LUND UNIVERSITY

Master Thesis

Iterative Learning Control for Preparative Chromatography

by

Daniel Espinoza

Department of Chemical Engineering
Lund University
Sweden
April 1, 2020

Supervisor: **Ph.D. Niklas Andersson**
Examiner: **Professor Bernt Nilsson**

Front page picture: Overall structure of an objective function-based ILC.

Postal address
PO-Box 124
SE-221 00 Lund, Sweden
Web address
www.lth.se/chemeng

Visiting address
Naturvetarvägen 14

Telephone
+46 46-222 82 85
+46 46-222 00 00

Printed in Sweden by Media-Tryck.
Lund 2020

Preface

What you are about to read is the report of my master thesis, the final project of the chemical engineering program at Lund University's Faculty of Engineering.

In this thesis work I have worked towards automatizing the downstream processing of proteins for pharmaceutical use by applying an iterative learning controller to the chromatographic purification step. Two controller configurations were tested, the first with the resolution between two peaks as the parameter to be controlled, and the other with an objective function constructed to yield an optimal separation. The controller was applied to a simulated chromatography process. The construction of the simulated process and the design of the controllers are detailed in the upcoming pages.

This thesis project has allowed me to apply much of the knowledge I have acquired during my specialization in process design within the program. Most notably, process simulation and process control were the core of this project. The simulations were run in Python, and thus I have also gotten to apply programming which I learned during my exchange semester at Delft University of Technology. I also learned about bioprocess design in Delft, which is closely related to the production of pharmaceutical proteins.

I have enjoyed the time I have spent producing this report, and it is my sincere wish that you enjoy reading it as well.

Best regards,
Daniel Espinoza
March 25th, 2020

Acknowledgements

This thesis marks the end of my time as a student of chemical engineering at the Faculty of Engineering at Lund University, and the transition into the unknown waters that lay beyond. It is by far the greatest undertaking of my life thus far, and I have many people to thank for having been able to reach this point. Listing them all has proven to be nigh impossible as once I think I have gotten them all, more people tend to come to mind. Still, I will attempt to do as many people justice as possible.

First off, I would like to thank my examiner, professor Bernt Nilsson, and my supervisor Ph.D. Niklas Andersson for their support and the opportunity to do my thesis project at the Department of Chemical Engineering at Lund University. You have given me countless hours of support during the project, despite both of your very busy schedules. Without you, this thesis would never have happened!

I would like to continue by thanking all Ph.D. students that are part of professor Bernt Nilsson's group, all of whom have helped shape my time at the deparment into the enjoyable and educative experience that it has been. Joaquín Gomis Fons has been fantastic in helping me familiarize myself with the laboratory. Anton Löfgren has been a huge help in learning how to analyze the data I have used in this thesis, and I have appreciated our discussions about training as well. Simon Tallvod has also helped me out quite a bit in the lab, and our shared interest in Asian cuisine has entertained many spirited discussions in the coffee room. Madèlene Isaksson has helped me understand some basic chromatography concepts better and has also shared many interesting ideas that could be applied to the concepts in this thesis in the future. And finally, Mikael Yamanee-Nolin has proofread this report, provided me a lot of useful feedback out of his own interest, and been an enthusiastic partner in conversations about pretty much anything. I would also like to thank my fellow master thesis student Hanna Danielsen, with whom I have shared an office, many hours of discussing our respective projects, and even more hours of less productive conversation. All of you have been great colleagues during my time creating this thesis.

I would also like to thank the rest of the Department of Chemical Engineering for creating such a pleasant atmosphere and for being so welcoming. It has been very liberating to be able to sit and talk with anyone and everyone during lunch and coffee breaks.

My partner, Ella Lagerwall, has been my pillar of support during these tumultuous past few months. Being together with her has made me grow into a more well-rounded and complete human being, and so I would like to thank you for being by my side throughout this project. Facing the future seems like a trivial challenge when we are together.

A special acknowledgement goes out to my family, in particular my mother Roxana and my brother Carlos. Together we came to this country 21 years ago, and together we have made it our home. I would literally not be where I am today without you!

Finally, I would like to thank the wealth of friends that I have made during my years as a student and as a member of the Guild of Chemical Engineering and Biotechnology. I could write an entire separate thesis on how meeting all of you has affected me in all the best ways, but I will leave that for a future project. For now, just know that I appreciate all of you and that I can only hope that I have been as big a part of your lives as you have been in mine.

Thank You!

Abstract

There is a desire for a fully automated downstream process in pharmaceutical protein production. One part of the downstreaming process is preparative chromatography. A good separation between the product and other proteins, as well as good productivity, are desired. Due to the batch nature of chromatographic separation, an iterative learning controller (ILC) could be a suitable choice for achieving these goals. ILC based on time-varying perturbation models have been successfully applied to control batch reactors in the past. The purpose of this master thesis was to test the application of time-varying perturbation model-based ILC for automation of preparative chromatography.

The application of an ILC was performed using simulations of an ion-exchange chromatographic purification process. Since protein purification by chromatography is commonly performed with gradient elution, the slope of the gradient was chosen as the input parameter for the controller. The slope was controlled via the gradient time, i.e. the time it takes for the elution buffer to go from its initial to its final concentration. First, the resolution between two peaks was used as the output parameter of the controller. The controller was able to successfully reach the desired resolution, however using only the gradient time resulted in a non-linear process trajectory, which resulted in difficulties in process control. Secondly, an objective function was constructed using the resolution and the productivity of the process. The objective function had a local extrema, which was considered the process optimum and thus the derivative of the objective function was used as output parameter. This configuration showed promise, although the estimation of the derivative during live runs was a limiting factor.

ILC shows promise for use in preparative chromatography. Multiple-input-multiple-output configurations should be considered for future applications, as such configurations could circumvent the problems caused by the non-linear process trajectory. Alternatively, a different objective function could possibly be applied. A natural next step is to apply the ILC to a real process, thus coming closer to full automation of preparative chromatography.

Sammanfattning

Det fins en önskan om fullständigt automatiserade nedströmsprocesser i framställandet av proteiner för läkemedelsbruk. En del av nedströmsprocessen är preparativ kromatografi. En god separation mellan produkt och andra proteiner, samt god produktivitet, är av intresse. På grund att kromatografisk separation ofta körs satsvis så skulle en iterative learning controller (ILC) vara ett bra val för att åstadkomma dessa mål. ILC baserad på tidsvarierande perturbationsmodeller har tidigare tillämpats för att reglera satsvisa reaktorer med goda resultat. Syftet med detta projekt var att testa tillämpningen av en sådan ILC för att automatisera preparativ kromatografi.

Tillämpningen av en ILC utfördes genom simuleringar av en kromatografisk reningsprocess i en jonbytarkolonn. Eftersom proteinupprening med kromatografi ofta utförs med gradienteluering så valdes lutningen av gradienten som in-parameter för styrning. Lutningen kontrollerades via gradienttiden, dvs tiden det tar för elueringsbufferten att gå från sin start- till slutkoncentration. Först användes upplösningen mellan två toppar i kromatogrammet som ut-parameter. Regulatorn lyckades nå den önskade upplösningen, men att enbart använda gradientlutningen som in-parameter visade sig leda till en icke-linjär processbana, vilket ledde till svårigheter i regleringen. Därefter konstruerades en målfunktion med hjälp av upplösningen och processens produktivitet. Målfunktionen hade en lokal extrempunkt som ansågs vara processoptimat och därav valdes målfunktionens derivata som ut-parameter. Denna konfiguration var lovande, men begränsades av derivataberäkningen under körningens gång.

ILC uppvisar potential för användning i preparativ kromatografi. Konfigurationer med flera in- och ut-parametrar bör övervägas för framtida tillämpning, då sådana konfigurationer skulle kunna kringgå problemen som orsakades av den icke-linjära processbanan. Alternativt så kan en annan objektivfunktion konstrueras. Ett naturligt nästa steg är att tillämpa ILC på en verklig process, och således komma närmare fullständig automatisering av preparativ kromatografi.

Popular science summary

Proteins are molecules that can possess many different qualities, some of which are beneficial to humans. An example of such a protein is insulin, which is used to treat diabetes. Proteins for use in pharmaceuticals are produced in living organisms along with other, less beneficial or even harmful proteins, and thus need to be properly purified before use. One part of the purification process is called preparative chromatography. Ion-exchange chromatography is commonly used for protein purification, and involves the use of a column to which the different proteins stick, while other molecules flow through. Salt is then flushed through the column to remove the stuck proteins. The strength with which the proteins stick to the column varies from protein to protein, so higher concentrations of salt are needed for proteins that are more strongly stuck. Thus, it is possible to separate one protein from the other by gradually increasing the salt concentration while the column is being flushed. Choosing how this salt concentration should increase requires many experiments and work hours.

In this master thesis, a method for making the separation of proteins as good as possible automatically is tested. The method used is called iterative learning control (ILC), which means that information from previous chromatography runs is used to predict how the salt concentration needs to change to give the desired separation. Several simulations of a chromatography process were run to develop the controller. In the end, the ILC was able to control the processes and thus shows promise for the automation of preparative chromatography processes, although more work needs to be performed before it can function optimally. In the future, an ILC for preparative chromatography could be combined with the other purification steps to create a fully automated protein purification process.

