NTNU
Norwegian University of
Science and Technology

Optimization of a Reactor Network for Production of Butyl Butyrate by an Attainable Region Analysis

Halvor Aarnes Krog

Chemical Engineering and Biotechnology
Submission date: June 2017
Supervisor: Heinz A. Preisig, IKP

Norwegian University of Science and Technology
Department of Chemical Engineering

Preface

This master thesis started off as a mini project in the module Advanced Simulation, at the institute for chemical engineering at the Norwegian University of Science and Technology (NTNU). In the module, reactor optimization by the attainable region methodology was the subject. After the module was finished, I was allowed by Professor Heinz Preisig to continue working on the subject on my master's thesis. I would like to thank my supervisor Professor Heinz Preisig for his interest in my work, and that he always had time to discuss whenever I had questions. I would also like to thank my co-surpervisor Cansu Birgen, who helped me a lot with finding and applying the kinetic models, discussing attainable region theory and for proof reading of my report.

Finally, I would like to thank my friends from NTNU for five memorable years. And, of course, also my family for all the support throughout these years.

Abstract

Butyl butyrate is a promising supplement for jet fuel and diesel. It can be produced in a biotechnological process, by an esterification reaction between butanol and butyric acid. These intermediates can in turn be produced by fermentation. This work focuses on optimizing the reactor network which are required to convert sugar to the intermediates butanol and butyric acid, and the subsequent esterification of these intermediates to form butyl butyrate. The optimization was done by performing an attainable region analysis on the system.

Two different process designs was evaluated. In the first design, there were two separate fermentation networks, and the outlet flows were mixed and fed to the esterification reactor. In the second design, butanol was produced separately by fermentation. The outlet flow consisted of both substrate and butanol. It was fed to a reactor where butyric acid was produced by fermentation, and simultaneously converted to butyl butyrate.

The results showed that the maximum concentration of butyl butyrate was obtained by the second process design, which yielded maximum 16,42 g/L of butyl butyrate. This was considerably higher than the first design, which resulted in a maximum of 11,28 g/L of butyl butyrate. A possible reason for this deviation, is that the product inhibition of the cells which are producing butyric acid are reduced in the second process design. This is due to that butyric acid is reacting with the butanol to form butyl butyrate, hence, the concentration of butyric acid is maintained at low values. In addition, the feed concentration of butanol to the reactor producing butanol was higher in the second process design. The reason is because the butanol concentration was in effect diluted when mixed with the outlet flow from the fermenter of butyric acid in the first design.

Sammendrag

Esteren butylbutanat har egenskaper som gjør at det er et lovende supplement til flybensin og diesel. Esteren kan dannes ved en kondensasjonsreaksjon av butanol og smørsyre. Butanol og smørsyre kan igjen bli dannet ved fermentering. I denne oppgaven vil et reaktornettverket som omdanner sukker til butanol og smørsyre, for deretter å danne butylbutanat, bli optimalisert med hensyn på å maksimere konsentrasjonen av butylbutanat.

To forskjellige prosessdesign ble sammenliknet. I det første scenarioet skjedde fermenteringsreaksjonene separat. Produktstrømmene fra de to fermenteringstankene ble blandet og sendt til en reaktor hvor butylbutanat ble dannet. I det andre scenarioet ble butanol dannet ved fermentering. Produktstrømmen fra fermenteringstanken inneholdt både butanol og sukker. Denne strømmen ble sendt til en reaktor hvor smørsyre både ble dannet ved fermentering, og samtidig omdannet til butylbutanat ble ved den nevnte kondensasjonsreaksjonen.

Simuleringsresultatene viste at scenario 2 gir den høyeste konsentrasjonen av butylbutanat. Her ble den maksimale konsentrasjonen 16,42 g/L, mens i scenario 1 var den høyeste oppnåelige konsentrasjonen av butylbutanat 11,28 g/L. Mulige årsaker til at scenario 2 er et bedre prosessdesign er at konsentrasjonen av smørsyre alltid er lav i den kombinerte reaktoren. Dette medfører at gjærcellene ikke blir inhibert av produktet. En annen mulig årsak er at fødekonsentrasjonen av butanol til reaktoren hvor esteren dannes er høyere i det andre scenarioet. Dette er på grunn av at når produktstrømmene ut fra de to fermenteringstankene blandes i scenario 1, så medfører det at konsentrasjonen av butanol blir fortynnet.

Nomenclature

Symbol	Description	Unit
MW	Molar mass	g/mole
τ	Residence time	h
r_i	Rate of reaction for component i	g/(L h)
$\underline{r}$	Rate of reaction vector	g/(L h)
C_i	Concentration of component i	g/L
$\underline{C}$	Concentration vector/state space vector	g/L
V	Volumetric flowrate	m^3/s
μ	Cell growth rate	1/h
μ_{max}	Maximum cell growth rate, butanol model	1/h
K_S	Substrate affinity constant	g/L
$I_i(C_i)$	Inhibition function of component i	-
I_{tot}	Total inhibition function	-
$K_i(BuOH)$	Inhibition constant of butanol	g/L
$C_{HBu,max}$	Toxic concentration of butyric acid	g/L
m_{HBu}	Constant, exponent value in inhibiton term for HBu	-
q_S	Specific rate of substrate consumption	g$_S$/(g$_X$ h)
$Y_{i/S}$	Growth yield coefficient of component i on the substrate	g$_i$/g$_S$
k_{HBu}	Rate constant for consumption of butyric acid	L/(h g$_X$)
k_{BuOH}	Rate constant for production of butanol	(g$_{BuOH}$ L)/(h g$_X$ g$_{HBu}$)
K_{HBu}	Rate constant for consumption of butyric acid	g/L
μ_m	Maximum cell growth rate, butyric acid model	1/h
α_i	Cell growth-associated product formation	g/g
β_i	Cell non-growth-associated product formation	g/(g h)
m_S	Specific maintenance coefficient	1/h
K_I	Substrate inhibition constant	g/L
P_d	Product concentration associated with no cell growth	g/L
i	Degree of inhibition	-
Y_i	Stoichiometric yield coefficients on glucose	g$_i$/g$_S$

Symbol	Description	Unit
γ	Collection of constants	g_S/g_X
$K_{i(HBu)}$	Inhibition constant of HBu	mmol/L
$K_{m,i}$	Michaelis constant of component i	mmol/L
r_{max}	Maximum rate of esterification reaction	mmol/(L min g_{enzyme})
ω	Collection of parameters	g_S/g_{HBu}

Acronyms and Synonyms

Acronym	Meaning
AR	Attainable region
BuOH	Butanol
EtOH	Ethanol
HBu	Butyric acid
HAc	Acetic acid
BuB	Butyl butyrate
X_{BuOH}	Cell, butanol is the main product
X_{BuOH}	Cell, butyric acid is the main product
PFR	Plug flow reactor
CSTR	Continously stirred tank reactor
DSR	Differential sidestream reactor
VdV	Van de Vusse (kinetic system)
ODE	Ordinary differential equation
ABE fermentation	AcetoneButanolEthanol fermentation

List of Figures

List of Tables

Contents

1 Introduction

1.1 Introduction

Concerns about the environmental impact, as well as the decrease of accessibility of crude oil, have generated an interest in the production of petrochemical products from bio-based sources. Biorefineries have been pinpointed as the most promising way of switching from a fossil based economy to a bio-based economy [1, 2]. In a biorefinery, organic material is converted to fuels, heat, power, added-value bulk platform chemicals and speciality chemicals. In the public discussion, biofuels have gained particular interest. The Norwegian parliament has decided that fuel consumption shall be bio-based by 2020 [3]. This will lead to a growing market demand for biofuels. However, to ensure a satisfactory economic potential of a plant, speciality and platform chemicals play a vital role in the product portfolio of the biorefinery [4].

The fatty acid ester butyl butyrate (BuB) has proven to have properties that makes it excellent to use as an additive to jet engine fuel [5], as well as diesel [6]. Current areas of usage of BuB are as a component of pineapple flavor in the food and beverage industry. The building blocks of BuB are butyric acid (HBu) and butanol (BuOH), which are themselves important platform chemicals. These building blocks may react in an enzymatically catalyzed esterification reaction to form BuB [7, 8]. Both HBu [9] and BuOH [10] may be produced by anaerobic fermentation of sugars derived from lignocellulosic biomass. A production of BuB and its building blocks by the aforementioned methods will thereby address the transition from petroleum based to a sustainable energy supply.

The Norwegian University of Science and Technology collaborates with SINTEF Energy Research and Jadavapur University from India to make a proof-of-concept for an integrated biotechnological production process of BuB. In this biotechnological process, a challenge is to obtain satisfactory high concentrations of the desired fermentation products. This is due to that the bacteria is inhibited by the product when the concentration is above a certain threshold. Low concentrations impose high processing cost downstream, as the required separation becomes more difficult with lower concentration differences. A previous attempt to circumvent these problems for BuB production was to use a single bioreactor, where the fermentation products, HBu and BuOH, reacted instantaneously to BuB. The idea with this approach was to prevent inhibition of the bacteria by HBu and BuOH. However, this approach resulted in only 0,5% enrichment of kerosene with BuB. In contrast to this approach, EcoLodge aims at optimizing the production of each component prior to the esterification of HBu and BuOH to BuB. New process designs have been investigated in order to minimize side streams and waste and to maximize production of BuB. Earlier work on this project includes investigation of separation technologies, thereby in-situ removal of the fermentation products, as well as kinetic studies of the fermentation products [11].

This work will focus on optimization of the reactor network, without assessing separation technologies. Optimization of each subprocess, which is production of BuOH, production of HBu and the final production of BuB, will be done. Also, an optimization which integrates the whole reactor network will be performed. Two different process configurations will be evaluated to determine which layout yields the highest concentration of BuB.

Optimization of reactor networks is not a trivial problem since it is difficult to determine if the truly optimal solution has been found. Popular approaches for solving this problem has been heuristic rules, superstructure methods or attainable region (AR) analysis

Heuristic rules rely on previous engineering experience, as found in [12]. Methods like the Levenspiel plot is also counted as a heuristic rule, and this procedure may optimize the reactor structure given there is only a single reaction and the performance target is set [13].

Optimizing reactor networks by superstructure methods are a popular approach in the literature [14, 15, 4]. All possible combinations of reactors may be incorporated into the superstructure, and the solution of the optimization is, therefore, a subset of the superstructure [14]. As a result, the problem is generally computationally expensive. Also, the uniqueness of the solution is not assured and it is difficult to say if the solution is a local or a global optimum [16]. Despite these disadvantages, superstructure methods are powerful tools, as a large amount of detail and flexibility may be embedded into the structure. For complex systems with a large number of independent reactions, superstructure methods may be the only feasible procedure to optimize a reactor network [17].

In an attainable region analysis, the central task is to construct a region, in the state space, which describes all possible outcomes of the system [18]. For problems which may be reduced to three or lower dimensions, the attainable region approach yields an intuitive graphical output, which easily lets the user select the reactor structure which is optimally suited for his objective. As the problems in this work turn out to have a dimensionality lower than three, the attainable region analysis has been selected as the preferred method for reactor optimization.

1.2 Objective

The objective of this work is to find the limits of what a biotechnological process for butyl butyrate production can achieve, regarding the concentration of butyl butyrate. Two different process designs will be evaluated and compared by the maximum concentration of BuB. A sub-goal is to separately optimize the fermentation reactions which yield BuOH and HBu. These two components are valuable by themselves as well as that they are the required reactants to form BuB. Another sub-goal is to minimize the residence times in each reactor network, given that it does not compromise on the final concentration of the components referred to above. The optimization will be based on an analysis of the attainable region for the system.

2 Attainable Region for Reactor Networks

Optimization of continuous reactor networks are a challenging task due to the inherent non-linearity of reaction kinetics. The attainable region methodology is based on an idea, initially stated by Horn, as "that if one could find the attainable region for the system ... then the problem of the optimization was essentially solved" [18].[1].

Based on this foundation, AR theory sets out to answer the following question: given a set of chemical reactions and their kinetics, what are all the possible outlet states from the reactor network [16]? This region in the state space is termed the *attainable region*. In terms of a classical optimization problem, where an objective function is to be minimized or maximized subject to some constraints, the AR is effectively the constraint region which the solution has to satisfy.

All AR theory presented here is based on two important assumptions:

1. Constant denisty throughout the reactor network

2. Isothermal conditions

As the systems which will be investigated are biological systems in liquid phase, it is reasonable to believe that both these assumptions are valid. The state of a system is uniquely determined if pressure, temperature and concentrations of the system are specified [20]. For an AR analysis, the state space for the system is therefore described solely by the concentrations of the present components. In some cases, the residence time, τ, is also of interest. The residence time is defined as $\tau = \frac{V_{reac}}{Q}$, where V_{reac} is the reactor volume and Q is the volumetric flowrate through the reactor. The residence time gives implicitly the reactor volume, and is therefore an important parameter in, for example, cost estimation. For this reason, residence time will sometimes be treated as a pseudo concentration, hence, it can be incorporated to be a part of the state space.

It turns out that all the systems which will be investigated in this work can be represented in a two dimensional state space. In order to keep this section

[1] The original paper from Horn was proven to be difficult to get hold of, and has therefore not been read by the author. The citation is therefore from the paper which first demonstrated how to construct the AR. The original paper from Horn is in the aforementioned paper referenced to as [19]

brief and relevant, only theory relevant for two dimensional constructions will be presented here. It should be noted that AR theory is still applicable for higher dimensional theory, although the construction gets more complex. In the rest of this section some mathematical terms, which are frequently used in AR literature, will be defined. Afterwards, the reactor types which were used to construct a two dimensional AR will be presented. Construction and interpretation of the AR will be discussed in the final parts of this section.

2.1 Mathematical Definitions

In order to accurately describe certain features of the AR, some mathematical terms need to be clearly defined. These definitions are in terms with Feinberg, who pioneered the mathematical foundations of AR theory in n dimensions [21, 17, 22]. In the following, vectors are denoted by an underline under the variable name. A matrix is denoted by a double underline. Thus, in $\underline{\underline{A}}\,\underline{x}$, $\underline{\underline{A}}$ is a matrix and $\underline{x}$ is a vector.

2.1.1 Convexity and Extreme Points

Let V denote a real finite dimensional vector space and S be a set of vectors in V. S is *convex* if there for whatever vector $\underline{u}$ and $\underline{v}$ in S, the whole line segment connecting $\underline{u}$ and $\underline{v}$ is in S [23]. The *convex hull* of S, $conv(S)$, is the smallest set in V which is containing S. The *closure* of S, $cl(S)$, is the set containing all points in S together with its boundary. An *extreme point* of S, $\underline{x^*} \in S$, is a point which lies in S but not inside any straight line segment contained in S. Thus, the points which form the convex hull are in fact the extreme points of S, while all other points in S are denoted as *interior points* of S. A set, $H \subset V$ is a *hyperplane* if there is a non-zero vector $\underline{a} \in V$ and a number β such that $H = \{\underline{x} \in V : \underline{a} \cdot \underline{x} = \beta\}$. If a hyperplane that supports S at $\underline{x^*} \in S$ and it meets S only in $\underline{x^*}$, then $\underline{x^*}$ is an *exposed point*. Hence, every exposed point is an extreme point, but not all extreme points are exposed points.

Examples of these definitions are shown in Figure 1. Points A and D are extreme points in S, but not exposed since the supporting hyperplane meets the whole line segment connecting A and B, or the segment C and D. B and C are also extreme points, but since there are infinetly many hyperplanes supporting B and C, also they are not exposed points. All points on the interior of the arc AD are exposed points. The interior points of S are all points which are encapsulated by the figure, as well as all the points on the straight lines AB, BC and CD, excluding points A,B,C and D. By the dimension of a point $\underline{x^*}$ in S, it is meant the dimension of the linear subspace $T(S,\underline{x^*})$. In Figure 1, the dimension of the boundary is one at all points, except at point B and C. This is due to that there exists only one supporting hyperplane for each point, hence there is only one normal vector a for every point, and the dimension of the hyperplane is in this case one. On point B and C, there exist infinetly many hyperplanes, and the only vector which is orthogonal to all hyperplanes are the zero vector. Hence, in points B and C the dimension is zero [21].

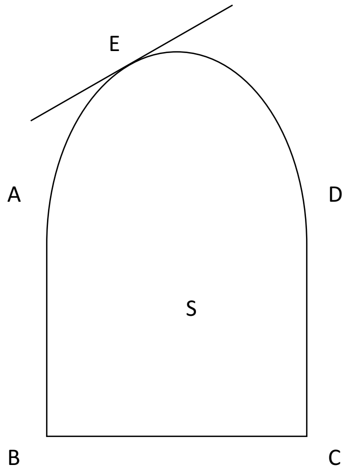


Figure 1: Illustration of some properties regarding convex sets. Points A, B, C and D are all extreme points, but not exposed since there exist several supporting hyperplanes (B and C) or the hyperplane touches several points in S (A and D). Point E is however an exposed point. The figure is a recreation of a figure from Feinberg [21]

2.1.2 Manifolds, Protrusions and Lineations

It is also required to describe the idea of a *manifold*. A manifold is a high dimensional generalization of a surface [23]. A manifold in n dimensions is a space which can cover any point in the neighbourhood with a n-dimensional patch. Since manifolds operate "locally", they ignore curvature. Hence, the surface of a sphere in $\mathbb{R}^3$ is therefore a two dimensional manifold, while a circle is a one dimensional manifold. For a more detailed mathematical definition, see elsewhere [24].

The idea of a manifold is vital for describing a *protrusion* and a *lineation*. As Feinberg states [21], a set P on the boundary of a closed convex set $S \subset V$ is an n-dimensional protrusion if

1. P is an n-dimensional manifold consisting entirely of extreme points on S
2. At each point in P, the dimension of S is n

On the other hand, a n-dimensional lineation, L, is a set on the boundary of the closed convex set S if

(i) L is a n-dimensional manifold and each point is interior to a line segment contained in L

(ii) At each point in L, the dimension of S is n

Thus, a lineation and a protrusion are complete opposite of each other in terms of extreme points. All points on a protrusion are extreme while no points in a lineation can be extreme. However, they are similar in the way that the dimension of the manifold is identical to the dimension of S in both cases.

2.1.3 Linear Subspaces and Linear Dependency

U is a linear subspace of $\mathbb{R}^n$ if it has the property that for all vectors $\underline{u_1}$ and $\underline{u_2}$ in U, and for all real numbers α_1 and α_2, the resultant vector $\alpha_1\underline{u_1} + \alpha_2\underline{u_2}$ also lies in U. The rank of U, also called its dimension, is the number of linearly independent vectors in U. If no vectors in the set can be written as a linear combination of the other vectors, then the set of vectors is linearly independent.

2.2 Allowable Operations in a Reactive System

In AR theory, two fundamental processes are thought to occur in any reactor network: mixing and reaction [18]. In order to describe these processes compactly, vector notation will be used to describe the system.

Let the system under investigation have n reacting species. As isothermal conditions and constant density are assumed, the state vector consists only of concentrations. Let $\underline{C}_{all} \in \mathbb{R}^n$ be the state vector, consisting of the concentrations of all the components present. Note that the residence time may be treated as a component in $\underline{C}_{all}$ if it is a feature of interest. The *phase vector* contains in principle exactly the same information, but only the minimal amount of information which is required to uniquely determine the state of the system is included in this vector. For a chemical reactive system, the dimension of the phase vector is equal to the number of independent reactions [16]. If residence time is a feature of interest, the dimensionality increases by one. The unique state of the system can then from the phase vector be calculated by the appropiate mole or mass balances. This allows the AR to reside in the phase space, hence, it reduces the dimensionality of the problem while the uniqueness of the solution is intact [16]. The phase vector incorporates therefore the concentrations of the *independent* components. This vector will be denoted by $\underline{C}$ in this work.

2.2.1 Mixing as a Fundamental Process

Mixing is an important process in any chemical reactor network. For example, mixing of effluent streams from different reactors result in a new stream with a different composition, which may be the input to yet another reactor. Mixing occurs also inside reactors. As an example, in an ideal continuous stirred tank reactor (CSTR), it is assumed that full mixing occurs. This is equivalent to that the concentration is homogenous everywhere in the reactor. This reactor type will be more described in section 2.3.2.

If mixing between two streams with concentration $\underline{C}_1$ and $\underline{C}_2$ occurs, the resultant stream, $\underline{C}^*$, will be a linear combination of the two streams. This means that $\underline{C}^*$, will be on the straight line between points 1 and 2 in a state space diagram, and that it obeys what is termed the lever arm rule [18]. The lever arm rule is a consequence of the conservation of moles and that there are no volume changes in the system. The lever arm rule is also valid for when residence time is a component in the concentration vector, see Appendix A for the proof. The resultant concentration after mixing is given by Equation (2.1)

$$\underline{C}^* = \lambda \underline{C}_1 + (1-\lambda)\underline{C}_2, \qquad \lambda \in [0,1] \tag{2.1}$$

Here, λ is the ratio of the volumetric flows, $\lambda = \frac{Q_1}{Q_1+Q_2}$.

A mixing process for a two dimensional state vector is visualized in Figure 2. Here, the state vector consists of two concentrations, $\underline{C}_i = [x_i, y_i]$. As it can be observed, the resultant concentration is on the straight line between $\underline{C}_1$ and $\underline{C}_2$. The mixing line also fills out the concave parts of the AR, hence, mixing can expand the AR and make it a convex figure. This is a highly useful property of mixing. The main reason is that obtaining the global minimum of an optimization problem is ensured if a local minimum exist, given that the constraint region is convex [25]. It also follows that all the interior points in the AR can be reached by mixing different streams with concentrations which reside on the AR boundary.

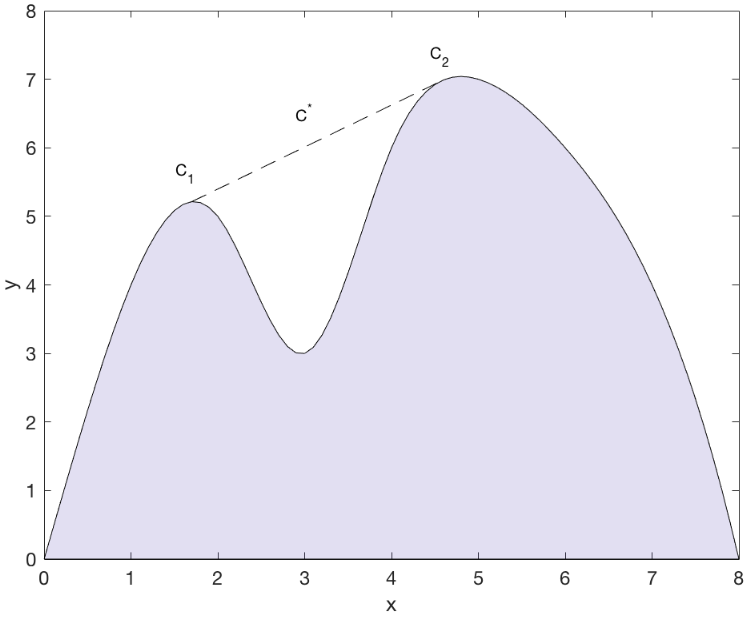


Figure 2: Mixing between two flows with different composition. The resultant concentration lies one the straight line between the two points. The shaded area is the current AR. Mixing fills out the concave holes, and thus expands the AR

2.2.2 Chemical Reactions as a Fundamental Process

The other allowable operation in the reactor network is, of course, chemical reactions. When the kinetics is known for the system, it is possible to compute the rate vector for any point in the state space. The rate vector for all components in a system is defined by Equation (2.2).

$$\underline{r}_{all}(\underline{C}_{all}) = \frac{d\underline{C}_{all}}{d\tau} = \left[\frac{dC_1}{d\tau}, \frac{dC_2}{d\tau}, .. \frac{dC_n}{d\tau}\right]^T \tag{2.2}$$

Note that if the residence time is incorporated into the state vector, then the reaction rate for τ is given as $\frac{d\tau}{d\tau} = 1$.

From how the rate vector was defined in Equation (2.2), it is clear that it describes the direction of change in the state space. When it is multiplied by a time step, a vector in the state space can be visualized. It is therefore possible to draw a rate field for the system.

In order to visualize a concrete example of a rate field, and other properties of AR theory later on, the Van de Vusse (VdV) system [26] will be used as an example. This is a reacting system which has been extensively researched and frequently used as an example in AR theory [18, 27, 21, 28, 17, 22, 16]. The chemical reactions in the system are given by Equation (2.3).

$$\begin{gathered} \mathrm{A} \xrightarrow{\mathrm{k_1}} \mathrm{B} \xrightarrow{\mathrm{k_2}} \mathrm{C} \\ 2\mathrm{A} \xrightarrow{\mathrm{k_3}} \mathrm{D} \end{gathered} \tag{2.3}$$

The reaction rates are as if the reactions were elementary reactions. Hence, the rate vector is given by Equation (2.4).

$$\underline{r} = \begin{bmatrix} r_A \\ r_B \\ r_C \\ r_D \end{bmatrix} = \begin{bmatrix} -k_1 C_A - k_3 C_A^2 \\ k_1 C_A - k_2 C_B \\ k_2 C_B \\ k_3 C_A^2 \end{bmatrix} \tag{2.4}$$

The initial concentrations were for all the simulations $\underline{C}_f = \left[C_{f,A}, C_{f,B}, C_{f,C}, C_{f,D}\right]^T = [1,0,0,0]^T$ mol/L. The kinetics parameters used are given in Table 1.

Table 1: Kinetics parameters used in the Van de Vusse kinetics

Symbol	Value
k_1	1 s^{-1}
k_2	1 s^{-1}
k_3	10 L/(mol s)

As there are three independent reactions in the system, the full AR can be represented in $\mathbb{R}^3$. The full AR is then usually given in $C_A - C_B - C_D$ space, where the concentration of C may be computed by a mass balance. However, all the rate of formations in Equation (2.4) can be computed based on only C_A and C_B. By assuming that B is the only valuable component, and no interest is taken in component C or D, no information is lost if the AR is made in $C_A - C_B$-space. In order to facilitate a relevant discussion of ARs in two dimensions, *this assumption will be done throughout this section*.

By meshing points in $C_A - C_B$-space, the rate field for the VdV kinetics can be computed. This is shown in Figure 3.

2.3 The Fundamental Reactor Types and Their Geometrical Properties

There exist a variety of different chemical reactors, and each of them behave differently in a rate field. However, Feinberg showed in a series of papers that only three fundamental reactor types are required to construct the AR in any dimension [21, 17, 22]. The fundamental reactors are plug flow reactors (PFRs), continously stirred tank reactors (CSTRs) and differential sidestream reactors (DSRs). For a two dimensional system, only the PFR and a CSTR are required and they were the only reactors used in this work. A brief introduction of the most interesting features of the PFR and CSTR will be given in this section.

2.3.1 PFR

The PFR is a tubular reactor where it is assumed that no mixing occurs in the radial direction [29]. The design equation for a PFR is given by Equation (2.5).

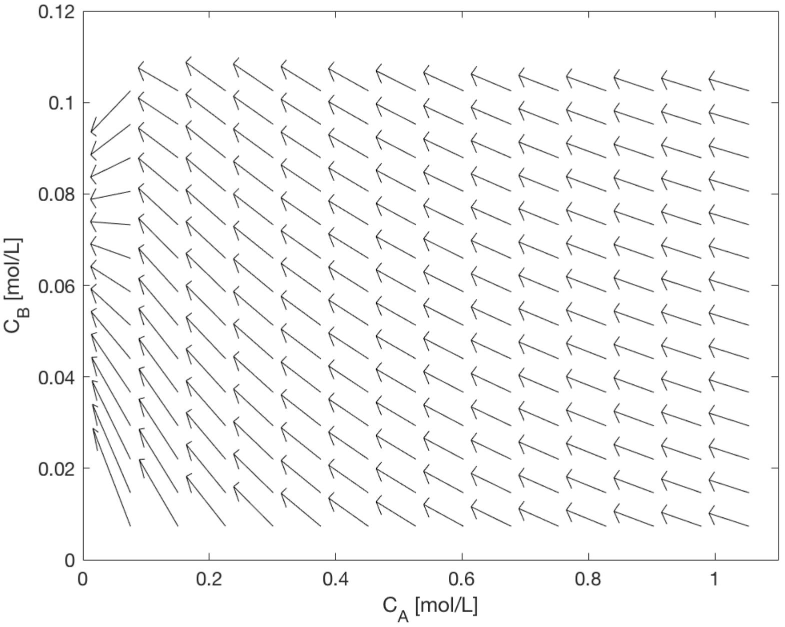


Figure 3: Example of a rate field. The rate field represents the Van de Vusse system in $C_A - C_B$ space.

It is a system of ODEs, and given the appropiate initial conditions, this system is readily solvable. When the solution of Equation (2.5) is plotted in the state space, it is referred to as the PFR solution *trajectory*, or just the PFR trajectory.

$$\frac{d\underline{C}}{d\tau} = \underline{r}(\underline{C}), \qquad \underline{C}(\tau = 0) = \underline{C}_f \tag{2.5}$$

As it can be observed from Equation (2.5), the rate vector evaluated at point $\underline{C}$ is always tangent to the PFR trajectory. Furthermore, from well known theory about the solution of an ODE, every solution is unique for a given set of initial conditions [30]. Also the rate field is unique for the given kinetics, which implies that PFR trajectories which are initiated from different points in space cannot cross each other [18]. All these features are important properties for the PFR.

As the PFR is a vital part in any AR construction, it is important to get a physical interpretation of its trajectory. The PFR trajectory shows how the composition of the stream changes as it flows through the reactor. This means that the PFR trajectory shows a continuum of solutions for *one* reactor. This physical interpretation of the trajectory is a distinct difference from the solution of the CSTR design equation, which in fact describes a range of differently sized CSTRs. This will be more discussed in the next section.

2.3.2 CSTR

The CSTR is conceptually a tank reactor where the content is stirred to such an extent that the concentration in the tank is homogenous [29]. The design equation for a CSTR is given by Equation (2.6). To obtain the outlet concentration, it is generally required to solve a system of algebraic non-linear equations. This is due to the complex form that the rate expressions might have. Depending on the kinetics, there can be multiple solutions for the same residence time, which implies that one of the (currently undiscovered) solutions may expand the current AR significantly.

$$\underline{C} - \underline{C}_f = \tau \underline{r}(\underline{C}) \tag{2.6}$$

From a geometric point of view, it is observed that the rate vector at point $\underline{C}$ is co-linear with the mixing vector from point $\underline{C}$ to the feed, $\underline{C} - \underline{C}_f$. The solution of the CSTR equation, when plotted in either concentration or residence time space, is called the CSTR *locus* [18].

As the CSTR locus is the solution of Equation (2.6), it should be physically interpreted as a range of CSTRs with different residence times. That is, given the same composition and flowrate in the inlet, the CSTR locus corresponds to physically different CSTRs with different reactor volumes. This is in contrast to a PFR, where the solution trajectory describes the change in composition throughout one reactor.

2.3.3 Geometry of the Fundamental Reactor Types

The geometric features of a PFR and a CSTR are readily visualized for the VdV system described in section 2.2.2. The design equations for a PFR and CSTR are the basis for making a state space diagram. From these equations, concentration of all components as a function of the residence time can be made. By pairing $C_A - C_B$-pairs which have the same residence time, a state space diagram can be made. The algorithm to make an AR diagram will be explained more thoroughly in section 2.6. Here, only the final result of the AR will be presented. This is done so that it is possible to envision the geometric properties of a PFR and CSTR, as these properties are important when proving the uniqueness of the AR. The full AR for the VdV system, with reaction vectors indicated on the PFR trajectory and CSTR locus, is shown in Figure 4.

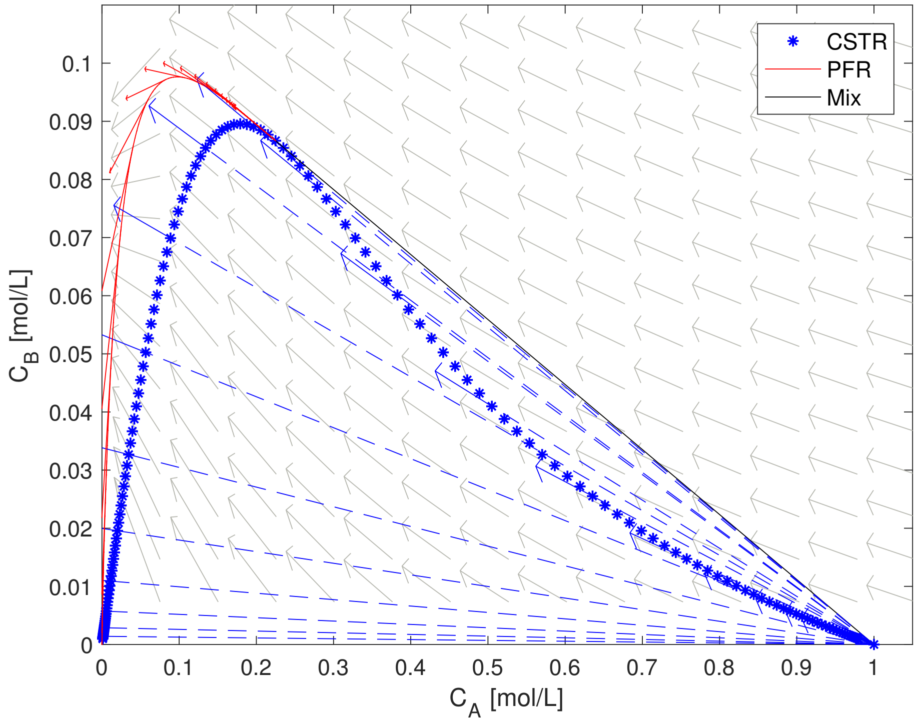


Figure 4: Geometrical properties of the PFR and CSTR. The CSTR locus is shown by the scattered blue line, while mixing lines with the feed to various concentrations from the outlet of the CSTR is shown with the dashed blue lines. Reaction rate vectors at the selected outlet CSTR concentrations are co-linear with the mixing vectors. The red line shows the PFR trajectory, initiated from a suitable CSTR outlet. Here, the reaction rate vectors are tangent to the PFR trajectory. This is the full AR for the VdV system.

