Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Natural Sciences
Department of Chemical Engineering

Anne Øyen Halås

Modelling and model predictive control of a semi-batch reactor for emulsion polymerization of PVC

Master's thesis in Chemical Engineering and Biotechnology
Supervisor: Magne Hillestad (IKP), Peter Singstad (Cybernetica AS)
June 2020

NTNU
Norwegian University of
Science and Technology

Anne Øyen Halås

Modelling and model predictive control of a semi-batch reactor for emulsion polymerization of PVC

Master's thesis in Chemical Engineering and Biotechnology
Supervisor: Magne Hillestad (IKP), Peter Singstad (Cybernetica AS)
June 2020

Norwegian University of Science and Technology
Faculty of Natural Sciences
Department of Chemical Engineering

NTNU
Norwegian University of
Science and Technology

Preface

This thesis was written during the spring of 2020. It concludes the work performed as the final part of a master's degree program within Chemical Engineering and Biotechnology at Norwegian University of Science and Technology (NTNU). This project was an extension of a specialization project, completed during the autumn of 2019. The work and results of this preliminary project are also utilized in this thesis. The relevant content from the preliminary project has been included, in condensed form, to give the reader a comprehensive understanding of the problem. Both projects were written in collaboration with Cybernetica AS, who proposed the projects, and Inovyn Norge, who provided necessary process data, from their facilities in Porsgrunn.

June 2020, Trondheim, Norway
Anne Øyen Halås

Acknowledgement

I want to express my gratitude towards Cybernetica AS, who granted me the opportunity of a summer internship, with subsequently specialization project and master thesis. Thanks to Peter Singstad, who believed in me and has been my co-supervisor at Cybernetica; your guidance and opinions have been of most importance during the course of my project. The learning outcome has been tremendous.

As I have spent a lot of my time at Cybernetica's offices, I would like to thank all employees for making me feel welcome. It has been inspiring to have such engaged co-workers. A special thanks to Fredrik Gjertsen; both your knowledge and patience has been admirable. I also wish to thank me supervisor at NTNU, Magne Hillestad, and Asbjørn Holt at Inovyn Norge for support and valuable inputs.

Thanks to the Department of Chemical Engineering for, two, most interesting years, and kudos to Victoria Glott for making me take breaks.

Finally, thanks to my parents, Anita Ø. Halås and Arnstein Halås, my sister, Aud Ø. Halås, and to Erik A. Koren; for your love, support and encouragement.

Abstract

The objective of this thesis is to study the implementation of Nonlinear Model Predictive Control (NMPC) on a reactor for an emulsion polymerization process. A model is derived using first principles and serves as a plant replacement model, in addition to an integrated part of the controller. The majority of this model was previously developed, and some customization is done to make it applicable for NMPC. The control system is tuned, and the performance is demonstrated by simulations.

The theoretical aspects of polymers and emulsion polymerization are explicitly provided in this thesis. Necessary model equations describing this process is implemented in C code, in one of Cybernetica's templates for polymer processes. The process studied is the emulsion polymerization of Vinyl Chloride Monomer (VCM) to the desired product Poly-Vinyl Chloride (PVC). The model is validated and modified against process data provided by Inovyn Norge. It is concluded that the model gave satisfactory results compared to process data. It is emphasized that the model can be done applicable for other systems than PVC, with some effort and an understanding of the process.

The concepts behind optimization, Model Predictive Control (MPC) and state- and parameter estimation is provided. In addition, is some specific aspects of control and NMPC implementation on this process, presented. Some deviations between the model used as plant replacement, and in the controller, is deliberately implemented. This is done to imitate the fact that a model never will be an exact representation of reality. A Kalman Filter (KF) was implemented as the state- and parameter estimator, and was tuned and tested.

The implementation of NMPC, in combination with state- and parameter estimation, on the reactor, is considered successful in this project. The results indicate that the control system can exploit the cooling capacity by feeding reactants and controlling the temperature in an efficient matter. This is done by still ensuring safe operation and meeting quality requirements.

Sammendrag

Målet med dette prosjektet er å studere implementering av ulineær modellprediktiv regulering (NMPC) for en reaktor for emulsjonpolymerisasjon. En modell utledes ved bruk av fysiske prinsipper, og brukes til å erstatte målte data fra prosessanlegget. Den brukes også som modell for NMPC'en. Størsteparten av denne modellen ble utviklet i et tidligere prosjekt, men noen tilpasninger utføres for å gjøre den anvendelig for NMPC. Ytelsen til kontrollsystemet testes og demonstreres ved simuleringer.

De teoretiske aspektene ved polymerer og emulsjonspolymerisasjon er spesifikt utdypet i denne oppgaven. Modellikningene som beskriver denne prosessen er implementert i C-kode, i en av Cyberneticas maler for polymerprosesser. Prosessen som studeres er emulsjonspolymerisasjonen av vinyl klorid (VCM), til det ønskede produktet poly-vinyl klorid (PVC). Modellen valideres og modifiseres mot prosessdata fra Inovyn Norge. Modellen ga akseptable resultater sammenlignet med prosessdata. Det understrekes at modellen kan gjøres anvendelig for andre systemer enn PVC, med litt anstrengelse og forståelse av prosessen.

Konseptene bak optimalisering, MPC (Modellprediktiv regulering) og tilstands- og parameterestimering er gitt. I tillegg er noen spesifikke aspekter ved regulering og NMPC implementering av en semi-batch reaktor for emulsjonspolymerisasjon presentert. Noen avvik mellom modellene som ble brukt til å erstatte målte data for prosessanlegget, og i NMPC'en, ble bevisst implementert. Dette skal etterligne det faktum at en modell aldri vil være en eksakt representasjon av virkeligheten. Et Kalman Filter (KF) ble implementert som tilstand- og parameterestimator. Applikasjonen testes så både med og uten estimatoren.

Implementeringen av NMPC, i kombinasjon med estimering av tilstand og parametere, ansees som vellykket i dette prosjektet. Resultatene indikerer at kontrollsystemet er i stand til å utnytte kjølekapasiteten ved å føde reaktanter og kontrollere temperaturen på en effektiv måte. Sikker drift og kvalitetskrav er da opprettholdt.

Table of Contents

Nomenclature

For consistency reasons, this will be the strategy used for the nomenclature

- Descriptive sub- and superscripts are of roman type, so are units.
- Italic type represents a quantity or a running number (indices).
- Compounds will be described using capital letters; Monomer (M), Water (W), Surfactant/Emulgator (S), Initiator (I), Polymer (P), Chain-transfer agents (AX), Radical (R). The relevant compound for compound-specific properties will be denoted with subscripts, for example, the molecular weight of monomer: M_{M}.
- Phases will be described using lower case letters; Polymer phase (p), water phase (w), gas phase (g), free phase (f). The given phase for phase-specific properties will be denoted with superscripts, for example, the volume of the gas phase: V^{g}.
- Dotted variables indicated time-derivatives, for example change in reactor mass; $\dot{m}_{\mathrm{R}}$.
- Hatted variables indicate flow, for example, the mass flow of fresh water; $\hat{m}_{\mathrm{fresh}}$. Note that volume flows are excepted from this standard, and is denoted by q.

Latin symbols

Symbol	Description	Unit
A	Jacobian matrices	
$A_{\mathrm{J,amb}}$	Jacket-ambient area	m^2
$A_{\mathrm{R,amb}}$	Reactor-ambient area	m^2
$A_{\mathrm{R,J}}$	Reactor-jacket area	m^2
a_{s}	Interfacial area of the particles	m^2/molecule
B	Jacobian matrices	
C	The relative rate coefficient of radical termination i polymer phase	

CTA	Chain transfer agent	
c	Rate coefficient for termination in Smith and Ewart's balance of number of radicals per particle	
$c_{p,i}$	Heat capacity of component i	J/kg/K
$c_{p,\mathrm{J}}$	Heat capacity of jacket content	J/kg/K
$c_{p,\mathrm{rec}}$	Heat capacity of the recycled stream	J/kg/K
$c_{p,\mathrm{steel}}$	Heat capacity of steel	J/kg/K
D_i	Number of inactive polymer chains of chain length i	
d	Diameter of the particles	m
d_u	Weight vector	
d_x	Weight vector	
f	Efficiency factor for the initiator	
$f(\cdot)$	Process model	
f_1	Efficiency factor for Initiator 1	
f_2	Efficiency factor for Initiator 2	
$\hat{f}^0_{\mathrm{M}}$	Fugacity of monomer at standard state	Pa
$\hat{f}^j_{\mathrm{M}}$	Fugacity of monomer in phase j, j = f, p, g, w	Pa
$g(\cdot)$	Measurement model	
H	Output matrix	
$h(\cdot)$	Function that adds noise to the parameters	
h_{fg}	Latent heat of steam	J/kg
I	Initiator	
I_1	Initiator 1	
I_2	Initiator 2	
$[i]$	Concentration of component i, i = M, I, S, P, W, AX, R	mol/m^3
$[i]^j$	Concentration of component i in phase j, i = M, I, S, P, W, AX, j = f, p, g, w	mol/m^3
J	Objective function	
K_1	Monomer in water solubility constant	
K_k	Estimator gain matric	
k	Current time sample	
k'	Rate coefficient for desorption in Smith and Ewart's balance of number of radicals per particle	
k''	Constant in Smith and Ewart's equation for nucleation	
k'''	The rate coefficient describing the radical exit from the particles to the water phase in Li and Brook's prediction	

k_{ads}	Rate constant for radical adsorption from water phase	$\mathrm{m}^3/\mathrm{mol/s}$
$k_{\mathrm{ct}}^{\mathrm{AX}}$	Rate constant for chain transfer to chain transfer agents	$\mathrm{m}^3/\mathrm{mol/s}$
$k_{\mathrm{ct}}^{\mathrm{M}}$	Rate constant for chain transfer to monomer	$\mathrm{m}^3/\mathrm{mol/s}$
$k_{\mathrm{ct}}^{\mathrm{P}}$	Rate constant for chain transfer to polymer	$\mathrm{m}^3/\mathrm{mol/s}$
k_{d}	Rate constant for decomposition of initiator	$1/\mathrm{s}$
k_{d_1}	Rate constant for decomposition of Initiator 1	$1/\mathrm{s}$
k_{d_2}	Rate constant for decomposition of Initiator 2	$1/\mathrm{s}$
k_{des}	Rate constant for desorption of radicals	$1/\mathrm{s}$
k_{i}	Rate constant for chain initiation	$1/\mathrm{s}$
$k_{\mathrm{i}}^{\mathrm{A}}$	Rate constant for chain initiation by radical fragment from the chain transfer agent	$1/\mathrm{s}$
k_{p}	Rate constant for propagation	$\mathrm{m}^3/\mathrm{mol/s}$
k_{t}	Rate constant for termination	$\mathrm{m}^3/\mathrm{mol/s}$
$k_{\mathrm{t}}^{\mathrm{w}}$	Rate constant for termination in water phase	$\mathrm{m}^3/\mathrm{mol/s}$
k_{tc}	Rate constant for termination by combination	$\mathrm{m}^3/\mathrm{mol/s}$
k_{td}	Rate constant for termination by disproportionation	$\mathrm{m}^3/\mathrm{mol/s}$
M	Monomer	
M	Control horizon	
M_i	Molecular weight of component i, i = M, I, S, P, W, AX	kg/mol
$\overline{M_{\mathrm{n}}}$	Number average molecular weight	kg/mol
$\overline{M_{\mathrm{w}}}$	Weight average molecular weight	kg/mol
$\hat{m}_{\mathrm{fresh}}$	Mass flow of fresh water feed	kg/s
$\hat{m}_{\mathrm{fresh}}^{\mathrm{delayed}}$	Mass flow of fresh water feed, taken into account the delay in the pipe lines	kg/s
m_i	Mass of component i, i = M, I, S, P, W, AX	kg
$\hat{m}_i$	Mass flow of component i, i = M, I, S, P, W, AX	kg/s
m_i^j	Mass of component i in phase j, i = M, I, S, P, W, AX, j = f, p, g, w	kg
m_{J}	Mass of content contained inside jacket (water)	kg
$\hat{m}_{\mathrm{J}}$	Mass flow of jacket content	kg/s
$m_{\mathrm{P}}^{\text{all particles}}$	Mass of all polymer particles	kg
$m_{\mathrm{P}}^{\text{one particle}}$	Mass of one polymer particle	kg
$\hat{m}_{\mathrm{rec}}$	Mass flow of recycled water, before addition of fresh water	kg/s
m_{steal}	Mass of steal (reactor)	kg
$\hat{m}_{\mathrm{steam}}$	Mass flow of steam	kg/s
N	Prediction horizon	
N_{A}	Avogadros constant	1/mol

N_n	Number of particles containing n radicals	
N_{T}	Total number of particles	mol
n	Number of radicals	
$\bar{n}$	Average number of radicals per particle	
n_i	Moles of specie i, i = M, I, S, P, W, AX, R	mol
$\hat{n}_i$	Molar flow of specie i, i = M, I, S, P, W, AX	mol/s
$n_{i,0}$	Initial moles of specie i, i = M, I, S, P, W, AX, R	mol
n_i^j	Moles of specie i in phase j, j = f, p, g, w, i = M, I, S, P, W, AX, R	mol
n_u	Number of inputs	
n_x	Number of states	
$\mathrm{P_i}$	Growing polymer of chain length i	
P_i	Number of active polymer chains of chain length i	
P_f	Weighting matrix	
$\bar{P}_k$	A priori covariance	
$\hat{P}_k$	A posteriori covariance	
P_{tot}	Total amount of radicals	mol
p	Unknown parameter which will be subject for estimation	
p_{M}	Partial pressure of monomer	Pa
$p_{\mathrm{M}}^{\mathrm{sat}}$	Saturation pressure of monomer	Pa
p_{R}	Pressure in the reactor	Pa
p_{W}	Partial pressure of water	Pa
$p_{\mathrm{W}}^{\mathrm{sat}}$	Saturation pressure of water	Pa
Q	Covariance matrix for process noise	
$\dot{Q}$	Heat transfer	W
Q_1	Weighting matrix	
q	Volume flow	m^3/s
q_{fresh}	Volume flow of fresh water	m^3/s
q_{steam}	Volume flow of steam	m^3/s
R	Covariance matrix for measurement noise	
$\mathrm{R}\cdot$	Radical	
R_1	Weighting matrix	
R_2	Weighting matrix	
R_3	Weighting matrix	
R_{fresh}	Design parameter for the valve opening of fresh water	
R_{steam}	Design parameter for the valve opening of steam	
R^{g}	Gas constant	$\mathrm{Pam}^3/\mathrm{K/mol}$
R_{d}	Reaction rate for composition of initiator	mol/s

R_{d_1}	Reaction rate for decomposition of Initiator 1	mol/s
R_{d_2}	Reaction rate for decomposition of Initiator 2	mol/s
R_{I}	Total rate of formation of radicals that can be used for polymerization	mol/s
R_{I_1}	Rate of formation of radicals produced from Initiator 1 that can be used for initiation	mol/s
R_{I_2}	Rate of formation of radicals produced from Initiator 2 that can be used for initiation	mol/s
R_{i}	Reaction rate for chain initiation	mol/s
R_i	Net generation or consumption of a specie i through chemical reactions	mol/s
R_{p}	Reaction rate for propagation	mol/s
R_{t}	Reaction rate for termination	mol/s
$R_{\mathrm{t}}^{\mathrm{w}}$	Reaction rate for termination in water phase	mol/s
r	Weighting vector	
S	Surfactant	
S	Amount of surfactants	
T_{amb}	Ambient temperature	K
$T_{\mathrm{feed},i}$	Temperature of reactant i fed into the reactor, i = M, I, S, W, AX	K
T_{fresh}	Temperature of the fresh water feed	K
T_{J}	Temperature of water inside the jacket	K
$T_{\mathrm{J}}^{\mathrm{min}}$	Minimum temperature possible for the water contained in the jacket	K
$T_{\mathrm{J,in}}$	Inlet jacket temperature	K
$T_{\mathrm{J,out}}$	Outlet jacket temperature	K
T_{R}	Reactor temperature	K
T_{rec}	Temperature of the recycled cooling water, after the heat exchanger	K
$\bar{T}_{X_{\mathrm{M}}}$	Conversion average temperature	K
t_{batch}	Batch time	s
$U_{\mathrm{J,amb}}$	Heat transfer coefficient between jacket and ambient	$\mathrm{W/m^2/K}$
$U_{\mathrm{R,amb}}$	Heat transfer coefficient between reactor and ambient	$\mathrm{W/m^2/K}$
$U_{\mathrm{R,J}}$	Heat transfer coefficient between reactor and jacket	$\mathrm{W/m^2/K}$
u	Inputs of the system	
V	Volume	m^3
$V_f(\cdot)$	Terminal cost function	
$V_{\mathrm{fluid,s}}$	Volume of liquid and solid in the system if no monomer or water was present in the gas phase	m^3

V^j	Volume of phase j, j = f, p, g, w	m^3
V_i^j	Volume of component i in phase j, j =f, p, g, w, i = M, I, S, P, W, AX	m^3
V_R	Volume of the reactor	m^3
v	Process noise	
$v_{p,k}$	Noise on the parameters	
v^p	Volume of one polymer particle	m^3
w	Measurement noise	
w_l	The liquid volume fraction in the reactor	
X_M	Monomer conversion	
X_M^c	Critical monomer conversion	
x	States of the system	
x'	Augmented states	
x^{ref}	Reference value of the state	
$\bar{x}_k$	A priori estimates	
$\hat{x}_k$	A posteriori estimates	
y	Measurements	
y_i	Volume fraction of component i in gas phase, $i = M, W$	
z	Valve position	
z_{fresh}	Valve position of the fresh water feed	
z_{steam}	Valve position of the steam inlet	
$\mathcal{E}$	Indices for equality constrains	
$\mathcal{I}$	Indices for inequality constrains	

Greek symbols

Symbol	Description	Unit
α_M	Monomer activity	
ρ'	Rate coefficient for adsorption in Smith and Ewart's balance of number of radicals per particle	rad/m^3
ρ''	Rate of formation of radicals per volume water (Smith and Ewart's equation for nucleation)	
ρ_{fresh}	Density of the recycled stream	kg/m^3
ρ_{steam}	Density of steam	kg/m^3
ρ_i	Density of component i, i = M, I, S, P, W, AX	kg/m^3
ρ_i^j	Density of component i in phase j, i = M, I, S, P, W, AX, j = f, p, g, w	kg/m^3

μ	Rate of increase in particle volume (Smith and Ewart's equation for nucleation)	m^3/s
μ_k	k'th live moment	mol/m^3
ϕ	Set of decision variables	
ψ	Parameter in Li and Brook's prediction	
φ	Volume fraction of polymer in the polymer particles	
ϵ	Ratio between termination by disproportionation and the total termination	
ε	Slack variable	
χ	Flory-Huggins interaction parameter	
χ_f	Terminal constraints	
θ	Collection of process parameters	
ν	Internal control signal	
ν_k	k'th dead moment	mol/m^3
$\nu^{\max}$	Maximum value of internal control signal	
$\nu^{\min}$	Minimum value of internal control signal	
σ	The average rate radicals enters the particles from water phase in Li and Brook's prediction	
τ_{fresh}	Time constant for delay in pipe lines for fresh water feed	s
$\Delta H^{\text{cooling capacity}}$	Available (maximum) cooling capacity	W
$\Delta H^{\text{cooling demand}}$	Cooling demand	W
ΔH_{rx}	Heat of reaction	W

List of Figures

List of Tables

List of Abbreviations

CMC	Critical Micelle Concentration
CTA	Chain Transfer Agent
DAEs	Differential Algebraic Equations
EKF	Extended Kalman Filter
KF	Kalman Filter
LP	Linear Programming
MHE	Moving Horizon Estimation
MPC	Model Predictive Control
MWD	Molecular Weight Distribution
NLP	Nonlinear Programming
NMPC	Nonlinear Model Predictive Control
NTNU	Norwegian University of Science and Technology
ODE	Ordinary Differential Equation
ODEs	Ordinary Differential Equations
OPC	Open Platform Communication
PDI	Polydispersity Index
PID	Proportional-Integral-Derivative Controller
PVC	Poly-Vinyl Chloride
PVT	Pressure, Vapour and Temperature
QP	Quadratic Programming
RK2	2nd order Runge Kutta
RTO	Real-Time Optimizer
SQP	Sequential Quadratic Programming
SRC	Split Range Controller
VCM	Vinyl Chloride Monomer

Chapter 1

Introduction

This chapter aims to give the reader an introduction to this thesis. The background for this project will be presented by reviewing some of the historical aspects of polymer and polymer science. The present perspective of the field will then highlight the motivation of implementing model-based control schemes, such as Model Predictive Control (MPC), to polymerization processes. The objective of this project will then be presented. Finally, the structure of the remaining chapters of this thesis will be covered.

1.1 Background and motivation

The first polymers used were natural occurring products, like cotton, starch and wool (Sperling, 2005). Polymer technology was utilized in the developments of fibres, additives, coating and leathers. Little was, however, known about the chemistry of these materials. Before 1770, when Joseph Priestley discovered natural rubbers was well applicable for erasers, natural rubbers were not utilized extensively. Synthetic rubbers were not produced to any particular extent before the late 1930s (Seymour, 1989). The research within polymer science was created in need to make new kinds of plastics, fibres and coatings, in addition to making an understanding of the relationship between the chemical structure and the resulting physical properties (Sperling, 2005). Synthetic polymers can today be produced through a variety of different processes and mechanisms (Kiparissides, 1996).

During the 1940s, Harkins (1945, 1946, 1947), published a framework that qualitatively describes the emulsion polymerization process. His theory has later been treated by numerous of other workers, as by Smith and Ewart (1948). It has later become evident that the mechanism of emulsion polymerization is much more complex than theories of the early workers. In addition, these theories have shown trends not to be applicable for all emulsion polymerization systems (Okamura and Motoyama, 1962; Peggion et al., 1964; Gardon, 1968a,b).

A number of different mechanisms have been proposed, for example by Min and Ray (1974), who proposed a more general modeling framework. Also, J. Ugelstad performed a significant work on kinetics and mechanisms of emulsion polymerization (1967; 1976), particulary on the polymerization of Vinyl Chloride Monomer (VCM) (1969; 1970; 1973).

The field of emulsion polymerization is by this time, quite well-understood (Gilbert, 1995). However, the polymer industry becomes more and more competitive. This increases the pressure on reducing the production cost and more strict product quality requirements. It is, therefore, essential to make use of the available resources and the knowledge people possess. By developing mathematical models describing the process, the behaviour of the system can be predicted. The system's response to different reactor configurations and operational conditions can then be explored (Kiparissides, 1996). This does, however, require a modeller with a deep understanding of the problem. Model-based control schemes are attractive to ensure both safe and stable operations. It can also be beneficial when aiming to meet the quality requirement, improve quality and reduce process variability, which can have significant economic benefits (Valappil and Georgakis, 2002). Model-based control schemes, such as MPC, can also be used to reduce emissions and waste, which can be advantageous both from an economic and environmental point of view.

Significant improvement can be obtained regarding plant operation and economics by implementing MPC. The dynamics of polymerization processes are highly nonlinear, and some, important, property measurements are not avail-

able during operation (Özkan et al., 2001). Other environmental measurements, however, like temperature, can readily be obtained (Kiparissides, 1996; Mutha et al., 1997). A model that predicts the behaviour of the quality parameters from the available measurements is, therefore, necessary. A challenge in on-line optimization is to develop a model with sufficient complexity to capture the dynamics of the system. The model should, however, not be too complicated for the solver not to find a solution in between samples. The optimization problem should be solved within the available sampling time (Rawlings et al., 2017).

1.2 Scope of work

From an industrial point of view, an interesting aspect would be to explore the possibility of shortening the duration of one batch. By exploiting the cooling capacity more efficiently, the production rate can be increased and thus decreasing the batch duration. As long as reducing the batch time does not compromise the quality of the product, the batch time will be the most critical economic variable in many systems. Dosing chemicals in an appropriate matter might also be a candidate for increasing the efficiency of the operation. The objective of this project is to investigate and implement Nonlinear Model Predictive Control (NMPC) of a semi-batch reactor for an emulsion polymerization process, for the purpose of the above-mentioned challenges.

In advance of this thesis, a preliminary project was conducted. This project included developing a dynamic model for an industrial polymerization process. More specifically, an emulsion polymerization process carried out in a semi-batch reactor. The model was validated against an industrial case, by data provided by Inovyn Norge. The results showed promising results for further implementation of NMPC.

This master project has consisted of a wide range of various tasks. Some important modelling aspects remained after the preliminary work to make the model applicable for NMPC. The first task of this project was to finalize the model and validate the model against process data. An environment for control system testing was then to be established, in which the model served as a plant replacement simulator, and as an integrated part of the controller. Schemes for state- and parameter estimation was then to be investigated, and assessed if it provides more efficient control. This is of particular interest for systems where the models in the plant replacement simulator and integrated into the controller, are dissimilar. The selected solution was to be implemented in Cybernetica's on-line environment, and the performance of the control system should be tested.

1.3 Thesis structure

This remaining chapters of this thesis will be structured in the following way.

Chapter 2 will introduce the theoretical aspects of polymerization processes. This will be done first on a general basis, before free-radical polymerization will be reviewed. This will together found the basis for the final, and most relevant topic for the purpose of this project; emulsion polymerization processes.

Chapter 3 will introduce the theoretical aspects of optimization, linear and nonlinear MPC, and state- and parameter estimation. The general theory of optimization problems will be stated first. The concept of MPC will then be presented, whereas the need for state- and parameter estimation will become clear.

Chapter 4 will give a description of the relevant process and the control structure of the system. Necessary information regarding reactor designed and chemical dosage will be reviewed to provide the reader with a proper understanding of the relevant process.

Chapter 5 provides the finalization of the set of model equation which will be used when implementing the NMPC application. There was done a thoroughly modelling work in the preliminary project, which is summarized in Appendix A, and Chapter 5 merely completes this work.

Chapter 6 introduces some key considerations when developing a control system for the relevant process. It will provide the reader with a comprehensive understanding of the addressed problem, and highlight the motivation for implementation of NMPC.

Chapter 7 presents the results of a selected number of simulations, which is considered of particular relevancy. They are then discussed consecutively.

Chapter 8 will give a conclusion based on the previous chapters. Finally, some recommendations and remarks on further work have been made.

Chapter 2

Theoretical aspects of polymerization

This section will give the reader an introduction to the theory behind polymerization processes. This is important when introducing the relevant process and the established model, in addition to the implementation of the control scheme. Polymerization processes will be described on a general basis, with a focus on emulsion polymerization specifically.

There was done thoroughly researched regarding the theory of polymerization during the preliminary project. This to be able to establish the model for the process (Chapter 5 and Appendix A). The theoretical basis will be presented here.

2.1 Polymerization processes

Polymers are macromolecules built up of smaller molecule units, called *monomers* (Odian, 1991). Polymers may consist of more than thousands of monomer units. Hence polymers are typically high molecular weight compounds. For monomers to form polymers, they have to have either reactive functional groups, double bonds or triple bonds. The reaction of generating polymers is called *polymerization* and is commonly a highly exothermic reaction (Kumar, 2003).

The properties of the polymer may be highly dependent on the arrangement of the polymer chains, called the *microstructure*. Polymers can be divided into three different classes, based on their structure (Painter, 1997);

1. Linear polymers

2. Branched polymers

3. Network polymers

Linear polymers are arranged in a linear sequence. The simplest type of polymer is *linear homopolymer* and is made up of one type of monomer (Painter, 1997). In comparison, polymers built up of two or more types of monomers are referred to as *copolymers*. *Branched polymers* may contain up to several secondary chains, either long or short, connected to the primary chain (Asua, 2008).

When linear chains pack into a three-dimensional form, they form a *crystalline phase*. Properties such as stiffness, strength and optical clarity are affected by the crystallinity of the three-dimensional polymer. Highly branched polymers are incapable of forming such a crystalline phase, as their branches prevent them of regular package. The density of linear polymers may, therefore, be higher than the equivalent branched polymer (Painter, 1997).

A network is defined to be a polymer where every unit is interconnected through some pathway. The network remains mixed with the monomer unless all the monomer units are connected to the network. A network can be formed in various ways, as by cross-linkage of linear polymers or by the formation of highly branched polymers (Painter, 1997). When a critical number of polymers has undergone cross-linkage or branching, polymers that initially was soluble might become insoluble (Flory, 1941). This phenomena is called *gelation*. A *gel* is defined to be the part of a polymer belonging to the infinite three-dimensional network (Asua, 2008).

There are multiple ways of classification within polymer science. One is the classification based on the mechanism of the polymerization reaction. In this case, the polymerization reactions are distinguished between *step-growth* and *chain-growth* polymerization (Odian, 1991).

The reaction mechanism of step-growth was introduced by Flory (1946), by the following scheme

$$\mathrm{x{-}mer} + \mathrm{y{-}mer} \longrightarrow (\mathrm{x} + \mathrm{y}){-}\mathrm{mer} \tag{2.1.1}$$

Here, x and y represent the chain length of the mer, and can, for example, be monomer, dimer and trimer. Step-growth polymerization consists of a reaction between the functional groups, where the reaction between monomers or polymers generates polymers with a longer chain length. This results in a slow-building of chains. All chains will be active throughout the batch, and the long-chained polymers will only be formed at the end.

In comparison to step-growth polymerization, chain-growth polymerization will show the presence of high-molecular-weight polymer chains at any given conversion, not only towards the end. For chain-growth polymerization, an initiator is used to generate a specie with a reactive centre. This reactive centre could be a free-radical, anion or cation. The polymerization reaction occurs by the propagation of the reactive centre by addition of monomer by a chain reaction. One of the most common example of a chain-growth reaction is the polymerization of vinyl monomers (Odian, 1991). Vinyl Chloride Monomer (VCM) is an example of such a monomer, and its corresponding polymer is Poly-Vinyl Chloride (PVC). PVC is an example of a linear homopolymer, and can be described by the following chemical formula.

$$[-\mathrm{C_2H_3Cl}-]_i \tag{2.1.2}$$

Here, i represents the number of repeated monomer units. Chain-growth polymerization consist mainly of the following features; *initiation*, *propagation*, *termination* and *chain transfer*. This will be described in detail in Section 2.2.

2.1.1 Molecular Weight Distribution

The molecular weight of a polymer can determine many of its important physical properties. A collection of polymer chains may have a large range of lengths, and both an average chain length and the Molecular Weight Distribution (MWD) can be useful when examining the physical properties. The MWD describes the number of chains with a particular molecular weight, and can also give information about which kinetic phenomena that is present in a system. The MWDs can be modelled, and some properties can also be measured. This opens up the opportunity to optimize properties of the polymer, which is affected by the MWD (Gilbert, 1995).

The properties of the polymer are affected by the whole MWD, but it is often described using average molecular weights. The *number average molecular weight*, $\overline{M_\mathrm{n}}$, and the *weight average molecular weight*, $\overline{M_\mathrm{w}}$, are described by Equation 2.1.3 and 2.1.4, respectively.

$$\overline{M_\mathrm{n}} = \frac{\sum i(D_i + P_i)}{\sum (D_i + P_i)} M_\mathrm{M} \tag{2.1.3}$$

$$\overline{M_\mathrm{w}} = \frac{\sum i^2(D_i + P_i)}{\sum i(D_i + P_i)} M_\mathrm{M} \tag{2.1.4}$$

Here, i represent the number of repeated monomer units, D_i represent the number of inactive chains of length i, P_i represent the number of active chains

of length i, and M_{M} represent the molecular weight of the monomer. Calculation of these molecular weights are elaborated in Appendix A.4. The ratio between the weight and number average molecular weight is defined to be the Polydispersity Index (PDI) of the MWD. The PDI is therefore described by Equation 2.1.5 (Asua, 2008).

$$\mathrm{PDI} = \frac{\overline{M_{\mathrm{w}}}}{\overline{M_{\mathrm{n}}}} \tag{2.1.5}$$

For a system where all chains are of the same length, the number and weight average molecular weight will be equal. Thus the PDI will be 1. In reality, there will always be some deviation between the two, and the weight average molecular weight will be larger than the number average. The PDI will be larger than 1.

$$\mathrm{PDI} > 1 \tag{2.1.6}$$

2.2 Free-radical polymerization

Free-radical polymerization is a type of chain-growth polymerization. As a consequence, individual polymer chains can grow very rapidly and will eventually terminate. New, active polymers are continuously generated throughout the process by initiation. The system will contain both living and dead radicals of different chain lengths at any given time, due to continuous initiation, propagation, termination and chain transfer (Gilbert, 1995). The steps of free-radical polymerization will now be presented.

2.2.1 Reaction mechanisms

Initiation

Initiation is necessary for the polymerization to start, as it is the first reaction step in the free-radical polymerization process and generates radicals. Chemical compounds that through reactions yield radicals are referred to as initiators. Initiation consist of two main steps; *initiator decomposition* and *chain initiation*. Radicals are formed through the initiator decomposition step. They are highly reactive species, and if there is monomer present, they react rapidly by chain initiation. The decomposition of initiator is a much slower reaction step than the chain initiation, and will, therefore, be the rate-determining step in the initiation mechanism. The choice of initiators and which characteristics they possess is, therefore, highly important in free-radical polymerization processes (Mishra and Yagci, 2016).

There are several methods of initiation, such as thermal initiation and reduction-oxidation initiating systems (Redox initiation). In thermal initiation, thermal energy will cause bond dissociation, leading to radical fragments. The activation energy of initiators is important, as they should be stable at room temperature, but highly reactive at certain temperature rise. Peroxides

are examples of chemical compounds that meet this requirement, and radical fragments are formed by cleavage of the O−O-bond. For thermal initiation systems, initiator decomposition is most commonly described by Equation 2.2.1, where one initiator molecule yields two radicals by homolytic decomposition (Matyjaszewski and Davis, 2003).

$$\mathrm{I} \xrightarrow{k_\mathrm{d}} 2\mathrm{R}\cdot \tag{2.2.1}$$

Here, I represent the initiator, k_d the rate constant for decomposition of the initiator, and $\mathrm{R}\cdot$ represent a radical fragment. $\mathrm{R}\cdot$ will, in this thesis, represent a radical, of chain length zero, generated from any arbitrary initiation mechanism. The rate of initiator decomposition can be described by the following equation

$$R_\mathrm{d} = k_\mathrm{d}[\mathrm{I}] \tag{2.2.2}$$

where R_d represent the reaction rate of initiator decomposition, and [I] represent the concentration of initiator. Not all radicals that are formed by initiator decomposition can take part in further polymerization. They may terminate and react further to inactive molecules. An initiator efficiency factor, f, is therefore introduced. This parameter will have a value between 0 and 1. The rate of radical formation that can be used for polymerization can then be described by Equation 2.2.3.

$$R_\mathrm{I} = 2fk_\mathrm{d}[\mathrm{I}] \tag{2.2.3}$$

Here, R_I represent the rate of radical formation that can be used for chain initiation. Since one initiator molecule will give rise to two radical fragments, the equation is multiplied by two (Asua, 2008).

Redox systems consist of a reducing agent and an oxidizing agent. It is commonly used in industry because of their low activation energy. Redox systems can, therefore, be performed at a lower temperature, that is under 50°C. The typical redox initiation mechanism is shown below (Mishra and Yagci, 2016).

$$\mathrm{ROOH} + \mathrm{Mt}^\mathrm{n} \longrightarrow \mathrm{RO}\cdot + \mathrm{Mt}^{\mathrm{n}+1} + \mathrm{OH}^- \tag{2.2.4}$$

Here, ROOH represent a hydroperoxide, Mt represent the metal compound, $\mathrm{RO}\cdot$ represent a alkoxyl radical, and OH^- the hydroxide. The alkoxyl radical will be the specie that will dominantly further take part in the polymerization. The metal compound will form a complex with additives of the reaction mixture, and cycle up and down in valency, and act as a catalyst of the reaction (Matyjaszewski and Davis, 2003).

The chain initiation step can be described by Equation 2.2.5.

$$\mathrm{R}\cdot + \mathrm{M} \xrightarrow{k_\mathrm{i}} \mathrm{P}_1 \tag{2.2.5}$$

Here, M represent the monomer, k_i the rate constant for chain initiation, and P_1 represent a growing polymer chain of length one. The rate of chain initiation

will be given by the following equation

$$R_{\mathrm{i}} = k_{\mathrm{i}}[\mathrm{M}][\mathrm{R}\cdot] \tag{2.2.6}$$

where R_{i} represent the rate of chain initiation, [M] represent the concentration of monomer and $[\mathrm{R}\cdot]$ represent the concentration of radicals with no attached monomer unit (Mishra and Yagci, 2016).

Propagation

Propagation consist of adding monomer units to a growing polymer chain, and is described by the following reaction mechanism for homo-polymerization reactions

$$\mathrm{P_i} + \mathrm{M} \xrightarrow{k_{\mathrm{p}}} \mathrm{P_{i+1}} \tag{2.2.7}$$

where k_{p} represent the rate constant for propagation, and $\mathrm{P_i}$ and $\mathrm{P_{i+1}}$ represent a growing polymer chain of length i and i+1, respectively (Butté et al., 2002). It is here assumed that the rate constant for propagation is independent of chain length. There have, however, been done work that indicates that k_{p} has a significant chain-length dependence for the first addition steps, as by Heuts and Russell (2006).

When assuming chain initiation is negligible contribution to the overall consumption of monomer, the production rate will be given as the propagation rate. This is described by Equation 2.2.8.

$$R_{\mathrm{p}} = k_{\mathrm{p}}[\mathrm{M}]P_{\mathrm{tot}} \tag{2.2.8}$$

Here, [M] represent the concentration of monomer, and P_{tot} the total amount of radicals (Asua, 2008).

Termination

Termination involves a reaction between two radicals, leading to inactive polymer chains, unable to undergo further polymerization. The rate of termination is determined by how often, how close two growing polymers approach each other and how long they stay close. Due to this diffusion dependency, the rate of termination can vary as a function of the viscosity of the reaction mixture, which may change during the batch time. This introduces the aspects of chain-length dependency termination, or alternatively conversion dependency (Stevenson, 1986). This is a complex feature and is, therefore, often neglected in terms of modelling.

Termination can occur by two modes; combination or disproportionation. The reaction mechanisms are shown in Equation 2.2.9 for combination, and 2.2.10 for disproportionation.

$$\mathrm{P_i} + \mathrm{P_j} \xrightarrow{k_{\mathrm{tc}}} \mathrm{D_{i+j}} \tag{2.2.9}$$

$$\mathrm{P_i} + \mathrm{P_j} \xrightarrow{k_{\mathrm{td}}} \mathrm{D_i} + \mathrm{D_j} \tag{2.2.10}$$

Here, P_i and P_j represent a growing polymer chain i and j respectively, and D_i, D_j and D_{i+j} represent a dead polymer chain of length i, j and i+j respectively. k_{tc} represent the rate constant for termination by combination, and k_{td} represent the rate constant for termination by disproportionation. The total termination rate constant will be given by the sum of the reaction rates of the two modes, expressed by k_t, in Equation 2.2.11.

$$k_t = k_{tc} + k_{td} \tag{2.2.11}$$

The total reaction rate of termination will then be given by R_t in Equation 2.2.12.

$$R_t = \frac{k_t {P_{tot}}^2}{V} \tag{2.2.12}$$

where V represent the volume of which the termination takes place (Painter, 1997; Butté et al., 2002).

Which one of the modes that are dominant vary from one emulsion system to another. For instance, Abdel-Alim and Hamielec (1972) reported that termination by disproportionation probably is the most dominant mode for emulsion polymerization of PVC.

Chain transfer

Chain transfer is a reaction where a growing polymer chain is terminated, and a new chain is initiated in its place. These reactions have commonly little effect on the rate of polymerization but can have the ability to affect the MWD. This is because they change the size of the chain lengths produced. Many compounds can take part in chain transfer reactions, such as monomer, Chain Transfer Agent (CTA), polymer, solvents and initiator. CTAs are often added to the reaction mixture to alter the MWD deliberately (Painter, 1997). Equation 2.2.14, 2.2.13 and 2.2.15 shows the reaction mechanism for chain transfer to CTA, monomer and polymer, respectively.

$$\begin{aligned} P_i + AX &\xrightarrow{k_{ct}^{AX}} D_iX + A\cdot \\ A\cdot + M &\xrightarrow{k_i^A} P_1 \end{aligned} \tag{2.2.13}$$

$$P_i + M \xrightarrow{k_{ct}^M} D_i + P_1 \tag{2.2.14}$$

$$P_i + D_j \xrightarrow{k_{ct}^P} D_i + P_j \tag{2.2.15}$$

Here, P_i, P_j and P_1 is growing polymer chains of length i, j and one, respectively. AX represent the CTA, and $A\cdot$ the radical fragment from the CTA. D_i and D_j is dead polymers of chain length i and j, respectively. k_{ct}^{AX} represent the rate constant for chain transfer to CTA, and k_i^A represent the rate constant for chain initiation by the radical fragment from the CTA. k_{ct}^M and k_{ct}^P represent the

rate constant for chain transfer to monomer and polymer, respectively (Butté et al., 2002; Mishra and Yagci, 2016).

2.3 Emulsion polymerization

Emulsion polymerization is a type of free-radical polymerization. It is usually carried out in batch or semi-batch reactors for commercial use, and almost complete conversion can be obtained (Matyjaszewski and Davis, 2003).

The basic principle of emulsion polymerization is to disperse a sparingly water-soluble monomer in the continuous water phase. The recipe for emulsion polymerization systems consists of water, a monomer which is sparingly soluble in water, surfactant and a water-soluble initiator generating radicals (Braun et al., 2012).

