State Estimation and Control of

mAb Production Processes

by

Sandra Ama Obiri

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science
in
Chemical Engineering

Department of Chemical and Materials Engineering

University of Alberta

Abstract

Monoclonal antibodies (mAbs) have become indispensable assets in modern medicine, necessitating advancements in large-scale production to meet the growing market demand. This thesis focuses on two critical aspects of mAb production: state estimation in the upstream process and the control and optimization of the downstream process.

Accurate state estimation is vital for optimizing the mAb production process and reducing costs. Therefore, this work presents guidelines for sensor selection to enhance state estimation accuracy, and illustrates an effective variable selection technique for simultaneous state and parameter estimation in the upstream process. Subsequently, a Moving Horizon Estimation (MHE) framework is developed and applied to three case studies to demonstrate the efficiency of estimating some parameters in addition to the states, with the Root Mean Squared Error (RMSE) serving as the evaluation criterion.

The switching of the downstream capture columns are pivotal for ensuring the continuity of the integrated continuous mAb production process. However, due to the discrete nature of the switching operation, advanced process control algorithms such as economic model predictive control (EMPC) are computationally difficult to implement. To address this issue, computationally-efficient approaches are explored to improve EMPC implementation. The first approach uses a sigmoid function to relax the discrete decision variables into continuous ones, which makes the optimization problem easier to solve. The second approach involves training a ReLU neural network to replace the original nonlinear model, leading to the conversion of the integer nonlinear program (INLP) into an integer linear program (ILP).

This modification facilitates a quicker solution to the optimization problem. Additionally, we explore a reinforcement learning (RL) method, which seeks to identify the most effective policy for addressing the optimization problem. Comparatively, all three techniques are evaluated against the conventional switching approach, which relies on a fixed switching product breakthrough rule. The integration of improved state estimation in the upstream process and optimized control strategies in the downstream process presents a comprehensive framework for enhancing mAb production efficiency.

Preface

The research presented in this thesis was conducted under the guidance of Dr. Jinfeng Liu. The models presented in Chapter 2 are originally taken from the work by Kontoravdi et al. [1], [2] for the upstream process, and Gomis Fons et al. [3] for the downstream process. These models have been adapted and modified for the purpose of this thesis.

Throughout this research, I have been actively involved in simulation design, analysis, and manuscript composition. S. Bo contributed by devising and implementing the reinforcement learning approach in Chapter 4. B. Decardi-Nelson provided valuable insights during the design of the sigmoid approach in Chapter 4, B. Agyeman provided valuable assistance in designing the control work, and contributed to manuscript edits. Furthermore, S. Debnath offered support in the design of the estimation work in Chapter 3.

Dr. Jinfeng Liu has been the supervisory author, actively participating in the concept formulation, manuscript composition, and editing process. His guidance and expertise have been instrumental in shaping the direction and quality of this research.

Together, this collaborative effort has led to significant advancements in our understanding of the subject matter, and we hope this thesis will contribute to the broader scientific community's knowledge and inspire further research in the field.

Acknowledgments

Above all, I express my heartfelt gratitude to the Almighty for providing me with the opportunity to pursue and successfully complete my master's degree at this esteemed university.

I am deeply thankful to my supervisor, Dr. Jinfeng Liu, for his unwavering guidance and support throughout my graduate studies. Dr. Liu's encouragement and patient assistance in overcoming research challenges have been invaluable to me.

My gratitude extends to my fellow colleagues from the Process Systems and Control Engineering group, whose unwavering support has been truly commendable. I want to extend special thanks to Bernard Agyeman, Dr. Benjamin Decardi-Nelson, Sarupa Debnath, and Song Bo for their willingness to help at every step. I am also indebted to my family for their love and support throughout my academic journey.

I must acknowledge the generous financial support from the Natural Sciences and Engineering Research Council (NSERC) of Canada, without which my research endeavors would not have been possible.

In closing, I offer my sincere appreciation to all those who have played a part in my academic journey and contributed to my personal growth and success. Your support and encouragement have been instrumental, and I am truly grateful for the experiences and opportunities I have had during my time at this university.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation and Research Overview

Monoclonal antibodies (mAbs) are monospecific and homogeneous antibodies that are developed in a laboratory to enhance the body's natural immune response against pathogens [6]. Due to their unique properties, they serve as powerful tools in the advancement of therapeutic treatments [7]. Over the years, they have become vital in the field of medicine, and have been applied in the treatment of diverse diseases including cancer and infectious diseases such as COVID-19 [6],[3],[8],[9]. However, the cost of treatments involving mAbs can be significantly high due to the necessity for high doses and the expensive production process [10]. Consequently, optimizing the production of mAbs has become imperative in order to minimize expenses and make these treatments more accessible.

Most of the processes for mAb production rely on batch operation, which involves using a batch or fed-batch bioreactor to provide nutrients for cells in order to facilitate antibody production. This is followed by several batch purification steps [3]. Batch processes, however, present several challenges, including difficulties in scaling up, limited adaptability to meet dynamic market demands, poor yield, and the additional expenses associated with storing and handling intermediate products. As a result of these drawbacks, there has been a growing interest in the continuous production of mAbs which has the potential to eliminate time-consuming and labor-intensive procedures, enhance product quality, and improve overall profitability [3], [11], [12]. The continuous production of mAbs can be divided into two primary stages: the upstream and downstream processes. The upstream process entails antibody production within a perfusion bioreactor, where a continuous supply of nutrients is introduced. The downstream process involves purifying the produced antibodies through a number of chromatography steps [4].

In the context of control, buffer tanks play a critical role by serving as an intermediary step between the continuous operation units of the upstream and downstream processes. They help regulate and manage fluctuations in flow rates, pressure, and other state variables [13]. The application of advanced sensor technologies, such as mass and Raman spectroscopy, allows for online or at-line measurement of cell characteristics, substrate concentrations, and metabolic byproduct concentrations thereby enabling effective control strategies [14], [15]. Feedback control in different scenarios can be achieved using either a proportional integral derivative (PID) controller or a model predictive controller (MPC), depending on the specific requirements. The implementation of the PID controller is relatively simple, widespread and faster in terms of computational time, while MPC offers more advanced capabilities for complex processes and utilizes process models and optimization tools to forecast the response of manipulated variables, albeit at the cost of increased computational time. Despite the remarkable technological advancements in control and automation facilitated by high-performance computing, these developments have primarily occurred outside the pharmaceutical industry, and biomanufacturing processes predominantly rely on classical control systems [12] as evidenced in studies such as [16], where a PID controller was utilized to regulate dissolved oxygen in vaccine production. However, there is a rising interest in implementing advanced process control (APC) strategies, such as model predictive control (MPC), within continuous biomanufacturing [17], [18]. For instance in [19], MPC was formulated to maximize the production of monoclonal antibodies and to showcase its superiority over the utilization of a PID controller.

Through the accurate estimation of state variables, processes can be efficiently monitored and controlled to achieve desired yield and product quality, while minimising resource utilisation. A number of studies have documented the application of the Extended Kalman Filter (EKF) in the field of biotechnological processes for state estimation [20], [21], [22]. While this technique has proven valuable in many applications, it is not without limitations. One challenge that arises when using EKF in highly nonlinear systems such as the one describing the mAb production process is that it may lead to reduced estimation accuracy [23]. Incorporating state constraints in bioprocesses is of utmost importance [24] as it ensures that operations are conducted within predefined boundaries. It enables various factors to be taken into consideration, including the maximum allowable levels of impurities, production capacity, and other safety factors. The utilization of Moving Horizon Estimation (MHE) presents notable advantages compared to EKF, particularly in its capacity to effectively handle nonlinear dynamics and incorporate constraints [25]. These advantages serve as a strong motivation for employing MHE in the context of this paper.

Achieving accurate state estimation also relies on the concept of observability. By placing the sensors strategically, it is possible to obtain measurements from which we can extract sufficient information to reconstruct the entire state information of a system. In the work by Liu et al. [26], a sensitivity-based approach was introduced for determining the optimal sensor combination for state estimation problems. The algorithm seeks to determine an optimal sensor subset that ensures that the entire state information can be reconstructed. The approach was applied to a wastewater treatment plant and a four continuous stirred-tank reactor to assess its efficiency. The algorithm involves several steps, including the construction of a sensitivity matrix, the use of successive orthogonalization, and the application of the degree of observability to identify the optimal sensor subset that satisfies the full rank condition.

Given that MHE is a model-based technique, enhancing the accuracy of the model is important and parameter estimation is one way to enhance model accuracy. In [27], a particle swarm optimization (PSO) algorithm was developed and used for parameter estimation in the mAb production process. One drawback of this algorithm is its associated high search complexity. As the search space increases, the computational effort required to find the optimal solution also grows, impacting the efficiency of the algorithm [28], [29]. A common approach is to estimate the sates and parameters simultaneously. It often involves augmenting the parameters as additional states of the system. However, this augmentation can introduce challenges, as it may lead to a loss of observability in the system, even with the use of the optimally-selected sensors. The augmented system may have more variables than can be accurately estimated, making it necessary to select a subset of variables for estimation. In [30], an algorithm for variable selection in augmented systems is proposed. It identifies the subset of variables that can be accurately estimated by effectively extracting information from the measured output. The selected variables are then estimated using MHE.

In Chapter 3 of this work, the objective is to estimate the states of the upstream process. To achieve this, we begin by addressing sensor placement, aiming to determine the minimum sensor combination and their optimal locations. Additionally, we focus on estimating a select set of parameters within the model to maximize the information extracted from the available measurements. Since the evaluation of each possible sensor combination would not be practical for a large-scale system such as the system under consideration, the sensor selection algorithm developed in the work by Liu et al. [26] is applied. Furthermore, simultaneous state and parameter estimation is performed by first augmenting the parameters of the system as additional states, and employing the variable selection technique used in [30] to select the variables to be estimated.

Chapter 4 focuses on optimizing the capture step of the downstream process, which is essential to achieving the integrated, continuous manufacturing of mAbs. Usually, two or more chromatography columns are employed for this step. For instance, in the work by Steinebach et al. [31], two columns were used to achieve the sequential countercurrent loading of mAbs for maximum resin utilization. Other studies have also utilized multiple columns for the same purpose, employing techniques such as simulated moving bed chromatography (SMB), sequential multi-column chromatography (SMCC), and periodic counter-current chromatography (PCC) [32], [33], [34]. In this work, we focus on the use of twin chromatography capture columns, as described by Gomis et al. [3]. The capture columns are operated such that while one column is loaded with product from upstream, the contents of the second column are eluted into the subsequent downstream component. The columns are switched when the loading column reaches a fixed predetermined percentage product breakthrough at the column outlet. However, to ensure maximum product yield and effective resource utilization, the optimization and control of the switching operation is crucial.

The downstream model used in this work involves complex nonlinear equations. It also involves spatial discretization which leads to a large number of states. Furthermore, due to the discrete nature of the switching operation, the optimization problem is formulated as a nonlinear integer problem. This poses an extra challenge for the implementation of EMPC because an integer nonlinear program (INLP) needs to be solved online at each sampling time, which is computationally-demanding and slow. To overcome these difficulties, this work investigates and proposes computationally-efficient approaches to improve EMPC implementation for controlling and optimizing the switching step.

In the first approach, a sigmoid activation function is used to relax the discrete decision variables into continuous decision variables, making the optimization problem easier to solve. The second approach involves training a ReLU neural network to replace the original nonlinear model, leading to the conversion of the INLP into an integer linear program (ILP). This modification facilitates a quicker solution to the optimization problem. Additionally, this work explores an RL method, which seeks to identify the most effective policy for addressing the optimization problem. Comparatively, all three techniques are evaluated against the conventional switching approach, which relies on a fixed switching product breakthrough rule. The assessment centers around product loss and the overall cost incurred, providing insights into their relative performance.

1.2 Thesis Outline and Contributions

The outline and contributions of the thesis is organized as follows:

In Chapter 2, the upstream and downstream models for the mAb production process are presented.

Chapter 3 presents a detailed consideration of sensor placement for continuous mAb production processes. The findings offer valuable guidelines for selecting suitable sensors in similar continuous mAb production setups. Additionally, a state estimation framework using MHE is introduced for continuous mAb production processes. This framework effectively extracts maximum information from measurements to improve state estimation accuracy.

In Chapter 4, the focus shifts to optimization strategies for enhancing the switching operation of the capture columns in the downstream process. Extensive simulation results are presented, comparing different approaches in terms of product loss and cost. The effect of various factors such as the weights used in the optimization problem design, prediction horizon, and noise are also investigated. The aim is to identify the most efficient and cost-effective methods for controlling the switching operation in the integrated continuous production system.

The collective efforts presented in this thesis contribute to the advancement of mAb production processes, both in terms of improving sensor placement for better state estimation and optimizing the downstream operation for increased efficiency.

Chapter 2

Mathematical Models for Upstream and Downstream Processes

This chapter presents the upstream and downstream models of the monoclonal antibody (mAb) production process. The upstream model is originally sourced from the work of Kontoravdi et al. [1], [2], while the downstream model is derived from the work of Gomis Fons et al. [3]. The models were adapted and modified for the purpose of this thesis.

2.1 Upstream System Description and Problem Formulation

The diagram presented in Figure 2.1 illustrates the upstream system of a continuous mAb production process, and is a modified version from [4]. The system primarily consists of a well-stirred bioreactor and a microfiltration unit/cell retention device. The buffer tank, introduced for smoother operation [35], is located between the upstream and downstream process components and is considered in this work alongside the upstream components.

In this setup, the bioreactor employs perfusion culture, where nutrients such as glucose and glutamine are continuously supplied with a flow rate represented by F_{in}, to create an

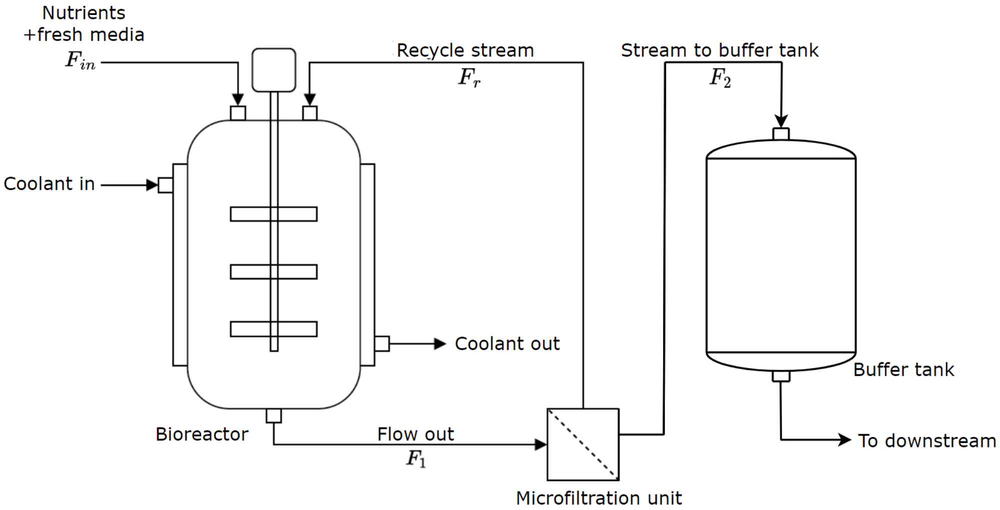

Figure 2.1: A schematic diagram of the upstream process for mAb production.

optimal environment for mAb production by a mammalian cell culture in the bioreactor. Specifically, the GS-NSO cell culture system as described in [36] is considered here. During the production of the desired mAbs, the cell culture undergoes metabolic activities which lead to the production of the key metabolites: lactate and ammonia.

The effluent stream, comprising the cells, mAbs, and metabolites exits the bioreactor at a flow rate denoted as F_1, into the microfiltration unit. The microfiltration unit recycles the cells back into the bioreactor through the recycle stream with a flow rate of F_r. This recycling process ensures the maintenance of a high cell density within the bioreactor. The remaining content of the stream is allowed to pass on into the buffer tank with a flow rate denoted as F_2. A cooling jacket is used around the bioreactor to regulate the temperature of the bioreactor components [4]. Finally, the stream from the buffer tank, containing the product, metabolites, and any toxic by-products is directed downstream for product purification. The mathematical model of the upstream process is based on the work by [1], [2], [4].

2.1.1 Bioreactor Modelling

The bioreactor model encompasses three key aspects: cell growth and death, cell metabolism, and mAb production. To describe cell growth and death, as well as cell metabolism, a Monod kinetics model from [36] was employed, and the dynamic model for the upstream system was developed under the following assumptions:

1. The content within the bioreactor is homogeneously mixed.
2. The enthalpy change resulting from cell death is negligible.
3. The dilution effect is negligible.
4. No heat is lost to the external environment.
5. The temperature of the recycled stream and the reaction mixture are equivalent.
6. The buffer tank level, volume of the bioreactor, and volume of the cell retention device remain constant throughout the process.