The scattered blue line represents $C_A - C_B$-pairs obtained by solving the CSTR design equation for a range of different residence times. The feed to the reactor had a concentration of 1 mol/L of pure A. The dashed blue line represents mixing from the feed with various outlet CSTR concentrations. From these outlet concentrations of the CSTR, the reaction rate vector at that point is indicated by a blue arrow. Note that the dashed blue mixing line and the reaction rate arrow is co-linear. This is expected from the CSTR design equation (2.6).

The solid red line shows $C_A - C_B$-paris obtained from the PFR design equation for various residence times. This reactor is *not* initiated from the feed, but from the outlet of a suitable CSTR. The process of selecting a suitable CSTR will be explained in section 2.6. It should be observed that the red arrows, symbolizing the reaction rate vectors, are tangent to the PFR trajectory. Also this is expected by inspection of Equation (2.5).

2.4 Necessary Conditions for the AR

Based on the geometric properties of the PFR, CSTR and the mixing process, it can be argued that in certain cases it is possible to enlarge the AR. Thus, some necessary conditions need to be fulfilled if a defined region in the state space is to be called the attainable region. These necessary conditions for the AR can be listed as follows [18, 16]

(a) The AR must be convex

(b) The AR must contain the feed point

(c) Reaction vectors cannot point out of the AR

(d) The AR is unique and it exists as a single region

(e) The AR resides in a subspace of $\mathbb{R}^n$

These necessary conditions can easily be justified. If there are concave holes in the AR, then mixing between two suitable extreme points from the AR can fill these holes. To see how mixing can make a region convex, refer to Figure 2. Hence, condition (a) must be true.

By definition, the feed is achievable and it must therefore be contained in the AR. This makes (b) valid.

To prove condition (c), suppose that a rate vector points out of the boundary of the AR on some point P. If this is the case, then a PFR initiated from P can "follow" the rate vector, and therefore expand the AR. Hence, all rate vectors on the boundary of the AR must point into the region or be tangent to the boundary [18].

To justify condition (d), assume first that it is not true. Then, for a given feed point, multiple candidate ARs can be made. However, these regions must be subregions of the true AR, as mixing between the different regions can connect them. Hence, condition (d) must be true [16].

The AR describes by its definition all the attainable concentrations of a system with n components. The AR must therefore reside in $\mathbb{R}^n$. As the concentrations must respect a non-negativity constraint, the AR is bounded in the subspace where the concentrations take values in the range $[0, \infty)$. However, this is an unnecessary loose bound on the AR. Given a feed point and a reaction stoichiometry, the range of possible concentration values are bounded by mass balance. As an example, consider the generic reaction A $\longrightarrow$ B, initiated with a feed of 1 mol/L of pure A. The concentration of both A and B may only take values in the range [0, 1] mol/L. Hence, the AR is finite and exists in a subspace in $\mathbb{R}^n$ [16]. This proves condition (e).

2.5 Reaction Surfaces and Mixing Surfaces

The extreme points which defines the boundary of the AR encapsulate all possible states the system can reach. The boundary of the AR is therefore of special interest when searching for the optimal reactor structure. A central task of AR theory is thus to describe how the states which reside on the boundary of the AR have been generated.

Specifically, it is of interest to explore which points or surfaces on the AR boundary that are a result of a mixing process and which ones are generated by reaction. Conceptually, the AR is the set of points in the state space which are reachable. Furthermore, the AR is always a convex set of states if a linear mixing law is assumed, as explained in section 2.2. All interior points can therefore be reached by mixing the states which form the extreme points of the AR boundary. This can be seen if the AR is made up of the extreme points A, B, C, D and the arc AD in Figure 1; all interior points can be reached by mixing these extreme points in various quantities. Regarding the extreme points themselves, by definition, they cannot be on the straight line segment contained in the set of points which defines the AR. This means that they cannot be expressed as a linear combination of other points in the AR. Thus, it can be deduced that the extreme points cannot be a result of a mixing process. As mixing and reaction are the only two operations which are allowed, it means that all extreme points in the AR are either feed points or they are generated by reaction [16].

It has been showed by Feinberg that all the extreme points in the AR can be reached by a combination of PFRs and CSTRs by exploiting properties of what is termed the *complement principle* [21]. For AR constructions with a dimensionality greater than two, also a DSR is required to obtain all the extreme points on the AR [17]. In the following part of this section, the complement principle will be briefly explained. Then, the role the PFR and CSTR play on the AR boundary will be highlighted.

2.5.1 The Complement Principle

The complement principle plays an important role in the theorems which states that all the extreme points on the AR can be reached by the means of PFRs and CSTRs. For any production unit where only reaction and mixing may occur, the complement principle places a constraint on the set of composition which may be present in the production unit. That is, the principle relates the mole balances over a region in a reactor with the underlying kinetics and feed. Here, only the principle itself and a short overview over the argumentation will be given. For the complete picture, see elsewhere [21].

To describe the complement principle, some definitions need to be in place. As before, $\underline{C}_{all}$ is the concentration vector which resides in $\mathbb{R}^n$. The set of vectors which resides in $\mathbb{R}^n$ and are strictly positive are denoted $\mathbb{P}^n$. If the concentration of one or more of the species is zero, then the concentration vector is residing in $\overline{\mathbb{P}}^n$. For such a system, the complement priciple is stated as [21]

The Complement Principle. *Let $\underline{r}_{all}$ be a (continously differentiable) species formation rate function, and let $\mathscr{F} \subset \overline{\mathbb{P}}^n$ be a set of feed compositions. Furthermore, let $\mathscr{C}$ be a set in $\overline{\mathbb{P}}^n$ that contains $\mathscr{F}$. In order for $\mathscr{C}$ to be the set of realized compositions in a steady state production unit involving only reaction and mixing* *(and with feed streams having composition in $\mathscr{F}$), it is necessary that the following condition must be satisfied: For every relatively open subset $\mathscr{D} \subset \mathscr{C}$ that contains an extreme point of conv($\mathscr{C}$) but no element of $\mathscr{F}$, there are vectors*

$$\underline{C}_{all,out} \in cl\ conv(\mathscr{D}) \tag{2.7a}$$

$$\underline{C}_{all,in} \in cl\ conv(\mathscr{C} \backslash \mathscr{D}) \tag{2.7b}$$

$$\underline{r}_{all} \in cl\ conv\{\underline{r}_{all}(\underline{C}_{all}) \in \mathbb{R}^n : \underline{C}_{all} \in \mathscr{D}\} \tag{2.7c}$$

and a positive number τ such that

$$\underline{C}_{all,out} - \underline{C}_{all,in} = \tau \underline{r}_{all} \tag{2.7d}$$

The argumentation for the principle is mainly based on the idea of a physical control region in the reactor, which only consists of the regions in the reactor where the compositions in $\mathscr{D}$ is present and it does not consists of the feed. The inflow to this region must necessarily have compositions which belong to the complement of $\mathscr{D}$ in $\mathscr{C}$, denoted by $\mathscr{C} \backslash \mathscr{D}$, and outlet compositions which resides in $\mathscr{D}$. As the compositions within the control region reside entirely in $\mathscr{D}$, the rate of formation is constrained on the form of Equation (2.7c). A steady state mole balance over the control region then gives Equation (2.7d). For a full description of the complement principle, see elsewhere [21].

2.5.2 PFRs Obtain the Extreme Points on the AR Boundary

When visualizing the AR for the VdV system in Figure 4, it can be observed that all the points which are on the AR boundary is either on a mixing line or on the PFR trajectory. The mixing line goes from the feed (1 mol/L of pure A) to the outlet of what was termed a "suitable" CSTR in section 2.3.3. This means that almost all of the extreme points of the AR for the VdV system is generated by the means of a PFR. This is actually a general result in AR theory, and it is stated in the following theorem from Feinberg [21]. Here, the union of plug flow trajectories is a feature of AR constructions in higher dimensionality systems. For a two dimensional case, the union of PFR is actually only a single PFR.

Theorem 2.1. *Suppose that $\mathscr{F} \subset \mathbb{R}^n$ is a specified (compact) feed set and that $\underline{r}_{all}(\underline{C}_{all})$ is a specified (continuously differentiable) species formation rate function defined on $\mathbb{R}^n$. Moreover, suppose that $\mathscr{C}_{all}$ is a compact convex set in $\mathbb{R}^n$ that complies with the complement principle and has the property that nowhere on the boundary of $\mathscr{C}$ does $\underline{r}_{all}(\underline{C}_{all})$ points outward. Then any protrusion in $\mathscr{C}$ disjoint from $\mathscr{F}$ is the union of plug-flow reactor trajectory segments- that is, of trajectory segments for the differential equation*

$$\frac{d\underline{C}_{all}}{dt} = \underline{r}_{all}(\underline{C}_{all})$$

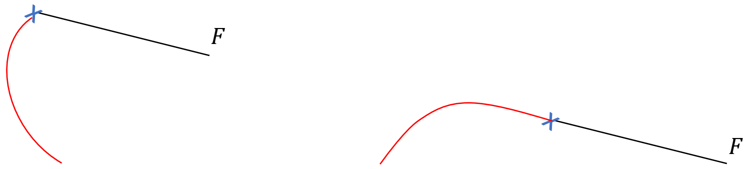
The proof of this theorem is based on that an exposed point in $\mathscr{C}$, say $\underline{C}^*_{all}$, has a rate vector, $\underline{r}_{all}(\underline{C}^*_{all})$, which is required by the complement principle to be tangent to $\mathscr{C}$ at $\underline{C}^*_{all}$. As the feed point is not included in the set $\mathscr{C}$, the requirement of $\underline{C}^*_{all}$ to be an exposed point on the AR, can be relaxed to that $\underline{C}^*_{all}$ can be an extreme point on the AR. It can then be demonstrated that a protrusion in $\mathscr{C}$ is a manifold with the property that at all of its points, the rate vector is tangent to the manifold. This fact coincides with the property of a union of PFRs. For the detailed proof, see elsewhere [21].

The practical essence of Theorem 2.1 is that all exposed points on the AR can be reached with a PFR. More precisely, the AR is composed entirely of mixing surfaces and manifolds of PFR trajectories [16]. This implies that generally, all reactor networks should optimally terminate with a PFR [21, 16].

2.5.3 The CSTR as a Connector to the AR Boundary

As the AR is built up of mixing surfaces and a PFR manifold, a central task is to obtain the initial conditions for this PFR trajectory. This initial condition has itself to reside on the AR boundary. For the VdV kinetics, Figure 4 shows that the optimal PFR is initiated from a CSTR. It turns out that this is a general result for two dimensional AR constructions.

The state which initiate the PFR on the manifold inhibit some special geometrical properties. This state is termed a *connector*, $\mathscr{K}$. These geometrical properties are in a large degree dependent on how a lineation, $\mathscr{L}$, meets a protrusion, $\mathscr{P}$. For the set of points to be called a conncetor, it is required that the protru-



(a) Example of a rough joining of a lineation and a protrusion. Notice the change in gradient along at the point of intersection.

(b) Smooth joining of a protrusion and lineation. There is no change in gradient at the point of intersection. The blue cross is therefore a connector.

Figure 5: Examples of a rough and smooth joining of a lineation and protrusion. The black line is a lineation, or a mixing surface in AR terminology. The red line is a protrusion, corresponding to a PFR trajectory, and the blue cross is a connector, or a specialized CSTR solution.

sion meets the lineation *smoothly*. For higher dimensional AR theory, Theorem 2.1 states that the union of PFRs form a protrusion. These PFRs need different initial conditions. The connector is then a set of states made up of outlet concentrations from specialized CSTRs *and* DSRs. The conncetor will therefore be a line in the state space. However, the role of DSRs will be omitted here, as a CSTR is all that is required in two dimensions. The conncetor will therefore only be a point in the state space. For the interested reader, see [21, 17, 22].

The idea of a lineation meeting a protrusion smoothly will be only described qualitatively in this text. For the full mathematical definition, see elsewhere [21].

Figure 5a and 5b show two examples of a lineation meeting a manifold. A PFR manifold is given by the red line, and a lineation caused by mixing point F with the blue cross is given by the black line. Hence, there are three different sets in Figure 5; the PFR manifold, the lineation *and* the blue cross.

In Figure 5a, there is a distinct change in gradient where the lineation meets the PFR manifold. It is thus said that the lineation meets the manifold *roughly*. This blue cross is a manifold of dimension zero, and it has also a dimension of zero by empowering definitions in 2.1. Hence, this point is itself a protrusion, and according to Theorem 2.1, it should be a PFR manifold. This is clearly impossible, as there is a unique solution for any ODE, given a specified feed point. Hence, the blue cross cannot be a feed point for the red PFR manifold, as well as a PFR manifold itself.

In Figure 5b, there is another situation. The blue cross "fuses" the black lineation and the red protrusion smoothly. That is, no distinct change in gradient is observed. When this is the case, it is understood that the lineation meets the manifold *smoothly*. The blue cross is still a manifold of dimension zero, but now the dimensionality of the point is one, as there are exactly one hyperplane supporting the point. The point is therefore *not* a protrusion, and Theorem 2.1 does not constrain this point to be a PFR trajectory. The qualitative criteria of smoothness can be translated to that the reaction rate vector at the point of intersection need to be co-linear with the lineation, which is a mixing vector. This criterion fits the geometrical property of a CSTR, as described in section 2.3.2. Hence, the CSTR act as a *connector* on the AR boundary, where it can initiate PFR manifolds. As these PFR manifolds are actually protrusions, and thereby by definition consisting of only extreme points, it is ensured that CSTRs and PFRs are the *only* reactor types which are required to construct the AR in two dimensions[2]. The CSTR which is operating on the boundary is called a *critcal CSTR* in the AR literature [22].

2.6 Construction of the AR

One of the main advantages of using AR for reactor optimization, is that it is easy to interpret the AR in terms of reactor configuration and design. For two dimensional systems, also the construction of the AR is relatively straightforward. As an example of how to construct and then interpret the AR in concentration space, the VdV kinetics described in section 2.2.2 will be used.

The first step when constructing the AR is to solve the CSTR and PFR design equations for a specified feed. In this case, a feed of 1 mol/L of pure A was used. The solutions yield a trajectory of concentrations of A and B as a function of the residence time. This is shown in Figure 6. As it can be observed, the CSTR uses a long time to approach equilibrium values. In order to approximate equilibrium, the residence time for the CSTR was set to 1000 h. However, to better visualize the dynamics of the system, a residence time of 20 h was chosen in Figure 6.

[2] For higher order dimensions, also a DSR is required. See [21] for more information.

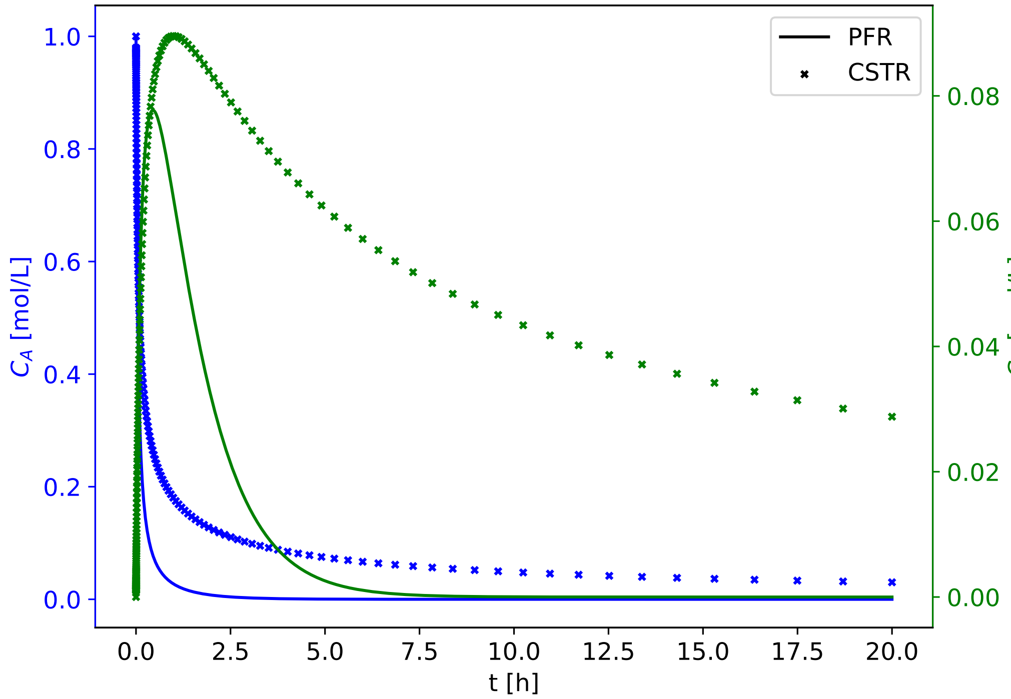


Figure 6: Solution trajectories of the PFR and CSTR design equation for the VdV kinetics.

The state space diagram can be made by pairing the $C_A - C_B$-pairs which have the same residence time. A plot of C_A as a function of C_B for the CSTR equation gives the scattered blue line in Figure 7. The solution of a PFR initated from the feed is shown as the dashed red line. As it can be observed, the CSTR locus is "outside" the area encapsulated by the dashed red line in a large part of the figure. Theorem 2.1, and the property of a CSTR as a connector, tells that somewhere on the CSTR locus, a PFR can be initiated which will form the extreme points on the AR boundary. Furthermore, there are concave holes in the CSTR locus. From section 2.2.1, it is known that the concave holes can be "filled" with mixing the feed with a point on the CSTR locus. A mixing line which makes the CSTR locus convex is shown as the solid black line in Figure 7. From section 2.5.3, it is known that the CSTR concentration which forms the end points of the mixing line, together with the feed, is a connector on the AR boundary. A PFR initiated from this outlet CSTR concentration must therefore reside on the AR boundary, according to Theorem 2.1.

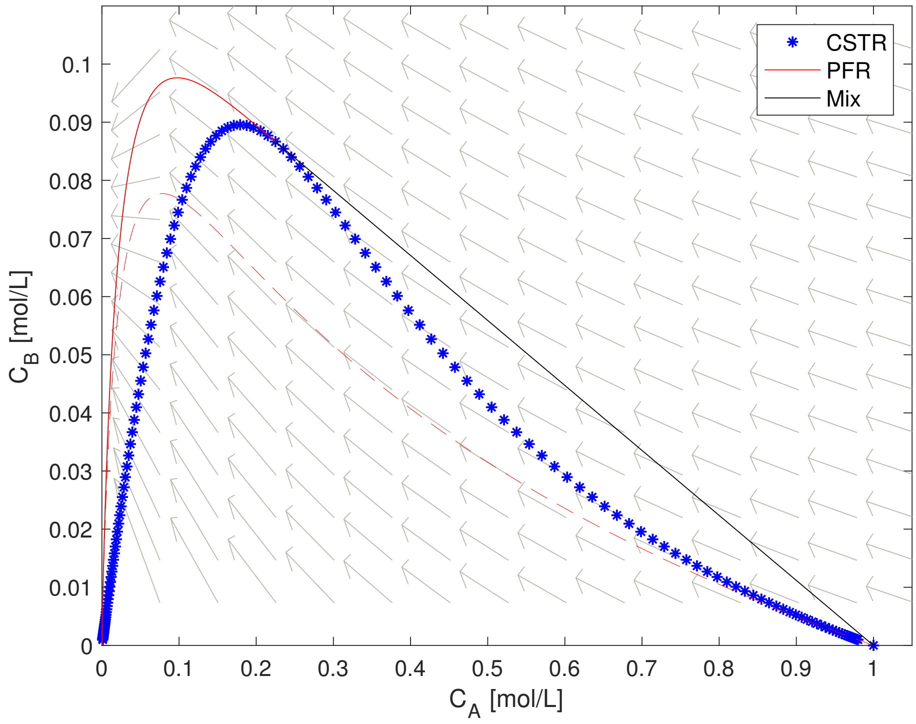


Figure 7: Solutions of the VdV kinetics in the $C_A - C_B$-space for a variety of reactors. The blue crosses are the CSTR locus, the black line is the mixing line from the feed to the optimal CSTR outlet and the solid red line is the PFR trajectory with feed from the outlet of the optimal CSTR. The dashed red line shows the PFR trajectory which is initiated from the feed point to the reactor network.

The complete AR for the VdV system, with only the reactor configurations which are required to operate on the AR boundary, is given in Figure 8. From the figure, it can be observed that the outlet concentrations from the critical CSTR are $C_A^{crit} = 0,225$ mol/L and $C_B^{crit} = 0,087$ mol/L. By comparing these values with Figure 6, then it can be observed that a critical CSTR for the VdV system has a residence time of 0,672 h.

By examining Figure 8, it is clear that this region fullfills all the necessary conditions in section 2.4. It is convex, the feed region is being contained within the AR, and all rate vectors on the boundary point either into the region or they are tangent to the boundary.

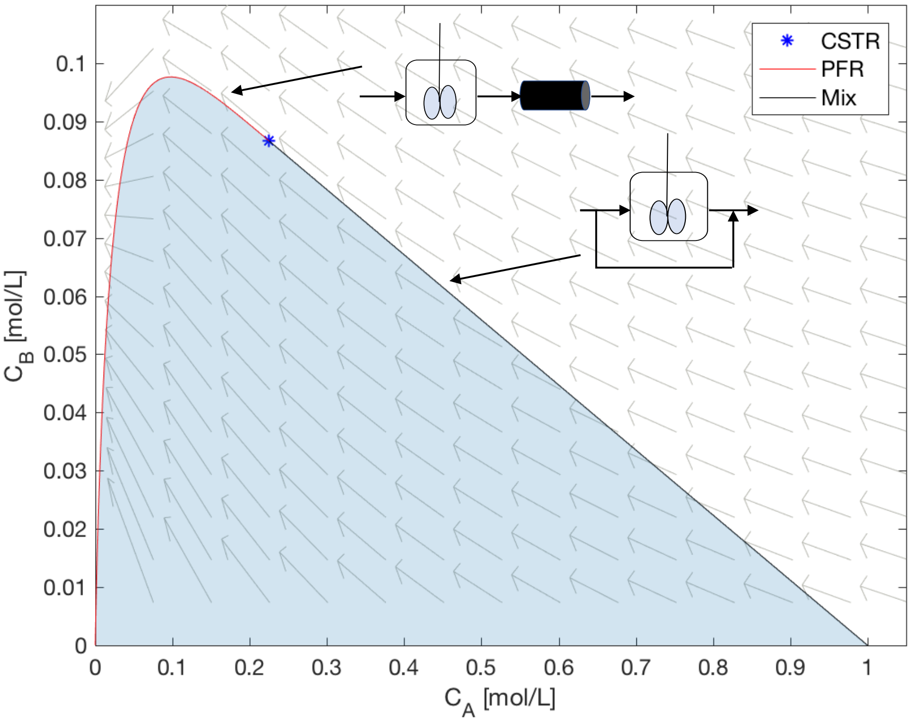


Figure 8: The attainable region for the two dimensional VdV kinetics. The blue crosses are the CSTR locus, the black line is the mixing line from the feed to the optimal CSTR outlet and the solid red line is the PFR trajectory with feed from the outlet of the optimal CSTR. The dashed red line shows the PFR trajectory with feed from the feed point to the reactor network.

3 Process Description for Butyl Butyrate Production

The esterification of BuOH and HBu gives BuB as a product. BuOH and HBu are in turn separately produced in fermentation reactors where the feed is sugar derived by lignocellulose. In order to make an AR analysis feasible for the whole process, kinetic models for all these three reactive systems need to be described. Two different scenarios for the total production will be investigated so that it may be possible to identify the most promising process configuration.

First, a configuration with separate production of BuOH and HBu by fermentation, followed by their esterification to BuB will be described. A separate reactor network for all three process will therefore be the output of this process configuration. This will be referred to as scenario 1. In scenario 2, there are two separate reactive systems: one is producing BuOH, while the other one is a combined production of HBu and BuB. The BuOH stream is fed to the combined production reactor, together with an additional stream of fresh substrate. The expected advantage of this process configuration is that the cells in the combined reactor are not inhibited by HBu, since it is simultaneously being consumed to produce BuB. Thus, scenario 1 has a strong focus on optimization of sub-processes, while scenario 2 has a higher degree of integration in the process design.

Both these process configurations will be described in more detail in sections 3.1 and 3.2. In the following, cell concentration (equivalent to bacteria) and substrate will be denoted by X and S, respectively. The molar mass of component i will be denoted by MW_i. In addition, the two fermentation models have many similarites. Many of the variables are termed the same, as they describe the same phenomenon but for different systems. The rate parameters are therefore valid only inside each subsection, as some parameters are named the same in section 3.1.1 and 3.1.2.

3.1 Scenario 1

In scenario 1, there are separate reactor networks producing BuOH and HBu, with a following esterification of these two intermediate streams to BuB. The process flow diagram is shown in Figure 9.

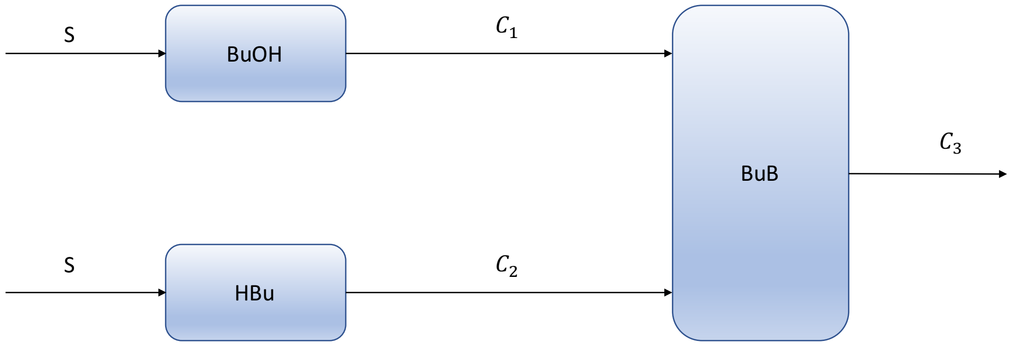


Figure 9: Process flow diagram for scenario 1. Separate fermentation reactions produce BuOH and HBu, while in a consecutive reaction network these two components undergo an esterification reaction to form BuB.

The kinetics for the production of BuOH, HBu and BuB are described in detail in sections 3.1.1, 3.1.2 and 3.1.3.

3.1.1 Production of Butanol

The kinetic equations for the butanol production in scenario 1 resembles the kinetics described by Votruba structurally [31]. However, some of the parameters have been experimentally determined in earlier parts of the EcoLodge project. The employed inhibition terms, with their corresponding parameters, are gathered from the literature [10]. The process is an acetonebutanolethanol (ABE) fermentation, and the bacteria used in the aforementioned papers were *Clostridium acetobutylicum ATCC 824*.

The present components in this system are cell mass, substrate, butyric acid, butanol and ethanol. The kinetic expression for the growth of the biomass is given by Equation (3.1).

$$r_X = \mu C_X \tag{3.1}$$

Here, μ is the specific cell growth rate and it is given by Equation (3.2).

$$\mu = \frac{\mu_{max} C_S}{K_S + C_S} I_{tot}(C_S, C_{BuOH}, C_{HBu}) \tag{3.2}$$

K_S is the substrate affinity constant. The term $I_{tot}(C_S, C_{BuOH}, C_{HBu})$ indicates the total inhibition of the cell mass due to too high values of C_S, C_{BuOH} or C_{HBu}. The inhibition terms may be written as in Equation (3.3) [10]. For the inhibition of HBu, Equation (3.5) with its corresponding parameters were obtained from [10], while the equation and parameters for I_{BuOH} were determined experimentally by co-supervisor. Due to lack of data, inhibition due to glucose was neglected in this work.

$$I_{tot}(C_S, C_{BuOH}, C_{HBu}) = I_{BuOH}(C_{BuOH}) I_{HBu}(C_{HBu}) I_S(C_S) \tag{3.3}$$

$$I_{BuOH}(C_{BuOH}) = \frac{K_{i(BuOH)}}{K_{i(BuOH)} + C_{BuOH}} \tag{3.4}$$

$$I_{HBu}(C_{HBu}) = 1 - \left(\frac{C_{HBu}}{C_{HBu,max}}\right)^{m_{HBu}} \tag{3.5}$$

$$I_S(C_S) = 1 \tag{3.6}$$

Here, $K_{i(BuOH)}$ is an inhibition constant. In Equation (3.5), $C_{HBu,max}$ is the toxic concentration of HBu to the cell, while m_{HBu} is a constant.

The rate of change of the substrate is given by Equation (3.7).

$$\begin{aligned} r_S &= q_S C_X, \qquad \text{where } q_S = \frac{\mu}{Y_{X/S}} \\ r_S &= \frac{\mu C_X}{Y_{X/S}} = \frac{1}{Y_{X/S}} r_X \end{aligned} \tag{3.7}$$

Here, q_S is the specific rate of substrate consumption and $Y_{X/S}$ is the cell growth yield coefficient on the substrate. Butyric acid is an intermediate in the system: it is synthesized by the cells by utilizing the substrate while it is simultaneously being converted to butanol by the same cell mass. This is shown in its reaction rate expression, given by Equation (3.8).

$$r_{HBu} = C_X \left(Y_{HBu/S} q_S - k_{HBu} \frac{C_{HBu}}{K_{HBu} + C_{HBu}} \right) \tag{3.8}$$

As HBu is converted to BuOH, terms in the kinetic expression for BuOH resembles the kinetics for HBu. The reaction rate for BuOH is given by Equation

(3.9).

$$r_{BuOH} = C_X \left(Y_{BuOH/S} q_S + k_{BuOH} \frac{C_{HBu}}{K_{HBu} + C_{HBu}} \right) \tag{3.9}$$

The parameters k_{HBu}, k_{BuOH} and K_{HBu} are rate parameters which were obtained earlier in the EcoLodge project.

Also ethanol is formed by the cells. The ethanol kinetics are given by Equation (3.10).

$$r_{EtOH} = Y_{EtOH/S} q_S C_X \tag{3.10}$$

The parameter $Y_{EtOH/S}$ is the yield coefficient of ethanol on the substrate.

Note that the reaction rate for biomass, substrate and ethanol are proportional, hence, they are linearly dependent. As the conversion of substrate is an interesting parameter, the first component in the state vector, $\underline{C}$, is the concentration of substrate. In addition, the reaction rate for HBu may be written as a linear combination of the reaction rate of the substrate and butanol. This is shown in Equation (3.11), where the final expression on the right hand side matches Equation (3.9).

$$\begin{aligned} r_{HBu} &= -\left(Y_{HBu/S} + \frac{k_{HBu} Y_{BuOH/S}}{k_{BuOH}} \right) r_S - \frac{k_{HBu}}{k_{BuOH}} r_{BuOH} \\ &= -\left(Y_{HBu/S} + \frac{k_{HBu} Y_{BuOH/S}}{k_{BuOH}} \right) (-q_S C_X) - \\ &\quad \frac{k_{HBu}}{k_{BuOH}} C_X \left(Y_{BuOH/S} q_S + k_{BuOH} \frac{C_{HBu}}{K_{HBu} + C_{HBu}} \right) \\ &= C_X \left(Y_{HBu/S} q_S - k_{HBu} \frac{C_{HBu}}{K_{HBu} + C_{HBu}} \right) \end{aligned} \tag{3.11}$$

As the rate of formation of HBu can be written as a linear combination of the reaction rates of the substrate and BuOH, then the reaction rate for all components in the system may be described by combinations of r_S and r_{BuOH}. This means that substrate and BuOH are the state variables of the system, hence, $\underline{C} = [C_S, C_{BuOH}]^T$. The rate of formation of all components may therefore be expressed as in Equation (3.12).

$$\underline{r}_{all} = \begin{bmatrix} r_X \\ r_S \\ r_{HBu} \\ r_{BuOH} \\ r_{EtOH} \end{bmatrix} = \begin{bmatrix} -Y_{X/S} & 0 \\ 1 & 0 \\ -\left(Y_{HBu/S} + \frac{k_{HBu}Y_{BuOH/S}}{k_{BuOH}}\right) & -\frac{k_{HBu}}{k_{BuOH}} \\ 0 & 1 \\ -\frac{1}{Y_{EtOH/S}} & 0 \end{bmatrix} \begin{bmatrix} r_S \\ r_{BuOH} \end{bmatrix} = \underline{\underline{A}}_{BuOH}\underline{r} \quad (3.12)$$

The matrix $\underline{\underline{A}}_{BuOH}$ gives the linear relationship between the independent species and the dependent species. As $r_i = \frac{dC_i}{d\tau}$, then Equation (3.13) must also be true.

$$\begin{aligned} \frac{d\underline{C}_{all}}{d\tau} &= \underline{\underline{A}}_{BuOH}\frac{d\underline{C}}{d\tau} \\ d\underline{C}_{all} &= \underline{\underline{A}}_{BuOH} d\underline{C} \\ \underline{C}_{all} &= \underline{C}_{f,all} + \underline{\underline{A}}_{BuOH}(\underline{C} - \underline{C}_f) \end{aligned} \quad (3.13)$$

Here, $\underline{C}_{f,all}$ is the feed concentration for all the components. That is, $\underline{C}_{f,all} = [C_{f,X}, C_{f,S}, C_{f,HBu}, C_{f,BuOH}, C_{f,EtOH}]^T$. As the concentration of all components can be calculated for any point in the state space $C_S - C_{BuOH}$, the reaction rates for the state variables can be calculated by using Equation (3.7) and (3.9).

