A MODEL-BASED APPROACH TO EVALUATING CHROMATOGRAPHY CAPTURE STEPS

Jakob Ulmestig

Environmental Engineering, LTH, 07/11/2017

Examiner: Bernt Nilsson,
Professor at the Department of Chemical Engineering,
Lund University

Supervisor: Niklas Andersson,
Postdoc at the Department of Chemical Engineering,
Lund University

ABSTRACT

Recent advances in upstream production of biopharmaceuticals have not yet been matched by equivalent advances in the downstream processing. As chromatography is currently the primary downstream processing method in production of biopharmaceuticals new methods for multicolumn processes are being developed. This paper investigates one such method, the three-column periodic counter-current (3C-PCC) process, through computer models and laboratory experiments. The process is optimized with respect to scheduling and the effect of feed concentration is studied.

3C-PCC is shown to have limited to no benefit from decreased flowrates, and the effects of the feed concentration shows clear signs of the internal scheduling limitations that arise when the feed concentration rises above a certain value. The results show an increase in resin utilization but a lower productivity for the multicolumn process compared to the base case batch process.

PREFACE

This report is the result of a degree project in Advanced Process Technology at the Department of Chemical Engineering at Lund University, where it provides an initial evaluation of a multicolumn capture process.

The author would like to thank Bernt Nilsson for providing an interesting subject, and being the examiner of this project.

The author would further like to thank Niklas Andersson for being the supervisor, providing insights when necessary and keeping the project moving.

Lastly, the author would like to thank Joaquín Gomis Fons for providing help with Orbit and getting started in the lab, and Anton Löfgren for insights concerning Orbit and Unicorn.

Jakob Ulmestig
November 2017
jakob.ulmestig@gmail.com

TABLE OF CONTENTS

INTRODUCTION

Chromatography is a chemical separation process which is frequently used in the pharmaceutical industry to remove impurities from the product after it has been produced.
In chromatography different compounds, dissolved in a mobile phase, are passed over a stationary phase and are separated based on the different compounds interaction with it. This makes it possible to separate the dissolved compounds based on different characteristics, such as hydrophobicity, size, and charge. As chromatography can consistently deliver high-purity products it is currently the primary downstream processing method in biopharmaceutical production [1].

Recent improvements in upstream processes has resulted in greatly increased fermentation productivity and titers [2]. As these improvements have not been matched by similar improvements in downstream processing this has resulted in bottlenecks in downstream processing [1]. Novel operation schemes connecting multiple columns, such as CaptureSMB [3] and sequential multicolumn chromatography [4] [5] are showing potential, especially as when these continuous operations are integrated with continuous upstream processing [6]. As the FDA states that the advantages of smaller equipment, fewer steps and improved on-line monitoring gained by moving towards a continuous process leads to better defined, characterized, and ultimately safer drugs a there are no regulatory reason why this move towards continuous processes should be avoided [1].

The downstream processing of a biomolecule is typically separated in several steps, see Figure 1 [1]. After a biologically derived therapeutic protein has been produced the product is present in a mixture with other proteins, either secreted from the host organism or released as the host has been lysed. As the properties of these impurities varies significantly the separation is divided into two main parts; the capture part and the polishing part.

The goal of this study is to evaluate new advances in chromatography techniques by studying a three-column periodic counter-current process (hereafter shortened as 3C-PCC). This study will focus on the capture step, and the system modelled will be based on a one component system calibrated to lysozyme.

WHAT IS THE CAPTURE STEP?

The purpose of the capture step is to reduce the working volume of the solution and remove the majority of non-product-related impurities [1]. This means that the capture step only separates the product from the impurities which does not adsorb to the stationary phase of the column, making separation easy. As such, the capture step is typically performed using step elution [1]. The concentrated product solution is then sent to further downstream processing, where the target protein is further purified [1].

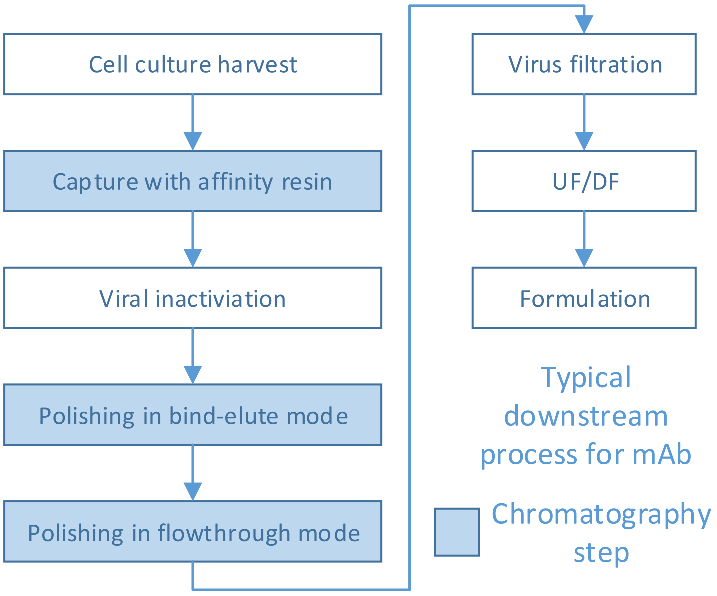

Cell culture harvest
Capture with affinity resin
Viral inactiviation
Polishing in bind-elute mode
Polishing in flowthrough mode
Virus filtration
UF/DF
Formulation
Typical downstream process for mAb
Chromatography step

Figure 1: Typical downstream processing for mAb. Adapted from Steinebach et. al. [1]

Due to mass transfer resistance there is a concentration profile within a column during the loading phase, which leads to loss of product before the static binding capacity is reached [6]. This is referred to as breakthrough, as shown in Figure 2, and is a limiting factor of batch chromatography [6]. Capture steps are designed for high yield, and have their feed load adjusted to prevent breakthrough [1]. In batch chromatography the capture step is typically loaded to 90% of 1% dynamic binding capacity, i.e. 90% of the load at which the product breakthrough is 1% [1] [2]. Traditionally this has led to chromatography processes being oversized to enable a decreased flowrate, thereby increasing the resin utilization [6]. As capture processes are designed for high yield this leads to low resin utilization and productivity [1].

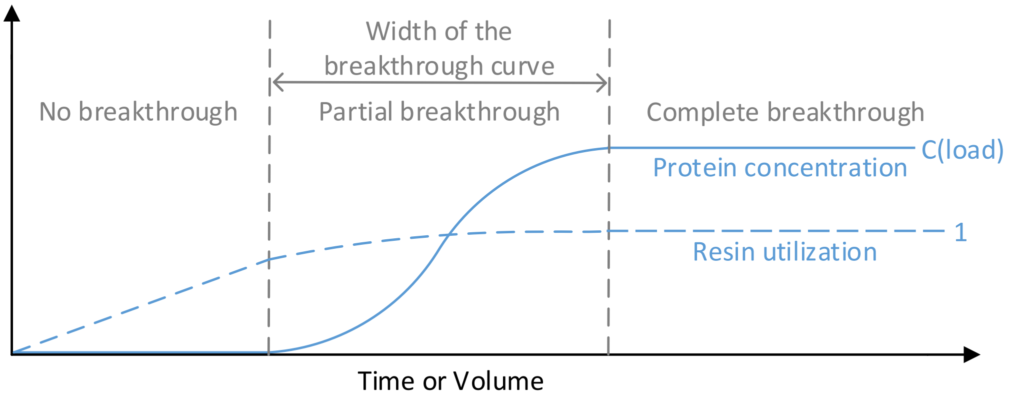


Figure 2: Illustration of breakthrough on a single component system

Computer simulations have been used for many years as a tool for investigating multicolumn processes [2]. In this way the cost of the often-expensive product can be reduced [2], as multicolumn processes typically require large volumes of a feed to be investigated. Previous studies have concluded that fitting batch breakthrough experiments has been sufficient for predicting the performance of a two-column process [2].

This study will model the chromatography process using MATLAB.

THEORY

Continuous counter-current chromatography has shown improved resin utilization and decreased buffer consumption by introducing internal recycling over multiple columns with the same resin [1] [2] [6]. The scheduling of a 3 column PCC process is shown in Figure 3 below:

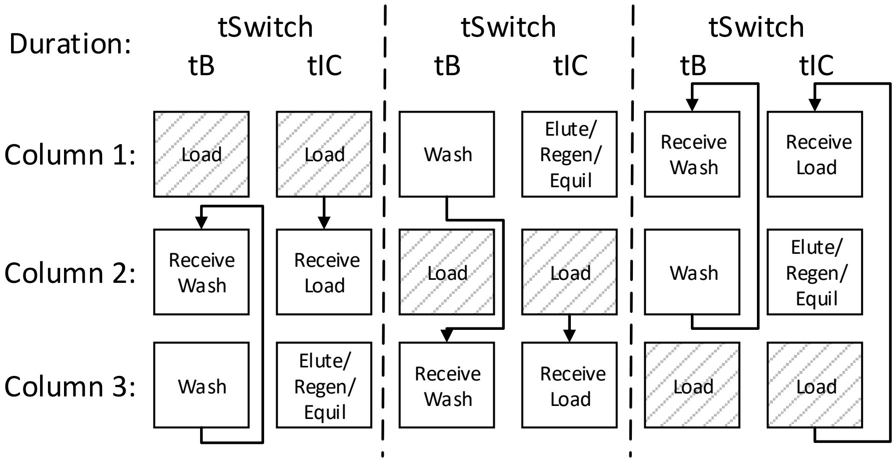


Figure 3: Scheduling of a 3 column PCC operation. Adapted from Godawat et. al. [6]

As the breakthrough in a PCC process is passed onto a second column with the same stationary phase breakthrough does not limit the resin utilization in the same way as in a batch process, for a given yield [1] [6]. This means that a column can be loaded beyond the limitations of batch chromatography [1], approaching full saturation, i.e. to its static binding capacity [6]. As the process is significantly less sensitive to the width of the breakthrough curve there is less of a need to decrease the flowrate to narrow the breakthrough curve, increasing the productivity of the process [6]. As any number of columns can be used in PCC the optimal number of columns for a given process can be adjusted to the width of the breakthrough curve [1].

RESIN UTILIZATION

As resins are used for a set number of cycles before they are replaced with virgin resin [1] [6] the increase in resin utilization per cycle also increases the amount of product produced for each unit of resin used. Being that the buffer used per cycle is generally unaffected by the resin utilization [1] a higher resin utilization also leads to reduced buffer consumption [1] [2] [6]. These factors improve the economics of the process, as resin and buffer consumption are important cost factors for downstream processing [6].

INTERCONNECTED WASH

As multicolumn processes can be loaded up to saturation the composition of the liquid phase inside the column will approach the composition of the feed at the end of the loading [6]. Thus, the yield of the process would suffer if the wash phase was directly discarded [6]. As such it is important for multicolumn processes to recycle the wash phase as well as the load phase [1] [6]. In the 3 column PCC setup shown by Godawat et. al. [6] this is solved by recycling the wash at the cost of not recycling the initial part of the loading, see Figure 3. This does not affect the yield of the process, but imposes a limitation on the process that the product breakthrough must start beyond this point.

RECOVER-REGEN PHASE

Any number of columns can be used in a PCC process, but the optimal number is generally determined by the duration of the recover-regen zone and the width of the breakthrough curve [1]. In this report, a 3 column PCC-scheme will be evaluated. A schematic of the scheduling is shown in Figure 3.

METHOD

MATHEMATICAL MODEL

The mathematical description of the chromatography column is based on the finite volume method [7] [8].

The chromatography column is approximated to have a perfect radial mixing and can therefore be modelled as a one-dimensional row of compartments. These compartments were modelled as homogenous, i.e. some mesoscopic behavior such as film mass transfer and pore diffusion were not included in the model.

The transport equation for the liquid phase are derived using a mass balance. For this mass balance, the liquid is assumed to have constant density, which means that isothermal conditions are necessary. It also assumes that the volume is a packed bed, and is therefore affected by both the column- and packing porosities. An external pressure on one side is also assumed to cause a flow through the volume, and it is assumed that the dispersion caused by the material in the packed bed can be described as diffusion along the direction of the flow. The mass balance is described by:

$$\begin{aligned} Rate\ of\ accumulation &= \quad Inflow \quad - \quad Outflow \\ \varepsilon\ A\ \Delta z\ d(\bar{c}_z)/dt &= (F_z\ c_z - \varepsilon\ A\ N_z)\ -\ F_{z+\Delta z}\ c_{z+\Delta z} - \varepsilon\ A\ N_{z+\Delta z} \end{aligned} \tag{1}$$

where A is the cross-section area, F is the flowrate, N is the dispersion and c is the concentration. Eq. (1) describes how the average concentration between z and $z + \Delta z$ changes over time as a function of the in- and outflows of said compartment. ε signifies the total void of the packed column, calculated from the column porosity, ε_c, and packing porosity, ε_p, as follows:

$$\varepsilon = \varepsilon_c + (1 - \varepsilon_c)\ \varepsilon_p \tag{2}$$

If both sides of the mass balance are divided by $\varepsilon\ A\ \Delta z$ and Δz goes to 0, we are left with:

$$\frac{\partial c}{\partial t} = -\frac{F}{\varepsilon\ A}\frac{\partial c}{\partial z} - \frac{\partial N}{\partial z} \tag{3}$$

As N describes the dispersion due to the packing and can be described as diffusion N can be written as $N = -D_{ax}\ \frac{dc}{dz}$. Using this the rewritten mass balance can be changed into:

$$\frac{\partial c}{\partial t} = D_{ax}\ \frac{\partial^2 c}{\partial z^2} - \frac{F}{\varepsilon\ A}\ \frac{\partial c}{\partial z} \tag{4}$$

The boundary condition at the inlet of the chromatography column was set to Dirichlet, i.e. the concentration at the inlet was set to a fixed value:

$$c(t,0) = c_{in} \tag{5}$$

With the value of c_{in} varies over the course of the simulation to portray the different steps of the chromatography process. The boundary condition at the outlet was set to no flux, i.e. the concentration within the chromatography column was not affected by the concentrations past the outlet. In mathematical term this is described as shown below:

$$\frac{dc(t,L)}{dz} = 0 \tag{6}$$

The initial state of the column is defined by the user input.