To describe the growth of cells, Equations (2.1) - (2.2) are used [36]:

$$\frac{dX_{v1}}{dt} = \mu X_{v1} - \mu_d X_{v1} - \frac{F_{in}}{V_1} X_{v1} + \frac{F_r}{V_1}(X_{vr} - X_{v1}) \tag{2.1}$$

$$\frac{dX_{t1}}{dt} = \mu X_{v1} - \frac{F_{in}}{V_1} X_{t1} + \frac{F_r}{V_1}(X_{tr} - X_{t1}) \tag{2.2}$$

They represent the conversion of viable cells and total cells within the culture [4], where X is the cell concentration in $cells/L$, μ is the specific growth rate in min^{-1}, and μ_d is the specific death rate in min^{-1}. The subscripts v and t signify viable and total cells, and the subscript 1 represents the bioreactor throughout the paper. V_1 represents the bioreactor volume in L, F_{in} is the volumetric flow rate of nutrients into the bioreactor, F_r is volumetric flow rate of the recycle stream, and F_{out} is the volumetric flow rate out of the bioreactor in L/min.

The specific cell growth rate depends on temperature, the concentrations of glucose and glutamine, as well as the metabolites lactate and ammonia. It can be calculated using the following Monod kinetic equations [36]:

$$\mu = \mu_{max} f_{lim} f_{inh} \tag{2.3}$$

$$f_{lim} = \left(\frac{[GLC]_1}{K_{glc} + [GLC]_1}\right)\left(\frac{[GLN]_1}{K_{gln} + [GLN]_1}\right) \tag{2.4}$$

$$f_{inh} = \left(\frac{KI_{lac}}{KI_{lac} + [LAC]_1}\right)\left(\frac{KI_{amn}}{KI_{amn} + [AMN]_1}\right) \tag{2.5}$$

where μ_{max} is the maximum specific growth rate in min^{-1}, and f_{lim} and f_{inh} are the nutrient limitation function and the product inhibition function respectively. The concentrations of glucose, glutamine, lactate, and ammonia are represented by $[GLC]$, $[GLN]$, $[LAC]$, and $[AMM]$ respectively, and measured in mM. The Monod constant in mM for glucose, glutamine, lactate and ammonia are also denoted as K_{glc}, K_{gln}, KI_{lac} and KI_{amm} respectively. The specific death rate is given by Equation (2.6), under the assumption that cell death is only a function of the accumulation of ammonia, and n is assumed to be greater than 1 to give a sharper increase in specific death of cells, with increasing ammonia concentration. The maximum cell death rate, $\mu_{d,\text{max}}$, was also fixed as a constant.

$$\mu_d = \frac{\mu_{d,max}}{1 + \left(\frac{K_{d,amm}}{[AMM]_1}\right)^n};\ n > 1 \tag{2.6}$$

$$\mu_{\text{max}} = 0.0016T - 0.0308 \tag{2.7}$$

In [4], a linear regression analysis was performed utilizing data sourced from [37] to establish a linear correlation between temperature and the maximum cell growth rate. This relationship is expressed by Equation (2.7), where T denotes the temperature of the bioreactor contents in $°C$. The dataset used for this analysis was obtained at the temperatures of $33°C$ and $37°C$.

The heat balance equation for the bioreactor is given by [4]:

$$\frac{dT}{dt} = \frac{F_{in}}{V_1}(T_{in} - T) + \frac{-\Delta H}{\rho c_p}(\mu X_{v1}) + \frac{U}{V_1 \rho c_p}(T_c - T) \tag{2.8}$$

where T_{in} is the temperature of media entering the bioreactor in $°C$, ΔH is the heat of reaction due to cell growth in J/mol, ρ is the density of the reaction mixture in g/L, c_p is the specific heat capacity of the reaction in $J/(g°C)$, U is the overall heat transfer coefficient in $J/(hr°C)$, and T_c is the temperature of coolant in the cooling jacket in °C. The first term in Equation (2.8) accounts for the heat transfer resulting from the inflow of the feed into the bioreactor. It takes into consideration the temperature difference between the incoming feed and the current temperature of the bioreactor contents. The second term represents the heat consumption due to cell growth, and the last term represents the external heat transfer due to the cooling jacket.

As the cells in the bioreactor consume nutrients, undergo growth, and engage in metabolic activities, they generate two main metabolites, namely, lactate and ammonia. The mass balance equations for the nutrients, glucose and glutamine, in the bioreactor can be described using Equations (2.9) - (2.13):

$$\frac{d[GLC]_1}{dt} = -Q_{glc}X_{v1} + \frac{F_{in}}{V_1}([GLC]_{in} - [GLC]_1) + \frac{F_r}{V_1}([GLC]_r - [GLC]_1) \tag{2.9}$$

$$Q_{glc} = \frac{\mu}{Y_{X,glc}} + m_{glc} \tag{2.10}$$

$$\begin{aligned} \frac{d[GLN]_1}{dt} &= -Q_{gln}X_{v1} - K_{d,gln}[GLN]_1 + \frac{F_{in}}{V_1}([GLN]_{in} - [GLN]_1) \\ &\qquad - \frac{F_r}{V_1}([GLC]_1 - [GLN]_1) \end{aligned} \tag{2.11}$$

$$Q_{gln} = \frac{\mu}{Y_{X,gln}} + m_{gln} \tag{2.12}$$

$$m_{gln} = \frac{\alpha_1[GLN]_1}{\alpha_2 + [GLN]_1} \tag{2.13}$$

where $Y_{X,glc}$ and $Y_{X,gln}$ are the yield of cells on glucose and glutamine respectively in $cell/mmol$, $K_{d,gln}$ is the constant for glutamine degradation in min^{-1}, and m_{glc} is the maintenance coefficient of glucose. α_1 and α_2 are the constants of glutamine maintenance coefficient. The mass balance equations for lactate and ammonia in the bioreactor can also be described using Equations (2.14) - (2.17):

$$\frac{d[LAC]_1}{dt} = Q_{lac}X_{v1} - \frac{F_{in}}{V_1}[LAC]_1 + \frac{F_r}{V_1}([LAC]_r - [LAC]_1) \tag{2.14}$$

$$Q_{lac} = Y_{lac,glc}Q_{glc} \tag{2.15}$$

$$\frac{d[AMM]_1}{dt} = Q_{amm}X_{v1} + K_{d,gln}[GLN]_1 - \frac{F_{in}}{V_1}[AMM]_1 + \frac{F_r}{V_1}([AMM]_r - [AMM]_1) \tag{2.16}$$

$$Q_{amm} = Y_{amm,gln}Q_{gln} \tag{2.17}$$

where $Y_{lac,glc}$ and $Y_{amm,gln}$ are the yield of lactate from glucose and the yield of ammonia from glutamine respectively in $mmol/mmol$.

The production rate of the mAbs in the bioreactor is described using Equations (2.18) - (2.19):

$$\frac{d[mAb]_1}{dt} = X_{v1}Q_{mAb} - \frac{F_{in}}{V_1}[mAb]_1 + \frac{F_r}{V_1}([mAb]_r - [mAb]_1) \tag{2.18}$$

$$Q_{mAb} = Q_{mAb}^{max} exp\left[-\frac{1}{2}\left(\frac{pH - pH_{opt}}{w_{mAb}}\right)^2\right] \tag{2.19}$$

where Q_{mAb}^{max} denotes the maximum specific productivity measured in $mg/cell/h$. w_{mAb} denotes the pH-dependent productivity constant, and pH_{opt} denotes the optimal culture pH as shown in [38]. The pH value in Equation (2.19) is assumed to be a function of state, and its model is given by Equations (2.20) - (2.21):

$$pH = \theta_1 - log_{10}(\theta_2[AMM] + \theta_3) \tag{2.20}$$

$$pH = 7.1697 - log_{10}(0.074028[AMM] + 0.968385) \tag{2.21}$$

where the constants θ_1 and θ_2 were obtained from nonlinear regression [4].

2.1.2 Microfiltration

As the stream leaving the bioreactor enters the cell retention device, the cells are separated from the remaining contents of the stream. It is assumed that in the separation process no reactions occur. The concentration of each variable in the recycle stream is given as follows [4]:

$$X_{vr} = \eta_{rec} X_{v1} \frac{F_1}{F_r} \tag{2.22}$$

$$X_{tr} = \eta_{rec} X_{t1} \frac{F_1}{F_r} \tag{2.23}$$

$$[GLC]_r = \eta_{ret} [GLC]_1 \frac{F_1}{F_r} \tag{2.24}$$

$$[GLN]_r = \eta_{ret} [GLN]_1 \frac{F_1}{F_r} \tag{2.25}$$

$$[LAC]_r = \eta_{ret} [LAC]_1 \frac{F_1}{F_r} \tag{2.26}$$

$$[AMM]_r = \eta_{ret} [AMM]_1 \frac{F_1}{F_r} \tag{2.27}$$

$$[mAb]_r = \eta_{ret} [mAb]_1 \frac{F_1}{F_r} \tag{2.28}$$

where (η_{rec}) is the cell recycle rate and (η_{ret}) is the retention rate of glucose, glutamine, lactate, ammonia, and mAb [39]. The mass balance for glucose, glutamine, lactate, ammonia, and mAb concentrations around the cell retention device can be expressed by Equations (2.29) - (2.35):

$$\frac{dX_{v2}}{dt} = \frac{F_1}{V_2}(X_{v1} - X_{v2}) - \frac{F_r}{V_2}(X_{vr} - X_{v2}) \tag{2.29}$$

$$\frac{dX_{t2}}{dt} = \frac{F_1}{V_2}(X_{t1} - X_{t2}) - \frac{F_r}{V_2}(X_{tr} - X_{t2}) \tag{2.30}$$

$$\frac{d[GLC]_2}{dt} = \frac{F_1}{V_2}([GLC]_1 - [GLC]_2) - \frac{F_r}{V_2}([GLC]_r - [GLC]_2) \tag{2.31}$$

$$\frac{d[GLN]_2}{dt} = \frac{F_1}{V_2}([GLN]_1 - [GLN]_2) - \frac{F_r}{V_2}([GLN]_r - [GLN]_2) \tag{2.32}$$

$$\frac{d[LAC]_2}{dt} = \frac{F_1}{V_2}([LAC]_1 - [LAC]_2) - \frac{F_r}{V_2}([LAC]_r - [LAC]_2) \tag{2.33}$$

$$\frac{d[AMM]_2}{dt} = \frac{F_1}{V_2}([AMM]_1 - [AMM]_2) - \frac{F_r}{V_2}([AMM]_r - [AMM]_2) \tag{2.34}$$

$$\frac{d[mAb]_2}{dt} = \frac{F_1}{V_2}([mAb]_1 - [mAb]_2) - \frac{F_r}{V_2}([mAb]_r - [mAb]_2) \tag{2.35}$$

where the subscript 2 denotes the cell retention device. It is assumed that the volume of the contents in the cell retention device, V_2, is constant. Additionally, X_{v2} and X_{t2} are the concentrations of viable cells and total cells respectively in the cell retention device.

2.1.3 Buffer Tank Model

Following the retention of cells by the microfiltration unit, the remaining contents of the stream, which has a high mAb concentration, flow into the buffer tank. The rate of change of mAb concentration in the buffer tank can be described by Equations (2.36) - (2.37):

$$\frac{dc}{dt} = \frac{F_{in,bf}}{(A_c h)([c_{in,bf}] - [c])} \tag{2.36}$$

$$A_c = \pi(\frac{D}{2})^2 \tag{2.37}$$

where $F_{in,bf}$ represents the inlet flow rate of the buffer tank in L/min, h represents the buffer tank level which is assumed to be constant, and $[c_{in,bf}]$ represents the inlet concentration of mAb in mg/L. $[c]$ denotes the concentration of mAb in the buffer tank in mg/L, and D represents the diameter of the buffer tank which is fixed.

The parameters used in the model described by Equations (2.1) - (2.37) are listed in Table 2.1. The values are obtained from literature or calculated by steady-state design. For instance, the constant buffer tank level used, as well as the fixed volumes of the bioreactor and cell retention device were obtained at steady-state operation. Furthermore, the parameters used in the equations describing mAb synthesis, cell growth, death and metabolism in the bioreactor were obtained from the works by Papathanasiou et al. [36] and Villiger et al. [38]. The cell recycle rate and the retention rates of the mAbs, nutrients and metabolites around the cell retention device were also obtained from the work by Clincke et al. [39]. Table 2.1 also provides a list of the inputs utilized in the simulations conducted in the subsequent sections.

Table 2.1: Inputs under nominal conditions and parameters of the model [4].

Parameter	Unit	Value	Parameter	Unit	Value
$K_{d,amm}$	mM	1.76	$-\Delta H$	J/mol	5.0×10^5
$K_{d,gln}$	hr^{-1}	$(9.6 \times 10^{-3})/60$	ρ	g/L	1560.0
K_{glc}	mM	0.75	c_p	$J/(g°C)$	1.244
K_{gln}	mM	0.075	U	$J/(h°C)$	4×10^2
KI_{amm}	mM	28.48	T_{in}	$°C$	37.0
KI_{lac}	mM	171.76	m_{glc}	$mmol/cell/min$	$4.9 \times 10^{-14/60}$
n	$-$	2	α_1	$mML/cell/min$	$3.4 \times 10^{13/60}$
$Y_{amm,gln}$	$mmol/mmol$	0.45	η_{rec}	-	92%
$Y_{lac,glc}$	$mmol/mmol$	2.0	η_{ret}	-	20%
$Y_{X,glc}$	$cell/mmol$	2.6×10^8	V_1	L	3×10^3
$Y_{X,gln}$	$cell/mmol$	8.0×10^8	V_2	L	2.976×10^3
α_2	mM	4.0	h	dm	15
$\mu_{d,max}$	min^{-1}	0.06/60	D	dm	5
Q_{mAb}^{max}	$mg/(cell \cdot hr)$	$(6.59 \times 10^{-10})/60$			
Input	Unit	Value	Input	Unit	Value
F_{in}	L/min	2.161×10^1	GLC_{in}	mM	2.199×10^{-3}
F_1	L/min	2.161×10^1	GLN_{in}	mM	1.306×10^1
F_r	L/min	5.501×10^{-3}	T_c	$°C$	4.069×10^1
F_2	L/min	2.161×10^1	F_{outbf}	L/min	2.161×10^1

2.1.4 Problem Formulation

The continuous-time model in Section 2.1 is discretized using the fourth-order Runge-Kutta method with a sampling time of $\Delta = 60$ minutes, and represented by Equations (2.38) - (2.39) for simplicity.

$$x(k+1) = F(x(k), u(k), P(k)) + w_x(k) \tag{2.38}$$

$$y(k) = H(x(k)) + v(k) \tag{2.39}$$

In Equation (2.38), the system state vector $x(k) \in \mathbb{X} \subset \mathbb{R}^{16}$ comprises 16 elements, representing the state variables listed in Table 2.2. The input vector $u(k) \in \mathbb{U} \subset \mathbb{R}^8$ consists of 8 elements and represents the inputs to the system. The parameter vector $P(k) \in \mathbb{P} \subset \mathbb{R}^{27}$ has 27 elements, the process noise vector $w_x(k) \in \mathbb{W} \subset \mathbb{R}^{16}$ accounts for disturbances in

Table 2.2: States for the upstream process model.

State	Unit	Definition
Xv_1	$cell/L$	concentration of viable cells in bioreactor
X_{t1}	$cell/L$	total concentration of cells in bioreactor
GLC_1	mM	glucose concentration in bioreactor
GLN_1	mM	glutamine concentration in bioreactor
LAC_1	mM	lactate concentration in bioreactor
AMM_1	mM	ammonia concentration in bioreactor
mAb_1	mg/L	mAb concentration in bioreactor
Xv_2	$cell/L$	concentration of viable cells in cell separator
Xt_2	$cell/L$	total concentration of cells in cell separator
GLC_2	mM	glucose concentration in cell separator
GLN_2	mM	glutamine concentration in cell separator
LAC_2	mM	lactate concentration in cell separator
AMM_2	mM	ammonia concentration in cell separator
mAb_2	mg/L	mAb concentration in cell separator
T	$°C$	temperature of bioreactor mixture
c	mg/L	mAb concentration in buffer tank

the system, and the nonlinear state equation is denoted by F. Equation (2.39) describes the measured output vector $y(k) \in \mathbb{Y} \subset \mathbb{R}^{N_y}$, which comprises N_y measurements obtained through the sensor selection process. The measurement noise vector $v(k) \in \mathbb{V} \subset \mathbb{R}^{N_y}$ represents the errors associated with the measurement process, and H represents the output equation that relates the system states to the measured outputs.

The main objective of this work is to estimate the states of the system described in Equations (2.38) - (2.39). To achieve this, the sensor placement problem is first considered, with the aim of determining the minimum number of sensors for which the system is observable, as well as the optimal sensor locations. The following section provides a comprehensive overview of the selection process for the optimal set of sensors for state estimation. We present the algorithm used for sensor selection and reveal the sensors that are ultimately chosen.

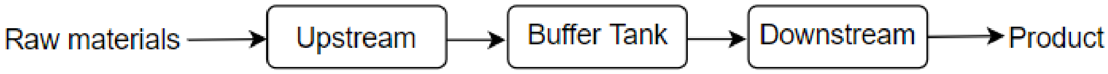

Figure 2.2: High-level flow diagram of mAb production.

2.2 Downstream System Description and Problem Formulation

This section provides a concise overview of the downstream production process, with a focus on the operation of the capture column.