The values of the parameters used for modelling the butanol fermentation are given in Table 2. These values are for a temperature of 37°C in the fermentation broth.

Table 2: Kinetics parameters used in the fermentation model for butanol for a temperature of 37°C.

Symbol	Value	Unit
μ_{max}	0,678	1/h
K_S	30	g_S/L
$Y_{X/S}$	0,06	g_X/g_S
$Y_{HBu/S}$	0,75	g_{HBu}/g_S
K_{HBu}	0,272	g_{HBu}/L
k_{HBu}	3,53	L/(h g_X)
k_{BuOH}	0,951	(g_{BuOH} L)/(h g_X g_{HBu})
$Y_{BuOH/S}$	0.01	g_{BuOH}/g_S
$Y_{EtOH/S}$	0,0137	g_{BuOH}/g_S
$K_{i(BuOH)}$	17,7	g/L
$C_{HBu,max}$	11,0125	g/L
m_{HBu}	2,5	-

3.1.2 Production of Butyric Acid

The kinetics for the production of butyric acid is described by Song [9]. In addition to the main fermentation product HBu, acetic acid (HAc) will be produced as a side product. Both these acids decrease the pH in the broth; their pKa values are 4,76 and 4,82 for HAc and HBu respectively [32]. The increase in acidity of the fermentation broth contributes to inhibit the production of HBu. The microorganism used in this study was *Clostridium tyrobutyricum*, and the stoichiometry of the system is given by Equation (3.14) and (3.15).

$$C_6H_{12}O_6 \longrightarrow C_3H_7COOH + 2H_2 + 2CO_2 \tag{3.14}$$

$$C_6H_{12}O_6 \longrightarrow 2CH_3COOH + 4H_2 + 2CO_2 \tag{3.15}$$

The gaseous compounds H_2 and CO_2 were of no interest, and therefore not included in the modeling. The rate of formations for the remaining components are given in Equation (3.16).

$$
\begin{aligned}
r_X &= \mu C_X \\
r_{HBu} &= \alpha_{HBu} r_X + \beta_{HBu} C_X \\
r_{HAc} &= \alpha_{HAc} r_X \\
r_S &= -\frac{r_X}{Y_X} - \left(1 + \frac{2MW_{CO_2}}{MW_{HBa}}\right)\frac{r_{HBu}}{Y_{HBu}} - \\
&\left(1 + \frac{MW_{CO_2}}{MW_{HAc}}\right)\frac{r_{HAc}}{Y_{HAc}} - m_S C_X
\end{aligned}
\tag{3.16}
$$

Here, μ is again the cell growth parameter while α_i and β_i are the cell growth-associated product formation, and cell non-growth-associated product formation of component i, respectively. Y_i is the stoichiometric yield coefficient on glucose for component i and m_S is the cell maintenance coefficient.

In the expression for r_X, the cell growth parameter, μ, has a slightly different form than for the butanol production. Here, μ is defined by Equation (3.17).

$$
\mu = \frac{\mu_m C_S}{C_S + K_S + (C_S)^2/K_I}\left(1 - \frac{P}{P_d}\right)^i \tag{3.17}
$$

Here, μ_m is the maximum specific growth rate, K_S and K_i are substrate saturation and inhibition constants and P_d is the product concentration at where no cell growth occurs.

Note that substrate and product inhibition are taken into account by the terms C_S^2/K_i and $\left(1 - \frac{P}{P_d}\right)^i$, respectively. The total product concentration, P, is assumed to consist only of the main fermentation product, as well as HAc which is an important side product which contributes to decrease the pH in the broth. Thus, P is given by Equation (3.18).

$$
P = C_{HBu} + C_{HAc} \tag{3.18}
$$

The values for the parameters in this system are given in Table 3. In addition, the term MW_i denotes the molecular weight for component i. These are given as $\underline{MW} = [MW_{CO_2}, MW_{HAc}, MW_{HBu}] = [44.01, 60.05, 88.10]^T$ g/moles. When obtaining the parameters in Table 3, the temperature was controlled at 37°C and the pH was maintained at 6,0 by the addition of NH_4OH [9].

Table 3: Kinetics parameters used in the fermentation for butyric acid. The temperature was controlled at 37°C and the pH was maintained at 6,0 by the addition of NH_4OH when obtaining these values [9].

Symbol	Value	Unit
μ_m	0,48	1/h
Y_X	0,807	g_X/g_S
Y_{HBu}	0,978	g_{HBu}/g_X
Y_{HAc}	0,999	g_{HAc}/g_X
K_d	0,0024	1/h
K_S	1,62	g/L
P_d	48,3	g/L
α_{HBu}	2,92	g_{HBu}/g_X
α_{HBu}	0,72	g_{HAc}/g_X
i	5,18	-
β_{HBu}	0,057	g_{HBu}/g_X
m_S	0,002	1/h

From the stoichiometry, there are two independent reactions. However, a reduction of the dimensionality of the system is desirable so that the AR for the HBu production can be visualized in two dimensions, where one dimension is the residence time, τ.

In order to reduce the dimensionality of the system, the rate expressions in Equation (3.16) can be simplified slightly. In the rate expression for the substrate, the specific maintainence coefficient, m_S, can be assumed to be negligible (real value: 0,002 [1/h]). Also the expression for the production of HBu should be simplified. The parameter which is associated with the cell non-growth product formation, β_{HBu}, may be assumed to be negligible (real value: 0,057 g HBu/g cell). In this way, the reaction rate vector may be expressed as in Equation (3.19).

$$\underline{r_{all}} = \begin{bmatrix} r_X \\ r_S \\ r_{HAc} \\ r_{HBu} \\ r_\tau \end{bmatrix} = \begin{bmatrix} \mu c_X \\ -\frac{r_X}{Y_X} - \left(1 + \frac{2MW_{CO_2}}{MW_{HBa}}\right)\frac{r_{HBu}}{Y_{HBu}} - \left(1 + \frac{MW_{CO_2}}{MW_{HAc}}\right)\frac{r_{HAc}}{Y_{HAc}} \\ \alpha_{HAc} r_X \\ \alpha_{HBu} r_X \\ 1 \end{bmatrix} \tag{3.19}$$

By substituting the rate equations for r_{HAc} and r_{HBu} into the rate equation for the substrate, the change of concentration for the substrate can be stated explicitly as a function of the cell mass. This is shown in Equation (3.20).

$$r_S = -\left(\frac{1}{Y_X} + \left(1 + \frac{2MW_{CO_2}}{MW_{HBu}}\right)\frac{\alpha_{HBu}}{Y_{HBu}} + \left(1 + \frac{MW_{CO_2}}{MW_{HAc}}\right)\frac{\alpha_{HAc}}{Y_{HAc}}\right) r_X \tag{3.20}$$

Thus, the rate of formation of the substrate is proportional to the rate of formation for the cell mass. The multiplier is simply a collection of constants, and to simplify notation, these constants are termed γ. Hence, γ is defined by Equation (3.21).

$$\gamma = \frac{1}{Y_X} + \left(1 + \frac{2MW_{CO_2}}{MW_{HBu}}\right)\frac{\alpha_{HBu}}{Y_{HBu}} + \left(1 + \frac{MW_{CO_2}}{MW_{HAc}}\right)\frac{\alpha_{HAc}}{Y_{HAc}} \tag{3.21}$$

Given that the previous approximations were valid, the state space vector is two dimensional, wheras one (pseudo) component is the residence time and the other is the concentration of a component of free choice. As the conversion of substrate is an interesting parameter, the state vector was chosen as $\underline{C} = [C_S, \tau]^T$, which means that $\underline{r} = [r_S, r_\tau]$. The rate vector for all the components can now be expressed as a linear combination of the independent components, as shown in Equation (3.22). Also here, the matrix $\underline{\underline{A}}_{HBu}$ gives the relationship between the linearly independent and dependent species.

$$\underline{r}_{all} = \begin{bmatrix} -\frac{1}{\gamma} & 0 \\ 1 & 0 \\ -\frac{\alpha_{HAc}}{\gamma} & 0 \\ -\frac{\alpha_{HBu}}{\gamma} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} r_S \\ r_\tau \end{bmatrix} = \underline{\underline{A}}_{HBu}\underline{r} \tag{3.22}$$

The concentrations of biomass, acetic acid and butyric acid can be calculated for any concentration of substrate by applying the same method as when Equation (3.13) was derived. The resulting equations are given in Equation (3.23). The vector $\underline{C}_{f,all}$ contain the feed concentrations for all components in the system, hence, it is defined as $\underline{C}_{f,all} = [C_{f,X}, C_{f,S}, C_{f,HAc}, C_{f,HBu}, C_{f,\tau}]^T$. The same indexation is used for the outlet concentration for all components, $\underline{C}_{all}$.

$$\underline{C}_{all} = \underline{C}_{f,all} + \underline{\underline{A}}_{HBu}(\underline{C} - \underline{C}_f) \tag{3.23}$$

In order to obtain all the concentrations of the system, only the feed concentrations need to be specified in order to have the concentrations uniquely determined for any point in $C_S - \tau$ space. The reaction rate vector for the independent species may therefore then be computed as in Equation (3.24).

$$\underline{r} = \begin{bmatrix} r_S \\ r_\tau \end{bmatrix} = \begin{bmatrix} -\gamma r_x \\ 1 \end{bmatrix} = \begin{bmatrix} -\gamma \mu C_X \\ 1 \end{bmatrix} \tag{3.24}$$

It should be noted that the preceeding system of equations were valid given the assumption that m_S and β_{HBu} were negligible. To check this assumption, a comparison between the full model, described by Equation (3.16), and the simplified model in Equation (3.22) was done. The result is shown in Figure 10.

As it can be observed, the concentration trajectories resembles each other closely while the residence time is less than 25 h. After 25h, the substrate is used up, and the growth of the cell mass ceases. For the simplified model, this means that equilibrium has been reached. This is due to that the values of the maintainance coefficient for the cells, m_S, as well as the non-growth product formation, β_{HBu}, have been approximated to be zero. The physical interpretation of setting $m_S = 0$, is that no substrate consumption is required to maintain the cells

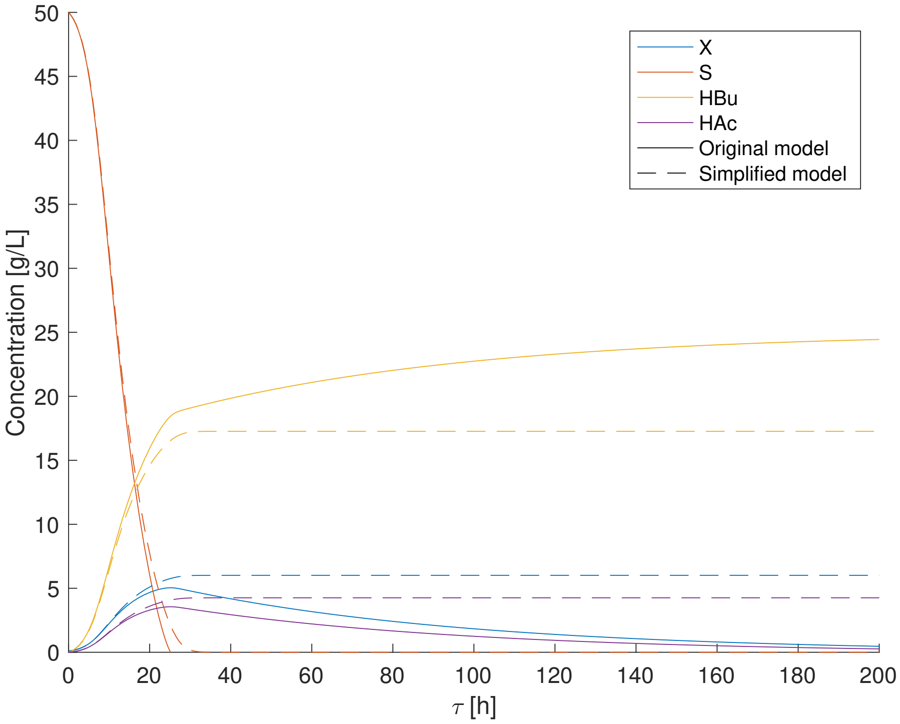


Figure 10: Comparison of the concentration profiles of the full model for HBu fermentation and the simplified model. The full model is given by Equation (3.16), and the simplified model is given by Equation (3.22).

[33]. By approximating $\beta_{HBu} = 0$, it is in effect assumed that the formation of HBu is only affected by the cell growth rate, and not by the amount of the cells. The validity of this assumption is strengthened by noting that $\alpha_{HBu} >> \beta_{HBu}$ [9]. Note that the physical interpretation of these two approximations lose their validity gradually as the substrate concentration decreases. In the full model, the substrate is depleted after 25 h, and the approximations have no longer any physical meaning.

However, for a continuous process, a reactor with a residence time larger than 25 h may be considered unfeasible. As an example, consider if the flowrate throughout the reactor is 1 m^3/h. Assume a single PFR is used, where the radius is set to be 1/10 of its length. The required length of the reactor would then be $\sqrt[3]{\frac{4}{\pi}\tau Q_{flow} 10^2} = 14,7$ m. The corresponding radius would be 1,5 m. From an engineering perspective, it can therefore be argued that a residence time larger than 25 h is not practically feasible, especially for continuous processes. Hence, only the first 25 h in Figure 10 will be used as a basis for evaluating the models.

The relative error of the simplified model compared to the full model is visualized in Figure 11. For HBu, the absolute value of the relative error in the simplified model is always smaller than 10,3%. For the substrate, the relative error at equilibrium is not displayed, but the maximum value of the relative error is 2290%.

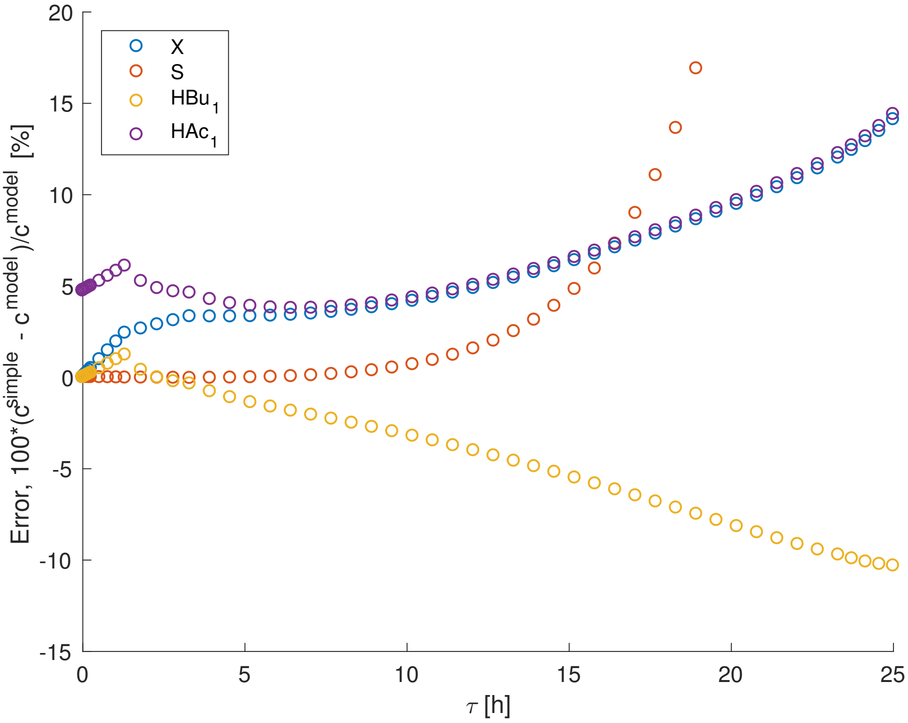


Figure 11: The relative error of each component in the simplified HBu model compared to the full model.

The reason for this very high relative error for the substrate is that the concentration of substrate is going towards zero at equilibrium. However, if the nominal error for the substrate is being considered, then a maximum of 2,13 g/L is observed. For HBu, the simplified model underestimates the concentration of HBu by 1,89 g/L in the worst case.

In this fermentation, HBu is the component of interest. By using the simplified model, an extra error source is introduced into the simulation. However, as Figure 10 shows, both the concentration of HBu and the utilization of substrate are *always* underestimated in the simplified model. Calculations based on the simplified HBu fermentation model will therefore consistently be conservative, both regarding what is attainable in terms of HBu concentrations as well as for conversion of substrate. This feature of the simplified model speaks in favor of accepting the simplification, as it also reduces the dimensionality of the HBu fermentation system and thus facilitates AR analysis. From now on, when referring to the HBu fermentation model, it is the simplified fermentation model which is meant, defined by Equation (3.22).

3.1.3 Production of Butyl Butyrate

The kinetics for the production of butyl butyrate, BuB, is described by Yadav [8]. The reaction is an esterification, which is catalyzed by the enzyme Novozym SP 435 from *Candida antarctica*. Butanol and butyric acid are the reactants. The stoichiometry of the reaction is given by Equation (3.25).

$$\mathrm{C_3H_7COOH + C_4H_9OH \rightleftharpoons H_2O + C_3COOC_4H_9} \tag{3.25}$$

As there is only one independent reaction, it is known that only the concentration of one component is required to describe the full AR. The concentration of the other components may then be calculated by their stoichiometric dependencies. As BuB is the component of interest, this is the component which will be included in the state vector. The residence time is also a feature of interest, and will therefore be included in the state vector. Hence, $\underline{C} = [C_{BuB}, \tau]^T$. The reaction rate for BuB is given by Equation (3.26).

$$r_{BuB} = \frac{MW_{BuB} r_{max} \left(\frac{C_{HBu}}{MW_{HBu}}\right)\left(\frac{C_{BuOH}}{MW_{BuOH}}\right)}{\left[K_{i(HBu)} + \left(\frac{C_{HBu}}{MW_{HBu}}\right)\right] K_{m(BuOH)} + \left[K_{m(HBu)} + \left(\frac{C_{HBu}}{MW_{HBu}}\right)\right]\left(\frac{C_{BuOH}}{MW_{BuOH}}\right)} \tag{3.26}$$

Here, r_{max} is the maximum reaction rate, $K_{i(HBu)}$ is the inhibition constant of butyric acid while $K_{m(BuOH)}$ and $K_{m(HBu)}$ are the Michaelis constants for BuOH and HBu, respectively.

The rate of reaction for all components can therefore be written in terms of the state variables of the system, as in Equation (3.27). Again, $\underline{\underline{A}}_{BuB}$ gives the linear relationship between the independent variables and the dependent variables.

$$\underline{r}_{all} = \begin{bmatrix} r_{BuOH} \\ r_{HBu} \\ r_{BuB} \\ r_{\tau} \end{bmatrix} = \begin{bmatrix} -\frac{MW_{BuOH}}{MW_{BuB}} & 0 \\ -\frac{MW_{HBu}}{MW_{BuB}} & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} r_{BuB} \\ r_{\tau} \end{bmatrix} = \underline{\underline{A}}_{BuB}\underline{r} \tag{3.27}$$

The concentrations of all the components present in the system may be calculated in the same way as in section 3.1.1 and 3.1.2. This is formulated in Equation (3.28). Here, $\underline{C}_{f,all}$ is the feed concentration of all the components, $\underline{C}_{f,all} = [C_{f,BuOH}, C_{f,HBu}, C_{f,BuB}]^T$.

$$\underline{C}_{all} = \underline{C}_{f,all} + \underline{\underline{A}}_{BuB}(\underline{C} - \underline{C}_f) \tag{3.28}$$

The values for the parameters in Equation (3.26) are given in Table 4. These values are for a temperature of 30°C in the reactor [8]. As in the the original article, 160 mg of enzyme was used.

Table 4: Parameters used in the kinetics describing the production of butyl butyrate for a temperature of 30°C.

Symbol	Value	Unit
r_{max}	0,02	mmol/(L min g_{enzyme})
$K_{i(HBu)}$	0,033	mmol/L
$K_{m(HBu)}$	0,051	mmol/L
$K_{m(BuOH)}$	0,028	mmol/L
MW_{BuOH}	74,12	g/mol
MW_{HBu}	88,11	g/mol
MW_{BuB}	144,21	g/mol

The kinetic model for production of BuB is valid in an organic solvent. In the literature, heptane was used as a solvent [8]. In this work, it has been assumed that the model is valid with the liquid which used in the fermentation model. It is assumed that this liquid is a pseudo-phase, as it consists of both water, cells and organic compounds, such as the alcohols and acids. The assumption of the liquid being a pseudo-phase will be more discussed in section 6.3.

3.2 Scenario 2

In case two, butanol will be produced separately, as described in section 3.1.1. The BuOH producing cells will be removed, and this cell-free product stream from the BuOH reactor will be an inlet to the second reactor block. In the second reactor block, butyric acid is formed continuously as a fermentation product. Simultaneously, the butyric acid is reacting with the inlet butanol stream to form butyl butyrate in an esterification reaction. This second reactor will sometimes be referred to as the combined reactor model. The layout of the process flow diagram for this case is shown in Figure 12.

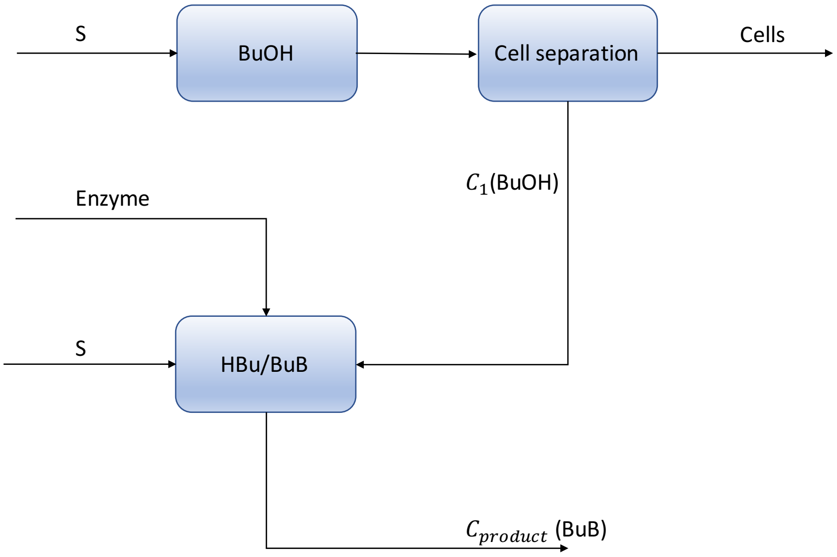


Figure 12: Process flow diagram for scenario 2.

As HBu is continuously being removed from the fermentation broth by the esterification reaction, the concentration of HBu is lower than in scenario 1. It is therefore assumed that the reaction rate for the cells are higher, due to less product inhibition. However, acetic acid is left unreacted in the broth. The relatively high concentration of HAc should contribute to a higher degree of product inhibition, which might counteract the aforementioned effect to some extent. In addition, the butanol concentration in the input stream might be high, hence, some inhibition due to butanol is also expected. Considering the overall effect, it is however expected that there will be less product inhibition than in scenario 1 due to the low concentration of HBu. A higher conversion of substrate is therefore expected by this setup. As more sugar is expected to converted into HBu by the cells, this also implies that the concentration of BuB will be higher than in scenario 1, as there are more reactants present.

The model equations for the butanol production are unchanged from section 3.1.1. Thus, all the concentrations from the butanol fermentation can be described in the $C_S - C_{BuOH}$-space, as these are the state variables. For the model of the reactor block with simultaneous production of HBu and BuB, the total rate of reaction for component i is described by Equation (3.29). This equation is verified by keeping in mind that $r_i = \frac{dC_i}{d\tau}$, thus multiplying with $d\tau$ on both sides of Equation (3.29), yields the change of concentration due to two different reactions.

$$r_{i,tot} = r_{i,fermentation} + r_{i,esterification} \tag{3.29}$$

The kinetic system for the simultaneous production of HBu and BuB are therefore given by Equation (3.30) - (3.35).

$$\begin{align}
r_{X_{HBu}} &= \mu C_{X_{HBu}} \tag{3.30}\\
r_{HAc} &= \alpha_{HAc} r_{X_{HBu}} \tag{3.31}\\
r_{BuB} &= \frac{MW_{BuB} r_{max} \left(\frac{C_{HBu}}{MW_{HBu}}\right)\left(\frac{C_{BuOH}}{MW_{BuOH}}\right)}{\left[K_{i(HBu)} + \left(\frac{C_{HBu}}{MW_{HBu}}\right)\right] K_{m(BuOH)} + \left[K_{m(HBu)} + \left(\frac{C_{HBu}}{MW_{HBu}}\right)\right]\left(\frac{C_{BuOH}}{MW_{BuOH}}\right)} \tag{3.32}\\
r_{BuOH} &= -\frac{MW_{BuOH}}{MW_{BuB}} r_{BuB} \tag{3.33}\\
r_{HBu} &= r_{HBu,fermentation} + r_{HBu,esterification} \notag\\
&= \alpha_{HBu} r_{X_{HBu}} - \frac{MW_{HBu}}{MW_{BuB}} r_{BuB} \tag{3.34}\\
r_S &= -\frac{r_X}{Y_X} - \left(1 + \frac{2MW_{CO_2}}{MW_{HBa}}\right)\frac{r_{HBu}}{Y_{HBu}} - \left(1 + \frac{MW_{CO_2}}{MW_{HAc}}\right)\frac{r_{HAc}}{Y_{HAc}} \tag{3.35}
\end{align}$$

By substituting the relevant rate of formations into the rate of reaction for the substrate, Equation (3.35) can be expressed only in terms of r_X and r_{BuB}. This is shown in Equation (3.36).

$$\begin{aligned}
r_S &= -\frac{r_X}{Y_X} - \left(1 + \frac{2MW_{CO_2}}{MW_{HBa}}\right)\frac{r_{HBu}}{Y_{HBu}} - \left(1 + \frac{MW_{CO_2}}{MW_{HAc}}\right)\frac{r_{HAc}}{Y_{HAc}} \\
&= -\frac{r_X}{Y_X} - \left(1 + \frac{2MW_{CO_2}}{MW_{HBu}}\right)\frac{1}{Y_{HBu}}\left(\alpha_{HBu} r_X - \frac{MW_{HBu}}{MW_{BuB}} r_{BuB}\right) \\
&\quad - \left(1 + \frac{MW_{CO_2}}{MW_{HAc}}\right)\frac{\alpha_{HAc} r_X}{Y_{HAc}} \\
&= -\gamma r_X + \frac{1}{Y_{HBu}}\left(1 + \frac{2MW_{CO_2}}{MW_{HBu}}\right)\left(\frac{MW_{HBu}}{MW_{BuB}}\right) r_{BuB}
\end{aligned} \tag{3.36}$$

To ease notation, the parameter ω, which is simply a collection of constants, is introduced in Equation (3.37).

$$\omega = \frac{1}{Y_{HBu}}\left(1 + \frac{2MW_{CO_2}}{MW_{HBu}}\right) \tag{3.37}$$

From the set of Equations (3.30) - (3.36), it can be seen that all reaction rates can be expressed as a linear combination of two components. As the concentration of BuB and utilization of substrate are two features of special interest, the state space vector is selected to be $\underline{C} = [C_S, C_{BuB}]^T$. By using Equation (3.36), the rate of reaction for cell mass can be written as in Equation (3.38).

$$r_X = -\frac{1}{\gamma} r_S + \frac{\omega}{\gamma}\left(\frac{MW_{HBu}}{MW_{BuB}}\right) r_{BuB} \tag{3.38}$$

By inserting Equation (3.38) into the rate expression for HBu, Equation (3.34), the reaction rate of HBu can be computed as a linear combination of r_S and r_{BuB}. This is shown in Equation (3.39).

$$\begin{aligned}
r_{HBu} &= -\alpha_{HBu}\left(\frac{1}{\gamma} r_S + \frac{\omega}{\gamma}\left(\frac{MW_{HBu}}{MW_{BuB}}\right) r_{BuB}\right) - \frac{MW_{HBu}}{MW_{BuB}} r_{BuB} \\
&= -\frac{\alpha_{HBu}}{\gamma} r_S + \left(\frac{\alpha_{HBu}\omega}{\gamma} - 1\right)\left(\frac{MW_{HBu}}{MW_{BuB}}\right) r_{BuB}
\end{aligned} \tag{3.39}$$

By following the same procedure for HAc, the reaction rate vector for all components can be expressed as a linear combination of r_S and r_{BuB}. The final expression is given by Equation (3.40).

$$\underline{r}_{all} = \begin{bmatrix} r_X \\ r_S \\ r_{HAc} \\ r_{HBu} \\ r_{BuOH} \\ r_{BuB} \end{bmatrix} = \begin{bmatrix} -\frac{1}{\gamma} & \frac{\omega}{\gamma}\frac{MW_{HBu}}{MW_{BuB}} \\ 1 & 0 \\ -\frac{\alpha_{HAc}}{\gamma} & \alpha_{HAc}\frac{\omega}{\gamma}\frac{MW_{HBu}}{MW_{BuB}} \\ -\frac{\alpha_{HBu}}{\gamma} & \frac{MW_{HBu}}{MW_{BuB}}\left(\alpha_{HBu}\frac{\omega}{\gamma}-1\right) \\ 0 & -\frac{MW_{BuOH}}{MW_{BuB}} \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} r_S \\ r_{BuB} \end{bmatrix} = \underline{\underline{A}}_{sc2}\underline{r} \tag{3.40}$$

By again using the methodology Equation (3.13) is based on, it is possible to compute the concentration of all components from the state space vector by Equation (3.41). Also here, $\underline{\underline{A}}_{sc2}$ gives the relationship between the independent and dependent species.

$$\underline{C}_{all} = \underline{C}_{f,all} + \underline{\underline{A}}_{sc2}(\underline{C} - \underline{C}_f) \tag{3.41}$$

When the concentration of all components are known, r_{BuB} can be calculated by (3.32) and r_S can be calculated by Equation

$$r_S = -\gamma\mu C_X + \omega\left(\frac{MW_{HBu}}{MW_{BuB}}\right) r_{BuB} \tag{3.42}$$

The cell growth parameter for the HBu productinging bacteria, μ, are again given by Equation (3.17). As the feed to this reactor has a stream with high concentration of BuOH, it is desirable to incorporate an inhibition term for BuOH into Equation (3.17). However, the BuOH inhibition term was neglected due to lack of data.

As the model for the simultaneous production of HBu and BuB is a mix of the two original models described in section 3.1.2 and 3.1.3, the parameters used in this subsection are the same as in the aforementioned sections. Refer to Table 3 and 4 for the values used.

4 Computer Tools

For the main part of the simulations, the Anaconda distribution of Python 3.6.0 was used [34]. The Anaconda version was 4.3.1. For mesh generation, MATLAB_R2017a was employed, as the built-in MATLAB function *inpolygon* facilitated to include only the desirable grid points. This will be more discussed in section 5.1.3 and 5.2. In addition, MATLAB was used as a plotting tool for all figures where a rate field was included. This was due to that the *quiver* function of Matplotlib turned out to distort rate vectors.

In order to construct ARs in a generic way, a Python module called *artools.py* was created. This module can be found in Appendix B.1. All external python packages which were used are given in Table 5. All these modules are included in the Anaconda distribution of Python [34].

