



UNIVERSITÀ  
DEGLI STUDI  
DI PADOVA



**DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE**

**CORSO DI LAUREA IN BIOINGEGNERIA**

**“Model predictive control for microalgae growth in a photobioreactor”**

**Relatore: Prof. Simone Del Favero**

**Laureando: Alessandro Bernardi**

**ANNO ACCADEMICO 2021 – 2022**

**Data di laurea 14 aprile 2022**



Ai miei genitori Mauro e Susanna  
Ai miei fratelli Riccardo e Francesco  
Alla mia compagna Giorgia

## Sommario

L'ottimizzazione della crescita nelle colture di microalghe è una missione rilevante per garantire una maggiore efficienza nei processi industriali, che si traduce nella riduzione dei costi e nella possibilità di aumentare il margine di guadagno sul prodotto finale. Negli impianti industriali spesso viene associato un gemello digitale ovvero una riproduzione virtuale dell'impianto, che permette di fare analisi e previsioni del sistema anche in tempo reale. Una possibile soluzione per ottimizzare la crescita di microalghe è quella di definire una loro rigorosa rappresentazione virtuale, che tenga conto dei fenomeni che influiscono sul loro sviluppo e che sia in grado di fornire le informazioni necessarie per gestire un corretto controllo delle varie variabili. Questa tesi vuole mostrare alcuni modelli matematici che descrivono la crescita di microalghe nei fotobioreattori e, tramite un simulatore, vuole fornire un metodo per il controllo della concentrazione di biomassa. Il primo capitolo è una introduzione sulla definizione e classificazione delle microalghe e del funzionamento del fotobioreattore. Nel secondo capitolo, viene fornita la descrizione matematica del funzionamento di un bioreattore e le formule matematiche che evidenziano la dinamica della crescita delle alghe. Il terzo capitolo serve come raccolta delle informazioni necessarie per l'implementazione in ambiente virtuale del modello. Il quarto capitolo fornisce due diverse strategie di controllo, PID e MPC, e mostra come questi siano stati regolati. Nel quinto capitolo si discute dei vantaggi e svantaggi dei due controllori e si paragonano tra loro le prestazioni, mentre il sesto capitolo lascia spazio alle conclusioni finali.

## **Abstract**

The optimization of growth in microalgae cultures is a relevant mission to ensure greater efficiency in industrial processes, which means reducing costs and increasing the profit margin on the final product. Often, in industrial plant, a digital twin is used, that is a virtual reproduction of the plant, which allows to make analysis and predictions of the system also in real time. A possible solution to optimize the growth of microalgae is to define a rigorous virtual representation, which takes into account the phenomena that affect their development and is able to provide the necessary information to manage a proper control of the various variables. This thesis wants to show some mathematical models that describe the growth of microalgae in photobioreactors and, through a simulator, wants to provide a method to control the biomass concentration. The first chapter is an introduction on the definition and classification of microalgae and the operation of the photobioreactor. In the second chapter, the mathematical description of the functioning of a bioreactor is provided as well as mathematical formulas that highlight the dynamics of algae growth. The third chapter collects the information needed for the virtual environment implementation of the model. The fourth chapter provides two different control strategies, PID and MPC, and shows how these were tuned. The fifth chapter reviews the advantages and disadvantages of the two controllers and compares their performance with each other, while the sixth chapter shows the final conclusions.



# Contents

<b>1</b>	<b>Introduction</b>	<b>9</b>
1.1	Microalgae . . . . .	9
1.2	Photobioreactor . . . . .	12
1.3	Aim of the thesis . . . . .	13
<b>2</b>	<b>Mathematical models for microalgae growth</b>	<b>15</b>
2.1	Basic bioreactor model . . . . .	15
2.1.1	Dilution rate . . . . .	16
2.1.2	Conversion function . . . . .	17
2.2	Single microorganism and single input flow . . . . .	17
2.2.1	Specific growth rate . . . . .	19
2.3	Droop model . . . . .	22
2.3.1	Metabolic rates in Droop model . . . . .	23
2.4	Lipid accounting model . . . . .	24
2.4.1	Metabolic rates in lipid model . . . . .	25
<b>3</b>	<b>Computational model implementation</b>	<b>27</b>
3.1	Model implementation . . . . .	27
3.1.1	Model info . . . . .	27
3.1.2	Model parameters . . . . .	28
3.1.3	Constraints . . . . .	29
3.1.4	Equilibrium point . . . . .	29
3.1.5	S-function . . . . .	30
3.2	Linearization . . . . .	31
3.3	Discretization . . . . .	33
<b>4</b>	<b>Controllers</b>	<b>35</b>
4.1	PID control . . . . .	37
4.1.1	Gain tuning . . . . .	37
4.2	Model predictive control . . . . .	39

4.2.1	Cost function . . . . .	39
4.2.2	Input constraints . . . . .	40
4.2.3	Perturbation domain . . . . .	40
4.3	Kalman filtering . . . . .	41
<b>5</b>	<b>Results and discussion</b>	<b>43</b>
5.1	PID performance . . . . .	43
5.2	MPC performance . . . . .	44
5.3	State estimation . . . . .	46
5.4	Different set-ups . . . . .	46
5.4.1	Change of input constraints and output tracking weight $Q$ . . . . .	46
5.4.2	Change of reference trajectory . . . . .	47
5.4.3	Change of sampling time . . . . .	48
<b>6</b>	<b>Conclusions</b>	<b>49</b>
<b>A</b>	<b>Simulink schemes</b>	<b>51</b>
<b>B</b>	<b>Code listing</b>	<b>55</b>



# List of Figures

1.1	Pigment absorption spectrum. Extracted from [1]	10
1.2	Flow chart showing applications of microalgae in different fields. Extracted from [2]	11
1.3	Three different example designs of open pond systems. Extracted from [3]	12
1.4	Laboratory stirred tank photobioreactors. Extracted from [4]	13
2.1	Schematic representation of bioreactors	19
2.2	Specific growth rate as function of substrate concentration $S$ ( $K_s = 0.1$ , $K_i = 0.4$ )	20
2.3	A schematic of photosynthesis rate curve on light limited, saturated and inhibited regions. Extracted from [5]	21
2.4	Specific growth rate as function of Temperature T, CTMI model.	22
3.1	Output from the true system and the linearized model	32
4.1	Trajectory of reference	35
4.2	Light intensity with disturbance and measured disturbance over 1 day	36
4.3	Effect of different gains $K_p$ on manipulated variable and on output	39
5.1	Input and output response from simulation with PID	44
5.2	Contribution of proportional, integral, derivative parts	44
5.3	Input and output response from simulation with MPC	45
5.4	States of plant and their estimate through Kalman filter	46
5.5	Input and output response whit different constraints and weight	47
5.6	Input and output response whit different reference trajectory	47
5.7	Input and output response whit different sampling time	48
A.1	Plant via s-function block	51
A.2	PID block	51
A.3	Simulation with PID controller	52
A.4	Simulation with MPC controller	53



# List of Tables

2.1	Dynamic equations of the three types of bioreactor . . . . .	18
3.1	Droop model parameters . . . . .	28
3.2	Operating point . . . . .	30
4.1	Proportional gain and relative cost function values . . . . .	38



# Chapter 1

## Introduction

In this chapter it is offered an overview of the background knowledge needed to understand the sector in which this thesis is located. In Section 1.1 is described what microalgae are and how they are used in industrial processes. In Section 1.2 is presented the tools used nowadays for the growth of microalgae in photo-bioreactors. In Section 1.3 is explained what is the purpose of this thesis.

### 1.1 Microalgae

Microalgae are microscopic unicellular organisms that live in aquatic environments. They have a simple reproductive structures and they do not have stems, roots or leaves [6]. Microalgae are considered photoautotrophic organisms, that is they are able to convert light and CO<sub>2</sub> into organic compounds, however, many microalgae are also heterotrophic, since they are able to take nutrients from organic molecules [7]. Microalgae that can undertake photosynthesis are essential for life on Earth as they produce around half of the oxygen in the atmosphere [8] while also reducing CO<sub>2</sub> in the atmosphere through carbon fixation. The peculiarity of microalgae is that are capable of doing photosynthesis with high photosynthetic yields compared to terrestrial plants and they can be grown quickly in non-arable lands [9]. They boast a huge biodiversity with hundreds of thousands of estimated species and thousands species described [10] and that is why they are very versatile and can grow in a wide variety of environments, from oceans, lake, rivers, to ponds and wastewater [11] and why are used in many fields.

#### Classification

Microalgae come in different sizes, shapes and pigments so it is a complex task to classify them under an unique criteria. One great distinction is between eukariotyc and prokaryotic [12], but algae can also be classified based on their pigments [13]. The main six

groups are:

- green algae (*Chlorophyceae*)
- blue-green algae, also called cyanobacteria (*Cyanophyceae*)
- red algae (*Rhodophyceae*)
- brown algae (*Phaeophyceae*)
- golden brown algae (*Chrysophyceae*)
- diatoms (*Bacillariophyceae*)

It should be stressed that each different specie needs a different type of light, according mainly on its pigment content [14]. Some of the most important pigments are:

- *Chlorophyll-a* (in all plants, algae, and bacteria);
- *Chlorophyll-b* (in chlorophytes and land plants);
- *Carotenoids* (in all plants and algae);
- *Phycoerythrin* (in red algae)
- *Phycocyanin* (in cyanobacteria)

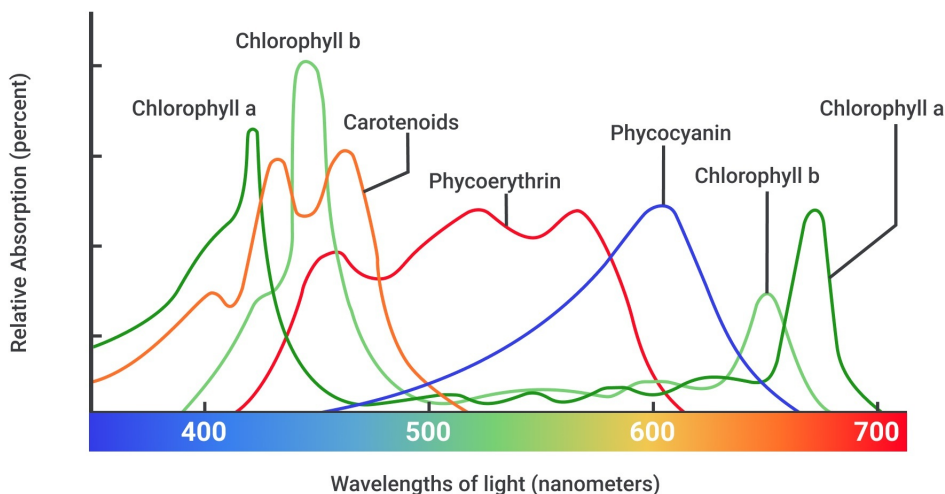


Figure 1.1: Pigment absorption spectrum. Extracted from [1]

Microalgae use the pigment chlorophyll-a to carry out the reaction of photosynthesis [15]. This pigment absorbs very well the light at the blue and red frequencies, but some species of microalgae can absorb light at others frequencies depending on the species.

To maximize photosynthetic efficiency, the light source wavelength should match the photosynthetically active spectrum, for example Chlorophyll b does not absorb green light, so it should be avoided to irradiate 550nm wavelength light organisms where this pigment is present.

## Application

Applications arise from the exploitation of properties of microalgae, like their ability of fixate CO<sub>2</sub> and reduce greenhouse gas in the atmosphere, or derive from a desire to extract a product. The final product of microalgae harvest may be exactly its own biomass or its lipids that, for example, through transesterification can lead to biofuel [16], or may be biochemicals produced directly by microalgae like  $\beta$ -carotene [17]. The wide biodiversity of microalgae make them suitable for many fields, like commercial, industrial and environmental [2].

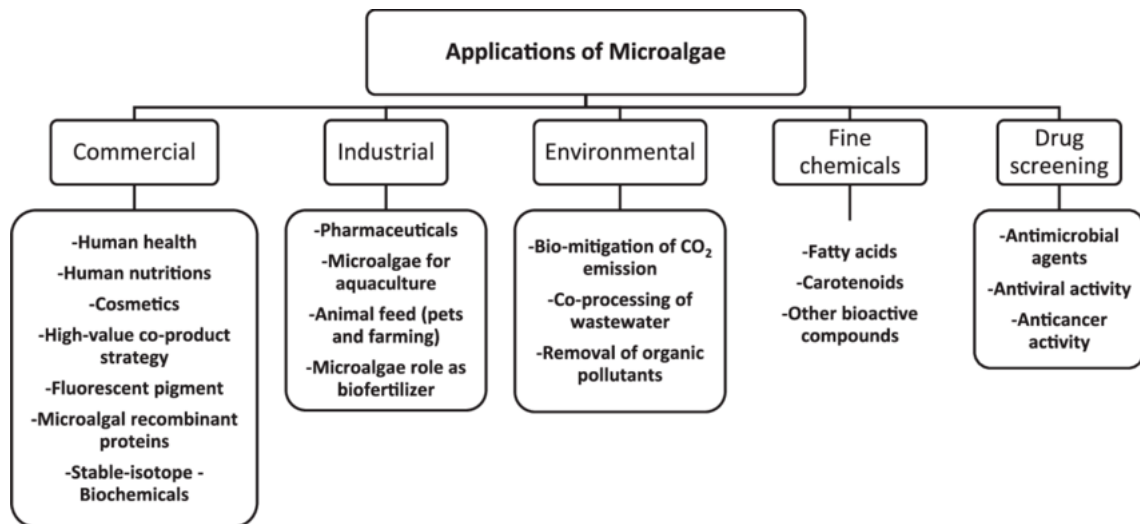


Figure 1.2: Flow chart showing applications of microalgae in different fields. Extracted from [2]

## Algae cultivation systems

There are two main methods to cultivate microalgae [18]:

- *Open* systems like open ponds, circular ponds, raceway ponds
- *Closed* systems like bioreactor and photobioreactor (PBR)

The first method is the most natural way to grow algae. These are placed in oval or circular tanks outdoors so that they receive light and carbon dioxide and are fed and left to grow. The tanks are designed in a raceway configuration and are equipped with

paddle wheels that provide circulation and mixing of cells, nutrients and dissolved gases [19]. The main advantages over the closed systems are: simple and cheap design, relative easy maintenance, cost efficient operation. However, there are significant drawbacks that limit this technology from being superior in some applications: high contamination risk, control of temperature, pH and nutrients can be more difficult given that the system is more exposed to environmental changes.