2.2.1 Overview of Production Process

The continuous integrated production of mAbs comprises the upstream and downstream processes. As depicted in Figure 2.2, a buffer tank is used between the two processes to ensure that disruptions in upstream operations do not significantly affect downstream operations [3]. While the upstream process is responsible for mAb production, the downstream process is responsible for the purification of the produced mAbs leaving the upstream process components. The downstream process involves the following steps: capture, virus inactivation, cation exchange chromatography (CEX) in bind-and-elute mode and anion exchange chromatography (AEX) in flow-through mode. The downstream process configuration used in this work is taken from the studies by Gomis Fons et al. [3].

The capture step is a chromatography step and is discontinuous by nature. Therefore, a set of twin protein A chromatography columns are used to achieve continuity. This step is implemented such that while one of the twin columns (column A) is in loading mode, the other column (column B) is in elution mode as depicted in Figure 2.3 and the rest of the downstream processes are carried out [3]. This means that by the time column A is fully loaded, column B is empty of product and has undergone regeneration. The two columns then switch roles so that column A will be connected to the rest of the downstream

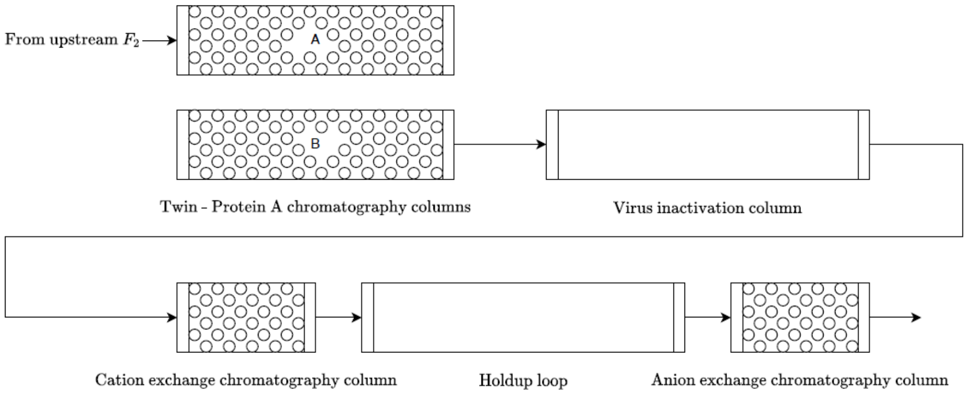
From upstream F_2
A
B
Twin - Protein A chromatography columns
Virus inactivation column
Cation exchange chromatography column
Holdup loop
Anion exchange chromatography column

Figure 2.3: Schematic diagram of the downstream process [4].

units and its contents eluted into the VI-loop, whereas column B will be connected to the upstream units and loaded [4]. Any virus present in the stream are inactivated during the virus inactivation step to prevent further mAb degradation. The CEX and AEX steps effectively remove other undesirable components or contaminants from the stream, ensuring the collection of a purified final product.

The twin chromatography columns used for the capture step are packed with protein A resins or beads, which have an affinity for the mAbs.As the product-saturated stream from upstream enters the capture column, the mAbs bind to the beads until the beads become fully saturated, preventing further adsorption. During elution mode, a buffer is employed to wash out (elute) the adsorbed mAbs from the capture column into the VI loop. In Figure 2.4, the red line represents the loading mode of the capture column and the blue line represents the elution mode of the capture column. Column 1 and column 2 represent the twin chromatography columns employed in the capture step. The switching of these columns is facilitated by two versatile valves, V_1 and V_2. As depicted in the diagram, failing to switch the columns frequently enough results in excess product from upstream being wasted. Conversely, switching too frequently may not allow sufficient time for mAb adsorption onto the beads. Hence, it is crucial to determine the optimal time for column switching to achieve the best performance.

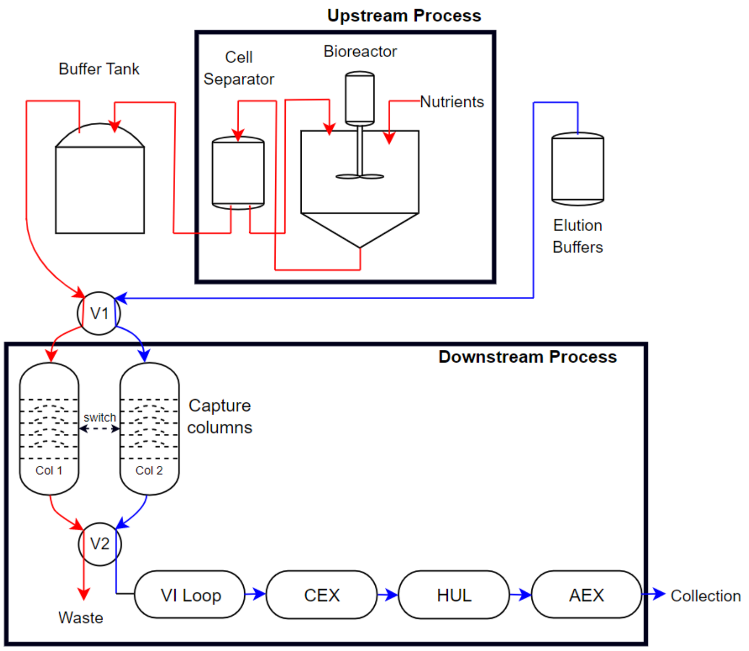

Figure 2.4: Schematic diagram depicting the switching of capture column [3].

2.2.1.1 Capture Column - Loading Mode

To understand the modelling of the capture process, we refer to the capture column as depicted in Figure 2.5. From the figure, the capture column is shown to be packed with protein A resins. The porous nature of the resins provides a large surface area for the binding of the antibodies.

The modeling of the capture step considers three key types of mass transfers within the column. The first is the convection flow resulting from the bulk movement of the fluid through the column in the axial direction. The second is the dispersion of mAb along the axial direction of the column as shown in the second subfigure. The third is the intra-particle diffusion within the beads [4]. The general rate model (GRM) by Perez-Almodovar and Carta [40] is used to describe the loading of the column using the following equation, which describes the mass transfer along the axial coordinate of the column:

$$\frac{\partial c}{\partial t} = D_{ax}\frac{\partial^2 c}{\partial z^2} - \frac{v}{\epsilon_c}\frac{\partial c}{\partial z} - \frac{1-\epsilon_c}{\epsilon_c}\frac{3}{r_p}k_f(c - c_p|_{r=r_p}) \tag{2.40}$$

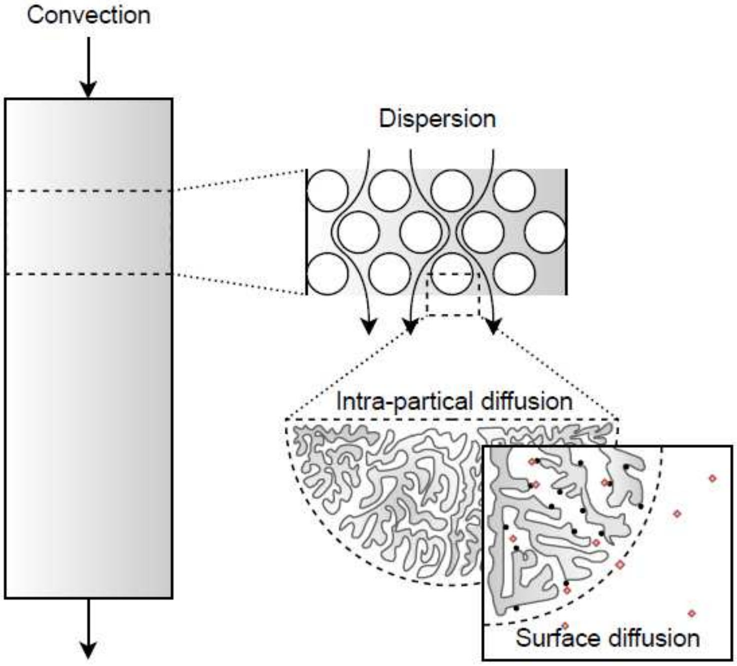

Figure 2.5: Schematic diagram of the capture column [5].

In Equation (2.40), z represents the axial coordinate of the column. c represents the concentration of mAb in the mobile phase which changes with time t, D_{ax} is the axial dispersion coefficient and v is the superficial fluid velocity. ϵ_c denotes the extra-particle column void, k_f is the mass transfer coefficient and r_p is the radius of the porous particles. Furthermore, the term $\frac{\partial^2 c}{\partial z^2}$ describes the movement of mAb through the column due to the concentration difference within the column. The term $\frac{\partial c}{\partial z}$ also represents the change in concentration of mAb along the z axis due to convection flow. The last term, $k_f(c - c_p|_{r=r_p})$, is used to describe the mass transfer between the mobile phase c and the surface of the beads $c_p|_{r=r_p}$. It should be noted that the general rate model is used under the assumption that the transfer along the radial direction of the column is negligible and the transfer along the axial direction of the column and the radial direction in the beads is considered [4]. The following equations are the boundary conditions for Equation (2.40):

$$\frac{\partial c}{\partial z} = \frac{v}{\epsilon_c D_{ax}}(c - c_F) \quad \text{at} \quad z = 0 \tag{2.41a}$$

$$\frac{\partial c}{\partial z} = 0 \quad \text{at} \quad z = L \tag{2.41b}$$

where c_F denotes the harvest mAb concentration from upstream. The mass balance for the diffusion of mAb inside the beads is represented by the following equation:

$$\frac{\partial c_p}{\partial t} = D_{eff}\frac{1}{r^2}\frac{\partial}{\partial r}(r^2\frac{\partial c_p}{\partial r}) - \frac{1}{\epsilon_p}\frac{\partial(q_1 + q_2)}{\partial t} \tag{2.42}$$

where c_p is the concentration of mAb along the radial coordinate of the beads, D_{eff} is the effective pore diffusivity, r is the distance from the current location to the center of the particle and q_1 and q_2 are the adsorbed mAb concentrations for the slow and fast-binding sites of the beads respectively. The boundary conditions for Equation (2.42) are given as:

$$\frac{\partial c_p}{\partial r} = 0 \quad \text{at} \quad r = 0 \tag{2.43a}$$

$$\frac{\partial c_p}{\partial r} = \frac{k_f}{D_{eff}}(c - cp) \quad \text{at} \quad r = r_p \tag{2.43b}$$

and the rates of change of the concentration of adsorbed mAbs (q_1 and q_2) are also given by the following equation:

$$\frac{\partial q_i}{\partial t} = k_i[(q_{max,i} - q_i)c_p|_{r=r_p} - \frac{q_i}{K}] \quad \text{for} \quad i = 1, 2 \tag{2.44}$$

where k_i is the adsorption kinetic constant, q_{max} is the column capacity, and K is the Langmuir equilibrium constant. In this work, we assume the porous beads have two binding sites; a fast-binding one and a slow-binding one. As a result, two indices are employed to represent these two sites, one representing the fast-binding site and the other representing the slow-binding site [4]. It should be noted that the optimization work focuses on the loading mode of the capture column, and thus, the elution mode model is not considered.

2.2.1.2 Compact Form of Capture Column Model

The model equations given by Equations (2.40)-(2.44) are partial differential equations (PDEs). The equations were discretized to convert to ordinary differential equations (ODEs) using the two point central difference method and the ODEs were solved using numerical methods. The beads within the column were discretized into 5 segments along the radial direction, resulting in a total of 5 c_p states. Consequently, the capture column has a total of 8 states. Furthermore, along the axial direction, the column is divided into 75 equal parts, leading to a grand total of 600 states. The compact form of the capture column model can be represented by Equation (2.45), where Equation (2.45b) represents the system state vector and Equation (2.45c) represents the continuous input vector which contains the inlet mAb concentration and flow rate which have the constant values 49.9219 mg/L and 21.6129 L/min respectively. These values were obtained from the upstream process at steady state.

$$x(k+1) = f(x(k), u^c(k)) \tag{2.45a}$$

$$x(k) \in \mathbb{R}^{600} \tag{2.45b}$$

$$u^c(k) \in \mathbb{R}^2 \tag{2.45c}$$

The modelling of the capture column with the switching operation is given by Equation (2.46a):

$$x(k+1) = f(x(k), u^c(k))[1 - u^d] \tag{2.46a}$$

$$u^d(k) \in \mathbb{R} \tag{2.46b}$$

where u^d is the discrete input which is either 0 or 1.

Chapter 3

State Estimation of Upstream Processes

In this chapter, state estimation is performed for the upstream mAb process. We begin by considering sensor placement to select the most estimable states for estimation. We also investigate the effect of estimation some parameters in addition.

3.1 Sensor Selection for State Estimation

The problem of optimal sensor selection is encountered in various areas such as robotics, healthcare, and industrial process control. While an exhaustive search of all possible sensor combinations may be feasible for small-scale systems, it becomes computationally inefficient for large-scale systems, where the complexity grows exponentially with the number of candidate sensors. Addressing this issue requires a reliable sensor selection scheme that offers improved performance and lower computational complexity [41], [42]. In this work, we apply a sensitivity-based approach for optimal sensor selection to the upstream process of mAb production. This approach, developed by Liu et al. [26], was utilized to determine the optimal number and placement of sensors in a wastewater treatment plant. The algorithm involves constructing the sensitivity matrix and employing orthogonalization and degree of observability for sensor selection. To begin, we take a look at the construction of the sensitivity matrix, and the sensor placement process.

3.1.1 Construction of the Sensitivity Matrix

Observability plays a crucial role in state estimation as it determines the possibility of deducing the internal states of a system based on the measurements of a subset of those states. The sensitivity matrix, in the context of nonlinear systems, is a valuable tool for assessing observability. The work presented in [30] sheds light on the relation between the sensitivity matrix and the observability of dynamical nonlinear systems. Moreover, it elucidates the correlation between the observability matrix and observability in linear systems. Through these insights, a deeper understanding of the relationship between observability and the sensitivity matrix can be gained. Here, we delve into the use of the sensitivity matrix for evaluating the observability of a discrete-time nonlinear system. Considering the discrete-time nonlinear system described by Equations (2.38) - (2.39), the sensitivity matrix, $S_{y,x(0)}(k)$, at sampling time k can be constructed as shown in Equation (3.1):

$$S_{y,x(0)}(k) = \begin{bmatrix} S_{11}(k) & \cdots & S_{1n}(k) \\ \vdots & \ddots & \vdots \\ S_{m1}(k) & \cdots & S_{mn}(k) \end{bmatrix} \in \mathbb{R}^{m\times n} \tag{3.1}$$

where each column represents the sensitivity of the different outputs to a specific state element. The individual entries of the matrix, which can be denoted as $S_{ij}(k)$, can be calculated using Equation (3.2a). Furthermore, Equation (3.2b) expresses the sensitivity of the state $x(k)$ with respect to the initial state $x(0)$, as mentioned in [26].

$$S_{i,j}(k) = \frac{\partial y_i(k)}{\partial x_j(0)} = \frac{\partial h_i}{\partial x_j}(k) S_{x,x(0)}(k) \tag{3.2a}$$

$$S_{x,x(0)}(k) = \frac{\partial x(k)}{\partial x(0)} \tag{3.2b}$$

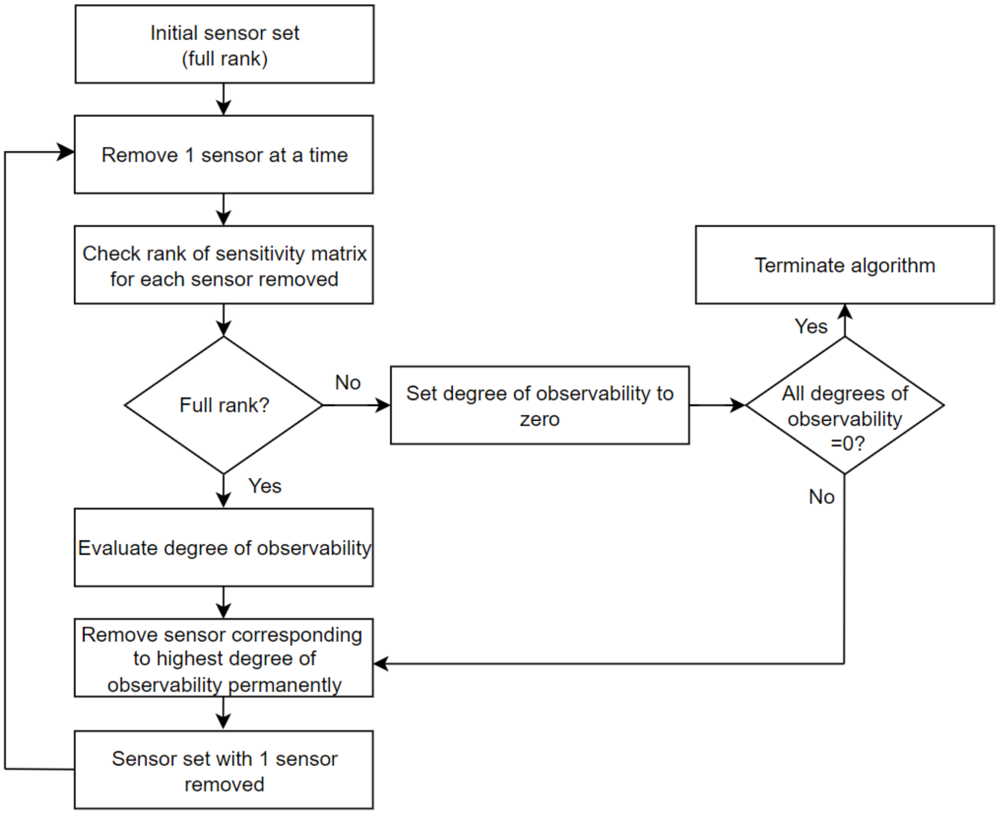
Initial sensor set (full rank)
Remove 1 sensor at a time
Check rank of sensitivity matrix for each sensor removed
Full rank?
No
Yes
Set degree of observability to zero
All degrees of observability =0?
Yes
No
Terminate algorithm
Evaluate degree of observability
Remove sensor corresponding to highest degree of observability permanently
Sensor set with 1 sensor removed

Figure 3.1: A schematic flow diagram of sensor selection process.

