n} W^c_i (\bar{S}^i_k - \mathbf{m}^-_k)(\psi^i_k - \boldsymbol{\mu}_k)^T$

7) Compute $\mathbf{K}_k$, $\mathbf{m}_k$, and $\mathbf{P}_k$, conditional on $\mathbf{y}_k$:

$\mathbf{K}_k = \mathbf{C}_k \mathbf{B}^{-1}_k$

$\mathbf{m}_k = \mathbf{m}^-_k + \mathbf{K}_k[\mathbf{y}_k - \boldsymbol{\mu}_k]$

$\mathbf{P}_k = \mathbf{P}^-_k - \mathbf{K}_k \mathbf{B}_k \mathbf{K}^T_k$

2.3.8 Cubature Kalman filter

CKF uses the spherical cubature rule to approximate the Gaussian filter [6]. The main idea is to use the third-order spherical cubature integration rule in Gaussian filter equations. It is important to point out that CKF is a special case of the UKF with $\alpha = 1$, $\beta = 0$, and $\kappa = 0$ (the mean weight becomes zero with these choices). The additive form of the CKF is shown in Algorithm 2.2 .

Algorithm 2.2 CKF Algorithm

Initialization

$\mathbf{P}_{k-1} = \mathbf{P}(0)$

$\mathbf{m}_{k-1} = \mathbf{x}(0)$

Prediction step

1) Form the sigma points as

$S_{k-1}^i = \boldsymbol{m}_{k-1} + \sqrt{\mathbf{P}_{k-1}}\xi^i, i = 1, ..., 2n$

where the unit sigma points are defined as

$$\xi^i = \begin{cases} \sqrt{n}\mathbf{e}_i, & i = 1, \cdots, n, \\ -\sqrt{n}\mathbf{e}_{i-n}, & i = n+1, \cdots, 2n. \end{cases}$$

2) Propagate the sigma points through $\chi_k^i = \mathbf{g}(S_{k-1}^i)$, i=1, $\cdots$, 2n.

3) Compute $\boldsymbol{m}_k^-$ and $\boldsymbol{P}_k^-$:

$\mathbf{m}_k^- = 1/2n \sum_{i=1}^{2n} \chi_k^i$

$\mathbf{P}_k^- = 1/2n \sum_{i=1}^{2n} (\chi_k^i - \mathbf{m}_k^-)(\chi_k^i - \mathbf{m}_k^-)^T + \mathbf{Q}$

Update step

4)Form the sigma points

$\bar{S}_k^i = \boldsymbol{m}_k^- + \sqrt{\mathbf{P}_k^-}\xi^i, \quad i = 1, ..., 2n$

where the unit sigma points are defined as

$$\xi^i = \begin{cases} \sqrt{n}\mathbf{e}_i, & i = 1, \cdots, n, \\ -\sqrt{n}\mathbf{e}_{i-n}, & i = n+1, \cdots, 2n. \end{cases}$$

5) Propagate the sigma points $\psi_k^i = \mathbf{h}(\bar{S}_k^i)$, i=1, ..., 2n.

6) Compute $\boldsymbol{\mu}_k$, $\mathbf{B}_k$, and $\mathbf{C}_k$:

$\boldsymbol{\mu}_k = 1/2n \sum_{i=0}^{2n} \psi_k^i$

$\mathbf{B}_k = 1/2n \sum_{i=0}^{2n} (\psi_k^i - \boldsymbol{\mu}_k)(\psi_k^i - \boldsymbol{\mu}_k)^T + \mathbf{R}$

$\mathbf{C}_k = 1/2n \sum_{i=0}^{2n} (\bar{S}_k^i - \mathbf{m}_k^-)(\psi_k^i - \boldsymbol{\mu}_k)^T$

7) Compute $\mathbf{K}_k$, $\mathbf{m}_k$, and $\mathbf{P}_k$, conditional on $\mathbf{y}_k$:

$\mathbf{K}_k = \mathbf{C}_k \mathbf{B}_k^{-1}$

$\mathbf{m}_k = \mathbf{m}_k^- + \mathbf{K}_k[\mathbf{y}_k - \boldsymbol{\mu}_k]$

$\mathbf{P}_k = \mathbf{P}_k^- - \mathbf{K}_k \mathbf{B}_k \mathbf{K}_k^T$

2.4 Joint and Dual Nonlinear Kalman Estimators

JEKF (Joint Extended Kalman Filter)

JEKF is a Bayesian filter-based approach for joint estimation in nonlinear dynamical systems. It concatenates states x_i and parameters θ of a process model into a single joint state vector. The state variables vector $\mathbf{x}(t)$ in JEKF is extended as:

$$\mathbf{x}(t) = [x_1, x_2, ..., x_n, \theta_1, ..., \theta_n]^T. \tag{2.28}$$

In JEKF, the learning involves both states x_i and parameters θ_i of a discrete-time nonlinear system (e.g., UMM). It corrects system states and model parameters simultaneously based on observed noisy signals $\mathbf{y}_k$. JEKF is recognized for parameter evolution, where parameters are treated as random variables with noise added at each timestep:

$$\boldsymbol{\theta}(t_k) = \boldsymbol{\theta}(t_{k-1}) + \text{noise}, \tag{2.29}$$

This approach is efficient for updating process model parameters, especially when near optimal parameters for specific conditions, such as similar biomanufacturing conditions. In JEKF, parameter estimation refers to this ongoing evolution of parameters [7].

JUKF (Joint Unscented Kalman Filter)

JUKF extends the UKF to joint estimation scenarios. Unlike JEKF, JUKF does not linearize the process and measurement models but instead uses a deterministic sampling technique (the unscented transform) to capture the mean and covariance estimates. This makes JUKF more accurate in capturing the true state of a nonlinear system. JUKF also concatenates states and parameters into a single state vector and simultaneously estimates them using the unscented transform and Kalman filter equations.

JCKF (Joint Cubature Kalman Filter)

Similar to JUKF, JCKF is designed for joint estimation of states and parameters in nonlinear systems. JCKF employs the cubature Kalman filter, which uses cubature rules to approximate the integrals in the state and covariance propagation. This approach avoids linearization errors and is computationally more efficient than JUKF. JCKF, like the other joint estimation methods, concatenates states and parameters into a single state vector for simultaneous estimation.

2.4.1 DEKF algorithm

This Dual EKF algorithm is a robust method from the literature for simultaneous state and parameter estimation in systems where model dynamics are influenced by unknown or time-varying parameters. It iteratively predicts and corrects estimates based on new measurements, adapting over time to improve accuracy. This approach is widely used in control systems, robotics, and areas requiring real-time estimation under uncertain conditions. Here is a detailed breakdown and description of each step within the DEKF Algorithm2.3:

Initialization: 1 - State Initialization: The initial state estimate, denoted as $\mathbf{x}_{0|0}$, is set to the expected value $E(\mathbf{x}_0)$. 2 - State Covariance Initialization: The initial covariance matrix $\mathbf{P}_{\mathbf{x}_{0|0}}$ of the state estimate is computed. It represents the expected error in the initial state estimate. 3 - Parameter Covariance Initialization: Similarly, $\mathbf{P}_{\boldsymbol{\theta}_0}$ is initialized, representing the covariance of the initial parameter estimates, indicating the uncertainty or variability in these initial parameter estimates.

Prediction Step for Parameters: 1 - The parameters $\boldsymbol{\theta}_{k|k-1}$ at step k given information up to step $k-1$ are assumed to be equal to the estimates from the previous step, reflecting a model where parameters do not evolve over time. 2 - The covariance of the parameter estimates, $\mathbf{P}^{\boldsymbol{\theta}}_{k|k-1}$, is updated by scaling the previous covariance $\mathbf{P}^{\boldsymbol{\theta}}_{k-1|k-1}$ by a factor λ^{-}, which adjusts the estimate's uncertainty.

Prediction Step for States: 1 - The state prediction $\mathbf{x}_{k|k-1}$ is calculated using the state transition function $\mathbf{g}$, which predicts the next state based on the previous state and current parameter estimates. 2 - The covariance of the state prediction, $\mathbf{P}^{\mathbf{x}}_{k|k-1}$, is updated to reflect the uncertainty due to the process noise and the propagation of the previous state's uncertainty.

Update Step for States: 1 - The Kalman Gain for the states, $\mathbf{K}^{\mathbf{x}}_k$, is computed, which determines how much the measurements taken at time k, denoted by ϵ_k, should impact the state estimate. 2 - The state estimate $\mathbf{x}_{k|k}$ is updated using the Kalman Gain and the measurement residual ϵ_k. 3 - The covariance of the updated state estimate, $\mathbf{P}^{\mathbf{x}}_{k|k}$, is calculated, indicating reduced uncertainty after incorporating the measurement.

Update Step for Parameters: 1 - Similarly to the states, a Kalman Gain for the parameters, $\mathbf{K}^{\boldsymbol{\theta}}_k$, is computed. 2 - The parameter estimates $\boldsymbol{\theta}_{k|k}$ are updated using this Kalman Gain and the measurement residual. 3 - The covariance of the updated parameter estimates, $\mathbf{P}^{\boldsymbol{\theta}}_{k|k}$, is updated to reflect the new level of uncertainty post-measurement.

Additional Mathematical Details: i) ϵ_k represents the innovation or measurement residual, which is the difference between the actual measurement and the predicted measure-

ment. ii) Various Jacobian matrices ($\mathbf{F}_k$, $\mathbf{H}_k$, $\mathbf{H}_k^{\boldsymbol{\theta}}$) are used to linearize the nonlinear state transition and measurement functions around the current estimates, which is a key aspect of the EKF methodology.

Algorithm 2.3 DEKF Algorithm

1)Initialize

$\mathbf{x}_{0|0} = E(\mathbf{x}_0)$

$\mathbf{P}_{\mathbf{x}_{0|0}} = E[(\mathbf{x}_0 - \mathbf{x}_{0|0})(\mathbf{x}_0 - \mathbf{x}_{0|0})^T]$

$\mathbf{P}_{\boldsymbol{\theta}_0} = E[(\boldsymbol{\theta}_0 - \boldsymbol{\theta}_{0|0})(\boldsymbol{\theta}_0 - \boldsymbol{\theta}_{0|0})^T]$

2) Prediction step of parameters

$\boldsymbol{\theta}_{k|k-1} = \boldsymbol{\theta}_{k-1|k-1}$

$\mathbf{P}^{\boldsymbol{\theta}}_{k|k-1} = \lambda^{-}\mathbf{P}^{\boldsymbol{\theta}}_{k-1|k-1}$

3) Prediction step of states

$\mathbf{x}_{k|k-1} = \mathbf{g}(\mathbf{x}_{k-1|k-1}, \boldsymbol{\theta}_{k|k-1})$

$\mathbf{P}^{\mathbf{x}}_{k|k-1} = \mathbf{F}_k\mathbf{P}^{\mathbf{x}}_{k-1|k-1} + \mathbf{P}^{\mathbf{x}}_{k-1|k-1}\mathbf{F}_k^T + \mathbf{Q}$

4) Update step of states

$\mathbf{K}^{\mathbf{x}}_k = \mathbf{P}^{\mathbf{x}}_{k|k-1}\mathbf{H}_k^T(\mathbf{H}_k\mathbf{P}^{\mathbf{x}}_{k|k-1}\mathbf{H}_k^T + \mathbf{R})^{-}$

$\mathbf{x}_{k|k} = \mathbf{x}_{k|k-1} + \mathbf{K}^{\mathbf{x}}_k\epsilon_k$

$\mathbf{P}^{\mathbf{x}}_{k|k} = (\mathbf{I} - \mathbf{K}^{\mathbf{x}}_k\mathbf{H}_k)\mathbf{P}^{\mathbf{x}}_{k|k-1}$

5) Update step of parameters

$\mathbf{K}^{\boldsymbol{\theta}}_k = \mathbf{P}^{\boldsymbol{\theta}}_{k|k-1}\mathbf{H}_k^{\boldsymbol{\theta}T}(\mathbf{H}^{\boldsymbol{\theta}}_k\mathbf{P}^{\boldsymbol{\theta}}_{k|k-1}\mathbf{H}_k^{\boldsymbol{\theta}T} + \mathbf{R}^{\boldsymbol{\theta}})^{-}$

$\boldsymbol{\theta}_{k|k} = \boldsymbol{\theta}_{k|k-1} + \mathbf{K}^{\boldsymbol{\theta}}_k\epsilon_k$

$\mathbf{P}^{\boldsymbol{\theta}}_{k|k} = (\mathbf{I} - \mathbf{K}^{\boldsymbol{\theta}}_k\mathbf{H}^{\boldsymbol{\theta}}_k)\mathbf{P}^{\boldsymbol{\theta}}_{k|k-1}$

where $\epsilon_k = \mathbf{y}_k - \mathbf{h}(\mathbf{x}_{k|k-1}) = \mathbf{y}_k - \bar{\mathbf{H}}\mathbf{x}_{k|k-1}$;

$\bar{\mathbf{H}} = [1\ 0\ ...\ 0]$

$\mathbf{F}_k = \frac{\partial \mathbf{g}(\mathbf{x}, \boldsymbol{\theta}_{k|k-1})}{\partial \mathbf{x}} \mid_{\mathbf{x}_{k-1|k-1}}$

$\mathbf{H}_k = \frac{\partial \mathbf{h}(\mathbf{x})}{\partial \mathbf{x}} \mid_{\mathbf{x}_{k|k-1}} = \bar{\mathbf{H}}$

$\mathbf{H}^{\boldsymbol{\theta}}_k = -\frac{\partial \epsilon_k}{\partial \boldsymbol{\theta}} = \frac{\partial \mathbf{h}(\mathbf{x}_{k|k-1})}{\partial \boldsymbol{\theta}} = \bar{\mathbf{H}}\frac{\partial \mathbf{x}_{k|k-1}}{\partial \boldsymbol{\theta}} \mid_{\boldsymbol{\theta}_{k|k-1}}$

2.5 Biomanufacturing conditions for fast and low-cost bioprocess monitoring

The biopharmaceutical industry aims fast and low-cost bioprocess monitoring approaches, see Figure 2.3. Fast (efficient) and low-cost (cost-effective) bioprocess monitoring can

be defined as a set of methods designed to track and analyze the parameters and states (critical process parameters and quality attributes) of biomanufacturing in real time to minimize both capital and operational expenses [62,63,113]. In this context, the word fast, means to perform real-time estimations of all system state variables. Because the current approaches for bioprocess monitoring rely on offline measurements and it take hours or days to deliver results about nutrients and production formation. The term "low-cost", in this context, means to reduce the use of multiple devices to perform offline measurements, consequently, reducing the operational cost of the monitoring process. For example, using only one device to provide online measurements of cell growth and estimate the others state variables based on this device, see Figure 2.3.

However, the biomanufacturing conditions for fast and low-cost bioprocess monitoring are challenge. They include UMM [74], and a limited amount of data [110]. However, before starting the description of the biomanufacturing conditions, we formally analyze UMM (defining unshared parameters and weak and strong terms/variables) and the difficulties of biopharm to generate large amount of data to development of soft sensors.

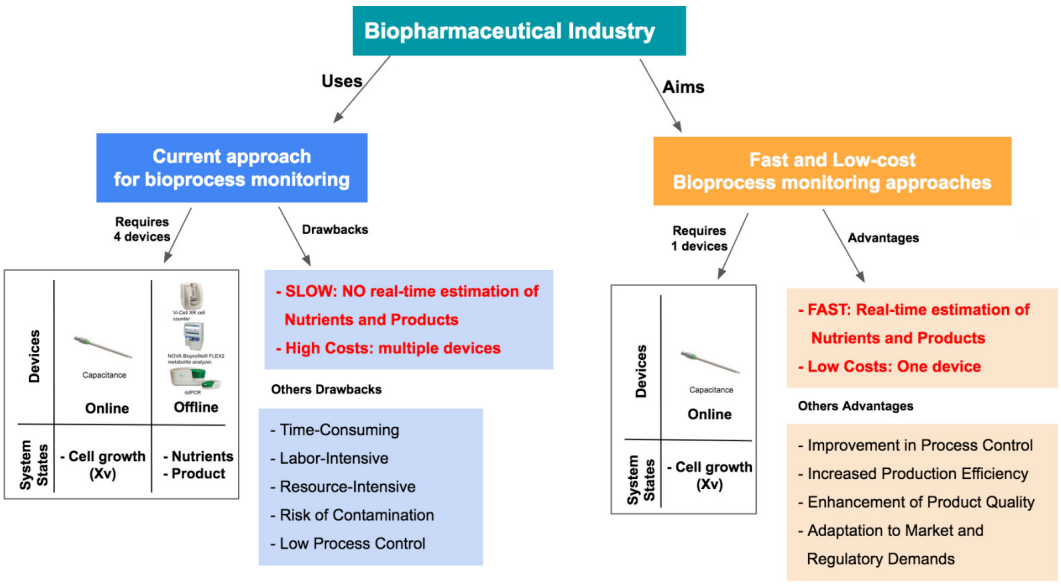

Figure 2.3: The biopharmaceutical industry aims fast and low-cost bioprocess monitoring approaches.

2.5.1 Analysis of Unstructured Mechanistic Models

As discussed before, UMMs can be divided into two groups. Specific UMMs provide a detailed and customized description of a particular bioprocess, while Generic UMMs (formed with only unshared parameters) offer a more generalized framework that can be adapted to various bioprocesses with the help of additional computational tools.

Unshared and shared parameters

Unshared parameters are parameters used only in one term of an ODE and not used by other ODEs of the same UMM. The UMM presented in Section 2.1.1 is the case of an ODE system with only **unshared parameters**. On the other hand, the SMM of Section 2.1.3 is a case of ODE system with only **shared parameters**. For example, the k_3 is used in different ODEs of this system. The UMM case 2 of Section 2.1.2 is a case of system with unshared and shared parameters.

Weak and strong terms

A *weak term* is a term of an ODE with a low percentage of variables of state variable vector, and a *"strong term"* is one with a high percentage of variables of state variable vector. The three mechanistic models of Sections 2.1.1, 2.1.2, and 2.1.3 are examples of ODE systems with weak and strong terms. The UMM of Section 2.1.1 is example of ODE system with only terms that could be considered weak, because they have low percentage of variables that compose the state variable vector. For example, let assume the following state variable vector $\mathbf{x}(t)_{case1} = [Xv, N, MP, \mu_{Xv}, \mu_N, \mu_{mp}]$ with six elements for the UMM of Section 2.1.1. We have that all terms of this UMM has a 1/3 of the state variable vector. On another hand, the UMM of Section 2.1.2 is example of ODE system with terms that could be considered weak and strong. For example, let consider the following state variable vector for UMM case 2 (Section 2.1.2),

$$\mathbf{x}(t)_{case2} = [X_v, fgr, X_d, GLC, GLN, LAC, ASN, ASP, ALA, AMM, Mab, \mu_{max}, k_{31}, k_d] \tag{2.30}$$

we have that the first term of equation $\frac{dX_v}{dt}$ is the strongest term in this system, since it has 7/14 of state variable vector, and the first term of the equation $\frac{dMab}{dt}$ as the weakest term, since it has 2/14 of state variable vector.

Weak and strong 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. The three mechanistic model cases (Sections 2.1.1, 2.1.2, and 2.1.3) presented above are examples of the ODE system with weak and strong variables. The variable MP in the UMM case1 (Section 2.1.1)and Mab in UMM case2 (Section 2.1.2)are examples of weak variable. In these cases, the first-order partial derivatives of all functions with respect to these two variables are equal to zero and this reflects that the variable has a column with zeros in the jacobian $\mathbf{J}_t^{\mathbf{g}}$. On another hand, in the UMM of Sections 2.1.1) and 2.1.2, Xv is an example of strong variable.

2.5.2 Biopharma is a data-limited industry

The biopharma industry is particularly data-limited regarding actively generated data, as each experiment and analysis is resource-intensive, limiting the number of experiments and corresponding data that can be generated [110]. Obtaining data in biopharmaceutical studies is costly and time-consuming due to the necessity of conducting biological experiments, which require significant time and resources. Additionally, these data are often linked to strategic products and confidential information of pharmaceutical companies, making it challenging to publicly share them with the research community. While the industry possesses substantial "historical data," these were not collected for AI/ML training purposes and contain biases in the design spaces explored and information recorded, affecting overall data quality. This lack of publicly available benchmark datasets poses a major challenge to the reproducibility of machine learning (ML), such as Data-Driven Models (DDM) for the development of soft sensor for fast and low-cost bioprocess monitoring. Without large datasets, it becomes difficult to assess the effectiveness of DDM or perform comparative analyses between different approaches. This is because DDM are suitable for learning patterns in data from high-dimensional design spaces with complex non-linear interactions [74, 110]. Then, the effective use of DDM requires high-quality data in significant quantities to develop useful and applicable models for biopharmaceutical applications. Therefore, the bioprocess data are expensive to obtain, requiring time-consuming, labor-intensive, and costly biological experiments [74]. Furthermore, these data are associated with strategic products and business secrets, making them unlikely to be publicly shared, preventing the explosive growth of DDM in biopharmaceuticals compared to fields like natural language processing and computer vision. However, it is important to point out that Hybrid dynamic models and NKE with UMM can overcome this need for large amount of

data and enable the development of soft sensor for fast and low-cost bioprocess monitoring.

Because traditional machine learning models (Data-Driven Models) typically require large datasets, and the biopharma industry faces significant limitations in generating such data, this thesis does not focus on comparing the proposed models against DDMs. The proposed models, including NKE with UMM, are designed to operate effectively with smaller datasets and are tailored to the data-limited environment of the biopharma industry. Hence, a comparison with traditional ML approaches, which would require vast data resources, is not included.

2.5.3 Challenge Conditions for NKE development

In Section 3.2, we present the theoretical analysis of the limitations of joint and dual NKE for fast and low-cost bioprocess monitoring with biomanufacturing conditions presented here. Given that, the biomanufacturing conditions for fast and low-cost bioprocess monitoring with a NKE are:

1. A state variables vector defined by $\mathbf{x} = [x_{msv}, x_2, \ldots, x_n]$.
2. x_{msv} as the unique measured state variable (MSV).
3. A system of nonlinear differential equations representing a UMM of the form:

$$\mathbf{g}(\mathbf{x}, \boldsymbol{\theta}) = \begin{array}{ll} \frac{dx_{msv}}{dt} & = f_1(x_{msv}, x_2, \ldots, x_{n-1}, \theta_1, \theta_2, \ldots, \theta_m), \\ \frac{dx_2}{dt} & = f_2(x_{msv}, x_2, \ldots, x_{n-1}, \theta_1, \theta_2, \ldots, \theta_m), \\ & \vdots \\ \frac{dx_n}{dt} & = f_n(x_{msv}, \theta_{up}) \end{array} \tag{2.31}$$

   where x_{msv} and $x_2, \ldots, x_n$ are the variables of the system, $f_1, f_2, \ldots, f_n$ are the functions defining the system represented by $\mathbf{g}(\cdot)$, and $\theta_1, \theta_2, \ldots, \theta_m$ are the parameters of the system, and θ_{up} an unshared parameter.
4. $\mathbf{R}$ as measurement noise variance of x_{msv}.
5. θ_{up} as the unshared parameter (UP) to be evolved (estimated) in real-time. It is important to note that UP is part of a weak term related to a weak variable (x_n).

6. $\mathbf{P}(0)$ and $\mathbf{Q}$ with uncorrelated elements. They are defined as

$$\begin{aligned}\mathbf{P}(0) &= Diag([P_{x_{msv},x_{msv}}, P_{x_2,x_2}, \cdots, P_{n,n}, P_{\theta_{up},\theta_{up}}]) \\ \mathbf{Q} &= Diag([Q_{x_{msv},x_{msv}}, Q_{x_2,x_2}, \cdots, Q_{n,n}, Q_{\theta_{up},\theta_{up}}]).\end{aligned} \tag{2.32}$$

In biomanufacturing, due to the limited amount of data, it is very common to assume $\mathbf{P}$ and $\mathbf{Q}$ with uncorrelated elements in NKE applications [63, 117, 120, 166]. This assumption means that the error covariance matrices $\mathbf{P}$ and $\mathbf{Q}$ are diagonal, with the diagonal elements being the noise variances ($\mathrm{P}_{i,i} \neq 0$ and $\mathrm{Q}_{i,i} \neq 0$) and off-diagonal elements equal to zero ($\mathrm{P}_{i,j} = 0$ and $\mathrm{Q}_{i,j} = 0$).

2.6 Comparison with State-of-the-Art

Recent studies in the state-of-the-art focus on bioprocess monitoring techniques and can be considered similar to the present work. These studies primarily involve implementing soft sensors for monitoring specific bioprocesses, offering insights into various approaches and their effectiveness.

Andrea Tuveri [155] presented the important role that state estimators have in the next-generation biotech industry. She advances bioprocess monitoring by developing and applying Bayesian state estimators to address uncertainties and plant-model mismatches. This research implements and compares constrained EKF, UKF, and Moving Horizon Estimators for fed-batch cultivation of *Corynebacterium glutamicum*. Methods to handle system dynamics uncertainties and the integration of infrequent and delayed measurements significantly enhance the accuracy and reliability of bioprocess monitoring systems.

Arnas Survyla [149] proposed the development of soft sensors based on bioreactor exhaust gas analysis. This approach increased productivity through precise and reliable estimation algorithms for biomass concentration, highlighting the potential for improved process efficiency and control. The study focuses on three key cultivation variables that cannot be measured online: specific cell growth rate, viable biomass concentration, and target product concentration. The aim is to increase cultivation productivity by using precise, reliable estimation algorithms based on exhaust gas analysis. Key findings indicate that in the early cultivation phases, cell maintenance has minimal impact on estimation accuracy; the synthesis efficiency of target products is directly tied to the growth rate at induction; bioprocess state assessments based on oxygen consumption rates are reliable and

repeatable, unaffected by E. coli metabolism; and cell culture age significantly improves the precision of viable cell concentration estimations.

Abdolrahim Yousefi-Darani [169] proposed the application of soft sensor based on NKE for yeast fermentation supervision. This work demonstrates the capability of NKEs in accurately monitoring and controlling fermentation processes, contributing to more robust bioprocess management [169]. In addition, this work exemplifies the effective use of a single measured state variable to estimate other critical variables in bioprocess monitoring through the application of NKE for yeast fermentation supervision. By leveraging the measurement of ethanol concentration via a gas sensor array, the study employs EKF and UKF, to predict essential parameters such as glucose concentration, biomass concentration, and growth rates. This approach demonstrates the capability of NKEs to integrate sparse, real-time data with a robust mathematical model to infer unmeasured variables accurately. The research highlights the reliability and precision of these soft sensors in managing and optimizing the fermentation process, thus contributing to more efficient and robust bioprocess management by reducing dependency on multiple physical sensors and enabling continuous process control with minimal manual intervention.

Vincent Brunner [18] proposed the application of soft sensors for Pichia pastoris bioprocesses, integrating process knowledge and sensor validation. This integration allows for more accurate monitoring and control, ensuring high-quality production outcomes. The methylotrophic yeast *Pichia pastoris* is a well-established organism for bioprocessing, but there remains potential for optimization in the areas of monitoring, control, and automation. Further, this bioprocess shows time-variant and non-linear behavior. This must be taken into account when developing soft sensors. Therefore, the proposed approaches represent important building blocks to fill the gaps between uncertain process data and knowledge in the development of soft sensors.

The studies described above do not deal with the biomanufacturing conditions for fast and cost-effective bioprocess monitoring. Contrary to these studies, the present work focuses on the development of new approaches to support the implementation of NKEs with UMM for fast and cost-effective bioprocess monitoring. While earlier works have focused on monitoring specific bioprocesses, such as the application of Bayesian state estimators for handling system uncertainties (Tuveri [155]), the precise estimation of biomass concentration using bioreactor exhaust gas analysis (Survyla [149]), or the use of soft sensors for yeast fermentation supervision with NKEs (Yousefi-Darani [169]), this thesis aims to extend the capabilities of NKEs beyond specific cases of biomanufacturing conditions. This thesis addresses the issues of integrate NKEs with both specific and generic UMMs under a range of biomanufacturing conditions. The integration of NKEs with both specific and generic UMMs allows the proposed methods to handle dynamic biomanufacturing scenarios

and highly nonlinear bioprocesses.

Chapter 3

Joint NKE with SANTO

3.1 Overview

This Chapter presents the proposed approach to address the RQ1. JEKF, JUKF, and JCKF face challenges in biomanufacturing applications such as fast and low-cost bioprocess monitoring, particularly with UMM that consists of ODEs with unshared parameters, weak variables, and weak terms. This research identifies a failure case where JEKF cannot simultaneously estimate unshared parameters and state variables due to the use of uncorrelated initial error covariance matrices and the presence of a single measured state variable. To address this, we propose a solution named SANTO (Specific initiAl coNdiTiOn) for the matrix Ricatti differential equation (MRDE), inspired by regularization techniques. SANTO modifies the initial state error covariance between the measured state variable and an unshared parameter, preventing the Kalman gain from being zero and ensuring robust parameter evolution. This solution supports real-time monitoring in biomanufacturing processes with unknown mechanisms, aiding the biopharmaceutical industry's transition to biomanufacturing 4.0 by improving product quality, optimizing operations, and reducing costs. The effectiveness of SANTO is demonstrated through theoretical analysis and empirical validation using synthetic dataset. SANTO is also applicable to JUKF and JCKF.

The main contributions of this chapter are:

- **Theoretical Proof**: We provide theoretical proof of joint and dual EKF limitations when acting as an unshared parameter estimator for fast and low-cost bioprocess monitoring. To our knowledge, this is the first study to formally report this limitation

case, which includes a vocabulary describing UMM. This is also the first study to formally define unshared parameters and weak and strong terms/variables of an ODE of UMM.

- **SANTO Approach**: We propose the SANTO approach to enable joint NKE (EKF, UKF, and CKF) with a specific UMM for fast and low-cost bioprocess monitoring under a set of similar biomanufacturing conditions. SANTO approach preserver the advantages of joint NKE, such as real-time parameter estimator, easy implementation, and low computational requirements ideal for implementation in micro-controllers (and/or minicomputers and/or microprocessors).

- **Empirical Validation**: Empirical results, using a synthetic dataset of monoclonal Antibody (mAb) productions with different challenge sceneries, show the effectiveness of SANTO in improving the performance of joint NKE (EKF, UKF, and CKF). The code and data used in this work are available on GitHub repository [1] to facilitate reproducibility.

3.2 Limitations of Joint and Dual NKE for fast and Low-cost bioprocess monitoring

This Section presents the theoretical analysis of the limitations of joint and dual NKE for fast and low-cost bioprocess monitoring, where joint and dual NKE cannot perform unshared parameter evolution simultaneously with the states.

Biomanufacturing conditions in the context of NKE development

The following conditions are prevalent in biomanufacturing and should be taken into consideration while developing JEKF applications for this area:

- **ODEs of UMM with unshared parameters.** As described before this parameter type is commonly used in ODE to model the dynamic of product formation in biomanufacturing [41, 78, 107].

[1] https://github.com/cristovaoiglesias/NKEs-SANTO

- **P and Q with uncorrelated elements.** The **Q** constant and with uncorrelated elements is used only to build the MRDE, and the **P** with uncorrelated elements can be used to build an MRDE and as an initial condition of MRDE (the initial predicted state error covariance $\mathbf{P}(\mathrm{t}=0)$). This assumption raises two scenarios:

  1. The use of **P** with uncorrelated elements to build the MRDE (Equation 2.15) and $\mathbf{P}(\mathrm{t}=0)$ with uncorrelated elements as an initial condition. When **P** with uncorrelated elements is used to build the MRDE, the ODEs of MRDE are based only on noise variance of $\mathrm{P}_{i,i}$ and $\mathrm{Q}_{i,i}$ and elements of Jacobian $\mathbf{J}_t^{\mathbf{g}}$. It is important to point out that depending on the partial derivative, the ODE to predict a state error covariance can be time-invariant $\frac{dP_{i,j}(t_{k|k-1})}{dt} = 0$.
  2. The use of **P** with correlated elements to build the MRDE (Equation 2.15) and $\mathbf{P}(\mathrm{t}=0)$ with uncorrelated elements as an initial condition . This means that the ODE of MRDE can be composed of off-diagonal elements of **P**, and it can reduce the number of the time-invariant ODE to predict a state error covariance between two state variables.

- **ODEs of UMM with weak terms.** A strong term contributes more than a weak term to compute the predicted state error covariance $\mathbf{P}(t_{k|k-1})$. Many elements of Jacobian $\mathbf{J}_t^{\mathbf{g}}$ result from the partial derivation of a strong term. In the context of JEKF, we have that a strong term in an UMM contributes more than a weak term to compute of predicted state error covariance $\mathbf{P}(t_{k|k-1})$. Since, many elements of Jacobian $\mathbf{J}_t^{\mathbf{g}}$, results from the first-order partial derivatives of strong term with respect to the variables of state variable vector. For example, let consider the following:

  – State variable vector $\mathbf{x}(t) = [x_1, x_2, x_3, x_4]$
  – An UMM composed of an ODE with a strong term by the function $S(x_1, x_2, x_3, x_4)$ and three ODEs composed of waek terms represented by the function $W_1(x_1)$, $W_2(x_2)$ and $W_3(x_3)$.
  – MRDE (Equation 2.15).
  – **P** and **Q** uncorrelated for the $\mathbf{x}(t)$,

$$\mathbf{P} = \begin{bmatrix} P_{1,1} & 0 & 0 & 0 \\ 0 & P_{2,2} & 0 & 0 \\ 0 & 0 & P_{3,3} & 0 \\ 0 & 0 & 0 & P_{4,4} \end{bmatrix}, \mathbf{Q} = \begin{bmatrix} Q_{1,1} & 0 & 0 & 0 \\ 0 & Q_{2,2} & 0 & 0 \\ 0 & 0 & Q_{3,3} & 0 \\ 0 & 0 & 0 & Q_{4,4} \end{bmatrix}. \tag{3.1}$$

Given this we have the following Jacobian

$$\textbf{Jacobian}(S, W_1, W_2, W_3) = \begin{bmatrix} \frac{\partial S}{\partial x_1} & \frac{\partial S}{\partial x_2} & \frac{\partial S}{\partial x_3} & \frac{\partial S}{\partial x_4} \\ \frac{\partial W_1}{\partial x_1} & 0 & 0 & 0 \\ 0 & \frac{\partial W_2}{\partial x_2} & 0 & 0 \\ 0 & 0 & \frac{\partial W_3}{\partial x_3} & 0 \end{bmatrix} \tag{3.2}$$

and the following MRDE to compute the Predicted state error covariance $\mathbf{P}(t_{k/k-1})$ from t_{k-1} to t_k,

$$\frac{d\mathbf{P}(t)}{dt} = \begin{bmatrix} Q_{1,1} + 2.P_{1,1}.\frac{\partial S}{\partial x_1} & P_{1,1}.\frac{\partial W_1}{\partial x_1} + P_{2,2}.\frac{\partial S}{\partial x_2} & P_{3,3}.\frac{\partial S}{\partial x_3} & P_{4,4}.\frac{\partial S}{\partial x_4} \\ P_{1,1}.\frac{\partial W_1}{\partial x_1} + P_{2,2}.\frac{\partial S}{\partial x_2} & Q_{2,2} & P_{2,2}.\frac{\partial W_2}{\partial x_2} & 0 \\ P_{3,3}.\frac{\partial S}{\partial x_3} & P_{2,2}.\frac{\partial W_2}{\partial x_2} & Q_{3,3} & P_{3,3}.\frac{\partial W_3}{\partial x_3} \\ P_{4,4}.\frac{\partial S}{\partial x_4} & 0 & P_{3,3}.\frac{\partial W_3}{\partial x_3} & Q_{4,4} \end{bmatrix}. \tag{3.3}$$

Then, we can see that S contribute with 7 partial derivatives and the others functions with 2 partial derivative only each one. If we consider the MRDE formed with $\mathbf{P}$ correlated, we have S contributing with 32 partial derivatives and the others functions with 8 partial derivative only each one. It is important point out that the element 0 in the MRDE (Equation 3.3) represents an time invariant ODE $\frac{dP_{x_2,x_4}(t_{k/k-1})}{dt} = 0$ to predicts the covariance between x_4 and x_2 when solved from t_{k-1} to t_k.

- **ODEs of UMM with weak variables.** In the Jacobian $\mathbf{J}_t^{\mathbf{g}}$, the first-order partial derivatives of all functions with respect to a *weak variable* are equal to zero. Consequently, this variable type does not contribute to the calculations of predicted error covariance $\mathbf{P}(t_{k|k-1})$ since it will not be part of any element of MRDE to predict the state error covariance matrix $\mathbf{P}(t_{k|k-1})$. On the other hand, a *strong variable* contributes to the calculations of predicted error covariance $\mathbf{P}(t_{k|k-1})$. See the example in Section 2.5.1.

- **Only one measured state variable.** In some cases (JEKF application), measuring only one state variable is possible. This measured state variable determines which column of predicted state error covariance $\mathbf{P}(t_{k|k-1})$ is used to compute the Kalman gain through $\mathbf{P}(t_{k|k-1})\mathbf{H}^T$ in the Equation (2.16). If this column has a row with a value equal to zero (no covariance between the measured variable and state variable represented by the row), the Kalman gain cannot be computed to the state variable defined by the row.

3.2.1 Lemma: Inability to Update Kalman Gain for Unshared Parameters based P(0) and Q with Uncorrelated Elements

Given the conditions described above, we have the following Lemma:

Lemma 3.1. *The Kalman gain cannot be updated (by Equation (2.16)) for an unshared parameter that is part of a state variable vector and part of a weak term in a UMM if the initial state error covariance matrix $\boldsymbol{P}(t = 0)$ and $\boldsymbol{Q}$ are formed by uncorrelated elements and there is only one state variable measured.*

The proof of the Lemma 3.1 is in the following, and after, we have an example.

Proof of Lemma 3.1. Let's consider the following:

- A general UMM with an unshared parameter in a weak term represented by a system of nonlinear differential equations of the form:

$$\mathbf{g}(\mathbf{x}, \boldsymbol{\theta}) = \begin{array}{ll} \frac{dx_{msv}}{dt} & = f_1(x_{msv}, x_2, \ldots, x_{n-1}, \theta_1, \theta_2, \ldots, \theta_m), \\ \frac{dx_2}{dt} & = f_2(x_{msv}, x_2, \ldots, x_{n-1}, \theta_1, \theta_2, \ldots, \theta_m), \\ & \vdots \\ \frac{dx_n}{dt} & = f_n(x_{msv}, \theta_{up}) \end{array} \tag{3.4}$$

  where x_{msv} and $x_2, \ldots, x_n$ are the variables of the system, $f_1, f_2, \ldots, f_n$ are the functions defining the system $\mathbf{g}(\cdot)$, and $\theta_1, \theta_2, \ldots, \theta_m$ are the parameters of the system and θ_{up} an unshared parameter.

- A joint state variables vector defined as

$$\mathbf{x}(t)_{general} = [x_{msv}, x_2, \ldots, x_n, \theta_{up}]. \tag{3.5}$$

- A process model defined as

$$\mathbf{x}(t_{k/k-1})_{general} = \mathbf{g}(\mathbf{x}(t_{k-1})_{general}) + \boldsymbol{\omega}(t) \tag{3.6}$$

- x_{msv} as the unique measured state variable (MSV) and $\mathbf{H} = [1\ 0\ \ldots\ 0\ 0]$.

- $\mathbf{R}$ as measurement noise variance of x_{msv}.

- θ_{up} as the unshared parameter (UP) to be evolved (estimated) and presented in only one weak term.
- $\mathbf{P}$ and $\mathbf{Q}$ with uncorrelated elements for the $\mathbf{x}(t)_{general}$ (Equation 3.5),

$$\mathbf{P} = \begin{bmatrix} P_{x_{msv},x_{msv}} & 0 & \dots & 0 & 0 \\ 0 & P_{x_2,x_2} & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & P_{n,n} & 0 \\ 0 & 0 & \dots & 0 & P_{\theta_{up},\theta_{up}} \end{bmatrix}, \tag{3.7}$$

$$\mathbf{Q} = \begin{bmatrix} Q_{x_{msv},x_{msv}} & 0 & \dots & 0 & 0 \\ 0 & Q_{x_2,x_2} & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & Q_{n,n} & 0 \\ 0 & 0 & \dots & 0 & Q_{\theta_{up},\theta_{up}} \end{bmatrix}. \tag{3.8}$$

- The Jacobian $\mathbf{J}_t^{\mathbf{g}}$, (Equation 3.9) with the $\mathbf{x}(t)_{general}$ (Equation 3.5),

$$\mathbf{J}_t^{\mathbf{g}} = \left.\frac{\partial \mathbf{g}(\mathbf{x}(t))}{\partial \mathbf{x}_i}\right|_{\mathbf{x}(t)=\hat{\mathbf{x}}(t-1)}, \tag{3.9}$$

$$\mathbf{J}_t^{\mathbf{g}}(\mathbf{g}(\mathbf{x}(t)_{general})) = \begin{bmatrix} \frac{\partial f_1}{\partial x_{msv}} & \frac{\partial f_1}{\partial x_2} & \dots & \frac{\partial f_1}{\partial x_n} & 0 \\ \frac{\partial f_2}{\partial x_{msv}} & \frac{\partial f_2}{\partial x_2} & \dots & \frac{\partial f_2}{\partial x_n} & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \frac{\partial f_n}{\partial x_{msv}} & \frac{\partial f_n}{\partial x_2} & \dots & \frac{\partial f_n}{\partial x_n} & \frac{\partial f_n}{\partial \theta_{up}} \\ 0 & 0 & \dots & 0 & 0 \end{bmatrix}. \tag{3.10}$$

Given these conditions and the Equation (2.15), we have the following MRDE (based on $\mathbf{P}$ uncorrelated)

$$\frac{d\mathbf{P}(t)}{dt} = \begin{bmatrix} \frac{dP_{x_{msv},x_{msv}}(t)}{dt} = Q_{1,1} + 2P_{1,1}\frac{\partial f_1}{\partial x_{msv}} & \frac{dP_{x_2,x_{msv}}(t)}{dt} = (P_{1,1}+P_{2,2})\frac{\partial f_1}{\partial x_2} & \dots & \frac{dP_{x_n,x_{msv}}(t)}{dt} = (P_{1,1}+P_{3,3})\frac{\partial f_1}{\partial x_n} & \frac{dP_{\theta_{up},x_{msv}}(t)}{dt} = 0 \\ \frac{dP_{x_{msv},x_2}(t)}{dt} = (P_{1,1}+P_{2,2})\frac{\partial f_2}{\partial x_{msv}} & \frac{dP_{x_2,x_2}(t)}{dt} = Q_{2,2} + 2P_{2,2}\frac{\partial f_2}{\partial x_2} & \dots & \frac{dP_{x_n,x_2}(t)}{dt} = (P_{3,3}+P_{2,2})\frac{\partial f_2}{\partial x_n} & \frac{dP_{\theta_{up},x_2}(t)}{dt} = 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \frac{dP_{x_{msv},x_n}(t)}{dt} = (P_{1,1}+P_{n,n})\frac{\partial f_n}{\partial x_{msv}} & \frac{dP_{x_2,x_n}(t)}{dt} = (P_{2,2}+P_{n,n})\frac{\partial f_n}{\partial x_2} & \dots & \frac{dP_{x_n,x_n}(t)}{dt} = Q_{n,n} + 2P_{n,n}\frac{\partial f_n}{\partial x_n} & \frac{dP_{\theta_{up},x_n}(t)}{dt} = 0 \\ \frac{dP_{x_{msv},\theta_{up}}(t)}{dt} = 0 & \frac{dP_{x_2,\theta_{up}}(t)}{dt} = 0 & \dots & \frac{dP_{x_n,\theta_{up}}(t)}{dt} = 0 & \frac{dP_{\theta_{up},\theta_{up}}(t)}{dt} = 0 \end{bmatrix}. \tag{3.11}$$

Now, using this Equation (3.11) to compute the predicted state error covariance matrix $\mathbf{P}(t_{k/k-1})$ from t_{k-1} to t_k with a initial predicted state error covariance matrix $\mathbf{P}(t_{k-1}) = \mathbf{P}_0 = \mathbf{P}(0) = \mathbf{P}_{init}(t=0)$ with uncorrelated elements as following

$$\mathbf{P}_{init}(t=0) = \begin{bmatrix} P_{x_{msv},x_{msv}}(t=0) & 0 & \dots & 0 & 0 \\ 0 & P_{x_2,x_2}(t=0) & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & P_{n,n}(t=0) & 0 \\ 0 & 0 & \dots & 0 & P_{\theta_{up},\theta_{up}}(t=0) \end{bmatrix}, \tag{3.12}$$

we have

$$\mathbf{P}(t_{k/k-1}) = \begin{bmatrix} P_{x_{msv},x_{msv}}(t_{k/k-1}) & P_{x_2,x_{msv}}(t_{k/k-1}) & \dots & P_{n,x_{msv}}(t_{k/k-1}) & P_{\theta_{up},x_{msv}}(t_{k/k-1}) \\ P_{x_{msv},x_2}(t_{k/k-1}) & P_{x_2,x_2}(t_{k/k-1}) & \dots & P_{n,x_2}(t_{k/k-1}) & P_{\theta_{up},x_2}(t_{k/k-1}) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ P_{x_{msv},n}(t_{k/k-1}) & P_{x_2,n}(t_{k/k-1}) & \dots & P_{n,n}(t_{k/k-1}) & P_{\theta_{up},n}(t_{k/k-1}) \\ P_{x_{msv},\theta_{up}}(t_{k/k-1}) = 0 & P_{x_2,\theta_{up}}(t_{k/k-1}) & \dots & P_{n,\theta_{up}}(t_{k/k-1}) & P_{\theta_{up},\theta_{up}}(t_{k/k-1}) \end{bmatrix}. \tag{3.13}$$

Now, using $\mathbf{P}(t_{k/k-1})$, $\mathbf{H}$ and $\mathbf{R}$ to compute the Kalman gain for all variables in the state variable vector $\mathbf{x}(t)_{general}$ (Equation 3.32), we have

$$\mathbf{K}_k = \mathbf{P}(t_{k|k-1})\mathbf{H}^T(\mathbf{H}\mathbf{P}(t_{k|k-1})\mathbf{H}^T + \mathbf{R})^{-1} = \begin{bmatrix} K_{x_{msv}} \\ K_{x_2} \\ \vdots \\ K_{x_n} \\ K_{\theta_{up}} \end{bmatrix} =$$

$$\begin{bmatrix} \frac{P_{x_{msv},x_{msv}}(t_{k/k-1})}{P_{x_{msv},x_{msv}}(t_{k/k-1})+R} \\ \frac{P_{x_{msv},x_2}(t_{k/k-1})}{P_{x_{msv},x_{msv}}(t_{k/k-1})+R} \\ \vdots \\ \frac{P_{x_{msv},n}(t_{k/k-1})}{P_{x_{msv},x_{msv}}(t_{k/k-1})+R} \\ \frac{P_{x_{msv},\theta_{up}}(t_{k/k-1})}{P_{x_{msv},x_{msv}}(t_{k/k-1})+R} \end{bmatrix} = \begin{bmatrix} \frac{P_{x_{msv},x_{msv}}(t_{k/k-1})}{P_{x_{msv},x_{msv}}(t_{k/k-1})+R} \\ \frac{P_{x_{msv},x_2}(t_{k/k-1})}{P_{x_{msv},x_{msv}}(t_{k/k-1})+R} \\ \vdots \\ \frac{P_{x_{msv},n}(t_{k/k-1})}{P_{x_{msv},x_{msv}}(t_{k/k-1})+R} \\ \frac{0}{P_{x_{msv},x_{msv}}(t_{k/k-1})+R} \end{bmatrix} = \begin{bmatrix} \frac{P_{x_{msv},x_{msv}}(t_{k/k-1})}{P_{x_{msv},x_{msv}}(t_{k/k-1})+R} \\ \frac{P_{x_{msv},x_2}(t_{k/k-1})}{P_{x_{msv},x_{msv}}(t_{k/k-1})+R} \\ \vdots \\ \frac{P_{x_{msv},n}(t_{k/k-1})}{P_{x_{msv},x_{msv}}(t_{k/k-1})+R} \\ 0 \end{bmatrix}. \tag{3.14}$$

$\mathbf{H}$ selected the first column of $\mathbf{P}(t_{k/k-1})$ since it is related to the measured value x_{msv}. However, in this column, we have that the predicted state error covariance between x_{msv}

and θ_{up} is zero, $P_{x_{msv},\theta_{up}}(t_{k/k-1}) = Cov(x_{msv}, \theta_{up}) = 0$. Since the solution of $\frac{dP_{x_{msv},\theta_{up}}(t)}{dt} = 0$ obtained from t_{k-1} to t_k is equal to the initial condition that is zero due to $\mathbf{P}(\mathrm{t} = 0)$ with uncorrelated elements, and we have $Cov(x_{msv}, \theta_{up}) = P_{x_{msv},\theta_{up}}(t_{k-1}) = P_{x_{msv},\theta_{up}}(t = 0) = 0$. Then, we have the Kalman gain value for the unshared parameter is zero, $K_{\theta_{up}} = 0$, and consequently the predicted state error covariance $P_{x_{msv},\theta_{up}}(t_{k/k-1})$ cannot be updated (by Equation (2.18)). Since

$$\mathbf{P}(t_{k|k}) = (\mathbf{I} - \mathbf{K}_k\mathbf{H})\mathbf{P}(t_{k|k-1}) = \begin{bmatrix} \vdots & \\ P_{x_{msv},\theta_{up}}(t_{k/k-1}) - K_{\theta_{up}}.P_{x_{msv},x_{msv}}(t_{k/k-1}) & \ldots \end{bmatrix} = \begin{bmatrix} \vdots & \\ 0 - 0.P_{x_{msv},x_{msv}}(t_{k/k-1}) & \ldots \end{bmatrix}. \tag{3.15}$$

Therefore, we have that $P_{x_{msv},\theta_{up}}(t_{k/k}) = P_{x_{msv},\theta_{up}}(t_{k/k-1}) = 0$, and as $P_{x_{msv},\theta_{up}}(t_{k/k}) = 0$ have to be used as a new initial condition for MRDE (Equation 3.11), we have $K_{\theta_{up}} = 0$ for all $P_{x_{msv},\theta_{up}}(t_{k/k-1})$ obtained from t_{k-1} to t_k using Equation (3.11) and consequently $K_{\theta_{up}}$ and $P_{x_{msv},\theta_{up}}(t_{k/k}) = P_{x_{msv},\theta_{up}}(t_{k/k-1}) = 0$ are always zero and cannot be updated. □

Example of Lemma 3.1

To illustrate this Lemma, we show that the Kalman gain value cannot be updated for the unshared parameter μ_{mp} of the UMM with weak terms presented in Case 1 (Section 2.1.1). Let's consider the following:

- The state variables vector

$$\mathbf{x}(t)_{case1} = [Xv, N, MP, \mu_{Xv}, \mu_N, \mu_{mp}]. \tag{3.16}$$

- Xv is the unique measured variable and $\mathbf{H} = [1\ 0\ 0\ 0\ 0\ 0]$.
- R as measurement noise variance of Xv.
- μ_{mp} the unshared parameter to be evolved (estimated) and that is related to a weak term.