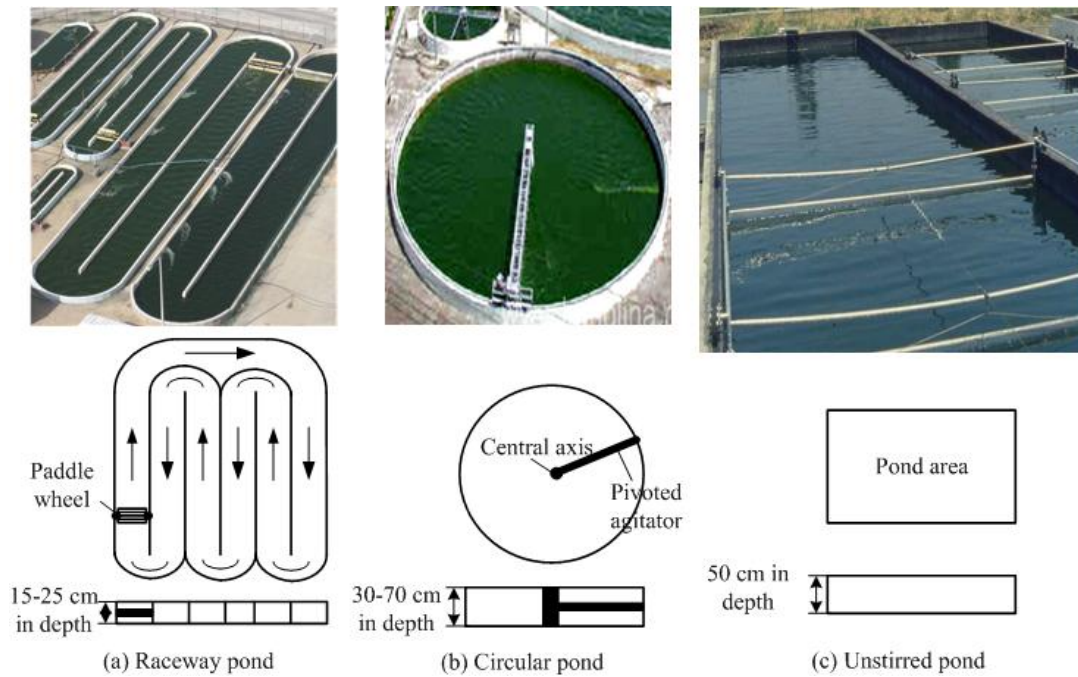


Figure 1.3: Three different example designs of open pond systems. Extracted from [3]

Closed systems are typically referred to as photobioreactors. Its description and operating principles are discussed in the next section.

## 1.2 Photobioreactor

A photobioreactor is a particular bioreactor that uses light as source of nutrient for phototropic organisms. Hence, it is an artificial environment where specific conditions are carefully controlled with the aim to obtain better growth rates of biomass or better purity levels compared to culture in the corresponding natural habitats [4]. In addition, growth cultures can be continuously supplied by nutrients, and by gas like oxygen and carbon dioxide and compensate the limited sun lighting with artificial lightning. It is important to control all these variables since it has been shown that in microalgae cultures, an optimal feeding strategy can lead to optimal growth or even a lipid induction, under environmental stresses [20]. Photobioreactors can be placed indoors or outdoors, but often these systems are placed inside greenhouses to reduce the cost of lighting. The



major advantage of using these systems is the increased surface area for a certain culture volume. Other advantages over open ponds are: lower contamination risk, higher cell concentration, easier mixing which improves mass transfer, easier to control temperature, pH and nutrient supply [18]. There are many types of bioreactors with different shapes: vertical tubular, flat panel, horizontal tubular and stirred tank. In particular, a stirred tank photobioreactor for laboratory use is a cylindrical vessel of a few liters equipped with agitators that can spread the compounds and make the concentrations homogeneous. The walls are generally made of glass, while the head plate is made of stainless steel metal, with various openings that allow the insertion of probes, media and gases, and the sampling of the solution [21]. In addition, the bioreactor is equipped with various systems for heat exchange, it may have an external coating or coil through which steam or cooling water is pumped or it may have helical coils located internally. Another method is to pump liquid from the reactor through a separate heat exchange [22]. Artificial illumination of the culture can be provided from the outside by LED lights and panels but there are also systems that carry light inside the tank through optical fibers [23].



Figure 1.4: Laboratory stirred tank photobioreactors. Extracted from [4]

### 1.3 Aim of the thesis

Photosynthetic yield from light energy to biomass is very low in industrial processes respect to theoretical limit. One of the reasons is the lack of a digital twin in automatic control, based on a mathematical description that gives a quantitative depiction of microalgae growth. A digital twin is a virtual representation of a real objects that can

be used to view the status of the actual physical object for monitoring, diagnostics and prognostics to optimize asset performance and utilization. In this case, be able to assess how much biomass and lipids can grow and how effectively light is utilized, may bring an optimization of the entire process and the possibility to scale the production from lab tank of few liters to industrial containers of thousands of liters. The objective of this thesis is to select the models available in the state of the art, implement a simulation of the microalgae growth through the Matlab software, define the control problem, build a model predictive controller (MPC) in the simulation environment, estimate the state of the system through a Kalman filter, and compare the performance to a standard PID controller.

# Chapter 2

## Mathematical models for microalgae growth

Modeling is a tool to understand, describe and analyze real world phenomena, and a model can be define as an approximate representation of reality. Models can be used to measure what is not accessible, to simulate and control systems and to predict their behaviour. Mathematical models are crucial in the formulation of the control problem as we use them to understand the effect of inputs to the system. Moreover, the capability of the MPC to predict the evolution of the system in the future is totally dependent by the model. Hence, the first main step of the project is to formulate a mathematical description of how a photobioreactor works, and how microalgae grow. In Section 2.1 it is described the basic bioreactor dynamics, in Section 2.2 it is analyzed a single organism and single feed rate bioreactor and its typology, in Section 2.3 it is shown a particular model of algae growth and in Section 2.4 a multi-input feed model that takes into account lipid production.

### 2.1 Basic bioreactor model

A bioreactor is a tank that contains a mixture of substrates, microorganisms and their products. Liquids can be injected and expelled in the tank, modifying the media composition and volume. To understand what changes inside the tank, we resort on mass balance equations [24], [25] that give an explanation on how mass is transformed and how volume changes over time. Hence, for a generic substance with concentration  $C$  [g/L] in a volume  $V$  [L], the accumulation of mass inside the tank is given by three distinct factors:

- Influent volume, that brings new mass through the incoming fluid with concentration  $C_{in}$

- Effluent volume, that takes mass from the reactor volume with concentration  $C^1$
- Conversion, that means reactions of internal processes which can increment or decrement the total concentration  $C$  inside the reactor

The mass balance equation is the following:

$$\frac{d(V(t)C(t))}{dt} = r_c(t)V(t) + F_{in}(t)C_{in}(t) - F_{out}(t)C(t) \quad (2.1)$$

Where  $r_c$  is the conversion function,  $F_{in} [L/d]$  (d=day) is the entering volume flow and  $F_{out} [L/d]$  is the exiting volume flow. Omitting time dependency and using the derivative of a product we obtain:

$$\dot{C}V + \dot{V}C = r_cV + F_{in}C_{in} - F_{out}C \quad (2.2)$$

And then, isolating the concentration change  $\dot{C}$ :

$$\dot{C} = r_c + \frac{F_{in}C_{in}}{V} - \frac{F_{out}C}{V} - \frac{\dot{V}C}{V} \quad (2.3)$$

Volume change over time is regulated by flows entering and exiting<sup>2</sup> in the tank:

$$\dot{V} = \sum_{i=1}^n F_{in,i} - F_{out} \quad (2.4)$$

Where  $i$  goes from one to  $n$  total influent flows.

### 2.1.1 Dilution rate

The dilution rate is a very important factor when working with a bioreactor and it is usually used as input for the system. It is defined as follow:

$$D(t) = \sum_{i=1}^n \frac{F_{in,i}(t)}{V(t)} \left[ \frac{1}{d} \right] \quad (2.5)$$

where  $d$  stands for days. From eq.(2.3) and (2.4) the concentration and volume derivatives become:

---

<sup>1</sup>Mass in the media is considered well mixed, so the density is homogeneous

<sup>2</sup>it is considered a single media output flow

$$\begin{aligned}
\dot{C} &= r_c + \frac{F_{in}C_{in}}{V} - \frac{F_{out}C}{V} - \frac{(\sum_{i=1}^n F_{in,i} - F_{out})C}{V} \\
&= r_c + \frac{F_{in}C_{in}}{V} - DC
\end{aligned} \tag{2.6}$$

$$\dot{V} = F_{in} - F_{out} = DV - F_{out}$$

Eqs.(2.6) are the standard mathematical description of a bioreactor.

### 2.1.2 Conversion function

The conversion function is used to describe how mass of the different concentrations inside the tank evolve. For example, the biomass is able to grow at a certain rate and is susceptible to death at a different rate, but also the substrate is consumed with some yield by the biomass. Taking this in consideration we can write:

- for biomass X

$$r_X(t) = \mu X(t) - k_d X(t) \tag{2.7}$$

- for substrate S

$$r_S(t) = -\frac{1}{Y_{XS}}\mu X(t) + k_m X(t) \tag{2.8}$$

Where  $X$  [g/L] is the biomass concentration,  $\mu$  [1/d] is the specific growth rate,  $S$  [g/L] is the substrate concentration,  $k_d$  [1/d] is the death rate,  $k_m$  [1/d]<sup>3</sup> is maintenance term for biomass survival,  $Y_{XS}$  is the yield coefficient that relates the consumption of substrate to the biomass growth.

## 2.2 Single microorganism and single input flow

Let's focus on a bioreactor with only one type of microorganism and only one source of influent and one source of outlet flow, assuming that no mass of microalgae are introduced inside the tank and the tank is well mixed. From eq.(2.6), (2.7) and (2.8) it is possible to write:

$$\begin{cases}
\dot{X} = \mu X - k_d X - DX \\
\dot{S} = -\frac{1}{Y_{XS}}\mu X + k_m X + DS_{in} - DS \\
\dot{V} = F_{in} - F_{out}
\end{cases} \tag{2.9}$$

---

<sup>3</sup>generally, death rate  $k_d$  and maintenance term  $k_m$  are neglected

From this configuration, acting on the inlet and outlet flow, it is possible to distinguish three main types of bioreactors:

- *Batch* reactor
- *Fed-Batch* reactor
- *Continuous stationary tank* reactor (CSTR)

A batch reactor is a cultivation method where no flow is inserted and no flow comes out. At the start of the process, a given amount of biomass and substrate are placed, the biomass will consume the nutrients and start to grow while the total volume remains constant<sup>4</sup>. The harvest is done when all the substrate is consumed. This setup can be useful to observe how microalgae grow from which one can derive the estimation of the model parameters. In a Fed-batch set up, only substrate valve is opened while the output flow is always shut down. So the initial conditions start with little volume that begins to increase over time and biomass that keeps getting fed by inflow. At the end of the process all the content is withdrawn and the tank is cleaned and sterilized for the next operation. The last type is a bioreactor that works in continuous mode, which means that an input and an output flows are always present and equal in such a way that the total volume stays constant. This system works on steady state conditions where medium is fed continuously and the product is harvested continuously.

Batch	Fed-Batch	CSTR
$F_{in} = F_{out} = 0$	$F_{out} = 0$	$F_{in} = F_{out} = F$
$\dot{X} = \mu X$	$\dot{X} = (\mu - D)X$	$\dot{X} = (\mu - D)X$
$\dot{S} = -\frac{1}{Y_{XS}}\mu X$	$\dot{S} = -\frac{1}{Y_{XS}}\mu X + D(S_{in} - S)$	$\dot{S} = -\frac{1}{Y_{XS}}\mu X + D(S_{in} - S)$
$\dot{V} = 0$	$\dot{V} = F_{in} = DV$	$\dot{V} = F_{in} - F_{out} = 0$

Table 2.1: Dynamic equations of the three types of bioreactor

Note: the total amount of volume in a fed-batch can be computed at any time  $t$  by  $V(t) = V(t_0) + \int_{t_0}^t F_{in} dt$  where  $t_0$  is the initial time of the experiment.

<sup>4</sup>Small fluid adjustment can be done for pH and against foam

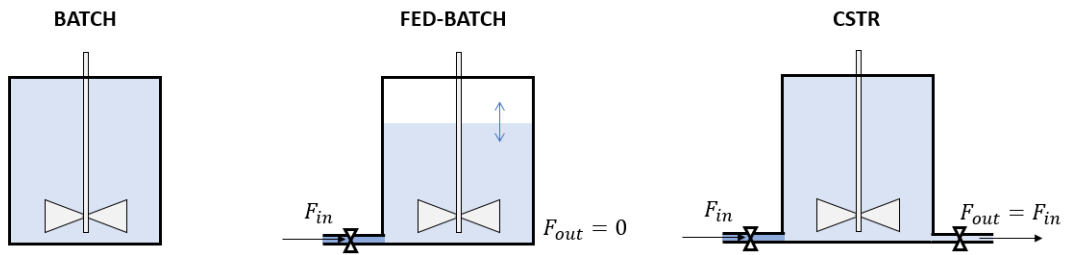


Figure 2.1: Schematic representation of bioreactors

### 2.2.1 Specific growth rate

In this subsection we describe more in details what is the specific growth rate. This is a function of different parameters that tells how fast microalgae can grow. Microalgae are affected by external environment like how much nutrients are accessible, if too much product is toxic, what is the outside temperature where they live, how acid is the medium, how much light is available for photosynthesis and so on [24]. The specific growth rate can be expressed as a function of products for each influencing factor:

$$\mu = \mu(X)\mu(S)\mu(P)\mu(pH)\mu(T)\mu(I)...$$

The aim of the thesis is not to give a wide overview of the formulations on the specific growth rate, so we show only the ones of major interest.

#### Influence of substrate **S** on $\mu$

Microalgae require several nutrients to grow, and when deprived of these nutrients, their growth can be severely reduced. Cells development are influenced by nitrogen and phosphorus molecules [26] and also by many micronutrients, such as iron, manganese, molybdenum, and nickel [27]. Many models have been tried to describe how nutrients affect microalgae development. Monod developed one of the first growth models [28], which was originally designed for bacterial cultures. To characterize the growth rate as a function of nutrient concentration, he used the Michaelis-Menten equation. Monod (Michaelis-Menten) equation:

$$\mu(S) = \mu_{max} \frac{S}{(K_s + S)} \quad (2.10)$$

Where  $\mu_{max}$  is maximum specific growth rate,  $S$  is concentration of the limiting nutrient, and  $K_s$  is concentration of limiting nutrient at which the specific growth rate is half of its maximum value.

Monod's equation, however, fails to explain the inhibitory effect that a high concentration of nutrients can lead to. The Haldane-Andrew equation takes into account this effect by adding the inhibitory factor into the equation [29]: Haldane equation<sup>5</sup>:

$$\mu(S) = \mu_{max} \frac{S}{(K_s + S + \frac{S^2}{K_i})} \quad (2.11)$$

where  $K_i$  is inhibition constant and  $\mu_{max}$  is a theoretical value representing maximum specific growth rate if there was no inhibition. The next figure shows the plot of the Monod and Haldane functions.

