TÉCNICO
LISBOA

Stochastic modelling of bioreactors for *Escherichia coli* fermentation with recombinant protein production

Mariana Assunção Albino

Thesis to obtain the Master of Science Degree in

Biological Engineering

Supervisors: Prof. Susana de Almeida Mendes Vinga Martins
Prof. Krist V. Gernaey

Examination Committee

Chairperson: Prof. Marília Clemente Velez Mateus
Supervisor: Prof. Susana de Almeida Mendes Vinga Martins
Member of the Committee: Prof. Nuno Gonçalo Pereira Mira

October 2022

Preface

The work presented in this thesis was performed at the Department of Chemical and Biochemical Engineering, Process and Systems Engineering Centre of Technical University of Denmark (Kongens Lygnby, Denmark), during the period February-July 2022, under the supervision of Prof. Krist V. Gernaey and Postdoc Carina Loureiro da Costa Lira Gargalo, within the frame of the Erasmus+ program. The thesis was co-supervised at Instituto Superior Técnico by Prof. Susana de Almeida Mendes Vinga Martins.

I declare that this document is an original work of my own authorship and that it fulfills all the requirements of the Code of Conduct and Good Practices of the Universidade de Lisboa.

Acknowledgments

I would like to start by thanking Carina Gargalo, for the all the guidance and supervision while at the same time supporting and giving me freedom in my choices. Always prepared to answer my questions and also to give me some words of reassurance when I had doubts and was the most stressed. To Krist Gernaey, thank you for helping me see the big picture and the great insights on writing this thesis. To Prof. Susana Vinga, thank you for advising me at a distance. Thank you Óscar Rubio and Anderson Isaza for helping get out of my biggest roadblock. And to João, thank you for always helping me with my code, even if you never really understood what I was doing.

I would like to thank my mom for always supporting me unconditionally and being there to listen. To my dad for always trusting my choices, even though you always thought I studied to much. To all my family for never doubting my capabilities.

A special thank you to my boyfriend for accompanying me throughout this thesis. Thank you for always being excited to hear about my day, whether it was good or amazingly bad. Can't stress enough how important that was.

To all my friends, thank you for being with me throughout this university years, could not have made it without you.

Abstract

The need to develop models capable of describing key processes in the biomanufacturing industry has been rising as this industry is in a rapid transition towards more automation and digitalisation. Fermentation for recombinant protein production is one key process, and as all biological processes, it is subject to the stochasticity of living beings. Because of this, it is important that these models take this stochasticity into account. Gillespie's algorithm is a method of doing so, with exact numerical calculations within the framework of the stochastic formulation without the need of solving the Chemical Master Equation (CME).

The aim of this project was to develop a model for fed-batch fermentation of *Escherichia coli* for recombinant protein production that was able to capture the uncertainty associated with this bioprocess. This model was developed for two case study proteins, Green Fluorescent Protein (GFP) and recombinant human Growth Hormone (rhGH).

This goal was achieved by using deterministic models available in the literature and adapting them to the Stochastic Simulation Algorithm (SSA). It was shown that the developed model was applicable and could describe well the production dynamics. By confronting the model with experimental data, some parameters were re-estimated using the ABC Rejection sampler method. In summary, it is possible to derive conclusions regarding the uncertainty associated with each species present in the model.

These results provide one more successful application of this algorithm, and stand as a foundation to expand and improve its use for industry relevant processes.

Keywords

Stochastic modelling; Gillespie algorithm; Recombinant protein production; *Escherichia coli*; Bioreactors

Resumo

A indústria biotecnológica encontra-se numa fase de transição, para uma maior automatização e digitalização. Por esta razão, há uma necessidade crescente de desenvolver modelos que descrevam os bioprocessos mais relevantes. A fermentação para a produção de proteínas recombinantes é um destes processos, e, como todos os processos biológicos, está sujeito à estocacidade característica dos processos químicos, metabólicos e fisiológicos dos seres vivos. Por esta razão, ao desenvolver estes modelos, é necessário que esta estocacidade seja tida em conta. Gillespie desenvolveu um algoritmo que permite fazê-lo, utilizando cálculos numéricos exatos dentro do enquadramento estocástico, sem a necessidade de resolver a *Chemical Master Equation (CME).*

Neste contexto, o objetivo desta tese foi o desenvolvimento de um modelo da fermentação *fed-bacth* de *Escherichia coli* para a produção de proteínas recombinantes, que fosse capaz de capturar a incerteza associada ao processo. Este modelo foi desenvolvido para duas proteínas, *Green Fluorescent Protein (GFP)* e *recombinant human Growth Hormone (rhGH).*

Recorreu-se a modelos determinísticos disponíveis na literatura, que foram adaptados para o *Stochastic Simulation Algorithm (SSA).* Mostrou-se que o modelo desenvolvido era aplicável no contexto desejado e descrevia adequadamente as dinâmicas de produção. Foi possível concluir acerca da incerteza associada a cada uma das espécies simuladas. Através do confronto do modelo com dados experimentais, foi ainda possível re-estimar alguns parâmetros recorrendo ao método *ABC Rejection Sampler.*

Os resultados obtidos ilustram mais um caso de aplicação bem-sucedida deste algoritmo e podem servir como base para expandir e melhorar o seu uso para processos industrialmente relevantes.

Palavras Chave

Modelamento estocástico; Algoritmo de Gillespie; Produção de proteínas recombinantes; *Escherichia coli*; Bioreactores

Contents

List of Figures

List of Tables

Acronyms

ABC	Approximate Bayesian Computation
CDW	cell dry weight
CME	Chemical Master Equation
GFP	Green Fluorescent Protein
GHD	growth hormone deficiency
GMP	Good manufacturing practice
hGH	Human Growth Hormone
ODEs	Ordinary Differential Equations
PBM	Population Balance Models
QSSA	Quasi-steady state assumption
rhGH	recombinant human Growth Hormone
SSA	Stochastic Simulation Algorithm

1

Introduction

Contents

1.1 Motivation

The concept of Digital Twins is now a hot topic in the bioprocessing industry [1]. Essentially, the first step to developing a digital twin is a model that can describe the process of interest well. Using mechanistic models is a helpful tool from process development to monitoring and control. They are a key part of making biomanufacturing more automated and digitalised [2].

Hand in hand with this, the relevance of this industry keeps growing. For example, fermentation processes and the production of recombinant proteins bring several advantages when producing therapeutics and food additives, among many other products. Regarding pharmaceuticals, these advantages are for example high effectiveness, less side effects and the capability of curing diseases instead of only curing symptoms. As well, there is an increasing number of diseases which can be treated with biopharmaceuticals [3]. Recombinant human growth hormone as a therapeutic agent is an example that could only start being used due to the development of recombinant protein production.

One of the disadvantages of biological processes when compared to chemical routes is that they are harder to control. No recipe guarantees that a process will run the same way every time, for example when talking about fermentation. Living organisms have inherent stochasticity which brings randomness and noise to said processes [4]. Due to this, it is important that when developing models, this uncertainty should be taken into account. Thus, different scenarios can be predicted *a priori*, and strategies can be developed to model, predict, and rectify such variations.

1.2 Objectives and Contributions

The initial goal of this work was the stochastic simulation of intracellular biochemical reactions and its applications to industry relevant cases of fed-batch fermentation processes. As it is described in the results chapter, there was a need to change strategies. So, the objective of this work was to develop a stochastic model of bioreactors for recombinant protein production, recurring instead to an unstructured model.

The desired output of this thesis was to validate the application of the Stochastic Simulation Algorithm (SSA) to recombinant protein production, which to the author's knowledge had not been done previously. It is thus expected to derive insights regarding the uncertainty of this process, how it propagates, and hence gain a better understanding of it.

That being said, the main outcome of this thesis was the validation of this model against deterministic simulations available in the literature. This concordance also attested that the assumptions made in the development of the propensity equations were valid in this scenario.

Moreover, the simulation results were confronted with experimental data. Parameter re-estimation using the Approximate Bayesian Computation (ABC) Rejection sampler was performed. It allowed to improve the model quality in the prediction of protein production trajectories.

1.3 Thesis Outline

This thesis is divided into 5 chapters, including the present introduction Chapter. In Chapter 2, the theoretical background necessary for the development and understanding of the work done is detailed. It starts by highlighting the different types and the advantages of bioreactor modelling. It moves on to the thorough explanation of the SSA, the assumptions behind it, the different steps in the algorithm and some situations where its application has been successful. Finally, there is a brief introduction to recombinant protein production in *E. coli* as well as to the two proteins of the case studies.

Chapter 3 details the methodology used for the development of the project. It includes all the equations that were taken from available models in the literature; adaptation of these to the stochastic simulation; detailed steps of the algorithm used. Finally, the method utilised for parameter estimation is explained.

Following, Chapter 4 presents all the relevant results obtained as well their discussion. At last, Chapter 5 highlights the conclusions that were possible to draw from the obtained results as well as the future perspectives for this topic.

2

Background

Contents

This chapter will concern the work that has been previously developed in the scope of this thesis. It is important for its understanding since it will as well detail the necessary theoretical bases. The most important section will be regarding the stochastic simulation algorithm which serves as the foundation for the developed models. It also includes important information on modelling of bioreactors, protein production in *E. coli* and a brief description of the proteins used as a case study.

2.1 Modelling of bioreactors

In the present time, the biomanufacturing industry is delving into the concept of Digital Twins and the benefits it can bring regarding process optimisation. Becoming more digitalised and automated is one of the requisites of Industry 4.0 [1]. This industry is still falling behind in the implementation of digitalisation and automated handling of data when compared to other industries, partly because the definition of digital twins is still not clear. So, in the biomanufacturing operations context, a Digital Twin can be defined as a comprehensive digital representation of a physical object that is capable of bidirectional communication with it. It is responsive to changes in the physical asset and can modify the behaviour of the real system [2].

In the domain of bioprocesses the term "Digital Twin" is still scarcely used, however, the same is not true for the term "model", which, in the end, is always the backbone of a digital twin [2]. So, accurate mathematical description of unit-operations can represent the digital model component of the digital twin. These are useful to design and optimise different processes. In the case of fermentation, the topic that concerns this thesis, the base for this model will be a description of the relationship between feed inputs and product outputs.

When concerning models, the first distinction must be made between **steady-state** and **dynamic** models. A steady-state model, is essentially a steady-state mass and energy balance of a process which employs equilibrium conditions to describe reactions or simple relations to elucidate the conversion of components [2]. These expressions are not time dependent and thus do not contain an accumulation term. They can be used mostly for a first optimisation in an early stage and uncertainty analysis. Their main use is in areas where kinetic information is not available making them the only viable possibility.

On the other hand, dynamic models pick up on the mass balances of the steady-state model, adding accumulation terms and system dynamics. They contain time-based derivative terms for variables of interest and can be used to determine optimal process parameters. These dynamic models can also be purely data-driven models which can be used for example for predicting the end time of a fermentation process [5]. Here lies the second distinction that should be made, **data-driven** and **mechanistic** models.

Data-driven models use parametric equations to compute process outputs (response variables) from inputs (predictor variables). These can identify correlations between critical process parameters and critical quality attributes while not focusing on mechanistic causalities that are governing the relationships [6]. The main advantage of these models is the automatic model assembly and reduced computational burden, which makes them suitable for real-time monitoring. However, their predictive capabilities are limited to the space where they were validated, constricting their use for bioprocess control and optimisation.

As for mechanistic models, they are mathematical descriptions of the current understanding of a current system. The model will be described by first-principle equations with parameters that have a physical meaning [7]. These are mass, heat and momentum balances with the addition of ,e.g., kinetic expressions to describe process dynamics, which are often empirical. If assuming a homogeneous reactor environment, mechanistic models can be classified in four different categories [8] illustrated in Figure 2.1.

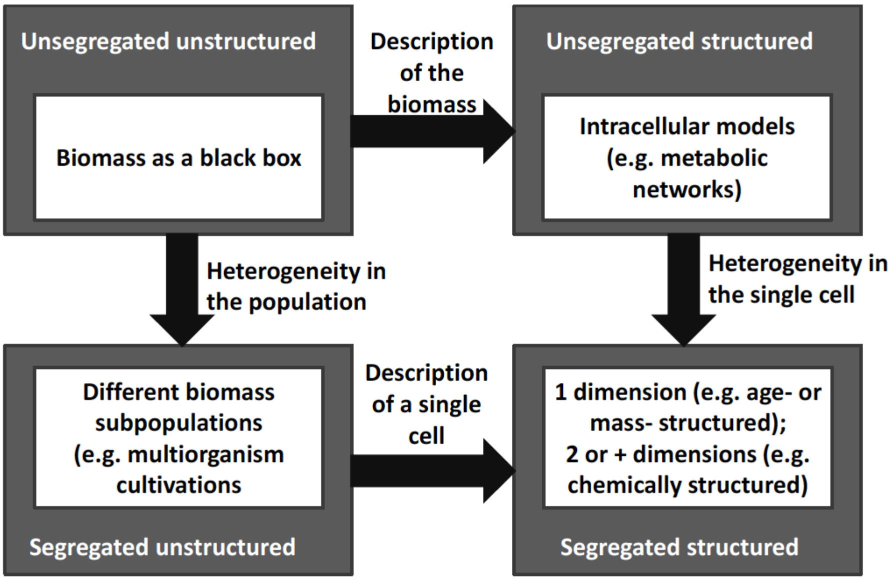
Unsegregated unstructured
Biomass as a black box
Description of the biomass
Unsegregated structured
Intracellular models (e.g. metabolic networks)
Heterogeneity in the population
Heterogeneity in the single cell
Different biomass subpopulations (e.g. multiorganism cultivations
Segregated unstructured
Description of a single cell
1 dimension (e.g. age- or mass- structured); 2 or + dimensions (e.g. chemically structured)
Segregated structured

Figure 2.1: Schematic classification of mechanistic models for fermentation processes [8].

The most common are **unsegregated** models which illustrate average cell description. Within these, the simplest and most popular are the **unstructured** models. They use a single variable to describe biomass, treating it as a black-box. Unsegregated **structured** models describe biomass as a group of different variables, e.g ATP, NADH, precursors and metabolites. These can be used to model complex processes such as intracellular metabolism [9].

Segregated models consider each cell individually accounting for the fact that in a population cells are different. These are usually formulated with Population Balance Models (PBM) [8]. Unstructured segregated models characterise cells by one distributed property (e.g. cell size or age) but without considering extracellular composition. Structured segregated models are more complex since the distribution of one or more intracellular variables is considered.

Having explained what are mechanistic models and the different classes, now the focus is brought back to how they can be applied and be useful in the context of Industry 4.0, as mentioned in the beginning of this section.

These mechanistic models require extensive time and resource investment when compared to data-driven models. However, since they are based on process understanding they can be applied to multiple processes by changing model parameters, making them highly valuable. They can also be used in several stages of process development and operation, as summarised in Figure 2.2 [7].

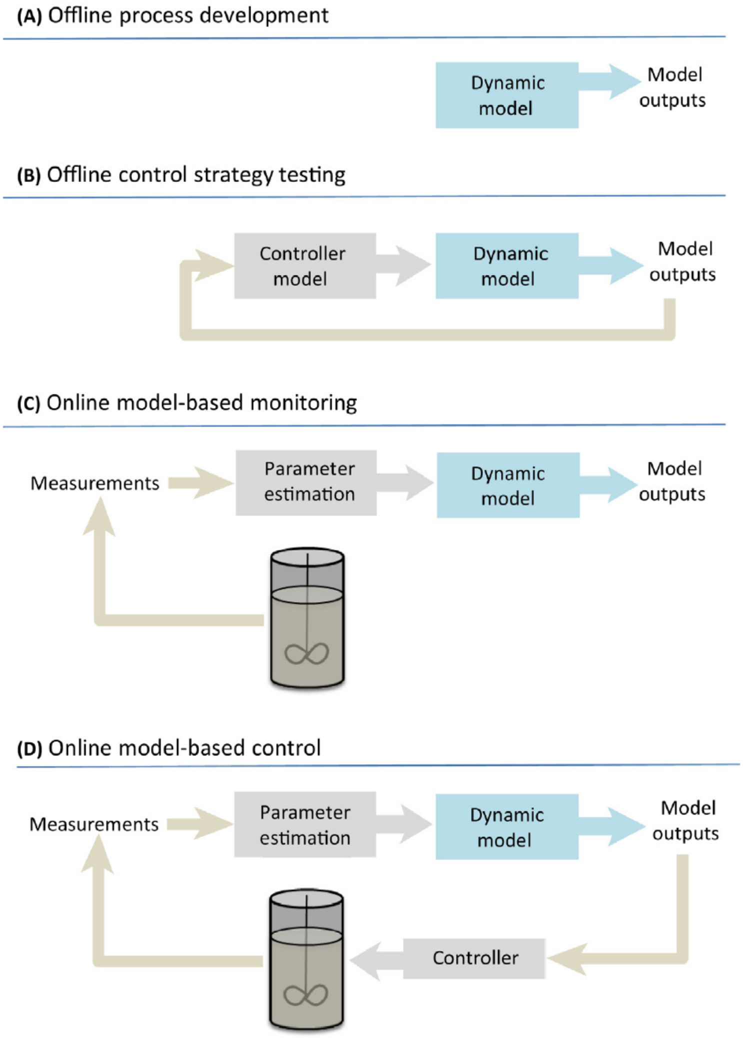

Figure 2.2: Mechanistic model structures for(A) offline process development, (B) offline control strategy testing, (C) online model-based monitoring, and(D) online model-based control [7].

Starting with offline development (Figure 2.2 (A)), the models can be used to asses the sensitivity of the process in different conditions, for example mass and heat transfer at different scales of operation. This is especially useful in scale-up studies, since it is critical to understand equipment limitations at different scales. So, these models are useful to determine what are the conditions that lead to a more profitable process. Mechanistic models are chosen at this stage because they can be extrapolated outside of the conditions for which they were developed (e.g, for different process parameters) and so can be used to test new equipment types or conditions for which there is no available data yet.

Regarding offline control strategy testing (Figure 2.2 (B)), a control algorithm can be coupled with the model to simulate different control strategies. This way, the control algorithm is implemented offline speeding up the online implementation that otherwise comes with considerable resource consumption for testing and tuning.

The biomanufacturing industry suffers from the lack of online sensors for key parameters, like substrate, biomass and product concentration. Some of the issues lie in the sterilisation of the probes as well as getting approval for changes in hardware according to Good manufacturing practice (GMP). As such, using mechanistic models for online based model-based monitoring (Figure 2.2 (C)), could be a good solution. The model can be implemented online, where the measured data is used to re-estimate parameters (online parameter estimation) and thus improving model prediction. This was done successfully by Mears et al. [10] for a case of industrially relevant filamentous fungus process at pilot scale, where the model was used to predict product concentration over time.

As for applying online model-based control (Figure 2.2 (D)), it allows for a flexible control objective, that can among others be glucose concentration control and avoid overflow metabolism. It also allows for control on key parameters that cannot be measured online. This works as available measurement data is fed to the model to update the predictions. The model output is used by the control algorithm.