- **P** and **Q** with uncorrelated elements for the $\mathbf{x}(t)_{case1}$ (Equation 3.16),

$$\mathbf{P} = \begin{bmatrix} P_{1,1} & 0 & 0 & 0 & 0 & 0 \\ 0 & P_{2,2} & 0 & 0 & 0 & 0 \\ 0 & 0 & P_{3,3} & 0 & 0 & 0 \\ 0 & 0 & 0 & P_{4,4} & 0 & 0 \\ 0 & 0 & 0 & 0 & P_{5,5} & 0 \\ 0 & 0 & 0 & 0 & 0 & P_{6,6} \end{bmatrix}, \mathbf{Q} = \begin{bmatrix} Q_{1,1} & 0 & 0 & 0 & 0 & 0 \\ 0 & Q_{2,2} & 0 & 0 & 0 & 0 \\ 0 & 0 & Q_{3,3} & 0 & 0 & 0 \\ 0 & 0 & 0 & Q_{4,4} & 0 & 0 \\ 0 & 0 & 0 & 0 & Q_{5,5} & 0 \\ 0 & 0 & 0 & 0 & 0 & Q_{6,6} \end{bmatrix}. \tag{3.17}$$

- The Jacobian $\mathbf{J}_t^{\mathbf{g}}$, (Equation 3.9) with the $\mathbf{x}(t)_{case1}$ (Equation 3.16),

$$\mathbf{J}_t^{\mathbf{g}} = \begin{bmatrix} \mu_{Xv} & 0 & 0 & Xv & 0 & 0 \\ -\mu_N & 0 & 0 & 0 & -Xv & 0 \\ \mu_{mp} & 0 & 0 & 0 & 0 & Xv \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{3.18}$$

Given these conditions and the Equation 2.15, we have the following MRDE (based on **P** with uncorrelated elements)

$$\frac{d\mathbf{P}(t)}{dt} = \begin{bmatrix} Q_{1,1} + 2.P_{1,1}.\mu_{Xv} & -P_{1,1}.\mu_N & P_{1,1}.\mu_{mp} & P_{4,4}.Xv & 0 & 0 \\ -P_{1,1}.\mu_N & Q_{2,2} & 0 & 0 & -P_{5,5}.Xv & 0 \\ P_{1,1}.\mu_{mp} & 0 & Q_{3,3} & 0 & 0 & P_{6,6}.Xv \\ P_{4,4}.Xv & 0 & 0 & Q_{4,4} & 0 & 0 \\ 0 & -P_{5,5}.Xv & 0 & 0 & Q_{5,5} & 0 \\ 0 & 0 & P_{6,6}.Xv & 0 & 0 & Q_{6,6} \end{bmatrix} = \tag{3.19}$$

$$\frac{d\mathbf{P}(t)}{dt} = \begin{bmatrix} \frac{P_{X_v,X_v}(t)}{dt} = Q_{1,1} + 2.P_{1,1}.\mu_{Xv} & \frac{P_{N,X_v}(t)}{dt} = -P_{1,1}.\mu_N & \dots & \frac{P_{\mu_N,X_v}(t)}{dt} = 0 & \frac{P_{\mu_{mp},X_v}(t)}{dt} = 0 \\ \frac{P_{X_v,N}(t)}{dt} = -P_{1,1}.\mu_N & \frac{P_{N,N}(t)}{dt} = Q_{2,2} & \dots & \frac{P_{\mu_N,N}(t)}{dt} = -P_{5,5}.Xv & \frac{P_{\mu_{mp},N}(t)}{dt} = 0 \\ \frac{P_{X_v,MP}(t)}{dt} = P_{1,1}.\mu_{mp} & \frac{P_{N,MP}(t)}{dt} = 0 & \dots & \frac{P_{\mu_N,MP}(t)}{dt} = 0 & \frac{P_{\mu_{mp},MP}(t)}{dt} = P_{6,6}.Xv \\ \frac{P_{X_v,\mu_{X_v}}(t)}{dt} = P_{4,4}.Xv & \frac{P_{N,\mu_{X_v}}(t)}{dt} = 0 & \dots & \frac{P_{\mu_N,\mu_{X_v}}(t)}{dt} = 0 & \frac{P_{\mu_{mp},\mu_{X_v}}(t)}{dt} = 0 \\ \frac{P_{X_v,\mu_N}(t)}{dt} = 0 & \frac{P_{N,\mu_N}(t)}{dt} = -P_{5,5}.Xv & \dots & \frac{P_{\mu_N,\mu_N}(t)}{dt} = Q_{5,5} & \frac{P_{\mu_{mp},\mu_N}(t)}{dt} = 0 \\ \frac{P_{X_v,\mu_{mp}}(t)}{dt} = 0 & \frac{P_{N,\mu_{mp}}(t)}{dt} = 0 & \dots & \frac{P_{\mu_N,\mu_{mp}}(t)}{dt} = 0 & \frac{P_{\mu_{mp},\mu_{mp}}(t)}{dt} = Q_{6,6} \end{bmatrix}. \tag{3.20}$$

Now, using this Equation 3.20 to compute the predicted state error covariance matrix $\mathbf{P}(t_{k/k-1})$ (for the Case1 of Section 2.1.1) from t_{k-1} to t_k with a initial predicted state error covariance matrix $\mathbf{P}(t_{k-1}) = \mathbf{P}(t=0)$ with uncorrelated elements as following

$$\mathbf{P}(t=0) = \begin{bmatrix} P_{X_v,X_v}(t=0) & 0 & 0 & 0 & 0 & 0 \\ 0 & P_{N,N}(t=0) & 0 & 0 & 0 & 0 \\ 0 & 0 & P_{MP,MP}(t=0) & 0 & 0 & 0 \\ 0 & 0 & 0 & P_{\mu_{X_v},\mu_{X_v}}(t=0) & 0 & 0 \\ 0 & 0 & 0 & 0 & P_{\mu_N,\mu_N}(t=0) & 0 \\ 0 & 0 & 0 & 0 & 0 & P_{\mu_{mp},\mu_{mp}}(t=0) \end{bmatrix}, \tag{3.21}$$

we have

$$\mathbf{P}(t_{k/k-1}) = \begin{bmatrix} P_{X_v,X_v}(t_{k/k-1}) & P_{N,X_v}(t_{k/k-1}) & P_{MP,X_v}(t_{k/k-1}) & P_{\mu_{X_v},X_v}(t_{k/k-1}) & P_{\mu_N,X_v}(t_{k/k-1}) & P_{\mu_{mp},X_v}(t_{k/k-1}) \\ P_{X_v,N}(t_{k/k-1}) & P_{N,N}(t_{k/k-1}) & P_{MP,N}(t_{k/k-1}) & P_{\mu_{X_v},N}(t_{k/k-1}) & P_{\mu_N,N}(t_{k/k-1}) & P_{\mu_{mp},N}(t_{k/k-1}) \\ P_{X_v,MP}(t_{k/k-1}) & P_{N,MP}(t_{k/k-1}) & P_{MP,MP}(t_{k/k-1}) & P_{\mu_{X_v},MP}(t_{k/k-1}) & P_{\mu_N,MP}(t_{k/k-1}) & P_{\mu_{mp},MP}(t_{k/k-1}) \\ P_{X_v,\mu_{X_v}}(t_{k/k-1}) & P_{N,\mu_{X_v}}(t_{k/k-1}) & P_{MP,\mu_{X_v}}(t_{k/k-1}) & P_{\mu_{X_v},\mu_{X_v}}(t_{k/k-1}) & P_{\mu_N,\mu_{X_v}}(t_{k/k-1}) & P_{\mu_{mp},\mu_{X_v}}(t_{k/k-1}) \\ P_{X_v,\mu_N}(t_{k/k-1}) & P_{N,\mu_N}(t_{k/k-1}) & P_{MP,\mu_N}(t_{k/k-1}) & P_{\mu_{X_v},\mu_N}(t_{k/k-1}) & P_{\mu_N,\mu_N}(t_{k/k-1}) & P_{\mu_{mp},\mu_N}(t_{k/k-1}) \\ P_{X_v,\mu_{mp}}(t_{k/k-1}) & P_{N,\mu_{mp}}(t_{k/k-1}) & P_{MP,\mu_{mp}}(t_{k/k-1}) & P_{\mu_{X_v},\mu_{mp}}(t_{k/k-1}) & P_{\mu_N,\mu_{mp}}(t_{k/k-1}) & P_{\mu_{mp},\mu_{mp}}(t_{k/k-1}) \end{bmatrix} = \tag{3.22}$$

$$\mathbf{P}(t_{k/k-1}) = \begin{bmatrix} Cov(X_v,X_v) & Cov(N,X_v) & Cov(MP,X_v) & Cov(\mu_{X_v},X_v) & Cov(\mu_N,X_v) & Cov(\mu_{mp},X_v) \\ Cov(X_v,N) & Cov(N,N) & Cov(MP,N) & Cov(\mu_{X_v},N) & Cov(\mu_N,N) & Cov(\mu_{mp},N) \\ Cov(X_v,MP) & Cov(N,MP) & Cov(MP,MP) & Cov(\mu_{X_v},MP) & Cov(\mu_N,MP) & Cov(\mu_{mp},MP) \\ Cov(X_v,\mu_{X_v}) & Cov(N,\mu_{X_v}) & Cov(MP,\mu_{X_v}) & Cov(\mu_{X_v},\mu_{X_v}) & Cov(\mu_N,\mu_{X_v}) & Cov(\mu_{mp},\mu_{X_v}) \\ Cov(X_v,\mu_N) & Cov(N,\mu_N) & Cov(MP,\mu_N) & Cov(\mu_{X_v},\mu_N) & Cov(\mu_N,\mu_N) & Cov(\mu_{mp},\mu_N) \\ Cov(X_v,\mu_{mp}) & Cov(N,\mu_{mp}) & Cov(MP,\mu_{mp}) & Cov(\mu_{X_v},\mu_{mp}) & Cov(\mu_N,\mu_{mp}) & Cov(\mu_{mp},\mu_{mp}) \end{bmatrix}. \tag{3.23}$$

Now, using $\mathbf{P}(t_{k/k-1})$, $\mathbf{H}$ and R to compute the Kalman gain values for all variables in the state variable vector, we have

$$\mathbf{K}_k = \mathbf{P}(t_{k|k-1})\mathbf{H}^T(\mathbf{H}\mathbf{P}(t_{k|k-1})\mathbf{H}^T + \mathbf{R})^{-1} = \begin{bmatrix} K_{X_v} \\ K_N \\ K_{MP} \\ K_{\mu_{X_v}} \\ K_{\mu_N} \\ K_{\mu_{mp}} \end{bmatrix} = \begin{bmatrix} \frac{P_{X_v,X_v}(t_{k/k-1})}{P_{X_v,X_v}(t_{k/k-1})+R} \\ \frac{P_{X_v,N}(t_{k/k-1})}{P_{X_v,X_v}(t_{k/k-1})+R} \\ \frac{P_{X_v,MP}(t_{k/k-1})}{P_{X_v,X_v}(t_{k/k-1})+R} \\ \frac{P_{X_v,\mu_{X_v}}(t_{k/k-1})}{P_{X_v,X_v}(t_{k/k-1})+R} \\ \frac{P_{X_v,\mu_N}(t_{k/k-1})}{P_{X_v,X_v}(t_{k/k-1})+R} \\ \frac{P_{X_v,\mu_{mp}}(t_{k/k-1})}{P_{X_v,X_v}(t_{k/k-1})+R} \end{bmatrix} = \begin{bmatrix} \frac{P_{X_v,X_v}(t_{k/k-1})}{P_{X_v,X_v}(t_{k/k-1})+R} \\ \frac{P_{X_v,N}(t_{k/k-1})}{P_{X_v,X_v}(t_{k/k-1})+R} \\ \frac{P_{X_v,MP}(t_{k/k-1})}{P_{X_v,X_v}(t_{k/k-1})+R} \\ \frac{P_{X_v,\mu_{X_v}}(t_{k/k-1})}{P_{X_v,X_v}(t_{k/k-1})+R} \\ \frac{P_{X_v,\mu_N}(t_{k/k-1})}{P_{X_v,X_v}(t_{k/k-1})+R} \\ \frac{0}{P_{X_v,X_v}(t_{k/k-1})+R} \end{bmatrix} = \begin{bmatrix} \frac{P_{X_v,X_v}(t_{k/k-1})}{P_{X_v,X_v}(t_{k/k-1})+R} \\ \frac{P_{X_v,N}(t_{k/k-1})}{P_{X_v,X_v}(t_{k/k-1})+R} \\ \frac{P_{X_v,MP}(t_{k/k-1})}{P_{X_v,X_v}(t_{k/k-1})+R} \\ \frac{P_{X_v,\mu_{X_v}}(t_{k/k-1})}{P_{X_v,X_v}(t_{k/k-1})+R} \\ \frac{P_{X_v,\mu_N}(t_{k/k-1})}{P_{X_v,X_v}(t_{k/k-1})+R} \\ 0 \end{bmatrix}. \tag{3.24}$$

$\mathbf{H}$ selected the first column of $\mathbf{P}(t_{k/k-1})$ since it is related to the measured value X_v. However, in this column, we have that the predicted error covariance between X_v and μ_{mp} is zero, $Cov(X_v, \mu_{mp}) = 0$. Since the solution of $\frac{P_{X_v,\mu_{mp}}(t)}{dt} = 0$ obtained from t_{k-1} to t_k is zero, and we have $P_{X_v,\mu_{mp}}(t_{k/k-1}) = P_{\mu_{X_v},\mu_{mp}}(t_{k-1}) = P_{\mu_{X_v},\mu_{mp}}(t = 0) = 0$. This means that due to $\mathbf{P}$(t=0) with uncorrelated elements the obtained solution is equal to the initial condition. Then, we have the kalman gain value for the unshared parameter is zero, $K_{\mu_{mp}} = 0$, and consequently the predicted state error covariance $P_{X_v,\mu_{mp}}(t_{k/k-1})$ cannot be updated in $\mathbf{P}(t_{k|k-1})$ by Eq 2.18. Since

$$\mathbf{P}(t_{k|k}) = (\mathbf{I} - \mathbf{K}_k\mathbf{H})\mathbf{P}(t_{k|k-1}) = \begin{bmatrix} P_{X_v,X_v}(t_{k/k-1}) - K_{Xv}.P_{X_v,X_v}(t_{k/k-1}) & ... \\ P_{X_v,N}(t_{k/k-1}) - K_N.P_{X_v,X_v}(t_{k/k-1}) & ... \\ P_{X_v,MP}(t_{k/k-1}) - K_{mp}.P_{X_v,X_v}(t_{k/k-1}) & ... \\ P_{X_v,\mu_{Xv}}(t_{k/k-1}) - K_{\mu_{X}v}.P_{X_v,X_v}(t_{k/k-1}) & ... \\ P_{X_v,\mu_N}(t_{k/k-1}) - K_{\mu_N}.P_{X_v,X_v}(t_{k/k-1}) & ... \\ P_{X_v,\mu_{mp}}(t_{k/k-1}) - K_{\mu_{mp}}.P_{X_v,X_v}(t_{k/k-1}) & ... \end{bmatrix} = \tag{3.25}$$

$$\mathbf{P}(t_{k|k}) = (\mathbf{I} - \mathbf{K}_k\mathbf{H})\mathbf{P}(t_{k|k-1}) = \begin{bmatrix} P_{X_v,X_v}(t_{k/k-1}) - K_{Xv}.P_{X_v,X_v}(t_{k/k-1}) & ... \\ P_{X_v,N}(t_{k/k-1}) - K_N.P_{X_v,X_v}(t_{k/k-1}) & ... \\ P_{X_v,MP}(t_{k/k-1}) - K_{mp}.P_{X_v,X_v}(t_{k/k-1}) & ... \\ P_{X_v,\mu_{Xv}}(t_{k/k-1}) - K_{\mu_{X}v}.P_{X_v,X_v}(t_{k/k-1}) & ... \\ P_{X_v,\mu_N}(t_{k/k-1}) - K_{\mu_N}.P_{X_v,X_v}(t_{k/k-1}) & ... \\ 0 - 0.P_{X_v,X_v}(t_{k/k-1}) & ... \end{bmatrix}. \tag{3.26}$$

We have that $P_{X_v,\mu_{mp}}(t_{k/k}) = P_{X_v,\mu_{mp}}(t_{k/k-1}) = 0$, and as the $P_{X_v,\mu_{mp}}(t_{k/k}) = 0$ have to be used as a new initial condition for MRDE (Equation 3.20), we have $K_{\mu_{mp}} = 0$ for all $P_{X_v,\mu_{mp}}(t_{k/k-1})$ in $\mathbf{P}(t_{k/k-1})$ that are obtained from t_{k-1} to t_k using Equation 3.20 and consequently no updates for $K_{\mu_{mp}}$ and $P_{X_v,\mu_{mp}}(t_{k/k-1})$.

3.2.2 Theorem: JEKF Limitation

The consequence of the Lemma 3.1 (Section 5.1) is the following Theorem:

Theorem 3.1. *The JEKF cannot estimate an unshared parameter (parameter evolution) that is part of a state variable vector and part of a weak term in a UMM if the initial state error covariance matrix $\boldsymbol{P}(t = 0)$ and $\boldsymbol{Q}$ are composed of uncorrelated elements, and there is only one state variable measured. This is because the Kalman gain value for the unshared parameter is equal to zero for all steps of execution of the JEKF algorithm.*

The proof of Theorem 3.1 is in the following, and an example.

Proof of Theorem 3.1. This proof can be done using the conditions and results described previously in the proof of the Lemma 3.1 (Section 3.2.1).

Then, let us consider the following:

- $\mathbf{H}$=[1 0 ... 0 0] and $\mathbf{K}_k = [K_{x_{msv}}, K_{x_2}, \ldots, K_{x_n}, K_{\theta_{up}}]^T$ as obtained in the proof of Lemma 3.1 in Section 3.2.1, where $K_{\theta_{up}} = 0$.
- $\mathbf{y}_k$ as measured value of x_{msv} .
- Predicted mean of the state variable vector $\hat{\mathbf{x}}(t_{k/k-1})_{general} = [\hat{x}_{msv}, \hat{x}_2, \ldots, \hat{x}_n, \hat{\theta}_{up}]^T$ regards to the general UMM used in the proof of Lemma 3.1 in Section 3.2.1.

Now, using the Equation (2.17) to compute the estimated mean of the state variable vector $\hat{\mathbf{x}}(t_{k/k})_{general}$, we have

$$\hat{\mathbf{x}}(t_{k/k})_{general} = \hat{\mathbf{x}}(t_{k/k-1})_{general} + \mathbf{K}_k(\mathbf{y}_k - \mathbf{H}\hat{\mathbf{x}}(t_{k/k-1})_{general}) \tag{3.27}$$

$$\hat{\mathbf{x}}(t_{k/k})_{general} = \begin{bmatrix} \hat{x}_{msv} \\ \hat{x}_2 \\ \vdots \\ \hat{x}_n \\ \hat{\theta}_{up} \end{bmatrix} + \begin{bmatrix} K_{x_{msv}} \\ K_{x_2} \\ \vdots \\ K_{x_n} \\ K_{\theta_{up}} \end{bmatrix} .(\mathbf{y}_k - \hat{x}_{msv}) = \begin{bmatrix} \hat{x}_{msv} + K_{x_{msv}}.(\mathbf{y}_k - \hat{x}_{msv}) \\ \hat{x}_2 + K_{x_2}.(\mathbf{y}_k - \hat{x}_{msv}) \\ \vdots \\ \hat{x}_n + K_{x_n}.(\mathbf{y}_k - \hat{x}_{msv}) \\ \hat{\theta}_{up} + 0 \end{bmatrix} \tag{3.28}$$

Then, we have that the estimated mean of the unshared parameter $\hat{\theta}_{up}(t_{k/k})$ (composing the $\hat{\mathbf{x}}(t_{k/k})_{general}$) is equal to the predicted mean of unshared parameter $\hat{\theta}_{up}(t_{k/k-1})$ (composing the $\hat{\mathbf{x}}(t_{k/k-1})_{general}$) for all step from t_{k-1} to t_k. In another words, the JEKF fails to perform the parameter evolution, since it does not have a noise component to evolve the parameter as described in the $\theta(t_k) = \theta(t_{k-1}) + noise$ (Equation 2.29), then $\hat{\theta}_{up}(t_{k/k}) = \hat{\theta}_{up}(t_{k/k-1})$ for all step from t_{k-1} to t_k. □

To illustrate this Theorem, we show the JEKF failure with the condition and results used previously in the example (Section 3.2.1) of the Lemma 3.1.

Example of Theorem 3.1

Then, let's consider the following:

- the UMM of Case1 (Section 2.1.1).
- $\mathbf{H}$=[1 0 0 0 0 0] and $\mathbf{K}_k = [\mathbf{K}_{X_v}, \mathbf{K}_N, \mathbf{K}_{MP}, \mathbf{K}_{\mu_{X_v}}, \mathbf{K}_{\mu_N}, \mathbf{K}_{\mu_{mp}}]^T$ as obtained in the proof of Lemma 3.1, where $\mathbf{K}_{\mu_{mp}} = 0$.
- $\mathbf{y}_k$ as measured value of X_v .
- $\hat{\mathbf{x}}(t_{k/k-1})_{case1} = [\hat{X_v}, \hat{N}, \hat{MP}, \hat{\mu_{Xv}}, \hat{\mu_N}, \hat{\mu_{mp}}]^T$.

Now, using the Equation 2.17 to provide the estimated mean of the state variable vector $\hat{\mathbf{x}}(t_{k/k})_{case1}$, we have

$$\hat{\mathbf{x}}(t_{k/k})_{case1} = \hat{\mathbf{x}}(t_{k/k-1})_{case1} + \mathbf{K}_k(\mathbf{y}_k - \mathbf{H}\hat{\mathbf{x}}(t_{k/k-1})_{case1}) \tag{3.29}$$

$$\hat{\mathbf{x}}(t_{k/k})_{case1} = \begin{bmatrix} \hat{X_v} \\ \hat{N} \\ \hat{MP} \\ \hat{\mu_{Xv}} \\ \hat{\mu_N} \\ \hat{\mu_{mp}} \end{bmatrix} + \begin{bmatrix} \mathbf{K}_{X_v} \\ \mathbf{K}_N \\ \mathbf{K}_{MP} \\ \mathbf{K}_{\mu_{X_v}} \\ \mathbf{K}_{\mu_N} \\ \mathbf{K}_{\mu_{mp}} \end{bmatrix} .(\mathbf{y}_k - X_v) = \begin{bmatrix} \hat{X_v} + \mathbf{K}_{X_v}.(\mathbf{y}_k - \hat{X_v}) \\ \hat{N} + \mathbf{K}_N.(\mathbf{y}_k - \hat{X_v}) \\ \hat{MP} + \mathbf{K}_{MP}.(\mathbf{y}_k - \hat{X_v}) \\ \hat{\mu_{Xv}} + \mathbf{K}_{\mu_{X_v}}.(\mathbf{y}_k - \hat{X_v}) \\ \hat{\mu_N} + \mathbf{K}_{\mu_{mp}}.(\mathbf{y}_k - \hat{X_v}) \\ \hat{\mu_{mp}} + 0 \end{bmatrix} \tag{3.30}$$

Then, we have the estimated mean $\hat{\mathbf{x}}(t_{k/k})_{case1}$ of the unshared parameter is equal to the predicted mean $\hat{\mathbf{x}}(t_{k/k-1})_{case1}$ of unshared parameter for all step from t_{k-1} to t_k. In another words, the JEKF failure to perform the parameter evolution, since $\hat{\mu_{mp}}(t_{k/k}) = \hat{\mu_{mp}}(t_{k/k-1})$ all step from t_{k-1} to t_k.

3.2.3 Theorem: DEKF Limitation

By using the DEKF (see algorithm 2.3), we can estimate the state and parameters of the state space model (Equation 2.7). However, in the presence of biomanufacturing conditions for fast and low-cost bioprocess monitoring, we have the issue case described by the following theorem.

Theorem 3.2. *The DEKF cannot estimate an unshared parameter (parameter evolution) that is part of a weak term in a UMM if the unshared parameter is not part of the nonlinear function that models the unique state variable measured.*

Theorem 3.2 shows that the "failure" arises from the inability of the algorithm to derive any information about the unshared parameter from the measured state variable.

The proof of Theorem 3.2 is in the following.

Proof. Let's consider the following:

- A general UMM with an unshared parameter in an weak term represented by a system of nonlinear differential equations of the form:

$$\begin{aligned} \frac{dx_{msv}}{dt} &= f_1(x_{msv}, x_2, \ldots, x_{n-1}, \theta_1, \theta_2, \ldots, \theta_m) \\ \frac{dx_2}{dt} &= f_2(x_{msv}, x_2, \ldots, x_{n-1}, \theta_1, \theta_2, \ldots, \theta_m) \\ &\vdots \\ \frac{dx_n}{dt} &= f_n(x_{msv}, \theta_{up}) \end{aligned} \tag{3.31}$$

  where x_{msv} and $x_2, \ldots, x_n$ are the variables of the system, $f_1, f_2, \ldots, f_n$ are the functions defining the system represented by $\mathbf{g}(\cdot)$, and $\theta_1, \theta_2, \ldots, \theta_m$ are the parameters of the system and θ_{up} an unshared parameter.

- A state variables vector defined as

$$\mathbf{x} = [x_{msv}, x_2, \ldots, x_n]^T. \tag{3.32}$$

- x_{msv} as the unique measured state variable (MSV) and $\bar{\mathbf{H}} = [1\ 0\ \ldots\ 0\ 0]$.

- $\mathbf{R}$ as measurement noise variance of x_{msv}.
- $\boldsymbol{\theta} = [\theta_{up}]$, where θ_{up} as the unshared parameter (UP) to be evolved (estimated) and presented in only one weak term.
- Euler–Maruyama discretization of general UMM (Equations 3.31)

$$\mathbf{x}_{k|k-1} = \begin{pmatrix} x_{msv,k|k-1} \\ x_{2,k|k-1} \\ \vdots \\ x_{n,k|k-1} \end{pmatrix} = \begin{pmatrix} x_{msv,k-1|k-1} + f_1(x_{msv,k-1|k-1}, x_{2,k-1|k-1}, \ldots, x_{n-1,k-1|k-1}, \theta_1, \theta_2, \ldots, \theta_m)\Delta t \\ x_{2,k-1|k-1} + f_2(x_{msv,k-1|k-1}, x_{2,k-1|k-1}, \ldots, x_{n-1,k-1|k-1}, \theta_1, \theta_2, \ldots, \theta_m)\Delta t \\ \vdots \\ x_{n,k-1|k-1} + f_n(x_{msv,k-1|k-1}, \theta_{up,k-1|k-1})\Delta t \end{pmatrix} \tag{3.33}$$

Given these conditions above and the DEKF Algorithm 2.3, we have that

$$\begin{aligned}
\mathbf{H}_k^{\boldsymbol{\theta}} &= -\frac{\partial \epsilon_k}{\partial \boldsymbol{\theta}} = \frac{\partial \mathbf{h}(\mathbf{x}_{k|k-1})}{\partial \boldsymbol{\theta}} = \bar{\mathbf{H}} \frac{\partial \mathbf{x}_{k|k-1}}{\partial \boldsymbol{\theta}} \mid_{\boldsymbol{\theta}_{k|k-1}} \\
\mathbf{H}_k^{\boldsymbol{\theta}} &= [1\ 0\ \ldots\ 0] \begin{bmatrix} \frac{d(x_{msv,k-1|k-1} + f_1(x_{msv,k-1|k-1}, x_{2,k-1|k-1}, \ldots, x_{n-1,k-1|k-1}, \theta_1, \theta_2, \ldots, \theta_m)\Delta t)}{d\theta_{up}} = 0 \\ \frac{d(x_{2,k-1|k-1} + f_2(x_{msv,k-1|k-1}, x_{2,k-1|k-1}, \ldots, x_{n-1,k-1|k-1}, \theta_1, \theta_2, \ldots, \theta_m)\Delta t)}{d\theta_{up}} = 0 \\ \vdots \\ \frac{d(x_{n,k-1|k-1} + f_n(x_{msv,k-1|k-1}, \theta_{up,k-1|k-1})\Delta t)}{d\theta_{up}} = \Delta t \frac{d(f_n(x_{msv,k-1|k-1}, \theta_{up,k-1|k-1}))}{d\theta_{up}} \end{bmatrix}^T \\
\mathbf{H}_k^{\boldsymbol{\theta}} &= [1\ 0\ \ldots\ 0] \begin{bmatrix} 0 \\ 0 \\ \vdots \\ \Delta t \frac{d(f_n(x_{msv,k-1|k-1}, \theta_{up,k-1|k-1}))}{d\theta_{up}} \end{bmatrix}^T = 0.
\end{aligned} \tag{3.34}$$

Therefore,

$$\begin{aligned}\mathbf{K}_k^{\boldsymbol{\theta}} &= \mathbf{P}_{k|k-1}^{\boldsymbol{\theta}}\mathbf{H}_k^{\boldsymbol{\theta}T}(\mathbf{H}_k^{\boldsymbol{\theta}}\mathbf{P}_{k|k-1}^{\boldsymbol{\theta}}\mathbf{H}_k^{\boldsymbol{\theta}T} + \mathbf{R}^{\boldsymbol{\theta}})^{-} \\ \mathbf{K}_k^{\boldsymbol{\theta}} &= 0\end{aligned} \tag{3.35}$$

Consequently, there is no update because $\boldsymbol{\theta}_{k|k} = \boldsymbol{\theta}_{k|k-1}$,

$$\begin{aligned}\boldsymbol{\theta}_{k|k} &= \boldsymbol{\theta}_{k|k-1} + \mathbf{K}_k^{\boldsymbol{\theta}}\epsilon_k, \\ \boldsymbol{\theta}_{k|k} &= \boldsymbol{\theta}_{k|k-1} + 0, \\ \boldsymbol{\theta}_{k|k} &= \boldsymbol{\theta}_{k|k-1}.\end{aligned} \tag{3.36}$$

□

3.3 SANTO: Specific initiAl coNdiTiOn

This section presents the SANTO approach to avoid the JEKF limitation described in Theorem 3.1. The initial condition of MRDE is the initial state error covariance matrix $\mathbf{P}_0 = \mathbf{P}(t = 0)$. When it is composed of uncorrelated elements ($\mathrm{P}_{i,j} = 0$), some initial conditions of time-invariant ODEs ($\frac{dP_{i,j}(t_{k|k-1})}{dt} = 0$) in the MRDE are zero, and consequently, the obtained solutions from t_{k-1} to t_k for some of these time-invariant ODEs are zero too. Furthermore, in the presence of the biomanufacturing conditions (failure case presented in Section 3.2), we have that the Kalman gain value regarding the unshared parameter (K_{UP}) and the predicted state error covariance between the unique measured state variable and the unshared parameter ($P_{MSV,UP}(t_{k|k-1})$), are zero too, $K_{UP} = 0$ and $P_{MSV,UP}(t_{k|k-1}) = 0$. Then, the K_{UP}, and $P_{MSV,UP}(t_{k|k-1})$ that compose $\mathbf{P}(t_{k|k-1})$ cannot be updated regards to the unshared parameter (see the Lemma 3.1), and they are constant and equal to zero during the entire process execution of JEKF. It is worth noting that $P_{MSV,UP}(t_{k|k-1})$ is an element of $\mathbf{P}(t_{k|k-1})$ such as $P_{MSV,UP}(t = 0)$ is an element of $\mathbf{P}(t = 0)$. Furthermore, the $K_{UP} = 0$ during the entire JEKF execution reflects an unrealistic situation. This would mean that the prediction regarding the unshared parameter is perfect and does not need the influence of the measurement in the correction step of JEKF since there is no uncertainty in the prediction regarding the unshared parameter. However, based on prior knowledge, we know that the process model predictions regarding the unshared parameter are imperfect since we need to perform the evolution of the unshared parameter; otherwise, they would be the same during the entire process. Therefore, we need $K_{UP} \neq 0$ and $P_{MSV,UP}(t_{k|k-1}) \neq 0$.

In general, the initial condition of MRDE is $\mathbf{P}(t = 0)$ with uncorrelated elements ($\mathrm{P}_{i,j} = 0$) due to the difficulty of estimating all covariances with a limited dataset. However, instead, considering all off-diagonal elements of $\mathbf{P}(t = 0)$ equal zero ($\mathrm{P}_{i,j} = 0$), we can consider only the key off-diagonal element (that is the $\mathrm{P}_{MSV,UP}(t = 0)$) with an initial value different of zero ($P_{MSV,UP}(t = 0) = P_{UP,MSV}(t = 0) \neq 0$) to avoid the failure case. This value could be a positive quantity, λ, since the off-diagonal elements of $\mathbf{P}(t = 0)$ can show a positive covariance between two variables, indicating that they tend to increase or decrease together. Furthermore, the value of λ should be different from zero and small enough to not significantly affect the filter's estimates, but large enough to prevent the failure case. Then, with this consideration, we can have a value for the initial state error covariance between the MSV and an UP ($\mathrm{P}_{MSV,UP}(t = 0)$). If we add it to the initial state error covariance matrix $\mathbf{P}(t = 0)$ with the other uncorrelated elements, we have a specific initial condition for MRDE that enables us to update the K_{UP} and $P_{MSV,UP}(t_{k|k-1})$ present in $\mathbf{P}(t_{k|k-1})$ and, consequently, avoids the JEKF failure.

Theorem 3.3 (SANTO—Proposed approach to avoid the JEKF failure)**.** *The addition of a positive quantity (λ) to the $P_{MSV,UP}(t = 0)$ and $P_{UP,MSV}(t = 0)$ (where $P_{MSV,UP} = P_{UP,MSV}$) in $\boldsymbol{P}(t = 0)$ to initialize the MRDE with a specific initial condition can prevent the Kalman gain being zero in the entire execution of JEKF, and prevent the JEKF failure (Section 3.2).*

Proof. The proof of SANTO approach can be done using the conditions described previously in the proof of the Lemma 3.1 (Section 3.2.1) and the Theorem 3.1 (Section 3.2.2).

Then, let us consider the following:

- A quantity $\lambda > 0$.
- x_{msv} as the unique measured state variable (MSV) and $\mathbf{H} = [1\ 0\ ...\ 0\ 0]$.
- R as measurement noise variance of x_{msv}.
- θ_{up} as the unshared parameter (UP) to be evolved (estimated) and presented in only one weak term.
- An specific initial predicted state error covariance matrix $\mathbf{P}(t_{k-1}) = \mathbf{P}_0 = \mathbf{P}_{santo}(t = 0)$ with uncorrelated elements and $P_{MSV,UP}(t = 0) = P_{x_{msv},\theta_{up}}(t = 0) = \lambda$ as following

$$\mathbf{P}_{santo}(t=0) = \begin{bmatrix} P_{x_{msv},x_{msv}}(t=0) & 0 & \dots & 0 & P_{\theta_{up},x_{msv}}(t=0)=\lambda \\ 0 & P_{x_2,x_2}(t=0) & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & P_{n,n}(t=0) & 0 \\ P_{x_{msv},\theta_{up}}(t=0)=\lambda & 0 & \dots & 0 & P_{\theta_{up},\theta_{up}}(t=0) \end{bmatrix}, \tag{3.37}$$

Now, using this Equation (3.11) to compute the predicted state error covariance matrix $\mathbf{P}(t_{k/k-1})$ from t_{k-1} to t_k with the specific initial predicted state error covariance matrix $\mathbf{P}(t_{k-1}) = \mathbf{P}_0 = \mathbf{P}_{santo}(t=0)$ we have

$$\mathbf{P}(t_{k/k-1}) = \begin{bmatrix} \vdots & \vdots & \\ P_{x_{msv},\theta_{up}}(t_{k/k-1})=\lambda & P_{x_2,\theta_{up}}(t_{k/k-1}) & \dots \end{bmatrix}. \tag{3.38}$$

where $P_{x_{msv},\theta_{up}}(t_{k/k-1}) = \lambda$ because the solution of $\frac{dP_{x_{msv},\theta_{up}}(t)}{dt} = 0$ obtained from t_{k-1} to t_k is equal to the initial condition that is λ in $\mathbf{P}_0$. Now, using $\mathbf{P}(t_{k/k-1})$, $\mathbf{H}$ and R to compute the Kalman gain for all variables in the state variable vector $\mathbf{x}(t)$ (Equation 3.32), we have

$$\mathbf{K}_k = \mathbf{P}(t_{k|k-1})\mathbf{H}^T(\mathbf{H}\mathbf{P}(t_{k|k-1})\mathbf{H}^T+\mathbf{R})^{-1} = \begin{bmatrix} \vdots \\ K_{\theta_{up}} \end{bmatrix} = \begin{bmatrix} \vdots \\ \frac{P_{x_{msv},\theta_{up}}(t_{k/k-1})}{P_{x_{msv},x_{msv}}(t_{k/k-1})+R} \end{bmatrix} = \begin{bmatrix} \vdots \\ \frac{\lambda}{P_{x_{msv},x_{msv}}(t_{k/k-1})+R} \end{bmatrix}. \tag{3.39}$$

Then, we have the Kalman gain value for the unshared parameter as

$$K_{\theta_{up}} = \lambda(P_{x_{msv},x_{msv}}(t_{k/k-1}) + R)^- \neq 0, \tag{3.40}$$

and consequently the predicted state error covariance $P_{x_{msv},\theta_{up}}(t_{k/k-1})$ can be updated by Equation (2.18) and predicted mean of the state variable vector regards to UP, $\hat{\theta}_{up}(t_{k/k-1})$, can be updated Equation (2.17). Therefore, we have $P_{x_{msv},\theta_{up}}(t_{k/k}) \neq P_{x_{msv},\theta_{up}}(t_{k/k-1})$ and $\hat{\theta}_{up}(t_{k/k}) \neq \hat{\theta}_{up}(t_{k/k-1})$ during the entire execution of JEKF. □

It is essential to point out that the SANTO is inspired by the idea of regularization technique used to avoid the singularity problem in the state error covariance matrix [158, 163]. However, instead of adding a small quantity to the diagonal elements of the state

error covariance matrix $\mathbf{P}$, such as the perturbed-P algorithm [163], we only add a quantity (λ) to the $P_{MSV,UP}(t=0)$ in $\mathbf{P}(t=0)$ to initialize the MRDE. Furthermore, a quantity λ to the $\mathrm{P}_{MSV,UP}(t=0)$ can be defined by empirical tuning. One of the most common ways to define a quantity is by trial and error. This involves running the filter with different values of λ and choosing the value that results in the best performance [103]. However, λ can be also defined with BAT (proposed approach of Chapter 4).

Soft sensor development based on JEKF-SANTO for bioprocess monitoring

Figure 3.1 shows the steps to develop a soft sensor for bioprocess monitoring based on JEKF-SANTO.

- Step 1: Data Collection and Preprocessing. The first step in developing a soft sensor for bioprocess monitoring using the JEKF-SANTO approach involves comprehensive data collection and preprocessing. Once collected, this data must be meticulously cleaned and preprocessed to remove outliers and address any missing values. This preprocessing is crucial to ensure the quality and reliability of the data, which forms the foundation for accurate modeling and estimation in subsequent steps.

- Step 2: Analyze the Biomanufacturing Conditions. This step involves a comprehensive analysis of the biomanufacturing conditions where JEKF fails to estimate an unshared parameter that is part of a state variable vector and part of a weak term in a UMM if the initial state error covariance matrix $\mathbf{P}(\mathrm{t}=0)$ and $\mathbf{Q}$ are composed of uncorrelated elements, and there is only one state variable measured.

- Step 3: Implement JEKF with the SANTO approach. Implement the JEKF algorithm, defining the process model and the measurement model. Modify the initial state error covariance matrix $\mathbf{P}(t=0)$ as per the SANTO approach, adding a specific quantity λ to the covariance between the measured state variable and the unshared parameter.

- Step 4: JEKF-SANTO calibration. Tune the R and Q of JEKF-SANTO based on consistency tests, and adjust the λ parameter model based on the estimates obtained from JEKF-SANTO related to the unshared parameter and the associated weak variable.

- Step 5: Deployment and Monitoring. Integrate the JEKF-SANTO as a soft sensor into the biomanufacturing process control system to monitor critical quality attributes (CQAs) and critical process parameters (CPPs) in real-time.