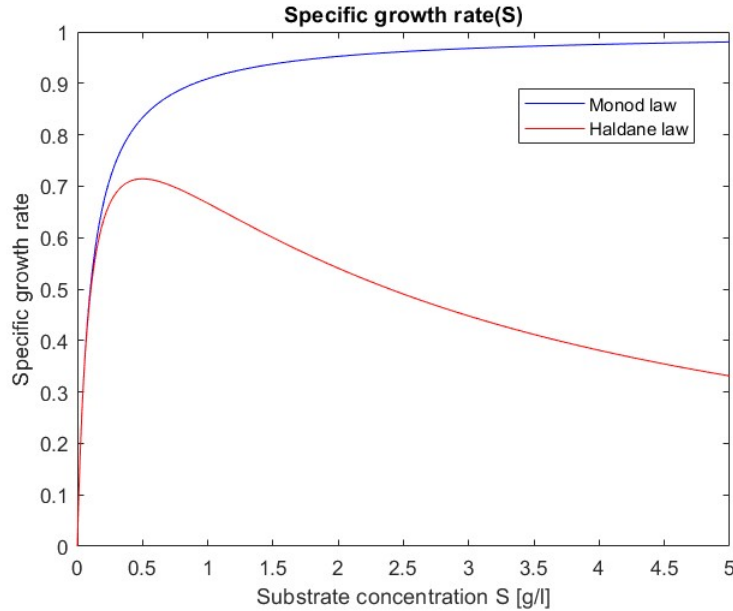


Figure 2.2: Specific growth rate as function of substrate concentration  $S$  ( $K_s = 0.1$ ,  $K_i = 0.4$ )

### Influence of light intensity $I$ on $\mu$

Light is the most crucial parameter in the modeling of microalgal growth because algae need light as a source of energy for photosynthesis. Based on the intensity of the light, microalgal cells are located in three different light zones: photolimited, photosaturated, and photoinhibited [5] as it can be see in Figure 2.3.

<sup>5</sup>The maximum of this function is when  $S = \sqrt{K_s K_i}$



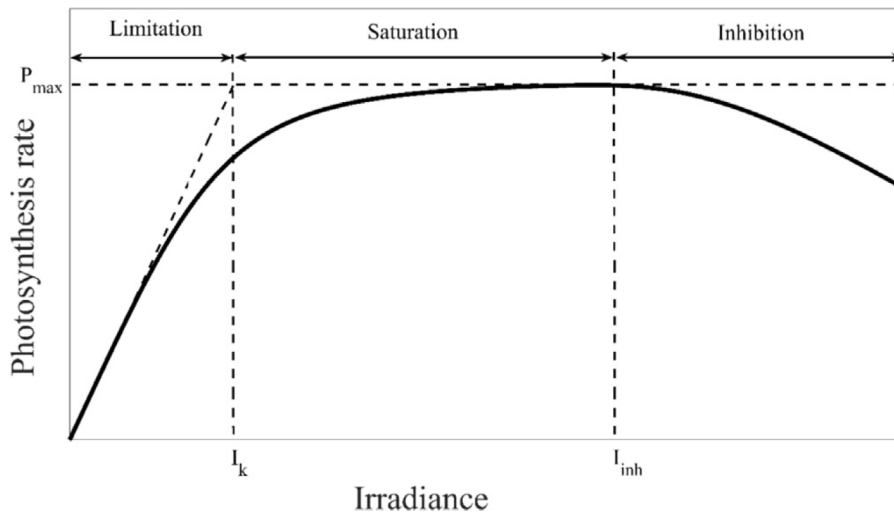


Figure 2.3: A schematic of photosynthesis rate curve on light limited, saturated and inhibited regions. Extracted from [5]

As light goes deeper, more and more light is attenuated and microalgae deeper in the media receive less light causing limitation on the growth. Besides light penetration, other phenomena should be taken in consideration, like the light exposure of algae in hydrodynamic moving environment where the cells move and can come closer to or farther away from a light source, or how light spectrum and different length waves can undermine growth. We simplify this behaviour, making the assumption that light affect growth as saturation like Monod equation:  $\mu(I) = \frac{I}{K_{sI}+I}$  or inhibition like Haldane equation:  $\mu(I) = \frac{I}{(K_{sI}+I+\frac{I^2}{K_{iI}})}$ .

### Influence of temperature $T$ on $\mu$

Temperature, after nutrition and light, is the most important factor in microalgae development. In fact, microalgae are not able to regulate their temperature, and every microalgae species has an optimal temperature, which is generally between 20 °C and 24 °C [30]. They all appear to be following the same patterns like the one in Figure 2.4: a region where the function reaches its peak at the optimum temperature, and the regions outside where no growth occur. For simplicity sake, we describe here only two of many models. The Hinshelwood model [31], which was created using the Arrhenius rule for bacteria growth and later utilized for microalgae development:

$$\mu(T) = k_1 e^{-\frac{E_1}{T}} - k_2 e^{\frac{E_2}{T}} \quad (2.12)$$

where  $T$  is absolute temperature and  $k_1$ ,  $k_2$ ,  $E_1$ , and  $E_2$  are model parameters.  $E_1$  and  $E_2$  replaced the activation energy over gas constant in the original form of Arrhenius law

and  $k_1$  and  $k_2$  are pre-exponential factors. Thermodynamically, in this equation, the first term can be interpreted as the production (growth) rate and the second term as the consumption (death) rate of the bacteria due to high temperature. However, as the temperature is decreased in this model, specific growth rate declines but never reaches zero except at temperature of absolute zero. This makes the model unrealistic at low temperatures. An alternative model proposed is the cardinal temperature model with inflection (CTMI) [32]. This model uses three cardinal temperatures ( $T_{min}$ ,  $T_{opt}$ ,  $T_{max}$ ). The growth takes place only when the temperature is between the maximum and the minimum<sup>6</sup>.

$$\mu(T) = \mu_{max} \frac{(T - T_{max})(T - T_{min})^2}{(T_{opt} - T_{min})[(T_{opt} - T_{min})(T - T_{opt}) - (T_{opt} - T_{max})(T_{opt} + T_{min} - 2T)]} \quad (2.13)$$

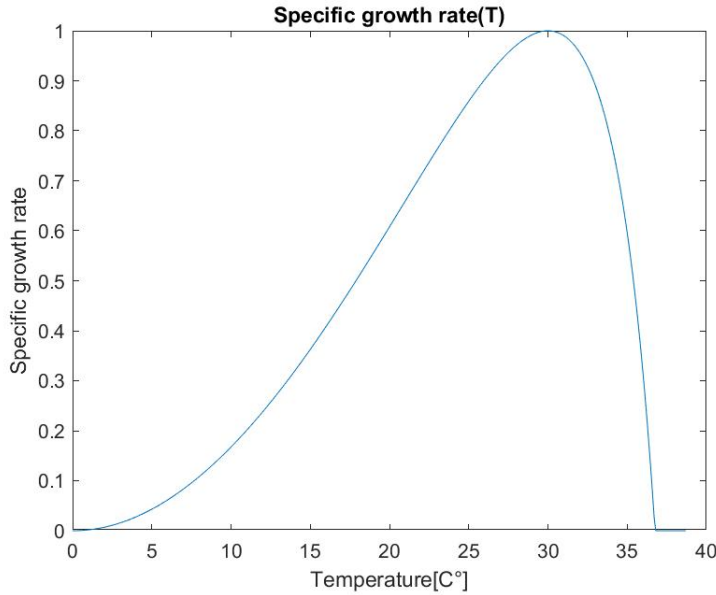


Figure 2.4: Specific growth rate as function of Temperature T, CTMI model.

### 2.3 Droop model

Differential equations like eq.(2.9) are not sufficient to describe very complex biosystems like algae growing in a bioreactor. In particular, in contrast to bacteria, it has been seen that some species of microalgae are able to uptake nitrogen and use their internal nitrogen pool as source of feed in the case of lacking nutrients. This strong uncoupling between

<sup>6</sup>under the condition:  $T_{opt} > \frac{T_{min} + T_{max}}{2}$

nutrient uptake and growth was highlighted by Droop [33]. His model represented growth dependent on an internal quota  $Q$  defined as the quantity of nutrient per biomass unit, instead of dependent on substrate concentration  $S$ . Hence, the substrate is consumed to increase directly and only the cell internal quota, which is subsequently utilized to increase the biomass. The resulting equations are:

$$\begin{cases} \dot{X} = \mu(Q)X - XD \\ \dot{Q} = \rho(S) - \mu(Q)Q \\ \dot{S} = -\rho(S)X + D(S_{in} - S) \end{cases} \quad (2.14)$$

Where  $\rho(S)$  is the uptake rate described in the next section.

### 2.3.1 Metabolic rates in Droop model

With metabolic rates are intended the functions that contribute on the development of the cell, like growth, uptake of nutrients etc. and they are part of the *conversion* function defined in Section 2.1.2.

#### Specific growth rate

As seen in Section 2.2.1 the specific growth rate is determined by the products of more factors. In the original Droop model, it only depends by the internal quota  $Q$ . In fact, it makes sense that the nitrogen pool has effect on the growth of the algae, which can grow by drawing nutrients directly form their feed stock.

$$\mu(Q) = \mu_{max}\mu(Q) = \mu_{max} \left(1 - \frac{q_0}{Q}\right) \quad (2.15)$$

where  $\mu_{max}$  is the hypothetical growth rate obtained for an infinite quota and  $q_0$  is the minimum internal quota allowing growth.

However, Droop model does not take into account the effect of light on the growth. In our implementation, light effect has been modeled with an Haldane inhibition. So it is added the multiplicative factor of light influence and the full specific growth rate becomes:

$$\mu(Q, I) = \mu_{max}\mu(Q)\mu(I) = \mu_{max} \left(1 - \frac{q_0}{Q}\right) \left(\frac{I}{K_{sI} + I + \frac{I^2}{K_{iI}}}\right) \quad (2.16)$$

#### Uptake rate

The uptake rate is modeled with a saturation Monod curve, with substrate as the saturating factor. The intuition is that if more nutrients are present, the microalgae can

uptake them until a certain point where more nutrients can not be absorbed more. Other models have been propose, like Hill allosteric regulation coefficient or model accounting for maximum storage of the cell, but Akaike and Bayesian information criteria are able to show little difference between the models [34], so in this case, the simple one is always preferred.

$$\rho(S) = \rho_{max} \frac{S}{(K_S + S)} \quad (2.17)$$

The parameters  $\rho_{max}$  and  $K_S$  represent respectively the maximum uptake rate and the half-saturation constant for the substrate.

## 2.4 Lipid accounting model

Microalgae are organisms that can store lipids inside the cell in condition of nutrients deficiency and environmental stresses. These lipids are a valuable product of the harvest because it is obtained biodiesel after refinery. So when it comes to maximize productivity, the aim is to harvest as much as biomass possible while you try to achieve high amount of lipids. Thus, another model is presented from the paper of De la Hoz Siegler et al, and Yoo et al [35], [36] derived from mass balance equations, so the mathematical description is the same as eq.(2.6) and eq.(2.9) except there are two input feeds (one for glycine and one for glucose) and three compartments: the active biomass, the nitrogen quota and the lipid quota.

$$\left\{ \begin{array}{l} \dot{X} = \mu(Q, S_2)X - XD \\ \dot{Q} = \rho(S_1)X - \frac{1}{Y_{XQ}}\mu(Q, S_2)X - QD \\ \dot{L} = \pi(L, S_2)X - \frac{1}{Y_{XL}}\mu(Q, S_2)X - LD \\ \dot{S}_1 = -\rho(S_1)X + S_{1,in}\frac{F_{1,in}}{V} - S_1D \\ \dot{S}_2 = -\frac{1}{Y_{XS}}\mu(Q, S_2)X - k_m X + S_{2,in}\frac{F_{2,in}}{V} - S_2D \\ \dot{V} = VD - F_{out} = F_{1,in} + F_{2,in} - F_{out} \end{array} \right. \quad (2.18)$$

This model consists of six states  $[X, Q, L, S_1, S_2, V]'$  and summarizes these features: cells are composed of three main compartments: metabolically active biomass  $X$  [g/L], lipid body  $L$  [g/L], nitrogen pool  $Q$  [g/L]; nitrogen source  $S_1$  [g/L] (glycine) is inserted through the nitrogen feed flow rate  $F_{1,in}$  [mL/h] and is taken up into the nitrogen pool and later converted, at a constant yield  $Y_{XQ}$ , into active biomass; carbon source is inserted through

the carbon feed flow rate  $F_{2,in}$  [mL/h] and is taken up by the cells and directly converted either into active biomass or oil, at constant yields  $Y_{XS}$  or  $Y_{LS}$  respectively; oil stored in the lipid bodies can be used to support growth at a constant yield  $Y_{XL}$ ; glucose  $S_2$  [g/L] is used for maintenance of active biomass; specific growth rate depends only by glucose and nitrogen pool; uptake rate is function of glycine, and lipid production rate is function of  $L$  and glucose. It is useful to represent also the total biomass given by the sum of the three compartments:

$$Z = X + Q + L$$

### 2.4.1 Metabolic rates in lipid model

#### Specific growth rate

In this model, the growth is influenced by four distinct factor: intracellular nitrogen  $Q$ , lipid concentration  $L$ , glucose  $S_2$  (as limiting/saturating factor), light intensity  $I$

$$\mu = \mu_{max} \left(1 - \frac{q_0}{Q/Z}\right) \left(1 - \frac{l_0}{L/Z}\right) \left(\frac{S_2}{K_{S_2} + S_2}\right) \left(\frac{I}{K_I + I}\right) \quad (2.19)$$

where  $\mu_{max}$  [1/L] is the maximum growth rate  $q_0$  is the minimum nitrogen quota for supporting growth,  $l_0$  is the minimum lipid quota for supporting growth,  $K_{S_2}$  [g/L] is the half saturation constant of glucose for growth,  $K_I$  [ $\mu\text{mol}/\text{m}^2\text{s}$ ] is the half saturation constant of light for growth.

#### Uptake rate

As mention in the Droop model, the nitrogen available affect directly the uptake of glycine substrate. In this particular model, there are also bounds for a minimum and maximum internal quota outside which the uptake stops:

$$\rho = \rho_{max} \left(\frac{S_1}{S_1 + K_{S_1}}\right) \left(\frac{q_m - Q/Z}{q_m - q_0}\right) \quad (2.20)$$

where  $\rho_{max}$  [1/h] is the maximum uptake rate,  $q_m$  is the maximum quota of nitrogen above which uptake rate stops,  $K_{S_1}$  [g/L] is the half saturation constant of glycine for uptake.

#### Lipid production rate

The lipid production rate is assumed to be affected by the intracellular nitrogen conditions as lipid production increases in nitrogen-deficiency conditions and also assumed to be affected by extracellular carbon concentration.

$$\pi = \pi_{max} \left( \frac{S_2}{S_2 + K_\pi} \right) \left( 1 - \frac{Q}{Z} \right) \left( \frac{l_m - L/Z}{l_m} \right) \quad (2.21)$$

where  $\pi_{max}$  [1/h] is the maximum lipid production rate,  $K_\pi$  [g/L] is the half saturation constant for oil production,  $l_m$  is the maximum quota of lipid above which lipid production stops.

# Chapter 3

## Computational model implementation

The next step after defining the model, is to evaluate the process dynamics through a development environment that can run the computation of the differential equations and give us the simulation in time of the growth of the microalgae. Matlab software is the one most suitable software for this purpose, since it can be used to utilize functions, generate plots, and perform simulation through the integrated software Simulink.

### 3.1 Model implementation

In this section it is presented all the model information necessary for the transposition in the virtual environment, it is provided the model information, parameters, constraints and equilibrium points that need to be used for the simulation.