The system consists of three or four different phases, depending on the course of the reaction. The four different phases are; water phase, polymer phase, gas phase and free phase. The water phase consists of water, surfactants, initiator, and radicals generated from initiator degradation. Surfactants will form micelles above Critical Micelle Concentration (CMC). These micelles will then be swelled by monomer. A fraction of these micelles will be transformed into polymer particles. The polymer phase consists of monomer, and growing polymer chains as monomer are consumed by propagation. In the gas phase, the monomer will occupy the largest fraction due to its volatility, in addition to some water and inert. The composition of the gas phase will vary throughout the batch, depending on the amount of monomer in the system. The so-called free phase consists of monomer droplets and acts as a reservoir for the polymerization reaction (Chern, 2006).

Emulsion polymerization is a complex process, with multiple phases and simultaneous reactions. A schematic illustration developed by Morrison et al. (1994) is shown in Figure 2.1. The figure shows the many outcomes of the species in an emulsion polymerization process. The initiator generates radicals in the water phase. Here, the radical may propagate or terminate, by reaction mechanisms described in Section 2.2.1. After some steps of aqueous propagation, the growing polymer radical may enter the polymer particles. Here, it might propagate further, undergo termination or chain transfer. Short-length radicals generated from chain transfer mechanism can exit from the particles. It may now undergo the same reaction mechanism as before.

Due to its complexity, emulsion polymerization processes may be hard to understand, model and control. The process, its course of reactions, phenomena and how it distinguishes from bulk polymerization will be described in more detail in the upcoming subchapters.

2.3.1 Development of the process during the batch

The emulsion polymerization process is often divided into three different intervals, all of which has its own characteristics.

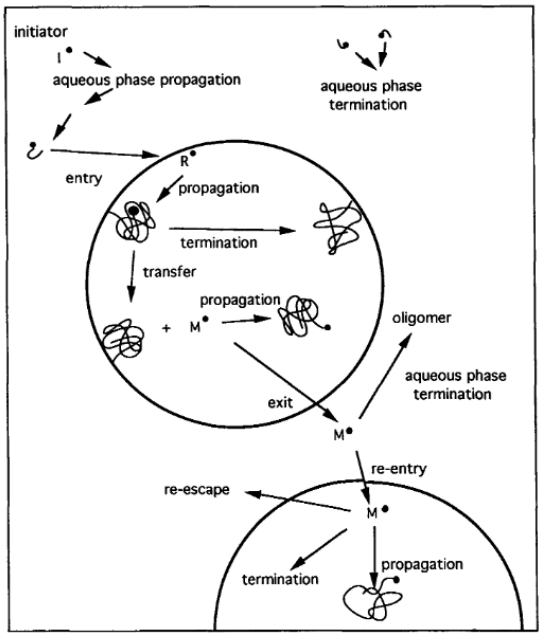


Figure 2.1: Schematic illustration of free-radical polymerization in emulsion polymerization processes (Morrison et al., 1994).

- Interval I consists of particle nucleation
- Interval II consists of particle growth
- Interval III the residual monomer is consumed, and the batch reaches towards its end

The evolution of the batch is illustrated in Figure 2.2. This division shall prove to be important for modelling purposes (Chapter 5 and Appendix A) and can be recognized when evaluating process measurements data as reactor pressure.

Interval I: Particle nucleation

The surfactants have one hydrophilic and one hydrophobic part, and above CMC, they form micelles (Hiemenz, 1997). Monomer diffuses from the monomer droplets and the aqueous phase, into the micelles and swell them. The micelles dissolve the monomer, in a process called solubilization (Harkins, 1947, 1950).

Adsorption is a critical phenomenon in emulsion polymerization and is followed by the entry of radicals into the micelles or polymer particles. Radicals may also enter monomer droplets, but the surface area of micelles is much larger than those of droplets, and it is, therefore, more likely for the radicals to enter the micelles. In emulsion polymerization, initiators are mostly water-soluble. Radicals are, therefore, formed in the water phase, and the chain initiation step

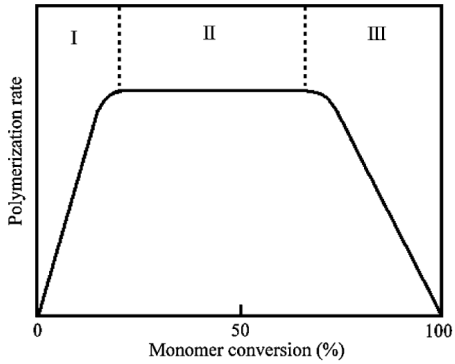


Figure 2.2: Illustration by Chern (2006) of the polymerization rate as function of conversion during the different intervals of emulsion polymerization.

will also take place in this phase (Matyjaszewski and Davis, 2003; Mishra and Yagci, 2016). The product of initiator decomposition is often hydrophilic, and will, therefore, not enter the organic polymer phase. They will instead undergo a few steps of propagation with monomer dissolved in the water phase, forming oligomers. As monomer is added to the growing polymer chain, the polymer chains become more and more water-insoluble. It will eventually reach a critical chain length, diffuse to the surface of the micelle, and enter it. As of the time a micelle contains a polymer chain, it will be referred to as a polymer particle. This is what is called particle formation by heterogeneous nucleation, or micellar nucleation. The growing chain may now propagate further inside the polymer particle (Asua, 2002).

It is important to notice that not all generated radicals will be able to undergo adsorption. This was briefly discussed in Section 2.2.1, and is due to bi-reactions in water-phase, making the radicals inactive and unable to undergo further polymerization. Another reason for the oligomers not to undergo adsorption would be that they propagate in water phase until they have become insoluble. They have then formed a new polymer particle by the collapse of the chains upon themselves and is called homogenous nucleation (Fitch, 1973; Hansen and Ugelstad, 1978).

The number of particles will increase during the first interval, as a growing number of micelles becomes polymer particles. Consequently, the polymerization rate will also be increasing in this interval, as shown in Figure 2.2. The particle nucleation may be very complex in emulsion polymerization systems and can vary dependent on operational conditions as the type of monomer(s), initiators, recipe or temperature. No general mechanism can capture all the different aspects of nucleation in a given system (Matyjaszewski and Davis, 2003).

The cease of nucleation marks the end of the rapid Interval I. At this point, 0.1-1% of the micelles will have become polymer particles. The remaining surfactants will now contribute to stabilizing the growing particles (Chern, 2006).

For emulsion polymerization of styrene, experiments indicates that Interval I ends at a conversion of 5-10% (De La Rosa et al., 1999), and 6-7% for VCM (Ugelstad et al., 1969).

Interval II: Particle growth and point of inversion

Interval II is characterized by particle growth and is the longest interval. All four phases are present; polymer phase, free phase, water phase and gas phase. During this interval, the number of particles and the concentration of monomer inside the particles remains approximately constant. Consequently, so will the polymerization rate (Gilbert, 1995; Matyjaszewski and Davis, 2003). This is shown in Figure 2.2. During the polymerization process, monomer from the monomer droplets diffuse into the particles as monomer inside the particles is consumed (Harkins, 1947).

The growing chain may also terminate by bimolecular reactions inside the particle, or desorb out of the particle and into the water phase again (Asua, 2008). The end of Interval II is characterized by the cease of monomer in free phase (Gilbert, 1995). This is also referred to as the inversion point.

Interval III: Consumption of residual monomer

Interval III is characterized by the consumption of residual monomer (Chern, 2006). As the free phase has ceased to exist, there will only be three phases in this interval; polymer phase, water phase and gas phase. As the residual monomer is consumed, the concentration of monomer in polymer, gas and water phase will decrease. Due to the decreasing concentration of monomer, the polymerization rate cannot be maintained, and will decrease, as shown in Figure 2.2. The reaction will continue until the monomer is fully consumed, or the reaction is stopped for other limiting reasons. Due to monomer being transported from gas phase to liquid phase, the inversion point is also characterized by a pressure drop. As a result, the temperature in the gas phase will also experience a decrease.

For many systems, the particle volume will decrease slightly during Interval III. This is due to the contraction by polymerization, and the residual monomer is consumed. The decrease in monomer concentration also leads to an increased viscosity within the particles, so the termination rate constant will, as a result, decrease. This decrease in k_t for increasing conversions, is known as the *gel effect* (Ugelstad and Hansen, 1976).

2.3.2 Radical distribution

Describing the whereabouts of the radicals is important to describe other phenomena, as propagation. Smith and Ewart (1948) presented a theory based on Harkins' work, a quantitative description of the the emulsion polymerization process. Smith and Ewart formulated a balance describing the number

of particles N_n containing n radicals. The following infinite set of Ordinary Differential Equations (ODEs) were obtained

$$\begin{aligned}\frac{dN_n}{dt} = \rho'\Big(N_{n-1} - N_n\Big) + k'\Big((n+1)N_{n+1} - nN_n\Big) \\ +c\Big((n+2)(n+1)N_{n+2} - n(n-1)N_n\Big)\end{aligned} \tag{2.3.1}$$

Here, $n = [0, 1, ..., \infty]$, ρ', k' and c is the rate coefficient for adsorption, desorption and termination respectively. Several people have reported different ways of solving these equations, as Stockmayer (1957), O'toole (1965) and Hawkett et al. (1977). However, Smith and Ewart presented three different limiting cases when determining the number of radicals per particle. These are presented below

Case 1: $\bar{n} \ll 0.5$ In this case, the probability of desorption is so high, that at any time only a small number of particles will contain a free radical. As a result, the number of radicals will be much smaller than unity (Smith and Ewart, 1948) These systems are characterized by small particles, water-soluble monomers and a large number of particles (Asua, 2008). As will be described in Section 2.3.5, desorption of radicals into water phase can only occur if short polymer chains are generated inside the particles. That is, chain transfer mechanisms generating these short-length polymers is dominant in these systems. Ugelstad et al. (1969) reported that this is the case for the polymerization of PVC, where the number of radicals per particle was calculated to be within the range of 0.0005 to 0.1.

Case 2: $\bar{n} = 0.5$ This situation occurs if the probability of desorption is low, and adsorption and termination will be dominating. The probability of mutual termination of two radicals in the same particle is large enough so that the time frame the particle contains two radicals is negligible compared to the time it contains one. The particle will then either contain zero or one radical and in average contain 0.5 radicals per particle (Smith and Ewart, 1948; Matyjaszewski and Davis, 2003)

Case 3: $\bar{n} \gg 0.5$ For the case of a number of radicals per particle to be higher than 0.5, the adsorption of radicals has to dominant, and termination is no longer instantaneous open entry (Matyjaszewski and Davis, 2003).

These systems typically contain large particles and high concentration of initiators (Asua, 2008).

An average number of radicals per particle can be calculated by the following equation.

$$\bar{n} = \frac{\sum_{n=0}^{\infty} nN_n}{\sum_{n=0}^{\infty} N_n} \tag{2.3.2}$$

Here, $\bar{n}$ represent the average number of radicals per particle, N_n represent the number of particles containing n radicals. This calculation requires solving the system of ODEs presented in Equation 2.3.1, and estimation of the unknown parameters. Methods for calculation $\bar{n}$ have been explored, for example by Li and Brooks (1993), in which the reader is advised to Appendix A.3.3.

Systems that satisfy either Case 1 and 2 are often referred to as a *zero-one system*. In these cases, the probability of the particle containing more than one radical is much less than containing zero and one.

2.3.3 Number of particles

Smith and Ewart stated that particles are nucleated by radicals entering the micelles and that monomer-swollen polymer particles are the locus of propagation. This is some of the assumptions that build the background for the theory. Furthermore, the particle nucleation will stop when the concentration of surfactant in the water phase is below CMC. The decreasing concentration of surfactant in the water phase is described to be due to surfactants adsorbing onto newly formed, growing, particles. Eventually, the micelles will cease to exist, and no new particles can be formed through heterogeneous nucleation. The following equation was obtained by Smith and Ewart for calculating the number of particles nucleated.

$$N_{\mathrm{T}} = k''\left(\frac{\rho''}{\mu}\right)^{0.4}\left(a_{\mathrm{s}}S\right)^{0.6} \tag{2.3.3}$$

Here, N_{T} represent the number of particles, k'' a parameter within the range of $0.37 - 0.53$, ρ'' the rate of radical formation per volume of water, μ the rate of particle growth in volume units, a_{s} the interfacial area of the particles, and S the amount of surfactant. The limits of the k''-parameter is derived from two idealized situations, and the reader is encouraged to read Smith and Ewart (1948) work for an elaboration.

This theory has later proven not to apply to all systems, for example by Peggion et al. (1964) for polymerization of PVC. Fitch (1973) proposed a model describing the initiation of free-radicals and growth in the continuous medium would be more likely. This could be followed by either homogeneous nucleation or capture by colloidal particles already present or previously formed.

Smith (1948) suggested that the case of emulsion polymerization of styrene was an example of Case 2 described in Section 2.3.2. However, Hansen and Ugelstad (1979) proposed that high desorptions rates cause the deviations in

the calculation of the number of particles in the system. Even for the system of emulsion polymerization of styrene, there will be a small effect of desorption of radicals, leading to an average number of radicals of particle slightly below 0.5. The desorbed radicals may also take part in particle formation, which leads to a higher number of particles. If chain transfer to monomer or CTA is eminent, this can reinforce this effect, as it leads to the formation of short-length polymer chains, that more easily can desorb.

The work of Harkins (1945, 1946, 1947) and Smith and Ewart (1948) is considered to be pioneer work within emulsion polymerization. In light of subsequent work, some statements are now seen to be incorrect. Others remain to be the common understanding of the process. Most of the subsequent work are based upon these theories.

Seeded systems

Emulsion polymerization processes are complex, and each interval has its own characteristics. To understand the different intervals, it would be beneficial to investigate them separately. The nucleation stage is particularly challenging to describe and model, which introduce to the concept of seeded systems. In such systems are the particles pre-formed, and the number of particles are known and will not variate. The polymerization process can then start directly in Interval II, and the irreproducible of nucleation formation from one batch to another is overcome (Gilbert, 1995; Asua, 2008).

2.3.4 Polymerization rate

The polymerization rate in an emulsion polymerization process can be described similar to other free-radical polymerization processes, by Equation 2.2.8. By assuming that most of the polymerization reactions take place inside the particles, the following expression can be obtained.

$$R_{\mathrm{p}} = k_{\mathrm{p}}[\mathrm{M}]^{\mathrm{p}} P_{\mathrm{tot}} \tag{2.3.4}$$

Here, $[\mathrm{M}]^{\mathrm{p}}$ represent the concentration of monomer inside the particles, which is determined thermodynamically and assumed to be the same in every particle (Ugelstad et al., 1969; Gilbert, 1995). P_{tot} is, as earlier mentioned, the total amount of radicals in the system. For emulsion polymerization, the relevant radicals will be placed inside the particles. This parameter may change during the batch, and the number of radicals inside a particle can vary from one particle to another. It is, therefore, often necessary, and also more practical, to simplify the system by calculating an average number of radicals inside the particles, $\bar{n}$. P_{tot} can then be approximated by Equation 2.3.5.

$$P_{\mathrm{tot}} = \bar{n} N_{\mathrm{T}} \tag{2.3.5}$$

where N_T represent the total amount of particles in the system. Finally, the polymerization rate can be described by Equation 2.3.6.

$$R_p = k_p[M]^p \bar{n} N_T \tag{2.3.6}$$

Note that the units of N_T is moles of particles. This parameter is in some literature given in number of particles. In that case, the polymerization rate will be given by the following equation

$$R_p = k_p[M]^p \frac{\bar{n} N_T}{N_A} \tag{2.3.7}$$

where N_A is Avogadro's constant.

2.3.5 Other phenomena in emulsion polymerization

Desorption

As already mentioned, radicals may undergo desorption from the polymer radicals, and diffuse into the water phase. This will decrease the average number of radicals per particle, and also increase the number of radicals in the water phase. The radical distribution was discussed theoretically in detail in Section 2.3.2, and for modelling purposes the reader is advised to read Appendix A.3.3.

A growing radical contained in the particle above the critical chain length is most likely not to go into the water phase again as it will have high free energy in the organic environment. The only possibility for desorption to occur is the formation of radicals below the critical chain length inside the particles. The only possible method for this is through chain transfer reaction inside the particles. Chain transfer gives rise to growing polymer radicals of a sufficiently short length for desorption. Also, after the chain transfer has occurred, the short-length radical must diffuse through the interior of the particle without undergoing too many propagation steps to again make it water-insoluble (Gilbert, 1995).

Radicals that have undergone desorption can have many fates, as was illustrated by Figure 2.1 (Morrison et al., 1994). It may enter other particles they may or may not already contain a growing polymer chain. Another possibility is that it will undergo termination in the water phase. It may also propagate further in the water phase until it becomes insoluble and thereby form a new polymer particle by homogeneous nucleation (Gilbert, 1995).

Compartmentalization

Segregation of growing polymer chains inside the polymer particles is called *compartmentalization*, and is characteristic for emulsion polymerization (Butté et al., 2002). In bulk polymerization, growing chains have equal access to one another, and can, therefore, easily terminate. In emulsion polymerization, however, only growing particles contained inside the same particle can react with each other and terminate. This significantly reduces the termination rate,

as polymer chains mostly are contained in separate particles. As a result, the termination rate in emulsion polymerization is experienced to be much lower than those of bulk polymerization. As the termination rate is reduced, the lifetime of a growing particle will increase. The polymerization rate is also found to be higher in emulsion polymerization, and molecular weights are found to be higher, most likely as a result of the effect of compartmentalization (Gilbert, 1995).

Since radicals contained in different particles cannot terminate by the bimolecular reaction, the overall radical concentration is much higher in emulsion polymerization than those of bulk polymerization. Consequently, the polymerization rate is observed to be higher for emulsion polymerization. As the number of particles in a system increases, the concentration of radicals will increase, and participate in increasing polymerization rate. The frequency of radical adsorption decreases with an increasing number of particles. Therefore, by increasing the number of particles, a second radical would be less likely to enter and contribute to bimolecular termination, resulting in higher molecular weights. To summarize; increasing the number of particles will contribute to increasing the polymerization rate, and increasing molecular weights (Asua, 2004).

Chapter 3

Theoretical aspects of optimization, MPC and estimation

This chapter will give the reader an introduction to optimization problems, Model Predictive Control (MPC), and state- and parameter estimation. Firstly, the theoretical background of optimization problems will be presented, which is essential when later introducing concepts of MPC. MPC will bind together the concepts of optimization and control. The need for state- and parameter estimation will become evident when discussing MPC, and an introduction to some well-known approaches will be given.

The theory provided in this section, in addition to the theory of polymerization process (Chapter 2), will prove essential when later discussing the relevant case; control of semi-batch reactor for emulsion polymerization (Chapter 6).

3.1 The optimization problem

The material in this sections is adapted from Hovd (2004), Nocedal (2006), Foss and Heirung (2013) and Rawlings et al. (2017).

Optimization is widely used within many fields, as engineering, science, economics and in industry. Solving optimization problems can be used in a large variety of applications, for example, controlling a chemical process to optimize the performance and minimizing emissions.

When solving an optimization problem, an *objective function* must be constructed. The objective function is a measure of the performance of the system, and could, for example, be profit, time, energy consumption or a combination of such quantities. This function depend on some variables, often called *decision variables*, which may be limited to some *constraines*. The goal of the optimization problem is to find the values of these variables which minimizes (or maximizes) the objective.

A collection of different optimization algorithms are available to solve an optimization problem. An optimization problem can be expressed using the following notation

$$\min_{\phi \in R^n} J(\phi) \tag{3.1.1a}$$

subject to

$$c_i(\phi) = 0, \quad i \in \mathcal{E} \tag{3.1.1b}$$

$$c_i(\phi) \geq 0, \quad i \in \mathcal{I} \tag{3.1.1c}$$

Here, $f(\phi)$ represent the objective function, ϕ is the set of decision variables, and $\mathcal{E}$ and $\mathcal{I}$ are indices for equality and inequality constrains, respectively.

When the objective function and the constrains all are linear functions, the problem is called a Linear Programming (LP) problem. Nonlinear Programming (NLP) problems, on the other hand, will consist of at least one nonlinear constraint or objective function. When facing nonlinear optimization problems, NLP solvers are needed. There are two main classes of such solvers, Sequential Quadratic Programming (SQP) and interior point methods. SQP is one of the most effective methods for nonlinear constrained optimization. A SQP problem consist of a sequence of Quadratic Programming (QP) problems, which are approximations to the nonlinear model. A QP problem consist of a quadratic objective function and all linear constraints (Hovd, 2004).

There are many solving methods for optimization problems, and many of them are iterative. They require an initial point, in addition to a termination criteria. Some algorithms require feasible initial points, while others, for example, the SQP method, does not (Foss and Heirung, 2013).

3.1.1 Dynamic systems

Dynamic systems are characterized by that they change over time. Such systems can be optimized in two different ways; by *quasi-dynamic optimization*

or *dynamic optimization*. Quasi-dynamic optimization optimizes a system by repetitively optimizing a static model. Dynamic optimization, on the other hand, optimizes a dynamic model. The latter approach is often necessary when the dynamics of a system plays an important role, and the operation conditions changes frequently (Foss and Heirung, 2013). Polymerization processes is an example of such a system.

Models of dynamic system may be represented in different ways, as by Ordinary Differential Equations (ODEs), Differential Algebraic Equations (DAEs) or transfer functions. Most nonlinear system descriptions derived from physical arguments are continuous time models, as shown in Equation 3.1.2.

$$\dot{x} = f(x, u) \tag{3.1.2a}$$

where

$$u \in R^{n_u} \tag{3.1.2b}$$

$$x \in R^{n_x} \tag{3.1.2c}$$

Here, $f(\,\cdot\,)$ represent the process model, n_u the number of inputs u, and n_x the number of states x (Rawlings et al., 2017).

Process measurements, however, are often sampled at discrete points in time, usually equidistant. The continuous time differential equation can then be replaced by a discrete time difference equation. Such system can be described in the following form

$$x_{k+1} = f(x_k, u_k) \tag{3.1.3a}$$

where

$$u_k \in R^{n_u} \tag{3.1.3b}$$

$$x_k \in R^{n_x} \tag{3.1.3c}$$

k denoted the sampling number, and is connected to time, t, by $k = t\Delta$, where Δ represent the sample time. The input u_k is constant within the interval $[k, k+1\rangle$, while the states only are defined at discrete points in time. A linear approximation of $f(\,\cdot\,)$ may be found by Taylor approximation, and will be given by Equation 3.1.4.

$$x_{k+1} = A_k x_k + B_k u_k \tag{3.1.4}$$

Here, A_k and B_k represent Jacobian matrices, and will be time-dependent.

For a dynamic optimization problem, the following objective function can be defined.

$$\begin{aligned} J(\phi) &= J(x_1, ..., x_k, u_0, ..., u_k) \\ &= \sum_{k=0}^{N-1} J_k(x_{k+1}, u_k) \end{aligned} \tag{3.1.5}$$

Here, N represent the prediction horizon, and $(x_1, ..., x_k, u_0, ..., u_k)$ represent the decision variables. The number of decision variables will increase linearly with the prediction horizon. The objective function sums up the contributions of each time step through J_k. x_0 is not a decision variable, and may be known or estimated. x_{k+1} is the state at the end of a time step, and is why u_k is paired with x_{k+1}.

Linear systems

Linear systems will first be reviewed, followed by an expansion to nonlinear system. The QP problem for linear systems may be described by Equation 3.1.6.

$$\min_{\phi \in R^n} J(\phi) = \sum_{k=0}^{N-1} \left(\frac{1}{2} x_{k+1}^\top Q_{1,k+1} x_{k+1} + d_{x_{t+1}}^\top x_{t+1} + \frac{1}{2} u_k^\top R_{1,k} u_k + d_{u_k}^\top u_k \right) \tag{3.1.6a}$$

subject to

$$x_{k+1} = A_k x_k + B_k u_k, \qquad k = 0, 1, ..., N-1 \tag{3.1.6b}$$

$$x_0, u_{-1} = \text{given} \tag{3.1.6c}$$

$$x_k^{\text{low}} \leq x_k \leq x_k^{\text{high}}, \qquad k = 0, 1, ..., N \tag{3.1.6d}$$

$$u_k^{\text{low}} \leq u_k \leq u_k^{\text{high}}, \qquad k = 0, 1, ..., N-1 \tag{3.1.6e}$$

$$-\Delta u^{\max} \leq \Delta u_k \leq \Delta u^{\max}, \qquad k = 0, 1, ..., N-1 \tag{3.1.6f}$$

where

$$Q_{1,k} \geq 0, \qquad k = 0, 1, ..., N \tag{3.1.6g}$$

$$R_{1,k} \geq 0, \qquad k = 0, 1, ..., N-1 \tag{3.1.6h}$$

$$\Delta u_k = u_k - u_{k-1} \tag{3.1.6i}$$

This formulation makes it possible to penalize some states or inputs differently then others, by adjusting the values of the elements of the matrices $Q_{1,k}$ and $R_{1,k}$. The quadratic objective function is widely used, and is a useful formulation for a dynamic optimization problem. It can easily be extended, for example by penalizing control moves by adding the following term

$$\frac{1}{2} \Delta u_k^\top R_{2,k} \Delta u_k, \quad R_{2,k} \geq 0 \tag{3.1.7}$$

In practice, this would penalize wear on equipment, and adjust the aggressiveness of the controller. Note that deviation variables are being used in the formulation in Equation 3.1.6, meaning that x_k represents the deviation between the actual value and the reference value (Foss and Heirung, 2013).

$$x_k - x_k^{ref} \tag{3.1.8}$$

Nonlinear systems

For nonlinear system, Equation 3.1.6 may be rewritten to the following equation

$$\min_{\phi \in R^n} J(\phi) = \sum_{k=0}^{N-1} \left(\frac{1}{2} x_{k+1}^\top Q_{1,k+1} x_{k+1} + d_{x_{t+1}}^\top x_{t+1} + \frac{1}{2} u_k^\top R_{1,k} u_k + d_{u_k}^\top u_k \right) \tag{3.1.9a}$$

subject to

$$x_{k+1} = f(x_k, u_k), \qquad k = 0, 1, ..., N-1 \tag{3.1.9b}$$

$$x_0, u_{-1} = \text{given} \tag{3.1.9c}$$

$$x_k^{\text{low}} \leq x_k \leq x_k^{\text{high}}, \qquad k = 0, 1, ..., N \tag{3.1.9d}$$

$$u_k^{\text{low}} \leq u_k \leq u_k^{\text{high}}, \qquad k = 0, 1, ..., N-1 \tag{3.1.9e}$$

$$-\Delta u^{\max} \leq \Delta u_k \leq \Delta u^{\max}, \qquad k = 0, 1, ..., N-1 \tag{3.1.9f}$$

where

$$Q_{1,k} \geq 0, \qquad k = 0, 1, ..., N \tag{3.1.9g}$$

$$R_{1,k} \geq 0, \qquad k = 0, 1, ..., N-1 \tag{3.1.9h}$$

$$\Delta u_k = u_k - u_{k-1} \tag{3.1.9i}$$

As for linear systems, penalties for control moves (Equation 3.1.7), in addition to other terms may also be added to this objective function (Foss and Heirung, 2013).

The main difference compared to the linear system, is the introduction of nonlinear equality constrains, $f(x_k, u_k)$. A nonlinear model often results in a non-convex optimization problem, and raises the need for a NLP solver, for example, the SQP approach. Optimization of such problems is often much more time consuming than convex problems (Hovd, 2004).

3.2 Model Predictive Control

The concept of MPC is to use a dynamic model to predict the future behaviour of a system and optimize some objective function to produce the best decisions. An important advantage of MPC is its ability to handle constraints, both for controllers and states. It has, therefore, been widely used in industries where satisfying the constraints are important, simultaneously to operating close to these constraints (Rawlings et al., 2017).

The concept of MPC is illustrated in Figure 3.1. In MPC, the current control move is found by solving on-line, at each sample instant, a finite horizon open-loop optimal control problem. The current state of the plant is used as the initial state. This results in a sequence of control actions, where only the first action is applied to the plant (Mayne et al., 2000). Essentially, a similar

problem is solved over and over again, at each time step. Hence, MPC is using a *moving horizon* approach, where the horizon is shifted one sample from one optimization problem to the next (Foss and Heirung, 2013).

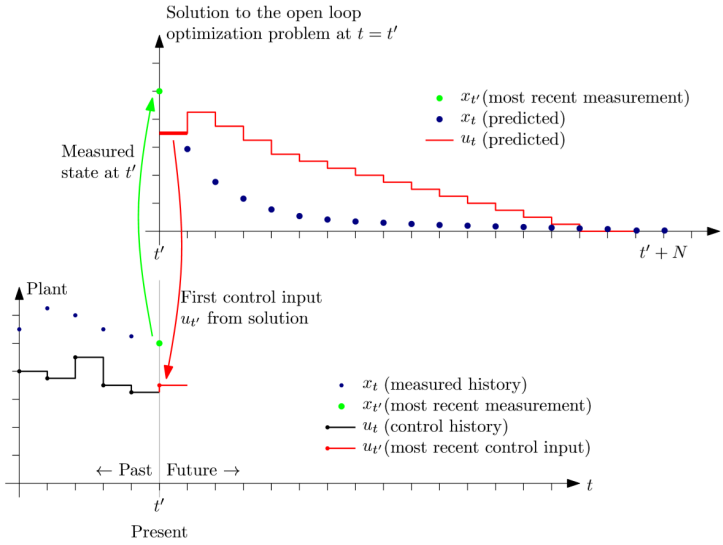


Figure 3.1: Illustration of the concept of MPC[1] from Foss and Heirung (2013)

MPC couples open-loop optimization with feedback by using the current state as the initial state in the optimization problem. This state can be obtained in a number of ways. One way is to use the first predicted state value, which is called *State Feedback MPC*. A drawback with this approach is that it does not account for error in the model or disturbances. Another approach is, therefore, to compute an estimate of the states which is dependent on the latest plant measurements, and use this as the initial state of the optimization problem. This is called *Output Feedback MPC*, and the estimate of the current state can for example be found by using Kalman Filter (KF) or Moving Horizon Estimation (MHE) (Foss and Heirung, 2013). This will be discussed further in Section 3.3.

The MPC generally predicts the behaviour of the outputs over a finite set of future time intervals, called the *prediction horizon*. The prediction horizon should be set long enough to capture the steady-state effect of the control inputs. The MPC computes a series of control inputs, spread over a finite *control horizon*. The performance increase as the control and prediction horizon increases, but at the expense of additional computation (Qin and Badgwell, 2003). In this thesis, the prediction horizon and control horizon will be denoted N and M, respectively. Both are tuning parameters in the MPC, and N must be at least as long as M.

[1] In this master thesis, k is used to denote the sampling number. t is however used in this figure.

A objective function for linear MPC, resembles to the QP problem in Equation 3.1.6. Typically, the objective function of MPC also contain a penalty for control moves, as described by Equation 3.1.7. The objective function may then be described by Equation 3.2.1.

$$\min_{\phi \in R^n} J(\phi) = \sum_{k=0}^{N-1} \left(\frac{1}{2} x_{k+1}^\top Q_{1,k+1} x_{k+1} + d_{x_{k+1}}^\top x_{k+1} + \frac{1}{2} u_k^\top R_{1,k} u_k + d_{u_k}^\top u_k + \frac{1}{2} \Delta u_k^\top R_{2,k} \Delta u_k \right) \tag{3.2.1a}$$

subject to

$$x_{k+1} = A_k x_k + B_k u_k \tag{3.2.1b}$$
$$x_0, u_{-1} = \text{given} \tag{3.2.1c}$$
$$x_k^{\text{low}} \leq x_k \leq x_k^{\text{high}} \tag{3.2.1d}$$
$$u_k^{\text{low}} \leq u_k \leq u_k^{\text{high}} \tag{3.2.1e}$$
$$-\Delta u^{\max} \leq \Delta u_k \leq \Delta u^{\max} \tag{3.2.1f}$$

where

$$Q_{1,k} \geq 0 \tag{3.2.1g}$$
$$R_{1,k} \geq 0 \tag{3.2.1h}$$
$$R_{2,k} \geq 0 \tag{3.2.1i}$$
$$\Delta u_k = u_k - u_{k-1} \tag{3.2.1j}$$

Linear MPC is a QP problem. The extension from linear MPC to a Nonlinear Model Predictive Control (NMPC), involves the introduction of a nonlinear model. The objective function for the nonlinear case is very much alike the linear case and can be described by Equation 3.2.2.

$$\min_{\phi \in R^n} J(\phi) = \sum_{k=0}^{N-1} \left(\frac{1}{2} x_{k+1}^\top Q_{1,k+1} x_{k+1} + d_{x_{k+1}}^\top x_{k+1} + \frac{1}{2} u_k^\top R_{1,k} u_k + d_{u_k}^\top u_k + \frac{1}{2} \Delta u_k^\top R_{2,k} \Delta u_k \right) \tag{3.2.2a}$$

subject to

$$x_{k+1} = f(x_k, u_k) \tag{3.2.2b}$$
$$x_0, u_{-1} = \text{given} \tag{3.2.2c}$$
$$x_k^{\text{low}} \leq x_k \leq x_k^{\text{high}} \tag{3.2.2d}$$
$$u_k^{\text{low}} \leq u_k \leq u_k^{\text{high}} \tag{3.2.2e}$$
$$-\Delta u^{\max} \leq \Delta u_k \leq \Delta u^{\max} \tag{3.2.2f}$$

where

$$Q_{1,k} \geq 0 \tag{3.2.2g}$$
$$R_{1,k} \geq 0 \tag{3.2.2h}$$
$$R_{2,k} \geq 0 \tag{3.2.2i}$$
$$\Delta u_k = u_k - u_{k-1} \tag{3.2.2j}$$

This transforms the QP problem of the linear MPC, to a nonlinear and non-convex problem, raising the need for a NLP solver (Foss and Heirung, 2013).

3.2.1 Feasibility and Stability

Constraints on the controller inputs are often related to physical limits, while constraints on states or output often are merely desirable. The optimization problem may, therefore, often define input constraints as hard constraints. Constraints on states and outputs may be relaxed in some manner and are called soft constraints. For example, it is physically impossible to open or close a valve more than 0% or 100%. Soft constraints, on the other hand, may be violated, usually at some cost, by introducing slack variables, ε. An advantage of introducing soft constraints is that the control problem will not be infeasible, as it can be relaxed by choosing an appropriate value of ε (Rawlings et al., 2017).

An optimization problem is infeasible if there exist no set of values of the decision variable so that all the constraints are fulfilled. If a large disturbance happens when using MPC and operating close to the constraint, such infeasibilities may occur. It is then crucial for the controller not to 'give up', and effectively steer the process into an operating region where all the constraints again are fulfilled. This can be ensured by introducing slack variables (Hovd, 2004).

Slack variables introduce additional terms to the objective function. Taking the linear system as an example, the QP problem in Equation 3.2.1 can now be written in the following way

$$\begin{aligned}\min_{\phi \in R^n} J(\phi) = \sum_{k=0}^{N-1} \Bigg(& \frac{1}{2} x_{k+1}^\top Q_{1,k+1} x_{k+1} + d_{x_{k+1}}^\top x_{k+1} + \frac{1}{2} u_k^\top R_{1,k} u_k \\ & + d_{u_k}^\top u_k + \frac{1}{2} \Delta u_k^\top R_{2,k} \Delta u_k + \frac{1}{2} \varepsilon^\top R_3 \varepsilon + r^\top \varepsilon \Bigg)\end{aligned} \tag{3.2.3a}$$

subject to

$$x_{k+1} = A_k x_k + B_k u_k \tag{3.2.3b}$$
$$x_0, u_{-1} = \text{given} \tag{3.2.3c}$$
$$x_k^{\text{low}} - \varepsilon \leq x_k \leq x_k^{\text{high}} + \varepsilon \tag{3.2.3d}$$
$$u_k^{\text{low}} \leq u_k \leq u_k^{\text{high}} \tag{3.2.3e}$$
$$-\Delta u^{\max} \leq \Delta u_k \leq \Delta u^{\max} \tag{3.2.3f}$$

where

$$Q_{1,k} \geq 0 \tag{3.2.3g}$$
$$R_{1,k} \geq 0 \tag{3.2.3h}$$
$$R_{2,k} \geq 0 \tag{3.2.3i}$$
$$\Delta u_k = u_k - u_{k-1} \tag{3.2.3j}$$
$$\varepsilon \in R^{n_x} \geq 0 \tag{3.2.3k}$$
$$r \in R^{n_x} \geq 0 \tag{3.2.3l}$$
$$R_3 \in \text{diag}\{r_1, ..., r_{n_x}\}, \quad r_i \geq 0, \quad i = \{1, ..., n_x\} \tag{3.2.3m}$$

The additional terms are $\frac{1}{2}\varepsilon^\top R_3 \varepsilon$ and $r^\top \varepsilon$, and it is desirable to drive these terms to zero. They should only be nonzero when the constraints are violated (Foss and Heirung, 2013).

Feasibility is, however, not a guarantee for stability, i.e. the trajectories may not converge. It can be ensured by reformulating the optimization problem. This can be done in several ways, for example, by adding a terminal cost function ($V_f(x_N)$) and terminal constraints (χ_f). Equation 3.2.3 can then be expressed in the following way

$$\min_{\phi \in R^n} J(\phi) = \sum_{k=0}^{N-1} \left(\frac{1}{2} x_{k+1}^\top Q_{1,k+1} x_{k+1} + d_{x_{k+1}}^\top x_{k+1} + \frac{1}{2} u_k^\top R_{1,k} u_k \right.$$
$$\left. + d_{u_k}^\top u_k + \frac{1}{2} \Delta u_k^\top R_{2,k} \Delta u_k + \frac{1}{2} \varepsilon^\top S \varepsilon + \rho^\top \varepsilon \right) + V_f(x_N) \tag{3.2.4a}$$

subject to

$$x_{k+1} = A_k x_k + B_k u_k \tag{3.2.4b}$$
$$x_0, u_{-1} = \text{given} \tag{3.2.4c}$$
$$x_k^{\text{low}} - \varepsilon \leq x_k \leq x_k^{\text{high}} + \varepsilon \tag{3.2.4d}$$
$$u_k^{\text{low}} \leq u_k \leq u_k^{\text{high}} \tag{3.2.4e}$$
$$-\Delta u^{\max} \leq \Delta u_k \leq \Delta u^{\max} \tag{3.2.4f}$$

where

$$Q_{1,k} \geq 0 \tag{3.2.4g}$$
$$R_{1,k} \geq 0 \tag{3.2.4h}$$
$$R_{2,k} \geq 0 \tag{3.2.4i}$$
$$\Delta u_k = u_k - u_{k-1} \tag{3.2.4j}$$
$$\varepsilon \in R^{n_x} \geq 0 \tag{3.2.4k}$$
$$\rho \in R^{n_x} \geq 0 \tag{3.2.4l}$$
$$S \in \text{diag}\{s_1, ..., s_{n_x}\}, \quad s_i \geq 0, \quad i = \{1, ..., n_x\} \tag{3.2.4m}$$
$$x_N \in \chi_f \tag{3.2.4n}$$

The terminal cost, $V_f(x_N)$, may be expressed as a quadratic function

$$V_f(x_N) = \frac{1}{2} x_N^\top P_f x_N, \quad P_f \geq 0 \tag{3.2.5}$$

Stability can also be achieved with sufficiently long enough prediction horizon compared to the control horizon (Alamir and Bornard, 1995; Hovd, 2004; Foss and Heirung, 2013).

3.2.2 Horizon Parameterization: Input blocking and coincidence points

Input blocking, or *move blocking* is a common way to parameterize the control input sequence (Rawlings et al., 2017). The user may then specify points on the control horizon, which will or will not be computed, and the control trajectory is forced to remain constant for some steps. This will reduce the number of decision variables and reduce runtime, especially for NMPC applications. This can, however, be at the expense of performance (Foss and Heirung, 2013; Qin and Badgwell, 2003). Figure 3.2 illustrates the parameterization of the input and size of the control horizon. The example shown is the control of the temperature of a reactor (output) using a Split Range Controller (SRC) (input). In this case, the input is parameterized with four parameters, which determines the input at 0, 500, 1000, 2000 and 4000 seconds. Hence, the control horizon is 4000 seconds. After the final block, the input will remain constant.

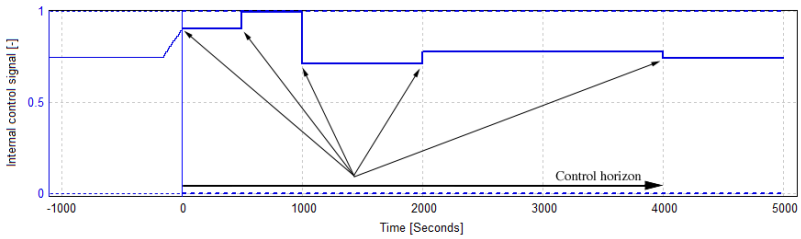


Figure 3.2: Illustration of the concept of input blocking, with a control horizon of 4000 seconds. Here, the input is parameterized with four parameters, which determines the input at 0, 500, 1000, 2000 and 4000 seconds.

Like input blocking, there are also techniques to reduce the number of points in the prediction horizon. This is done by using *coincidence points*, which is a number of appropriate chosen points in the prediction horizon. The length of the prediction horizon will remain the same, but the runtime can be reduced significantly. For short prediction horizons, the coincidence points may be set at every sampling point. For longer prediction horizons, the calculation may, for example, be simplified by choosing coincidence points at equidistant intervals (Zhu and Zhou, 2009). Figure 3.3 illustrates the parameterization of the output, and the length of the prediction horizon. The coincidence points are

marked by black dots. In this case, the output is weighted every 500 seconds, starting at 500 seconds and ending at 5000 seconds. That is, the output is weighted in equidistant intervals. The prediction horizon, in this case, is 5000 seconds.