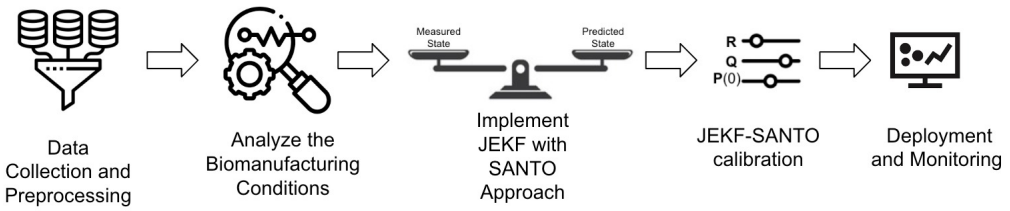

Figure 3.1: The basic steps to develop a soft sensor for bioprocess monitoring based on JEKF-SANTO.

3.4 Empirical Evaluation

In our evaluation, we aim to answer questions **Q1** and **Q2** by assessing the ability of the joint NKEs to simultaneously estimate state variables and unshared parameters in a synthetic monoclonal Antibody (mAb) production dataset and evaluating the improvement in performance with the SANTO approach.

- **Q1)** Can the classic JEKF, JUKF, and JCKF estimate simultaneously the unshared parameters and states of a UMM under biomanufacturing conditions for fast and low-cost bioprocess monitoring? This question aims to experimentally test the Theorem 3.1.
- **Q2)** Can the SANTO approach improve the performance of JEKF, JUKF, and JCKF? This question aims to test whether SANTO can avoid the JEKF, JUKF, and JCKF failure.

3.4.1 Experimental setup

Synthetic Dataset - mAb production

The dataset used in this thesis is synthetic. This is because of the scarcity of open upstream cell culture data, and several studies have used mechanistic models to generate synthetic data for empirical evaluation [74]. 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 [70]. The SD is composed of three runs (A-SD, B-SD, and C-SD) with different cell expansions and

different maximums of mAb (titer) production. The runs of SD can be seen in Figure 3.2, The runs have samples regarding the state variables viable cell density (Xv), glucose (GLC), glutamine (GLN), lactate (LAC), ammonium (AMM) and mAb (titer). They were generated by solving the specific UMM (case 4) of Section 2.1.4 from 0h to 103h with three sets of different parameters (Table 3.1), and the same initial condition (Table 3.2). The run A-SD (red lines in plots of Figure 3.2) was generated using the original parameters proposed by [90], that are the parameters $\mu_{max} = 5.8 \times 10^{-9}(h^{-})$ and QmAb = 7.21 ($\times 10^{-9}$ $mg\ cells^{-1}h^{-1}$). Run B-SD (black lines in plots of Figure 3.2) has the maximum cell expansions and maximum of mAb (titer) production of SD, and they were obtained with the parameters $\mu_{max} = 7.5 \times 10^{-9}(h^{-})$ and QmAb = 9.21 ($\times 10^{-9}$ $mg\ cells^{-1}h^{-1}$). On the other hand, the run C-SD (brown lines in plots of Figure 3.2) 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 ($\times 10^{-9}$ $mg\ cells^{-1}h^{-1}$). Furthermore, the noise data regarding X_V (cell/L) in the runs B-SD and C-SD were generated by adding the Gaussian white noise with standard deviation of 20×10^7 to the data ground truth of runs B-SD and C-SD, represented in black and brown lines in the first plots of Figure 3.2. The X_v of B-SD and C-SD with noise is highlighted in light grey and in orange in the first plot. It is important to point out that X_V samples with Gaussian white noise represent a possible online measurement with sensor including noises. These noises are used to evaluate the performance of the NKEs to estimate mAb and QmAb.

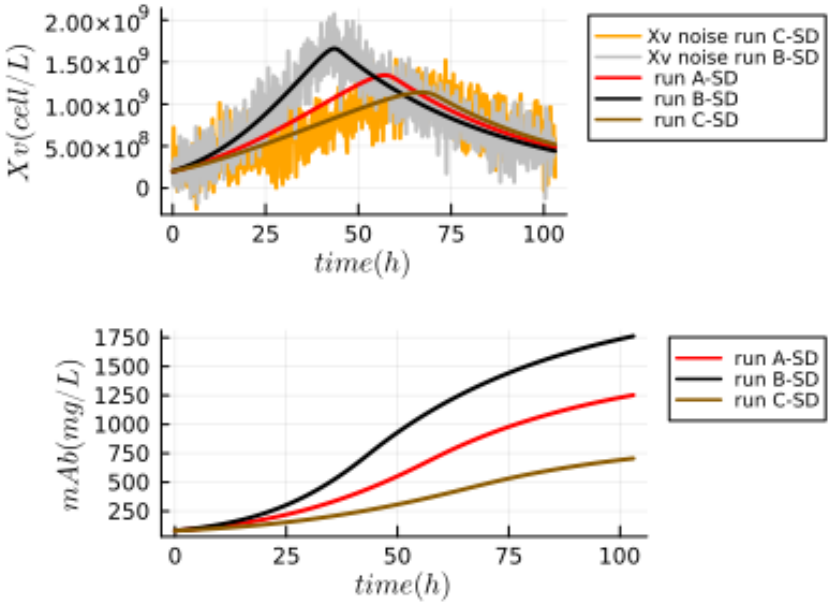

Figure 3.2: Synthetic dataset regarding the mAb production. The run A-SD (red lines) was generated using the original parameters proposed by [90]. Run B-SD (black lines - ground truth) has the maximum cell expansions and maximum of mAb (titer) production of SD. On the other hand, the run C-SD (brown lines - ground truth) has the minimum cell expansions and mAb (titer) production. The X_v of B-SD and C-SD with noise is highlighted in grey and in orange in the first plot. These noises are used to evaluate the performance of the NKEs to estimate mAb and QmAb.

Table 3.1: Initial parameters used in UMM case 4 (Section 2.1.4) to generate the main solutions (ground truth) the runs A-SD, B-SD and C-SD of Synthetic Dataset (SD).

Parameter	**Name**	**run A-SD**	**run B-SD**	**run C-SD**
$\mu_{max}(h^-)$	Maximum growth rate	5.8×10^{-2}	7.5×10^{-2}	5×10^{-2}
$k_{glc}(mM)$	Monod constant glucose	7.5×10^{-1}	7.5×10^{-1}	7.5×10^{-1}
$k_{gln}(mM)$	Monod constant glutamine	7.5×10^{-2}	7.5×10^{-2}	7.5×10^{-2}
$k_{Ilac}(mM)$	Monod constant lactate for inhibition	1.72×10^{2}	1.72×10^{2}	1.72×10^{2}
$k_{Iamm}(mM)$	Monod constant ammonium for inhibition	2.85×10^{1}	2.85×10^{1}	2.85×10^{1}
$\mu_{d,max}(h^-)$	Maximum death rate	3.0×10^{-2}	3.0×10^{-2}	3.0×10^{-2}
$K_{d,amm}(mM)$	Monod constant ammonium for death	1.76	1.76	1.76
$K_{lysis}(h^-)$	Breakdown of cell membranes	5.51×10^{-2}	5.51×10^{-2}	5.51×10^{-2}
$Y_{X,glc}(cells\ mmol^-)$	Yield coefficient cell conc./glucose	1.06×10^{8}	1.06×10^{8}	1.06×10^{8}
$m_{glc}(mmol/cells\ h)$	Glucose maintenance coefficient	4.85×10^{-14}	4.85×10^{-14}	4.85×10^{-14}
$Y_{X,gln}(cells/mmmol)$	Yield coefficient cell conc./glutamine	5.57×10^{8}	5.57×10^{8}	5.57×10^{8}
$\alpha_1(mmols\ cells^-\ h^-)$	Coefficient for m_{gln}	3.40×10^{-13}	3.40×10^{-13}	3.40×10^{-13}
$\alpha_2(mM)$	Coefficient for m_{gln}	4.0	4.0	4.0
$k_{d,gln}(h^-)$	Monod constant glutamine for death	9.6×10^{-3}	9.6×10^{-3}	9.6×10^{-3}
$Y_{lac/glc}(1)$	Yield coefficient lactate/glucose	1.4	1.4	1.4
$Y_{amm/gln}(1)$	Yield coefficient ammonium/glutamine	4.27×10^{-1}	4.27×10^{-1}	4.27×10^{-1}
γ	constant parameter	4.27×10^{-1}	4.27×10^{-1}	4.27×10^{-1}
$Q_{mAb}(mg\ cells^-\ h^-)$	mAb specific production rate	7.21×10^{-9}	9.21×10^{-9}	4.21×10^{-9}

Table 3.2: Initial conditions of state variables of joint NKEs with UMM case4 (Section 2.1.4).

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
QmAb	Specific production rate of mAb	7.21 $\times 10^{-9}$ $mg\ cells^{-1}h^{-1}$

NKEs assessment with synthetic dataset to address RQ1 and RQ2

All NKEs (JEKF-Classic, JUKF-Classic, JCKF-Classic, JEKF-SANTO, JUKF-SANTO, and JCKF-SANTO) used the UMM described in Section 2.1.4 as part of process model (Equation 3.42) and the same initial concentration regarding the state variables, (see Table 3.2). The NKEs were used to correct (estimate) the predictions regarding state variables (Xv and mAb) and to evolve the unshared parameter (QmAb) of the process model simultaneously. This was done using the Xv samples with the noise of the run B-SD and C-SD as the unique measured state variable and the parameters used to generate the run A-SD as initial parameters of the UMM (see Table 3.1). This situation represents a joint estimation problem where the prediction and parameter of the UMM should be corrected by the NKEs based on measured state variable Xv with noise. For example, the initial value used for QmAb is the value of run A-SD (QmAb = 7.21 $\times 10^{-9}$ $mg\ cells^{-1}h^{-1}$), and it should be evolved to the value of run B-SD (9.21 $\times 10^{-9}$ $mg\ cells^{-1}h^{-1}$) based on Xv with the noise of run B-SD. In the case of run C-SD, QmAb should evolve from 7.21 $\times 10^{-9}$ $mg\ cells^{-1}h^{-1}$ to 4.21 $\times 10^{-9}$ $mg\ cells^{-1}h^{-1}$. Furthermore, the Xv (without noise) and mAb samples of run B-SD and C-SD were used as ground truth too, see black and brown lines in Figure 3.2 . In addition, the root mean square percentage error (RMSPE) was used as a metric to assess the similarity between NKEs estimations and the ground truth of run B-SD and C-SD. It is important to point out that in the case of run B-SD, we applied the SANTO approach by adding a positive quantity λ to $\mathrm{P}_{\mathrm{X_v,QmAb}} = \mathrm{P}_{\mathrm{QmAb,X_v}}$ of $\mathbf{P}(0)$, and in case of run C-SD we added a negative quantity. This was done because

$P_{X_v,QmAb}$ is an off-diagonal element of $\mathbf{P}(0)$, and it can be a positive or negative quantity. λ as an off-diagonal element of $\mathbf{P}(t=0)$ can show a positive covariance between two variables, indicating that they tend to increase or decrease together, or a negative covariance, suggesting that they are inversely correlated. In the case of run C-SD, QmAb should decrease from 7.21 $\times 10^{-9}$ $mg\ cells^{-1}h^{-1}$ to 4.21 $\times 10^{-9}$ $mg\ cells^{-1}h^{-1}$, then we added a negative quantity. The details about the design of NKEs related to $\mathbf{R}$, $\mathbf{Q}$, and $\mathbf{P}(0)$ with SD can be found in the next Section 3.4.1.

Table 3.3: RMSPE between NKEs estimations about mAb and ground truth of run B-SD and run C-SD

NKE	RMSPE (run B-SD)	RMSPE (run C-SD)
JEKF-SANTO	1.92%	1.83%
JUKF-SANTO	1.75%	1.80%
JCKF-SANTO	1.11%	1.12%
JEKF-Classic	18.6%	48.1%
JUKF-Classic	18.4%	48.1%
JCKF-Classic	18.4%	48.3%

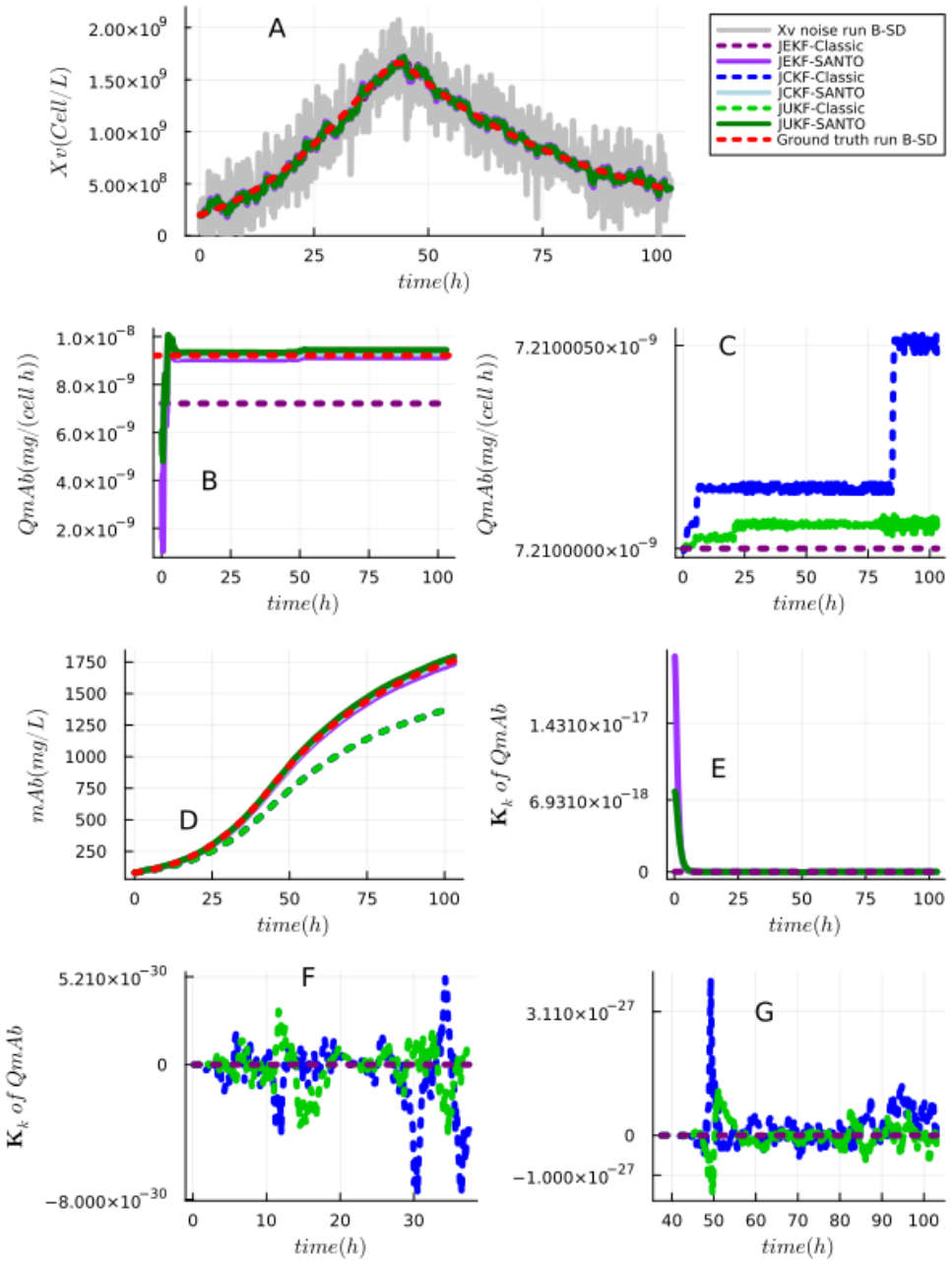

Figure 3.3: Empirical test with **run B-SD**. Plot A demonstrates noise levels and estimations of Xv, closely aligning with the ground truth. Plot B covers QmAb estimations by JEKF-Classic, JEKF-SANTO, JUKF-SANTO and JCKF-SANTO, while Plot C focuses on the classic versions of JUKF-Classic and JCKF-Classic. Plot D shows the mAb estimation done by all NKEs . Plots E, F and G show the Kalmain Gain over time for the NKEs.

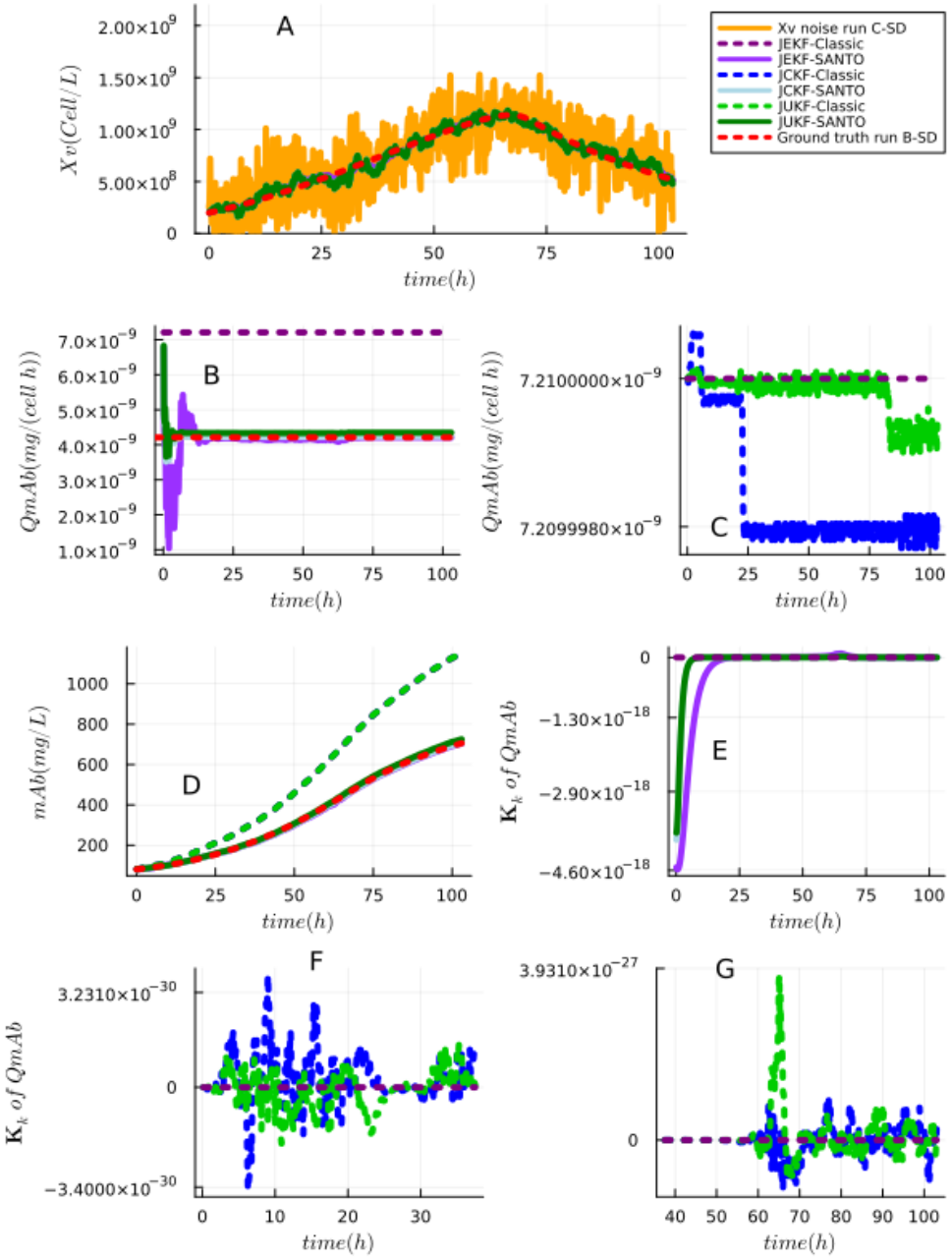

Figure 3.4: Empirical test with **run C-SD**. Plot A demonstrates noise levels and estimations of Xv, closely aligning with the ground truth. Plot B covers QmAb estimations by JEKF-Classic, JEKF-SANTO, JUKF-SANTO and JCKF-SANTO, while Plot C focuses on the classic versions of JUKF-Classic and JCKF-Classic. Plot D shows the mAb estimation done by all NKEs . Plots E, F and G show the Kalmain Gain over time for the NKEs.

NKEs design to address RQ1 and RQ2

The process model (based on UMM case4 of Section 2.1.4) and joint state variable vector used by NKEs (JEKF-Classic, JUKF-Classic, JCKF-Classic, JEKF-SANTO, JUKF-SANTO and JCKF-SANTO) considering $\frac{d\ QmAb}{dt} = 0$ are the following:

$$\mathbf{x}(t)_{case4} = [X_V, X_t, GLC, GLN, LAC, AMM, mAb, QmAb]^T, \tag{3.41}$$

and

$$\mathbf{x}(t_{k/k-1})_{case4} = \mathbf{g}(\mathbf{x}(t_{k-1})_{case4}) + \boldsymbol{q}(t). \tag{3.42}$$

The $\mathbf{P}$(t=0) that was used by the NKEs with run B-SD is in Tables 3.4, 3.5, and 3.6. Furthermore, the $\mathbf{P}$(t=0) that was used by the NKEs with run C-SD are in Tables 3.9, 3.10, and 3.11. It is important to point out that in case of run B-SD, we applied the SANTO approach by adding a small positive quantity to $P_{X_v,QmAb}$ and P_{QmAb,X_v}, and in case of run C-SD we added a small negative quantity to $P_{X_v,QmAb}$ and P_{QmAb,X_v}, see Tables 3.4, 3.5, 3.6, 3.9, 3.10, and 3.11.

The $\mathbf{R}$ and $\mathbf{Q}$ used by the NKEs (for run B-SD) are presented in Tables 3.8 and 3.7. Furthermore, The $\mathbf{R}$ and $\mathbf{Q}$ used by the NKEs (for run C-SD) are presented in Tables 3.13 and 3.12.

It is important point out that all NKEs used a $\mathbf{R}$, $\mathbf{P}$(t=0), and $\mathbf{Q}$ that were obtained by trial and error until achieve positive results in the Normalized Innovations Squared Chi-square Test.

Table 3.4: Standard initial state error covariance matrix (standard $\mathbf{P}$(t=0)) for JEKF-Classic, and JEKF-SANTO with run B of Synthetic Dataset.

Parameter	$\mathbf{P}_{i,i}$ for JEKF-Classic	$\mathbf{P}_{i,i}$ for JEKF-SANTO
P_{X_v,X_v} (c^2/mL^2)	0.00	0.00
P_{X_t,X_t} (c^2/mL^2)	0.00	0.00
$P_{GLC,GLC}$ (mM^2)	0.00	0.00
$P_{GLN,GLN}$ (mM^2)	0.00	0.00
$P_{LAC,LAC}$ (mM^2)	0.00	0.00
$P_{AMM,AMM}$ (mM^2)	0.00	0.00
$P_{mAb,mAb}$ $(mg/L)^2$	0.00	0.00
$P_{QmAb,QmAb}$ (g cells^{-1}h^{-1})^2	3.9e-18	3.9e-18
$P_{X_v,QmAb}$ (c^2/mL^2)(g cells^{-1}h^{-1})	0.0	0.8404

Table 3.5: Standard initial state error covariance matrix (standard $\mathbf{P}$(t=0)) for JUKF-Classic, and JUKF-SANTO with run B of Synthetic Dataset.

Parameter	$\mathbf{P}_{i,i}$ for JCKF-Classic	$\mathbf{P}_{i,i}$ for JCKF-SANTO
P_{X_v,X_v} (c^2/mL^2)	0.00001	0.00001
P_{X_t,X_t} (c^2/mL^2)	0.00001	0.00001
$P_{GLC,GLC}$ (mM^2)	0.00001	0.00001
$P_{GLN,GLN}$ (mM^2)	0.00001	0.00001
$P_{LAC,LAC}$ (mM^2)	0.00001	0.00001
$P_{AMM,AMM}$ (mM^2)	0.00001	0.00001
$P_{mAb,mAb}$ $(mg/L)^2$	0.00001	0.00001
$P_{QmAb,QmAb}$ (g cells^{-1}h^{-1})^2	0.01	10000.01
$P_{X_v,QmAb}$ (c^2/mL^2)(g cells^{-1}h^{-1})	0.0	3.1e-1

Table 3.6: Standard initial state error covariance matrix (standard $\mathbf{P}$(t=0)) for JCKF-Classic, and JCKF-SANTO with run B of Synthetic Dataset.

Parameter	$\mathbf{P}_{i,i}$ for JCKF-Classic	$\mathbf{P}_{i,i}$ for JCKF-SANTO
P_{X_v,X_v} (c^2/mL^2)	0.00001	0.00001
P_{X_t,X_t} (c^2/mL^2)	0.00001	0.00001
$P_{GLC,GLC}$ (mM^2)	0.00001	0.00001
$P_{GLN,GLN}$ (mM^2)	0.00001	0.00001
$P_{LAC,LAC}$ (mM^2)	0.00001	0.00001
$P_{AMM,AMM}$ (mM^2)	0.00001	0.00001
$P_{mAb,mAb}$ $(mg/L)^2$	0.00001	0.00001
$P_{QmAb,QmAb}$ (g cells^{-1}h^{-1})^2	0.01	10000.01
$P_{X_v,QmAb}$ (c^2/mL^2)(g cells^{-1}h^{-1})	0.0	2.9e-1

Table 3.7: Measurement noise variance $\mathbf{R}$ and error covariance matrix of process model ($\mathbf{Q}$) for the JEKF-Classic, JUKF-Classic and JCKF-Classic with run B of Synthetic Dataset.

Parameter	**Name**	JEKF-Classic	JUKF-Classic	JCKF-Classic
R^2 (c^2/mL^2)	Viable cells MNV [1]	$(20\times 10^7)^2$	$(20\times 10^7)^2$	$(20\times 10^7)^2$
Q_{X_v,X_v} (c^2/mL^2)	Viable cells PNV [2]	$(20\times 10^6)^2$	$(20\times 10^6)^2$	$(20\times 10^6)^2$
Q_{X_t,X_t} (c^2/mL^2)	Viable cells PNV [2]	0.001	0.00001	0.00001
$Q_{GLC,GLC}$ (mM^2)	Glucose PNV	0.001	0.00001	0.00001
$Q_{GLN,GLN}$ mM^2	Glutamine PNV	0.001	0.00001	0.00001
$Q_{LAC,LAC}$ (mM^2)	Lactate PNV	0.001	0.00001	0.00001
$Q_{AMM,AMM}$ (mM^2)	Ammonium PNV	0.001	0.00001	0.00001
$Q_{mAb,mAb}$ (VG^2/mL^2)	Monoclonal Antibody (titer) PNV	0.001	00001	0.001
$Q_{QmAb,QmAb}$ (h^{-2})	Specific production rate of mAb	1×10^{-18}	0.01	0.1

[1] MNV—measurement noise value; [2] PNV—process noise value.

Table 3.8: Measurement noise variance $\mathbf{R}$ and error covariance matrix of process model ($\mathbf{Q}$) for the JEKF-SANTO, JUKF-SANTO and JCKF-SANTO with run B of Synthetic Dataset.

Parameter	**Name**	JEKF-SANTO	JUKF-SANTO	JCKF-SANTO
R^2 (c^2/mL^2)	Viable cells MNV [1]	$(20\times 10^7)^2$	$(20\times 10^7)^2$	$(20\times 10^7)^2$
Q_{X_v,X_v} (c^2/mL^2)	Viable cells PNV [2]	$(80\times 10^6)^2$	$(20\times 10^6)^2$	$(20\times 10^6)^2$
Q_{X_t,X_t} (c^2/mL^2)	Viable cells PNV [2]	0.001	0.00001	0.00001
$Q_{GLC,GLC}$ (mM^2)	Glucose PNV	0.001	0.00001	0.00001
$Q_{GLN,GLN}$ mM^2	Glutamine PNV	0.001	0.00001	0.00001
$Q_{LAC,LAC}$ (mM^2)	Lactate PNV	0.001	0.00001	0.00001
$Q_{AMM,AMM}$ (mM^2)	Ammonium PNV	0.001	0.00001	0.00001
$Q_{mAb,mAb}$ (VG^2/mL^2)	Monoclonal Antibody (titer) PNV	0.001	0.001	0.001
$Q_{QmAb,QmAb}$ (h^{-2})	Specific production rate of mAb	0.001	0.001	0.001

[1] MNV—measurement noise value; [2] PNV—process noise value.

Table 3.9: Standard initial state error covariance matrix (standard $\mathbf{P}$(t=0)) for JEKF-Classic, and JEKF-SANTO with run C of Synthetic Dataset.

Parameter	$\mathbf{P}_{i,i}$ for JEKF-Classic	$\mathbf{P}_{i,i}$ for JEKF-SANTO
P_{X_v,X_v} (c^2/mL^2)	0.00	0.00
P_{X_t,X_t} (c^2/mL^2)	0.00	0.00
$P_{GLC,GLC}$ (mM^2)	0.00	0.00
$P_{GLN,GLN}$ (mM^2)	0.00	0.00
$P_{LAC,LAC}$ (mM^2)	0.00	0.00
$P_{AMM,AMM}$ (mM^2)	0.00	0.00
$P_{mAb,mAb}$ $(mg/L)^2$	0.00	0.00
$P_{QmAb,QmAb}$ (g $cells^{-1}h^{-1})^2$	8.9e-18	8.9e-18
$P_{X_v,QmAb}$ (c^2/mL^2)(g $cells^{-1}h^{-1}$)	0.0	-0.1805

Table 3.10: Standard initial state error covariance matrix (standard $\mathbf{P}$(t=0)) for JUKF-Classic, and JUKF-SANTO with run C of Synthetic Dataset.

Parameter	$\mathbf{P}_{i,i}$ for JUKF-Classic	$\mathbf{P}_{i,i}$ for JUKF-SANTO
P_{X_v,X_v} (c^2/mL^2)	0.00001	0.00001
P_{X_t,X_t} (c^2/mL^2)	0.00001	0.00001
$P_{GLC,GLC}$ (mM^2)	0.00001	0.00001
$P_{GLN,GLN}$ (mM^2)	0.00001	0.00001
$P_{LAC,LAC}$ (mM^2)	0.00001	0.00001
$P_{AMM,AMM}$ (mM^2)	0.00001	0.00001
$P_{mAb,mAb}$ $(mg/L)^2$	0.00001	0.00001
$P_{QmAb,QmAb}$ (g cells^{-1}h^{-1})^2	0.01	10000.01
$P_{X_v,QmAb}$ (c^2/mL^2)(g cells^{-1}h^{-1})	0.0	-1.518e-1

Table 3.11: Standard initial state error covariance matrix (standard $\mathbf{P}$(t=0)) for JCKF-Classic, and JCKF-SANTO with run C of Synthetic Dataset.

Parameter	$\mathbf{P}_{i,i}$ for JCKF-Classic	$\mathbf{P}_{i,i}$ for JCKF-SANTO
P_{X_v,X_v} (c^2/mL^2)	0.00001	0.00001
P_{X_t,X_t} (c^2/mL^2)	0.00001	0.00001
$P_{GLC,GLC}$ (mM^2)	0.00001	0.00001
$P_{GLN,GLN}$ (mM^2)	0.00001	0.00001
$P_{LAC,LAC}$ (mM^2)	0.00001	0.00001
$P_{AMM,AMM}$ (mM^2)	0.00001	0.00001
$P_{mAb,mAb}$ $(mg/L)^2$	0.00001	0.00001
$P_{QmAb,QmAb}$ (g cells^{-1}h^{-1})^2	0.01	10000.01
$P_{X_v,QmAb}$ (c^2/mL^2)(g cells^{-1}h^{-1})	0.0	-1.581e-1

Table 3.12: Measurement noise variance **R** and error covariance matrix of process model (**Q**) for the JEKF-Classic, JUKF-Classic and JCKF-Classic with run C of Synthetic Dataset.

Parameter	**Name**	JEKF-Classic	JUKF-Classic	JCKF-Classic
R^2 (c^2/mL^2)	Viable cells MNV [1]	$(20\times 10^7)^2$	$(20\times 10^7)^2$	$(20\times 10^7)^2$
Q_{X_v,X_v} (c^2/mL^2)	Viable cells PNV [2]	$(20\times 10^6)^2$	$(20\times 10^6)^2$	$(20\times 10^6)^2$
Q_{X_t,X_t} (c^2/mL^2)	Viable cells PNV [2]	0.001	0.00001	0.00001
$Q_{GLC,GLC}$ (mM^2)	Glucose PNV	0.001	0.00001	0.00001
$Q_{GLN,GLN}$ mM^2	Glutamine PNV	0.001	0.00001	0.00001
$Q_{LAC,LAC}$ (mM^2)	Lactate PNV	0.001	0.00001	0.00001
$Q_{AMM,AMM}$ (mM^2)	Ammonium PNV	0.001	0.00001	0.00001
$Q_{mAb,mAb}$ (VG^2/mL^2)	Monoclonal Antibody (titer) PNV	0.001	001	0.001
$Q_{QmAb,QmAb}$ (h^{-2})	Specific production rate of mAb	0.001	0.01	0.1

[1] MNV—measurement noise value; [2] PNV—process noise value.

Table 3.13: Measurement noise variance **R** and error covariance matrix of process model (**Q**) for the JEKF-SANTO, JUKF-SANTO and JCKF-SANTO with run C of Synthetic Dataset.

Parameter	**Name**	JEKF-SANTO	JUKF-SANTO	JCKF-SANTO
R^2 (c^2/mL^2)	Viable cells MNV [1]	$(20\times 10^7)^2$	$(20\times 10^7)^2$	$(20\times 10^7)^2$
Q_{X_v,X_v} (c^2/mL^2)	Viable cells PNV [2]	$(20\times 10^6)^2$	$(20\times 10^6)^2$	$(20\times 10^6)^2$
Q_{X_t,X_t} (c^2/mL^2)	Viable cells PNV [2]	0.001	0.00001	0.00001
$Q_{GLC,GLC}$ (mM^2)	Glucose PNV	0.001	0.00001	0.00001
$Q_{GLN,GLN}$ mM^2	Glutamine PNV	0.001	0.00001	0.00001
$Q_{LAC,LAC}$ (mM^2)	Lactate PNV	0.001	0.00001	0.00001
$Q_{AMM,AMM}$ (mM^2)	Ammonium PNV	0.001	0.00001	0.00001
$Q_{mAb,mAb}$ (VG^2/mL^2)	Monoclonal Antibody (titer) PNV	0.001	0.001	0.001
$Q_{QmAb,QmAb}$ (h^{-2})	Specific production rate of mAb	0.001	0.001	0.001

[1] MNV—measurement noise value; [2] PNV—process noise value.

3.4.2 Results

The results are organized by research questions **Q1**, and **Q2**.

Answer to Q1. The estimations regarding the states of Xv and mAb, and the unshared parameter (QmAb) done by classic JEKF, JUKF, and JCKF can be seen in Figures 3.3, and 3.4. In the case of run B-SD (Figure 3.3), the classic JEKF was not able to evolve (update) the QmAb, because the estimations were constant and equal to the initial value of 7.21 $\times 10^{-9}$ $mg\ cells^{-1}h^{-1}$ (purple dash line in Plots B and C). On the other hand, the classic JUKF and JCKF evolved QmAb (green and blue dash line in Plot C) but not significantly to arrive close to ground truth (red dash line in Plot B). Consequently, the classic JEKF, JUKF, and JCKF estimation regarding mAb were far from the ground truth of run B-SD (red dash line in plot D), and they had the highest RMSPE values, see Table 3.3. It is essential to point out that the Kalman gain over time (related to QmAb) that was obtained by JEKF-Classic is constant and equal to zero (purple dash line in Plots E, F, and G) as described in [60]. However, the Kalman gain values obtained by classic JUKF and JCKF presented fluctuation around zero. We created two plots in different scales to visualize the fluctuation of their Kalman gain values (green and blue dash lines in Plots F and G). Furthermore, the obtained results with run C-SD (Figure 3.4) by classic JEKF, JUKF, and JCKF were similar to the ones obtained with run B-SD (Figure 3.3). They also could not simultaneously estimate the Xv, mAb, and QmAb (the estimations were far from the red dash line in Plots B and C), and the Kalman gain values also had the same behavior.

Answer to Q2. The estimations regarding Xv, mAb, and QmAb done by the NKEs (JEKF, JUKF, and JCKF) with the SANTO approach can be seen in Figures 3.3, and 3.4. In the case of run B-SD (Figure 3.3), all NKEs with the SANTO approach evolved the QmAb from the initial value to the ground truth (red dash line in Plot B), and consequently, they estimated the mAb close to the ground truth of run B-SD (red dash line Plot D). This is confirmed by their smallest RMSPE values; see Table 3.3. Furthermore, the Kalman gain values obtained by all NKEs with SANTO presented a normal and stable behavior starting from a positive value and converging to a small value close to zero, see plot E. However, in the case of run C-SD (Figure 3.4), despite all NKEs with SANTO estimate simultaneously mAb and QmAb close to the ground truth (Plots B and D), their Kalman gain values presented an unconventional behavior. They started from a negative value and converged to a small value close to zero; see plot E. This is a consequence of adding a small negative quantity to $P_{X_v,QmAb}$ of $\mathbf{P}(0)$ that includes a negative component in the Kalman gain computation.

3.5 Discussion

Results. Our theoretical and empirical results showed the joint NKE failure with biomanufacturing conditions. These results showed that the classic joint NKE could not estimate the unshared parameters and the state simultaneously since the Kalman gain related to the unshared parameter was constant and equal to zero from the beginning to the end of the processes tested. It is essential to point out that the joint NKE with SANTO approach did not change the probabilistic view of joint NKE, and the minimization cost function in joint NKE remained the same. Therefore, the joint NKE-SANTO approach can be viewed as an artifact that prevents the Kalman gain from becoming zero with the biomanufacturing conditions (failure case). In addition, the joint NKE-SANTO approach only addresses the failure case. It does not solve other issues, such as nonlinearity or high dimensionality, and should be used as a complementary approach. In Figure 3.3 of the document, the failure cases presented show significant deviations, particularly in the estimation of critical parameters like $QmAb$ and mAb. These deviations are critical as they highlight the limitations of the classic JEKF, JUKF, and JCKF, where estimations of parameters and state variables diverged significantly from the ground truth. In fact, the Root Mean Square Percentage Error (RMSPE) values for the classic estimators were notably high, especially for run B-SD and run C-SD, as shown in Table 3.3. Therefore, the results emphasize that, although the errors in the estimation of $QmAb$ are relatively small, they are far from insignificant. These errors directly impact the accuracy of key parameters, such as the monoclonal antibody titer, which is crucial for ensuring product quality in biomanufacturing. This underscores the importance of developing new models or enhanced approaches, such as the SANTO method, to improve the performance of joint NKEs. Notably, the SANTO approach was able to significantly reduce these errors.

Singularities and convergence issues. Beyond the SANTO approach, several methods have been established to tackle singularities and convergence issues in NKE. Rank reduction techniques address ill-conditioned covariance matrices by reducing their dimensionality, thus preventing singularities [143]. Time-correlated noise analysis allows for a more accurate state estimation by adjusting noise covariance matrices based on observed temporal correlations in system noise, providing a more realistic noise model [99]. These methods, used in conjunction with joint NKE-SANTO, offer a comprehensive approach to NKE optimization in challenging scenarios like biomanufacturing.

Low observability. The SANTO approach enables NKEs to effectively track states and update parameters by addressing challenges of low observability, particularly in bioprocess monitoring with limited data and unshared parameters. The theoretical foundation for this approach lies in its modification of the initial state error covariance matrix,

preventing the Kalman gain from becoming zero. In bioprocesses with UMMs and low observability conditions, such as when only one state variable is measured, conventional joint NKEs often fail because the initial covariance matrix lacks sufficient correlation between the measured state variable and unshared parameters, leading to a zero Kalman gain that halts parameter updates. SANTO solves this by introducing a correlation between the measured state and unshared parameters in the initial covariance, ensuring the Kalman gain remains non-zero and allowing both state and parameter updates throughout the estimation process. For the EKF, the SANTO approach works based on modifying the initial conditions in the MRDE, which is used to propagate the state error covariance matrix. By introducing a non-zero covariance between the measured state and unshared parameters, SANTO ensures that the Kalman gain remains positive, preventing the failure of parameter estimation due to zero covariance in low-observability settings. For the CKF and UKF, the SANTO approach enhances these filters by modifying the initial state error covariance matrix, ensuring that the non-zero correlation between measured states and unshared parameters is preserved during the propagation of cubature or sigma points. This prevents the Kalman gain from becoming zero and allows continuous parameter updates. Thus, SANTO ensures that key covariance information is retained, enabling continuous updates and improving NKE performance across EKF, CKF, and UKF.

Dual NKE. It is important to point out that SANTO approach was designed for joint NKE. In contrast to joint NKE, the dual NKE employs two consecutive NKEs, separating the estimation of system states and parameters [66]. This separation can be advantageous in certain scenarios, but joint NKE offers two important benefits, particularly in the context of the practical implementation of adaptive controllers using microcontrollers in bio-manufacturing that requires numerically economical and robust algorithms such as joint NKE [10, 111]. First, joint NKE avoids the computational overhead associated with running two separate filters, as in dual NKE, enhancing computational efficiency [98]. Lastly, the single-filter structure of joint NKE is simpler to tune compared to the dual-filter approach of dual NKE [44].

Unconventional Kalman gain values. The SANTO approach improved the performance of JEKF, JUKF and JCKF, achieving lower RMSPE values than the classic versions in both cases (runs B-SD and C-SD). Nevertheless, the empirical test using run C-SD showed unconventional Kalman gain values for all NKEs with the SANTO approach, starting from a negative value and converging to near zero. This is the result of $\lambda < 0$ due to the need to decrease the initial value of the unshared parameter (QmAb) in the run C-SD. However, this unconventional values are not problematic and did not lead to system unrealiability because the filters achived positive results in the normalized innovations squared chi-square test that is a filter consistency test (see Table 5.6).

Computational complexity. Considering n as the number of states in the system (dimension of the state vector $\mathbf{x}_k \in \mathbb{R}^n$), we have that the classical EKF is of order $\mathcal{O}(n^3)$ and particle filter is $\mathcal{O}(Nn^2)$, where N is the number of particles Thus, a PF is more complex than an EKF by a factor of N/n depending on the number of particles [147]. In addition, UKF and CKF, compared to EKF, offers better accuracy for highly nonlinear systems without requiring Jacobian computations, and the computational complexity of the UKF, CKF and EKF are proportional to $\mathcal{O}(n^3)$ [27,102,129,161]. Therefore, as SANTO approach did not change the probabilistic view of joint NKE, and the minimization cost function in joint NKE remained the same, we have that the computational complexity of the joint UKF, CKF, and EKF remained the same.

Joint NKEs for Time-Evolving Parameters. The dynamics of parameters being estimated using joint filters involve both state tracking and parameter estimation, which typically evolve over time. In the case of joint NKEs, states and parameters are concatenated into a single state vector, allowing simultaneous estimation. The parameters are treated as time-varying and evolve with each time step to account for changes in the system or external factors influencing the process. These parameters generally need to be updated over time, particularly in systems where bioprocess conditions or mechanisms may change. The estimation is done in a recursive manner, with each step updating both the states and parameters based on the latest available data. This is critical for ensuring that the model remains accurate and adaptable to new conditions, especially when the system dynamics or measurement data evolve.

3.6 Summary

Our analysis and empirical results demonstrated that the classic DEKF, JEKF, JUKF, and JCKF are inefficient under the conditions studied. The SANTO approach, however, improved the performance of JEKF, JUKF and JCKF, achieving lower RMSPE values than the classic versions in both cases (runs B-SD and C-SD).

Chapter 4

Batch Bayesian Auto-Tuning for Nonlinear Kalman Estimators

4.1 Overview

This Chapter presents the proposed approach to address the RQ2. Optimal NKE performance relies on tuning five components: process noise covariance, measurement noise covariance, initial state noise covariance, initial state conditions, and UMM parameters. However, traditional auto-tuning approaches based on normalized estimation error squared (NEES) or normalized innovation squared (NIS) are inefficient for estimating all NKE components due to their dependence on ground truth state models or a subset of measured data. In addition, manual tuning is labor-intensive and prone to errors. To address these challenges, we introduce a novel approach called batch Bayesian auto-tuning (BAT) for NKEs, which utilizes all available measured data during the tuning process by defining a comprehensive posterior distribution of all NKE components outside the recursive process. This approach is based on the equivalence between posterior distributions in batch and recursive Bayesian inference. Empirical validation on a synthetic bioprocess dataset demonstrates that BAT significantly improves the accuracy of NKE estimations compared to baseline methods, showing that BAT can effectively optimize NKE tuning, thereby enhancing performance and reliability in practical applications. This research identifies the limitations of traditional methods in estimating unshared parameters in biomanufacturing processes and highlights the necessity of auto-tuning all NKE components, including unshared parameters, for real-time monitoring of novel bioprocesses with unknown mechanisms. The BAT approach integrates offline measurements as additional data related to

state variables, refining the tuning process and improving the estimation of challenging parameters.

The main contributions of this chapter are:

- **BAT:** To the best of our knowledge, we are the first to introduce an approach (BAT posterior) 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, and CKF, by enabling the use all available measured data, not just those selected for generating innovation errors.

- **Enhanced Performance:** The BAT approach, in addition to addressing the RQ2, also addresses the limitation of the SANTO approach. The empirical results show that JEKF, JUKF and JCKF performed better only with the SANTO approach but with an unconventional behavior of Kalman gain in one of the studied cases, starting from a negative value and converging to near zero. BAT solves this issue under the same case study by directly estimating the unshared parameter. Furthermore, BAT can be used to calibrate the SANTO approach.