The interaction between the liquid and stationary phases are modelled in each compartment using the Langmuir isotherm as described below:

$$r = k_{ads}\, c\, (q_{max} - q) - k_{des,0}\, s^{\beta}\, q \tag{7}$$

The equation is rewritten as shown in [9]:

$$r = k_{kin}(H_0\, s^{-\beta}\, c(1 - q/q_{max}) - q) \tag{8}$$

Using the relations $H_0 = (k_{ads}\, q_{max})/k_{des,0}$ and $k_{kin} = k_{des,0}\, s^{\beta}$. The isotherm is rewritten in this way to make the parameters, now H_0, β, q_{max} & k_{in}, more closely related to peak behavior as k_{in} describes the width of the peak, and $H = H_0\, s^{-\beta}$ describes at what time the peak will elute.

Now the derived reaction rate r can be included in the previously derived transport equations for the liquid phase to get the complete model of the chromatography column. Assuming that the reaction only occurs on the packing the full domain equation becomes:

$$\frac{\partial c}{\partial t} = D_{ax}\, \frac{\partial^2 c}{\partial z^2} - \frac{F}{\varepsilon\, A}\, \frac{\partial c}{\partial z} - \frac{1 - \varepsilon_c}{\varepsilon}\, r \tag{9}$$

CODE STRUCTURE

The initial program was divided into three layers; a main script, ‘*ColumnOperation*’ and ‘*ColumnModel*’. See Figure 17 in the Appendix for a schematic of the code structure of the original simulator.

COLUMNMODEL-SCRIPT

ColumnModel assembles the mathematical model of the chromatography column. The function of the script can be summed up in a few points; it unpacks the values fed to it, checks if the current time is during the elution phase and adjusts the inlet salt concentration accordingly, calculates the adsorption rate for each compartment based on current concentrations, solves the domain equation & outputs the results.

COLUMNOPERATION-SCRIPT

The *ColumnOperation*-script handles all the calculations that can be made outside of the ODE-solver. These can be divided into pre- and post-processing. The preprocessing consists of calculating the necessary variables from the user input data as well as performing as much of the calculations as possible outside of the ode-solver. This includes, among other things, preparing the dispersion and convection matrices used in the domain equation in *ColumnModel*. The discretization is performed using ‘*FVMtools*’, which is a toolbox developed by Bernt Nilsson for the course *KETN01 – Process Simulation* at Lund University [10]. The convection matrices are discretized using 2-point backwards approximation, and the dispersion matrices are discretized using a 3-point central approximation. The ‘*FVMtools*’-toolbox was also used to calculate the Jacobian matrix for the ode-solver.

After the preprocessing, the *ColumnOperation*-script calls the *ColumnModel*-script using the ode-solver ‘*ode15s*’. Each step of the chromatography process is simulated separately, meaning that *ColumnOperation* calls *ColumnModel* five times; one each for the loading, washing, elution, regeneration and equilibration steps. The final state of each step is used as the initial state of the following. The ode-solver was limited to positive answers, to prevent the occurrence of negative concentrations in the simulation. To decrease the computation time the concentrations are scaled. The concentrations, in both mobile & stationary phase, are all scaled by their start value at the respective steps of the chromatography operation. Any concentrations that start at zero are scaled by 1 to prevent division by zero.

In the post processing, the resin utilization is calculated for the column. This is calculated by summing the amount of adsorbed protein in the column for each time step after which they are divided by the maximum capacity of the column for the given inlet concentration, normalizing the amount of adsorbed protein to one. The post processing also extracts all concentrations for the final compartment for all time steps, which based on the no flux-boundary condition is the same as the concentrations in the outlet, as well as all the concentrations in all the compartments for the final time step, giving the final state of the column.

MainScript

The main script is a way for the user to state the values for the used variables. In it the user defines all the physical and operational variables, such as the parameters used in the Langmuir isotherm (*Eq. 8*), scheduling and initial column states. It then calls *ColumnOperation* to run the simulation. The final part of the main script handles the graphical presentation of the calculated data.

Calibration experiments

For the simulated model to correlate to the specific case of interest the model parameters must be calibrated to the system. For this to be possible calibration data from experiments is necessary.

The calibration experiments are performed on an GE ÄKTA Explorer. The protein used is lysozyme in a distilled water suspension of 200 mM sodium phosphate. Buffer A consists of distilled water with 200 mM sodium phosphate, and buffer B consists of buffer A with the addition of 0.5 M sodium chloride. The experiments used 1 ml HiTrap SP HP columns from GE Healthcare Life Sciences [11].

The GE ÄKTA Explorer is controlled using an external controller; Orbit. Orbit is a research software developed using the OPC interface of the native control system (Unicorn 6.4), implemented in Python. The chromatography process is then executed following the instruction in a script. [12] [13]

Four calibration experiments are made, and a fifth one for validation. As the purpose of the simulation was to simulate a chromatography processes with full breakthrough the calibration experiments also include full breakthrough to improve fit of the loading. As such the recovery part of the experiments becomes independent of the loading part, and variations in both the loading and recovery could be part of the same experiments.

The two variables varied for the loading part of the experiments were the flowrate and the feed concentration. The two variables tested were flowrate and the steepness of salt gradient in the elution. The values used for each experiment are shown in Table 1 below:

Table 1: Calibration and validation experiment conditions

		Calibration data				**Validation data**
Experiment #		***1***	***2***	***3***	***4***	***5***
Feed concentration	[g/l]	5	5	1	5	3
Loading flowrate	[ml/min]	1	1	1	0.5	0.75
Salt gradient length	[CV]	25	40	30	1	15
Recovery flowrate	[ml/min]	1	1	0.5	1	0.75

CALIBRATION OF THE MODEL

For the program to be able to calibrate to experimental data a few changes were necessary. See Figure 18 in the Appendix for the code structure of the calibration code.

CHANGES TO *COLUMNOPERATION*-SCRIPT

For calibration to be possible the simulation must deliver values which are comparable with the experimental data. The *ColumnOperation*-script is therefore given an input variable for specified measurement times, so that the simulation data set can be directly comparable to the experimental.

CALIBRATION-SCRIPT

A new script was added above the *ColumnOperation*-script to handle the comparison of the data sets and calibration of the parameters. The *Calibration*-script contains two functions; the lower level *calibrationHandler*-function, which takes a number of experimental conditions and runs all the simulations, and the *Calibration*-function for preprocessing of the data and iterating over the *calibrationHandler*-function using the nonlinear curve-fitting function *lsqcurvefit*.

The higher-level *calibration*-function separates the concentration data points from the experimental values and concatenate them into one matrix. The weighting is applied as a simple multiplication of a constant to a data point to either increase or decrease the difference to be minimized during the fitting. No weighting is applied apart from removing the flattened tops of the elution peaks by weighing them as 0. The final adjustment to the data is scaling the parameters and the datasets to decrease the computation time during the fitting and make the relative changes equally sensitive for all the parameters. Both these variables are scaled by their own values, with zeros replaced with ones. This scaling assumes that any errors in the model appears as a fraction of the value.

The *calibration*-function then iterates the lower-level *calibrationHandler*-function through the nonlinear curve-fitting function *lsqcurvefit*, which uses a least-square evaluation, shown below:

$$\min_{p} \sum_{i} \sum_{j} \left(y_{i,j} - \hat{y}_{i,j}\right)^{2}, i \in experiment \ \& \ j \in time \tag{10}$$

where p are the calibrated parameters, y are the experimental values and $\hat{y}$ are the simulation response using the parameters p. The *CalibrationHandler*-function scales back the parameters and takes the experimental conditions and runs the simulation for each scenario. It then extracts the same values as supplied by the *calibration*-function and applies the same weighing and scaling. The simulated dataset is then compared to the experimental one by *lsqcurvefit*, which adjusts the isotherm parameters accordingly.

CHANGES TO *MAINSCRIPT*

Some changes to *MainScript* are necessary as well for the calibration to be implemented. The majority of the changes pertained to loading in and changing the units of the measured data.

The raw experimental data is stored in a separate MATLAB-script, which returned the experimental data for a called experiment.

The protein concentration in the outlet is indirectly measured with a UV-sensor and the salt concentration in the outlet is indirectly measured with a conductivity sensor. As such the units of the measured values are in absorbance and conductivity. For the calibration to work the units of the results calculated by the simulation and the experimental data must have the same unit, so that they are comparable. The experimental data is therefore converted from absorbance & conductivity to concentrations. This was done by using two known concentrations and using linear interpolation to convert the other measured data. Any measured value below zero are set to zero before the conversion was done.

In the case of the protein concentration the concentration at the start of the experiments is zero, and the concentration at complete breakthrough is the same as the feed concentration, which is known. The specific value used is the UV-measurement at the end of the load phase. Note that the measured concentration at the end of the load phase is not affected by the change at the inlet as the absorbance is measured downstream of the column, and is therefore separated from the inlet by more than one column volume. The equation used is Lambert-Beers law shown below [14]:

$$c_{calc.} = \frac{A_{m.}}{A_{m.,Full\ Breakthrough}}\ c_{feed} \tag{11}$$

In the case of the salt, the known salt concentrations used are the end of the equilibration, for pure buffer A, and the highest measured conductivity, for pure buffer B. This assumes that full equilibration occurs in all the experiments and that there are no high outliers in conductivity due to measurement errors. As the salt concentration of pure buffer A is not zero a more complex formula must be used when converting the measured conductivity to a concentration. This formula is shown below:

$$s_{calc.} = \left(\frac{\gamma - \gamma_A}{\gamma_B - \gamma_A}\right)(s_B - s_A) + s_A \tag{12}$$

where γ refers to the conductivity of a given point, γ_A & γ_B refers to the known conductivity of the buffers and s_A & s_B refers to the known salt concentrations of the buffers.

After this a shifted based on the flowrate and a given volume is applied to all the data points, to account of the dead volume in the system.

The experimental datasets are then interpolated to a fix number of data points before calling the *calibration*-script. This is necessary for the different datasets to have a uniform length, which enables storing all the datasets in a single matrix.

The weighting of the experimental data is then specified by the user before the *calibration*-script is called. The calibration is finally validated by comparing an experimental dataset which is not used in the calibration to the simulated results for the same experimental conditions.

FITTING THE ELUTION PEAK

As the UV-sensor naturally has a limitation for the absorbance range it cannot cover the full spectrum of the protein concentration in the chromatography operation. As the protein concentrations in the elution peak exceed this range there are no reliable measurements for the protein concentration at these points. These points are therefore weighted as zero, and thereby excluded from the calibration.

OPTIMIZATION OF THE MULTICOLUMN PROCESS.

After the calibrated parameters are validated the work shifts to simulating a multicolumn process. The approach to simulating a multicolumn process is similar to the approach taken to simulating the different chromatography steps separately, where a new script called *MultiColumnOperation* relates to *ColumnOperation* similarly to how *ColumnOperation* relates to *ColumnModel*. The code structure can be seen in Figure 19 in the Appendix.

HOW WAS THE OPTIMIZATION PERFORMED?

The initial step of optimization was performed by simulating an area with varying t_B & t_{IC}. After this the effect of varying feed concentrations were studied. In this second step t_B & t_{IC} were removed as user determined variable, and were calculated based on the column volumes of the load-, wash-, and recover-regen phase. Using the known volumes t_B & t_{IC} were minimized to maximize productivity. All simulated cases use step elution.

INCREASED INITIAL LOAD

The goal for the multicolumn process is for each cycle to load and elute the equivalent of the static binding capacity on each column. However, as the breakthrough curve has a width, an excess of protein is necessary in the system. As, in an ideal process, the amount of protein sent from one column onto the next equals this amount this excess only needs to be added once. In this case this excess of protein is added to the system on the first cycle of the first column, by extending the load phase by a factor of 1.4. Note that this load can be retrieved on the final elution of the final column, so this excess is not lost in the process.

This is solved by extending the load phase for the first cycle on the first column.

t_B & t_{IC} AS PARAMETERS

In the first scenario t_B & t_{IC} are the varied parameters. This examines the effect of varying the flowrates of the different phases, as the process times are varied with the volumes of each phase set.

CHANGES TO *COLUMNMODEL*-SCRIPT

Two changes are made to the *ColumnModel*-script to accommodate a multicolumn process. The first is to add the ability to have a variable feed, beyond the linear salt gradient of the elution step. As the time steps taken by the ode-solver are unknown until they are taken this is solved by giving a matrix with the concentration data throughout the receive phase, and then interpolate the feed concentration for the given time from the concentration matrix. The second change introduced to the *ColumnModel*-script is the ability to have up to two different flowrates within a single call. This is necessary as the load- and receive phases normally change flowrate as it transitions from t_B to t_{IC}.