To sum up this section, mechanistic models are highly useful for the biomanufacturing industry, in several stages of process development. However, most of these models are based on deterministic equations, and as such do not account for the stochasticity in the process. For this reason, the focus of this thesis was to create a mechanistic model that would take the inherent randomness of biological processes into account. For that purpose, the stochastic simulation algorithm was used, which is detailed in the following section.

2.2 Stochastic Simulation

Stochasticity is an inherent characteristic of multiple biochemical processes meaning that randomness and noise are an essential part of living beings. This is especially prevalent in processes where low copy numbers of chemical species have a big influence on the system dynamics and the best example of this is the regulation of gene expression [4]. The stochastic effects on biochemical processes are also influencing onset of disease and immune response. Given this, it is clear that models that are not deterministic, but that take this randomness into account are essential to better understand these biological phenomena.

Over recent decades considerable attention has been given to this problem and in 1977, Daniel Gillespie described the SSA, also known as the Gillespie algorithm. His motivation relied on the fact that the stochastic approach a firmer physical basis than the deterministic formulation. It describes the time behavior of a spatially homogeneous chemical system, regarding this process as a random-walk, which is governed by a single differential equation, the Chemical Master Equation (CME). This formulation is, however, often mathematically intractable, which leads the author of said work to develop a way to derive exact numerical calculations within the stochastic formulation framework, without the need to solve the CME. This framework uses a rigorously derived Monte Carlo procedure to numerically simulate the time evolution of the given system [11].

The general problem was formulated in the following way: 'If a fixed volume V contains a spatially uniform mixture of N chemical species which can interact through M specified chemical reaction channels, then given the number of molecules of each species present at some initial time, what will these molecular population levels be at any later time?'[11].

The deterministic solution would be to assume that the number of molecules of the i^{th} species in V at time t can be represented by a continuous function, $X_i(t)$ $(i = 1, ..., N)$, and that each of the chemical reactions M can be considered as a continuous rate process, then it is easy to construct a set of coupled first-order Ordinary Differential Equations (ODEs), where the specific functions for each reaction would be determined by the respective structure and rate constant. Even though this approach is of great importance and in many cases the chemical systems can be treated as deterministic and continuous, it must not be forgotten that the physical approach behind it is not the most solid. This is due to the fact that molecular population levels can only change by discrete integer amounts and time evolution itself is not a deterministic process. Moreover, since more and more attention is given to chemical reactions within biological processes, with the inability of the reaction-rate equations to describe the fluctuations in the molecular population levels, this becomes particularly relevant [11].

On the other hand, the stochastic solution takes into consideration the known fact that, in general, a chemical reaction occurs when two or more molecules of appropriate kinds collide in the right way, and that in a system of molecules in thermal equilibrium this happens essentially in a random manner. This argument supports the idea that chemical reactions are better described by a 'reaction probability per unit of time' instead of a 'reaction rate'. This way, considering the reaction described by Eq. 2.1, there will exist a c_1 constant which only depends on the physical properties of the two molecules and temperature of the system, such that c_1dt is the probability that the molecular pair $A - B$ will react according to R_1 in the next infinitesimal time interval dt.

$$R_1 : A + B \rightarrow AB. \tag{2.1}$$

So, in general, supposing that volume V contains a spatially homogeneous mixture of X_i molecules of chemical species S_i $(i = 1, ..., N)$ and that these species can react through M specified chemical reaction channels R_μ $(\mu = 1, ..., M)$, M constants c_μ $(\mu = 1, ..., M)$ exist depending only on the physical properties of the molecules and the temperature of the system, such that $c_\mu dt$ represents the average probability that a particular combination of R_μ reactant molecules will react accordingly in the next infinitesimal time interval dt.

It is important now to understand how this stochastic rate constant relates to the more well known reaction-rate constant, k_μ, since this will allow for the direct application of the latter in the Gillespie algorithm simulation. For first order reactions, both constants are equal and in the case of second order reactions, the stochastic rate constant is equal to the reaction-rate constant, divided by the volume of the reaction environment (Equation 2.2) [12].

$$c_\mu = \frac{k_\mu}{N_{Av} V}, \tag{2.2}$$

where N_{Av} is the Avogadro number. For second order reactions of two molecules of the same species, for example $A + A \rightarrow AA$, the stochastic rate constant would be calculated by Equation 2.3 [12].

$$c_\mu = \frac{2k_\mu}{N_{Av} V}. \tag{2.3}$$

This is because the number of distinct pairs of molecules that can collide is smaller in the second case. In the $A + B \rightarrow AB$ reaction, if X_A and X_B denote the number of molecules of A and B, the number of distinct molecular encounters is $X_A X_B$. As for the homodimer formation reaction $A + A \rightarrow AA$, the number of distinct molecular encounters is $\frac{X_A(X_A-1)}{2}$.

For reactions described by Michaelis-Menten dynamics and not by the Law of mass action, it is no longer possible to apply the equations described above. This was addressed by Rao and Arkin [13] who proposed a way to translate kinetic ODEs into propensities through the application of the Quasi-steady state assumption (QSSA). Using as an example the enzymatic reaction $\mathrm{E} + \mathrm{S} \underset{k_2}{\overset{k_1}{\rightleftarrows}} \mathrm{ES} \overset{k_3}{\rightarrow} \mathrm{E} + \mathrm{P}$, the ODE for the substrate would correspond to equation 2.4 [14]

$$\frac{dS}{dt} = -\frac{k_3 E_0 S}{S + K_{ES}}. \tag{2.4}$$

Recurring to the fact that the system volume is constant, concentrations can be replaced by the number of molecules through Equation 2.5.

$$C_i = N_i \left(\frac{S_0}{N_{S,0}} \right), \tag{2.5}$$

where S_0 and $N_{S,0}$ are the substrate's initial concentration and the corresponding number of molecules, respectively. Through this, Eq. 2.4 becomes Equation 2.6.

$$\frac{dN_S}{dt} = -k_3 \frac{N_{E,0} N_S}{N_S + K_{ES} \cdot (\frac{N_{S,0}}{S_0})}. \tag{2.6}$$

which would correspond to the following propensity function

$$a = -k_3 \frac{N_{E,0} N_S}{N_S + K_{ES} (\frac{N_{S,0}}{S_0})}. \tag{2.7}$$

Once the relationship between the reaction-rate constant and the stochastic rate constant, the attention is brought back to the stochastic description of reactions.

The number of distinct reactant combinations available for reaction R_μ is denoted by h_μ

Considering the definitions given above, the algorithm proceeds as follows: In each step of the simulation, two things must be determined,

(1) what is the waiting time for the next reaction to occur and,

(2) which of the reactions in the system will occur.

This is determined by generating two random numbers according to the following probability density function:

$$P(\tau, \mu) = a_\mu e^{-a_0 \tau}, \tag{2.8}$$

where

$$a_\mu = h_\mu c_\mu, \tag{2.9}$$

$$a_0 = \sum a_\mu. \tag{2.10}$$

$P(\tau, \mu) d\tau$ is the probability that the next reaction will occur in the next infinitesimal time interval $d\tau$ and that the reaction will be R_μ. Once the waiting time and the reaction to occur are determined, the number of molecules in the system and the simulation time are updated and the algorithm proceeds to the next iteration. The step by step description of the algorithm is presented in Chapter 3.

From the definition of the algorithm it is intuitive to understand that the bigger the reaction rate is or the larger the number of substrate molecules, the greater the chance that the given reaction will occur in the next step of the simulation. This also means that there is no constant time step in the simulation, given that this is determined in every iteration and is dependant on the current state of the system. Thus, it isn't easy to anticipate the simulation's computational cost and there is no way of determining the required simulation time in advance, since the total number of iterations is depending on the time step change and consequently the number of reactant molecules [12].

The algorithm that has been described thus far is the result of Gillespie's work in 1977 and is denominated the *Gillespie direct method*. It is clear that this algorithm is computationally very intensive since for each iteration two random numbers must be generated to determine the next reaction to occur. As such, some other exact SSAs were formulated to try to improve the efficiency without trading accuracy. An example is the one developed by Gibson and Bruck [15] named the *Next reaction method*, in which the next reaction time of each reaction is determined independently. The next reaction to occur is the one with the smallest next reaction time, and thus there is no need for random selection of reaction events, and so it is only necessary to generate one random number per iteration.

Anderson [16] further improves this method by scaling the times of each reaction so that these follow unit-rate exponential random variables. This way, this algorithm can be applied to more complex biochemical reaction networks with time-dependant propensity functions.

Even though the above mentioned algorithms present computational improvements in comparison to the *Gillespie direct method*, all the exact SSAs, meaning the algorithms that identically follow the probability laws of stochastic chemical kinetics, become computationally intractable for large biochemical networks where many chemical species and reactions are involved. This comes as a result of every reaction event being simulated. So, in order to try to reduce the computational burden, several approximate SSAs have been developed. It is to be noted, that with the gain of computational efficiency there is a sacrifice in accuracy.

One of the main methods developed in this category was developed by Gillespie [17], the author of the first exact SSA and is named the *τ-leaping method*. In this algorithm the system evolves in discrete time steps of defined length τ such that in the time interval $[t, t+\tau]$ the propensity functions are held constant. Then the number of reaction events that occur are counted and the state vector is updated accordingly. The number of reaction events that occur within the time interval can be considered a Poisson distribution with a mean $a_j(X(t)\tau)$, $(j = 1, ...M)$.

As mentioned, this produces results that require less computational effort at the cost of accuracy. This is observable in the clear difference in the noise patterns of the generated paths, as can be seen in Figure 2.3. This figure shows one of the obtained sample paths by each of the described methods for two different reaction networks (mono-molecular chain, a),c) and e) and enzyme kinetics, b),d) and f)). It is also possible to observe that the two exact methods (*Gillespie direct method* (a) and (b); *Next reaction method* (c) and (d)) produce different trajectories, while the noise patterns remain the same.

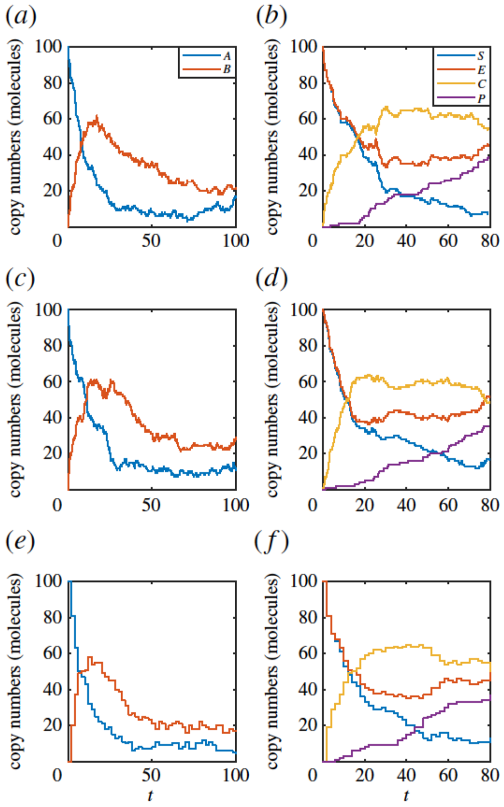

Figure 2.3: Plots illustrating the difference between the discussed methods, *Gillespie direct method* (a) and (b); *Next reaction method* (c) and (d); τ*-leaping method* (e) and (f) [4].

2.2.1 Practical examples

In this section some practical examples where stochastic simulation has been used and proved useful will be illustrated. This includes the modelling of gene expression, human immune system response and fermentation in an industrial bioreactor.

Example 1

There is strong experimental evidence that the same gene has significantly different levels of expression from cell to cell [18]. Combined with having mRNA molecules in such small numbers, the stochastic nature of the associated biochemical reactions leads to large fluctuations [19]. Thus, it is clear that these fluctuations need to be taken into account when modelling this type of systems and the Gillespie algorithm is the standard method for doing so. It does not, however account for volume changes during the course of the simulation. Cell division and growth play an important role on genetic regulatory and protein-protein interaction networks since the cell volume inversely relates to the association rate for a bi-molecular reaction [19]. With this in mind, this approach to the SSA accounts for deterministic time dependence of rate constants arising from cellular growth and division.

To do so, some modifications were made to the original algorithm. This time, there is a volume variable dependant on time. The reaction channels are divided into those whose propensities depend on the time and the ones that do not. Time was normalised in such a way that the volume doubles in a unit time interval after which the cell divides. The simplifying assumption that cell division is deterministic is made, and so the cell division time is constant and all molecules are divided uniformly among the daughter cells [19]. The algorithm proceeds in a similar way to the original, the difference being in accounting for the different types of propensities when defining the time of the next reaction.

The authors applied this to a case of transcriptional regulation without feedback. In this example, there is a single gene that can exist in two states, S_0 and S_1, where the transition between states occurs when a regulator protein is bound to the gene's promoter. This transition probability is inversely proportional to the cell volume and when cell division occurs both the volume and number of proteins are halved. The propensity of the reverse transition is considered to be time independent. The transcription of the gene leads to the production of protein X.

Given that the protein production and degradation reactions are fast when compared with the cell division time it would be too computationally expensive to use the Gillespie algorithm. The authors opted for a hybrid simulation technique where the dynamics for the fast reactions are simulated deterministically or using Langevin equations while the slow ones are simulated through the Gillespie method.

It was concluded that even though the Gillespie approach produced accurate results it was not feasible to simulate all the reactions his way. Concerning the hybrid technique, the simulations achieved good results when compared to the theoretical predictions while not increasing significantly the computational expense. This way, it was shown that it is possible to incorporate periodic deterministic events into the SSA. This opens up doors for including other sources of noise such as stochastic growth rate or unequal partition of molecules upon cell division.

Example 2

In the following example, the author's goal was to simulate the human immune response to the Yellow Fever (YF) vaccine. The Human Immune System (HIS) is fundamental in the body's defense against diseases. It has, however, many components interacting, which makes the understanding of these mechanisms complicated. Mathematical models are a useful tool for this goal. Distinct techniques can be used to achieve this, the most common being the modelling of the HIS using differential equations (ODEs or PDEs), agents (ADM) or cellular automaton. ODEs or PDEs describe solely the average behaviour of the system and as such cannot describe the behaviour of small populations or individuals as ABM and cellular automaton can. These however require significant computational power to simulate large populations [20].

Given this, a stochastic approach based on Gillespie's algorithm seems interesting since it can provide the description of distinct behaviours at a lower computational cost. It can be used to understand the dynamics of the human response to vaccines, since for example the yellow fever vaccine presents a seroconversion rate of 95% - 98% and it becomes relevant to understand what factors explain the 2% - 5% of individuals that are not seroconverted.

The developed model represents five populations: virus, antibodies and three types of lymphocyte B cells (naive, active and memory). For the deterministic model this is translated into five ODEs which are then converted into equivalent reactions for the stochastic model. Without going too much in detail about this model, in general the conversion of ODEs is done in the following way: For each species, the positive terms in the corresponding ODE will correspond to the probability of the reaction that leads to addition of one unit of the species to the simulation, as for the negative terms, they will correspond to the probability of the reaction that leads to one unit of the species to be removed.

For validation, both the deterministic and the stochastic model simulates a scenario where an individual was vaccinated against YF for the first time. Out of the 400 realisations of the stochastic model, 20 were plotted against the deterministic result for comparison. The results obtained are shown in Figure 2.4.

For the virus results, it can be seen that the stochastic simulation is in agreement with the deterministic one. Moreover, both are consistent with the literature regarding the time point in which the viraemia peak occurs. The antibody results are more interesting since it is possible to detect an important difference between the deterministic and the stochastic model. From the scientific knowledge, it is expected that after vaccination, an individual's antibody level remains different from zero, which is observed in the

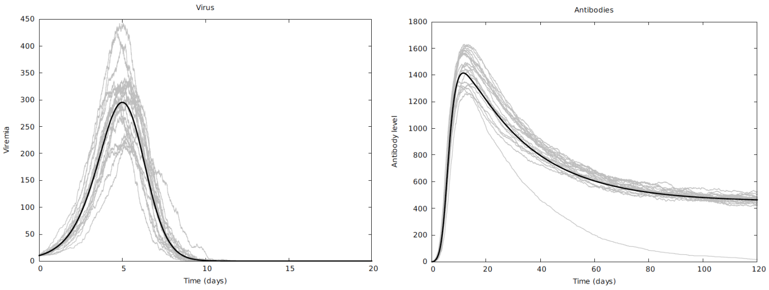

Figure 2.4: Simulation results obtained by [20]; The black line represents the deterministic simulations, the light grey line, represents the different realisations of the stochastic simulation.

deterministic model. One specific stochastic simulation, however, shows otherwise, that after 120 days the antibody level is almost equal to zero. When accounting for all 400 realisations, this happened in around 4% of the cases. This is within the range of 2% - 5% of experimental non seroconversion mentioned above. So, in conclusion, the stochastic model was able to reproduce what the deterministic one could not, that not all individuals seroconvert after vaccination.

Example 3

In the third chapter of his Master Thesis [14], Isaza approaches the production of xylitol at an industrial level. Since it is desired to predict the behaviour of these bioprocesses while also accounting for their inherent stochasticity, stochastic modelling is of great relevance. So in his work, both the deterministic and stochastic models for the fermentation in question are developed.

The model includes as species biomass (*Candida moggi*), glucose, xylose and intra and extracellular xylitol in a batch bioreactor. As in the previous example, the construction of the stochastic model is based on the deterministic one. The conversion of ODEs into propensities is the one adopted in the present thesis and as such is explained in detail in Chapter 3.2.

In the obtained results, the stochastic simulation is not shown with all its realisations, but instead it is shown the uncertainty of the system concluded from those. This results are shown in Figure 2.5, where the red full line is the result of the deterministic simulation and the grey dashed lines the deviation envelope.

First of all, from the results it is possible to observe good agreement between the stochastic model and the deterministic one as well as with the experimental data. For xylitol, the biggest difference between the models is observed, since the simulation was conducted with a smaller volume $5 \cdot 10^{-22}$ in order to increase uncertainty and thus contain the experimental results. The model allows to evaluate the different scenarios on which the concentration may vary as well as predict the most probable behaviour from the average.

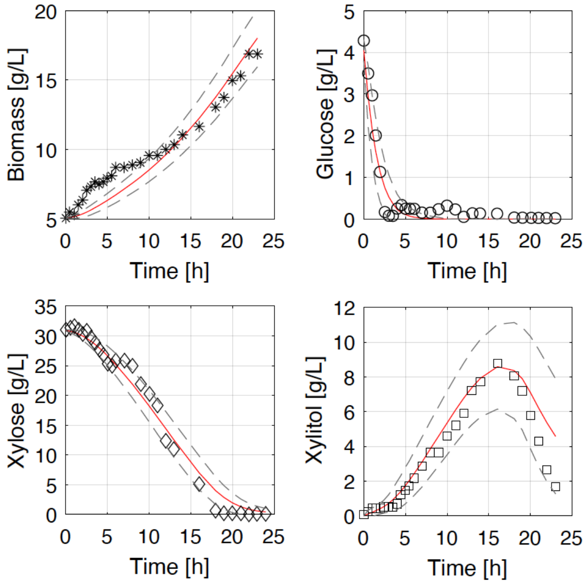

Figure 2.5: Simulation results obtained by[14]; The red line represents the deterministic solution, the stars, circles, diamonds and squares, represent the experimental results, and the space in between the dashed grey lines corresponds to the envelope predicted by the stochastic simulation.