The normalized sensitivity matrix from sampling time 0 to k can be constructed as Equation (3.3). If the observability matrix, $S(k, 0)$, is full rank along the entire trajectory from sampling time 0 to k, then the nonlinear system is locally observable along the trajectory. This implies that the current states $x(k)$ can be fully estimated using input and output data [30], [43].

$$S(k,0) = \begin{bmatrix} S_{y,x(0)}(0) \\ S_{y,x(0)}(1) \\ \vdots \\ S_{y,x(0)}(k) \end{bmatrix} \in \mathbb{R}^{(k+1)m \times n} \tag{3.3}$$

3.1.2 Procedure to Determine Minimum Number and Optimal Placement of Sensors

The sensitivity matrix is normalized to ensure that the magnitudes of its elements are on a consistent scale. The norm of each column of the sensitivity matrix is used as a measure of the degree of observability. Larger norms indicate greater sensitivity of the output to the corresponding state. However, the direct use of the norms is limited by the potential linear dependence among some columns. To address this, successive orthogonalization is applied to identify the strongly linearly independent columns. Starting with the sensor corresponding to the column with the smallest norm, sensors are successively removed for states that contribute the least to the overall system observability. The process continues until removing another sensor would render the system unobservable. The orthogonalization procedure used to rank the columns by magnitude, and the final algorithm used to obtain the minimum sensor set have been detailed in [26].

To derive the ultimate minimum sensor set, the procedure in Figure 3.1 is implemented. Given the initial sensor set containing all the sensors, the sensors are removed one at a time. For each sensor removed from the set, the rank of the normalized sensitivity matrix is checked to ensure that the system is still observable. If the matrix is rank deficient, the degree of observability is set to zero. If the matrix is full rank, the total degree of observability is calculated based on the remaining sensors in the set. This ensures that the algorithm considers only observable systems moving forward. The total degree of observability is the sum of the norms of the columns of the normalized sensitivity matrix, where the norms of the matrix have been obtained and ordered using orthogonalization to eliminate linear dependence. The sensor which when removed from the set results in the highest degree of observability is chosen and removed permanently from the set. This is because that sensor contributes the least to the overall observability of the system. This process continues until the further removal of any sensor would cause the system to lose observability. Consequently, the algorithm terminates at the step where for all the remaining sensors, the degree of

Table 3.1: Sensor selection process for mAb process.

m	$\mathcal{U}$	$(S^{(m)})$	*rank*	degree$_{max}$	sensor$_{rem}$
16	{}	{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16}	16	64.025	None
15	{2}	{1,3,4,5,6,7,8,9,10,11,12,13,14,15,16}	16	61.994	2
14	{2,1}	{3,4,5,6,7,8,9,10,11,12,13,14,15,16}	16	59.894	1
13	{2,1,3}	{4,5,6,7,8,9,10,11,12,13,14,15,16}	16	57.717	3
12	{2,1,3,7}	{4,5,6,8,9,10,11,12,13,14,15,16}	16	55.255	7
11	{2,1,3,7,14}	{4,5,6,8,9,10,11,12,13,15,16}	16	53.096	14
10	{2,1,3,7,14,6}	{4,5,8,9,10,11,12,13,15,16}	16	50.628	6
9	{2,1,3,7,14,6,15}	{4,5,8,9,10,11,12,13,16}	16	48.032	15
8	{2,1,3,7,14,6,15,5}	{4,8,9,10,11,12,13,16}	16	45.289	5
7	{2,1,3,7,14,6,15,5,4}	{8,9,10,11,12,13,16}	16	42.367	4
m	$\mathcal{U}$	$(S^{(m)})$	*rank*	D(S^m)	sensor taken out
6	{2,1,3,7,14,6,15,5,4,8}	{9,10,11,12,13,16}	15	36.777 → 0	8
6	{2,1,3,7,14,6,15,5,4,9}	{8,10,11,12,13,16}	14	34.325 → 0	9
6	{2,1,3,7,14,6,15,5,4,10}	{8,9,11,12,13,15}	15	36.777 → 0	10
6	{2,1,3,7,14,6,15,5,4,11}	{8,9,10,12,13,16}	15	36.777 → 0	11
6	{2,1,3,7,14,6,15,5,4,12}	{8,9,10,11,13,16}	15	36.777 → 0	12
6	{2,1,3,7,14,6,15,5,4,13}	{8,9,10,11,12,16}	15	36.777 → 0	13
6	{2,1,3,7,14,6,15,5,4,16}	{8,9,10,11,12,13}	13	31.845 → 0	16

observability has been set to zero due to rank deficiency.

3.1.3 Minimum Sensor Set Selection

Consider the mAb process described in Section 2.1 and its compact representation given by Equations (2.38) - (2.39). We assume the presence of commercially-available sensors, as discussed in [44], [45], [46], capable of measuring all 16 states representing temperature and the concentrations of the nutrients, metabolites, and mAbs. To initiate the sensor selection process, we begin with the original full sensor set denoted as $S^{(16)}$, which comprises all 16 sensors. The initial set $S^{(16)}$ is defined as $\{Xv_1, X_{t1}, GLC_1, GLn_1, LAC_1, AMM_1, mAb_1, Xv_2, Xt_2, GLC_2, GLN_2, LAC_2, AMM_2, mAb_2, T, c\}$ which can be simplified as $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16\}$ for convenience.

Table 3.1 illustrates the step-by-step removal of sensors using the sensor selection algorithm to determine the final optimal minimum sensor set. Following the algorithm, for each sensor removed, the degree of observability for the remaining set is calculated and the corresponding sensor for the set with the highest degree of observability is removed. This removal is performed because the removed sensor contributes the least to the overall observability of the system. In Table 3.1, m represents the number of sensors remaining in the sensor set,

$\mathcal{U}$ denotes the set of removed sensors, $rank$ is the rank of the sensitivity matrix which is constructed based on the current sensor set combination. degree$_{max}$ is the maximum degree of observability at each step of the removal process and for which the corresponding sensor is permanently removed, sensor$_{rem}$ indicates the sensor corresponding to the maximum degree of observability and which is removed permanently at the current step. $D(S^m)$ also represents the degree of observability for a particular sensor taken out.

We initiate the sensor selection algorithm with the full sensor set $S^{(16)}$, which exhibits a total degree of observability of 64.025 and a full-rank sensitivity matrix. The algorithm proceeds sequentially by removing one sensor at a time from the set of 16 sensors. For each iteration, the sensitivity matrix is reconstructed and its rank is assessed. If the rank of the sensitivity matrix is less than 16, indicating rank deficiency, the degree of observability is set to zero. Conversely, if the rank is 16, the degree of observability remains effective. Analyzing the degrees of observability for each sensor removal, we find that removing sensor 2 yields the highest degree of observability at 61.994. Hence, sensor 2 is permanently eliminated from the set as it contributes the least to the overall observability of the system.

Moving forward, we proceed to the next step of the algorithm with a sensor set of 15 sensors. The objective is to remove one more sensor to obtain an optimal set containing 14 sensors. After evaluating the degrees of observability for each sensor removal, it is observed that removing sensor 1 leads to the maximum degree of observability. Consequently, sensor 1 is removed permanently from the set. Following a similar procedure, sensors 3, 7, 14, 6, 15, 5, and 4 are subsequently removed permanently from the set, one after another.

After the removal of sensor 4, the remaining set consists of sensors $\{8, 9, 10, 11, 12, 13, 16\}$. Upon removing sensor 8, the sensitivity matrix is rank deficient, therefore the degree of observability, $D(S^m)$, is set to 0 and sensor 8 is not permanently removed. Similar observations are made for sensors 9, 10, 11, 12, 13, and 16, therefore none of these sensors is removed permanently as the removal of any of these sensors results in a rank-deficient sensitivity matrix. Thus, the algorithm terminates at $m = 7$, and the final optimal minimum sensor set is $S^{(7)} =$

$\{8, 9, 10, 11, 12, 13, 16\}$, which corresponds to $\{Xv_2, Xt_2, GLC_2, GLN_2, LAC_2, AMM_2, c\}$.

3.2 State Estimation Method

The sensors selected in Section 3.1.3 play a crucial role in obtaining measurements which when combined with the process model, enable the reconstruction of the system's states. However, relying solely on state estimation may not capture all the valuable information embedded within the measurements. To fully exploit the available information, it is advantageous to also estimate some parameters. This approach enhances the algorithm's resilience to model uncertainty and improves overall estimation performance. Simultaneous state and parameter estimation has garnered significant attention, as it offers the capability to achieve better estimation outcomes [30], [47], [48].

Estimating parameters in complex systems usually presents a challenge, as there are often more parameters than can be reliably estimated. Consequently, it becomes crucial to determine a subset of parameters that can be accurately estimated. How to select the appropriate subset of parameters becomes a significant question to address [49]. To determine which parameters should be included for estimation, variable selection is employed. First, the parameters are augmented as extra states of the original system. Next, the sensitivity matrix for the augmented system, S_a, is constructed and normalized. Subsequently, the rank of the normalized sensitivity matrix is checked. If the sensitivity matrix is not full rank, it indicates that not all the elements of the augmented system can be uniquely estimated, and it becomes important to identify the subset of elements that can be estimated accurately.

Using orthogonalization, the norms of the columns of S_a are evaluated and the column with the largest norm is selected and denoted as X_j. Next, the residual information matrix is calculated. The column of S_a which corresponds to the largest norm in the residual matrix is taken and added to X_j to form X_{j+1}. The cut-off criterion for algorithm termination is the rank of the sensitivity matrix. Columns are added successively in this order until the rank of

X_j is equal to the rank of S_a and the algorithm is terminated. The variables corresponding to the columns of X_{j+i} are identified as the most estimable variables which correspond to the strongly linearly-independent columns of the sensitivity matrix. This variable selection procedure used can be found in [30], where it is illustrated with an example.

Here, the variable selection process considers a total of 18 parameters, which are the parameters of interest and are listed as follows: $\{K_{d,amm}$, $K_{d,gln}$, K_{glc}, K_{gln}, KI_{amm}, KI_{lac}, n, Xv_2, $Y_{amm,gln}$, $Y_{lac,glc}$, $Y_{X,glc}$, $Y_{X,gln}$, α_2, $\mu_{d,max}$, Q_{mAb}^{max}, $-\Delta H$, ρ, c_p, $U\}$. To estimate the states and parameters simultaneously, the parameters are augmented as extra states by combining Equation (3.4) with Equation (3.5).

$$x(k+1) = F(x(k), u(k), p(k)) + w_x(k) \tag{3.4}$$

$$p(k+1) = p(k) + w_p(k) \tag{3.5}$$

Equation (3.4) represents the state equation, considering the parameters of interest which are represented by the vector $p(k) \subset \mathbb{R}^{18} \in P(k)$. Equation (3.5) represents the parameter equation, where $w_p(k) \in \mathbb{W}_p \subset \mathbb{R}^{18}$ is a vector representing process noise and containing 18 elements. The parameters, represented by the vector p, are assumed to be constant. The augmented state vector is represented by X, where $X = [x^T, p^T]^T$. The discrete-time representation of the augmented system is shown in Equations (3.6) - (3.7):

$$X(k+1) = F_a(X(k), u(k)) + w_a(k) \tag{3.6}$$

$$y(k) = H_a(X(k)) + v(k) \tag{3.7}$$

where the subscript ‘a’ represents the augmented system. In Equation (3.6), $X(k) \in \mathbb{X}_a \subset \mathbb{R}^{34}$ represents the augmented system state vector, $u(k) \in \mathbb{U} \subset \mathbb{R}^8$ represents the input vector, and $w_a(k) \in \mathbb{W}_a \subset \mathbb{R}^{34}$ represents the process noise vector. Equation (3.7) describes the measured output vector $y(k) \in \mathbb{R}^7$, which contains the measurements obtained from the 7 selected sensors, and the measurement noise vector is represented by $v(k) \in \mathbb{V} \subset \mathbb{R}^7$.

Through Singular Value Decomposition (SVD) analysis, the rank of the normalized sensitivity matrix of the augmented system was determined to be 21. Therefore, the variable selection algorithm terminates after the selection of the first 21 most estimable variables. The selected variables consist of all 16 states and 5 of the parameters. The chosen parameters are: $\{Q_{mAb}^{max}, K_{d,amm}, \rho, \mu_{d,max}, K_{gln}\}$. The 21 selected variables are estimated using MHE, which is an estimation technique in which the estimation problem is formulated as an optimization problem [50], [51], [52]. The MHE formulation at sampling time k for the simultaneous state and parameter estimation of the augmented system is given by Equations (3.8a) - (3.8f):

$$\min_{\hat{X}(k-N),..,\hat{X}(k),\hat{w}_a(k-N),..,\hat{w}_a(k-1)} \quad \left\|\hat{X}(k-N) - \bar{X}(k-N)\right\|_{P^{-1}}^2 + \sum_{j=k-N}^{k-1} \|\hat{w}_a(j)\|_{Q^{-1}}^2 + \sum_{j=k-N}^{k} \|\hat{v}(j)\|_{R^{-1}}^2 \tag{3.8a}$$

$$\text{s.t} \quad \hat{X}(j+1) = F_a(\hat{X}(j), u(j)) + \hat{w}_a(j), j \in [k-N, k-1] \subset \mathbb{Z} \tag{3.8b}$$

$$\hat{v}(j) = y(j) - H_a(\hat{X}(j)), j \in [k-N, k] \subset \mathbb{Z} \tag{3.8c}$$

$$\bar{X}(k-N) = \hat{X}(k-N|k-N) \tag{3.8d}$$

$$\hat{X}(j) \in \mathbb{X}_a, \hat{v}(j) \in \mathbb{V}, j \in [k-N, k] \subset \mathbb{Z} \tag{3.8e}$$

$$\hat{w}_a(j) \in \mathbb{W}_a, j \in [k-N, k-1] \subset \mathbb{Z} \tag{3.8f}$$

The objective of MHE optimization is to minimize the mismatch between the predicted and actual measurements, which is represented by the term $\|\hat{v}\|_{R^{-1}}^2$ in Equation (3.8a). The term $\hat{v}$ is defined in Equation (3.8c), and the caret sign ˆ indicates an estimated variable. The process disturbance is also accounted for by the term $\|\hat{w}_a\|_{Q^{-1}}^2$ in Equation (3.8a), and $\hat{w}_a$ is defined in Equation (3.8f). The arrival cost, which represents the information from the initial state of the model to the beginning of the estimation window, is given by $\left\|\hat{X} - \bar{X}\right\|_{P^{-1}}^2$. Here, N denotes the length of the estimation window, and $\hat{X}$ and $\hat{w}_a$ are the decision variables of the optimization problem. $\hat{X}$ represents the estimated variable and is defined in Equation (3.8e). $\hat{X}(k-N|k-N)$ represents the estimated state, $\hat{X}$, at time instant $k-N$, and

$\bar{X}$ is given by Equation (3.8d). The matrices P, Q, and R are the covariance matrices for state uncertainty, process noise, and measurement noise, respectively. Furthermore, MHE incorporates constraints on the states, parameters, and model uncertainties in Equations (3.8e) - (3.8f).

3.3 State Estimation Results

In this section, the focus is on performing state estimation using the selected sensors determined by the sensor placement algorithm described in Section 3.1. The MHE technique is employed to estimate all 16 states of the system using the nominal parameter values provided in Table 2.1. The estimation results are then examined by comparing the estimated state trajectories with the actual state trajectories.

Furthermore, to investigate the effect of estimating some parameters in addition to the states, three different cases are analyzed. In Case 1, state estimation alone is conducted, and in Case 2, all 16 states and 18 parameters are estimated simultaneously. In Case 3, only the selected variables (16 states and 5 parameters) obtained through the variable selection algorithm described in Section 3.2 are estimated, representing the proposed approach. In all three cases, some bias was added to the parameters of the plant model, whereas the nominal parameters were used for the estimator. The initial parameter guess for MHE are the nominal parameter values.

The evaluation of the estimation results in all three cases involves comparing the Root Mean Squared Errors (RMSEs) and plotting the estimated and actual values. Additionally, the error evolution for all three cases is analyzed to assess the estimation performance comprehensively.