CHANGES TO *COLUMNOPERATION*-SCRIPT

The number of chromatography steps in the *ColumnOperation*-script is increased to six to include a receive phase when necessary. As the interpolation function used to determine inlet concentration in *ColumnModel* is unable to handle repeating identical data points any repetition in the receive-vector must be removed. To reduce computation time this was done in *ColumnOperation* rather than in *ColumnModel*.

MULTICOLUMNOPERATION-SCRIPT

Similarly to how the *ColumnOperation*-script is introduced between the *MainScript* and the *ColumnModel*-script to iterate over the separate phases of the chromatography process, a new script, *MultiColumnOperation*, is added between *MainScript* and the *ColumnOperation*-script to handle multicolumn processes. The function of the *MultiColumnOperation*-script can be divided into three parts; scheduling, iterating over the *ColumnOperation*-script and calculating performance of the multicolumn process.

The initial step of the scheduling is to determine if it is possible to make a schedule with the given parameters, notably the column volumes in each step, t_B, t_{IC} & maximum flowrate. If the scheduling is possible the next step is to determine the flowrates for the different steps.

In the case of 3 column PCC the number of equations describing the flowrates during the loading are outnumbered by the number of flowrates to determine below:

$$\begin{aligned}
&Equations:\\
&(I)\ V_{load,tB} + V_{load,tIC} = V_{load}\\
&(II)\ V_{load,tB}/Q_{load,tB} = t_B\\
&(III)\ V_{load,tIC}/Q_{load,tIC} = t_{IC}\\
&Unknowns:\\
&(i)\ V_{load,tB}\\
&(ii)\ V_{load,tIC}\\
&(iii)\ Q_{load,tB}\\
&(iv)\ Q_{load,tIC}
\end{aligned} \tag{13}$$

where V is the loaded volume and Q is the flowrate. Note that the flowrates can vary within a phase here, increasing the number of flowrates to be calculated. The solution chosen for this system is to maximize $Q_{load,tIC}$, i.e. $Q_{load,tIC} = Q_{max,Load}$, and can be calculated based on the schedule and the actual maximum flowrate. The values of the flowrates $Q_{load,tIC}$, $Q_{load,tIC}$, Q_{wash} & $Q_{elu.,reg.,equ.}$ are determined as shown in Figure 4 below:

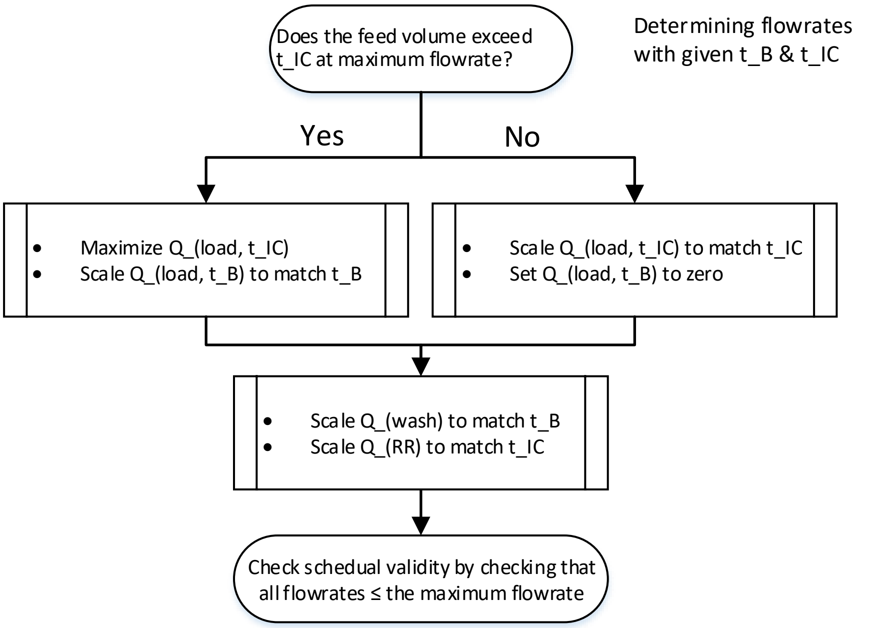

Does the feed volume exceed t_IC at maximum flowrate?
Determining flowrates with given t_B & t_IC
Yes
No
• Maximize Q_(load, t_IC)
• Scale Q_(load, t_B) to match t_B
• Scale Q_(load, t_IC) to match t_IC
• Set Q_(load, t_B) to zero
• Scale Q_(wash) to match t_B
• Scale Q_(RR) to match t_IC
Check schedual validity by checking that all flowrates ≤ the maximum flowrate

Figure 4: Determining flowrates with given tB and tIC

The script then schedules the phases based on t_B & t_{IC}. The initial load phase on column 1 is extended to ensure saturation of column 1 by compensating for the lack of recycled protein loaded onto the column. This initial saturation ensures enough breakthrough that all following column operations can load and elute a product load close to the static binding capacity.

The program then iterates over the columns and cycles, calling *ColumnOperation* for each operation, storing the data from each column in cell matrices. The concentrations for the recycled load and wash are extracted and queued in the *receive*-variable for the relevant columns. Note that the load phase breakthrough and the recycled wash have to be extracted separately, as they are sent to different columns.

As a final step the productivity and yield are calculated for each column operation. For these to be calculated the amount of eluted product must be calculated. This is done using a trapezoidal, numerical integration over the outlet past the start of the elution phase using the MATLAB function *trapz*. The formula used in *trapz* is shown below:

$$\int_a^b f(x)dx \approx \frac{1}{2}\sum_{n=1}^{N}(x_{n+1} - x_n)[f(x_n) + f(x_{n+1})] \tag{14}$$

where a & b are the start & end of the integrated region and $(x_{n+1} - x_n)$ is the spacing between each pair of data points [15] . The productivity per volume of resin is then calculated as the amount of product eluted in a cycle divided by the time of a column operation and the volume of resin in a column.

The yield is calculated by comparing the amount of eluted product to the amount of product loaded onto the column in a given cycle. The amount added to a column is calculated as the feed concentration times the volume loaded onto the column.

CHANGES TO *MAINSCRIPT*

Limited changes are implemented to *MainScript* to accommodate calling *MultiColumn-Operation*. New process variables are introduced; t_B, t_{IC}, $N_{columns}$, N_{cycles} and $RecycleShift$, where the last describes how many columns the recycled load and wash are shifted.

$RecycleShift$ is best illustrated by studying the 3C PCC schematic shown in Figure 3. In the process studied the recycled load is shifted a single column forward, whereas the recycled wash is shifted two columns forward.

The column volumes of the recover-regen part of the process are lumped into a single parameter, CV_{RR}, and three parameters describing the fraction of this which is to be used for elution, regeneration and equilibration, respectively. This change is made as to make the comparison of the column volumes used in the loading, the wash and the recover-regen part easier.

Finally, an option is introduced to the script to solve multiple systems, with a specified variation in t_B & t_{IC}. The results are then presented in grids as productivity, normalized productivity, yield & productivity times yield. The results are also presented in a figure with productivity and yield on the axes.

FEED CONCENTRATION AS A PARAMETER

In this scenario the effects of a varied feed concentration are studied. The scripts used for this are very similar to the ones used for the first multicolumn study, but the differences are noted below.

CHANGES TO *MULTICOLUMNOPERATION*-SCRIPT

The most notable difference in *MultiColumnOperation* is a change in how the flowrates are calculated. This method uses variable t_B & t_{IC}, minimizing both by always maximizing the flowrate of the time limiting step. The process of determining the flowrates and process times is shown in Figure 5 below. τ_L signifies the time of the loading phase and τ_R refers to the time of the wash, elution, regeneration and equilibration phases combined.

As the flowrates are calculated, rather than given by the used, the values of t_B & t_{IC} returned to *MainScript*.

Since the base case batch scenario relies on a constant feed concentration *MultiColumnOperation* is adjusted to also handle single column processes. As there is only a single flowrate in a batch process, that singular flowrate is always the limiting. The flowrate is therefore always equal to the maximum flowrate in a batch process.

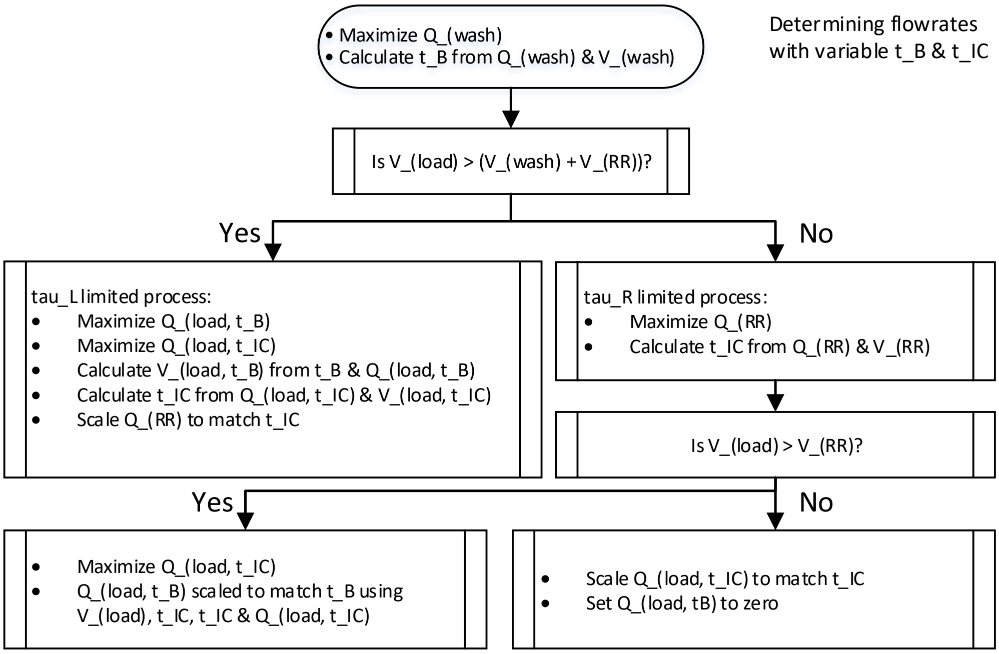

• Maximize Q_(wash)
• Calculate t_B from Q_(wash) & V_(wash)
Determining flowrates with variable t_B & t_IC
Is V_(load) > (V_(wash) + V_(RR))?
Yes
No
tau_L limited process:
• Maximize Q_(load, t_B)
• Maximize Q_(load, t_IC)
• Calculate V_(load, t_B) from t_B & Q_(load, t_B)
• Calculate t_IC from Q_(load, t_IC) & V_(load, t_IC)
• Scale Q_(RR) to match t_IC
tau_R limited process:
• Maximize Q_(RR)
• Calculate t_IC from Q_(RR) & V_(RR)
Is V_(load) > V_(RR)?
Yes
No
• Maximize Q_(load, t_IC)
• Q_(load, t_B) scaled to match t_B using V_(load), t_IC, t_IC & Q_(load, t_IC)
• Scale Q_(load, t_IC) to match t_IC
• Set Q_(load, tB) to zero

Figure 5: Determining flowrates with variable tB & tIC

CHANGES TO *MAINSCRIPT*

In this second rendition of *MainScript* t_B & t_{IC} are not specified, but the feed concentration is varied. Note that since a single variable is varied the grid-solution functionality of the previous case is changed to a vector of solutions instead.
Since t_B & t_{IC} are calculated rather than specified they are also plotted as a function of the feed concentration.

In order to maintain the amount of protein loaded onto any given column when varying the feed concentration, the volumes of the load phase are divided by the factor varying the feed concentration.

VALIDATE THE OPTIMIZED MULTICOLUMN PROCESS IN THE LAB.

The simulated process is then validated by a laboratory experiment. This validation experiment is performed on a GE ÄKTA Pure system. The system setup can be seen in Figure 6, and is controlled using the Orbit controller [12] [13].