2.3 Recombinant protein production in *E. coli*

Escherichia coli is a Gram-negative bacterium that is a facultative anaerobe. It is the most well studied micro-organism which is one of the characteristics that makes it an attractive choice for the production of biological products.

Even though many organisms can and are used for recombinant protein production, *E. coli* remains the favourite due to ease of cultivation, short life-cycle and easy genetic manipulations because of the vast knowledge of its genetics [21]. Thus, using *E. coli* as a microbial cell factory for producing recombinant proteins reduces the cost of production and improves yield [22].

Nonetheless, producing in *E. coli* comes with its challenges. This can be caused by protein toxicity to the host organism or protein aggregation in inclusion bodies. This can either harm biomass growth or complicate the protein structure and consequent downstream process. For this reason among others, there is still a considerable interest in research on novel strategies to improve this process [22].

Fermentations for protein production usually run at high growth rates to maximize growth and production. This, however, also brings some challenges. One is the significant acetate accumulation as a by-product that affects both biomass growth and the production of the product of interest. Glucose is usually fed as a carbon source, and even in a medium with multiple carbon sources its the preferred one by the bacteria due to catabolite repression [23]. Once the glucose is taken up, it is simply metabolised through glycolysis and central carbon metabolism with acetate as a secondary product. This acetate is both a loss of the carbon source and thus financial handicap but as well inhibiting for cell growth and protein production [24]. Besides this, it can as well have a negative effect on the stability of intracellular proteins [25].

There can be two reasons behind acetate production. One is a drop in dissolved oxygen that activates the fermentation pathway causing acetate excretion. The other one is overflow metabolism, which is caused by imbalance between the fast glucose uptake and the biomass and other products formation rate. Acetyl-CoA is then diverted from the TCA cycle to the formation of acetate. Figure 2.6 [9] schematises the glucose path in the central carbon metabolism, where the conversion of AcCoA to acetate is also shown.

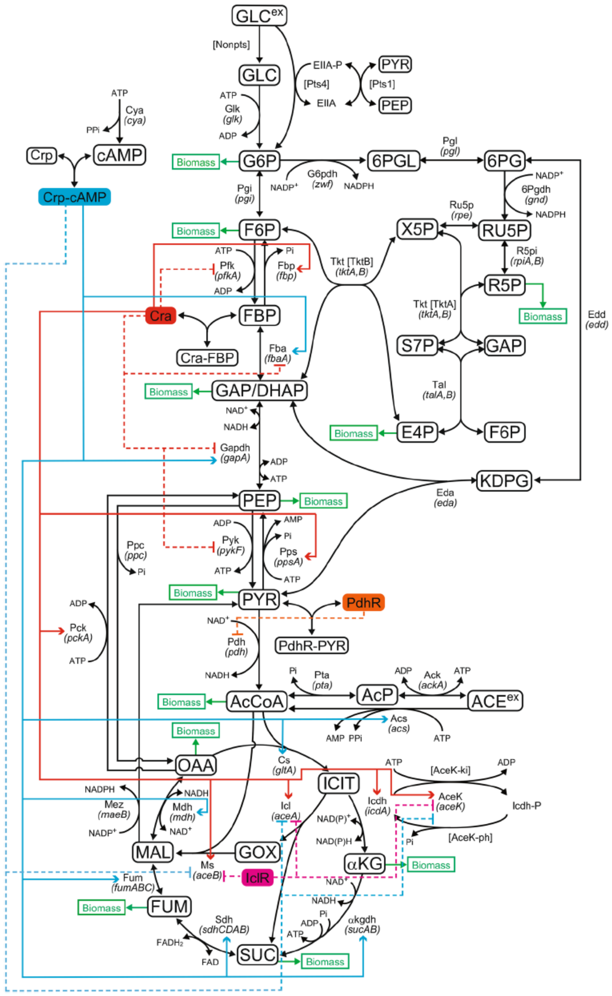

Figure 2.6: Schematic representation of the central carbon metabolism of *E. coli* [9].

The amount of acetate produced depends on several factors such as the producing strain, growth conditions and glucose concentration. Strategies to control acetate secretion can be done at the genetic level, or at the bioprocess level which is the concern of this thesis.

These methods consist mostly in controlling the medium composition and fermentation conditions, like temperature, pH, dissolved oxygen and substrate concentration. The *E. coli* culture will generate acetate when the substrate consumption rate surpasses a certain threshold. To overcome this different strategies can be use such as limitation of growth through substrate-limited fed-batch schemes and utilisation of different substrates such as glycerol or fructose.

To adopt these efficient strategies, relevant mathematical models that predict growth and nutrient demand are a key factor. An example of this is the growth model developed by Anane et al. [26] on which the model developed in this thesis is partly based on.

2.4 Case study: GFP and rhGH

2.4.1 Introduction to Green Fluorescent Protein

Green Fluorescent Protein (GFP) is a 238 amino acid protein with a molecular weight of around 27 kDa. Figure 2.7 shows its tertiary structure. It was first identified in the jellyfish *Aequorea victoria.* This organism gets its bioluminescence from two photo proteins, one of which is GFP. The fluorescence originates in a sequential activation of aequorin and GFP. When binding to calcium, aequorin emits blue light which then excites GFP to fluoresce green [27].

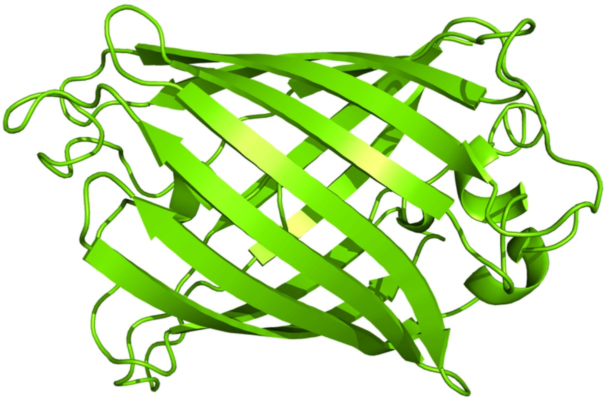

Figure 2.7: Tertiary structure of GFP.

GFP has a particularity that it forms a chromophore of three amino acids within its primary structure and so, unlike other bioluminescent molecules it can operate in the absence of a co-factor. It is also a very stable protein. Even though the molecule's properties have been known years before, only in 1992 when its cDNA was cloned, and with subsequent heterologous expression in *E. coli* and *C. elegans*, the research community became aware of its potential applications [28]. In more recent years, through direct-site mutagenesis, GFP can now emit blue, cyan and yellow [29].

A few examples are the use of GFP as a reporter for gene expression; a marker to study cell lineage during development and a tag to localize proteins in living cells [29].

The first applications were as a reporter gene. The gene expression can be measured by placing GFP under the control of the promoter of the gene of interest and measuring GFP fluorescence to directly indicate the gene expression level in living cells or tissue. Other application are fusion tags. In this case GFP is fused with a cloned gene. With this, dynamic cellular events can be visualised and protein location can be monitored. GFP is ideal for this since it doesn't require the presence of any cofactors or substrates. As a final example, GFP can be used as a marker for tumor cells, illuminating tumor progression and allowing the detection of metastases up to single-cell level [30].

GFP is also used for research purposes, for example for testing fermentation conditions for the production of recombinant proteins [31] due to its ease of quantification.

2.4.2 Introduction to Recombinant Human Growth Hormone

Human Growth Hormone (hGH) is a single-chain polypeptide hormone. It is mainly synthestised in the anterior pituitary gland by the acidophilic somatrophs. The most common form of this hormone contains 191 amino acid residues and has a molecular weight of around 22 kDa. It is released in the blood stream where it participates in several biological functions, like protein synthesis, cell proliferation, lactation, immune regulation and metabolism of proteins, carbohydrates and lipids [32].

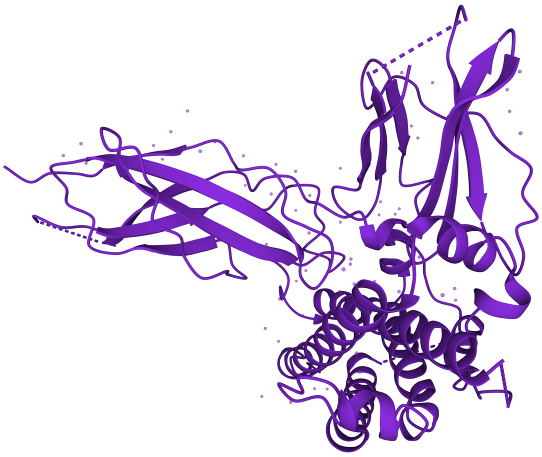

Figure 2.8: Tertiary structure of rhGH.

Before the 1980s, the only available source for hGH was human cadaver tissue, which made its use as a therapeutic agent forbidden. With the advances made in recombinant DNA technology it is now possible to express proteins, such as hGH, in host cells, eliminating the risk of transfer of human pathogens and requirement of pituitary-derived preparation. Now, safe and abundant production of recombinant human Growth Hormone (rhGH) is possible, as well as its therapeutic use in diseases related with growth hormone deficiency (GHD) [32].

rhGH can be used in a multitude of health applications. It is FDA-approved for treatment of several conditions such as growth hormone deficiency, chronic renal insufficiency, and some genetic diseases like Turner syndrome and Prader-Willi syndrome. GHD is caused by hypothalamic or pituitary dysfunction. Multiple studies have demonstrated that rhGH is effective in the treatment of GHD, with patients achieving adult heights consistent with family related parameters [33]. In the case of chronic renal insufficiency, children suffering from it usually experience a linear decrease in growth. Studies have shown that treatment with rhGH increases linear growth, bone mineral density, body weight and lean mass. Also, it doesn't appear to negatively affect the progression of disease [34]. Turner syndrome is characterised by an abnormal or missing X chromosome which results in girls having short stature and ovarian failure. Therapy with rhGH increases growth rate and adult height of girls suffering from this syndrome [34]. Additionally to the mentioned diseases, rhGH also has a positive effect on bone fractures, skin burns and bleeding ulcers [32].

Regarding its recombinant production, since nonglycosylated hGH is biologically active, prokaryotic expression systems are the favourites for its production [32]. Like most recombinant proteins, production is more efficient when growth and production phase are separated, therefore, recombinant organisms that have inducible promoters are preferred. *E. coli*, as mentioned in the previous chapter, is one of the most widely used hosts for production of heterologous proteins, and it has several inducible promoter systems. These are advantageous for the over production of recombinant proteins in high cell density fermentations.

3

Methodology

Contents

This chapter will detail the methodology used throughout the development of this work. This includes the algorithm used as well as the equations used for the models and the chosen method for parameter estimation. Figure 3.1 illustrates the workflow.

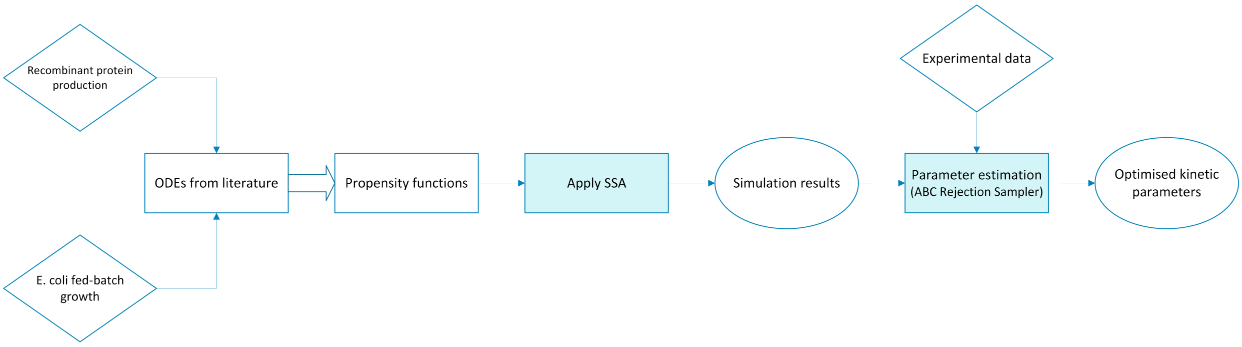

Figure 3.1: Information flow of the developed methodology. The diamond shapes symbolize the inputs, the circle shapes the results/outputs, and the rectangles show the intermediate steps. The highlighted rectangles represent the two main steps.

3.1 Model development

In this section, the equations implemented in each of the models and the literature they were based on are detailed.

3.1.1 *Escherichia coli* central carbon metabolism

The model used was based on the one developed by Jahan et al. [9]. It describes the central carbon metabolism of *E. coli* including the glycolytic pathway, TCA cycle, pentose phosphate pathway, Entner-Doudoroff (ED) pathway, anaplerotic pathway, glyoxylate shunt, oxidative phosphorylation, Pts and non-Pts. The kinetic model contains 27 metabolites, 22 enzymes and Pts proteins, 38 fluxes, 21 gene expressions and 12 biomass production rate equations for precursor intracellular metabolites [9].

Due to the significant number of equations and the fact that no changes are made to the model equations, these are not presented here. The interested reader is referred to the original paper ([9]), where the model is described in detail.

3.1.2 *Escherichia coli* fed-batch fermentation

The model for *E.coli* growth is based on the model developed by Anane et al. [26] which takes into account acetate cycling on the bacterial growth. This is of great relevance since *E. coli*'s overflow metabolism and acetate excretion to the fermentation broth are biological phenomena that that have most impact on industrial fermentation. This is not only because the extracellular acetate inhibits bacterial growth, but also because the carbon source diverted into the overflow metabolism affects the production of recombinant proteins [26].

Regarding the ordinary differential equations that this model consists of, for the state variables, X, S, A, these equations follow the general Equation 3.1.

$$\frac{dx}{dt} = \frac{F}{V}\left(x_i - x\right) + rX, \tag{3.1}$$

where $x \in \{X, S, A\}$ represents the respective state variable in [g/L], subscript i the inlet concentration, F the feed, V the volume and r is the corresponding specific rate. For the biomass, considering the inlet concentration is zero, Equation 3.2 is obtained.

$$\frac{dX}{dt} = \mu X - \frac{F}{V}X, \tag{3.2}$$

where X represents the cell concentration, expressed as cell dry weight (CDW), and μ is the non-inhibited Monod-type specific growth rate, that is given by Equation 3.3.

$$\mu = \left(q_{sox} - q_m\right) Y_{em} + q_{sof} Y_{Xsof} + q_{sA} Y_{Xa}, \tag{3.3}$$

where q_{sox}, q_{sof}, q_{sA} represent the substrate uptake rates for oxidation, metabolism through the overflow route and acetate, respectively. As for the $Y_{...}$ constants, they define the respective yield coefficients, q_m represents the substrate consumed for cell maintenance. The mass balance for glucose (S) is represented by Equation 3.4.

$$\frac{dS}{dt} = \frac{F_s}{V}\left(S_f - S\right) - q_s X. \tag{3.4}$$

The specific substrate uptake rate, q_S, takes into account the acetate inhibition, as such is modelled with Monod-type kinetics with non-competitive inhibition (Equation 3.5).

$$q_s = \frac{q_{\text{smax}}}{1 + \frac{A}{K_{ia}}} \cdot \frac{S}{S + K_s}, \tag{3.5}$$

where K_{ia} and K_s are the acetate inhibition and substrate affinity constant, respectively. The substrate that is consumed, q_S is not only metabolized in the TCA cycle (q_{sox}), but also through the overflow path, q_{sof} (Equations 3.6 and 3.7).

$$q_{sox} = \left(q_s - q_{sof}\right) \cdot \frac{DO}{DO + K_0}, \tag{3.6}$$

$$q_{sof} = \frac{P_{Amax} q_s}{q_s + K_{ap}}, \tag{3.7}$$

K_o is a dimensionless constant set to 0.1, to increase the numeric stability of the simulation. P_{Amax} and K_{ap} are the maximum acetate production and the production affinity constants, respectively. The acetate production/consumption is considered a cyclic process that is described by the mass balance of Equation 3.8.

$$\frac{dA}{dt} = q_A X - \frac{F}{V} A. \tag{3.8}$$

When the acetate produced through the overflow route, p_A, is equal to the acetate consumed, q_{sA}, equilibrium is reached ($q_A = 0$), as described by Equation 3.9.

$$q_A = p_A - q_{sA}, \tag{3.9}$$

$$p_A = q_{sof} Y_{as}. \tag{3.10}$$

In Equation 3.10, Y_{as} is the yield of acetate relative to the substrate consumed through the overflow route [g/g]. The specific acetate consumption rate is modelled similarly to the substrate uptake rate, following a non-competitive inhibition Monod-type kinetic (Equation 3.11).

$$q_{sA} = \frac{q_{Amax}}{1 + \frac{q_s}{K_{is}}} \cdot \frac{A}{A + K_{sa}}, \tag{3.11}$$

where q_{Amax}, K_{is} and K_{sa} are the maximum acetate uptake rate, the acetate uptake inhibition and acetate affinity constants, respectively. Through Equations 3.4 and 3.11, the glucose and acetate counter inhibition effect is illustrated.

Finally, the dissolved oxygen is calculated in % of saturation, assuming that the feeding solution in the fed-batch phase is fully saturated with dissolved oxygen. The oxygen profile is described by Equation 3.12.

$$\frac{dDO}{dt} = K_{La} \left(DO^* - DO\right) - q_o X, \tag{3.12}$$

where DO^* is the saturation value of dissolved oxygen in the medium, K_{La} the volumetric mass transfer coefficient for oxygen and q_o the oxygen uptake rate described by Equation 3.13. It should be noted that the above equation differs from the one presented by Anane et al., as the authors include Henry's constant, H, in the second term. This is omitted here since due to the algorithm's nature, the constant is used in the DO conversion equation (Equation 3.29) in the following sub-chapter 3.2.

$$q_o = (q_{sox} - q_{san})\,Y_{os} + q_{sA}Y_{oa}. \tag{3.13}$$

Regarding the equation above, Y_{oa} and Y_{os} are the yield coefficients for substrate and acetate to oxygen consumption, respectively. q_{san} is defined by Eq. 3.14.

$$q_{san} = (q_{sox} - q_m)\,Y_{em} \cdot \frac{C_x}{C_s}, \tag{3.14}$$

where C_x and C_s are the carbon content of biomass and glucose, respectively.

Regarding the feeding strategy, just as adopted by Anane et al., the fed-batch phase is initiated when there is an exhaustion of the batch phase glucose (activated when glucose drops below 0.05 g/L) and consisted of two-feeding regimes. First an exponential feed that is applied that was used to maintain a set-point specific growth rate μ_{set}.

$$F_s(t) = \frac{\mu_{set}}{Y_{x/s}S_f}(X_0V_0)e^{\mu_{set}t_f}. \tag{3.15}$$

The exponential feeding rate F_s [L/h] that is calculated through Equation 3.15 uses biomass concentration (X_0) and volume (V_0) at the end of the batch phase, biomass to substrate yield ($Y_{x/s}$) and glucose concentration in the feed solution (S_f) and it is a function of feeding time, t_f.