- **Accuracy:** Our empirical evaluation of a synthetic bioprocess dataset demonstrates that BAT outperforms the baselines in the auto-tuning process by significantly enhancing the accuracy of a tuned EKF related to state variables strongly associated with unshared parameters. The code and data used in this work are available on GitHub repository [1] to facilitate reproducibility.

- **Reduced Data Requirements**: BAT is an offline approach with reduced data requirements applicable to any new biomanufacturing condition. The empirical evaluation across all testing sets shows that BAT performs well with limited data, addressing the limitation of the biopharma industry to generate a large amount of data to develop monitoring tools

- **Unimodal Posterior Distributions**: Unlike baseline methods, BAT consistently estimates unimodal posterior distributions, which closely align with ground truth values, reducing ambiguity in parameter estimation.

- **Instructional Application**: 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 for instructional purposes.

[1] https://github.com/cristovaoiglesias/BAT

4.2 Related Work

The literature has reported several techniques to tune some NKE components; however, to 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 [140]. 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 [14].

State augmentation (Joint NKE). It is an online approach that aims to auto-tune the parameters $\boldsymbol{\theta}$ of a UMM in NKEs by expanding the state variable vector to include both the system states $\mathbf{x}_k$ and $\boldsymbol{\theta}$ [9, 55, 166]; see Section 2.4. This approach is widely used due to the need to correct the prediction of a UMM regarding state variables and to update the UMM by evolving its parameters based on the corrections done [44]. However, the increased dimensionality and complexity of state augmentation can make accurately estimating states and parameters challenging. Furthermore, this approach fails to estimate the state and the unshared parameters $\boldsymbol{\theta}_{up}$ simultaneously for fast and low-cost bioprocess monitoring; see Section 3.2. The SANTO approach, however, can improve the performance of joint NKEs. Nevertheless, in some cases, SANTO can lead to an unconventional Kalman gain values. A solution to avoid this failure case is to estimate $\boldsymbol{\theta}_{up}$ during the NKE auto-tuning process. Therefore, we explore the case where the SANTO approach results in unconventional Kalman gain values (as reported in Section 3.4.2) in our empirical evaluation to show that BAT is a solution to this problem.

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 [11, 19, 22, 88, 135]. Various optimization methods, such as Genetic Algorithms [118], Particle Swarm Optimization [154], and particularly Bayesian Optimization (BO) [22–24], have been proposed to optimize objective functions. In [14], the authors present an objective function for tuning the EKF using chi-square tests on NIS samples. This novel approach combines several metrics, including RMSE, estimation error covariance, and consistency measures, into a weighted objective function to avoid local minima and ensure filter consistency. This method is detailed in Section A.3.1 of the Appendix, and serves as a baseline in our empirical evaluation due to its focus on innovation errors and ensures filter consistency. Additionally, it represents the latest advancement in

the field. In addition, the method proposed in [1] is also used as a baseline and detailed in Section A.3.2 of the Appendix. The authors proposed minimizing the residual prediction error, which uses the outputs of the EKF update step and can be adapted to the use of all measurement data available (which motives its use as a baseline). 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 "eliminate" $\mathbf{M}$ (when it has the same noises for all state variables) before applying this tuning approach, see Section A.3.2 of the Appendix.

However, both methods presented previously (objective function [14] and residual prediction error minimization [1]) tend to estimate multimodal distributions for NKE components. 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. This enables comprehensive data utilization, facilitating the estimation of unimodal distributions of 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 [12, 54, 114].

4.3 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 based on the concepts presented in Sections 2.3.1, 2.3.3 and 2.3.4.

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 2.2. 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 [71]. 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 a measurement model in the batch Bayesian inference (BBI) framework outside of the NKE (see Figure 4.1) as,*

$$\mathbf{y}_k = \mathbf{h}(\hat{\mathbf{x}}_k^{\mathrm{NKE_{PS}}}) + \boldsymbol{\delta}_k, \tag{4.1}$$

given

$$\langle \hat{\mathbf{x}}_k^{\mathrm{NKE_{PS}}}, \mathbf{P}_k^{\mathrm{NKE_{PS}}} \rangle = \mathrm{NKE_{PS}}(\mathbf{g}, \boldsymbol{\theta}, t_{k-1}, t_k, \hat{\mathbf{x}}_{0,k-1} = \hat{\mathbf{x}}_{k-1}^{\mathrm{NKE_{US}}}, \mathbf{P}_{0,k-1} = \mathbf{P}_{k-1}^{\mathrm{NKE_{US}}}, \mathbf{Q}_{k-1}, \mathbf{R}_{k-1}). \tag{4.2}$$

The $\hat{\mathbf{x}}_k^{NKE_{PS}}$ (predicted mean of state variables) is the solution obtained by the **NKE prediction step** at time t_k, beside of predicted error covariance matrix $\mathbf{P}_k^{NKE_{PS}}$, see Figure 4.1. 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 $\mathbf{h}(\cdot)$ 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 UMM described by $\mathbf{g}(\hat{\mathbf{x}}_{0,k-1} = \hat{\mathbf{x}}_{k-1}^{NKE_{US}}; \boldsymbol{\theta})$, $\mathbf{Q}_{k-1}$, and the results of the **NKE update step** at time 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 4.1. 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 t_{k-1} to t_k .

Then, based on the previous assumptions, and assuming the conditional independence of measurement $\mathbf{y}_k$ in Equation 4.1, we can define the BAT likelihood of measurements such that

$$\begin{aligned} p(\mathbf{y}|\mathbf{h}(\hat{\mathbf{x}}^{NKE_{PS}}), \boldsymbol{\sigma}^2) = \prod_{k=1}^{T} p(\mathbf{y}_k|\mathbf{h}(\hat{\mathbf{x}}_k^{NKE_{PS}}), \boldsymbol{\sigma}^2) = \\ \prod_{k=1}^{T} p(\mathbf{y}_k|\mathbf{h}(NKE_{PS}(\mathbf{g}, \boldsymbol{\theta}, t_{k-1}, t_k, \hat{\mathbf{x}}_{0,k-1}, \mathbf{P}_{0,k-1}, \mathbf{Q}_{k-1}, \mathbf{R}_{k-1})), \boldsymbol{\sigma}^2). \end{aligned} \tag{4.3}$$

BAT Priors. Given the Equation 4.3, 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 estimates 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 4.3 to have a likelihood of the measurement $\mathbf{y}_k$ given all NKE components, we have

$$p(\mathbf{y}|\mathbf{h}(\hat{\mathbf{x}}^{NKE_{PS}}), \boldsymbol{\sigma}^2) = \prod_{k=1}^{T} p(\mathbf{y}_k|\boldsymbol{\theta}, \hat{\mathbf{x}}_{0,0}, \mathbf{P}_{0,0}, \mathbf{Q}_{k-1}, \mathbf{R}_{k-1}, \boldsymbol{\sigma}^2), \tag{4.4}$$

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

$$\begin{aligned} p(\boldsymbol{\theta}, \hat{\mathbf{x}}_{0,0}, \mathbf{P}_{0,0}, \mathbf{Q}, \mathbf{R}, \boldsymbol{\sigma}^2|\mathbf{y}) \propto p(\hat{\mathbf{x}}_{0,0}) \times p(\mathbf{P}_{0,0}) \times \\ p(\boldsymbol{\sigma}^2) \times \prod_{k=1}^{T} p(\mathbf{y}_k|\boldsymbol{\theta}, \hat{\mathbf{x}}_{0,0}, \mathbf{P}_{0,0}, \mathbf{Q}_{k-1}, \mathbf{R}_{k-1}, \boldsymbol{\sigma}^2) \times p(\mathbf{Q}_{k-1}) \times p(\mathbf{R}_{k-1}). \end{aligned} \tag{4.5}$$

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 4.5 can be sampled by MCMC methods.*

Consequently, we have the following Theorem.

Theorem 4.1. *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.*

The proof of Theorem 4.1 is the following.

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

$$\mathbf{y}_k = \mathbf{h}(\hat{\mathbf{x}}_k^{NKE_{PS}}) + \boldsymbol{\delta}_k, \tag{4.6}$$

  Where $\boldsymbol{\delta}_k \sim N(0, \boldsymbol{\sigma}^2)$, and $\mathbf{h}(\cdot)$ 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

$$\begin{aligned} p(\mathbf{y}|\mathbf{h}(\hat{\mathbf{x}}^{NKE_{PS}}), \boldsymbol{\sigma}^{\mathbf{2}}) = \prod_{k=1}^{T} p(\mathbf{y}_k|\mathbf{h}(NKE_{PS}(\mathbf{g}, \boldsymbol{\theta}, t_{k-1}, t_k, \hat{\mathbf{x}}_{0,k-1}, \mathbf{P}_{0,k-1}, \mathbf{Q}_{k-1}, \mathbf{R}_{k-1}))) = \\ \prod_{k=1}^{T} p(\mathbf{y}_k|\boldsymbol{\theta}, \hat{\mathbf{x}}_{0,0}, \mathbf{P}_{0,0}, \mathbf{Q}_{k-1}, \mathbf{R}_{k-1}, \boldsymbol{\sigma}^{\mathbf{2}}) = \\ \prod_{k=1}^{T} p(\mathbf{y}_{online,k}, \mathbf{y}_{offline,k}|\boldsymbol{\theta}, \hat{\mathbf{x}}_{0,0}, \mathbf{P}_{0,0}, \mathbf{Q}_{k-1}, \mathbf{R}_{k-1}, \boldsymbol{\sigma}^{\mathbf{2}}), \end{aligned} \tag{4.7}$$

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

$$p(\boldsymbol{\theta}, \hat{\mathbf{x}}_{0,0}, \mathbf{P}_{0,0}, \mathbf{Q}, \mathbf{R}, \boldsymbol{\sigma^2} | \mathbf{y}_{online}, \mathbf{y}_{offline}) \propto p(\hat{\mathbf{x}}_{0,0}) \times p(\mathbf{P}_{0,0}) \times$$
$$p(\boldsymbol{\sigma}^2) \times \prod_{k=1}^{T} p(\mathbf{y}_{online,k}, \mathbf{y}_{offline,k} | \boldsymbol{\theta}, \hat{\mathbf{x}}_{0,0}, \mathbf{P}_{0,0}, \mathbf{Q}_{k-1}, \mathbf{R}_{k-1}, \boldsymbol{\sigma^2}) \tag{4.8}$$
$$\times p(\mathbf{Q}_{k-1}) \times p(\mathbf{R}_{k-1}).$$

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.

□

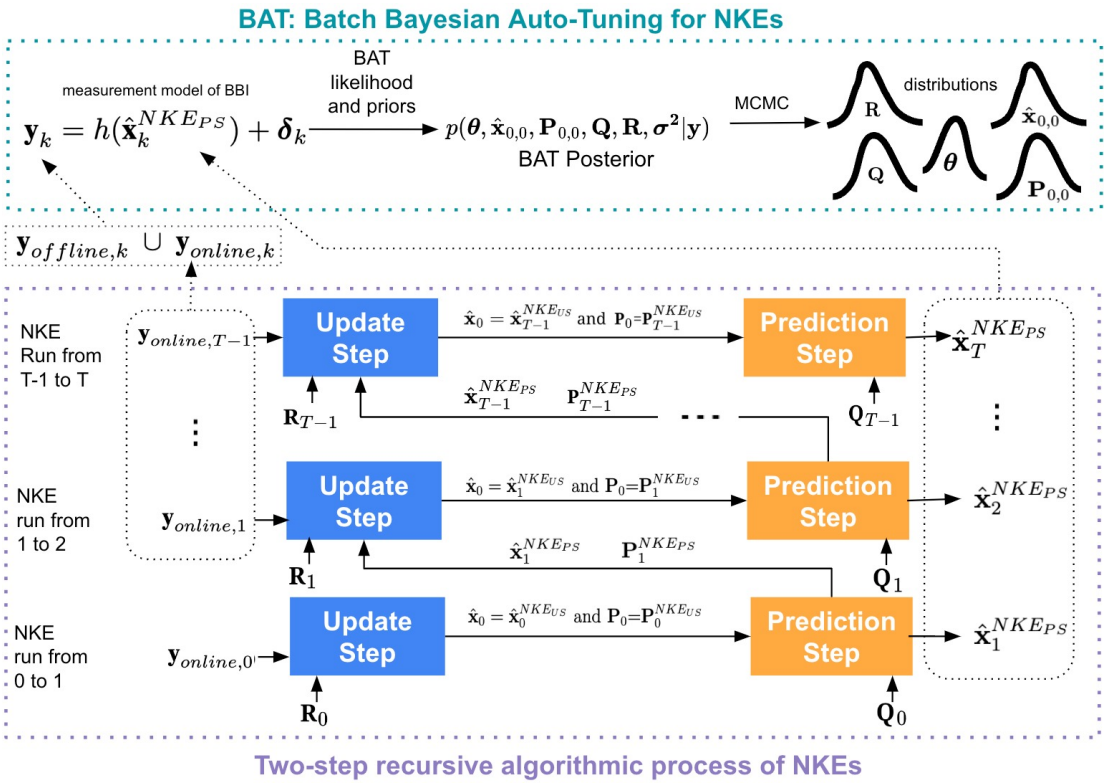

Figure 4.1: BAT: Batch Bayesian Auto-Tuning for Nonlinear Kalman Estimators. The online measurements $\mathbf{y}_{online,k}$ (used to generate innovation errors) and offline measurements $\mathbf{y}_{offline,k}$ of other state variables compose a full set $\mathbf{y}_k$ of measured data of all state 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 4.4) and priors to obtain the BAT posterior (Equation 4.5). Then, samples can be extracted by MCMC methods to auto-tune NKEs by extracting the mean values from the obtained distribution of $\boldsymbol{\theta}, \hat{\mathbf{x}}_{0,0}, \mathbf{P}_{0,0}, \mathbf{Q}, \mathbf{R}$.

4.3.1 BAT for EKF

Here, we describe how BAT can be applied to EKF. However, a similar approach can be used for UKF and CKF (see Section A.2 in Appendix). 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

$$\mathbf{y}_k = \mathbf{h}(\hat{\mathbf{x}}_k^{EKF_{PS}}) + \boldsymbol{\delta}_k, \tag{4.9}$$

$$\langle \hat{\mathbf{x}}_k^{NKE_{PS}}, \mathbf{P}_k^{NKE_{PS}} \rangle = EKF_{PS}(\hat{\mathbf{x}}_{0,k-1} = \hat{\mathbf{x}}_{k-1}^{EKF_{US}}, \mathbf{g}, \boldsymbol{\theta}, t_{k-1}, t_k, \mathbf{Q}_{k-1}, \mathbf{R}_{k-1}, \mathbf{P}_{0,k-1} = \mathbf{P}_{k-1}^{EKF_{US}}) \tag{4.10}$$

where $\boldsymbol{\delta}_k$ is a D $\times$ 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 UMM $\mathbf{g}(\hat{\mathbf{x}}_{0,k-1} = \hat{\mathbf{x}}_{k-1}^{EKF_{US}}; \boldsymbol{\theta})$ 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 an 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 4.10 as following

$$\langle \hat{\mathbf{x}}_k^{EKF_{PS}}, \mathbf{P}_k^{EKF_{PS}} \rangle = ODESolve^{EKF_{PS}}(\hat{\mathbf{x}}_{0,k-1} = \hat{\mathbf{x}}_{k-1}^{EKF_{US}}, \mathbf{g}, \boldsymbol{\theta}, t_{k-1}, t_k, \mathbf{Q}_{k-1}, \mathbf{R}_{k-1}, \mathbf{P}_{0,k-1} = \mathbf{P}_{k-1}^{EKF_{US}}). \tag{4.11}$$

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:

$$\mathbf{y}_k \approx MVN(\mathbf{h}(\hat{\mathbf{x}}_k^{EKF_{PS}}), \sum(\boldsymbol{\sigma}^{\mathbf{2}})), \tag{4.12}$$

$$\mathbf{y}_k \approx MVN(\mathbf{h}(ODESolve^{EKF_{PS}}(\hat{\mathbf{x}}_{0,k-1}, \mathbf{g}, \boldsymbol{\theta}, t_{k-1}, t_k, \mathbf{Q}_{k-1}, \mathbf{R}_{k-1}, \mathbf{P}_{0,k-1})), \sum(\boldsymbol{\sigma}^{\mathbf{2}})), \tag{4.13}$$

where $\sum(\boldsymbol{\sigma}^{\mathbf{2}})$ is a diagonal matrix with variances related to each observed state variable in $\mathbf{y}_k$. Hence, the likelihood function is given by

$$\mathcal{L}(\mathbf{y}|\mathbf{h}(\hat{\mathbf{x}}_k^{EKF_{PS}}), \boldsymbol{\sigma^2}) = \prod_{k=1}^{T} MVN(\mathbf{y}_k; \mathbf{h}(\hat{\mathbf{x}}_k^{EKF_{PS}}), \sum(\boldsymbol{\sigma^2})) =$$

$$\prod_{k=1}^{T} MVN(\mathbf{y}_k; \mathbf{h}(ODESolve^{EKF_{PS}}(\hat{\mathbf{x}}_{0,k-1}, \mathbf{g}, \boldsymbol{\theta}, t_{k-1}, t_k, \mathbf{Q}_{k-1}, \mathbf{R}_{k-1}, \mathbf{P}_{0,k-1})), \sum(\boldsymbol{\sigma^2})), \tag{4.14}$$

and the posterior density is

$$p(\boldsymbol{\theta}, \boldsymbol{\sigma}^2, \hat{\mathbf{x}}_{0,set}, \mathbf{Q}, \mathbf{R}, \mathbf{P}_{0,set}|\mathbf{y}) \propto p(\boldsymbol{\theta}) \times p(\boldsymbol{\sigma}^2) \times \prod_{k=1}^{T} MVN(\mathbf{y}_k; \mathbf{h}(\hat{\mathbf{x}}_k^{EKF_{PS}}), \sum(\boldsymbol{\sigma^2}))$$
$$\times p(\hat{\mathbf{x}}_{0,k-1}) \times p(\mathbf{P}_{0,k-1}) \times p(\mathbf{Q}_{k-1}) \times p(\mathbf{R}_{k-1}). \tag{4.15}$$

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 4.15. This process iteratively refines the estimates of all NKE components, leading to an auto-tuned EKF better adapted to the specific characteristics of the observed data.

It is essential to point out that in our experiments with EKF, we used the Bogacki-Shampine 3/2 method (BS3) as an ODE solver. The BS3 ODE solver is recommended for efficiently solving non-stiff problems at higher tolerances, mainly 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.

4.4 Empirical Evaluation

We executed detailed experiments to answer the following questions and understand the effectiveness of the proposed approach:

- **Q1**) How does the performance of BAT compare with the performance of baseline methods? and

- **Q2**) Does the BAT posterior density (Equation 4.5) enable an efficient sampling with MCMC?

These 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 during the execution of an empirical task. The EKF was chosen for this empirical evaluation due to its status as the most popular and widely used NKE. However, in the Section A.2.1 and A.2.2 of the Appendix, we show examples of the BAT application with UKF and CKF in a classical problem.

4.4.1 Experimental Setup

Datasets

The synthetic dataset (SD) has data regarding Monoclonal Antibody (mAb) productions [70] such as those used in the evaluation of the SANTO approach (Section 3.4.1). The synthetic dataset is composed of three runs: one training D1 set and two testing sets (D2 and D3). The runs have data related to 7 state variables (viable cell density (Xv), total cell (Xt), glucose (GLC), glutamine (GLN), lactate (LAC), ammonium (AMM), and mAb). The training set D1 has noisy data related to all state variables, representing the case where we have the measured data to generate innovation errors (Xv) and 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 4.2. In training set D1, the Gaussian white noises with standard deviations (std) of $10{\times}10^7$, $10{\times}10^7$, 1, 0.5, 2.0, 0.1, and 40.5 were added to the main solution (ground truth) D of state variables Xv, Xt, GLC, GLN, LAC, AMM and mAb, respectively, from 0h to 103h. The testing sets D2 and D3 have the Xv noisy data generated from white Gaussian noises with standard deviations (std) of $10{\times}10^7$, see Figure 4.2. 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 an NKE with a small dataset (offline measurements), apply the NKE in online mode with a high frequency of noisy data. The training set D1 has a sample rate of 7h for all state variables, and the testing sets D2 and D3 have a sample rate of 7h and 7.5 min respectively, see Figure 4.2. The training set D1 and testing set D2 represent a very small dataset with 15 data points for each state variable. This kind of small dataset commonly results from offline measurements for all state variables. Where the state variables Xv, GLC, GLN, LAC, AMM, and mAb are obtained through offline measurements by a researcher at a low sampling rate in the order of hours or days. The

offline data mimicked the measurements from a hemocytometer (for Xv and Xt), YSI Bioprofiler 200 (for GLC, GLN, LAC, and AMM), and ELISA (for mAb) [76]. The testing set D3 represent a online measurements of the state variable Xv. Where Xv is obtained by online measurements from sensors in the bioreactor at the high sampling rate in the order of minutes. The online data mimic measurements from a capacitance sensor or Raman spectroscopy [107, 115, 128]. The procedure to generate the synthetic dataset used in this thesis is the same as used for the empirical test of SANTO approach in Section 3.4.1. The training and testing sets of runs of the synthetic dataset (D1, D2, and D3) 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 [90] from 0h to 103h with the parameters of run C-SD presented in Table 3.1, and with the same initial concentrations of states variables (Xv, GLC, GLN, LAC, AMM, and mAb) presented in Table 3.2. 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. It is important to point out that we used the parameters of run C-SD because it represents a challenging scenario for the SANTO approach and is useful for showing the efficacy of BAT.

Baselines

Our baseline methods are based on the objective function (OF) [14] and residual prediction error (RPE) minimization [1] (see Sections A.3.1 and A.3.2 in Appendix A.3 for more details about these methods). The baseline methods comprise a total of 8 methods and are divided into two groups. The first group comprises the numerical optimization solvers to optimize the OF and RPE. The optimization methods used in the empirical test are Bayesian optimization (BO) [22–24], particle swarm (PS), [162], and genetic algorithm (GA) [118]. When applied to OF and RPE, these methods are denoted as OF-BO and RPE-BO for Bayesian optimization, OF-PS and RPE-PS for particle swarm, and OF-GA and RPE-GA in the case of genetic algorithm. More details about these optimization methods can be found in Section A.3.3 of the Appendix. The last group is formed by the No-U-Turn Sampler (NUTS) approach [51]. The No-U-Turn Sampler (NUTS) approach is the MCMC method used to sample the distribution of $\boldsymbol{\Theta} = \{\mathbf{x}_{0,0}, \boldsymbol{\theta}, \mathbf{Q}, \mathbf{R}\}$ that minimizes the OF and RPE through a probabilistic model (see Section A.3.4 in Appendix A.3 for more details). We refer to these baselines as OF-NUTS and RPE-NUTS.

Task

The training and testing sets of SD are used in empirical task (see Figure 4.2). It is important to point out that in the empirical task only Xv (noisy data) is used as the measured state variable to generate innovation errors during the execution of the EKF during the process of auto-tuning.

It comprises three steps:

1. Using the BAT and baselines with training set D1 to estimate the optimal $\mathbf{\Theta} = \{\mathbf{x}_{0,0}, \boldsymbol{\theta}, \mathbf{Q}, \mathbf{R}\}$ 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 UMM (see Equation 2.4) used in the EKF. It is important to note that BAT is designed to be able to use the noisy data related to the state variable (Xv) that generates innovation errors and the additional noisy data (offline measurements of Xt, GLC, GLN, LAC, AMM, and mAb) associated with the other state variables that do not generate innovation errors. The baselines, due their limitations, can only use the noisy data related to Xv to auto-tune the EKF during the execution of the task (see Sections A.3.1 and A.3.2 in Appendix for more details about the limitations of baselines).

2. Testing the filter consistency of the designed EKFs with the estimated $\mathbf{\Theta}$ using the testing sets D2 and D3, and

3. Assessing the accuracy between designed EKFs estimations and ground truth values of the testing set D2 and D3 through the root-mean-square percentage error (RMSPE).

It is essential to point out that the empirical task in our experiments represents a challenging scenario where the stage augmentation approach does not succeed, and this is the motivation to include this task in our experiments; see Section 4.2. For example, the application of state augmentation with EKF to simultaneously estimate the following states LAC, AMM, mAb, and the following parameters $QmAb, Y_{lac,glc}$ and $Y_{amm,gln}$ of UMM (Equation 2.4) "fails" with the biomanufacturing conditions described previously. Therefore, this is the motivation for the empirical task, 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 (Equation 2.4) with an EKF.

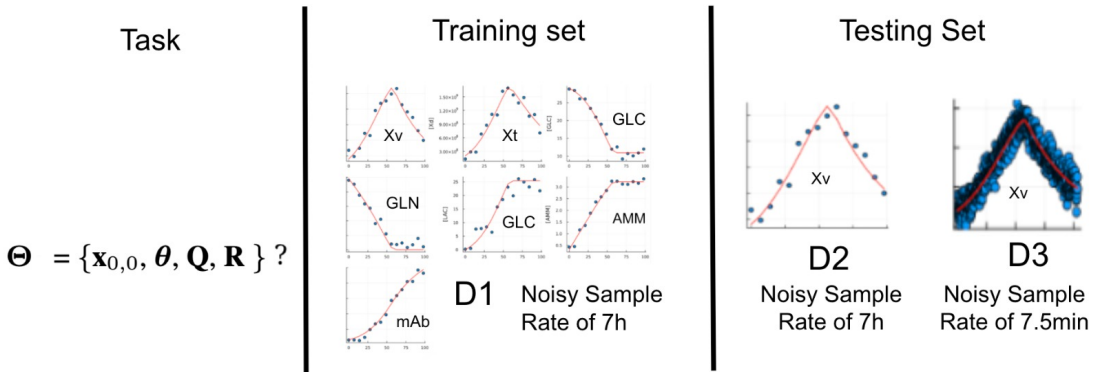

Figure 4.2: Datasets and empirical task. The main task requires the estimation of $\mathbf{\Theta} = \{\mathbf{x}_{0,0}, \boldsymbol{\theta}, \mathbf{Q}, \mathbf{R}\}$ using the training set D1 (representing offline measurements) that has a sample rate of 7h related to noisy data of 7 state variables (blue) where each one has a specific noise. The testing sets used to assess the designed EKF with the estimated $\mathbf{\Theta}$ are D2 (representing offline measurements) and D3 (representing online measurements of Xv). They have sample rates of 7h and 7.5 min, respectively. It is important to point out that the red lines in the plots represent the main solution (ground truth) D.

Metrics

Filter consistency test

First, to specifically answer Q1, we utilize the following filter consistency test as the metric for comparing the final estimations of $\mathbf{\Theta}$ achieved by both BAT and baselines. In addition, we used RMSPE to asses the estimations done by the designed EKFs in the empirical task. To evaluate the performance of designed EKF by BAT and baselines, we can use performance measures based on the innovation process [11, 132, 137]. For a properly functioning filter, the innovation sequence $e_{y,k}$ should have a zero mean and be white, with a covariance denoted as $\mathbf{S}_k$ [11, 132, 137]. 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

$$\mu(NIS) = \frac{1}{N}\sum_{k=1}^{N} e_{y,k}\mathbf{S}_k^{-}e_{y,k} \tag{4.16}$$

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

$$P(N \times \mu(NIS) \in [r1, r2] | H_0) = 1 - \alpha \tag{4.17}$$

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 [11, 132, 137]

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

- 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: $P(N \times \mu(NIS) \in [745.3, 904.39] | H_0) = 1 - \alpha$?

Auto-correlation function (ACF) plot

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 [17]. 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 [17]. 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 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.

Ground Truth values of $\boldsymbol{\Theta}$

Lastly, the ground truth values of $\boldsymbol{\Theta}$ were defined through a consistency test of EKF. The ground truth values of $\boldsymbol{\Theta}$ are defined in Table 4.6. They allow the design of a consistent EKF with all testing sets. In Table 4.5, we can see that ground truth values were acceptable in FCT2 and FCT3 defined in Section 4.4.1. In addition, the N $\times\mu(NIS)$ obtained by the designed EKF with ground truth values of $\boldsymbol{\Theta}$ for FCT1 was 12.35 (acceptable). The setup information about the BAT and the baselines can be seen in Section 4.4.1.

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

It is important to note that BAT and the baselines used the same type of priors to guarantee a fair comparison. The priors are divided into two groups: informative priors using normal distribution and uninformative priors (less informative priors) using uniform distribution (see Table 4.2). These two groups were used to assess the performances of BAT and baselines (OF-NUTS and RPE-NUTS). The informative priors were used to define the priors of $\mathbf{Q}$, and the uninformative priors were used to define the priors of $\mathbf{R}$, $\mathbf{x}_{0,0}$=$\{\mathrm{u0}_{LAC}, \mathrm{u0}_{mAb}\}$ that represent the initial concentrations of LAC and mAb, and $\boldsymbol{\theta}$ =$\{\mathrm{Y}_{lac,glc}, \mathrm{Y}_{amm,gln}, \lambda\}$ that represent three unshared parameters of the UMM (see Equation 2.4). Furthermore, the baselines (OF-BO, OF-PS, OF-GA, RPE-BO, RPE-PS, and RPE-GA) used the lower and upper bound defined in Table 4.3. Furthermore, in the BAT approach, the priors used to model the variances $\boldsymbol{\sigma}^2$ of additive noise $\boldsymbol{\delta}_k$ present in the measurement model outside of the recursive NKE process can be seen in Table 4.4. The initial conditions $\mathbf{x}_0$ and $\mathbf{P}(0)$ used during the execution of the task are in the Tables 3.2, and 4.1.

It is important to point out that the empirical task is executed using the UMM (Equations 2.4), with the parameters used to generate the run A-SD (see Table 3.1) as initial parameters. The idea is to estimate the parameters $\boldsymbol{\theta}$ =$\{\mathrm{Y}_{lac,glc}, \mathrm{Y}_{amm,gln}, \lambda\}$ used to generate the main solution (run D) based on the measured data of run D1, besides to estimate the other components of $\boldsymbol{\Theta}_2$, such as $\{\mathbf{x}_{0,0}$, $\mathbf{Q}$, $\mathbf{R}$. $\}$. This is a challenging task for the SANTO approach, see Section 3.4.1. 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 3.2. However, more details about how to define priors for full covariance matrices such as $\mathbf{Q}$ can be seen in Section A.4 in the Appendix.

Table 4.1: Initial state error covariance matrix $\mathbf{P}$(t=0) for BAT and baselines during the empirical task.

Parameter	**Name**	value
P_{X_v,X_v} (c^2/mL^2)	Viable cells	0.0
P_{X_t,X_t} (c^2/mL^2)	Viable cells	0.0
$P_{GLC,GLC}$ (mM^2)	Glucose	0.0
$P_{GLN,GLN}$ (mM^2)	Glutamine	0.0
$P_{LAC,LAC}$ (mM^2)	Lactate	0.0
$P_{AMM,AMM}$ (mM^2)	Ammonium	0.0
$P_{mAb,mAb}$ $(mg/L)^2$	Monoclonal Antibody (titer)	0.0

Table 4.2: 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)
$P(Y_{lac,glc})$	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)

Table 4.3: Upper And Lower Bounds (priors) of OF-BO, OF-PS, OF-GA, RPE-BO, RPE-PS, and RPE-GA used in the first step of the empirical task with the training set D1.

EKF component	(Lower Bound, Upper Bound)
R_{X_v}	(9.5e7,18e7)
Q_{X_v}	(10e6, 10e4)
Q_{X_t}	(0,1)
Q_{GLC}	(0,1)
Q_{GLN}	(0,1)
Q_{LAC}	(0,1)
Q_{AMM}	(0,1)
Q_{mAb}	(0,1)
$Y_{lac,glc}$	(1.35 , 4.0)
$Y_{amm,gln}$	(4.0e-1 , 20.5e-1)
λ	(2.e-9, 12.21e-9)
$u0_{LAC}$	(0, 10)
$u0_{mAb}$	(50, 300)

Table 4.4: Priors of $\boldsymbol{\sigma^2}$ 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)
$P(\sigma^2_{GLC})$	InverseGamma(2, 3)
$P(\sigma^2_{GLN})$	InverseGamma(2, 3)
$P(\sigma^2_{LAC})$	InverseGamma(2, 3)
$P(\sigma^2_{AMM})$	InverseGamma(2, 3)
$P(\sigma^2_{mAb})$	InverseGamma(2, 3)

4.4.2 Results

The results are organized by questions **Q1** and **Q2**.

Answering Q1. The BAT outperforms the baselines. Columns A, B, C, and D of Figures 4.3 and 4.5 present the estimations of OF-NUTS and RPE-NUTS, OF-BO and RPE-BO, OF-PS and RPE-PS, and OF-GA and RPE-GA, respectively, compared to BAT estimations. The majority of the baselines 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 properly estimate LAC, AMM, and mAb close to the ground truth (red dotted lines), in both cases of testing set D2 and D3, see Figures 4.3 and 4.5. This poor performance was confirmed by high RMSPE values for LAC, AMM, and mAb, see the heatmap of RMSPE values in Figures 4.4 and 4.6. The best estimations of the baselines were related to Xv, Xt, GLC, and GLN which are the state variables without relation with unshared parameters. Furthermore, Table 4.5 presents the results of the filter consistency test of the designed EKFs using the optimal $\boldsymbol{\Theta}$ estimated by BAT and baselines during the execution of the first step of the empirical task. Only three (RPE-BO, OF-NUTS, and OF-PS) of the eight baselines were accepted in the filter consistency tests with testing sets D2 and D3. This suggests that the EKFs designed by baselines might be inconsistent in practical applications (see Table 4.5) because the filter consistency test with testing set D2 is similar to the training set and represents offline measurements, but the testing set D3 simulates a real-world scenario. On the other hand, EKF designed by BAT was accepted in both filter consistency tests (see Table 4.5). Furthermore, the designed EKF presents the best estimations of LAC, AMM and mAB with lowest RMSPE values in both testing sets D2 and D3; see Figures 4.3, 4.4, 4.5 and 4.6.

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 all parameters ($\mathbf{R}$, $\mathbf{x}_{0,0}$, $\boldsymbol{\theta}$ and $\mathbf{Q}$) as unimodal, closely aligning with the ground truth values; see Figure 4.7 and Figure 4.8. It is essential to note that $\mathbf{R}$, $\mathbf{x}_{0,0}$=$\{u0_{LAC}$, $u0_{mAb}\}$, and $\boldsymbol{\theta}$ =$\{Y_{lac,glc}$, $Y_{amm,gln}, \lambda\}$) were modeled with uninformative priors, and $\mathbf{Q}$ with informative priors. The outcome of BAT contrasts with the baseline methods, which estimate 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. They only estimated the posterior distribution of $\mathbf{Q}$ as an unimodal distribution (see Figure 4.8), and only the estimations of OF-NUTS related to $\mathbf{R}$ and $Y_{amm,gln}$ that can be considered unimodal. Furthermore, in the majority of the cases, the highest peaks and mean obtained from the estimated distribution of $\mathbf{R}$, $\mathbf{x}_{0,0}$, and $\boldsymbol{\theta}$ (by the baselines) are far from the ground truth (see Figure 4.7). The mean and standard deviation of the estimated distributions ($\boldsymbol{\Theta}$) by

BAT, RPE-NUTS, and OF-NUTS using the training set D1 during the first step of the empirical task can be seen in Table 4.6. RPE-NUTS and OF-NUTS have the highest standard deviation values related to parameters ($\mathbf{R}$, $\mathbf{x}_{0,0}$=$\{\mathrm{u0}_{LAC}, \mathrm{u0}_{mAb}\}$, and $\boldsymbol{\theta}$ =$\{\mathrm{Y}_{lac,glc}, \mathrm{Y}_{amm,gln}, \lambda\}$) representing high uncertainty about the mean. It is important to point out that the highest peaks were chosen as the optimal $\boldsymbol{\Theta}$ estimated by BAT, OF-NUTS, and RPE-NUTS to design the EKF tested in the second step of the empirical task (see Figures 4.7 and Figure 4.8. In the case of baselines (OF-BO, OF-PS, OF-GA, RPE-BO, RPE-PS, and RPE-GA), the optimal $\boldsymbol{\Theta}$ estimated are in Table 4.7.

Answering Q2. The BAT posterior density enables efficient sampling with MCMC Figure 4.9 shows the ACF plots for the BAT, OF-NUTS, and RPE-NUTS applied during the empirical task. 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. These 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. It is important to note that the baseline methods estimated a total of 13 components of EKF during the first step of the empirical task, but BAT estimated a total of 20 parameters (13 components of EKF and 7 different variances $\boldsymbol{\sigma}^2$). The variances are related to the additive noise $\boldsymbol{\delta}_k$ present in the measurement model outside of the recursive NKE process defined by BAT posterior. See Tables 4.4 and 4.8.

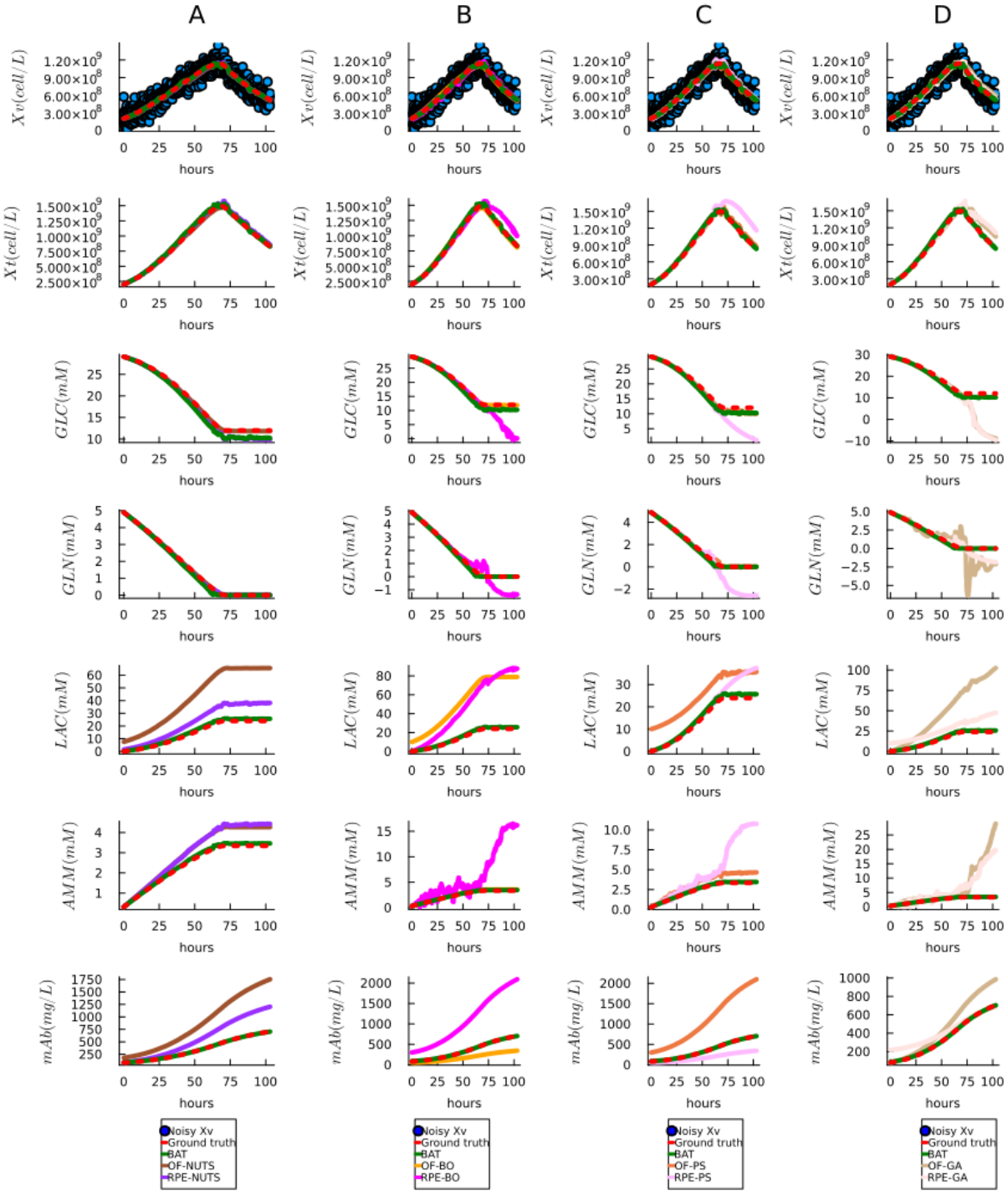

Figure 4.3: State variables estimations from EKF designed by BAT, baselines during task using the noisy Xv data of testing set D3. Columns A, B, C, and D present the estimations of OF-NUTS and RPE-NUTS, OF-BO and RPE-BO, OF-PS and RPE-PS, and OF-GA and RPE-GA, respectively, compared to BAT estimations. Baselines had similar performances, with the worst estimation of LAC, AMM, and mAb. Figure 4.4 confirms these results by showing the heatmap of RMSPE values between designed EKF (during the empirical task) and ground truth values of the testing set D3.

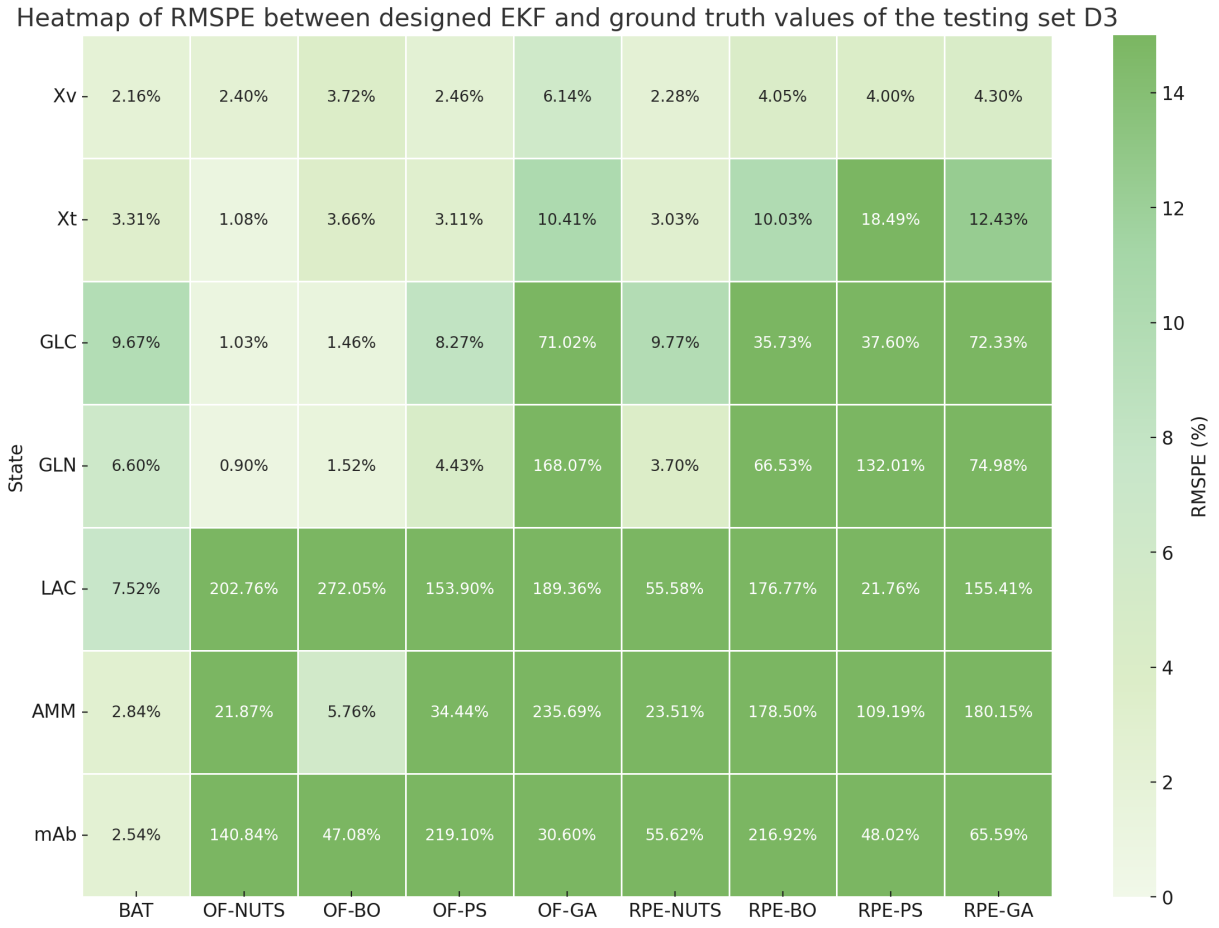

Figure 4.4: Heatmap of RMSPE between designed EKF (during the empirical task) and ground truth values of the testing set D3 (sample rate of 7.5 min).