The multicolumn validation experiment used the same composition of buffer A & buffer B, and the same type of columns as the batch experiments. The experimental conditions are shown in Table 2 below:

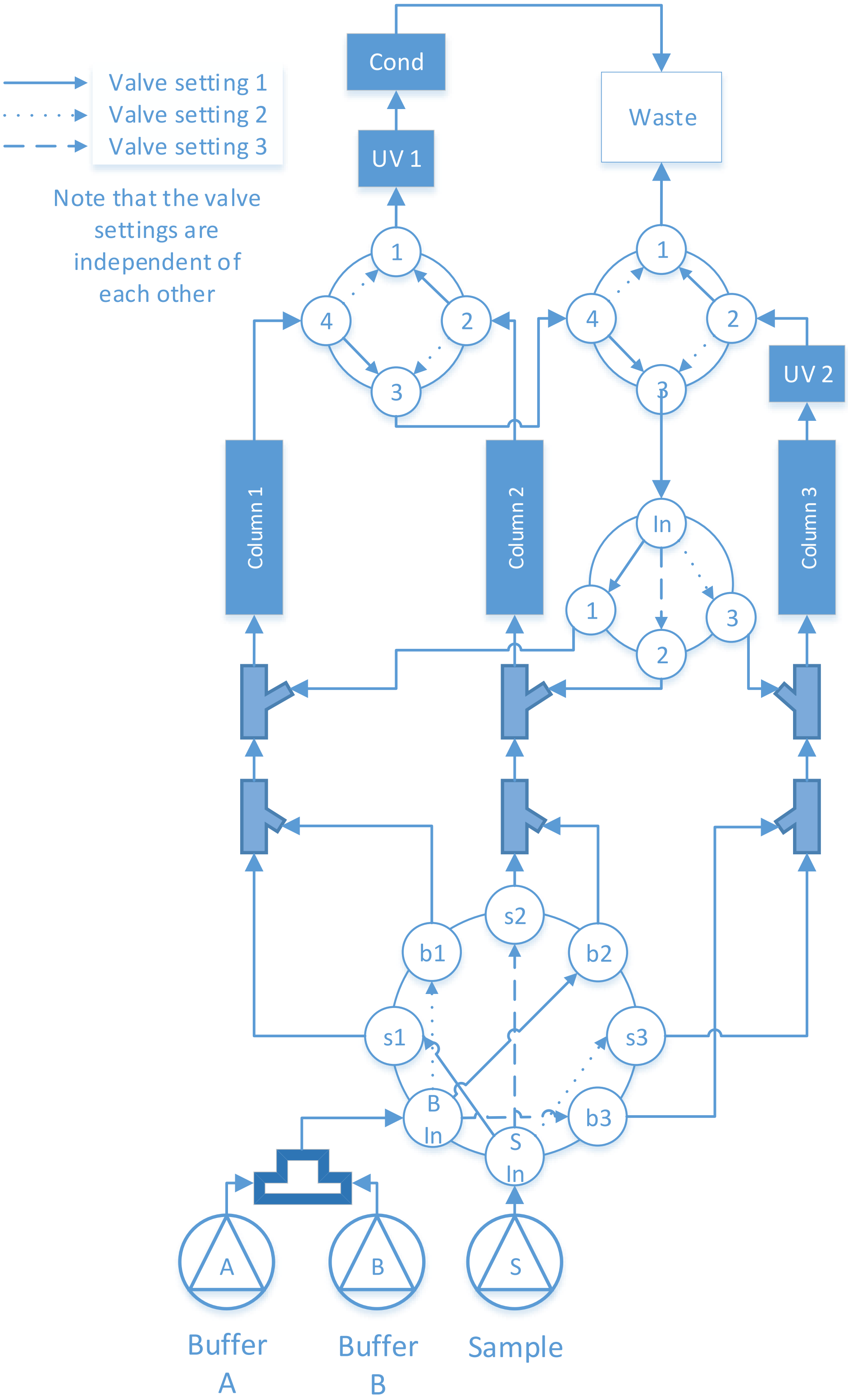


Figure 6: The experimental setup for the validation experiment

Table 2: 3C-PCC validation experiment conditions

		Validation experiment
Feed concentration	[g/l]	3
$\boldsymbol{t_B}$	[min]	10
$\boldsymbol{t_{IC}}$	[min]	20
Flowrates:		
Load, $\boldsymbol{t_B}$	[ml/min]	0.55
Load, $\boldsymbol{t_{IC}}$	[ml/min]	1
Wash	[ml/min]	1
RR	[ml/min]	1
Extra load on 1:st column	[-]	1.6

RESULTS & DISCUSSION

CALIBRATION RESULTS

The results of the calibration and batch validation experiments are shown in Figure 7 and Figure 8. As can be seen the elution peaks of the experiments exceed the measurement range of the UV-sensors, cutting them off. These data points were excluded in the calibration of the parameters; the skipped intervals are shown in Table 3 below. It should also be noted that the loading phase plateaus were consistently below the cutoff concentration.

Table 3: The intervals of the data points excluded from the calibration. Values given in minutes

Experiment:	Excluded values	
	Start:	End:
1	57	68.5
2	55	70.2
3	186.5	210
4	119	137.5

It should be noted that the timespan excluded in experiment 4 continues past the maximized absorbance measurements. This was due to problems with the front of the simulated elution peak calibrating to the back end of the measured elution peak.

The calibration returned the parameter values presented in Table 4 below:

Table 4: Calibrated parameters and standard deviation.

$\boldsymbol{H_0}$ $[-]$	$\boldsymbol{\beta}$ $[-]$	$\boldsymbol{q_{Max}}$ $[\boldsymbol{mol/m^3}]$	$\boldsymbol{k_{kin}}$ $[\boldsymbol{1/h}]$	$\boldsymbol{Standard\ Deviation}$ $[-]$
$3.37144e^{-20}$	8.44790	877722	14.8237	19.2134

The standard deviation shown in Table 4 is calculated from the difference between scaled response of the calibrated model and the scaled experimental data.

The response of the calibrated model is also shown in Figure 7 and Figure 8. Note that Figure 7 shows the same experiments twice, once with a larger concentration span and once with a narrower one. Figure 7 shows that there are no high valued outliers in the measured salt concentration, making our assumption that the highest measured conductivity value corresponds to pure buffer B reasonable. The dead volume of the system was estimated by aligning the start of the elution curve between the experimental data and the simulation response. This resulted in an estimated 2.5 ml dead volume in the experimental setup. A single value was used to shift the entire dataset. As this single value was adjusted to fit the phase change from the wash phase to the elution phase, i.e. from buffer A to buffer B, this single value also includes the dead volume between the buffer storage and the buffer/feed-valve. This means that the adjusted dead volume was overestimated for the switch between buffer and feed. This can be seen when comparing the experimental values and the simulation response in the transition from the loading phase to the wash phase in Figure 7 and Figure 8.

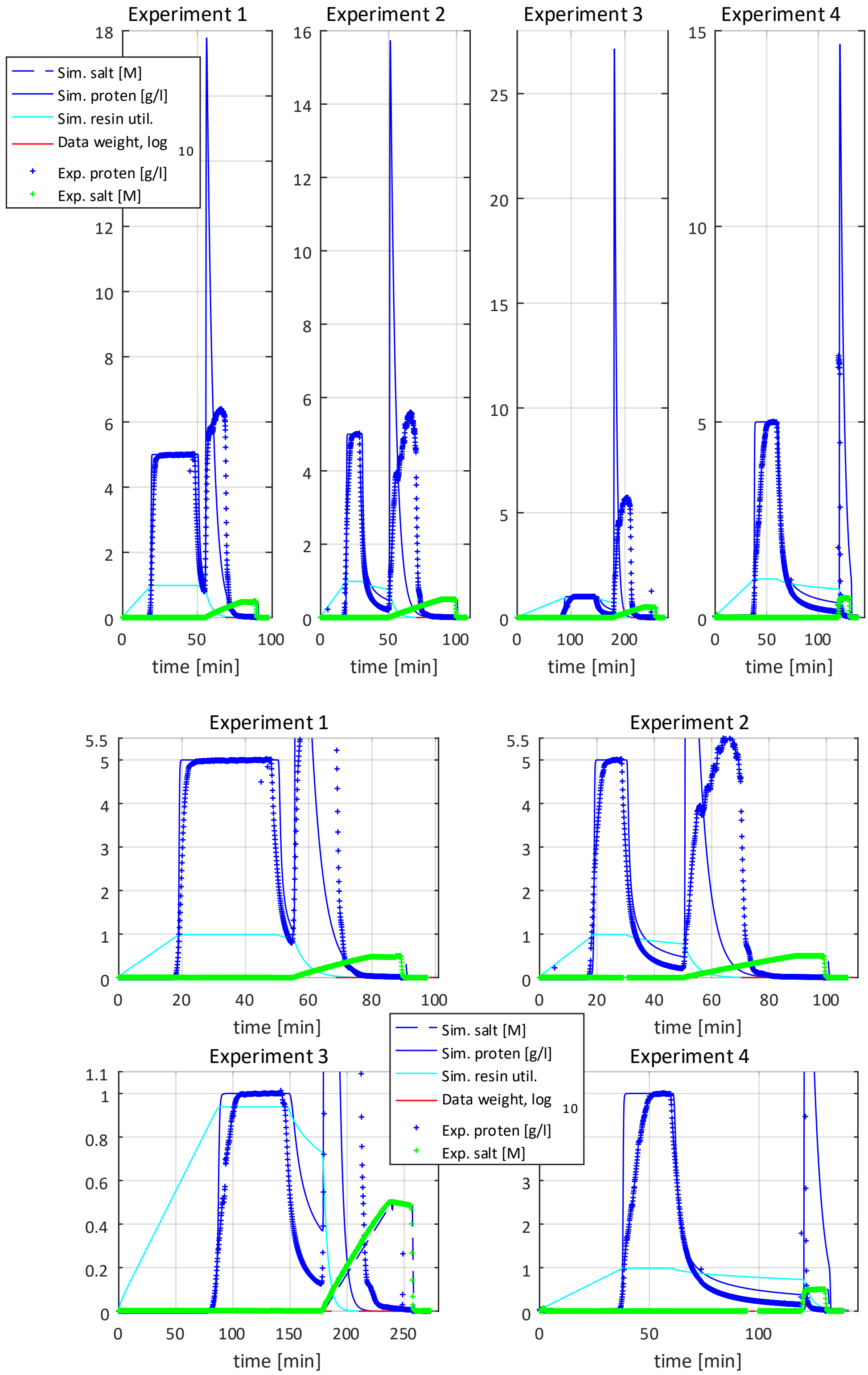


Figure 7: Calibrated model plotted against the experimental values.

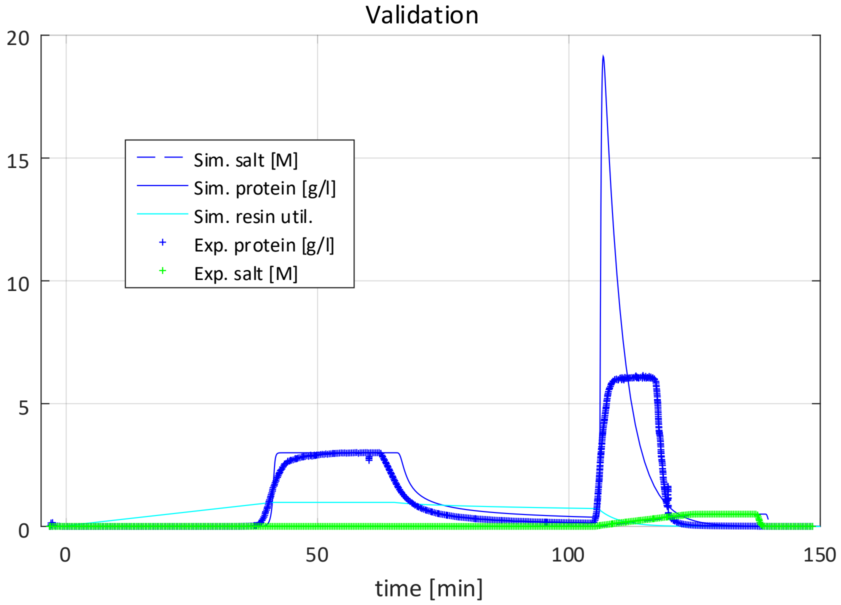


Figure 8: Calibrated model plotted against the validation data.

As can be seen in Figure 7 and Figure 8 the calibrated failed to accurately predict the width of the breakthrough curve. This is a reasonable as a homogenous model was used. Since the width of the breakthrough curve is normally limited by the mass transfer resistance within a system the homogenous model used was unable to accurately evaluate it.

As the resin utilization of the multicolumn process is significantly less sensitive to the width of the breakthrough curve than the batch process, the resin utilization of a simulated batch process will be overestimated more than that of a simulated multicolumn process.

Experiment 1-3, seen in Figure 7, and in the validation experiment, seen in Figure 8, shows that the elution is overestimated when salt gradients are used to elute the protein. If the elution peak in experiment 4 in Figure 7 is observed the opposite can be seen. The simulation response returned a much wider elution peak than experiment 4, which used step elution. This shows that the simulation struggles to accurately represent varying salt gradients at once.

OPTIMIZATION RESULTS

One batch process and one 3C-PCC process of similar yields were then run using the calibrated parameters. The process conditions for the simulations are shown in As can be seen in the second half of Figure 9 the excess protein added to the first column is redistributed to the other two columns through the recycled wash and feed. The figure also shows how there was just enough product breakthrough in the PCC operation to achieve close to full breakthrough, as well as the protein that was bound to the column during the receive phase. Studying the resin utilization during the receive phase closer shows that most of the recycled product is recycled during the first 10 minutes of the receive phase, i.e. all the recycled protein was recycled during the wash. There is a small increase in resin utilization towards the end of the receive phase, which corresponds to the product breakthrough at the end of the loading phase.

Table 5, and a few key performance numbers are presented in Table 6 below. The concentrations over time are shown in Figure 9.

As can be seen in the second half of Figure 9 the excess protein added to the first column is redistributed to the other two columns through the recycled wash and feed. The figure also shows how there was just enough product breakthrough in the PCC operation to achieve close to full breakthrough, as well as the protein that was bound to the column during the receive phase. Studying the resin utilization during the receive phase closer shows that most of the recycled product is recycled during the first 10 minutes of the receive phase, i.e. all the recycled protein was recycled during the wash. There is a small increase in resin utilization towards the end of the receive phase, which corresponds to the product breakthrough at the end of the loading phase.

Table 5: Process conditions for the baseline simulations.