Populärvetenskaplig sammanfattning

Proteiner är molekyler som kan ha många olika egenskaper, varav vissa är förmånliga för människor. Ett exempel på ett sådant protein är insulin, som används för att behandla diabetes. Proteiner för läkemedelsbruk produceras ofta i levande organismer tillsammans med andra, mindre gynnsamma eller till och med skadliga proteiner, och behöver således renas ordentligt före användning. En del av reningsprocessen kallas för preparativ kromatografi. Jonbytarkromatografi är vanligt för upprening av proteiner, och involverar användandet av en kolonn som de olika proteinerna fastnar på, medan andra molekyler flödar igenom. Salt spolas sedan genom kolonnen för att ta bort proteinerna som fastnat. Proteiner binder olika hårt till kolonnen, vilket innebär att högre koncentrationer av salt krävs för proteinerna som sitter fast hårdare. Därför går det att separera ett protein från ett annat genom att gradvis öka koncentrationen salt medan kolonnen spolas. Att välja hur denna ökning av koncentration ska gå till kräver många experiment och arbetstimmar.

I detta examensarbete har en metod för att göra separation av proteiner så bra som möjligt automatiskt testats. Metoden som använts kallas för reglering med iterativ inlärning (iterative learning controller, eller ILC), som innebär att information från tidigare kromatografikörningar används för att förutsäga hur saltkoncentrationen behöver ändras för att ge den önskade separationen. Flera simuleringar av en kromatografiprocess kördes för att utveckla regulatorn. I slutändan så var ILC:n kapabel till att reglera processerna och således visar metoden potential för automatisering av preparativ kromatografi, men mer arbete behöver utföras innan det fungerar optimalt. I framtiden kan ILC för preparativ kromatografi kombineras med andra reningssteg för att skapa en fullständigt automatisk proteinuppreningsprocess.

Contents

1 Introduction

Automatically finding and applying settings for optimal separation and productivity in downstream processing of biopharmaceutical proteins is a prospect of great interest. A method to achieve this could reduce the time spent experimentally deriving these settings for every new protein mixture, for example in separation by chromatography. In the long term, this could become another step in the creation of fully automated bioprocess downstreaming.

The research group at the Department of Chemical Engineering at Lund University led by Bernt Nilsson is devoted to development and optimization of downstream processing systems for production of biopharmaceuticals, with full automation as the end goal. One part of the automation process is the derivation and application of settings for optimal chromatographic purificaiton. Two parameters are of particular interest. Firstly, good separation between product and other compounds in the process stream is desired. Secondly, a short process is desired to maximize productivity. In addition, automatic adjustments to disturbances that may occur in the process are also desired. One proposed method to achieve these three conditions is the use of an iterative learning controller, which is a method of controlling parameters of interest to keep a desired value in processes that repeat over time, such as batch processes.

The purpose of this thesis is to evaluate the use of iterative learning in controlling preparative chromatography processes, as well as its use in finding optimal separation parameters.

2 Background

2.1 Chromatography

Chromatography is a separation technique that is based on a fluid mobile phase flowing through a stationary phase. The former is a mixture of molecules of interest, whereas the latter is a material to which the molecules in the mobile phase (Schulte & Epping, 2005) adsorb. The components in the mobile phase are separated based on their interactions with the stationary phase (Jandera & Henze, 2011).

Ion-exchange chromatography is commonly used for separation of proteins, the mechanisms for which are illustrated by Figure 2.1. The stationary phase in an ion-exchange chromatography column consists of a packing material with high surface area, on which positively or negatively charged groups are immobilized. Ions in the mobile phase with the opposite charge are in equilibrium with these charged sites. Once the column is loaded with proteins, the proteins switch places with these counter-ions and bind to the sites via charged groups on the protein surface. The

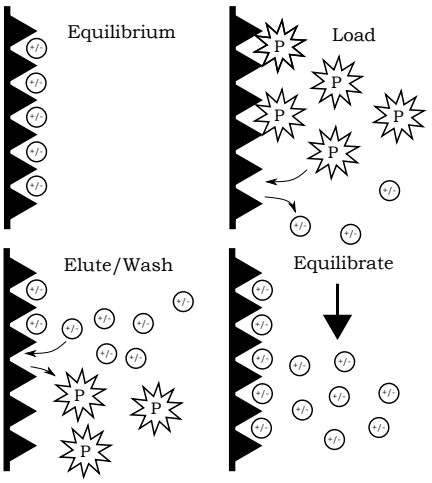


Figure 2.1: Illustration of ion-exchange chromatography, where at first counter-ions are adsorbed to the charged sites in the column packing. Proteins are then loaded, and eluted with salt solution before the column is equilibrated and may be loaded with protein once more (Roos, 2000).

adsorbed proteins are then eluted by increasing the concentration of ions in the mobile phase, for example by injecting a solution of salt such as NaCl (Roos, 2000). To facilitate the separation in cases where there is a difference in adsorption strength between components in the mobile phase, linear gradient elution may be applied. This is done by gradually increasing the elution strength during the elution process. This may improve the separation of the proteins compared to having a constant gradient strength during the elution (Schulte, Wekenborg, & Wewers, 2005).

The quality of a chromatographic separation can be quantified in several ways. One way is by looking at the *retention time* of the process. A short retention time is beneficial in batch chromatography, since it allows for a larger number of batches to be run in a fixed amount of time than a longer retention time does. However, a short retention time is only useful if there is good *resolution* between the peaks in the chromatogram. Defined according to Equation 2.1 and demonstrated in Figure 2.2, the resolution of two peaks is a function of the base width of the peaks (w_b), as well as of the retention time (t_r) of the top of each peak, and is a measure of how well the peaks are separated (Schulte & Epping, 2005).

$$res = 2\frac{t_{r2} - t_{r1}}{w_{b1} + w_{b2}} \tag{2.1}$$

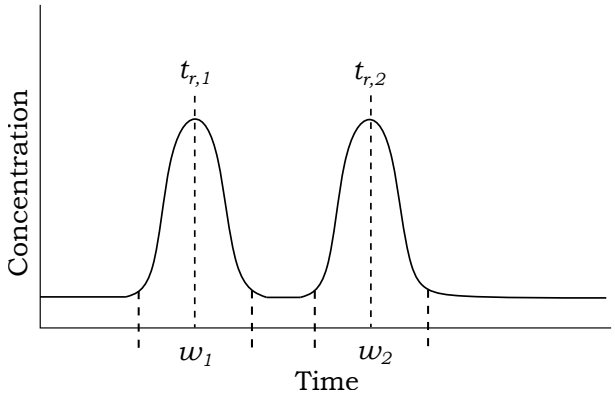


Figure 2.2: Illustration of the calculation of the resolution between two peaks in a chromatogram. The retention time t_r is defined as the time at which the top of a peak is eluted. The width w of a peak is the width of the baseline of a peak (Schulte & Epping, 2005).

2.2 Iterative Learning Control

Bajpai (2018) describes process control as a means to maintain a desired process parameter at a desired value. For example, maintaining a certain indoor temperature by stoking the flames in a fireplace can be considered a type of manual process control. If, instead, one uses electrical heating and a thermostat to maintain an indoor temperature, it could be considered a form of automatic process control. For control to be possible in this latter example, the process needs a thermometer (a sensor), a thermostat (the controller) and a heater (an actuator). The thermostat calculates the difference between the current and the desired temperature (also known as the error). It then sends instructions to the heater to either turn on or off depending on what direction the temperature needs to be changed in to achieve the desired value.

Iterative learning control (ILC) is a method used to improve performance in processes that repeat over time. The main idea is to use existing operation data to obtain a control input that gives a desirable process output. This style of control is commonly applied to robotic production processes or chemical batch reactors (Ahn, Chen, & Moore, 2007). Figure 2.3 illustrates the general structure of an ILC. A batch system is subject to an input u_k at batch number k, which results in a system output y_k. Both the input and output are added to a memory, where inputs and their corresponding outputs from previous runs are stored. This memory is then used by the iterative learning controller, together with the desired output value y_d, to compute the next batch input u_{k+1}. This new input is fed to the system, and the process repeats (Ahn et al., 2007).

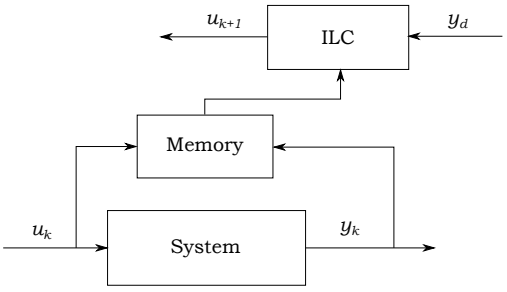


Figure 2.3: General ILC structure (Ahn, Chen, & Moore, 2007).

2.3 Batchwise Perturbation Model-Based ILC

Xiong and Zhang (2003) describe the application of an optimal iterative learning control algorithm based on time-varying perturbation models. The algorithm acts as the ILC-block in the schematic shown in Figure 2.3. Xiong and Zhang (2003) successfully applied this ILC algorithm to a simulated chemical batch reactor, using the algorithm to maximize the concentration of a reaction product by controlling the reactor temperature. The study was based on a batch process with fixed run-length and a certain number of sampling intervals, N. The goal of the algorithm was to steer a process variable towards following a given *trajectory*, meaning that at each sampling time during the process, the process variable had a set point that the ILC would steer towards. As an example, in a process with a run-length of 20 minutes and with $N = 10$ sampling intervals, the ILC was given 10 set points and would control the system inputs so that the set point was reached at each sampling interval.