Table 5: External Python packages used for the simulations

External package	Version
NumPy [35]	1.11.3
SciPy [36]	0.18.1
SymPy [37]	1.0
Matplotlib [38]	2.0.0

4.1 Numerical Solvers and Convex Hull Computation

Two solvers were used in the simulations. For solving ODEs, scipy.integrate.odeint() was used. This function uses LSODA from the FORTAN library odepack, and the solver works for both stiff and non-stiff systems. For stiff systems, the Backward Differentiation Method (BDF) is employed. For non-stiff systems, the Adams method is used [39].

In order to solve non-linear equations, such as the CSTR design equation, scipy.optimize.fsolve() was used. This is a wrapper on MINPACK's hybrd and hybrdj algorithms, which use a modification of Powell's hybrid algorithm [40].

To compute the convex hull for a set of points, scipy.spatial.ConvexHull() was employed. This function uses the Qhull-library to compute the convex hull [41].

5 Attainable Region Analysis for Butyl Butyrate Production

In this section, the results of the AR analysis for the two different process scenarios in section 3 will be presented. A comparison of the two models will be made based on the maximum attainable BuB concentration. The concentration in each flow in the resultant process flow diagram are given in a table at the end of each subsection.

5.1 The AR for Scenario 1

In scenario 1, there were separate reactor networks producing BuOH and HBu, with the subsequent esterification of these two streams to BuB. Separate ARs for each reactor network were therefore made for the BuOH and HBu reactors. Based on these ARs, a variety of feed points were available for the esterification reactor. The convex hull of these two separate fermentation reactor networks will be termed the *feed region* to the BuB reactor. A regular grid was made in this feed region, and each grid point represented a possible feed composition to the BuB reactor. The AR for the BuB reactor was made based on each grid point. The selection criteria for the optimal reactor configuration was the one which maximized the concentration of BuB. The construction of the ARs for the BuOH, HBu and BuB reactor networks are described in section 5.1.1, 5.1.2 and 5.1.3, respectively.

5.1.1 AR for BuOH Production

The model for the BuOH production is given in section 3.1.1. The feed to the reactor network is given in Table 6.

For the BuOH fermentation network, the independent variables were chosen as $\underline{C} = [C_S, C_{BuOH}]^T$. By solving the design equations for a PFR and CSTR, Equation (2.5) and (2.6) respectively, the concentration profiles for the independent species were calculated. This is shown in Figure 13.

Equation (3.13) gives $\underline{C}_{all}$ when $\underline{C}$ is defined. The concentration profiles for X, EtOH and HBu were calculated by using this relationship. The result is given

Table 6: Feed concentrations to the BuOH reactor network

Component	Feed concentration [g/L]
X	0.1
S	50
HBu	0
BuOH	0
EtOH	0

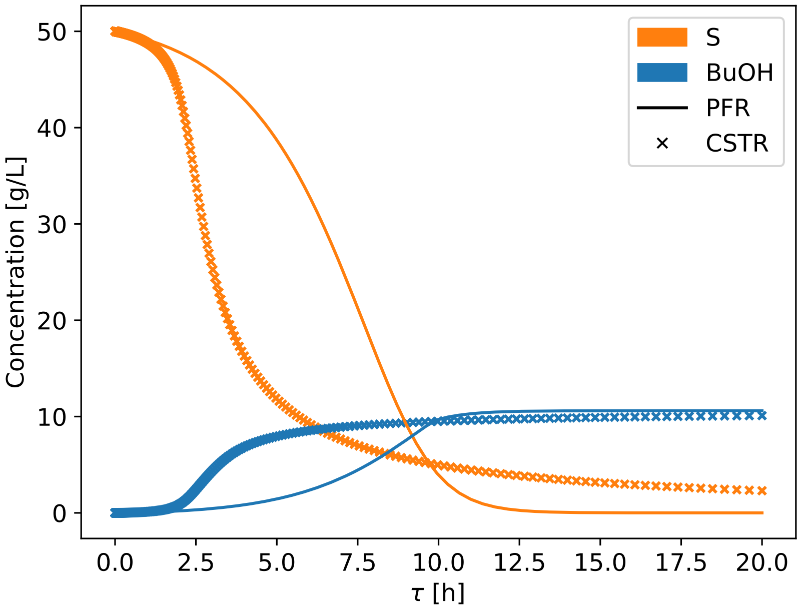


Figure 13: Concentration profiles for substrate and BuOH in the BuOH fermenter. The scattered line indicates the CSTR solution, while the solid line is the concentration profile in a PFR.

in Figure 14.

When solving the CSTR design equation, a variety of different guesses of the solution were supplied to the solver, in order to check for multiple solutions. Multiple solutions for the independent species were indeed found, however, these solutions always resulted in negative concentrations for one or more of the dependent species. Hence, it is believed that there exist no other physically realizable solutions for the CSTR design equation than the one visualized in Figure 13 and

14.

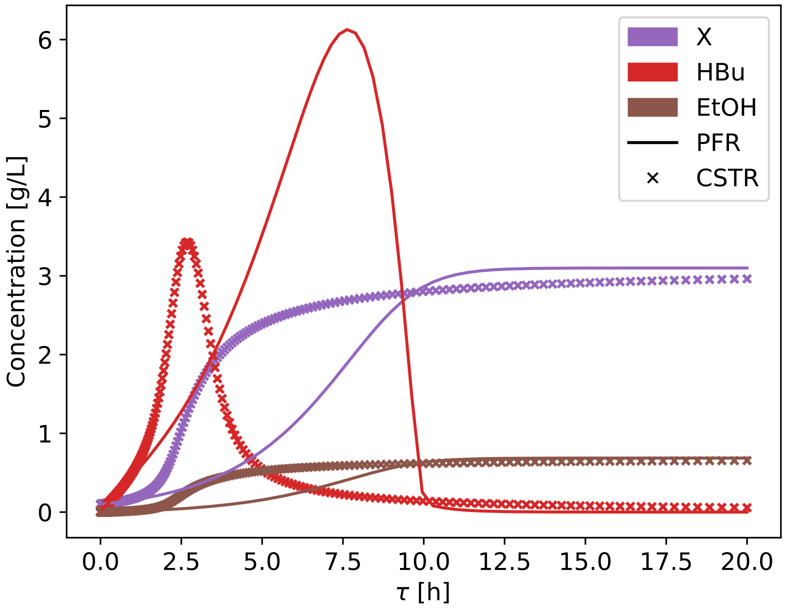


Figure 14: Concentration profiles for the cells, ethanol and butyric acid in the BuOH fermenter. The scattered line indicates the CSTR solution, while the solid line is the concentration profile in a PFR.

By pairing the $C_S - C_{BuOH}$ pairs in Figure 13, the AR for the BuOH fermenter can be made. This is shown in Figure 15. The reason why no rate vectors are displayed on the upper part of the figure, is that negative concentrations of some of the linearly dependent components are obtained in this area.

By drawing the convex hull based on the CSTR locus and PFR trajectory in Figure 15, it can be shown that all the points on the boundary are formed by a PFR with a residence time which ensures equilibrium. The upper part of the AR in Figure 15 can be formed by having a bypass of feed to the PFR outlet. This might be difficult to see since the CSTR locus operates close to this mixing line. However, the secondary axis in Figure 15 which is zoomed in close to the equilibrium point, shows that the mixing line is in fact obtained by a PFR with bypass. As all points on the boundary can be obtained by a PFR, theorem 2.1 ensures that the AR cannot be expanded by the means of any other reactor. The maximum attainable concentration of BuOH was therefore found to be 10,60 g/L.

As a final remark, from the concentration profiles in Figure 13 and 14, it can be observed that the reaction is faster in the CSTR than in the PFR. Thus, if residence time is a parameter of interest as well as the concentrations, then it might be a better solution to use a properly sized CSTR instead of a PFR operated at equilibrium. The time it takes to reach equilibrium for a PFR is approximately 12,5 h, as indicated by Figure 13 and 14. From Figure 13 it can be observed that for a CSTR with residence time of 5 h, the corresponding concentration of BuOH will be 7,95 g/L. As the analysis is considering the attainable concentrations as the main goal, the PFR with bypass will be representing the AR for the BuOH fermentation in the following sections.

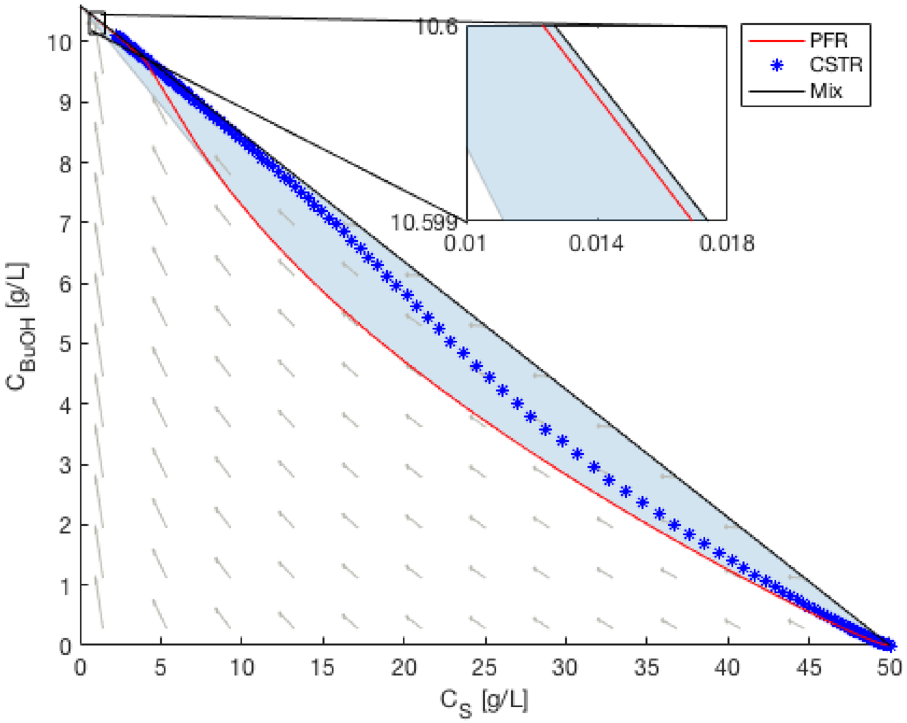


Figure 15: The AR for the BuOH fermenter. The scattered blue line indicates the CSTR solution while the solid red line is the concentration profile in a PFR. The secondary axis shows that all points on the boundary can be obtained by a PFR with bypass.

5.1.2 AR for HBu Production

The kinetic model for the HBU fermentation is described in section 3.1.2. The feed to the reactor network is given in Table 7.

Table 7: Feed concentrations to the HBu reactor network

Component	Feed concentration [g/L]
X	0.1
S	50
HAc	0
HBu	0

The independent variables in the HBu model are the concentration of substrate and the residence time of the system. The concentration of the cells and the substrate as a function of the residence time for a PFR and a CSTR with a feed as in Table 7, is given in Figure 16.

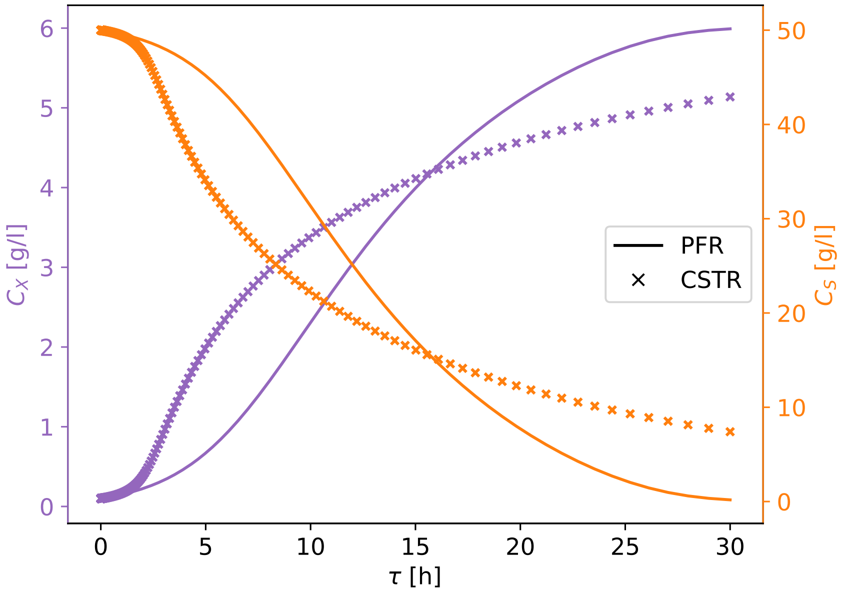


Figure 16: Concentration profiles for the substrate and cells in the HBu fermenter. The scattered line indicates the CSTR solution, while the solid line is the concentration profile in a PFR.

Based on the concentration of the substrate, the concentration of the other components were uniquely determined by Equation (3.23). The concentration profiles for the acids are given in Figure 17.

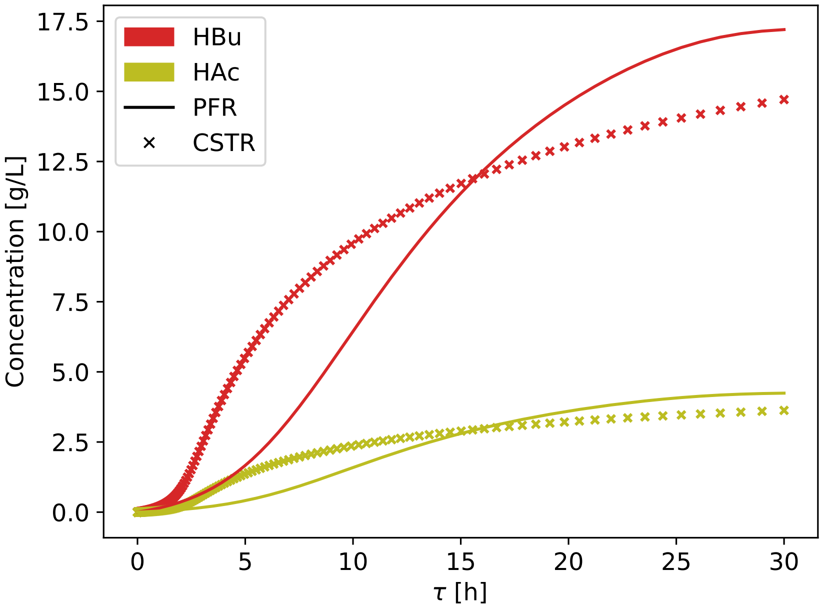


Figure 17: Concentration profiles of the acids in the HBu fermenter. The scattered line indicates the CSTR solution while the solid line is the concentration profile in a PFR.

As the AR can be expanded significantly if the CSTR design equations have multiple roots, it is necessary to perform a thorough search every time a CSTR is used. As the system had residence time and the concentration of a single component as independent variables, the search for multiple solutions of Equation (2.6) was done by meshing points in $C_S - \tau$-space, and based on the grid point, the residual of Equation (5.1) was calculated.

$$res = (\underline{C} - \underline{C}_f) - \tau \underline{r}(\underline{C}) \tag{5.1}$$

A surface was made in $C_S - \tau - res$-space, which stated that whenever $res = 0$, the CSTR had a solution. This procedure is facilitated by its easy grid generation, and that it is possible to visualize the result as it is a three dimensional surface.

This was done for the HBu fermentation, and the result is shown in Figure 18. Here, the curved surface is the value Equation (5.1) takes for every gridpoint in the $C_S - \tau$-space. The red plane takes the value zero in the res-direction. This means that a solution of the CSTR design equation is obtained only when the curved surface and the red plane intersects.

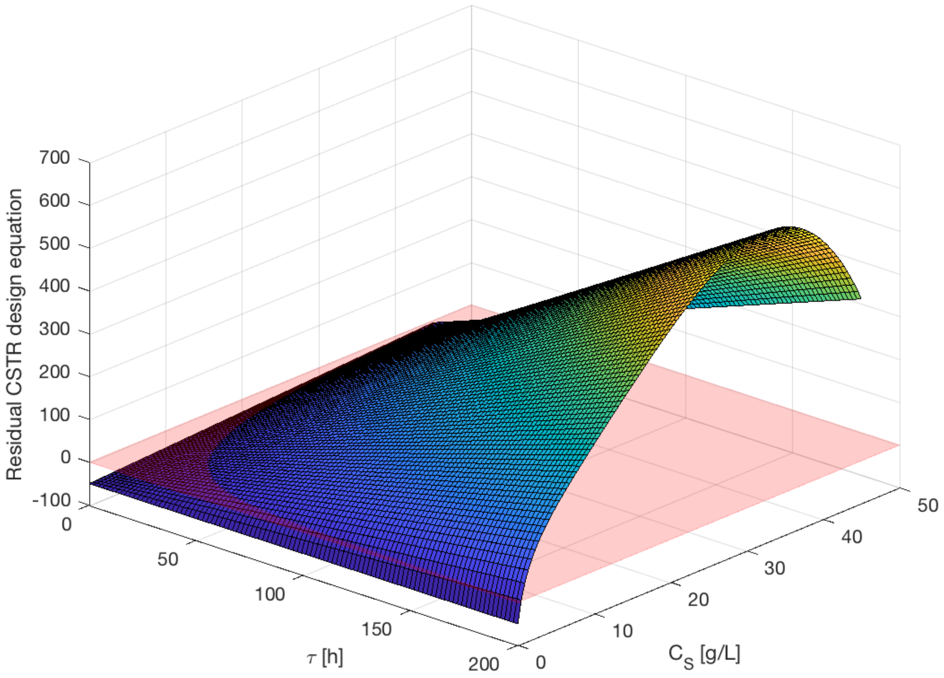


Figure 18: Visualization of a search for multiple solutions of the CSTR design equation for the HBu fermentation model. The curved surface is the residual of the CSTR design equation given by Equation (5.1), and the red plane indicates when the residual is zero. When these surfaces intersect, the CSTR design equation has a solution. Multiple solutions is obtained if the curved surface intersects the red plane for the same residence time.

As it can be observed from Figure 18, the curved surface intersects the red plane only once for any value of the residence time. This implies that there are only one solution of the CSTR design equation for the HBu fermentation model.

As it was ensured that the model yields only one solution of the CSTR design equation, construction of the AR could proceed. From the concentration profiles, each substrate concentration was paired with its corresponding residence time. In AR literature, it is customary to plot the residence time on the ordinate axis and the concentration on the abscissa. The resulting AR plot in $C_S - \tau$-space is given in Figure 19. Note that this AR plot is unbounded on the τ axis. This is natural, as the reactants may be in the reactor in an infinite amount. Therefore, a constraint on the τ-axis is set, and this value of τ corresponds to the time it takes to reach equilibrium.

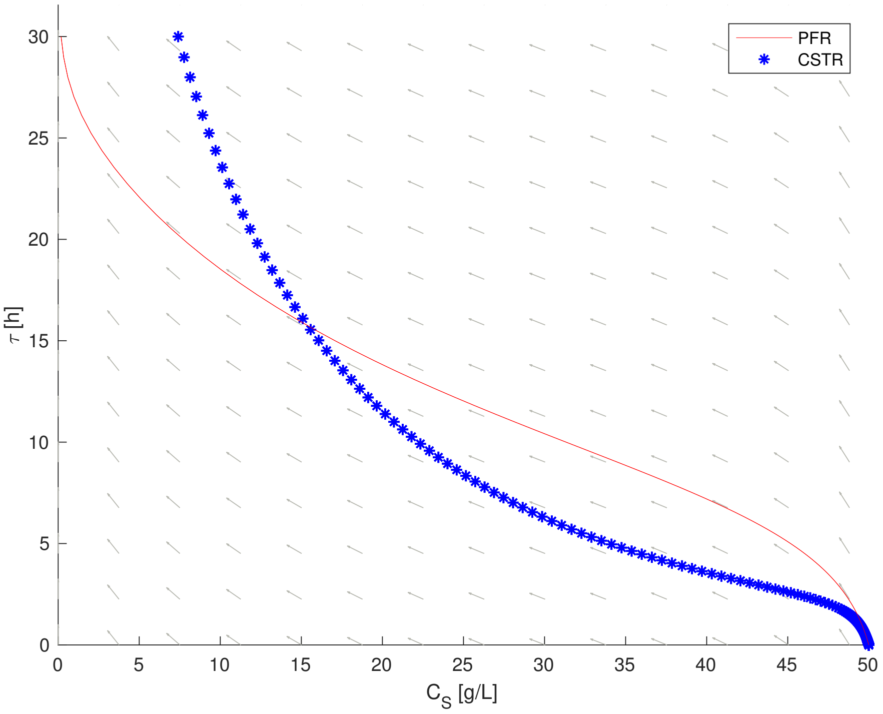


Figure 19: First candidate AR for the HBu fermentation model. The scattered line indicates CSTR solution while the solid line is the concentration profile in a PFR. This is not the full AR as the reactor network is not terminated with a PFR.

As it can be observed from Figure 19, the residence time for a CSTR is lower than the residence time for a PFR when comparing any substrate concentration higher than 15,6 g/L. This means that a smaller volume is required in a CSTR than in a PFR in order to obtain the same outlet concentration, given that $C_S > 15,6$ g/L. Hence, the CSTR contributes to form a large part of the boundary of this first approximation of the AR.

From section 2.5, it is known that the AR boundary consists entirely of mixing surfaces and rection surfaces. The mixing surface is obtained by drawing the convex hull for the two reactors in Figure 19. The concave hull which is given by the feed point, $C_{S,feed} = 50$ g/L, and a suitable point on the CSTR locus, can be filled with a mixing line. This CSTR will serve as a connector on the AR boundary, and it is then known by Theorem 2.1 that the next reactor in the network should be a PFR. By taking the convex hull of all these configurations, it can be observed that the AR is convex, and all rate vectors point into the region or they are tangent to the boundary. Hence, the AR fulfills the necessary conditions stated in section 2.4. All these reactor configurations are given in Figure 20. The residence time of the critical CSTR was found to be 5,51 h.

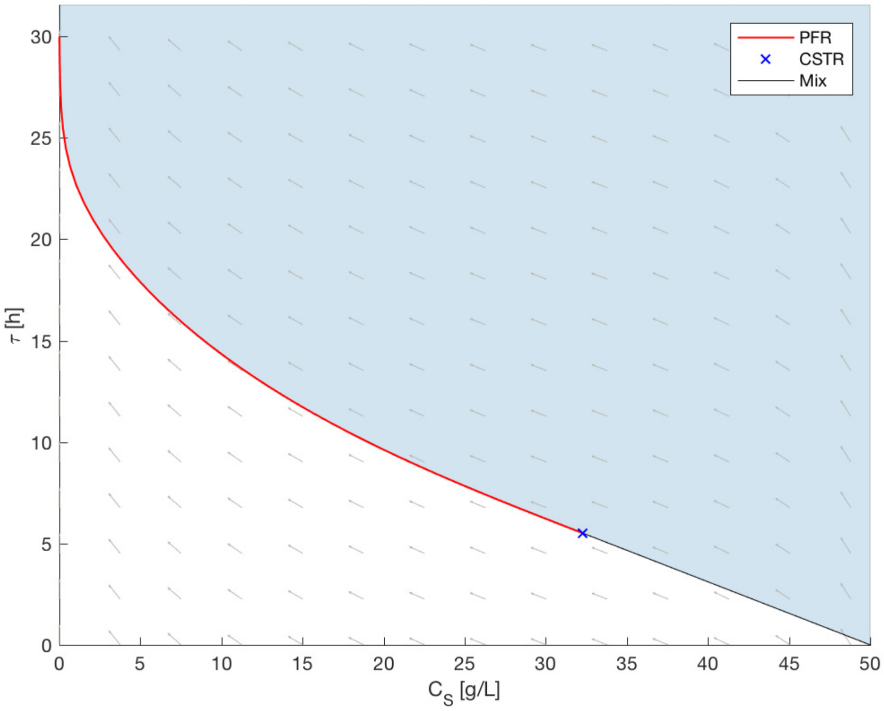


Figure 20: The full AR for the HBu fermentation model. The blue cross indicates a critical CSTR, while the solid red line is the PFR trajectory. The PFR is initiated from the outlet of the CSTR.

The concentration profiles for all the species in the solution, based on the optimal process configuration, is given in Figure 21. It is observed that equilibrium is reached after a total residence time of 30 h. The corresponding concentration of HBu is then 17,26 g/L. This is the maximum concentration of HBu which is possible to achieve for a feed of 50 g/L of substrate. However, the concentration profile for HBu are relatively flat after 20 h. The trade-off between residence time and concentration may speak in favor for selecting a lower residence time, as the concentrations are only changing marginally. As an example, at a total residence time of 20 h, the HBu concentration is 16,28 g/L. As the main objective of this work is to investigate all the attainable concentrations in the system, a maximum concentration of HBu will be referred to as 17,26 g/L, with a corresponding total residence time of 30 h. The residence time of the individual reactors are then 5,51 h for the CSTR and 24,49 h for the subsequent PFR.

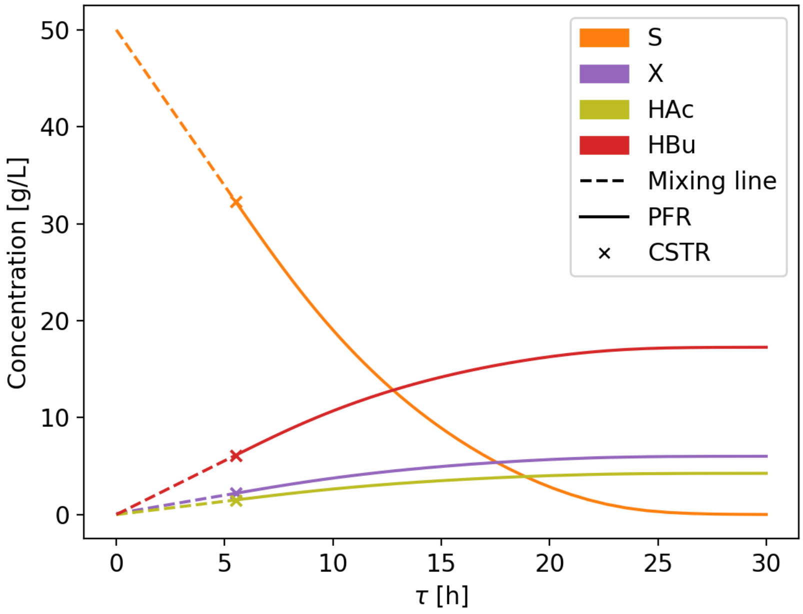


Figure 21: Concentration profiles of all species in the optimal reactor network for the HBu fermentation. The dashed line is the mixing line associated with bypass from the feed to the outlet of the optimal CSTR. The conctration from the optimal CSTR is marked with a cross. This is the feed point to the following PFR, indicated by a solid line. This is consistent with the optimal process design for the HBu fermentation

In section 3.1.2, it was stated that the simplifications of the HBu model was assumed to be valid for residence times lower than 25 h. Here, the maximum residence time is however set to 30 h. This was done in order to ensure that the model went to equilibrium. However, it should be kept in mind that the physical interpretation of the simplification of the HBu model lost its validity for $\tau > 25$ h, as the concentration of substrate in the full model was zero at that time.

5.1.3 AR for BuB Production

In scenario 1, BuB is produced in a reactor network, where the feed is a mix of the outlet flows from the BuOH and HBu fermentation networks. Based on the ARs for the fermentation networks, there are a wide range of entry points to the BuB reactor. The entry points can be visualized by plotting the AR for the fermentation reactors in $C_{BuOH} - C_{HBu} - C_S$-space. This is done in Figure 22.

It is known that the AR for the BuOH fermentation resides in $\mathbb{R}^2$, as there are two independent reaction rates for the system. This plane is made up of two "operating lines"; namely the PFR trajectory and a mixing line, see Figure 15. Along the PFR trajectory, HBu is formed and then later converted to BuOH, as indicated in the model equations and shown in the concentration profile plot, Figure 14. The mixing line is formed by mixing the feed point with the outlet of a PFR operated at equilibrium. At the feed and at equilibrium, there were no HBu in the solution. This explains why there is a straight line from the BuOH fermenter in Figure 22 which always takes the value $C_{HBu} = 0$, and why there is a curved line from the BuOH fermenter which has positive values for C_{HBu}.

The AR for the HBu fermentation was made in $C_S - \tau$-space in section 5.1.2. If only concentrations are considered, then the AR is in fact a one-dimensional construction, hence, it is a line in the state space and it resides in $\mathbb{R}^1$. This is shown in Figure 22 as the green line, which represents the AR from the HBu fermentation in the concentration space. As no BuOH is produced in the fermentation for HBu, the line takes the value $C_{BuOH} = 0$ everywhere.

The convex hull of the ARs from the fermentation network is shown as the semi-transparent red 3D-surface in Figure 22. This convex hull can be formed by mixing states from the AR related to the BuOH fermentation with the AR related to the HBu fermentation. The projection of this plane in the $C_{BuOH} - C_{HBu}$-space

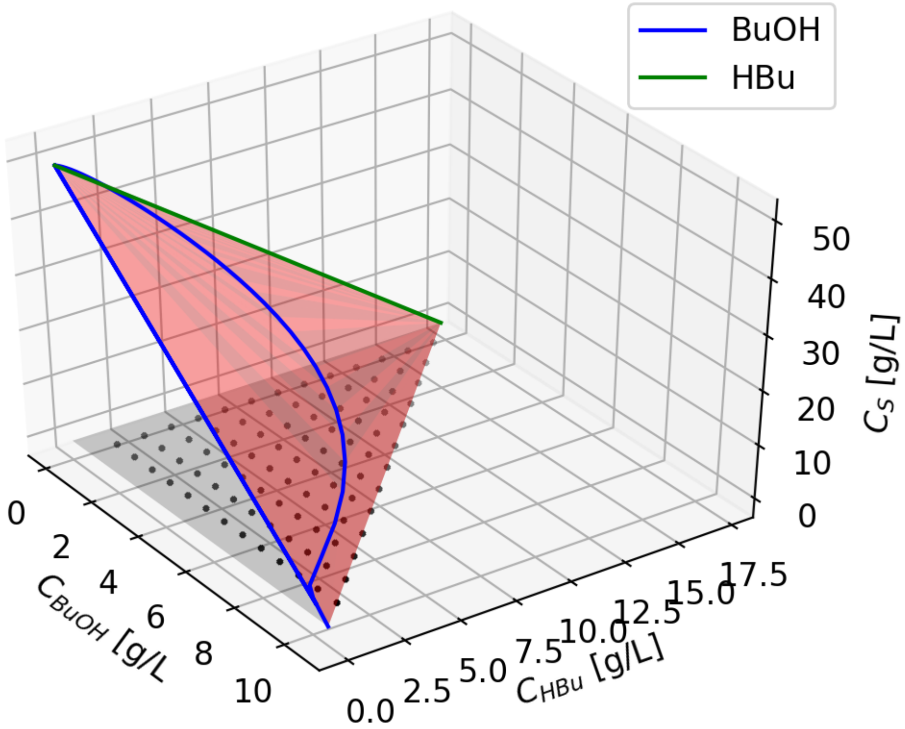


Figure 22: The feed region for the BuB reaction network in scenario 1. The blue lines are the AR from the BuOH fermentation network and the green line is the AR from the HBu fermentation network. The semi-transparent red object is the convex hull from the AR of the HBu and BuOH reactors, which may be formed by mixing points on the outlet of the AR from the BuOH and HBu fermenters. The grey surface is the projection of this convex hull in $C_{BuOH} - C_{HBu}$ space, and the black dots are a grid in that projection.

is shown as the grey surface. From section 3.1.3, it is known that the reaction rates for the enzymatic reaction of BuB is only dependent on the concentrations of BuOH and HBu. The AR for the BuB reactor is therefore unique for every particular feed combination of BuOH and HBu supplied to the BuB model. A 15 × 15-regular grid was made in $C_{BuOH} - C_{HBu}$-space, where the endpoints were specified from 0 to the maximum concentrations in the $C_{BuOH} - C_{HBu}$-projection.

By using the built-in function *inpolygon* in MATLAB, only the grid points which were inside or on the boundary of the projection was included. Points outside the triangle were given the value NaN (not a number), and not included in the further analysis. Also points on the C_{BuOH} or C_{HBu}-axes were excluded, as the reaction rate would be zero along these lines due to that one reactant was missing.

Mixing between the outlet of the BuOH and the HBu fermentation networks are required to physically obtain the grid points in Figure 22. It is therefore of interest to relate the grid points to points on the AR diagrams for the fermentation networks, namely Figure 15 and 20, as well as the required mixing ratios from the outlet of these reactor networks.