After 3h, the feeding is switched, Anane et al. [26] proceeds with a constant feed, but it was decided to do the switch for a decreasing feed. This is based on the finding in [35] that substrate accumulation, which can lead to inducer exclusion, would occur before batch completion. So, when the substrate concentration exceeds 0.03 g/L, the exponential feed is substituted by a linearly decreasing feed for the remaining duration of the fermentation.

3.1.3 Protein production

To account for the recombinant protein production, ODEs describing these dynamics were necessary.

3.1.3.A Green Fluorescent Protein

In the case of GFP the equation is based on the paper by Aucoin et al. [31]. It was assumed that the production follows a mixed-growth associated product formation kinetic. It was considered that the production is temperature induced, so the equation does not require a term for inducer dynamic. As such, the production is controlled by an inducer switch, which is activated when desired.

So, before when the inducer switch is turned OFF, the ODE for the biomass remains the same as Eq. 3.2, and the equation for protein production corresponds to Eq. 3.16.

$$\frac{dP}{dt} = 0. \tag{3.16}$$

When the inducer switch is turned ON, the ODE for biomass becomes Eq. 3.17 and the ODE for protein production becomes Eq. 3.18.

$$\frac{dX}{dt} = \mu X(1 - x_{ind}) - \frac{F}{V}X, \tag{3.17}$$

$$\frac{dP}{dt} = \left(Y_{P/X}\mu + \beta\right) X \cdot x_{ind} - \frac{F}{V}P. \tag{3.18}$$

Thus, GFP production is described by Equation 3.18 and is dependant of two terms. The first one corresponds to the growth associated product, where $Y_{P/X}$ is the protein to biomass yield and μ the specific growth rate described above by Equation 3.3. As for the second term, it represents the non-growth-associated production, where β is a protein production constant. The term x_{ind} represents an attenuating term for protein production.

3.1.3.B Recombinant Human Growth Hormone

In the case of the rhGH, it was considered that the induction is done in response to IPTG, one of the most common inducers used in the biotechnology industry. As such, the protein equation has to include an inducer dependant term. Also, a new species, the inductor, must be introduced and have an equation describing its dynamics.

The equations are based on the model defined by Chae et al. [36] for recombinant protein expression in *E. coli*. Since foreign protein production also occurs before inducer addition due to read through transcription and translation, it is useful that the model includes this type of production. This results in Equation 3.19.

$$\frac{dP}{dt} = \left(k_1\mu\frac{I}{I + K_i} + k_2\right) X - k_3P - P\frac{F}{V}, \tag{3.19}$$

where k_1 and k_2 are the induced and constitutive protein biosynthesis rates respectively, K_i is the induction constant and k_3 a constant first order degradation term. As for the inductor, the ODE follows the general Equation 3.1.

$$\frac{dI}{dt} = \frac{F_i}{V}(I_f - I) - I\frac{F}{V}, \tag{3.20}$$

where I_f is the concentration of the inducer on the inducer feed stream.

In addition, a feeding strategy for the induction needs to be defined. According to Ruiz et al. [37]

it is adequate to start induction when biomass reaches the concentration of 20 g/L. As such, following the strategy outlined in [35], when $X = 20$ g/L a constant flow rate of $F_i = 200$ g/L is set to rapidly rise IPTG levels. When the concentration reaches 1 mM or 0.238 g/L, the flow rate is determined by setting Equation 3.20 to 0 and the inducer concentration to $I = 0.238$ g/L, so it remains constant for the remaining part of the process. Solving this equation for F_i, Eq. 3.21 is obtained.

$$F_i = \frac{0.238 F_s}{I_f - 0.238}. \tag{3.21}$$

As for the substrate feed, the same exponential feed equation (3.15) as in section 3.1.2 was used. However, in this case the feed is switched on when glucose drops below 0.5 g/L. After 3h, the feed control is the same as described previously.

3.2 Model implementation

When it comes to first order reactions, the propensity function is the same as the deterministic kinetic equation, however, the same is not true for more complex ODEs. As such, it was necessary to transform the ODEs from the reference models so they could be incorporated into the SSA. This was done according with the work of [14].

The most straightforward case occurs when there is the same species on both sides of the equations, as presented, for example, in Equation 3.2. In this case, it is only necessary to replace the concentrations for number of molecules, through Equation 3.22. This is the unit needed to run the SSA.

$$C_i = \frac{N_i mw_i}{N_{Av} V}. \tag{3.22}$$

This leads to Equation 3.23.

$$\frac{dN_X}{dt} \cdot \frac{mw_X}{N_{Av} V} = \mu N_X \cdot \frac{mw_X}{N_{Av} V} - \frac{F}{V} N_X \cdot \frac{mw_X}{N_{Av} V}. \tag{3.23}$$

When 3.23 is simplified it originates the propensity function (Equation 3.24). It is equivalent to the ODE, except where the concentration is replaced by the number of molecules.

$$a_X = \mu N_X - \frac{F}{V} N_X. \tag{3.24}$$

In the situations where there are different species on each side of the equation, the first step is again to replace Equation 3.22 into, for example, Equation 3.4, which then leads to Equation 3.25.

$$\frac{dN_s}{dt} \cdot \frac{mw_s}{N_{A_v} V} = \frac{F_s}{V} \left(S_f - N_s \cdot \frac{mw_s}{N_{A_v} V} \right) - q_s \cdot N_x \cdot \frac{mw_x}{N_{A_v} V}. \tag{3.25}$$

This generates the following propensity function (Equation 3.26).

$$a_S = \frac{F_s}{V}\left(S_f \cdot \frac{N_{Av}V}{mw_S} - N_s\right) - q_s \cdot N_X \cdot \frac{mw_X}{mw_S}. \tag{3.26}$$

Intending to simplify when programming the equations, for each chemical species, a conversion constant was defined and applied (Equation 3.27). Replacing this constant into Equation 3.26 yields the final propensity function, Equation 3.28.

$$K_{conv,i} = \frac{N_{Av}V}{mw_i}, \tag{3.27}$$

$$a_S = \frac{F_s}{V}(S_f \cdot K_{conv,S} - N_s) - q_s \cdot N_X \cdot \frac{mw_X}{mw_S}. \tag{3.28}$$

In the particular case of DO, the conversion constant differs from the remaining ones, due to the input initial condition being in units of %DO. As such, Henry's constant was applied to convert this value to mol/L (M) and afterwards to number of molecules (Equation 3.29).

$$K_{conv,DO} = \frac{N_{Av}V}{H}. \tag{3.29}$$

As for the growth equations such as Equations 3.5 and 3.6, it is only necessary to replace the concentration terms using Equation 3.22. Then, using the conversion constant for each species, it finally leads to Equations 3.30 and 3.31.

$$q_s = \frac{q_{\text{smax}}}{1 + \frac{N_A/K_{conv,A}}{K_{ia}}} \cdot \frac{N_S/K_{conv,S}}{S + K_s}, \tag{3.30}$$

$$q_{sox} = (q_s - q_{sof}) \cdot \frac{N_{DO}/K_{conv,DO}}{N_{DO}/K_{conv,DO} + K_o}. \tag{3.31}$$

Since both the remaining propensity functions and growth equations are modified in the same way, there is no need to exhaustively explain all of them. As such, all the final equations are presented in Table 3.1. Table 3.2 summarises all the parameters used in the model.

Table 3.1: Final model equations.

Growth equations

$$\mu = \left(q_{sox} - q_m\right) Y_{em} + q_{sof} Y_{Xsof} + q_{sA} Y_{Xa}$$

$$q_s = \frac{q_{\mathrm{smax}}}{1+\frac{N_A/K_{conv,A}}{K_{ia}}} \cdot \frac{N_S/K_{conv,S}}{S+K_s}$$

$$q_{sox} = \left(q_s - q_{sof}\right) \cdot \frac{N_{DO}/K_{conv,DO}}{N_{DO}/K_{conv,DO}+K_o}$$

$$q_{sof} = \frac{P_{Amax} q_s}{q_s+K_{ap}}$$

$$q_A = p_A - q_{sA}$$

$$p_A = q_{sof} Y_{as}$$

$$q_{sA} = \frac{q_{Amax}}{1+\frac{q_s}{K_{is}}} \cdot \frac{N_A/K_{conv,A}}{N_A/K_{conv,A}+K_{sa}}$$

$$q_{san} = \left(q_{sox} - q_m\right) Y_{em} + \frac{C_x}{C_s}$$

$$q_o = \left(q_{sox} - q_m\right) Y_{os} + q_{sA} Y_{oa}$$

Propensity functions - Fed-batch growth

$$a_X = \mu N_X - \frac{F}{V} N_X$$

$$a_S = \frac{F_s}{V}\left(S_f \cdot K_{conv,S} - N_s\right) - q_s \cdot N_X \cdot \frac{mw_X}{mw_S}$$

$$a_A = q_A \cdot N_X \cdot \frac{mw_X}{mw_A} - \frac{F}{V} N_A$$

$$a_{DO} = K_{La}\left(DO^* \cdot K_{conv,DO} - N_{DO}\right) - q_o \cdot N_X \cdot mw_X$$

Propensity functions - GFP production

IND=OFF:
$a_X = \mu N_X - \frac{F}{V} N_X$
$a_P = 0$

IND=ON:
$a_X = \mu N_X (1 - x_{ind}) - \frac{F}{V} N_X$
$a_P = \left(Y_{P/X} \mu + \beta\right) N_X x_{ind} \cdot \frac{mw_X}{mw_P} - \frac{F}{V} N_P$

Propensity functions - rhGH production

$$a_P = \left(k_1 \mu \frac{N_I/K_{conv,I}}{N_I/K_{conv,I}+K_i} + k_2\right) \cdot N_X \cdot \frac{mw_X}{mw_P} - k_3 N_P - N_P \frac{F}{V}$$

$$\mathrm{a}_I = \frac{F_i}{V}\left(I_f \cdot K_{conv,I} - N_I\right) - N_I \frac{F}{V}$$

Feeding control

$$F_s(t) = \frac{\mu_{set}}{Y_{x/s} S_f}(X_0 V_0) e^{\mu_{set} t_f}$$

$$F_i = \frac{0.238 F_s}{I_f - 0.238}$$

Table 3.2: Final parameters.

Parameter	Value
Fed-batch growth	
q_m (Specific maintenance coefficient)	0.0129
Y_{em} (Yield for exclusive maintenance)	0.5333
Y_{xsof} (Biomass yield from overflow route)	0.2268
Y_{xa} (Yield of biomass on acetate)	0.5178
C_x (Carbon content of biomass)	0.488
C_s (Carbon content of glucose)	0.391
S_f (Glucose concentration in feed)	300
q_{smax} (Maximum specific glucose uptake rate)	0.6356
K_{ia} (Inhibition of glucose uptake by acetate)	1.2399
K_s (Affinity constant for substrate consumption)	0.037
K_O (Affinity constant for oxygen consumption)	0.0001
P_{amax} (Maximum specific acetate production rate)	0.2268
K_{ap} (Monod-type saturation constant for intracellular acetate production)	0.5052
Y_{as} (Yield of acetate on substrate)	0,9097
q_{amax} (Maximum specific acetate consumption rate)	0.1148
K_{is} (Inhibition of acetate uptake by glucose)	2.1231
K_{sa} (Affinity constant for acetate consumption)	0.0134
K_{La} (Volumetric mass transfer coefficient)	220
DO^* (Saturation value of dissolved oxygen)	99
Y_{os} (Yield of oxygen on glucose)	1.552
Y_{oa} (Yield of oxygen on acetate)	0.544
$Y_{x/s}$ (Yield of biomass on substrate)	$\mathrm{Y}_{em} + q_m$
μ_{set}(Set specific growth rate during exponential feed)	0.25
GFP production	
$Y_{p/x}$ (Yield of protein on biomass)	0.05
β (Protein production constant)	0.014
x_{ind} (Attenuating term)	0.8
q_m (Specific maintenance coefficient)	0.0133
Y_{em} (Yield for exclusive maintenence)	0.5
Y_{xsof} (Biomass yield from overflow route)	0.229
Y_{xa} (Yield of biomass on acetate)	0.5794
q_{smax} (Maximum specific glucose uptake rate)	1.5
K_{ia} (Inhibition of glucose uptake by acetate)	1.0062
K_s (Affinity constant for substrate consumption)	0.05
K_{ap} (Monod-type saturation constant for intracellular acetate production)	0.56
K_{is} (Inhibition of acetate uptake by glucose)	1.8383
K_{sa} (Affinity constant for acetate consumption)	0.0128
Y_{os} (Yield of oxygen on glucose)	1.5722
Y_{oa} (Yield of oxygen on acetate)	0.5221
rhgh production	
k_1 (Induced rhGH biosynthesis rate)	0.32
K_i (Inducer saturation constant)	0.55
k_2 (Constitutive rhGH biosynthesis rate)	0.00044
k_3 (First order degradation term)	0
I_f (Inducer concentration in feed)	50
Y_{em} (Yield for exclusive maintenance)	0.65
S_f (Glucose concentration in feed)	600
P_{amax} (Maximum specific acetate production rate)	0.41
K_{is} (Inhibition of acetate uptake by glucose)	1.8363
μ_{set} (Set specific growth rate during exponential feed)	0.3

To run the Gillespie algorithm, the equations need to define reaction channels with the corresponding stoichiometry. In this way, once the ODEs have been transformed in propensity functions, a reaction was associated to each of them. At this point, an assumption taken here was that the stochiometry would be 1 to 1 as described in Isaza [14].

Even though some species are more likely to be produced (increase in concentration) and others to be consumed (decrease in concentration), both cases can happen for every species, due to for example the dilution when feeding occurs. As such, in this work, it was defined that it would be a production reaction when the associated propensity was positive, and a consumption reaction when the propensity was negative. The possibility of implementing two separate channels for each species, one for production and one for consumption, was also evaluated, however it would increase the complexity of the model and produce the same results. All the reactions implemented in the model are summarised in Table 3.3.

Table 3.3: Reaction channels.

$a_i > 0$	$a_i < 0$
$N_X \rightarrow N_X + 1$	$N_X \rightarrow N_X - 1$
$N_S \rightarrow N_S + 1$	$N_S \rightarrow N_S - 1$
$N_A \rightarrow N_A + 1$	$N_A \rightarrow N_A - 1$
$N_{DO} \rightarrow N_{DO} + 1$	$N_{DO} \rightarrow N_{DO} - 1$
$N_P \rightarrow N_P + 1$	$N_P \rightarrow N_P - 1$
$N_I \rightarrow N_I + 1$	$N_I \rightarrow N_I - 1$

Volume is a key variable in this simulation since it is used in the C to N conversion and also is changing throughout the simulation due to the fed-batch mode of operation.

Since the models deal with bioreactors either of industrial or pilot-scale, the volumes in question are considerably big. As such, if this same volume was to be used in the calculation of N, it would result in a very big number of molecules for each species which significantly slows the computation or even renders it impossible, due to the big memory burden inherent to the algorithm. This way, it was decided to have two separate volume variables, a simulation volume, V_{sim}, used in the computation of N and the reactor's working volume, V.

For the simulation volume, it was appropriate to keep it at cell scale, between $1 \cdot 10^{-17}$ L and $1 \cdot 10^{-18}$ L. This allows for a computing time that is feasible, even though still extensive, as well as producing a good match with the deterministic models.

This solution only fixed the first half of the volume "problem". Since it is required that the bioreactor volume changes once the feeding is initiated, the first solution was to make it a chemical species to include in the SSA. The conversion to N in this case would be described by Equation 3.32. Considering that N is directly proportional to V, an industrial scale volume, would lead to a number of molecules, that would make the model computation infeasible.

$$N_V = \frac{\rho N_{Av} V}{mw_V}. \tag{3.32}$$

With this in mind, another solution was proposed. Consider the volume as a separate value, with a starting value of V_0, equal to the desired bioreactor volume, which is maintained constant through the batch-phase. Once the feeding phase is initiated, a new variable V, starts being updated through Equation 3.33 on each iteration of the SSA.

$$V_{new} = V_{current} + F \cdot \tau. \tag{3.33}$$

3.3 Stochastic Simulation Algorithm

The stochastic simulations were performed using the *Gillespie direct method* [11], previously described in Chapter 2. The algorithm hereafter detailed is based on the work of [11] which is described more straightforwardly by Kierzek et al. [12].

Initialisation:

- Input values for the stochastic rate constants c_i $(i = 1, ..., M)$.
- Input initial values for initial number of reactant molecules X_i $(i = 1, ..., M)$
- Set initial time of simulation $t = 0$.

Iteration:

- For every reaction calculate the propensity $a_\mu = h_\mu c_\mu$ $(\mu = 1, ..., M)$
- Calculate $a_0 = \sum_{n=1}^{M} a_\mu$
- Generate two random numbers r_1 and r_2 uniformly distributed over $(0, 1)$.
- Calculate the waiting time for the next reaction $\tau = (1/a_0)ln(1/r_1)$
- Choose the μ index for the next reaction so that $(a_1 + a_2 + ... + a_{\mu-1}) < r_2 a_0 < (a_1 + a_2 + ... + a_\mu)$
- Update the number of molecules in the system by executing reaction μ according to the stoichiometry.
- Set simulation time to $t = t + \tau$

Termination:

- Terminate simulation when time of simulation t exceeds predetermined maximum simulation time or when all substrates of all reactions in the system are consumed ($a_0 = 0$)

Given that the algorithm is based on the chemical interactions of molecules on a sub-microscopic level, the inputs of the propensity functions must be in a unit of molecules and not concentrations. Since concentration units are widely used, it was imperative to adapt the algorithm to have these as input and output, so that comparison with other models and experimental data is possible.

First of all, the conversion from concentrations to number of molecules should be done by using to the Avogadro number according to Equation 3.34.

$$N_i = \frac{C_i N_{Av} V}{mw_i}. \tag{3.34}$$

When the algorithm terminates, the reverse conversion is done by applying Equation 3.22.

3.4 Parameter Estimation

Given the implicit randomness of the SSA, for parameter estimation, it was required to use a method that would take this into consideration. ABC, is a widely used likelihood-free approach, especially in the field of systems biology [4]. This method is based on a discrepancy metric, ρ, described by Equation 3.35, which measures the difference between the measured data, Y_{obs}, and the simulated data, S_{obs} that is generated trough the SSA. The tested parameter's acceptance depends on the discrepancy threshold, ϵ.

$$\rho\left(Y_{\text{obs}}, S_{\text{obs}}\right) = \left[\sum_{i=1}^{n_t}\left(Y\left(t_i\right) - S\left(t_i\right)\right)^2\right]^{1/2}. \tag{3.35}$$

The algorithm used to implement this method can be described with the following steps:

- Define the starting value of the parameter to test, θ, and the range in which it should be varied.
- Generate a trial parameter, θ_{trial}, using a random uniform distribution in the interval defined in the previous step.
- Generate the simulated data with θ_{trial} as an input by running the SSA, and generating data for the time points present in the experimental data.
- Calculate the value for the discrepancy metric, ρ.
- If $\rho \leq \epsilon$, θ_{trial} is accepted.