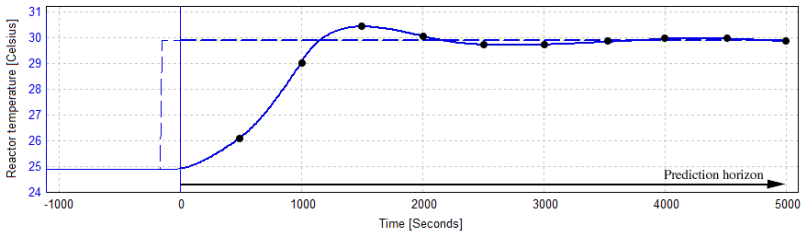


Figure 3.3: Illustration of the concept of coincidence points (black dots), by using temperature of a reactor (solid) and its setpoint (dashed), as an example. Here, the output is weighted at every 500 seconds, starting at 500 seconds and ending at 5000 seconds. That is, the prediction horizon is parameterized by equidistant intervals and the prediction horizon is 5000 seconds.

3.3 State- and parameter estimation

As mentioned in Section 3.2, MPC uses the current state as the initial state in the optimization problem. The state could be obtained by prediction, but this approach does not account for any errors in the model or disturbances. Another approach is to compute an estimate of the states depending on plant measurements. Model errors and disturbances are expected, and it is, therefore, necessary to update the process model to ensure good quality of the predictions of the process behaviour. The update will, however, not be as dramatic if the model predicts the behaviour of the plant by high accuracy, compared to a more poor model (Hovd, 2004; Foss and Heirung, 2013).

Which measurements that should be used to update the model most be determined in advance. For instance, the emulsion polymerization process of Poly-Vinyl Chloride (PVC) at Inovyn's facilities, the reactor temperature is measured. If the temperature also is calculated by the model, the deviation between the two can be used to update the model, making the model reflect the actual process even better. Temperature measurements are often candidates for model updating, as these measurements often are fast, accurate, frequently sampled and not very delayed.

There is a number of approaches that may be used for estimation, as MHE and KF, which probably are some of the most common and recognized approaches. The basic KF (first proposed by Kalman (1960)) is derived for linear system, and may be extended for nonlinear systems, as in the Extended Kalman Filter (EKF). MHE is optimization-based, where the state estimation is determined on-line by solving a finite horizon state estimation problem (Rao et al.,

2003). The material is adapted from Simon (2006), and the reader is advised to this book for a more elaborated description. In this thesis, KF was used, and will, therefore, be elaborated in the following subchapters, both for linear and nonlinear time discrete systems.

3.3.1 Kalman Filter for linear systems

It is assumed that a linear model is formulated discretely in the following way

$$x_{k+1} = A_k x_k + B_k u_k + v_k \tag{3.3.1}$$
$$y_k = H_k x_k + w_k \tag{3.3.2}$$

Here, v_k is a vector of process noise and w_k is a vector of measurement noise. They are assumed to be white, zero-mean and uncorrelated. v_k has the known covariance matrix Q_k, and w_k has the known covariance matrix R_k. H_k is often called the output matrix and decides which states, or combination of states, that will go into the measurement vector, y_k.

The purpose of the estimator is to give an estimate of the state at k, which in this case is to be used as the initial state of the optimization problem. Depending on the amount of information the system is given, either a *a priori* or a *a posteriori* estimate can be calculated. The *a priori* estimate will be denoted as $\bar{x}_k$, and is calculated when the system holds the measurements from y_1 to y_{k-1}. The *a posteriori* estimate will be denoted as $\hat{x}_k$, and can be calculated when the system holds the measurements from y_1 to y_k.

$$\bar{x}_k, \quad \textit{a priori} \text{ estimate} \tag{3.3.3}$$
$$\hat{x}_k, \quad \textit{a posteriori} \text{ estimate} \tag{3.3.4}$$

$\bar{x}_k$ is estimated before y_k is processed, and $\hat{x}_k$ is estimated after y_k is processed. The calculation of the *a priori* estimates are often referred to as the prediction, and predict the states for the current time step, k, by the model and information from the previous time step, $k-1$. The calculation of the *a posteriori* estimates are often referred to as the correction and corrects the prediction done at the current time step, k.

P_k will be used to denote the covariance of the estimation error, and follows the same notation as for the state estimates.

$$\bar{P}_k = E[(x_k - \bar{x}_k)(x_k - \bar{x}_k)^\top], \quad \textit{a priori} \text{ covariance} \tag{3.3.5}$$
$$\hat{P}_k = E[(x_k - \hat{x}_k)(x_k - \hat{x}_k)^\top], \quad \textit{a posteriori} \text{ covariance} \tag{3.3.6}$$

The time update for the state, from the a posteriori time at $k-1$ to the a priori time at k, is obtained by the following equation.

$$\bar{x}_k = F_{k-1}\hat{x}_{k-1} + G_{k-1}u_{k-1} \tag{3.3.7}$$

Here, F and G is Jacobian matrices. As no information about the measurement at the current time step, k, is yet obtained, the estimate is only updated based

on available information. Then, the time update for the covariance matrix is performed

$$\bar{P}_k = F_{k-1}\hat{P}_{k-1}F_{k-1}^\top + Q_{k-1} \tag{3.3.8}$$

As already mentioned, Q_k represent the covariance matrix for the process noise. When information about the measurements at k is obtained, the measurement-update can be performed

$$\hat{x}_k = \bar{x}_k + K_k(y_k - H_k\bar{x}_k) \tag{3.3.9}$$

$$K_k = \bar{P}_k H_k^\top \left(H_k\bar{P}_k H_k^\top + R_k\right)^{-1} \tag{3.3.10}$$

$$\hat{P}_k = (I - K_k H_k)\bar{P}_k \tag{3.3.11}$$

Here, K_k is the estimator gain matrix, R_k represent the covariance matrix for the measurement noise, and H_k represent the output matrix. Both $\bar{x}_k$ and $\hat{x}_k$ are estimates of x_k, but $\hat{x}_k$ is expected to be more accurate, as more information is used to calculate this estimate.

3.3.2 Extended Kalman Filter

An abbreviated summary of the theoretical aspects of the discrete-time EKF will now be presented to illustrate some similarities and differences to the linear case. All systems will, to some extent, be nonlinear. However, many systems will be close enough to linear so that linear estimation approaches may be used. EKF is probably the most common used for nonlinear state estimation. The idea of the EKF is to linearize a nonlinear system around the Kalman filter estimate, which is based on a linearized system (Simon, 2006).

It is assumed that a nonlinear model is formulated discretely

$$x_{k+1} = f(x_k, v_k, u_k) \tag{3.3.12}$$

$$y_k = g(x_k, w_k) \tag{3.3.13}$$

Here, $f(\,\cdot\,)$ represent the the nonlinear process model, and g represent a nonlinear model of the measurements. The *a priori* estimates are calculated by the following equations

$$\bar{x}_k = f(\hat{x}_{k-1}, 0, u_{k-1}) \tag{3.3.14}$$

$$\bar{y}_k = g(\bar{x}_k, 0) \tag{3.3.15}$$

$$\bar{P}_k = F\hat{P}_{k-1}F^\top + LQ_{k-1}L^\top \tag{3.3.16}$$

Again, Q_k represent the covariance matrix for the process noise. Note that during the Taylor series expansion, the state equation was linearized around $x_{k-1} = \hat{x}_{k-1}$ and $v_{k-1} = 0$, and the measurement equation around $x_k = \bar{x}_k$ and $w_k = 0$. Here, F and L are given by the following relations, evaluated at

the posteriori state estimate at $k-1$, that is $x = \hat{x}_{k-1}$.

$$F = \left.\frac{\partial f}{\partial x}\right|_{\hat{x}_{k-1}} \tag{3.3.17}$$

$$L = \left.\frac{\partial f}{\partial v}\right|_{\hat{x}_{k-1}} \tag{3.3.18}$$

The *a posteriori* estimates are calculated by the following equations

$$\hat{x}_k = \bar{x}_k + K_k(y_k + \bar{y}_k) \tag{3.3.19}$$

$$K_k = \bar{P}_k H_k^\top \left(H_k \bar{P}_k H_k^\top + M_k R_k M_k^\top\right)^{-1} \tag{3.3.20}$$

$$\hat{P}_k = (I - K_k H_k)\bar{P}_k \tag{3.3.21}$$

Here, K_k is the estimator gain matrix, y_k is the process measurement, and R_k is the measurement noise covariance matrix. I represent the identity matrix, and H and M are given by the following relations, evaluated at the current a priori state estimate, that is $x = \bar{x}_k$.

$$H_k = \left.\frac{\partial g}{\partial x}\right|_{\bar{x}_k} \tag{3.3.22}$$

$$M_k = \left.\frac{\partial g}{\partial w}\right|_{\bar{x}_k} \tag{3.3.23}$$

Other linearization techniques may be used to reduce the error that occurs by linearizing. Iterative EKF and higher-order EKF are examples of such approaches. These approaches typically has a higher accuracy than EKF, but it often comes at the price of longer runtime.

The specific algorithm used in this thesis was a divided difference approach, which resembles very much to the EKF. The differences lie in the way the covariance matrixes are updated, while the state update is generally the same for EKF and the divided difference approach. The reader is advised to read Schei (1997) and Noergaard et al. (2000) on this topic.

3.3.3 Parameter estimation

The theory stated above can also be used to estimate unknown process parameters. Consider the following nonlinear system

$$x_{k+1} = f(x_k, v_k, u_k, p) \tag{3.3.24}$$

$$p = \text{unknown process parameter} \tag{3.3.25}$$

Here, p is a vector of unknown process parameters, which should be subject to on-line parameter estimation. p will be a subset of the entire collection of process parameters, θ.

$$p \in \theta \tag{3.3.26}$$

The states are augmented with the parameter vector, and will be denoted x'_k

$$x'_k = \begin{pmatrix} x_k \\ p_k \end{pmatrix} \tag{3.3.27}$$

Noise can then be included in the parameters the following way

$$p_{k+1} = h(p_k, v_{p,k}) \tag{3.3.28}$$

Here, $h(\,\cdot\,)$ is a function that includes the noise, $v_{p,k}$, to the parameters. The augmented model can then be described by the following equations

$$x'_{k+1} = \begin{pmatrix} f(x_k, v_k, u_k) \\ h(p_k, v_{p,k}) \end{pmatrix} = f(x'_k, v_k, u_k, v_{p,k}) \tag{3.3.29}$$

The augmented state vector, x'_k, can now be estimated. This can for example be done by using some of the approaches mentioned in previous subchapters (Simon, 2006). Examples of parameters which could benefit from parameter estimation may be overall and individual heat transfer coefficients, thermal conductivities, model correction factors, and so on. Take a emulsion polymerization reactor with a cooling jacket as an example. The reactor content may vary throughout the batch as the process reaches higher monomer conversion. The viscosity of the content might change, which can have an significant effect on the heat transfer coefficient between the reactor and the cooling jacket. As a result, the heat transfer coefficient may then also vary. The heat transfer coefficient can then be a promising candidate for estimation.

Chapter 4

Process description and control structure

The purpose of this section is to give the reader insight into the process by describing reactor design and chemical dosage. This chapter will, therefore, contain a summarized description of the process and its control structure. Together with the modelling of the process, which will be described in Chapter 5, this will be important when later discussing the topic of controlling the reactor.

4.1 Process description

The recipe of this process is confidential, and will, therefore, not be stated in this report. Some of the chemicals used will also be kept confidential.

In this process, two different surfactants are used and will be denoted *Surfactant 1* and *Surfactant 2*, or just generalized *Surfactant* if there is no need to distinguish between the two. Two different initiators are used, and will in the same manner be denoted *Initiator 1* and *Initiator 2*, or just *Initiator*. Initiator 1 is a hydroperoxide, ROOH, and Initiator 2 a persulfate. Some additives, which will be denoted *Additives*, are both initial dosed and post-dosed. Neither the amount nor composition of the additives are relevant for the model, but they still are important for the real process. Vinyl Chloride Monomer (VCM) is the monomer which will polymerize to the desired product, Poly-Vinyl Chloride (PVC).

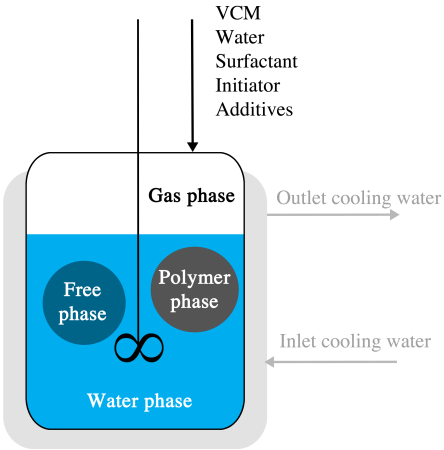


Figure 4.1: An illustration of the relevant process. A stirred semi-batch reactor is surrounded by a cooling jacket with in- and outlet streams (light grey). The process consists of four phases; gas, free, polymer and water phase. The reactor is initially loaded with monomer, water, surfactant, initiator and other additives. During the batch, surfactant, water, initiator and additives are post-dosed. Nothing is drained from the reactor before the batch is over.

The reaction unit will consist of three or four phases, depending on the current interval, as described in Section 2.3.1. They are all shown in Figure 4.1; gas phase, polymer phase, free phase and water phase. Initially, the reactor is loaded with water, monomer, some surfactants, initiator and additives. During the batch, Initiator 1, water, surfactants and additives are post-dosed. That is, many of the chemicals may be post-dosed, but the amount of monomer is only dependent on the initial loading.

The initiation mechanism taking place is a combined reduction-oxidation

and thermal system. Even though some initiator is initially dosed, the polymerization will not start taking place before some additives are dosed.

In this thesis, the initial loading of the reactor is assumed not to vary from one batch to another. This is for the purpose of modelling and simulation. The required feed rate and the total amount of surfactants and additives, may, however, variate with different temperature and initiator conditions. It is assumed in the calculations that neither water, surfactants or additives will be rate determining. This might not be the case in the real plant, and the addition of these species might be necessary.

4.2 Control structure

The reactor is a continuously stirred semi-batch, where nothing is drained from the reactor before the batch is finished. As shown in Figure 4.2, the reactor is surrounded by a cooling jacket. The inlet cooling water is located at the bottom of the reactor, while the outlet at the top. Some of the cooling water is drained, and some are recycled.

The jacket is used to regulate the temperature inside the reactor. This is done by regulating the temperature of the inlet of the jacket, by either heating the recycled stream or feeding cold, fresh water into the circuit. Its primary function, however, is to cool the reaction unit. As polymerization processes may be highly exothermic, controlling the temperature, avoiding reaction runaway and ensuring safe production are important. The structure is shown in Figure 4.2.

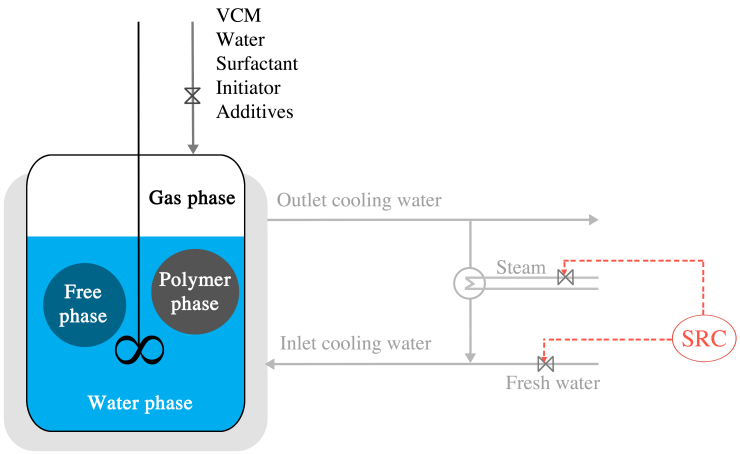


Figure 4.2: An illustration of the relevant process, as in Figure 4.1, including the valves which are relevant for this project. The temperature of the cooling jacket can be controlled by either heating the recycled water, or by addition of cold, fresh water. A Split Range Controller (SRC) is controlling these two valves.

The valve position of the steam inlet and fresh water inlet is regulated by a SRC, which will be described in detail in Section 5.2. The SRC will control the temperature of the inlet cooling water, which in its turn will contribute to heating or cooling the reactor content. The second manipulated variable is the feed rate of initiator into the system. These two control signals will be the only manipulated variables considered in this thesis.

Chapter 5

Modelling of a semi-batch reactor for emulsion polymerization

As already mentioned, a preliminary project was conducted in advance of this master thesis. The main goal of this preliminary work was to establish a model describing an industrial polymerization process, more specifically an emulsion polymerization. The model was validated against process data, and parameters were adjusted to minimize the error between the model and measured outputs. The results were then evaluated. As polymerization processes often have troublesome nonlinear dynamics (Ray and Villa, 2000), it was discussed if Nonlinear Model Predictive Control (NMPC) could be beneficial for this process. The work conducted during the previous project and its associated equations and assumptions can be found in Appendix A.

Some modelling work remained after the previous project. During the initial phase of this master thesis, some work was, therefore, carried out to finalize the model and making it suitable for NMPC. The model equations that follows in this section is a continuation of the preliminary project. The finalization includes the following features, which will be described in this chapter;

- Modelling parameters describing the product quality important to the manufacturer
- Complete the modelling of the cooling circuit
- A description of the unmodelled aspects
- Definition of a termination criteria
- A description of the maximum cooling capacity of the system.

5.1 Conversion average temperature

The K-value is a quality parameter, calculated as a function of the conversion average temperature, $K(\bar{T}_{X_\mathrm{M}})$. The relation will not be given as it is confidential. However, it is a measure of the quality of the end product. The conversion average temperature is defined by the following equation.

$$\bar{T}_{X_\mathrm{M}} = \frac{\int_0^t R_\mathrm{p} T_\mathrm{R} \mathrm{d}t}{\int_0^t R_\mathrm{p} \mathrm{d}t} \tag{5.1.1}$$

where $\bar{T}_{X_\mathrm{M}}$ is the conversion average temperature, R_p is the polymerization rate and T_R is the temperature in the reactor. This will introduce one additional state to the system, as the integral in the denominator can be described as

$$\int_0^t R_\mathrm{p} \mathrm{d}t = X_\mathrm{M} \int_0^t \hat{n}_\mathrm{M} \mathrm{d}t \tag{5.1.2}$$

where X_M is a state describing the monomer conversion, and the second integral represent the total molar amount of fed monomer into the reactor. Consequently, the conversion average temperature can be described by the following equation

$$\bar{T}_{X_\mathrm{M}} = \frac{\int_0^t R_\mathrm{p} T_\mathrm{R} \mathrm{d}t}{X_\mathrm{M} \int_0^t \hat{n}_\mathrm{M} \mathrm{d}t} \tag{5.1.3}$$

5.2 Cooling circuit

For the preliminary work, the measured inlet temperature and the initiator feed were the inputs of the system. At the real plant, the cooling system is partially closed, as was shown by Figure 4.2. Some of the water is recycled, while some are drained. The temperature can be adjusted by either heating the recycled stream using a heat exchanger or cooled by adding cold, fresh water. The opening of these valves is controlled using a Split Range Controller (SRC).

The model was updated so that the inlet temperature could be calculated by the model. Figure 5.1 shows the cooling circuit, which also was shown in Figure 4.2, in addition to the temperature of the flows.

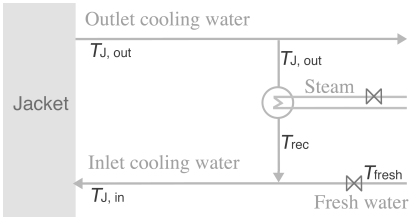


Figure 5.1: Illustration of the cooling circuit, in addition to temperature of the flows.

By making a steady state mass and energy balance over the mixing point, the inlet temperature can be calculated by the following equation.

$$T_{\mathrm{J,in}} = \frac{T_{\mathrm{fresh}}\hat{m}_{\mathrm{fresh}} + T_{\mathrm{rec}}\hat{m}_{\mathrm{rec}}}{\hat{m}_{\mathrm{fresh}} + \hat{m}_{\mathrm{rec}}} \tag{5.2.1}$$

where T_{fresh} is the temperature of the fresh water, $\hat{m}_{\mathrm{fresh}}$ is the mass flow of the fresh water, T_{rec} is the temperature of the recycled water after the heat exchanger, and $\hat{m}_{\mathrm{rec}}$ is the mass flow of the recycled water. In this derivation it was assumed that the heat capacity was independent of temperature. This was done as it was assumed that the change in temperature, and followingly the heat capacity, would be quite small.

Both valves were assumed to follow an exponential characteristic, and is given by the following equation.

$$q(z) = az^R, \quad 0 \leq z \leq 1 \tag{5.2.2}$$

Here, $q(z)$ represent a volume flow, a is a constant, R is a design parameter, and z is the valve opening. By using the following end condition, a can be obtained.

$$q(1) = q^{\max} \tag{5.2.3}$$

Here, $q^{\max}$ represent the maximum volume flow through the pipe. $q(z)$ will then be given by the following equation.

$$q(z) = q^{\max}z^R \tag{5.2.4}$$

The volume flow of steam (q_{steam}) and fresh water (q_{fresh}) can then be calculated.

$$q_{\mathrm{fresh}} = q_{\mathrm{fresh}}^{\max} {z_{\mathrm{fresh}}}^{R_{\mathrm{fresh}}} \tag{5.2.5}$$

$$q_{\mathrm{steam}} = q_{\mathrm{steam}}^{\max} {z_{\mathrm{steam}}}^{R_{\mathrm{steam}}} \tag{5.2.6}$$

Here, z_{fresh} represent the valve opening of the feed of fresh water, z_{steam} represent the valve opening of steam into the heat exchanger, R_{fresh} is the design parameter for the valve for fresh water, and R_{steam} is the design parameter for the valve for steam. By using Equation 5.2.5 and 5.2.6, the mass flows, $\hat{m}_{\mathrm{fresh}}$ and $\hat{m}_{\mathrm{steam}}$ can be calculated.

$$\hat{m}_{\mathrm{fresh}} = q_{\mathrm{fresh}}\rho_{\mathrm{fresh}} \tag{5.2.7}$$

$$\hat{m}_{\mathrm{steam}} = q_{\mathrm{steam}}\rho_{\mathrm{steam}} \tag{5.2.8}$$

Here, ρ_{fresh} represent the density of the fresh water, and ρ_{steam} represent the density of the steam. The heat transfer from the steam to the recycled water can then be calculated.

$$\dot{Q} = \hat{m}_{\mathrm{steam}}h_{\mathrm{fg}} \tag{5.2.9}$$

Here, h_{fg} is the latent heat of vaporization. The temperature of the recycled water after the heat exchanger, but before the feed for fresh water, can then be calculated.

$$T_{\text{rec}} = \frac{\dot{Q}}{\hat{m}_{\text{rec}} c_{p,\text{rec}}} + T_{\text{J,out}} \tag{5.2.10}$$

Here, $c_{p,\text{rec}}$ represent the heat capacity of the recycled stream. The temperature of the inlet stream can then be calculated using Equation 5.2.1.

It is assumed that $\hat{m}_{\text{rec}}$ is kept at a constant value. Note that heating and cooling is not used simultaneously. So when heating, the $T_{\text{J,in}}$ corresponds to T_{rec} and when cooling, T_{rec} corresponds to $T_{\text{J,out}}$.

As mentioned, this is controlled using SRC, and the criteria for this controller is summarized in Equation 5.2.11 and 5.2.12, and visualized by Figure 5.2.

$$\left.\begin{aligned} z_{\text{fresh}} &= z_{\text{fresh}}^{\max} - \tfrac{1}{\nu^*}\nu \\ z_{\text{steam}} &= z_{\text{steam}}^{\min} \end{aligned}\right\} \text{ for } \nu^{\min} \leq \nu \leq \nu^* \tag{5.2.11}$$

$$\left.\begin{aligned} z_{\text{fresh}} &= z_{\text{fresh}}^{\min} \\ z_{\text{steam}} &= z_{\text{steam}}^{\min} + \tfrac{1}{z_{\text{steam}}^{\max} - \nu^*}(\nu - \nu^*) \end{aligned}\right\} \text{ for } \nu^* < \nu \leq \nu^{\max} \tag{5.2.12}$$

Here, ν represent the internal control signal of the SRC, $\nu^{\min}$ and $\nu^{\max}$ represent the minimum and maximum value of the internal control signal respectively, and ν^* represent the critical value of the internal control signal.

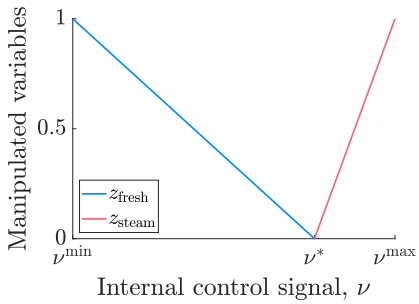


Figure 5.2: The split range controller action, equivalent to what is described by Equation 5.2.11 and Equation 5.2.12. For values of the internal control signal below ν^*, fresh water is fed into the circuit. For values over ν^*, steam is used to heat of the recycled water. For the critical value ν^*, neither fresh water or steam are used.

A delay to the feed of fresh water into the cooling circuit was also added. This introduces one additional state to the system and is described by the following Ordinary Differential Equation (ODE).

$$\frac{\mathrm{d}\hat{m}_{\text{fresh}}^{\text{delayed}}}{\mathrm{d}t} = \left(\hat{m}_{\text{fresh}} - \hat{m}_{\text{fresh}}^{\text{delayed}}\right)\frac{1}{\tau_{\text{fresh}}} \tag{5.2.13}$$

Here, $\hat{m}_{\text{fresh}}^{\text{delayed}}$ represent the state for the delayed mass flow of fresh water into the cooling circuit. This state is meant to capture the delay in the pipelines. That is; the time from opening the valve, until the fresh water has reached to cooling jacket. $\hat{m}_{\text{fresh}}$ is the non-delayed fresh water feed, calculated by Equation 5.2.7, and τ_{fresh} represent a time constant for the delay.

5.3 Unmodelled aspects

During the preliminary work, the correction factor for the kinetic model (CF), and the heat transfer coefficient between the reactor and the jacket ($U_{\text{R,J}}$), were estimated during the batch for three different datasets. Profiles for these parameters were then found. These results can be found in Appendix B, represented by Figure B.2. During the initial phase of this master thesis, average profiles were found, and polynomials describing these parameters as a function of conversion were obtained. These polynomials will be considered to describe the unmodelled aspects of the process.

5.4 Termination criteria and batch time

An important aspect for control purposes is the batch time. This parameter can be included in the NMPC-criteria and minimized to shorten the total duration of the batch. The batch time is defined to begin the moment the polymerization starts and ends when 96% monomer conversion has been reached. Equation 5.4.1 represents the ODE describing the evolution of the batch time.

$$\frac{\mathrm{d}t_{\text{batch}}}{\mathrm{d}t} = \begin{cases} 1 & \text{for } 0 < X_{\text{M}} \leq 96\% \\ 0 & \text{else} \end{cases} \tag{5.4.1}$$

5.5 Maximum cooling capacity

Maximum cooling capacity is important in terms of safety. Polymerization processes may be extremely exothermic, and the cooling demand can not exceed the available cooling capacity.

The available cooling capacity will be defined by the following equation.

$$\Delta H^{\text{cooling capacity}} = U_{\text{R,J}} A_{\text{R,J}} (T_{\text{J}}^{\text{min}} - T_{\text{R}}) \tag{5.5.1}$$

where $T_{\text{J}}^{\text{min}}$ will be the minimum temperature possible for the water contained in the jacket, $U_{\text{R,J}}$ is the heat transfer coefficient between the reactor and the jacket and $A_{\text{R,J}}$ is the area of heat transfer between the reactor and the jacket. This will be equivalent to the feed temperature of fresh water into the cooling

circuit. The cooling demand can be calculated.

$$\begin{aligned}\Delta H^{\text{cooling demand}} = & \frac{\mathrm{d}T_{\mathrm{R}}}{\mathrm{d}t}\Big(\sum m_i c_{p,i} + m_{\text{steel}} c_{p,\text{steel}}\Big) - \\ & \big(-\Delta H_{\text{rx}} R_{\text{p}} - U_{\mathrm{R,amb}} A_{\mathrm{R,amb}} (T_{\text{amb}} - T_{\mathrm{R}}) \\ & \qquad + \sum \hat{m}_{\text{feed},i} c_{p,i} (T_{\text{feed},i} - T_{\mathrm{R}})\big) \\ = & \, U_{\mathrm{R,J}} A_{\mathrm{R,J}} (T_{\mathrm{J}} - T_{\mathrm{R}})\end{aligned} \tag{5.5.2}$$

Details on the right hand side of this equation can be found in Appendix A.2. To ensure that the cooling demand do not exceed the available cooling capacity, the relationship described in Equation 5.5.3 must be satisfied.

$$-\Delta H^{\text{cooling capacity}} > -\Delta H^{\text{cooling demand}} \tag{5.5.3}$$

Chapter 6

Control of a semi-batch reactor for emulsion polymerization

This chapter will introduce some key consideration on control of a semi-batch reactor for an emulsion polymerization process. This includes control of the batch time, temperature control, pressure control, quality parameter control and safety considerations. All of these topics are presented to motivate for the implementation of Nonlinear Model Predictive Control (NMPC), and formulation of the optimization problem.

This chapter will bind together the previous chapters, and the goal is to provide a comprehensive understanding of the problem addressed in this thesis.

6.1 The control problem

Controlling a semi-batch reactor for emulsion polymerization can be complex due to multiple reasons. During the different intervals of the batch (See Section 2.3.1), operational conditions may change. The viscosity of the reactor content can vary throughout the batch, leading to variations in the cooling capacity and cooling demand. Sudden or local changes in the liquid phase viscosity can cause the rate of the termination reaction to slow down, and there will be nothing to hold back to propagation, causing a sudden increase in production rate. Such changes may cause unexpected temperature increase and instabilities in the system. The quality of the polymer is also an important aspect as it determines the market value of the product. These parameters are often not measurable online, and the model must, therefore, estimate them. The operational conditions must ensure that the quality requirements of the polymer are met, at least within a reasonable error interval.

The dynamics of the polymerization process is highly nonlinear. Consequently, if Model Predictive Control (MPC) was to be implemented, a Nonlinear Programming (NLP) solver would be needed, which often is much more time consuming than linear solvers. MPC is, therefore, referred to as NMPC for the relevant process in this thesis.

When building a control problem, some of the main decisions are which variables to optimize, defining the constraints and the formulation of the objective function. The rest of this chapter will introduce some of these aspects.

6.2 Minimization of batch time

The batch time was defined in Chapter 5, by Equation 5.4.1. The batch time is an essential parameter for the manufacturer, as it is expensive to operate a reactor. These costs include salaries, materials, power for production, maintenance, wear of equipment, in addition to a whole range of other factors. Shorter batch times can lead to higher production volume and thus higher profitability. Besides, there would be more room for flexibility in terms of scheduling and maintenance. Minimizing the batch time will, therefore, be equivalent to reducing some of the expenses related to the operation.

As already mentioned, the batch time could be minimized if it is included in the objective function of the optimization problem in some fashion. This can be performed in numerous ways. One alternative is to introduce a setpoint for the batch time, and the objective function will penalize for any deviation from this setpoint. The setpoint could be set to some desired value or some unrealizable value. If the setpoint is set to zero, every additional time step will be penalized, and the optimization problem will always work towards minimum batch time. Another opportunity is to define an upper constraint for the batch time.

The above-mentioned approach will, however, require a prediction horizon at least as long as the duration of the batch, which can be computational

demanding. If the solver cannot find a solution in between samples, the controller will fall further and further behind. Rephrased, the numerical efficiency will be too low, and the model does not fulfil the real-time requirement. This is not a unique issue related to minimizing the batch time, but for all optimization problem if it is too computational demanding. Prediction horizons, control horizons, horizon parameterization, number of controlled and manipulated variables, the model itself and numerous other aspects will be important factors affecting the runtime. An alternative would be to define a setpoint to the production rate. A high setpoint for this variable would make the optimize work to maximize the production rate, and thus minimizing the batch time.

An important aspect of minimization of the batch time is not setting the penalty for setpoint deviation as high that the NMPC never will start the batch. In addition, some terminating criteria most be pre-defined. In this model, the termination criteria was set to the point the reaction reached 96% conversion. It is, therefore, important that the model predicts this behaviour, for example, by freezing the state of the batch time when the system reaches this criterion. Figure 6.1 shows the intuitive relation of the reactor batch time as function of time. The simulation, the setpoint and the prediction are shown in the figure. In the first 720 seconds, pre-dosing of chemicals are executed. When the polymerization process is started, the batch time follows the relation given in Equation 5.4.1. When the batch is finished according to the termination criteria, the state is frozen, which is shown as the plateau at the end of the prediction. When minimizing the batch time, it is desirable to shift this plateau as far down as possible, while still meeting other requirements, as safety and quality.

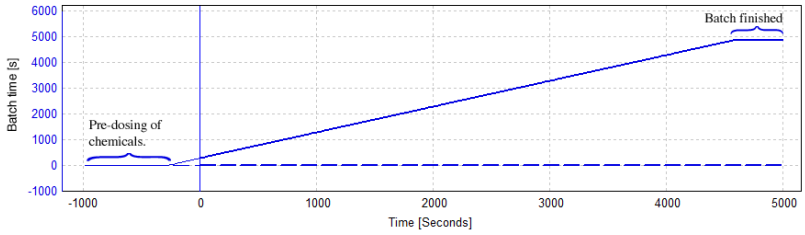


Figure 6.1: Batch time as function of time. The figure shows the simulation (solid) and the setpoint (dashed). Predicted values are illustrated in bold. The pre-dosing of chemicals are illustrated in the figure, in addition to the finalization of the batch. In between this time frame, the polymerization reaction is taking place.

6.3 Temperature and pressure control

As already emphasized, is the polymerization reaction a highly exothermic process. Controlling the reactor temperature (T_R) is, therefore, important, both for safety consideration, but also to meet quality requirements which may be

temperature-dependent. An example of this is the conversion average temperature ($\bar{T}_{X_M}$), which was elaborated in Section 5.1. The reactor temperature will be dependent on different factors, as heat transfer between a cooling or heating unit, heat transfer between the reactor and the environment, and the heat of reaction. The reactor temperature can be controlled to meet a certain setpoint requirement, be kept within some constraints, or both.

In the case of the process described in Chapter 4, there are two main ways to heat the reactor, and one to actively cool down the reactor. For heating, steam can be used to heat the recycled water in the cooling circuit, or initiator can be dosed to speed up the reaction, thus increase the heat of reaction. The only actuator for cooling is the feed of fresh water into the cooling circuit. If initiator is not in excess, and the valve for initiator feed is closed, the reaction will of slow down, resulting in less heat from the reaction. From this point of view, the initiator feed can be considered as an actuator for cooling, but the time delay will be significantly large. In reality, the plant might have other methods for controlling the reaction. This could, for example, be by adding inhibitors to the reaction mixture. Consequently, the reaction will slow down, thus lowering the heat of reaction. This is, however, not considered in this project, and can, therefore, not be used as an actuator.

In Section 5.5, the concept of maximum cooling capacity and cooling demand was introduced, making the cooling demand not to exceed the maximum cooling capacity. When implementing the NMPC, this can be included in the objective function by adding a constraint on the cooling demand. When this constraint is included in the prediction, the objective function will penalize for violation of the constraint, and the controller will seek not to perform any action that will lead to violation of the constraint.

Controlling the pressure in a reactor is an important concept, and pressurized systems are usually equipped with a range of safety features, as, for example, pressure-relief valves. However, minimizing the use of such features is desirable. This motivates the implementation of an upper-pressure constraint. The penalty for not satisfying the constraint should be set high enough that no present or future control action will lead to violation. The reactor pressure and temperature are highly correlated, and the reader is advised to Appendix A for further elaboration on the modelling of this relation. Given a certain reactor temperature, the reactor pressure will consequently also be given. The reactor temperature is an important factor for speeding up the reaction. If the goal is to minimize the batch time, the NMPC could be implemented to find an optimal trajectory for the reactor temperature, simultaneously keeping the pressure below some upper constraint. *Optimal trajectory for the reactor temperature* will in the case of the relevant process means the *optimal trajectory for Split Range Controller (SRC) and the initiator feed*, as these are the two actuators.

6.4 Quality parameter control

In Section 5.1 the concept of the K-value, its relation to the conversion average temperature, $\bar{T}_{X_M}$, and the calculation of this variable were introduced. Controlling this parameters are important, as it quantifies the quality of the product.

For the conversion average temperature, only the parameter value at the end of the batch will be of any importance for the final product. That is, the value of the parameter has no significance during the batch, but when the batch is finished, it must satisfy the required setpoint. This can be performed by introducing a setpoint for the conversion average temperature, which only is weighted at the end of the batch. This will, however, require a prediction horizon at least as long as the batch time. If not, there will be a chance that by the time the end of the batch is within the horizon, there will be too late for the controllers to take action to meet the requirement. As already mentioned, too long prediction horizons is computationally demanding, and the solver may take too long finding a solution to the optimization problem.

Alternatively, the setpoint can be weighted at the end of the prediction horizon, which can be defined as long or short as desired. The point at which the deviation from the setpoint is weighted will shift together with the end of the prediction horizon, making this system always meet the requirement at the end of the horizon. When the end of the batch is reached, the states for computing the quality parameter is frozen.

6.5 Initiator feed as actuator and cost of initiator

The initiator feed is an important actuator to minimize the batch time. High initiator feed results in a large number of radicals in the water phase, n_R^w, and hence also the average number of radicals per particle, $\bar{n}$, which contributes to high production rates. The reader is advised to read Appendix A for the modelling of these variables. For cheap initiators, any reduction in the batch time will be economically profitable. For expensive initiators, a trade-off between lowering the batch time and using more initiator most be considered. The initiator feed may, however, be limited by the requirements as not violating the cooling capacity and fulfilling some quality parameter.

A penalty for any additional feed of initiator can be included in the objective function of the optimization problem to reduce the usage of this chemical. This can be performed in numerous ways. One possible solution could be to penalize the waste of initiator in the system, that is the rate of termination in the water phase, R_t^w. The rate of termination was described in Section 2.2.1, and the reader is advised to Appendix A.3.3 for the details on modelling termination in the water phase specifically. The setpoint may be set to zero, and the optimizer will always work towards minimum waste of initiator. An

alternative, and perhaps a more intuitive approach, would be to introduce a variable that tracks the total cost of fed initiator to the system and let the setpoint of this variable be zero. The optimizer will then always work towards minimizing the cost of the initiator. As for the case of minimizing the batch time, it is important not to penalize the setpoint deviation too much. This can cause the batch never to reach the required termination criterion.

Chapter 7

Results and discussion

The purpose of this chapter is to present the results of this project. It aims to demonstrate the concept of estimation and Nonlinear Model Predictive Control (NMPC) by addressing the relevant problems described in Chapter 6. The content of the different sections are listed below, and they will be discussed consecutively.

- Section 7.1 present the results from the finalization of the model. Here, the results from the on-line estimation performed in the preliminary project are made part of the model, and the model outputs are compared to measured data.
- Section 7.2 present the choice of decision variables, setpoints and constraint for this NMPC implementation.
- Section 7.3 present the result from Case Study 1, where the effect of state- and parameter estimation was investigated.
- Section 7.4 present the result from Case Study 2, where a relationship between the total amount of fed initiator and the batch time was established.
- Section 7.5 present the result from Case Study 3, where different lengths of the control and prediction horizon were tested.
- Section 7.6 present the result from Case Study 4, where different horizon parameterization were tested.

Cybernetica's ModelFit was used when validating the finalized model. For all the following case studies, Cybernetica's CENIT was used. The interested reader is advised to Appendix F for a more elaborate description of the software. Prior to simulations presented in this chapter, the model and controllers were tested in CENIT. This, introductory case, and its results can be found in Appendix G.

7.1 Results from the finalization of the model

Some simulations were performed to validate the new aspects of the model, which were described in Chapter 5. The results from modelling the conversion average temperature, including the pre-estimated parameters and modelling the cooling circuit are compared to data from Inovyn.

7.1.1 Conversion average temperature

Figure 7.1 shows the conversion average temperature calculated both by Inovyn and the model, from a recursive simulation. The model developed in this project shows some deviations from Inovyn's calculation in the beginning and at the end of the batch, but throughout the batch, they show a close resemblance to each other.

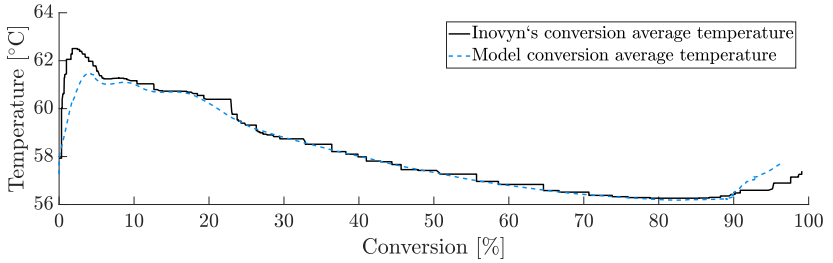


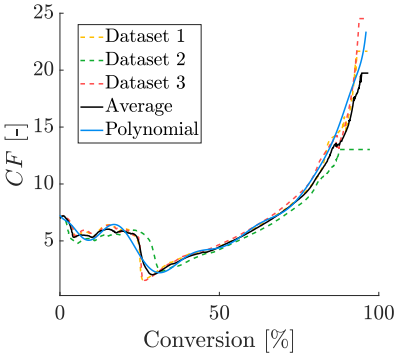
Figure 7.1: A plot showing the conversion average temperature calculated both by Inovyn and the model. The simulation is run recursively.