First, it is of interest to determine the outlet conditions from the reactor network. From Figure 15, it can be observed that for any sugar concentration, more BuOH is obtained if the configuration with a PFR operated at equilibrium with a bypass of fresh feed is used, than a single PFR. It will be beneficial to use this bypass configuration, as more substrate can be recovered in a separation process downstream of the reactor network (which is not considered in this work) and this lowers the total cost for raw materials. From now on, it will be assumed that the process configuration for the BuOH fermenter is as stated above. This means that there are two straight line sections in Figure 22 which gives the outlet conditions from the fermentation network. It is therefore possible to mathematically describe a plane which gives the substrate concentration as a function of the BuOH and HBu concentration (the grid point). Equation (5.2) describes this relationship. Here, $C_{S,max}$ is the feed to the fermentation network of the substrate, and $C_{BuOH,max}$ and $C_{HBu,max}$ are the equilibrium concentrations of BuOH and HBu out of the fermentation networks. This surface thus coincides with a part of the convex hull in Figure 22.

$$C_S = \frac{1}{C_{S,max}}\left(1 - \frac{C_{BuOH}}{C_{BuOH,max}} - \frac{C_{HBu}}{C_{HBu,max}}\right) \tag{5.2}$$

Every grid point can in this way be assigned to a unique substrate concentration. However, there are infinitely many ways the points which reside on the surface described by Equation (5.2) can be obtained by mixing various outlet states from the BuOH and HBu fermentation networks. This is visualized in Figure

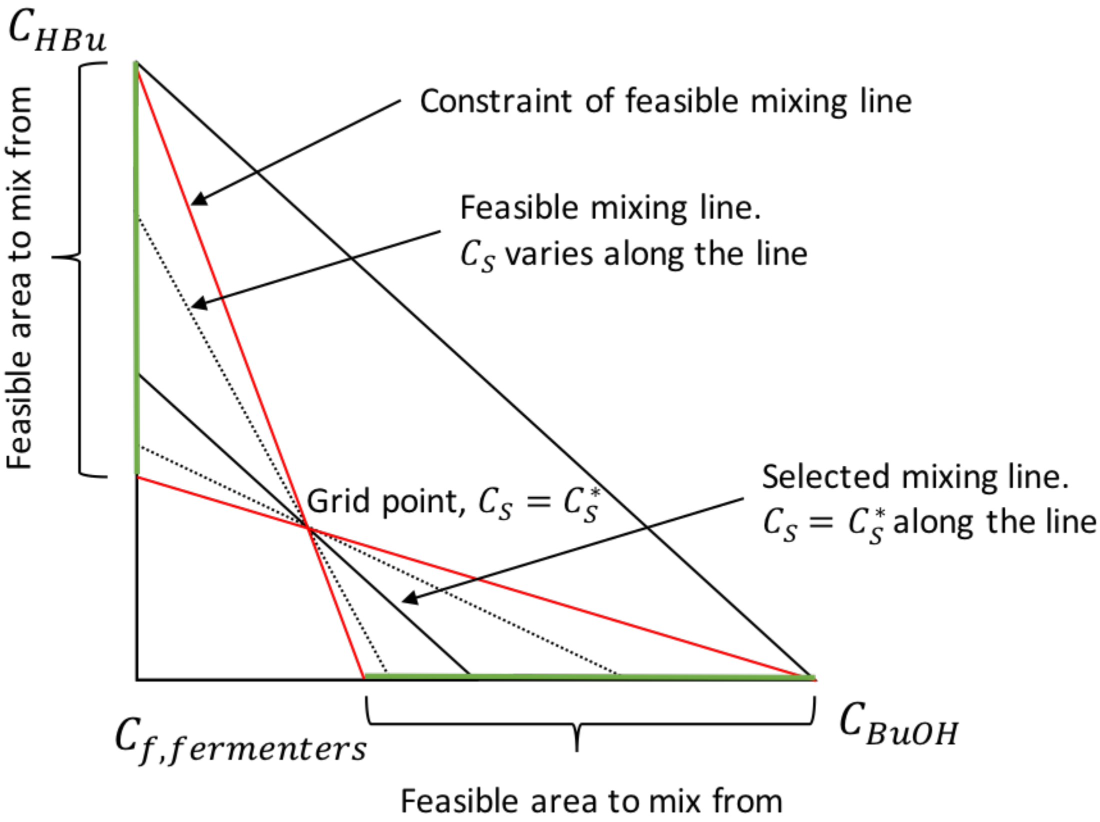

C_{HBu}
Feasible area to mix from
Constraint of feasible mixing line
Feasible mixing line.
C_S varies along the line
Grid point, $C_S = C_S^*$
Selected mixing line.
$C_S = C_S^*$ along the line
$C_{f,fermenters}$
C_{BuOH}
Feasible area to mix from

Figure 23: Selected method to obtain the grid points by mixing outlet concentrations from the fermentation networks. The triangle is the surface described by Equation (5.2). It is possible to obtain the grid point in infinitely many ways by making mixing lines from within the feasible regions, described by the green lines on each axis. However, there is only one line where the substrate concentration is constant throughout the mixing line. This was the selection criteria for obtaining mixing lines for each grid point.

23; the grid point is attainable in infinitely many ways by making mixing lines from within the feasible regions, described by the green lines on each axis. To get a unique solution, it was set as a criteria that the sugar concentration was to be kept constant throughout the mixing line. This implies that the sugar concentration in the outlet of both the BuOH and HBu fermentation networks was assumed to be equal in each grid point. This selection criteria may not give the optimal solution in terms of minimal residence time for the system, but it does provide reactor configurations which makes it possible to obtain each grid point.

As the sugar concentration and the mixing line is known, all points on Figure 24 which are required to calculate the lengths a and b are known. The mixing ratio, λ, for each grid point was then calculated by Equation (5.3).

$$
\begin{aligned}
a &= \sqrt{(C^*_{BuOH} - C_{BuOH,m})^2 + (C^*_{HBu} - 0)^2} \\
b &= \sqrt{(C^*_{BuOH} - 0)^2 + (C^*_{HBu} - C_{HBu,m})^2} \\
\lambda &= \frac{a}{a+b}
\end{aligned}
\tag{5.3}
$$

Here, the mixing ratio is defined as $\lambda = \frac{Q_{HBu}}{Q_{HBu}+Q_{BuOH}}$.

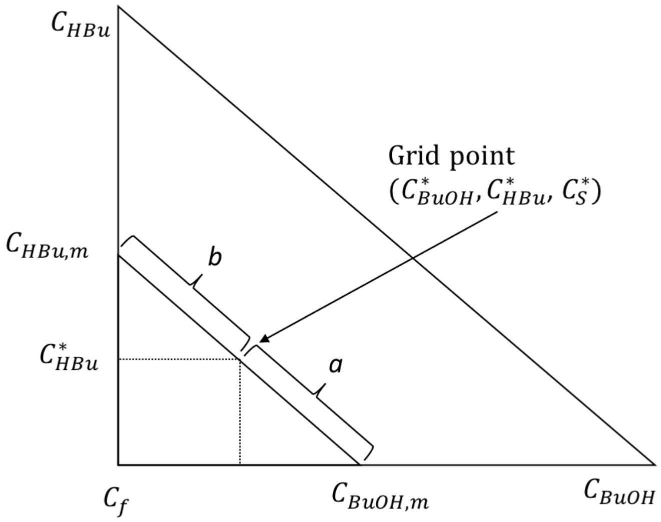


Figure 24: Geometrical method to determine the mixing ratio, λ, for each grid point. The triangle represents the surface described by Equation (5.2).

By following the aforementioned procedure, it was possible to obtain a reactor configuration for both fermentation networks, as well as the required mixing ratio, in order to obtain any grid point in Figure 22.

For a given feed point, it was expected that the AR for the BuB model was a line in the concentration space, as the model has only one concentration as an independent variable. As every grid point in Figure 22 represented different initial conditions to the BuB model described in section 3.1.3, different ARs should arise from each grid point. A grid was translated from a $C_{BuOH} - C_{HBu}$-space in Figure 22 to a $C_{BuOH} - C_S$-space by using Equation (5.2). This made it possible to make the AR plot for every feed point in the $C_{BuOH} - C_S - C_{BuB}$-space. The reason for this choice of variables, was that it made it easy to interpret how the substrate utilization in the fermenters affected the final concentration of BuB. Note that Equation (5.2) connects the grid points in $C_{BuOH} - C_{HBu}$-space to grid points in $C_{BuOH} - C_S$-space uniquely. The result of plotting all the ARs in $C_{BuOH} - C_S - C_{BuB}$-space is shown in Figure 25. The AR which maximized the concentration of BuB was marked with a red line.

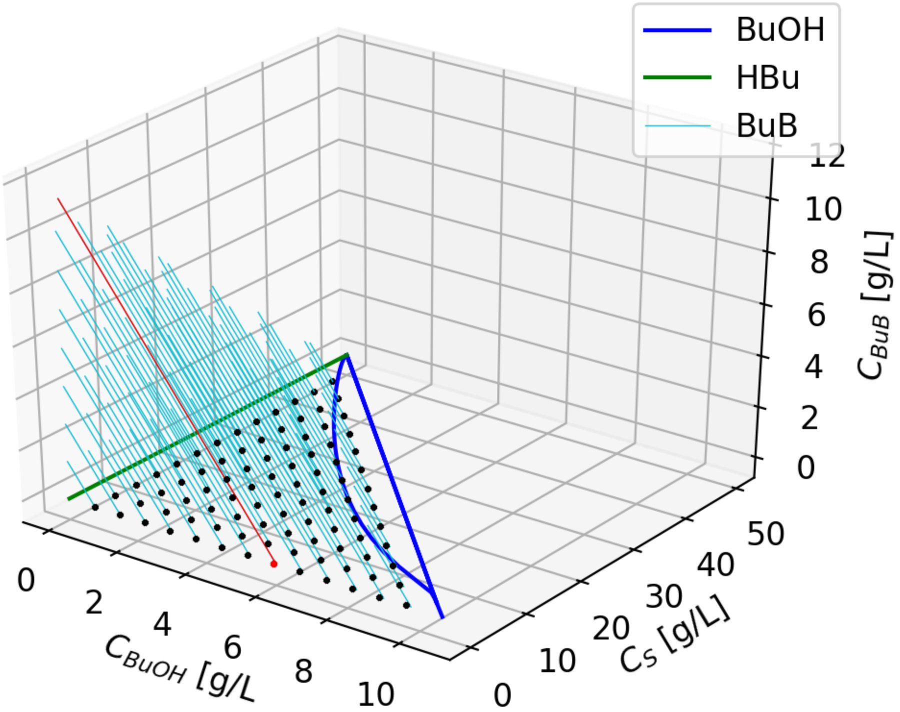


Figure 25: The AR for scenario 1. Separate ARs for the BuB model, based on the initial conditions in the grid, was made. The one which yielded the highest concentration of BuB is marked as a red line.

A 15 × 15 grid was chosen so that it was possible to visualize the grid and the resulting AR in $C_{BuOH} - C_S - C_{BuB}$-space in a clear manner. In order to obtain a more precise solution of optimal operating conditions, also a simulation of a 40×40 grid was done. The resulting contour plot is shown in Figure 26.

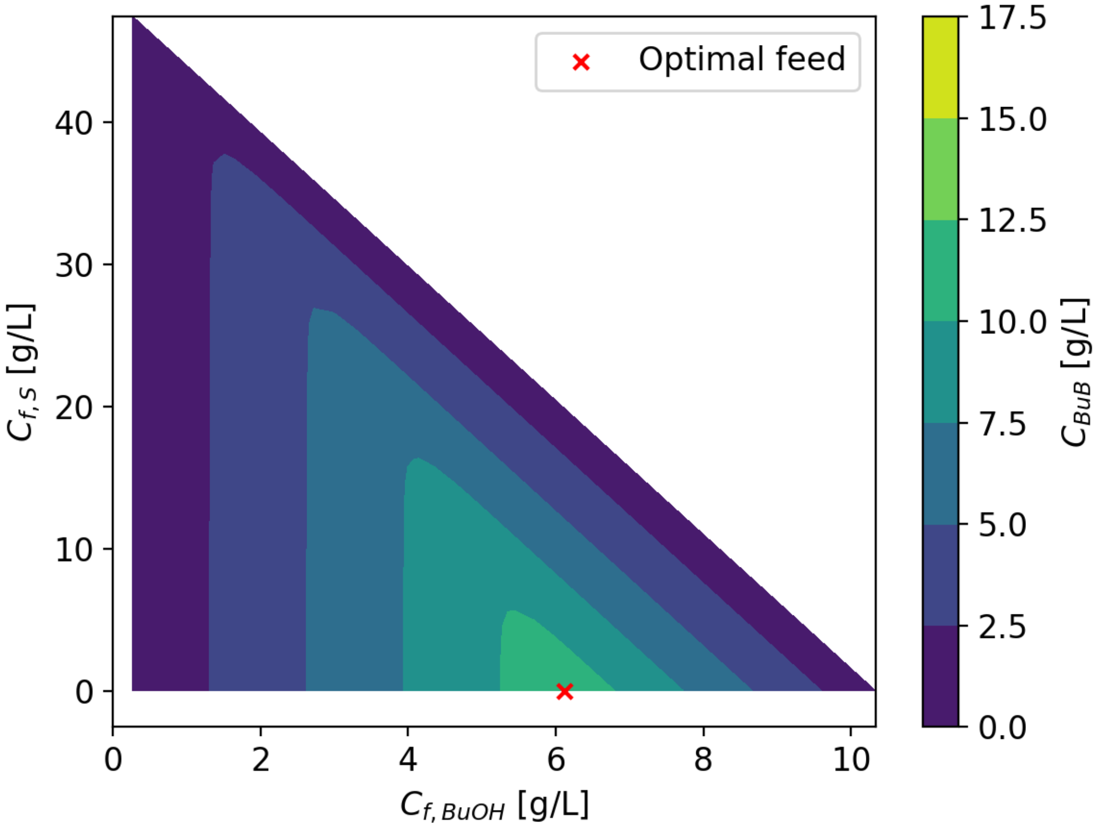


Figure 26: Contour plot of the AR for scenario 1 with a grid of 40 × 40 points. The $x-y$-plane represents possible feed points for the BuB reactor. The height describes the maximum concentration of BuB based on each possible feed point. The optimal feed is marked with a red cross.

From Figure 26, it can be observed that there are areas in the feed region to the BuB reactor which resulted in approximately the same concentration of BuB. As expected, the feed point which gave the highest concentration of BuB was located on the boundary of the convex hull of the feed region in Figure 22. The optimal feeding condition to the BuB reactor was to use 6,12 g/L BuOH and 7,00 g/L HBu.

The optimal mole ratio between the reactants was *not* 1, which was expeceted, but instead $\frac{n_{HBu}}{n_{BuOH}} = 0,96$. Here, n_i is the molar concentration of component i.

By using the method of obtaining λ described by Equation (5.3), the appropiate conditions for operating the BuOH and HBu fermentation network were obtained. Note that as the optimal feeding composition is at the boundary of the convex hull of the feed region in Figure 22, it is actually already given that both fermentation networks need to be operated to their maximum point, which is equilibrium. Hence, the optimal output from the BuOH fermentation network was 10,30 g/L BuOH, and 17.26 g/L of HBu from the HBu fermentation network. These two streams need then to be mixed with the ratio $\lambda = \frac{Q_{HBu}}{Q_{HBu}+Q_{BuOH}} = 0,41$ to obtain the optimal feeding point to the BuB reactor.

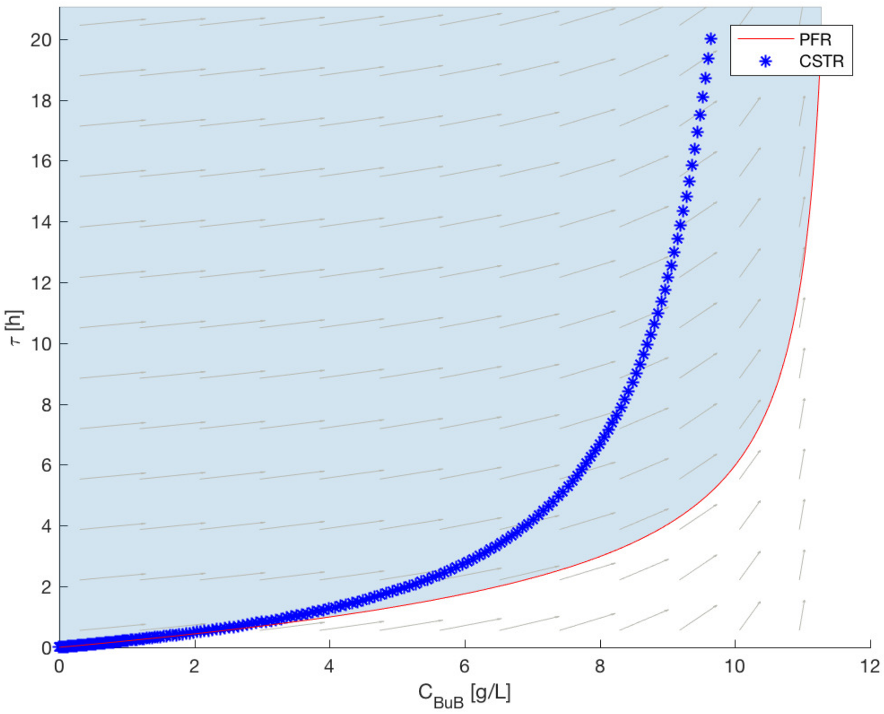


Figure 27: The AR for the BuB reactor with the optimal feed supplied. The red line is the PFR trajectory, and the scattered line is the CSTR locus, both initiated from the feed point. All points on the boundary of the AR is obtained by a PFR.

The AR based on the optimal feed is given in Figure 27. The maximum attainable concentration of BuB, with the process design of scenario 1, was found to be 11,28 g/L. The outlet concentrations of BuOH and HBu are in this case 0,32 g/L and 0,12 g/L, respectively. As it can be seen from Figure 27, all points on the boundary of the AR is obtained by a PFR which is initiated from the optimal feed. The required residence time to obtain equilibrium is 20 h.

The concentration profiles in the BuB reactor, based on the optimal feed, are shown in Figure 28. It is observed that after 10 h, the concentration of BuB is at 10,80 g/L. Hence, in the last 10 h, the concentration of BuB increases by only 0,48 g/L. As the main objective of this work is to investigate what is attainable in terms of outlet concentrations of BuB, a residence time of 20 h and a concentration of 11,28 g/L of BuB will be used as the reference. From a practical point of view, there might be a tradeoff between concentration and cost, implicitly given by the residence time, which speaks in favour of using a lower residence time.

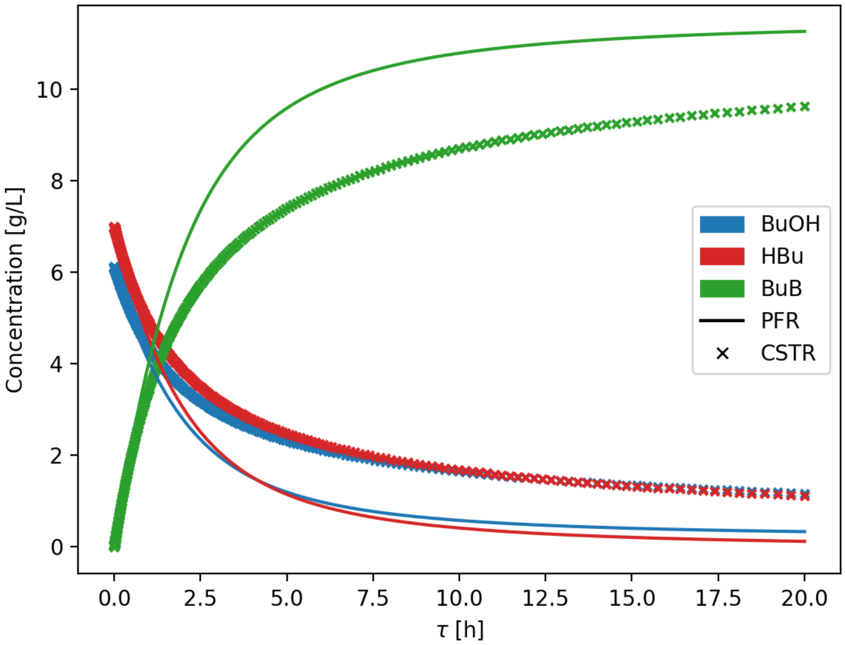


Figure 28: Concentration profiles in the BuB reactor in scenario 1, based on the optimal feed concentrations.

Based on the analysis for the whole scenario 1, Figure 29 shows all the process configurations which are required to operate on the AR boundary for any of the three separate reactor networks. Note that if the optimal concentrationof BuB is to be obtained, the flowrate in the bypass streams indicated in the figure should be set to zero. It was assumed that adding the cells only affect the cell concentrations, hence, stream 1^* and 1^{**} are identical to stream 1, except that the indicated cells have been added.

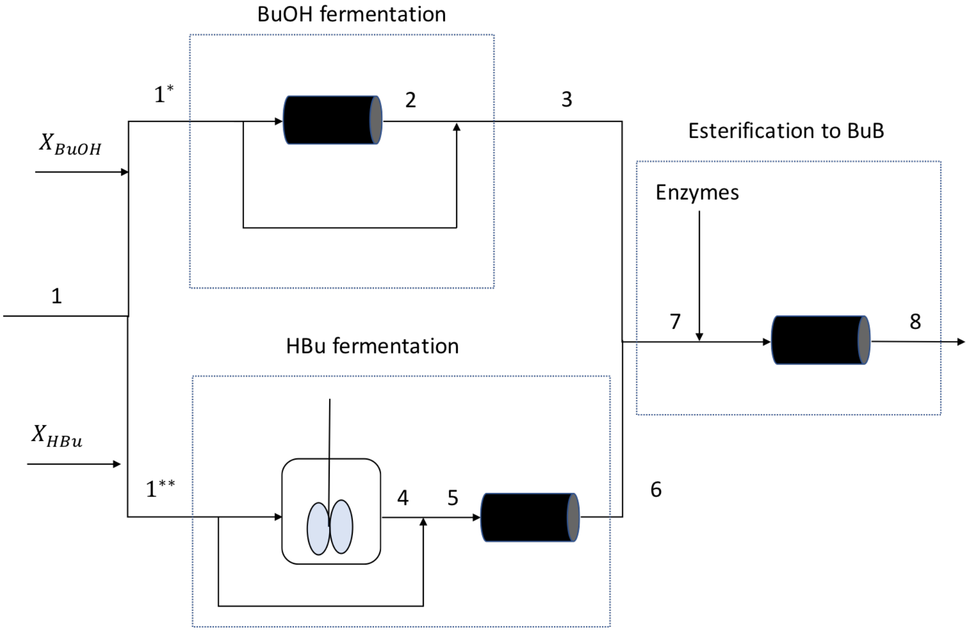


Figure 29: Process configurations in scenario 1 which ensures that the all reactor networks are operated on the boundary of the AR. If the concentration of BuB is to be maximized, then the flowrates in the bypasses should be set to zero.

The required residence times for each reactor in Figure 29 are given in Table 8. These are the residence times and the bypasses which are required if the final concentration of BuB is to be maximized.

Given the residence times and mixing ratios in Table 8, the composition in each flow indicated in Figure 29 can be calculated. These are therefore the concentrations throughout the reactor networks which will result in the maximum BuB concentration. All the concentrations are given in Table 9.

Table 8: Residence times of the reactors in scenario 1 which are required to operate on the AR boundaries. The mixing ratios indicate the ratios which are required to obtain the maximum concentration of BuB in the esterification reaction.

Model	Reactor type	Residence time [h]	Mixing ratio
BuOH fermentation	PFR	12,50	$\lambda = \frac{Q_{bypass}}{Q_3} = 0$
HBu fermentation	CSTR	5,51	$\lambda = \frac{Q_{bypass}}{Q_3} = 0$
	PFR	24,49	
BuB esterification	PFR	20,00	$\lambda = \frac{Q_6}{Q_7} = 0,41$

Table 9: Concentrations in the flows for scenario 1. The streams are indicated in Figure 29 and the reactor network is set to maximize the concentration of BuB.

Flow	C_S [g/L]	C_{BuOH} [g/L]	C_{HBu} [g/L]	C_{BuB} [g/L]	C_{HAc} [g/L]	C_{EtOH} [g/L]	$C_{X_{BuOH}}$ [g/L]	$C_{X_{HBu}}$ [g/L]
1	50	0	0	0	0	0	0	0
1^*	50	0	0	0	0	0	0,1	0
1^{**}	50	0	0	0	0	0	0	0,1
2	0	10,60	0	0	0	0,69	3,10	0
3	0	10,60	0	0	0	0,69	3,10	0
4	32,27	0	6,12	0	1,51	0	0	2,20
5	32,27	0	6,12	0	1,51	0	0	2,20
6	0,01	0	17,26	0	4,26	0	0	6,01
7	0	6,12	7,00	0	1,73	0,41	1,84	2,44
8	0	0,32	0,11	11,28	1,73	0,41	1,84	2,44

5.2 The AR for Scenario 2

The model for scenario 2 is described in section 3.2. The AR analysis for the BuOH fermentation will be as in section 5.1.1. It was assumed that in the feed of fresh substrate, the concentration was 50 g/L, the same as in the feed to the BuOH fermentation network. However, as this point is by definition already included in the AR for the BuOH process, the feed region to the combined reactor cannot be expanded by this fresh substrate stream. This means that all feed points to the BuB reactor is already contained within the AR for the BuOH fermentation model. If the feed region should have been expanded, another concentration of substrate in the fresh feed stream would have been required.

Various feed points to the combined reactor network was obtained by meshing the AR from the BuOH fermenter. The built-in function *inpolygon* in MATLAB was once again used, so that only mesh points inside or on the boundary of the region was included. The meshing generation was more difficult than in scenario 1, as the shape of the AR from the BuOH fermenter is not a usual polygon. Again, a regular grid was used, with endpoints specified from 0 to the maximum concentrations in the $C_{BuOH} - C_S$-projection. From each point in the mesh, the AR for the combined reactor was constructed. Each AR resides in $\mathbb{R}^2$, as there are two independent reaction rates describring the model in section 3.2. As the AR from the BuOH fermentation model and the combined reactor both have C_S as a common independent variable, the AR for the whole scenario 2 was made in $C_{BuOH} - C_S - C_{BuB}$-space. For a regular grid of 30×30 points in the $C_{BuOH} - C_S$-space, the resulting AR for the whole system is shown in Figure 30. This amount of points were chosen for clarity of the figure.

In order to obtain a more precise solution of the optimal CSTR, the grid size was increased to contain 60×60 points[3]. A contour plot which shows the maximum attainable concentration of BuB, given the grid point as the feed, is shown in Figure 31. The blue region is the AR from the BuOH fermenter, which serves as the feed region. White spaces inside this region was not covered by any grid point. A larger grid would have filled out the region even more.

[3] For this grid size, the computation time was around 2 h. Considering this computation time, the accuracy of the solution was found to be satisfactory. No parallell computing was implemented, which would have decreased the compuation time significantly.

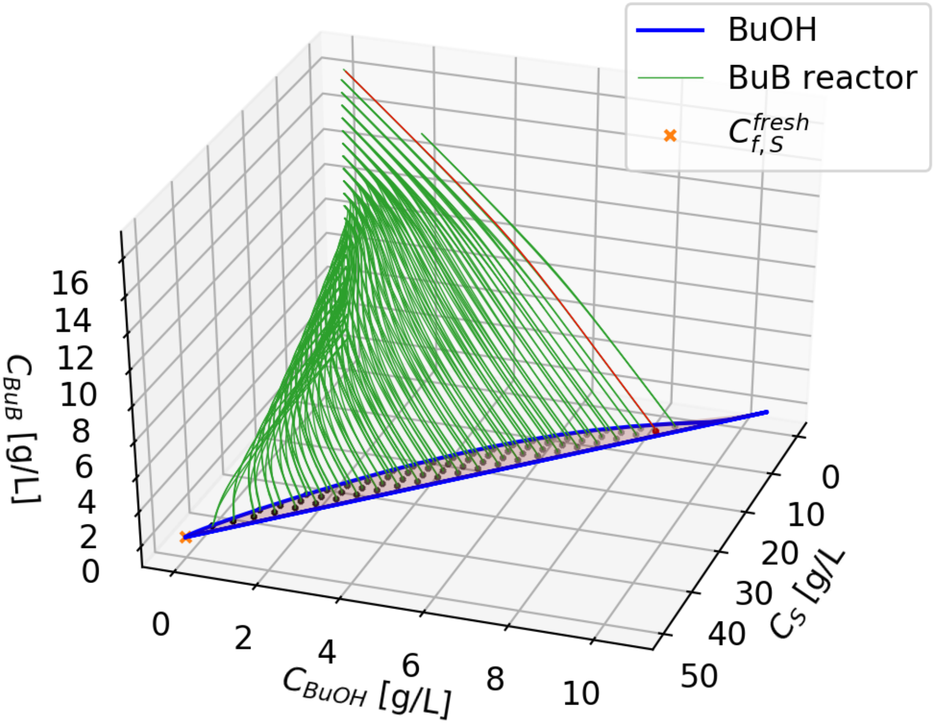


Figure 30: The AR for scenario 2. The blue region are the operating lines from the AR in the BuOH fermentation. The concentration of fresh feed are the same as the feed to the BuOH fermenter, hence, this concentration is contained inside the AR from the BuOH fermenter. The convex hull from this AR is the whole feed region to the combined reactor. A grid was inserted, and from each grid point the AR for the combined reactor was calculated. The AR which obtained the maximum conncetration of BuB is marked with a red line.

It was found that the concentration in the optimal feed was $C_{f,S}^{opt} = 9,32$ g/L and $C_{f,BuOH}^{opt} = 8,45$ g/L. This optimal feed point is marked with a red cross in Figure 31. This feed resulted in a final concentration of BuB of 16,43 g/L. Compared with scenario 1, this process design yields an increase in the maximum attainable concentration of BuB by 45,66 %. It is believed that the increased concentration of BuB is related to the HBu producing cells. As the HBu is reacted to BuB once

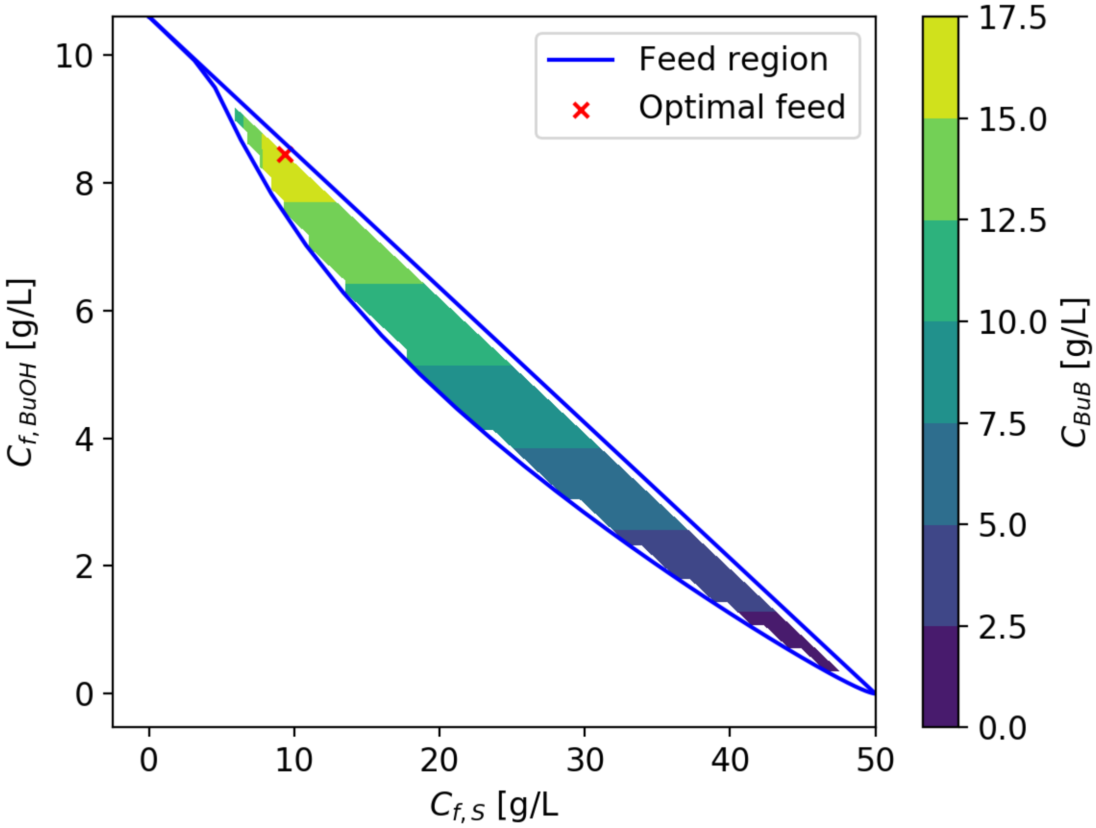


Figure 31: Contour plot of the AR from scenario 2. The x- and y-axis are the feed concentration of substrate and BuOH, respectively, and the z-value denotes the maximum concentration of BuB obtained given these feed points. The optimal input feed is shown by the red cross.

it is produced by the cells, the concentration of HBu is maintained at a low level. This reduces the product inhibition of the cells, which can be seen by inspection of Equation (3.18). The reduced inhibition of the HBu producing bacteria due to a maintained low concentration of HBu, is therefore believed to be the main reason for the increase in concentration.