#### 3.1.1 Model info

This work is based on the Droop model introduced in Section 2.3 through a state-space representation, with a specific growth rate and uptake rate functions define in eqs.(2.16),(2.17). A state-space representation is a mathematical model of a physical system as set of input, output and state variables related by differential equations. State variables are variables whose values change over time based on the values they hold at any given time and the values of input variables imposed externally. The values of output variables are determined by the values of state variables. The advantage is that all the information of the past history of the system is condensed in the state space at the present time. A continuous stationary harvest mode is used so it is filled in as much volume as it

is removed and the total volume remains constant. From eq.(2.14):

$$\begin{cases} \dot{x} = f(x(t), u(t)) \\ y(t) = h(x(t), u(t)) \end{cases} \implies \begin{cases} \dot{X} = \mu(Q, I)X - XD \\ \dot{Q} = \rho(S) - \mu(Q, I)Q \\ \dot{S} = -\rho(S)X + D(S_{in} - S) \\ \dot{V} = DV - F_{out} = 0 \\ y = X \end{cases} \quad (3.1)$$

This system has three states  $[X, Q, S]'$  ( $V$  not included in the state because it does not change), the dilution rate  $D$  [ $1/d$ ] as the single **input** and the biomass concentration (biovolume)  $X$  [ $\mu m^3/L$ ] as the single measured **output** expressed in biovolume on liters.

### 3.1.2 Model parameters

The difficulty in implementing a mathematical model, developed theoretically, is that not all system parameters, appearing in the model equations, are known a priori. Experiments with the real-world system can be used to identify unknown parameter values. In theory, this may be accomplished by evaluating data measured at the system's input and output using parameter-identification methods. This is a simulation thesis and real time data and measurements are not accessible, so the parameter values are based on the paper of Benattia et. al [37] and are reported on the following table.

Parameter	Description	Value	Unit
$\mu_{max}$	Maximum specific growth rate	2	$1/d$
$q_0$	Minimum cell quota allowing growth	1.8	$\mu mol/\mu m^3$
$K_{sI}$	Light saturation constant	150	$\mu E m^{-2} s^{-1}$
$K_{iI}$	Light inhibition constant	2000	$\mu E m^{-2} s^{-1}$
$S_{in}$	Substrate concentration in inlet feed	100	$\mu mol/L$
$\rho_{max}$	Maximum uptake rate	9.3	$\mu mol/\mu m^3 d$
$K_S$	Half saturation constant of substrate	0.105	$\mu mol/L$

Table 3.1: Droop model parameters



### 3.1.3 Constraints

$X, S$  are mass concentrations and their value can not be below zero. Similarly, dilution rate is expressed as the ratio between input flow and volume, so it can not go below zero. Furthermore the concentration of nutrients inside the tank may not exceed the external concentration supply. Minimal and maximal internal quota are derived directly from the model [38]:

$$\begin{cases} X \geq 0 \\ q_0 \leq Q \leq \frac{\rho_{max}}{\mu_{max}\mu_I} + q_0 \\ 0 \leq S \leq S_{in} \\ D \geq 0 \end{cases} \quad (3.2)$$

where  $\mu_I = \mu(I)$

It should be noticed that these constraints are automatically enforced by the model, except for the input constraint. Therefore, the output of the controller is saturated limiting the input value between 0 (we can not put a negative volume flux) and the double of input reference.

### 3.1.4 Equilibrium point

The equilibrium trajectories are calculated for equilibrium at an operating point such that  $X_r = 25 \mu m^3/L$  [39]. The equilibrium point for this equilibrium value are calculated when the dynamics of the system do not change i.e when the derivatives are zero:

$$\begin{cases} D_r = \mu_{max}\mu_I(1 - \frac{q_0}{Q_r}) \\ Q_r = \frac{\mu_{max}\mu_I(S_{in} + K_S) + (\rho_{max} + \mu_{max}\mu_I q_0)X_r - \sqrt{\Delta}}{2\mu_{max}\mu_I X_r} \\ S_r = \frac{\mu(Q_r, I_r)K_S Q_r}{\rho_{max} - \mu(Q_r, I_r)Q_r} \\ I_r = I_{opt} \\ \Delta = aX_r^2 + bX_r + c \\ a = (\rho_{max} + \mu_{max}\mu_I q_0)^2 \\ b = 2\mu_{max}\mu_I[(K_S - S_{in})(\rho_{max} + \mu_{max}\mu_I q_0) - 2\mu_{max}\mu_I q_0 K_S] \\ c = \mu_{max}^2 \mu_I^2 (S_{in} + K_S)^2 \end{cases} \quad (3.3)$$

In table 3.2 are shown the values of the working point.

Reference	Description	Value	Unit
$D_r$	Dilution rate	0.711	$1/d$
$X_r$	Biomass concentration	25	$\mu m^3/L$
$Q_r$	Nitrogen quota	3.998	$\mu mol/\mu m^3$
$S_r$	Substrate concentration	0.046	$\mu mol/L$
$I_r$	Light intensity	547	$\mu E m^{-2} s^{-1}$

Table 3.2: Operating point

### 3.1.5 S-function

An S-function is a computer language description of a Simulink block written in MATLAB, C, C++, or Fortran. C, C++, and Fortran S-functions are compiled as MEX files using the mex utility. S-functions follow a general form and can accommodate continuous, discrete, and hybrid systems. By following a set of simple rules, one can implement an algorithm in an S-function and use the S-Function block to add it to a Simulink model. S-functions define how a block works during different parts of simulation, such as initialization, update, derivatives, outputs and termination. In every step of a simulation, a method is invoked by the simulation engine to fulfill a specific task. S-function basics require fundamental knowledge of mathematical relationships between the block inputs, states, and outputs. Basically, we use a s-function to simulate the “true” plant, calculating at every sample time what is the output given a certain input. The states update is defined by continuous derivatives, but the input and output are sampled, giving us access to the true system output only at the time we sample. In addition to the system input, the s-function wants additional settings all gathered in a matlab structure called *DroopData* which has model parameters, initial conditions, reference values, input constraints, sampling time. In this case, the s-function takes the dilution rate as input, and light disturbance as second input, while in output it gives the states at the next simulation time. However, only the biomass concentration is the measured output, so we exclude the direct knowledge of the substrate concentration and nitrogen quota through a gain that selects only the first state. Simulink blocks and scheme are shown in Appendix A.

## 3.2 Linearization

Linear models frequently arise as descriptions of small perturbations away from a nominal solution of the system. The urge to linearize a model comes because linear systems are easier to study and analyze, computation is also faster, and it is more practical to build a controller. Now let the control be perturbed from their nominal values to  $u(t) = u_r(t) + \delta u(t)$  and let the state trajectory accordingly be perturbed to  $x(t) = x_r(t) + \delta x(t)$  where the subscript stands for reference. Performing a Taylor series expansion to first-order terms, we can write the Droop model in a linear state space representation:

$$\begin{cases} \delta \dot{x} = A\delta x + B\delta u \\ \delta y = C\delta x + D\delta u \end{cases} \quad (3.4)$$

with

$$A = \begin{bmatrix} \frac{\partial \dot{X}}{\partial X} & \frac{\partial \dot{X}}{\partial Q} & \frac{\partial \dot{X}}{\partial S} \\ \frac{\partial \dot{Q}}{\partial X} & \frac{\partial \dot{Q}}{\partial Q} & \frac{\partial \dot{Q}}{\partial S} \\ \frac{\partial \dot{S}}{\partial X} & \frac{\partial \dot{S}}{\partial Q} & \frac{\partial \dot{S}}{\partial S} \end{bmatrix} \bigg|_{\substack{x = x_r \\ u = u_r}} \quad B = \begin{bmatrix} \frac{\partial \dot{X}}{\partial D} & \frac{\partial \dot{X}}{\partial I} \\ \frac{\partial \dot{Q}}{\partial D} & \frac{\partial \dot{Q}}{\partial I} \\ \frac{\partial \dot{S}}{\partial D} & \frac{\partial \dot{S}}{\partial I} \end{bmatrix} \bigg|_{\substack{x = x_r \\ u = u_r \\ I = I_r}} \quad (3.5)$$

$$C = [1 \ 0 \ 0]$$

$$D = [0 \ 0]$$

In more details, matrix  $A$ :

$$A = \begin{bmatrix} \mu(Q_r, I_r) - D_r & \frac{q_0 \mu_{max} \mu_I X_r}{Q_r^2} & 0 \\ 0 & -\mu_{max} \mu_I & \frac{\rho_{max} K_S}{(S_r + K_S)^2} \\ -\frac{\rho_{max} S_r}{S_r + K_S} & 0 & -D_r - X_r \frac{\rho_{max} K_S}{(S_r + K_S)^2} \end{bmatrix} \bigg|_{\substack{x = x_r \\ u = u_r}} \quad (3.6)$$

and  $B$ :

$$B = \begin{bmatrix} -X_r & X_r \mu_{max} \left(1 - \frac{q_0}{Q_r}\right) \frac{(K_{sI} - I_r^2/KiI)}{(I_r + K_{sI} + I_r^2/KiI)^2} \\ 0 & -Q_r \mu_{max} \left(1 - \frac{q_0}{Q_r}\right) \frac{(K_{sI} - I_r^2/KiI)}{(I_r + K_{sI} + I_r^2/KiI)^2} \\ S_{in} - S_r & 0 \end{bmatrix} \begin{matrix} x = x_r \\ u = u_r \\ I = I_r \end{matrix} \quad (3.7)$$

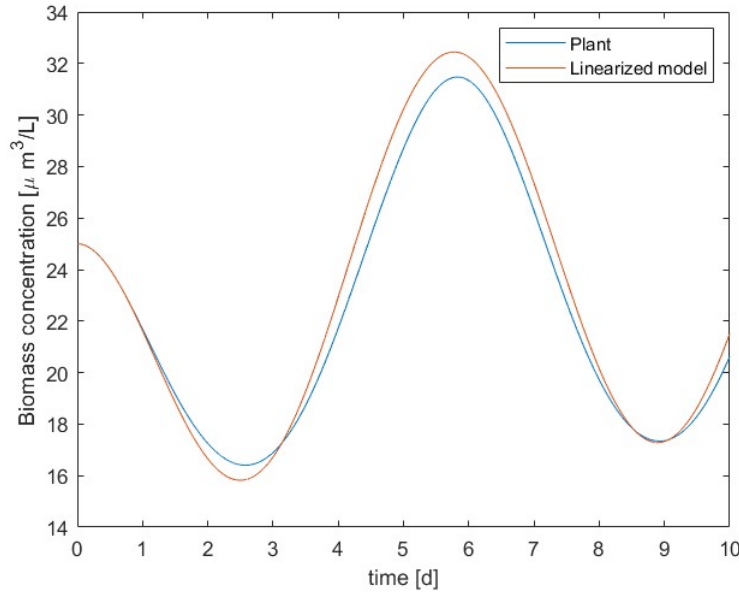


Figure 3.1: Output from the true system and the linearized model

It can be seen in Figure 3.1 that the linearized model describes well the dynamics of the nonlinear model even far from the equilibrium point with a maximum error between the two outputs of approximately 5%, but one of the possibilities to improve further is to make the linearization making the derivative directly by  $\mu_I$  under the strong assumption to know how exactly the light influences growth every time as well, that is perfectly know

parameter values in the function  $\mu(I)$ . This would return another matrix B as follow:

$$B = \left[ \begin{array}{cc} -X_r & X_r \mu_{max} (1 - q_0/Q_r) \\ 0 & -Q_r \mu_{max} (1 - q_0/Q_r) \\ S_{in} - S_r & 0 \end{array} \right] \left| \begin{array}{l} x = x_r \\ u = u_r \\ \mu(I) = \mu(I_r) \end{array} \right. \quad (3.8)$$

We notice, however, that the influence of light effect compared to the dilution rate is much smaller, so we decided to keep the first approximation to not complicate the assumptions and be content of the linearization error. So now it is possible to write the derivatives for small perturbation from the equilibrium defined in eq.(3.3) in a linear state space representation:

$$\left\{ \begin{array}{l} \delta \dot{x}(t) = \begin{bmatrix} 0 & 3.64 & 0 \\ 0 & -1.29 & 42.73 \\ -2.84 & 0 & -1068.94 \end{bmatrix} \delta x(t) + \begin{bmatrix} -25 & 1.52e - 05 \\ 0 & -2.43e - 06 \\ 99.95 & 0 \end{bmatrix} \delta u(t) \\ \delta y(t) = [1 \ 0 \ 0] \delta x(t) + [0 \ 0] \delta u(t) \end{array} \right. \quad (3.9)$$

### 3.3 Discretization

Physical real world systems are naturally modelled as continuous-time, but when we approach them through sampling, we need to threat a discrete-time model, in order to design a controller. In matlab we can discretize a state space system with the `c2d()` command which receives as input, the `ss` object and a sampling time ( $T_s = 0.01 \text{ days} \sim 15 \text{ min}$  in our case). The default continuous-discrete conversion methods is the zero order hold which provides an exact match between the continuous and discrete time systems in the time domain for staircase inputs. The new **linear discrete-time** state space system is:

$$\left\{ \begin{array}{l} \delta x(k+1) = \begin{bmatrix} 0.99 & 3.64 & 0.03 \\ -0.02 & 0.71 & 0.03 \\ -0.003 & -0.002 & -8.17e-05 \end{bmatrix} \delta x(k) + \begin{bmatrix} -5.82 & 3.53e-06 \\ 0.93 & -5.64e-07 \\ 0.11 & -9.35e-09 \end{bmatrix} \delta u(k) \\ \delta y(k) = [1 \ 0 \ 0] \delta x(k) + [0 \ 0] \delta u(k) \end{array} \right. \quad (3.10)$$

# Chapter 4

## Controllers

Maintaining an optimum operating point is crucial to maximize the productivity in cultivation of microalgae. The need to follow suitable harvest profiles shall be satisfied in order to maintain a high level yields while the control also has to reject the many sources of disturbance that can cause a reduction of the biomass density and compromise the output. Hence, the control problem is formulated so as to follow a certain reference trajectory for biomass that changes over time near the operating region and to reject the disturbance represented by light variation from the optimal value. To pursue this purpose, a generic closed-loop control scheme is adopted: the output is measured and its value is used by the controller to compute the next input that will feed the plant. The trajectory reference start at the equilibrium for the biomass concentration, then, after 2 days, drops to  $23 [\mu\text{m}^3/L]$  to jumps at  $28 [\mu\text{m}^3/L]$  at 6.5 days. Figure 4.1 shows the reference profile.

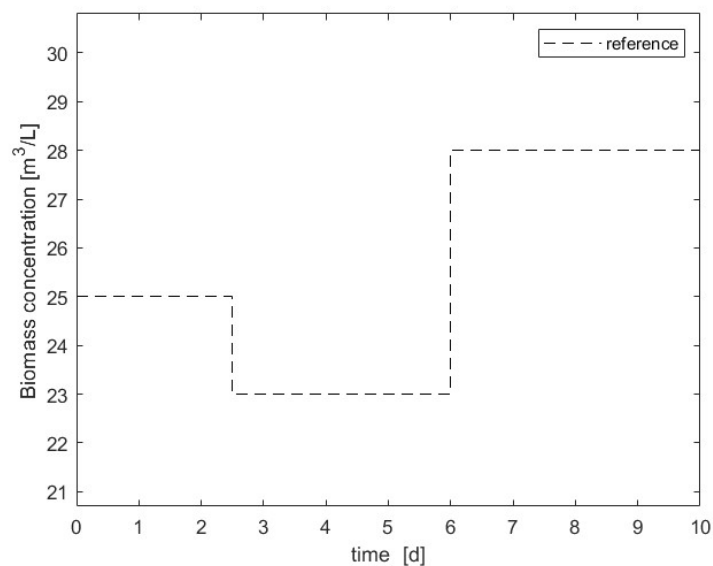


Figure 4.1: Trajectory of reference

Two different controllers were considered in this work: a control through a *proportional, integral, derivative* action (PID); a control through a model predictive control (MPC).