- The algorithm terminates when a predefined number of parameters has been accepted.

According to Warne et al. [4], the value for ϵ should be chosen such that it is small enough in order for a good approximation to the experimental data to be obtained, but not so small that it will make the estimation too long. As such, for the data sets in question, a value of $\epsilon = 5$ for the GFP and $\epsilon = 0.01$ for the rhGH, was assumed to be acceptable. The number of accepted parameters was decided to be 100 to assure a sufficiently low standard deviation.

Once the set of accepted parameters was determined, a histogram was plotted to determine what interval is most suited. Finally, the new parameter is selected using the average value within the selected range.

As an explanatory example, Figure 3.2 represents one of the histograms obtained.

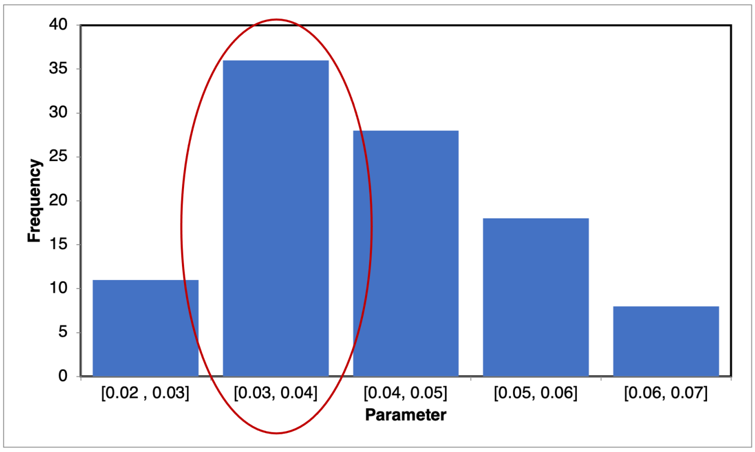

Figure 3.2: Example histogram for one of the estimated parameters; the red circle highlights the chosen interval.

So, in this case, the interval $[0.03, 0.04]$ would be chosen, and the selected new parameter value would be the average value of the interval, 0.035.

4

Results and discussion

Contents

The present chapter will detail the main results obtained throughout this project. These results will be discussed and compared with the available knowledge from literature.

4.1 *E. coli* central carbon metabolism

The first approach was to model *E. coli's* growth using a structured model. This was because SSA's use has been reported for systems containing a small number of molecules and for models that consist of reaction networks. So, it was decided to use the Jahan et al. [9] model for the central carbon metabolism of *E. coli*. This model encompasses the main steps of the carbon source metabolism and can predict the effects of multi layer regulations, such as allosteric effectors and gene regulation, on the biomass growth.

However, this model is constituted by 129 reactions and includes 51 species. As mentioned in Chapter 2, this algorithm can be very computationally intensive. This is explained because for each iteration, two random numbers must be generated and, the propensity of each reaction is re-calculated. If added to this, the number of reactions is very big, it is clear that the computation time will be very large as well. Therefore, while using this model, it was not possible to get relevant results. Figure 4.1 exemplifies the results that were obtained.

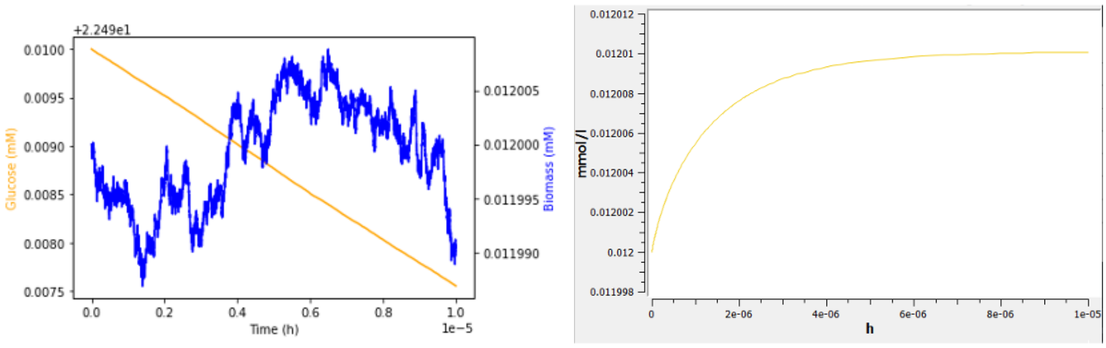

Figure 4.1: Comparison between one realisation of the stochastic simulation, depicting the evolution of the glucose (orange) and biomass (blue) concentration (left hand side plot), and the deterministic simulation depicting the evolution of biomass concentration [9] (right hand side plot).

The simulation depicted in Figure 4.1 only ran for a simulated time of $1 \cdot 10^{-5}$ h. The reason is mentioned above, the algorithm is very computationally demanding and thus simulating this high number of reactions takes a long time. This high demand is not only translating in the running time but as well to a very large memory requirement. Even recurring to a HPC with 400GB of RAM memory, it was not possible to simulate an amount of time with biological relevance.

Given this, from this first model still some conclusions can be drawn. The range of concentrations in which biomass varies is the same in the present simulation and the deterministic one conducted by the authors of the original work. This indicates that the strategy used here for the conversion between number of molecules and concentration units was successful. Also, it can be seen, that even though the trajectories are varying in the same range, the stochastic simulation dynamics has very different characteristics. The evolution is noisy when compared to the deterministic one which is a smooth line. This goes in hand with the goal of this work, which was to highlight the uncertainty associated with biological processes. Finally, the conclusion was reached, that the original Gillespie algorithm is not adequate for such a high number of reactions. Thus, in these cases, one of the optimised versions mentioned in Chapter 2 should be used.

4.2 *E. coli* fed-batch fermentation

Given the results from the first model, it was clear that a new strategy had to be adopted. So, an unstructured model was utilised. The model, adapted from the one of Anane et al. [26], simulates the growth of *E. coli* based on growth rate and substrate consumption equations and uses ODEs to describe the evolution of species over time. As explained in Chapter 3, each ODE corresponds to one reaction channel and so it was possible to reduce the number of reactions to be simulated to the number of species in question. In this first case, four.

This reduction in model complexity, reduced greatly the required computational power. Thus, even though the simulation was still considerably slower than the ones conducted deterministically, it was now possible to simulated the full duration of the fermentation. Since each realisation of the stochastic simulation can originate a different trajectory, each plot that is shown represents 100 realisations of the same simulation.

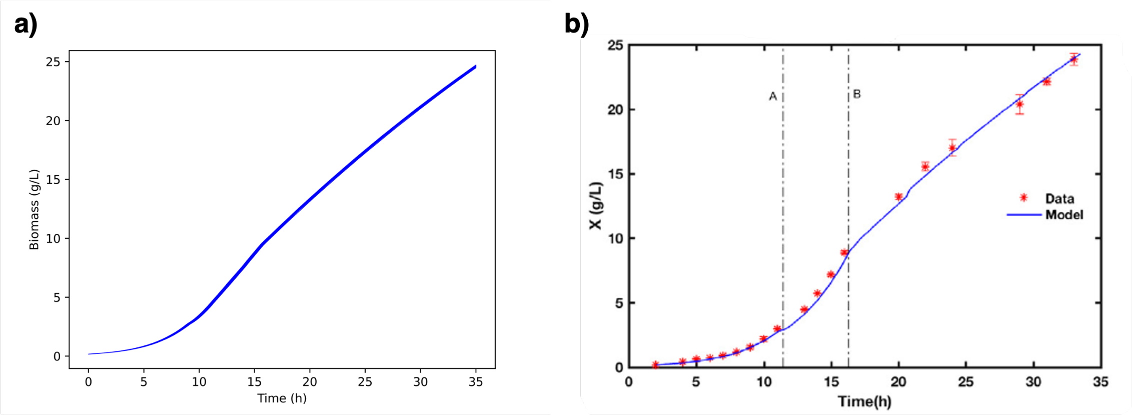

Figure 4.2: Evolution of biomass concentration over time: **a)** 100 realisations of the developed stochastic simulation model; **b)** results of the deterministic model (blue) and experimental data (red). Line A corresponds to the start of the exponential glucose feed fed-batch phase and line B to the start of the constant glucose feed fed-batch phase [26].

Starting by analysing the time evolution of the biomass concentration, it can be seen that there are no significant differences between the two plots. It is a plot characteristic of a fed-batch fermentation. There is a slower growth in the beginning, corresponding to the batch mode, which accelerates once the feeding is initiated. There is no exhaustion of the carbon source and thus biomass does not enters a stationary phase, unlike the characteristic curve of a batch fermentation.

Focusing solely on stochastic simulation, it can be observed that the line starts thinner and becomes wider until time has reached around 10 hours, from which point the line width is maintained. Since the plot is representing 100 realisations, the thinner the line, the higher the overlap between the simulations. So, even though the uncertainty is very small for this species, the smallest corresponds to the batch phase of the fermentation, increasing when the feed starts. This is in accordance with the expectation, because one more control variable is included in the system (the feed rate), increasing the sources of uncertainty.

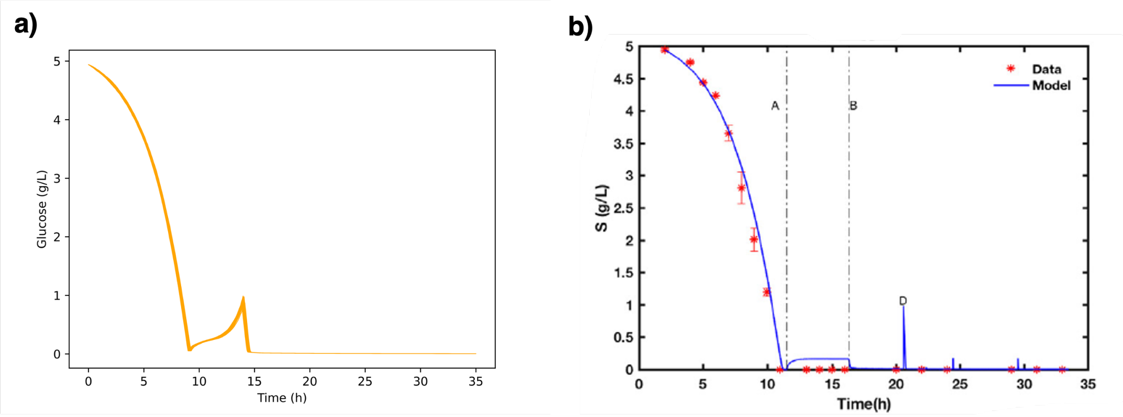

Figure 4.3: Evolution of glucose concentration over time: **a)** 100 realisations of the developed stochastic simulation model; **b)** results of the deterministic model (blue) and experimental data (red). Line A corresponds to the start of the exponential glucose feed fed-batch phase and line B to the start of the constant glucose feed fed-batch phase [26].

Regarding the evolution of the glucose concentration, the stochastic simulation is in general agreement with the deterministic results. Still, batch completion happens around 2 hours earlier than in the deterministic case. This happens in spite of exactly the same equations being used as well as the same initial conditions. Also, even though the exponential feeding phase lasts for the same number of hours, and the same feed equation is applied, a higher glucose concentration is achieved. Notwithstanding, this indicates that the performed simulation can reproduce the results to a good extent, even though not the exact same results are obtained. Although, these discrepancies do not seem to influence the behaviour of biomass, which is the variable of most interest in a fermentation operation as this one, where no other product is being produced.

Regarding the period after the exponential feed is switched to a linearly decreasing one, first it should be noted that the spikes observed in the deterministic simulation (plot **b)**), that start after around 20h, are due to glucose pulses introduced by the authors to test the robustness of the model. This is not done in the model developed in this thesis and thus there are no spikes present in plot **a)** . Secondly, in contrast to the biomass case, it is the final stage of the fermentation that has less uncertainty associated. The line is very thin after a fermentation time of 15h meaning that the 100 realisations are highly overlapping. At this stage, glucose concentration is made to be very low and constant. Because of this, on the stochastic algorithm, the glucose reaction channel is not selected as the reaction that should occur leaving little to no space for uncertainty.

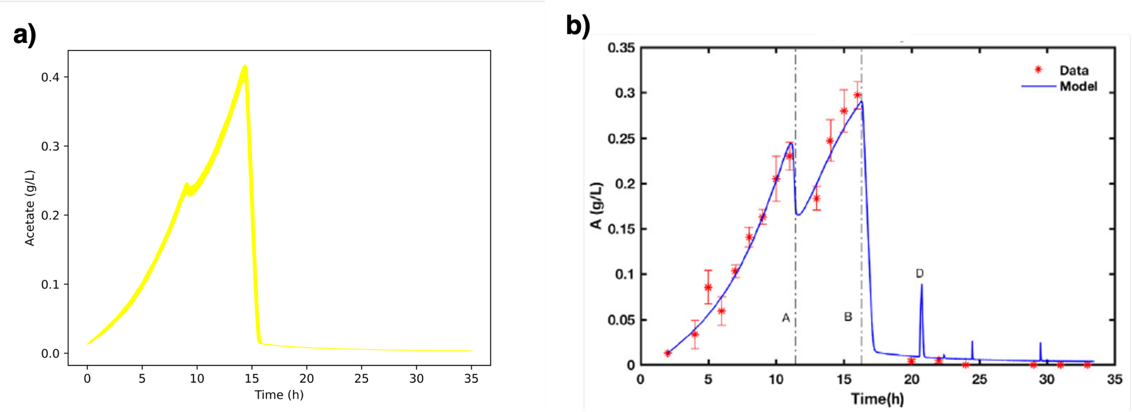

Figure 4.4: Evolution of acetate concentration over time: **a)** 100 realisations of the developed stochastic simulation model; **b)** results of the deterministic model (blue) and experimental data (red). Line A corresponds to the start of the exponential glucose feed fed-batch phase and line B to the start of the constant glucose feed fed-batch phase [26].

Now analysing the evolution of the acetate concentration, it is the variable with the biggest difference between the two plots thus far. The beginning of the plot is very similar, with the first peak in acetate concentration reaching around 0.25 g/L in both cases, only with a slight shift to the left in the stochastic simulation. This is due to the difference in batch completion time previously mentioned for glucose. As mentioned in Chapter 2.3, acetate production is related with both glucose and dissolved oxygen, so its evolution can be explained by the evolution of the other two. In the deterministic simulation it is possible to observe an abrupt decrease of the acetate concentration once the batch glucose is consumed; however, in the stochastic simulation results, this drop is very small and not very noticeable. An explanation to this can be that the SSA does not respond to these changes so rapidly. It could be due to the fact that in each time-step only one reaction is selected to occur and thus, when the biggest changes are occurring in glucose, the acetate reaction is selected less frequently.

After the feeding starts, the evolution of the plots is mostly the same, with the quick rise in glucose concentration, there is a rise in acetate production. In plot **a)**, the achieved concentration is higher, but this can be easily explained. Since the concentration does not drop to the same values, even though the difference is about the same, the maximum attained value is higher. Once the exponential feed is finished, there is a sharp decrease in acetate concentration which remains close to zero until the end of the fermentation. This part is equivalent in both plots and like in the glucose plots it is where the line is thinner and so less uncertainty is present.

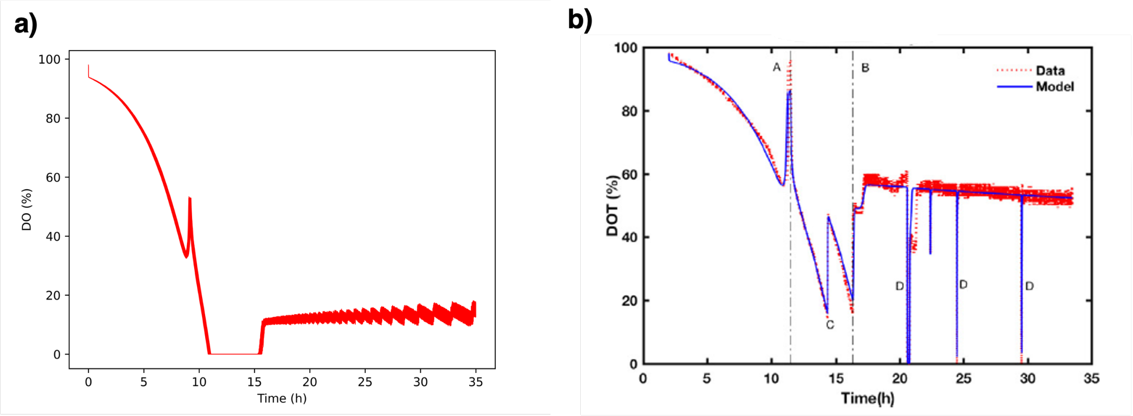

Figure 4.5: Evolution of dissolved oxygen concentration over time: **a)** 100 realisations of the developed stochastic simulation model; **b)** results of the deterministic model (blue) and experimental data (red). Line A corresponds to the start of the exponential glucose feed fed-batch phase and line B to the start of the constant glucose feed fed-batch phase [26].

The evolution of dissolved oxygen (DO) is where the stochastic simulation seems the least accurate. The general pattern is present but mostly ranges between very different values. Excluding the first five hours, the stochastic simulation always gives lower values than the deterministic simulation as well as the experimental data, even though the decrease and increase dynamics are captured. A possible explanation for the big discrepancy, is that the conversion for number of molecules in the case of oxygen is more complicated than the other species. So, it is probable that this conversion has a bigger impact and influences the rise and fall of the concentration. Overall, it does not seem that these discrepancies are affecting the behaviour of the remaining species; therefore it does not appear to be a big fault of this model.

4.3 Production of recombinant GFP

Having established the model for *E.coli* growth, with overall good results, the next step was to integrate the production of the proteins of interest into the model. For the production of GFP, the deterministic model available for comparison, from Muldbak et al. [38], was for a batch mode fermentation. Given this, the first simulation for GFP production was done for batch mode, so it could be possible to compare. The same equations as in the previous model were used, only the feed related terms were disregarded. Because of this, only the plots for GFP and glucose are included in this section, in order not to be repetitive. The remaining results of the model can be found in Appendix B.

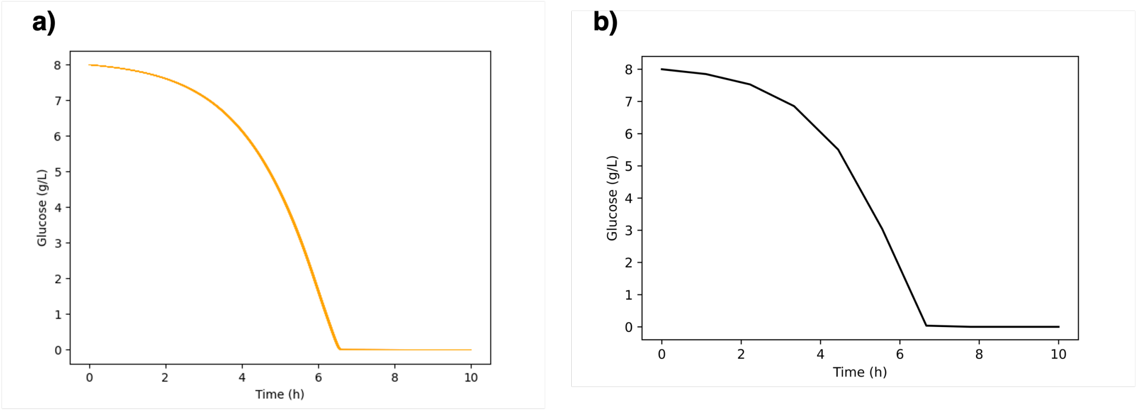

Figure 4.6: Evolution of glucose concentration over time: **a)** 100 realisations of the developed stochastic simulation model; **b)** results of the deterministic model [38].