The sampling is the equivalent to gathering data on the input and output parameters at each sampling point and add them to the ILC history. If the process had I input variables and O output variables, then at each sampling point I values would be added to the input history and O values would be added to the output history. Together, these points of input and output values, u and y, form the trajectory of the process. At batch k, if $I = O = 1$, these trajectories can be represented as the vectors defined by Equations 2.2 and 2.3.

$$\boldsymbol{U}_k = [u_k^T(1), u_k^T(2), ..., u_k^T(N-1), u_k^T(N)]^T \tag{2.2}$$

$$\boldsymbol{Y}_k = [y_k^T(1), y_k^T(2), ..., y_k^T(N-1), y_k^T(N)]^T \tag{2.3}$$

The desired process output trajectory would then be defined as Equation 2.4.

$$\boldsymbol{Y}_d = [y_d^T(1), y_d^T(2), ..., y_d^T(N-1), y_d^T(N)]^T \tag{2.4}$$

Xiong and Zhang (2003) then represent the causal relation between $\boldsymbol{Y}_k$ and $\boldsymbol{U}_k$ with a system of static, non-linear functions $\boldsymbol{F}(\cdot)$ as Equation 2.5,

$$\boldsymbol{Y}_k = \boldsymbol{F}(\boldsymbol{U}_k) + \boldsymbol{v}_k \tag{2.5}$$

where $\boldsymbol{v}_k$ is a vector of measurement noise. By then linearizing the system around a nominal trajectory $\boldsymbol{Y}_s$ and $\boldsymbol{U}_s$, i.e. a supposed trajectory that the system would follow in theory, the model described by Equation 2.6 is obtained,

$$\overline{\boldsymbol{Y}}_k = \boldsymbol{G}_s\overline{\boldsymbol{U}}_k + \boldsymbol{d}_k \tag{2.6}$$

where $\overline{\boldsymbol{Y}_k}$ and $\overline{\boldsymbol{U}_k}$ are the perturbation variables of the process and are defined according to Equation 2.7.

$$\overline{\boldsymbol{Y}}_k = \boldsymbol{Y}_k - \boldsymbol{Y}_s \qquad\qquad \overline{\boldsymbol{U}}_k = \boldsymbol{U}_k - \boldsymbol{U}_s \tag{2.7}$$

$\boldsymbol{d}_k$ in Equation 2.6 represents a combination of measurement noise and the model errors that follow the linearization. $\boldsymbol{G}_s$ then represents the linear time-varying model operator, which transforms the input variables into the process output. Xiong and Zhang (2003) eventually reach the ILC law in Equation 2.8.

$$\boldsymbol{U}_{k+1} = \hat{\boldsymbol{K}}_k\boldsymbol{e}_k + \boldsymbol{U}_k \tag{2.8}$$

Here, $\boldsymbol{e}_k$ denotes the difference between the desired process output, $\boldsymbol{Y}_d$, and the actual output $\boldsymbol{Y}_k$, at batch k. This is the tracking error of the process. $\hat{\boldsymbol{K}}_k$ is the learning rate of the controller and is derived as described in Equation 2.9.

$$\hat{\boldsymbol{K}}_k = [\hat{\boldsymbol{G}}_s^T \boldsymbol{Q} \hat{\boldsymbol{G}}_s + \boldsymbol{R}]^{-1} \hat{\boldsymbol{G}}_s^T \boldsymbol{Q} \tag{2.9}$$

$\boldsymbol{Q}$ and $\boldsymbol{R}$ are positive-definite matrices that act as weighting of the error and the input change, respectively. They stem from the derivation of the ILC law, in which the quadratic objective function in Equation 2.10 is applied.

$$J_{k+1} = \min_{\Delta \bar{\boldsymbol{U}}_{k+1}} \frac{1}{2} [\tilde{\boldsymbol{e}}_{k+1}^T \boldsymbol{Q} \tilde{\boldsymbol{e}}_{k+1} + \Delta \bar{\boldsymbol{U}}_{k+1}^T \boldsymbol{R} \Delta \bar{\boldsymbol{U}}_{k+1}] \tag{2.10}$$

$\tilde{\boldsymbol{e}}$ denotes the modified prediction error of the perturbation model. Thus, $\boldsymbol{Q}$ penalizes large errors $\tilde{\boldsymbol{e}}$ by increasing the control action, whereas $\boldsymbol{R}$ has a dampening effect by penalizing large changes in the control action $\Delta \boldsymbol{U}_{k+1}$. $\hat{\boldsymbol{G}}_s$ describes the prediction of the linear time-varying model. In the model constructed by Xiong and Zhang (2003), this prediction was updated at every batch according to Equation 2.11.

$$\hat{\boldsymbol{G}}_s = \hat{\boldsymbol{G}}_k = [\hat{g}_{k,1}^T, \hat{g}_{k,2}^T, \ldots, \hat{g}_{k,N}^T] \tag{2.11}$$

$\hat{g}_k$ at time i is defined as Equation 2.12.

$$\hat{g}_{k,i} = (\boldsymbol{H}_k^{iT} \boldsymbol{H}_k^i)^{-1} \boldsymbol{H}_k^{iT} \boldsymbol{Z}_k^i \tag{2.12}$$

$\boldsymbol{H}_k$ and $\boldsymbol{Z}_k$ are the previous batch history matrices described by Equation 2.13.

$$\boldsymbol{Z}_k^i = \begin{bmatrix} \beta^{L+k-1}\overline{y}_1^0(i) \\ \vdots \\ \beta^k\overline{y}_L^0(i) \\ \beta^{k-1}\overline{y}_1(i) \\ \vdots \\ \beta^1\overline{y}_{k-1}(i) \\ 0 \end{bmatrix} \qquad \boldsymbol{H}_k^i = \begin{bmatrix} \beta^{L+k-1}\overline{\boldsymbol{h}}_1^0(i) \\ \vdots \\ \beta^k\overline{\boldsymbol{h}}_L^0(i) \\ \beta^{k-1}\overline{\boldsymbol{h}}_1(i) \\ \vdots \\ \beta^1\overline{\boldsymbol{h}}_{k-1}(i) \\ 0 \end{bmatrix} \tag{2.13}$$

The superscript 0 denotes the L number of batches in the history, i.e. before the current k batches were run. β is a forgetting factor with the purpose of diminishing the weight of older batches in the estimation of $\hat{\boldsymbol{K}}_k$. $\boldsymbol{h}$ and $\overline{y}$ are historical input and output values, respectively, that have been modified by subtracting the values of the input and output at batch k.

3 Materials and Methods

The application of a learning controller was done on a simulated ion-exchange chromatography column separating two proteins in an ion-exchange chromatography column. Once a working model was established, two different ILC configurations were applied: one simple controller where one parameter was chosen to be controlled with a desired set point, and one controller where an objective function was constructed and its optimum was chosen as control parameter. The methods and tools used to perform this application are detailed in the following sections.

3.1 Model and Simulation

Three mass transport phenomena were taken into consideration when modelling the behavior of the column: axial dispersion, convection, and adsorption to the column packing. The change in concentration c of component i at time t in the mobile phase is described by the partial differential Equation 3.1 (Arkell, 2017):

$$\frac{\partial c_i}{\partial t} = -\frac{F}{A\varepsilon}\frac{\partial c_i}{\partial z} + D_{app}\frac{\partial^2 c_i}{\partial z^2} - \frac{(1-\varepsilon_c)}{\varepsilon}\frac{\partial q_i}{\partial t} \tag{3.1}$$

The parameter D_{app} describes the apparent axial dispersion in the column (Arkell, 2017). Column dimensions such as bed height, inner diameter and bed volume are displayed in table 3.1 ("HiTrap Capto S cation exchange chromatography column", 2020), as are the experimentally derived total porosity ε, void fraction ε_c and apparent dispersion coefficient D_{app} (Al-Kaisy, 2015).

Al-Kaisy (2015) performed model calibration on a HiTrap® Capto™ S ion-exchange column from GE Healthcare using a ternary mixture of proteins (lysozyme, cytochrome C and ribonuclease A). Four different models for adsorption were tested. Out of those four models, the Langmuir model with mobile phase modulators (MPM), where proteins compete for available adsorption space in the column,

Table 3.1: Void fractions, apparent dispersion coefficient (Al-Kaisy, 2015) and dimensional parameters of a HiTrap® Capto™ S ion-exchange column ("HiTrap Capto S cation exchange chromatography column", 2020).

ε_c	0.3123	-
ε	0.8468	-
D_{app}	$4.6026 \cdot 10^{-8}$	$m^2\ s^{-1}$
Bed height	25	mm
Inner diameter	7	mm
Bed volume	1	ml

worked particularly well for low protein loads. The Langmuir MPM model that describes the concentration q of adsorbed component i at time t is shown in Equation 3.2 (Jakobsson, 2006):

$$\frac{\partial q_i}{\partial t} = k_{kin,i}\left(B_i c_i c_s^{-\beta_i}\left(1 - \sum_{j=1}^{N}\frac{q_j}{q_{max,j}}\right) - q_i\right) \tag{3.2}$$

The adsorbed concentration q_i is dependent on the concentration of the same component, c_i, as well as the concentration of salt, c_s, in the mobile phase. The parameter β_i describes the ion-exchange characteristics of the compound (Jakobsson, 2006). The $q_{max,j}$ parameter describes the maximum concentration of adsorbed compound j. Finally, B_i and $k_{kin,i}$ describe various properties such as adsorption-desorption kinetics, lumped together to simplify model calibration (Al-Kaisy, 2015).

To reduce the complexity of the problem, the simulated experiments were performed on a binary separation. Lysozyme and cytochrome C were chosen as proteins, using NaH_2PO_4 as buffer and NaCl as salt for elution. The simulation was based on separation in a HiTrap® Capto™ S chromatography column. Equation 3.1 was set up for three compounds: each of the proteins, as well as salt. NaCl and NaH_2PO_4 were regarded as the same salt and were lumped into the same equation in this application. The parameters used for the adsorption kinetics in the Langmuir MPM model are presented in table 3.2 (Al-Kaisy, 2015).

Table 3.2: Adsorption kinetic parameters for lysozyme and cytochrome C in the Langmuir MPM model (Al-Kaisy, 2015).