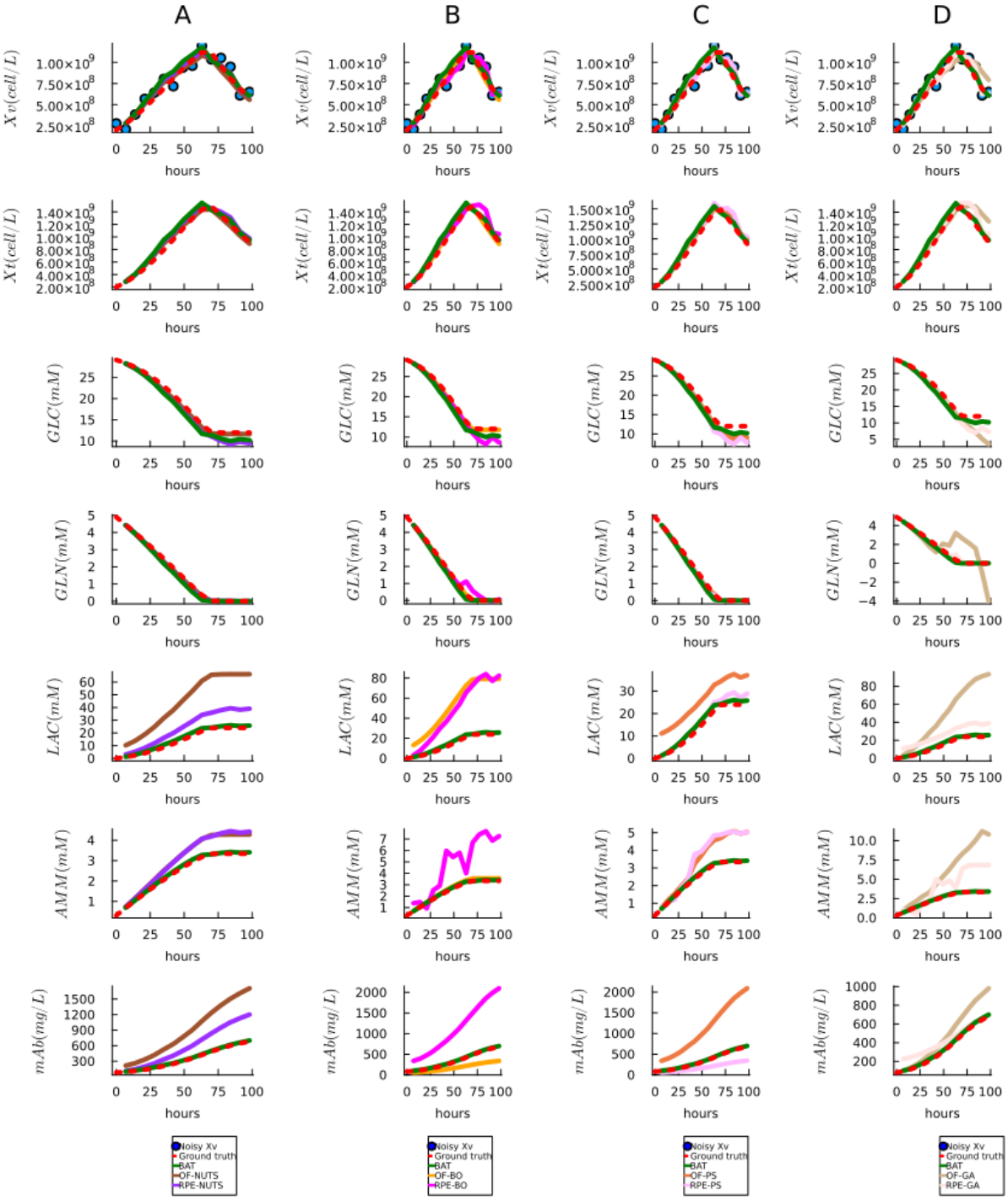

Figure 4.5: State variables estimations from EKF designed by BAT, baselines during task using the noisy Xv data of testing set D2. Columns A, B, C, and D present the estimations of OF-NUTS and RPE-NUTS, OF-BO and RPE-BO, OF-PS and RPE-PS, and OF-GA and RPE-GA, respectively, compared to BAT estimations. Baselines had similar performances, with the worst estimation of LAC, AMM, and mAb. Figure 4.6 confirms these results by showing the heatmap of RMSPE values between designed EKF (during the empirical task) and ground truth values of the testing set D2.

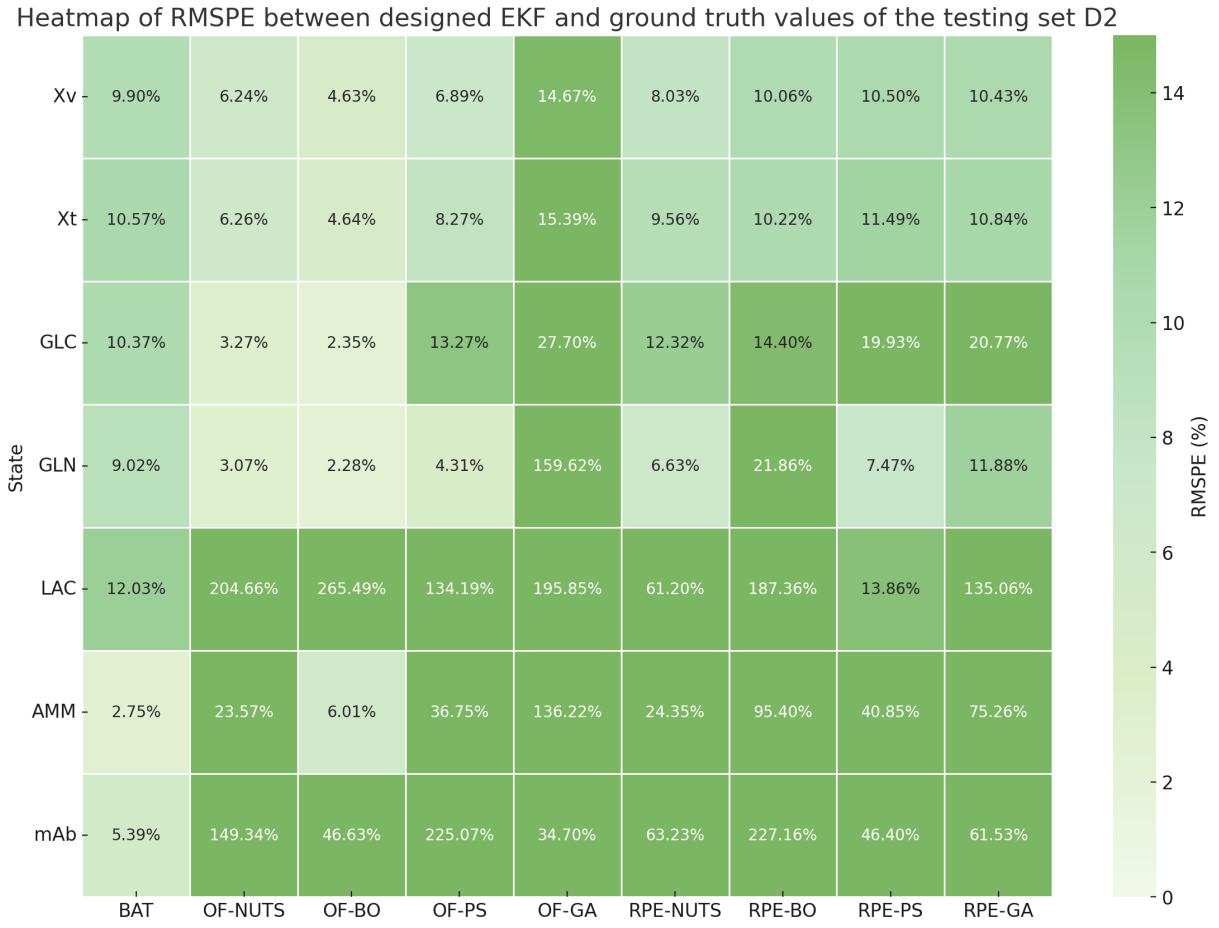

Figure 4.6: Heatmap of RMSPE between designed EKF (during empirical task) and ground truth values of the testing set D2 (sample rate of 7h).

Table 4.5: Results of filter consistency test of the designed EKF with testing sets D2 and D3 during the execution of the empirical task. The N $\times\mu(NIS)$ obtained by the designed EKF with BAT and baselines estimations were assessed in the filter consistency tests FCT2 ($P(N \times \mu(NIS) \in [5.629, 26.1]|H_0) = 1-\alpha?$) and FCT3 ($P(N \times \mu(NIS) \in [745.3, 904.39]$ — $H_0) = 1 - \alpha?$) defined in Section 4.4.1. 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	8.99	713.68
RPE-BO	5.99	870.03
RPE-PS	4.01	894.11
RPE-GA	4.99	874.79
OF-NUTS	11.68	776.67
OF-BO	3.32	265.31
OF-PS	8.27	870.33
OF-GA	22.93	1066.9
BAT	13.71	786.68
Ground Truth	7.173	774.8

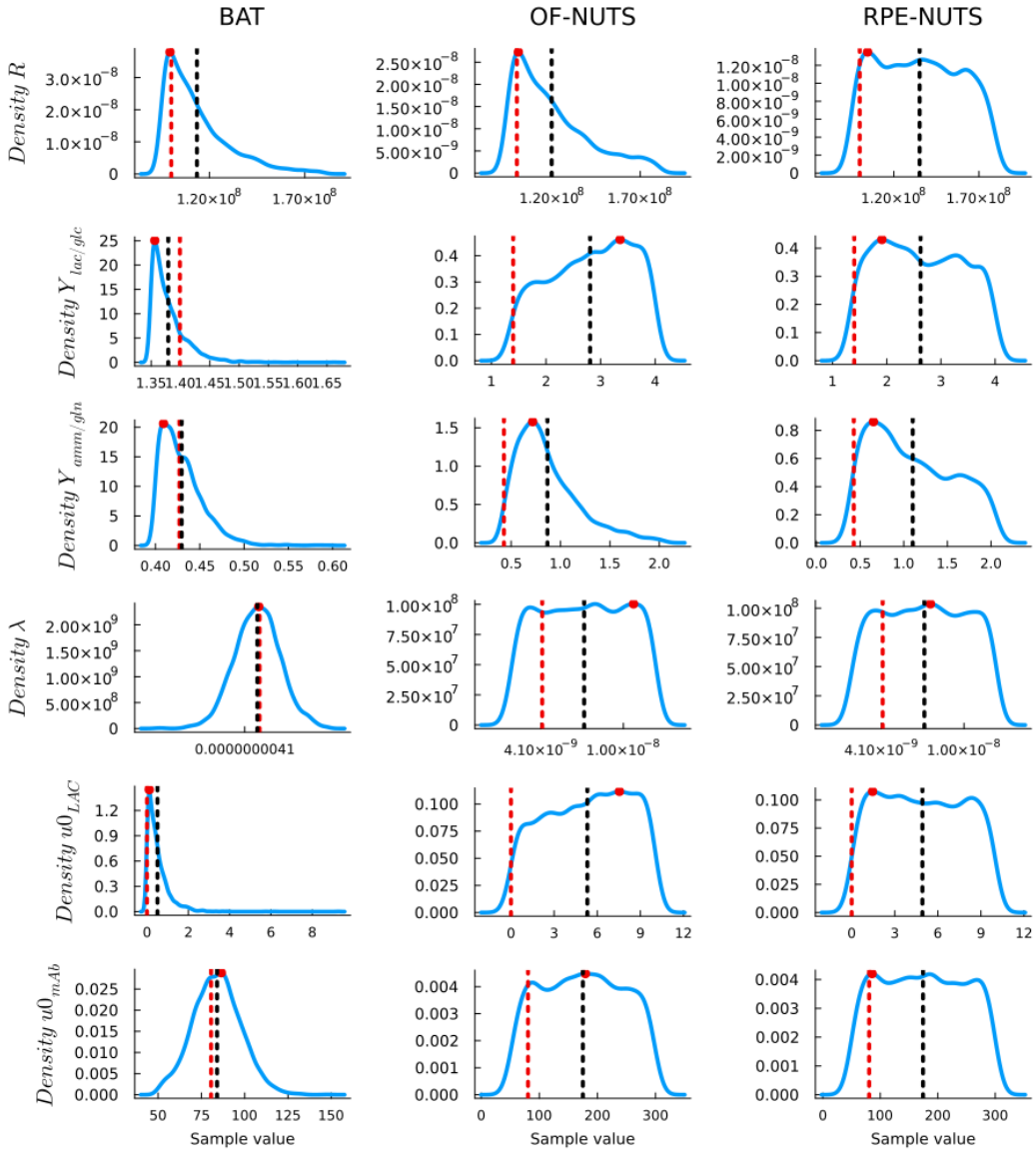

Figure 4.7: Distribution of the sampled values (for $\mathbf{R}$, $u0_{LAC}$, $u0_{mAb}$, $Y_{lac,glc}$, $Y_{amm,gln}$, and λ) by BAT, OF-NUTS, and RPE-NUTS with training set D1. OF-NUTS and RPE-NUTS produced multimodal distributions, whereas BAT generated unimodal distributions with peaks near the actual values (red vertical dotted line). The black vertical dotted lines indicate the mean of the sampled values for $\mathbf{R}$, $u0_{LAC}$, $u0_{mAb}$, $Y_{lac,glc}$, $Y_{amm,gln}$, and λ , and the red dot represents the highest peak. The highest peaks were chosen as the optimal $\mathbf{\Theta}$ estimated by BAT, OF-NUTS, and RPE-NUTS to design the EKF.

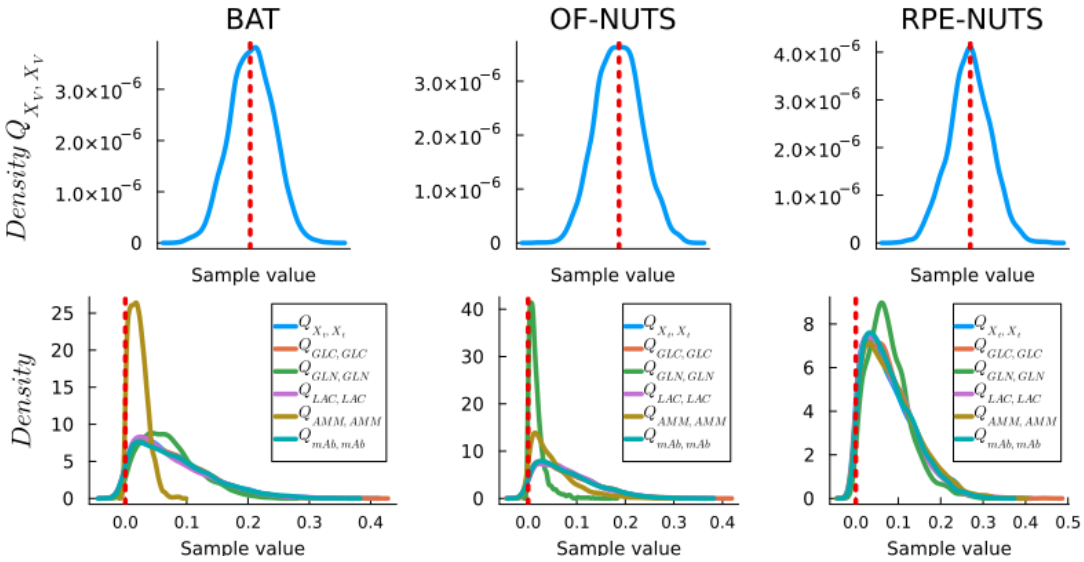

Figure 4.8: Distribution of the sampled values (for **Q** related to the seven state variables) by BAT, OF-NUTS and RPE-NUTS with training set D1. BAT, OF-NUTS, and RPE-NUTS estimated unimodal distributions with peaks close to the ground truth values (red vertical line).

Table 4.6: Mean and standard deviation of the distribution estimated by BAT, RPE-NUTS and OF-NUTS of Θ using training set D1 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	1.1.9e8	2.11e7	1.35e8	2.40e7
Q_{X_v}	10e6	1.06e7	948902.5	9.99e7	9.9e4	9.998e6	102054.7
Q_{X_t}	0.1	0.0773	0.0598	0.0783	0.0588	0.0801	0.0615
Q_{GLC}	0.1	0.0764	0.0595	0.0794	0.0625	0.0795	0.0590
Q_{GLN}	0.1	0.0707	0.0441	0.0167	0.0170	0.0780	0.0499
Q_{LAC}	0.1	0.0767	0.0594	0.0802	0.0593	0.0800	0.0591
Q_{AMM}	0.1	0.0214	0.0149	0.0500	0.0445	0.0852	0.0624
Q_{mAb}	0.1	0.0798	0.06	0.0789	0.0604	0.0803	0.0582
$Y_{lac,glc}$	1.399	1.3798	0.0286	2.8081	0.7462	2.6238	0.7664
$Y_{amm,gln}$	0.427	0.43	0.0232	0.8674	0.3199	1.1024	0.4724
λ	4.21e-9	4.18e-9	1.68e-10	7.2e-9	2.91e-9	7.1e-9	2.971e-9
$u0_{LAC}$	0	0.4894	0.4664	5.3134	2.8699	4.9260	2.8925
$u0_{mAb}$	80.6	85.2374	13.6992	175.0046	69.2860	174.4094	72.0334

Table 4.7: Θ estimated by BAT, RPE-NUTS and OF-NUTS using training set D1 during the task.

Θ	Ground Truth	BAT	OF-NUTS	OF-BO	OF-PS	OF-GA
R_{X_v}	10E7	9.933E7	1.008E8	1.8E8	9.5E7	9.50E7
Q_{X_v}	10E6	1.002E7	1.001E7	1.0E6	1.0E6	1.11E6
Q_{X_t}	0.1	0.038	0.024	8.075E-12	0.01	0.4310
Q_{GLC}	0.1	0.028	0.0248	0.01	0.09	0.326
Q_{GLN}	0.1	0.043	0.006	4.343E-11	0.001	0.7288
Q_{LAC}	0.1	0.024	0.023	1.1406E-11	0.001	0.144
Q_{AMM}	0.1	0.017	0.014	3.088E-11	0.001	0.151
Q_{mAb}	0.1	0.020	0.025	6.852E-12	0.07	0.723
$Y_{lac,glc}$	1.399	1.356	3.358	4.0	1.35	3.999
$Y_{amm,gln}$	0.427	0.409	0.717	0.5	0.744	1.920
λ	4.21E-9	4.203E-9	1.0725E-8	2.0E-9	1.221E-8	6.15E-9
$U0_{LAC}$	0	0.111	7.54	10.0	10.0	0.016
$U0_{mAb}$	80.6	86.51	179.32	50.0	300.0	78.47
Θ	Ground Truth	BAT	RPE-NUTS	RPE-BO	RPE-PS	RPE-GA
R_{X_v}	10E7	9.933E7	1.046E8	9.5E7	9.5E7	9.500E7
Q_{X_v}	10E6	1.002E7	9.998E6	2.0E7	2.0E7	1.999E7
Q_{X_t}	0.1	0.038	0.041	0.999	0.001	0.948
Q_{GLC}	0.1	0.028	0.023	0.627	0.002	0.997
Q_{GLN}	0.1	0.043	0.060	0.008	0.365	0.2659
Q_{LAC}	0.1	0.024	0.025	0.0086	0.005	0.976
Q_{AMM}	0.1	0.017	0.029	0.999	0.088	0.8669
Q_{mAb}	0.1	0.020	0.033	0.0050	0.074	0.387
$Y_{lac,glc}$	1.399	1.356	1.910	3.999	1.35	1.350
$Y_{amm,gln}$	0.427	0.409	0.652	2.05	0.4	0.4000
λ	4.21E-9	4.203E-9	7.612E-9	1.22096E-8	2.0E-9	3.191E-9
$U0_{LAC}$	0	0.111	1.455	0.00079	0.0	9.99
$U0_{mAb}$	80.6	86.51	85.698	300.0	50.0	217.024

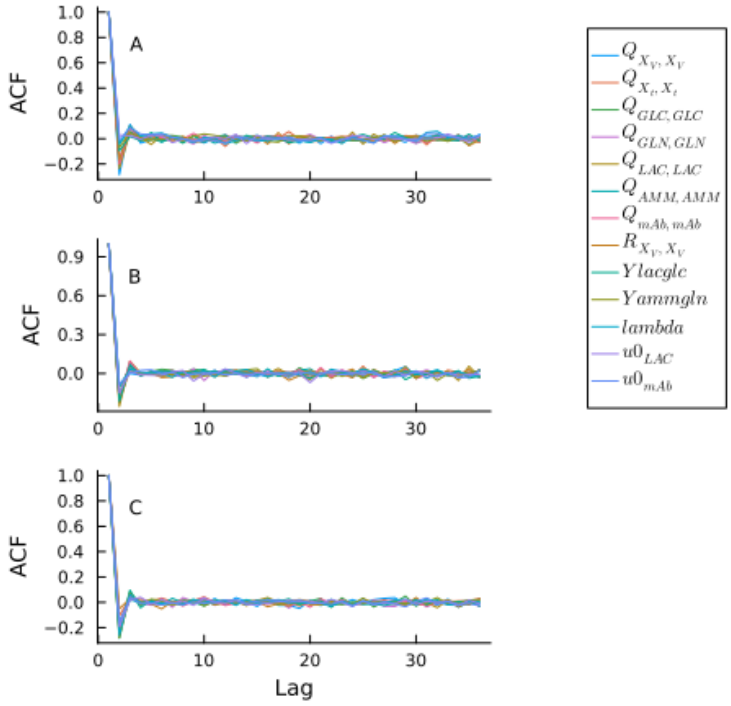

Figure 4.9: ACF plots of BAT (A), OF-NUTS (B) and RPE-NUTS (C) samples computed up to a lag of 30. This samples were obtained with training set D1 during the task .

Table 4.8: 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

4.5 Discussion

Based on our empirical results, some points should be discussed.

Approximating the BAT posterior. The ACF plots (see Figure 4.9) show that BAT posterior can be efficiently sampled with NUTS since the obtained results show: 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 important to highlight that MCMC was employed to approximate the BAT posterior in this thesis. However, variational inference (VI) can also be utilized, particularly in situations involving large datasets or high-dimensional parameter spaces where MCMC faces challenges with convergence and demands an excessive number of samples. VI is more efficient in handling high-dimensional spaces and scales better with data size, making it appropriate for large-scale problems.

Limitation. The main limitation of BAT is the need for good prior selection. 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 posterior density. Furthermore, another BAT limitation is that it needs observational data related to all state variables beyond those used to generate the innovation errors. Without that, the BAT may have similar performance to the baseline approaches.

Unimodal distribution estimation. The empirical results show that BAT is more able than the baselines to estimate the unimodal distribution of unshared parameters with peaks close to the ground truth values. 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. By performing batch Bayesian inference outside of the recursive NKE process, BAT can integrate measurement data and prior knowledge more effectively. For example, BAT posterior enabled the estimation of the variance $\boldsymbol{\sigma}^2$ of additive noise $\boldsymbol{\delta}_k \sim N(0, \boldsymbol{\sigma}^2)$ present in the measurement model defined outside of the recursive NKE process.

Small dataset. It is essential 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 an NKE with a small dataset. Consequently, this helped to demonstrate that BAT approach can address the limitation of the biopharma industry to generate a large amount of data to develop monitoring tools [110], as BAT performed well across all testing sets.

BAT and SANTO. SANTO approach improves the performance of JNKE with spe-

cific UMM within a set of similar biomanufacturing conditions, see Chapter 3. However, in one of the empirical tests, the Kalman gain exhibited unusual behavior, starting from a negative value and converging to near zero. Although this does not affect the performance, this anomaly can avoided with BAT by tuning all NKE components, including these unshared parameters, as described in the results. BAT directly estimates these unshared parameters, resolving the unconventional Kalman gain behavior observed with SANTO. In addition, BAT also can be applied to tune λ value of SANTO approach.

4.6 Summary

This thesis introduces Batch Bayesian Auto-Tuning, a novel approach for auto-tuning all NKE components, including the $\mathbf{Q}$, $\mathbf{R}$, $\mathbf{P}(0)$, $\mathbf{x}_{0,0}$, and $\boldsymbol{\theta}$. BAT enables utilizing all available measured data, not just those selected for generating innovation errors in NKEs. Our empirical results demonstrate that BAT not only exhibits efficient sampling with MCMC but also outperforms the baselines in tuning NKEs with better filter consistency, lower RMSPE values, and more precise unimodal parameter estimations. EKFs designed using BAT were accepted in filter consistency tests for both offline (testing set D2) and real-world (testing set D3) scenarios. Additionally, BAT provided more accurate estimations of key variables (LAC, AMM, mAb) with the lowest RMSPE values compared to the baseline methods. BAT estimated unimodal posterior distributions closely aligning with the ground truth, while the baseline methods resulted in multimodal distributions. It is important to point out that the effectiveness of BAT in estimating unimodal posterior distributions of all NKE components can be attributed to the well-defined BAT posterior distribution outside of the NKE recursive loop. This enables integrating all available measured data and prior knowledge more effectively. These aspects collectively contribute to its ability to produce more precise and less uncertain estimates compared to baselines.

Chapter 5

Hybrid Nonlinear Kalman Estimators

5.1 Overview

This Chapter presents the proposed approach to address the RQ3. This thesis introduces the hybrid nonlinear Kalman estimator, a novel hybrid Gaussian filter designed to address the challenges of fast and low-cost bioprocess monitoring. Traditional methods often require extensive tuning and can fail under biomanufacturing conditions characterized by limited data and the use of generic UMMs. HNKE auto-initializes the state error covariance matrix $\mathbf{P}(0)$ and iteratively estimates UMM parameters in real time by integrating a hybrid dynamic model with uncertainty quantification using unscented transformation or cubature rule in a hybrid Gaussian filter framework. Empirical evaluation with synthetic bioprocess data, representing monoclonal antibody production, demonstrates that HNKE requires minimal data for training and only the definition of measurement $\mathbf{R}$ and process $\mathbf{Q}$ noise covariance matrices, outperforming baseline models in monitoring highly nonlinear bioprocesses. This approach offers a practical, robust, adaptable, and cost-effective solution for bioprocess monitoring, aligning with the biopharmaceutical industry's goals of enhancing efficiency and reducing costs. HNKE enables real-time estimation of state variables under various conditions by auto-initializing $\mathbf{P}(0)$, reducing setup time and effort, and facilitating parameter estimation in a wide range of conditions without requiring large data volumes. Through empirical evaluations, HNKE demonstrated superior performance and reliability compared to baseline methods.

The main contributions of this chapter are:

- **Novel hybrid Gaussian filter:** HNKE is the first hybrid Gaussian filter that com-

bines HMuq with an unscented transformation or cubature rule for fast and low-cost monitoring of highly nonlinear bioprocess within a set of different biomanufacturing conditions.

- **Innovative auto-initialization of $\mathbf{P}(0)$:** This is the first study to introduce this feature, which, along with the iterative estimation of generic UMM parameters and the ability to use initial noise state variable values, simplifies the setup process, requiring only the definition of measurement noise $\mathbf{R}$ and process noise $\mathbf{Q}$.
- **Solution for generic UMM**: HNKE also addresses the limitation of the BAT and SANTO approach with generic UMM modeling a highly nonlinear bioprocess.
- **Reduced large data requirements:** HNKE can be trained using smaller amounts of data, making it more efficient and bypassing data availability issues in biopharmaceutical monitoring tool development.
- **Empirical performance superiority:** Through empirical testing with real-world scenarios, the HNKE outperformed the baselines, confirming its effectiveness and offering a more reliable and accurate tool for fast and low-cost bioprocess monitoring showing that HNKE is a real-time approach with high adaptability. The code and data used in this work are available on GitHub repository [1] to facilitate reproducibility.

5.2 Related work

The literature has reported several hybrid models and techniques to tune some NKE components, such as $\mathbf{P}$, $\mathbf{Q}$, $\mathbf{R}$ and UMM parameters; however, to the best of our knowledge, no hybrid Gaussian filter integrating HMuq with unscented transformation or cubature rule, and an approach to auto-initialize $\mathbf{P}(0)$ has been reported yet.

Hybrid models (HM). A Hybrid model (HM) is a combination of two or more models blended together to enhance the accuracy. It aims to leverage the strengths of different methods to achieve better performance, flexibility, and interpretability. A common type of HM is the Hybrid dynamic models with the integration of data-driven and knowledge-based approaches, thereby enabling greater accuracy and flexibility [109, 125, 139, 145]. By integrating process knowledge and data, these models can bridge the gap in the understanding of complex systems with limited data. This approach is particularly valuable

[1] https://github.com/cristovaoiglesias/HNKE

in fields such as biopharma, where experiments are costly and data are limited. Hybrid models require less data than pure Data-Driven Models (DDM), offer better accuracy, and allow extrapolation and interpretation [108, 110]. However, the use of HM with NKE for bioprocess monitoring remains in its infancy. For instance, there are no documented cases of hybrid models integrated with a cubature Kalman filter (CKF) or an unscented Kalman filter (UKF), with only one incorporating the EKF. In [107], researchers developed a hybrid model coupled with an extended Kalman filter for real-time monitoring and control of mammalian cell cultures. However, that study did not follow the manufacturing conditions presented in this paper for fast and low-cost bioprocess monitoring. For example, [107] relied on the measurements of all variables except titter to estimate all state variables. Our study uses only one measured state variable (Xv).

Tuning techniques. They can be grouped as online (real-time) and offline. They are poorly adapted to changing biomanufacturing conditions and need to be adjusted for each new condition. JNKE and DNKE are online approaches. JNKE aims to auto-tune the parameters $\boldsymbol{\theta}$ of a UMM in NKEs by expanding the state variable vector to include both system states $\mathbf{x}_k$ and $\boldsymbol{\theta}$ [9, 55, 166]. The evolution of $\boldsymbol{\theta}$ is based on the following equation: $\boldsymbol{\theta}_k = \boldsymbol{\theta}_{k-1} + \delta_{k-1}$, where δ_{k-1} is a perturbation (noise) added at each time step. However, the increased dimensionality and complexity of state augmentation can make the accurate estimation of states and parameters challenging. The problem with JNKE is the singularity of the UMM for the parameter. It works well when the entire system is not highly nonlinear [137]. Furthermore, as described in chapter 3, joint and dual NKE "fail" to estimate the state and unshared parameters $\boldsymbol{\theta}_{up}$ simultaneously under biomanufacturing conditions for fast and low-cost bioprocess monitoring, see Section 3.2. Most of offline approaches optimize an objective function based on metrics such as Normalized Estimation Error Squared (NEES) or Normalized Innovation Squared (NIS) [11, 19, 22, 88, 135]. Various optimization methods, such as Genetic Algorithms [118], Particle Swarm Optimization [154], and particularly Bayesian Optimization (BO) [22–24], have been proposed to optimize objective functions as described in Chapter 4.

5.3 Hybrid Nonlinear Kalman Estimators

HNKE uses Gaussian filtering, where the prediction step approximates $\mathbf{x}_k|\mathbf{y}_{1:k-1} \sim \mathcal{N}(\boldsymbol{m}_k^-, \boldsymbol{P}_k^-)$ using a hybrid dynamic model with uncertainty quantification (HMuq), and update step approximates $\mathbf{x}_k|\mathbf{y}_{1:k} \sim \mathcal{N}(\boldsymbol{m}_k, \boldsymbol{P}_k)$ using unscented transformation or cubature role. Therefore, in this section, we show that i) HMuq estimates the mean and covariance matrix of $\mathbf{g}(\mathbf{x}_{k-1})$; ii) HMuq can perform Gaussian filtering prediction step by moment matching;

and iii) HNKE can auto-initialize $\mathbf{P}(0)$.

5.3.1 Hybrid dynamic model with uncertainty quantification (HMuq)

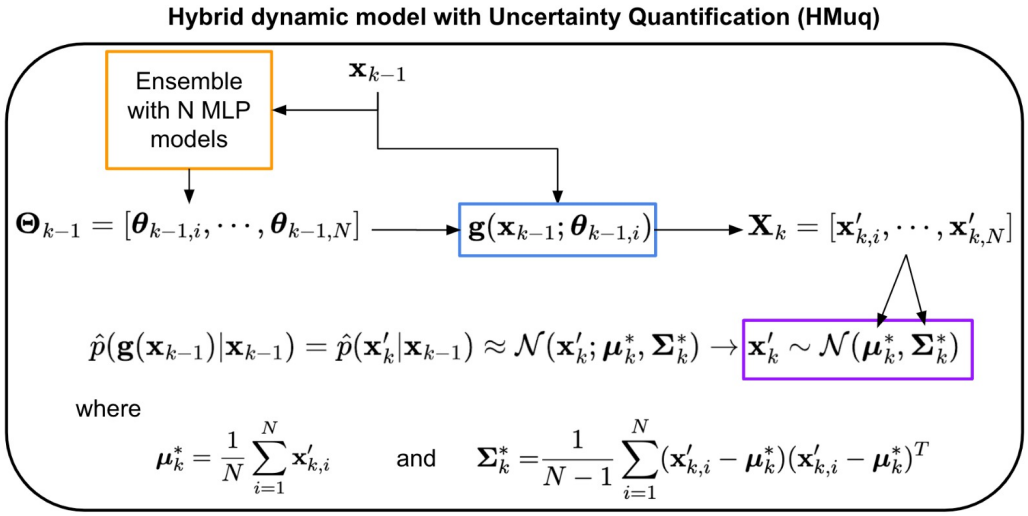

Figure 5.1: Hybrid dynamic model with uncertainty quantification (HMuq). It integrates an ensemble of N MLP models, each represented by $MLP(x_{k-1}, w_i)$, where w_i varies for each model in the ensemble, reflecting different parameter estimations. These models collectively predict a set of N possible parameters Θ_{k-1}, which are then used to solve the dynamic model $\mathbf{g}(\mathbf{x}_{k-1}; \boldsymbol{\theta}_{k-1,i})$, for example, via an ODE solver from t_{k-1} to t_k. Each parameter set $\theta_{k-1,i}$ provides a unique trajectory, $\mathbf{x}'_{k,i}$, which enables the computations of the moments of $\mathbf{x}'_k$.

An HMuq is a model composed of an ensemble of N multilayer perceptron (MLP) models, $[MLP(\mathbf{x}_{k-1}, \boldsymbol{w}_i), \cdots, MLP(\mathbf{x}_{k-1}, \boldsymbol{w}_N)]$, and a mechanistic model as $\mathbf{g}(\cdot)$, as shown in Figure 5.1. The ensemble predicts a set of N possible parameters $\mathbf{\Theta}_{k-1} = [\boldsymbol{\theta}_{k-1,i}, \cdots, \boldsymbol{\theta}_{k-1,N}]$ based on the state variables $\mathbf{x}_{k-1}$, for k=1, 2, $\cdots$ K. The $\mathbf{\Theta}_{k-1}$ enables to obtain a set of N solution $\mathbf{X}_k = [\mathbf{x}'_{k,i}, \cdots, \mathbf{x}'_{k,N}]$ by solving the mechanistic model $\mathbf{g}(\mathbf{x}_{k-1}; \boldsymbol{\theta}_{k-1,i})$ with a ODEsolver from t_{k-1} to t_k using each parameter of $\boldsymbol{\theta}_{k-1,i}$ and $\mathbf{x}_{k-1}$ as initial state variables, see Figure 5.1. Here, bagging is the ensemble strategy considered; however, other strategies can also be used.

It is important to point out that ensembling naturally can be viewed as an approximate Bayesian inference method that enables us to estimate both aleatoric and epistemic uncertainty of the prediction [43,85,94]. The Bayesian viewpoint of HMuq is then adopted from the perspective of a predictive posterior distribution. In the case of the ensemble of N MLP models, the essence of the Bayesian inference framework is to treat the MLP model parameters $\boldsymbol{w}$ as random variables and place a prior distribution $p(\boldsymbol{w})$ over them to compute a posterior distribution $p(\boldsymbol{w}|\mathcal{D})$ via Bayes' rule, $p(\boldsymbol{w}|\mathcal{D}) \propto p(\mathcal{D}|\boldsymbol{w})p(\boldsymbol{w})$, where $\mathcal{D}$ consists of M i.i.d data points $\mathcal{D} = \{\mathbf{x}_{k-1}, \boldsymbol{\theta}_{k-1}\}_{i=1}^{M}$. Therefore, the uncertainty in the model parameters $p(\boldsymbol{w}|\mathcal{D})$ is then marginalized out to obtain the predictive posterior distribution of HMuq as,

$$\begin{aligned} p(\mathbf{x}'_k|\mathbf{x}_{k-1}, \mathcal{D}) &= \int p(\mathbf{x}'_k|\mathbf{x}_{k-1}, \boldsymbol{\theta}_{k-1}) p(\boldsymbol{\theta}_{k-1}|\mathcal{D}) d\boldsymbol{\theta}_{k-1} \approx \frac{1}{N} \sum_{i=1}^{N} p(\mathbf{x}'_k|\mathbf{x}_{k-1}, \boldsymbol{\theta}_{k-1,i}), \\ \boldsymbol{\theta}_{k-1,i} &= MLP(\mathbf{x}_{k-1}, \boldsymbol{w}_i), \boldsymbol{w}_i \sim p(\boldsymbol{w}|\mathcal{D}). \end{aligned} \tag{5.1}$$

The generally intractable integral in Equation 5.1 is approximated using N Monte Carlo samples $\boldsymbol{w}_i$, ideally drawn from the $p(\boldsymbol{w}|\mathcal{D})$. In practice however, obtaining samples from the true posterior $p(\boldsymbol{w}|\mathcal{D})$ is virtually impossible, requiring to use an implicit sampling distribution $q(\boldsymbol{w})$ to approximate the posterior $p(\boldsymbol{w}|\mathcal{D})$, $q(\boldsymbol{w}) \approx p(\boldsymbol{w}|\mathcal{D})$ [43,85,94]. Thus, we obtain the approximate predictive posterior distribution as,

$$\begin{aligned} \hat{p}(\mathbf{x}'_k|\mathbf{x}_{k-1}) &= \frac{1}{N} \sum_{i=1}^{N} p(\mathbf{x}'_k|\mathbf{x}_{k-1}, \boldsymbol{\theta}_{k-1,i}), \\ \boldsymbol{\theta}_{k-1,i} &= MLP(\mathbf{x}_{k-1}, \boldsymbol{w}_i), \boldsymbol{w}_i \sim q(\boldsymbol{w}). \end{aligned} \tag{5.2}$$

The quality of the approximate predictive posterior depends on the number of samples N and the method employed to generate $q(\boldsymbol{w})$ [43,94]. It is essential to point out that, in this thesis, bagging is the ensemble method for generating $q(\boldsymbol{w})$, and $\boldsymbol{\theta}_{k-1,i}$ is the output of each ensemble sub-model $MLP(\mathbf{x}_{k-1}, \boldsymbol{w}_i)$ given $\mathbf{x}_{k-1}$ and the sampled $\boldsymbol{w}_i$. Then, $\boldsymbol{\theta}_{k-1,i}$ can be considered a sample from this *implicit distribution* $q(\boldsymbol{w})$, approximating the predictive posterior distribution $p(\mathbf{x}'_k|\mathbf{x}_{k-1}, \mathcal{D})$ by averaging these samples [94].

Assumption 5.1. *The number of ensemble members N is large enough for the Monte Carlo integration to hold (motivated by the Strong Law of Large Numbers).*

Assumption 5.2. *$q(\boldsymbol{w})$ sufficiently approximates $p(\boldsymbol{w}|\mathcal{D})$ [43, 85, 94]. Bagging can be used as an ensemble strategy, but other methods might be required depending on the quality of the approximation.*

Lemma 5.1. *If $N \to \infty$ and $q(\boldsymbol{w}) \approx p(\boldsymbol{w}|\mathcal{D})$, the predictive posterior distribution of HMuq can be approximated as a single Gaussian distribution (multivariate) $\hat{p}(\mathbf{g}(\mathbf{x}_{k-1})|\mathbf{x}_{k-1}) = \hat{p}(\mathbf{x}'_k|\mathbf{x}_{k-1}) \approx \mathcal{N}(\mathbf{x}'_k; \boldsymbol{\mu}^*_k, \boldsymbol{\Sigma}^*_k) \to \mathbf{x}'_k \sim \mathcal{N}(\boldsymbol{\mu}^*_k, \boldsymbol{\Sigma}^*_k)$ where $\boldsymbol{\mu}^*_k = \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}'_{k,i}$ and $\boldsymbol{\Sigma}^*_k = \frac{1}{N-1}\sum_{i=1}^{N}(\mathbf{x}'_{k,i} - \boldsymbol{\mu}^*_k)(\mathbf{x}'_{k,i} - \boldsymbol{\mu}^*_k)^T$ are mean and covariance matrix.*

Proof. Given that the Monte Carlo integration method is motivated by the Strong Law of Large Numbers, we have that expected predictive posterior distribution of HMuq is

$$\begin{aligned}
\mathbb{E}[\hat{p}(\mathbf{x}'_k|\mathbf{x}_{k-1})] &= \lim_{N\to\infty} \frac{1}{N}\sum_{i=1}^{N} p(\mathbf{x}'_k|\mathbf{x}_{k-1}, \boldsymbol{\theta}_{k-1,i}), \quad \boldsymbol{\theta}_{k-1,i} = MLP(\mathbf{x}_{k-1}, \boldsymbol{w}_i), \boldsymbol{w}_i \sim q(\boldsymbol{w}) \\
\mathbb{E}[\mathbf{g}(\mathbf{x}_{k-1})|\mathbf{x}_{k-1}] &= \lim_{N\to\infty} \frac{1}{N}\sum_{i=1}^{N} p(\mathbf{x}'_k|\mathbf{x}_{k-1}, \boldsymbol{\theta}_{k-1,i}), \quad \boldsymbol{\theta}_{k-1,i} = MLP(\mathbf{x}_{k-1}, \boldsymbol{w}_i), \boldsymbol{w}_i \sim q(\boldsymbol{w}) \\
&= \lim_{N\to\infty} \frac{1}{N}\sum_{i=1}^{N} \mathbf{g}(\mathbf{x}_{k-1}; \boldsymbol{\theta}_{k-1,i}), \quad \boldsymbol{\theta}_{k-1,i} = MLP(\mathbf{x}_{k-1}, \boldsymbol{w}_i), \boldsymbol{w}_i \sim q(\boldsymbol{w}).
\end{aligned} \tag{5.3}$$

Then, mean $\mathbb{E}[\mathbf{g}(\mathbf{x}_{k-1})|\mathbf{x}_{k-1}] = \boldsymbol{\mu}^*_k$ and covariance matrix $Cov(\mathbf{g}(\mathbf{x}_{k-1})|\mathbf{x}_{k-1}) = \boldsymbol{\Sigma}^*_k$ are approximated using N Monte Carlo samples $\boldsymbol{w}_i$ drawn from the $q(\boldsymbol{w})$, as follows:

$$\begin{aligned}
\mathbb{E}[\hat{p}(\mathbf{x}'_k|\mathbf{x}_{k-1})] = \mathbb{E}[\mathbf{g}(\mathbf{x}_{k-1})|\mathbf{x}_{k-1}] &\approx \frac{1}{N}\sum_{i=1}^{N} \mathbf{g}(\mathbf{x}_{k-1}; \boldsymbol{\theta}_{k-1,i}), \quad \boldsymbol{\theta}_{k-1,i} = MLP(\mathbf{x}_{k-1}, \boldsymbol{w}_i), \boldsymbol{w}_i \sim q(\boldsymbol{w}), \\
\boldsymbol{\mu}^*_k = \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}'_{k,i} &= \frac{1}{N}\sum_{i=1}^{N} \mathbf{g}(\mathbf{x}_{k-1}; \boldsymbol{\theta}_{k-1,i}), \quad \boldsymbol{\theta}_{k-1,i} = MLP(\mathbf{x}_{k-1}, \boldsymbol{w}_i), \boldsymbol{w}_i \sim q(\boldsymbol{w}) \\
&\text{and} \\
\boldsymbol{\Sigma}^*_k &= \frac{1}{N-1}\sum_{i=1}^{N}(\mathbf{x}'_{k,i} - \boldsymbol{\mu}^*_k)(\mathbf{x}'_{k,i} - \boldsymbol{\mu}^*_k)^T.
\end{aligned} \tag{5.4}$$

Therefore, under the conditions of a large ensemble size N and a well-approximated implicit sampling distribution $q(\boldsymbol{w})$, the predictive posterior distribution of HMuq can

be effectively approximated as a single multivariate Gaussian distribution $\hat{p}(\mathbf{x}'_k|\mathbf{x}_{k-1}) \approx \mathcal{N}(\mathbf{x}'_k; \boldsymbol{\mu}^*_k, \boldsymbol{\Sigma}^*_k)$ by moment matching using the mean $\mathbb{E}[\mathbf{g}(\mathbf{x}_{k-1})|\mathbf{x}_{k-1}] = \boldsymbol{\mu}^*_k$ and covariance matrix $Cov(\mathbf{g}(\mathbf{x}_{k-1})|\mathbf{x}_{k-1}) = Cov(\mathbf{X}_k) = \boldsymbol{\Sigma}^*_k$. □

5.3.2 HMuq as Gaussian filter prediction step

As described before, the unscented transformation or cubature rule is used in the Gaussian prediction step to estimate the mean and covariance of $\mathbf{x}_k = \mathbf{g}(\mathbf{x}_{k-1}) + \boldsymbol{q}_{k-1}$. However, based on the Lemma 5.1, we also can use HMuq to estimate the moments of $\mathbf{x}_k$.

Lemma 5.2. *Assume that HMuq approximates $\mathbf{g}(\mathbf{x}_{k-1}) \sim \mathcal{N}(\boldsymbol{\mu}^*_k, \boldsymbol{\Sigma}^*_k)$. If the joint distribution of $\mathbf{x}_{k-1} \sim \mathcal{N}(\boldsymbol{m}_{k-1}, \boldsymbol{P}_{k-1})$ and $\mathbf{x}_k = \mathbf{g}(\mathbf{x}_{k-1}) + \boldsymbol{q}_{k-1}$ can be approximated by*

$$p(\mathbf{x}_{k-1}, \mathbf{x}_k|\mathbf{y}_{1:k-1}) \approx \mathcal{N}\left(\begin{pmatrix}\mathbf{x}_{k-1}\\ \mathbf{x}_k\end{pmatrix} | \begin{pmatrix}\boldsymbol{m}_{k-1}\\ \boldsymbol{\mu}^*_k\end{pmatrix}, \begin{pmatrix}\boldsymbol{P}_{k-1} & \boldsymbol{C}\\ \boldsymbol{C}^T & \boldsymbol{\Sigma}^*_k + \boldsymbol{Q}\end{pmatrix}\right) \tag{5.5}$$

where $\boldsymbol{C}$ and $\boldsymbol{C}^T$ represent the cross-covariance between $\mathbf{x}_{k-1}$ and $\mathbf{x}_k$, the mean $\boldsymbol{m}^-_k = \boldsymbol{\mu}^*_k$ and covariance $\boldsymbol{P}^-_k = \boldsymbol{\Sigma}^*_k + \boldsymbol{Q}$ of the marginal distribution $\mathbf{x}_k \sim \mathcal{N}(\boldsymbol{m}^-_k, \boldsymbol{P}^-_k)$ are the prediction equations for the prediction step of Gaussian filtering.