7.1.2 Unmodelled aspects

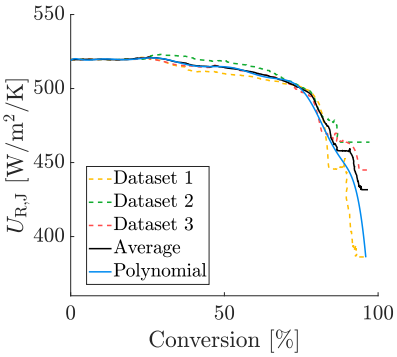
As already emphasized in Section 5.3, some deviations were experienced between measured and modelled output during the preliminary work when the simulation is run ballistic. An example of this is shown in Appendix B. Three batches were, therefore, run recursively, with Kalman Filter (KF), and it was attempted to find some polynomials which would describe the unmodelled aspects. Figure 7.2 shows the estimated parameter profiles of the correction factor, CF, and the heat transfer coefficient, $U_{\mathrm{R,J}}$. It shows the profiles for the three datasets, which were used to find an average, and polynomials were then generated to fit the average. These two polynomials will be considered to describe the unmodelled aspects of the batch.

Three, new, batches were then simulated ballistic with these polynomials as part of the model. This was done to validate the generated polynomials and ensure that they gave satisfactory results also for other, arbitrary, batches. Figure 7.3 shows the modelled reactor temperature for one of these datasets, both when taking into account (b), and not taking into account these pre-estimated profiles (a). Significant improvement can be seen when examining

the results.

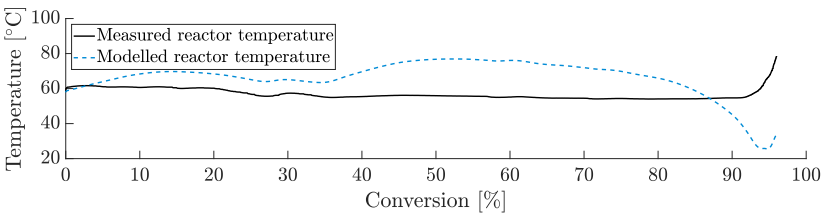


(a) The correction factor, CF, as function of conversion

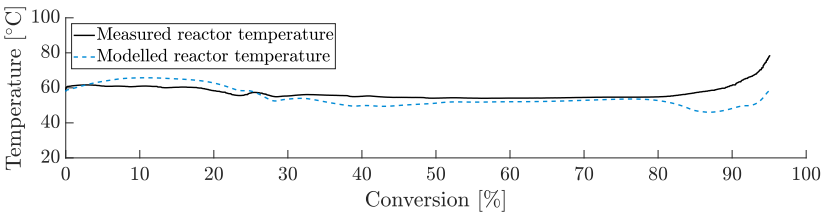


(b) The heat transfer coefficient between the reactor and the jacket, $U_{\mathrm{R,J}}$, as function of conversion

Figure 7.2: The estimated profiles when performing recursive filtering on three datasets during the preliminary work, in addition to the average and the generated polynomials.



(a) Without the pre-estimated profiles



(b) With the pre-estimated profiles

Figure 7.3: Modelled and measured reactor temperature from a ballistic simulation of a new, arbitrary, batch.

7.1.3 Cooling circuit

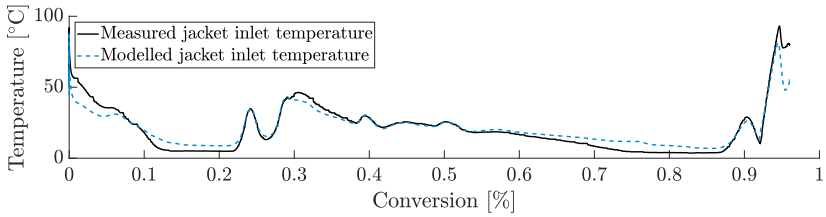
The design parameters for the choke openings, R_{steam} and R_{fresh}, and the time constant for the delay of fresh water, τ_{fresh}, was tuned to fit measured data

by trial-and-error. Their values are summarized in Table 7.1. The table also states the parameter values necessary for the Split Range Controller (SRC) block calculation, provided from the manufacturer. The inlet temperature, $T_{\mathrm{J,in}}$, was then calculated as described in Section 5.2.

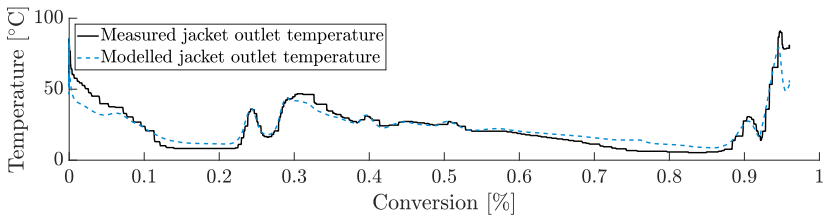
Table 7.1: Design parameters for the valve opening for fresh water and steam, time constant for the delay of fresh water into the cooling circuit, and parameter values necessary for the SRC block calculation.

Parameter	Unit	Value
R_{fresh}		1.0
R_{steam}		1.5
τ_{fresh}	s	167
ν^{min}		0
ν^{max}		1
ν^{*}		0.75
$z^{\mathrm{min}}_{\mathrm{fresh}}$		0
$z^{\mathrm{max}}_{\mathrm{fresh}}$		1
$z^{\mathrm{min}}_{\mathrm{steam}}$		0
$z^{\mathrm{max}}_{\mathrm{steam}}$		1

Figure 7.4 shows the measured and modelled jacket temperatures, for both inlet (a) and outlet (b) of the jacket. The simulation is run ballistic, with the pre-estimated profiles of CF and $U_{\mathrm{R,J}}$. Some deviations can be seen. However, the modelled temperatures generally have a large resembles the measurements.



(a) Inlet jacket temperature



(b) Outlet jacket temperature

Figure 7.4: Jacket temperatures, both measured and modelled temperature. The simulation is run ballistic, with the pre-estimated profiles of CF and $U_{\mathrm{R,J}}$.

7.1.4 Discussion

When controlling this emulsion polymerization reaction, quality parameter set-point must be reached. In this case, the desired quality parameter, the K-value, is a function of the conversion average temperature, $\bar{T}_{X_{\mathrm{M}}}$. This parameter was therefore modelled as described in Section 5.1. Data was provided from Inovyn, and compared to the model in Figure 7.1. When comparing the model and Inovyn's calculation, some deviation could be seen at the start and the end of the batch. For a recursive simulation without the pre-estimated profiles, the modelled and measured reactor temperature almost perfectly matches (As shown in Figure B.1). This means that the reactor temperature most likely is not the cause of the deviation in the conversion average temperatures. This indicates that there is some deviation in the modelled production rate, and the production rate calculated by Inovyn. This parameter is not easily measurable, and reasoning which calculation would be the best reflection of reality is therefore difficult. As measurement cannot be acquired, and the deviations in conversion average temperature not are severe, the modelling of this parameter is considered sufficient for control purpose and for this project.

Figure 7.2 illustrates that all three datasets show the same trends for the estimation of the correction factor, CF, and the heat transfer coefficient between the reactor and the jacket, $U_{\mathrm{R,J}}$. The rapid change towards the end can be due to a decreasing concentration of monomer inside the particles during Interval III of the process. The KF try to compensate for this by increasing the correction factor and lowering the heat transfer coefficient. An increasing correction factor causes higher propagation rate, which leads to more heat of reaction and hence also high reactor temperature. Lowering the heat transfer coefficient causes less efficient cooling, which also contributes to higher reactor temperature. Higher reactor temperature will, in its turn contribute to higher production rate. The decreasing heat transfer coefficient can also be explained by increasing viscosity of the reaction mixture, causes the heat transfer to be less efficient.

Some deviation between the estimated profiles for the three datasets can be seen, particularly towards the end of the batch. During the preliminary project, input data indicated when the polymerization stage was ending, and the estimation is switched off. However, the model keeps the polymerization reaction going until the termination criteria are reached. Batches where the input indicates that the polymerization stage finishes early, will not have time to reach such high correction factor, or low heat transfer coefficient, as other batches. Another reason can be due to model error, which the KF tries to compensate for. Different profiles can then be due to different magnitudes of the model error. Some deviations between the three datasets are, however, to be expected due to disturbances and variations from one batch to another.

However, the resemblance between the estimated profiles for the three batches is evident. An average of these profiles will, therefore, be a well-suited description of the unmodelled aspects. During the initial phase of this project, average profiles of the two estimated parameters were found, followed by finding

polynomials describing these averages. The purpose of finding these polynomials was to implement this in the application easily. They could not be functions of time, as this would cause a mismatch when simulating batches of different duration. The polynomials are therefore functions of conversion. In addition to the three datasets, Figure 7.2 shows the averages and the fitted polynomials. The polynomials were, by inspection, considered valid approximations of the averages.

The mismatch between measured and the modelled temperature is expected to be smaller for the case with these polynomials in the model, compared to without. This was shown by the ballistic simulation of the reactor temperature in Figure 7.3, where the modelled temperature in Figure 7.3b shows notable smaller mismatch to the measured temperature, than Figure 7.3a. The improvement is not as significant as the recursive simulation performed during the preliminary work (Figure B.1), nor can this be expected. There will always be some deviation in operation, and there will be disturbances from one batch to the other. An average or a fitted curve will not necessarily be representative for all possible batches. However, it was decided that the polynomials gave a large improvement of the modelled outputs, and they were therefore implemented and accepted to represent the unmodelled aspects of the system.

During the preliminary project, the inlet jacket temperature was used as input directly. When validating the model, this was a good approach to filter out any error that could have been added through modelling the inlet jacket temperature. For control purposes, however, it would be unrealistic to choose the inlet jacket temperature as an actuator. A model describing the cooling circuit was therefore derived, as described in Section 5.2, with the pre-estimated profiles. The new model was tested on a new, arbitrary batch for verification. The results were presented by Figure 7.4, and shows that the model output has a considerable resemblance to the measured output. Some deviations can be seen throughout the batch. The deviations at the beginning of the batch could be due to the initialization of the batch, which is set to be equal from one simulation to another. Alternatively, the deviations can be explained by the same reasoning as for the reactor temperature in Figure 7.3. The pre-estimated profiles will result in some error in the model due to variations from one batch to another. Another reason for the deviation can be that a delay in the effect of heating the recycled water, is not included in the model. This can theoretically be done in the same manner as for the fresh water feed. It was however decided not to include this feature as it would introduce an additional state to the system, and the yield of this introduction would be small. In addition, steam is used much less frequently than the fresh water feed, as the heat of reaction itself often is sufficient for heating. As penalization for any control move is implemented, the system will work to minimize the use of the actuator. Overall, the values of the design parameter for the valve opening for fresh water and steam, in addition to the time constant for the delay of fresh water into the cooling circuit (Table 7.1), gave satisfactory result with respect to the jacket temperatures.

Some aspects were left unmodelled during this work for several reasons. The model should, however, capture the most critical dynamics of the system. When modelling, there is almost no limit to how complex the model can be made. A more complex model will often lead to more accurate results if performed by a skilled and knowledgeable modeller. For a NMPC application with a state- and parameter estimator, this means not as dramatic updates, as for a less perfect model. However, a higher complexity will require more computational power, and a trade-off between simplicity and complexity had to be found.

One of the unmodelled aspects is the initiator system, which was modelled as a thermal system, but in reality, it is a combined redox-thermal system. In addition, perfect mixture and equilibrium between phases at all time is assumed. However, establishing equilibrium will have some time delay. It can be limited by mass transfer, and a time constant can be introduced to capture this delay. The procedure would then be the same as for the delay in fresh water feed, as described in Section 5.5. This will introduce additional states to the system, and also affect the pressure in the reactor.

The actuator for the initiator feed was set to be in mass units directly. However, controlling the valve position would probably be closer to reality. This would then require an equation describing the relationship between valve position and the feed of initiator. This approach would be similar to the modelling of the fresh water feed in Section 5.5. It would, however, require more information about the valve position from measured data. Some delay in pipelines may also be realistic.

The particles were assumed to be monodisperse, and the rate of adsorption and desorption were modelled independently of particle size. The size of the particles can be found easily enough if monodispersity is assumed, but to further model the dependency of the kinetic parameters on the particle size based on first principles would require a much more elaborate description of the system. This includes, for example, information about diffusion coefficients.

Both termination and propagation were modelled independently of chain length. If this was to be included in the model, a full dynamic population balance had to be done over the radials. Furthermore, the dependency of chain length on termination and propagation had to be found, as the rate constants. The result would be a large number of rate constants, which had to be found experimentally, or by estimation. Either approach would be very challenging, and probably a reason why chain length dependency rarely is included when modelling such systems.

The features just mentioned are only some of the aspects which are left unmodelled. If some were implemented successfully, CF and $U_{\mathrm{R,J}}$ could be re-estimated, new polynomials could be found, and a more accurate model may be obtained. Some of the features will, however, be more easily implemented than others. In addition, the complexity of the model will increase, and so will the computational load.

7.2 Implementation of the NMPC

When building the objective function, the decision variable and their setpoints and constraints were mainly obtained from engineering intuition or demands from the manufacturer.

At the plant, the operators are notified when the reactor pressure exceeds 10 barg. The upper constraint for the pressure was therefore set to 10 barg. The lower limit is set to -1 barg. Even though -1 barg is an unrealistic value for a pressurized tank during polymerization, process indicators logged negative gauge pressures during loading and drainage. Setting the lower limit to -1 barg accounts for this singularity.

The lower limit for the cooling demand was set to the maximum cooling capacity, as described in Section 5.5 and reasoned in Section 6.3. The upper limit was set unrealistically high, making sure that the upper boundary not will be active when the jacket is used for heating, and the cooling demand is positive. This would be the case when the temperature in the reactor is lower than the jacket, at which the recycled stream is heated with steam.

A setpoint for the production rate was specified and set unrealistically high for the optimizer always working towards maximizing the production rate and thereby minimizing the batch time. The duration of an average batch lies between 6 and 7 hours, and minimizing the batch time directly would require a prediction horizon at least as long. This was attempted, but it became clear that the optimizer could not find a solution in between samples. The measurements are logged every 10 seconds, and the optimizer could, on average, not find a solution within this time frame. Note that the coincidence points for R_p after the batch is finished, is removed. This was done for the controller not aiming to increase the production rate when the termination point is reached, as this would cause the batch never to end.

A setpoint for the monomer conversion was specified, and set to 100%. This will also contribute to minimizing the batch time, but as the system approaches 100% conversion, the deviation, and hence also the penalty for the deviation, will decrease. It was therefore decided to have setpoints both for the production rate and the monomer conversion. Note that even though the setpoint is at 100% conversion, the termination criteria at 96%.

The conversion average temperature is the quality parameter used for this reaction, which by the manufacturer is determined to be 57°C. After ensuring safe operation, meeting this requirement is considered the most important. As it is only the end-value of this parameter that is of any significant meaning, this decision variable will only have one coincidence point, which will be located at the end of the prediction horizon. Weighting setpoint at the end of a shorter prediction horizon ensures that not any action that can cause the quality parameter not being able to reach its setpoint. At least if it is weighted properly.

A setpoint was added to the waste of initiator, that is the rate of termination in the water phase: R_t^w. The reader is advised to read Appendix A.3.3 for an

elaboration on this subject. Its respective setpoint was set to zero. This is not a realistic value, as it would cause no feed of initiator, and the reaction would die out rather quickly in lack of radicals. It will, however, work towards minimizing the use of initiator, as discussed in Section 6.5. Note that an alternative would be to track the amount of initiator fed to the reactor, or the cost of fed initiator. However, it was decided to choose $R_{\mathrm{t}}^{\mathrm{w}}$ as the controlled variable as it will capture the dynamics of the kinetic model. For example, kinetics will make feeding initiator some points during a batch more efficient than others.

All outputs, except the conversion average temperature, has identical parameterization for the coincidence points. Some coincidence point for the production rate is also deleted consecutively. The inputs are also parameterized identically by input blocking. The parameter values for the SRC block calculation is as stated in Table 7.1. The lower limit for the feed of initiator is 0, and the upper limit was set to 10 g/s.

The setpoints and constraints mentioned in this section are summarized below. Unless something else is explicitly stated, these are the decision variables and their respective setpoints and constraints, in the optimization problem.

Pressure control: An upper and lower limit of the pressure was specified; $p_{\mathrm{R}}^{\mathrm{high}} = 10$ barg, $p_{\mathrm{R}}^{\mathrm{low}} = -1$ barg.

Cooling demand control: An lower limit of the cooling demand was specified; $\Delta H^{\text{cooling demand, low}} = \Delta H^{\text{cooling capacity}}$.

Production rate control: The setpoint was set unrealiticly high, for the NMPC to work towards maximizing the production rate: $R_{\mathrm{p}}^{\mathrm{sp}} = \text{high}$, with coincidence points only before the termination criteria.

Monomer conversion control: A setpoint was defined for the monomer conversion: $X_{\mathrm{M}}^{\mathrm{sp}} = 100\%$.

Conversion average temperature control: A setpoint was defined for the conversion average temperature: $\bar{T}_{X_{\mathrm{M}}}^{\mathrm{sp}} = 57°\mathrm{C}$. Note that for all simulations, the $\bar{T}_{X_{\mathrm{M}}}$ is only weighted at the end of the prediction horizon, by one coincidence point.

Initiator feed control: The setpoint for the waste of initiator was set to zero: $R_{\mathrm{t}}^{\mathrm{w,sp}} = 0$.

Constraints for the SRC: Lower and upper limit for the SRC: $\nu^{\min} = 0$, $\nu^{\max} = 1$.

Constraints for the feed rate of initiator: Lower and upper limit for the feed rate of initiator: $\hat{m}_{\mathrm{I}_1}^{min} = 0$, $\hat{m}_{\mathrm{I}_1}^{max} = 10$ g/s.

7.3 Case Study 1: Effect of state- and parameter estimation

A simulation without state- and parameter estimation was performed. This was done to establish an opinion about the need for including the KF. During this simulation, the same features as listed in Section 7.2 were included in the optimization problem.

The aspects of the system which is considered unmodelled (See Section 5.3), is only included in the simulator, and not in the estimator. This makes the model used in the simulator, and the model used in the controller, dissimilar. A large deviation in the outputs between the simulator and the estimator is, therefore, to be expected, especially when no state- and parameter estimation is implemented. Two separate case studies will, therefore, be performed to study the effect of the KF. The cases are listed below, and they will be presented systematically.

Case Study 1.1: Pre-estimated profiles were included in the simulator, but not in the prediction. State- and parameter estimation not active.

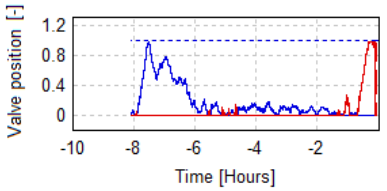
Case Study 1.2: Pre-estimated profiles were included in the simulator, and also in the prediction. State- and parameter estimation active.

The goal of Case Study 1 was to confirm the bias between the model in the estimator and simulator, and then to tune the Kalman filter for smooth estimation. This will show the effect of the Kalman Filter, utilized in Case Study 1.2.

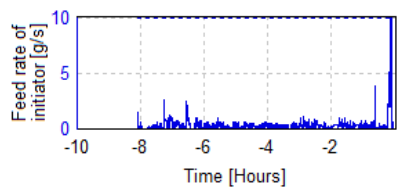
7.3.1 Case Study 1.1: Inactive Kalman Filter

Figure 7.5 - 7.8 show the results from Case Study 1.1. The KF was not active during this case study. Hence, neither state or parameter estimation were performed on-line. The pre-estimated profiles were not included in the prediction.

Figure 7.5a shows the valve positions for the feed of fresh water and steam, from the internal input signal (the SRC) calculated by CENIT, and their upper constraint. The valve position of fresh water feed increases rapidly at the beginning of the batch and decays throughout the batch. Steam is mainly utilized towards the end. Figure 7.5b shows the feed of initiator to the system, calculated by CENIT. The feed of initiator is quite steady during the batch before it increases rapidly towards the end.



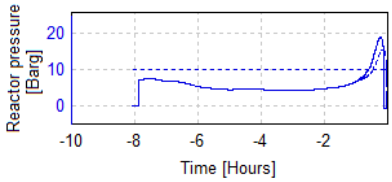
(a) Valve position for fresh water (blue) and steam (red), in addition to their upper constraint (dotted)



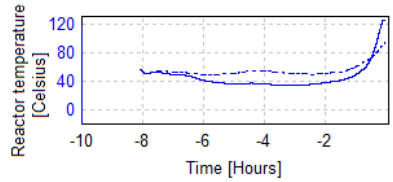
(b) Feed rate of initiator

Figure 7.5: Inputs calculated by CENIT, with inactive Kalman Filter, and pre-estimated profiles not in prediction.

Figure 7.6a shows the reactor pressure calculated by the simulator (solid) and the estimator (dashed), in addition to the upper constraint (dotted). As can be seen, both the simulator and the estimator violates the upper constraint. At the real plant, such high pressure could not be obtained as the operators would take action by venting or adding inhibitors. Alternatively, would other safety feature been activated before the pressure reaches such high values. Figure 7.6b shows the reactor temperature calculated by the simulator (solid) and the estimator (dashed). As can be seen, the simulated temperature reaches values far over 100°C. When analyzing logged data from the plant, the reactor temperature rarely goes above 90°C. The high reactor temperature is correlated to the high reactor pressure.



(a) Simulated (solid), estimated (dashed) reactor pressure and upper-constraint (dotted)

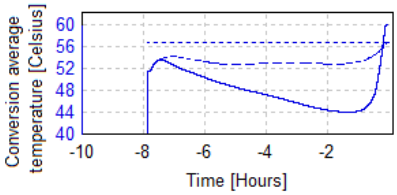


(b) Simulated (solid) and estimated (dashed) reactor temperature

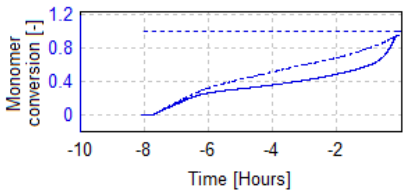
Figure 7.6: Reactor pressure and temperature, from simulation with inactive Kalman Filter, and pre-estimated profiles not in prediction.

Figure 7.7a shows the conversion average temperature calculated by the simulator (solid), the estimator (dashed) and its setpoint (dotted). As can be seen, only the estimator reaches the setpoint of 57°C, and the fluctuations are much smaller for the estimated profile. Figure 7.7b shows the monomer conversion calculated by the simulator (solid) and the estimator (dashed), both

reaching the termination criteria of 96%, approximately at the same time.



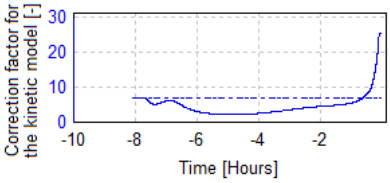
(a) Simulated (solid), estimated (dashed) conversion average temperature and its setpoint (dotted)



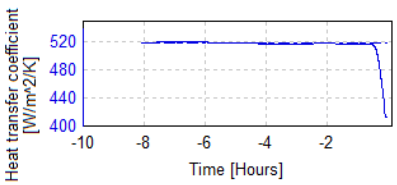
(b) Simulated (solid), estimated (dashed) monomer conversion and its setpoint (dotted)

Figure 7.7: Conversions, from simulation with inactive Kalman Filter, and pre-estimated profiles not in prediction.

Figure 7.8a shows the correction factor calculated by the simulator (solid) and the estimator (dashed). As can be seen, the correction factor for the simulator follows the polynomial shown in Figure 7.2. The correction factor for the estimator is constant due to no model updates. Figure 7.8b shows the heat transfer coefficient calculated by the simulator (solid) and the estimator (dashed). Again, the heat transfer coefficient for the simulator follows the polynomial shown in Figure 7.2, while the estimated value is constant.



(a) Simulated (solid) and estimated (dashed) correction factor for the kinetic model



(b) Simulated (solid) and estimated (dashed) heat transfer coefficient between reactor and jacket

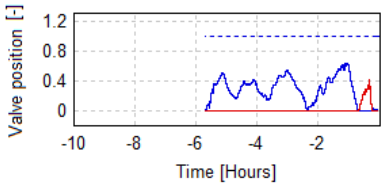
Figure 7.8: Parameters, from simulation with inactive Kalman Filter, and pre-estimated profiles not in prediction.

7.3.2 Case Study 1.2: Active Kalman Filter

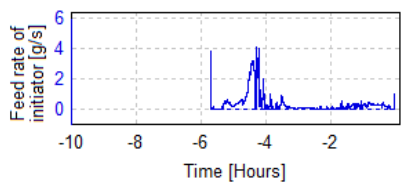
In this case study, the KF was active. Hence, state- and parameter estimation were performed on-line. The pre-estimated profiles for the unmodelled aspects were included in the prediction. The reactor temperature, T_{R}, and the outlet jacket temperature, $T_{\mathrm{J,out}}$, were used as active measurements, while the esti-

mated parameters were the correction factor for the kinetic model, CF, and the heat transfer coefficient between the reactor and the jacket, $U_{R,J}$. The relevant results are shown in Figure 7.9 - 7.12. Note that both simulated and estimated calculations are shown for the relevant outputs (pressure, temperatures, conversions). However, compared to Case Study 1.1, they almost perfectly coincide, and the difference between them is therefore not visible.

Figure 7.9a shows the valve position of the feed of fresh water and steam, in addition to their upper constraint. Note that the valve positions are calculated from the actual manipulated input, the SRC (not shown), by CENIT. Steam is almost never utilized, except at the very end of the batch. Figure 7.9b shows the feed for initiator. The feed of initiator increases early in the batch, before it decays and remains approximately steady throughout the batch.



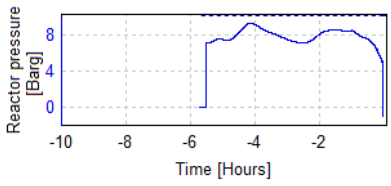
(a) Valve position for fresh water (blue) and steam (red), in addition to their upper constraint (dotted)



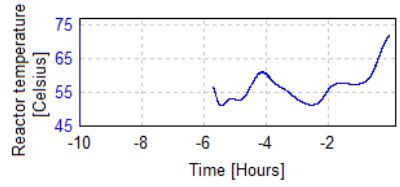
(b) Feed rate of initiator

Figure 7.9: Inputs calculated by CENIT, from simulation with active Kalman Filter, and pre-estimated profiles in prediction.

Figure 7.10a shows the reactor pressure and its upper constraint, which was not violated. Figure 7.10b shows the reactor temperature, which fluctuates a bit through the batch, before increasing considerably at the end.



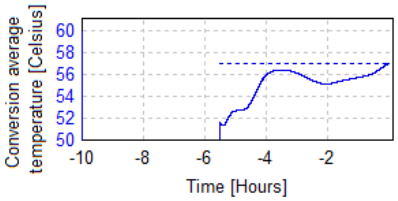
(a) Simulated (solid), estimated (dashed) reactor pressure and upper-constraint (dotted)



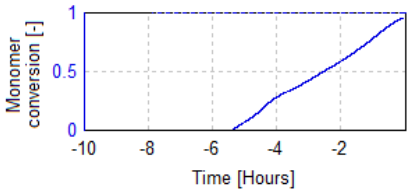
(b) Simulated (solid) and estimated (dashed) reactor temperature

Figure 7.10: Reactor pressure and temperature, from simulation with active Kalman Filter, and pre-estimated profiles in prediction.

Figure 7.11a shows the conversion average temperature which reaches the setpoint (dotted) of 57°C at the end of the batch. Figure 7.11b shows the monomer conversion, which reaches the termination criteria of 96% at the end.



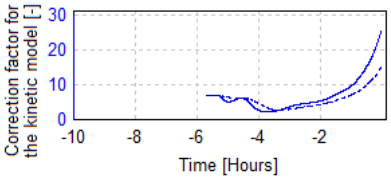
(a) Simulated (solid) estimated (dashed) conversion average temperature, and its setpoint (dotted)



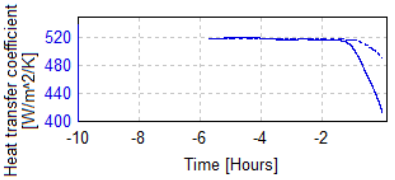
(b) Simulated (solid), estimated (dashed) monomer conversion and its setpoint (dotted)

Figure 7.11: Conversions, from the simulation with active Kalman Filter, and pre-estimated profiles in prediction.

Figure 7.12a shows the correction factor calculated by the simulator (solid) and the estimator (dashed). As can be seen, the correction factor for the simulator follows the polynomial shown in Figure 7.2. The correction factor estimated by the KF follows the same trend as the simulator but does not reach as high values towards the end. Figure 7.12b shows the heat transfer coefficient calculated by the simulator (solid) and the estimator (dashed). Again, the heat transfer coefficient for the simulator follows the polynomial shown in Figure 7.2, while the estimated value follows the same trend as the simulator, but does not reach as low values towards the end.



(a) Simulated (solid) and estimated (dotted) correction factor for the kinetic model



(b) Simulated (solid) and estimated (dotted) heat transfer coefficient between reactor and jacket

Figure 7.12: Parameters, from simulation with active Kalman Filter, and pre-estimated profiles in prediction.

7.3.3 Discussion

Case Study 1.1, showed that there was a significant need for implementing a state- and parameter estimator. This was an expected result as the simulator and estimator deliberately had different models integrated. If the models were similar, however, there would be no need for estimation, and the output from the simulator and the estimation would yield the same results.

The observant reader may have noticed the resemblance between the simulated reactor temperature in Figure 7.6b, and the evolution of the pre-estimated profile of the correction factor for the kinetic model in Figure 7.2. This is a result of the reactor temperature being very dependent on the rate constant of many of the kinetic parameter ($k_{\mathrm{i}}, k_{\mathrm{d}}, k_{\mathrm{p}}, k_{\mathrm{t}}$), which will all vary linearly with the correction factor. In the first approximately 3 hours of the batch, the reactor temperature in the simulator experience a decrease due to the correction factor. The temperature in the estimator does not experience the same and is on the correct course to reach the setpoint for the conversion average temperature (Figure 7.7a). The controller will therefore not perform any action to adjust the temperature, at least not dramatically. As a result, the conversion average temperature decreases in the simulator, and the monomer conversion at this point almost flattens out (Figure 7.7b).

Figure 7.6a shows that the pressure constraint was violated for the case where the KF was inactive. In addition, the reactor temperature shows much higher values than any of the logged material from previous batches provided from Inovyn. The batch was re-simulated with larger penalties for violating the pressure constraint, without any significant improvement. This can be explained by the following reasoning; Figure 7.7b shows that the estimator is approaching the termination criteria (96% monomer conversion) at about -1.5 hour, and at this point, it will be within the prediction horizon. To obtain the final temperature increase to reach the setpoint of the conversion average temperature (Figure 7.7a), it increases the temperature by using steam and keep on feeding initiator. However, the correction factor for the kinetic model rapidly increases to high values in this case for the simulator (Figure 7.8a), which causes a large temperature increase. This is evident from Figure 7.6b, where the simulator calculates very high temperatures. Consequently, the reactor pressure also increases.

The steep increase for Case Study 1.1 in the monomer conversion and the conversion average temperature for the simulator is a result of this increase in temperature. The NMPC aims for the estimator to obtain its required constraints and setpoints, and the off-set in the simulator may, therefore, be significant when the deviation between the model for the estimator and simulator is this large.

Overall, the results from Case Study 1.1 emphasized the need for state- and parameter estimation, and the results from Case Study 1.2 showed significant improvements after this implementation. The estimated parameters from Case Study 1.2 (Figure 7.12), did not perfectly coincide with their simulated values, which would be the idealized case. A reason for the deviation can be that the

KF was not tuned aggressively enough. Other tuning values could have been explored to minimize this deviation, which could yield improved prediction. However, the results were considered satisfactory.

7.4 Case Study 2: Initiator feed versus batch time

An interesting aspect of this polymerization process is the relationship between the amount of necessary initiator, and the batch time. Several simulations were therefore performed with the objective to establish this relationship. If such a relation is successfully found, one could go back and check what amount of initiator is necessary to achieve a certain batch time. For a cheap initiator where any reduction in the batch time would be profitable, finding the maximal amount of initiator that still would ensure safe operation and satisfy quality parameter, is desirable. During this case study, the same features as listed in Section 7.2 were included in the optimization problem.

Figure 7.13 shows the data points that were obtained for a prediction horizon of 1000 samples, and a control horizon of 200 samples. For the prediction horizon, coincidence points were chosen every 50 samples. For the control horizon, inputs were parameterized with three parameters, which determines the input at 0, 50, 100 and 200 samples. The results indicate an inverse exponential, or almost linear, relationship.

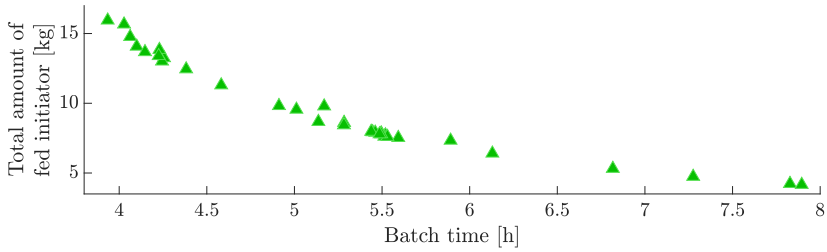


Figure 7.13: Initiator feed versus batch time. The prediction horizon for this simulation was 1000 samples (coincidence points every 50 sample), and the control horizon 200 samples (inputs determined at 0, 50, 100 and 200 samples).

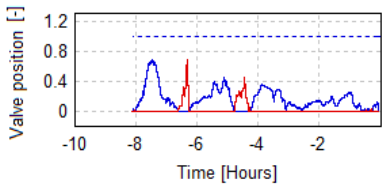
7.4.1 Example simulations

The results from some selected data points will now be presented. This is to illustrate the different operation for some batches shown in Figure 7.13. One batch will have a very long batch time, while the other has significant lower batch time.

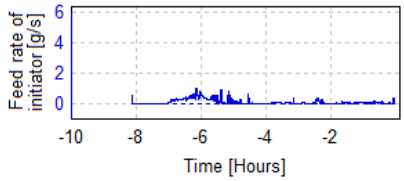
Case Study 2.1: Long batch time

Figure 7.14 shows the results from a simulation with a long batch time. Both fresh water and steam are utilized during the batch, illustrated in Figure 7.14a. Only a small amount of initiator is fed, which is shown in Figure 7.14b. Initiator is mainly fed early in the batch batch, approximately at -6 hours. In total 4167 g, and the batch duration was 7.9 hours. Figure 7.14c shows that

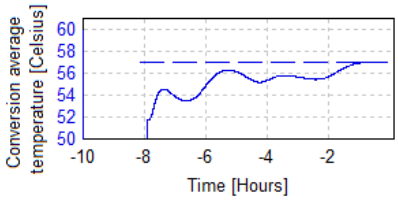
the conversion average temperature (solid) reaches its setpoint (dashed). The conversion average temperature is kept below its setpoint during the whole batch. Figure 7.14d shows the reactor temperature, which varies throughout the batch and decreases towards the end. The average number of radicals per particle is kept within a value of approximately 0 and 0.1, shown by Figure 7.14e. Figure 7.14f shows that the cooling demand (solid) does not exceed the cooling capacity (dotted).



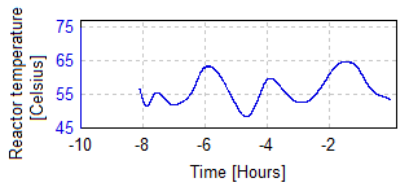
(a) Valve position for fresh water (blue) and steam (red), in addition to their upper constraint (dotted)



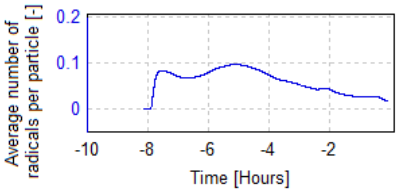
(b) Feed rate of initiator



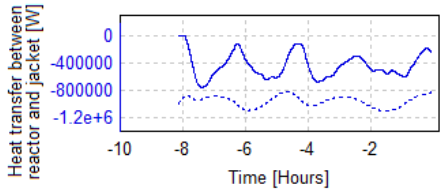
(c) Conversion average temperature (solid), and setpoint (dashed)



(d) Reactor temperature



(e) Average number of radicals per particle, $\bar{n}$



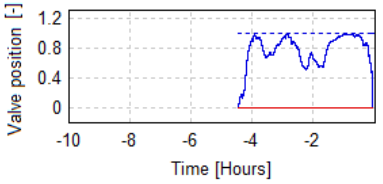
(f) Cooling efficiency (solid), and maximum cooling capacity (dotted)

Figure 7.14: Results from simulation of a batch with a long batch time (Case Study 2.1).

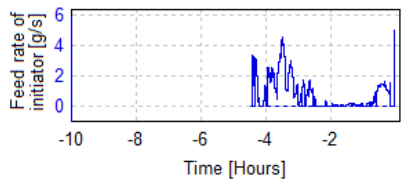
Case Study 2.2: Short batch time

Figure 7.15 show the results from a simulation with a short batch time. Only fresh water is utilized during the batch, as illustrated in Figure 7.15a. A large

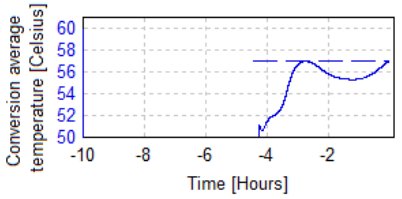
amount of initiator is fed, which is shown in Figure 7.15b. Initiator is mainly fed early in the batch, approximately at -4 hours. In total, 13399 g of initiator was fed, and the batch duration was 4.2 hours. Figure 7.15c shows that the conversion average temperature (solid) reaches its setpoint (dashed). Figure 7.15d shows the reactor temperature, which varies throughout the batch, increasing at the end. The average number of radicals per particle is kept within a value of approximately 0 and 0.15, shown by Figure 7.15e. Figure 7.15f shows that the cooling demand (solid) does not exceed the cooling capacity (dotted). A much smaller cooling margin, that is, the gap between the cooling efficiency and the maximum cooling capacity, can be observed for this case in comparison to Case Study 2.1.



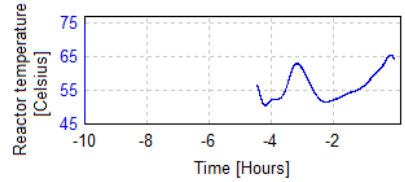
(a) Valve position for fresh water (blue) and steam (red), in addition to their upper constraint (dotted)



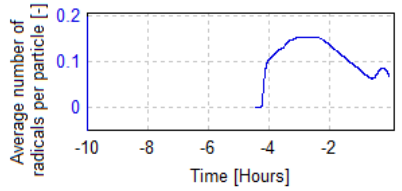
(b) Feed rate of initiator



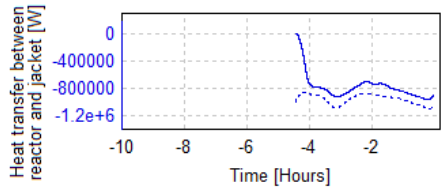
(c) Conversion average temperature (solid), and setpoint (dashed)



(d) Reactor temperature



(e) Average number of radicals per particle, $\bar{n}$



(f) Cooling efficiency (solid), and maximum cooling capacity (dotted)

Figure 7.15: Results from simulation of a batch with a short batch time (Case Study 2.2).

7.4.2 Discussion

As already stated, Figure 7.13 showed an inversely exponential, or possibly linear, relation between the total amount of initiator fed to the reactor and the batch time. For an inverse exponential function, when operating conditions are such that batch time gets very shorts, the effect of adding additional initiator will be small. When the operating condition is such that the batch time gets very long, the effect of adding additional initiator will be large. From a quantitative perspective, this might be the case for an inverse exponential relation. Examining the data point at the far right, only 2.2 kg of initiator, will reduce the batch time by 1.70 hours. To obtain the same amount of reduction in batch time from this point, another 6.04 kg of initiator is needed.

The relationship between the total amount of initiator added and the batch time is reasonable in the sense that adding more initiator, naturally will lead to shorter batch times. More initiator fed to the system means higher reaction rates and a greater amount of initiator decomposes to radicals. This results in a larger amount of radicals in the water phase. This will increase the rate of termination in the water phase, but also the rate at which radicals enter the particles. Once a larger number of radicals have entered the particles, the termination in the polymer phase will increase, but so will the rate of propagation. Overall, the production rate will increase when a larger amount of initiator is added.

There is also a qualitative argument that can be made to reason for an inverse exponential relation. When studying the kinetic expressions for the propagation and termination rate, which was established in the Section 2.2.1, the rate of propagation is linearly dependent on the total amount of radicals in the system by Equation 2.2.8, while the termination is quadratically dependent by Equation 2.2.12. Summarized, the following relations hold;

$$R_{\mathrm{p}} \propto \bar{n}$$
$$R_{\mathrm{t}} \propto \bar{n}^2$$

The purpose of emphasizing these relations is that for higher values of $\bar{n}$, termination reactions will be more prominent, and high values of $\bar{n}$ are obtained by large amounts of fed initiator. In other words, the addition of initiator has the purpose of speeding up the reaction, but at a certain point, the termination reactions will become more dominant. This makes the addition of more initiator less effective for large values of $\bar{n}$.

In both cases, a large portion of initiator is fed early in the batch. This can be reasoned by the nucleation of particles, which result in an increasing production rate. The theoretical aspects of this were introduces in Section 2.3.1. As a result, a larger amount of initiator is required, leading to a high feed rate. The increment in initiator feed is, however, not in the very beginning. This can be justified by pre-dosing of both Initiator 1 and 2. Initiator will at this point be in excess, and there is not necessary to add initiator to the system immediately.