In order to obtain this concentration, a PFR with bypass is required. The amount of bypass was once again determined by geometrical methods, namely by the familiar Pythagora's equation. The figure which was used to determine the mixing ratio between PFR outlet and the bypass stream is shown in Figure 32.

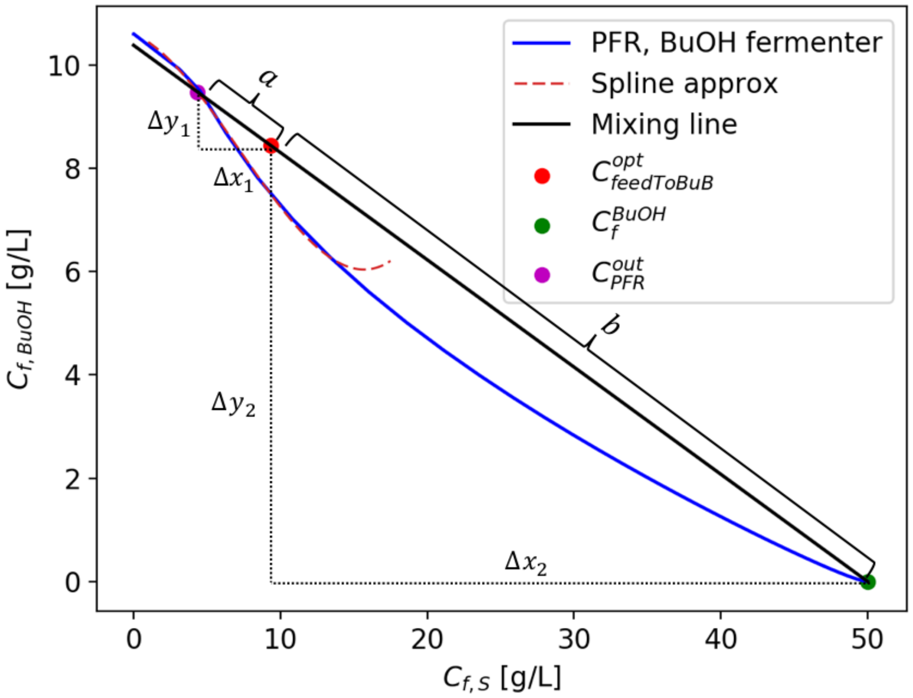


Figure 32: Sketch of the method to determine the mixing ratio between the BuOH producing PFR and its bypass to obtain the optimal feed point to the combined reactor. A part of the PFR trajectory was approximated with a cubic spline, so that a numerical solution could be obtained.

A linear equation for the mixing line on the form $C^{mix}_{BuOH}(C_S) = p + mC_S$ was obtained by using the known information about the indicated points $C^{opt}_{feedToBuB}$ and C^{BuOH}_{f}. An equation for parts of the PFR trajectory was approximated by interpolating with a cubic spline function. The interpolation range was from 0 to 13,4 g/L C_S and the interpolation function was *scipy.interpolate.UnivariateSpline()*.[4] The outlet concentration of the PFR in the BuOH fermenter was then obtained by numerically solving the equation $C^{spline}_{BuOH}(C_S) - C^{mix}_{BuOH}(C_S) = 0$ by using the root-solver *scipy.optimize.fsolve()*. The concentrations in the outlet of the PFR were found to be $C^{out}_{PFR,S} = 4,36$ g/L and $C^{out}_{PFR,BuOH} = 9,48$ g/L. By comparing these val-

[4] Only this part of the PFR trajectory was interpolated, as it was known that the solution would be in this area. A short interpolation range also enhances the fit of the interpolation function in the desired area.

ues with the concentration profiles for the BuOH fermenter, see Figure 13, the residence time of the PFR was found to be 9,9 h. The mixing ratio was then determined by $\lambda = b/(a+b)$, where a and b were calculated by Pythagora's equation, with the triangles indicated in Figure 32. It was found that $\lambda_{sc2,opt} = 0,89$, which means that 11% of the volumetric flowrate to the inlet of the BuOH fermenter should be a bypass, while the rest is sent to the BuOH fermenter.

For the BuOH fermenter as a system, it is interesting to note that the bypass reduces the total residence time. The flow out of the PFR has a residence time of 9,9 h in a reactor, while the flow in the bypass has not been in a reactor, and has therefore a residence time of 0 h. As the residence time obeys a linear mixing law in the same way as a concentration, see section 2.2.1, the total residence time for the BuOH fermenter may be calculated by the lever arm rule, Equation (2.1): $\tau_{tot} = \lambda\tau_{PFR} + (1-\lambda)\tau_{bypass} = 0,89 \cdot 9,9\text{h} + 0,11 \cdot 0\text{h} = 8,81$ h. As an example, if the PFR has a volume of 1 m^3, then it can process a volume flow of $Q_{PFR} = V_{PFR}/\tau_{PFR} = 0,101$ m^3/h. Due to the bypass, the total volumetric flowrate can be $Q_{tot} = V_{tot}/\tau_{tot} = (V_{PFR}+0)/\tau_{tot} = 0,114$ m^3/h.

The calculation of the mixing ratio introduces a substantial degree of uncertainty, as the spline interpolation does not represent the true PFR trajectory. The benefit of this procedure is that it does not require to read off the concentration values from the PFR trajectory. This means that it gives automatically the mixing ratio if a different grid size is used, and a different grid size could mean that the optimal feed point may change slightly.

Based on the optimal feed point, the AR for the combined reactor is shown in Figure 33. The boundary of the AR is formed by a PFR initiated from the optimal feed point, and a mixing line from the feed point to the outlet of the PFR.

In order to obtain the maximum concentration of BuB, a long residence time in the PFR is required. In cases where residence time is a parameter of interest, as in cost estimations, it may be desirable to find a trade-off between concentration of the components and the residence time. The concentration trajectories for BuOH, HBu and BuB for the combined reactor, based on the optimal feeding conditions, are therefore given in Figure 34. Only a residence time up to 80 h were included. The CSTR approaches equilibrium after 1000 h. However, a CSTR has a lower residence time for the same BuB concentration for any value of τ lower than 11,5

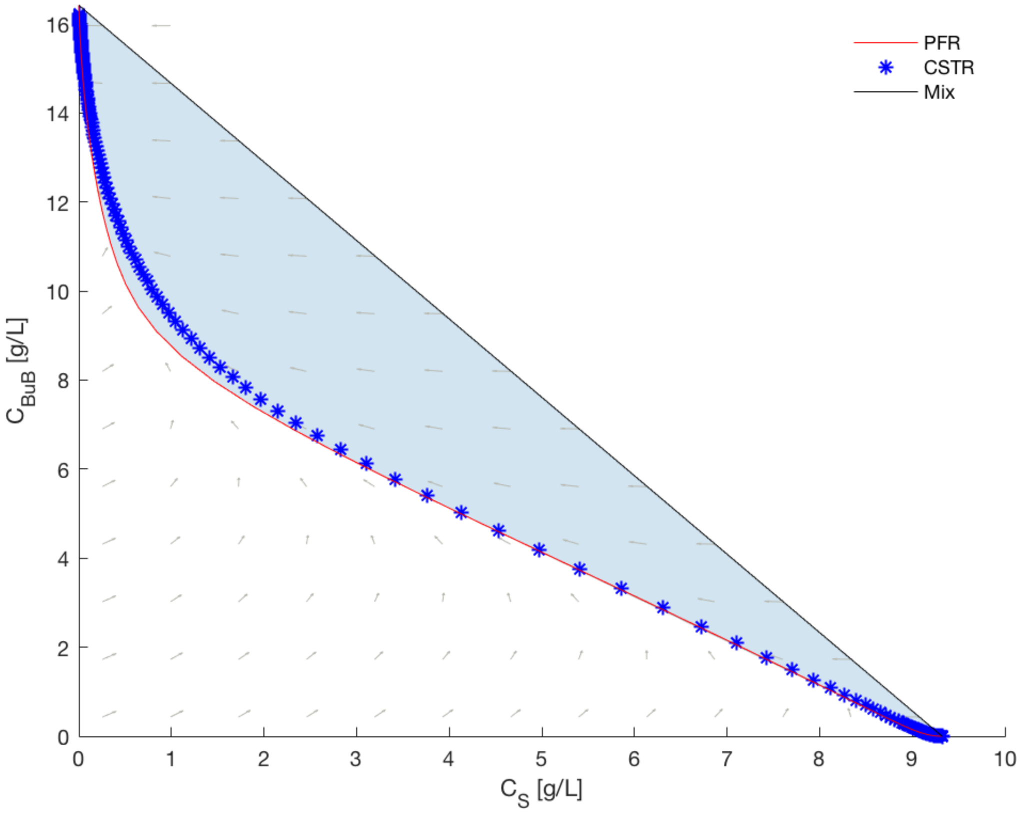


Figure 33: The AR for the combined reactor, given optimal feed concentrations. The red line is a PFR, initiated from the optimal feed, and the scattered blue line is a CSTR with the same feed. The black line is a mixing line from the feed to the outlet concentration of the PFR. The AR boundary is formed by the PFR trajectory and the mixing line.

h. Note that in order to obtain the same concentration of BuB as in the optimized scenario 1, a PFR with a residence time of 11,77 h is required by the process configuration of scenario 2.

The concentration profiles for sugar, biomass and acetic acid are given in Figure 35. Almost all the sugar is consumed after 15 h. From Figure 34, it can be seen that the concentration of both HBu and BuOH are also low after 15 h. This low concentration of reactants to the esterification reaction gives a low reaction rate, which ultimately explains the long residence time to reach equilibrium in the combined reactor. The concentration profiles for BuOH, HBu and the substrate shows

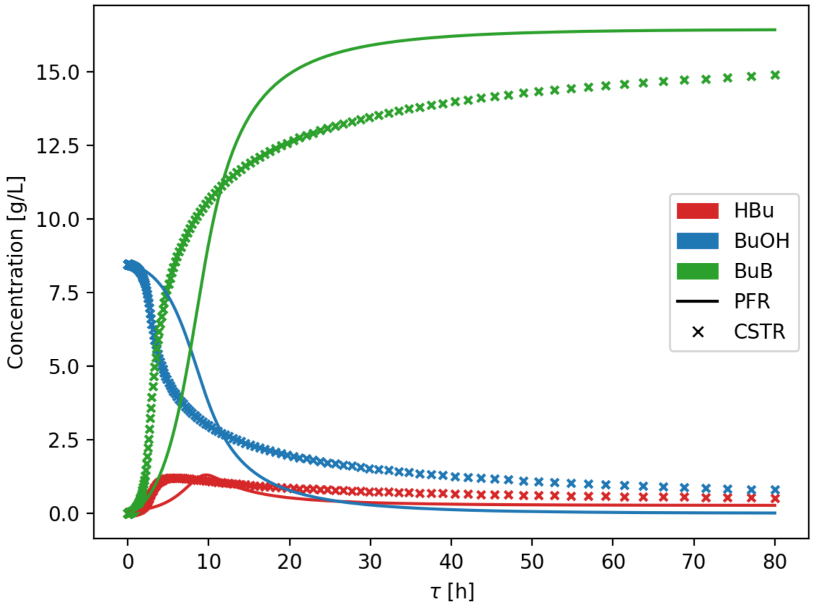


Figure 34: Concentration trajectories for BuOH, HBu and BuB in the combined reactor in scenario 2, based on optimal feed conditions.

that the optimal feed to the combined reactor, is the feed which balances the ratio between substrate and butanol. The optimal ratio ensures that there is just enough substrate left in the liquid to produce the adequate amount of HBu which is required to convert all the BuOH and form BuB, and thereby maximizing the concentration of BuB.

The total process configuration which is required to operate on the AR boundary for scenario 2 is shown in Figure 36. Here, the process configuration is purely sequential. This is due to that the fresh feed of substrate supplied to the combined reactor was assumed to have the same concentration as the substrate feed to the BuOH fermenter. This means that the substrate stream to the combined reactor in Figure 12 is redundant, as that point in the state space is already included in the AR from the BuOH fermenter. Given another concentration of substrate in stream supplied to the combined reactor, the process configuration would have been different.

Table 10 summarizes the conditions which are required to obtain the maxi-

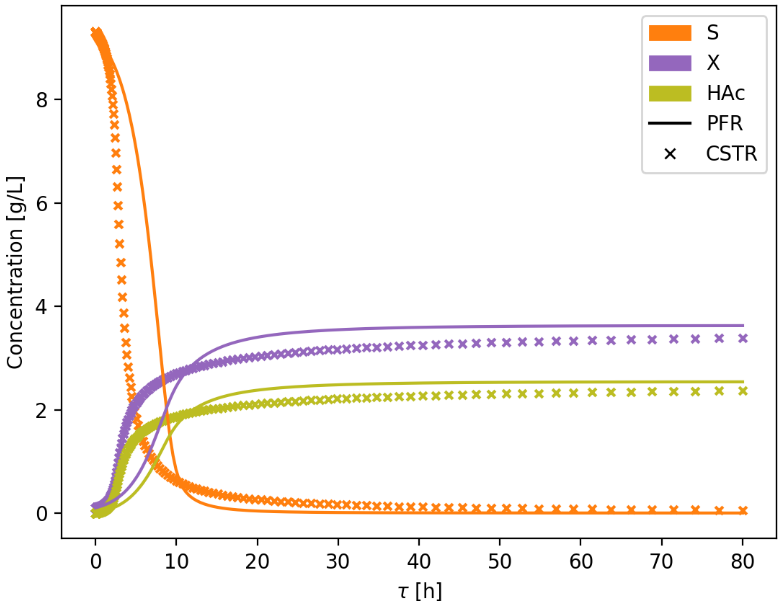


Figure 35: Concentration trajectories for the cells, substrate and HAc in the combined reactor in scenario 2, based on optimal feed conditions.

mum concentration of BuB in scenario 2. Note that the residence time for the combined reactor are the one which is required to approximate the equilibrium

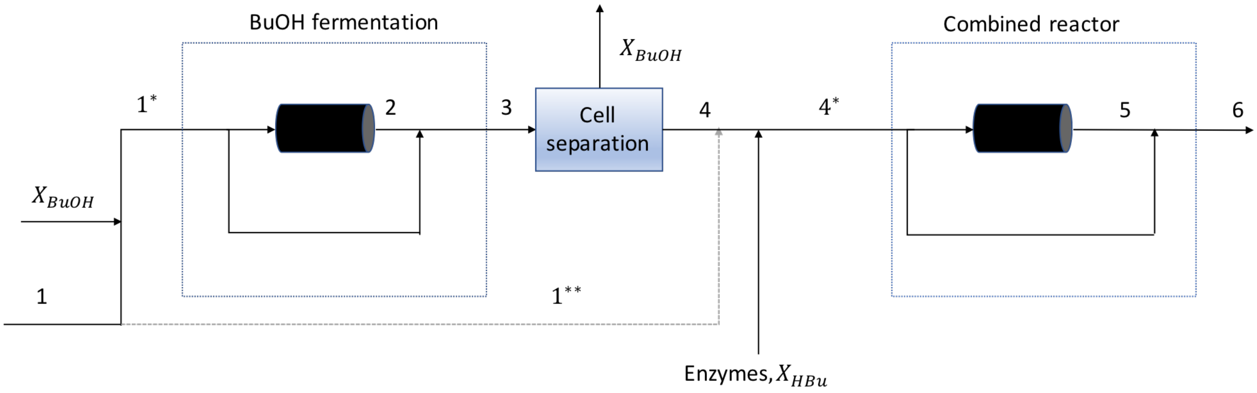


Figure 36: The process configurations which are required to operate on the border of the AR for scenario 2. As it was assumed that the fresh feed of substrate to the combined reactor had the same concentration as the feed concentration to the BuOH fermenter, flow 1^{**} is redundant.

concentration. However, a suitable tradeoff between the residence time and concentration of the components can be obtained by using the concentration profiles for the combined reactor given in Figure 34 and 35. As the main objective in this work deals with what is attainable in terms of concentrations, a residence time of 80 h will be used as the reference value when the residence times of each reactor is reported in Table 10.

Table 10: Residence times and mixing ratios for the reactors in scenario 2. The values of the residence time and mixing ratios are the ones which are required to obtain the maximum concentration of BuB. Note that for the PFR in the BuOH fermentation, it is the residence time for the reactor which is given. The total residence time for the BuOH fermentation system is the product of the given residence time and the mixing ratio. Q_i denotes the volumetric flowrate of stream i, where the number of the stream is shown in Figure 36

Model	Reactor type	Residence time [h]	Mixing ratio
BuOH fermentation	PFR	9,90	$\lambda = \frac{Q_2}{Q_3} = 0,89$
Combined reactor	PFR	80,00	$\lambda = \frac{Q_4}{Q_5} = 0$

The concentrations in each flow indicated in Figure 36 are given in Table 11. These concentrations correspond to the mixing ratios and residence times which are reported in Table 10.

Table 11: Concentrations in the flows for scenario 2. The streams are indicated in Figure 36 and the reactor network is set to maximize the concentration of BuB.

Flow	C_S [g/L]	C_{BuOH} [g/L]	C_{HBu} [g/L]	C_{BuB} [g/L]	C_{HAc} [g/L]	C_{EtOH} [g/L]	$C_{X_{BuOH}}$ [g/L]	$C_{X_{HBu}}$ [g/L]
1	50	0	0	0	0	0	0	0
1*	50	0	0	0	0	0	0,1	0
1**	50	0	0	0	0	0	0	0
2	4,36	9,48	0,45	0	0	0,63	2,84	0
3	9,32	8,45	0,40	0	0	0,56	2,53	0
4	9,32	8,45	0*	0	0	0,56	0	0
4*	9,32	8,45	0	0	0	0,56	0	0,1
5	0	0	0,27	16,42	2,54	0,56	0	3,63
6	0	0	0,27	16,42	2,54	0,56	0	3,63

5.3 Comparison of Scenario 1 and Scenario 2

The main objective in this work was to investigate what the maximum attainable concentration of BuB was, given scenario 1 and scenario 2 as process designs. Table 12 summarizes the oulet concentrations of BuB, BuOH and HBu, as these are the main building blocks in the esterification reaction.

Table 12: Comparison of final concentrations of BuOH, HBu and BuB in scenario 1 and scenario 2.

Species	C_i^{sc1} [g/L]	C_i^{sc2} [g/L]	$\frac{C_i^{sc2}-C_i^{sc1}}{C_i^{sc1}} \times 100$ [%]
BuB	11,28	16,42	+45,57
BuOH	0,32	0	-100
HBu	0,11	0,27	45,45

From Table 12, it can be observed that the process configuration in scenario 2 resulted in an increase of 45,57% in concentration of BuB. In order to describe possible reasons why there is such a great difference, it is interesting to inspect the feed to the BuB producing reactors in both scenarios. In scenario 1, this BuB-producing reactor was the esterification reactor, and in scenario 2 this was the combined reactor.

In both scenarios, BuOH was a given reactant to the reactor producing BuB, and almost all of the BuOH was completely converted to BuB. An interesting parameter which might explain some of the benefit scenario 2 offers is therefore the difference in feed concentration of BuOH in the two scenarios. The feed concentration of BuOH to the BuB producing reactor for both scenarios are given in Table 13.

It can be observed that the feed concentration of BuOH to the last reactor was 38,07% higher in scenario 2 than in scenario 1, even though a larger PFR was used in scenario 1. The reason for this lower BuOH concentration in scenario 1 is therefore due to the mixing with the flow from the HBu fermenter. The mixing process effectively dilutes the concentration of BuOH, which results in that the feed concentration of BuOH to the BuB producing reactor was lower in scenario 1 than in scenario 2. This difference in concentration of BuOH to the BuB-producing

Table 13: Feed concentrations of BuOH to the final reactor model in both scenarios. This corresponds to the concentrations from flow 7 in scenario 1 and from flow 4* in scenario 2. Also the relative change in concentration from scenario 1 to scenario 2 is included.

Species	$C_{f,i}^{sc1}$ [g/L]	$C_{f,i}^{sc2}$ [g/L]	$\frac{C_{f,i}^{sc2}-C_{f,i}^{sc1}}{C_{f,i}^{sc1}} \times 100$ [%]
BuOH	6,12	8,45	+38,07

reactors is therefore believed to be an important factor of why scenario 2 yields a higher final concentration BuB.

Another possible reason for the increased BuB production in scenario 2, is the reduced inhibition of the HBu producing cells. As Figure 34 shows, the concentration of HBu in the combined reactor is always below 1,3 g/L. The cells are therefore not inhibited by high HBu concentrations, as the HBu is simultaneously being converted to BuB. The cells are actually not directly inhibited by HBu itself, but by a low pH in the solution. The concentration of the side product HAc plays therefore an important role. HAc has a pKa value of 4,76, which is close to the pKa value of HBu, which is 4,82. An indication of the degree of product inhibition to the cells can therefore be obtained by evaluating the sum of C_{HBu} and C_{HAc}, as given in Equation (3.18). However, this degree of product inhibition should be viewed in context with the concentration of the substrate which was fed to the HBu producing cells. As this substrate concentration was widely different in the two scenarios, the possibility of the cells to produce the acids were altered as well. From Table 9 and 11, the substrate concentration to the reactor units where HBu was produced (flow 1 in scenario 1 and flow 4* in scenario 2) was 50 g/L for scenario 1 and 9,32 g/L in scenario 2. The decrease in substrate concentration for scenario 2 was therefore 81,36%. The decrease in product inhibition are given in Table 14.

From Table 14, the decrease in product inhibition was 86,94% in scenario 2, while the decrease in available sugar for cells to grow on was 81,36%. However, the cells are also inhibited by high sugar concentrations, expressed by the term $\frac{C_S^2}{K_I}$ in Equation (3.17). As the system is complex, and both the substrate concentration fed to the HBu fermenter and the kinetic model itself was varied between

Table 14: Outlet concentration of HBu and HAc from the reactor unit where HBu fermentation takes place in both scenarios. *P* denotes the total acid concentration, equivalent to the sum of HBu and HAc. This corresponds to the concentrations from flow 6 in scenario 1 and from flow 5 in scenario 2. Also the relative change in the values from scenario 1 to scenario 2 is included.

Species	C_i^{sc1} [g/L]	C_i^{sc2} [g/L]	$\frac{C_{f,i}^{sc2}-C_{f,i}^{sc1}}{C_{f,i}^{sc1}} \times 100$ [%]
HBu	17,26	0,27	-98,44
HAc	4,26	2,54	-40,38
P	21.52	2.81	-86,94

the two scenarios, it is difficult to isolate and quantify the effect of each inhibition phenomenon. However, it is believed that the product inhibition to the HBu producing bacteria was decreased significantly, which causes the rate of reaction to be faster and residence time to decrease.

6 Validity of the Analysis

In the AR analysis for the system, several assumptions and simplifications were made which raises questions about the validity of the analysis. The main assumptions are

- Constant density in the system
- Isothermal conditions
- The liquid solution is a pseudo-phase

Each of these assumptions will be discussed in the following sections. In addition, there were made simplifications of the system which affects the uncertainty of the obtained solution in various degrees. The most prominent sources for uncertainty of the obtained results, as well as the general assumptions, will be discussed in this section.

6.1 Constant Density Assumption

In order to have a linear mixing law in concentration space, constant density for the liquid was assumed. When mixing two liquids, the mass is as always conserved. The total volume of the mixture may however not be equal to the sum of the two separate original volumes, due to the mixing effect between different components. This effect is related to the partial molar volumes of each component, and the composition in the mixture. Especially, the mixture of sugar in water is known to vary a lot in density. In this work, a maximum substrate concentration of 50 g/L was used. The density of such a mixture is 1,02 g/mL, while the density of pure water is 0,99 g/mL [42]. In the case of maximum sugar concentration in the water, the deviation in density was therefore approximately 2% . Other mixtures which are known to vary in density are ethanol in water. However, as Table 9 and 11 shows, the ethanol concentrations in all flows were found to be maximum 0,69 g/L. The effect of ethanol on the density was therefore considered to be negligible.

As a high substrate concentration was considered to be the parameter which affected the density of the solution the most, a maximum increase in density of approximately 2% is considered to be tolerable. The constant density assumption is therefore believed to be valid for this system.

6.2 Isothermal Conditions in the Reactors

In AR analysis, a common simplification is to assume isothermal reactors. This is not a necessary assumption to do an AR analysis, but it simplifies the procedure. In the fermentation reactors, the assumption of isothermal conditions are considered to be valid, as temperature control is usually applied in such systems. The parameters in the rate equations for the fermentation reactions were also given for 37 °C, hence, there will be no temperature differences in flow 3 and 6 in scenario 1. The temperature in flow 6 may therefore only be affected by the increase in enthalpy due to mixing, also known as heat of mixing. For liquids, the heat of mixing can potentially be large, especially for strong mineral acids and alkalis [12]. The heat of mixing was neglected in this work, and it is believed to be a reasonable assumption as no mineral or alkali acids were included.

The kinetic parameters for the esterification reaction was reported for a temperature of 30°C. In scenario 1, it is possible to think that flow 7 is cooled from the fermentation temperature of 37°C to 30°C. For scenario 2, the kinetics for the HBu fermentation and BuB esterification was combined. This poses a conflict, as the kinetic parameters are reported for different temperatures. No information about the temperature dependency for the HBu fermentation kinetics were found [9], but the kinetics for the BuB esterification are shown to be dependent of the temperature. For a solution consisting of 0,01 mol of n-butanol and 0,01 mol HBu, diluted to a 15 mL solution with heptane, the conversion after 7 h were reported to be 56% and 68% for a temperature of 30°C and 40°C, respectively [8]. The kinetic parameters in the model were however only reported for 30°C, and this may be a significant source of error in this model. A suggestion for future work is therefore to obtain kinetic models which are based on the same temperature.

6.3 Pseudo-Phase in the Liquid

The kinetic model of BuB is describing how the reaction is proceeding in an organic solution. Specifically, heptane is used as a solute [8]. This is clearly breaking with the modeling framework which has been used, where it has been assumed that the kinetic model for BuB is valid for the liquid supplied to the reactor producing BuB. As this liquid clearly is not an organic solution, it is questionable wheter the results in section 5 are reliable or not.

In this work, it has in effect been assumed that the liquid is a pseudo-phase. The liquid is consisting of water, organic compounds and two different cell types. The idea of cells floating around in the fermenters is perhaps the most striking example of a component which makes the liquid a pseudo-phase. Clearly, the cells are solid particles which are not completely dissolved into the liquid. The organic compounds in the system are the alcohols and the acids. By considering BuOH as an example, the maximum concentration throughout the reactor network is 10,60 g/L, which is obtained in the BuOH fermenter in scenario 1. The solubility of BuOH in pure water is reported to be 63,2 g/L [43]. Hence, BuOH is completely dissolved into the liquid phase and it does not form a separate "layer" in the solution. This is also the case with the rest of the organic compounds. It is therefore expected that the liquid has properties which resembles aqueous solutions, but at the same time it also has some characterisitcs from its organic compounds. This mixture of characteristics therefore leads to the assumption that the liquid can be characterized as a pseudo-phase.

To the authors knowledge, there are no information about how the esterification reaction behaves in a pseudo-phase such as the one described above. Therefore, the kinetic models were taken *as is* and the system was analyzed based on this. A kinetic model for the esterification reaction in such a pseudo-phase is a recommendation for future work in the EcoLodge project.

Clearly, there are problems regarding the esterification reaction when treating the liquid as a pseudo-phase. Care should be taken when using the results in section 5, as parts of the applied model equations are based on process conditions which are clearly different than the conditions used in this work. The model also does not take into consideration separation processes, and it is plausible that there exist a separation process which may transfer BuOH and HBu from the pseudo-phase to a purely organic phase. In this case, at least the results from scenario 1 are plausible. For scenario 2, this is not possible due to that fermentation and esterification happens in the same reactor. The implications of the pseudo-phase approximation should therefore always be considered when interpreting the results in section 5, and especially for scenario 2.

6.4 Simplification of the HBu Model

In section 3.1.2, a simplification of the original HBu model was done in order to reduce the dimensionality of the problem. By comparing the full HBu model with the simplified model, it was shown in Figure 11 that the relative error was always less than 10,3% for any residence time lower than 25 h. The reason for this constraint on the residence time was that the substrate in the full model was depleted after 25 h. It was argued that the deviation of the simplified model was acceptable, especially as it consequently underestimates the concentration of HBu and it is therefore a conservative estimate.

However, from Table 8 it can be seen that the residence time in the HBu fermenter was above the predefined limit of 25 h. This violation of the constraint was done in order to approximate equilibrium in the simplified model. All the concentration values from the HBu fermentation are indeed attainable, but the optimal values for the residence times in the HBu fermentation should be viewed best as a guidance for the reactor optimization. Considerable uncertainty for these residence time values are inherent, as the simplified model loses its predictive capabilities after 25 h. This is shown by Figure 10.

A suggestion for future work is to do an AR analysis, without simplifying the HBu model. The benefit of doing an AR analysis based on the full model is that it will be investigated what is theoretically achievable in such a fermentation process. However, the construction and meshing of the feed region to the BuB model will increase in complexity. It is also believed that the practical utility of such an analysis is limited, as the residence time may be prohibitely large, as discussed in section 3.1.2.

6.5 Optimal Feed for BuB Production

When searching for the optimal feed for the BuB production, a regular grid was made. The grid was based on the maximum concentration of each component in the selected projection. For scenario 1, the projection was in $C_{BuOH} - C_{HBu}$-space, and in scenario 2 the grid was in $C_S - C_{BuOH}$-space. Only the grid points which were inside the projection of the feed region was selected, and these grid points represented unique, attainable feed concentrations to the reactor producing BuB. This approach resulted in the surfaces of solutions visualized Figure 26 and 31. This approach is an exhaustive search through all possible combinations of feed points, and it is therefore a brute force method.

If computational time and extreme accuracy of the solution had been critical parameters, the search for the optimal feed concentration would have been improved by selecting another method. Common optimization methods are gradient based, and given a well-conditioned problem, these algorithms typically outperform a brute force approach in both computational time and accuracy of the solution.

In this work, it is however interesting to observe the full surface of the solutions. This surface gives additional information to the designer of the reactor network. There might be areas in the feed region which yields approximately the same solutions as what was found to be the optimal point. In such a case, other parameters may also be decision variables for the designer. A typical example of such a decision is the capital cost of the reactors, where the residence time of the reactor plays an important role. In addition, there are a lot of uncertainty in this model, partly due to the assumptions made and partly due the difficulty of building both simple and accurate models of biological systems. The uncertainty of an optimal feed point will therefore in any case be high, and it is more interesting to observe which areas in Figure 26 and 31 that results in high final concentrations of BuB.

7 Conclusion

Two process configurations for a biotechnological production of butyl butyrate have been investigated. An attainable region analysis was performed on both scenarios, with the objective of determining the limits of the maximum reachable concentration of butyl butyrate. In both scenarios, a feed of 50 g/L pure substrate was supplied to the system.

In the first scenario, there were two separate fermentation reactor networks producing butanol and butyric acid, respectively. These flows were mixed and fed to a reactor, where butyl butyrate was formed. By evaluating all possible feed combinations to the last reactor, the maximum attainable concentration of butyl butyrate was found to be 11,28 g/L with this process configuration.

In the second scenario, a fermentation reactor producing butanol was modeled. The outlet flow was fed to a reactor structure where two reactions happened simultaneously: butyric acid was formed by fermentation of the remaining substrate, while it was simultaneously reacting with butanol to form butyl butyrate. The maximum attainable concentration of butyl butyrate was found to be 16,42 g/L with this process configuration.

One possible reason for the increase in concentration of butyl butyrate with the second process configuration, is the higher inlet concentration of butanol to the reactor producing butyl butyrate. In the first scenario, the concentration of both reactants out of the fermenters were diluted due to the mixing process. In the second scenario, an optimal balance between the inlet butanol concentration and the sugar for the cells producing butyric acid was found. This butanol concentration was higher than resulting butanol concentration after the mixing in the first scenario. Another possible reason for the increase in butyl butyrate concentration in the second scenario is that the butyric acid concentration is maintained low. This reduces the product inhibition of the cells producing butyric acid.

A sub-goal in this work was to separately optimize the fermentation networks for the production of butanol and butyric acid. For butanol, a maximum concentration of 10,60 g/L was obtained by using a single PFR. For butyric acid, a combination of a CSTR followed by a PFR gave a maximum concentration of 17,26 g/L.

7.1 Recommendations for Future Work

It is questionable whether or not these results are valid, as the kinetic model for butyl butyrate production is valid in an organic solvent. This problem are discussed more thoroughly in section 6.3. Development of a kinetic model for the esterification reaction in a liquid which resembles the fermentation broths are a recommendation for future work.

Another recommendation is to investigate how the AR changes for different inlet conditions. Especially, if the feed of fresh substrate had a higher concentration of sugar to the combined reactor in scenario 2, then the feed region to the combined reactor, depicted as the shaded yellow surface in Figure 30, would have increased. It would have been interesting to observe how the maximum attainable concentration of BuB had changed in such a scenario.