## Plant

The true system in the simulation is represented by the s-function and it has been also shown that the non-linear state space system in eq.(3.1) can be linearized around the equilibrium point. The output measures are taken from the true system, while the linear-discrete model is used from the MPC to do the calculations and find the optimal discrete input, which, once is made continuous, is used to feed the true system given by the s-function. Both PID and MPC simulations have a sampling time  $T_s = 0.01 d$ .

## Disturbance

The disturbance in the system is represented by the light intensity that hits the photobioreactor. We suppose a constant value  $I_{opt} = 547 \mu E m^{-2} s^{-1}$  at the equilibrium always available, while a sine fluctuation of amplitude  $I_{opt}/4$  and of one period per day is added as disturbance from the working point. Figure 4.2 shows the light intensity measured over the course of a day.

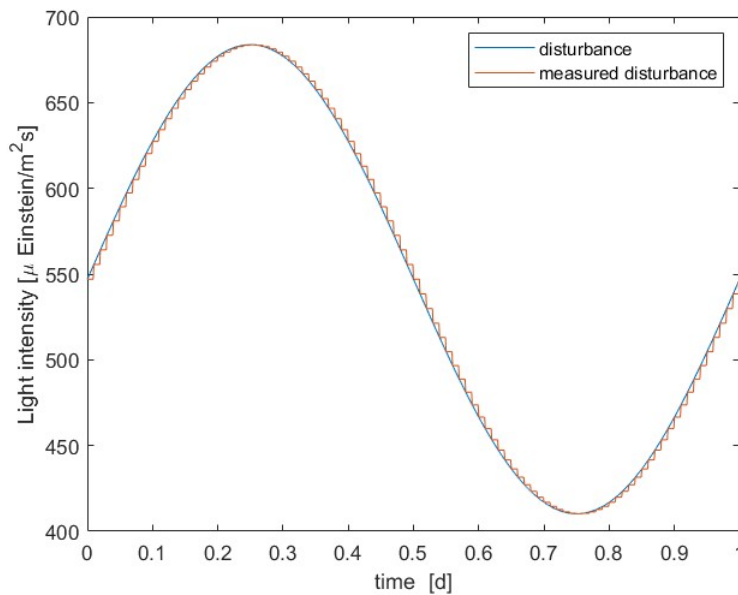


Figure 4.2: Light intensity with disturbance and measured disturbance over 1 day



## 4.1 PID control

The PID is a feedback-based control loop commonly used in industrial control systems and other applications that need constantly modulated control. It calculates an error value as the difference between a desired reference point and a measured process output and applies a correction based on proportional, integral, and derivative terms. Evaluating the performance of PID control method is important to be able to compare the results against the MPC. Due to the fact that we are operating at a certain region, we must inject the constant reference input  $D_r$  to the system. The error term, defined by the difference from the reference value and the output measure, is sent directly to the PID. Also a negative gain is inserted after the PID to change sign on the control because we want to manage the two cases:

1.  $e(k) = r(k) - y(k) \leq 0$  i.e. when the output value is above the reference. In this case the error is negative but we want to increment the dilution rate to lower biomass concentration
2.  $e(k) = r(k) - y(k) \geq 0$  i.e. when the output value is under the reference. In this case the error is positive but we want to decrement the dilution rate to let biomass grow

where  $e(k)$  is the error,  $r(k)$  is the reference and  $y(k)$  is the output at time  $k$ . This explains why we need to insert the negative gain at the end of PID controller.

Because the reference trajectories is a piece-wise constant function (see Figure 4.1), the derivative term of the PID would have some jumps and would saturate the input. To overcome this, for the derivative part, the reference was neglected and only the output is taken in consideration. To be able to tune the controller as we want, we do not use the matlab PID block (that uses transfer function) already available on simulink but we implement it manually. We need also to set the input initial value for the derivative block to  $\frac{-K_d X_r}{T_s}$  (it is multiplied by gain  $K_d$  because the block is after the derivative gain).

### 4.1.1 Gain tuning

The three degrees of freedom of the controller are the proportional gain  $K_p$ , the time constant of the integral action  $T_i$ , and the time constant of the derivative action  $T_d$ . The effect of the integral and derivative action are briefly recalled: An integral term increases action in relation not only to the error but also the time for which it has persisted. So, if the applied force is not enough to bring the error to zero, this force will be increased as time goes by. A derivative term does not consider the magnitude of the error, but the rate of change of error, trying to bring this rate to zero. It aims at flattening the

error trajectory into a horizontal line, damping the force applied, and so reduces overshoot. Knowing the effect of integral and derivative action on the control,  $T_i$  and  $T_d$  are manually tuned via trial and error verifying that the saturation of input is avoided and seeking a almost zero error at steady state while avoiding overshoot. The values found are  $T_i = 2$  and  $T_d = 0.01$ . For the tuning of the proportional gain, a cost function is used, define as the integral absolute error through the entire simulation:

$$J_e = \sum_{k=1}^N |r(k) - y(k)|$$

The goal is to find the proportional gain  $K_p$  that minimizes the cost function, meaning that the output is close to reference as long as possible. Different simulations are run with different  $K_p$  ranging from 0.01 to 1 and the cost function is calculated for every simulation. From table 4.1 it is possible to see that the optimal gain should be  $K_p = 0.5$ , however, analyzing the plots of inputs and outputs at different values of  $K_p$  shown in Figure 4.3, it can be seen that for values of  $K_p$  greater than 0.1 the proportional part makes the input to saturate, while the output even overshoots. It is preferable to avoid saturation and overshooting, thus,  $K_p = 0.1$  has been chosen for the proportional gain as trade-off between proper control aggressiveness and enough distance from saturation limits.

$K_p$	$J_e$
0.01	1550
0.05	606.79
<b>0.1</b>	<b>316.41</b>
0.3	125.39
0.5	122.43
1	134.69

Table 4.1: Proportional gain and relative cost function values

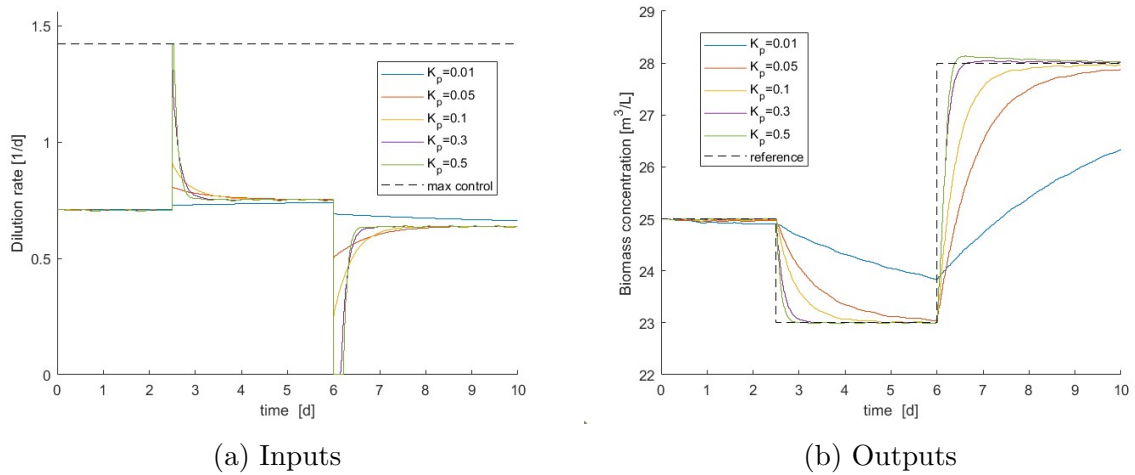


Figure 4.3: Effect of different gains  $K_p$  on manipulated variable and on output

## 4.2 Model predictive control

Model predictive control is an advanced approach of process control. Model predictive controllers rely on dynamic process models to anticipate future events and can take control actions accordingly. PID controllers do not have this predictive ability. Therefore, at each control step, the MPC controller:

1. Explores all possible sequences<sup>1</sup> of  $N$  control-actions, considering their effect on the system in the next PH (Prediction Horizon) steps. Then it picks-up the “optimal” sequence
2. Applies only the first control-action of the “optimal” sequence of  $N$  control actions and wait for a new measurement
3. When a new measurement comes in, the controller updates/re-computes the optimal sequence of control actions for the next PH steps
4. Then it will apply the first move of the optimal sequence and so on

### 4.2.1 Cost function

The goal of the MPC control, is to find the optimal control action sequence that minimize a predefined cost function  $J$ . There are a lot of different optimization problem, some very hard to solve. In this thesis, the aim is to follow a reference trajectory without departing too much from the suggested input. Assuming a quadratic weights of the difference

<sup>1</sup>compatible within constraints

between the values,  $J$  takes exactly the form of a *quadratic cost for reference tracking*:

$$J = \sum_{k=1}^{PH} (\hat{y}(kT) - y_0(kT))^T Q (\hat{y}(kT) - y_0(kT)) + (u(kT) - u_0(kT))^T R (u(kT) - u_0(kT)) \quad (4.1)$$

where  $PH$  is the prediction horizon,  $y_0(kT)$  is the reference signal we want to track at time  $kT$ ,  $\hat{y}(kT)$  is the output value,  $u_0(kT)$  is the reference input value equal to  $D_r$ , and  $u(kT)$  is the input feed at time  $kT$ ,  $Q > 0$ ,  $R > 0$  are positive definite matrices. Finding the optimal sequence of control action means minimize this cost function, that provides a trade-off between the predicted control error and the cost of a control action deviation from the suggested input, limiting controller aggressiveness through the weights  $Q$  and  $R$ . The weights  $Q$  and  $R$  are set after trial and error, keeping in mind the different orders of magnitude. We find a good balance of following reference, without saturating the input, for  $Q$  and  $R$  equal to 0.015 and 1 respectively.

### 4.2.2 Input constraints

Input constraints can be represented as a linear inequality. This arrangement is crucial when it comes to solve a quadratic programming problem like the one in our case.

$$\begin{bmatrix} -1 \\ 1 \end{bmatrix} u(kT) \leq \begin{bmatrix} -u_{min} \\ u_{max} \end{bmatrix} \quad (4.2)$$

where  $u_{min} = 0 \frac{1}{d}$  and  $u_{max} = 1.42 \frac{1}{d}$

### 4.2.3 Perturbation domain

These constraints work in the original system. The process of linearization bind us to work around the equilibrium point where the linearization takes place, so we now have to translate the problem in the *perturbation* domain instead of the full system. This brings the need to define the reference values and constraints in the new domain. All the new reference values are now set to 0, because we do not want to move away from the equilibrium point. The input constraints are calculated by subtracting the old constraints

with the reference value  $D_r$  and they become:

$$\begin{bmatrix} -1 \\ 1 \end{bmatrix} \delta u(kT) \leq \begin{bmatrix} 0.71 \\ 0.71 \end{bmatrix} \quad (4.3)$$

### 4.3 Kalman filtering

In control theory, a state observer or state estimator is a system that provides an estimate of the internal state of a given real system, from measurements of the input and output of the real system. The problem of find the estimate for the states occurs when not all these are directly measurable from sensors. To know at every sample time which is the state, observers are used that can give the estimation of all the states utilizing the information that measures can provide. The gold standard for state estimation it is the so called Kalman filter, which is the statistically optimal observer given the model error and measurement noise properties. It is assumed a gaussian random process for noise with zero mean and a given variance of  $W = 10^{-4}$  for model error and a variance of  $V = 10^{-2}$  for measurements noise and uncorrelation between each other. The Kalman filter is implemented through the simulink block which takes as inputs the input  $u$ , the disturbance  $d$  and the measurement affected by noise  $y$  and returns the estimates of the state.



# Chapter 5

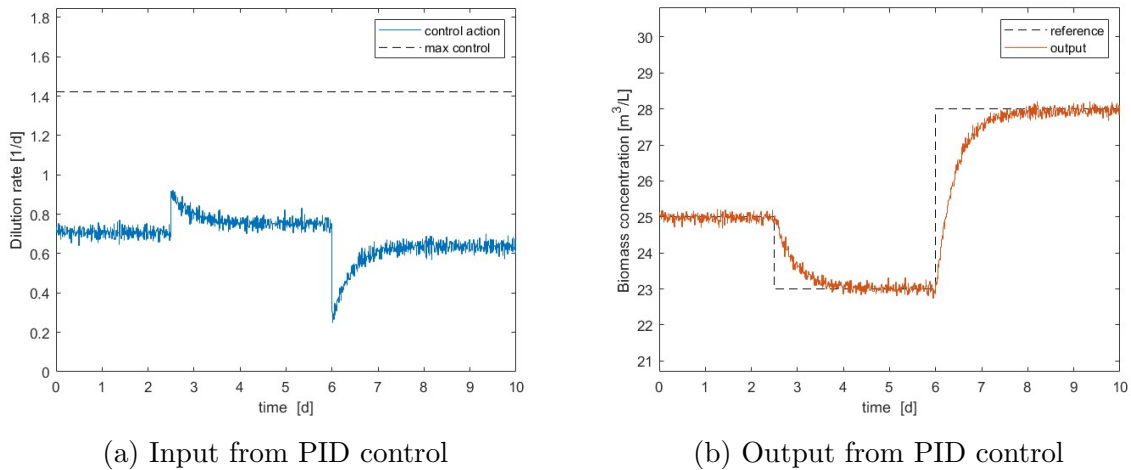
## Results and discussion

In this chapter it is shown the result of the simulation between the two control strategies. The simulation scheme is structured in this way: the real photobioreactor is represented by a set of nonlinear equations, and the dynamics are processed by the s-function. The measurements obtained are used for feedback, influencing the computation of the input of the two controllers, thus closing the feedback loop. The MPC also uses the linearized version at the equilibrium point of the Droop model and calculates for that model the optimal sequence of input for each new sample. Since the model has been linearized, the MPC is evaluating and calculating the optimal input sequence in the perturbed system. Nominal values are added when returning to the true unperturbed system. To update the state, Kalman estimator is placed, which uses the linear model and measurements to obtain the best estimate of the state. The model parameters do not vary over time, the reference sequence is a step function, and the disturbance is represented by light intensity and it is a sine wave that sums to the optimal reference value. The initial conditions are the equilibrium values and then control applies to track the reference. The duration of the experiment is set to 10 days, measurement are available every 15 minutes and are affected by white noise of zero mean and variance equal to 0.01.

### 5.1 PID performance

PID control is used as a landmark to evaluate controllers performance. In fact, PID control is well suited for reference tracking in systems that receive a single input and provide a single output. The PID controller provides good performance and it allows the output to reach steady state after approximately two days with almost a zero steady state error. The control input never saturates but it can be seen that is a lot noisy, making the control to changes very fast between one sample to the next one. That is because the noise on the measurements makes the signal to rapidly vary, and that makes the derivative part

of PID to amplify noise as it is shown in Figure 5.2. To remedy the problem, it should be inserted before the derivative block a low-pass filter capable of eliminating noise at high frequencies.