Proof. Let's consider

$$\begin{aligned}
&\mathbf{x}_{k-1} \sim \mathcal{N}(\boldsymbol{m}_{k-1}, \boldsymbol{P}_{k-1}),\\
&\mathbf{g}(\mathbf{x}_{k-1}) \sim \mathcal{N}(\boldsymbol{\mu}^*_k, \boldsymbol{\Sigma}^*_k),\\
&\mathbf{q}_{k-1} \sim \mathcal{N}(0, \boldsymbol{Q}) \text{ and}\\
&\mathbf{x}_k = \mathbf{g}(\mathbf{x}_{k-1}) + \boldsymbol{q}_{k-1}.
\end{aligned} \tag{5.6}$$

We have that $\mathbf{x}_k \sim \mathcal{N}(\boldsymbol{\mu}^*_k, \boldsymbol{\Sigma}^*_k + \boldsymbol{Q})$, given that $\mathbf{x}_k = \mathbf{g}(\mathbf{x}_{k-1}) + \boldsymbol{q}_{k-1}$ is the sum of two normally distributed variables. Subsequently, the joint moments of $\mathbf{x}_{k-1}$ and $\mathbf{x}_k$ are approximated as

$$\begin{aligned}
&\mathbb{E}\left[\begin{pmatrix}\mathbf{x}_{k-1}\\ \mathbf{x}_k\end{pmatrix}\right] \approx \begin{pmatrix}\boldsymbol{m}_{k-1}\\ \boldsymbol{\mu}^*_k\end{pmatrix},\\
&Cov\left[\begin{pmatrix}\mathbf{x}_{k-1}\\ \mathbf{x}_k\end{pmatrix}\right] \approx \begin{pmatrix}\boldsymbol{P}_{k-1} & \boldsymbol{C}\\ \boldsymbol{C}^T & \boldsymbol{\Sigma}^*_k + \boldsymbol{Q}\end{pmatrix}.
\end{aligned} \tag{5.7}$$

Consequently, based on the joint distribution of Gaussian variables (see Appendix A.1), the joint distribution of $\mathbf{x}_{k-1}$ and $\mathbf{x}_k$ can be approximated as Gaussian

$$p(\mathbf{x}_{k-1}, \mathbf{x}_k | \mathbf{y}_{1:k-1}) \approx \mathcal{N}\left(\begin{pmatrix} \mathbf{x}_{k-1} \\ \mathbf{x}_k \end{pmatrix} | \begin{pmatrix} \boldsymbol{m}_{k-1} \\ \boldsymbol{\mu}_k^* \end{pmatrix}, \begin{pmatrix} \boldsymbol{P}_{k-1} & \boldsymbol{C} \\ \boldsymbol{C}^T & \boldsymbol{\Sigma}_k^* + \boldsymbol{Q} \end{pmatrix} \right). \tag{5.8}$$

Because the prediction equations of general Gaussian filtering are obtained from the marginal distribution $\mathbf{x}_k \sim \mathcal{N}(\boldsymbol{m}_k^-, \boldsymbol{P}_k^-)$ [137], see Equations 2.21 and 2.25. The prediction step of a Gaussian filter (based on HMuq) consists of computing the mean $\boldsymbol{m}_k^- = \boldsymbol{\mu}_k^*$ and covariance $\boldsymbol{P}_k^- = \boldsymbol{\Sigma}_k^* + \boldsymbol{Q}$.

□

5.3.3 Auto-initialization of P(0) and iterative estimation of UMM parameters

Based on Lemmas 5.1 and 5.2, we can combine HMuq with UKF or CKF in a hybrid Gaussian filter that allows the auto-initialization of $\mathbf{P}(0)$ and the iterative estimation of MM parameters, as described in the following theorem.

Theorem 5.1 (HNKE). *Assume that HMuq approximates $\mathbf{x}_k|\mathbf{y}_{1:k-1} \sim \mathcal{N}(\boldsymbol{m}_k^-, \boldsymbol{P}_k^-)$ where $\boldsymbol{m}_k^- = \boldsymbol{\mu}_k^*$ and $\boldsymbol{P}_k^- = \boldsymbol{\Sigma}_k^* + \boldsymbol{Q}$ are the prediction equations of Gaussian filtering (Lemma 5.2). If the unscented transform or cubature rule can approximate $\mathbf{x}_k|\mathbf{y}_{1:k} \sim \mathcal{N}(\boldsymbol{m}_k, \boldsymbol{P}_k)$ (update Equation 2.23 of Gaussian filtering) based on $\boldsymbol{m}_k^-$ and $\boldsymbol{P}_k^-$ estimated by HMuq, then we have a hybrid Gaussian filter (HNKE) with the auto-initialization of $\boldsymbol{P}(0)$ and iterative estimation of the UMM parameters at each step k.*

Proof. Let us consider the state space model (SSM) in Equations 2.7. Then, we have that the prediction step of a Gaussian filter for the SSM (Equations 2.7) can be performed by HMuq (Lemma 5.2) as:

$$\mathbf{x}_k|\mathbf{y}_{1:k-1} \sim \mathcal{N}(\boldsymbol{m}_k^-, \boldsymbol{P}_k^-) = \mathcal{N}(\boldsymbol{\mu}_k^*, \boldsymbol{\Sigma}_k^* + \boldsymbol{Q}) \tag{5.9}$$

given that $\mathbf{x}_k = \mathbf{g}(\mathbf{x}_{k-1}) + \boldsymbol{q}_{k-1}$ is the sum of two normally distributed variables, where $\mathbf{g}(\mathbf{x}_{k-1}) \sim \mathcal{N}(\boldsymbol{\mu}_k^*, \boldsymbol{\Sigma}_k^*)$ based on Lemma 5.1 and $\mathbf{q}_{k-1} \sim \mathcal{N}(0, \boldsymbol{Q})$.

Now, considering that the update step performed by unscented transformation can approximate the joint distribution of $\mathbf{x}_k$ and $\mathbf{y}_k = \mathbf{h}(\mathbf{x}_k) + \boldsymbol{r}_k$ as

$$p(\mathbf{x}_k, \mathbf{y}_k|\mathbf{y}_{1:k-1}) \approx \mathcal{N}\left(\begin{pmatrix}\mathbf{x}_k\\ \mathbf{y}_k\end{pmatrix} | \begin{pmatrix}\boldsymbol{\mu}_k^*\\ \boldsymbol{\mu}_k\end{pmatrix}, \begin{pmatrix}\boldsymbol{\Sigma}_k^* + \boldsymbol{Q} & \mathbf{C}_k\\ \mathbf{C}_k^T & \mathbf{B}_k\end{pmatrix}\right). \tag{5.10}$$

We have that

$$p(\mathbf{x}_k|\mathbf{y}_k) \sim \mathcal{N}(\boldsymbol{\mu}_k^* + \mathbf{C}\mathbf{B}^{-1}(\mathbf{y}_k - \boldsymbol{\mu}_k), (\boldsymbol{\Sigma}_k^* + \boldsymbol{Q}) - \mathbf{C}\mathbf{B}^{-1}\mathbf{C}^T) = \mathcal{N}(\mathbf{m}_k, \mathbf{P}_k). \tag{5.11}$$

Therefore, for each measurement step k=1, ... ,K, the updated mean $\boldsymbol{m}_k$ and covariance $\boldsymbol{P}_k$ of $\mathbf{x}_k$ can be computed by performing the following operations.

1. Form sigma points $\bar{S}_k^i$ of $\mathbf{x}_k \sim \mathcal{N}(\boldsymbol{\mu}_k^*, \boldsymbol{\Sigma}_k^* + \boldsymbol{Q})$:

$$\begin{aligned}
&\bar{S}_k^0 = \boldsymbol{\mu}_k^*,\\
&\bar{S}_k^i = \boldsymbol{\mu}_k^* + \sqrt{(n+\lambda)}[\sqrt{(\boldsymbol{\Sigma}_k^* + \boldsymbol{Q})}]_i,\\
&\bar{S}_k^{i+n} = \boldsymbol{\mu}_k^* - \sqrt{(n+\lambda)}[\sqrt{(\boldsymbol{\Sigma}_k^* + \boldsymbol{Q})}]_i, i = 1, ..., n.
\end{aligned} \tag{5.12}$$

2. Propagate the sigma points through the measurement model $\psi_k^i = \mathbf{h}(\bar{S}_k^i)$, i=0, ..., 2n.

3. Compute the predicted mean $\boldsymbol{\mu}_k$, the predicted covariance of the measurement $\mathbf{B}_k$, and the cross-covariance of the state and measurement $\mathbf{C}_k$:

$$\begin{aligned}
\boldsymbol{\mu}_k &= \sum_{i=0}^{2n} W_i^m \psi_k^i\\
\mathbf{B}_k &= \sum_{i=0}^{2n} W_i^c (\psi_k^i - \boldsymbol{\mu}_k)(\psi_k^i - \boldsymbol{\mu}_k)^T + \mathbf{R}\\
\mathbf{C}_k &= \sum_{i=0}^{2n} W_i^c (\bar{S}_k^i - \mathbf{m}_k^-)(\psi_k^i - \boldsymbol{\mu}_k)^T
\end{aligned} \tag{5.13}$$

4. Compute the filter gain $\mathbf{K}_k$, filtered state mean $\mathbf{m}_k$, and covariance $\mathbf{P}_k$, conditional on the measurement $\mathbf{y}_k$:

$$
\begin{aligned}
\mathbf{K}_k &= \mathbf{C}_k \mathbf{B}_k^{-1}, \\
\mathbf{m}_k &= \boldsymbol{\mu}_k^* + \mathbf{K}_k[\mathbf{y}_k - \boldsymbol{\mu}_k], \\
\mathbf{P}_k &= (\boldsymbol{\Sigma}_k^* + \boldsymbol{Q}) - \mathbf{K}_k \mathbf{B}_k \mathbf{K}_k^T.
\end{aligned} \tag{5.14}
$$

The UKF and CKF require initial $\mathbf{P}_{k-1} = \mathbf{P}(0)$ and $\mathbf{m}_{k-1} = \mathbf{x}(0)$ in the prediction step to form sigma points to obtain the $\mathbf{P}_k^-$ and $\mathbf{m}_k^-$. However, because the prediction step in this Gaussian filter is performed by HMuq, we have an auto-initialization of $\mathbf{P}(0)$. Because HMuq can obtain $\mathbf{P}_k^-$ and $\mathbf{m}_k^-$ based only on $\mathbf{m}_{k-1} = \mathbf{x}(0) = \mathbf{x}_{k-1}$, the uncertainty in the HMuq prediction of $\mathbf{x}'_k$ comes from the sample from $q(\boldsymbol{w})$ as follows:

$$
\begin{aligned}
&\hat{p}(\mathbf{x}'_k|\mathbf{x}_{k-1}) = \frac{1}{N}\sum_{i=1}^{N} p(\mathbf{x}'_k|\mathbf{x}_{k-1}, \boldsymbol{\theta}_{k-1,i}); \quad \boldsymbol{\theta}_{k-1,i} = MLP(\mathbf{x}_{k-1}, \boldsymbol{w}_i), \boldsymbol{w}_i \sim q(\boldsymbol{w}), \\
&\hat{p}(\mathbf{x}'_k|\mathbf{x}_{k-1}) \approx \mathcal{N}(\mathbf{x}'_k; \boldsymbol{\mu}_k^*, \boldsymbol{\Sigma}_k^*) \rightarrow \mathbf{x}'_k \sim \mathcal{N}(\boldsymbol{\mu}_k^*, \boldsymbol{\Sigma}_k^*).
\end{aligned} \tag{5.15}
$$

Each ensemble sub-model's output ($\boldsymbol{\theta}_{k-1,i}$) is a sample from the implicit distribution $q(\boldsymbol{w})$ which enables us to estimate both the aleatoric and epistemic uncertainties of the prediction of $\mathbf{x}'_k$ [43, 85, 94] . It is essential to note that in this proof, we use the unscented transformation, but the same procedure can be performed with the cubature rule.

□

5.4 Empirical Evaluation

This section provides experimental results to demonstrate the superior adaptability and estimation properties of the HNKE over baselines. The experiments were designed to demonstrate the robustness and precision of HNKE across five operational settings and to highlight its superior effectiveness in handling highly nonlinear systems without specific setups, such as those required by baseline methods. Therefore, our empirical evaluation aimed to address the following key research questions:

- **Q1**) How does the ensemble size influence the performance of the HNKE based on CKF (HNKE-C) and UKF (HNKE-U)? As posited in Lemma 5.1, we expect that increasing the ensemble size will enhance prediction accuracy by better capturing the

underlying distribution of the system parameters. We varied the ensemble size from 2 to 100 in increments of 1, and measured the impact on the estimation accuracy of the state variables.

- **Q2**) How do the performances of HNKE-C and HNKE-U compare with those of the baseline methods? This comparison is crucial to demonstrate the adaptability and effectiveness of HNKE in handling highly nonlinear system behaviors without requiring different setups varying the initial state covariance matrices (P(0)) such as done by the baselines.

The primary task used to answer both Q1 and Q2 involves using noisy measurements of Xv from test set B to estimate the ground-truth values of all the state variables viable cell density (Xv), total cell (Xt), glucose (GLC), glutamine (GLN), lactate (LAC), ammonium (AMM), and mAb titer. This task represents a fast and low-cost bioprocess monitoring scenario in which only one device is used for measurements, and other state variables are estimated in real time.

5.4.1 Experimental setup

Datasets

The synthetic dataset (SD) has data regarding Monoclonal Antibody (mAb) productions [70] such as those used in the evaluation of the SANTO and BAT approaches (Section 3.4.1). The state variables were Xv, Xt, Glc, Gln, Lac, Amm, and mAb. The SD is composed of 25 runs divided in the following manner: 15 runs used as a training set in the development of the ensemble of MLP (EMLP) and 10 runs to test the HNKE-C, HNKE-U, and baselines (see Table 5.1). In addition, these 25 runs were also divided in terms of the optimal operating setpoint (SP), which represents five different optimal conditions for mAb production in a bioreactor in batch mode, as described in [90, 97]. The runs were generated using the **specific UMM** (Section 2.1.4) for mAb production in a batch bioreactor (three-phase mechanically agitated batch bioreactor) with suspended mammalian hybridoma cell culture at a temperature of 35-37°C, pH of 7 and batch time of 103 h [90, 97]. First, five ground-truth data points (related to each optimal operating setpoint) were generated by solving the specific UMM from 0h to 103h using the parameters proposed by [90] and the initial condition (GLC, GLN, Xv, and Xt) of an optimal operating setpoint [97] (see Table 5.1). The initial conditions for Lac (0 mM), Amm (0.31 mM) and mAb titer (80.6 mg/L) were the same for the five ground truths generated. Finally,

Table 5.1: The synthetic dataset of mAb production has 25 runs divided in 5 different initial condition of GLC, GLN, Xv and Xt which represent 5 different biomanufacturing conditions.

	Initial condition			Number of runs	
	[GLC] mM	[GLN] mM	Xv(0)=Xt(0) Cell/L	Training set A {A.1,A.2,A.3}	Testing set B {B.1,B.2}
SP0	29.1	4.9	2e8	3	2
SP1	100	4.9	2e8	3	2
SP2	29.1	9.0	2e8	3	2
SP3	45	10	2e8	3	2
SP4	100	25	2e9	3	2

Gaussian noise was added to the ground truth to mimic offline and online measurements. The offline measurements were simulated at a frequency of 7h with 15% added Gaussian noise, and the online measurements were simulated at a frequency of 7.5min with also 15% added Gaussian noise, as in [107]. The offline data mimicked the measurements from a hemocytometer (for Xv and Xt), YSI Bioprofiler 200 (for GLC, GLN, LAC, and AMM), and ELISA (for mAb) [76]. The online data mimic measurements from a capacitance sensor or Raman spectroscopy [107, 115, 128]. Therefore, the training set (A) represents an offline measurement of all state variables with a frequency (sample rate) of 7h, resulting in 15 data points for all state variables, where each SP has three runs. Finally, the testing set (B) represents an online measurement of only Xv at a frequency of 7.5 min with 825 data points for all state variables where each SP has two runs, (see Table 5.1).

HMuq development

HMuq uses an ensemble of multilayer perceptrons (MLP) to estimate the $\boldsymbol{\theta}_{k-1,i}$ using $\mathbf{x}_{k-1}$ as the input. Then, the estimated $\boldsymbol{\theta}_{k-1,i}$ is used in the **generic UMM** $\mathbf{g}(\mathbf{x}_{k-1}; \boldsymbol{\theta}_{k-1,i})$ through a serial architecture that has been reported as the most robust even with few data points [110], as shown in Figure 5.1.

Generic UMM. The generic UMM used by HMuq is an ODE system 5.16 representing the cell culture based on mass balances in a general form [41, 107, 109] defined as

$$\mathbf{g}(\mathbf{x}_t;\boldsymbol{\theta}_t) = \begin{array}{l} \frac{dX_V(t)}{dt} = \mu_{X_v}(t)X_V(t), \\ \frac{d[Glc(t)]}{dt} = -\mu_{Glc}(t)X_V(t), \\ \frac{d[Gln(t)]}{dt} = -\mu_{Gln}(t)X_V(t), \\ \frac{d[Lac(t)]}{dt} = \mu_{Lac}(t)X_V(t), \\ \frac{d[Amm(t)]}{dt} = \mu_{Amm}(t)X_V(t), \\ \frac{d[mAb(t)]}{dt} = \mu_{mAb}(t)X_V(t), \end{array} \tag{5.16}$$

where the state variables $\mathbf{x}_t$ are represented by [Xv(t), Glc(t), Gln(t), Lac(t), Amm(t), mAb(t)], and the parameters $\boldsymbol{\theta}_{k-1}$ by [$\mu_{Xv}(t)$, $\mu_{Glc}(t)$, $\mu_{Gln}(t)$, $\mu_{Lac}(t)$, $\mu_{Amm}(t)$, $\mu_{mAb}(t)$]. This system represents the cell growth, uptake of substrates, metabolism, and production process with six parameters: the specific cell growth rate (μ_{X_v}), the specific rates of uptake (consumption) of the main nutrients, glucose (μ_{Glc}) and glutamine (μ_{Gln}), the specific rates of production of the metabolite waste, lactate (μ_{Lac}) and ammonium (μ_{Amm}), and the specific rate of production of mAb(μ_{mAb}). Furthermore, the lack of knowledge in this generic UMM is compensated by the ensemble of MLP, as shown in the schematics in Figure 5.1, which estimates specific rates at $k-1$ based on the information from the culture experiments at $k-1$.

Ensemble of MLP. The training set used to develop the ensemble

$$\mathcal{D}_{EMLP} = \{\{\mathcal{D}_{SPi,j} = \{(\mathbf{x}_{(k-1)\times 7}, \boldsymbol{\theta}_{(k-1)\times 7})\}_{k=1}^{14}\}_{i=0}^{4}\}_{j=1}^{3} \tag{5.17}$$

is composed of 210 samples of $(\mathbf{x}_{(k-1)}, \boldsymbol{\theta}_{(k-1)})$ where each k step represents the sample rate of 7 h used in training set A, and the index j represents the three runs for each of the five optimal conditions represented by i from 0 to 4 (see Table 5.1). In addition, each $\boldsymbol{\theta}_{k-1}$ in 210 samples was obtained by optimizing the $\mathbf{g}(\cdot)$ for each $\{(\mathbf{y}_{(k-1)\times 7}, \mathbf{y}_{(k)\times 7})\}_{i=1}^{15}$ of the training set A. The optimization of $\mathbf{g}(\cdot)$ to obtain each $\boldsymbol{\theta}_{k-1}$ was performed using a neural ODE formulation with one layer [21,59]. Bagging was the ensemble strategy used to create 100 sub-models (MLP) of the ensemble [15]. Each MLP was trained with one of the 100 sub-datasets, where each sub-dataset contained 70% of the original data (147 samples from a total of 210 in $\mathcal{D}_{EMLP}$). This approach to selecting the number of sub-datasets and their size is consistent with common practices in machine learning, particularly to ensure that each model has enough data to learn effectively while maintaining sufficient diversity among the models [15]. It is important to note that the MLP architecture of each sub-model consists of four fully connected layers. The input layer accepts seven features (state variables $\mathbf{x}_{k-1}$, and time) and has 32 neurons using the softsign activation function.

Following this, two hidden layers, each with 32 neurons, also employ softsign activation. The final layer comprises six neurons $\boldsymbol{\theta}_{k-1}$ and does not include an activation function, which is typical in regression models, where the output needs to represent continuous values directly.

HNKE development

HNKE-C and HNKE-U were executed with $\mathbf{R}$ and $\mathbf{Q}$ as defined in Tables 5.3 and 5.4 (see Section 5.4.1), and the initial concentrations of noisy Xv (testing set) are listed in Table 5.2. These initial nosy concentrations are the first state in testing set B, and they were selected to show that HNKE does not require optimal initial conditions such as baselines. Then, it illustrates the robustness of HNKE in handling variations without the need for optimal state adjustments as required by baselines.

Baselines

For comparative analysis, we selected the Joint Cubature Kalman Filter (JCKF) and Joint Unscented Kalman Filter (JUKF) as baseline methods. These were chosen because they can evolve the unshared parameters of a general UMM (Equation 5.16) by using the SANTO approach [60]. The SANTO approach consists of adding a positive quantity for each off-diagonal element of $\mathbf{P}(0)$ which represents the state error covariance between the measured state variable (Xv) and unshared parameters ($\mu_{Glc}(t)$, $\mu_{Gln}(t)$, $\mu_{Lac}(t)$, $\mu_{Amm}(t)$, $\mu_{mAb}(t)$), (see Tables 5.3 and 5.4). In contrast, DNKE was excluded from our study because the current literature does not provide a methodology that allows DNKE to evolve unshared parameters of a general UMM under the specific conditions investigated here. JUKF and JCKF use the generic UMM with the initial parameters estimated by HMuq (see Table 5.5). These initial parameters were evolved by JUKF and JCKF with the SANTO approach during the execution of the algorithms with different setups of $\mathbf{P}(0)$ in Tables 5.3 and 5.4. It is important to point out that each setup of $\mathbf{P}(0)$ was tailored to one optimal operating setpoint and that the initial concentrations of the state variables were the initial concentrations without noise (see Table 5.1). This was performed because the use of initial concentrations with noise (as used by HNKE) would require one setup for each run of the testing set.

Table 5.2: Initial state variables with noise for HNKE-C and HNKE-U during the execution of the empirical task.

Testing Set	X_v(c/mL)	GLC(mM)	GLN(mM)	LAC(mM)	AMM(mM)	mAb(mg/L)
SP0 B1	3.7069e8	29.1	4.9	0.0	0.31	80.6
SPN B2	2.9866e8	29.1	4.9	0.0	0.31	80.6
SP1 B1	2.2116e8	100.0	4.9	0.0	0.31	80.6
SP1 B2	2.7206e8	100.0	4.9	0.0	0.31	80.6
SP2 B1	2.1897e8	29.1	9.0	0.0	0.31	80.6
SP2 B2	1.8335e8	29.1	9.0	0.0	0.31	80.6
SP3 B1	1.4507e8	45.0	10.0	0.0	0.31	80.6
SP3 B2	1.4138e8	45.0	10.0	0.0	0.31	80.6
SP4 B1	2.1481e9	100.0	25.0	0.0	0.31	80.6
SP4 B2	2.1041e9	100.0	25.0	0.0	0.31	80.6

Table 5.3: The measurement noise variance $\mathbf{R}$, and error covariance matrix of process model $\mathbf{Q}_{i,i}$ for HNKE-C, HNKE-U, JCKF and JUKF during the execution of task.

NKE	Error Covariance Matrix	SP0 {B.1,B.2}	SP1 {B.1,B.2}	SP2 {B.1,B.2}	SP3 {B.1,B.2}	SP4 {B.1,B.2}
HNKE-C	R (c^2/mL^2)	$(0.969e8)^2$	$(1.05e8)^2$	$(1.1e8)^2$	$(1.0e8)^2$	$(1.1e8)^2$
	Q_{X_v,X_v} (c^2/mL^2)	$(6e5)^2$	$(6e5)^2$	$(6e5)^2$	$(6e5)^2$	$(3.1e7)^2$
HNKE-U	R (c^2/mL^2)	$(0.967e8)^2$	$(1.03e8)^2$	$(1.12e8)^2$	$(0.97e8)^2$	$(1.08e8)^2$
	Q_{X_v,X_v} (c^2/mL^2)	$(6.2e5)^2$	$(6.2e5)^2$	$(6.2e5)^2$	$(6.2e5)^2$	$(3.2e7)^2$
JUKF	R (c^2/mL^2)	$(7.5e7)^2$	$(7.53e7)^2$	$(8.53e7)^2$	$(0.9e8)^2$	$(0.995e8)^2$
	Q_{X_v,X_v} (c^2/mL^2)	$(6.2e5)^2$	$(6.2e5)^2$	$(6.2e5)^2$	$(6.2e5)^2$	$(3.2e7)^2$
JCKF	R (c^2/mL^2)	$(7.35e7)^2$	$(7.6e7)^2$	$(8.45e7)^2$	$(0.91e8)^2$	$(1e8)^2$
	Q_{X_v,X_v} (c^2/mL^2)	$(5.95e5)^2$	$(6e5)^2$	$(6e5)^2$	$(6e5)^2$	$(3.1e7)^2$
HNKE-C,	$Q_{GLC,GLC}$ (mM^2)	$(6.02)^2$	$(6.02)^2$	$(6.02)^2$	$(6.02)^2$	$(6.02)^2$
HNKE-U,	$Q_{GLN,GLN}$ mM^2	$(0.61)^2$	$(0.61)^2$	$(0.61)^2$	$(0.61)^2$	$(0.61)^2$
JCKF,	$Q_{LAC,LAC}$ (mM^2)	$(4.29)^2$	$(4.29)^2$	$(4.29)^2$	$(4.29)^2$	$(4.29)^2$
JUKF	$Q_{AMM,AMM}$ (mM^2)	$(0.70)^2$	$(0.70)^2$	$(0.70)^2$	$(0.70)^2$	$(0.70)^2$
	$Q_{mAb,mAb}$ (mg^2/L^2)	$(141.98)^2$	$(141.98)^2$	$(141.98)^2$	$(141.98)^2$	$(141.98)^2$

Table 5.4: Initial state error covariance matrix $\mathbf{P}$(t=0), and error covariance matrix of process model $\mathbf{Q}_{i,i}$ for JCKF and JUKF during the execution of task.

Error Covariance Matrix	SP0 {B.1,B.2}	SP1 {B.1,B.2}	SP2 {B.1,B.2}	SP3 {B.1,B.2}	SP4 {B.1,B.2}
$Q_{\mu_{X_v},\mu_{X_v}}$ (c^2/mL^2)	$(0.001)^2$	$(0.001)^2$	$(0.001)^2$	$(0.001)^2$	$(0.001)^2$
$Q_{\mu_{GLC},\mu_{GLC}}$ (mM^2)	$(0.4)^2$	$(0.4)^2$	$(0.4)^2$	$(0.4)^2$	$(0.4)^2$
$Q_{\mu_{GLN},\mu_{GLN}}$ (mM^2)	$(0.05)^2$	$(0.05)^2$	$(0.05)^2$	$(0.05)^2$	$(0.05)^2$
$Q_{\mu_{LAC},\mu_{LAC}}$ (mM^2)	$(0.2)^2$	$(0.2)^2$	$(0.2)^2$	$(0.2)^2$	$(0.2)^2$
$Q_{\mu_{AMM},\mu_{AMM}}$ (mM^2)	$(0.06)^2$	$(0.06)^2$	$(0.06)^2$	$(0.06)^2$	$(0.06)^2$
$Q_{\mu_{mAb},\mu_{mAb}}$ (mg^2/L^2)	$(10.1)^2$	$(10.1)^2$	$(10.1)^2$	$(10.1)^2$	$(10.1)^2$
$P(0)_{X_v,X_v}$ (c^2/mL^2)	$(1.7e8)^2$	$(7.2e7)^2$	$(1.8e7)^2$	$(5.8e7)^2$	$(1.4e8)^2$
$P(0)_{GLC,GLC}$ (mM^2)	$(0.1)^2$	$(0.1)^2$	$(0.1)^2$	$(0.1)^2$	$(0.1)^2$
$P(0)_{GLN,GLN}$ (mM^2)	$(0.1)^2$	$(0.1)^2$	$(0.1)^2$	$(0.1)^2$	$(0.1)^2$
$P(0)_{LAC,LAC}$ (mM^2)	$(0.1)^2$	$(0.1)^2$	$(0.1)^2$	$(0.1)^2$	$(0.1)^2$
$P(0)_{AMM,AMM}$ (mM^2)	$(0.1)^2$	$(0.1)^2$	$(0.1)^2$	$(0.1)^2$	$(0.1)^2$
$P(0)_{mAb,mAb}$ (mg^2/L^2)	$(0.1)^2$	$(0.1)^2$	$(0.1)^2$	$(0.1)^2$	$(0.1)^2$
$P(0)_{\mu_{X_v},\mu_{X_v}}$ (c^2/mL^2)	$(0.1)^2$	$(0.1)^2$	$(0.1)^2$	$(0.1)^2$	$(0.1)^2$
$P(0)_{\mu_{GLC},\mu_{GLC}}$ (mM^2)	$(0.1)^2$	$(0.1)^2$	$(0.1)^2$	$(0.1)^2$	$(0.1)^2$
$P(0)_{\mu_{GLN},\mu_{GLN}}$ (mM^2)	$(0.1)^2$	$(0.1)^2$	$(0.1)^2$	$(0.1)^2$	$(0.1)^2$
$P(0)_{\mu_{LAC},\mu_{LAC}}$ (mM^2)	$(0.1)^2$	$(0.1)^2$	$(0.1)^2$	$(0.1)^2$	$(0.1)^2$
$P(0)_{\mu_{AMM},\mu_{AMM}}$ (mM^2)	$(0.1)^2$	$(0.1)^2$	$(0.1)^2$	$(0.1)^2$	$(0.1)^2$
$P(0)_{\mu_{mAb},\mu_{mAb}}$ (mg^2/L^2)	$(0.1)^2$	$(0.1)^2$	$(0.1)^2$	$(0.1)^2$	$(0.1)^2$
$P(0)_{X_v,\mu_{GLC}}$ (mM/h)	1e-6	2e-6	0.008	0.04	0.0001
$P(0)_{X_v,\mu_{GLN}}$ (mM/h)	0.001	0.0011	0.0011	0.001	0.001
$P(0)_{X_v,\mu_{LAC}}$ (mM/h)	0.055	0.00056	0.015	0.015	0.055
$P(0)_{X_v,\mu_{AMM}}$ (mM/h)	0.011	0.00012	0.000115	0.00011	0.011
$P(0)_{X_v,\mu_{mAb}}$ (mg/h)	0.01	0.02	0.000025	0.000001	0.000514

Table 5.5: Initial parameters of general UMM during the execution of for JCKF and JUKF.

Initial Parameters	SP0 {B.1,B.2}	SP1 {B.1,B.2}	SP2 {B.1,B.2}	SP3 {B.1,B.2}	SP4 {B.1,B.2}
$\mu_{\mathrm{X_v}}$ (c/mL)	0.033	0.0042	0.0042	0.0042	0.04
μ_{GLC} (mM)	0.40e-9	0.44e-9	0.0027e-9	0.24e-9	0.25e-9
μ_{GLN} (mM)	1.20e-10	1.21e-10	0.717e-10	1.15e-10	10.04e-11
μ_{LAC} (mM)	0.58e-9	7.046e-10	0.000043e-10	4.49e-10	6.72e-10
μ_{AMM} (mM)	1.17e-10	4.59e-11	7.39e-11	8.96e-11	1.22e-10
μ_{mAb} (mg/L)	1.85e-8	1.87e-8	1.87e-8	1.77e-8	1.29e-8

Metrics

The filter consistency test is the metric used to define the covariance matrices $\mathbf{Q}$, $\mathbf{R}$, and $\mathbf{P}$ used by NHKE-U, NHKE-C, and the baselines (see Tables 5.3 and 5.4). This test is the normalized innovations squared (NIS) Chi-square Test [35]. It is used to verify that the NHKE-U, NHKE-C, and baselines perform correctly with $\mathbf{P}$, $\mathbf{Q}$, and $\mathbf{R}$ selected [11, 137]. The NIS Chi-square test verifies that the innovation is unbiased and white using hypothesis testing (χ^2 test) [11, 137]. For a properly functioning filter, the innovation sequence $e_{y,k} = [\mathbf{y}_k - \boldsymbol{\mu}_k]$ should have zero mean and be white, with covariance denoted as $\mathbf{B}_k$ [11, 132, 137]. The NIS is defined as $NIS_k = e_{y,k}\mathbf{B}_k^{-}e_{y,k}$, and the mean of NIS as $\mu(NIS) = \frac{1}{N}\sum_{k=1}^{N} e_{y,k}\mathbf{B}_k^{-}e_{y,k}$ from a single run of the NHKE-U, NHKE-C, and baselines. Therefore, to test for 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 $P(N \times \mu(NIS) \in [r1, r2]|H_0) = 1-\alpha$ where m is the number of measured state variables and N is the number of samples from the measured state variables. Furthermore, in the two-sided 95% confidence region, we have $[r1, r2] = [\chi^2_{Nm}(0.025), \chi^2_{Nm}(0.975)]$. Further details are provided in [11, 132, 137]. In our experiments, the unique measured state variable was Xv. Then, we obtained the following filter consistent test (FCT) based on the testing set B2 with N=824: $P(N \times \mu(NIS) \in [745.3, 904.39]|H_0) = 1-\alpha$? Table 5.6 lists the acceptable values for the filter consistency test.

Finally, to specifically answer Q1 and Q2, we used the root-mean-square percentage error (RMSPE) to assess the estimations performed by HNKE-U, HNKE-C, and baselines during the execution of the task.

Table 5.6: The N $\times\mu(NIS)$ obtained by the HNKE-U, HNKE-C and baselines are acceptable values on the filter consistent test with N=824: $P(N\times\mu(NIS) \in [745.3, 904.39]|H_0) = 1-\alpha$?.

Testing set	HNKE-C	HNKE-U	JCKF	JUKF
SP0 B2.1	894.23	897.75	891.33	856.86
SP0 B2.2	746.65	749.64	899.56	864.75
SP1 B2.1	841.18	873.33	863.66	879.31
SP1 B2.2	815.67	846.95	881.01	896.95
SP2 B2.1	814.16	786.59	864.74	848.82
SP2 B2.2	810.20	782.62	894.42	877.96
SP3 B2.1	749.82	795.86	753.34	769.87
SP3 B2.2	832.09	883.02	817.98	835.92
SP4 B2.1	750.34	768.01	750.56	752.55
SP4 B2.2	873.75	896.00	886.94	889.44

5.4.2 Results

The results are organized by research questions **Q1**, and **Q2**.

Answering Q1. Figure 5.2 shows the RMSPE between the ground truth of test set B.1 and the estimations of HNKE-C and HNKE-U during the execution of empirical task using different the ensemble sizes. Here, we can see that increasing the ensemble size enhances the prediction accuracy by better capturing the underlying distribution of general UMM parameters. All RMSPE values dropped quickly and started to converge, which suggests that good estimation can also be obtained with ensembles with sizes ranging from 50. It is important to point out that HNKE-C and HNKE-U presented similar performances, since the RMSPE values over the different ensemble sizes were very similar (see Figure 5.2). Upon detailed analysis of the RMSPE values using a heat-map of RMSPE values with test set B.1 (see Figure 5.4) and considering the small variance in these values across the methods, we found that while HNKE-C frequently achieved lower RMSPEs (in testing set B.1), the practical difference in performance between HNKE-C and HNKE-U is minimal. This is confirmed by the plots of the state variables estimations in Figure 5.3, where we cannot visualize the difference between the estimation done by HNKE-C (blue dash-dot-dot lines) and HNKE-U (light-blue dotted lines), and both had estimations close to the ground truth values (red lines). Both versions performed robustly, with HNKE-C slightly outperforming HNKE-U in terms of achieving the minimum RMSPE more frequently in testing set B.1.

HNKE-C achieved the minimum RMSPE in 16 of the 30 comparisons, whereas HNKE-U achieved the minimum RMSPE in 14 instances (see Figure 5.4). However, the small scale of the variance suggests that both methods are comparably effective. Furthermore, HNKE-U achieved the minimum RMSPE in 16 of the 30 comparisons with testing set B.2, whereas HNKE-C achieved the minimum RMSPE in 14 instances, see Figure 5.6.

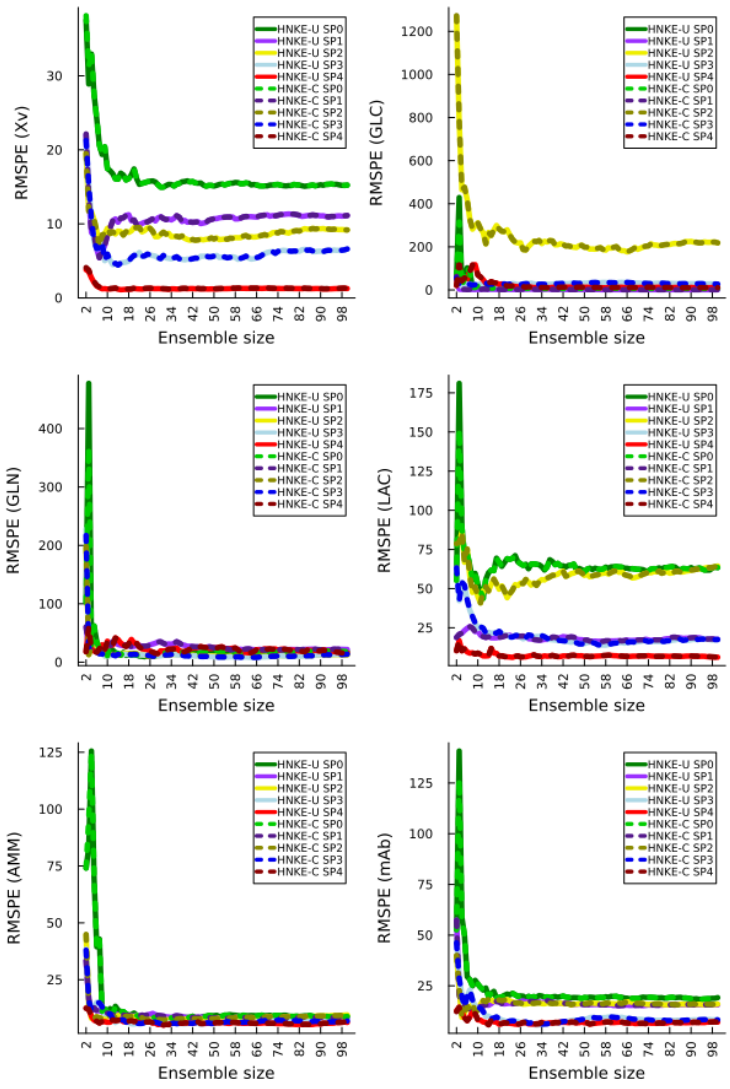

Figure 5.2: RMSPE values between ground truth state variables of testing set B.1 and the estimation of HNKE-C and HNKE-U during the execution of empirical task using different ensemble size. The ensemble vary from 2 to 100 by incrementing 1. The plot shows that as the ensemble size increases, the prediction accuracy improves, and that HNKE-C and HNKE-U show similar performance trends.

Answering Q2. Both HNKE-C and HNKE-U show better performance compared to JCKF and JUKF during the execution of the empirical task to estimate the state variables of mAb production (see Figure 5.3 and Figure 5.5). This is also confirmed by the heat-map plots in Figure 5.4 and Figure 5.6 in the Appendix, where the estimation performed by HNKE-C and HNKE-U presented the lowest RMSPE values. This implies that HNKE-C and HNKE-U are capable of adequately capturing the highly nonlinear dynamics of mAb production. The auto-initialization feature of HNKE models, which simplifies the model initialization process without compromising performance, is a notable advantage over traditional methods that require explicit initialization of $\mathbf{P}(0)$ for each operational condition. In the majority of cases, the JCKF (light purple) and JUKF (purple) estimations did not follow the highly nonlinear behavior of the ground truth (red lines). JCKF and JUKF had better performance with not highly nonlinear cases of the testing set B.1, such as GLC in SP2, LAC in SP3, and mAb in SP4 (see Figure 5.3). It is important to note that JUKF and JCKF had the worst estimation related to the state variables that were not measured but the best estimation related to Xv (measured variable). This indicates that JUKF and JCKF perform well with directly observed data, but struggle with unmeasured state variables. Furthermore, the results of the JCKF and JUKF with the SANTO approach can be improved with negative values in off-diagonal elements of P(0). However, we did not use this strategy because it generates unconventional behavior in the Kalman gain, where the values converge from negative values to zero.

5.5 Discussion

Based on our empirical results, some points should be discussed.

Reduced reliance on large volumes of data. The empirical results support HNKE as an effective approach to address the limitation of the biopharma industry in generating large amounts of data for the development of robust soft sensors. In this thesis, HNKE models were trained on offline data (representing a limited amount of data) and tested on online data, demonstrating their versatility and applicability in real-world scenarios. This approach aligns with the trend in biopharma to enhance process development efficiency, reduce costs, and increase production, all while dealing with limited data resources.

Adaptability. The capability of HNKE models to automatically initialize $\mathbf{P}(0)$ offers a considerable benefit compared to the baselines, which depend on adjusting the diagonal and off-diagonal elements of $\mathbf{P}(0)$. This adjustment process is challenging, particularly in highly nonlinear systems, where the characteristics of HNKE provide a more efficient and reliable alternative. Furthermore, the baselines required specific sets of $\mathbf{P}(0)$ and

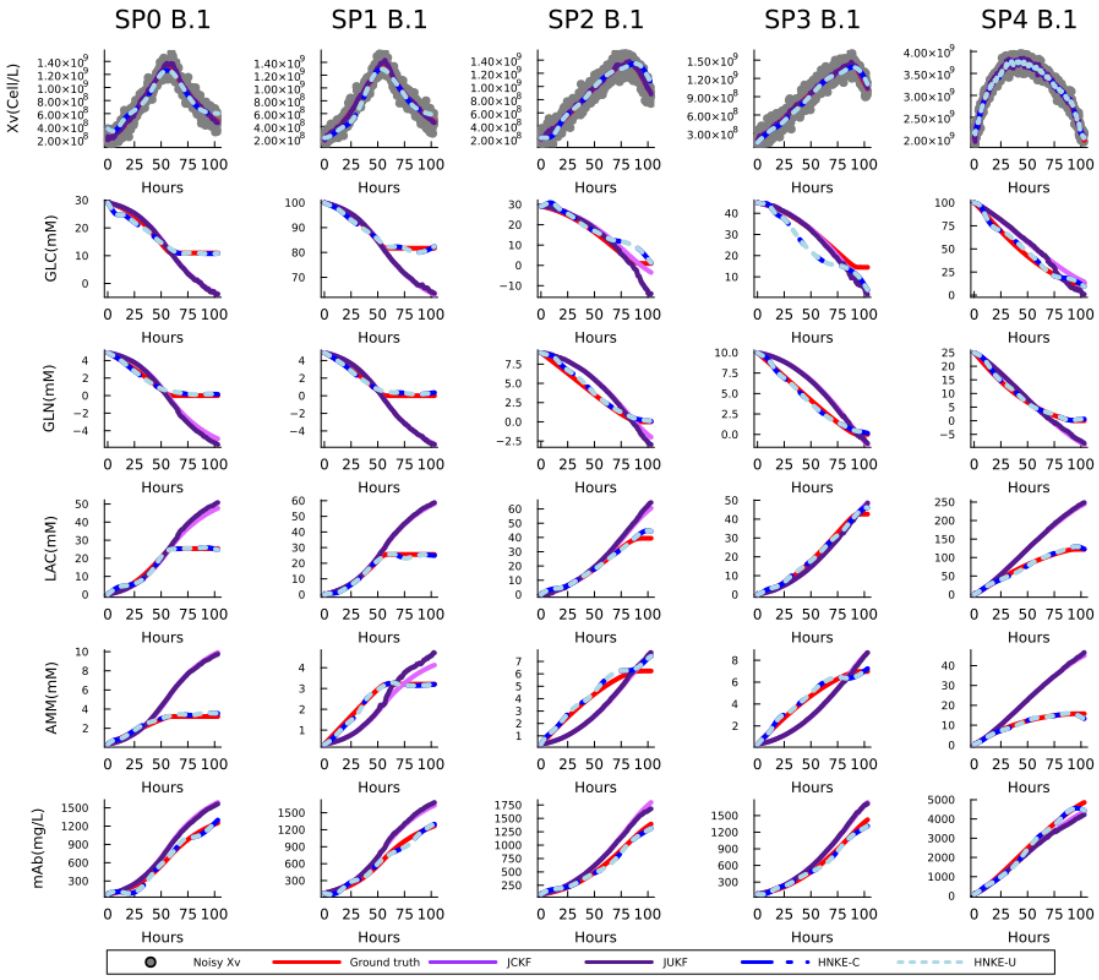

Figure 5.3: Comparison of state variable estimations by JCKF, JUKF, HNKE-C, and HNKE-U using the noisy measurements of Xv from testing set B.1. The plot highlights the precision and adaptability of HNKE in capturing the highly nonlinear dynamics of mAb production across different real operational conditions, contrasted against the performance of baselines.