As mentioned in Chapter 3.1.3.A, the production of GFP is controlled by an inducer switch. This is activated when glucose concentration decreases below 2 g/L. When this switch is off, the propensity of the GFP production reaction is set to 0. This explains why in Figure 4.7, GFP only starts increasing at the same time as glucose is seen to be below 2 g/L, as shown in Fig. 4.6. Consequently, it can be deduced that the switch dynamics are well captured by the model.

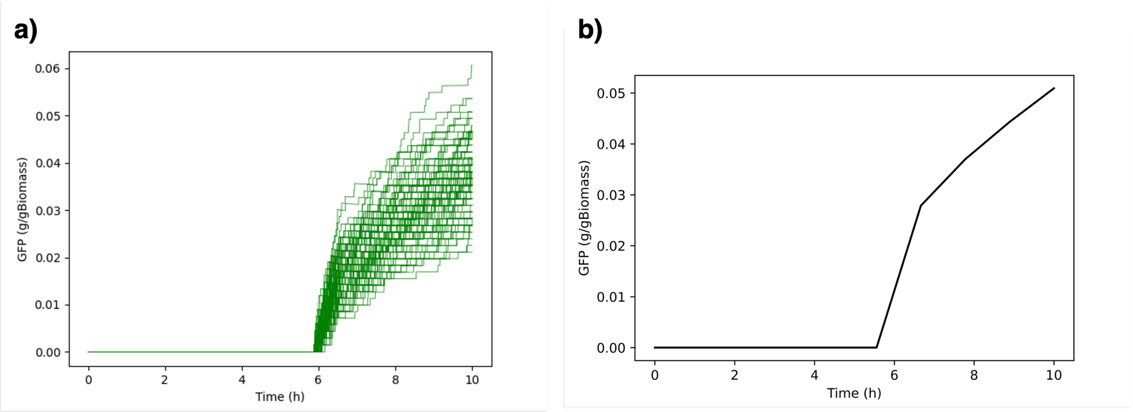

Figure 4.7: Evolution of GFP production over time: **a)** 100 realisations of the developed stochastic simulation model; **b)** results of the deterministic model [38].

Focusing now on the time evolution of the protein. Firstly, it is clear in this plot that the biggest variance in trajectories is observed. Nonetheless, the deterministic simulation line, falls within the prediction envelope of the stochastic simulation. As previously mentioned, stochasticity is particularly relevant when there is a small number of molecules involved. So, considering that at the peak, there is only 0.05 g_{GFP} per each $g_{Biomass}$, the order of magnitude of protein concentration is significantly lower than the remaining species. This is the most likely explanation for why the discrepancy is so obvious. Likewise, this is in accordance to what would be expected in a real-life production case. Product titers are subject to big variations, even if biomass titers remain constant. This is because there can be a a numerous amount of factors that influence it, since it is the production of a foreign protein.

4.3.1 Including fed-batch growth

Once the comparison with the deterministic model was made, it was decided to add a substrate feed to the model, since fed-batch mode is the most well-documented mode of operation in fermentations with the goal of protein production.

The feeding strategy used was the same as in the previous model for *E. coli* fed-batch growth. The substrate feeding would start when the concentration of glucose dropped below 0.5 g/L and the induction would occur 30 minutes later, in accordance with Aucoin et al. [31].

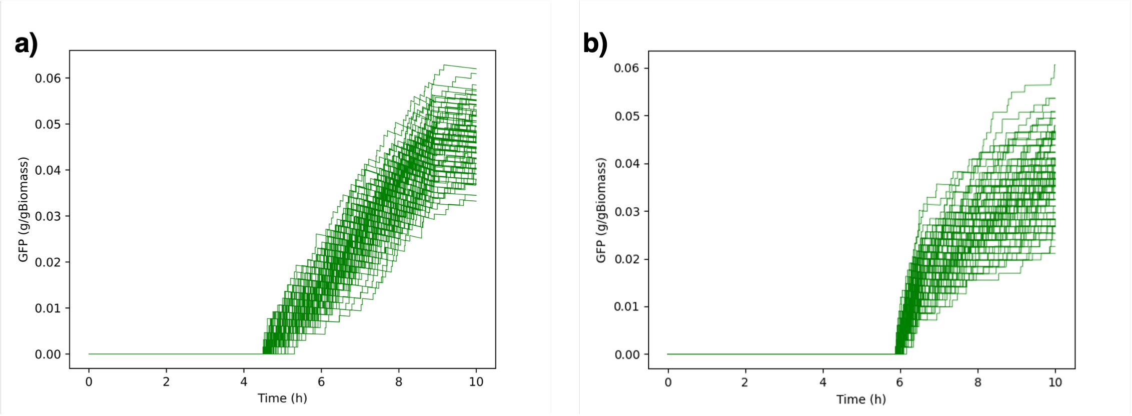

Figure 4.8: Evolution of GFP production over time: **a)** 100 realisations of the developed stochastic simulation model including a substrate feed; **b)** results of the same model for batch mode operation.

Comparing both results, it can be observed that the fed-batch mode is more advantageous, as expected. The production of protein starts earlier and the increase in production is faster. As well, the prediction space is now smaller. It means that there is less uncertainty associated with the production of protein which results in a more accurate prediction.

4.4 Production of rhGH

In the case of the production of rhGH, the dynamics are different. Instead of an inducer switch, the production of the protein is linked to the inducer concentration that enters the system as a feed stream.

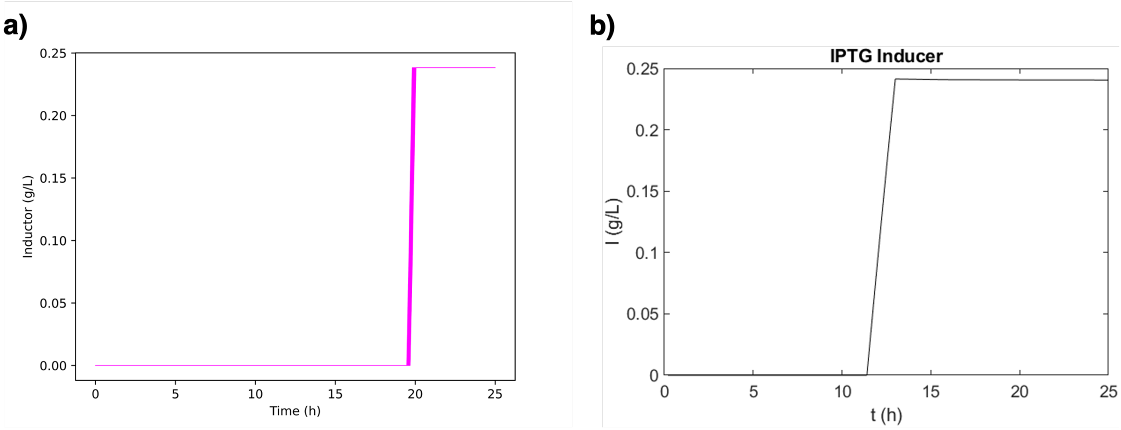

Figure 4.9: Evolution of rhGH production over time: **a)** 100 realisations of the developed stochastic simulation model; **b)** results of the deterministic model [35].

The inducer feed is set to start when biomass concentration reaches 20 g/L. As can be seen in Fig. B.4 of Appendix B, the plots for the biomass are different. However, since the results obtained by the implemented model are in agreement with the ones of Anane et al. [26] (the evolution is of the same shape), the discrepancy with the deterministic simulation is regarded as an error from the model proposed in [35]. Nonetheless, this means that this difference will propagate to the other species, which depend on the biomass concentration, such as the inducer or rhGH.

Thus, overlooking the fact that the inducer feed starts some hours later on plot **a)**, the evolution is otherwise very similar. The inducer concentration is null, followed by a very rapid increase to the desired inducing concentration (1 mM or 0.238 g/L), from which point it remains constant until the end of the fermentation.

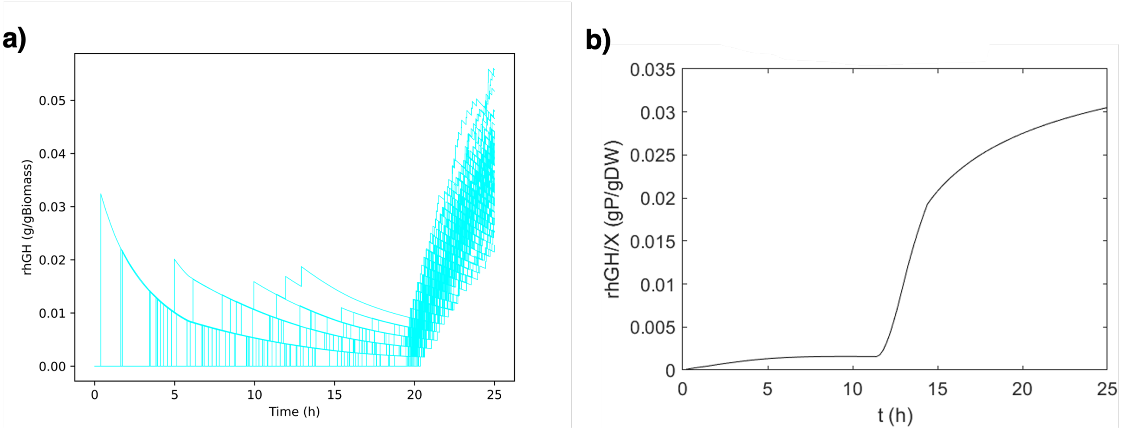

Figure 4.10: Evolution of rhGH production over time: **a)** 100 realisations of the developed stochastic simulation model; **b)** results of the deterministic model [35].

The model includes a term for constitutive expression in the protein production equation. Therefore, in both plots some production before the inducer feed is started. This phase seems to have almost as much uncertainty associated as the end of the fermentation. Especially at the beginning of the fermentation, when the biomass concentrations are very low, a very small concentration of protein is necessary to create a spike in relative protein production. As for the period when the inducer is being fed, the deterministic line is contained in the range predicted by the stochastic simulation. Regarding the associated uncertainty, the same reasoning as presented for GFP can be applied.

4.5 Parameter Estimation

After validating the model against the deterministic solutions, it was important to validate it against experimental data. The Anane et al. [26] model already included re-estimated parameters and was in accordance with experimental data. This can be observed in Figures 4.2 - 4.5. Then, a decision was made that the focus will be put on the protein production for this final task (parameter estimation).

4.5.1 For GFP production

The results of the simulation of the GFP production were compared with the experimental data of [31]. After adjusting some parameters of the model, such as the feed rate and substrate concentration in the feed, to match the ones described in the article, the simulation results were plotted against the experimental data (Fig. 4.11).

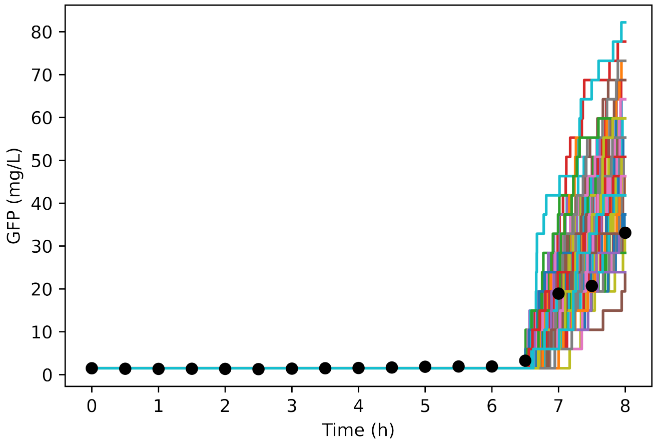

Figure 4.11: Comparison of the simulation results for GFP production (100 realisations) with experimental data (black dots) [31], using the parameters from the literature [38].

From this comparison, it can be seen that some of the realisations of the simulation can describe the experimental data satisfactorily. However, looking at the general trend, the model predictions are considerably higher than the experimental results. In order to obtain a better model fit, some changes and adjustments were performed to the parameters in the equation for protein production (3.18).

$$\frac{dP}{dt} = \left(Y_{P/X}\mu + \beta\right) X - \frac{F}{V}P. \tag{3.18}$$

Looking again at this equation, there are two parameters that influence the production rate, $Y_{P/X}$ and β, the protein to biomass yield and a protein production constant, respectively. The first had a value of 0.05 and the second of 0.014. and then evaluate if the model predictions fitted better to the experimental data. The first parameter was selected. This parameter was varied by ±50%. The algorithm was stopped when 100 parameters are accepted that guarantee a discrepancy threshold (ϵ) below 5.

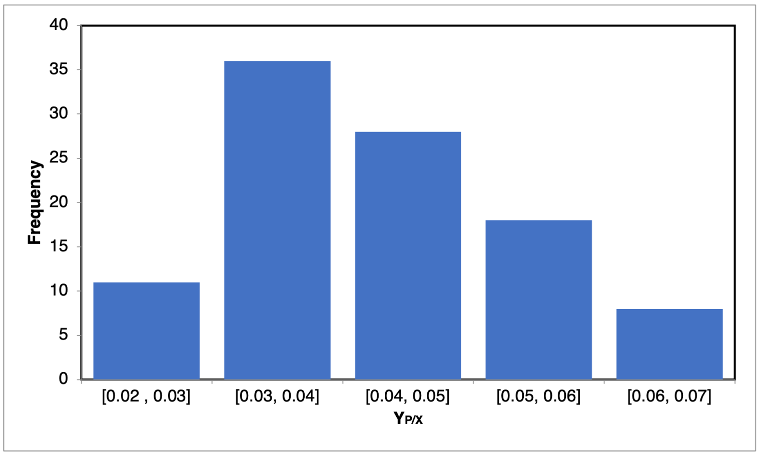

Figure 4.12: Histogram illustrating the distribution of the accepted parameters in different intervals.

The 100 accepted values are plotted in a histogram to determine which interval of parameters leads to the best simulation results. As can be seen in Fig. 4.12, this interval is $[0.03, 0.04]$. Hence, the values that belonged to this range are taken and their average value is calculated, this corresponded to the average value of the interval itself, 0.035.

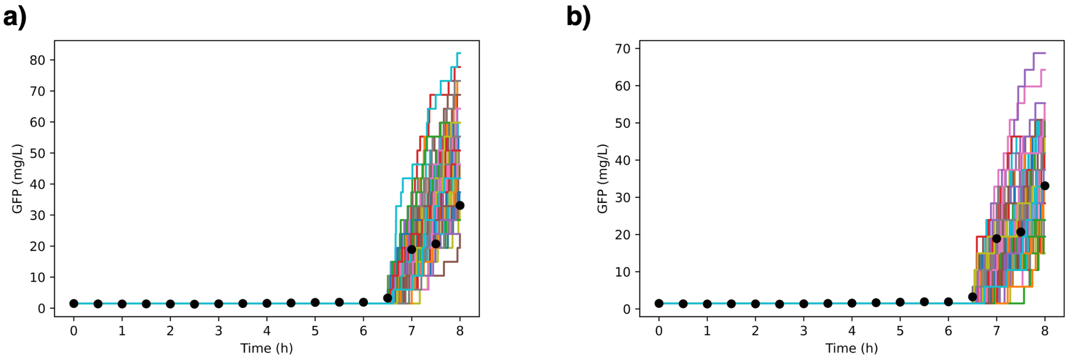

Figure 4.13: Comparison of the simulation results for GFP production (100 realisations) with experimental data (black dots) [31]: **a)** using the literature parameter value; **b)** using the re-estimated parameter.

In Figure 4.13, plot **a)** represents the simulation results using the value of of 0.05 for $Y_{P/X}$, and in plot **b)**, the results obtained for the new parameter value of 0.035. Even though the prediction range is still rather wide, just like in Figure 4.7, the improvement is evident. The experimental data now falls in the middle of the simulation results, instead of in the lower range of the simulation results like previously. In summary, the re-estimation of this parameter was enough to see a significant improvement in the simulation results, and thus no further estimation was performed. Now, with this new parameter, the model can predict well the envelope for the production of the protein.

4.5.2 For rhGH production

Moving on to rhGH, finding literature with experimental results for the conditions for which the model was developed proved to be very difficult. Hence, it was then assumed that the best approach was to use the experimental data from Bylund et al. [39]. The issue was that the plots presented in the article all had arbitrary units (au). So, given some concrete results presented in the text and some knowledge of the typical range for these values, some interpretation assumptions were made. The most relevant were the relative protein production units, where it was assumed that 100 a.u corresponded to 100 $mg_{Protein}/g_{Biomass}$. Considering this, the parameter value that was obtained might not be the most accurate, but this exercise was useful for observing the model's sensitivity to the changes in the parameter.

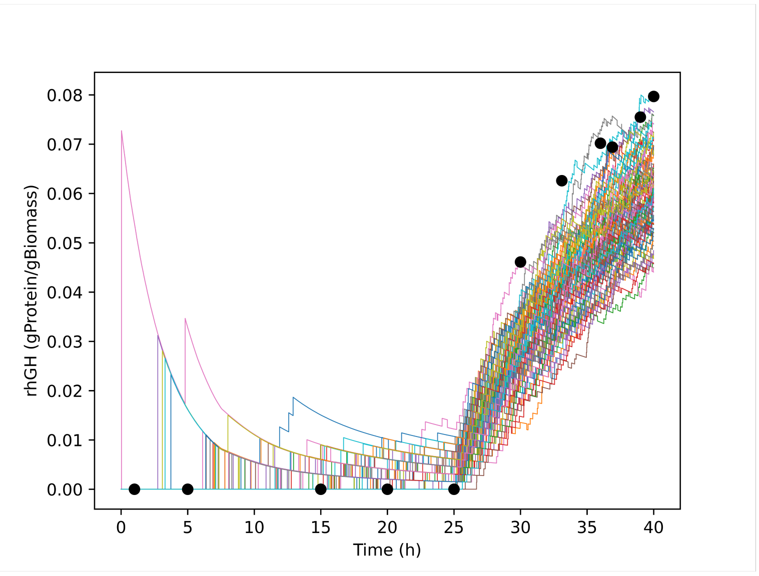

Figure 4.14: Comparison of the simulation results for rhGH production (100 realisations) with experimental data (black dots). The experimental data was taken from Bylund et al. [39] Figure 6 a), using the results corresponding to the SDR bioreactor.