$\boldsymbol{Process}$	$\boldsymbol{Batch}$	$\boldsymbol{3C\text{-}PCC}$	$\boldsymbol{Process}$	$\boldsymbol{Batch}$	$\boldsymbol{3C\text{-}PCC}$
$\boldsymbol{\#\ of\ columns}$ $[-]$	1	3	$\boldsymbol{t\_B}$ $[\boldsymbol{min}]$	N/A	10
$\boldsymbol{Feed\ concentration}$ $[\boldsymbol{g_{product}/l}]$	3		$\boldsymbol{t\_IC}$ $[\boldsymbol{min}]$	N/A	20
$\boldsymbol{V_{receive,t_B}}$ $[\boldsymbol{ml}]$	N/A	10	$\boldsymbol{Q_{receive,t_B}}$ $[\boldsymbol{ml/min}]$	N/A	1
$\boldsymbol{V_{receive,t_{IC}}}$ $[\boldsymbol{ml}]$		20	$\boldsymbol{Q_{receive,t_{IC}}}$ $[\boldsymbol{ml/min}]$		1
$\boldsymbol{V_{load,t_B}}$ $[\boldsymbol{ml}]$	25.16	5.5	$\boldsymbol{Q_{load,t_B}}$ $[\boldsymbol{ml/min}]$	1	0.55
$\boldsymbol{V_{load,t_{IC}}}$ $[\boldsymbol{ml}]$		20	$\boldsymbol{Q_{load,t_{IC}}}$ $[\boldsymbol{ml/min}]$		1
$\boldsymbol{V_{wash}}$ $[\boldsymbol{ml}]$	10		$\boldsymbol{Q_{wash}}$ $[\boldsymbol{ml/min}]$	1	1
$\boldsymbol{V_{RR}}$ $[\boldsymbol{ml}]$	20		$\boldsymbol{Q_{RR}}$ $[\boldsymbol{ml/min}]$	1	1

Table 6: The yield, productivity and resin utilization of the baseline configurations.

$\boldsymbol{Process}$	$\boldsymbol{Yield}$ $[-]$	$\boldsymbol{Productivity}$ $[\boldsymbol{g_{product}/h\ l_{resin}}]$	$\boldsymbol{Max\ Resin}$ $\boldsymbol{Utilization}$ [%]	$\boldsymbol{Resin\ Utilization}$ $\boldsymbol{at\ elution\ start}$ [%]
$\boldsymbol{Batch}$	0.997	73.8	83.7	79.6
$\boldsymbol{3C\text{-}PCC}$	0.996	45.8	97.9	84.0

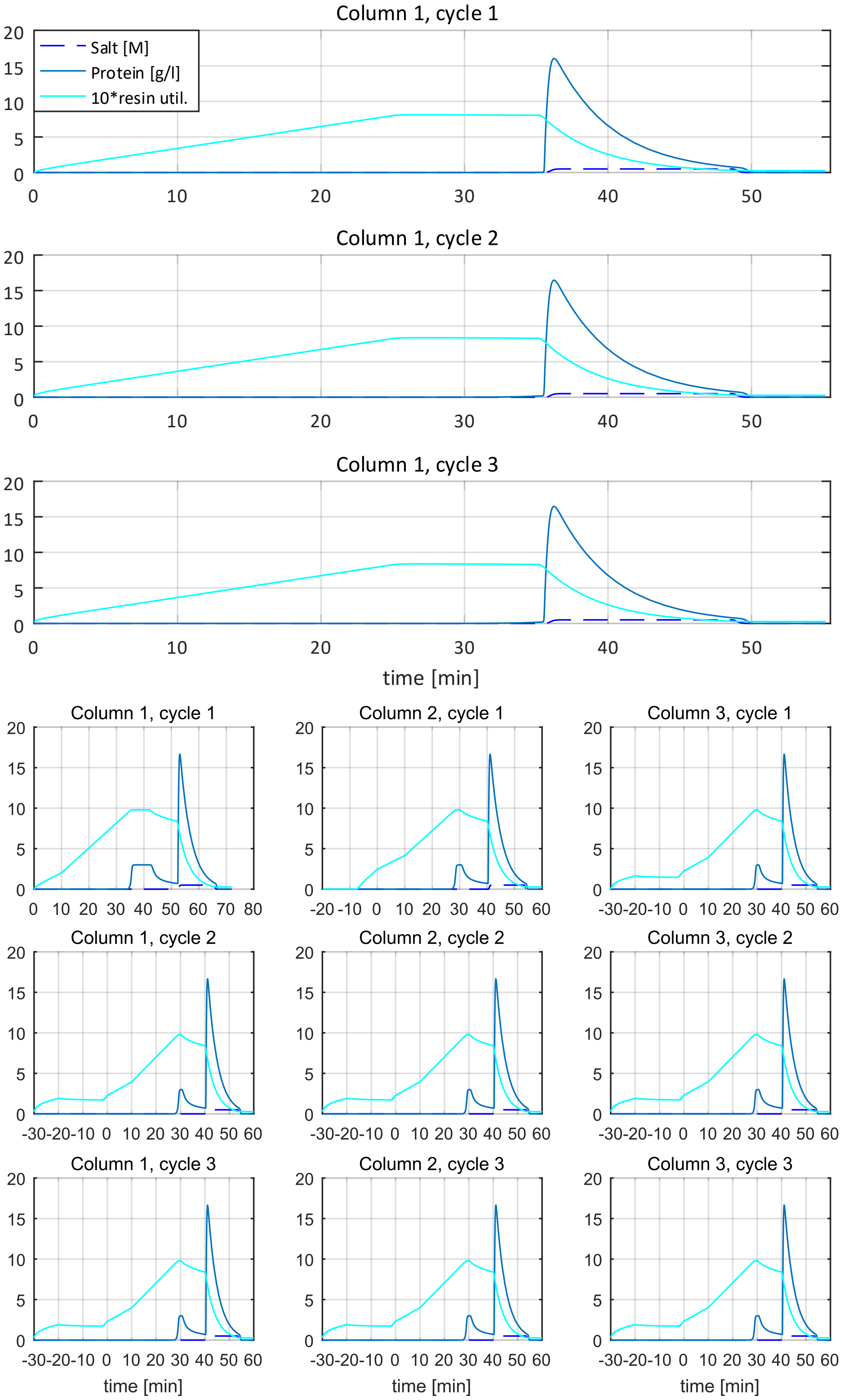


Figure 9: Baseline batch and 3C-PCC operations. Note that the receive phase ends at t = 0 in the multicolumn process

The retrieved values for yield, productivity and resin utilizations shown in Table 6 relates to a stable process, and is therefore not valid for the first and last cycle. The values for a stable process were chosen as the effect of the first and last cycles depends on the number of cycles run.

EFFECT OF VARYING PROCESS TIMES t_B & t_{IC}

The next step was to investigate the effect of t_B & t_{IC} on the yield and productivity. The column volumes applied in each phase was kept constant, and the flowrates were calculated as described under '*MultiColumnOperation*-script'.

The effect t_B & t_{IC} has on the yield can be seen in Figure 10 below. The yield is strictly decreasing as t_B increases, whereas it increases slightly with t_{IC} before it plateaus. Both these behaviors were the result of the load of the stable, baseline process being tuned by hand. As t_B increases Q_{wash} decreases. This gives the bound protein more time to elute during the wash phase, resulting in more protein being washed out, decreasing the resin utilization at the start of the elution phase. Unlike a batch process, the protein that was eluted during the wash phase was not directly lost, but instead captured by the receiving column. This increased the amount of protein sent to the next column, decreasing the amount of product retrieved in the elution phase. This leads to a column behavior where there is a buildup of protein being circulated around the system, decreasing the yield by the amount of protein retained within the system. This behavior is illustrated in Figure 11, where t_B has been extended to 40 min.

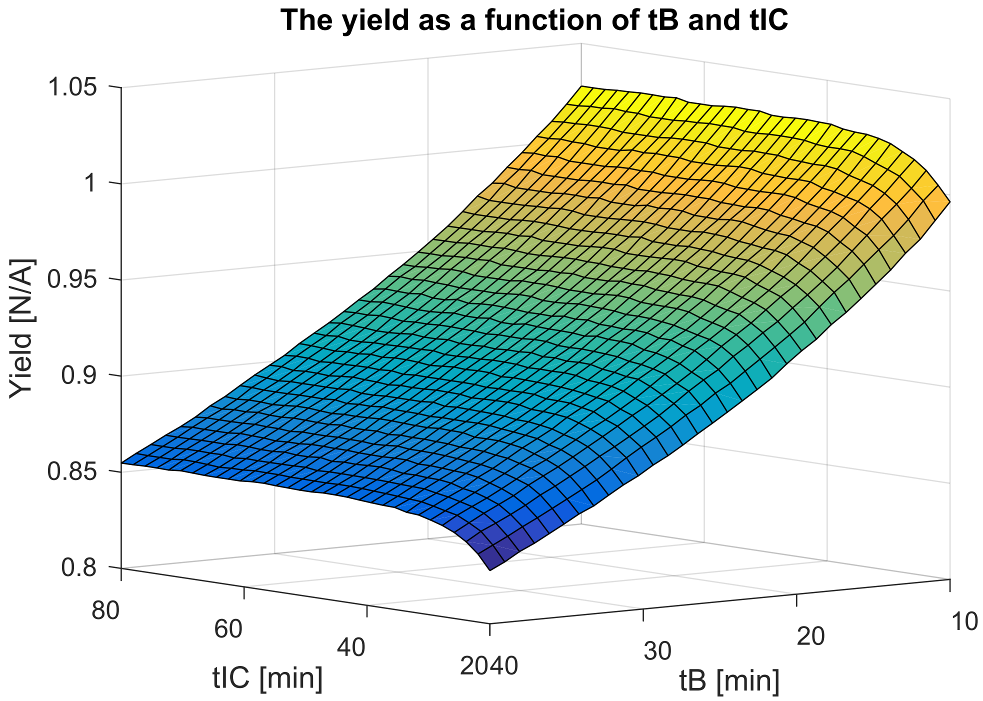


Figure 10: Yield of a 3C-PCC process as a function of tB & tIC

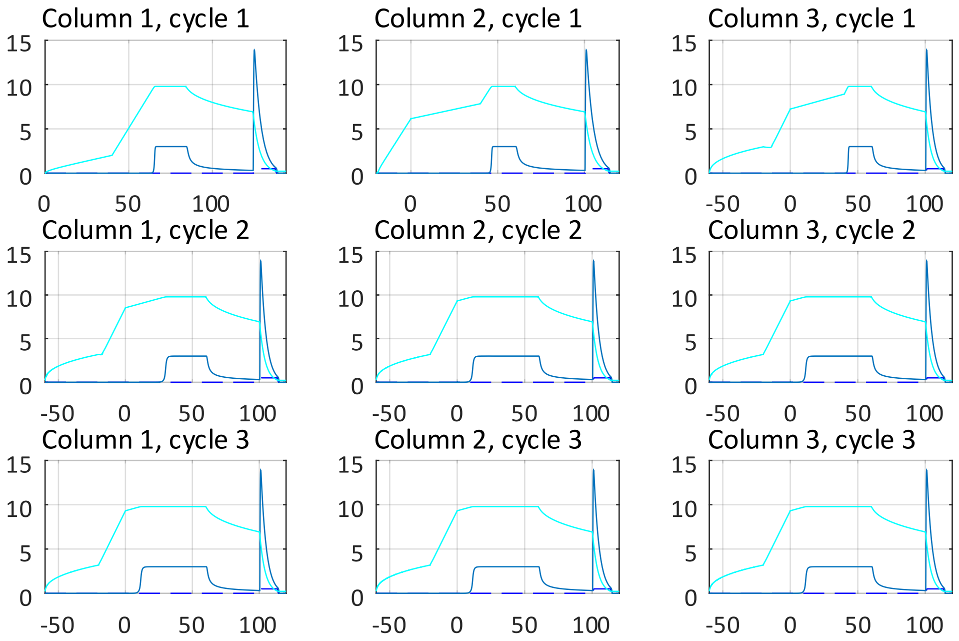


Figure 11: 3C-PCC operation with extended tB. Same legend as Figure 8 above

Conversely, in the case with extended t_{IC} the recover-regen part, i.e. the elution, regeneration and equilibration phases, were extended. This lead to lower flowrates, improving elution. As the baseline process had roughly 2.7 % of the resin still utilized at the end of the regeneration protocol, more protein was eluted per cycle with an increase in t_{IC}, until full regeneration was reached. This is the reason the yield plateaued after a certain increase in t_{IC}. This increase in eluted protein also disturbed the balance of protein loaded versus protein eluted, leading to a depletion of the extra load provided during the initial loading, enabling a yield greater than one. This can be seen in Figure 12, where t_{IC} was extended to 80 min.

The productivity decreased with both increased t_B and t_{IC}, as shown in Figure 13. This was to be expected as the cycle time was increased without increasing the load per cycle. Increasing t_B appeared to have a more significant effect on the productivity, but this was likely an effect of the decreased yield due to product build-up within the system.