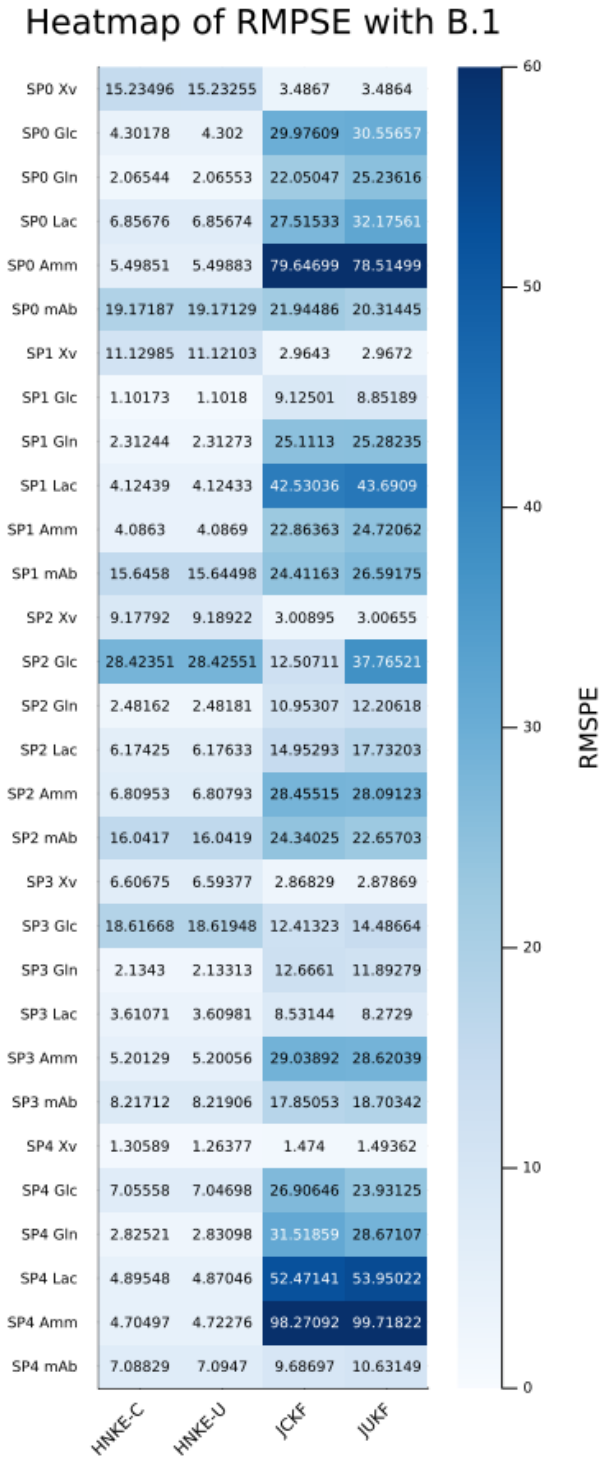

Figure 5.4: Heatmap of RMSPE comparing the performance of HNKE-C, HNKE-U, JCKF, and JUKF in estimating state variables from testing set B.1. The heatmap color codes the RMSPE values to visually represent the accuracy of each method across the five different conditions from SP0 to SP4. This visualization underscores the accuracy of HNKE-C and HNKE-U in modeling highly nonlinear bioprocess dynamics, with consistently lower RMSPE values compared to the baselines.

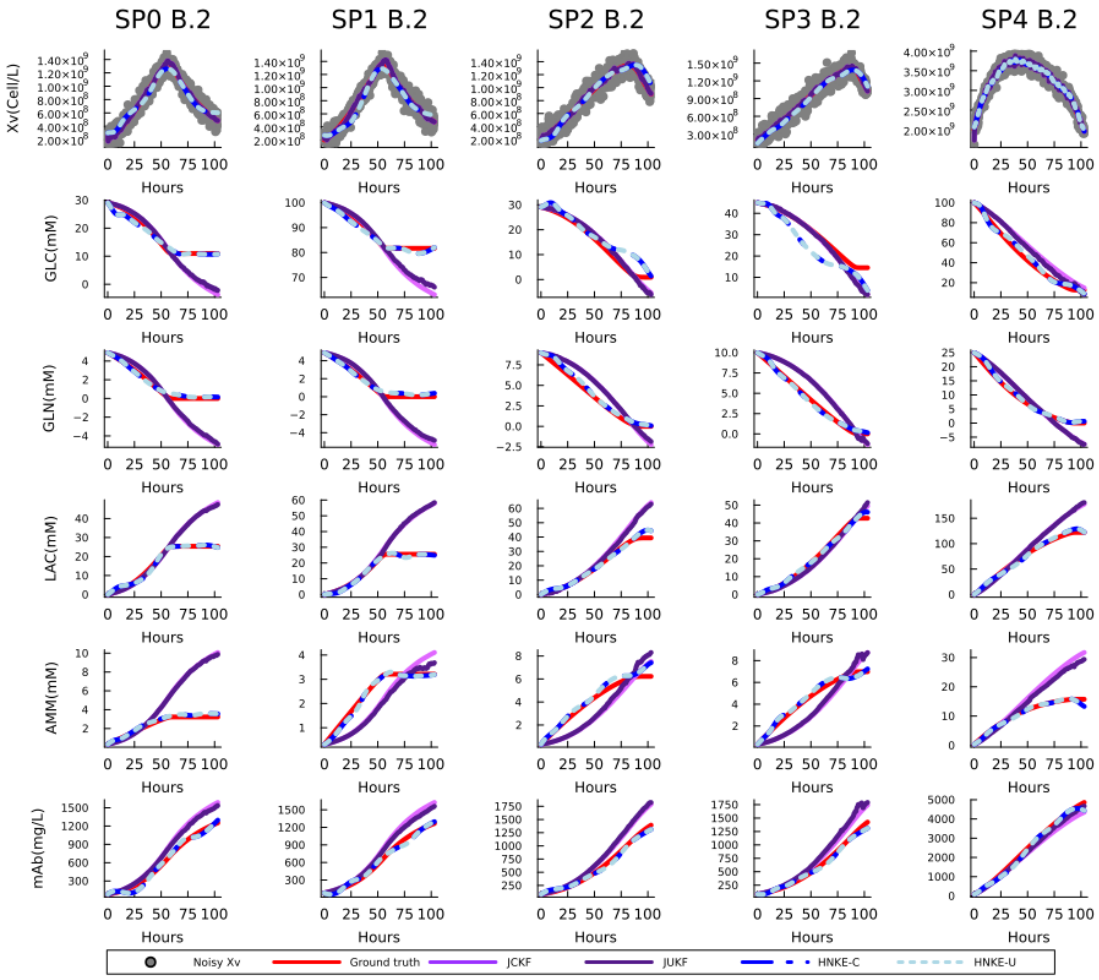

Figure 5.5: Comparison of state variable estimations by JCKF, JUKF, HNKE-C, and HNKE-U using the noisy measurements of Xv from testing set B.2. The plot highlights the precision and adaptability of HNKE in capturing the highly nonlinear dynamics of mAb production across different real operational conditions, contrasted against the performance of baselines.

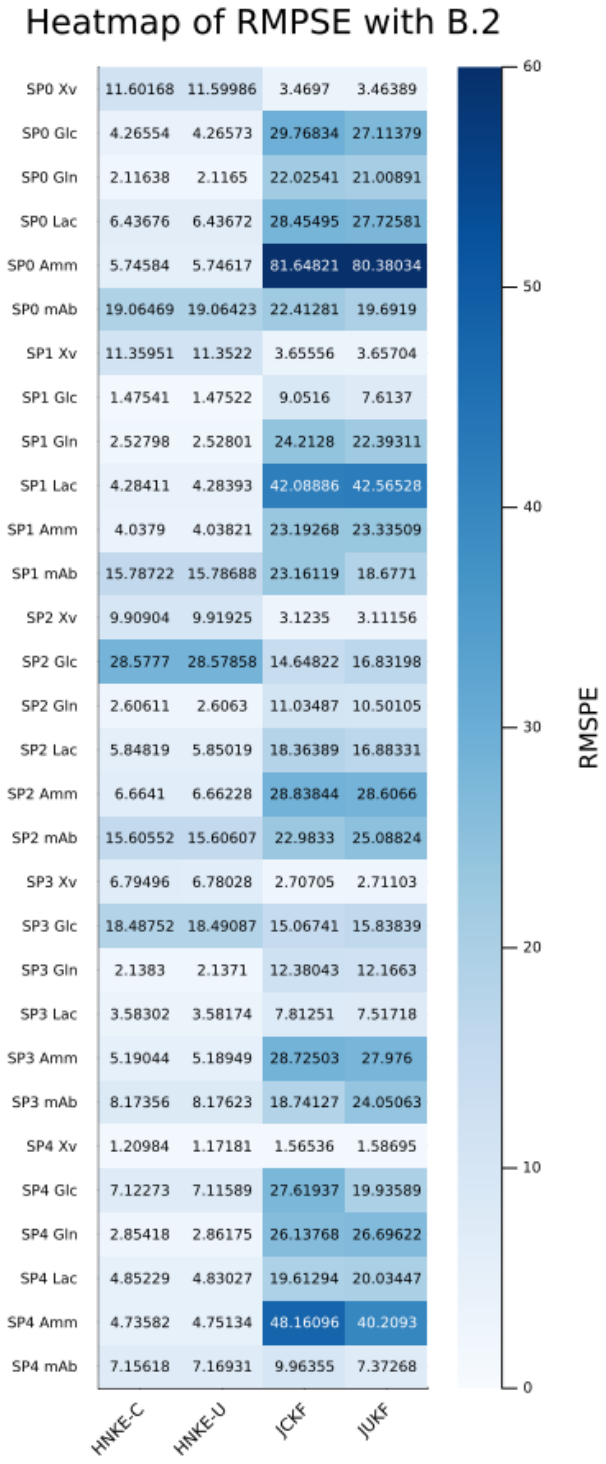

Figure 5.6: Heatmap of RMSPE comparing the performance of HNKE-C, HNKE-U, JCKF, and JUKF in estimating state variables from testing set B.2. The heatmap color codes the RMSPE values to visually represent the accuracy of each method across the five different conditions from SP0 to SP4. This visualization underscores the accuracy of HNKE-C and HNKE-U in modeling highly nonlinear bioprocess dynamics, with consistently lower RMSPE values compared to the baselines.

initial concentrations for each operational setpoint (see Table 5.5 and Table 5.1), making them difficult to set up and tune. HNKE eliminated the need to set these specific initial conditions and allowed the filter to adapt to updating these conditions in real time. This feature is especially beneficial in complex biomanufacturing scenarios, where setting precise initial conditions for different operational setpoints is challenging.

Practicality. The empirical findings underscore the operational efficiency of HNKE, which relies solely on the definition of measurement noise $\mathbf{R}$ and process noise $\mathbf{Q}$ matrices. Unlike JCKF and JUKF, which require the specification of initial state error covariance $\mathbf{P}(0)$ and optimal state variables (ground truth), HNKE simplifies the process by iteratively estimating $\mathbf{P}$ and the parameters $\boldsymbol{\theta}$ during the filter execution. Furthermore, HNKE can use the initial state concentrations directly obtained from the testing set (without the need for optimal state variables), facilitating a more straightforward setup.

Integration. The adaptability of the HNKE can be a key factor in facilitating its integration with advanced control systems, such as Model Predictive Control (MPC) and digital twins, enhancing bioprocess monitoring and control. This is because the MPC and digital twins are systems that demand rapid adaptation to changing process conditions. For instance, in a bioreactor, MPC can utilize real-time state estimates of HNKE to dynamically adjust inputs such as nutrient feeds, optimizing conditions for cell growth and productivity. Similarly, integrating HNKE with a digital twin allows for real-time process simulations, supporting proactive adjustments and scenario analysis to prevent issues like culture contamination and optimize production outcomes.

Applicability. Although the present study used synthetic data from batch production in a bioreactor, the HNKE approach can be used for fed-batch processes with appropriate training data. By training the ensemble of MLP with new data for fed-batch production, HNKE can be adapted for broader applications in biomanufacturing. Moreover, the input to MLP can be extended beyond the current state variables and time to include additional factors, such as pH and temperature, further expanding the versatility of the model.

Limitations. It is essential to point out that the performance of HNKE relies on the size of the ensemble and the strategy used for generating $q(\boldsymbol{w})$ as described by the Lemma 5.1, and this can be seen as the HNKE limitations.

Computational Complexity. The HNKE framework incorporates an ensemble of N MLP models to estimate the parameters of the UMM at each time step k. Each MLP receives the previous state vector of size n as input and outputs a set of parameters of size n_{out}. The computational complexity of a single MLP with l layers and n_l neurons in each hidden layer is approximately:

$$\mathcal{O}(n \cdot n_l + (l-2) \cdot n_l^2 + n_l \cdot n_{out})$$

For simplicity, assuming that the sizes of the input and output layers are small compared to the hidden layers, this can be approximated as:

$$\mathcal{O}(l \cdot n_l^2)$$

In the case of an ensemble consisting of N MLPs, the total complexity is the product of the complexity of a single MLP and the number of MLPs in the ensemble, since each MLP operates independently. Therefore, the total computational complexity of the MLP ensemble is [173]:

$$\mathcal{O}(N \cdot l \cdot n_l^2)$$

This additional complexity makes HNKE more computationally demanding than both CKF and UKF, especially when the number of MLPs (N) or the number of neurons per layer (n_l) grows larger than the state dimension n. While the computational complexity of CKF and UKF is typically $\mathcal{O}(n^3)$ [27, 102, 129, 161], the inclusion of the MLP ensemble in HNKE results in a higher overall complexity. Thus, for scenarios where $N \geq n$, $n_l \geq n$, and $l \geq n$, HNKE exceeds the computational complexity of classical UKF and CKF filters.

5.6 Summary

This thesis introduced a hybrid nonlinear Kalman estimator, which is a novel hybrid Gaussian filter for fast and low-cost bioprocess monitoring. The key features of HNKE include i) auto-initialization of the $\mathbf{P}(0)$, ii) iterative update of generic UMM parameters in real-time for highly nonlinear bioprocesses, and iii) reduced reliance on large data volumes during the training phase. Our empirical results, using synthetic bioprocess data, demonstrate that HNKE models outperform the JCKF and JUKF under the conditions studied. HNKE eliminates the need to define specific sets of NKE components (parameters) for each operational setpoint, thereby reducing the effort and expertise required for model tuning. Our results also indicate that HNKE models are robust, with consistent filter performance across different biomanufacturing conditions. Furthermore, a comparative analysis of HNKE-C and HNKE-U revealed minimal practical differences in performance. Our findings suggest that HNKE provides a robust, adaptable, and cost-effective solution for bioprocess monitoring,

aligning well with the goals of the biopharma industry to enhance monitoring efficiency and reduce operational costs.

Chapter 6

Conclusion and Future work

Fast and low-cost bioprocess monitoring is crucial for the biopharmaceutical industry. It can reduce operational costs by minimizing reliance on offline measurements and supports regulatory compliance, ultimately making advanced therapies more accessible and affordable for patients. However, the development of soft sensors based on classic NKEs with UMM faces significant limitations in achieving fast and low-cost bioprocess monitoring, particularly when addressing diverse biomanufacturing conditions. Therefore, the main goal of this thesis is to support the design and development of soft sensors based on NKEs with UMM for fast and low-cost monitoring of highly nonlinear bioprocesses, which was achieved through the following contributions.

6.1 Contributions

The main goal was achieved with the three main contributions of this thesis: SANTO, BAT and HNKE. The flowchart (Figure 6.1) illustrates the versatility of the proposed approaches (HNKE, SANTO, and BAT) each tailored to different biomanufacturing complexities, ensuring that the appropriate approach is selected based on the specific system requirements and computational constraints. The process of design and development a soft sensor based on SANTO, BAT and HNKE for fast and low-cost bioprocess monitoring begins by collecting data and analyzing biomanufacturing conditions. If the system requires integration with complex control systems such as Model Predictive Control (MPC) or Digital Twins (DT), the HNKE is the most indicated, as it provides high adaptability and real-time monitoring with generic UMMs. For systems without complex control integration but with low computational requirements, HNKE is also preferred if a generic UMM

is used; otherwise, the SANTO approach is suitable for simpler bioprocesses and specific UMMs, offering real-time parameter estimation with low computational overhead. If tuning of all NKE components is required for new biomanufacturing conditions, the Batch Bayesian Auto-Tuning (BAT) approach is ideal, allowing comprehensive tuning of NKE components. In addition, the three methods proposed in this thesis (SANTO, BAT, and HNKE) each address specific aspects of the challenges faced in fast and low-cost bioprocess monitoring. They complement each other but were not merged into a single implementation.Each method was developed to solve particular problems within the scope of this thesis, ensuring clarity and focus in addressing the identified research questions.

The proposed approaches in the thesis are designed to be general and can be applied to any NKE, including the EKF, UKF, and CKF. Therefore, the selection of a specific NKE depends primarily on the characteristics of the problem at hand or the preferences of the researcher or engineer [95]. Each filter has its strengths and weaknesses, the decision should be based on the specific needs of the bioprocess monitoring task [52, 95]. More specifically, we have that SANTO and BAT can be applied to any NKE and the HNKE can be based on the unscented transformation or cubature rule.

It is important to point out that the proposed approaches (SANTO, BAT and HNKE) can be translated to fields beyond bioprocess monitoring, such as navigation. The approaches developed are grounded in general mathematical frameworks that are applicable across a range of dynamic systems where real-time state estimation and uncertainty quantification are critical. In navigation systems, for instance, these filters are already widely used for tasks such as trajectory estimation, sensor fusion, and vehicle tracking. The generality of the mathematical formulations (state-space models, recursive estimation, etc.) allows the principles to be adapted for different applications with minimal adjustments. The same core steps (prediction, update, and error covariance adjustment) used in bioprocess monitoring can be employed in navigation systems to estimate the position and velocity of a vehicle based on sensor data. Thus, while this thesis focuses on bioprocesses, the fundamental methodologies are versatile and can be adapted for fields such as robotics, aerospace, and autonomous systems. The primary requirement for translation is adjusting the model dynamics and observation functions to suit the specific characteristics of the new field.

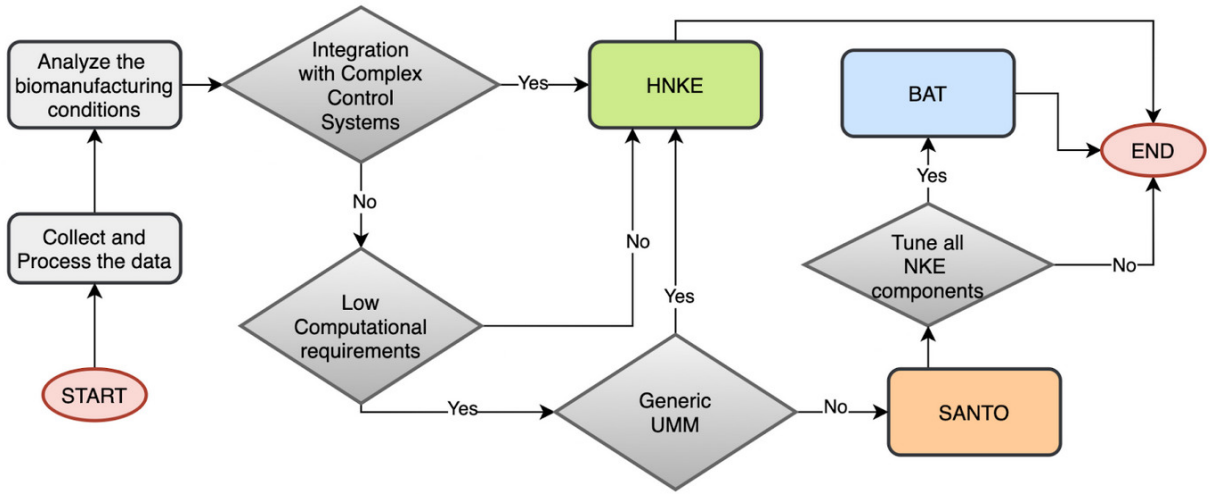

Figure 6.1: Design and development a soft sensor based on SANTO, BAT and HNKE (the main contributions) for fast and low-cost bioprocess monitoring.

6.1.1 Specific initiAl coNdiTiOn (SANTO) in joint NKEs

SANTO is the first main contribution of this thesis. In the chapter 3, we described the SANTO approach, which was proposed to address the RQ1 (Section 1.2). The SANTO approach consists of adding a quantity to the state error covariances between the measured state variable and an unshared parameter in $\mathbf{P}(t=0)$ to enable the Kalman gain to be updated during the execution of joint NKE. The SANTO approach offers several advantages, making it a versatile solution for any NKE such as EKF, UKF, and CKF. It facilitates real-time monitoring for specific and similar biomanufacturing conditions, enabling the accurate estimation of unshared parameters within specific UMM. This approach is particularly beneficial for bioprocesses monitoring, as it requires low computational power and is easy to implement, making it suitable for implementation on micro-controllers and other low-cost hardware. However, SANTO has some limitations, including its low adaptability (generalization), necessitating re-tuning for each new biomanufacturing condition. Additionally, it may exhibit unconventional Kalman gain behavior under certain conditions and is ineffective with generic UMMs for highly nonlinear bioprocesses. Therefore, on the basis of the advantages and limitations of the SANTO approach, it is better suited (applicable) for developing specific and simple soft sensors because of its straightforward implementation and low computational requirements, which make it ideal for real-time monitoring of

bioprocesses under similar conditions. The simplicity and ease of use of SANTO make it particularly suited for applications where specific UMMs are used to model the bioprocess, requiring minimal tuning and adjustment. Its applicability to low-cost and fast monitoring setups, especially on microcontrollers, makes it a perfect fit for straightforward monitoring tasks that do not require extensive computational resources or adaptability to varied conditions, see Figure 6.1.

6.1.2 Batch Bayesian Auto-Tuning (BAT) for NKEs

BAT is the second main contribution of this thesis. In the chapter 4, we described the BAT approach, which was proposed to address the RQ2 (Section 1.2). BAT 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. As the first method capable of auto-tuning all NKE components, BAT operates offline with reduced data requirements, making it applicable to any new biomanufacturing condition. It empirically outperforms traditional auto-tuning methods by consistently estimating unimodal posterior distributions, ensuring accurate and stable parameter estimation (especially related to unshared parameters). This accuracy facilitates fast convergence and stability in NKE applications. Additionally, BAT can estimate the initial condition and parameters required by the SANTO approach, thus avoiding the unconventional behavior of the Kalman gain caused by SANTO. While BAT excels with specific UMM for highly nonlinear bioprocesses, it demands high computational resources in high dimensions and is not suitable for generic UMMs. Despite these limitations, BAT is crucial for supporting the design and development of soft sensors based on NKEs with UMMs for fast and low-cost monitoring of highly nonlinear bioprocesses. The ability of BAT to automatically tune all NKE components ensures accurate and stable parameter estimation, which is essential for the reliable performance of soft sensors, see Figure 6.1.

6.1.3 Hybrid Nonlinear Kalman Estimator (HNKE)

HNKE is the third main contribution of this thesis. In the chapter 5, we described the HNKE approach, which was proposed to address the RQ3 (Section 1.2). Furthermore, HNKE overcomes the limitations of SANTO. HNKE is the firt hybrid Gaussian filter to integrate a hybrid dynamic model with uncertainty quantification (HMuq) and Cubature

or Unscented Kalman Filters (CKF/UKF). It introduces the innovative feature of auto-initializing the state error covariance matrix $\mathbf{P}(0)$. This reduces the time and effort required for tuning, making it a fast and low-cost approach. HNKE is particularly effective for monitoring bioprocesses with unknown mechanisms using generic UMMs and can handle highly nonlinear bioprocesses across a wide range of conditions. It reduces the need for large data sets, ensuring high adaptability and real-time parameter estimation. However, its implementation comes with the limitation of higher computational requirements than the SANTO approach. Therefore, on the basis of the advantages and limitations of the HNKE approach, we have that, unlike SANTO, HNKE is more applicable for developing complex control tools such as digital twins and Model Predictive Control (MPC) due to its advanced capabilities in handling diverse and highly nonlinear bioprocesses across different conditions. The ability of HNKE to operate with reduced data requirements and to adapt dynamically to changing conditions makes it ideal for integration with sophisticated control systems like digital twins (DT) and MPC, which demand high adaptability, robustness, and the capacity to handle complex scenarios in real-time. These systems benefit from the capability of HNKE to provide accurate state estimations and predictions, facilitating proactive adjustments and optimizations in the biomanufacturing process, see Figure 6.1.

6.2 Future work

6.2.1 Handling massive proportion of missing labels in multivariate long-term time series forecasting

Training Deep Learning (DL) models with missing labels is a challenge in diverse engineering applications. Missing value imputation methods have been proposed to try to address this problem, but their performance is affected with Massive Proportion of Missing Labels (MPML). In our preliminar work [55], we present an approach for handling MPML in Multivariate Long-Term Time Series Forecasting. It is an two-step process where interpolation (using Gaussian Processes Regression (GPR) and domain knowledge from experts) and prediction model are separated to enable the integration of prior domain knowledge. First, a set of samples of the possible interpolation of the missing outputs are generated by the GPR based on the domain knowledge. Second, the observed input sensor data and interpolated labels from GPR are used to train the prediction model. We evaluated our approach with the development of a soft-sensor with one real datasets to forecast the biomass during recombinant adeno-associated virus (rAAV) production in bioreactors. Our experimental results demonstrate the potential of the approach through quantitative evaluation of the

generated forecasts in a case that would be extremely difficult to train a DL model due to MPML. The future works are: i) include uncertainty quantification in the prediction, ii) evaluate the approach with a real extremely unlabelled multivariate time series dataset of bioprocess.

6.2.2 BAT online version

While BAT currently is an offline approach, future research could explore the development of an online version, if it is possible to get a closed-form of BAT posterior Equation 4.5 (for example, using conjugate priors), BAT can be used as an online application. However, the online application of a closed form of BAT posterior may require the use of 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 (online measurements) is used to generate innovation errors in NKEs.

6.2.3 HNKE based on Bayesian neural network and Kolmogorov Arnold networks

Future work may focus on exploring other ensemble methods or Bayesian neural networks to further enhance the capabilities of the HNKE and extend its applicability to a broader range of bioprocess monitoring scenarios, including cases of culture contamination. However, replacing Multi-Layer Perceptrons (MLPs) with Kolmogorov-Arnold Networks (KANs) [91] in the development of HNKE may significantly increase its adaptability to different scenarios and the estimation accuracy. KANs are proposed as promising alternatives to MLPs for various reasons, making them potentially beneficial for HNKE development. KANs can significantly enhance the accuracy, interpretability, and efficiency of bioprocess monitoring systems. They have demonstrated superior accuracy in function fitting tasks compared to MLPs, which is ideal for the precise requirements of bioprocess control. Their greater interpretability is crucial for understanding model decisions and optimizing bioprocess parameters. Additionally, KANs' flexibility, due to their learnable activation functions on edges, allows for better modeling of complex nonlinear relationships typical in bioprocesses. This supports the goals of fast and low-cost bioprocess monitoring essential for advancing biomanufacturing processes and ensuring high-quality biopharmaceutical products.

6.2.4 Challenges of the design phase of DT Development Life Cycle based on soft sensors (NKE with UMM)

DT is critical to the pharmaceutical industry because it is a key enabler to biomanufacturing in industry 4.0 [116, 145, 156]. However, despite its potential benefits, digital-twin-based biomanufacturing has not been fully implemented [142, 171]. Additionally, unlike many other manufacturing industries, biopharma still needs to fulfill the requirements of Industry 3.0, that is, moving from manual to automated systems [145]. In this thesis, the proposed approaches (SANTO, BAT, and HNKE) to support the development of soft sensors based on NKEs with UMM as the foundation of DT for fast and low-cost bioprocess monitoring [48, 49]. However, implementing digital twins in a real-world biomanufacturing environment poses significant technical challenges, including the need for advanced modeling techniques in the design phase of DT Development Life Cycle (SDLC) [39, 42, 142]. In the Design phase, two significant issues are general (often emerge across various applications) in DT [39, 142]:

Lack of approach for Modelling Decision making in the high level of abstraction [2, 136]: Designing a digital twin system involves planning how will be created the replica of the biomanufacturing process that can accurately represent its behavior under different conditions. A key aspect of this is modeling the decision-making process. The decisions could range from adjustments in the manufacturing process due to changes in input materials and environmental conditions to decisions on maintenance schedules, shutdowns, alert messages, and more. Capturing the complexity of these decision-making processes in a model can be challenging, especially in a field as complex and variable-dependent as biomanufacturing. This is because decisions in biomanufacturing are often dependent on a multitude of factors, many of which are probabilistic or uncertain, making it a complex task to develop an accurate and robust decision-making model. However, to the best of our knowledge, there is a lack of specifically tailored approaches for modeling the decision-making performed by DT in high-level [39]. The effects of this issue are poor representation of the system, and low transparency, explainability, and risk management.

Our preliminary work [57] addresses this issue by examining the feasibility and effectiveness of a probabilistic domain model which integrates Decision Making in Personalized Monitoring Systems (DM-PMS) with Bayesian Networks (BNs) as a methodology for designing decision-making processes in digital twins at the early stages of the systems development life cycle. However, this probabilistic domain model requires more validations in different scenarios.

Addressing Resilience manually at a high level of abstraction [104, 150, 160]: Resilience in a digital twin system for biomanufacturing refers to the system's ability

to withstand changes, disturbances, or unexpected events, and to continue functioning correctly. Resilience should be addressed in the early stages of implementing the DT system by identifying IoT critical objects and threats and selecting the best mitigation strategies. It should be done in the design phase [29, 47, 79, 80, 89] because it allows: (i) to deal with the complexity of the problems, (ii) effective communication, (iii) complete understanding, (iv) reduce costs, (v) predict behavior, (vi) reuse and (vii) analyze the feasibility (financial and practical) of the DT system. However, designing for Resilience often requires a deep understanding of the potential sources of disturbances, their impacts, and how the system can respond to ensure continued operation. Given the complexity and variability of biomanufacturing processes, manually addressing Resilience can be a significant challenge. As an effect, this process can be time-consuming and prone to error as it involves anticipating a wide range of potential scenarios and formulating appropriate responses.

Our preliminary works [56,61] indicate that the integration of NER model, Architectural Design Decision model and database of threats and mitigation strategies in an automatic framework can enable the automatic extraction of IoT critical objects from documents and the guided selection of threats and mitigation strategies. Consequently, enabling the designing resilient DT system.

References

[1] Pieter Abbeel, Adam Coates, Michael Montemerlo, Andrew Y Ng, Sebastian Thrun, et al. Discriminative training of kalman filters. In *Robotics: Science and systems*, volume 2, page 1, 2005.

[2] Ashwin Agrawal, Robert Thiel, Pooja Jain, Vishal Singh, and Martin Fischer. Digital twin: Where do humans fit in? *Automation in Construction*, 148:104749, 2023.

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

[4] Amani A Alahmadi, Jennifer A Flegg, Davis G Cochrane, Christopher C Drovandi, and Jonathan M Keith. 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.

[5] Joan Albiol, Jordi Robusté, Carles Casas, and Manel Poch. Biomass estimation in plant cell cultures using an extended kalman filter. *Biotechnology progress*, 9(2):174–178, 1993.

[6] Ienkaran Arasaratnam and Simon Haykin. Cubature kalman filters. *IEEE Transactions on automatic control*, 54(6):1254–1269, 2009.

[7] Andrea Arnold. When artificial parameter evolution gets real: particle filtering for time-varying parameter estimation in deterministic dynamical systems. *Inverse Problems*, 39(1):014002, 2022.

[8] Nicholas Assimakis and Maria Adam. Kalman filter riccati equation for the prediction, estimation, and smoothing error covariance matrices. *International Scholarly Research Notices*, 2013, 2013.

[9] Neha Aswal, Baidurya Bhattacharya, and Subhamoy Sen. 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*, pages 223–252. World Scientific, 2022.

[10] Neha Aswal, Subhamoy Sen, and Laurent Mevel. Switching kalman filter for damage estimation in the presence of sensor faults. *Mechanical Systems and Signal Processing*, 175:109116, 2022.

[11] Yaakov Bar-Shalom, X Rong Li, and Thiagalingam Kirubarajan. *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons, 2001.

[12] Alberto Bertipaglia, Barys Shyrokau, Mohsen Alirezaei, and Riender Happee. 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)*, pages 670–677. IEEE, 2022.

[13] Biomatik. How much does it cost to make a custom antibody?, 2022. Accessed: 2024-05-30.

[14] Boulaid Boulkroune, Kurt Geebelen, Jia Wan, and Ellen van Nunen. Auto-tuning extended kalman filters to improve state estimation. In *2023 IEEE Intelligent Vehicles Symposium (IV)*, pages 1–6. IEEE, 2023.

[15] Leo Breiman. Bagging predictors. *Machine learning*, 24:123–140, 1996.

[16] Peter J. Brockwell. Time series analysis. *Encyclopedia of Statistics in Behavioral Science*, 2005.

[17] Steve Brooks, Andrew Gelman, Galin Jones, and Xiao-Li Meng. *Handbook of markov chain monte carlo*. CRC press, 2011.

[18] Vincent Brunner. *Soft sensors for Pichia pastoris bioprocesses: filling the gaps between uncertain process data and knowledge*. PhD thesis, Technische Universität München, 2022.

[19] Michael C Burkhart, David M Brandman, Carlos E Vargas-Irwin, and Matthew T Harrison. The discriminative kalman filter for nonlinear and non-gaussian sequential bayesian filtering. *arXiv preprint arXiv:1608.06622*, 2016.

[20] Emily Capra, A Godfre, Alberto Loche, and Jeff Smith. Gene-therapy innovation: Unlocking the promise of viral vectors. *Recuperate by: https://www. mckinsey. com/industries/life-sciences/our-insights/gene-therapy-innovat ion-unlocking-the-promise-of-viral-vectors*, 2021.

[21] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations. *arXiv preprint arXiv:1806.07366*, 2018.

[22] Zhaozhong Chen, Harel Biggie, Nisar Ahmed, Simon Julier, and Christoffer Heckman. Kalman filter auto-tuning through enforcing chi-squared normalized error distributions with bayesian optimization. *arXiv preprint arXiv:2306.07225*, 2023.

[23] Zhaozhong Chen, Christoffer Heckman, Simon Julier, and Nisar Ahmed. Weak in the nees?: Auto-tuning kalman filters with bayesian optimization. In *2018 21st International Conference on Information Fusion (FUSION)*, pages 1072–1079. IEEE, 2018.

[24] Zhaozhong Chen, Christoffer Heckman, Simon Julier, and Nisar Ahmed. Time dependence in kalman filter tuning. In *2021 IEEE 24th International Conference on Information Fusion (FUSION)*, pages 1–8. IEEE, 2021.

[25] Viki Chopda, Aron Gyorgypal, Ou Yang, Ravendra Singh, Rohit Ramachandran, Haoran Zhang, George Tsilomelekis, Shishir PS Chundawat, and Marianthi G Ierapetritou. 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.

[26] Ioannis Christakis, Odysseas Tsakiridis, Dionisis Kandris, and Ilias Stavrakas. A kalman filter scheme for the optimization of low-cost gas sensor measurements. *Electronics*, 13(1):25, 2023.

[27] Fred Daum. Nonlinear filters: beyond the kalman filter. *IEEE Aerospace and Electronic Systems Magazine*, 20(8):57–69, 2005.

[28] Jesús-Adolfo Mejía de Dios and Efrén Mezura-Montes. Metaheuristics: A julia package for single- and multi-objective optimization. *Journal of Open Source Software*, 7(78):4723, 2022.

[29] Kemal A Delic. On resilience of iot systems: The internet of things (ubiquity symposium). *Ubiquity*, 2016(February):1–7, 2016.

[30] Vivek Dua and Pinky Dua. A simultaneous approach for parameter estimation of a system of ordinary differential equations, using artificial neural network approximation. *Industrial & engineering chemistry research*, 51(4):1809–1814, 2012.

[31] Mary J Dunlop and Richard M Murray. Towards biological system identification: fast and accurate estimates of parameters in genetic regulatory networks. *parameters*, 1000(4), 2006.

[32] Robert Dürr and Steffen Waldherr. Hybrid simulation algorithm for efficient numerical solution of population balance equations. *IFAC-PapersOnLine*, 51(2):290–295, 2018.

[33] Joseph Emerson, Bo Kara, and Jarka Glassey. Multivariate data analysis in cell gene therapy manufacturing. *Biotechnology Advances*, page 107637, 2020.

[34] Mohammad Ennab and Hamid Mcheick. Designing an interpretability-based model to explain the artificial intelligence algorithms in healthcare. *Diagnostics*, 12(7):1557, 2022.

[35] Alexandros Evangelidis and David Parker. Quantitative verification of kalman filters. *Formal Aspects of Computing*, 33(4-5):669–693, 2021.

[36] World Economic Forum. The bio revolution: Innovations transforming economies, societies, and our lives, 2020.

[37] Cory Tyler Fraser. *Adaptive extended Kalman filtering strategies for autonomous relative navigation of formation flying spacecraft*. PhD thesis, Carleton University, 2019.

[38] Paul Frogerais, Jean-Jacques Bellanger, and Lotfi Senhadji. Various ways to compute the continuous-discrete extended kalman filter. *IEEE Transactions on Automatic Control*, 57(4):1000–1004, 2011.

[39] Aidan Fuller, Zhong Fan, Charles Day, and Chris Barlow. Digital twin: Enabling technologies, challenges and open research. *IEEE access*, 8:108952–108971, 2020.

[40] Carina L Gargalo, Simoneta Caño de las Heras, Mark Nicholas Jones, Isuru Udugama, Seyed Soheil Mansouri, Ulrich Krühne, and Krist V Gernaey. Towards the development of digital twins for the bio-manufacturing industry. In *Digital Twins*, pages 1–34. Springer, 2020.

[41] Chetan T Goudar. Computer programs for modeling mammalian cell batch and fed-batch cultures using logistic equations. *Cytotechnology*, 64(4):465–475, 2012.

[42] Michael Grieves and John Vickers. *Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems*, pages 85–113. Springer International Publishing, Cham, 2017.

[43] Fredrik K Gustafsson, Martin Danelljan, and Thomas B Schon. Evaluating scalable bayesian deep learning methods for robust computer vision. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 318–319, 2020.

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

[45] Shan He, Zixiong Han, Cristóvão Iglesias, Varun Mehta, and Miodrag Bolic. A real-time respiration monitoring and classification system using a depth camera and radars. *Frontiers in Physiology*, 13:799621, 2022.

[46] Inmaculada Hernandez, Samuel W Bott, ASet al Patel, Collin G Wolf, Alexa R Hospodar, Shivani Sampathkumar, and William H Shrank. Pricing of monoclonal antibody therapies: higher if used for cancer. *Am J Manag Care*, 24(2):109–112, 2018.

[47] Jeffrey W Herrmann. *Engineering decision making and risk management*. John Wiley & Sons, 2015.

[48] Christoph Herwig, Ralf Pörtner, and Johannes Möller. *Digital Twins: Applications to the Design and Optimization of Bioprocesses*. Springer, 2021.

[49] Christoph Herwig, Ralf Pörtner, and Johannes Möller. *Digital Twins: tools and concepts for smart biomanufacturing*. Springer, 2021.

[50] Zach Hetzler, Noah Lott, Aditi Dey Poonam, Selen Dalgan, and Qingshan Wei. Single-use biosensors for biomanufacturing: Perspective on the state-of-the-art. *Current Opinion in Biomedical Engineering*, page 100512, 2023.

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

[52] Dai Hong-de, Dai Shao-wu, Cong Yuan-cai, and Wu Guang-bin. Performance comparison of ekf/ukf/ckf for the tracking of ballistic target. *TELKOMNIKA Indonesian Journal of Electrical Engineering*, 10(7), 2012.

[53] Hanwen Huang, Andreas Handel, and Xiao Song. A bayesian approach to estimate parameters of ordinary differential equation. *Computational statistics*, 35:1481–1499, 2020.

[54] Mohammed Khalil Hussain, Bajel Mohammed Alshadeedi, and Rashid Hejeejo. Optimized kalman filters for sensorless vector control induction motor drives. *International Journal of Electrical and Computer Engineering*, 13(1):17, 2023.

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

[56] Cristovão F Iglesias, Rongchen Guo, Pedro Nucci, Claudio Miceli, and Miodrag Bolic. Automated extraction of iot critical objects from iot storylines, requirements and user stories via nlp. In *2023 10th IEEE Swiss Conference on Data Science (SDS)*, pages 104–107. IEEE, 2023.

[57] Cristovão F Iglesias, Pedro Nucci, Claudio Miceli, and Miodrag Bolic. Demde: Decision making design based on bayesian network for personalized monitoring system. In *2023 26th International Conference on Information Fusion (FUSION)*, pages 1–8. IEEE, 2023.

[58] Cristovão Freitas Iglesias and Miodrag Bolic. How not to make the joint extended kalman filter fail with unstructured mechanistic models. *Sensors*, 24(2), 2024.

[59] Jr. Xingge Xu Varun Mehta Mounia Akassou Alina Venereo-Sanchez Nabil Belacel Amine Kamen Iglesias, Cristovão Freitas and Miodrag Bolic. Monitoring the recombinant adeno-associated virus production using extended kalman filter. *Processes*, 10(11):2180, 2022.

[60] Cristovão Freitas Iglesias Jr and Miodrag Bolic. How not to make the joint extended kalman filter fail with unstructured mechanistic models. *Sensors*, 24(2):653, 2024.

[61] Cristovao Freitas Iglesias Jr, Claudio Miceli, and Miodrag Bolic. An architectural design decision model for resilient iot application. *arXiv preprint arXiv:2306.10429*, 2023.

[62] Cristovão Freitas Iglesias Jr, Milica Ristovski, Miodrag Bolic, and Miroslava Cuperlovic-Culf. raav manufacturing: The challenges of soft sensing during upstream processing. *Bioengineering*, 10(2):229, 2023.

[63] Cristovão Freitas Iglesias Jr, Xingge Xu, Varun Mehta, Mounia Akassou, Alina Venereo-Sanchez, Nabil Belacel, Amine Kamen, and Miodrag Bolic. Monitoring the recombinant adeno-associated virus production using extended kalman filter. *Processes*, 10(11):2180, 2022.

[64] Hiroaki Inoue, Koji Hukushima, and Toshiaki Omori. Estimating distributions of parameters in nonlinear state space models with replica exchange particle marginal metropolis–hastings method. *Entropy*, 24(1):115, 2022.

[65] Andrew H. Jazwinski. Stochastic processes and filtering theory. *Mathematics in science and engineering*, (64), 1970.

[66] Zhong Ji and Martin Brown. Joint state and parameter estimation for biochemical dynamic pathways with iterative extended kalman filter: comparison with dual state and parameter estimation. *The Open Automation and Control Systems Journal*, 2(1), 2009.

[67] Xue-Bo Jin, Ruben Jonhson Robert Jeremiah, Ting-Li Su, Yu-Ting Bai, and Jian-Lei Kong. The new trend of state estimation: from model-driven to hybrid-driven methods. *Sensors*, 21(6):2085, 2021.

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

[69] Simon J Julier and Jeffrey K Uhlmann. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3):401–422, 2004.

[70] I Jyothilekshmi and NS Jayaprakash. Trends in monoclonal antibody production using various bioreactor systems. *Journal of microbiology and biotechnology*, 31(3):349, 2021.

[71] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.

[72] Karel J Keesman. *System identification: an introduction*. Springer Science & Business Media, 2011.

[73] Masoud Khodarahmi and Vafa Maihami. A review on kalman filter models. *Archives of Computational Methods in Engineering*, pages 1–21, 2022.

[74] Thanh Tung Khuat, Robert Bassett, Ellen Otte, Alistair Grevis-James, and Bogdan Gabrys. Applications of machine learning in antibody discovery, process development, manufacturing and formulation: Current trends, challenges, and opportunities. *Computers & Chemical Engineering*, page 108585, 2024.

[75] Sukkeun Kim, Ivan Petrunin, and Hyo-Sang Shin. A review of kalman filter with artificial intelligence techniques. In *2022 Integrated Communication, Navigation and Surveillance Conference (ICNS)*, pages 1–12. IEEE, 2022.

[76] Cleo Kontoravdi, Efstratios N Pistikopoulos, and Athanasios Mantalaris. Systematic development of predictive mathematical models for animal cell cultures. *Computers & Chemical Engineering*, 34(8):1192–1198, 2010.

[77] Richard E Kopp and Richard J Orford. Linear regression applied to system identification for adaptive control systems. *Aiaa Journal*, 1(10):2300–2306, 1963.

[78] Martin Kornecki and Jochen Strube. Accelerating biologics manufacturing by upstream process modelling. *Processes*, 7(3):166, 2019.

[79] Alexander Kossiakoff, Steven M Biemer, Samuel J Seymour, and David A Flanigan. *Systems engineering principles and practice*. John Wiley & Sons, 2020.

[80] Gerald Kotonya and Ian Sommerville. *Requirements engineering: processes and techniques*. Wiley Publishing, 1998.

[81] Gennady Yu Kulikov and Maria V Kulikova. Accurate numerical implementation of the continuous-discrete extended kalman filter. *IEEE Transactions on Automatic Control*, 59(1):273–279, 2013.

[82] Maria V Kulikova and G Yu Kulikov. Adaptive ode solvers in extended kalman filtering algorithms. *Journal of Computational and Applied Mathematics*, 262:205–216, 2014.

[83] Sarantos Kyriakopoulos, Kok Siong Ang, Meiyappan Lakshmanan, Zhuangrong Huang, Seongkyu Yoon, Rudiyanto Gunawan, and Dong-Yup Lee. Kinetic modeling of mammalian cell culture bioprocessing: the quest to advance biomanufacturing. *Biotechnology Journal*, 13(3):1700229, 2018.