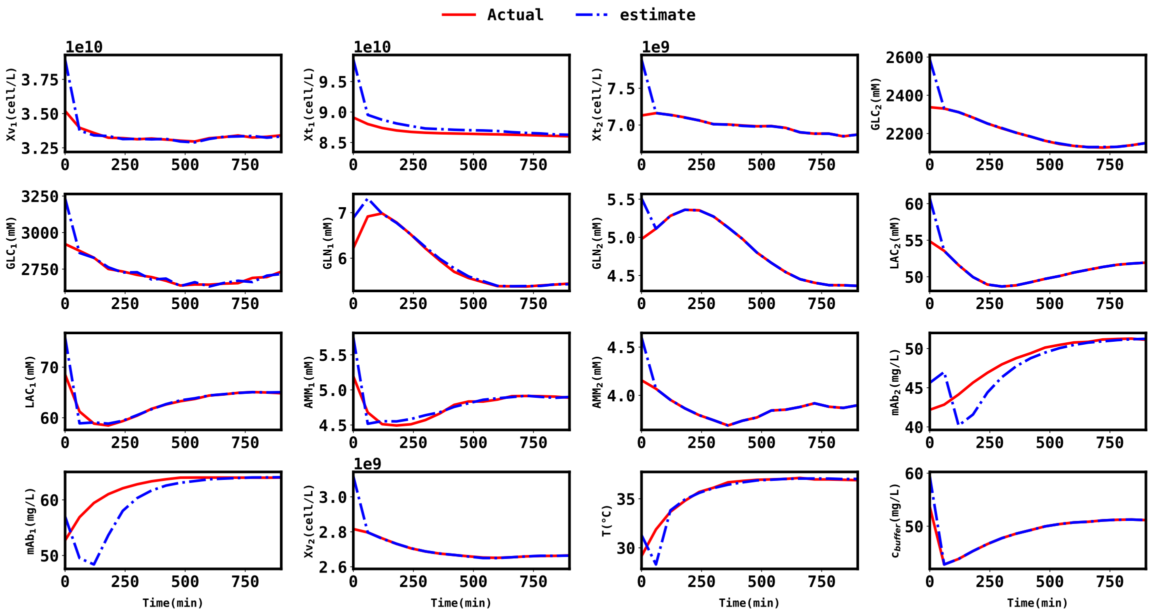

Figure 3.2: Plot of the actual states and their estimates based on the minimum sensor set.

3.3.1 Simulation Settings

To begin, state estimation is performed using the nominal parameters listed in Table 2.1. For the MHE estimator, the weighting matrices are diagonal matrices, where the state covariance matrix, $P = diag\{[50^2, 50^2, 50^2, 50^2, 50^2, 50^2, 50^2, 50^2, 50^2, 50^2, 50^2, 50^2, 50^2, 50^2, 50^2, 50^2]\}$, and the process covariance matrix and measurement covariance matrix are given by Q and R respectively: $Q = diag\{[6.25 \times 10^{12}, 6.25 \times 10^8, 0.2025, 6.25 \times 10^{-8}, 6.25 \times 10^{-6}, 1.6 \times 10^{-7}, 2.25 \times 10^{-6}, 6.25 \times 10^{-6}, 2.5 \times 10^{11}, 5.625 \times 10^{-5}, 4 \times 10^{-8}, 2.25 \times 10^{-6}, 1.41 \times 10^{-7}, 6.25 \times 10^{-6}, 2.25 \times 10^{-6}, 2.5 \times 10^{-9}]\}$, $R = diag\{[2.26 \times 10^{-7}, 2.26 \times 10^{-7}, 2.26 \times 10^{-7}, 7.31 \times 10^{-9}, 7.31 \times 10^{-9}, 8.15 \times 10^{-9}, 7.31 \times 10^{-9}]\}$. Using a moving horizon length of 40 for MHE, the results obtained were plotted in Figure 3.2, where the red line represents the actual state trajectories and the blue line represents the estimates. Since the estimated states are able to closely track the actual state trajectory, it can be concluded that the sensor selection algorithm was effective in selecting the minimum number of sensors, and their optimal locations for estimation.

Furthermore, to asses the benefit of estimating some parameters in addition, estimation is performed for the three cases mentioned above, and the state trajectories are compared in Figure 3.3. In the plots, the red line represents the actual state trajectories, the blue line represents the estimates from Case 1, the black line represents the estimates from Case 2, and the green line represents the estimates from Case 3. The estimated parameters were also plotted and compared in Figure 3.4. The average RMSE for all three cases over time, from $k = 0$ to $k = N_{sim} - 1$ was calculated using Equations (3.9a) - (3.9b) and compared in Table 3.2.

$$RMSE_x = \frac{1}{N_{sim}} \sum_{k=0}^{N_{sim}-1} \sqrt{\frac{1}{N_x} \sum_{i=1}^{N_x} \left(\frac{\hat{x}_i(k) - x_i(k)}{x_i(k)} \right)^2} \tag{3.9a}$$

$$RMSE_p = \frac{1}{N_{sim}} \sum_{k=0}^{N_{sim}-1} \sqrt{\frac{1}{N_p} \sum_{i=1}^{N_p} \left(\frac{\hat{p}_i(k) - p_i(k)}{p_i(k)} \right)^2} \tag{3.9b}$$

Equation (3.9a) represents the RMSE for the states, and Equation (3.9b) represents the RMSE for the parameters, where N_{sim} represents the total number of simulation steps, N_x represents the total number of states, x_i denotes the actual states, and $\hat{x}_i$ denotes the estimated states. Similarly, N_p represents the total number of parameters, p_i denotes the true parameters, and $\hat{p}_i$ denotes the estimated parameters.

From Table 3.2, we see that estimating parameters in addition to the states(Case 2 and Case 3) leads to better estimation performance. This is evident from the fact that Case 1 exhibits the poorest performance. The better performance of the proposed approach, Case 3, compared to Case 2 shows that estimating more than the selected variables of the augmented system leads to a degradation in state estimation performance. Table 3.2 clearly demonstrates that the proposed approach, Case 3, yields the most favorable outcome as evidenced by its lowest RMSE values for both the states and parameters. The parameter RMSE recorded in Case 1 is constant because in Case 1 no parameters are estimated, and the RMSE calculation is based on the true parameters used in the plant and the nominal

Case	x_0	$RMSE_x$	$RMSE_\theta$
Case 1	$0.95x_s$	0.02773	0.048646
	$0.97x_s$	0.02777	
	$1.1x_s$	0.02858	
	$1.15x_s$	0.02850	
	$1.2x_s$	0.02875	
Average		**0.02827**	**0.04865**
Case 2	$0.95x_s$	0.01750	0.01745
	$0.97x_s$	0.01462	0.01733
	$1.1x_s$	0.01363	0.01679
	$1.15x_s$	0.01315	0.01676
	$1.2x_s$	0.01281	0.01664
Average		**0.01434**	**0.01699**
Case 3	$0.95x_0$	0.01597	0.00765
	$0.97x_0$	0.01336	0.00769
	$1.1x_0$	0.01317	0.00789
	$1.15x_0$	0.01284	0.00777
	$1.2x_0$	0.01262	0.00769
Average		**0.01359**	**0.00774**

Table 3.2: Average RMSE considering different initial states in all three cases.

parameters used in the estimator. The true parameters are represented as:

$$\theta_{true} = \theta_{nominal} + 10\% \times \theta_{nominal}\ (for\ unselected\ parameters) \tag{3.10a}$$

$$\theta_{true} = \theta_{nominal} + 1\% \times \theta_{nominal}\ (for\ selected\ parameters) \tag{3.10b}$$

To quantify the improvement percentage of the state RMSE for Case 3 compared to Case 1 and Case 2, we utilize Equations (3.11a) - (3.11b):

$$Improvement\ percentage = \left(\frac{RMSE_x(Case\ 1) - RMSE_x(Case\ 3)}{RMSE_x(Case\ 1)}\right) \times 100 \tag{3.11a}$$

$$Improvement\ percentage = \left(\frac{RMSE_x(Case\ 2) - RMSE_x(Case\ 3)}{RMSE_x(Case\ 2)}\right) \times 100 \tag{3.11b}$$

Considering the average state RMSE, Case 3 shows a 51.92% and 5.2% improvement over Case 1 and Case 2 respectively. Similarly, Considering the average parameter RMSE, Case 3 shows a 54.4% improvement over Case 2. The error evolution for the states and parameters

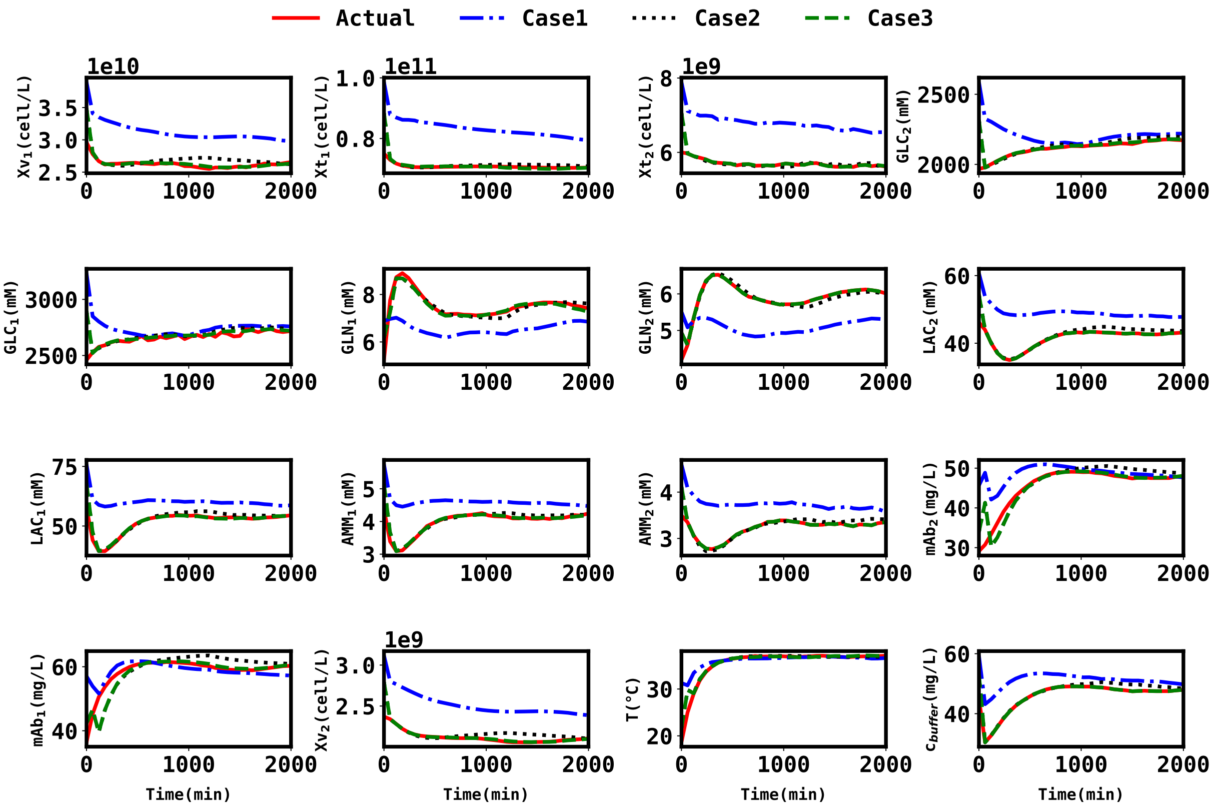

Figure 3.3: Plot of the actual states and estimates for all three cases.

were also plotted and compared in Figures 3.5 and 3.6 respectively.

3.4 Summary

In conclusion, the results presented in Table 3.2 exhibit the superiority of the proposed approach, Case 3, in terms of estimation performance. Case 3 consistently exhibits the lowest RMSE values for both the states and parameters, indicating the benefit of estimating the additional parameters selected through the variable selection procedure.

Conversely, Case 1, which solely focuses on state estimation without considering parameter estimation, demonstrates the poorest performance, as reflected by its highest RMSE. This emphasizes the importance of including parameter estimation in achieving improved estimation results.

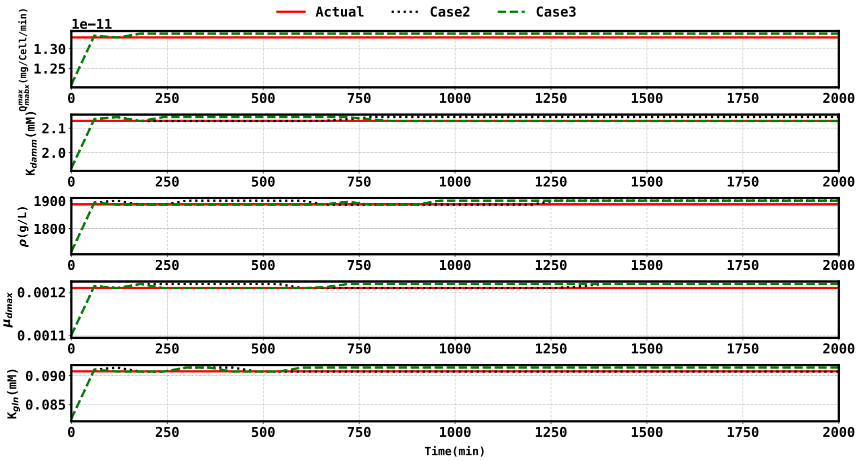

Figure 3.4: Plot of nominal parameters and estimates for Case 2 and Case 3.

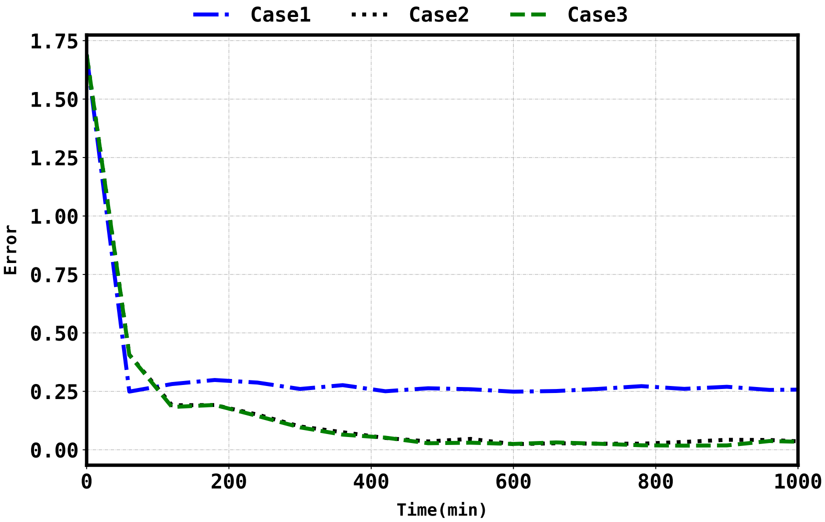

Figure 3.5: State error evolution in all three cases.

Furthermore, comparing Case 3 to Case 2 in which all variables of the augmented system are estimated, we see that estimating all the parameters alongside the states can lead to a degradation in estimation performance. The average RMSE of Case 3 is found to be lower than that of Case 2, indicating the potential adverse effect of estimating unobservable parameters.

Quantifying the improvement percentage of Case 3 over Case 1 and Case 2 in terms of state RMSE, we observe an impressive 51.92% improvement over Case 1 and a modest 5.2% improvement over Case 2. Similarly, comparing the parameter RMSE of Case 3 to Case 2,

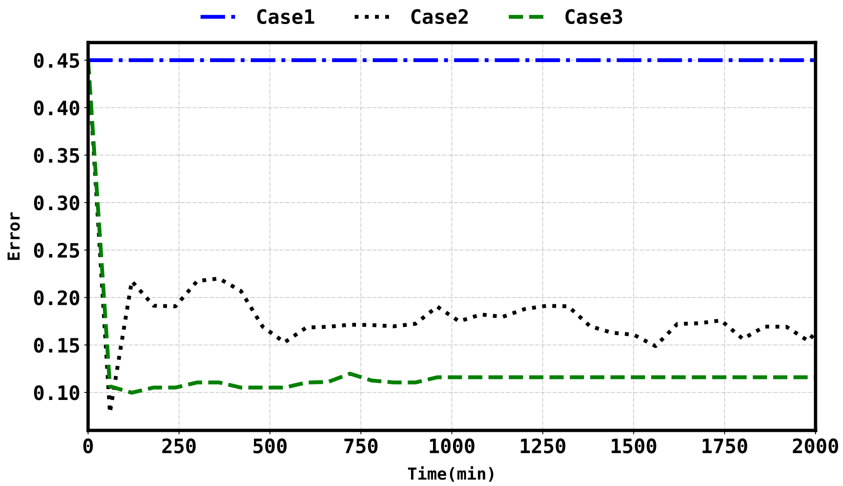

Figure 3.6: Parameter error evolution in Case 2 and Case 3.

Case 3 exhibits a noteworthy 54.4% improvement. The plotted error evolution for the states and parameters, as shown in Figures 3.5 and 3.6 respectively, further supports the findings and provides a visual representation of the performance differences between the cases.

In summary, the inclusion of parameter estimation, as demonstrated in Case 3, enhances the overall estimation performance. By leveraging a selective approach that considers both state and parameter estimation, more accurate and reliable estimation results can be obtained, enabling better understanding and control of the system under consideration.

Chapter 4

Control of Downstream Process

This chapter focuses on the control of the downstream process in mAb production. The downstream process includes critical steps such as capture, virus inactivation, polishing, and filtration, all of which are essential for refining the mAb product. Particular attention is given to optimizing the switching operation at the capture step, as it plays a crucial role in achieving continuity of the fully-continuous integrated production process.

The traditional approach to switching relies on fixed thresholds based on product breakthrough values, which may result in inefficient product loss and increased costs. To address this, this chapter explores the implementation of three distinct control techniques: the sigmoid function approximation approach, the ReLU approximation approach, and the RL approach.