	Lysozyme	Cytochrome C	
β	6.1698	5.8758	-
k_{kin}	$2.3286 \cdot 10^{-2}$	$5.36785 \cdot 10^{-2}$	s^{-1}
B	$9.50824 \cdot 10^{16}$	$6.48485 \cdot 10^{15}$	-
q_{max}	8.3875	11.1946	mol m^{-3}

The partial derivatives in Equation 3.1 were discretized along the axial (z) dimension to reduce the problem into a system of ordinary differential equations (ODEs) by means of the method of lines (Davis, 1984). The column was discretized to 100 points. The finite volume method was used to approximate the derivatives in the differential equations, using two-point backward and three-point central differences for first- and second-order derivatives, respectively (LeVeque, 2002).

The simulations were performed using Python's SciPy library and its numerical methods for solving of initial value problems. The system of differential equations was solved using a backward differentiation formula (BDF) implemented via SciPy's *solve_ivp* function.

3.2 Chromatography Process

A simulation of a basic batch chromatography process was set up to study the separation of the two proteins. The controller was to be applied to this system. The process consisted of five phases: load, wash, elution, regeneration and equilibration.

During the *loading phase*, the chromatography column is loaded with 1 column volumes (CV) of the protein sample, a mix of 1 g/l each of lysozyme and cytochrome C in a solution of 20 mM NaH_2PO_4 (henceforth referred to as Buffer A). The loading phase is followed by a *washing phase*, during which the system is flushed with pure Buffer A for 5 CV with the purpose of washing out any non-adsorbed protein in the column. During the *elution phase*, a linear gradient is applied by pumping a mixture of Buffer A and a 750 mM solution of NaCl (henceforth referred to as Buffer

B) through the column, successively increasing the amount of Buffer B in relation to Buffer A. Three parameters are of importance in this phase: the initial fraction of Buffer B ($x_{B,i}$), the final fraction of Buffer B ($x_{B,f}$), and the elution time (or gradient time, t_{grad}). For example, if the initial fraction of Buffer B is set to 0, the final fraction to 1 and the elution time is set to 20 minutes, then the elution will begin by pumping pure Buffer A through the system. The flow of Buffer B will increase linearly with a slope such that the fraction of Buffer B reaches 1 after 20 minutes. Together, these three parameters determine the slope of the linear gradient which in turn controls the retention time and resolution of the peaks.

The elution phase is followed by a *regeneration phase*, during which the column is flushed with pure Buffer B for 5 CV. This elutes any protein that remains adsorbed on the column.

Finally, the column is flushed with pure Buffer A during the *equilibration phase*. This returns the column to a state where a new protein sample injection can be performed and allows the process to be repeated.

3.3 Controller Design

Two controller configurations were designed for ILC implementation. Both configurations were applied to the simulated chromatography process in a Python program. The main difference between these was the choice of *output parameter*, denoted as y, which is the parameter to be controlled. This parameter was different for the two configurations. However the general structure of the controllers was the same: the controllers took a *set point* value, y_d, and controlled the process by adjusting an *input parameter*, u, to steer y towards y_d by applying the control algorithm to minimize the *error*, e. The input parameter chosen for both configurations was the *gradient time*, t_{grad}. Indirectly, the gradient time controls the slope of the gradient by setting the time required for the buffer composition to go from $x_{B,i}$ to $x_{B,f}$, if these are set to fixed values.

The ILC algorithm needed some existing operating points to run, a *memory* of past inputs and their respective outputs. For both configurations, the same four initial

points were given to the algorithm. Additionally, both y and u were *scaled* by some factor with the intention of keeping them in the same order of magnitude. The selection of the initial points in the memory and the scaling factors is detailed in section 3.5. Once applied, the ILC algorithm would use the initial memory points to suggest a new input value to apply to the next batch.

In ***Configuration 1***, the chromatography peak resolution, res, was chosen as output parameter. To find an achievable resolution to use as set point, a series of simulations of the chromatography process were run. Each simulation was run with a different input value, ranging from the lowest possible to as high a value as was acceptable. The lower bound of this interval was set to a t_{grad} as short as possible while still resulting in two separate peaks. The upper bound, on the other hand, was set to 500 minutes. This gradient time was chosen as a safeguard against unreasonably long gradient times. While a gradient time of this length would be still impractical due to the long batch cycle times it would result in, it was deemed acceptable in the evaluation of a controller. Once the simulations were run, the resulting data was plotted. From this data, the achievable span of resolution in the gradient time interval was identified and a suitable set point was chosen. The structure of Configuration 1 is demonstrated in Figure 3.1.

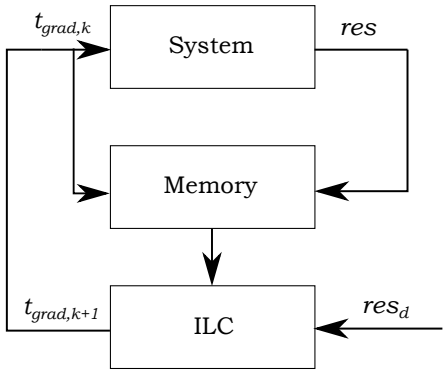


Figure 3.1: The overall structure of Configuration 1.

Configuration 2 was constructed using an *objective function*. The resolution and the *productivity* of the process were used to construct a function with a local extreme value that could be used as set point. The assumption was made that the productivity is more or less equal to the inverse of the retention time of the second peak in the chromatogram, according to Equation 3.3. This way, the productivity

decreased as the retention time increased, which was deemed reasonable as a longer retention time results in a longer process cycle time. An offset time t_0 was added to the denominator to avoid division by zero. This offset time was set to 5 minutes.

$$Prod \approx \frac{1}{t_{r2} + t_0} \tag{3.3}$$

Assuming that the resolution of two peaks increased with a lower gradient slope, the definition of productivity in Equation 3.3 combined with the resolution would result in a function, shown in Equation 3.4, that had a local extreme value with regards to gradient time.

$$Obj = res \cdot Prod \approx res \cdot (\frac{1}{t_{r2} + t_0}) \tag{3.4}$$

The main motivation of using an objective function such as Equation 3.4 was that the extreme point of the function could serve as a compromise between a good separation and a high productivity. The extreme point of the objective function could be found by studying its derivative: the derivative is 0 at the extreme point. This was also deemed advantageous, as compared to Configuration 1, one would not need to know what resolution was possible in order to find a suitable set point value. The set point value could simply be set to 0, and the controller would steer the process towards that value. The idea was that an optimal separation could be found automatically by applying this configuration.

Furthermore, a *weighting* of resolution versus productivity was introduced via exponents on the resolution and productivity as presented in Equation 3.5. The idea behind this was that the exponents n and m could be set to values < 1 to decrease or values > 1 to increase the weight of each parameter.

$$Obj = res^n \cdot Prod^m \approx res^n \cdot (\frac{1}{t_{r2} + t_0})^m \tag{3.5}$$

Several series of simulations were performed in a similar fashion to what was done

for Configuration 1, the difference being that instead of finding the resolution, the objective function value was calculated for each gradient time. Each series had a different exponent n, while the exponent m was kept at a constant value of 1 in order to reduce the number of combinations tested. The values tested were $\frac{3}{2}$, $\frac{4}{3}$, 1, $\frac{1}{2}$ and $\frac{1}{3}$.

Since the derivative of the objective function was chosen as set point, it needed to be calculated in some way. The decision was made to use a two-point linear derivative. The derivative of the objective at batch k being calculated using the point closest to it, denoted with the index c, along the u-axis according to Equation 3.6.

$$\frac{dObj_k}{du} = \frac{Obj_k - Obj_c}{u_k - u_c} \tag{3.6}$$

There was a risk of numerical errors as the distance between u_k and u_c approached 0. To avoid this, all points within a distance δ along the u-axis surrounding the point u_k were removed. This also served the purpose of introducing an intentional error as the controller steered the derivative $\frac{dObj_k}{du}$ towards the set point, which could avoid oscillatory behaviour. This is illustrated in Figure 3.2. The value of δ was set to $1 \cdot 10^{-2}$.

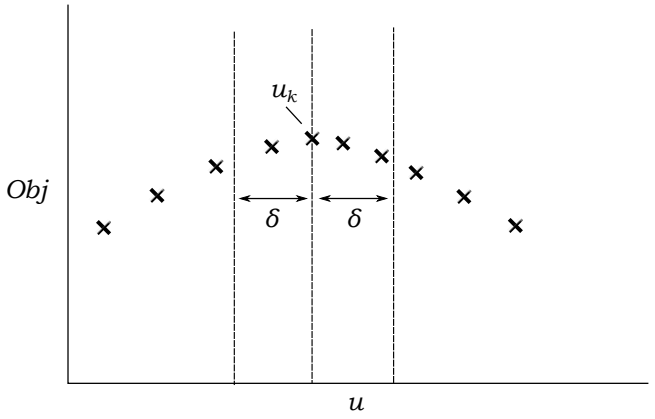


Figure 3.2: An illustration of the removal of points surrounding the current point k. All points within a distance δ from the current point in the u-axis direction are removed to avoid numerical problems when calculating the derivative of the objective function.

The controller was constructed using two memories. Gradient times and objective function values were stored in Memory 1, which was used to calculate the derivative of the objective function at batch k. The derivative calculated using Memory 1 was then stored in Memory 2, along with the gradient time. Memory 2 was then used by the ILC algorithm to compute the gradient time for the next batch, $k+1$. The overall structure of Configuration 2 is illustrated in Figure 3.3.