(a) Input from PID control

(b) Output from PID control

Figure 5.1: Input and output response from simulation with PID

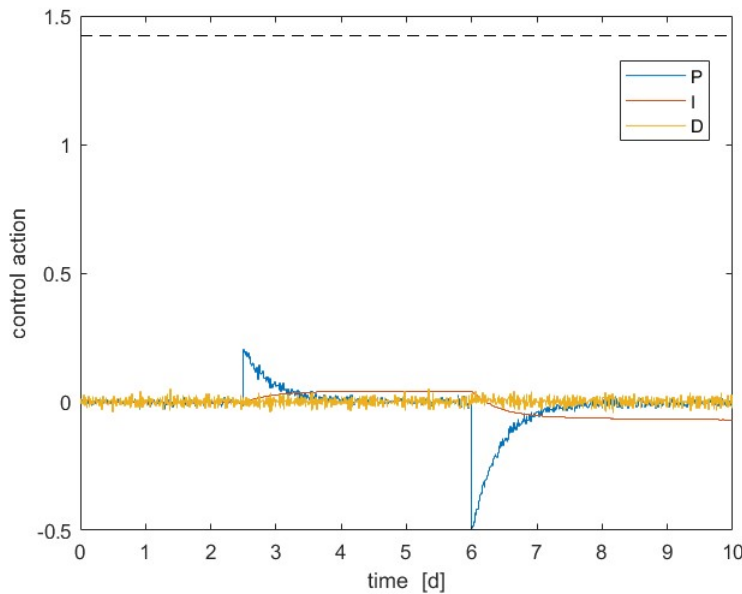


Figure 5.2: Contribution of proportional, integral, derivative parts

## 5.2 MPC performance

MPC plots show good performance for the simulation with the MPC controller: fast tracking for reference without any overshoot and the control input never saturate through



the entire simulation. However, it can be seen a small error at steady state. Probably, the trade off between the amount of dilution rate to be applied and the reference to be followed makes the system settle below the target. Increasing the weight for reference tracking within the cost function, i.e. increasing the value of  $Q$ , could achieve zero error but with the risk of saturating the input. Another possible solution could be to implement an integral action that would bring the error to zero in steady state.

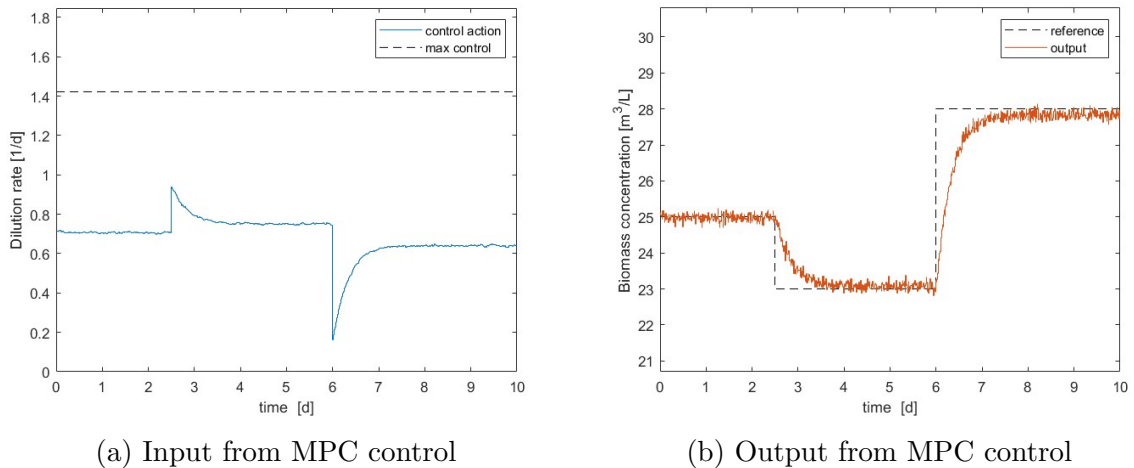


Figure 5.3: Input and output response from simulation with MPC

MPC is a very advanced tool. It can handle multiple inputs and multiple outputs, control even nonlinear systems and work on certain input and output constraints. In addition, the cost function is fully customizable according to the type of the problem. One can give heavier penalties to input action, to output and to input changes, penalize just one, two or all. One can also use cost functions for regulation to 0 or for tracking a reference, or for the objective of the maximization or minimization of one or more variables. This customization allows control via MPC to be much more flexible and address many more complex problems than PID can. However, the popularity of PID lies in its easy use, and there are plenty of methods that help with weight tuning. The MPC, in addition, is extremely dependent on the model used. Working with biological systems, many variables come into play, and it is not trivial to understand how these variables affect each other, and which of these stand out for importance and for effects on the system. This thesis aims to solve a simple problem: tracking of a reference in a single-input single-output system. This is the kind of problem very suitable for a PID. The main objective, however, is to implement an MPC knowing that in the future it will be increased in complexity, adding more inputs and more sensors and further improving the mathematical model, adding for example the lipid component that constitutes the microalgae and their dynamics, and evaluate other cost functions that takes in consideration the maximization

of a variable or cost functions that puts constraints in the input variations.

### 5.3 State estimation

In this simulation, a linear Kalman filter was implemented in order to estimate the state of the system. The estimate is subsequently sent to the MPC block to update it on the plant condition. The values of the model error variance and measurement noise variance are assumed to be known. In order to compare the goodness of the estimate, a version of the simulation allows to derive the true values of the three states, which are paired with their own estimates provided by the Kalman filter, as shown in Figure 5.4. The plots show

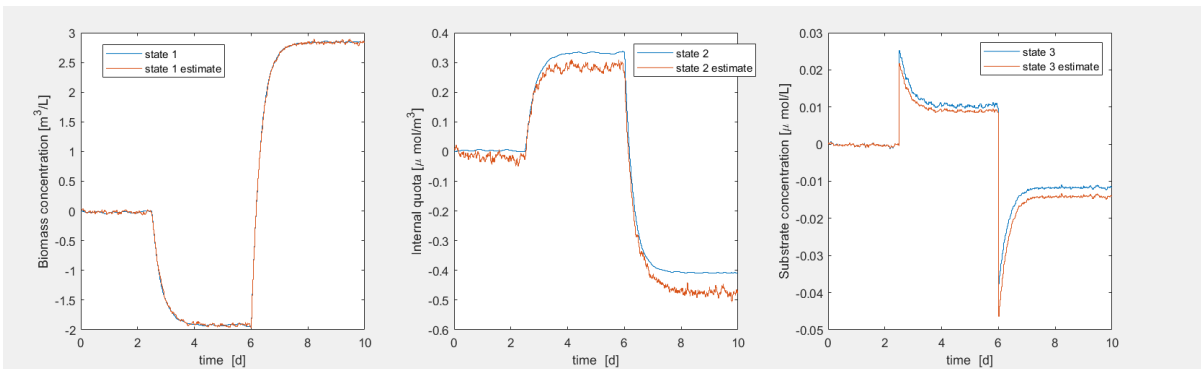


Figure 5.4: States of plant and their estimate through Kalman filter

that the estimator was able to correctly calculate the state even with noisy measurements. However, a bias error can be seen for state 2 and state 3 due to the discrepancy between the linear model used for Kalman filtering and the nonlinear model represented by the photobioreactor.

### 5.4 Different set-ups

This section aims to demonstrate the customization of the simulator for different experimental set ups. For example, one can change the duration of the trial, the data sampling time, the reference trajectory, the available measurements, the constraints or the parameters. The simulator is now used for other types of experiments. All of the following plots are taken from the MPC simulation.

#### 5.4.1 Change of input constraints and output tracking weight $Q$

For this simulation, tighter input constraint have been chosen: the minimum value that the dilution rate can take is changed to  $u_{min} = D_r/2 = 0.36 \frac{1}{d}$  and it is changed the maximum

input value at  $u_{max} = 1 \frac{1}{d}$ . Then, it is tested a more aggressive control increasing the weight  $Q$  to 10 in the cost function. In Figure 5.5 it is showed the result of the simulation.

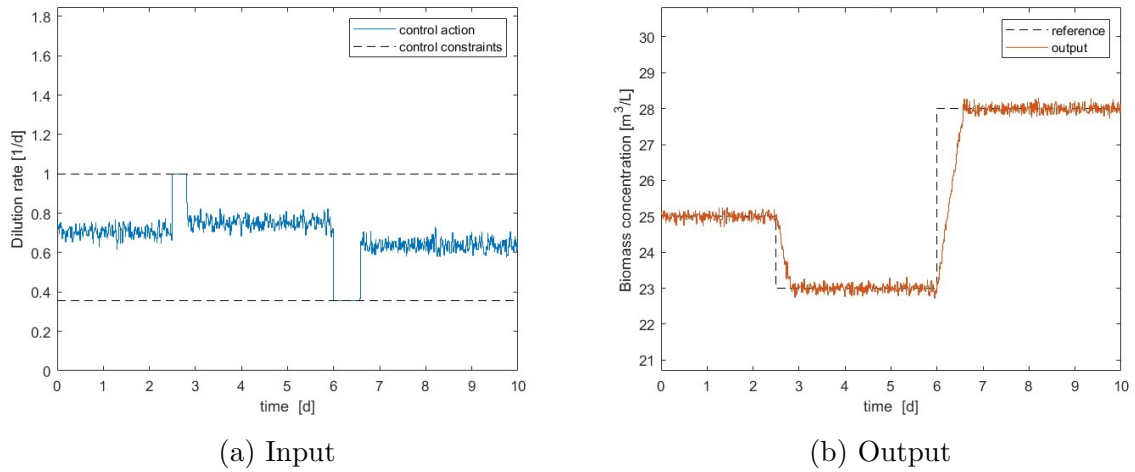


Figure 5.5: Input and output response with different constraints and weight

### 5.4.2 Change of reference trajectory

For this simulation, a different reference trajectory is selected.

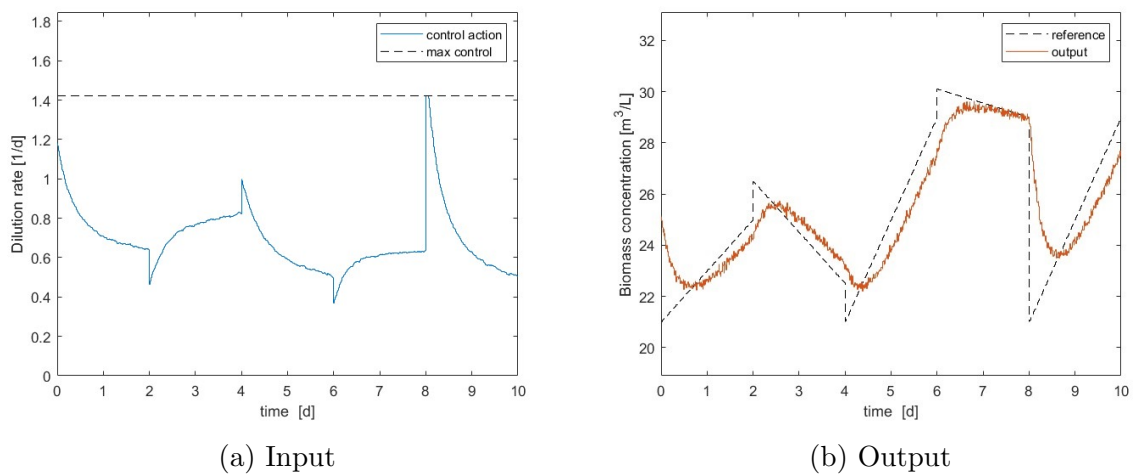


Figure 5.6: Input and output response with different reference trajectory

### 5.4.3 Change of sampling time

For this simulation, the sampling time is  $T_s = 0.3 d$

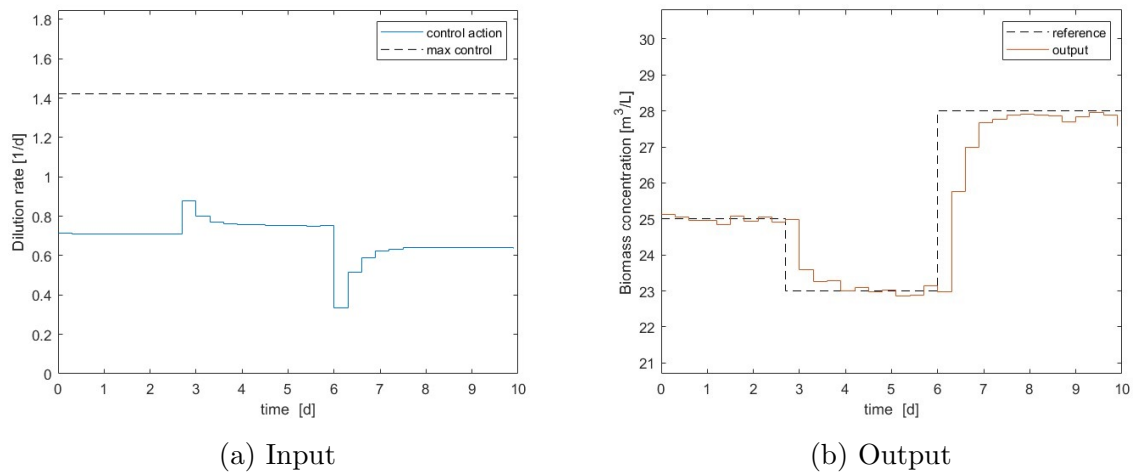


Figure 5.7: Input and output response with different sampling time

# Chapter 6

## Conclusions

The cultivation of algae is a very complex operation since their growth is influenced by many factors for which is difficult to attribute the exact effects that affect the culture and how these effects act, in a combined way, on the organism. In this thesis a mathematical model was developed to describe the ability of microalgae to absorb nutrients in the solution, and to use their energy reserves to grow and duplicate. In particular, it was taken into account the inhibitory effect of light, which at high intensity, can damage the proper development of algae. Then, a simulator was built from scratch, it can take advantage of the knowledge provided by the mathematical model to predict the growth of microalgae and it can give a prediction of the biomass inside the tank. The concentration had to follow a specific trend, so it was built a model predictive controller, able to predict the effects of the input sequence on the system and determine via numerical optimization the most effective dilution rate sequence to apply. Next to the controller a Kalman state estimator was placed which is capable, using the linear model and measurements, to provide state estimation not only of the biomass, but also of the internal nitrogen quota and nutrient concentration in the photobioreactor. The controller showed good performance at tracking the reference trajectory, and was able to handle the disturbance that caused the light to move away from the optimal value. The estimator proved useful in calculating the state estimate, providing good results with noisy measurements and model uncertainty. An additional experiment was done with a PID controller in order to have a reference of the controller qualities. The PID and MPC showed comparable results to each other, which therefore suggests that good efficiency has been achieved in handling the control problem. MPC is a very advanced tool, and its advantages over PID are not appreciated in this thesis. By adding complexity to the model, putting more inputs and more outputs to the system, applying more constraints, and using a different cost function, the benefits of MPC over traditional PID can be more easily highlighted. The simulator is still in an embryonic stage, further developments are needed to make the

tool more effective and reliable but it stands as a solid starting point that can already be applied and associated to small experiments with laboratory bioreactors.