[84] Roger Labbe. Kalman and bayesian filters in python. *Chap*, 7(246):4, 2020.

[85] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

[86] Annu Lambora, Kunal Gupta, and Kriti Chopra. Genetic algorithm-a literature review. In *2019 international conference on machine learning, big data, cloud and parallel computing (COMITCon)*, pages 380–384. IEEE, 2019.

[87] Albert H Li, Philipp Wu, and Monroe Kennedy. Replay overshooting: Learning stochastic latent dynamics with the extended kalman filter. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 852–858. IEEE, 2021.

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

[89] Dong Liu, Ralph Deters, and Wen-Jun Zhang. Architectural design for resilience. *Enterprise Information Systems*, 4(2):137–152, 2010.

[90] Yang Liu and Rudiyanto Gunawan. Bioprocess optimization under uncertainty using ensemble modeling. *Journal of biotechnology*, 244:34–44, 2017.

[91] Ziming Liu, Yixuan Wang, Sachin Vaidya, Fabian Ruehle, James Halverson, Marin Soljačić, Thomas Y Hou, and Max Tegmark. Kan: Kolmogorov-arnold networks. *arXiv preprint arXiv:2404.19756*, 2024.

[92] Lennart Ljung. Asymptotic behavior of the extended kalman filter as a parameter estimator for linear systems. *IEEE Transactions on Automatic Control*, 24(1):36–50, 1979.

[93] Yu Luo, Varghese Kurian, and Babatunde A Ogunnaike. Bioprocess systems analysis, modeling, estimation, and control. *Current Opinion in Chemical Engineering*, 33:100705, 2021.

[94] Andrey Malinin, Bruno Mlodozeniec, and Mark Gales. Ensemble distribution distillation. *arXiv preprint arXiv:1905.00076*, 2019.

[95] Mahendra Mallick, Xiaoqing Tian, and Jing Liu. Evaluation of measurement converted kf, ekf, ukf, ckf, and pf in gmti filtering. In *2021 International Conference on Control, Automation and Information Sciences (ICCAIS)*, pages 21–27. IEEE, 2021.

[96] Carl-Fredrik Mandenius and Robert Gustavsson. Mini-review: Soft sensors as means for pat in the manufacture of bio-therapeutics. *Journal of Chemical Technology & Biotechnology*, 90(2):215–227, 2015.

[97] Gheorghe Maria, Cristiana Luminita Gijiu, Iulian Cebanu, Cristina Maria, and Carmen Tociu. In-silico optimization of a batch bioreactor for mabs production in relationship to the net evolution of the hybridoma cell culture. *methods*, 6:7, 2019.

[98] Stefano Mariani and Alberto Corigliano. Impact induced composite delamination: state and parameter identification via joint and dual extended kalman filters. *Computer methods in applied mechanics and engineering*, 194(50-52):5242–5272, 2005.

[99] Peter S. Maybeck. *Stochastic Models, Estimation, and Control*. Mathematics in Science and Engineering. Academic Press, 1982.

[100] Lisa Mears, Stuart M Stocks, Mads O Albaek, Gürkan Sin, and Krist V Gernaey. Mechanistic fermentation models for process design, monitoring, and control. *Trends in biotechnology*, 35(10):914–924, 2017.

[101] David Medina-Ortiz, Ashkan Khalifeh, Hoda Anvari-Kazemabad, and Mehdi D Davari. Interpretable and explainable predictive machine learning models for data-driven protein engineering. *bioRxiv*, pages 2024–02, 2024.

[102] Qingwen Meng and Xuyou Li. Minimum cauchy kernel loss based robust cubature kalman filter and its low complexity cost version with application on ins/od integrated navigation system. *IEEE Sensors Journal*, 22(10):9534–9542, 2022.

[103] Vincent Michel, Alexandre Gramfort, Gaël Varoquaux, Evelyn Eger, and Bertrand Thirion. Total variation regularization for fmri-based prediction of behavior. *IEEE transactions on medical imaging*, 30(7):1328–1340, 2011.

[104] Partha Pratim Mondal, Abhinav Galodha, Vishal Kumar Verma, Vijai Singh, Pau Loke Show, Mukesh Kumar Awasthi, Brejesh Lall, Sanya Anees, Katrin Pollmann, and Rohan Jain. Review on machine learning-based bioprocess optimization, monitoring, and control systems. *Bioresource Technology*, page 128523, 2022.

[105] André Moser, Christian Appl, Simone Brüning, and Volker C Hass. Mechanistic mathematical models as a basis for digital twins. In *Digital Twins*, pages 133–180. Springer, 2020.

[106] Chitra Murugan. Soft sensors for biomass monitoring during low cost cellulase production. In *Biomass*. IntechOpen, 2021.

[107] Harini Narayanan, Lars Behle, Martin F Luna, Michael Sokolov, Gonzalo Guillén-Gosálbez, Massimo Morbidelli, and Alessandro Butté. Hybrid-ekf: Hybrid model coupled with extended kalman filter for real-time monitoring and control of mammalian cell culture. *Biotechnology and Bioengineering*, 117(9):2703–2714, 2020.

[108] Harini Narayanan, Martin Luna, Michael Sokolov, Alessandro Butté, and Massimo Morbidelli. Hybrid models based on machine learning and an increasing degree of process knowledge: application to cell culture processes. *Industrial & Engineering Chemistry Research*, 61(25):8658–8672, 2022.

[109] Harini Narayanan, Michael Sokolov, Massimo Morbidelli, and Alessandro Butté. A new generation of predictive models: the added value of hybrid models for manufacturing processes of therapeutic proteins. *Biotechnology and Bioengineering*, 116(10):2540–2549, 2019.

[110] Harini Narayanan, Moritz von Stosch, Fabian Feidl, Michael Sokolov, Massimo Morbidelli, and Alessandro Butté. Hybrid modeling for biopharmaceutical processes: advantages, opportunities, and implementation. *Frontiers in Chemical Engineering*, 5:1157889, 2023.

[111] Lawrence Nelson and Edwin Stear. The simultaneous on-line estimation of parameters and states in linear systems. *IEEE Transactions on automatic Control*, 21(1):94–98, 1976.

[112] Tam NT Nguyen, Sha Sha, Moo Sun Hong, Andrew J Maloney, Paul W Barone, Caleb Neufeld, Jacqueline Wolfrum, Stacy L Springs, Anthony J Sinskey, and Richard D Braatz. Mechanistic model for production of recombinant adeno-associated virus via triple transfection of hek293 cells. *Molecular Therapy-Methods & Clinical Development*, 21:642–655, 2021.

[113] Sarfaraz Niazi and Sunitha Lokesh. Understanding bioprocessing. In *Biopharmaceutical Manufacturing, Volume 2*, 2053-2563, pages 1–1 to 1–42. IOP Publishing, 2022.

[114] Maximilian Nitsch, David Stenger, and Dirk Abel. Automated tuning of nonlinear kalman filters for optimal trajectory tracking performance of auvs. *arXiv preprint arXiv:2304.03565*, 2023.

[115] Thomas Noll and M Biselli. Dielectric spectroscopy in the cultivation of suspended and immobilized hybridoma cells. *Journal of biotechnology*, 63(3):187–198, 1998.

[116] Michael Ogunsanya and Salil Desai. Physics-based and data-driven modeling for biomanufacturing 4.0. *Manufacturing Letters*, 2023.

[117] Kaveh Ohadi, Raymond L Legge, and Hector M Budman. Development of a soft-sensor based on multi-wavelength fluorescence spectroscopy and a dynamic metabolic model for monitoring mammalian cell cultures. *Biotechnology and bioengineering*, 112(1):197–208, 2015.

[118] Yaakov Oshman and Ilan Shaviv. Optimal tuning of a kalman filter using genetic algorithms. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, page 4558, 2000.

[119] Maria M Papathanasiou, Baris Burnak, Justin Katz, Nilay Shah, and Efstratios N Pistikopoulos. Assisting continuous biomanufacturing through advanced control in downstream purification. *Computers & Chemical Engineering*, 125:232–248, 2019.

[120] Olivier Paquet-Durand, Viktoria Zettel, Abdolrahim Yousefi-Darani, and Bernd Hitzmann. The supervision of dough fermentation using image analysis complemented by a continuous discrete extended kalman filter. *Processes*, 8(12):1669, 2020.

[121] Seo-Young Park, Cheol-Hwan Park, Dong-Hyuk Choi, Jong Kwang Hong, and Dong-Yup Lee. Bioprocess digital twins of mammalian cell culture for advanced biomanufacturing. *Current Opinion in Chemical Engineering*, 33:100702, 2021.

[122] Sara Pérez-Vieites and Joaquín Míguez. Nested gaussian filters for recursive bayesian inference and nonlinear tracking in state space models. *Signal Processing*, 189:108295, 2021.

[123] Leander AH Petersen, B Wayne Bequette, Sten B Jørgensen, John Villadsen, Ib Christensen, and Krist V Gernaey. Modeling and system identification of an unconventional bioreactor used for single cell protein production. *Chemical Engineering Journal*, 390:124438, 2020.

[124] Emil Petre and Dan Selişteanu. Model approximation and simulations of a class of nonlinear propagation bioprocesses. *Numerical Analysis-Theory and Application*, pages 211–230, 2011.

[125] José Pinto, Mykaella Mestre, J Ramos, Rafael S Costa, Gerald Striedner, and Rui Oliveira. A general deep hybrid model for bioreactor systems: Combining first principles with deep neural networks. *Computers & Chemical Engineering*, 165:107952, 2022.

[126] Ralf Pörtner and Johannes Möller. *Bioprocess Systems Engineering Applications in Pharmaceutical Manufacturing*. MDPI Books, 2022.

[127] Chris Rackauckas, Mike Innes, Yingbo Ma, Jesse Bettencourt, Lyndon White, and Vaibhav Dixit. Diffeqflux. jl-a julia library for neural differential equations. *arXiv preprint arXiv:1902.02376*, 2019.

[128] Carl Rafferty, Jim O'Mahony, Rosemary Rea, Barbara Burgoyne, Karin M Balss, Olav Lyngberg, Caitlin O'Mahony-Hartnett, Dan Hill, and Eugene Schaefer. Raman spectroscopic based chemometric models to support a dynamic capacitance based cell culture feeding strategy. *Bioprocess and biosystems engineering*, 43:1415–1429, 2020.

[129] Matti Raitoharju and Robert Piché. On computational complexity reduction methods for kalman filter extensions. *IEEE Aerospace and Electronic Systems Magazine*, 34(10):2–19, 2019.

[130] Judit Randek and Carl-Fredrik Mandenius. On-line soft sensing in upstream bioprocessing. *Critical reviews in biotechnology*, 38(1):106–121, 2018.

[131] Anurag S Rathore, Saxena Nikita, Garima Thakur, and Somesh Mishra. Artificial intelligence and machine learning applications in biopharmaceutical manufacturing. *Trends in Biotechnology*, 41(4):497–510, 2023.

[132] Ian Reid and Hilary Term. Estimation ii discrete-time kalman filter. *Hilary Term*, pages 1–44, 2001.

[133] Grand View Research. Biotechnology market size, share & trends analysis report by application (health, agriculture, food, natural resources & environment, industrial processing bioinformatics), by technology, and segment forecasts, 2020 - 2027, 2020.

[134] Sebastian Juan Reyes, Yves Durocher, Phuong Lan Pham, and Olivier Henry. Modern sensor tools and techniques for monitoring, controlling, and improving cell culture processes. *Processes*, 10(2):189, 2022.

[135] Atsushi Sakai and Yoji Kuroda. Discriminative parameter training of unscented kalman filter. *IFAC Proceedings Volumes*, 43(18):677–682, 2010.

[136] Omer San. The digital twin revolution. *Nature Computational Science*, 1(5):307–308, 2021.

[137] Simo Särkkä and Lennart Svensson. *Bayesian filtering and smoothing*, volume 17. Cambridge university press, 2023.

[138] Artur M Schweidtmann, Erik Esche, Asja Fischer, Marius Kloft, Jens-Uwe Repke, Sebastian Sager, and Alexander Mitsos. Machine learning in chemical engineering: A perspective. *Chemie Ingenieur Technik*, 93(12):2029–2039, 2021.

[139] Artur M Schweidtmann, Dongda Zhang, and Moritz von Stosch. A review and perspective on hybrid modelling methodologies. *Digital Chemical Engineering*, page 100136, 2023.

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

[141] Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P Adams, and Nando De Freitas. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2015.

[142] Angira Sharma, Edward Kosasih, Jie Zhang, Alexandra Brintrup, and Anisoara Calinescu. Digital twins: State of the art theory and practice, challenges, and open research questions. *Journal of Industrial Information Integration*, page 100383, 2022.

[143] Dan Simon. *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*. John Wiley & Sons, 2006.

[144] Peter Sinner, Sven Daume, Christoph Herwig, and Julian Kager. *Usage of Digital Twins Along a Typical Process Development Cycle*, pages 71–96. Springer International Publishing, Cham, 2021.

[145] Michael Sokolov, Moritz von Stosch, Harini Narayanan, Fabian Feidl, and Alessandro Butté. Hybrid modeling—a key enabler towards realizing digital twins in biopharma? *Current Opinion in Chemical Engineering*, 34:100715, 2021.

[146] Dimitri Solomatine, Linda M See, and RJ Abrahart. Data-driven modelling: concepts, approaches and experiences. *Practical hydroinformatics*, pages 17–30, 2009.

[147] Ines V Stelzer, Julian Kager, and Christoph Herwig. Comparison of particle filter and extended kalman filter algorithms for monitoring of bioprocesses. In *Computer Aided Chemical Engineering*, volume 40, pages 1483–1488. Elsevier, 2017.

[148] SU Support. Costs of monoclonal antibody production, 2024. Accessed: 2024-05-30.

[149] Arnas Survyla. *Novel soft sensors for bioprocess state estimation.* PhD thesis, Kauno technologijos universitetas., 2024.

[150] Christos Tsigkanos, Stefan Nastic, and Schahram Dustdar. Towards resilient internet of things: Vision, challenges, and research roadmap. In *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, pages 1754–1764. IEEE, 2019.

[151] Apostolos Tsopanoglou and Ioscani Jiménez del Val. 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.

[152] Aditya Tulsyan, Hamid Khodabandehlou, Tony Wang, Gregg Schorner, Myra Coufal, and Cenk Undey. Spectroscopic models for real-time monitoring of cell culture processes using spatiotemporal just-in-time gaussian processes. *AIChE Journal*, 67(5):e17210, 2021.

[153] Aditya Tulsyan, Gregg Schorner, Hamid Khodabandehlou, Tony Wang, Myra Coufal, and Cenk Undey. A machine-learning approach to calibrate generic raman models for real-time monitoring of cell culture processes. *Biotechnology and Bioengineering*, 116(10):2575–2586, 2019.

[154] Ayyarao SLV Tummala and PV Ramanarao. Tuning of extended kalman filter for power systems using two lbest particle swarm optimization, 2017.

[155] Andrea Tuveri. *Bayesian Estimators for Bioprocess Monitoring Under Uncertainty.* PhD thesis, NTNU, 2023.

[156] Isuru A Udugama, Pau C Lopez, Carina L Gargalo, Xueliang Li, Christoph Bayer, and Krist V Gernaey. Digital twin in biomanufacturing: Challenges and opportunities towards its implementation. *Systems Microbiology and Biomanufacturing*, 1:257–274, 2021.

[157] Isuru A Udugama, Merve Öner, Pau C Lopez, Christan Beenfeldt, Christoph Bayer, Jakob K Huusom, Krist V Gernaey, and Gürkan Sin. 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.

[158] Genta Ueno and Nagatomo Nakamura. Bayesian estimation of the observation-error covariance matrix in ensemble-based filters. *Quarterly Journal of the Royal Meteorological Society*, 142(698):2055–2080, 2016.

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

[160] Wil MP van der Aalst, Oliver Hinz, and Christof Weinhardt. Resilient digital twins: organizations need to prepare for the unexpected, 2021.

[161] Eric A Wan and Rudolph Van Der Merwe. 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)*, pages 153–158. Ieee, 2000.

[162] Dongshu Wang, Dapei Tan, and Lei Liu. Particle swarm optimization algorithm: an overview. *Soft computing*, 22:387–408, 2018.

[163] Kedong Wang, Yong Li, and Chris Rizos. Practical approaches to kalman filtering with time-correlated measurement errors. *IEEE Transactions on Aerospace and Electronic Systems*, 48(2):1669–1681, 2012.

[164] Peng Xu. Analytical solution for a hybrid logistic-monod cell growth model in batch and continuous stirred tank reactor culture. *Biotechnology and bioengineering*, 117(3):873–878, 2020.

[165] Bhagya S Yatipanthalawa and Sally L Gras. Predictive models for upstream mammalian cell culture development-a review. *Digital Chemical Engineering*, page 100137, 2023.

[166] Abdolrahim Yousefi-Darani, Olivier Paquet-Durand, and Bernd Hitzmann. The kalman filter for the supervision of cultivation processes. *Digital Twins*, pages 95–125, 2020.

[167] Abdolrahim Yousefi-Darani, Olivier Paquet-Durand, and Bernd Hitzmann. The Kalman Filter for the Supervision of Cultivation Processes. *Advances in biochemical engineering/biotechnology*, 177:95–125, 2020.

[168] Abdolrahimahim Yousefi-Darani, Olivier Paquet-Durand, Jörg Hinrichs, and Bernd Hitzmann. Parameter and state estimation of backers yeast cultivation with a gas sensor array and unscented kalman filter. *Engineering in Life Sciences*, 21(3-4):170–180, 2021.

[169] Abdolrahim Yousefidarani. *Development of software sensors for on-line monitoring of baker's yeast fermentation process*. PhD thesis, Universität Hohenheim, 2021.

[170] Dongda Zhang, Ehecatl Antonio Del Rio-Chanona, Panagiotis Petsagkourakis, and Jonathan Wagner. Hybrid physics-based and data-driven modeling for bioprocess online simulation and optimization. *Biotechnology and bioengineering*, 116(11):2919–2930, 2019.

[171] Beichen Zhao, Xueliang Li, Wanqiang Sun, Juntao Qian, Jin Liu, Minjie Gao, Xin Guan, Zhenwu Ma, and Jianghua Li. Biodt: An integrated digital-twin-based framework for intelligent biomanufacturing. *Processes*, 11(4):1213, 2023.

[172] Robert Zimmerleiter, Julian Kager, Ramin Nikzad-Langerodi, Vladimir Berezhinskiy, Frank Westad, Christoph Herwig, and Markus Brandstetter. Probeless noninvasive near-infrared spectroscopic bioprocess monitoring using microspectrometer technology. *Analytical and Bioanalytical Chemistry*, 412(9):2103–2109, 2020.

[173] Patrick J Zwietering, Emile HL Aarts, and Jaap Wessels. The design and complexity of exact multilayered perceptrons. *International Journal of Neural Systems*, 2(03):185–199, 1991.

APPENDICES

Appendix A

NKE Auto-Tuning

A.1 Properties of Gaussian Distribution

Properties and formulations of Gaussian (normal) distributions and their characteristics, especially in the context of joint, marginal, and conditional distributions.

Definition A.1 (Gaussian distribution)**.** *A random variable* $\mathbf{x} \in \mathbb{R}^n$ *is said to follow a Gaussian distribution with mean* $\mathbf{m} \in \mathbb{R}^n$ *and covariance matrix* $\mathbf{P} \in \mathbb{R}^{n \times n}$ *if its probability density function (PDF) is given by:*

$$N(\mathbf{x} \mid \mathbf{m}, \mathbf{P}) = \frac{1}{(2\pi)^{n/2}|\mathbf{P}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\mathbf{m})^T\mathbf{P}^{-1}(\mathbf{x}-\mathbf{m})\right)$$

where $|\mathbf{P}|$ *is the determinant of covariance matrix* $\mathbf{P}$*.*

The exponential term $\exp\left(-\frac{1}{2}(\mathbf{x}-\mathbf{m})^T\mathbf{P}^{-1}(\mathbf{x}-\mathbf{m})\right)$ *quantifies the probability density based on the distance* $\mathbf{x}$ *from the mean* $\mathbf{m}$*, scaled by the covariance matrix* $\mathbf{P}$*.*

Lemma A.1 (Joint Distribution of Gaussian Variables)**.** *If* $\mathbf{x} \in \mathbb{R}^n$ *and* $\mathbf{y} \in \mathbb{R}^m$ *are Gaussian-distributed variables such that:*

$$\begin{aligned} \mathbf{x} &\sim N(\mathbf{m}, \mathbf{P}) \\ \mathbf{y} \mid \mathbf{x} &\sim N(\mathbf{H}\mathbf{x} + \mathbf{u}, \mathbf{R}) \end{aligned} \tag{A.1}$$

where $\mathbf{H}$ *is a linear transformation matrix,* $\mathbf{u}$ *is a translation vector, and the joint distribution of* $\mathbf{x}$ *and* $\mathbf{y}$*, and the marginal distribution of* $\mathbf{y}$ *are given by*

$$\begin{aligned} \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} \sim N\left(\begin{pmatrix} \mathbf{m} \\ \mathbf{Hm} + \mathbf{u} \end{pmatrix}, \begin{pmatrix} \mathbf{P} & \mathbf{PH}^T \\ \mathbf{HP} & \mathbf{HPH}^T + \mathbf{R} \end{pmatrix} \right) \\ \mathbf{y} \sim N(\mathbf{Hm} + \mathbf{u}, \mathbf{HPH}^T + \mathbf{R}) \end{aligned} \tag{A.2}$$

Lemma A.2 (Conditional Distribution of Gaussian Variables)**.** *If* $\mathbf{x}$ *and* $\mathbf{y}$ *have the joint Gaussian distribution:*

$$\begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} \sim N\left(\begin{pmatrix} \mathbf{a} \\ \mathbf{b} \end{pmatrix}, \begin{pmatrix} \mathbf{A} & \mathbf{C} \\ \mathbf{C}^T & \mathbf{B} \end{pmatrix} \right)$$

then the conditional and marginal distributions are:

$$\begin{aligned} \mathbf{x} &\sim N(\mathbf{a}, \mathbf{A}) \\ \mathbf{y} &\sim N(\mathbf{b}, \mathbf{B}) \end{aligned}$$

Conditional on $\mathbf{y}$*,* $\mathbf{x}$ *has the distribution:*

$$\mathbf{x} \mid \mathbf{y} \sim N(\mathbf{a} + \mathbf{CB}^{-1}(\mathbf{y} - \mathbf{b}), \mathbf{A} - \mathbf{C}\ \mathbf{B}^{-1}\mathbf{C}^T)$$

Conditional on $\mathbf{x}$*,* $\mathbf{y}$ *has the distribution:*

$$\mathbf{y} \mid \mathbf{x} \sim N(\mathbf{b} + \mathbf{C}^T\mathbf{A}^{-1}(\mathbf{x} - \mathbf{a}), \mathbf{B} - \mathbf{C}^T\mathbf{A}^{-1}\mathbf{C})$$

This formulation highlights the relationships and dependencies between $\mathbf{x}$ *and* $\mathbf{y}$ *in the joint Gaussian setting.*

- *Marginal Distributions: The distributions of* $\mathbf{x}$ *and* $\mathbf{y}$ *are considered independent of each other. The marginal distributions were derived from the joint distribution by ignoring the other variables.*
- *Conditional Distributions: These provide the distributions of one variable given specific values of the other variable. The expressions for conditional distributions in Gaussian cases are particularly neat because they also result in Gaussian distributions. This is a special property of Gaussian distributions known as the closure under conditioning.*

- *The term* $\mathbf{CB}^{-1}(\mathbf{y}-\mathbf{b})$ *represents the correction to the mean of* $\mathbf{x}$ *based on the value* $\mathbf{y}$*, adjusted by the covariance between* $\mathbf{x}$ *and* $\mathbf{y}$*. This correction is a measure of how much* $\mathbf{y}$ *informs us of* $\mathbf{x}$*.*

- *The covariance term* $\mathbf{A}-\mathbf{CB}^{-1}\mathbf{C}^T$ *quantifies the uncertainty in* $\mathbf{x}$ *after knowing* $\mathbf{y}$*. It is reduced by the amount of information* $\mathbf{y}$ *provides about* $\mathbf{x}$*, through the covariance* $\mathbf{C}$*.*

- *Conversely,* $\mathbf{C}^T\mathbf{A}^{-1}(\mathbf{x}-\mathbf{a})$ *adjusts* $\mathbf{b}$*, the mean of* $\mathbf{y}$*, based on the observed* $\mathbf{x}$ *value. Similarly,* $\mathbf{B}-\mathbf{C}^T\mathbf{A}^{-1}\mathbf{C}$ *reduces the uncertainty in* $\mathbf{y}$ *based on the information provided by* $\mathbf{x}$*.*

A.2 Theoretical Application of BAT to Estimate all NKEs components

The Figure A.1 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.

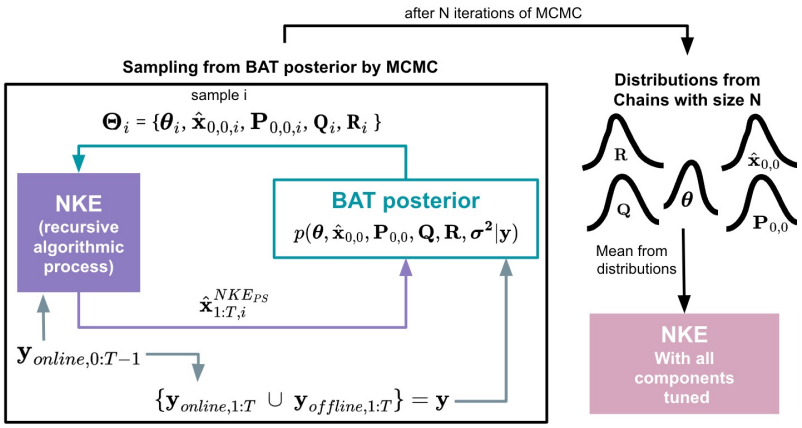

Figure A.1: BAT: Batch Bayesian Auto-Tuning for Nonlinear Kalman Estimators. BAT posterior of $\boldsymbol{\theta}, \hat{\mathbf{x}}_{0,0}, \mathbf{P}_{0,0}, \mathbf{Q}, \mathbf{R}$) given $\mathbf{y}$ is defined outside of NKE recursive algorithm that after complete T steps produce a set of predicted 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 $\mathbf{y}_k = h(\hat{\mathbf{x}}_k^{NKE_{PS}}) + \boldsymbol{\delta}_k$ (Equation 4.6) 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 $\boldsymbol{\theta}, \hat{\mathbf{x}}_{0,0}, \mathbf{P}_{0,0}, \mathbf{Q}, \mathbf{R}$ 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 $\mathbf{y}_{online,1:T}$.

A.2.1 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 4.6 with the specific prediction mechanism of UKF:

$$\mathbf{y}_k = h(\hat{\mathbf{x}}_k^{UKF_{PS}}) + \boldsymbol{\delta}_k, \tag{A.3}$$

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:

$$\langle \hat{\mathbf{x}}_k^{UKF_{PS}}, \mathbf{P}_k^{UKF_{PS}} \rangle = UKF_{PS}(\hat{\mathbf{x}}_{0,k-1} = \hat{\mathbf{x}}_{k-1}^{UKF_{US}}, \mathbf{f}, \boldsymbol{\theta}, t_{k-1}, t_k, \mathbf{Q}_{k-1}, \mathbf{R}_{k-1}, \mathbf{P}_{0,k-1} = \mathbf{P}_{k-1}^{UKF_{US}}), \tag{A.4}$$

where:

- $\mathbf{f}$ is the nonlinear process model function (unstructured mechanistic model).
- $\boldsymbol{\theta}$ 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: $\hat{\mathbf{x}}_k^{UKF_{PS}} = \sum_{i=0}^{2n} W_i^m \mathbf{s}_i'$, $\mathbf{P}_k^{UKF_{PS}} = \sum_{i=0}^{2n} W_i^c (\mathbf{s}_i' - \hat{\mathbf{x}}_k^{UKF_{PS}})(\mathbf{s}_i' - \hat{\mathbf{x}}_k^{UKF_{PS}})^T + \mathbf{Q}$, 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:

$$p(\boldsymbol{\theta}, \boldsymbol{\sigma}^2, \hat{\mathbf{x}}_{0,set}, \mathbf{Q}, \mathbf{R}, \mathbf{P}_{0,set} | \mathbf{y}) \propto p(\boldsymbol{\theta}) \times p(\boldsymbol{\sigma}^2) \times \prod_{k=1}^{T} MVN(\mathbf{y}_k; h(\hat{\mathbf{x}}_k^{UKF_{PS}}), \sum(\boldsymbol{\sigma}^2)) \times p(\hat{\mathbf{x}}_{0,k-1}) \times p(\mathbf{P}_{0,k-1}) \times p(\mathbf{Q}_{k-1}) \times p(\mathbf{R}_{k-1}). \tag{A.5}$$

Pendulum case

Aiming to illustrate the application of BAT for UKF, we explore the Pendulum tracking with UKF presented in [137], which is a classic problem. The main idea is to estimate the UKF parameters based on the simulated data of the pendulum. Table A.1 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 A.1) to the pendulum model and simulated data is shown in A.2. The details about the dynamic model of the pendulum and simulated data can be seen in [137]. Furthermore, the code for this example is available in our repository https://github.com/cristovaoiglesias/BAT.

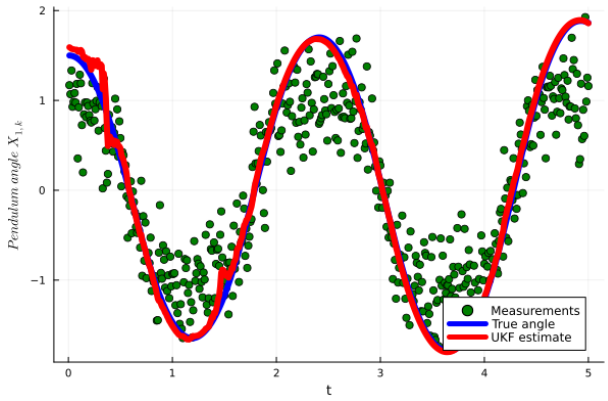

Figure A.2: Simulated pendulum data and the result of tracking The pendulum using the UKF parameters estimated by BAT posterior.

Table A.1: 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

A.2.2 BAT for CKF

Again, let us consider an observation model as defined in Equation 4.6 with the specific prediction mechanism of CKF:

$$\mathbf{y}_k = h(\hat{\mathbf{x}}_k^{CKF_{PS}}) + \boldsymbol{\delta}_k, \tag{A.6}$$

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:

$$\langle \hat{\mathbf{x}}_k^{CKF_{PS}}, \mathbf{P}_k^{CKF_{PS}} \rangle = CKF_{PS}(\hat{\mathbf{x}}_{0,k-1} = \hat{\mathbf{x}}_{k-1}^{CKF_{US}}, \mathbf{f}, \boldsymbol{\theta}, t_{k-1}, t_k, \mathbf{Q}_{k-1}, \mathbf{R}_{k-1}, \mathbf{P}_{0,k-1} = \mathbf{P}_{k-1}^{CKF_{US}}), \tag{A.7}$$

where:

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

$$\mathbf{s}_i = \sqrt{\mathbf{P}_{k-1}}\,\mathbf{e}_i, \quad i = 1, \ldots, 2n \tag{A.8}$$

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:

$$\mathbf{s}'_i = \mathbf{f}(\mathbf{s}_i, \boldsymbol{\theta}, t_k), \quad i = 1, \ldots, 2n \tag{A.9}$$

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

$$\hat{\mathbf{x}}_k^{CKF_{PS}} = \frac{1}{2n} \sum_{i=1}^{2n} \mathbf{s}'_i \tag{A.10}$$

$$\mathbf{P}_k^{CKF_{PS}} = \frac{1}{2n} \sum_{i=1}^{2n} (\mathbf{s}'_i - \hat{\mathbf{x}}_k^{CKF_{PS}})(\mathbf{s}'_i - \hat{\mathbf{x}}_k^{CKF_{PS}})^T + \mathbf{Q} \tag{A.11}$$

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:

$$\begin{aligned} p(\boldsymbol{\theta}, \boldsymbol{\sigma}^2, \hat{\mathbf{x}}_{0,set}, \mathbf{Q}, \mathbf{R}, \mathbf{P}_{0,set} | \mathbf{y}) \propto p(\boldsymbol{\theta}) \times p(\boldsymbol{\sigma}^2) \times \prod_{k=1}^{T} MVN(\mathbf{y}_k; h(\hat{\mathbf{x}}_k^{CKF_{PS}}), \sum(\boldsymbol{\sigma}^2)) \\ \times p(\hat{\mathbf{x}}_{0,k-1}) \times p(\mathbf{P}_{0,k-1}) \times p(\mathbf{Q}_{k-1}) \times p(\mathbf{R}_{k-1}). \end{aligned} \tag{A.12}$$

Pendulum case

Aiming to illustrate the application of BAT for CKF, we explore the same classical problem of subsection A.2.1. The main idea is also to estimate the CKF parameters based on the simulated data of the pendulum. Table A.2 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 [137]. Furthermore, the code of this example is available in our repository https://github.com/cristovaoiglesias/BAT.

Table A.2: Results of sampling the BAT for CKF with simulated data of pendulum.

CKF components	ground truth	BAT mean	BAT std
R	0.1	0.1009	0.0101
g	9.81	9.8097	0.0204

A.3 Baseline methods to auto-tuning NKEs

A.3.1 Objective Function with five metrics

The objective function proposed in [14] is the most recent approach in the SOTA to Estimate **Q** and **R** of EKF, and therefore, it is used in our empirical evaluation. In [14], 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 aim is to keep the problem easy to solve and does not need any extra effort from the filter designer. The objective function proposed in [14] is defined as follows:

$$min_{Q,R} = \omega_1 \mid \frac{\mu(\varepsilon_{y,k})}{m} - 1 \mid + \omega_2 \mid \frac{\sigma^2(\varepsilon_{y,k})}{m} - 1 \mid + \omega_3 \parallel trace(P) \parallel + \omega_4 RMSE(e_y) + \omega_5 \mid \frac{N_{\chi^2_m \geq \alpha}}{0.05 \times \bar{N}} - 1 \mid \tag{A.13}$$

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: $\omega_1 = 0.39999$, $\omega_2 = 0.39999999999999$, $\omega_3 = 1 \times 10^{-14}$, $\omega_4 = 1 \times 10^{-5}$, and $\omega_5 = 0.2$.

A.3.2 Minimizing the residual prediction error (RPE)

The prediction error minimization technique proposed in [1] 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 $\mathbf{M}$

$$\langle \mathbf{R}, \mathbf{Q} \rangle = \arg\min_{\mathbf{R},\ \mathbf{Q}} \sum_{k=0}^{T} (\mathbf{y}_k - h(\boldsymbol{\mu}_t)) \mathbf{M}^{-1} (\mathbf{y}_k - h(\boldsymbol{\mu}_t)) \tag{A.14}$$

In this Equation A.14:

- $\mathbf{R}$ 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 μ_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 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.

Therefore, RPE (Equation A.14) can be adapted to the use of all measurement data available (which motives its use as a baseline), but it requires $\mathbf{M}$ to be known or be the same for all measured state variables which is not possible in the majority of the real-world applications, including our empirical evaluation.

If the covariance matrix $\mathbf{M}$ is not known but is the same for all state variables, assuming it is a multiple of the identity matrix is a common simplification. This is based on the assumption that the noise in the measurements is isotropic, meaning it has the same variance in all directions and there are no correlations between different measurements. Then, in the context of the RPE minimization technique, if the matrix $\mathbf{M}$ is any multiple of the identity matrix, $\mathbf{M} = \sigma^2\mathbf{I}$ where σ^2 is a scalar, then the weighting factor simplifies the minimization problem of Equation A.14 in the following way.

First, substituting $\mathbf{M}$ with $\sigma^2\mathbf{I}$

$$\langle \mathbf{R}, \mathbf{Q} \rangle = \arg\min_{\mathbf{R},\ \mathbf{Q}} \sum_{k=0}^{T} (\mathbf{y}_k - h(\boldsymbol{\mu}_t))^T (\sigma^2\mathbf{I})^{-1} (\mathbf{y}_k - h(\boldsymbol{\mu}_t)),$$

and simplify using properties of identity matrix $(\sigma^2\mathbf{I})^{-1} = \frac{1}{\sigma^2}\mathbf{I}$

$$\langle \mathbf{R}, \mathbf{Q} \rangle = \arg\min_{\mathbf{R},\ \mathbf{Q}} \sum_{k=0}^{T} \frac{1}{\sigma^2} (\mathbf{y}_k - h(\boldsymbol{\mu}_t))^T \mathbf{I} (\mathbf{y}_k - h(\boldsymbol{\mu}_t)).$$

Now, factoring out $\frac{1}{\sigma^2}$

$$\langle \mathbf{R}, \mathbf{Q} \rangle = \arg\min_{\mathbf{R},\ \mathbf{Q}} \frac{1}{\sigma^2} \sum_{k=0}^{T} (\mathbf{y}_k - h(\boldsymbol{\mu}_t))^T (\mathbf{y}_k - h(\boldsymbol{\mu}_t))$$

and simplifying the quadratic term

$$(\mathbf{y}_k - h(\boldsymbol{\mu}_t))^T (\mathbf{y}_k - h(\boldsymbol{\mu}_t)) = \|\mathbf{y}_k - h(\boldsymbol{\mu}_t)\|_2^2 ,$$

we have the following simplified Equation

$$\langle \mathbf{R}, \mathbf{Q} \rangle = \arg\min_{\mathbf{R},\ \mathbf{Q}} \frac{1}{\sigma^2} \sum_{k=0}^{T} \|\mathbf{y}_k - h(\boldsymbol{\mu}_t)\|_2^2 .$$

Since $\frac{1}{\sigma^2}$ is a constant scaling factor and does not affect the location of the minimum, it can be omitted in the optimization process, leading to

$$\langle \mathbf{R}, \mathbf{Q} \rangle = \arg\min_{\mathbf{R},\ \mathbf{Q}} \sum_{k=0}^{T} \|\mathbf{y}_k - h(\mu_t)\|_2^2 \tag{A.15}$$

In our empirical evaluation, such as in the many real-world applications, we do not know $\mathbf{M}$ but we know that it is not the same for all state variables since each state variable have different measurement noises associated (see the description of datasets in Section 4.4.1). Consequently, in our case, the simplification (Equation A.15) is invalid because you have more than one variable with different measurement noises and we cannot generally assume $\mathbf{M}$ is a multiple of the identity matrix. However, if only one measured state variable is present, we can consider $\mathbf{M}$ a multiple of the identity matrix and use Equation A.15. Therefore, in our empirical evaluation, we use the Equation A.14 by considering only the measurements of Xv to estimate $\boldsymbol{\Theta}$ as

$$\langle \boldsymbol{\Theta} \rangle = \arg\min_{\boldsymbol{\Theta}} \sum_{k=0}^{T} \|\mathbf{y}_k - h(\mu_t)\|_2^2 . \tag{A.16}$$

It is important to point out that the Equation A.15 can also be considered an objective function. Therefore, NUTS is the MCMC method used to sample the distribution of $\boldsymbol{\Theta}$ that minimize the Equation A.14 through a probabilistic model. We refer to these approaches as RPE-NUTS, see the next Section for more details about this probabilistic model.

A.3.3 Optimization methods for objective function

BO (Bayesian Optimization): Bayesian Optimization is a powerful method for optimizing expensive black-box functions. It builds a probabilistic model of the objective function, often using Gaussian processes, and uses this model to make decisions about where to sample next. This approach is particularly useful for functions that are expensive

to evaluate, as it aims to find the global optimum with a minimal number of function evaluations by balancing exploration and exploitation [141].

In this study, the BO was used to optimize OF and RPE using the Julia programming language with the package BayesOpt.jl, and the following setup. A standard Gaussian process where the hyperparameters are estimated directly from data using maximum likelihood estimates was used as surrogate model, and kMaternARD5 as kernel function. Maximum a posteriori was used as method for learning the kernel parameters, and cLCBa as selection criteria. Furthermore, a total of 400 iterations were used.

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 [162].

In this study, the PS was used to optimize OF and RPE using the Julia programming language with the package Metaheuristics.jl [28]. The Parameters for PS algorithm: learning rates C1=1.5 and C2=1.5, the population size N=200 and ω=0.7 controls the inertia weight. Furthermore, a total of 4000 iterations were used.

GA (Genetic Algorithm): Genetic Algorithm is a heuristic search and optimization technique inspired by the principles of natural selection and genetics. It operates on a population of potential solutions using operators such as selection, crossover, and mutation to evolve the solutions over generations. GAs are particularly effective for large, complex search spaces where traditional optimization methods struggle, and they are known for their flexibility and robustness in solving diverse optimization problems [86].

In this study, the PS was used to optimize OF and RPE using the Julia programming language with the package Metaheuristics.jl [28]. The Parameters for GA: PolynomialMutation as mutation operator, SBX as crossover operators, and GenerationalReplacement as replacement method. Furthermore, a total of 4000 iterations were used.

A.3.4 Method to sample $\boldsymbol{\Theta}$ that optimizes the objective function

Given that the approaches presented in Section A.3.3 minimize OF (Section A.3.1) and RPE (Section A.3.2). Then, after the minimization process of the OF and RPE where we denote them as ObjF($\boldsymbol{\Theta}$) (since RPE can also be considered an objective function), we have

$$\begin{aligned} \nu_{min} &= min\ ObjF(\boldsymbol{\Theta}) \\ \boldsymbol{\Theta}_{opt} &= arg\ min\ ObjF(\boldsymbol{\Theta}), \end{aligned} \tag{A.17}$$

where $\boldsymbol{\Theta} = \{\mathbf{R}, \mathbf{Q}, \mathbf{P}, \boldsymbol{\theta}, \mathbf{x}_0\}$ are the variables of ObjF (in our study, $\boldsymbol{\Theta}$ represents the NKE components), ν_{min} is the minimum value of an ObjF, and $\boldsymbol{\Theta}_{opt}$ is the specific values of the parameters that yield the minimum value of ObjF. However, besides finding $\boldsymbol{\Theta}_{opt}$, it is important to find the distributions of $\boldsymbol{\Theta}$ related to a minimum ν_{min} or a value ϑ ¡ ν_{min} due to the following reasons:

- **Uncertainty Quantification.** The distributions provide insights into the uncertainty associated with the objective function. Knowing the distributions of $\boldsymbol{\Theta}$ allows us to understand the variability and robustness of the objective function. If the distributions are narrow (i.e., low variance), it indicates that the objective function leads to a robust and stable solution. Conversely, wide distributions (high variance) imply greater uncertainty and variability in these parameters, indicating a less stable solution.
- **Exploring Solution Space**. Analyzing the distributions can give insights about the complexity and landscape of the objective functions. For example, if the distributions are multimodal, it may indicate the presence of multiple local minima in the objective function, suggesting that the optimization problem has a complex landscape, making it harder to find the global minimum. This can be crucial information for further analysis of the objective function or for selecting a new one with a simpler and more well-behaved landscape, with fewer local minima and a clearer path to the global minimum (likely easier to optimize).

The approach for analyzing the distributions of $\boldsymbol{\Theta}$ at the minimum ν_{min} of an objective function is to use a MCMC method to sample from the posterior distributions of the $\boldsymbol{\Theta}$. This is done through a probabilistic model that embeds the objective function as a surrogate for likelihood and the $\boldsymbol{\Theta}$ parameters as prior distributions. The probabilistic model is defined in the following way. Given $\boldsymbol{\vartheta}$ observations related to a target minimum of ObjF, we have the following observational model

$$\boldsymbol{\vartheta} = \nu + \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \xi) \tag{A.18}$$

where $\boldsymbol{\vartheta}$ is a set of N observations $\in \{0, \nu_{min}\}$, ν is a value computed by ObjF($\boldsymbol{\Theta}$) representing a measure of how well the $\boldsymbol{\Theta}$ is to design an EKF, and $\boldsymbol{\epsilon}$ is an additive noise.

Then, under a Gaussian error model, we assume that $\boldsymbol{\vartheta}$ follows a normal distribution, and the likelihood is given by

$$P(\boldsymbol{\vartheta}|\boldsymbol{\Theta}) = \mathcal{N}(\nu, \xi), \tag{A.19}$$

and the posterior density as

$$\begin{aligned} P(\boldsymbol{\Theta}|\boldsymbol{\vartheta}) &\propto P(\boldsymbol{\vartheta}|\boldsymbol{\Theta}) \times P(\boldsymbol{\Theta}) \\ &\propto \mathcal{N}(\boldsymbol{\vartheta}|\nu, \xi) \times P(\boldsymbol{\Theta}). \end{aligned} \tag{A.20}$$

In the context of Bayesian inference, the posterior density (Equation A.20) allows the MCMC methods, such as NUTS, to sample (explore) different $\boldsymbol{\Theta}$ parameter values and form the distributions of $\boldsymbol{\Theta}$ by analyzing $\boldsymbol{\vartheta}$. This provides insights into the robustness, variability, and stability of the EKF design by the objective function.

The priors $P(\boldsymbol{\Theta})$ used in our empirical evaluation (by OF-NUTS and RPE-NUTS) are the same as those used in the BAT approach and are defined in Table 4.2.

A.4 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:

$L_{1,1} \sim$ 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 $\mathbf{Q} = \mathbf{L}\mathbf{L}^T$. 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:

  $L_{1,1} \sim$ 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 $\mathbf{Q}$ is computed as $\mathbf{Q} = \mathbf{L}\mathbf{L}^T$. By construction, this $\mathbf{Q}$ 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, $\mathbf{I}$ 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.