Figure 4.14 plots the experimental data against the simulation results. Of note is that, the experimental results provided data from the point when induction started (25 h), and so, the values of the time preceding this were assumed to be 0. Since the model predicts some constitutive protein production in this stage, the initial discrepancy was expected. However, it was decided to not remove or re-estimate the parameter responsible for this, because said production is documented in the literature. Concerning the period for which there is effectively data available, it can be observed that the model predictions are, in every realisation, too low when compared to the experimental results.

$$\frac{dP}{dt} = \left(k_1\mu\frac{I}{I+K_i} + k_2\right)X - k_3P - P\frac{F}{V}. \tag{3.19}$$

Looking once more into Eq. 3.19, four parameters influence the production of rhGH, k_1, k_2, K_i and k_3. These represent induced and constitutive protein biosynthesis rates,the induction constant and a constant first order degradation term, respectively. Similar to the previous case, it was decided to start with the re-estimation of k_1 which had an initial value of 0.32. In this case, the value chosen for ϵ was 0.01. Once 100 parameters were accepted, they were plotted in a histogram (Fig. 4.15).

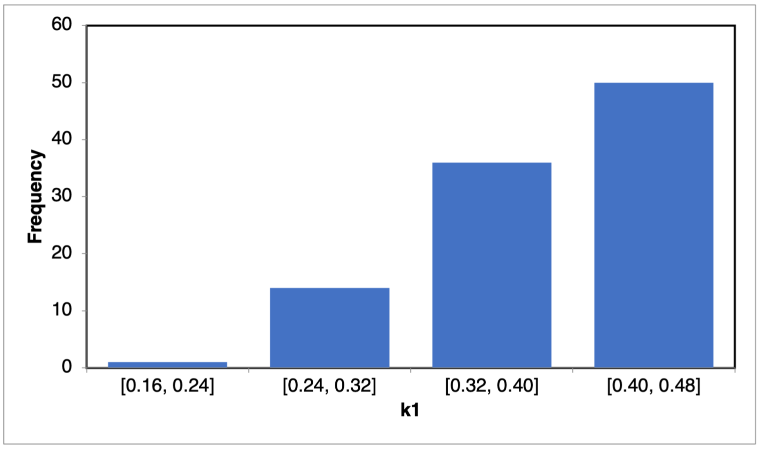

Figure 4.15: Histogram illustrating the distribution of the accepted parameters in different intervals.

From this, it can be established that the interval chosen is $[0.40, 0.48]$. From the values within this interval, the average value is picked, corresponding to 0.44.

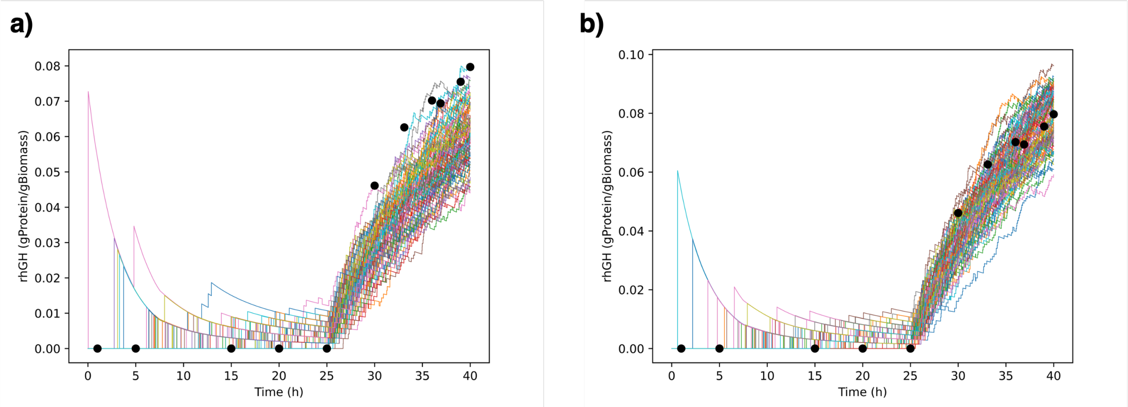

Figure 4.16: Comparison of the simulation results for rhGH production (100 realisations) with experimental data (black dots)[39]: **a)** using the literature parameter value; **b)** using the re-estimated parameter.

In Figure 4.16, plot **a)** represents the simulation using the value of 0.32 for k_1 and plot **b)**, the new value of 0.44. First of all, it is observed that now the experimental data is falling mostly in the center of the simulation predictions. This improvement is very significant, since, to start with, these values were in every case outside of the prediction space. Similarly, the prediction range was decreased giving a smaller envelope. All in all, the re-estimated parameter made the prediction not only more accurate, but also more constrained. It should be noted once again, that the interpretation of the experimental results required assumptions, and thus this new parameter value should not be taken by itself. Instead, this procedure illustrated that the model is sensitive to this parameter and for the hypothetical case of the presented experimental results, its decrease is beneficial for the quality of the predictions.

5

Conclusion and future perspectives

Contents

5.1 Conclusions

Stochasticity is an inherent part of biological processes, since randomness and noise are an essential characteristic of living beings. Thus, this should be taken into account when trying to model such systems in order to gain a better understanding of certain biological phenomena.

The production of recombinant proteins is a major part of the biotechnological industry, since these can have several applications ranging from therapeutics, to food supplements and detergents, to mention a few. Part of the goal of Industry 4.0, is to get the biomanufacturing industry more automated and digitised. Models of key processes are in this an important to tool to gain better understanding and allow the prediction and control of different production scenarios.

Given these two things it seemed relevant to adapt these bioprocess models, namely fermentation, to include stochasticity. The developed work aimed then at the application of the SSA to predictive models of recombinant protein production in bioreactors.

The first approach used a detailed structured model to predict *E. coli*'s growth. It was concluded that simulating a model with this level of complexity was not feasible recurring to the Gillespie direct method. It was computationally too intensive, preventing that relevant results could be obtained.

Moving on, unstructured models were developed. This allowed to reduce to a very significant extent the number of reactions as well as the number of simulated variables. Both these things relieved the computational burden of the model, allowing for simulation times that gave significant results. The results of the simulations were in general good, with good agreement with the deterministic results.

However, the results for all the species excluding the recombinant proteins showed very little uncertainty. So, on the one hand the model was validated by the deterministic results, but on the other hand, it did not seem to bring a significant advantage. The results predicted a solution space not much bigger than the deterministic model, while having a much slower computation. So, even though it is a good result in terms of the applicability of this type of algorithm to large scale bioproduction, the deterministic solution seems to have the upper hand. Its calculation of trajectories is almost immediate and gives a good description of the average behaviour of the system, which for the large number of molecules involved is adequate.

Regarding the case studies for both recombinant proteins, GFP and rhGH, the results obtained were very different than the ones described above. Considering that the concentrations of these molecules are significantly smaller when compared to the remaining species, it was expected that the uncertainty captured by the model would be higher. This is exactly what was observed. Even though the deterministic solution was in all cases included in the stochastic simulation envelope, the stochastic information provided more information on the different possible scenarios. So, as a compromise for losing a single "exact" solution, information about the uncertainty associated with the process is gained.

Moreover, this positive outcome also attested that the assumptions made in developing the propensity equations were valid under this scenario.

Finally, using ABC rejection sampler, new parameters for the equations of protein production were estimated. Even though it was decided to perform this for only one parameter for each protein, in order to reduce the computational effort, significant improvements were observed for both the proteins.

To conclude, the goal of this work was achieved, providing one more example of successful implementation of the SSA in a case that presumably had not been done previously, the production of recombinant proteins.

5.2 Future Perspectives

Now, with the model successfully implemented and proof of concept of its usability for industry relevant bioproduction cases, it is important to explore the insights that can be drawn with this type of model and as well point out what can be improved.

It was outside the objective of this thesis and as well there was not enough time to conduct some production optimisation studies. Nonetheless, in future work on this subject, the model can be used to study how different initial conditions and process parameters can influence the production, and help determine optimal conditions. Also, the inherent uncertainty of the model can be used to study its sensitivity to the different parameters, and with that draw conclusions on which parameters can influence production the most.

On trying to improve the efficiency of the model, different strategies can be proposed. One is to utilise one of the more recent methods mentioned in Chapter 2, like the *Next reaction method* or the *τ-leaping method*. This could be especially relevant when trying to implement a structured model, as the first one herein proposed.

Another suggestion is on the implementation of a hybrid model. As it has been mentioned, the species excluding recombinant protein, had very little uncertainty associated when using this algorithm. Because of that, an approach could be adopted where biomass and glucose are modelled recurring to deterministic equations and protein production is modelled in a stochastic way. This way, the computation time would be greatly decreased, without loosing the advantages of stochastic simulation on the species for which it is relevant.

Bibliography

[1] H. Narayanan, M. F. Luna, M. Stosch, M. N. Cruz Bournazou, G. Polotti, M. Morbidelli, A. Butté, and M. Sokolov, “Bioprocessing in the digital age: The role of process models,” *Biotechnology Journal*, vol. 15, no. 1, p. 1900172, 2019.

[2] I. A. Udugama, P. C. Lopez, C. L. Gargalo, X. Li, C. Bayer, and K. V. Gernaey, “Digital twin in biomanufacturing: Challenges and opportunities towards its implementation,” *Systems Microbiology and Biomanufacturing*, vol. 1, no. 3, p. 257–274, 2021.

[3] M. Debnath, G. B. Prasad, and P. S. Bisen, “Biopharmaceutical industry and health care,” *Molecular Diagnostics: Promises and Possibilities*, p. 413–424, 2009.

[4] D. J. Warne, R. E. Baker, and M. J. Simpson, “Simulation and inference algorithms for stochastic biochemical reaction networks: From basic concepts to state-of-the-art,” *Journal of The Royal Society Interface*, vol. 16, no. 151, p. 20180943, 2019.

[5] M. Spooner, D. Kold, and M. Kulahci, “Harvest time prediction for batch processes,” *Computers amp; Chemical Engineering*, vol. 117, p. 32–41, 2018.

[6] A. Tsopanoglou and I. Jiménez del Val, “Moving towards an era of hybrid modelling: Advantages and challenges of coupling mechanistic and data-driven models for upstream pharmaceutical bioprocesses,” *Current Opinion in Chemical Engineering*, vol. 32, p. 100691, 2021.

[7] L. Mears, S. M. Stocks, M. O. Albaek, G. Sin, and K. V. Gernaey, “Mechanistic fermentation models for process design, monitoring, and control,” *Trends in Biotechnology*, vol. 35, no. 10, p. 914–924, 2017.

[8] K. V. Gernaey, “A perspective on pse in fermentation process development and operation,” *12th International Symposium on Process Systems Engineering and 25th European Symposium on Computer Aided Process Engineering*, p. 123–130, 2015.

[9] N. Jahan, K. Maeda, Y. Matsuoka, Y. Sugimoto, and H. Kurata, "Development of an accurate kinetic model for the central carbon metabolism of *Escherichia coli*," *Microbial Cell Factories*, vol. 15, no. 1, 2016.

[10] L. Mears, S. M. Stocks, M. O. Albaek, G. Sin, and K. V. Gernaey, "Application of a mechanistic model as a tool for on-line monitoring of pilot scale filamentous fungal fermentation processes-the importance of evaporation effects," *Biotechnology and Bioengineering*, vol. 114, no. 3, p. 589–599, 2016.

[11] D. T. Gillespie, "Exact stochastic simulation of coupled chemical reactions," *The Journal of Physical Chemistry*, vol. 81, no. 25, p. 2340–2361, 1977.

[12] A. M. Kierzek, "Stocks: Stochastic kinetic simulations of biochemical systems with gillespie algorithm," *Bioinformatics*, vol. 18, no. 3, p. 470–481, 2002.

[13] C. V. Rao and A. P. Arkin, "Stochastic chemical kinetics and the quasi-steady-state assumption: Application to the gillespie algorithm," *The Journal of Chemical Physics*, vol. 118, no. 11, p. 4999–5010, 2003.

[14] A. V. Isaza, "Stochastic modeling of bioreactors," Master's thesis, Universidad Nacional de Colombia, 2021.

[15] M. A. Gibson and J. Bruck, "Efficient exact stochastic simulation of chemical systems with many species and many channels," *The Journal of Physical Chemistry A*, vol. 104, no. 9, p. 1876–1889, 2000.

[16] D. F. Anderson, "A modified next reaction method for simulating chemical systems with time dependent propensities and delays," *The Journal of Chemical Physics*, vol. 127, no. 21, p. 214107, 2007.

[17] D. T. Gillespie, "Approximate accelerated stochastic simulation of chemically reacting systems," *The Journal of Chemical Physics*, vol. 115, no. 4, p. 1716–1733, 2001.

[18] M. B. Elowitz, A. J. Levine, E. D. Siggia, and P. S. Swain, "Stochastic gene expression in a single cell," *Science*, vol. 297, no. 5584, p. 1183–1186, 2002.

[19] T. Lu, D. Volfson, L. Tsimring, and J. Hasty, "Cellular growth and division in the gillespie algorithm," *Systems Biology*, vol. 1, no. 1, p. 121–128, 2004.

[20] M. P. Xavier, C. R. Bonin, R. W. dos Santos, and M. Lobosco, "On the use of gillespie stochastic simulation algorithm in a model of the human immune system response to the yellow fever vaccine,"

2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), vol. 6, no. 1, Jan 2017.

[21] G. J. Gopal and A. Kumar, “Strategies for the production of recombinant protein in *Escherichia coli*,” *The Protein Journal*, vol. 32, no. 6, p. 419–425, 2013.

[22] G. L. Rosano, E. S. Morales, and E. A. Ceccarelli, “New tools for recombinant protein production in *Escherichia coli* : A 5 year update,” *Protein Science*, vol. 28, no. 8, p. 1412–1422, 2019.

[23] M. De Mey, S. De Maeseneire, W. Soetaert, and E. Vandamme, “Minimizing acetate formation in *E. coli* fermentations,” *Journal of Industrial Microbiology amp; Biotechnology*, vol. 34, no. 11, p. 689–700, 2007.

[24] C. R. Dittrich, R. V. Vadali, G. N. Bennett, and K.-Y. San, “Redistribution of metabolic fluxes in the central aerobic metabolic pathway of *E. coli* mutant strains with deletion of the acka-pta and poxb pathways for the synthesis of isoamyl acetate,” *Biotechnology Progress*, vol. 21, no. 2, p. 627–631, 2008.

[25] G. Stephanopoulos, “Metabolic engineering,” *Biotechnology and Bioengineering*, vol. 58, no. 2-3, p. 119–120, 1998.

[26] E. Anane, D. C. López C, P. Neubauer, and M. N. Cruz Bournazou, “Modelling overflow metabolism in *Escherichia coli* by acetate cycling,” *Biochemical Engineering Journal*, vol. 125, p. 23–30, 2017.

[27] D. C. Prasher, “Using gfp to see the light,” *Trends in Genetics*, vol. 11, no. 8, p. 320–323, 1995.

[28] H.-H. Gerdes and C. Kaether, “Green fluorescent protein: Applications in cell biology,” *FEBS Letters*, vol. 389, no. 1, p. 44–47, 1996.

[29] J. Kong, Y. Wang, W. Qi, M. Huang, R. Su, and Z. He, “Green fluorescent protein inspired fluorophores,” *Advances in Colloid and Interface Science*, vol. 285, p. 102286, 2020.

[30] M. Zimmer, “Green fluorescent protein (gfp): applications, structure, and related photophysical behavior,” *Chemical Reviews*, vol. 102, no. 3, p. 759–782, 2002.

[31] M. G. Aucoin, V. McMurray-Beaulieu, F. Poulin, E. B. Boivin, J. Chen, F. M. Ardelean, M. Cloutier, Y. J. Choi, C. B. Miguez, M. Jolicoeur, and et al., “Identifying conditions for inducible protein production in *E. coli*: Combining a fed-batch and multiple induction approach,” *Microbial Cell Factories*, vol. 5, no. 1, 2006.

[32] M. Rezaei and S. H. Zarkesh-Esfahani, “Optimization of production of recombinant human growth hormone in *Escherichia coli*,” *Journal of Research in Medical Sciences : the Official Journal of Isfahan University of Medical Sciences*, vol. 17, no. 7, pp. 681–685, 2012.

[33] M. J. Henwood, A. Grimberg, and T. Moshang, “Expanded spectrum of recombinant human growth hormone therapy,” *Current Opinion in Pediatrics*, vol. 14, no. 4, p. 437–442, 2002.

[34] D. S. Hardin, S. F. Kemp, and D. B. Allen, “Twenty years of recombinant human growth hormone in children: Relevance to pediatric care providers,” *Clinical Pediatrics*, vol. 46, no. 4, p. 279–286, 2007.

[35] E. N. Stavad, “Dynamic modelling and simulation of the production of human growth hormone - towards the development of digital twins in biopharma,” Bachelor’s Project, Technical University of Denmark, 2021.

[36] H. J. Chae, M. P. Delisa, H. J. Cha, W. A. Weigand, G. Rao, and W. E. Bentley, “Framework for online optimization of recombinant protein expression in high-cell-density *Escherichia coli* cultures using gfp-fusion monitoring,” *Biotechnology and Bioengineering*, vol. 69, no. 3, pp. 275–285, 1999.

[37] J. Ruiz, G. González, C. de Mas, and J. López-Santín, “A semiempirical model to control the production of a recombinant aldolase in high cell density cultures of *Escherichia coli*,” *Biochemical Engineering Journal*, vol. 55, no. 2, p. 82–91, 2011.

[38] M. Muldbak, C. Gargalo, U. Krühne, I. Udugama, and K. V. Gernaey, “Digital twin of a pilot-scale bio-production setup,” *Computer Aided Chemical Engineering*, p. 1417–1422, 2022.

[39] F. Bylund, A. Castan, R. Mikkola, A. Veide, and G. Larsson, “Influence of scale-up on the quality of recombinant human growth hormone,” *Biotechnology and Bioengineering*, vol. 69, no. 2, p. 119–128, 2000.

A

Code for the developed model

This appendix shows the Python code for the developed model. It is the code corresponding to the rhGH production model, since this was the msot complex one.