When comparing the cooling efficiency for Case Study 2.1 and Case Study 2.2, by examining Figure 7.14f and Figure 7.15f, it is obvious that the cooling capacity is better utilized for the shortest batch. The cooling margin is much smaller for Case Study 2.2. It is, however, never violated. The valve position of fresh water is more stable and even saturates at some points. Steam is nor utilized, and all of this leads to fewer fluctuations in reactor temperature. Hence, the reactor pressure and the conversion average temperature will also have more stable profiles. The feed of initiator is higher for Case Study 2.2, resulting in a larger values of $\bar{n}$. This increases the production rate and reduces the batch time. During the last phase of the batch, the reactor temperature raises for the short batch, while decreases for the longest batch. From an industrial perspective, it is desirable for the reactor to obtain a high temperature towards the end. Vinyl Chloride Monomer (VCM) is toxic, and high reactor temperature results in the residual VCM being burned off.

At the real plant, there are several different reasons why it would be unrealistic to add up to 16 kilos of initiator. Firstly, there is the safety aspect. Even though the NMPC is implemented to stay under some pressure boundary and to ensure enough cooling capacity at all time, all scenarios can not possibly be foreseen or modelled. 16 kilos of initiator is almost four times as much as a typical batch is operating under today at the plant. If the application were to be tested in industry, a less aggressive tuning would be a more realistic starting point. It is important to remember that the simulator uses a plant-replacement model, which never could resemble the real plant perfectly.

7.5 Case Study 3: Length of horizons

This case study was performed to investigate the possibility of a different relationship between the amount of initiator and batch time for different horizons. Table 7.2 shows the different horizon parameterization that was examined during this case study. The numbers in parenthesis indicate the parameterization. Note that one sample, in this case, is equivalent to 10 seconds. Some example simulations from these case studies will be presented afterwards.

Table 7.2: Length of prediction horizon and control horizon for all the different case studies simulated in Section 7.5. Numbers in parenthesis indicates the parameterization. Both horizons is given in number of samples.

	Control horizon	Prediction horizon
Case Study 3.1	200 (0, 50, 100, 200)	750 (every 50)
Case Study 3.2	200 (0, 50, 100, 200)	1000 (every 50)
Case Study 3.3	400 (0, 50, 100, 200, 400)	1500 (every 50)

Figure 7.16 shows the relationship established between the total amount of initiator fed to the reactor. The different data points for each case study were obtained only by adjusting the weights in the objective function of the optimization problem. The figure shows that for a specific batch time, approximately the same amount of initiator is needed, independently of the length of the horizon. This can be seen, for example, at a batch time of 5.5 hours, which corresponds to approximately 8 kilos of initiator, for all three cases.

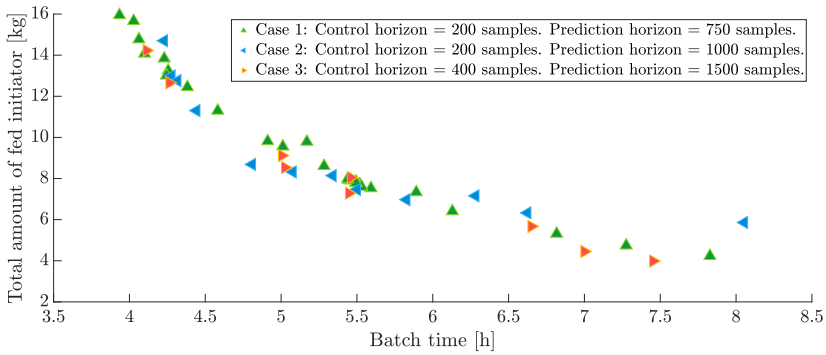


Figure 7.16: Initiator feed versus batch time, for the cases described by Table 7.2.

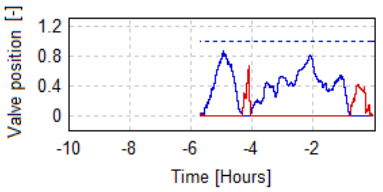
7.5.1 Example Simulations 1: Similar tuning

The results from some selected data point will now be presented. All three case studies were simulated with the same tuning, except different control and prediction horizons. Only the most important results will be presented; valve

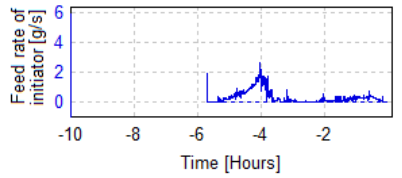
positions, feed of initiator, reactor temperature, conversion average temperature and batch time.

Case Study 3.1.1: $M = 200$, $N = 750$

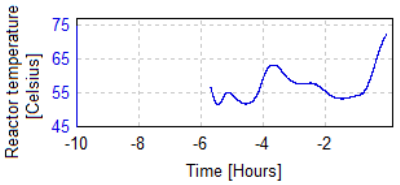
This case study has the same control and prediction horizons as described for Case Study 3.1 in Table 7.2. Figure 7.17 shows the main results from Case Study 3.1.1. The valve positions (Figure 7.17a), the feed of initiator (Figure 7.17b), the reactor temperature (Figure 7.17c) and the conversion average temperature (Figure 7.17d). In total 7199 g of initiator was used, and the duration of the batch was approximately 5.47 hours. As can be seen, mainly fresh water is utilized, in addition to some steam. Initiator is fed mainly early in the batch, and a low feed rate can also be observed throughout the batch. The reactor temperature varies through the batch and increases towards the end. As a result, the conversion average temperature also increases towards the end, reaching its setpoint.



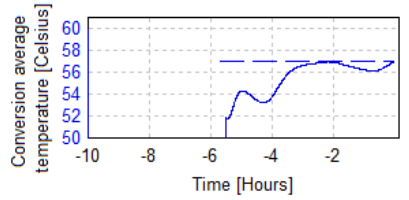
(a) Valve position for fresh water (blue) and steam (red), in addition to their upper constraint (dotted)



(b) Feed rate of initiator



(c) Reactor temperature



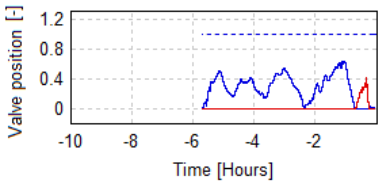
(d) Conversion average temperature (solid) and its setpoint (dashed)

Figure 7.17: Results from Case Study 3.1.1.

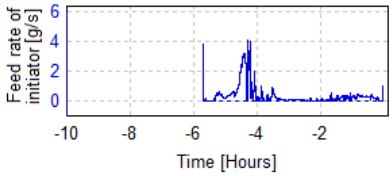
Case Study 3.2.1: $M = 200$, $N = 1000$

This case study has the same control and prediction horizons as described for Case Study 3.2 in Table 7.2. Figure 7.18 shows the main results from Case Study 3.2.1. The valve positions (Figure 7.18a), the feed of initiator (Figure

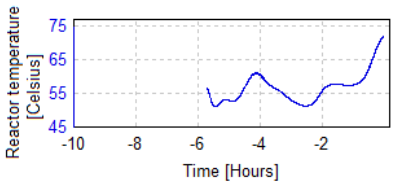
7.18b), the reactor temperature (Figure 7.18c) and the conversion average temperature (Figure 7.18d). In total 7791 g of initiator was used, and the duration of the batch was approximately 5.40 hours. As can be seen, mainly fresh water is utilized, in addition to some steam towards the end of the batch. Initiator is fed mainly early in the batch, and a low feed rate can also be observed throughout the batch. The reactor temperature varies through the batch and increases towards the end. As a result, the conversion average temperature also increases towards the end, reaching its setpoint.



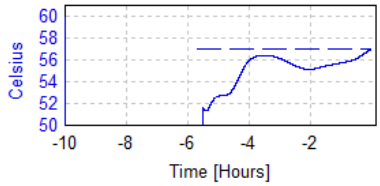
(a) Valve position for fresh water (blue) and steam (red), in addition to their upper constraint (dotted)



(b) Feed rate of initiator



(c) Reactor temperature

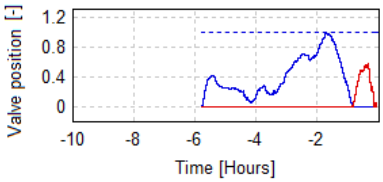


(d) Conversion average temperature (solid) and its setpoint (dashed)

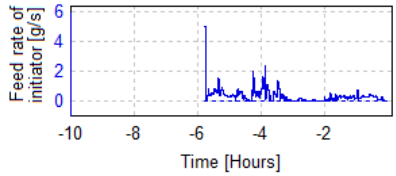
Figure 7.18: Results from Case Study 3.2.1.

Case Study 3.3.1: $M = 400$, $N = 1500$

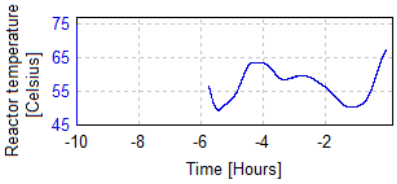
This case study has the same control and prediction horizons as described for Case Study 3.3 in Table 7.2. Figure 7.19 shows the main results from Case Study 3.3.1. The valve positions (Figure 7.19a), the feed of initiator (Figure 7.19b), the reactor temperature (Figure 7.19c) and the conversion average temperature (Figure 7.19d). In total 7095 g of initiator was used, and the duration of the batch was approximately 5.54 hours. As can be seen, mainly fresh water is utilized, which even saturates at some point. In addition to some steam towards the end of the batch. A more spread feed of initiator can be observed for this case. The reactor temperature varies through the batch, and increases towards the end. It does, however, not reach as high temperature as the other cases. Also, the conversion average temperature lies above its setpoint a large fraction of the batch for this case.



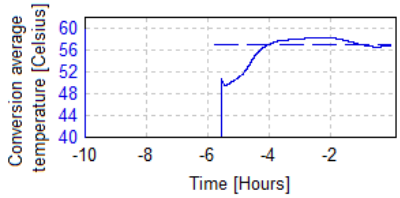
(a) Valve position for fresh water (blue) and steam (red), in addition to their upper constraint (dotted)



(b) Feed rate of initiator



(c) Reactor temperature



(d) Conversion average temperature (solid) and its setpoint (dashed)

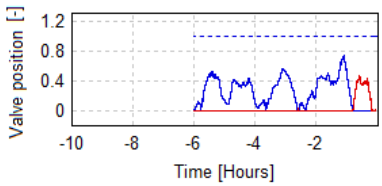
Figure 7.19: Results from Case Study 3.3.1.

7.5.2 Example Simulations 2: Dissimilar tuning

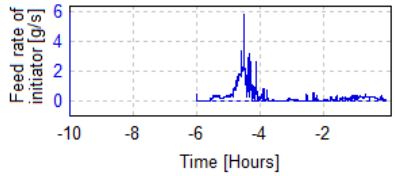
It was attempted to change the penalties on Case Study 3.2.1 to obtain more similar results with respect to the total amount of fed initiator. This was obtained by changing the size of the penalty on the waste of initiator's deviation from setpoint (R_t^w). This will be denoted Case Study 3.2.2.

Case Study 3.2.2: $M = 200$, $N = 1000$

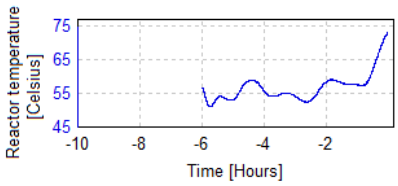
This case study has the same control and prediction horizons as described for Case Study 3.2 in Table 7.2. Figure 7.20 shows the main results from Case Study 3.2.2. The valve positions (Figure 7.20a), the feed of initiator (Figure 7.20b), the reactor temperature (Figure 7.20c) and the conversion average temperature (Figure 7.20d). In total 7181 g of initiator was used, and the duration of the batch was approximately 5.75 hours. As can be seen, mainly fresh water is utilized, in addition to some steam. Initiator is fed mainly early in the batch, and a low feed rate can also be observed throughout the batch. The reactor temperature varies through the batch and increases towards the end. As a result, the conversion average temperature also increases towards the end, reaching its setpoint.



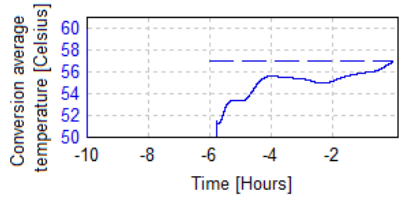
(a) Valve position for fresh water (blue) and steam (red), in addition to their upper constraint (dotted)



(b) Feed rate of initiator



(c) Reactor temperature



(d) Conversion average temperature (solid) and its setpoint (dashed)

Figure 7.20: Results from Case Study 3.2.2.

7.5.3 Summarized results from example simulations

Figure 7.21 summarized the results found with respect to batch time and initiator, from Example Simulations 1 and Example Simulations 2. As can be seen, Case Study 3.2.1 and Case Study 3.2.2 are less efficient in terms of the trade-off between initiator and batch time.

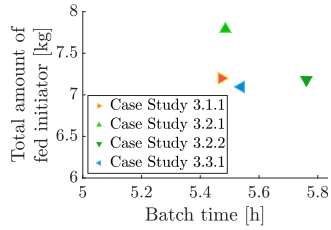


Figure 7.21: Summarized results from the example simulations shown both with similar and dissimilar penalties. Case Study 3.1.1, Case Study 3.2.1 and Case Study 3.3.1 all have the same tuning, while Case Study 3.2.2 have other values of the tuning parameters.

7.5.4 Discussion

Figure 7.16 showed the relation between the total amount of fed initiator and the batch time, for different lengths of control and prediction horizons. Before

conducting the case study, one hypothesis was that with longer horizons, the NMPC would be able to predict a more efficient profile for the two controllers (the SRC and initiator feed). However, the figure shows no obvious trend of this. Case Study 3.1, Case Study 3.2 and Case Study 3.3 follow the same trend as was established by Case Study 2 for the relation between total amount of initiator and batch time.

It might also seem like the collection of all data points are more spread in the right half of the figure, and more concentrated to the left. This can be seen in context to the results from Case Study 2, where the addition of initiator at for batches with already a small amount of initiator, will have a more significant effect than for systems already containing a large amount of initiator. The system will then also be more sensitive to disturbances, and a larger amount of outliers can be expected.

Figure 7.16 did not show any precise results in terms of the relation between the total amount of fed initiator and control and prediction horizon. However, the simulations are insensitive to this variable change and manage to produce approximately the same result independently of the length of the horizons. This indicates a robust application and that the yield of additional computational power is low for this parameter change. Approximately the same result can be obtained by using long horizons, which requires high computational power, as for shorter horizons, which has shorter runtime. Short horizon would then be preferred to ensure that the real-time criterion is kept. To confirm this, however, the additional case studies were performed.

When comparing the data points from Example Simulations 1 in Figure 7.21, the simulation for Case Study 3.2.1 used much more initiator then Case Study 3.1.1 and Case Study 3.3.1, without obtaining any shorter batch time. The conversion average temperature for Case Study 3.2.1 (Figure 7.17d), is much lower throughout the batch then Case Study 3.1.1 (Figure 7.18d). This can be due to a longer prediction horizon for Case Study 3.2.1. The setpoint will then be obtained further into the prediction horizon, and lower temperatures can therefore be allowed during the batch. Lower temperatures will, however, cause a lower production rate, and therefore longer batch times.

It was attempted to alter the penalty on $R_{\mathrm{t}}^{\mathrm{w}}$ deviations from its setpoints, to obtain the same amount of fed initiator. This is shown by the dark green data point in Figure 7.21 (Case Study 3.2.2). However, a longer batch time was then obtained. This supports the argument in the previous paragraph; a longer prediction horizon compared to Case Study 3.1.1 can lead to generally lower reactor temperature throughout the batch.

When examining Case Study 3.3.1, which has both longer control and prediction horizon, a higher reactor temperature was obtained during the middle part of the simulation compared to the other two. As a result, the conversion average temperature lies above the setpoint a large fraction of the batch. It manages, however, to obtain a large enough temperature drop, before increasing towards the end. The final temperature increase is, as already mentioned, important to burn off the residual monomer. Besides, it can lead to the batch

time being significantly reduced due to an increased production rate at high temperatures. This does require the conversion average temperature to approach its setpoint from underneath the final part of the batch, unless the NMPC would rather cool the reactor content in order to reach the setpoint.

Comparing all four data points in Figure 7.21, Case Study 3.1.1 or Case Study 3.1.1, would be preferable, as both shortest batch time and least initiator is used in these cases. To ensure that the setpoint for the conversion average temperature is reached from below, Case Study 3.1.1 would be the favourable choice. This will also be the least computational demanding case.

7.6 Case Study 4: Horizon parameterization

The purpose of this case study was to investigate the effect of variously located coincidence points and input blocking. The different cases investigated is summarized in Table 7.3. The effect on input blocking will be investigated firstly (Case Study 4.1), followed by the effect on coincidence points (Case Study 4.2). In all cases, the length of the control horizon and the prediction horizon were kept constant.

Table 7.3: Horizon parameterization for all the different cases simulated in Case Study 4. Numbers in parenthesis indicates the parameterization. Case Study 4.1 involves the investigation of input blocking, and Case Study 4.2 involves the investigation of coincidence points. Both horizons is given in number of samples.

	Control horizon	Prediction horizon
Case Study 4.1.1	200 (0, 50, 100, 200)	1000 (every 50)
Case Study 4.1.2	200 (0, 25, 50, 100, 150, 200)	1000 (every 50)
Case Study 4.1.3	200 (0, 25, 50, 75, 100, 150, 200)	1000 (every 50)
Case Study 4.2.1	200 (0, 50, 100, 200)	1000 (every 20)
Case Study 4.2.2	200 (0, 50, 100, 200)	1000 (every 50)
Case Study 4.2.3	200 (0, 50, 100, 200)	1000 (every 125)

Table 7.4 summarizes the results for Case Study 4 in terms of runtime. For Case Study 4.1, no other tuning parameters were change except the input parameterization. For Case Study 4.2, no other tuning parameters were changed except the distance between every coincidence point, and hence also the number of coincidence points.

Table 7.4: The results from Case Study 4. The max time CENIT used to solve the optimization problem for one sample and the mean time CENIT uses. Note that all values represent av average, where three simulations have been performed for every case study.

	CENIT mean [s]	CENIT max [s]
Case Study 4.1.1	2.738	8.308
Case Study 4.1.2	3.104	10.030
Case Study 4.1.3	3.634	10.604
Case Study 4.2.1	3.102	11.406
Case Study 4.2.2	2.738	8.308
Case Study 4.2.3	2.614	7.080

As can be seen when examining the results from Case Study 4.1, the mean time CENIT used for every sample[1], from start to finish[2], increases with an

[1]Sample meaning real-time sample. The optimizer solves the finite horizon open-loop optimal control problem for each sample instant.

[2]Finish here meaning when the batch reaches the terminal criteria (96% conversion)

increasing number of parameters the control horizon is parameterized by. The same applies for the maximum time spent on one sample. For all simulations in Case Study 4.1, the setpoint for the conversion average temperature was reached with a sufficient margin of error. In addition, they all fit into the relation found between the total amount of initiator and batch time in Figure 7.16, for any length of the horizon. The maximum time CENIT used to solve a sample exceeds the sampling time for Case Study 4.1.2 and Case Study 4.1.3.

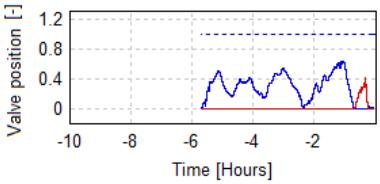
As can be seen, when examining the results form Case Study 4.2, the mean time CENIT used for every sample decreases with decreasing number of coincidence points. The same trend can be seen for the maximum time CENIT uses for a sample during a batch. For all simulations in Case Study 4.2, the setpoint for the conversion average temperature was reached, with a sufficient margin of error. In addition, they all fit into the relation found between the total amount of initiator and batch time in Figure 7.16, for any length of the horizon. The maximum time CENIT used to solve a sample exceeds the sampling time for Case Study 4.2.1.

7.6.1 Case Study 4.1: Simulation results

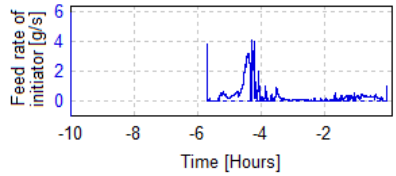
Some example simulation from Case Study 4.1 will now be presented to illustrate the performance of the application. The horizon parameterization and lengths are as given in for Case Study 4.1 in Table 7.3.

Case Study 4.1.1

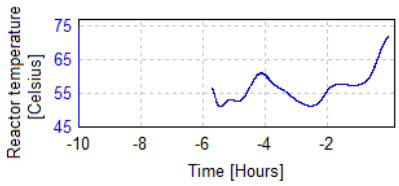
Figure 7.22 shows the results from a simulation with the horizon parameterization described for Case Study 4.1.1. The valve positions (Figure 7.22a), feed of initiator (7.22b) reactor temperature (Figure 7.22c) and the conversion average temperature (Figure 7.22d) is illustrated. In total, 7791 g of initiator was used, and the batch time was 5.49 hours. As can be seen, fresh water is mainly utilized, even though some steam is used towards the end of the batch. Initiator is fed mainly early in the batch, while an approximately constant feed rate is maintained throughout the batch. The reactor temperature varies during the batch and increases towards the end. As a result, the conversion average temperature also increases at the end of the batch, reaching its respective setpoint.



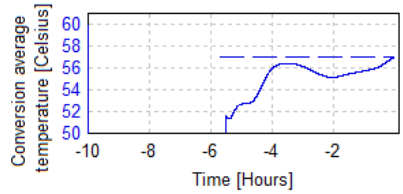
(a) Valve position for fresh water (blue) and steam (red), in addition to their upper constraint (dotted)



(b) Feed rate of initiator



(c) Reactor temperature

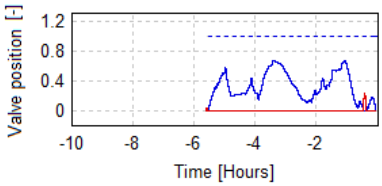


(d) Conversion average temperature (solid) and its setpoint (dashed)

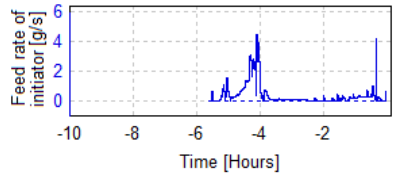
Figure 7.22: Results from Case Study 4.1.1.

Case Study 4.1.2

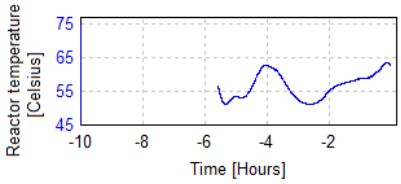
Figure 7.23 shows the results from a simulation with the horizon parameterization described for Case Study 4.1.2. The valve positions (Figure 7.23a), feed of initiator (7.23b) reactor temperature (Figure 7.23c) and the conversion average temperature (Figure 7.23d) is illustrated. In total, 8053 g of initiator was used, and the batch time was 5.39 hours. As can be seen, also here, fresh water is mainly utilized, and only a small portion of steam. Initiator is fed mainly early in the batch. A low feed rate can also be observed during the middle of the batch, in addition to a bigger portion towards the end. The reactor temperature varies during the batch and increases towards the end. However, it does not reach as high temperature as for Case Study 4.1.1. The conversion average temperature also increases at the end of the batch, reaching its respective setpoint. The increment is, however, not as steep as the previous case.



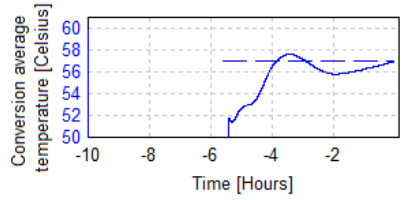
(a) Valve position for fresh water (blue) and steam (red), in addition to their upper constraint (dotted)



(b) Feed rate of initiator



(c) Reactor temperature

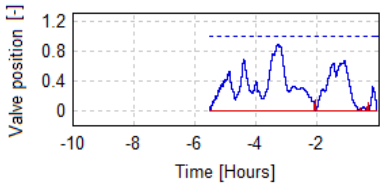


(d) Conversion average temperature (solid) and its setpoint (dashed)

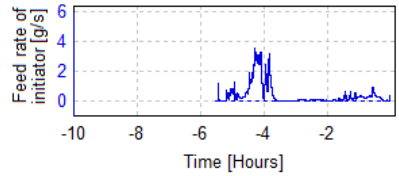
Figure 7.23: Results from Case Study 4.1.2.

Case Study 4.1.3

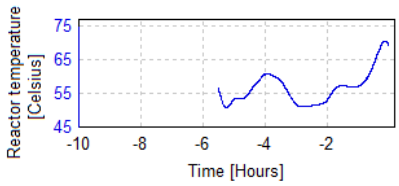
Figure 7.24 shows the results from a simulation with the horizon described for Case Study 4.1.3. The valve positions (Figure 7.24a), feed of initiator (7.24b) reactor temperature (Figure 7.24c) and the conversion average temperature (Figure 7.24d) is illustrated. In total, 8374 g of initiator was used, and the batch time was 5.30 hours. As can be seen, also here, fresh water is mainly utilized. Initiator is fed mainly early in the batch, in addition to a smaller portion throughout the batch. The reactor temperature varies during the batch and increases towards the end. It resembles much to the temperature profile from Case Study 4.1.1. The conversion average temperature also increases at the end of the batch, reaching its respective setpoint.



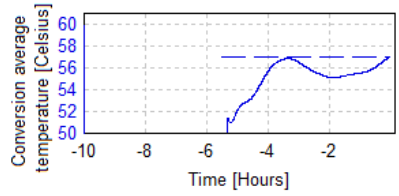
(a) Valve position for fresh water (blue) and steam (red), in addition to their upper constraint (dotted)



(b) Feed rate of initiator



(c) Reactor temperature



(d) Conversion average temperature (solid) and its setpoint (dashed)

Figure 7.24: Results from Case Study 4.1.3.

Summarization

Figure 7.25 summarized the data point found from Case Study 4.1, with respect to total amount of fed initiator and batch time. As can be seen, the results agree with the other data points found in Case Study 3 (Figure 7.16). In addition, an increasing number of blocks the control horizon is parameterized results in shorter batch time. Hence, also a larger amount of initiator is used.

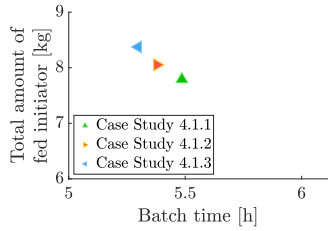


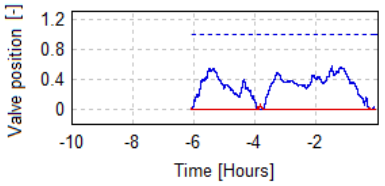
Figure 7.25: Summarized results from Case Study 4.1.1, Case Study 4.1.2 and Case Study 4.1.3.

7.6.2 Case Study 4.2: Simulation results

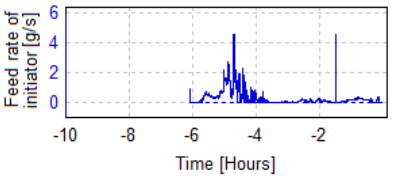
Some example simulation from Case Study 4.2 will now be presented to illustrate the performance of the application. The horizon parameterization and lengths are as given in Table 7.3.

Case Study 4.2.1

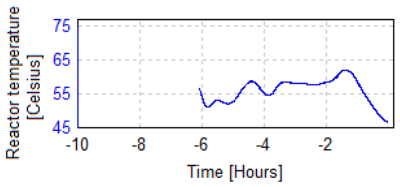
Figure 7.26 shows the results from a simulation with the horizon described for Case Study 4.2.1. The valve positions (Figure 7.26a), feed of initiator (7.26b) reactor temperature (Figure 7.26c) and the conversion average temperature (Figure 7.26d) is illustrated. In total, 7291 g of initiator was used, and the batch time was 5.86 hours. As can be seen, fresh water is mainly utilized, in addition to some small fraction of steam. Initiator is fed mainly early in the batch, in addition to some small portion throughout the batch. The reactor temperature varies during the batch and decreases towards the end. A relatively low conversion average temperature is maintained throughout the batch, and even crosses the setpoint before it decreases toward the end. However, it reaches its respective setpoint.



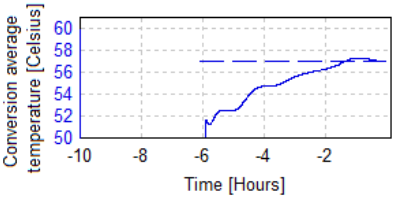
(a) Valve positions for fresh water (blue) and steam (red), in addition the their upper constraint (dotted)



(b) Feed rate of initiator



(c) Reactor temperature



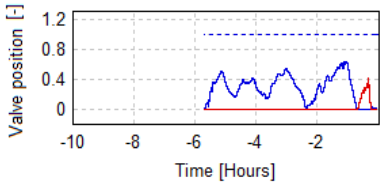
(d) Conversion average temperature

Figure 7.26: Results from Case Study 4.2.1.

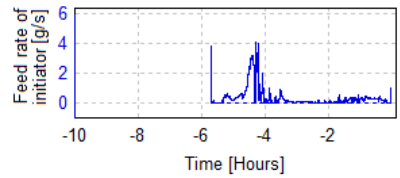
Case Study 4.2.2

Figure 7.27 shows the results from a simulation with the horizon described for Case Study 4.2.2. Note that this case is equivalent to Case Study 4.1.1.

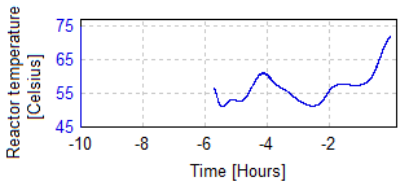
The valve positions (Figure 7.27a), feed of initiator (7.27b) reactor temperature (Figure 7.27c) and the conversion average temperature (Figure 7.27d) is illustrated. In total, 7791 g of initiator was used, and the batch time was 5.49 hours. As can be seen, fresh water is mainly utilized, even though some steam is used towards the end of the batch. Initiator is fed mainly early in the batch, while an approximately constant feed rate is maintained throughout the batch. The reactor temperature varies during the batch and increases towards the end. As a result, the conversion average temperature also increases at the end of the batch, reaching its respective setpoint.



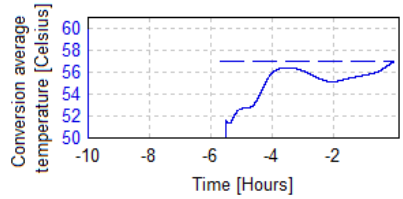
(a) Valve positions for fresh water (blue) and steam (red), in addition the their upper constraint (dotted)



(b) Feed rate of initiator



(c) Reactor temperature

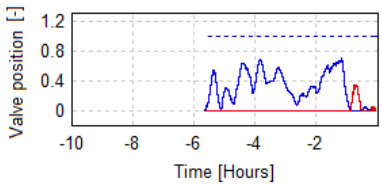


(d) Conversion average temperature

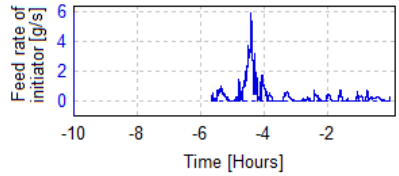
Figure 7.27: Results from Case Study 4.2.2.

Case Study 4.2.3

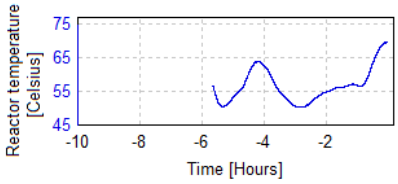
Figure 7.28 shows the results from a simulation with the horizon described for Case Study 4.2.3. The valve positions (Figure 7.28a), feed of initiator (7.28b) reactor temperature (Figure 7.28c) and the conversion average temperature (Figure 7.28d) is illustrated. In total, 7971 g of initiator was used, and the batch time was 5.44 hours. As can be seen, fresh water is mainly utilized, even though some steam is used towards the end of the batch. Initiator is fed mainly early in the batch, while an approximately constant feed rate is maintained throughout the batch. The reactor temperature varies during the batch and increases towards the end. As a result, the conversion average temperature also increases at the end of the batch, reaching its respective setpoint.



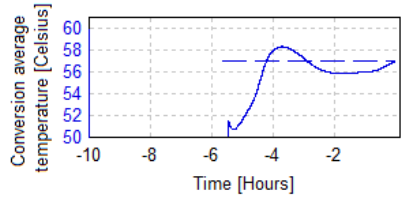
(a) Valve positions for fresh water (blue) and steam (red), in addition the their upper constraint (dotted)



(b) Feed rate of initiator



(c) Reactor temperature



(d) Conversion average temperature

Figure 7.28: Results from Case Study 4.2.3.

Summarization

Figure 7.29 summarized the data point found from Case Study 4.2, with respect to total amount of fed initiator and batch time. As can be seen, the data point from Case Study 4.2.1 stands out in the sense that it used more initiator and obtains a longer batch time than the other two. The data points for Case Study 4.2.2 and Case Study 4.2.3 almost coincide.

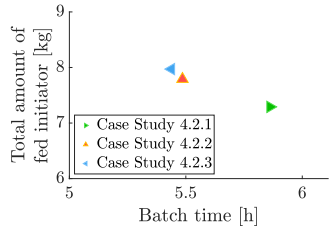


Figure 7.29: Summarized results from Case Study 4.2.1, Case Study 4.2.2 and Case Study 4.2.3.

7.6.3 Discussion

Case Study 4.1 showed expected results with respect to the mean time CENIT used to solve one sample. An increasing number of parameters the control

horizon is parameterized by increases the size of the problem the optimizer has to solve.

Fewer input blocks could have led to less accurate prediction. However, the example simulations presented from Case Study 4.1 did not show any indication of reduced performance. Even more extreme causes could have been tested to verify this, for example, by using only one input block. However, the data points fit into the relation found between initiator and batch time. By adjusting the penalties of the objective function, the data points in Figure 7.25, could have been forced to coincide. The results indicate that the case studies have enough inputs blocks to predict the behaviour of the process sufficiently. As there was no sign of that a reduction in the number of input parameters lead to reduced controller performance, the tuning of the case with the best numerical efficiency, Case Study 4.1.1, would, therefore, be preferable.

Case Study 4.2 showed expected results for the mean time CENIT used to solve one sample, and the maximum time CENIT used to solve one sample. An increasing number of coincidence points increases the size of the problem the optimized has to solve.

The data points from Figure 7.29 also fit into the relation found between initiator and batch time. An increasing number of coincidence points can lead to a more accurate prediction. However, Case Study 4.1.1 results in a decreasing temperature at the end of the batch, even though it is parameterized with the largest number of coincidence point. This behaviour in the reactor temperature is undesirable. One reason for the temperature decrease could be that increasing coincidence points on the waste of initiator, $R_{\mathrm{t}}^{\mathrm{w}}$, causes larger penalties on the feed of initiator. If this is dominating over batch time minimizing outputs, as monomer conversion (X_{M}) and production rate (R_{p}), it can cause less feed of initiator. This can, however, be easily counteracted by adjusting the weights.

As can be seen from the result (Table 7.4), the maximum time CENIT used for one sample exceeds the sampling time, meaning that the real-time criterion was not adhered at all times (Case Study 4.1.2, Case Study 4.1.3, Case Study 4.2.1). If this happens repetitively and often, the optimization problem is too demanding to be solved for every sample. The problem can be simplified by, for example, a simplified model, fewer coincidence points, fewer input parameters, shorter prediction horizon and shorter control horizon. However, the simulations in these case studies showed such long runtimes only for a few samples throughout the batch. When the average runtime also is this low, these few violations will not be critical for the overall performance.

Chapter 8

Conclusion, further work and recommendations

This chapter aims to summarize the findings of this project. Some comments on further work and recommendations will be made.

The purpose of this project was to finalize a mathematical model for emulsion polymerization of Poly-Vinyl Chloride (PVC) in a semi-batch reactor, and further implement Nonlinear Model Predictive Control (NMPC) using this model in Cybernetica's software. The aim was to build a working application, testing it with different tuning parameters, and evaluating its usability.

8.1 Conclusion

This report established the necessary theoretical background of polymerization, optimization, Model Predictive Control (MPC) and state- and parameter estimation. In resemblance to the preliminary project, the remaining modelling work was developed based on first principles. The results were then simulated and validated against process data. An environment for the control system was developed based on the process model and implemented in Cybernetica's software. Schemes for the on-line state- and parameter estimation was then implemented and investigated. The performance of the application in terms of runtime and accuracy, was then evaluated.

Firstly, the quality parameter was validated against Inovyn's calculations. The small deviation was considered tolerable for control purposes and for this project. Also, the simulations where the pre-estimated polynomials describing the unmodelled aspects were included, gave significant improvements for ballistic simulations. The model outputs, as, for example, the reactor temperatures, were of much larger resemblance to the measurements when the polynomials were implemented. The mismatch was however more significant than for a recursive simulation, which was expected.

When performing Case Study 1.1, it became evident that there was a need to implement a state- and parameter estimator. This was expected, as the plant replacement model and the model integrated into the controller, were dissimilar. Significant improvement was then obtained by implementing and tuning a Kalman Filter. There is a resemblance in the correction factor for the kinetic model, CF, and the heat transfer coefficient between the reactor and the jacket, $U_{\mathrm{R,J}}$, from the simulator and the estimator after included the Kalman Filter. This was as expected since the Kalman Filter updates the model to minimize the error between the model outputs and the measurements, and the difference between the two is precisely these profiles. The ideal situation would be that the Kalman Filter estimates profiles the same as the pre-estimated ones. Overall, the implementation of the Kalman Filter was regarded as successful.

For the studied simulations, the feed of initiator was high early in the batch, however, not in the very beginning. It was argued that this was due to particle nucleation, and that the lack of immediate increment was due to pre-loading of initiator.

The relation between the total amount of initiator feed and batch time was found. It was concluded to follow an inverse exponential function, which coincides with the ratio between the propagation and termination reaction, and their relationship to the average number of radicals per particle (Case Study 2). Approximately the same results were shown for different lengths of the prediction and control horizon (Case Study 3). However, it was argued that a short control and prediction horizon would be preferable. This indicates a robust application, and that the yield of additional computational power for the purpose of lowering the amount of initiator is low.

8.2 Further work and recommendations

For further work, the model could have been extended in complexity. The average runtime of most of the case studies performed indicates that the real-time criterion is satisfied the majority of the time. A more complex model could, therefore, be beneficial, even though it would increase the computational load. A time delay on the effect of opening the valve for steam could be implemented, or implementation of a more advanced model over the heat exchanger, in general, could have been investigated. More data on this part of the process would then probably be needed.

Different tuning values for the noise on the states and parameters should also be investigated, aiming for the profiles estimated by the Kalman Filter, and in the simulator, to match even better. This could lead to better prediction, and hence also better performance.

NMPC in combination a Real-Time Optimizer (RTO), was not investigated in this project. This could be beneficial as the value of the conversion average temperature only should be weighted at the end of the batch time. For further work, a higher-level RTO could be implemented with larger sampling time. The RTO would then provide some setpoints to the NMPC.

8.3 Closing comment

The established model and the use of NMPC one the reactor, were considered successful in this project. The implementation of the Kalman Filter as the state- and parameter estimator showed promising results. The results indicate that the control system ensures safe operation with respect to pressure and cooling capacity, in addition to fulfilling quality parameter setpoints. However, this can not be confirmed before the application is implemented and tested at the facility.

Bibliography

Abdel-Alim, A.H., Hamielec, A., 1972. Bulk polymerization of vinyl chloride. Journal of applied polymer Science 16, 783–799.

Alamir, M., Bornard, G., 1995. Stability of a truncated infinite constrained receding horizon scheme: the general discrete nonlinear case. Automatica 31, 1353–1356. Publisher: Pergamon.

Asua, J., 2008. Polymer Reaction Engineering. Blackwell Publishing Ltd, Oxford, UK.

Asua, J.M., 2002. Miniemulsion polymerization. Progress in polymer science 27, 1283–1346. Publisher: Elsevier.

Asua, J.M., 2004. Emulsion polymerization: from fundamental mechanisms to process developments. Journal of Polymer Science Part A: Polymer Chemistry 42, 1025–1041.

Brandrup, J., Immergut, E.H., Grulke, E.A., Abe, A., Bloch, D.R., . Physical Constants of Poly(vinyl Chloride), in: Polymer Handbook (4th Edition). John Wiley & Sons. URL: `https://app.knovel.com/hotlink/pdf/id:kt003S16LE/polymer-handbook-4th/polymer-ha-physical-constants-6`.

Braun, D., Cherdron, H., Rehahn, M., Ritter, H., Voit, B., 2012. Polymer synthesis: theory and practice: fundamentals, methods, experiments. Springer Science & Business Media.

Butté, A., Storti, G., Morbidelli, M., 2002. Evaluation of the Chain Length Distribution in Free-Radical Polymerization, 2. Emulsion Polymerization. Macromolecular theory and simulations 11, 37–52. Publisher: Wiley Online Library.

Chern, C., 2006. Emulsion polymerization mechanisms and kinetics. Progress in polymer science 31, 443–486.

De La Rosa, L.V., Sudol, E., El-Aasser, M., Klein, A., 1999. Emulsion polymerization of styrene using reaction calorimeter. III. Effect of initial

monomer/water ratio. Journal of Polymer Science Part A: Polymer Chemistry 37, 4073–4089.

Fitch, R.M., 1973. The homogeneous nucleation of polymer colloids. British polymer journal 5, 467–483. Publisher: Wiley Online Library.