# Appendix A

## Simulink schemes

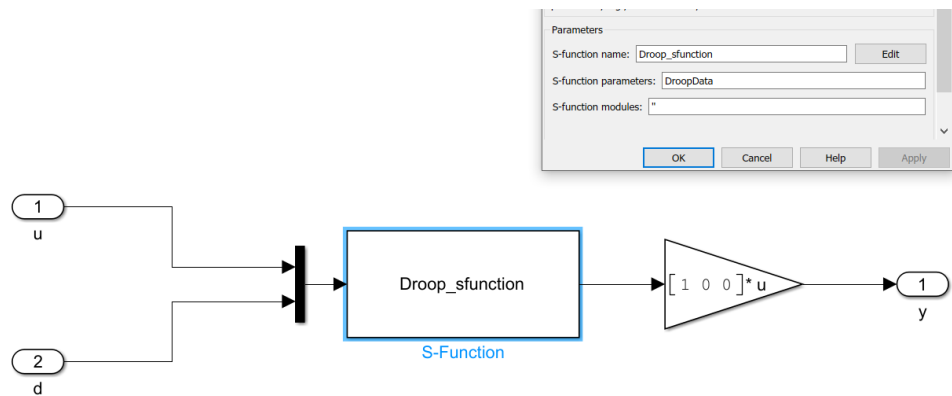


Figure A.1: Plant via s-function block

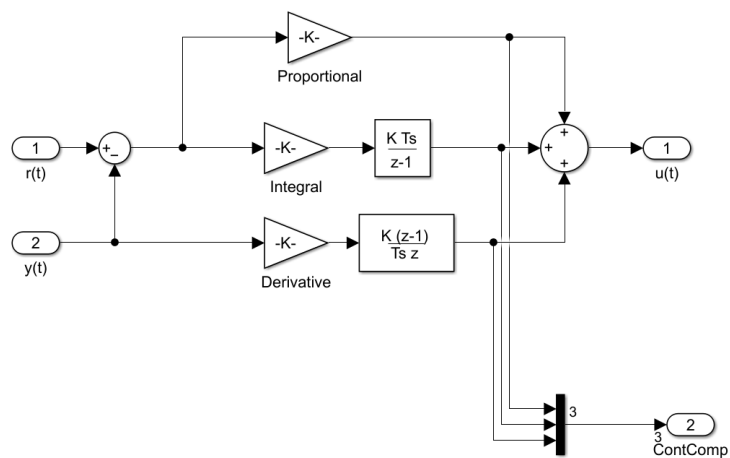


Figure A.2: PID block

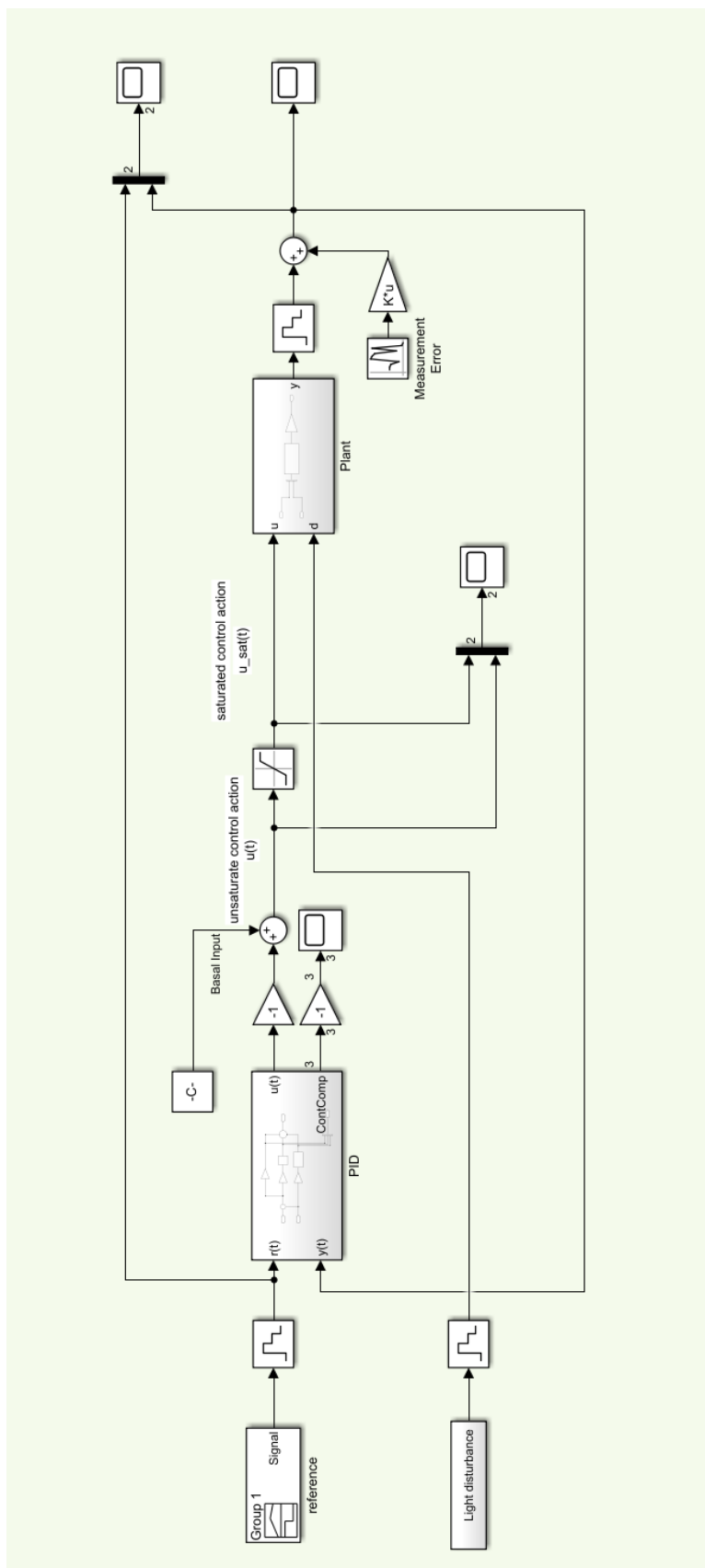


Figure A.3: Simulation with PID controller



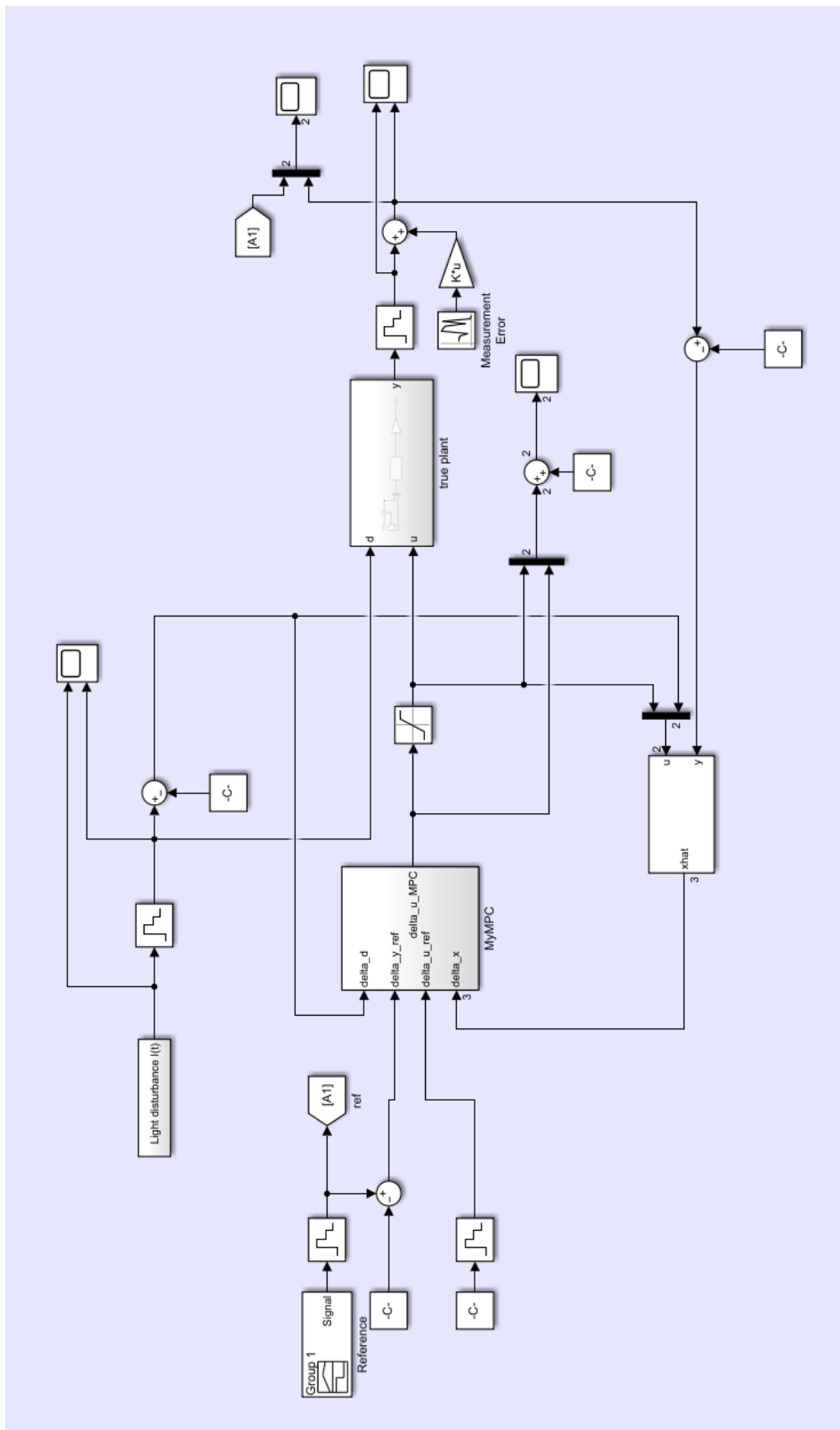


Figure A.4: Simulation with MPC controller



# Appendix B

## Code listing

Listing B.1: s-function of the Droop model

```
1 function [sys ,x0 ,str ,ts] = Droop_sfunction(t ,x ,u ,flag ,par)
2
3 % x(1)=X biomass concentration [micro m^3/L]
4 % x(2)=Q internal quota [micro mol/micro m^3]
5 % x(3)=S substrate [concentration micro mol/L]
6 switch flag ,
7
8     %%%%%%%%%%%
9     % Initialization %
10    %%%%%%%%%%%
11    case 0 ,
12        [sys ,x0 ,str ,ts]=mdlInitializeSizes (par) ;
13
14    %%%%%%%%%%%
15    % Derivatives %
16    %%%%%%%%%%%
17    case 1 ,
18        sys=mdlDerivatives (t ,x ,u ,par) ;
19
20    %%%%%%%%%%%
21    % Outputs %
22    %%%%%%%%%%%
23    case 3 ,
24        sys=mdlOutputs (t ,x ,u ,par) ;
25
```

```

26     case { 2, 4, 9 },
27         sys = [];
28
29     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
30     % Unexpected flags %
31     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
32     otherwise
33         DASTudio.error('Simulink:blocks:unhandledFlag', num2str(flag
34             ));
35 end
36
37 % end sfuntmpl
38
39 %
40 %

```

---

```

41 % mdlInitializeSizes
42 % Return the sizes, initial conditions, and sample times for the
43 % S-function.

```

---

```

44 %
45 function [sys,x0,str,ts]=mdlInitializeSizes(par)
46
47 sizes = simsizes;
48
49 sizes.NumContStates = 3;
50 sizes.NumDiscStates = 0;
51 sizes.NumOutputs = 3;
52 sizes.NumInputs = 2;
53 sizes.DirFeedthrough = 0;
54 sizes.NumSampleTimes = 1; % at least one sample time is needed
55
56 sys = simsizes(sizes);

```

```

57
58 %
59 % initialize the initial conditions
60 %
61 x0 = par.x_0;
62
63 %
64 % str is always an empty matrix
65 %
66 str = [];
67
68 %
69 % initialize the array of sample times
70 %
71 ts = [-1 0]; %inherited sample time
72
73 % end mdlInitializeSizes
74
75 %
76 %

```

---

```

77 % mdlDerivatives
78 % Return the derivatives for the continuous states.
79 %

```

---

```

80 %
81 function sys=mdlDerivatives(t,x,u,par)
82 muI=(u(2)/(par.KsI+u(2)+u(2)^2/par.KiI));
83 mu= par.mumax*(1-par.q0/x(2))*muI;
84 ro= par.romax*(x(3)/(x(3)+par.KS));
85
86 x1dot= x(1)*(mu-u(1));
87 x2dot= ro-mu*x(2);
88 x3dot= -ro*x(1)+u(1)*(par.Sin-x(3));
89

```

```

90 sys = [x1dot x2dot x3dot];
91 % end mdlDerivatives
92
93 %
94 %

```

---

```

95 % mdlOutputs
96 % Return the block outputs.
97 %

```

---

```

98 %
99 function sys=mdlOutputs(t,x,u,par)
100
101 sys = [x(1) x(2) x(3)];
102
103 % end mdlOutputs

```

Listing B.2: function that loads Droop model parameters

```

1 function DroopData=LoadMPCDroop()
2
3     %% Droop parameters
4
5     % Simulation parameters
6     DroopData.Ts=0.01; %[d] d=days
7     DroopData.SimTime=10; %[d]
8
9     % System parameters
10    mumax=2; % [1/d]
11    KsI=150; % [micro Einstein/m^2*s]
12    KiI=2000; % [micro Einstein/m^2*s]
13    Iopt=547; % [micro Einstein/m^2*s]
14    romax=9.3; % [micro mol/L]
15    q0=1.8; % [micro mol/micro m^3]
16    KS=0.105; % [micro mol/L]
17    Sin=100; % [micro mol/L]

```

```

18
19 % Equilibrium point
20 Xr=25; % Benattia et al (2015) IFAC–PapersOnLine
      192–197
21 Ir=Iopt ;
22 muI=Ir / ( Ir+KsI+( Ir * Ir / KiI ) ) ;
23 delta = ((romax+mumax*muI*q0) *Xr–mumax*muI * ( Sin+KS) )
      ^2+4*mumax*muI*KS*romax*Xr ;
24 Qr=(mumax*muI * ( Sin+KS) +(romax+mumax*muI*q0) *Xr–sqrt (
      delta ) ) / ( 2*mumax*muI*Xr ) ;
25 Dr=mumax*(1–q0/Qr) *muI ;
26 mu=mumax*(1–q0/Qr) *muI ;
27 Sr=((KS*mu*Qr) / ( romax–mu*Qr ) ) ;
28 ro=romax*( Sr / ( Sr+KS) ) ;
29
30 % Continuous Time State Space Model (Linearization on
      equilibrium point)
31 Ac=[mu–Dr, q0*mumax*muI*Xr/(Qr^2), 0;
32      0, –mumax*muI, romax*KS/((Sr+KS)^2);
33      –romax*(Sr/(Sr+KS)), 0, –Dr–((Xr*romax*KS)/((Sr+KS)
      ^2))];
34 dXdI=Xr*mumax*(1–q0/Qr) * ((KsI–Ir * Ir / KiI) / (( Ir+KsI+Ir
      * Ir / KiI ) ^2)); %partial derivative of Xdot respect
      to I
35 dQdI=–Qr*mumax*(1–q0/Qr) * ((KsI–Ir * Ir / KiI) / (( Ir+KsI+
      Ir * Ir / KiI ) ^2)); %partial derivative of Qdot
      respect to I
36 Bc=[–Xr; 0; Sin–Sr];
37 Mc=[dXdI; dQdI; 0];
38 Cc=[1 0 0];
39 Dc=[0,0];
40
41 % Exact Discretization using Matlab Toolbox
42 sys_c=ss (Ac, [Bc,Mc], Cc, Dc);
43 sys_d=c2d (sys_c, DroopData.Ts);
44
45 % System