The effectiveness of each method is evaluated based on product loss, cost optimization, and computational efficiency. Additionally, the impact of noise, as well as the weight factors influencing product loss and the switching action, are thoroughly investigated. The results provide valuable insights into the performance and robustness of each control approach, paving the way for enhanced downstream process control in mAb production.

4.1 Traditional Control Method

In many industries, profitability serves as the primary factor influencing decision-making. Therefore, it becomes crucial to adjust processes to optimize specified parameters and bring them closer to their optimal values while ensuring that environmental, safety, and customer requirements are met. In the absence of proper process control and optimization, a process may run below its maximum efficiency and may be more expensive to operate [53], [54].

The conventional operation of the capture column relies on a fixed switching strategy, where the switching event takes place once a predetermined percentage product breakthrough is observed. This approach lacks optimization measures to minimize product loss and cost. In a study conducted by Warikoo et al. [34], ultraviolet (UV) sensors were strategically placed at the capture column inlet feed and outlet. The decision to switch was determined based on the UV absorbance difference between these points, which indicated the attainment of a predefined percentage product breakthrough threshold. For instance, the study mentions a 3% product breakthrough value [34], while another study [3] mentions a 1% product breakthrough threshold. In this work, a 1% breakthrough threshold was used for the simulation of the traditional approach.

This work aims to enhance the operation of the capture step through the implementation of EMPC. However, EMPC implementation for integer optimization problems is known to be challenging. To overcome this challenge, we investigate two distinct approaches to improve the implementation of EMPC, namely, a sigmoid function approximation approach, and a ReLU approximation approach. Furthermore, a third approach which explores the use of RL is also explored, since it is able to handle the discrete action space. The subsequent sections provide a comprehensive overview of the problem formulation for the two EMPC optimization approaches and a detailed description of all three approaches.

4.2 Optimization-Based Control Problem Formulation

Given that the primary control objective is to prevent unnecessary product loss, the optimization problem is formulated to minimize product loss. Switching is triggered when the mAb concentration at the capture column outlet starts to rise. The EMPC formulation for the integer nonlinear problem is presented below:

$$\min_{u_i^d} \quad \sum_{i=k}^{k+N-1} \left(W_s x_{out,i} + W_d u_i^d\right) \tag{4.1a}$$

$$\text{s.t} \quad x_{i+1} = f(x_i, u_i^c)[1 - u_i^d]; i = k, .., k+N-1 \tag{4.1b}$$

$$x_k = x(k) \tag{4.1c}$$

$$u_i^d \in \{0, 1\}; i = k, .., k+N-1 \tag{4.1d}$$

where Equation (4.1a) represents the cost function, and Equations (4.1b)-(4.1d) represent the constraints. N represents the prediction horizon, x_{out} denotes the concentration of mAb at the column outlet, and W_s is the weight on x_{out}. u_i^d represents the discrete decision variable, which is binary and has a value of 0 or 1. It represents the decision to either switch or not switch, with 0 representing the decision not to switch and 1 representing the decision to switch. Furthermore, W_d represents the weight on the discrete decision variable.

Equation (4.1b) represents the system state equation, which is multiplied by the term $[1 - u_i^d]$. In this equation, x_i denotes the system states, and u_i^c represents the continuous inputs of the system. The multiplication of the state equation by $[1 - u_i^d]$ ensures that when $u_i^d = 0$, the states continue to increase with time. Conversely, when $u_i^d = 1$, the term $[1 - u_i^d]$ becomes 0, causing the states to go to zero. This is akin to resetting or switching the column.

Practically, very frequent column switching would keep x_{out} at zero, however, this is undesirable since it does not allow sufficient time for the product to reach the column outlet before switching occurs. Optimal operation requires the mAbs to have enough residence time to bind to the beads within the column during the loading process, as excessive switching would result in the underutilization of the columns. To address this issue, the penalty on the switching action, W_d, was introduced. By adjusting the value of W_d, the frequency of switching can be controlled. Increasing or reducing W_d and W_s enables the fine-tuning of the switching behavior.

4.2.1 Sigmoid Function Approximation Approach

Sigmoid functions have a wide range of applications and are applied in numerous areas such as computer science, engineering, finance and physics [55]. They are bounded between 0 and 1, differentiable, and have a characteristic s-shape [56]. They are widely used as activation functions in neural networks to convert a real number to a probability, and can be used in logistic regression to predict the outcome of binary classification problems.

In the work by Agyeman et al. [57], a sigmoid function was used to simplify a mixed-integer MPC problem in order to enhance the computational efficiency of an irrigation scheduler. In the work by Shao et al. [58], a sigmoid function-based integral-derivative observer was applied to an autopilot design, and in the work by Khairunnahar et al. [59], a sigmoid function was used in logistic regression to classify malignant and benign tissue to improve breast cancer detection in women.

Without the implementation of the sigmoid activation function, determining the optimal discrete decision variable that minimizes the cost function would necessitate the utilization of a nonlinear integer solver like the Basic Open-source Nonlinear Mixed Integer programming (BONMIN) solver to solve the integer problem [60]. However, this approach typically demands significant computational resources and considerable time. Incorporating a sigmoid function allows for the utilization of continuous decision variables within defined boundaries. This transformation converts the original integer optimization problem into a continuous optimization problem, which can be addressed more efficiently using a continuous nonlinear solver such as the Interior Point Optimizer (IPOPT) solver [61]. By solving the nonlinear optimization problem, a continuous decision variable is obtained, which is subsequently approximated as either 0 or 1 prior to its application to the actual system. The sigmoid function is mathematically defined as:

$$\sigma(r) = \frac{1}{1 + e^{-\beta r}} \tag{4.2}$$

where β is the slope of the sigmoid curve whose numerical value was chosen as 15, and r is the continuous decision variable selected within the range $r_{min} \leq r \leq r_{max}$, where r_{min} and r_{max} were chosen as -10 and 10 respectively. The solver selects a continuous variable, r, within this range as the optimal solution. The selected r value is applied to the sigmoid function, which returns an output between 0 and 1. The output is represented by $\sigma(r)$. When $\sigma(r) \geq 0.5$, it is assigned a discrete value of 1 before it is applied to the actual system. Similarly, when $\sigma(r) < 0.5$, it is assigned a discrete value of 0 and applied to the system. The optimisation problem can therefore be re-written as:

$$\min_{r} \quad \sum_{i=k}^{k+N-1} \left(W_s x_{out,i} + W_d \sigma_i(r)\right) \tag{4.3a}$$

$$\text{s.t} \quad x_{i+1} = f\left(x_i, u_i^c\right)\left[1 - \sigma_i(r)\right]; i = k, .., k+N-1 \tag{4.3b}$$

$$r_{min} \leq r \leq r_{max} \tag{4.3c}$$

$$x_k = x(k) \tag{4.3d}$$

A plot of the sigmoid function is shown in Figure 4.1.

4.2.2 Rectified Linear Unit Approach

Artificial neural networks (ANNs) are data-driven models designed to identify relationships within datasets. Unlike traditional regression models, ANNs possess the capability to learn and represent complex nonlinear relationships [62], [63]. The architecture of an ANN consists of interconnected nodes organized into layers. The basic building block is the neuron, which receives inputs, performs a mathematical computation, and generates an output for the

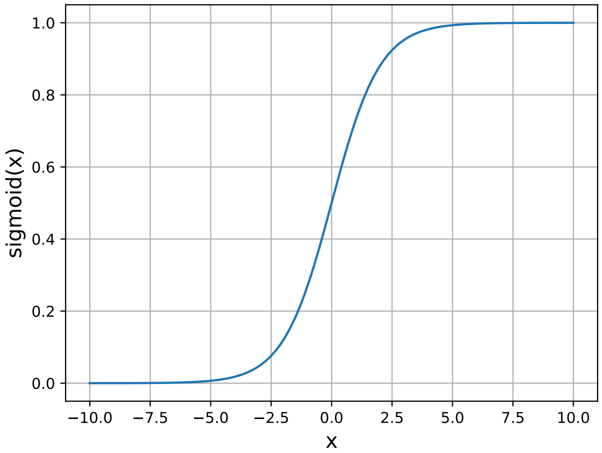

Figure 4.1: Plot of sigmoid function.

subsequent layer. Neurons are activated when the weighted sum of inputs exceeds a threshold value. Once activated, the neuron applies a transfer function to the signal before passing it to neighboring nodes [62].

Transfer functions come in various forms, with nonlinear functions proving more valuable for handling a wide range of data patterns [64], [65]. While the sigmoid and hyperbolic tangent (tanh) functions have long been popular, they suffer from the vanishing gradient problem. In contrast, the ReLU transfer function has gained attention for its superior performance in training ANNs [65].

By employing the ReLU activation function, we can convert the original nonlinear optimization problem into an Integer Linear Program (ILP) by leveraging the integer linear formulation of ReLU neural networks. This transformation, done using the optimization and machine learning toolkit (OMLT) [66], allows us to efficiently solve the resulting ILP to global optimality using readily available solvers such as Gurobi [67]. The ILP formulation is based on the fact that a ReLU network can be described by a set of integer linear constraints. The ReLU activation function is defined as $ReLU(x) = \max(0, x)$, which returns the same x input if x is positive, and returns 0 for negative values of x. Figure 4.2 presents a plot of the ReLU function. The ReLU neural network for this system can be represented by Equations

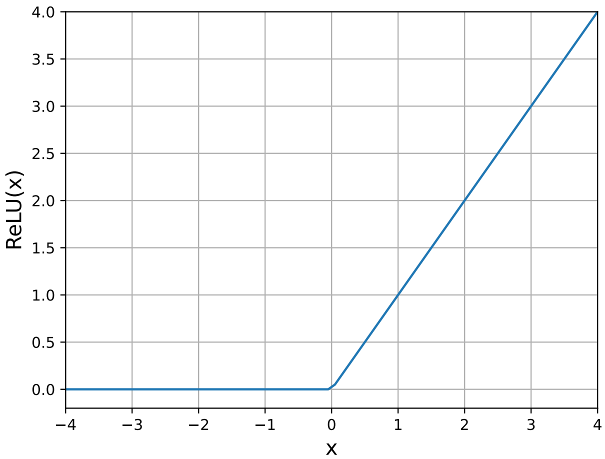

Figure 4.2: Plot of ReLU Function.

(4.4a) - (4.4c):

$$z_0 = \begin{bmatrix} x_{out,t} \\ x_{out,t-1} \\ u_t^{c_1} \\ u_t^{c_2} \\ u_{t-1}^{c_1} \\ u_{t-1}^{c_2} \end{bmatrix} \tag{4.4a}$$

$$z_{\ell+1} = max\left(W_\ell z_\ell + b_\ell, 0\right); \ell = 0, \ldots, L-1 \tag{4.4b}$$

$$y_t = x_{out,t+1} \tag{4.4c}$$

where z_0 represents the input to the neural network, which includes the state of interest x_{out} and the two continuous inputs from Equation (2.45c) at the current and previous time steps. $z_{\ell+1}$ denotes the output vector of the $(\ell+1)$th hidden layer, and y_t is the output of the neural network, representing x_{out} at the next time step. W_ℓ and b_ℓ are the weight matrix and bias vector of the $(\ell+1)$th hidden layer, respectively. For the $(\ell+1)$th layer in the ReLU network described by Equation (4.4b), let $\underline{\mathrm{m}}^{\ell}$ and $\bar{m}^{\ell}$ denote the lower and upper bounds on the input such that $\underline{m}^{\ell} \leq W_{\ell} z_{\ell} + b_{\ell} \leq \bar{m}^{\ell}$. The ReLU activation function can be expressed using the following integer linear constraints:

$z_{\ell+1} = max\left(W_{\ell} z_{\ell+1} + b_{\ell}, 0\right) \iff$

$$\begin{cases} z_{\ell+1} & \geq W_{\ell} z_{\ell+1} + b_{\ell} \\ z_{\ell+1} & \leq W_{\ell} z_{\ell+1} + b_{\ell} - diag(\underline{m}_{\ell})(1 - t_{\ell}) \\ z_{\ell+1} & \geq 0 \\ z_{\ell+1} & \leq diag(\bar{m}_{\ell}) t_{\ell} \end{cases}$$

where $t_{\ell} \in \{0,1\}$ is a vector of binary variables for the $(\ell+1)$th layer. For further information on this transformation, refer to the work by Tjeng et al. [68] and Chen et al. [69].

To train the ReLU neural network, a dataset of 50,000 data points was generated for the continuous inputs $u^{c}(k) \in \mathbb{R}^{2}$ in Equation (2.45c). Additionally, open-loop simulations using the first principles downstream model described in Chapter 2 provided a corresponding set of 50,000 data points for x_{out}. This input/output dataset was used to train the ReLU neural network. The ReLU network architecture consists of two hidden layers with 200 and 160 nodes, respectively. The training process utilized current and past time steps to predict the next time step. Default learning rate and batch size values (0.001 and 32, respectively) from the Keras library were used, and a total of 80 epochs were executed. The performance of the selected ReLU model was evaluated through single-step ahead and multiple-step ahead prediction plots in Figure 4.3, where the red and blue lines represented the predicted and actual state trajectories respectively. The data-driven model obtained through training the ReLU neural network is utilized by the EMPC to predict the discrete decision variable. Subsequently, the resulting solution is applied to the actual model for implementation. The

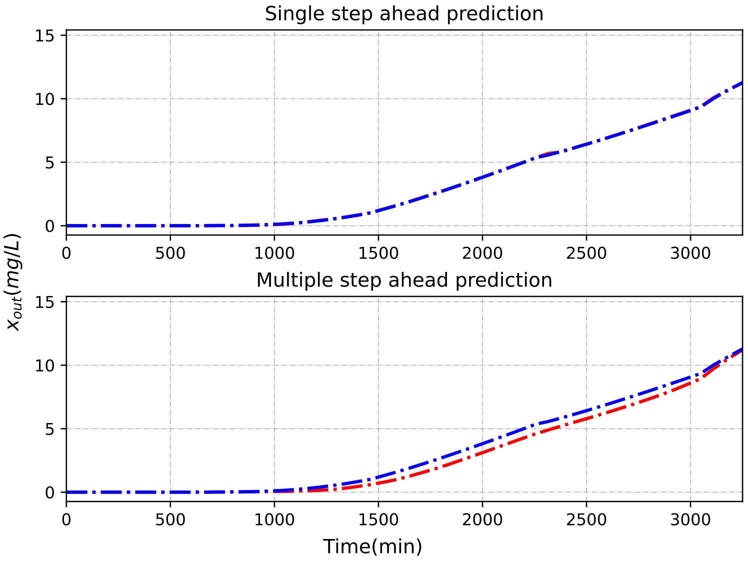

Figure 4.3: ReLU model performance.

optimization problem can then be represented as:

$$\min_{u_i^d} \sum_{i=k}^{k+N-1} \left(W_s y_{i+1} + W_d u_i^d\right) \tag{4.6a}$$

$$\text{s.t} \quad y_{i+1} = f_{NN}(y_i, y_{i-1}, u_i^c, u_{i-1}^c)[1 - u_i^d]; i = k, .., k+N-1 \tag{4.6b}$$

$$y_k = y(k) \tag{4.6c}$$

$$u_i^d \in \{0,1\}; i = k, .., k+N-1 \tag{4.6d}$$

where y_{i+1} is the output of the neural network, and f_{NN} is the neural network.

4.3 Reinforcement Learning Approach

RL represents a class of data-driven learning algorithms where an agent learns a closed-loop policy $\pi(u|x)$ by interacting with the environment. The environment is formulated as a Markov decision process (MDP), in which the agent's actions influence the system's state and result in rewards. A typical data tuple required by RL consists of the current state x_k, the action prescribed by the agent u_k, the resulting reward r_{k+1}, and the next state x_{k+1}.

The state transition dynamics of the MDP are represented by the conditional probability $P(r_{k+1}, x_{k+1}|x_k, u_k)$. In this work, the transition dynamics are described by the nonlinear system given by Equation (2.46).

Following [70], the RL problem can be formulated as finding the optimal policy π^* that maximizes the expected return G_k given the current state x_k and action u_k:

$$\pi^* = \arg\max_{\pi} \quad \mathbb{E}_{\pi}[G_k|x_k, u_k] \tag{4.7}$$

where G_k represents the accumulated reward r. In the context of mAb switching operation, the reward r is defined in Equation (4.8), following the same idea of the optimization problem in Equation (4.1a).

$$r_k = W_s x_{out,k} + W_d u_k^d \tag{4.8}$$

4.4 Simulation Results

4.4.1 Comparison of All Approaches

In this section, we compare the simulation results of the four different approaches. A prediction horizon of 20 was used in the simulations involving EMPC (sigmoid and ReLU). Additionally, a step size of 60 minutes and 50 simulation steps were used across all cases, and the weights W_s and W_d were chosen as 1 and 0.5 respectively. For RL, the Proximal Policy Optimization (PPO) was selected as the agent. The design of the agent is summarized in Table 4.1. Due to the stochasticity nature of the RL training process, 20 RLs were trained with the same parameters, and the one with the best performance was reported in the rest of the work.