Listing A.1: PYTHON Code

```python
import numpy as np
import random
import math
import copy

n=6.022e23
Vsim=1e-18
mwBM=25
mwS=180.156
mwA=60.052
mwI=238.31
```

```
mwP=22124.76
mwH2O=18.01528
H=14000
C={'BM':n*Vsim/mwBM,'S':n*Vsim/mwS,'A':n*Vsim/mwA,'I':n*Vsim/mwI,'P':n*Vsim/mwP,'DO':n*Vsim/H} #C to N conversion constant

#parameters
If=50
Sf=600
k1=0.32
k2=0.00044
k3=0.6
Ki=0.55
Kap=0.5088
Ksa=0.0128
Ko=0.0001
Ks=0.0381
Kia=1.2602
Kis=1.8363
Pamax=0.41
qAMAX=0.1148
Qm=0.0129
qSMAX=0.6356
Yoa=0.5221
Yxa=0.5718
Yem=0.65
Yos=1.5722
Yxs=Yem+Qm
Yxsof=0.229
muset=0.3
Cx=0.488
Cs=0.391
Sc=1.5
Yas=0.9097
Xc=4
Vi=1.4e3
kla=220
DOTstar=99
```

```

class rxnetwork:
    def __init__ (self, X,react):

        self.M=len(react) #number of reactions

        self.N=len(X) #number of species

        self.r=react #stoichiometry

        self.X0=X #initial conditions


def Gillespie(tend):

    num_simulations=100

    for i in range(num_simulations):

        X={'BM':[1],'S':[2],'A':[0.0129],'I':[0],'P':[0],'DO':[98]} #initial conditions
        V=[Vi] #variable for volume update
        X0=[] #variable to store X before feeding start

        n=6.022e23
        Vsim=1e-18
        mwBM=25
        mwS=180.156
        mwA=60.052
        mwI=238.31
        mwP=22124.76
        mwH2O=18.01528
        H=14000
        C={'BM':n*Vsim/mwBM,'S':n*Vsim/mwS,'A':n*Vsim/mwA,'I':n*Vsim/mwI,'P':n*Vsim/mwP,'DO':n*Vsim/H}

        for key in X:
            X[key][0]=X[key][0]*C[key]

        a=[[] for x in range (6)] #list for propensities

        def prp(t, tf ,X):

            if tf[-1]==0:
                Fs=0
            else:
                if tf[-1]<=3:
                    Fs=((X0[0]*Vi*muset)/(Yxs*Sf))*math.exp(muset*tf[-1])
                elif t[-1]>3:
                    if (X['S'][-1]/C['S'])>=0.03:
                        Fs=3e3*-120*tf[-1]
                        if Fs<=0:
                            Fs=0
                    else:
                        Fs=((X0[0]*Vi*muset)/(Yxs*Sf))*math.exp(muset*tf[-1])


            if X['BM'][-1]/C['BM']<20:
                Fi=0
            else:
                if (X['I'][-1]/C['I'])<0.238:
                    Fi=20
                else:
                    Fi=0.238*Fs/(If-0.238)

            F=Fi+Fs

            if tf[-1]>0:
                V.append(V[-1]+(F*(tf[-1]-tf[-2])))

            else:
                V.append(Vi)

            qS=(qSMAX/(1+((X['A'][-1]/C['A'])/Kia)))*((X['S'][-1]/C['S'])/((X['S'][-1]/C['S'])+Ks))
            qSA=(qAMAX/(1+(qS/Kis)))*((X['A'][-1]/C['A'])/((X['A'][-1]/C['A'])+Ksa))
            qSOF=Pamax*(qS/(qS+Kap))
            qSOX=(qS-qSOF)*(X['DO'][-1]/C['DO'])/((X['DO'][-1]/C['DO'])+Ko)
            mu=(qSOX-Qm)*Yem+qSOF*Yxsof+qSA*Yxa
            pA=qSOF*Yas
            qA=pA-qSA

            if X['S'][-1]<=0:
                mu=0

            a[0]=mu*X['BM'][-1]-X['BM'][-1]*F/V[-1]
            a[1]=Fs/(V[-1])*(Sf*C['S']-X['S'][-1])-qS*X['BM'][-1]*(mwBM/mwS)
            a[2]=qA*X['BM'][-1]*(mwBM/mwA)-X['A'][-1]*F/V[-1]
            a[3]=(Fi/V[-1])*(If*C['I']-X['I'][-1])-Fs/V[-1]*X['I'][-1]
            a[4]=((k1*mu*(X['I'][-1]/C['I']))/(Ki+(X['I'][-1]/C['I']))+k2)*X['BM'][-1]*(mwBM/mwP)-X['P'][-1]*F/V[-1]
            qSan=(qSOX-Qm)*Yem*Cx/Cs
            qo=Yos*(qSOX-qSan)+qSA*Yoa
            a[5]=kla*(DOTstar*C['DO']-X['DO'][-1])-qo*X['BM'][-1]*mwBM

            return a

        r=[] #stoichiometry matrix

        dict1=copy.deepcopy(X)
        for x in dict1:
            dict1[x]=0
        dict1['BM']=1
        r.append(dict1)

        dict4=copy.deepcopy(X)
        for x in dict4:
            dict4[x]=0
        dict4['S']=1
        r.append(dict4)

        dict5=copy.deepcopy(X)
        for x in dict5:
            dict5[x]=0
        dict5['A']=1
        r.append(dict5)

        dict7=copy.deepcopy(X)
        for x in dict7:
            dict7[x]=0
        dict7['I']=1
        r.append(dict7)

        dict9=copy.deepcopy(X)
        for x in dict9:
            dict9[x]=0
        dict9['P']=1
        r.append(dict9)

        dict11=copy.deepcopy(X)
        for x in dict11:
            dict11[x]=0
        dict11['DO']=1
        r.append(dict11)

        rx=rxnetwork(X,r)

        t=[0] #initial time
        tf=[0] #variable to store t after feeding start
        X=rx.X0 #set X to initial conditions
        stc=rx.r #set stoichiometry matrix

        rev=copy.deepcopy(stc) #create matrix for stoichiometry of "reverse" reaction
        for i in range(len(stc)):
            for key in stc[i]:
                rev[i][key]=-stc[i][key]

        while t[-1]<tend:

```

```
        notabsprp=prp(t,tf,X) #actual propensity values

        updtprp=copy.deepcopy(notabsprp) #absolute propensity values
        for i in range(len(updtprp)):
            updtprp[i]=abs(updtprp[i])

        prp_sum=sum(updtprp) #sum of all propensities

        if prp_sum==0: #stop algorithm when there is no more substrate molecules
            break

        rand=random.uniform(0,1)

        tau=abs(1/prp_sum*np.log(1/rand)) #timestep

        t.append(t[-1]+tau)

        rand2=random.uniform(0,1)

        for j in range(rx.M): #selection of reaction channel
            if rand2*prp_sum>sum(updtprp[:j]) and rand2*prp_sum<=sum(updtprp[:j+1]):
                for key in X:
                    if notabsprp[j]>0:
                        X[key].append(X[key][-1]+stc[j][key])
                    elif notabsprp[j]<0:
                        X[key].append(X[key][-1]+rev[j][key])

        for key in X: #condition to prevent negative concentration values
            if X[key][-1]<0:
                X[key][-1]=0

        if (X['S'][-1]/C['S'])<0.5: #initiate substrate feed
            tf.append(tf[-1]+tau)
            X0.append(X['BM'][-1]/C['BM'])
        else:
            if tf[-1]>0:
```

```python
                    tf.append(tf[-1]+tau)

        conc=copy.deepcopy(X) #convert N to C
        for key in X:
            for i in range(len (X[key])) :
                conc[key][i]=X[key][i]/C[key]

        pratio=[0 for x in range(len(conc['BM']))]
        for i in range(len(conc['BM'])):
            pratio[i]=conc['P'][i]/conc['BM'][i]

    return (t, conc)
```

B

Additional simulation results

This appendix presents the plots obtained throughout the results of thesis that were not considered relevant to include in Chapter 4.

B.1 GFP production

Figures B.1 and B.2illustrate the comparison between the stochastic simulation and deterministic results for biomass and acetate. There is no relevant difference to be commented.

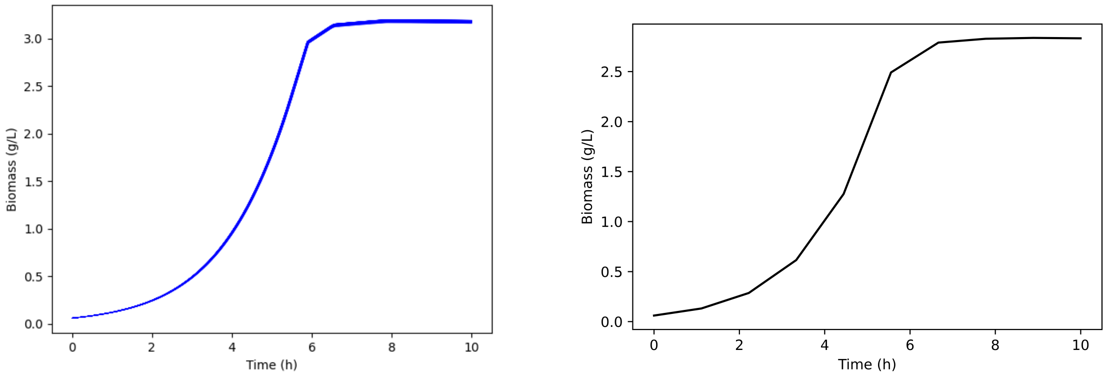

Figure B.1: Evolution of biomass overtime: On the left side, 100 realisations of the developed stochastic simulation model; on the right side, results of the deterministic model [38].

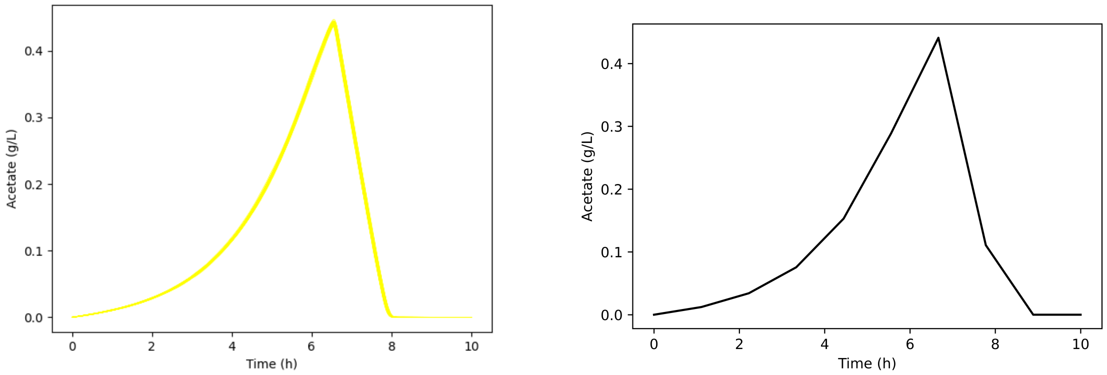

Figure B.2: Evolution acetate overtime: On the left side, 100 realisations of the developed stochastic simulation model; on the right side, results of the deterministic model [38].

Figure B.3 illustrates the comparison between the stochastic simulation and deterministic results for dissolved oxygen. The observed differences can be explained by what is suggested in Chapter 4 for Model 4.2.

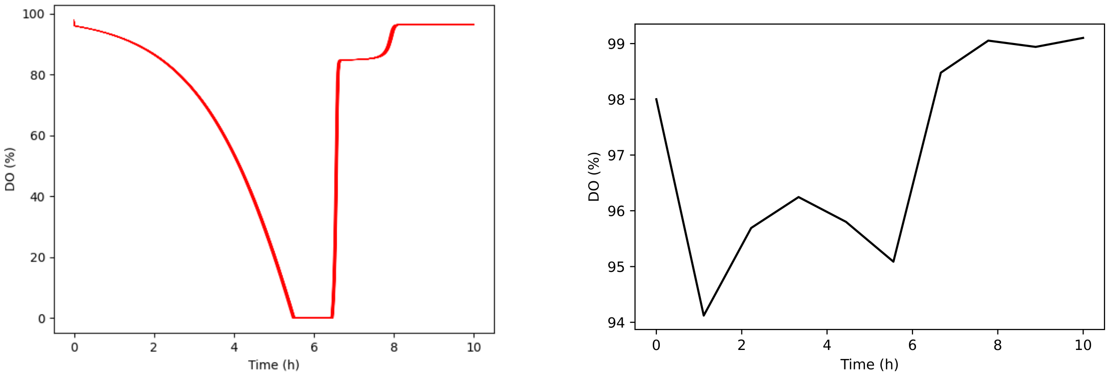

Figure B.3: Evolution of dissolved oxygen overtime: On the left side, 100 realisations of the developed stochastic simulation model; on the right side, results of the deterministic model [38].

B.2 rhGH production

Figure B.4 illustrates the comparison between the stochastic simulation and deterministic results for biomass. The differences are commented in Chapter 4.4.

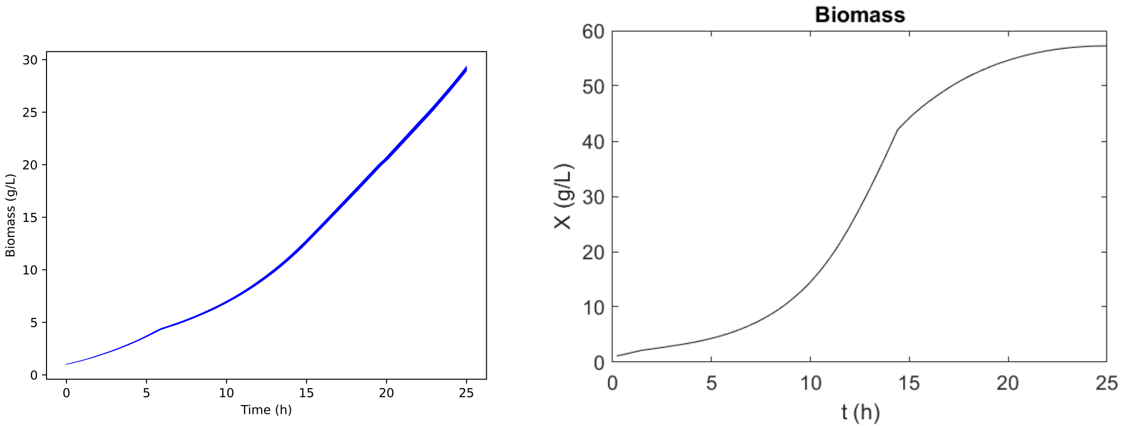

Figure B.4: Evolution of biomass overtime: On the left side, 100 realisations of the developed stochastic simulation model; on the right side, results of the deterministic model [35].

Figures B.5 and B.6 illustrate the comparison between the stochastic simulation and deterministic results for glucose and acetate. The differences are not relevant for discussion.

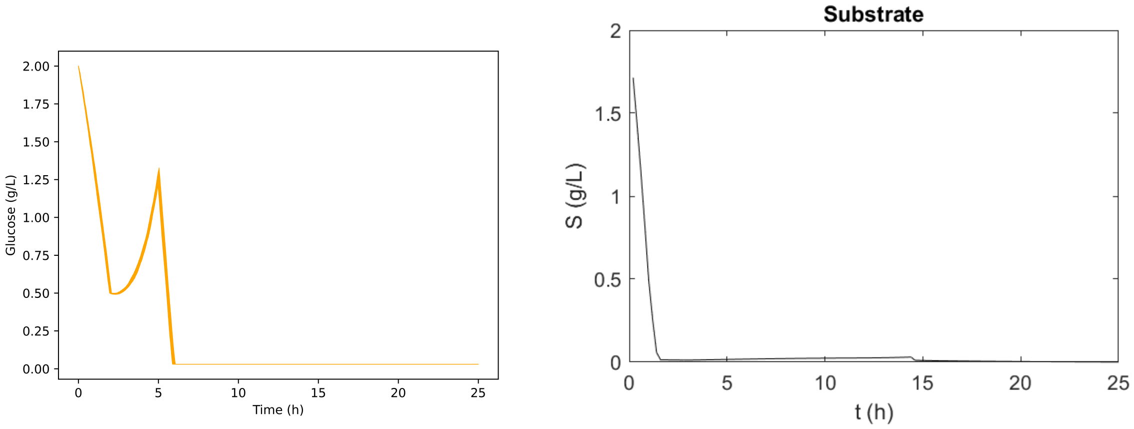

Figure B.5: Evolution of glucose overtime: On the left side, 100 realisations of the developed stochastic simulation model; on the right side, results of the deterministic model [35].

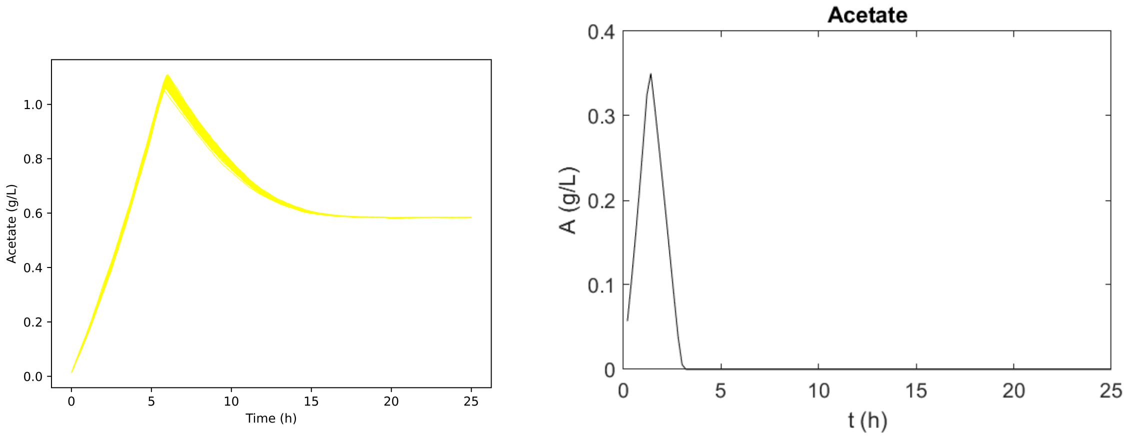

Figure B.6: Evolution of acetate overtime: On the left side, 100 realisations of the developed stochastic simulation model; on the right side, results of the deterministic model [35].

Figure B.7 illustrates the comparison between the stochastic simulation and deterministic results for dissolved oxygen. The observed differences can be explained by what is suggested in Chapter 4 for Model 4.2.

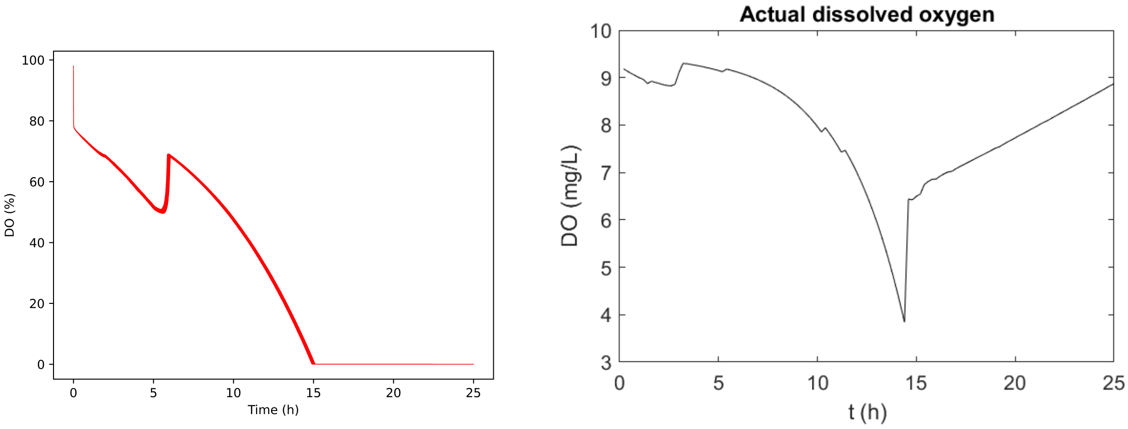
Actual dissolved oxygen

Figure B.7: Evolution of dissolved oxygen overtime: On the left side, 100 realisations of the developed stochastic simulation model; on the right side, results of the deterministic model [35].