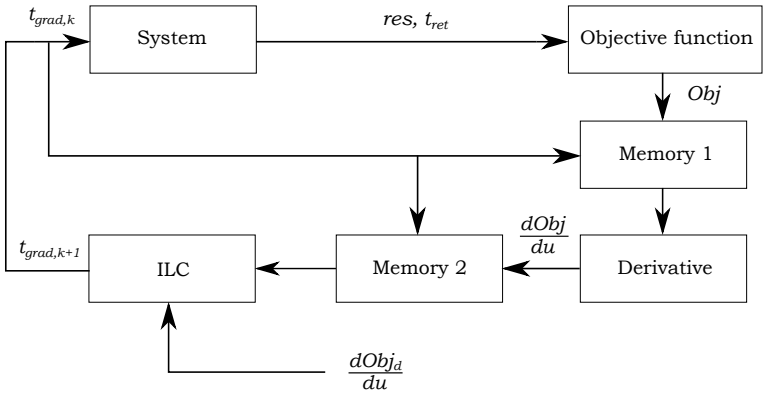


Figure 3.3: The overall structure of Configuration 2

3.4 Adapted ILC Algorithm

The ILC algorithm used by Xiong and Zhang (2003) was modified to function on a single-input-single-output (SISO) system with only one sample point, being the end of a batch run. This reduced the ILC law to Equation 3.7

$$\boldsymbol{U}_{k+1} = \hat{K}_k \boldsymbol{e}_k + \boldsymbol{U}_k \tag{3.7}$$

where $\hat{K}_k$ is calculated by modifying the definition of the learning rate in Equation 2.9, changing the amplifying and damping matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$ to scalar values q and r since there was only one sample point. The modified learning rate is described by Equation 3.8.

$$\hat{K}_k = \frac{G_s q}{G_s^2 q + r} \tag{3.8}$$

The prediction of the linear time-varying model at batch k also became a scalar value and could be calculated according to Equation 3.9

$$G_s = G_k = \frac{\boldsymbol{H}^T \cdot \boldsymbol{Z}}{\boldsymbol{H}^T \cdot \boldsymbol{H}} \tag{3.9}$$

Finally, the memory matrices $\boldsymbol{Z}$ and $\boldsymbol{H}$ were reduced to the vectors described by Equation 3.10

$$\boldsymbol{Z}_k = \begin{bmatrix} \beta^{L+k-1}\overline{y}_1^0 \\ \vdots \\ \beta^{k}\overline{y}_L^0 \\ \beta^{k-1}\overline{y}_1 \\ \vdots \\ \beta^{1}\overline{y}_{k-1} \\ 0 \end{bmatrix} \qquad \boldsymbol{H}_k = \begin{bmatrix} \beta^{L+k-1}\overline{u}_1^0 \\ \vdots \\ \beta^{k}\overline{u}_L^0 \\ \beta^{k-1}\overline{u}_1 \\ \vdots \\ \beta^{1}\overline{u}_{k-1} \\ 0 \end{bmatrix} \tag{3.10}$$

where every element $\overline{y}$ and $\overline{u}$ in position i in the vectors was defined as described by Equation 3.11

$$\overline{y} = y_i - y_k \qquad\qquad \overline{u} = u_i - u_k \tag{3.11}$$

The forgetting factor β was set to 0.98. At every new batch k that was run, the input and the corresponding output were appended to the vectors $\boldsymbol{H}_k$ and $\boldsymbol{Z}_k$, and the new learning rate K_k was computed. The error was calculated and the corrected input for the upcoming batch was calculated using Equation 3.7. The current batch

number was updated to $k = k+1$ and the cycle was repeated for the desired number of batches.

3.5 Experimental Design

The experiments performed could be divided into two parts, the first of which took place before applying the controller algorithm. These experiments were to identify the behavior the resolution and the objective as functions of gradient time with fixed $x_{B,i}$ and $x_{B,f}$. First, a few manual simulations were run with different combinations of $x_{B,i}$ and $x_{B,f}$ values, starting with 0 and 1, respectively. The gradient time was fixed at 100 minutes. The aim of these simulations was to find values that resulted in a short residence time and separation of peaks to set up for the following experiments.

Following this, more manual simulations were run with different gradient times until the lowest gradient time that still resulted in two distinct peaks was found. Then, as previously detailed, simulations of the chromatography process were run with gradient times between this value and 500 minutes. The resolution was calculated by using the data points generated by the *solve_ivp* function to find the top of the peaks and their widths. The tops of the peaks were found using the first and second derivatives of the chromatogram. A point was deemed a peak when the chromatogram's derivative at that point was zero, while the second derivative was negative. The base width of a peak was determined by finding the point between the beginning (the point where the concentration of compound rose above zero) of the peak and the top of the peak where the derivative was the highest, and extrapolating a straight line with that derivative as its slope. The point where the straight line crossed the baseline of the chromatogram was determined to be the beginning of the chromatogram. The same operation was done to find the end of the chromatogram, using instead the lowest value of the derivative between the top of the peak and its end. The advantage of this method was that a resolution could be calculated even when the peaks were not well resolved, as shown in Figure 3.4.

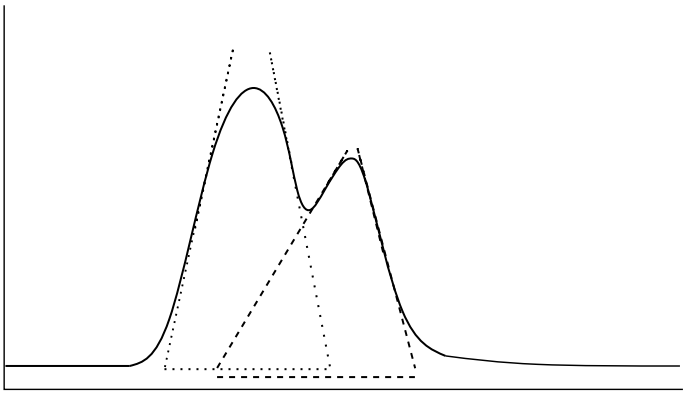


Figure 3.4: A chromatogram with poor resolution between two peaks, and the straight lines used to calculate the peak widths. The dotted lines show the calculation of the first peak's width, while the dashed lines show that of the second peak.

It is apparent from Figure 3.4 that this method is not always reliable: when the resolution is very poor, the width of the lower peak could be computed to be larger than it actually is and thus result in low resolutions being evaluated to even lower values. However, for the purposes of this project, this issue was considered acceptable. Suggestions on other methods for calculation of the resolution are detailed in Section 5.1.

In conjunction with this, the behavior of the objective function was mapped through several simulations with different exponents on the resolution in the objective function defined in Equation 3.5. The results from these simulations were plotted to show a $u - y$-graph, that is, a graph showing the output signal's dependence on the input signal. From these graphs, the initial memory points for the upcoming studies were chosen.

A scaling of both the input and output values was performed. The gradient time was scaled using a reference gradient time, $t_{ret,ref}$, which was chosen to 200 minutes. The scaling value of Configurations 1 and 2 were set to the maximum value found in the four initial points of their respective memory.

Following the completion of these primary studies, the actual ILC application was performed. First, a basic application was run with arbitrarily chosen parameters. Then, different parameters were tested based on the behavior of the controller. The parameters to be studied were the damping action of the controller, r, and the

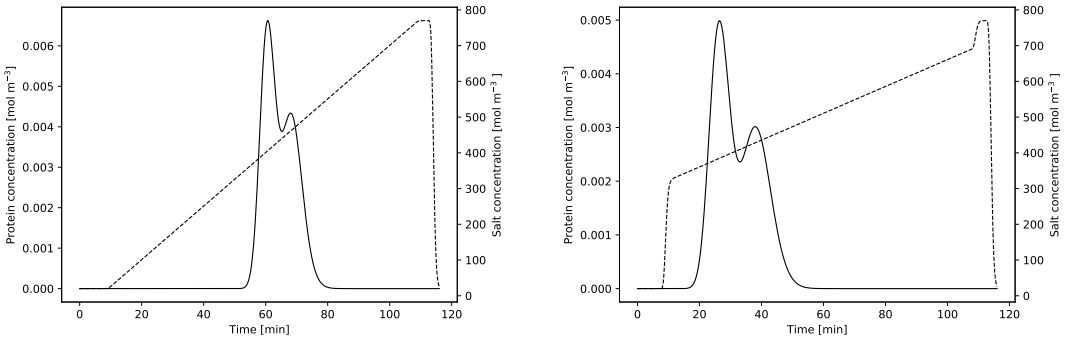
amplifying action, q. A simulation of 30 batches with a set pair of q and r values was run, and the found operating points were plotted in graphs showing the y values for each batch in relation to the set point, y_d. If the controller was found to be "too cautious", i.e. it took steps too small to come close to the set point, the damping was decreased. If the controller was "too aggressive" and displayed an overshoot of the set point, the damping was increased. Generally, only the value of r was changed while q was kept at a constant value of 1.

To evaluate the choice of parameters for a controller in a real application, two conditions are of particular interest: the *number of batches* it takes for the controller to reach y_d, and the *stability* of the controller once y_d is reached. Stability here is defined as the ability of the controller to maintain a steady output when the error, e, becomes so small that it is essentially only caused by noise in the data. The number of batches to reach y_d is of interest as having fewer batches results in a more efficient use of resources in a practical application, since each run consumes buffers and protein solution. Stability, on the other hand, is interesting as it applies a constraint on the aggressiveness of the controller: having a very low value of r in relation to q could result in a controller that amplifies the control action in an exaggerated manner due to noise in the data. However, as the 30-batch tests were performed by simulation and thus there was no measurement noise to speak of, only the number of batches to reach y_d was evaluated.

4 Result and Discussion

4.1 Primary study

Manual simulations led to the values $x_{B,i} = 0.4$ and $x_{B,f} = 0.9$ being found as a good span for short retention times and high resolutions when the gradient time was fixed to 100 minutes. Figure 4.1 displays the chromatogram resulting from these settings compared to a run based on the same gradient time, but with $x_{B,i} = 0$ and $x_{B,f} = 1$.