The results for all four cases are presented in Table 4.2, where Time$_{sim}$ represents simulation time, $\mathcal{PL}$ represents product loss, and $\mathcal{TC}$ represents total cost. Among the optimization approaches, the sigmoid approach had the longest simulation time, taking 6 hours to com-

Parameters	Values
Episode	10,000
Steps per episode	200
Batch size	10
Learning rate	1e-4
Discounted factor	0.99

Table 4.1: RL agent design parameters.

Method	Time$_{sim}$	$\mathcal{PL}(mg/L)$	$\mathcal{TC}$	Switches
Sigmoid	6 h	0.0145	2.01	4
ReLU	1 h, 9 m	0.2421	1.74	3
RL	1.0713 s	0.3472	1.63	2
Traditional	0.1718 s	2.7161	3.71	2

Table 4.2: Comparison of all four approaches.

plete the closed-loop simulation of 50 steps. On the other hand, the traditional approach took only 0.1718 seconds because it involves no optimization. Furthermore, since the RL policy was obtained offline and directly implemented online, there was no optimization conducted online. As a result, the simulation time was significantly reduced, taking only 1.07 seconds to complete.

In terms of product loss, the sigmoid approach saved more product at the column outlet compared to the ReLU and RL approaches. However, this came at the expense of higher cost, as the sigmoid approach proposed more switching actions. The ReLU approach, while saving less product than the sigmoid approach, still outperformed the RL approach in terms of product loss. However in terms of cost, the RL approach shows the best results since it exhibited the lowest cost among the three proposed approaches. Overall, it is evident that the proposed approaches outperformed the traditional approach since they provided better results in terms of product loss and cost optimization.

Moving forward, we decided to focus on evaluating the ReLU and RL approaches in more detail due to their significantly faster simulation times compared to the sigmoid approach. This will allow us to explore the impact of various factors on the simulation results more efficiently. First, the effect of different EMPC prediction horizons is investigated for the

Horizon	Time_{sim}	$\mathcal{PL}(mg/L)$	$\mathcal{TC}$	Switches	Steps
1	2.14 m	1.57692	2.52833	2	18,36
3	3.76 m	0.55275	2.05275	3	16,32,48
4	4.54 m	0.55275	2.05275	3	16,32,48
5	5.11 m	0.24205	1.74205	3	15,30,45
10	9.60 m	0.24205	1.74205	3	15,30,45
20	1h, 9 m	0.24205	1.74205	3	15,30,45
30	2 days, 12h	0.24205	1.74205	3	15,30,45

Table 4.3: Effect of different horizons.

ReLU approach and analyzed in Table 4.3. The subsequent simulation results reported in Subsection 4.4.2 are based on 50 simulation steps.

4.4.2 Effect of Different Factors on Simulation Results - ReLU

4.4.2.1 Impact of Different Prediction Horizons

The effect of different horizons on the simulation results can be analysed using Table 4.3. The results indicate that a smaller horizon leads to delayed switching, resulting in higher product loss and increased cost. On the other hand, as the horizon value increases, the controller gains the ability to look further into the future, enabling quicker decision-making. Consequently, this facilitates early switching, leading to reduced product loss and lower overall cost. For instance, for a horizon of 3, switching occurs first at the $16th$ step, whereas for a horizon of 5, the first switch occurs at the $15th$ step.

After analyzing the results, a horizon value of 10 was selected for subsequent simulations. This value strikes a balance between minimizing product loss and cost while still allowing the controller to make timely decisions based on a reasonable forecasting window. By optimizing the horizon value, the ReLU approach can achieve more efficient control and better overall performance in terms of product loss and cost optimization.

W_s	W_d	Time$_{sim}$	$\mathcal{PL}(mg/L)$	Switches	Steps
0.9	0.1	11.27 m	0.09631	3	14,28,42
0.7	0.3	9.68 m	0.24205	3	15,30,45
0.5	0.5	9.60 m	0.55275	3	16,32,48
0.3	0.7	11.11 m	1.15787	2	17,34
0.1	0.9	11.21 m	1.15787	2	17,34

Table 4.4: Effect of weights W_d and W_s.

4.4.2.2 Impact of Different Weights

The impact of varying the weights, W_s and W_d, was investigated to assess their influence on the simulation results. A prediction horizon of 10 was used for the simulations, and the results were presented in Table 4.4, displaying the variations in simulation time, product loss, number of switches, and switching steps for different weight combinations.

A smaller switching penalty, represented by a smaller value of W_d, allows for more frequent switching. Consequently, when the penalty is smaller, switching commences at an earlier step, resulting in reduced product loss. As the weights differ in each case, a direct comparison of the cost is not feasible. Instead, we can focus on comparing the effects of the weights on switching and product loss. Generally, it is observed from Table 4.4 that a higher value of W_s and a lower value of W_d result in a smaller number of switches and minimal product loss. In contrast, a lower value of W_s and a higher value of W_d lead to a reduction in the number of switches, but with an associated increase in product loss.

Moving forward, the weights W_s and W_d were selected as 0.7 and 0.3, respectively, based on a careful consideration of their impact on the optimization process. A weight of 0.7 was assigned to W_s, which controls the penalty for product loss. This weight was chosen to be relatively high, indicating the importance of minimizing product loss in the system. By placing a greater emphasis on reducing product loss, the optimization algorithm is incentivized to make decisions that prioritize maintaining product quality.

Furthermore, the weight on W_d was set to 0.3, indicating the penalty associated with switching actions. This weight was selected to strike a balance between minimizing unneces-

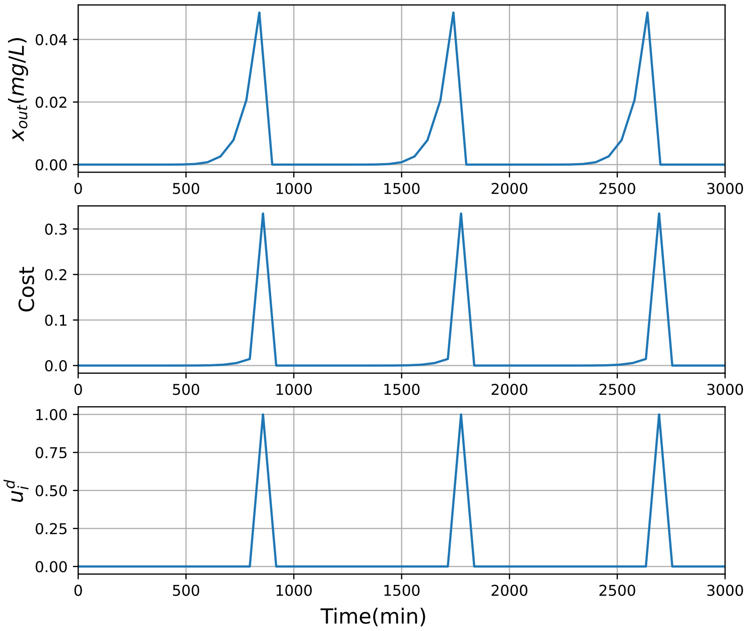

Figure 4.4: Plots of x_{out}, cost and u_i^d for selected weights.

sary switches and optimizing the utilization of system resources. With a moderate penalty for switching, the algorithm is encouraged to find an efficient trade-off between the number of switches and overall system performance.

By assigning weights of 0.7 and 0.3 to W_s and W_d respectively, the chosen values reflect a strategic approach to address the dual objectives of minimizing product loss while maximizing resource efficiency. It is expected that the selected weights will facilitate effective decision-making, leading to improved performance and cost-effectiveness in subsequent simulations and practical implementations. Plots of the cost, u_i^d and x_{out} for the selected weights can be found in Figure 4.4.

4.4.2.3 Impact of Process Noise

In this section, the impact of process noise on the system performance is investigated by adding randomly-generated noise of 0 mean and different standard deviations to the two constant continuous inputs, u_i^c, defined in Equation (2.45c). The simulations are conducted using a horizon of 10, and weights of 0.7 and 0.3 for W_s and W_d respectively. The results

Noise$_{std}$	$\mathcal{PL}_{ave}(mg/L)$	$\mathcal{TC}_{ave}$	Steps
2	0.19261	1.03482	14,28,42 15,29,44
3	0.22476	1.05733	16,31,46 14,29,44 15,29,44
4	0.22320	1.05624	16,31,47 13,28,43 15,30,45 15,29,44
5	0.21808	1.05264	16,32,47 15,30,45 15,29,44 14,27,42

Table 4.5: Effect of process noise.

of the simulations under the different levels of noise are summarized in Table 4.5, where Noise$_{std}$ represents the standard deviation of the noise used, $\mathcal{PL}_{ave}$ denotes the average product lost, and $\mathcal{TC}_{ave}$ represents the average total cost. For each standard deviation of noise, five simulations were run, and the average product lost and average total cost were calculated and recorded in the table.

In the presence of process noise, the switching behavior is not as perfectly periodic as it is in the noise-free cases reported in Tables 4.3 - 4.4, and switching may occur a step earlier or later than anticipated. The noise introduces some level of variability in the system, resulting in fluctuations in the total product lost and total cost. The analysis of the impact of process noise provides valuable insights into the robustness and performance of the system under realistic operating conditions. It highlights the need for appropriate control strategies to mitigate the effects of noise and maintain desired performance levels.

4.4.2.4 Impact of Measurement Noise

To analyse the effect of measurement noise, randomly generated noise of zero mean, normal distribution and varying standard deviations was added to the outlet mAb concentration, x_{out}. The simulation results under different levels of measurement noise are summarized in

α	Noise$_{std}$	$\mathcal{PL}_{ave}(mg/L)$	$\mathcal{TC}_{ave}$	Steps
10^{-4}	2	0.242053	1.742053	15,30,45
10^{-4}	4	0.242053	1.203960	15,30,45
10^{-4}	5	0.242053	1.069437	15,30,45
10^{-4}	6	0.242053	1.069437	15,30,45
10^{-4}	10	0.242053	1.069437	15,30,45
10^{-3}	10	0.110225	0.977039	14,28,42 14,28,40 15,29,43
10^{-3}	12	0.096796	1.26775	6,16,30,40,50 15,18,32,46 14,17,32,43 14,28,42

Table 4.6: Effect of measurement noise.

Table 4.6, where α represents the multiplication factor used to adjust the intensity of the noise level.

As the standard deviation of the measurement noise increases, the system exhibits robustness in maintaining a periodic switching pattern, similar to the noise-free case. The average product lost remains relatively constant, indicating the controller's ability to adapt to the measurement noise, nevertheless, slight variations in the total cost are observed due to the influence of the noise.

As the noise level is further increased by adjusting α, however, it can be observed that the switching pattern becomes less periodic. This suggests that the controller's performance may be impacted by higher levels of measurement noise, leading to deviations from the expected behavior.

4.4.3 Effect of Different Factors on Simulation Results - RL

In this section, the same study was conducted to examine the effects of different weights, process noises, and measurement noises on the RL controller. Because the RL design did not involve the prediction horizon, we decided not to conduct a study on it.

W_s	W_d	Time$_{sim}$	$\mathcal{PL}(mg/L)$	Switches	Steps
0.9	0.1	0.87 s	0.096	3	14,28,42
0.7	0.3	0.87 s	0.24	3	15,30,45
0.5	0.5	0.81 s	1.58	2	18,36
0.3	0.7	0.84 s	2.72	2	19,38
0.1	0.9	0.81 s	11.25	2	22,44

Table 4.7: Effect of weights W_d and W_s for RL.

4.4.3.1 Impact of Different Weights

Using the same experimental design applied to ReLU with weights W_s and W_d, we investigated the effect of five different weight pairs. The results are summarized in Table 4.7. Similar to the ReLU approach, an increase in the value of W_d resulted in a greater penalty on the switching action, leading to a decreasing trend in the number of switches. For the last three cases, the number of switches was the same. However, as W_d increased from 0.5 to 0.9, the first switch happened at a later time step, from the 18^{th} to the 22^{nd}. Consequently, there were more production losses as the mAb escaped from the capture column outlet due to fewer and delayed switches. Also, we can observe from the last column that the switching was periodic because there was no disturbance considered. Moving forward, the experiments are conducted with W_s and W_d set to 0.7 and 0.3, respectively.

4.4.3.2 Impact of Process Noise

For the capture column, the inlet flow rate and concentration of mAb from the buffer tank are identified as disturbances, and the noises were simulated with a mean of 0 and a standard deviation ranging from 2 to 4. For each noise level, 5 simulations were conducted, and the average production loss and total cost are reported in Table 4.8. With noisier disturbances, both the production loss and total cost increased, indicating that the performance of the trained RL controller was affected by the presented disturbances. The switching steps are summarized in the last column of the table, and it shows that the periodic pattern in Table 4.7 is not preserved anymore due to the stochasticity. With a higher noise standard deviation,

Noise_{std}	$\mathcal{PL}_{ave}(mg/L)$	$\mathcal{TC}_{ave}$	Steps
2	0.39	1.69	15,30,45 15,30,47 15,31,46 15,31,47 16,31,46
3	0.40	1.70	15,30,44 15,31,47 16,29,44 16,31,48 16,32,47
4	0.63	1.71	15,30,44 15,31,47 16,29,44 16,31,48 18,36
5	0.71	1.75	15,30,44 15,32,49 16,29,44 16,31,48 18,36

Table 4.8: Effect of process noise for RL.

the first step to switch happened at or after the 15^{th} step, resulting in more production loss.

4.4.3.3 Impact of Measurement Noise

The study of the impact of measurement noise on RL performance was conducted, and the experimental setup and results are summarized in Table 4.9. Interestingly, given that the magnitude of α is higher than that in the EMPC case, the results for the 3 cases are the same as the deterministic case. This implies that the effect of the measurement noise is negligible with the given α and standard deviation. One possible reason is that the policy obtained by RL takes the whole state space as the input. When only the measured state, x_{out}, is subjected to the measurement noise, the modification to the state space is minimal. On the contrary, EMPC, as expressed in Equation (4.6), explicitly considers x_{out} and hence will be more responsive to the measurement noise.

α	Noise$_{std}$	$\mathcal{PL}_{ave}(mg/L)$	$\mathcal{TC}_{ave}$	Steps
1	2	0.24	1.62	15,30,45
1	3	0.24	1.62	15,30,45
1	4	0.24	1.62	15,30,45

Table 4.9: Effect of measurement noise for RL.

4.5 Summary

The analysis of the simulation results from the different approaches provides valuable insights into the performance and effectiveness of each approach. The proposed approaches outperformed the traditional approach in terms of product loss and cost optimization. The sigmoid approach, while saving more product at the column outlet, resulted in higher cost and took a significantly longer time for computation. On the other hand, the ReLU approach showed a balance between product loss and cost, outperforming the RL approach in terms of product loss. However, the RL approach exhibited the best results with the lowest overall cost.

The impact of different factors such as prediction horizon, noise, and the weights on product loss and switching action on the simulation results were investigated on the ReLU, revealing that a smaller horizon led to delayed switching, increased product loss, and higher cost. Increasing the horizon allowed for earlier switching, resulting in reduced product loss and lower cost. Based on these findings, a horizon value of 10 was selected as a suitable compromise between minimizing product loss and cost while maintaining timely decision-making.

The impact of varying the weights, W_s and W_d, was thoroughly examined to understand their influence on the simulation results. The analysis revealed that the selection of weights significantly affects the system behavior, particularly in terms of the number of switches and product loss. A smaller W_d allows for more frequent switching, resulting in reduced product loss. Conversely, higher values of W_s and lower values of W_d lead to a smaller number of switches and minimal product loss. Considering these findings, the weights W_s and W_d

were carefully chosen as 0.7 and 0.3, respectively. A weight of 0.7 for W_s emphasizes the importance of minimizing product loss. On the other hand, a weight of 0.3 for W_d strikes a balance between minimizing unnecessary switches and optimizing resource utilization. This approach ensures efficient decision-making that aligns with the dual objectives of minimizing product loss and maximizing resource efficiency.

The effect of process noise was also analyzed, indicating that the presence of noise disrupted the periodic switching behavior observed in the noise-free cases, with fluctuations in product loss and cost due to the influence of the noise.

Additionally, the impact of measurement noise was examined, showing that the system exhibited robustness in maintaining a periodic switching pattern even under higher levels of measurement noise. The average product loss remained relatively constant, highlighting the controller's ability to adapt to the noise. However, at much higher noise levels for the ReLU case, deviations from the expected behavior were observed, indicating potential challenges in the controller's performance.

For the studies on RL, a similar observation was made while varying W_s and W_d as in the ReLU case. Regarding the impact of process noise, worse economic performance was observed as higher process noise was introduced to the system. However, in the case of measurement noise, since the noise was applied to only one state out of the full state space, the RL agent did not detect it, resulting in the same results as the deterministic case. By understanding these factors and their impact, more efficient control strategies can be developed to enhance the system's performance and robustness.

In conclusion, the proposed approaches demonstrate improved performance compared to the traditional approach. The analysis of different factors provides valuable insights for optimizing control strategies and addressing challenges such as process and measurement noise. This research contributes to the development of effective control methods in the context of the studied system, with potential applications in various industries.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

The combined findings from both the estimation and control works provide valuable insights into enhancing the overall performance and understanding of the system under consideration. In the estimation work, Case 3 emerges as the superior approach, demonstrating the importance of including parameter estimation alongside state estimation. By leveraging a selective approach that considers both variables, more accurate and reliable estimation results can be achieved, leading to improved system understanding and control.