Finally, other process designs may yield better results. One possibility is to have first a separate fermentation of HBu, followed by a combined reactor where BuOH is produced by fermentation and simultaneously converted to BuB by the esterification reaction.

References

[1] A. Demirbas, “Biorefineries: Current activities and future developments,” *Energy Conversion and Management*, vol. 50, no. 11, pp. 2782–2801, 2009. [Online]. Available: http://dx.doi.org/10.1016/j.enconman.2009.06.035

[2] A. C. Kokossis and A. Yang, “On the use of systems technologies and a systematic approach for the synthesis and the design of future biorefineries,” *Computers and Chemical Engineering*, vol. 34, no. 9, pp. 1397–1405, 2010. [Online]. Available: http://dx.doi.org/10.1016/j.compchemeng.2010.02.021

[3] Miljodirektoratet, “Norwegian Environment Agency- biofuels in cars,” 2017. [Online]. Available: http://www.miljodirektoratet.no/no/Nyheter/Nyheter/2017/Februar-2017/Fakta-om-biodrivstoff1/

[4] A. C. Kokossis, M. Tsakalova, and K. Pyrgakis, “Design of integrated biorefineries,” *Computers and Chemical Engineering*, vol. 81, pp. 40–56, 2014. [Online]. Available: http://dx.doi.org/10.1016/j.compchemeng.2015.05.021

[5] C. J. Chuck, H. J. Parker, R. W. Jenkins, and J. Donnelly, “Renewable biofuel additives from the ozonolysis of lignin,” *Bioresource Technology*, vol. 143, pp. 549–554, 2013. [Online]. Available: http://dx.doi.org/10.1016/j.biortech.2013.06.048

[6] J. P. Lange, R. Price, P. M. Ayoub, J. Louis, L. Petrus, L. Clarke, and H. Gosselink, “Valeric biofuels: A platform of cellulosic transportation fuels,” *Angewandte Chemie - International Edition*, vol. 49, no. 26, pp. 4479–4483, 2010.

[7] M. N. Varma and G. Madras, “Kinetics of synthesis of butyl butyrate by esterification and transesterification in supercritical carbon dioxide,” *Journal of Chemical Technology & Biotechnology*, vol. 83, no. May, pp. 1135–1144, 2008.

[8] G. D. Yadav and P. S. Lathi, “Kinetics and mechanism of synthesis of butyl isobutyrate over immobilised lipases,” *Biochemical Engineering Journal*, vol. 16, no. 3, pp. 245–252, 2003.

[9] H. Song, M. H. Eom, S. Lee, J. Lee, J. H. Cho, and D. Seung, “Modeling of batch experimental kinetics and application to fed-batch fermentation of Clostridium tyrobutyricum for enhanced butyric acid production,” *Biochemical Engineering Journal*, vol. 53, no. 1, pp. 71–76, 2010. [Online]. Available: http://dx.doi.org/10.1016/j.bej.2010.09.010

[10] E. A. Buehler and A. Mesbah, “Kinetic study of acetone-butanol-ethanol fermentation in continuous culture,” *PLoS ONE*, vol. 11, no. 8, pp. 1–22, 2016.

[11] C. Birgen, H. A. Preisig, A. Wentzel, and S. Markussen, “Anaerobic Bioreactor Modeling,” *Proceddings of the 26th European Symposium on Computer Aided Process Engineering*, 2016.

[12] R. Sinnott and G. Towler, “Chapter 10.13: Reactors,” in *Chemical Engineering Design*, 5th ed. Butterworth- Heinemann, 2009, pp. 662–668.

[13] S. Skogestad, “Sammenlikning av reaktortyper,” in *Prosessteknikk masse- og energibalanser*, 2nd ed. Trondheim: Tapir Akademisk Forlag, 2003, ch. 10.2.4, pp. 210–212.

[14] A. C. Kokossis and C. A. Floudas, “Optimization of Complex Reactor Networks - I. Isothermal Operation,” *Chemical Engineering Science*, vol. 45, no. 3, pp. 595–614, 1990. [Online]. Available: http://www.sciencedirect.com/science/article/pii/000925099087004C{\% }5Cnpapers3://publication/doi/10.1016@0009-2509(90)87004-c

[15] V. L. Mehta and A. C. Kokossis, “Nonisothermal synthesis of homogeneous and multiphase reactor networks,” *AIChE Journal*, vol. 46, no. 11, pp. 2256–2273, 2000. [Online]. Available: http://doi.wiley.com/10.1002/aic.690461117

[16] D. Ming, D. Glasser, D. Hildebrandt, B. Glasser, and M. Metzeger, *Attainable Region Theory: An Introduction to Choosing an Optimal Reactor*, 1st ed. Hoboken, New Jersey: John Wiley & Sons, Inc., 2016.

[17] M. Feinberg, “Optimal reactor design from a geometric viewpoint. Part II. Critical sidestream reactors,” *Chemical Engineering Science*, vol. 55, no. 13, pp. 2455–2479, 2000.

[18] D. Glasser, C. M. Crowe, and D. Hildebrandt, “A geometric approach to steady flow reactors: the attainable region and optimization in concentration space,” *Industrial & Engineering Chemistry Research*, vol. 26, no. 1980, pp. 1803–1810, 1987. [Online]. Available: http://pubs.acs.org/doi/abs/10.1021/ie00069a014

[19] F. Horn, “Attainable and Non-Attainable Regions in Chemical Reaction Technique,” *Third European Symposium on Chemical Reaction Engineering*, pp. 1–10, 1964.

[20] S. Kjeldstrup, *Fysikalsk Kjemi*, 2nd ed. Fagbokforlaget Vigmostad & Bjørke AS, 2009.

[21] M. Feinberg and D. Hildebrandt, “Optimal reactor design from a geometric viewpoint- I. Universal properties of the attainable region,” *Chemical Engineering Science*, vol. 52, no. 10, pp. 1637–1665, 1997.

[22] M. Feinberg, “Optimal reactor design from a geometric viewpoint - III. Critical CFSTRs,” *Chemical Engineering Science*, vol. 55, no. 17, pp. 3553–3565, 2000.

[23] G. A. Venema, *Foundations of Geometry*, 2nd ed. Boston: Pearson Education, 2012.

[24] V. Guillemin and A. Pollack, *Differential Topology*, 1st ed. Englewood Cliffs, N.J: AMS Chealse Publishing, 1974.

[25] R. C. Qiu, Z. Hu, H. Li, and M. C. Wicks, “Convex Optimization,” in *Cognitive Radio Communications and Networking*, 1st ed. John Wiley & Sons, 2012, ch. Convex Opt, pp. 235–282.

[26] J. van de Vusse, “Plug-flow type reactor versus tank reactor,” *Chemical Engineering Science*, vol. 19, no. 12, pp. 994–996, 1964.

[27] D. Hildebrandt and D. Glasser, “The attainable region and optimal reactor structures,” *Chemical Engineering Science*, vol. 45, pp. 2161–2168, 1990.

[28] M. Feinberg, “Recent results in optimal reactor synthesis via attainable region theory,” *Chemical Engineering Science*, vol. 54, no. 13-14, pp. 2535–2543, 1999.

[29] S. H. Fogler, *Elements of Chemical Reaction Engineering*, 4th ed. Pearson Education, 2006.

[30] J. Polking, A. Bogges, and D. Arnold, *Differential Equations with Boundary Value Problems*, 2nd ed. Pearson Education, 2006.

[31] J. Votruba, B. Volesky, and L. Yerushalmi, “Mathematical model of a batch acetone–butanol fermentation,” *Biotechnology and Bioengineering*, vol. 28, no. 2, pp. 247–255, 1986.

[32] G. Aylward and T. Findlay, *SI Chemical Data*, 6th ed. Milton: John Wiley & Sons Australia Ltd., 2008.

[33] P. Van Bodegom, “Microbial maintenance: A critical review on its quantification,” *Microbial Ecology*, vol. 53, no. 4, pp. 513–523, 2007.

[34] C. Analytics, “Continuum Analytics- Anaconda distribution of Python,” 2017. [Online]. Available: https://www.continuum.io/

[35] N. Developers, “NumPy,” 2017. [Online]. Available: http://www.numpy.org/

[36] S. Developers, “SciPy,” 2017. [Online]. Available: https://www.scipy.org/

[37] S. D. Team, “SymPy,” 2016. [Online]. Available: http://www.sympy.org/en/index.html

[38] J. Hunter, D. Dale, E. Firing, and M. Droettboom, “Matplotlib,” 2016. [Online]. Available: https://matplotlib.org/

[39] A. C. Hindmarsh, “LSODA- odepackage in FORTRAN,” 2006. [Online]. Available: https://computation.llnl.gov/casc/odepack/

[40] J. J. Moré, B. S. Garbow, and K. E. Hillstrom, *User Guide for MINPACK-1*, Argonne, 1980. [Online]. Available: http://www.mcs.anl.gov/{~}more/ANL8074a.pdf

[41] B. Barber, “Qhull- Computation of Convex Hull,” 1995.

[42] A. F. Fucaloro, Y. Pu, K. Cha, A. Williams, and K. Conrad, “Partial molar volumes and refractions of aqueous solutions of fructose, glucose, mannose, and sucrose at 15.00, 20.00, and 25.00°C,” *Journal of Solution Chemistry*, vol. 36, no. 1, pp. 61–80, 2007.

[43] PubChem, “Solubility of Butanol in Water,” 2017.

[44] W. Weckesser, “nullspace- retrieved 02.02.2017,” 2015. [Online]. Available: http://scipy-cookbook.readthedocs.io/items/RankNullspace.html

A Linear Mixing Law in a Constant Density System

In a constant density system, the mixing process is linear in concentration space. Let $\underline{C}_1$ and $\underline{C}_2$ be two different concentration vectors and let V be the volume of the liquid. By mixing $\underline{C}_1$ and $\underline{C}_2$, the resultant concentration will be $\underline{C}^*$. The concentration of $\underline{C}^*$ is then given by the mass balance in Equation (A.1).

$$\begin{aligned}\underline{C^*} &= \frac{V_1\underline{C_1} + V_2\underline{C_2}}{V^*} \\ &= \lambda\underline{C_1} + (1-\lambda)\underline{C_2}, \qquad \text{where } \lambda = \frac{V_1}{V^*} = \frac{V_1}{V_1+V_2}\end{aligned} \tag{A.1}$$

Also residence times obey a linear mixing law. Let τ^* be the residence time of the system. Then, Equation A.2 must be true.

$$\begin{aligned}\tau^* &= \frac{V_{tot,reac}}{Q_{tot}} \\ &= \frac{V_{1,reac} + V_{2,reac}}{Q_{tot}} \\ &= \frac{\tau_1 Q_1 + \tau_2 Q_2}{Q_{tot}} \\ &= \lambda\tau_1 + (1-\lambda)\tau_2, \qquad \text{where } \lambda = \frac{Q_1}{Q_{tot}}\end{aligned} \tag{A.2}$$

Hence, residence time and concentrations inhibit the same properties with regards to mixing, and τ may therefore be incorporated as a pseudo-component in the concentration vector.

B Code

This appendix contains the code used for generating the ARs in this thesis. First, the module which were implemented to generally facilitate construction of an AR are given. Then, all the kinetic models are presented. Finally, the main.py-file for the HBu fermentation model is shown. The rest of the main-files bear close resemblance, and are omitted due to space considerations. They are however handed in to the supervisor.

B.1 artools.py

The function "nullspace" is obtained from the scipy-cookbook [44].

```
import numpy as np
import scipy as sp
import sympy
import matplotlib as mpl
mpl.use(’Qt5Agg’) #works as an interactive backend.
#There is currently a bug with the backend macosx
# which makes it not possible to pan 3D plots
from sympy.interactive.printing import init_printing
init_printing(use_unicode = True)#, wrap_line = False, no_global = True)
from sympy.matrices import Matrix, zeros, ones, diag, GramSchmidt
import scipy.integrate
import scipy.optimize
import scipy.spatial
import scipy.linalg
import numpy.linalg
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
import matplotlib.tri as mtri
from sympy.utilities.lambdify import lambdastr
import copy



def PFRtrajectory(rxRate, c0, tEnd, numPoints = 200, lintime = False):
    """
    Computes the PFR trajectory for a system
    INPUT
    rxRate: lambda function which specifies the reaction rate vector.
        rxRate(c0) gives numeric value for the reaction rate in the point c0
    c0: Feed concentration to the reactor (a numpy array)
    tEnd: stop of the integration time
    numPoints: Number of poitns evaluated
    #linTime: default false (=> use logarithmic timescale)

    OUTPUT
    cPFR: concentration trajectory for the components
    tPFR: Time points evaluated
    """
    if lintime:
        tPFR = sp.linspace(0, tEnd, numPoints)
    else:
        #scipy don't have log10(0), therefore use append first and decrease
        # numPoints by 1
        tPFR = sp.append(0.0, sp.logspace(-3, sp.log10(tEnd), numPoints - 1))
    odeSys = lambda conc, time: rxRate(conc)
    cPFR = sp.integrate.odeint(odeSys, c0, tPFR)
    return cPFR, tPFR

def CSTRlocusFast(rxRate, c0, tEnd, numPoints = 150, lintime = False):
    """
    Computes the CSTR locus for a system
    INPUT
    rxRate: lambda function which specifies the reaction rate vector. rxRate(c0)
        # gives numeric value for the reaction rate in the point c0
    c0: Feed concentration to the reactor (a numpy array)
    tEnd: specifies the time length evaluated (an int)
    numPoints: Number of poitns evaluated
    #linTime: default false (=> use logarithmic timescale)

    OUTPUT
    cCSTR: concentration loci for the components
    tCSTR: Time points evaluated
    """
    if lintime:
        tCSTR = sp.linspace(0, tEnd, numPoints)
    else:
        #scipy don't have log10(0), therefore use append first and decrease
        # numPoints by 1
        tCSTR = sp.append(0.0, sp.logspace(-3, sp.log10(tEnd), numPoints - 1))
    cCSTR = np.zeros((numPoints, c0.shape[0]))
    cCSTR[0, :] = c0
    for i in range(1, numPoints):
        eqCSTR = lambda conc: (conc - c0) - tCSTR[i]*rxRate(conc)
        cCSTR[i, :] = sp.optimize.fsolve(eqCSTR, cCSTR[i-1, :])
    return cCSTR, tCSTR

def residual1DimCSTReq(rxRate, cf, c0End, tEnd, numPoints = 100):
    """
    Compute the residual of the CSTR equation for a 1D AR (2D with tau) in order
    to check for multiple solutions. The output will be a meshgrid so that a
    visual confirmation of the number of solutions is possible.
    INPUT
    rxRate: lambda function which specifies the reaction rate vector. rxRate(c0)
        gives numeric value for the reaction rate in the point c0
    c0End: Initial guesses for the CSTR design equation goes in a linear
        grid from [0,..,c0End]
    tEnd: specifies the time length evaluated (an int)
    numPoints: Number of poitns evaluated

    OUTPUT
    tv, cv, res: meshgrid of tEnd, c0End and corresponding value of the residual

    """
    tarr = np.linspace(0, tEnd, num = numPoints) # time array
    carr = np.linspace(0, c0End, num = numPoints) # concentration array
    tv, cv = np.meshgrid(tarr, carr, indexing = 'ij')
    res = np.zeros(tv.shape)
    for i in range(numPoints):
        for j in range(numPoints):
            myinput = np.array([cv[i, j], tv[i,j]])
            res[i,j] = (cv[i, j] - cf) - tv[i, j]*rxRate(myinput)[0]
    return tv, cv, res



def CSTRlocusSetInitialGuess(rxRate, c0, tEnd, cGuess,
    numPoints = 150, lintime = False):
    """
    Computes the CSTR locus for a system
    INPUT
    rxRate: lambda function which specifies the reaction rate vector.
        rxRate(c0) gives numeric value for the reaction rate in the point c0
    c0: Feed concentration to the reactor (a numpy array)
    tEnd: specifies the time length evaluated (an int)
    cGuess: Guess of the concentration vector
    numPoints: Number of poitns evaluated
    #linTime: default false (=> use logarithmic timescale)

    OUTPUT
    cCSTR: concentration loci for the components
    tCSTR: Time points evaluated
    """
    if lintime:
        tCSTR = sp.linspace(0, tEnd, numPoints)
    else:
        #scipy don't have log10(0), therefore use append first and
        # decrease numPoints by 1
        tCSTR = sp.append(0.0, sp.logspace(-3, sp.log10(tEnd), numPoints - 1))
    cCSTR = np.zeros((numPoints, c0.shape[0]))
    cCSTR[0, :] = c0
    for i in range(1, numPoints):
        eqCSTR = lambda conc: (conc - c0) - tCSTR[i]*rxRate(conc)
        cCSTR[i, :] = sp.optimize.fsolve(eqCSTR, cGuess)
        #Need to check if we've a valid solution, that is all concentrations
        # are positive
        if not all(num >= 0 for num in cCSTR[i, :]):
            cCSTR[i, :] = np.full(cCSTR[i, :].size, np.nan) #fills the row
            # with 'nan' if there are negative values
    return cCSTR, tCSTR

def convhull(points, incrementalBool = False, qhull_optionsChosen = None):
    """
    A wrapper for the scipy.spatial.ConvexHull function. See the doc for
    this function
    """
    return scipy.spatial.ConvexHull(points, incremental = incrementalBool,
        qhull_options = qhull_optionsChosen)

def optimCSTRforTau2D(hull0, cCSTR, rxRate, tEndPFR,
    makePlot = False, numPointsPFR = 200):
    """
    Determine the optimal CSTR in a 2D AR-plot, where residence
    time is a state space variable.
    From each CSTR effluent concentration, a new PFR trajectory is computed.
    The optimal CSTR is the one which, combined with the "new"
    subsequent PFR, maximizes the area of the convex hull

    INPUT:
    hull0: a scipy.spatial.ConvexHull(incremental = True) object. Consists
        of initial PFR and CSTR concentrations
    cCSTR: CSTR locus, initiated from the feed point
    rxRate: lambda function which specifies the reaction rate vector.
         rxRate(c0) gives numeric valuefor the reaction rate in the point c0
    tEndPFR: End-time of PFR
    makePlot: boolean variable if a graph of (concentration,
        increase in area of convex hull) should be made

    OUTPUT:
    optimIndex: The optimal index for the CSTR, the index which
        maximizes the convex hull when a PFR is initiated from the
        corresponding CSTR effluent concentration

    """
    #Iterate through all the points on the CSTR (optimal
    #solution is to check whether the points are below the
    #PFR curve, but use brute force for simplicity)
    hull = {}
    hull['h'] = {} #save the hulls
    hull['da'] = np.zeros(cCSTR[:, 0].shape) #da = dArea, increase in
    # the area of the convex hull
    for i in range(len(cCSTR[:, 0])):
        # print('loop started')
        c0 = cCSTR[i, :] #inlet concentration to the PFR
        cNewPFR, tWaste = PFRtrajectory(rxRate, c0,
            tEndPFR, numPoints = numPointsPFR)
        """
        Don't want the plot to be expanded over the equilibrium point
        (know that equilibrium is reached at tEndPFR. If it is
        expanded, the algorithm with searching for the maximum convex hole
        will not work, since it's constantly being expanded in the
        tau-direction, but not in the concentration direction). Hence,
        all residence time points in the new cPFR which is larger than
        tEndPFR needs to be set to tEndPFR (no information is lost
        since the concentration doesn't change). This is done in
        the next line. cNewPFR[:, -1] > tEndPFR makes a boolean array,
        andwhen the indexed array is set to true, the
        new time will be tEndPFR
        """
        cNewPFR[cNewPFR[:, -1] > tEndPFR, -1] = tEndPFR# It’s assumed
        #that the residence time data  is the last entry in cNewPFR
        h2 = copy.copy(hull0)# copy.copy(hull0) #copy the values, not
        # the reference
        h2.add_points(cNewPFR, restart = True)
        hull[’da’][i] = h2.volume - hull0.volume #hull.volume and
        # hull.area are 3D attributes.
        # In 2D, volume thus refers to the area of the convex hull, while
        # hull.area refers to the circumference of the hull


    optimIndex = hull[’da’].argmax()#find the maximum increase of the area
    if makePlot:
        fig = plt.figure(’Max␣area␣of␣convex␣hull␣for␣optimal␣CSTR␣index’)
        ax = fig.add_subplot(111)
        ax.set_xlabel(’C’)
        ax.set_ylabel(’Increase␣in␣area␣of␣the␣convex␣hull’)
        ax.scatter(cCSTR[:, 0], hull[’da’], s = 5)
        return optimIndex, fig, ax
    else:
        return optimIndex



def nullspace(A, atol=1e-13, rtol=0):
    """Compute an approximate basis for the nullspace of A.

    The algorithm used by this function is based on the singular value
    decomposition of ‘A‘.

    Parameters
    ----------
    A : ndarray
        A should be at most 2-D.  A 1-D array with length k will be treated
        as a 2-D with shape (1, k)
    atol : float
        The absolute tolerance for a zero singular value.  Singular values
        smaller than ‘atol‘ are considered to be zero.
    rtol : float
        The relative tolerance.  Singular values less than rtol*smax are
        considered to be zero, where smax is the largest singular value.

    If both ‘atol‘ and ‘rtol‘ are positive, the combined tolerance is the
    maximum of the two; that is::
        tol = max(atol, rtol * smax)
    Singular values smaller than ‘tol‘ are considered to be zero.

    Return value
    ------------
    ns : ndarray
        If ‘A‘ is an array with shape (m, k), then ‘ns‘ will be an array
        with shape (k, n), where n is the estimated dimension of the
        nullspace of ‘A‘.  The columns of ‘ns‘ are a basis for the
        nullspace; each element in numpy.dot(A, ns) will be approximately
        zero.
    """

    A = np.atleast_2d(A)
    u, s, vh = np.linalg.svd(A)
    tol = max(atol, rtol * s[0])
    nnz = (s >= tol).sum()
    ns = vh[nnz:].conj().T
    return ns
```

B.2 Model Equations

B.2.1 Scenario 1: Fermentation Model for BuOH

```
import numpy as np
import sympy
from sympy.matrices import Matrix


def rxRateFun(c, rc):
        """
        Computes the reaction rate vector for a given concentration

        INPUT
        c: numpy array. State space vector which the AR will be
                computed in (usually concentrations or
                concentrations and residence time space)
        rc: dictionary. Information about the rate constants
        OUTPUT
        r: numpy array. The rate vector
        """
        # Unpack the rate vector
        B = rc[’B’]
        cf = rc[’cf’]
        cfAll = rc[’cfAll’]
        ix = rc[’ix’]
        isub = rc[’isub’]
        ihbu = rc[’ihbu’]# initial concentrations [g/L]
        ibuoh = rc[’ibuoh’]
        cs = c[0]
        cbuoh = c[1]
        r = np.zeros(np.shape(c))

        #Write the mass balance matrix. Have s and buoh as independent species
        #                               s, buoh
        A = np.matrix([[-B[2], 0],#x
                [1, 0],#s
                [-B[3] - B[4]*B[9]/B[10], -B[4]/B[10]],#hbu
                [0, 1], #buoh
                [-B[13], 0]])#etoh
        # write delta-concentrations = c - cf for the independent species
        cfAll = np.asmatrix(cfAll[:-1]).T
        cf = np.asmatrix(cf).T
        c = np.asmatrix(c).T
        dci = c - cf #change of independent concentrations
        cAll = cfAll + A*dci #mass balance for dependent and independent species

        cx = cAll[ix, 0]
        chbu = cAll[ihbu, 0]

        rs = ((B[0] * cs) / (B[1] + cs))/B[2]

        r[0] = -rs*cx #dcs/dt
        r[1] = -B[9]*r[0] + B[10]*(chbu/(B[5] + chbu))*cx # dcbuoh/dt
        return r



def solveMoleBalance(cin, rc):
        """
        Solve the mole/mass balance for the system (determine the
        dependent concentrations given the
        independent concentrations)
        INPUT
        cin: numpy array. State space vector which the AR will be
                computed in (usually concentrations or
                concentrations and residence time space). The independent variables
        rc: dictionary. Information about the rate constants
        OUTPUT
        cout: numpy array. State space vector for all independent and
                dependent variabels (independent concentrations + residence
                time (if this is variabel in the specific AR) AND dependent
                concentrations
        """

        B = rc[’B’]
        cf = rc[’cf’]
        cfAll = rc[’cfAll’]
        ix = rc[’ix’]
        isub = rc[’isub’]
        ihbu = rc[’ihbu’]# initial concentrations [g/L]
        ibuoh = rc[’ibuoh’]
        ietoh = rc[’ietoh’]
        itau = rc[’itau’]

        #Write the mass balance matrix. Have s and buoh as independent species
        #                               s, buoh
        A = np.matrix([[-B[2], 0],#x
                [1, 0],#s
                [-B[3] - B[4]*B[9]/B[10], -B[4]/B[10]],#hbu
                [0, 1], #buoh
                [-B[13], 0]])#etoh
        # write delta-concentrations = c - cf for the independent species
        cfAll = np.asmatrix(cfAll[:-1]).T
        cf = np.asmatrix(cf).T
        cAll = np.zeros((cin.shape[0], len(cfAll)))
        for i in range(len(cin[:, 0])):
                cInd = np.asmatrix(cin[i, :]).T
                # ci = np.matrix([[cs], [cbuoh]])
                dci = cInd - cf #change of independent concentrations
                cAlli = cfAll + A*dci #mass balance
                cAll[i, :] = np.asarray(cAlli)[:, 0]
        cout = np.column_stack((cAll[:, ix], cAll[:, isub],
                cAll[:, ihbu], cAll[:, ibuoh], cAll[:, ietoh]))
        return cout
```

B.2.2 Scenario 1: Fermentation Model for HBu

```
import numpy as np
import sympy
from sympy.matrices import Matrix
def rxRateFun(c, rc):
        """
        Computes the reaction rate vector for a given concentration

        INPUT
        c: numpy array. State space vector which the AR will be
                computed in (usually concentrations or
                concentrations and residence time space)
        rc: dictionary. Information about the rate constants
        OUTPUT
        r: numpy array. The rate vector
        """

        # Unpack the rate vector. For units, check the definition
        # of rc in the file "main.py"
        mu_m = rc[’mu_m’]
        exponent = rc[’exponent’]
        m_S = rc[’m_S’]
        K_I = rc[’K_I’]
        K_d = rc[’K_d’]
        K_S = rc[’K_S’]
        P_d = rc[’P_d’]
        alpha_Aa = rc[’alpha_Aa’]
        alpha_Ba = rc[’alpha_Ba’]
        beta_Aa = rc[’beta_Aa’]
        beta_Ba = rc[’beta_Ba’]
        Y_X = rc[’Y_X’]
        Y_Aa = rc[’Y_Aa’]
        Y_Ba = rc[’Y_Ba’]
        x = rc[’x_ind’]# initial concentrations [g/L]
        s = rc[’s_ind’]
        aa = rc[’aa_ind’]
        ba = rc[’ba_ind’]
        tau_ind = rc[’tau_ind’]
        cf = rc[’cfAll’] #containes the feed concentration of all
        # the components in the system
        N = rc[’N’]
        MW = rc[’MW’]

        #unpack the concentration vector
        cs = c[0]
        tau = c[1]

        #simplify notation, "invent" a variable for dsdt
        alpha_S = 1/Y_X + (alpha_Aa/Y_Aa)*(1 + 1*44.01/60.05) +
                (alpha_Ba/Y_Ba)*(1 + 2*44.01/88.1)

        #For simplification of the rate expression, compute cx, cs and caa
        #(could’ve substituted inside the function)
        cx = cf[x] - (1/alpha_S) * (cs - cf[s]) #from dxdt = alpha_ba*dba/dt
        cba = cf[ba] - (alpha_Ba/alpha_S) * (cs - cf[s]) # f
        caa = cf[aa] - (alpha_Aa/alpha_S) * (cs - cf[s])
        # print(cx*MW[x],cs*MW[s],caa*MW[aa], cba*MW[ba], tau)
        #compute the rate vector
        mu = ((mu_m * cs) / (cs + K_S +
                ((cs**2)/K_I))) * ((1 - (cba + caa)/P_d)**exponent)
        r = np.zeros(np.shape(c))
        r[0] = -alpha_S*(mu*cx) # [g/(L h)], dcs/dt
        r[1] = 1 #dt/dt
        return r

def solveMoleBalance(cin, rc):
        """
        Solve the mole/mass balance for the system (determine
        the dependent concentrations given the
        independent concentrations)
        INPUT
        cin: numpy array. State space vector which the AR will be
                computed in (usually concentrations or
                concentrations and residence time space). The independent variables
        rc: dictionary. Information about the rate constants
        OUTPUT
        cout: numpy array. State space vector for all independent and d
                ependent variabels (independent concentrations + residence time
                (if this is variabel in the specific AR) AND dependent
                concentrations

        """

        # Unpack the rate vector
        mu_m = rc[’mu_m’]
        exponent = rc[’exponent’]
        m_S = rc[’m_S’]
        K_I = rc[’K_I’]
        K_d = rc[’K_d’]
        K_S = rc[’K_S’]
        P_d = rc[’P_d’]
        alpha_Aa = rc[’alpha_Aa’]
        alpha_Ba = rc[’alpha_Ba’]
        beta_Aa = rc[’beta_Aa’]
        beta_Ba = rc[’beta_Ba’]
        Y_X = rc[’Y_X’]
        Y_Aa = rc[’Y_Aa’]
        Y_Ba = rc[’Y_Ba’]
        x = rc[’x_ind’]# initial concentrations [g/L]
        s = rc[’s_ind’]
        aa = rc[’aa_ind’]
        ba = rc[’ba_ind’]
        tau_ind = rc[’tau_ind’]
        cf = rc[’cfAll’] #feed concentration of all components in the system
        N = rc[’N’]
        MW = rc[’MW’]

        #unpack the concentration vector
        cs = cin[:, 0]
        tau = cin[:, 1]

        #simplify notation, "invent" a variable for dsdt
        alpha_S = 1/Y_X + (alpha_Aa/Y_Aa)*(1 + 1*44.01/60.05) +
                (alpha_Ba/Y_Ba)*(1 + 2*44.01/88.1)

        #For simplification of the rate expression, compute cx, cs and caa
        #(could’ve substituted inside the function)
        cx = cf[x] - (1/alpha_S) * (cs - cf[s]) #from dxdt = alpha_ba*dba/dt
        cba = cf[ba] - (alpha_Ba/alpha_S) * (cs - cf[s])
        caa = cf[aa] - (alpha_Aa/alpha_S) * (cs - cf[s])
        #stack the variables in the output concentrations vector
        cout = np.column_stack((cs, cx, caa, cba, tau))

        return cout
```

B.2.3 Scenario 1: Esterification Model of BuB

```
import numpy as np
import sympy
from sympy.matrices import Matrix
def rxRateFun(c, rc):
        """
        Computes the reaction rate vector for a given concentration

        INPUT
        c: numpy array. State space vector which the AR will be computed
                in (usually concentrations or
                concentrations and residence time space)
        rc: dictionary. Information about the rate constants
        OUTPUT
        r: numpy array. The rate vector
        """
        # Unpack the rate vector
        rmax = rc[’rmax’] # mol/(L h)
        Kma = rc[’Kma’] # [mol/L]
        Kmb = rc[’Kmb’] # [mol/L]
        Kia = rc[’Kia’] # [mol/L]
```

```python
        cf = np.asmatrix(rc['cf']).T # [g/L]
        cfAll = np.asmatrix(rc['cfAll']).T # [g/L], np.array([cbuoh,
        # chbu, cbub, cs, tau])
        isub = rc['isub'] # indices in vectors
        ihbu = rc['ihbu']
        ibuoh = rc['ibuoh']
        ibub = rc['ibub']
        MW = rc['MW'] #[g/mol], np.array

        #make c = [cbub, tau] a matrix in order to multiply
        ci = np.asmatrix(c).T # independent concentrations
        # mass balance matrix
        A = np.matrix([[-MW[ibuoh]/MW[ibub], 0], #buoh
                [-MW[ihbu]/MW[ibub], 0], # hbu
                [1, 0], # bub
                [0, 0], # s
                [0, 1]]) # tau

        cAll = cfAll + A*(ci - cf)

        # compute the rate of reaction, in molar form. Rewrite chbu = ca and
        # cbuoh = cb, where ca,cb are in mol/L
        ca = cAll[ihbu, 0]/MW[ihbu]
        cb = cAll[ibuoh, 0]/MW[ibuoh]

        rateOfRx = (rmax * ca * cb) / (Kia*Kmb + Kma*cb + Kmb*ca + ca*cb)

        r = np.zeros(np.shape(c))
        r[0] = MW[ibub] * rateOfRx # [g/L h], dbub/dt
        r[1] = 1 # [-] dt/dt
        return r

def solveMoleBalance(cin, rc):
        """
        Solve the mole/mass balance for the system (determine the
        dependent concentrations given the
        independent concentrations)
        INPUT
        cin: numpy array. State space vector which the AR will be computed
                in (usually concentrations or
                concentrations and residence time space). The independent variables
        rc: dictionary. Information about the rate constants
        OUTPUT
        cout: numpy array. State space vector for all independent and
                dependent variabels (independent concentrations + residence time
                (if this is variabel in the specific AR) AND dependent
                concentrations
        """

        # Unpack the rate vector
        # Unpack the rate vector
        rmax = rc['rmax'] # mol/(L h)
        Kma = rc['Kma'] # [mol/L]
        Kmb = rc['Kmb'] # [mol/L]
        Kia = rc['Kia'] # [mol/L]
        cf = np.asmatrix(rc['cf']).T # [g/L]
        cfAll = np.asmatrix(rc['cfAll']).T # [g/L], np.array([cbuoh,
        # chbu, cbub, cs, tau])
        isub = rc['isub'] # indices in vectors
        ihbu = rc['ihbu']
        ibuoh = rc['ibuoh']
        ibub = rc['ibub']
        MW = rc['MW'] #[g/mol], np.array

        #make c = [cbub, tau] a matrix in order to multiply
        # ci = np.asmatrix(c).T # independent concentrations
        # mass balance matrix
        A = np.matrix([[-MW[ibuoh]/MW[ibub], 0], #buoh
                [-MW[ihbu]/MW[ibub], 0], # hbu
                [1, 0], # bub
                [0, 0], # s
                [0, 1]]) # tau

        # cAll = cfAll + A*(ci - cf)
        # write delta-concentrations = c - cf for the independent species
        cAll = np.zeros((cin.shape[0], len(cfAll)))
        for i in range(len(cin[:, 0])):
                cInd = np.asmatrix(cin[i, :]).T
                # ci = np.matrix([[cs], [cbuoh]])
                dci = cInd - cf #change of independent concentrations
                cAlli = cfAll + A*dci #mass balance
                cAll[i, :] = np.asarray(cAlli)[:, 0]

        return cAll
```