Flory, P.J., 1941. Molecular size distribution in three dimensional polymers. I. Gelation1. Journal of the American Chemical Society 63, 3083–3090.

Flory, P.J., 1946. Fundamental principles of condensation polymerization. Chemical Reviews 39, 137–197.

Flory, P.J., 1953. Principles of polymer chemistry. Cornell University Press, Ithaca.

Foss, B., Heirung, T.A.N., 2013. Merging optimization and control. Lecture Notes .

Gardon, J., 1968a. Emulsion polymerization. II. Review of experimental data in the context of the revised Smith-Ewart theory. Journal of Polymer Science Part A-1: Polymer Chemistry 6, 643–664. Publisher: Wiley Online Library.

Gardon, J.L., 1968b. Emulsion polymerization. I. Recalculation and extension of the Smith-Ewart theory. Journal of Polymer Science Part A-1: Polymer Chemistry 6, 623–641. Publisher: Wiley Online Library.

Gilbert, R.G., 1995. Emulsion polymerization: a mechanistic approach. Academic Pr.

Hansen, F., Ugelstad, J., 1978. Particle nucleation in emulsion polymerization. I. A theory for homogeneous nucleation. Journal of Polymer Science: Polymer Chemistry Edition 16, 1953–1979. Publisher: Wiley Online Library.

Hansen, F.K., Ugelstad, J., 1979. The effect of desorption in micellar particle nucleation in emulsion polymerization. Die Makromolekulare Chemie: Macromolecular Chemistry and Physics 180, 2423–2434. Publisher: Wiley Online Library.

Harkins, W.D., 1945. A General Theory of the Reaction Loci in Emulsion Polymerization. The Journal of Chemical Physics 13, 381–382. URL: https://doi.org/10.1063/1.1724054, doi:10.1063/1.1724054.

Harkins, W.D., 1946. A General Theory of the Reaction Loci in Emulsion Polymerization. II. The Journal of Chemical Physics 14, 47–48. URL: https://doi.org/10.1063/1.1724062, doi:10.1063/1.1724062.

Harkins, W.D., 1947. A general theory of the mechanism of emulsion polymerization1. Journal of the American Chemical Society 69, 1428–1444.

Harkins, W.D., 1950. General theory of mechanism of emulsion polymerization. II. Journal of polymer science 5, 217–251. Publisher: Wiley Online Library.

Hawkett, B.S., Napper, D.H., Gilbert, R.G., 1977. General solution to the Smith–Ewart equation for emulsion polymerization kinetics. Journal of the Chemical Society, Faraday Transactions 1: Physical Chemistry in Condensed Phases 73, 690–698. Publisher: Royal Society of Chemistry.

Heuts, J.P., Russell, G.T., 2006. The nature of the chain-length dependence of the propagation rate coefficient and its effect on the kinetics of free-radical polymerization. 1. Small-molecule studies. European polymer journal 42, 3–20. Publisher: Elsevier.

Hiemenz, P.C., 1997. Principles of colloid and surface chemistry. 3rd ed., rev. and expanded. ed., Marcel Dekker, New York.

Hovd, M., 2004. A brief introduction to Model Predictive Control. URL= http://www. itk. ntnu. no/fag/TTK4135/viktig/MPCkompendium% 20HOvd. pdf .

Hulburt, H., Katz, S., 1964. Some problems in particle technology: A statistical mechanical formulation. Chemical Engineering Science 19, 555–574. doi:10.1016/0009-2509(64)85047-8.

Kalman, R.E., 1960. A new approach to linear filtering and prediction problems. Journal of Basic Engineering 82, 35–45.

Kiparissides, C., 1996. Polymerization reactor modeling: a review of recent developments and future directions. Chemical Engineering Science 51, 1637–1659.

Kiparissides, C., Daskalakis, G., Achilias, D., Sidiropoulou, E., 1997. Dynamic simulation of industrial poly (vinyl chloride) batch suspension polymerization reactors. Industrial & engineering chemistry research 36, 1253–1267.

Kumar, A., 2003. Fundamentals of polymer engineering. volume 66 of *Plastics engineering*. 2nd ed., rev. and expanded. ed., Marcel Dekker, New York.

Li, B., Brooks, B.W., 1993. Prediction of the average number of radicals per particle for emulsion polymerization. Journal of Polymer Science Part A: Polymer Chemistry 31, 2397–2402.

Matyjaszewski, K., Davis, T.P., 2003. Handbook of radical polymerization. John Wiley & Sons.

Mayne, D.Q., Rawlings, J.B., Rao, C.V., Scokaert, P.O., 2000. Constrained model predictive control: Stability and optimality. Automatica 36, 789–814. Publisher: Elsevier.

Min, K., Ray, W., 1974. On the mathematical modeling of emulsion polymerization reactors Publisher: Taylor & Francis.

Mishra, M., Yagci, Y., 2016. Handbook of vinyl polymers: radical polymerization, process, and technology. CRC press.

Morrison, B.R., Casey, B.S., Lacik, I., Leslie, G.L., Sangster, D.F., Gilbert, R.G., Napper, D.H., 1994. Free radical exit in emulsion polymerization. II. Model discrimination via experiment. Journal of Polymer Science Part A: Polymer Chemistry 32, 631–649. doi:10.1002/pola.1994.080320403.

Mutha, R.K., Cluett, W.R., Penlidis, A., 1997. On-line nonlinear model-based estimation and control of a polymer reactor. AIChE Journal 43, 3042–3058. Publisher: Wiley Online Library.

Nilsson, H., Silvegren, C., Törnell, B., 1978. Swelling of PVC latex particles by VCM. European Polymer Journal 14, 737–741.

Nocedal, J., 2006. Numerical Optimization. Springer Series in Operations Research and Financial Engineering. 2nd ed. ed., Springer New York : Imprint: Springer, New York, NY.

Noergaard, M., Poulsen, N.K., Ravn, O., 2000. New developments in state estimation for nonlinear systems. Automatica 36, 1627–1638. Publisher: Elsevier.

Odian, G., 1991. Principles of polymerization. 3rd ed. ed., Wiley, New York.

Okamura, S., Motoyama, T., 1962. Emulsion polymerization of vinyl acetate. Journal of Polymer Science 58, 221–228. Publisher: Wiley Online Library.

O'toole, J., 1965. Kinetics of emulsion polymerization. Journal of Applied Polymer Science 9, 1291–1297. Publisher: Wiley Online Library.

Painter, P.C., 1997. Fundamentals of polymer science : an introductory text. 2nd ed. ed., Technomic, Lancaster, Pa.

Peggion, E., Testa, F., Talamini, G., 1964. A kinetic study on the emulsion polymerization of vinyl chloride. Die Makromolekulare Chemie: Macromolecular Chemistry and Physics 71, 173–183.

Qin, S.J., Badgwell, T.A., 2003. A survey of industrial model predictive control technology. Control engineering practice 11, 733–764. Publisher: Elsevier.

Rao, C.V., Rawlings, J.B., Mayne, D.Q., 2003. Constrained state estimation for nonlinear discrete-time systems: Stability and moving horizon approximations. IEEE transactions on automatic control 48, 246–258. Publisher: IEEE.

Rawlings, J., Mayne, D., Diehl, M., 2017. Model Predictive Control: Theory, Computation, and Design.

Ray, W.H., Villa, C.M., 2000. Nonlinear dynamics found in polymerization processes—a review. Chemical Engineering Science 55, 275–290.

Schei, T.S., 1997. A finite-difference method for linearization in nonlinear estimation algorithms. Automatica 33, 2053–2058. Publisher: Elsevier.

Seymour, R.B., 1989. Polymer science before and after 1899: notable developments during the lifetime of Maurits Dekker. Journal of Macromolecular Science—Chemistry 26, 1023–1032. Publisher: Taylor & Francis.

Simon, D., 2006. Optimal State Estimation: Kalman, H, and Nonlinear Approaches. John Wiley & Sons, Inc., Hoboken, NJ, USA. Publication Title: Optimal State Estimation.

Smith, W.V., 1948. The Kinetics of Styrene Emulsion Polymerization1a. Journal of the American Chemical Society 70, 3695–3702. Publisher: ACS Publications.

Smith, W.V., Ewart, R.H., 1948. Kinetics of emulsion polymerization. The journal of chemical physics 16, 592–599.

Sperling, L.H., 2005. Introduction to physical polymer science. John Wiley & Sons.

Stevenson, J.K., 1986. Free radical polymerization models for simulating reactive processing. Polymer Engineering & Science 26, 746–759. Publisher: Wiley Online Library.

Stockmayer, W., 1957. Note on the kinetics of emulsion polymerization. Journal of Polymer Science 24, 314–317. Publisher: Wiley Online Library.

Ugelstad, J., Fløgstad, H., Hansen, F., Ellingsen, T., 1973. Studies on the emulsion polymerization of vinyl chloride by seed polymerization, Wiley Online Library. pp. 473–485. Issue: 1.

Ugelstad, J., Hansen, F., 1976. Kinetics and mechanism of emulsion polymerization. Rubber chemistry and technology 49, 536–609.

Ugelstad, J., Mork, P., Dahl, P., Rangnes, P., 1969. A kinetic investigation of the emulsion polymerization of vinyl chloride, Wiley Online Library. pp. 49–68.

Ugelstad, J., Mörk, P., Aasen, J., 1967. Kinetics of emulsion polymerization. Journal of Polymer Science Part A-1: Polymer Chemistry 5, 2281–2288. Publisher: Wiley Online Library.

Ugelstad, J., Mørk, P., 1970. A kinetic study of the mechanism of emulsion polymerisation of vinyl chloride. British Polymer Journal 2, 31–39. Publisher: Wiley Online Library.

Valappil, J., Georgakis, C., 2002. Nonlinear model predictive control of end-use properties in batch reactors. AIChE Journal 48, 2006–2021. doi:10.1002/aic.690480915. place: Hoboken.

Xie, T., Hamielec, A., Wood, P., Woods, D., 1987. Experimental investigation of vinyl chloride polymerization at high conversion—temperature/pressure/conversion and monomer phase distribution relationships. Journal of Applied Polymer Science 34, 1749–1766.

Zhu, N., Zhou, L., 2009. The study of predictive control system based on coincidence points technology, IEEE. pp. 4157–4161.

Özkan, G., Özen, , Erdoğan, S., Hapoğlu, H., Alpbaz, M., 2001. Nonlinear control of polymerization reactor. Computers & Chemical Engineering 25, 757–763. Publisher: Elsevier.

Appendix A

Model equations

In advance of this master thesis, a preliminary project was conducted. The assumptions and model equations that follow in this appendix is therefore adapted from this project. They are re-stated as they are used as the model in the Nonlinear Model Predictive Control (NMPC) implementation.

The main goal of this preliminary work was to establish a model describing an industrial polymerization process, more specifically an emulsion polymerization. The model was to be validated against process data, and parameters were adjusted to minimize the error between the model and measured outputs. The results were then evaluated. As polymerization processes often have troublesome nonlinear dynamics (Ray and Villa, 2000), it was discussed if NMPC could be beneficial for this process.

Firstly, the assumptions made will be stated in Appendix A.1, followed by the model equation for the pheriphery model in Appendix A.2 and the kinetic model in Appendix A.3. The kinetic parameters estimated or found by trial-and-error can be found in Appendix B, the physical properties will be stated in Appendix C and the derivation and calculation of the monomer distribution are elaborated in Appendix D.

A.1 Assumptions

Unless something else is stated, these assumptions will be applicable for the preliminary work and also this master thesis. When establishing the model, some assumptions had to be made, both for clarification and simplification. They are all listed below.

1. Coagulation between particles is neglected.

2. Due to the complexity of the redox initiation mechanism, the initiation is assumed to be described by thermal initiation. Two independent thermal initiation mechanism is therefore assumed to be

present in the reactor. Initiator is assumed to dissolve in the water phase only.

3. Both monomer droplets and polymer particles are assumed to be monodisperse for the sake of volume calculations.

4. The reaction unit is divided into three or four phases, according to Section 2.3.1. Equilibrium between phases is maintained at all times, and the mixture is assumed to be ideal. The gas phase is assumed to be a real gas described by the ideal gas law, corrected by the compressibility factor z.

5. Small polymer radicals may desorb out of the particles, and be reabsorb to particles. The average number of radicals per particle is assumed to be small due to rapid desorption and reabsorption. (Ugelstad et al., 1969; Gilbert, 1995)

6. Molecular weight is mainly determined by chain transfer to monomer (Ugelstad et al., 1969). Chain transfer to polymer and Chain Transfer Agent (CTA) is neglected.

7. Quality parameters are estimated from moments of the Molecular Weight Distribution (MWD).

8. Solvability of monomer in the polymer particles is calculated by the Flory-Huggins equation. Solvability of monomer in water is assumed to follow Henry's law.

9. Nucleation occurs through classical micellar mechanism, controlled by a sufficient amount of surfactant. Homogeneous nucleation is assumed to be neglected. The nucleation stage is though modelled as seeded polymerization.

10. Termination rate is independent of chain-length.

A.2 Periphery Model

The periphery model may take into account the heat loss from the reaction unit, mechanical power as stirring, energy change due to dosing or drainage of chemicals, or energy transfer between the reactor and a cooling jacket. In this case, an energy balance over the reactor will introduce two more states in the system; the temperature in the reactor and the outlet temperature of the cooling jacket. These balances have to take into account the heat generated from the reaction, heat loss the environment, heat transfer between the reactor and the jacket, and change in energy due to dosing of chemicals. Equation

A.2.1 and A.2.2 represent the state equations for the reactor temperature and the outlet jacket temperature, respectively.

$$\frac{\mathrm{d}T_{\mathrm{R}}}{\mathrm{d}t} = \frac{\begin{array}{c}-\Delta H_{\mathrm{rx}} R_{\mathrm{p}} - U_{\mathrm{R,amb}} A_{\mathrm{R,amb}} (T_{\mathrm{amb}} - T_{\mathrm{R}}) \\ + U_{\mathrm{R,J}} A_{\mathrm{R,J}} (T_{\mathrm{J}} - T_{\mathrm{R}}) + \sum \hat{m}_{\mathrm{feed},i} c_{p,i} (T_{\mathrm{feed},i} - T_{\mathrm{R}})\end{array}}{\sum m_i c_{p,i} + m_{\mathrm{steel}} c_{p,\mathrm{steel}}} \tag{A.2.1}$$

$$\frac{\mathrm{d}T_{\mathrm{J,out}}}{\mathrm{d}t} = \frac{\begin{array}{c}-U_{\mathrm{J,amb}} A_{\mathrm{J,amb}} (T_{\mathrm{amb}} - T_{\mathrm{J}}) \\ - U_{\mathrm{R,J}} A_{\mathrm{R,J}} (T_{\mathrm{J}} - T_{\mathrm{R}}) + \hat{m}_{\mathrm{J}} c_{p,\mathrm{J}} (T_{\mathrm{J,in}} - T_{\mathrm{J,out}})\end{array}}{m_{\mathrm{J}} c_{p,\mathrm{J}}} \tag{A.2.2}$$

In Equation A.2.1, the first term represent the heat of reaction, the second term represent heat loss to the environment, the third term represent heat transfer between the reactor and the jacket, and the last term represent change in energy due to post-dosing of chemicals. The reaction unit is considered to be the total weight of the reactor, that is both steel and the content inside the reactor.

In Equation A.2.2, the first term represents the heat loss to the environment, the second term represents heat transfer between the reactor and the jacket, and the second term represents the change in energy due to water flowing through the jacket.

Note that in these two equations, T_{J}, represent the temperature in the jacket, which can be calculated as the average between the in- and outlet stream, as shown in Equation A.2.2.

$$T_{\mathrm{J}} = \frac{T_{\mathrm{J,in}} + T_{\mathrm{J,out}}}{2} \tag{A.2.3}$$

A.3 Kinetic Model

The kinetics model describing the process are chosen in such a way that the model is believed to give accurate results, without introducing unnecessary complexity. A more complex model might be more precise, but it will also most likely require higher computational power. Finding a trade-off and considering the dividend of extra complexity, was therefore important during the preliminary work.

A.3.1 Number of particles

In the preliminary work, it was decided that the number of polymer particles was to be calculated from a previous batch. This was done by a backwards calculation, knowing the amount of fed monomer, the monomer conversion,

particle size and the particle density. The number was therefore kept constant during the whole batch. The calculated is shown in Equation A.3.1

$$N_{\mathrm{T}} = \frac{m_{\mathrm{P}}^{\text{all particles}}}{m_{\mathrm{P}}^{\text{one particle}}} = \frac{m_{\mathrm{M}} X_{\mathrm{M}}}{v^{\mathrm{P}} \rho_{\mathrm{P}}} = \frac{m_{\mathrm{M}} X_{\mathrm{M}}}{\frac{4\pi(d/2)^3}{3} \rho_{\mathrm{P}}} \tag{A.3.1}$$

where m_{M} represent the amount of fed monomer, X_{M} is the monomer conversion, v^{P} is the volume of one polymer particle, ρ_{P} is the density of the polymer particle, and d is the diameter of the particle. The diameter and the density of the particles is measured after at the end of the batch.

There was several reason for not taking in hand the expression for this quantity derived by Smith and Ewart (1948). This backwards calculation will not introduce any additional states to the system, reducing the need for computational power. Also, some of the parameters in Smith and Ewart equations may not be readily obtained, and some studies even indicate that emulsion polymerization of Poly-Vinyl Chloride (PVC) does not follow Smith and Ewart's theory (Peggion et al., 1964).

A.3.2 Monomer Distribution

This model calculates the amount of monomer in all phases. In addition, the volume of each phase is calculated, and the reactor pressure. This provides a full description of the system, at every point through the batch. The model bases on the Flory-Huggins equation (Flory, 1953), and a model developed by Xie et al. (1987) for the case of polymerization of PVC. The C code for the monomer distribution can be found in Appendix D. Some assumptions were made to establish the model for the monomer distribution, and these are summarized below.

1. Equilibrium between phases established immediately
2. The polymer is insoluble in its own monomer
3. The solubility of monomer in water follows Henry's law
4. The solubility of monomer in the polymer follows the Flory-Huggins equation
5. The Pressure, Vapour and Temperature (PVT) properties of the vapour phase follow ideal gas law[1]

The calculation of the monomer distribution will be dependent on the current interval of the process. Due to the short length of Interval I, Interval I and II will be merged for the purpose of calculation of the monomer distribution, volumes and reactor pressure. The system is then considered as seeded

[1] In this master thesis, the compressibility factor z is taken into account to describe the deviation of a real gas from an ideal gas.

for modelling purposes. Interval III will be calculated separately because of different conditions before and after the point of inversion.

Firstly, Assumption 1 assumes that the equilibrium between the phases is established immediately. As a result, the fugacity of monomer in all four phases has to be equal;

$$\hat{f}_{\mathrm{M}}^{\mathrm{g}} = \hat{f}_{\mathrm{M}}^{\mathrm{f}} = \hat{f}_{\mathrm{M}}^{\mathrm{w}} = \hat{f}_{\mathrm{M}}^{\mathrm{p}} \tag{A.3.2}$$

The activity of monomer will be given by the ratio between the monomer fugacity coefficient in the polymer phase over the coefficient in a standard state.

$$\alpha_{\mathrm{M}} = \frac{\hat{f}_{\mathrm{M}}^{\mathrm{p}}}{\hat{f}_{\mathrm{M}}^{0}} \tag{A.3.3}$$

Then, from Assumption 4 and the Flory-Huggins equation, the volume fraction of polymer in polymer phase can be obtained

$$\ln \alpha_{\mathrm{M}} = \ln \left(\frac{\hat{f}_{\mathrm{M}}^{\mathrm{p}}}{\hat{f}_{\mathrm{M}}^{0}} \right) = \ln (1 - \varphi) + \varphi + \chi \varphi^2 \tag{A.3.4}$$

where φ represent the volume fraction of polymer in the polymer particles, and χ represent the Flory-Huggins interaction parameter (Kiparissides, 1996). The code for calculating the monomer distribution in Interval I and II can be found in Appendix D.3, and for Interval III in Appendix D.4.

Interval I and II

For the first two intervals, the activity of monomer is assumed to be equal to one. This simplifies the Flory-Huggins equation to Equation A.3.5.

$$0 = \ln (1 - \varphi) + \varphi + \chi \varphi^2 \tag{A.3.5}$$

φ can then easily be obtained, and the volume fraction of polymer in polymer phase can be calculated. The amount of monomer in polymer phase can then be calculated by Equation A.3.6. The derivation can be found in Appendix D.2.

$$m_{\mathrm{M}}^{\mathrm{p}} = V_{\mathrm{M}}^{\mathrm{p}} \rho_{\mathrm{M}}^{\mathrm{l}} = \frac{m_{\mathrm{P}} \rho_{\mathrm{M}}^{\mathrm{l}}}{\rho_{\mathrm{P}}} \left(\frac{1 - \varphi}{\varphi} \right) \tag{A.3.6}$$

Here, $V_{\mathrm{M}}^{\mathrm{p}}$ represent the volume of monomer in polymer phase, ρ_{M}^{l} represent the density of monomer in liquid phase, m_{P} is the total mass of polymer in the reactor and ρ_{P} represent the density of the polymer. The partial pressure of water and monomer will be equal to their respectively saturation pressures. The reactor pressure will then, as a result, be the sum of the saturation pressures. This is shown in Equation A.3.7. The volume fraction of water (W)

and monomer (M) in gas phase can then be calculated by Equation A.3.8 and A.3.9.

$$\begin{aligned} p_{\mathrm{R}} &= p_{\mathrm{M}} + p_{\mathrm{W}} \\ &= p_{\mathrm{M}}^{\mathrm{sat}} + p_{\mathrm{W}}^{\mathrm{sat}} \end{aligned} \tag{A.3.7}$$

$$y_{\mathrm{M}} = \frac{p_{\mathrm{M}}^{\mathrm{sat}}}{p_{\mathrm{R}}} \tag{A.3.8}$$

$$y_{\mathrm{W}} = \frac{p_{\mathrm{W}}^{\mathrm{sat}}}{p_{\mathrm{R}}} \tag{A.3.9}$$

Here p_{R} represent the reactor pressure, p_{M} and p_{W} represent of partial pressure of monomer and water respectively, $p_{\mathrm{M}}^{\mathrm{sat}}$ and $p_{\mathrm{W}}^{\mathrm{sat}}$ represent the saturation pressure of monomer and water respectively, and y_{M} and y_{W} represent the volume fraction of monomer and water respectively.

The mass of monomer in the gas phase can then be calculated. The derivation can be found in Appendix D.1.

$$m_{\mathrm{M}}^{\mathrm{g}} = V^{\mathrm{g}} y_{\mathrm{M}} \rho_{\mathrm{M}}^{\mathrm{g}} = \frac{(V_{\mathrm{R}} - V_{\mathrm{fluid,s}}) y_{\mathrm{M}} \rho_{\mathrm{M}}^{\mathrm{g}}}{1 - \frac{M_{\mathrm{W}} y_{\mathrm{W}} p_{\mathrm{R}}}{z R^{\mathrm{g}} T^{\mathrm{g}} \rho_{\mathrm{W}}^{\mathrm{l}}} - \frac{y_{\mathrm{M}} \rho_{\mathrm{M}}^{\mathrm{g}}}{\rho_{\mathrm{M}}^{\mathrm{l}}}}} \tag{A.3.10}$$

Here, $\rho_{\mathrm{M}}^{\mathrm{g}}$ represent the density of monomer in gas phase, $\rho_{\mathrm{W}}^{\mathrm{l}}$ represent the density of water in liquid phase, and M_{W} represent the molecular weight of water. For Interval I and II, the temperature in the gas phase, T^{g}, will be equal to the temperature in liquid phase. The compressability factor z is calculated from fugacity parameters, V_{R} is the total volume of the reactor, and $V_{\mathrm{fluid,s}}$ is the volume of the liquid and solid material if no monomer and water was contained in the gas phase. The calculation of $V_{\mathrm{fluid,s}}$ is shown in Equation A.3.11.

$$V_{\mathrm{fluid,s}} = \frac{m_{\mathrm{W}}}{\rho_{\mathrm{W}}^{\mathrm{l}}} + \frac{m_{\mathrm{M}}}{\rho_{\mathrm{M}}^{\mathrm{l}}} + \frac{m_{\mathrm{P}}}{\rho_{\mathrm{P}}} \tag{A.3.11}$$

As already mention in Assumption 3, the solubility of monomer in water phase is assumed to follow Henry's law. The mass of monomer in water phase will then be given by Equation A.3.12.

$$m_{\mathrm{M}}^{\mathrm{w}} = K_1 \alpha_{\mathrm{M}} m_{\mathrm{W}} \tag{A.3.12}$$

where K_1 represent the monomer in water solubility constant, and will be expressed as 0.088 kg VCM/kg H_2O (Nilsson et al., 1978).

The mass of monomer in the free phase can then be calculated by mass conservation. This is shown in Equation A.3.13.

$$m_{\mathrm{M}}^{\mathrm{f}} = m_{\mathrm{M}} - m_{\mathrm{M}}^{\mathrm{p}} - m_{\mathrm{M}}^{\mathrm{g}} - m_{\mathrm{M}}^{\mathrm{w}} \tag{A.3.13}$$

When the mass and densities of each component in every phase are calculated, the volume of all phases can be calculated from Equation A.3.14 - A.3.18.

$$V^{\mathrm{g}} = \frac{m_{\mathrm{M}}^{\mathrm{g}}}{y_{\mathrm{M}} \rho_{\mathrm{M}}^{\mathrm{g}}} \tag{A.3.14}$$

$$V^{\mathrm{p}} = \frac{m_{\mathrm{P}}}{\rho_{\mathrm{P}}} + \frac{m_{\mathrm{M}}^{\mathrm{p}}}{\rho_{\mathrm{M}}^{\mathrm{l}}} \tag{A.3.15}$$

$$V^{\mathrm{f}} = \frac{m_{\mathrm{M}}^{\mathrm{f}}}{\rho_{\mathrm{M}}^{\mathrm{l}}} \tag{A.3.16}$$

$$V^{\mathrm{w}} = \sum_i \frac{m_i^{\mathrm{w}}}{\rho_i^{\mathrm{l}}},\ i = \text{component in water phase} \tag{A.3.17}$$

$$V^{\mathrm{l}} = V_{\mathrm{R}} - V^{\mathrm{g}} = V^{\mathrm{p}} + V^{\mathrm{f}} + V^{\mathrm{w}} \tag{A.3.18}$$

To be able to calculate the production rate of polymer (See Equation A.3.34), the concentration of monomer in the polymer phase has to be obtained. This parameter can be calculated by the following equation.

$$[\mathrm{M}]^{\mathrm{p}} = \frac{m_{\mathrm{M}}^{\mathrm{p}}}{M_{\mathrm{M}} V^{\mathrm{p}}} \tag{A.3.19}$$

where M_{M} represent the molecular weight of monomer.

Interval III

For Interval III, the activity of monomer is no longer equal to 1. The partial pressure of monomer will no longer be equal to its saturation pressure, and the temperature in the gas phase will not be equal to the temperature in the liquid phase. In this interval, the temperature in the gas phase will be modelled to be equal to the saturation temperature. The partial pressure of monomer will be given by the following equation

$$p_{\mathrm{M}} = p_{\mathrm{M}}^{\mathrm{sat}} \alpha_{\mathrm{M}} \tag{A.3.20}$$

The reactor pressure will then be given by Equation A.3.21.

$$p_{\mathrm{R}} = p_{\mathrm{M}}^{\mathrm{sat}} \alpha_{\mathrm{M}} + p_{\mathrm{W}}^{\mathrm{sat}} \tag{A.3.21}$$

Equation A.3.5 can not be utilized in this interval. The volume fraction of polymer in the polymer phase will be found iteratively in this interval. The activity is then found by inverse calculation of the Flory-Huggins equation. The approach for this calculation can be found in Algorithm 1, and the belonging code can be found in Appendix D.4.

Set mass of monomer in the polymer phase to be the sum of monomer in the system (initial value);
for *i = 1:3* **do**
  Calculate volume fraction from initial value;
  Inversely calculate the activity;
  Calculate saturation pressures, pressure, saturation temperature and densities;
  Calculate z and volume fractions;
  Calculate monomer in gas and water phase;
  Calculate monomer in the polymer phase from mass balance;
  Set the initial value to be the sum of some weight of the previous initial value and some weight of the value just calculated;
end
Follow same procedure as for Interval I and II for the remaining calculations

Algorithm 1: Monomer distribution calculation for Interval III

A.3.3 Radical Distribution

Modelling the radical distribution is necessary to calculate the polymerization rate, as it is a function of the number of radicals inside a particle. In the preliminary work, this parameter was modelled to be the average of all particles. As mention, in emulsion polymerization number of radicals inside a particle will vary both in time and from one particle to another. This was done by utilizing Li and Brooks (1993) prediction. This approximation introduces few additional states to the system and is a good compromise between accuracy and complexity. The result of Li and Brooks work was the following Ordinary Differential Equation (ODE) describing the average number of radicals per particle.

$$\frac{\mathrm{d}\bar{n}}{\mathrm{d}t} = \sigma - k'''\bar{n} - \psi C\bar{n}^2 \tag{A.3.22}$$

Here, $\bar{n}$ represent the average number of radicals per particle, σ is the average rate radicals enters the particles from water phase, k''' is the rate coefficient describing the radical exit from the particles to the water phase, C is the relative rate coefficient of radical termination i polymer phase, and ψ will be a parameter that varies between 0 and 2. These parameters is calculated by Equation A.3.23 - A.3.26.

$$\sigma = k_{\mathrm{ads}}\frac{n_{\mathrm{R}}^{\mathrm{w}}}{V^{\mathrm{w}}} \tag{A.3.23}$$

$$k''' = k_{\mathrm{des}} \tag{A.3.24}$$

$$C = \frac{k_{\mathrm{t}}^{\mathrm{p}} N_{\mathrm{T}}}{V^{\mathrm{p}}} = \frac{k_{\mathrm{t}}^{\mathrm{p}}}{v^{\mathrm{p}}} \tag{A.3.25}$$

$$\psi = \frac{2(2\sigma + k''')}{2\sigma + k''' + C} \tag{A.3.26}$$

Here, k_{ads} is the rate constant for radical adsorption on the particles, n_{R}^{w} is the number of radicals in water phase, V^{w} is the volume of the water phase, k_{des} is the desorption rate of radicals from the particles, k_{t}^{p} is the rate constant for termination in particle phase, V^{p} is the total volume of the polymer phase and v^{p} is the volume of one polymer particle.

By Equation A.3.23, it becomes clear that it is necessary to have information about the number of radicals in the water phase. This is done by introducing another ODE describing the change of radicals in this phase. The following ODE is obtained by setting up a material balance over the radicals in the water phase.

$$\frac{\mathrm{d}n_{\text{R}}^{\text{w}}}{\mathrm{d}t} = R_{\text{I}} + k_{\text{des}}\bar{n}N_{\text{T}} - k_{\text{ads}}N_{\text{T}}\frac{n_{\text{R}}^{\text{w}}}{V^{\text{w}}} - k_{\text{t}}^{\text{w}}\left(\frac{n_{\text{R}}^{\text{w}}}{V^{\text{w}}}\right)^2 V^{\text{w}} \tag{A.3.27}$$

where n_{R}^{w} represents the moles of radicals in water phase, R_{I} is the total rate of radical formation that can be used for initiation, V^{w} is the volume of the water phase and k_{t}^{w} is the radical termination in water phase. Note that the last term in the equation will represent the rate of termination in water phase, R_{t}^{w}.

$$R_{\text{t}}^{\text{w}} = k_{\text{t}}^{\text{w}}\left(\frac{n_{\text{R}}^{\text{w}}}{V^{\text{w}}}\right)^2 V^{\text{w}} \tag{A.3.28}$$

Since it is assumed that no long-chained polymer will be present in the water phase, and that after termination the radicals will be inactive and incapable to undergo polymerization, this will be equivalent to a loss, or 'waste', of initiator.

By implementation of Equation A.3.22 and A.3.27, the model will give a approximate description of the radical distribution of the system.

A.3.4 Chemical Reactions

The main chemical reactions taking place in this system is initiation, propagation, termination and chain transfer, as described in Section 2.2. Equation A.3.29 and A.3.30 are used to describe the decomposition of initiator, and Equation A.3.31 - A.3.33 are used to describe the chain initiation.

$$R_{\text{d}_1} = k_{\text{d}_1}[\text{I}_1]^{\text{w}} \tag{A.3.29}$$
$$R_{\text{d}_2} = k_{\text{d}_2}[\text{I}_2]^{\text{w}} \tag{A.3.30}$$

$$R_{\text{I}_1} = f_1 k_{\text{d}_1}[\text{I}_1]^{\text{w}} \tag{A.3.31}$$
$$R_{\text{I}_2} = 2f_2 k_{\text{d}_2}[\text{I}_2]^{\text{w}} \tag{A.3.32}$$
$$R_{\text{I}} = R_{\text{I}_1} + R_{\text{I}_2} \tag{A.3.33}$$

Here, R_{d_1} and R_{d_2} describe the rate of decomposition of Initiator 1 and 2, k_{d_1} and k_{d_2} represent the rate constant for decomposition of Initiator 1 and 2, $[I_1]^w$ and $[I_2]^w$ represent the concentration of Initiator 1 and 2 in water phase. R_{I_1} and R_{I_2} represent the rate of formation of radicals from Initiator 1 and 2 which can be used for polymerization, and R_I represent the total rate of formation of radicals that can be used for polymerization. As Initiator 1 represent the redox initiator, which is assumed to give one radical fragment for each initiator molecule, Equation A.3.31 is not multiplied by 2, in comparison with Equation A.3.32.

The rate of reaction, which is defined to be the propagation reaction, is defined by Equation A.3.34.

$$R_p = k_p[M]^p P_{tot} \tag{A.3.34}$$

Here, k_p is the rate constant for the propagation and $[M]^p$ is the concentration of monomer in polymer phase, and P_{tot} is the total moles of radicals inside the polymer particles. As the number of radicals inside a particle will vary through the batch, and also number of radicals will vary from one particle to another, using $\bar{n}$ is often sufficient to calculate P_{tot}. In that case, P_{tot} will be calculated by $P_{tot} = \bar{n}N_T$.

$$R_p = k_p[M]^p \bar{n} N_T \tag{A.3.35}$$

An ODE describing the monomer conversion can be established when the production rate is calculated.

$$\frac{dX_M}{dt} = \frac{R_p}{\int_0^t \hat{n}_M dt} \tag{A.3.36}$$

Here, $\hat{n}_M$ represent the molar flow of monomer into the reactor, and the integral will then represent the total amount of monomer fed into the reactor from the beginning of the batch until time t.

Termination and chain transfer can have a big impact on the MWD of the system. The model separates between termination by combination and disproportionation, and it requires a ratio between the two defined by the following equation.

$$\begin{aligned} \epsilon &= 1 - \frac{k_{tc}}{k_{td}} \\ k_{td} &= \epsilon k_t \\ k_{tc} &= (1-\epsilon)k_t \end{aligned} \tag{A.3.37}$$

An expression for the rate of termination can be found by the following equation

$$R_t = k_t \left(\frac{\bar{n}N_T}{V_p} \right)^2 V_p \tag{A.3.38}$$

In the case of polymerization of PVC, it is assumed that termination by disproportionation will be the dominant mode of termination (Abdel-Alim and

Hamielec, 1972), and ϵ will be set to 1. Note that the model is not limited by this choice, and ϵ can be set to desired value.

In addition to the reactions described in this section, termination in water phase ($k_{\mathrm{t}}^{\mathrm{w}}$), adsorption of radicals into the polymer particles (k_{ads}) and desorption of radicals into the water phase (k_{des}), were also taken into account when modelling the reactions of the process (From Appendix A.3.3). These are parameters that can have a large impact on the radical distribution of the system, and therefore also the production rate of polymer.

Finally, chain transfer may also be important in emulsion polymerization processes. The model opens up the opportunity to include chain transfer to monomer, polymer and chain transfer agents. These reactions may alter the MWD of the system, as will be described in the upcoming section. In the case of polymerization of PVC, chain transfer to monomer is expected to be of most importance.

A correction factor, CF, was implemented in the model. This factor was used to adjust all the kinetic parameters by the same scale easily. This is shown in Equation A.3.39.

$$\begin{aligned}
k_{\mathrm{p}} &= k_{\mathrm{p}}^{'} CF \\
k_{\mathrm{t}} &= k_{\mathrm{t}}^{'} CF \\
k_{\mathrm{t}}^{\mathrm{w}} &= k_{\mathrm{t}}^{\mathrm{w}'} CF \\
k_{\mathrm{d_1}} &= k_{\mathrm{d_1}}^{\prime} CF \\
k_{\mathrm{d_2}} &= k_{\mathrm{d_2}}^{\prime} CF \\
k_{\mathrm{des}} &= k_{\mathrm{des}}^{'} CF \\
k_{\mathrm{ads}} &= k_{\mathrm{ads}}^{'} CF \\
k_{\mathrm{ct}}^{\mathrm{M}} &= k_{\mathrm{ct}}^{\mathrm{M}'} CF \\
k_{\mathrm{ct}}^{\mathrm{AX}} &= k_{\mathrm{ct}}^{\mathrm{AX}'} CF \\
k_{\mathrm{ct}}^{\mathrm{P}} &= k_{\mathrm{ct}}^{\mathrm{P}'} CF
\end{aligned} \tag{A.3.39}$$

Here, the dashed parameters represent their values before correction.

A.3.5 Material Balances

A balance over every species in the reactor has to be performed. The result will be a set of ODEs, taking into account dosing and consumption of chemicals. In this industrial case, there is no drainage of chemicals during the batch. The general expression for the rate of change of a specie in molar basis is given in the equation below.

$$\frac{\mathrm{d}n_i}{\mathrm{d}t} = \hat{n}_i + R_i \tag{A.3.40}$$

where $\hat{n}_i$ represent the dosing of specie i, and R_i represent the net generation or consumption of specie i through chemical reactions. The main chemical species in the reactor will be water (W), monomer (M), polymer (P), surfactant (S) and initiator (I_1 and I_2), and their balances is described by Equation A.3.41 - A.3.46.

$$\frac{\mathrm{d}n_{\mathrm{M}}}{\mathrm{d}t} = \hat{n}_{\mathrm{M}} - R_{\mathrm{p}},\ n_{\mathrm{M}}(0) = n_{\mathrm{M},0} \tag{A.3.41}$$

$$\frac{\mathrm{d}n_{\mathrm{I}_1}}{\mathrm{d}t} = \hat{n}_{\mathrm{I}_1} - R_{\mathrm{d}_1},\ n_{\mathrm{I}_1}(0) = n_{\mathrm{I}_1,0} \tag{A.3.42}$$

$$\frac{\mathrm{d}n_{\mathrm{I}_2}}{\mathrm{d}t} = \hat{n}_{\mathrm{I}_2} - R_{\mathrm{d}_2},\ n_{\mathrm{I}_2}(0) = n_{\mathrm{I}_2,0} \tag{A.3.43}$$

$$\frac{\mathrm{d}n_{\mathrm{S}}}{\mathrm{d}t} = \hat{n}_{\mathrm{S}},\ n_{\mathrm{S}}(0) = n_{\mathrm{S},0} \tag{A.3.44}$$

$$\frac{\mathrm{d}n_{\mathrm{W}}}{\mathrm{d}t} = \hat{n}_{\mathrm{W}},\ n_{\mathrm{W}}(0) = n_{\mathrm{W},0} \tag{A.3.45}$$

As indicated by the equations, it is assumed that surfactant and water not will take part in any reactions, but merely depend on the dosing. The rate of change of monomer will depend on the dosing, and the rate of propagation. It is assumed that the consumption of monomer through propagation is much larger than those of chain initiation. Even though the chain initiation is not included in this balance, it is important for the generation of oligomers used in further polymerization. The rate of change of initiator will depend on the dosing, and the decomposition rate of the initiators.

There is no need to establish a balance over the amount of polymer in the reactor, as this will be directly given by the amount of consumed monomer. Regardless, since the molecular weight of the polymer will vary through the batch and is not easily obtained without measurements, a balance would be difficult to obtain. The calculation of the mass of polymer in the reactor is given by Equation A.3.46.

$$m_{\mathrm{P}} = X_{\mathrm{M}} M_{\mathrm{M}} \int_0^t \hat{n}_{\mathrm{M}} \mathrm{d}t \tag{A.3.46}$$

Here, M_{M} represent the molecular weight of monomer.