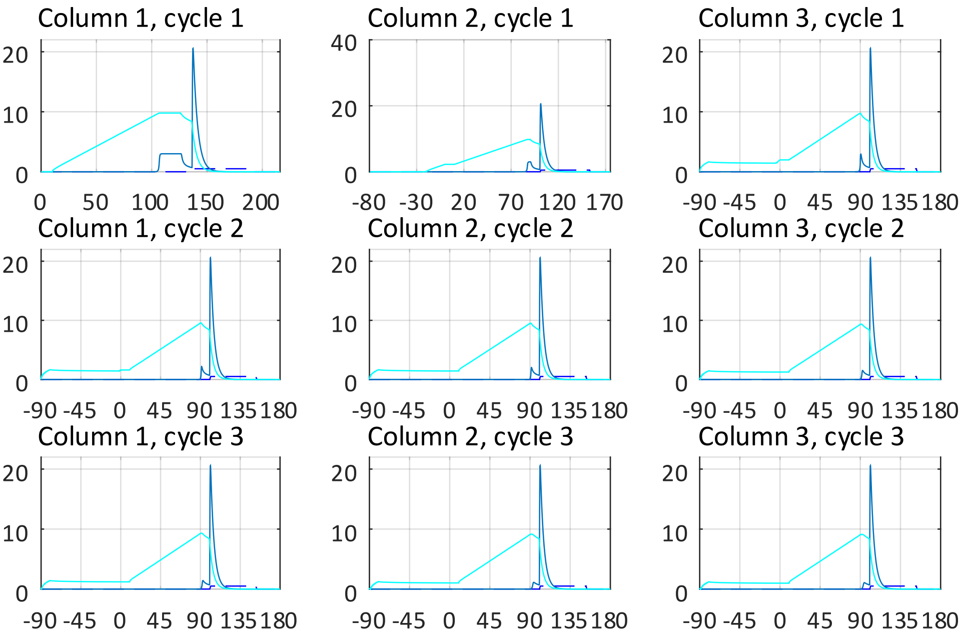


Figure 12: 3C-PCC operation with extended tIC. Same legend as Figure 8 above

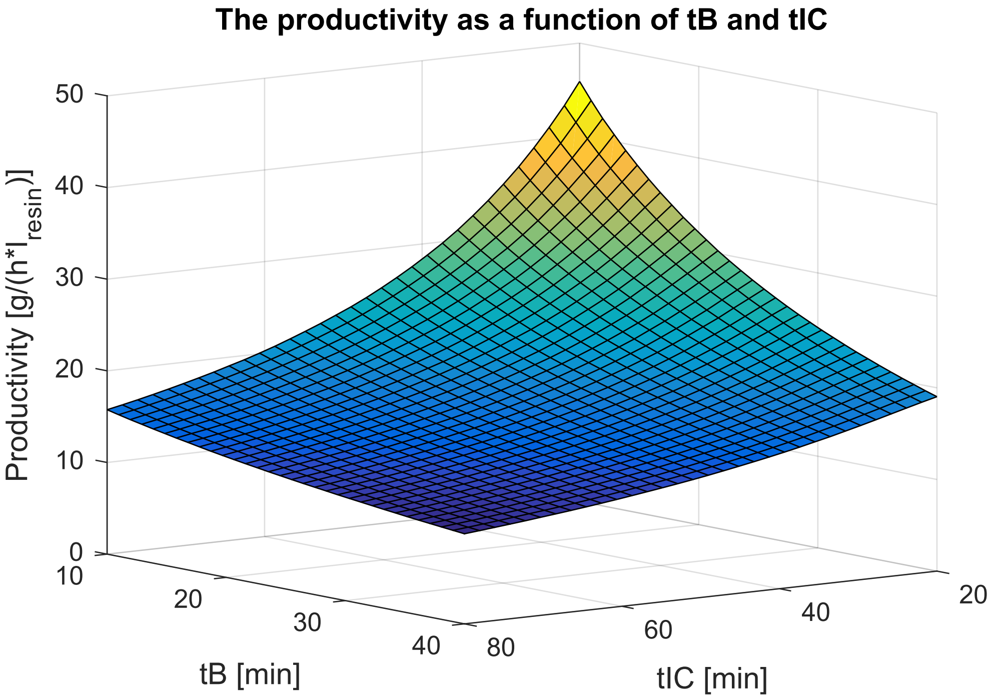


Figure 13: Productivity of a 3C-PCC process as a function of tB & tIC

EFFECT OF VARYING FEED CONCENTRATION

BATCH PROCESS

After the scheduling, the effect of the feed concentration was studied. As can be seen in Figure 14, the productivity and yield both increased with increasing feed concentration for the batch configuration. This increase in yield is due to a small, but constant loss of product during the loading phase. As the volume fed during the loading phase increases, due to decreased product concentration, the more protein was lost which in turn decreased yield.

The productivity increased continuously as the loading volume was decreased. Plotting the yield versus the productivity showed that batch chromatography strictly became better with increased feed concentration.

3C-PCC PROCESS

The 3C-PCC process showed a more complicated relationship with feed concentration due to the coupling of the loading and washing times. If the feed concentration was high enough for the process to be τ_R-limited t_B & t_{IC} was constant. Above this point only a slight improvement in yield is achieved, due to the decrease in volume fed. The productivity does not change at all with feed concentration once the process is τ_R-limited, since the loading phase is coupled to the wash and recover-regen protocol and cannot be reduced past $t_B + t_{IC}$. Therefore, there is a very tight grouping of data points on the right side of the pareto plot.

Once the feed concentration was reduced below $2.55\ \mathrm{g/l}$ the process was no longer limited by τ_R, but rather by the loading. At this point t_{IC} starts to increase to accommodate for the greater load volume, and the same effects are observed as when t_{IC} was increased previously.

VALIDATION OF 3C-PCC SIMULATION

The results of the simulation can be compared to experimental results in Figure 16. The positions of the measuring devices can be seen in Figure 6 in the Appendix. When studying the experimental results, it should be noted that the experimental data are presented in absorbance and conductivity, rather than the protein and salt concentrations. If the experimental setup shown in Figure 6 is studied it is evident that the measurements by UV1 can never be converted to protein concentration unless the process deviates from the PCC schedule. This is because the second known concentration, that at full breakthrough, will never be measured by UV1 unless there is full breakthrough on two columns, or during t_B. Neither of these scenarios should occur if the PCC schedule has been devised correctly. The measured conductivity could be converted to salt concentration. This conversion was not done as the main focus of this study is the protein concentration.

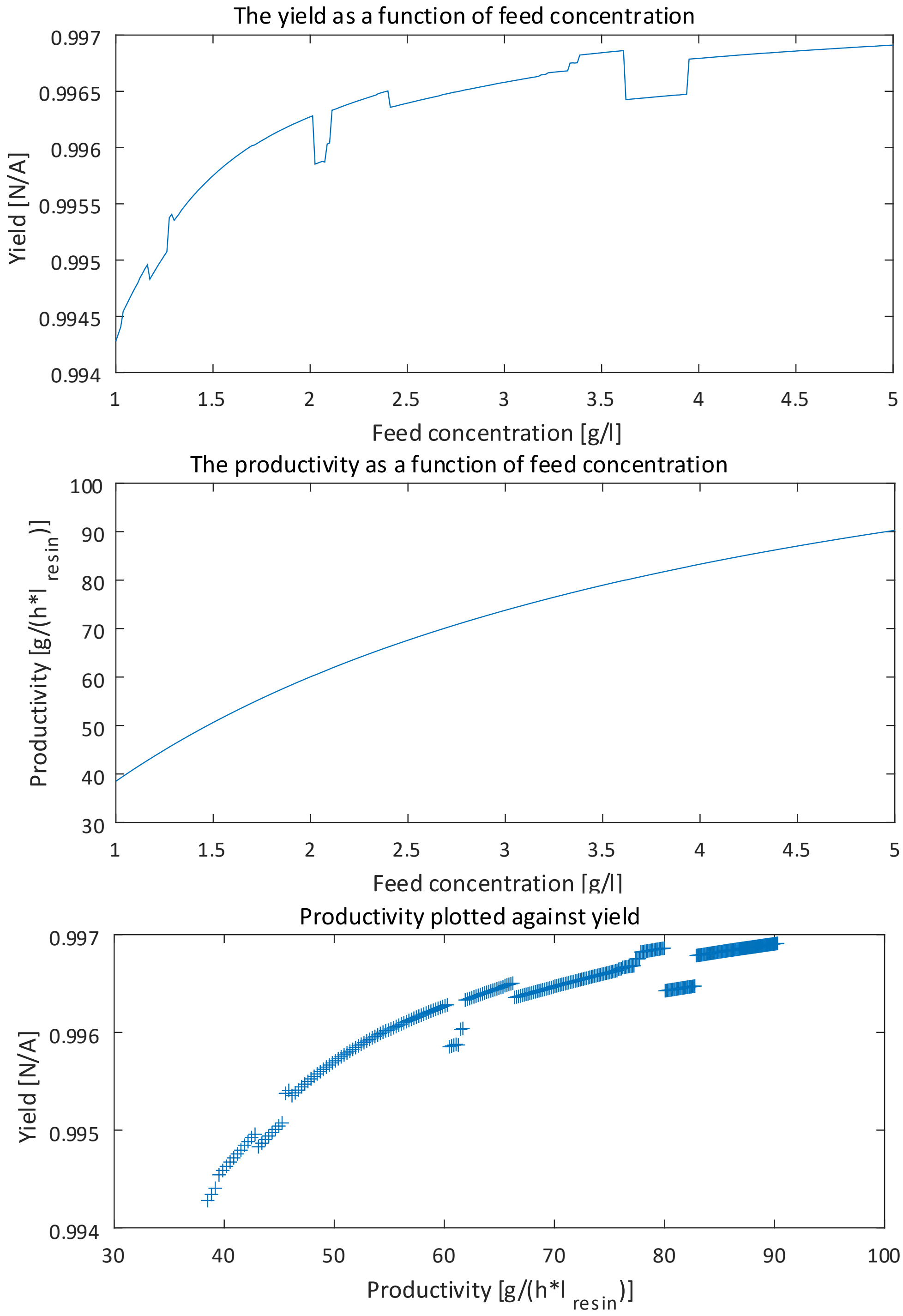


Figure 14: Effect of a varying feed concentration on a batch process

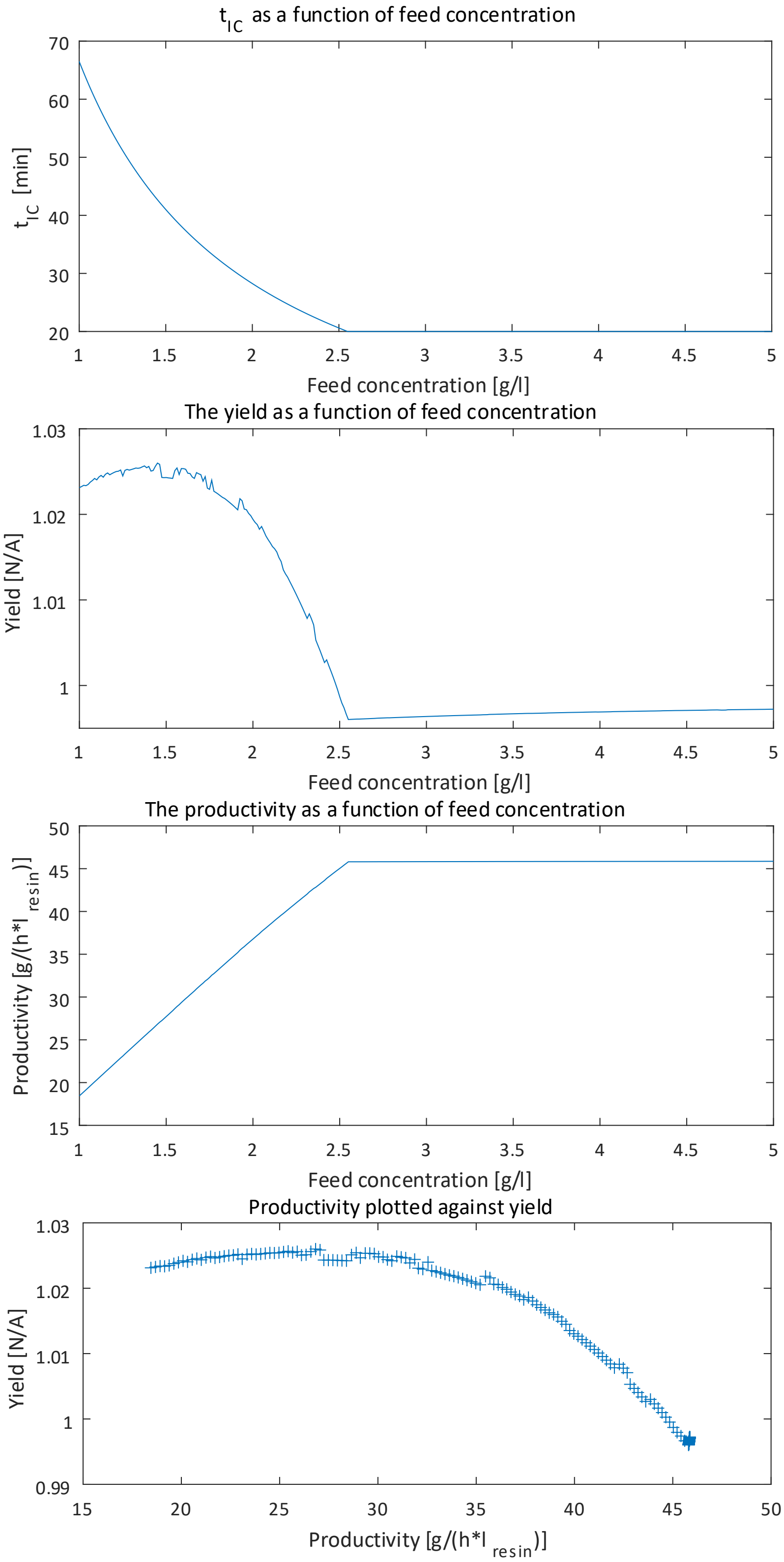


Figure 15: Effect of a varying feed concentration on a 3C-PCC process

It should be noted that the experimental PCC operation started on column 3, as the initial breakthrough curve could be captured by UV2 in this way. This adjustment was made since the initial feeding of the first column is the point in the schedule where the longest time of complete breakthrough occurs. If UV2 in Figure 16, two important facts can be gleaned; the initial breakthrough never stabilizes, i.e. full breakthrough does not occur in the initial extended load phase, and there is no breakthrough recorded on column 3 after the initial loading. As full breakthrough is never reached, the second known concentration needed to convert the measured absorbance to protein concentration is missing, and as such a more in-depth analysis of the simulation performance is not possible. We can, however, still compare the breakthrough curve of the initial load. Product breakthrough starts to occur at the same time in the simulation and the validation experiment. But as the breakthrough curve is much wider in the experimental results, we can draw the conclusion that more protein is adsorbed onto the column in the experiment than in the simulation, i.e. the value of q_{max} is too small. This could also help to explain why there is no product breakthrough during the load phase of the second cycle, as the extra load of the initial loading has been divided between the three columns.