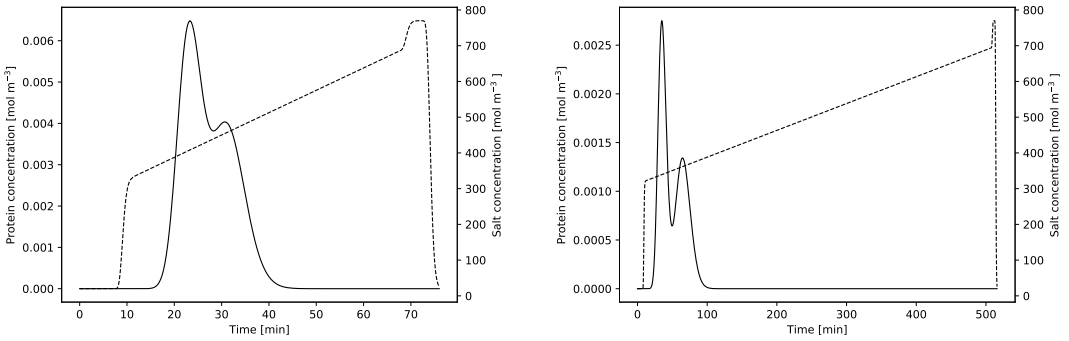


(a) **(b)**

Figure 4.1: Chromatography simulation results with a gradient time of 100 minutes. The dashed line represents the salt buffer concentration. A fraction of Buffer B going from 0 to 1 resulted in a resolution of 0.39 and a retention time of around 68 minutes (4.1a). A fraction going from 0.4 to 0.9 resulted in a resolution of 0.57 and a retention time of around 38 minutes (4.1b)

As seen in Figure 4.1b, the values $x_{B,i} = 0.4$ and $x_{B,f} = 0.9$ resulted in a resolution of 0.57 and a retention time of 38 minutes, which out-performed the base case which yielded $res = 0.39$ and $t_r = 68$ minutes. Thus, $x_{B,i} = 0.4$ and $x_{B,f} = 0.9$ were chosen as a foundation for all future experiments with variable gradient time.

The shortest viable gradient time found from manual simulations was determined to be 60 minutes, with a resulting resolution of 0.29. The resolution from a run with a t_{grad} of 500 minutes was found to be 0.87. The chromatograms from these runs are presented in Figure 4.2.



(a) **(b)**

Figure 4.2: Chromatography simulation results for $t_{grad} = 60$ min (4.2a) and $t_{grad} = 500$ min (4.2b). The resolutions were found to be 0.29 and 0.87, respectively.

Figure 4.2b highlights a problem with using only the gradient time to control the gradient slope: it has the demerit of causing long gradient times where the proteins have already eluted after only a small portion of the gradient time has passed. Such a problem could be avoided if either the final or initial fractions of Buffer B were used to control the process along with the gradient time, in a multiple-input configuration. Alternatively, the controller could be programmed to begin the regeneration phase once all components have been eluted.

The behavior of the resolution as a function of the gradient time in the interval $60 \leq t_{grad} \leq 500$ is displayed in Figure 4.3. The processes were run with gradients ranging between $x_{B,i} = 0.4$ and $x_{B,f} = 0.9$.

It is apparent from Figure 4.3 that, when only the gradient time is used to control the process, the resolution experiences great diminishing returns as t_{grad} increases, or even approaches a value asymptotically. A consequence of this is that the chosen system - the separation of cytochrome C and lysozyme in a HiTrap® Capto™ S ion-exchange column - is fairly difficult to control using the gradient time, especially if a higher resolution is desired. If a resolution of, for example, 1.5 was desired, it is likely that a controller would increase the gradient time towards infinity if that resolution is impossible to reach.

The set point value for the Configuration 1 experiments was set to 0.7, so that the resolution was achievable while still being in a range where a relatively small change

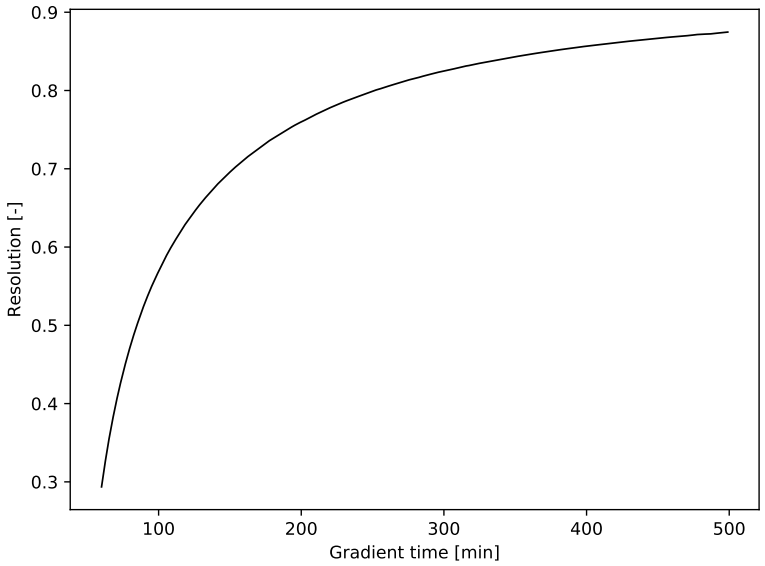


Figure 4.3: Resolution as a function of gradient time for gradient times $60 \leq t_{grad} \leq 500$, running between $x_{B,i} = 0.4$ and $x_{B,f} = 0.9$.

in gradient time yielded a significant change in resolution. The initial t_{grad} values in the memory chosen for the Configuration 1 and 2 simulations were 70, 80, 110 and 150.

Similarly, the objective as a function of gradient time for different exponents n on the resolution is presented in Figure 4.4.

Two decisions were made based on these results: firstly, the exponent of res was set to $\frac{1}{2}$. This was to ensure precision when calculating the derivative $\frac{dObj}{du}$, since the objective function was "sharper" around its maximum value when $n = \frac{1}{2}$ than when it was higher. However, since the optimum was also displaced when an exponent was used, the problem of poor precision remained if a higher weight of the resolution was desired. The constructed objective function and its behavior, for example when manipulating the exponent m on the productivity instead, could be investigated further. A different objective function could also possibly be constructed.

Secondly, the initial memory points for Configuration 2 were set. The values chosen for t_{grad} were 70, 80, 110 and 150, identical to those of Configuration 1. This would result in four values of Obj, and in turn three values of $\frac{dObj}{du}$. The reasoning was that the optimum, and thus the set point, was situated in the middle of this interval.

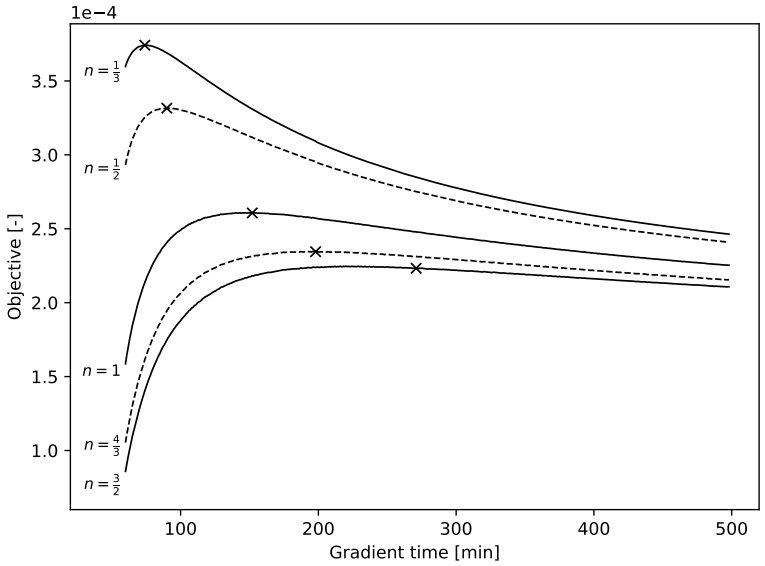


Figure 4.4: Objective function as a function of gradient time for $60 \leq t_{grad} \leq 500$. Different exponents n were applied to the resolution. The cross-marked points indicate the location of the local extrema, that is, the points where the derivative $\frac{dObj}{du} = 0$.

Another reason was that for Configuration 2 in particular, a good spread of the points was important since the calculation of the derivative was quite simple. As a consequence, the calculation of the derivative was heavily reliant on the points being spread out well enough so that two adjacent points were not far from each other along the u-axis but close to each other along the y-axis. If such a scenario occurred, there was a risk that a derivative of 0 could be found without the controller being anywhere near the optimum.

The constructed objective function may be too complex for straight-forward application in an ILC. The different optima shown in Figure 4.4 imply that the optimum shifts according to some logarithmic function when the exponent n is changed. This complicates the decision of how a parameter in the objective function should be weighted.

4.2 Configuration 1

Figure 4.5 shows the controller output from the 30-batch simulations on Configuration 1 for four different values of the damping r.