A.4 Molecular Weight Distribution and Moment Balances

The theory of MWD was briefly introduced in Section 2.1.1, and the procedure for calculating these will now be described. As described in Section 2.1.1, the concept of MWD introduces a problematic aspect when modelling the system. The average molecular weights in the form they are written in Equation 2.1.3

and 2.1.4, introduce a large number of individual species and requires the number of both active and inactive chains, of every possible chain length. To be able to track the average molecular weights, moment balances are introduced (Asua, 2008; Matyjaszewski and Davis, 2003).

$$[\mu_k] = \sum_{i=1}^{\infty} i^k [P_i] \tag{A.4.1}$$

$$[\nu_k] = \sum_{i=1}^{\infty} i^k [D_i] \tag{A.4.2}$$

Here μ_k represent live moments, and ν_k represent dead moments. Equation 2.1.3 and 2.1.4 can then be rewritten to the following equations

$$\overline{M_{\mathrm{n}}} = \frac{\mu_1 + \nu_1}{\mu_0 + \nu_0} M_{\mathrm{M}} \tag{A.4.3}$$

$$\overline{M_{\mathrm{w}}} = \frac{\mu_2 + \nu_2}{\mu_1 + \nu_1} M_{\mathrm{M}} \tag{A.4.4}$$

Here μ_0, μ_1 and μ_2 represent the zeroth, first and second live moment respectively, and ν_0, ν_1 and ν_2 represent the zeroth, first and second dead moment respectively. For modelling purposes during the preliminary project, the zeroth, first and second, both live and dead, moments had to be calculated. They were therefore re-stated below. Note that the equation take into account chain transfer to monomer, CTA and polymer, in addition to termination by both combination and disproportionation. An example of the derivation of these balances is given in Appendix E.

$$\frac{\mathrm{d}[\mu_0]}{\mathrm{d}t} = R_{\mathrm{I}} - (k_{\mathrm{td}} + k_{\mathrm{tc}})[\mu_0]^2 \tag{A.4.5}$$

$$\begin{aligned}\frac{\mathrm{d}[\mu_1]}{\mathrm{d}t} = R_{\mathrm{I}} + k_{\mathrm{ct}}^{\mathrm{M}}[\mathrm{M}]([\mu_0] - [\mu_1]) + k_{\mathrm{ct}}^{\mathrm{AX}}[\mathrm{AX}]([\mu_0] - [\mu_1]) + k_{\mathrm{p}}[\mathrm{M}][\mu_0] \\ +(k_{\mathrm{td}} + k_{\mathrm{tc}})[\mu_0][\mu_1] + k_{\mathrm{ct}}^{\mathrm{P}}([\mu_0][\nu_2] - [\mu_1][\nu_1])\end{aligned} \tag{A.4.6}$$

$$\begin{aligned}\frac{\mathrm{d}[\mu_2]}{\mathrm{d}t} = R_{\mathrm{I}} + k_{\mathrm{ct}}^{\mathrm{M}}[\mathrm{M}]([\mu_0] - [\mu_2]) + k_{\mathrm{ct}}^{\mathrm{AX}}[\mathrm{AX}]([\mu_0] - [\mu_2]) + k_{\mathrm{p}}[\mathrm{M}]([\mu_0] \\ -2[\mu_1]) + (k_{\mathrm{td}} + k_{\mathrm{tc}})[\mu_0][\mu_1] + k_{\mathrm{ct}}^{\mathrm{P}}([\mu_0][\nu_3] - [\mu_2][\nu_2])\end{aligned} \tag{A.4.7}$$

$$\frac{\mathrm{d}[\nu_0]}{\mathrm{d}t} = k_{\mathrm{ct}}^{\mathrm{M}}[\mathrm{M}][\mu_0] + k_{\mathrm{ct}}^{\mathrm{AX}}[\mathrm{AX}][\mu_0] + (k_{\mathrm{td}} + (0.5k_{\mathrm{tc}})[\mu_0]^2 \tag{A.4.8}$$

$$\begin{aligned}\frac{\mathrm{d}[\nu_1]}{\mathrm{d}t} = k_{\mathrm{ct}}^{\mathrm{M}}[\mathrm{M}][\mu_1] + k_{\mathrm{ct}}^{\mathrm{AX}}[\mathrm{AX}][\mu_1] + (k_{\mathrm{td}} + k_{\mathrm{tc}})[\mu_0][\mu_1] + k_{\mathrm{ct}}^{\mathrm{P}}([\mu_1][\nu_1] \\ -[\mu_0][\nu_2])\end{aligned} \tag{A.4.9}$$

$$\frac{\mathrm{d}[\nu_2]}{\mathrm{d}t} = k_{\mathrm{ct}}^{\mathrm{M}}[\mathrm{M}][\mu_2] + k_{\mathrm{ct}}^{\mathrm{AX}}[\mathrm{AX}][\mu_2] + (k_{\mathrm{td}} + k_{\mathrm{tc}})[\mu_0][\mu_2] + k_{\mathrm{tc}}[\mu_1]^2 + k_{\mathrm{ct}}^{\mathrm{P}}([\mu_2][\nu_1] - [\mu_0][\nu_3]) \tag{A.4.10}$$

The MWD calculation only require the zeroth, first and second moments dead moments. But as can be seen the ODE for the second moments are dependent on the third live moment, $[\nu_3]$. Setting up a balance for this moment will only yield another, unknown, variable. The set suffers from a moment closure problem (Asua, 2008). Hulburt and Katz (1964) purposed to describe the MWD as a truncate series of Laguerre polynomials, and a steady state approximation of $[\nu_3]$ will as a result be given by A.4.11, only described by $[\nu_0]$, $[\nu_1]$ and $[\nu_2]$.

$$[\nu_3] = \frac{[\nu_2]}{[\nu_0][\nu_1]}(2[\nu_0][\nu_2] - [\nu_1]^2) \tag{A.4.11}$$

These equations states the necessary information to calculate the number and weight average molecular weights, and hence also the Polydispersity Index (PDI) of the distribution.

Appendix B

Results from preliminary work

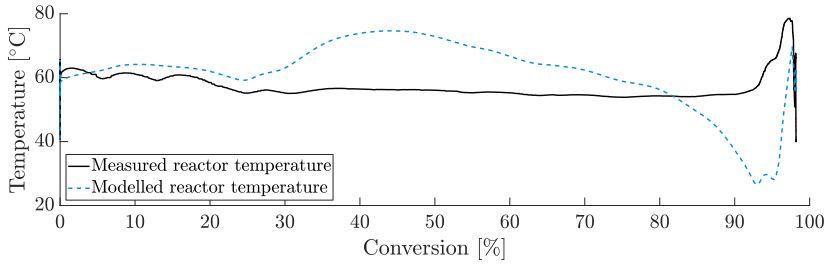
The kinetic parameters used in the simulations are shown in Table B.1, and were obtained during the preliminary project. The rate constants for the propagation and termination were found in literature (Kiparissides et al., 1997), while the others were found by estimation in ModelFit or trial-and-error. The rate constant for chain transfer to Chain Transfer Agent (CTA) and polymer is set to zero, at it was assumed that chain transfer to monomer was dominating. Also, as literature indicated that termination by disproportionation would be dominating in systems of emulsion polymerization of Poly-Vinyl Chloride (PVC) (Abdel-Alim and Hamielec, 1972), ϵ was set to one.

Table B.1: Kinetic parameters that were used as basis in the preliminary project; Rate constant for propagation, termination, adsorption, desorption, termination in water phase and chain transfer to monomer. In addition, the ratio between the modes of termination, and the efficiency factor for both initiators is given. Kinetic parameters for the propagation and termination was obtained from literature (Kiparissides et al., 1997).

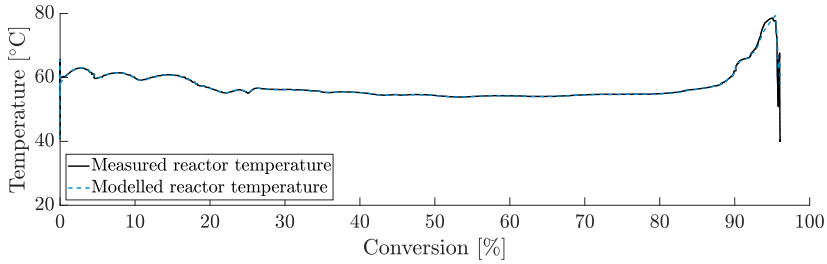
Parameter	Value	Unit
k_p	$5.0 \cdot 10^{4} \exp(-3320/T_\mathrm{R})$	$\mathrm{m}^3/\mathrm{mol/s}$
K_c	$1.01 \cdot 10^{-7} \exp(-5740\,(1/T_\mathrm{R} - 1/333.15)$	$\mathrm{m}^3/\mathrm{mol/s}$
k_t	$2k_\mathrm{p}/K_\mathrm{c}$	$\mathrm{m}^3/\mathrm{mol/s}$
k_ads	0.500	$1/\mathrm{s}$
k_des	$2.857 \cdot 10^{-4}$	$1/\mathrm{s}$
k_t^w	$8.500 \cdot 10^{-3}$	$\mathrm{m}^3/\mathrm{mol/s}$
k_ct^M	$1.05 \cdot 10^{7} \exp(-61203/RT_\mathrm{R})$	$\mathrm{m}^3/\mathrm{mol/s}$
$k_\mathrm{ct}^\mathrm{AX}$	0	
k_ct^P	0	
ϵ	1	
f_1	0.7	
f_2	0.7	

The results from the preliminary work showed deviations from measured outputs, especially with respect to the reactor temperature. This is shown in Figure B.1a. Tight temperature control is important in polymerization processes, as the reactions often are exothermic and thermal runaway must be avoided. It was concluded that recursive filtering was necessary to obtain satisfactory results. State- and parameter estimation was therefore implemented using Kalman Filter (KF). For the filtering, the parameters chosen to be estimated was the correction factor for the kinetic parameters, CF, and the heat transfer coefficient between the reactor and the jacket, $U_{\mathrm{R,J}}$. The active measurements were chosen to be the reactor temperature, T_{R}, and the outlet jacket temperature, $T_{\mathrm{J,out}}$. This gave a temperature profile with a much smaller bias compared to the measurements. An example simulation is shown in Figure B.1b. As can be seen, Figure B.1a shows a significantly larger mismatch between the measured and the modelled temperature, compared to Figure B.1b. The filtering was then concluded to be successful.

Three arbitrary batches were simulated, all showing the same trends for the two estimated parameters, as shown in Figure B.2.

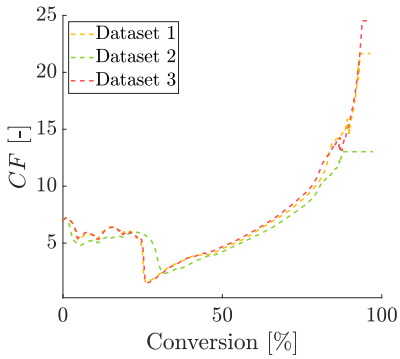


(a) Reactor temperature from a ballistic simulation.

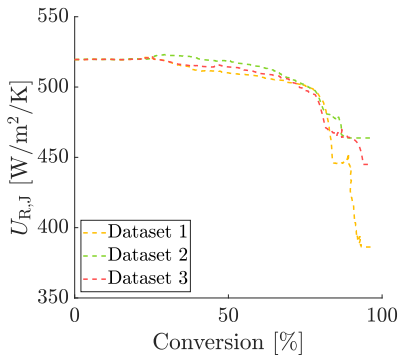


(b) Reactor temperature from a recursive simulation.

Figure B.1: Measured and modelled reactor temperature, both from ballistic and recursive simulation.



(a) The estimated profiles of the correction factor, CF, as function of conversion



(b) The estimated profiles of the heat transfer coefficient between the reactor and the jacket, $U_{\mathrm{R,J}}$, as function of conversion.

Figure B.2: The estimated profiles when performing recursive filtering on three datasets during the preliminary work.

Appendix C

Physical properties

These equation and code snippets are re-stated from the preliminary project to show the calculation of the physical properties used in the model equations (Appendix A).

Code given in this appendix is printed with permission from Peter Singstad, Cybernetica AS. Heat of reaction for propagation used for the simulations is -97.6 kJ/kg. An interval range of the heat of polymerization of Poly-Vinyl Chloride (PVC) is given in Brandrup et al.. Densities [kg/m^3], heat capacities [J/kg/K] and saturation pressures [Pa] can be found bellow (Kiparissides et al., 1997):

$$\rho_{\mathrm{W}} = 1011.0 - 0.4485\theta \tag{C.0.1}$$
$$c_{p,\mathrm{W}} = 4.02\exp(-1.5366 \cdot 10^{-2}T) \tag{C.0.2}$$
$$p_{\mathrm{W}}^{\mathrm{sat}} = exp(72.55 - 7206.7/T - 7.1386\log(T) + 4.046 \cdot 10^{-6}T^2) \tag{C.0.3}$$

$$\rho_{\mathrm{M}}^{\mathrm{l}} = 947.1 - 1.746\theta - 3.24 \cdot 10^{-3}\theta^2 \tag{C.0.4}$$
$$c_{p,\mathrm{M}} = 4.178(18.67 + 0.0758\theta)/62.5 \tag{C.0.5}$$
$$p_{\mathrm{M}}^{\mathrm{sat}} = \exp(126.85 - 5760.1/T - 17.914\log(T) + 2.4917 \cdot 10^{-2}T) \tag{C.0.6}$$

$$\rho_{\mathrm{P}} = 1000\exp(0.4296 - 3.2743 \cdot 10^{-4}T) \tag{C.0.7}$$
$$c_{p,\mathrm{P}} = 0.934 \tag{C.0.8}$$

$$\theta = T - 273.15 \tag{C.0.9}$$

The density of the gaseous Vinyl Chloride Monomer (VCM) is calculated from the viral equation of state. The necessary C-code to perform this calculation can be found below. Note that molecular weights, gas constant and reactor volume are assumed to be global variables.

```
double rho_vcm_g(            // Out: Dens. for water, kg/m3
    double T,                // In: temperature, K
    double p                 // In: Pressure, Pa
    )
{
    double rho, Bm, Bw, Bmw;

    // Obtaining fugacity parameters from fugpar-function
    fugpar( &Bm, &Bw, &Bmw, T );

    // Calculating density from viral equation
    rho = MWvcm * p / ( Rgas * T + Bm * p );

    return rho;
}
```

```
void fugpar(
    double *Bm,              // Out: [m3/kmol]
    double *Bw,              // Out: [m3/kmol]
    double *Bmw,             // Out: [m3/kmol]
    double T                 // In:  [K]
    )
{
    double Tc_m  = 432.0;    // [K]
    double Tc_w  = 647.5;    // [K]
    double Tc_mw = 528.9;    // [K]
    double Pc_m  = 56.e5;    // [Pa]
    double Pc_w  = 220.5e5;  // [Pa]
    double Pc_mw = 107.e5;   // [Pa]
    double acfm  = 0.1048;   // [-]
    double acfw  = 0.3342;   // [-]
    double acfmw = 0.2195;   // [-]
    double Trm, Trw, Trmw;

    Trm  = T/Tc_m;
    Trw  = T/Tc_w;
    Trmw = T/Tc_mw;

    *Bm  = Rgas*Tc_m /Pc_m  *((0.083-0.422/pow(Trm,1.6))
               -acfm *(0.139-0.172/pow(Trm,4.2)));
    *Bw  = Rgas*Tc_w /Pc_w  *((0.083-0.422/pow(Trw,1.6))
               -acfw *(0.139-0.172/pow(Trw,4.2)));
    *Bmw = Rgas*Tc_mw/Pc_mw *((0.083-0.422/pow(Trmw,1.6))
               -acfmw*(0.139-0.172/pow(Trmw,4.2)));
}
```

Appendix D

Monomer distribution: Derivation and code

This derivation and the code snippets given is re-stated from the preliminary project. This is done to show the reader how parts of the monomer distribution (Appendix A.3.2) are derived and calculated.

Code given in this Appendix is printed with permission from Peter Singstad, Cybernetica AS. The following snippets of the monomer distribution calculation that is referred to in the thesis. Note that molecular weights, gas constant and reactor volume are assumed to be global variables. Density calculations can be found in Appendix C.

D.1 Monomer in gas phase

To calculate the mass of monomer in gas phase, the following derivation was done.

$$\begin{aligned} m_{\mathrm{M}}^{\mathrm{g}} &= V_{\mathrm{m}}^{\mathrm{g}} \rho_{\mathrm{M}}^{\mathrm{g}} \\ &= V^{\mathrm{g}} y_{\mathrm{M}} \rho_{\mathrm{M}}^{\mathrm{g}} \\ &= (1 - w_{\mathrm{l}}) V_{\mathrm{R}} y_{\mathrm{M}} \rho_{\mathrm{M}}^{\mathrm{g}} \end{aligned} \tag{D.1.1}$$

Here, V_{R} represent the total reactor volume and w_{l} represent the liquid volume fraction in the reactor and is given by the following equation.

$$w_{\mathrm{l}} = \left(\frac{m_{\mathrm{M}} - m_{\mathrm{M}}^{\mathrm{g}}}{\rho_{\mathrm{M}}^{\mathrm{l}}} + \frac{m_{\mathrm{P}}}{\rho_{\mathrm{P}}} + \frac{m_{\mathrm{W}} - m_{\mathrm{W}}^{\mathrm{g}}}{\rho_{\mathrm{W}}^{\mathrm{l}}} \right) / V_{\mathrm{R}} \tag{D.1.2}$$

Inserting this in Equation D.1.1 and rearranging

$$\begin{aligned} m_{\mathrm{M}}^{\mathrm{g}} &= \left[V_{\mathrm{R}} - \left(\frac{m_{\mathrm{M}} - m_{\mathrm{M}}^{\mathrm{g}}}{\rho_{\mathrm{M}}^{\mathrm{l}}} + \frac{m_{\mathrm{P}}}{\rho_{\mathrm{P}}} + \frac{m_{\mathrm{W}} - m_{\mathrm{W}}^{\mathrm{g}}}{\rho_{\mathrm{W}}^{\mathrm{l}}}\right)\right] y_{\mathrm{M}}\rho_{\mathrm{M}}^{\mathrm{g}} \\ &= \left[V_{\mathrm{R}} - \left(\frac{m_{\mathrm{M}}}{\rho_{\mathrm{M}}^{\mathrm{l}}} + \frac{m_{\mathrm{P}}}{\rho_{\mathrm{P}}} + \frac{m_{\mathrm{W}}}{\rho_{\mathrm{W}}^{\mathrm{l}}}\right) + \frac{m_{\mathrm{M}}^{\mathrm{g}}}{\rho_{\mathrm{M}}^{\mathrm{l}}} + \frac{m_{\mathrm{W}}^{\mathrm{g}}}{\rho_{\mathrm{W}}^{\mathrm{l}}}\right] y_{\mathrm{M}}\rho_{\mathrm{M}}^{\mathrm{g}} \\ &= \left[V_{\mathrm{R}} - V_{\mathrm{fluid,s}} + \frac{m_{\mathrm{M}}^{\mathrm{g}}}{\rho_{\mathrm{M}}^{\mathrm{l}}} + \frac{m_{\mathrm{W}}^{\mathrm{g}}}{\rho_{\mathrm{W}}^{\mathrm{l}}}\right] y_{\mathrm{M}}\rho_{\mathrm{M}}^{\mathrm{g}} \end{aligned} \tag{D.1.3}$$

Here, $V_{\mathrm{fluid_s}}$ is given by

$$V_{\mathrm{fluid,s}} = \frac{m_{\mathrm{M}}}{\rho_{\mathrm{M}}^{\mathrm{l}}} + \frac{m_{\mathrm{P}}}{\rho_{\mathrm{P}}} + \frac{m_{\mathrm{W}}}{\rho_{\mathrm{W}}^{\mathrm{l}}} \tag{D.1.4}$$

and the mass of water in gas phase, $m_{\mathrm{W}}^{\mathrm{g}}$

$$\begin{aligned} m_{\mathrm{W}}^{\mathrm{g}} &= n_{\mathrm{W}}^{\mathrm{g}} M_{\mathrm{W}} \\ &= \frac{p_{\mathrm{R}} V_{\mathrm{W}}^{\mathrm{g}} M_{\mathrm{W}}}{z R^{\mathrm{g}} T^{\mathrm{g}}} \\ &= \frac{p_{\mathrm{R}} V^{\mathrm{g}} y_{\mathrm{W}} M_{\mathrm{W}}}{z R^{\mathrm{g}} T^{\mathrm{g}}} \\ &= \frac{p_{\mathrm{R}} V_{\mathrm{M}}^{\mathrm{g}} y_{\mathrm{W}} M_{\mathrm{W}}}{z R^{\mathrm{g}} T^{\mathrm{g}} y_{\mathrm{M}}} \\ &= \frac{p_{\mathrm{R}} m_{\mathrm{M}}^{\mathrm{g}} y_{\mathrm{W}} M_{\mathrm{W}}}{z R^{\mathrm{g}} T^{\mathrm{g}} y_{\mathrm{M}} \rho_{\mathrm{M}}^{\mathrm{g}}} \end{aligned} \tag{D.1.5}$$

Equation D.1.3 then becomes

$$m_{\mathrm{M}}^{\mathrm{g}} = \left[V_{\mathrm{R}} - V_{\mathrm{fluid,s}} + \frac{m_{\mathrm{M}}^{\mathrm{g}}}{\rho_{\mathrm{M}}^{\mathrm{l}}} + \frac{\frac{p_{\mathrm{R}} m_{\mathrm{M}}^{\mathrm{g}} y_{\mathrm{W}} M_{\mathrm{W}}}{z R^{\mathrm{g}} T^{\mathrm{g}} y_{\mathrm{M}} \rho_{\mathrm{M}}^{\mathrm{g}}}}{\rho_{\mathrm{W}}^{\mathrm{l}}}\right] y_{\mathrm{M}}\rho_{\mathrm{M}}^{\mathrm{g}} \tag{D.1.6}$$

Rearrangement gives the following equation.

$$m_{\mathrm{M}}^{\mathrm{g}} = \frac{(V_{\mathrm{R}} - V_{\mathrm{fluid,s}}) y_{\mathrm{M}} \rho_{\mathrm{M}}^{\mathrm{g}}}{1 - \frac{\rho_{\mathrm{M}}^{\mathrm{g}} y_{\mathrm{M}}}{\rho_{\mathrm{M}}^{\mathrm{l}}} - \frac{p_{\mathrm{R}} y_{\mathrm{W}} M_{\mathrm{W}}}{z R^{\mathrm{g}} T^{\mathrm{g}} \rho_{\mathrm{W}}^{\mathrm{l}}}}} \tag{D.1.7}$$

Which was to be derived. How to calculate the z-factor can be found in Appendix D.5.

D.2 Monomer in polymer phase

To calculate the mass of monomer in the polymer phase, the following derivation was done.

$$\begin{aligned} m_{\mathrm{M}}^{\mathrm{p}} &= V_{\mathrm{M}}^{\mathrm{p}} \rho_{\mathrm{M}}^{\mathrm{l}} \\ &= V^{\mathrm{p}} (1 - \varphi) \rho_{\mathrm{M}}^{\mathrm{l}} \end{aligned} \tag{D.2.1}$$

Here, φ represent the volume fraction of polymer in polymer phase. Substituting for the volume of the particle phase;

$$m_{\mathrm{M}}^{\mathrm{p}} = \left(\frac{m_{\mathrm{M}}^{\mathrm{p}}}{\rho_{\mathrm{M}}^{\mathrm{l}}} + \frac{m_{\mathrm{P}}}{\rho_{\mathrm{P}}} \right) (1-\varphi) \rho_{\mathrm{M}}^{\mathrm{l}} \tag{D.2.2}$$

Solving with respect to $m_{\mathrm{M}}^{\mathrm{p}}$ gives the following equation.

$$m_{\mathrm{M}}^{\mathrm{p}} = \frac{m_{\mathrm{P}} \rho_{\mathrm{M}}^{\mathrm{l}}}{\rho_{\mathrm{M}}^{\mathrm{p}}} \left(\frac{1-\varphi}{\varphi} \right) \tag{D.2.3}$$

Which was to be derived.

D.3 Monomer distribution for Interval I and II

The following code was used to calculate the monomer distribution in Interval I and II. The code for the `psat`-function can be found in Appendix D.9, the densities is calculated by equation provided in Appendix C, the code for the `z_factor` can be found in Appendix D.5.

```
int phase_distribution_free(    // Out: Value is 1 if no
                                // free phase is found
    double *p,      // Out: Pressure, Pa
    double *Tg,     // Out: Gas temperature, K
    double *mm_p,   // Out: Mass of monomer solved in polymer
    double *mm_w,   // Out: Mass of monomer solved in water
    double *mm_g,   // Out: Mass of monomer in gas phase
    double *mm_f,   // Out: Mass of monomer in free phase
    double *cm_p,   // Out: Concentration of monomer in
                    //      polymer phase, mol/m3
    double *V_l,    // Out: Volume of liquid and solids, m3
    double *V_p,    // Out: Volume of polymer phase, m3
    double *V_g,    // Out: Volume of gaseous phase, m3
    double *V_f,    // Out: Volume of free phase, m3
    double *V_w,    // Out: Volume of water phase, m3
    double T,       // In:  Temperature, K
    double mm,      // In:  Monomer mass, kg
    double mw,      // In:  Water mass, kg
    double mp,      // In:  Polymer mass, kg
    double Vr,      // In:  Reactor volume, m3
    double mm0,     // In:  Amount of total added monomer
    double X_M,     // In:  Monomer conversion
    )
{
    double rhoW, rhoP, rhoM, rhoG, vf, am, pM, pW, V_fluid_s,
    V_W_g_parameter, z, y_M, y_W;

    // Finding saturation pressures of monomer and water
    psat( &pM, &pW, T );

    // Calculating total pressure, assuming saturation pressure
    equals the partial pressure
    *p = pM + pW;
```

```
    // The temperature in gas phase will then equal the
    // temperature in liquid phase
    *Tg = T;

    // Activity is assumed to be 1 during Interval I and II
    am = 1.0;

    // Calculating densities
    rhoW = rho_w( T );
    rhoP = rho_pvc( T );
    rhoM = rho_vcm( T );
    rhoG = rho_vcm_g( *Tg, pM );

    // Calculating the volume fraction of polymer in polymer phase
    vf = Flory( am, T );

    // Calculating compressability factor
    z = z_factor(T, p);

    // Calculating volume fraction of monomer and water in gas
    phase
    y_M = p_sat_M / p;
    y_W = p_sat_W / p;

    // Parameter to account for water in gas phase
    V_g_parameter = (1 - molarmass_W * y_W * p / (rhoW * z *
    R_un_gas * (*Tg)) - y_M * rhoG / rhoM);

    // Volume of fluid and solid of all water and monomer were in
    liquid phase
    V_fluid_s = mw / rhoW + mp / rhoP + mm / rhoM;

    // Calculating the monomer distribution
    *mm_g = ( Vr - V_fluid_s ) * rhoM * y_M / V_g_parameter;
    *mm_w = 0.0088 * am * mw;
    *mm_p = rhoM / rhoP * (mp + 1e-20) * ( 1.0 - vf ) / vf;
    *mm_f = mm - *mm_g - *mm_w - *mm_p;

    // Calculating volume of gas, liquid, polymer phase, free
    phase, and water phase
    *V_g = *mm_g / rhoG / y_M;
    *V_l = V_r - V_g;
    *V_p = (*mm_p / rhoM) + (mm0 * molarmass_M * X_M) / rhoP;
    *V_f = *mm_f / rhoM;
    *V_w = *V_l - *V_p - *V_f;

    // Concentration of monomer in polymer phase
    *cm_p = *mm_p / (molarmass_M * (*V_p));

    // Mass of water in gas phase
    *mw_g = molarmass_W * y_W * (*V_g) * p / (z * R_un_gas * (*Tg)
    );

    // Checking if amount of monomer in free phase is negative.
    Then there is no free phase left, and will therefore return 1
    if ( *mm_f < 0 ) {
      *mm_f = 0.0;
```

```
        return 1;
    }

    // Still some monomer in free phase left, returning 0
    return 0;
}
```

D.4 Monomer distribution for Interval III

The following code was used to calculate the monomer distribution in Interval III. The densities is calculated by equation provided in Appendix C, the code for the `InvFloryPVC`-function can be found in Appendix D.7, the code for the `psat`-function can be found in Appendix D.9, the code for the `Tsat_vcm`-function can be found in Appendix D.10 and the code for the `z_factor`-function can be found in Appendix D.5.

```
void phase_distribution_unfree(
    double *p,      // Out: Pressure, Pa
    double *Tg,     // Out: Gas temperature, K
    double *mm_p,   // Out: Mass of monomer solved in polymer
    double *mm_w,   // Out: Mass of monomer solved in water
    double *mm_g,   // Out: Mass of monomer in gas phase
    double *mw_g,   // Out: Mass of water in gas phase
    double *cm_p,   // Out: Concentration of monomer in polymer
    phase
    double *V_l,    // Out: Volume of liquid and solids, m3
    double *V_g,    // Out: Volume of gas phase, m3
    double *V_p,    // Out: Volume of particle phase, m3
    double *V_w,    // Out: Volume of water phase, m3
    double T,       // In:  Temperature, K
    double mm,      // In:  Monomer mass, kg
    double mw,      // In:  Water mass, kg
    double mp,      // In:  Polymer mass, kg
    double Vr,      // In:  Reactor volume, m3
    double mm0,     // In:  Amount of total added monomer
    double X_M,     // In:  Monomer conversion
    )
{
    // This function assumes no free VCM in the system
    double rhoW, rhoP, rhoM, rhoG, mm_p_iter, vf, am, pM, pW,
    V_fluid_s, y_M, y_W, z;
    int i;

    // Parameters in the numerical solution
    double zeta = 0.5;
    int N = 3;

    // Finding the density of water, polymer and monomer
    rhoW = rho_w( T );
    rhoP = rho_pvc( T );
    rhoM = rho_vcm( T );

    // Assume first all VCM in polymer phase
    mm_p_iter = mm;
```

```
    // Iteratively finding a solution
    for (i = 1; i <= N; i++) {
        // Calculating the volume fraction of polymer in polymer
    phase
        vf = rhoM * mp / ( rhoM * mp + rhoP * mm_p_iter );

        // Inversely finding the activity
        InvFloryPVC( &am, T, vf );

        // Calculating the saturation pressure
        psat( &pM, &pW, T );

        if ( am > 1 ) {
            am = 1.0;
        }

        // Calculating the reactor pressure. Partial pressure of
    water is assumed to be the same as the saturation pressure.
    Saturation pressure of monomer is corrected with the activity
    to find the partial pressure
        *p = pM * am + pW;

        // Calculating the temperature of the gas phase from
    partial pressure of monomer
        *Tg = Tsat_vcm( pM*am );

        // Finding the density of monomer in gas phase
        rhoG = rho_vcm_g( *Tg, pM*am );

        // Calculating the compressability factor
        z = z_factor(*Tg, *p)

        // Parmeter to account for water being in gas phase
        V_g_parameter = (1 - molarmass_W * y_W * (*p) / (rhoW * z
    * R_un_gas * (*Tg)) - y_M * rhoG / rhoM);

        // Volume of fluid and solid of no water and monomer were
    present in gas phase
        V_fluid_s = mw / rhoW + mp / rhoP + mm / rhoM;

        // Monomer distribution calculation
        *mm_g = (con_V_tot - V_fluid_s) * y_M * rho_M_g /
    V_W_g_parameter;
        *mm_w = 0.0088 * am * mw;
        *mm_p = mm - *mm_g - *mm_w;
        mm_p_iter = zeta * *mm_p + (1-zeta) * mm_p_iter;
    }

    // Volume of gas phase, and total liquid phase (water+free+
    polymer), polymer phase and water phase
    *V_g = *mm_g / rhoG / y_M;
    *V_l = Vr - *V_g;
    *V_p = (*mm_p / rhoM) + (mm0 * molarmass_M * X_M) / rhoP;
    *V_w = *V_l - *V_p

    // Concentration of monomer in polymer phase
```

```
    *cm_p = *mm_p / (molarmass_M * (*V_p));

    // Mass of water in gas phase
    *mw_g = molarmass_W * y_W * (*V_g) * (*p) / (z * R_un_gas * (*
    Tg));
}
```

D.5 z-factor

To calculate the compressability factor, z, the following code was used. The code for the `fugpar`-function can be found in Appendix C.

```
double z_factor(    // Out: Compressibility factor z
    double T_R,     // In: temperature, K
    double p        // In: pressure, Pa
    )
{
    double B_m, B_w, B_mw;

    // Obtaining fugacity parameters from fugpar-function
    fugPar(T_R, &B_m, &B_w, &B_mw);

    // Calculating the compressibility factor from viral equation
    z = 1 + (B_2 * p) / (R_un_gas * T_R * 1000);

    return z;
}
```

D.6 φ for Interval I and II

To calculate the volume fraction of polymer in polymer phase for Interval I and II, φ, the following code was used. Note that the activity, α_{M}, will be one for Interval I and II. This method solves the Flory-Huggins equation for Poly-Vinyl Chloride (PVC). The code for the `coeffFlory`-function can be found in Appendix D.8. Referance: Kiparissides et al., 1997, Ind.Eng. Chem. Res., 36,1253-1267

```
double Flory(       // Out: volume fraction polymer in polymer
    phase.
    double T,       // In: temperature, K
    double am       // In: activity
    )
{

    if (am < 1.0e-15) {
        return 1.0;
    } else if (am > 1.0) {
        am = 1.0;
    }

    double c0;
```

```
    double c1;
    double c2;
    double c3;
    double c4;
    coeffFlory(T, am, &c0, &c1, &c2, &c3, &c4);

    double x = 1.0 - 0.25 *am * am * am * (1.0 + (T - 323.15) /
    500);

    double r, dr;
    for (int i = 0; i < 4; i++) {
        r = log(1 - x) + ((((c4)*x + (c3))*x + (c2))*x + (c1))*x +
     c0;
        dr = 1 / (x - 1) + ((4 * (c4)*x + 3 * (c3))*x + 2 * c2)*x
    + c1;
        x = x - r / dr;
    }

    return x;
}
```

D.7 α_{M} for Interval III

To calculate the activity coefficient for Vinyl Chloride Monomer (VCM), α_{M}, solved in PVC, for Interval III, the following code was used. This method solves the Flory-Huggins equation for PVC. The code for the `coeffFlory`-function can be found in Appendix D.8. Referance: Kiparissides et al., 1997, Ind.Eng. Chem. Res., 36,1253-1267.

```
double InvFlory(    // Out: monomer activity
    double T,       // In: temperature, K
    double vf       // In: volume fraction of polymer in polymer
    phase
    )
{
    double c0;
    double c1;
    double c2;
    double c3;
    double c4;
    coeffFlory(T, 1.0, &c0, &c1, &c2, &c3, &c4);

    if (vf >= 1.0) {
        return 0.0;
    } else if (vf <= 0.0) {
        return 1.0;
    } else {
        c0 = log(1 - vf) + ((((c4) * vf + (c3)) * vf + (c2)) * vf
    + (c1)) * vf;
        return exp(c0);
    }
}
```

D.8 Flory-Huggins coefficients

The code for the `coeffFlory`-function used in `Flory` and `InvFlory`. Returns parameters for Flory-Huggins equation for PVC. Referance: Kiparissides et al., 1997, Ind.Eng. Chem. Res., 36,1253-1267.

```
void coeffFlory(
    double T,
    double am,
    double *c0,
    double *c1,
    double *c2,
    double *c3,
    double *c4
    )
{
    double Xs = 0.26;
    double a = 0.15524;
    double b = 0.35311;
    double c = -0.50527;
    double d = 11.3605;
    double e = 199.96;
    double f = 6244.49;

    *c4 = f * b / T;
    *c3 = f * c / T;
    *c2 = Xs + e / T + f*a / T + f*d / (T*T);
    *c1 = 1.0;
    *c0 = -log(am);
}
```

D.9 Saturation pressure

The code for the `psat`-function. Referance: Kiparissides et al., 1997, Ind.Eng. Chem. Res., 36,1253-1267.

```
void psat(
  double T,           // In: reactor temperature
  double *p_sat_M,    // Out: saturation pressure of monomer
  double *p_sat_W     // Out: saturation pressure of water
  )
{
  *p_sat_M = exp(126.85 - 5760.1 / T - 17.914*log(T) + 2.4917e-2 *
     T);
  *p_sat_W = exp(72.55 - 7206.7 / T - 7.1386*log(T) + 4.046e-6 * T
     * T);
}
```

D.10 Temperature of the gas phase in Interval III

The code for the `Tsat_vcm`-function used in `phase distribution unfree`-function.

```
double Tsat_vcm(    // Out: Saturation temperature of VCM
    double p_R      // In: reactor pressure
    )
{
    double a = 5e5 / 20;
    double T_g = (p_R - 1e6) / a + 60 + TC0;
    double p_sat_M;
    double p_sat_W;;

    for (int i = 0; i < 5; i++) {
        psat(T_g, &p_sat_M, &p_sat_W);
        T_g = (p_R - p_sat_M) / a + T_g;
    }
    return T_g;
}
```

Appendix E

Moment balances: Derivation

The moment balances for emulsion polymerization process have been extensively studied the last decades, and can easily be obtained from the literature. During the preliminary project, they were anyway derived to confirm that the balances found in the literature were applicable for this system. The derivation of the zeroth live moment was stated in the preliminary report as an example and is re-stated below.

E.1 General relations

In the derivation, chain transfer to monomer, Chain Transfer Agent (CTA) and polymer is taken into account. The following relations are used for deriving the moment balances (Asua, 2008).

$$[\mu_k] = \sum_{i=1}^{\infty} i^k [P_i] \tag{E.1.1}$$

$$[\nu_k] = \sum_{i=1}^{\infty} i^k [D_i] \tag{E.1.2}$$

$$\frac{\mathrm{d}[\mu_k]}{\mathrm{d}t} = \sum_{n=1}^{\infty} \frac{\mathrm{d}(i^k [P_i])}{\mathrm{d}t} \tag{E.1.3}$$

$$\frac{\mathrm{d}[\nu_k]}{\mathrm{d}t} = \sum_{n=1}^{\infty} \frac{\mathrm{d}(i^k [D_i])}{\mathrm{d}t} \tag{E.1.4}$$

$$\frac{\mathrm{d}[P_i]}{\mathrm{d}t} = \left(R_\mathrm{I} + (k_\mathrm{ct}^\mathrm{M}[\mathrm{M}] + k_\mathrm{ct}^\mathrm{AX}[\mathrm{AX}]) \sum_{j=1}^{\infty}[P_j] \right) \delta(i-1) + k_\mathrm{p}[\mathrm{M}]([P_{i-1}] - [P_i]) - \left(k_\mathrm{ct}^\mathrm{M}[\mathrm{M}] + k_\mathrm{ct}^\mathrm{AX}[\mathrm{AX}] + k_\mathrm{t} \sum_{j=1}^{\infty}[P_j] \right)[P_i] + k_\mathrm{ct}^\mathrm{P} i[D_i] \sum_{j=1}^{\infty}[P_j] - k_\mathrm{ct}^\mathrm{P}[P_i] \sum_{j=1}^{\infty} j[D_j] \tag{E.1.5}$$

where P_i represent live polymer chain of length i, D_i represent dead polymer chain of length i and $\delta(i-1)$ represent the Kronecker delta, where $\delta(i-1) = 1$ when $i = 1$, and 0 otherwise.