```

```

46     DroopData.System.A=sys_d.A;
47     DroopData.System.B=sys_d.B(:,1);
48     DroopData.System.M=sys_d.B(:,2);
49     DroopData.System.C=sys_d.C;
50     DroopData.System.D=sys_d.D;
51     DroopData.x_0=[Xr Qr Sr];
52     DroopData.SystemModel=DroopData.System;
53
54     DroopData.mumax=mumax;
55     DroopData.muI=muI;
56     DroopData.KsI=KsI;
57     DroopData.KiI=KiI;
58     DroopData.KS=KS;
59     DroopData.romax=romax;
60     DroopData.q0=q0;
61     DroopData.Sin=Sin;
62     DroopData.Iopt=Iopt;
63     DroopData.Xr=Xr;
64     DroopData.Qr=Qr;
65     DroopData.Sr=Sr;
66     DroopData.Ir=Ir;
67     DroopData.Dr=Dr;
68     DroopData.u_max=2*Dr;
69     DroopData.u_min=0;
70     DroopData.delta_u_max=DroopData.u_max-Dr;
71     DroopData.delta_u_min=DroopData.u_min-Dr;
72     DroopData.sys_c=sys_c;
73     DroopData.sys_d=sys_d;

```

Listing B.3: main file for simulation run

```

1  clc
2  close all
3  clear all
4
5  addpath('Sources')
6
7  DroopData=LoadDroop();

```



```

8
9 %% PID Controller Parameters
10     PID_controller_parameters.Ti=2;
11     PID_controller_parameters.Td=0.01;
12     PID_controller_parameters.Kp=0.1;
13     PID_controller_parameters.Ki=PID_controller_parameters .
        Kp/PID_controller_parameters.Ti;
14     PID_controller_parameters.Kd=PID_controller_parameters .
        Kp*PID_controller_parameters.Td;
15     PID_controller_parameters.Ts=DroopData.Ts;
16     PID_controller_parameters.Ie_max=0.5;
17     PID_controller_parameters.Ie_min=-0.5;
18
19 %% MPC Controller Parameters
20     % Prediction horizon
21     PH=100; %[steps]
22
23     % Cost Weights
24     Q_MPC=0.015;
25     R_MPC=1;
26
27     % References input
28     u_0=DroopData.Dr;
29     delta_u_0=0;
30
31     % Constraints
32     F=[-1
33         1];
34
35     f=[-DroopData.delta_u_min
36         DroopData.delta_u_max
37         ];
38     % Condensing
39     condensedMatrices=build_MPC_matrices(DroopData .
        SystemModel ,R_MPC,Q_MPC,F , f ,PH) ;
40
41     % System Dimension

```

```

42     n=size(DroopData.SystemModel.A,1);
43     p=size(DroopData.SystemModel.C,1);
44     m_input=size(DroopData.SystemModel.B,2);
45     m_dist=size(DroopData.SystemModel.M,2);
46
47 %% Model error and measurement noise
48     Q_model=10-4*eye(size(DroopData.SystemModel.A,1));
49     R_model=10-2*eye(size(DroopData.SystemModel.C,1));
50     S_model=zeros(size(DroopData.SystemModel.A,1),size(
        DroopData.SystemModel.C,1));
51 %% Simulating
52     sim('Full_simulation.slx')
53
54 %% Plots
55     figure
56
57 % PID
58     subplot(2,2,1)
59     stairs(SimulationVariables_PID.time,SimulationVariables_PID.
        signals.values(:,1),'—k')
60     hold on
61     stairs(SystemOutput_PID.time,SystemOutput_PID.signals.values)
62     hold off
63     ylim([0.9*min(SimulationVariables_PID.signals.values(:,1)) 1.1*
        max(SimulationVariables_PID.signals.values(:,1))])
64     ylabel('Biomass concentration [m3/L]')
65     xlabel('time [d]')
66     legend('reference','output')
67     title('PID')
68
69     subplot(2,2,3)
70     stairs(ControlAction_PID.time,ControlAction_PID.signals.values
       (:,1))
71     hold on
72     stairs([0 DroopData.SimTime],DroopData.u_max*[1 1],'—k')
73     hold off
74     ylim([DroopData.u_min 1.3*DroopData.u_max])

```

```

75 ylabel('Dilution rate [1/d]')
76 xlabel('time [d]')
77 legend('control action', 'max control')
78
79 % Plot of P, I and D contributions
80 subplot(3,1,3)
81 stairs(ControlComponents_PID.time, ControlComponents_PID.signals .
      values)
82 hold on
83 stairs([0 DroopData.SimTime], DroopData.u_max*[1 1], '—k')
84 hold off
85 ylabel('control action')
86 xlabel('time [d]')
87 legend('P', 'I', 'D')
88
89 % MPC
90 subplot(2,2,2)
91 stairs(SimulationVariables_myMPC.time, SimulationVariables_myMPC .
      signals.values(:,1), '—k')
92 hold on
93 stairs(SimulationVariables_myMPC.time, SimulationVariables_myMPC .
      signals.values(:,2))
94 hold off
95 ylim([0.9*min(SimulationVariables_myMPC.signals.values(:,1))
      1.1*max(SimulationVariables_myMPC.signals.values(:,1))])
96 ylabel('Biomass concentration [m3/L]')
97 xlabel('time [d]')
98 legend('reference', 'output')
99 title('MPC')
100
101 subplot(2,2,4)
102 stairs(ControlAction_myMPC.time, ControlAction_myMPC.signals .
      values(:,1))
103 hold on
104 stairs([0 DroopData.SimTime], DroopData.u_max*ones*[1 1], '—k')
105 hold off
106 ylim([DroopData.u_min 1.3*DroopData.u_max])

```

```
107 ylabel('Dilution rate [1/d]')
108 xlabel('time [d]')
109 legend('control action', 'max control')
```

# Bibliography

- [1] Susie Maestre. How are leds used for growing plants?, May 2020. <https://www.circuitbread.com/ee-faq/how-are-leds-used-for-growing-plants>, accessed 15/02/2022.
- [2] Rizwan Muhammad, Ghulam Mujtaba, Sheeraz Memon, Kisay Lee, and Naim Rashid. Exploring the potential of microalgae for new biotechnology applications and beyond: A review. *Renewable and Sustainable Energy Reviews*, 92, 09 2018.
- [3] Ying Shen, W. Yuan, Z.J. Pei, Q. Wu, and E. Mao. Microalgae mass production methods. *Transactions of the ASABE*, 52:1275–1287, 07 2009.
- [4] Wikipedia. Photobioreactor — Wikipedia, the free encyclopedia. [Online; accessed 12/03/2022].
- [5] Pooya Darvehei, Parisa A. Bahri, and Navid R. Moheimani. Model development for the growth of microalgae: A review. *Renewable and Sustainable Energy Reviews*, 97:233–258, 2018.
- [6] Gilbert Randrianarison and Muhammad Aqeel Ashraf. Microalgae: a potential plant for energy production. *Geology, Ecology, and Landscapes*, 1(2):104–120, 2017.
- [7] Beth A. Rasala and Stephen P Mayfield. Photosynthetic biomanufacturing in green algae; production of recombinant proteins for industrial, nutritional, and medical uses. *Photosynthesis Research*, 123:227–239, 2014.
- [8] Michael J. Behrenfeld, James T. Randerson, Charles R. McClain, Gene C. Feldman, Sietse O. Los, Compton J. Tucker, Paul G. Falkowski, Christopher B. Field, Robert Frouin, Wayne E. Esaias, Dorota D. Kolber, and Nathan H. Pollack. Biospheric primary production during an enso transition. *Science*, 291(5513):2594–2597, 2001.
- [9] Mayfield Stephen P. Georgianna D. Ryan. Exploiting diversity and synthetic biology for the production of algal biofuels. *Nature*, 488, 2012.

- [10] Jr. Metting, F B. Biodiversity and application of microalgae. *Journal of Industrial Microbiology and Biotechnology*, 17(5-6):477–489, 11 1996.
- [11] Laura Barsanti, Primo Coltelli, Valtere Evangelista, Anna Maria Frassanito, Vincenzo Passarelli, Nicoletta Vesentini, and Paolo Gualtieri. Oddities and curiosities in the algal world. In Valtere Evangelista, Laura Barsanti, Anna Maria Frassanito, Vincenzo Passarelli, and Paolo Gualtieri, editors, *Algal Toxins: Nature, Occurrence, Effect and Detection*, pages 353–391, Dordrecht, 2008. Springer Netherlands.
- [12] Rukhsar Shaikh, Afshan Rizvi, Soumya Pandit, Niteen Desai, and Ravishankar Patil. Chapter 21 - microalgae: Classification, bioactives, medicinal properties, industrial applications, and future perspectives. In Maulin Shah, Susana Rodriguez-Couto, Celia Bertha Vargas De La Cruz, and Jayanta Biswas, editors, *An Integration of Phycoremediation Processes in Wastewater Treatment*, pages 451–486. Elsevier, 2022.
- [13] Shanmugam Hemaiswarya, Rathinam Raja, R. Ravikumar, and Isabel Carvalho. Microalgae taxonomy and breeding. *Biofuel Crops: Production, Physiology and Genetics*, pages 44–53, 06 2013.
- [14] Sara Fantini. Use of variable intensity led illuminator to optimize microalgal growth in a planar photobioreactor. Master’s thesis, Politecnico di Torino, Turin, 2017/2018.
- [15] Industrial Plankton. Microalgae growth light spectrum, Jun 2021. <https://industrialplankton.com/2021/05/15/microalgae-light-spectrum/>, accessed 12/03/2022.
- [16] Azad Kalam. *Advances in eco-fuels for a sustainable environment*. Woodhead Publishing, an imprint of Elsevier, Duxford, United Kingdom, 2019.
- [17] Miguel G. García-González, Mercedes Guerrero. Outdoor cultivation of microalgae for carotenoid production: current state and perspectives. *Applied Microbiology and Biotechnology*, 04 2007.
- [18] John Kyndt. Algae cultivation systems – open ponds vs pbr, Nov 2010. [Online; accessed 20/02/2022].
- [19] Abu Yousuf. Chapter 1 - fundamentals of microalgae cultivation. In Abu Yousuf, editor, *Microalgae Cultivation for Biofuels Production*, pages 1–9. Academic Press, 2020.
- [20] Kalpesh K. Sharma, Holger Schuhmann, and Peer M. Schenk. High lipid induction in microalgae for biodiesel production. *Energies*, 5(5):1532–1553, 2012.

- [21] Michael Butler. Hybridomas, genetic engineering of. In Robert A. Meyers, editor, *Encyclopedia of Physical Science and Technology (Third Edition)*, pages 427–443. Academic Press, New York, third edition edition, 2003.
- [22] Pauline M. Doran. Chapter 9 - heat transfer. In Pauline M. Doran, editor, *Bioprocess Engineering Principles (Second Edition)*, pages 333–377. Academic Press, London, second edition edition, 2013.
- [23] James C Ogbonna, Toshihiko Soejima, and Hideo Tanaka. An integrated solar and artificial light system for internal illumination of photobioreactors. *Journal of Biotechnology*, 70(1):289–297, 1999. Biotechnological Aspects of Marine Sponges.
- [24] G. Bastin. *On-line Estimation and Adaptive Control of Bioreactors*. Process Measurement and Control. Elsevier Science, 2013.
- [25] Michael Alexander Brik Ternbach. *Modeling based process development of fed-batch bioprocesses: L-valine production by Corynebacterium glutamicum*. PhD thesis, Aachen University, Aachen, 2005. Aachen, Techn. Hochsch., Diss., 2005.
- [26] Christopher Klausmeier, Elena Litchman, Tanguy Daufresne, and Simon Levin. Phytoplankton stoichiometry. *Ecological Research*, 23:479–485, 05 2008.
- [27] Liang Song, Jian G. Qin, Shengqi Su, Jianhe Xu, Stephen Clarke, and Yichu Shan. Micronutrient requirements for growth and hydrocarbon production in the oil producing green alga botryococcus braunii (chlorophyta). *PLOS ONE*, 7(7):1–9, 07 2012.
- [28] Jacques Monod. The growth of bacterial cultures. *Annual Review of Microbiology*, 3(1):371–394, 1949.
- [29] J. B. S. Haldane. *Enzymes*. MIT Press, Cambridge, 1965.
- [30] Patrick Sorgeloos Patrick Lavens. Algal production, 1996. <https://www.fao.org/3/w3732e/w3732e06.htm#b15-2.3.1.5.\%20Temperature>, accessed 27/02/2022.
- [31] Laurent Rosso, Jean R. Lobry, and Jean Pierre Flandrois. An unexpected correlation between cardinal temperatures of microbial growth highlighted by a new model. *Journal of theoretical biology*, 162 4:447–63, 1993.
- [32] Ghjuvan Micaelu Grimaud, Francis Mairet, Antoine Sciandra, and Olivier Bernard. Modeling the temperature effect on the specific growth rate of phytoplankton: a review. *Reviews in Environmental Science and Bio/Technology*, 16:625–645, 2017.

- [33] M. R. Droop. Vitamin b12 and marine ecology. iv. the kinetics of uptake, growth and inhibition in *monochrysis lutheri*. *Journal of the Marine Biological Association of the United Kingdom*, 48(3):689–733, 1968.
- [34] H. De la Hoz Siegler, A. Ben-Zvi, R.E. Burrell, and W.C. McCaffrey. The dynamics of heterotrophic algal cultures. *Bioresource Technology*, 102(10):5764–5774, 2011.
- [35] H. De la Hoz Siegler, W.C. McCaffrey, R.E. Burrell, and A. Ben-Zvi. Optimization of microalgal productivity using an adaptive, non-linear model based strategy. *Bioresource Technology*, 104:537–546, 2012.
- [36] Sung Yoo, Dong Jeong, Jung Kim, and Jong Lee. Optimization of microalgal photobioreactor system using model predictive control with experimental validation. *Bioprocess and Biosystems Engineering*, 39, 08 2016.
- [37] S. Benattia, S. Tebbani, and D. Dumur. Nonlinear model predictive control for regulation of microalgae culture in a continuous photobioreactor. In *22nd Mediterranean Conference on Control and Automation*, pages 469–474, 2014.
- [38] Guillaume Goffaux. Exploration of robust software sensor techniques with applications in vehicle positioning and bioprocess state estimation. *Ph. D. dissertation, Ph. D. thesis*, 2010.
- [39] Seif Eddine Benattia, Sihem Tebbani, and Didier Dumur. Robust Nonlinear Model Predictive Control for Regulation of Microalgae Culture in a Continuous Photobioreactor. In *1st IFAC Conference on Modelling, Identification and Control of Nonlinear Systems (MICNON)*, saint-petersbourg, Russia, June 2015.