B.2.4 Scenario 2: Model of Combined Reactor

```
import numpy as np
import sympy
from sympy.matrices import Matrix
def rxRateFun(c, rc):
        """
        Computes the reaction rate vector for a given concentration

        INPUT
        c: numpy array. State space vector which the AR will be computed in
                (usually concentrations or
                concentrations and residence time space)
        rc: dictionary. Information about the rate constants
        OUTPUT
        r: numpy array. The rate vector
        """

        # Unpack the rate vector. For units, check the definition of
        # rc in the file "main.py"
        phbu = rc['phbu'] # parameters for hbu model
        pbub = rc['pbub'] # parameters for bub model

        # Unpack hbu model
        mu_m = phbu['mu_m']
        exponent = phbu['exponent']
        m_S = phbu['m_S']
        K_I = phbu['K_I']
        K_d = phbu['K_d']
        K_S = phbu['K_S']
        P_d = phbu['P_d']
        alpha_Aa = phbu['alpha_Aa']
        alpha_Ba = phbu['alpha_Ba']
        beta_Aa = phbu['beta_Aa']
        beta_Ba = phbu['beta_Ba']
        Y_X = phbu['Y_X']
        Y_Aa = phbu['Y_Aa']
        Y_Ba = phbu['Y_Ba']

        # Unpack bub model
        rmax = pbub['rmax'] # mol/(L h)
        Kma = pbub['Kma'] # [mol/L]
        Kmb = pbub['Kmb'] # [mol/L]
        Kia = pbub['Kia'] # [mol/L]

        # Unpack rest of the parameters
        cfAll = np.asmatrix(rc['cfAll']).T #feed concentrations, all components
        cf = np.asmatrix(rc['cf']).T
        MW = rc['MW']
        isub = rc['isub'] # indices in vectors
        ihbu = rc['ihbu']
        ibuoh = rc['ibuoh']
        ibub = rc['ibub']
        ix = rc['ix']
        ihac = rc['ihac']

        # Make the independent concentrations in matrix form
        ci = np.asmatrix(c).T #independt species: S, HBu, BuB

        # Write the (combined) kinetic model
        # First; simplify notation. "Invent" helping variables gamma and omega
        gamma = 1/Y_X + (alpha_Aa/Y_Aa)*(1 + 1*44.01/60.05) +
                (alpha_Ba/Y_Ba)*(1 + 2*44.01/88.1)
        omega = (1/Y_Ba)*(1 + 2*44.01/88.1)

        # Make the reaction rate matrix for all the components
        A = np.matrix([[-1/gamma, (omega/gamma) * (MW[ihbu]/MW[ibub])],# dcX/dt
                [1, 0], #dcS/dt
                [-alpha_Aa/gamma, alpha_Aa * (omega/gamma) * (MW[ihbu]/MW[ibub])], # dcHac/dt
                [-alpha_Ba/gamma, (MW[ihbu]/MW[ibub]) * (alpha_Ba * (omega/gamma) - 1)], #dcHBu/dt
                [0, -MW[ibuoh]/MW[ibub]], # dcBuOH/dt
                [0, 1]]) # dcBub/dt

        # Compute the concentration of all components
        cAll = cfAll + A*(ci - cf)


        # For the Bub model: compute the rate of
        #reaction, in molar form. Rewrite chbu = ca and
        # cbuoh = cb, where ca,cb are in mol/L
        ca = cAll[ihbu, 0]/MW[ihbu]
        cb = cAll[ibuoh, 0]/MW[ibuoh]

        rateOfRxBubModelMolar = (rmax * ca * cb) /
                (Kia*Kmb + Kma*cb + Kmb*ca + ca*cb) #[mol/(L h)]

        # Calculate the cell growth rate (HBu-model)
        mu = ((mu_m * cAll[isub, 0]) / (cAll[isub, 0] + K_S +
                ((cAll[isub, 0]**2)/K_I))) *
                ((1 - (cAll[ihbu, 0] + cAll[ihac, 0])/P_d)**exponent)

        # Get the rate vector for the independent species
        r = np.zeros(np.shape(c))
        r[0] = -gamma*(mu*cAll[ix, 0]) +
                omega * MW[ihbu] * rateOfRxBubModelMolar # dcS/dt, [g/(L h)]
        r[1] = MW[ibub] * rateOfRxBubModelMolar # dcBub/dt [g/(L h)]
        return r

def solveMoleBalance(cin, rc):
        """
        Solve the mole/mass balance for the system (determine the
        dependent concentrations given the
        independent concentrations)
        INPUT
        cin: numpy array. State space vector which the AR will be computed
                in (usually concentrations or
                concentrations and residence time space).
        rc: dictionary. Information about the rate constants
        OUTPUT
        cout: numpy array. State space vector for all independent and
                dependent variabels (independent concentrations + residence time
                (if this is variabel in the specific AR) AND dependent
                concentrations

        """

        # Unpack the rate vector
        phbu = rc[’phbu’] # parameters for hbu model
        pbub = rc[’pbub’] # parameters for bub model

        # Unpack hbu model
        mu_m = phbu[’mu_m’]
        exponent = phbu[’exponent’]
        m_S = phbu[’m_S’]
        K_I = phbu[’K_I’]
        K_d = phbu[’K_d’]
        K_S = phbu[’K_S’]
        P_d = phbu[’P_d’]
        alpha_Aa = phbu[’alpha_Aa’]
        alpha_Ba = phbu[’alpha_Ba’]
        beta_Aa = phbu[’beta_Aa’]
        beta_Ba = phbu[’beta_Ba’]
        Y_X = phbu[’Y_X’]
        Y_Aa = phbu[’Y_Aa’]
        Y_Ba = phbu[’Y_Ba’]

        # Unpack bub model
        rmax = pbub[’rmax’] # mol/(L h)
        Kma = pbub[’Kma’] # [mol/L]
        Kmb = pbub[’Kmb’] # [mol/L]
        Kia = pbub[’Kia’] # [mol/L]

        # Unpack rest of the parameters
        cfAll = np.asmatrix(rc[’cfAll’]).T #feed concentration, all components
        cf = np.asmatrix(rc[’cf’]).T
        MW = rc[’MW’]
        isub = rc[’isub’] # indices in vectors
        ihbu = rc[’ihbu’]
        ibuoh = rc[’ibuoh’]
        ibub = rc[’ibub’]
        ix = rc[’ix’]
        ihac = rc[’ihac’]

        # Write the (combined) kinetic model
        # First; simplify notation. "Invent" helping variables gamma and omega
        gamma = 1/Y_X + (alpha_Aa/Y_Aa)*(1 + 1*44.01/60.05) +
                (alpha_Ba/Y_Ba)*(1 + 2*44.01/88.1)
        omega = (1/Y_Ba)*(1 + 2*44.01/88.1)

        # Make the reaction rate matrix for all the components
        A = np.matrix([[-1/gamma, (omega/gamma) * (MW[ihbu]/MW[ibub])],# dcX/dt
                [1, 0], #dcS/dt
                [-alpha_Aa/gamma, alpha_Aa *
                        (omega/gamma) * (MW[ihbu]/MW[ibub])], # dcHac/dt
                [-alpha_Ba/gamma, (MW[ihbu]/MW[ibub]) *
                        (alpha_Ba * (omega/gamma) - 1)], #dcHBu/dt
                [0, -MW[ibuoh]/MW[ibub]], # dcBuOH/dt
                [0, 1]]) # dcBub/dt

        cAll = np.zeros((cin.shape[0], len(cfAll)))
        for i in range(len(cin[:, 0])):
                cInd = np.asmatrix(cin[i, :]).T
                # ci = np.matrix([[cs], [cbuoh]])
                dci = cInd - cf #change of independent concentrations
                cAlli = cfAll + A*dci #mass balance
                cAll[i, :] = np.asarray(cAlli)[:, 0]
        return cAll

def solveMoleBalancePointStateSpace(cin, rc):
        """
        Solve the mole/mass balance for a single point in
        the state fspace for the system
        (determine the dependent concentrations given the
        independent concentrations)
        INPUT
        cin: 1D numpy array. State space vector which the AR
                will be computed in (usually concentrations or
                concentrations and residence time space).
        rc: dictionary. Information about the rate constants
        OUTPUT
        cout: 1D numpy array. State space vector for all independent
                and dependent variabels (independent
                concentrations + residence time (if this is variabel in
                the specific AR) AND dependent
                concentrations

        """

        # Unpack the rate vector
        phbu = rc['phbu'] # parameters for hbu model
        pbub = rc['pbub'] # parameters for bub model

        # Unpack hbu model
        mu_m = phbu['mu_m']
        exponent = phbu['exponent']
        m_S = phbu['m_S']
        K_I = phbu['K_I']
        K_d = phbu['K_d']
        K_S = phbu['K_S']
        P_d = phbu['P_d']
        alpha_Aa = phbu['alpha_Aa']
        alpha_Ba = phbu['alpha_Ba']
        beta_Aa = phbu['beta_Aa']
        beta_Ba = phbu['beta_Ba']
        Y_X = phbu['Y_X']
        Y_Aa = phbu['Y_Aa']
        Y_Ba = phbu['Y_Ba']

        # Unpack bub model
        rmax = pbub['rmax'] # mol/(L h)
        Kma = pbub['Kma'] # [mol/L]
        Kmb = pbub['Kmb'] # [mol/L]
        Kia = pbub['Kia'] # [mol/L]

        # Unpack rest of the parameters
        cfAll = np.asmatrix(rc['cfAll']).T #feed concentration, all components
        cf = np.asmatrix(rc['cf']).T
        MW = rc['MW']
        isub = rc['isub'] # indices in vectors
        ihbu = rc['ihbu']
        ibuoh = rc['ibuoh']
        ibub = rc['ibub']
        ix = rc['ix']
        ihac = rc['ihac']

        # Write the (combined) kinetic model
        # First; simplify notation. "Invent" helping variables gamma and omega
        gamma = 1/Y_X + (alpha_Aa/Y_Aa)*(1 + 1*44.01/60.05) +
                (alpha_Ba/Y_Ba)*(1 + 2*44.01/88.1)
        omega = (1/Y_Ba)*(1 + 2*44.01/88.1)

        # Make the reaction rate matrix for all the components
        A = np.matrix([[-1/gamma, (omega/gamma) * (MW[ihbu]/MW[ibub])],# dcX/dt
                [1, 0], #dcS/dt
                [-alpha_Aa/gamma, alpha_Aa * (omega/gamma) *
                        (MW[ihbu]/MW[ibub])], # dcHac/dt
                [-alpha_Ba/gamma, (MW[ihbu]/MW[ibub]) *
                        (alpha_Ba * (omega/gamma) - 1)], #dcHBu/dt
                [0, -MW[ibuoh]/MW[ibub]], # dcBuOH/dt
                [0, 1]]) # dcBub/dt

        ci = np.asmatrix(cin).T

        cAll = cfAll + A*(ci - cf)
        cout = np.asarray(cAll.T)[0, :]

        return cout
```

B.3 Main-files

As the structure of each main.py-file were similar, only the file for the construction of HBu are included. The rest of the files were handed to supervisor.

```
import numpy as np
import sympy
import matplotlib as mpl
mpl.use('Qt5Agg')
import matplotlib.lines#for manipulating legend entries
import matplotlib.patches#for manipulating legend entries
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
import os
import numpy.matlib
import time
import sys
homeDir = os.path.expanduser('~')
sys.path.insert(0, homeDir + '/Dropbox/Masteroppgave/PythonFiles')#Need
#to do this in order
#to import artools, which is a file in another directory
import artools
from kineticsHBu import rxRateFun, solveMoleBalance
import scipy.optimize

font = {'size': 12}
mpl.rc('font', **font)

#Make/check folders for storage of the dictionaries which will be computed
dirmain = os.getcwd()
dirCSV = dirmain + '/computedCSV'
if not os.path.exists(dirCSV):
        os.makedirs(dirCSV)
```

```
dirPlots = dirmain + ’/plots’
if not os.path.exists(dirPlots):
        os.makedirs(dirPlots)

#indices in cfAll and other matrices
s = 0 # substrate (glucose)
x = 1 # biomass
aa = 2 # acetic acid
ba = 3 # butyric acid
tauInd = 4 # residence time

#stoichiometric matrix
#       s   x aa ba H2 CO2 H2O
N = np.matrix([[
        -1, 0, 0, 1, 2, 2, 0], #rx1
        -1, 0, 1, 0, 4, 2, -2]) #rx2
N = N.T

MW = np.array([12.01*6 + 12*1.008 + 6*15.9994,
        12.01 + 1.6*1.008 + 0.43*15.9994 + .25*14.0067,
        60.05, 88.1]) # [g/mol]. Molar weights for MW = [MWS, MWX, MWAa, MWBa]

#Declare initial concentrations
cfAll = np.array([50, .1, 0, 0, 0])#[g/L, g/L, g/L, g/L, h]. Feed conc.
#Contains c0 = [cs0, cx0 caa0, cba0, tau0]
cf = np.array([cfAll[s], cfAll[tauInd]]) # [g/L, h]. Contains cbb and tau

#Declare parameters in the simulation
rc = {’mu_m’: 0.48, ’K_d’: 0.0024, ’m_S’: 0.002, #[1/h]
        ’K_I’: 372, ’K_S’: 1.62, ’P_d’: 48.3, #[g/L]
        ’Y_Aa’: 0.999, ’alpha_Aa’: 0.72, ’alpha_Ba’: 2.92,
        ’Y_X’: 0.807, ’Y_Ba’: 0.978, #[g/g]
        ’beta_Aa’: 0, ’beta_Ba’: 0.057, # [g/(gh)]
        ’exponent’: 5.18, # [-]
        ’s_ind’: s, ’x_ind’: x, ’aa_ind’: aa, ’ba_ind’: ba,
        ’tau_ind’: tauInd,#indices in cfAll
        ’cfAll’: cfAll, # [mol/l]
        ’MW’: MW, #[g/mol], np.matrix of molecular weights
        ’N’: N} #[-] stoichiometric matrix

tEnd = {’PFR’: 30, ’CSTR’: 30} #[h]
epsNonNeg = -1e-3 #threshold value for respecting
# non-negativity constraint in the reactors

#Make dicts for saving the data from AR calculations
cPFR = {}
tPFR = {}
cCSTR = {}
tCSTR = {}
cPFRall = {} #include for all components
cCSTRall = {}

#The key ’f’ refers to ’feed’
rxRate = lambda conc: rxRateFun(conc, rc) #conc is state vector, c = [cp, tau]
```

```
cPFR[’f’], tPFR[’f’] = artools.PFRtrajectory(rxRate, cf,
        tEnd[’PFR’], numPoints = 300)#1e3h integration time was found suitable
#for the biomass to reach equilibrium
cPFRall[’f’] = solveMoleBalance(cPFR[’f’], rc)

#respect non-negativity constraint
cPFR[’f’] = cPFR[’f’][cPFRall[’f’].min(axis=1) >= epsNonNeg, :]
cPFRall[’f’] = cPFRall[’f’][cPFRall[’f’].min(axis=1) >= epsNonNeg, :]
tPFR[’f’] = cPFR[’f’][:, -1]

cCSTR[’f’], tPFR[’f’] = artools.CSTRlocusFast(rxRate, cf,
        tEnd[’CSTR’], numPoints = 300)
cCSTRall[’f’] = solveMoleBalance(cCSTR[’f’], rc)

#respect the non-negativity constraints
cCSTR[’f’] = cCSTR[’f’][cCSTRall[’f’].min(axis=1) >= epsNonNeg, :]
cCSTRall[’f’] = cCSTRall[’f’][cCSTRall[’f’].min(axis=1) >= epsNonNeg, :]
tCSTR[’f’] = cCSTR[’f’][:, -1]



#start plotting
fig = {}
ax = {}

if True: #wheter or not to make the concentration vs time plots

        #check for more than one solution of the CSTR
        tv, cv, res = artools.residual1DimCSTReq(rxRate,
                cf[0], 45, 200, numPoints = 100)
        fig[’resCSTR’] = plt.figure(’resCSTR’)
        ax[’resCSTR’] = fig[’resCSTR’].add_subplot(111,
                projection = ’3d’)
        ax[’resCSTR’].set_xlabel(’t␣[h]’)
        ax[’resCSTR’].set_ylabel(’cGuess␣[g/L]’)
        ax[’resCSTR’].set_zlabel(’Residual␣CSTR␣design␣equation’)
        ax[’resCSTR’].plot_surface(tv,cv,res)
        xl = ax[’resCSTR’].get_xlim()
        yl = ax[’resCSTR’].get_ylim()
        xv, yv = np.meshgrid(xl, yl)
        sol = np.zeros(xv.shape)
        ax[’resCSTR’].plot_surface(xv,yv,sol, alpha = .2)
        ax[’resCSTR’].plot_surface(tv,cv,res)
        np.savetxt(dirCSV + ’/solCSTRtv.csv’, tv, delimiter = ’,’)
        np.savetxt(dirCSV + ’/solCSTRcv.csv’, cv, delimiter = ’,’)
        np.savetxt(dirCSV + ’/solCSTRres.csv’, res, delimiter = ’,’)
        plt.close(fig[’resCSTR’])


        #Plot concentration as a function of time
        fig[’cVst’] = {}
        ax[’cVst’] = {}
        #Make the biomass and substrate plot
        fig[’cVst’][’X’] = plt.figure(’Xvst’)
        ax['cVst']['X'] = fig['cVst']['X'].add_subplot(111)
        #ax['cVst']['X'].set_title('Biomass and glucose')
        ax['cVst']['X'].set_xlabel(r'$\tau$␣[h]')
        ax['cVst']['X'].set_ylabel('$C_X$␣[g/l]', color = 'C4')
        ax['cVst']['X'].plot(tPFR['f'], cPFRall['f'][:, x],
                c = 'C4', label = 'PFR')
        ax['cVst']['X'].scatter(tCSTR['f'], cCSTRall['f'][:, x],
                c = 'C4',
                marker = 'x', s = 15, label = 'CSTR')
        ax['cVst']['S'] = ax['cVst']['X'].twinx()
        ax['cVst']['S'].set_ylabel('$C_S$␣[g/l]',
                color = 'C1')
        ax['cVst']['S'].plot(tPFR['f'], cPFRall['f'][:, s],
                c = 'C1')
        ax['cVst']['S'].scatter(tCSTR['f'], cCSTRall['f'][:, s],
                c = 'C1',
                marker = 'x', s = 15,)
        handlePFR = mpl.lines.Line2D([], [], c = 'k',
                label = 'PFR')
        handleCSTR = mpl.lines.Line2D([], [], c = 'k',
                marker = 'x', linewidth = 0, label = 'CSTR')
        ax['cVst']['S'].spines['left'].set_color('C4')
        ax['cVst']['S'].spines['right'].set_color('C1')
        ax['cVst']['X'].tick_params(axis = 'y', colors = 'C4')
        ax['cVst']['S'].tick_params(axis = 'y', colors = 'C1')
        #ax['cVst']['X'].grid(True)
        ax['cVst']['X'].legend(handles = [handlePFR, handleCSTR],
                loc = 'right')

        #Butyric acid and acetic acid plot
        fig['cVst']['ba'] = plt.figure('bavst')
        ax['cVst']['ba'] = fig['cVst']['ba'].add_subplot(111)
        ax['cVst']['ba'].set_xlabel(r'$\tau$␣[h]')
        ax['cVst']['ba'].set_ylabel('Concentration␣[g/L]')
        ax['cVst']['ba'].plot(tPFR['f'], cPFRall['f'][:, ba],
                c = 'C3')
        ax['cVst']['ba'].scatter(tCSTR['f'],
                cCSTRall['f'][:, ba], c = 'C3',
                marker = 'x', s = 15)
        ax['cVst']['ba'].plot(tPFR['f'],
                cPFRall['f'][:, aa], c = 'C8')
        ax['cVst']['ba'].scatter(tCSTR['f'],
                cCSTRall['f'][:, aa], c = 'C8',
                marker = 'x', s = 15)
        handleBa = mpl.patches.Patch(color = 'C3',
                label = 'HBu')
        handleAa = mpl.patches.Patch(color = 'C8',
                label = 'HAc')
        handleCSTR = mpl.lines.Line2D([], [], c = 'k',
                linestyle = '', marker = 'x', markersize = 5,
                label = 'CSTR')
        handlePFR = mpl.lines.Line2D([], [], c = 'k',
                label = 'PFR')
        #ax['cVst']['ba'].grid(True)
        ax['cVst']['ba'].legend(handles = [handleBa,
                handleAa, handlePFR, handleCSTR])

#Make the AR plot
#Decorate
labels = ['C_S', r'\tau']

fig['AR'] = plt.figure('AR')

ax['AR'] = fig['AR'].add_subplot(111)

ax['AR'].set_xlabel('$' + labels[0] + '$' + '␣[g/L]')
ax['AR'].set_ylabel('$' + labels[1] + '$' + '␣[h]')


#Plot the (kinetic) AR. Start with PFR from the feed point
lineWidthSecImportance = .1 # for plotting reactors which is
#not part of the AR boundary
ax['AR'].plot(cPFR['f'][:, 0], tPFR['f'], c = 'r',
        linewidth = lineWidthSecImportance)
ax['AR'].scatter(cCSTR['f'][:, 0], tCSTR['f'], c = 'b',
        marker = 'x', label = 'CSTR', s = 5)#CSTR from the feed

"""
Make and draw the convex hull. Make sure that the hull is
containing a vertical line from
(max(X),y_data)-(max(X),max(ylim)) and from (0,0)-(0, max(ylim))
Steps:
        1) Gather all data
        2) Compute the convex hull
        3) Plot the convex hull

"""
#data gathering
cDataAll = cf
for key in cPFR.keys():
        cDataAll = np.vstack((cDataAll, cPFR[key]))
for key in cCSTR.keys():
        cDataAll = np.vstack((cDataAll, cCSTR[key]))

# Add maximum points (actual max x-value and the
#y-limit of the current axis, as well as the line
#from the feed point to max y-lim)
maxXdata = np.amax(cDataAll[:, 0])
maxYlim = ax['AR'].get_ylim()[1]
maximumPoints = np.array([[maxXdata, maxYlim],
        [cf[0], maxYlim]])
cDataAll = np.vstack((cDataAll, maximumPoints))#add the new points

#computation
hull = artools.convhull(cDataAll, incrementalBool = True)
# print(cCSTR['f'])
makeOptimCSTRindexPlot = False
if makeOptimCSTRindexPlot:
    optimCSTRIndex, fig[’optimCSTRIndex’],
        ax[’optimCSTRIndex’] = artools.optimCSTRforTau2D(hull,
            cCSTR[’f’], rxRate, tEnd[’PFR’],
            makePlot = makeOptimCSTRindexPlot)
else:
    optimCSTRIndex = artools.optimCSTRforTau2D(hull,
        cCSTR[’f’], rxRate, tEnd[’PFR’],
        makePlot = makeOptimCSTRindexPlot)

# Make the new PFR trajectory from the optimal CSTR outlet
#concentration. Also expand the plot
cPFR[’2’], tPFR[’2’] = artools.PFRtrajectory(rxRate,
    cCSTR[’f’][optimCSTRIndex, :],
    tEnd[’PFR’] - cCSTR[’f’][optimCSTRIndex, 1])

cPFRall[’2’] = solveMoleBalance(cPFR[’2’], rc)

#respect non-negativity constraint
cPFR[’2’] = cPFR[’2’][cPFRall[’2’].min(axis=1) >= epsNonNeg, :]
cPFRall[’2’] = cPFRall[’2’][cPFRall[’2’].min(axis=1) >= epsNonNeg, :]
tPFR[’2’] = cPFR[’2’][:, -1]

ax[’AR’].plot(cPFR[’2’][:, 0], cPFR[’2’][:, 1], c = ’r’,
    label = ’PFR’)

#Include the mixing line
mixingLine = np.vstack((cf, cCSTR[’f’][optimCSTRIndex, :]))
ax[’AR’].plot(mixingLine[:, 0], mixingLine[:, 1], c = ’k’,
    label = ’Mixing␣line’, linewidth = 0.5)

cDataAll = np.vstack((cDataAll, cPFR[’2’], mixingLine))


#Set the limits on the figure to include
#only the first quadrant (only positive values)
ax[’AR’].set_xlim((0, ax[’AR’].get_xlim()[1]))#set 0 as
#the first value, keep automatic value for xmax
ax[’AR’].set_ylim((0, ax[’AR’].get_ylim()[1])) #same for ymax

"""
Make a quiver plot of the rate vectors on the plot.
In this section, we’ll work in a X-Y space since
it’s two dimensions.

Here: x-dimension is cp, y-dimension is tau
"""
xlim = ax[’AR’].get_xlim()
ylim = ax[’AR’].get_ylim()
npPlt = 15 #number of points in each direction the rate
#vector will be evaluated in each direction.
Xp = np.linspace(*xlim, num = npPlt)# *xlim unpacks the data
#(tuple) in xlim. p stands here for "points"
Yp = np.linspace(*ylim, num = npPlt)

```

```
Xm, Ym = np.meshgrid(Xp, Yp)#Make a meshgrid. Don’t
#include the points on the limit of the
#axes on the figure

#compute the rate vector in each point in the meshgrid
VxMesh = np.zeros(Xm.shape) #V stands for velocity in a quiver plot
VyMesh = np.zeros(Ym.shape)

for i in range(len(Xm)):
        for j in range(len(Xm[0,:])):
                cxi = Xm[i, j]#Obtain the current concentrations of X and p
                cyi = Ym[i, j]
                #Check for non-negativity constraint on all the concentrations
                cAlli = solveMoleBalance(np.array([[cxi, cyi]]), rc)
                if np.all(cAlli >= epsNonNeg): #Check non-negativity constraint
                        ri = rxRate(np.array([cxi, cyi])) #compute the rate vector.
                        #Units for the rate vector is dcdt. However,
                        #it is not required to multiply the rate vector
                        # with a timestep, since the vector
                        #is going to be normalized soon anyway
                        dcx = ri[0] #rate vector for X (decomposed
                        #concentration vector)
                        dcy = ri[1] # same for p
                        dcxn = dcx/np.sqrt(dcx**2 + dcy**2)#normalized vectors
                        dcyn = dcy/np.sqrt(dcx**2 + dcy**2)
                        VxMesh[i, j] = dcxn
                        VyMesh[i, j] = dcyn
                else:
                        VxMesh[i, j] = np.nan
                        VyMesh[i, j] = np.nan

ax['AR'].quiver(Xm, Ym, VxMesh, VyMesh,
        color = 'k', alpha = 0.25)

#Compute the convex hull again so that
#the plot includes all the extreme points

maxXdata = np.amax(cDataAll[:, 0])
maxYlim = ax['AR'].get_ylim()[1]
maximumPoints = np.array([[maxXdata, maxYlim],
        [0, maxYlim]])
cDataAll = np.vstack((cDataAll, maximumPoints))#add new points
hull = artools.convhull(cDataAll) #this is the true convex hull
#do the plotting
h = ax['AR'].fill_between(cDataAll[hull.vertices,0],
        cDataAll[hull.vertices,1], maxYlim,
        alpha = 0.2)

h.set_edgecolor(h.get_facecolor()) #sets the edgecolor
#to the same as the facecolor



#Make a concentration vs time plot for all
```

```python
#the components given the optimal residence time
fig['optCvst'] = plt.figure('optCvst')
ax['optCvst'] = fig['optCvst'].add_subplot(111)
ax['optCvst'].set_xlabel(r'$\tau$␣[h]')
ax['optCvst'].set_ylabel('Concentration␣[g/L]')
myColors = ['C1', 'C4', 'C8', 'C3']
myLabels = ['S', 'X', 'HAc', 'HBu']
myLegend = []
for i in range(len(cPFRall['f'][0, :])-1):
        ax['optCvst'].plot(np.array([tCSTR['f'][0],
                tCSTR['f'][optimCSTRIndex]]), # mixing feed-cstr
                np.array([cCSTRall['f'][0, i],
                        cCSTRall['f'][optimCSTRIndex, i]]),
                color = myColors[i], linestyle = '--')
        ax['optCvst'].scatter(tCSTR['f'][optimCSTRIndex], #outlet CSTR
                cCSTRall['f'][optimCSTRIndex, i],
                color = myColors[i], marker = 'x', s = 30)

        ax['optCvst'].plot(tPFR['2'], cPFRall['2'][:, i],
                color = myColors[i]) #PFR from cstr out

        myLegend.append(mpl.patches.Patch(myColors[i],
                label = str(myLabels[i]), #colors for legend
                FaceColor = myColors[i]))
#attach legends to the different linestyles
myLegend.append(mpl.lines.Line2D([], [], color = 'k',
        linestyle = '--', label = 'Mixing␣line'))
myLegend.append(mpl.lines.Line2D([], [], color = 'k',
 label = 'PFR'))
myLegend.append(mpl.lines.Line2D([], [], color = 'k',
        marker = 'x', markersize = 5, LineWidth = 0,
        label = 'CSTR'))
ax['optCvst'].legend(handles=myLegend)





#save the data to csv files so that it may be
#used in different plotting software
np.savetxt(dirCSV + '/dataPFRf.csv', cPFR['f'],
        delimiter = ',')
np.savetxt(dirCSV + '/dataPFRallf.csv', cPFRall['f'],
        delimiter = ',')
np.savetxt(dirCSV + '/dataCSTR.csv', cCSTR['f'],
        delimiter = ',')
np.savetxt(dirCSV + '/dataCSTRall.csv', cCSTRall['f'],
        delimiter = ',')
with open(dirCSV + '/optimCSTRindex.txt', 'w') as f:
        f.write(str(optimCSTRIndex))
f.close()
np.savetxt(dirCSV + '/dataPFR2.csv', cPFR['2'],
        delimiter = ',')
np.savetxt(dirCSV + '/dataPFRall2.csv', cPFRall['2'],
        delimiter = ’,’)
np.savetxt(dirCSV + ’/dataCall.csv’, cDataAll,
        delimiter = ’,’)
np.savetxt(dirCSV + ’/hullVertices.csv’, hull.vertices,
        delimiter = ’,’)
np.savetxt(dirCSV + ’/dataGridX.csv’, Xm,
        delimiter = ’,’)
np.savetxt(dirCSV + ’/dataGridY.csv’, Ym,
        delimiter = ’,’)
np.savetxt(dirCSV + ’/dataGridVx.csv’, VxMesh,
        delimiter = ’,’)
np.savetxt(dirCSV + ’/dataGridVy.csv’, VyMesh,
        delimiter = ’,’)
# np.savetxt(’datarCSTR.csv’, rCSTR, delimiter = ’,’)


ax[’AR’].legend(loc = ’upper␣left’).get_frame().set_linewidth(0.5)
# for key in fig.keys(): #This procedure
#probably don’t work if the fig[’Cvst’]-plots are not uncommented
#       fig[key].savefig(dirPlots + ’/’ +
#       str(fig[key].canvas.get_window_title()),
#               format = ’eps’)
#       fig[key].savefig(dirPlots + ’/’ +
#       str(fig[key].canvas.get_window_title()) + ’.pdf’)
# # # mpldatacursor.datacursor()
plt.show()
plt.close(’all’)
```