All the experimental elution peaks appear as expected, except for the final peak which appears slightly early. An explanation for this has not been found. Some other instabilities can be found in the experimental data. The elution peak towards the end of the initial loading is most likely due to the dead volume in the buffer pipe leading to column 3 being full of buffer B. As the washing starts buffer A is pumped through the pipe, pushing the trapped buffer B through the column.

The elution peaks of the second column are smaller than the equivalent peaks past the first elution. No explanation for this behavior has been found.

CODE PHILOSOPHY

Multiple different techniques and philosophies were applied to the code to make the code more stable, more versatile and faster.

One of the coding philosophies used to write a more stable and predictable code was to attempt to treat as many variables as possible as immutable variables. As a part of this effort structures has been avoided as much as possible to prevent risking that a function edits a value of a variable used in another script.
The code was also designed to be easy to pick apart, by for example changing the *ColumnModel*-script at the bottom of the hierarchy with minimal impact on the rest of the program. This was a way to futureproof the code, so that as little as possible needs to be rewritten even if major changes are to be implemented.
The computation time was minimized by rewriting as much of the calculations as possible to matrix operation, and by moving as much of the calculations out of ode- and lsqcurvefit-loops. Currently the slowest part of the code is the interpolation necessary to determine the feed concentration during the receive phase. This can however not possible to move out of the ode-loop, as the timesteps taken by the ode-solver are not predefined. The ode- and lsqcurvefit-calls that were necessary were helped by scaling all the relevant input data to ones.

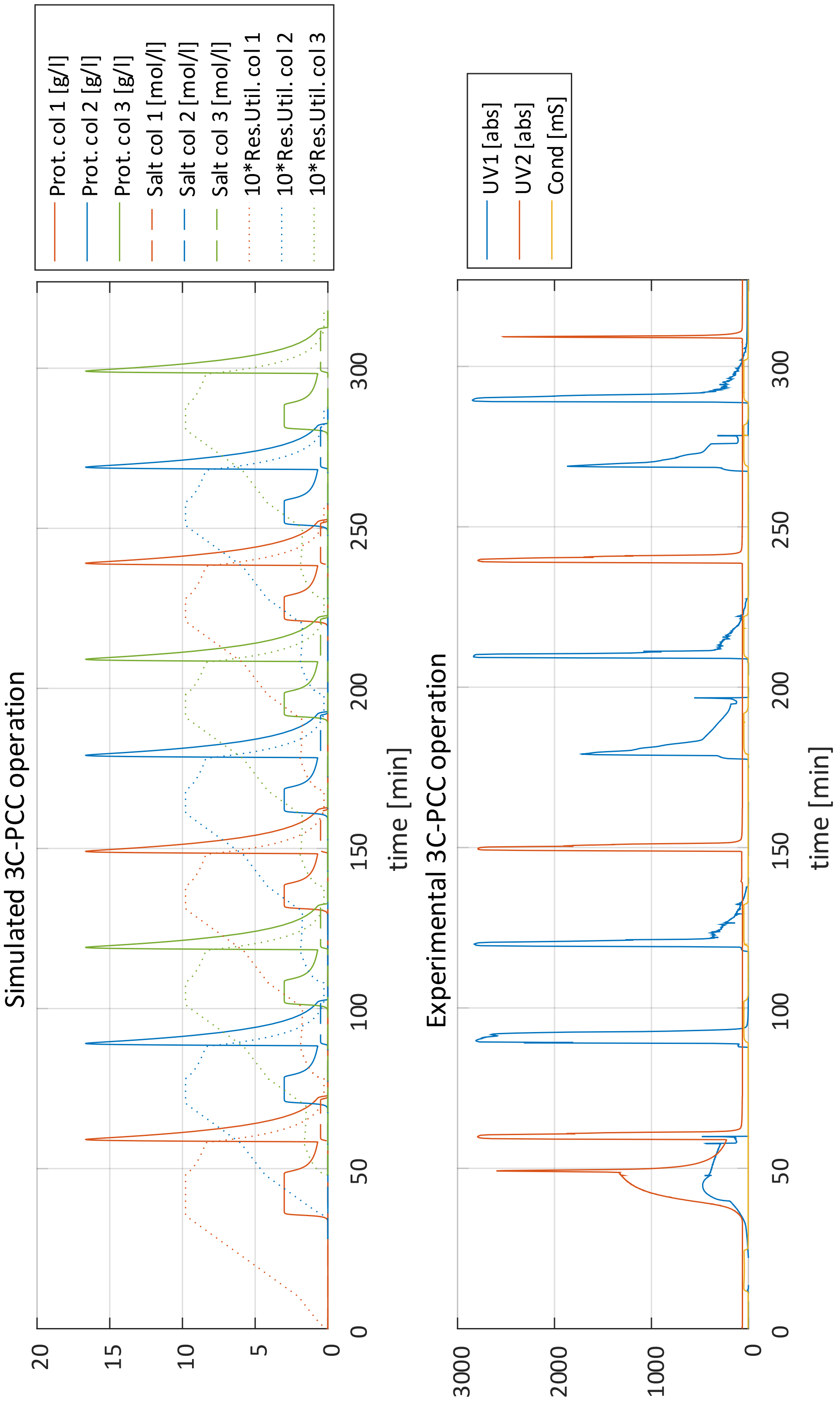


Figure 16: Comparison of 3C-PCC simulation and validation experiment

The choice of storing experimental data in separate files was based on two factors. The first was an attempt to increase the reusability of the code, decreasing the work necessary to change the experimental data used. The second was to decrease the amount of text that a user is forced to keep open in a MATLAB editor, as the large datasets recorded in the lab tends to slow down computers to a halt when they are being edited.

In hindsight it would have been a more elegant solution to simply call the *ColumnModel*-script two times for the receive and load phases than to add support for two separate flowrates within a single call.

The code does not currently calculate the concentration of a cut of the product. This could be implemented in the future with a parameter for a specific yield. The cut could be determined by starting at the highest concentration of the elution peak, and then stepping down the peak ether to the left or the right depending on which has the highest concentration. This could continue until the specified yield is achieved, and then the concentration of the cut could easily be determined.

GENERAL COMMENTS

Table 6 seems to indicate that there is a tradeoff of productivity per resin volume versus resin utilization when choosing between a batch and a 3C-PCC process. The increased resin utilization of the 3C-PCC process aligns with similar results reported by Steinebach [1], Godawat [6] & Baur [2], but Steinebach & Baur reports increased productivity as well. This result could be an effect of the narrow breakthrough curve exhibited by the model, be due to the baseline scenario having a high feed concentration, or a combination of both.

The literature suggests resin and buffer consumptions to be reduced by as much as 40 % [1], but the results in this study only points towards a decrease in buffer and resin consumption of about 5.3 %. This could partly be because a shorter wash has been used, favoring the multicolumn processes, as the elution rate during the wash is dependent on the amount of protein adsorbed to the resin. But considering the vastly different results it is reasonable to suspect that the previously mentioned breakthrough curves width and the high feed concentration also affect this result.

If there is in fact a trade of between productivity and resin utilization this could result in 3C-PCC having a higher investment cost, accounting for both the increased need for pipes & valves and the increase in resin needed to compensate for the decreased productivity per volume resin. But this higher investment cost might be balanced out in the long run in terms of savings due to more product produced per volume resin over its lifetime as well as decreased buffer consumption, with both these effects stemming from the increased resin utilization of the multicolumn process.

The choice of studying a single component system simplifies the problem, but it could be argued that it simplifies it beyond producing useful results. It is for example impossible to adjust the recover-regen protocol after a desired purity, and it is impossible to evaluate the effects of 3C-PCC on purity.

Godawat et. al. suggests a control strategy based on ΔA over the column, i.e. comparing the absorbance just before and just after the column. In this way the load phase can be defined in terms of a percental breakthrough on the column. I believe that this method has significant advantages over the fixed volume-approach which was used in this paper, as this method can adapt in real time to the fluctuations in feed concentration which would undoubtedly occur in a real life process. Using this method would also prevent the trouble with processes becoming ‘unstable’ when the process parameters are varied, either resulting in a build-up or depletion in the recycled protein. The downside of using this approach is that the non-limiting flowrates cannot be slowed down as much, since this could cause a non-limiting step to become limiting if the feed concentration is unusually high or low. But based on the studies in this report I have concluded that Q_{wash} should *never* be decreased, even when it is a non-limiting step. This is due to the functionality of the wash phase; to remove impurities which are inert to the adsorption sites on the stationary phase. As a decrease in Q_{wash} does not affect the removal of these inert impurities, but does increase the amount of product which is eluted into the wash volume, Q_{wash} should never be decreased.

The current scheduling also allows for varying flowrates within the receive and load phases. This decision was made to minimize the risk of breakthrough of the process, as the feed is exposed to more spare capacity at the end of the load phase, when it is recycled onto an empty column, than at the end of the t_B part of the loading. However, as product breakthrough has not been a problem it might be worth more to use a constant flowrate, so that this can be matched to a continuous upstream process as this would eliminate the need for intermittent storage.

CONCLUSIONS

While the multicolumn process exhibits better performance in terms of resin utilization the difference is not as large as is reported in the literature. The productivity per volume resin is better for the batch simulations than for the 3C-PCC simulations. This makes sense based on the very narrow breakthrough curve of the model.

Unable to fit the breakthrough curve properly, the unnaturally narrow breakthrough curve will have benefitted the batch-case in the scenario. More advanced models than homogenous Langmuir necessary to study multicolumn processes.

After studying the effects of slowing down flowrates past the time limiting steps it has been determined that the benefits are marginal, if existent. As such it is always best to minimize the process times t_B & t_{IC}. The flowrate of the non-limiting step can then be reduced by scaling it to match the time-limiting step, unless it is the wash, which should always be at maximum flowrate.

When studying the effects of feed concentration on the batch and 3C-PCC processes, respectively, it was discovered that batch processes always benefit from higher titers, as all the timesteps are independent of each other. With the link between the loading phase and the wash & recover-regen parts of the multicolumn process the 3C-PCC process only benefits from increased titers up to a certain point.

REFERENCES

[1] F. Steinebach, T. Müller-Späth and M. Morbidelli, "Continuous counter-current chromatography for capture and polishing steps in biopharmeceutical production," *Biotechnology Journal,* pp. 1126-1141, 11 2016.

[2] D. Baur, M. Angarita, T. Müller-Späth, F. Steinebach and M. Morbidelli, "Comparison of batch and continuous multi-column protein A capture processes by optimal design," *Biotechnology Journal,* pp. 920-931, 11 2016.

[3] D. Baur, M. Angarita, T. Müller-Späth and M. Morbidelli, "Optimal model-based design of the twin-column CaptureSMB process improves capacity utilization and productivity in protein A affinity capture," *Biotechnol J,* vol. 11, no. 1, pp. 135-145, 2016.

[4] V. Girard, N.-J. Hilbold, C. K. Ng, L. Pegon, W. Chahim, F. Rousset and V. Monchois, "Large-scale monoclonal antibody purification by continuous chromatography, from process design to scale-up," *Biotechnol J,* vol. 213, pp. 65-73, 2015.

[5] C. K. Ng, F. Rousset, E. Valery, D. G. Bracewell and E. Sorensen, "Design of high productivity sequential multi-column chromatography for antibody capture," *FBP,* vol. 92, no. 2, pp. 233-241, 2014.

[6] R. Godawat, K. Brower, S. Jain, K. Konstantinov, F. Riske and V. Warikoo, "Periodic counter-current chromatography - design and operational considerations for integrated and continuous purification of proteins," *Biotechnology Journal,* p. ??, ?? 2012.

[7] B. Nilsson, "Modeling and Simulation III," Dept. of Chemical Engineering, Lund University, Lund, 2016.

[8] R. J. LeVeque, Finite Volume Methods for Hyperbolic Problems, Cambridge: Cambridge University Press, 2002.

[9] W. Melander, Z. el Rassi and C. Horváth, "Interplay of hydrophobic and electrostatic interactions in biopolymer chromatography. Effect of salts on the retention of proteins.," *J. Chromatogr. A,* no. 3, p. 469, 1989.

[10] B. Nilsson, *FVMtools,* Lund: Faculty of Chemical Engineering, Lund University.

[11] GE Healthcare Life Sciences, "GE Healthcare Life Sciences "HiTrap SP HP, 5 x 1 ml"," General Electric, [Online]. Available: http://www.gelifesciences.com/webapp/wcs/stores/servlet/catalog/en/GELifeSciences-se/products/AlternativeProductStructure_17470/17115101. [Accessed 12 10 2017].