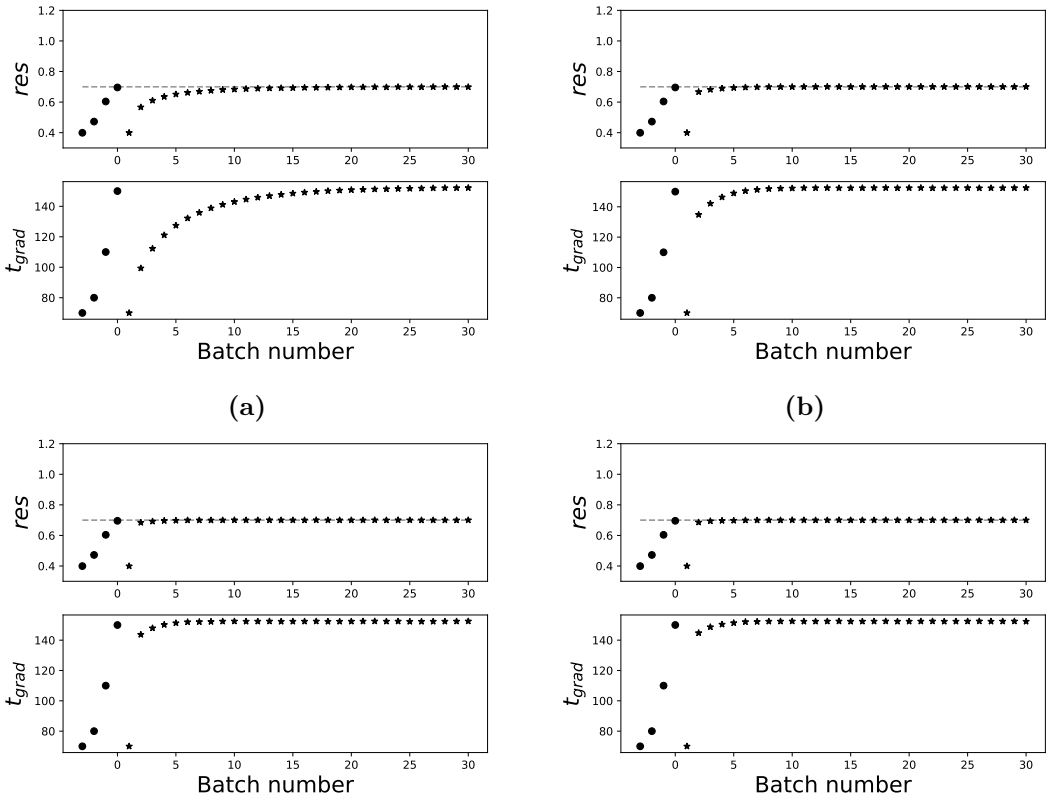


Figure 4.5: Process control output graphs for Configuration 1, where the amplifying parameter q is fixed to 1, and the damping parameter r is set to 1 (4.5a), $1 \cdot 10^{-1}$ (4.5b), $1 \cdot 10^{-2}$ (4.5c) and $1 \cdot 10^{-3}$ (4.5d). The circle markers represent the four initial memory data points, whereas the star-shaped markers show the operating points chosen by the controller. The dashed line marks the set point value of 0.7.

As Figure 4.5 shows, decreasing the damping in relation to the amplification q increases the "aggressiveness" of the controller, allowing it to find the desired set point more quickly. Fewer batches were needed before the set point was reached. However, it is important to take into consideration that the experiments that were run were based on simulations of a chromatography process, and thus were free from measurement noise. If this configuration was to be applied to a real process, the damping r could not be recklessly decreased to improve the control speed, as the controller would possibly be more sensitive to measurement noise and thus unstable.

4.3 Configuration 2

The 30-batch simulation results from Configuration 2 are presented in Figure 4.6. The experiments were run with a constant amplification $q = 1$ and differing damping parameters r. The values of r shown are 1, $1 \cdot 10^1$, $1 \cdot 10^2$ and $1 \cdot 10^3$.

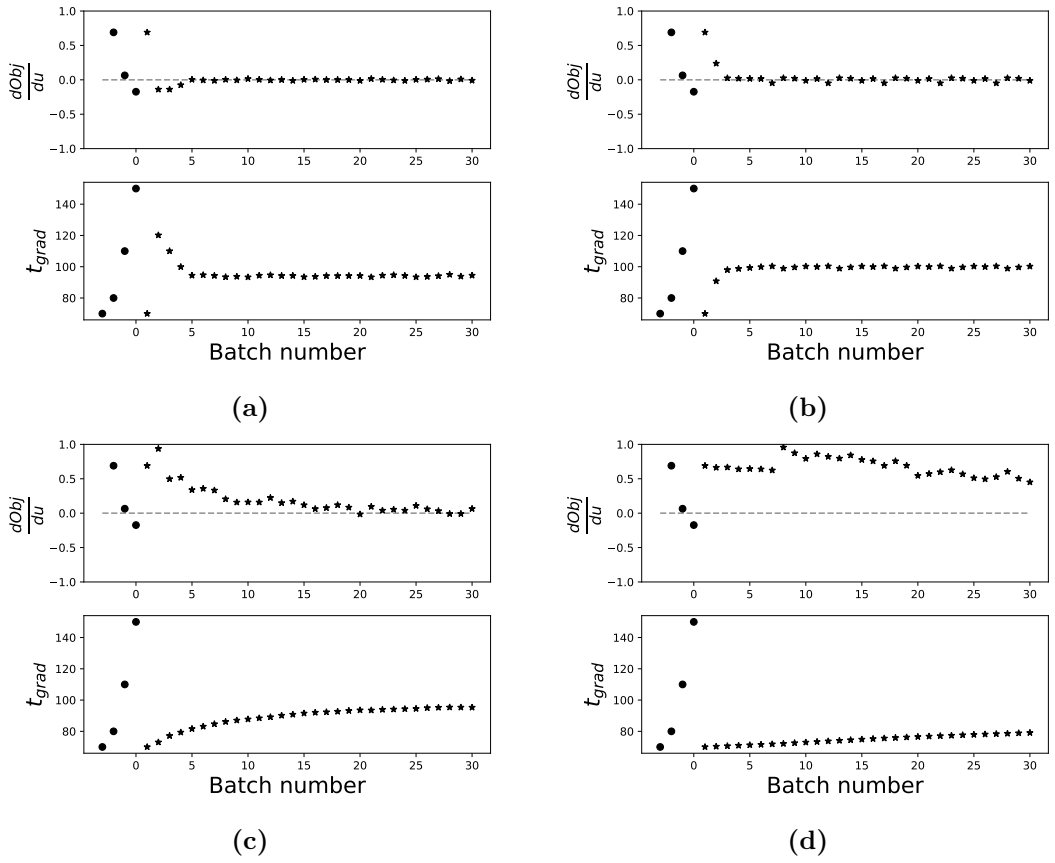


Figure 4.6: Process control output graphs for Configuration 2, where the amplifying parameter q is fixed to 1, and the damping parameter r is set to 1 (4.5a), $1 \cdot 10^1$ (4.6b), $1 \cdot 10^2$ (4.6c) and $1 \cdot 10^3$ (4.6d). The circle markers represent the four initial memory data points, whereas the star-shaped markers show the operating points chosen by the controller. The dashed line marks the set point value of 0.

From this first overview, it appears that the experiments indicate a similar behavior to those from Configuration 1: a lower damping results in faster control action and the set point is reached rather quickly. However, a closer look of the control graphs, shown in Figure 4.7, gives a more detailed picture of the control action.

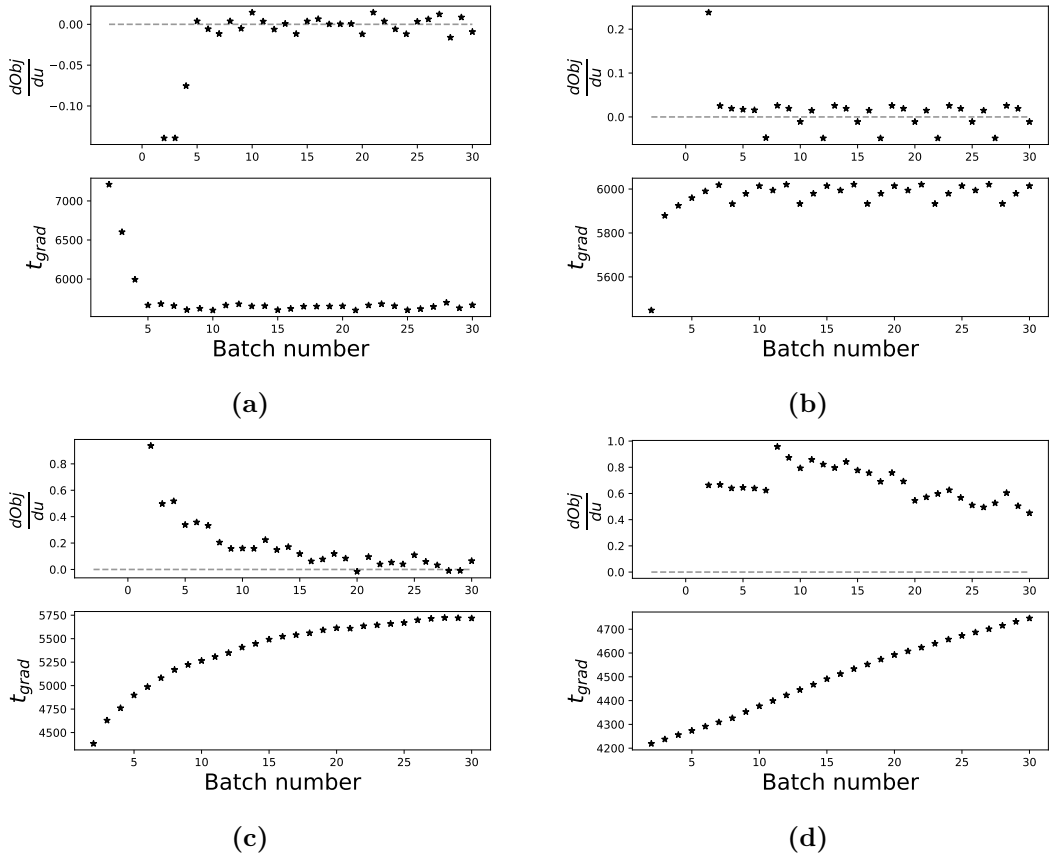


Figure 4.7: Process control output graphs for Configuration 2, where the amplifying parameter q is fixed to 1, and the damping parameter r is set to 1 (4.7a), $1 \cdot 10^1$ (4.7b), $1 \cdot 10^2$ (4.7c) and $1 \cdot 10^3$ (4.7d). The graphs are more closely centered around the controller trajectory, giving a closer look at the input and output values when they approach the set point