E.2 Zeroth live moment

The zeroth live moment can be described by the following equation,

$$\frac{\mathrm{d}[\mu_0]}{\mathrm{d}t} = \frac{\mathrm{d}[P_1]}{\mathrm{d}t} + \sum_{i=2}^{\infty} \frac{\mathrm{d}[P_i]}{\mathrm{d}t} \tag{E.2.1}$$

where the following relations apply,

$$\frac{\mathrm{d}[P_1]}{\mathrm{d}t} = R_\mathrm{I} + \left(k_\mathrm{ct}^\mathrm{M}[\mathrm{M}] + k_\mathrm{ct}^\mathrm{AX}[\mathrm{AX}] \right) \sum_{j=1}^{\infty}[P_j] + k_\mathrm{p}[\mathrm{M}]([P_0] - [P_1]) - \left(k_\mathrm{ct}^\mathrm{M}[\mathrm{M}] + k_\mathrm{ct}^\mathrm{AX}[\mathrm{AX}] + k_\mathrm{t} \sum_{j=1}^{\infty}[P_j] \right)[P_1] + k_\mathrm{ct}^\mathrm{P}[D_1] \sum_{j=1}^{\infty}[P_j] - k_\mathrm{ct}^\mathrm{P} \sum_{j=1}^{\infty} j[D_j] \tag{E.2.2}$$

$$\sum_{i=2}^{\infty} \frac{\mathrm{d}[P_i]}{\mathrm{d}t} = \sum_{i=2}^{\infty} \Big(k_\mathrm{p}[\mathrm{M}]([P_{i-1}] - [P_i]) - \big(k_\mathrm{ct}^\mathrm{M}[\mathrm{M}] + k_\mathrm{ct}^\mathrm{AX}[\mathrm{AX}] + k_\mathrm{t} \sum_{j=1}^{\infty}[P_j] \big)[P_i] + k_\mathrm{ct}^\mathrm{P} i[D_i] \sum_{j=1}^{\infty}[P_j] - k_\mathrm{ct}^\mathrm{P} \sum_{j=1}^{\infty} j[D_j] \Big) \tag{E.2.3}$$

This give the following expression describing the zeroth live moment.

$$\frac{\mathrm{d}[\mu_0]}{\mathrm{d}t} = R_\mathrm{I} + \left(k_\mathrm{ct}^\mathrm{M}[\mathrm{M}] + k_\mathrm{ct}^\mathrm{AX}[\mathrm{AX}] \right) \sum_{j=1}^{\infty}[P_j] + k_\mathrm{p}[\mathrm{M}]([P_0] - [P_1]) - \left(k_\mathrm{ct}^\mathrm{M}[\mathrm{M}] + k_\mathrm{ct}^\mathrm{AX}[\mathrm{AX}] + k_\mathrm{t} \sum_{j=1}^{\infty}[P_j] \right)[P_1] +$$

$$k_{\mathrm{ct}}^{\mathrm{P}}[D_1]\sum_{j=1}^{\infty}[P_j] - k_{\mathrm{ct}}^{\mathrm{P}}\sum_{j=1}^{\infty} j[D_j] + \sum_{i=2}^{\infty}\Big(k_{\mathrm{p}}[\mathrm{M}]([P_{i-1}] - [P_i])$$
$$-\Big(k_{\mathrm{ct}}^{\mathrm{M}}[\mathrm{M}] + k_{\mathrm{ct}}^{\mathrm{AX}}[\mathrm{AX}] + k_{\mathrm{t}}\sum_{j=1}^{\infty}[P_j]\Big)[P_i] +$$
$$k_{\mathrm{ct}}^{\mathrm{P}} i[D_i]\sum_{j=1}^{\infty}[P_j] - k_{\mathrm{ct}}^{\mathrm{P}}\sum_{j=1}^{\infty} j[D_j]\Big)$$
$$\frac{\mathrm{d}[\mu_0]}{\mathrm{d}t} = R_{\mathrm{I}} + (k_{\mathrm{ct}}^{\mathrm{M}}[\mathrm{M}] + k_{\mathrm{ct}}^{\mathrm{AX}}[\mathrm{AX}])[\mu_0] - k_{\mathrm{p}}[\mathrm{M}][P_1] - \Big(k_{\mathrm{ct}}^{\mathrm{M}}[\mathrm{M}] +$$
$$k_{\mathrm{ct}}^{\mathrm{AX}}[\mathrm{AX}] + k_{\mathrm{t}}[\mu_0]\Big)[P_1] + k_{\mathrm{ct}}^{\mathrm{P}}[D_1][\mu_0] - k_{\mathrm{ct}}^{\mathrm{P}}[\nu_1] + k_{\mathrm{p}}[\mathrm{M}]\Big(\sum_{i=2}^{\infty}[P_{i-1}] -$$
$$\sum_{i=2}^{\infty}[P_i]\Big) - \Big(k_{\mathrm{ct}}^{\mathrm{M}}[\mathrm{M}] + k_{\mathrm{ct}}^{\mathrm{AX}}[\mathrm{AX}] + k_{\mathrm{t}}[\mu_0]\Big)\sum_{i=2}^{\infty}[P_i] +$$
$$k_{\mathrm{ct}}^{\mathrm{P}}[\mu_0]\sum_{i=2}^{\infty} i[D_i] - k_{\mathrm{ct}}^{\mathrm{P}}[\nu_1]\sum_{i=2}^{\infty}[P_i]$$

Using the following relations

$$[P_0] = 0 \tag{E.2.4}$$
$$\sum_{i=2}^{\infty}[P_{i-1}] = [\mu_0] \tag{E.2.5}$$
$$\sum_{i=2}^{\infty}[P_i] = [\mu_0] - [P_1] \tag{E.2.6}$$
$$\sum_{i=2}^{\infty} i[D_i] = [\nu_1] - [D_1] \tag{E.2.7}$$

$$\frac{\mathrm{d}\mu_0}{\mathrm{d}t} = R_{\mathrm{I}} + ([\mu_0] - [P_1])(k_{\mathrm{ct}}^{\mathrm{M}}[\mathrm{M}] + k_{\mathrm{ct}}^{\mathrm{AX}}[\mathrm{AX}]) + k_{\mathrm{t}}[P_1][\mu_0] - k_{\mathrm{p}}[\mathrm{M}][P_1]$$
$$+ k_{\mathrm{ct}}^{\mathrm{P}}[D_1][\mu_0] - k_{\mathrm{ct}}^{\mathrm{P}}[P_1][\nu_1] + k_{\mathrm{p}}[\mathrm{M}][P_1] - (k_{\mathrm{ct}}^{\mathrm{M}}[\mathrm{M}] + k_{\mathrm{ct}}^{\mathrm{AX}}[\mathrm{AX}]$$
$$+ k_{\mathrm{t}}[\mu_0])([\mu_0] - [P_1]) + k_{\mathrm{ct}}^{\mathrm{P}}[\mu_0]([\nu_1] - [D_1]) - k_{\mathrm{ct}}^{\mathrm{P}}([\mu_0] - [P_1])[\nu_1]$$

The zeroth live moment is then given by the following equation

$$\frac{\mathrm{d}[\mu_0]}{\mathrm{d}t} = R_{\mathrm{I}} - k_{\mathrm{t}}[\mu_0]^2 = R_{\mathrm{I}} - (k_{\mathrm{tc}} + k_{\mathrm{td}})[\mu_0]^2 \tag{E.2.8}$$

Appendix F

Introduction to Cybernetica's software

The purpose of this chapter is to introduce the software utilized in this project. Section F.1 consist of a short description of how the process model has been implemented. Cybernetica's ModelFit will then be introduced in Section F.2, which was used both during the preliminary work and in the initial phase of this project. ModelFit was used both for simulation, off-line parameter estimation, and tuning the Kalman Filter (KF). A short example of an off-line estimation will be presented. Section F.3 will give an introduction to Cybernetica CENIT, which consist of a collection of components, and was the software used for Nonlinear Model Predictive Control (NMPC). Section F.4 introduces Cybernetica RealSim, which will work as the plant replacement process simulator for the application.

F.1 Process model

The process model was implemented in C code as a set of Ordinary Differential Equations (ODEs) in one of Cybernetica's template for polymer reactions. The process model, as well as specific code for implementation of estimators and controllers, is contained in *Cybernetica Model and Application Component*, which is linked to *Cybernetica CENIT*. Inside this component, all states, inputs, outputs, constants, parameters and constraints can be defined. Reactor temperature, pressure, monomer conversion and some quality parameters, are outputs that are of particular interest. To solve these ODEs, several different solvers are available, as Euler integration, 2nd order Runge Kutta (RK2) and Sundials CVODE. In this case, Sundials CVODE was chosen for integration.

The template encourages a separation between the periphery model and reaction kinetics. They are implemented in separate files in the template, and one may easily be changes independently of the other. This is done to make

implementation trivial, and changes can be done with minimal effort. Different reaction kinetics can in such way be tested with variable periphery models, for example, with different cooling systems or reactor design. From another perspective can, for example, different chemical processes be tested with the same reactor design.

F.2 ModelFit

The template is compatible with Cybernetica's simulation tool, ModelFit. This program gives the opportunity to plot the result from a simulation easily. ModelFit is initialized with values from the template, but values can also be changes inside this environment. In addition to doing the simulation, initial values, parameters values, constants, measurement noise and validity, measured values and inputs can also be edited inside this environment. Multiple datasets with different operating conditions, constant and parameter values may easily be simulated.

ModelFit can be used for off-line estimation of parameters and initial values. The user may choose which parameter or initial value to be optimized, and which measurements it should be optimized with respect to. Afterwards, the parameter or initial values may be assigned its optimized value. The simulation can then be run ballistic with its new values. This is done to adjust the model and making it a better reflection of the real plant.

As the process model may be very nonlinear, ModelFit used a Sequential Quadratic Programming (SQP) for off-line estimation of parameter and initial values. The optimization problem is shown below

$$\min_{\phi \in R^{n_\theta}} J(\phi) = \sum_{i=1}^{n_y} \frac{1}{2} (y_i - y_i^{\mathrm{M}})^\top Q_1 (y_i - y_i^{\mathrm{M}}) \tag{F.2.1a}$$

subject to the model description, for example

$$x_{k+1} = f(x_k, u_k, \theta) \tag{F.2.1b}$$

$$y_k = g(x_k, u_k, \theta) \tag{F.2.1c}$$

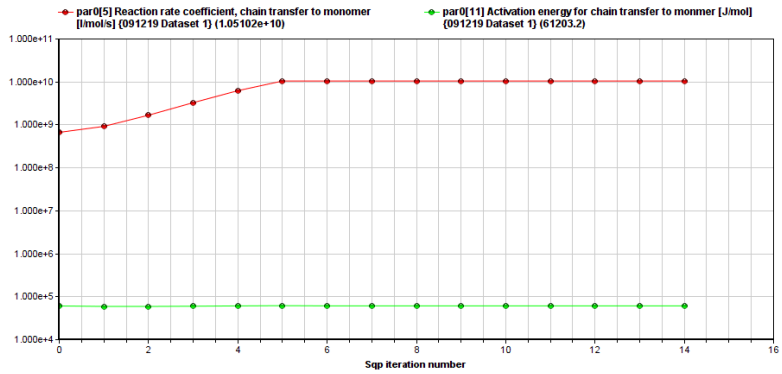
where

$$\theta_{\min} \leq \theta \leq \theta_{\max} \tag{F.2.1d}$$

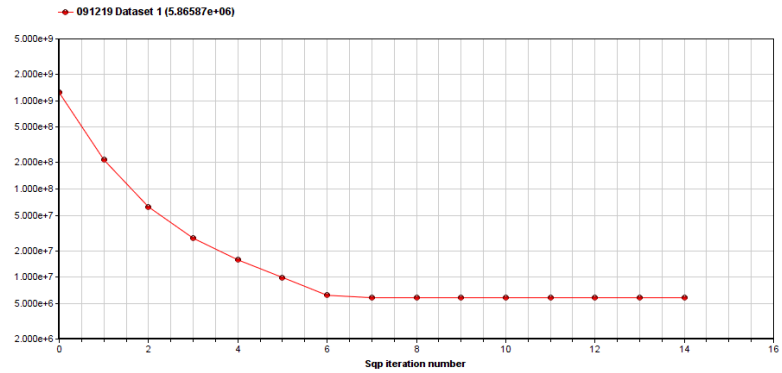
Here, n_y is the number of measurements, y is the measurement vector, y_i is the model output value of measurement number i, y_i^{M} is the measured value of measurement number i, Q_1 is a weighting matrix, x represent the states of the system, $f(\,\cdot\,)$ is the nonlinear model, $g(\,\cdot\,)$ is the measurement equations, u is the process input, θ represent the parameter vector with lower and upper boundaries, and n_θ is the number of parameters.

Figure F.1 shows an example of the ModelFit interface when preforming an off-line estimation. In this example, the reaction rate coefficient of chain transfer to monomer, $k_{\mathrm{ct}}^{\mathrm{M}}$, and the activation energy for chain transfer to monomer,

$E_{\mathrm{ct}}^{\mathrm{M}}$, have been estimated off-line. The goal was to find an Arrhenius expression to describe the reaction rate for chain transfer to monomer. The two parameters were estimated with respect to Inovyn's model for number average molecular weight, $\overline{M}_{\mathrm{n}}$. The optimized variables is shown in Figure F.1a, and the objective function is shown in Figure F.1b is shown for this estimation.



(a) Optimized variables, $k_{\mathrm{ct}}^{\mathrm{M}}$ and $E_{\mathrm{ct}}^{\mathrm{M}}$, for iteration number 0 to 14.



(b) Objective function for iteration number 0 to 14.

Figure F.1: Screenshot of ModelFit. The optimized variables (a) and the objective function (b) for iteration number 0 to 14 for a given off-line estimation, where number average molecular weight, $\overline{M}_{\mathrm{n}}$, was used as active measurement.

ModelFit can be used to decide which parameters should be estimated on-line and can be used to design on-line state- and parameter estimators. This is done in such a way that the program sees the incoming measurements as they were on-line, and performs a state- and parameter estimation to update the model with respect to some valid measurements. The estimators that is available is KF and Moving Horizon Estimation (MHE). In this case, KF was used.

ModelFit was an important tool, particularly in the preliminary project,

and also in the initial phase of this project. The model was adjusted and validated against process data. The aim was to develop a satisfactory model that could further be used for process control purposes.

F.3 CENIT

Cybernetica CENIT is a Windows application, built on top of the *Cybernetica CENIT CDK* software. CENIT consists of a collection of components; *CenitKernel*, *CenitMMI*, the process model and a database. CenitKernel is the main component. It communicates with the calculation algorithms, and with the process control system through the Open Platform Communication (OPC). Based on a prediction from the model, it performs on-line model updates and makes the controller calculations. CenitMMI is an engineering interface used mainly during the engineering phase. It is used to configure and supervise the CenitKernel. Settings can easily be adjusted inside the CenitMMI environment, for example, controller tunings, and constraints of the controlled variables and penalties on constraints violations.

CENIT is the software used for NMPC in this thesis, and is specifically designed for this purpose. CENIT uses nonlinear mechanistic models and is therefore well suited for nonlinear processes, as for example a polymerization batch reactor. The NMPC may control the input action directly, or get setpoints from a higher level, as for example a Real-Time Optimizer (RTO). The NMPC may also provide setpoints for lower levels, as for example a Proportional-Integral-Derivative Controller (PID).

F.4 RealSim

In this thesis, *Cybernetica RealSim* will work as a plant replacement process simulator, and is used for testing the on-line application and tuning of controllers. RealSim will simulate the process according to the controller input action, and it gathers the necessary measurements to the controller. At a real plant, however, RealSim will log the behaviour of the process.

The model used for plant replacement might be similar to the model used in CENIT, or the models might be different to evaluate how the controllers and estimators respond to process disturbances. As an example, in this project, will the polynomials describing the unmodelled aspects, described in Section 5.3, only be included in the plant replacement model. The estimator will then perform an on-line estimation, and update the state or parameter chosen to be estimated. These profiles may or may not be included in the NMPC prediction.

Figure F.2 shows a screenshot of RealSim while running the application. The plant replacement can either run continuously, be paused, or one sample or module can be executed. This figure shows the process measurements simulated by the plant replacement model. The process inputs, parameters and constant may be edited in RealSim if the plant is paused.

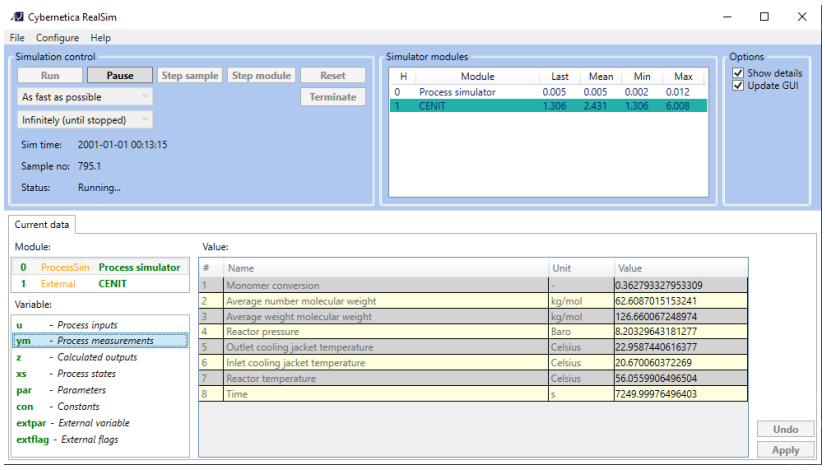


Figure F.2: Screenshot of RealSim while running the application.

Appendix G

Testing the model and the controllers in CENIT

An introductory case was conducted to test the two controllers; The Split Range Controller (SRC) and the feed of initiator to the system. All subcases in this chapter were simulated using Cybernetica's software for Nonlinear Model Predictive Control (NMPC), CENIT. Firstly, an open-loop analysis was performed, followed by tests where the controllers were set in feedback. Some comments are made to both the open-loop and closed-loop cases.

Testing the controllers in feedback is important to ensure that they are implementation is correct, and also for tuning purposes. They were tuned in such way that a change in the setpoint for the controlled variable, which here was the reactor temperature and the average number of radicals per particle, did not result in big overshoots or off-sets, neither in the simulation or prediction.

G.1 Open-loop analysis

The model was tested with no controllers in feedback. Step changes were performed in the SRC and the initiator feed. As the reactor is a semi-batch, it would be difficult to perform an open-loop analysis with ongoing chemical reactions. The chemical reaction is, therefore disconnected, and the effect on properties as the reactor temperature is examined. In the open-loop cases, the reactor is filled with water. Step changes in some of the disturbances were also performed, and the effect of them was examined. An overview of the cases is shown below, and the result will then be presented systematically.

Case 1: Step change in the SRC ($u_1 = \nu$).

Case 2: Step change in the initiator feed ($u_2 = \hat{m}_{\mathrm{I}_1}$)

Case 3: Step change in a disturbance: Temperature of the fresh water feed ($d = T_{\mathrm{fresh}}$)

G.1.1 Case 1: Step change in the SRC (ν)

Step changes were performed in the SRC, and is shown in Figure G.1. The duration of the simulation was 10 hours. The logic for the SRC was implemented as described in Section 5.2. Both step changes to yield fresh water feed, and steam were preformed, that is values of ν both over and under ν^*.

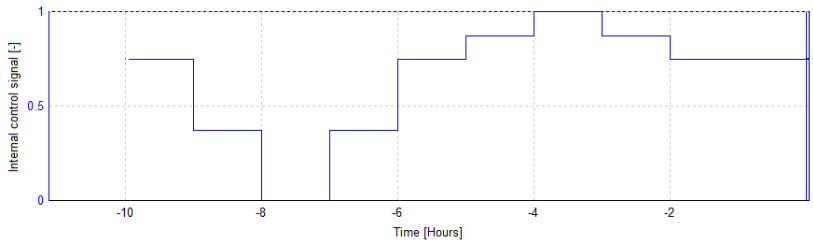


Figure G.1: Manual step changes made in the internal control signal of the SRC, ν.

The valve position for the feed of fresh water and steam is shown in Figure G.2. The SRC was manually changed so that the valve position for the fresh water feed was changed to 50%, 100%, 50% and 0%, one hour in between. The same procedure performed on the steam inlet. By the end of the simulation, both valves were closed.

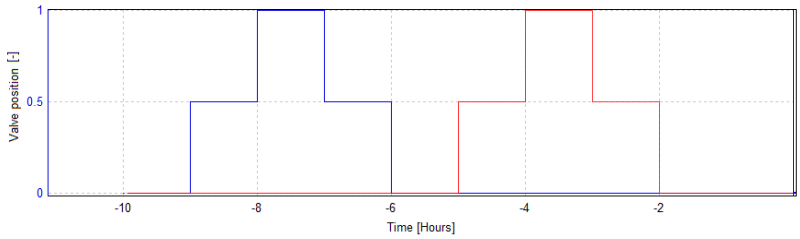


Figure G.2: Step changes in the feed of fresh water (blue) and steam inlet (red), as a result of step change in the SRC shown in Figure G.1.

The total water flow in the cooling jacket is shown in Figure G.3, with the delay of the fresh water feed taken into account and constant flow of water, is recycled. During the first hour, neither cooling nor heating was utilized, and the valves for both fresh water and steam were closed.

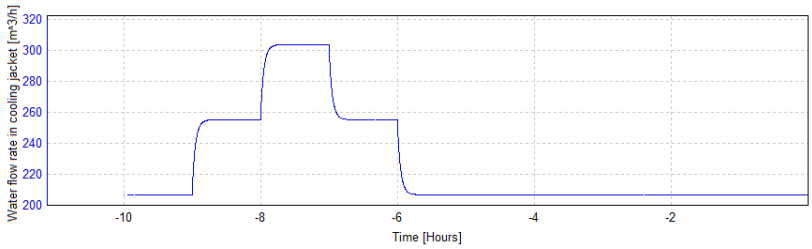


Figure G.3: Water flow in cooling jacket as a result of step change in the SRC shown in Figure G.1, with the delay in fresh water feed is taken into account ($\hat{m}_{\text{fresh}}^{\text{delayed}}$). The amount of water in the recycle was set constant.

The inlet jacket temperature, outlet jacket temperature, and the reactor temperature are shown in Figure G.4. They are all the responses to the step changes made in the internal control signal shown in Figure G.1. The starting temperature for the reactor, and in- and outlet temperatures were set equally to 57°C, and the ambient temperature to 24.95°C. As shown in the figures, the deviation in temperature between the inlet and outlet of the jacket is quite small, with a maximum deviation of approximately 3°C. Some, small, temperature drop can be seen at the beginning of the batch, even though both valves are closed at this point. The reactor temperature follows the same trend as the jacket temperature, but with some delay.

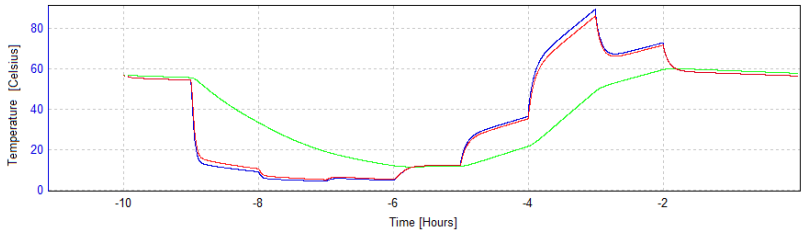


Figure G.4: The inlet jacket temperature (blue), outlet jacket temperature (red) and reactor temperature (green) corresponding to the step changes made in the SRC shown in Figure G.1.

G.1.2 Case 2: Step change in the initiator feed ($\hat{m}_{\text{I}_1}$)

Step changes were performed in the initiator feed, and is shown in Figure G.5. The integrated value, that is the total amount of initiator added, is shown in Figure G.6. The duration of the simulation was 10 hours. In this case, the feed temperature of initiator was set to a temperature of 16.85°C, and the initial temperature of the reactor and inlet- and outlet of the jacket were set to 24.90°C. The ambient temperature was set to 24.95°C.

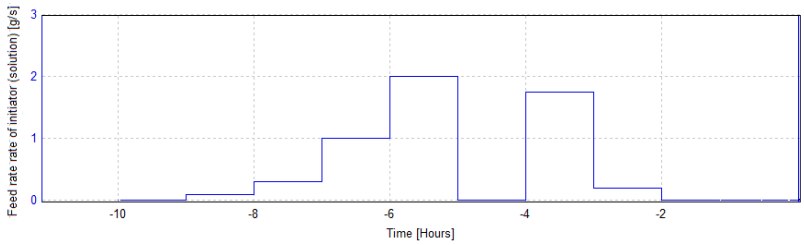


Figure G.5: Manual step changes made in the feed rate of initiator solution, $\hat{m}_{\mathrm{I}_1}$.

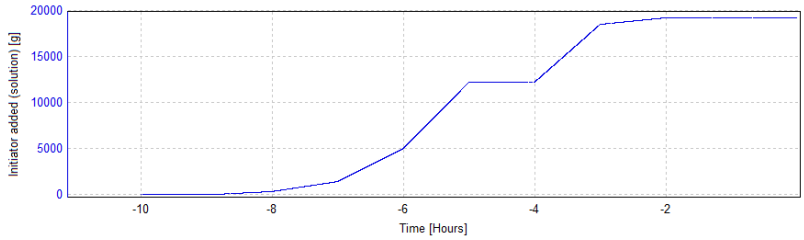


Figure G.6: The total amount of initiator added corresponding to the step changes made in the initiator feed shown in Figure G.5.

As there is a deviation between the temperature of the reactor feed and the reactor content, there will be some energy contribution from the feed. This is shown in Figure G.7.

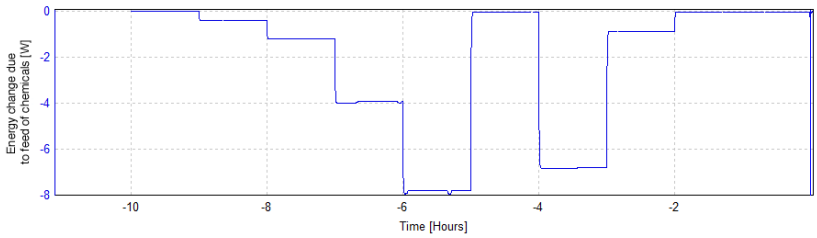


Figure G.7: Energy contribution from feed corresponding to the step changes made in the initiator feed shown in Figure G.5.

As the ambient temperature is lower than the initial temperature of the reactor and the jacket, a temperature drop will be experienced. A faster temperature drop in the jacket then in the reactor is observed. This is shown in Figure G.8.

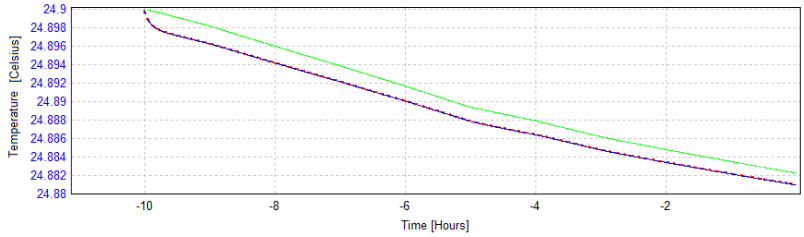


Figure G.8: Inlet cooling jacket temperature (blue), outlet cooling jacket temperature (red) and reactor temperature (green) corresponding to the step changes made in the initiator feed shown in Figure G.5.

Due to the difference in the reactor and jacket temperatures, there will be some energy transfer between the reactor and the jacket. This is shown in Figure G.9. The energy transfer is observed to be largest when the temperature deviation between the jacket and the reactor is most evident.

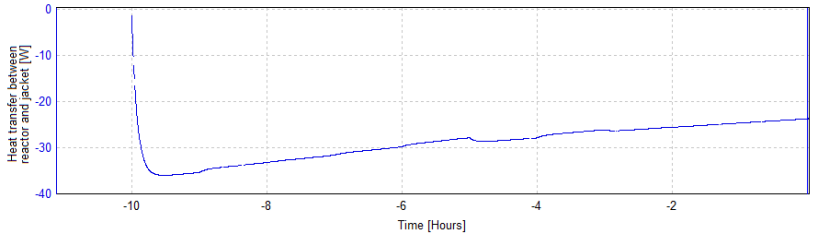


Figure G.9: Cooling effect corresponding to the step changes made in the initiator feed shown in Figure G.5.

G.1.3 Case 3: Step change in a disturbance (T_{fresh})

Step changes in the temperature of the fresh water feed (T_{fresh}) were performed. In the model, this temperature was set constant. In reality, however, one could imagine that this temperature may change depending on process operation or season. The step changes, reactor temperature, and jacket temperatures are all shown in Figure G.10. The initial temperature of the outlet and inlet of the jacket, the reactor and the fresh water feed, were set to 3°C, and the SRC was set manually to 0.375, corresponding to 50% opening of the valve for the fresh water.

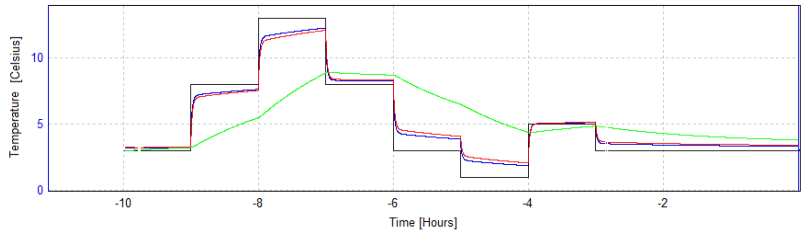


Figure G.10: Outlet and inlet temperature of jacket, and reactor temperature corresponding to the step changes in the fresh water feed.

The cooling efficiency (the energy transfer between the reactor and the jacket) is shown in Figure G.11. It is observed to be positive for reactor temperatures lower than the jacket temperatures, and negative for reactor temperatures higher than the jacket temperatures.

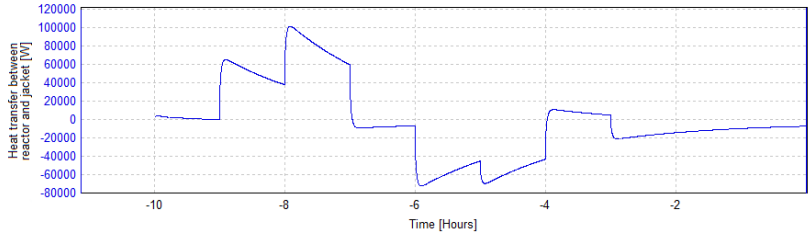


Figure G.11: Heat transfer between the reactor and the jacket corresponding to the step changes in T_{fresh} shown in Figure G.10.

G.1.4 Comments to the open-loop analysis

The open-loop analysis showed that the connection between the SRC and the valve position for fresh water and steam was implemented correctly according to the definition made in Section 5.2. The temperature profiles in Figure G.4, shows that the difference in inlet and outlet temperature of the jacket is quite small. This was also evident from the preliminary work where process measurements were examined. This implies that the flow of water in the cooling jacket is quite high and that the heat transfer per volume cooling fluid is low. The profile of the reactor temperature showed meaningful results, as it follows the temperature in the jacket, but with some delay. This is due to the heat transfer from the reactor to the jacket. During the first hour of the batch, both the jacket temperatures and the reactor experienced a small temperature drop, even though both valves were closed. This is due to heat loss to the environment.

The open-loop analysis when performing step changes in the initiator feed shows that the connection between the control signal ($\hat{m}_{\text{I}_1}$), and its corresponding accumulated value ($\hat{m}_{\text{I}_1}^{\text{acc}}$), was implemented correctly. This is evident from

Figure G.5 and G.6, where the accumulated variable represent the integrated value the feed rate. Further more, the energy contribution from feeding initiator (Figure G.7), showed reasonable results in the sense that the temperature of the fed chemicals are set lower then the calculated reactor temperature, showed in Figure G.8. This will have some cooling effect on the reactor content, evident from Figure G.9.

The open-loop analysis, when performing a step-change in the disturbance, shows that both the reactor temperature and the jacket temperature highly depends on the temperature of the fresh water feed (Figure G.10). As in the open-loop analysis, when performing step changes in the SRC, the inlet and outlet temperature of the jacket is quite similar. The reactor temperature will, if given enough time, approach the jacket temperature. This emphasizes the delay in the system. Another important aspect is that when cooling the reactor, neither the reactor temperature nor the jacket temperature can achieve any lower temperature than the inlet of the fresh water feed. The temperature of the fresh water feed will, therefore, be determinative for the cooling capacity of the system. Furthermore, Figure G.11 shows logical results with respect to the cooling effect from the jacket, as it indicates that the reactor content is heated when the jacket temperature is higher than the reactor temperature, and visa versa.

G.2 Testing the SRC

The SRC was tested by examining two different cases. Both cases had only the SRC in feedback. The chemical reactions were disconnected, so the temperature changes were only due to heating, cooling or energy transfer with the environment. An overview of the cases is shown below, and the result will then be presented.

Case 1: Setpoint changes in the reactor temperature, without reference trajectories and SRC in feedback.

Case 2: Setpoint changes in the reactor temperature, with reference trajectories and SRC in feedback.

The starting temperature of the reactor, the setpoint reactor temperature, and the inlet and outlet temperature of the jacket were set to 25°C initially. The reactor setpoint temperature was then increased by 5°C twice, followed by a decrease by 5°C four times. Finally, it was increased by 5°C twice and the end setpoint temperature is equal to 25°C. This procedure is equal for both cases.

G.2.1 Case 1: Setpoint changes in the reactor temperature, without reference trajectories, and SRC in feedback.

Setpoint changes in the reactor temperature were performed, and the SRC was set in feedback. The setpoint changes had no reference trajectories, that is, the controller will not be able to act on the step changes in advance. The response of the SRC was examined. The reactor temperature and the setpoint changes are shown in Figure G.12. To illustrate the concept of no reference trajectory, Figure G.13 is included. As the figure illustrates, there are no setpoint changes seen in the prediction. No visible overshoots were observed from this simulation.

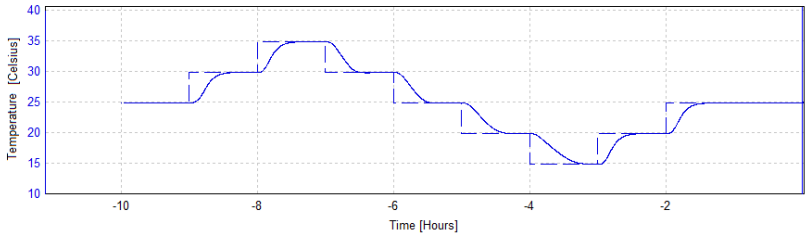


Figure G.12: Reactor temperature (solid) and reactor setpoint temperature (dashed).

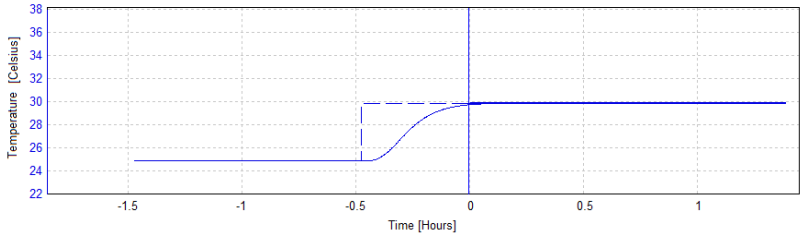


Figure G.13: Illustrating the concept of no reference trajectory of the reactor temperature (solid), with its respective setpoint (dashed). No setpoint changes is seen in the prediction. Time instant is 5310 seconds.

The valve position of steam and fresh water is shown in Figure G.14, and temperatures in the jacket is shown in Figure G.15.

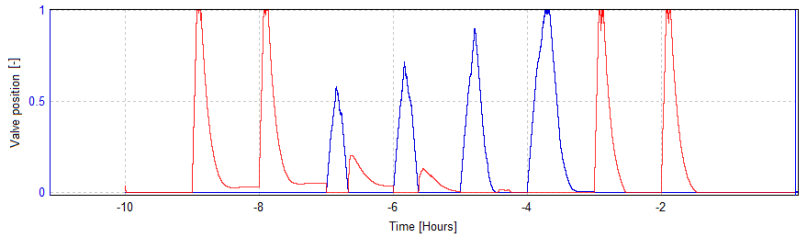


Figure G.14: The valve position of fresh water (blue) and steam (red) corresponding to the setpoint changes made in the reactor temperature shown in Figure G.12.

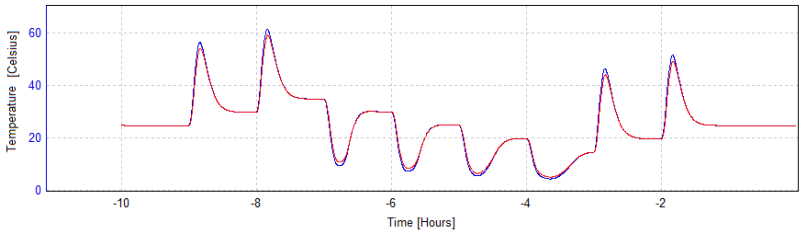


Figure G.15: The inlet jacket temperature (blue) and outlet jacket temperature (red) corresponding to the setpoint changes made in the reactor temperature shown in Figure G.12.

G.2.2 Case 2: Setpoint changes in the reactor temperature, with reference trajectories, and SRC in feedback.

Setpoint changes in the reactor temperature were performed, and the SRC was set in feedback. The reactor temperature is shown in Figure G.16. The setpoint changes had reference trajectories, such that the controller will be able to act on the step changes in advance. This is shown in Figure G.17, where to positive time frame shows the prediction of the reactor temperature and the setpoint changes. More significant overshoots can be seen from this simulation compared to the case without the setpoints in the prediction.

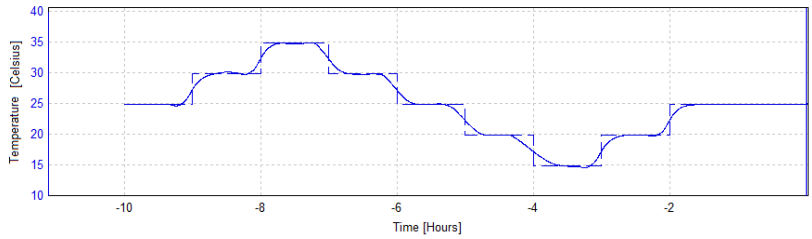


Figure G.16: Reactor temperature (solid) and reactor setpoint temperature (dashed).

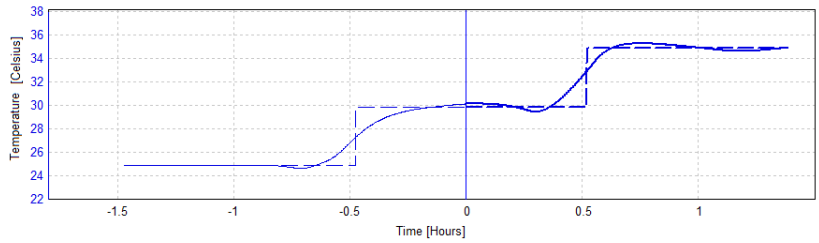


Figure G.17: Illustrating the concept of reference trajectory, showing both simulated and predicted reactor temperature (solid) and reactor setpoint temperature (dashed). Time instant is 5310 seconds.

The valve position of steam and fresh water is shown in Figure G.18, and temperatures in the jacket is shown in Figure G.19.

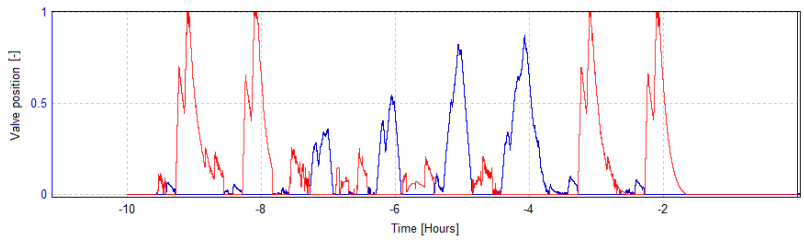


Figure G.18: The valve position of fresh water (blue) and steam (red) corresponding to the setpoint changes made in the reactor temperature shown in Figure G.16.

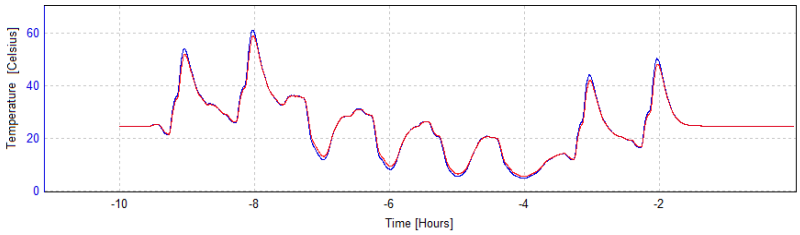


Figure G.19: The temperatures in the jacket, both inlet (blue) and outlet (red), corresponding to the setpoint changes made in the reactor temperature shown in Figure G.16.

G.2.3 Comments to the SRC testing

For both cases, the reactor temperature reaches its setpoints, and the corresponding valve positions (Figure G.14) and jacket temperatures (Figure G.15) is reasonable in that sense that opening the valve for steam leads to heating, and opening the valve for fresh water leads to cooling, both of jacket content and reactor. Also, for the cases where the setpoint for the reactor temperature over the ambient temperature, the controller will keep heating to compensate for the heat loss to the environment.

When comparing the cases with and without the reference trajectory, it is evident that the controllers are more aggressive if the reference trajectory is included. This can be seen by comparing the valve positions (Figure G.14 and G.18), and also the reactor temperature profiles (Figure G.13 and G.17) as more significant overshoots was experienced. In the case of no reference trajectory, the controllers do not start to aim for the setpoint before it is simulated by the plant replacement model. The reactor temperature is typically a controlled variable, and predictions may, therefore, exist for such variables. Reference trajectories can for example be used if the temperature is to be kept at a certain setpoint. Alternatively, some trajectory for the temperature may be pre-generated to obtain certain quality requirement.

Other variables might be disturbances, which is not as easy to predict a reference trajectory for. This can, for example, be changes in the fresh water feed or changes in the ambient temperature. Either way, the controller showed satisfactory results, making the controlled variable reach its setpoint within an acceptable range. The tuning could, however, be further explored to reduce the overshoots.

G.3 Testing the initiator feed

The initiator feed was tested by making setpoint changes in the average number of radicals per particle, $\bar{n}$. The reactor temperature should be insensitive to changes in the initiator dosage, except for the sudden change of temperature

due to the feed. It was therefore kept constant during this simulation, and the effect of the initiator dosing on $\bar{n}$ was examined.

For testing the initiator dosage, only the case without reference trajectory with the initiator feed in feedback is shown. However, a similar case can be simulated with reference trajectories, as shown for the temperature in the previous section.

The reactor is initially loaded with some Initiator 1 and 2, but Initiator 1 is the only initiator which is post-dosed. The setpoint for the average number of radicals per particle, $\bar{n}$, was initially set to 0.1, then changed to 0.125, 0.15, 0.125, 0.1, 0.75, 0.5, 0.75 and 0.1, one hour in between. The setpoint changes is shown in Figure G.20 and the feed rate of initiator is shown in Figure G.21.

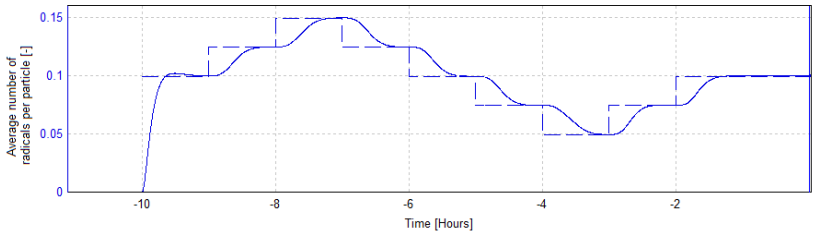


Figure G.20: The average number of radicals per particle (solid) and its setpoint (dashed).

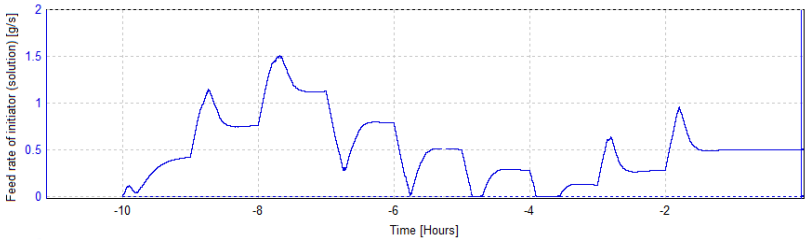


Figure G.21: The feed rate of initiator corresponding to the setpoint changes in the average number of radicals per particle, $\bar{n}$, in Figure G.20.

G.3.1 Comments to the initiator feed testing

The results of testing the controller for initiator dosing showed logical results. The desired valve for the average number of radicals per particle could easily be achieved by controlling the feed of initiator. Due to termination, continuous feed of initiator is needed to maintain the setpoint. The controlled variable does not show any significant overshoot or off-set. This indicates a good trade-off between aggressive and slow control.

Anne Øyen Halås

Master's thesis

NTNU
Norwegian University of
Science and Technology