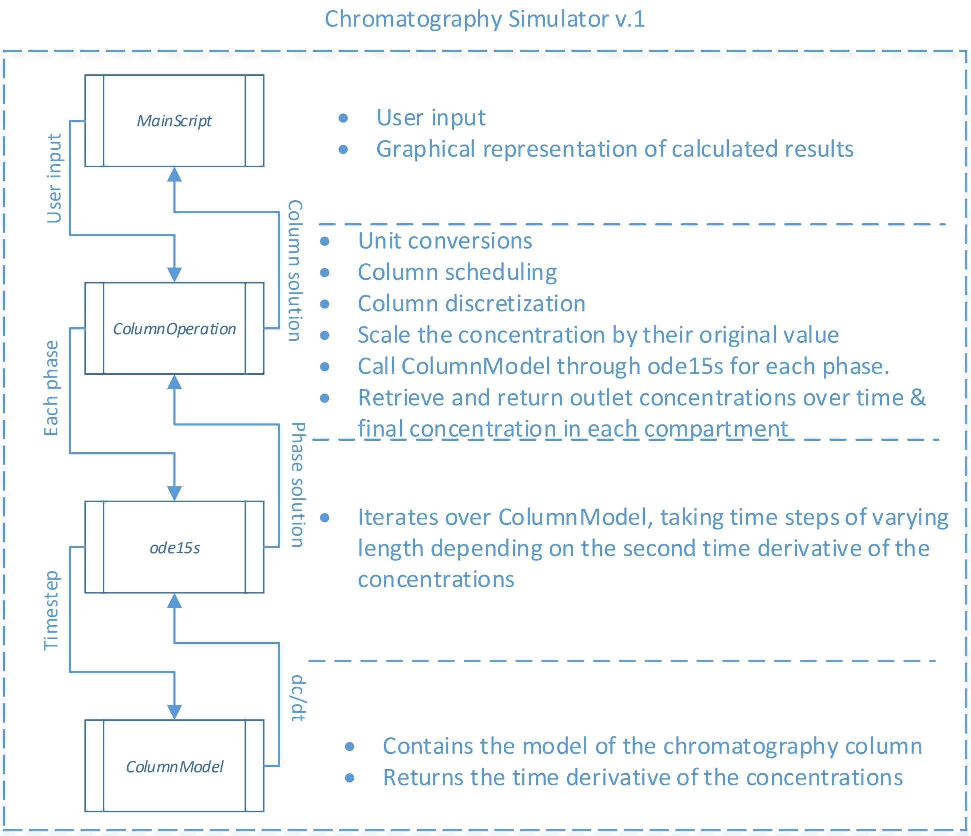
[12] N. Andersson, A. Löfgren, M. Olofsson, A. Sellberg, B. Nilsson and P. Tiainen, "Design and Control of Integrated Chromatography Column Sequences," *Biotechnology Progress,* vol. 33, no. 4, pp. 923-930, 2017.

[13] N. Andersson, A. Löfgren, M. Olofsson, A. Sellberg and B. Nilsson, "The Orbit Controller," Department of Chemical Engineering, Lund University, Lund, 2016.

[14] S. Braslavsky and K. Houk, "Chapter 30 Glossary of Terms used in Photochemistry," in *Photochromism - Molecules and Systems (Revised Edition)*, Amsterdam, Elsevier B.V., 2003, pp. 977-1032.

[15] The MathWorks, Inc., "Trapezoidal numerical integration - MATLAB trapz - MathWorks Nordic," The MathWorks, Inc., [Online]. Available: The MathWorks, Inc.. [Accessed 12 10 2017].

APPENDIX 1. CODE STRUCTURES



- User input
- Graphical representation of calculated results

- Unit conversions
- Column scheduling
- Column discretization
- Scale the concentration by their original value
- Call ColumnModel through ode15s for each phase.
- Retrieve and return outlet concentrations over time & final concentration in each compartment

- Iterates over ColumnModel, taking time steps of varying length depending on the second time derivative of the concentrations

- Contains the model of the chromatography column
- Returns the time derivative of the concentrations

Figure 17: Code structure of the initial code

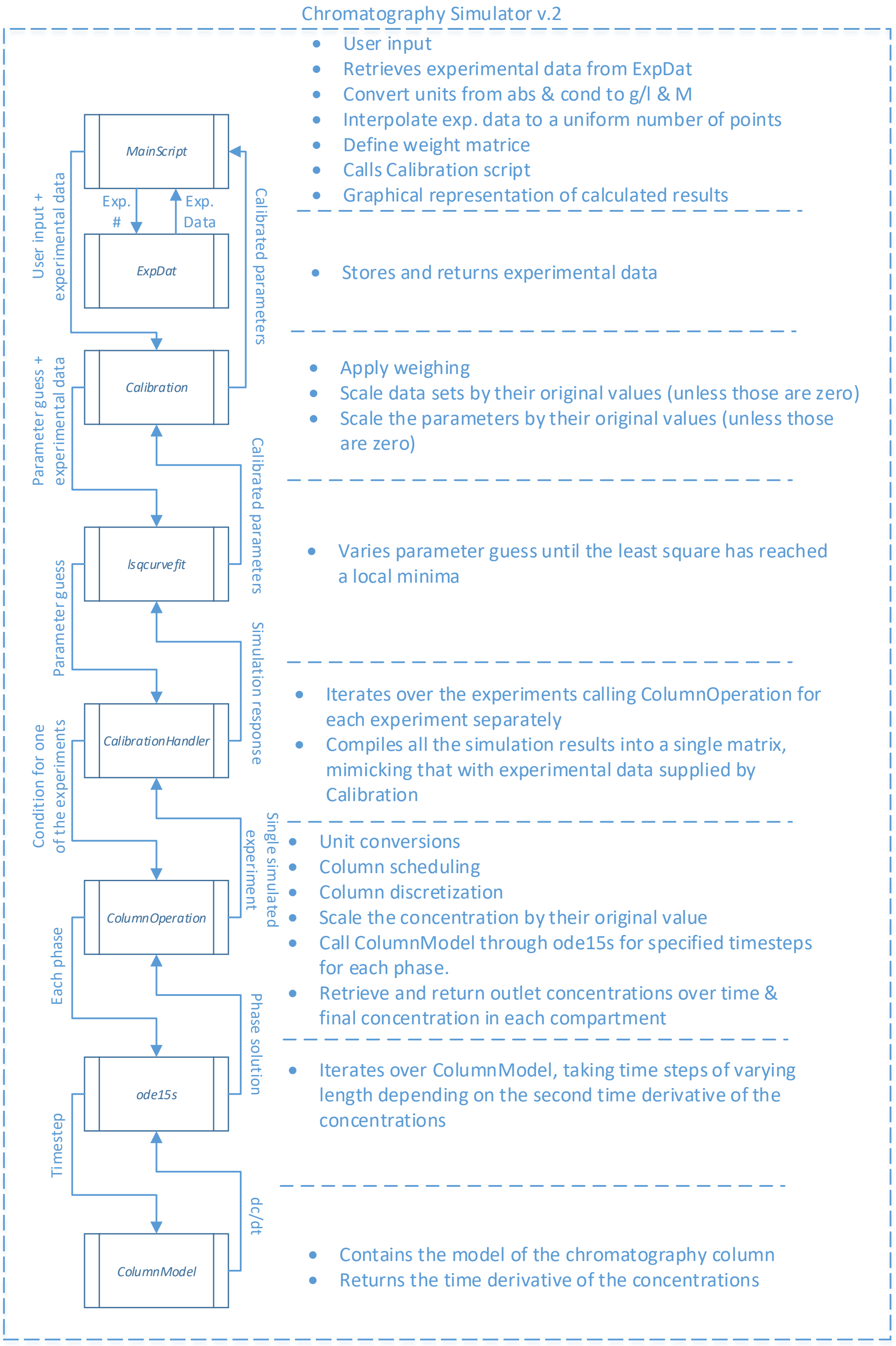
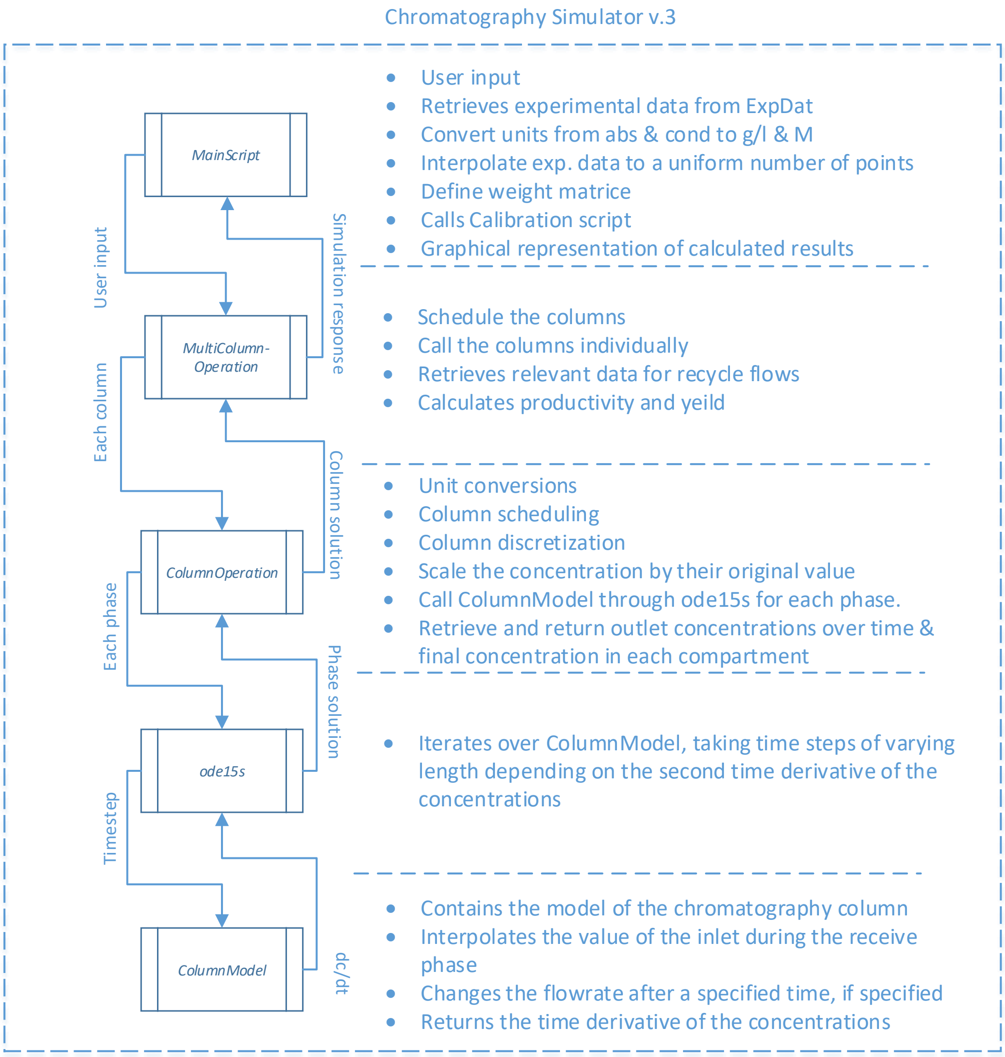

Chromatography Simulator v.2
MainScript
Exp. #
Exp. Data
ExpDat
User input + experimental data
Calibrated parameters
Calibration
Parameter guess + experimental data
lsqcurvefit
Calibrated parameters
Parameter guess
Simulation response
CalibrationHandler
Condition for one of the experiments
Single simulated experiment
ColumnOperation
Each phase
Phase solution
ode15s
Timestep
dc/dt
ColumnModel
User input
Retrieves experimental data from ExpDat
Convert units from abs & cond to g/l & M
Interpolate exp. data to a uniform number of points
Define weight matrice
Calls Calibration script
Graphical representation of calculated results
Stores and returns experimental data
Apply weighing
Scale data sets by their original values (unless those are zero)
Scale the parameters by their original values (unless those are zero)
Varies parameter guess until the least square has reached a local minima
Iterates over the experiments calling ColumnOperation for each experiment separately
Compiles all the simulation results into a single matrix, mimicking that with experimental data supplied by Calibration
Unit conversions
Column scheduling
Column discretization
Scale the concentration by their original value
Call ColumnModel through ode15s for specified timesteps for each phase.
Retrieve and return outlet concentrations over time & final concentration in each compartment
Iterates over ColumnModel, taking time steps of varying length depending on the second time derivative of the concentrations
Contains the model of the chromatography column
Returns the time derivative of the concentrations

Figure 18: Code structure of the calibration code



- User input
- Retrieves experimental data from ExpDat
- Convert units from abs & cond to g/l & M
- Interpolate exp. data to a uniform number of points
- Define weight matrice
- Calls Calibration script
- Graphical representation of calculated results

- Schedule the columns
- Call the columns individually
- Retrieves relevant data for recycle flows
- Calculates productivity and yeild

- Unit conversions
- Column scheduling
- Column discretization
- Scale the concentration by their original value
- Call ColumnModel through ode15s for each phase.
- Retrieve and return outlet concentrations over time & final concentration in each compartment

- Iterates over ColumnModel, taking time steps of varying length depending on the second time derivative of the concentrations

- Contains the model of the chromatography column
- Interpolates the value of the inlet during the receive phase
- Changes the flowrate after a specified time, if specified
- Returns the time derivative of the concentrations

Figure 19: Code structure of multicolumn code.