Figure 4.7, indicates that while the controller appears to move aggressively towards the set point, it oscillates quite heavily around it once nearby. This could be attributed to the roughness of the derivative approximation: since the derivative is calculated linearly between two adjacent points, it is going to deviate heavily from its true value when there are very few, very distant points available in memory. Since the controllers in 4.7a and 4.7b are relatively aggressive, they took larger steps when correcting the error and thus made the derivative calculation less accurate at the beginning of the run. In addition, since these past points with unreliably calculated derivatives remain in the ILC memory, they will continue to affect the controller until their effect decreases enough due to the forgetting factor β. If the objective function is not "sharp enough", i.e. the local extrema is not well-defined, it could take several batches before the controller had enough points to accurately estimate the derivative and find the real optimum. This means that it is more critical to have

a well-defined optimum in the process if an aggressive controller is desired, especially if high precision is desired. If a controller that reaches the set point quickly and accuracy is of no great concern, then these parameters may work well.

The less aggressive controllers, on the other hand, achieved more precise estimates of the derivative due to the smaller steps taken towards the optimum. The effects of this are quite visible in Figure 4.7c: the points oscillate less around their trajectories. However, this has the consequence of the controller being slow in taking action and reaching the set point, and Figure 4.7d shows that it does not reach the set point at all during the first 30 batches. Figure 4.7d shows another anomalous behavior: the controller actually oscillates significantly in the beginning of the run. This could be explained by the removal of nearby points from the process memory, detailed in Section 3.3. Since the derivative calculation ignores points that are too close to the current point along the u-axis, there is a possibility of the derivative calculation being wrongly estimated. This behavior is especially clear around batch 12 in Figure 4.7d, where the value of the derivative suddenly spikes once the input value has increased enough. If a higher precision is desired and speed of control (and thus, also consumption of buffers and protein sample solution) is of no particular interest, then these less aggressive controllers may work well.

While all the results from Configuration 2 imply that it is possible for an ILC algorithm to find an optimal separation using an objective function, the fact that the function chosen in this thesis requires a derivative to be calculated makes it difficult to apply. The controller is too dependent on already having several points in memory that are well-spread around the optimum. If such a memory exists, the objective function may very well be applicable. However, if the point is to be able to find the optimum in as few batches as possible and without having an extensive memory of operating points available, then the choice of objective function leaves a lot to be desired.

Table 4.1 summarizes the results of Configuration 2.

Table 4.1: The summarized results from Configuration 2.

Behavior of controller	Parameters	Pros	Cons
Aggressive	$q = 1$, $r_1 = 1$, $r_2 = 1 \cdot 10^1$	Fast action, approaches the set point quickly	Less accurate for the first batches, sensitive to less well-defined objective functions
Cautious	$q = 1$, $r_1 = 1 \cdot 10^2$, $r_2 = 1 \cdot 10^3$	More accurate estimate of the derivative	Slow action, reaches the set point very slowly, risk of wrongly estimated derivative at the beginning of a run

5 Conclusion

This thesis aimed to evaluate the use of iterative learning control to control and automatize separation in downstream processing of pharmaceutical proteins. ILC shows promise as both a control and automation method, particularly when directly controlling a process variable such as the resolution of two peaks. However, the design of an ILC does require some knowledge about the separation and what resolution is achievable before applying the controller.

When it comes to finding optimal separation parameters by means of an objective function, the iterative learning controller does show that it is possible to find an optimum. However, the objective function chosen in this thesis may not be the optimal choice if quick and precise control is desired, since improving its speed sacrifices its accuracy. A different objective function that does not require a derivative to be estimated, or a more accurate method to calculate the derivative, may be preferable.

5.1 Further Work

Applying a new method for calculation of the resolution between peaks would benefit both controller configurations. The method showcased by 3.4 shows a risk of resulting in lower resolutions for already low-resolved peaks. Several alternatives are available. For example, instead of using the derivatives on both sides of each individual peak to compute the width, the derivative of the "outermost" side of a peak could be mirrored around the top of the peak. Such a method would rely on the peaks being somewhat symmetrical to be accurate. Another method would be to fit Gaussian curves to each peak.

It would be interesting to see the basic ILC configuration applied to a multiple-input-multiple-output (MIMO) case. For example, it would be possible to control both the resolution and the retention time by letting the controller change the initial fraction

of buffer B, $x_{B,i}$, and the gradient time t_{grad}. This would increase the dimension of the damping and amplification and turn them into matrices, $\boldsymbol{Q}$ and $\boldsymbol{R}$. These matrices would then act as weighting on the two inputs and the two outputs, in a similar fashion to what the exponents on the objective function do in configuration 2. There is a possibility that using this MIMO configuration, an optimal control of the separation could be achieved without applying an objective function.

An alternative would of course be to develop a new objective function. If a linear objective function, constructed in such a way that no derivative estimation is required, could be constructed, the issues stemming from the derivative estimations in this thesis could be circumvented. Another option would be to apply some sort of regression on the existing data points in the current objective function approach. If a curve is fitted to the operating points when there are few points available, it is possible that a better estimate of the derivative could be achieved, thus allowing for a more aggressive control to be applied without sacrificing precision. Of course, great care would need to be taken when deciding on such a curve fit.

The configurations tested in this thesis were all based on simulations, meaning that no measurement noise was present during the experiments. It would be interesting to apply a small noise to these simulations and see how the controller behaves with the damping and amplification parameter values tried in this project. Adding disturbances midway through an experiment, or studying the effect of the initial history points would also be interesting. An example of how this could be done is to, at some batch number k, add a disturbance to the previous batch's control input to see how the controller adapts to it. One could also set different initial history points by, for example, having all of them placed closer or further appart from each other, or having them placed before or after the optimum in configuration 2. Eventually, applying a controller tuned using simulations on a real chromatography system would be interesting as well.

References

Ahn, H.-S., Chen, Y., & Moore, K. L. (2007). Iterative learning control: Brief survey and categorization. *Trans. Sys. Man Cyber Part C*, *37*(6), 1099–1121. doi:10.1109/TSMCC.2007.905759

Arkell, K. (2017). *Modeling and optimization of reversed-phase chromatography: Effects of modulators and temperature* (Doctoral dissertation, Lund University). Defence details Date: 2017-12-01 Time: 13:15 Place: lecture hall K:B, Kemicentrum, Getingevägen 60, Lund University, Faculty of Engineering LTH, Lund External reviewer Name: Lenhoff, Abraham Title: Professor Affiliation: University of Delaware, USA —.

Bajpai, P. (2018). Chapter 24 - process control. In P. Bajpai (Ed.), *Biermann's handbook of pulp and paper (third edition)* (Third Edition, pp. 483–492). doi:https://doi.org/10.1016/B978-0-12-814238-7.00024-6

Davis, M. (1984). Numerical methods & modeling for chemical engineers.

HiTrap Capto S cation exchange chromatography column. (2020). Retrieved March 11, 2020, from https://www.gelifesciences.com/en/us/shop/chromatography/prepacked - columns / ion - exchange / hitrap - capto - s - cation - exchange - chromatography-column-p-00689#tech-spec-table

Jakobsson, N. (2006). *A model based approach to robustness analysis and optimisation of chromatography processes* (Doctoral dissertation, Department of Chemical Engineering). Defence details Date: 2006-05-24 Time: 13:15 Place: Room A, Chemical centre, Getingevägen 60, Lund Institute of Technology External reviewer(s) Name: Staby, Arne Title: PhD, Manager Protein Purification Affiliation: Novo Nordisk —.

Jandera, P., & Henze, G. (2011). Liquid chromatography, 1. fundamentals, history, instrumentation, materials. In *Ullmann's encyclopedia of industrial chemistry*. doi:10.1002/14356007.b05_237.pub2. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/14356007.b05_237.pub2

Al-Kaisy, S. (2015). Model calibration and optimization of a protein purification process. Student Paper.

LeVeque, R. J. (2002). Finite volume methods. In *Finite volume methods for hyperbolic problems* (pp. 64–86). Cambridge Texts in Applied Mathematics. doi:10.1017/CBO9780511791253.005

Roos, P. H. (2000). Chapter 1 - ion exchange chromatography. In M. Kastner (Ed.), *Protein liquid chromatography* (Vol. 61, pp. 3–88). Journal of Chromatography Library. doi:https://doi.org/10.1016/S0301-4770(08)60529-1

Schulte, M., & Epping, A. (2005). Fundamentals and general terminology. In *Preparative chromatography* (Chap. 2, pp. 9–49). doi:10.1002/3527603484.ch2. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/3527603484.ch2

Schulte, M., Wekenborg, K., & Wewers, W. (2005). Process concepts. In *Preparative chromatography* (Chap. 5, pp. 173–214). doi:10.1002/3527603484.ch5. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/3527603484.ch5

Xiong, Z., & Zhang, J. (2003). Product quality trajectory tracking in batch processes using iterative learning control based on time-varying perturbation models. *Industrial & Engineering Chemistry Research*, *42*(26), 6802–6814. doi:10.1021/ie034006j. eprint: https://doi.org/10.1021/ie034006j