In the control work, the proposed approaches outperform the traditional method, highlighting their effectiveness in minimizing product loss and optimizing costs. The sigmoid approach, while efficient in saving product, incurs higher costs and longer computation times. The ReLU approach strikes a balance between product loss and cost, outperforming the RL approach in terms of product loss. However, the RL approach exhibits the best results with the lowest overall cost. The impact of factors such as prediction horizon, noise, and weights on product loss and switching action were thoroughly investigated, providing critical insights for developing efficient control strategies.

5.2 Future Work

- *Consideration of cost in sensor selection process:* The sensor selection process can be further enhanced by incorporating cost considerations. Currently, the sensor selection focuses on optimizing the observability of system states. However, in real-world applications, the cost of sensors can be a critical factor. Introducing cost constraints into the sensor selection problem adds complexity but can lead to more practical and cost-effective solutions. This area merits further study to develop sensor selection strategies that strike a balance between observability and affordability.

- *State estimation of downstream process:* The downstream process model involves a larger number of states due to spatial discretization. Estimating this significant number of states accurately can be challenging and requires robust estimation techniques. Future research can focus on developing advanced state estimation algorithms that can handle the increased complexity of the downstream process.

- *Experimental validation and implementation:* The proposed control and estimation strategies can be experimentally validated in real mAb production systems. Implementing these techniques in actual industrial settings will provide valuable insights into their effectiveness and practicality. Collaborating with industry partners for real-world validation can bridge the gap between research and industrial applications, leading to impactful advancements in mAb production processes.

Bibliography

[1] Cleo Kontoravdi, Steven P Asprey, Efstratios N Pistikopoulos, and Athanasios Mantalaris. Application of global sensitivity analysis to determine goals for design of experiments: an example study on antibody-producing cell cultures. *Biotechnology progress*, 21(4):1128–1135, 2005.

[2] 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.

[3] Joaquín Gomis-Fons, Hubert Schwarz, Liang Zhang, Niklas Andersson, Bernt Nilsson, Andreas Castan, Anita Solbrand, Joanne Stevenson, and Véronique Chotteau. Model-based design and control of a small-scale integrated continuous end-to-end mab platform. *Biotechnology progress*, 36(4):e2995, 2020.

[4] Mohan Zhang, Xiaozhou Wang, Benjamin Decardi-Nelson, Bo Song, An Zhang, Jinfeng Liu, Sile Tao, Jiayi Cheng, Xiaohong Liu, DengDeng Yu, et al. Smpl: Simulated industrial manufacturing and process control learning environments. *arXiv preprint arXiv:2206.08851*, 2022.

[5] Naeimeh Faraji Dizaji. *Minor Whey Protein Purification Using Ion-Exchange Column Chromatography.* PhD thesis, The University of Western Ontario (Canada), 2016.

[6] Elizabeth C Lloyd, Tejal N Gandhi, and Lindsay A Petty. Monoclonal antibodies for covid-19. *JAMA*, 325(10):1015–1015, 2021.

[7] FC Breedveld. Therapeutic monoclonal antibodies. *The Lancet*, 355(9205):735–740, 2000.

[8] Ron J Keizer, Alwin DR Huitema, Jan HM Schellens, and Jos H Beijnen. Clinical pharmacokinetics of therapeutic monoclonal antibodies. *Clinical pharmacokinetics*, 49:493–507, 2010.

[9] Maria Elisa Rodrigues, Ana Rita Costa, Mariana Henriques, Joana Azeredo, and Rosário Oliveira. Technological progresses in monoclonal antibody production systems. *Biotechnology Progress*, 26(2):332–351, 2010.

[10] Suzanne S Farid. Process economics of industrial monoclonal antibody manufacture. *Journal of Chromatography B*, 848(1):8–18, 2007.

[11] Keith Plumb. Continuous processing in the pharmaceutical industry: changing the mind set. *Chemical Engineering Research and Design*, 83(6):730–738, 2005.

[12] 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.

[13] Garima Thakur, Saxena Nikita, Anamika Tiwari, and Anurag S Rathore. Control of surge tanks for continuous manufacturing of monoclonal antibodies. *Biotechnology and bioengineering*, 118(5):1913–1931, 2021.

[14] Stephen Craven, Jessica Whelan, and Brian Glennon. Glucose concentration control of a fed-batch mammalian cell bioprocess using a nonlinear model predictive controller. *Journal of Process Control*, 24(4):344–357, 2014.

[15] João A Lopes, Paula F Costa, Teresa P Alves, and José C Menezes. Chemometrics in bioprocess engineering: process analytical technology (pat) applications. *Chemometrics and Intelligent Laboratory Systems*, 74(2):269–275, 2004.

[16] Omar Khan, Chandra Mouli R Madhuranthakam, Peter Douglas, Heron Lau, Jacob Sun, and Patrick Farrell. Optimized pid controller for an industrial biological fermentation process. *Journal of process control*, 71:75–89, 2018.

[17] Aparajith Bhaskar, Fernando N Barros, and Ravendra Singh. Development and implementation of an advanced model predictive control system into continuous pharmaceutical tablet compaction process. *International journal of pharmaceutics*, 534(1-2):159–178, 2017.

[18] Ravendra Singh, Marianthi Ierapetritou, and Rohit Ramachandran. System-wide hybrid mpc–pid control of a continuous pharmaceutical tablet manufacturing process via direct compaction. *European Journal of Pharmaceutics and Biopharmaceutics*, 85(3):1164–1182, 2013.

[19] Samardeep Sarna, Nikesh Patel, Brandon Corbett, Chris McCready, and Prashant Mhaskar. Process-aware data-driven modelling and model predictive control of bioreactor for the production of monoclonal antibodies. *The Canadian Journal of Chemical Engineering*, 101(5):2677–2692, 2023.

[20] PA Montgomery and D Williams. On-line estimation of cell mass using an extended kalman filter. *Computer Applications in Fermentation Technology: Modelling and Control of Biotechnological Processes*, pages 221–225, 1989.

[21] Hiroshi Shimizu, Takeichiro Takamatsu, Suteaki Shioya, and Ken-Ichi Suga. An algorithmic approach to constructing the on-line estimation system for the specific growth rate. *Biotechnology and Bioengineering*, 33(3):354–364, 1989.

[22] Frank-Jan JI Nagel, Johannes Tramper, Marjolein SN Bakker, and Arjen Rinzema. Temperature control in a continuously mixed bioreactor for solid-state fermentation. *Biotechnology and Bioengineering*, 72(2):219–230, 2001.

[23] Khaireddine Zarai and Adnane Cherif. Adaptive filter based on monte carlo method to improve the non-linear target tracking in the radar system. *Aerospace Systems*, 4(1):67–74, 2021.

[24] Andrea Tuveri, Fernando Perez-Garcia, Pedro A Lira-Parada, Lars Imsland, and Nadav Bar. Sensor fusion based on extended and unscented kalman filter for bioprocess monitoring. *Journal of process control*, 106:195–207, 2021.

[25] Christopher V Rao and James B Rawlings. Constrained process monitoring: Moving-horizon approach. *AIChE journal*, 48(1):97–109, 2002.

[26] Siyu Liu, Xunyuan Yin, Zhichao Pan, and Jinfeng Liu. A sensitivity-based approach to optimal sensor selection for process networks. *arXiv preprint arXiv:2208.00584*, 2022.

[27] Dan Selişteanu, Dorin Şendrescu, Vlad Georgeanu, and Monica Roman. Mammalian cell culture process for monoclonal antibody production: nonlinear modelling and parameter estimation. *BioMed research international*, 2015, 2015.

[28] Mudita Juneja and SK Nagar. Particle swarm optimization algorithm and its parameters: A review. In *2016 International Conference on Control, Computing, Communication and Materials (ICCCCM)*, pages 1–5. IEEE, 2016.

[29] Riccardo Poli, James Kennedy, and Tim Blackwell. Particle swarm optimization: An overview. *Swarm intelligence*, 1:33–57, 2007.

[30] Jianbang Liu, Aristarchus Gnanasekar, Yi Zhang, Song Bo, Jinfeng Liu, Jingtao Hu, and Tao Zou. Simultaneous state and parameter estimation: the role of sensitivity analysis. *Industrial & Engineering Chemistry Research*, 60(7):2971–2982, 2021.

[31] Fabian Steinebach, Monica Angarita, Daniel J Karst, Thomas Müller-Späth, and Massimo Morbidelli. Model based adaptive control of a continuous capture process for monoclonal antibodies production. *Journal of Chromatography A*, 1444:50–56, 2016.

[32] Ziyang Zhang, Marco Mazzotti, and Massimo Morbidelli. Multiobjective optimization of simulated moving bed and varicol processes using a genetic algorithm. *Journal of chromatography A*, 989(1):95–108, 2003.

[33] Daniel Baur, Monica Angarita, Thomas Müller-Späth, and Massimo Morbidelli. Optimal model-based design of the twin-column capturesmb process improves capacity utilization and productivity in protein a affinity capture. *Biotechnology journal*, 11(1):135–145, 2016.

[34] Veena Warikoo, Rahul Godawat, Kevin Brower, Sujit Jain, Daniel Cummings, Elizabeth Simons, Timothy Johnson, Jason Walther, Marcella Yu, Benjamin Wright, et al. Integrated continuous production of recombinant therapeutic proteins. *Biotechnology and bioengineering*, 109(12):3018–3029, 2012.

[35] Audun Faanes and Sigurd Skogestad. Buffer tank design for acceptable control performance. *Industrial & engineering chemistry research*, 42(10):2198–2208, 2003.

[36] Maria M Papathanasiou, Ana L Quiroga-Campano, Fabian Steinebach, Montaña Elviro, Athanasios Mantalaris, and Efstratios N Pistikopoulos. Advanced model-based control strategies for the intensification of upstream and downstream processing in mab production. *Biotechnology progress*, 33(4):966–988, 2017.

[37] Ioscani Jimenez del Val, Yuzhou Fan, and Dietmar Weilguny. Dynamics of immature mab glycoform secretion during cho cell culture: An integrated modelling framework. *Biotechnology journal*, 11(5):610–623, 2016.

[38] Thomas K Villiger, Ernesto Scibona, Matthieu Stettler, Hervé Broly, Massimo Mor-

bidelli, and Miroslav Soos. Controlling the time evolution of mab n-linked glycosylation-part ii: Model-based predictions. *Biotechnology progress*, 32(5):1135–1148, 2016.

[39] Marie-Françoise Clincke, Carin Mölleryd, Puneeth K Samani, Eva Lindskog, Eric Fäldt, Kieron Walsh, and Véronique Chotteau. Very high density of chinese hamster ovary cells in perfusion by alternating tangential flow or tangential flow filtration in wave bioreactor™—part ii: Applications for antibody production and cryopreservation. *Biotechnology progress*, 29(3):768–777, 2013.

[40] Ernie X Perez-Almodovar and Giorgio Carta. Igg adsorption on a new protein a adsorbent based on macroporous hydrophilic polymers. i. adsorption equilibrium and kinetics. *Journal of Chromatography A*, 1216(47):8339–8347, 2009.

[41] Sergio Contreras, Pushkin Kachroo, and Shaurya Agarwal. Observability and sensor placement problem on highway segments: A traffic dynamics-based approach. *IEEE Transactions on Intelligent Transportation Systems*, 17(3):848–858, 2015.

[42] M Naeem, S Xue, and DC Lee. Cross-entropy optimization for sensor selection problems. In *2009 9th International Symposium on Communications and Information Technology*, pages 396–401. IEEE, 2009.

[43] D Joubert, JD Stigter, and J Molenaar. An efficient procedure to assist in the re-parametrization of structurally unidentifiable models. *Mathematical biosciences*, 323:108328, 2020.

[44] R Renneberg, G Trott-Kriegeskorte, M Lietz, V Jäger, M Pawlowa, G Kaiser, U Wollenberger, F Schubert, R Wagner, RD Schmid, et al. Enzyme sensor-fia-system for on-line monitoring of glucose, lactate and glutamine in animal cell cultures. *Journal of biotechnology*, 21(1-2):173–185, 1991.

[45] Feng Li, Natarajan Vijayasankaran, Amy Shen, Robert Kiss, and Ashraf Amanullah.

Cell culture processes for monoclonal antibody production. In *MAbs*, volume 2, pages 466–479. Taylor & Francis, 2010.

[46] Murali K Maruthamuthu, Scott R Rudge, Arezoo M Ardekani, Michael R Ladisch, and Mohit S Verma. Process analytical technologies and data analytics for the manufacture of monoclonal antibodies. *Trends in biotechnology*, 38(10):1169–1186, 2020.

[47] Salim Ibrir. Joint state and parameter estimation of non-linearly parameterized discrete-time nonlinear systems. *Automatica*, 97:226–233, 2018.

[48] Jonathan R Stroud, Matthias Katzfuss, and Christopher K Wikle. A bayesian adaptive ensemble kalman filter for sequential state and parameter estimation. *Monthly Weather Review*, 146(1):373–386, 2018.

[49] Costas Kravaris, Juergen Hahn, and Yunfei Chu. Advances and selected recent developments in state and parameter estimation. *Computers & chemical engineering*, 51:111–123, 2013.

[50] Frank Allgöwer, Thomas A Badgwell, Joe S Qin, James B Rawlings, and Steven J Wright. Nonlinear predictive control and moving horizon estimation—an introductory overview. *Advances in control: Highlights of ECC'99*, pages 391–449, 1999.

[51] Song Bo, Soumya R Sahoo, Xunyuan Yin, Jinfeng Liu, and Sirish L Shah. Parameter and state estimation of one-dimensional infiltration processes: A simultaneous approach. *Mathematics*, 8(1):134, 2020.

[52] Xunyuan Yin and Jinfeng Liu. Event-triggered state estimation of linear systems using moving horizon estimation. *IEEE Transactions on Control Systems Technology*, 29(2):901–909, 2020.

[53] Thomas F Edgar. Control and operations: when does controllability equal profitability? *Computers & chemical engineering*, 29(1):41–49, 2004.

[54] Margret Bauer and Ian K Craig. Economic assessment of advanced process control–a survey and framework. *Journal of process control*, 18(1):2–18, 2008.

[55] Nikolay Kyurkchiev and Svetoslav Markov. Sigmoid functions: some approximation and modelling aspects. *LAP LAMBERT Academic Publishing, Saarbrucken*, 4, 2015.

[56] Sagar Sharma, Simone Sharma, and Anidhya Athaiya. Activation functions in neural networks. *towards data science*, 6(12):310–316, 2017.

[57] Bernard T Agyeman, Soumya R Sahoo, Jinfeng Liu, and Sirish L Shah. Lstm-based model predictive control with discrete actuators for irrigation scheduling. *IFAC-PapersOnLine*, 55(7):334–339, 2022.

[58] Xingling Shao, Honglun Wang, Jun Liu, Jun Tang, Jie Li, Xiaoming Zhang, and Chong Shen. Sigmoid function based integral-derivative observer and application to autopilot design. *Mechanical Systems and Signal Processing*, 84:113–127, 2017.

[59] Laila Khairunnahar, Mohammad Abdul Hasib, Razib Hasan Bin Rezanur, Mohammad Rakibul Islam, and Md Kamal Hosain. Classification of malignant and benign tissue with logistic regression. *Informatics in Medicine Unlocked*, 16:100189, 2019.

[60] P Bonami, LT Biegler, AR Conn, G Cornuéjols, IE Grossmann, CD Laird, Lee J, a Lodi A, Margot F, and A Sawaya N, Wächter. An algorithmic framework for convex mixed integer nonlinear programs. *Discrete optimization*, pages 186–204, 2008.

[61] Andreas Wächter and Lorenz T Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical programming*, pages 25–57, 2006.

[62] Jinming Zou, Yi Han, and Sung-Sau So. Overview of artificial neural networks. *Artificial neural networks: methods and applications*, pages 14–22, 2009.

[63] Abdul-Muaizz Koray, Dung Bui, William Ampomah, Emmanuel Appiah Kubi, and Joshua Klumpenhower. Application of machine learning optimization workflow to improve oil recovery. In *SPE Oklahoma City Oil and Gas Symposium/Production and Operations Symposium*, page D021S003R001. SPE, 2023.

[64] Ting-Lan Chiu and Sung-Sau So. Genetic neural networks for functional approximation. *QSAR & Combinatorial Science*, 22(5):519–526, 2003.

[65] Andrinandrasana David Rasamoelina, Fouzia Adjailia, and Peter Sinčák. A review of activation function for artificial neural network. In *2020 IEEE 18th World Symposium on Applied Machine Intelligence and Informatics (SAMI)*, pages 281–286. IEEE, 2020.

[66] Francesco Ceccon, Jordan Jalving, Joshua Haddad, Alexander Thebelt, Calvin Tsay, Carl D Laird, and Ruth Misener. Omlt: Optimization & machine learning toolkit. *The Journal of Machine Learning Research*, 23(1):15829–15836, 2022.

[67] Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2023.

[68] Vincent Tjeng, Kai Xiao, and Russ Tedrake. Evaluating robustness of neural networks with mixed integer programming. *arXiv preprint arXiv:1711.07356*, 2017.

[69] Shaoru Chen, Mahyar Fazlyab, Manfred Morari, George J Pappas, and Victor M Preciado. Learning region of attraction for nonlinear systems. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pages 6477–6484. IEEE, 2021.

[70] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction.* MIT press, 2018.