



NTNU – Trondheim
Norwegian University of
Science and Technology

A Modeling Framework for Control of Smart-Scale Tubular Polymerization Reactors

A Case Study on Nonlinear Model-based
Predictive Control of an Emulsion
Copolymerization Process

Fredrik Gjertsen

Chemical Engineering and Biotechnology
Submission date: June 2014
Supervisor: Sigurd Skogestad, IKP
Co-supervisor: Peter Singstad, Cybernetica AS

Norwegian University of Science and Technology
Department of Chemical Engineering

Preface

This master's thesis was written during the spring of 2014 as the final part of the study program leading to the degree of M.Sc. in Chemical Engineering at the Norwegian University of Science and Technology (NTNU). I have been a student of the Process Systems Engineering group at the Department of Chemical Engineering, and the thesis has been written in close collaboration with the company Cybernetica AS, who proposed the thesis.

Above all, I would like to express my utmost gratitude towards the employees of Cybernetica AS, who have provided me with valuable support and guidance along the entire duration of my internship. I have learned a lot from numerous interesting discussions and sessions at Cybernetica, and I appreciate the flexibility I have been granted in my work. I particularly wish to thank Peter Singstad, who has been my supervisor in Cybernetica, for showing belief in me and being beyond helpful in all aspects of the work.

I also wish to thank my supervisor at NTNU, Professor Sigurd Skogestad, who deserves credit for his encouragement and support to the work. To work under the supervision of the competent staff of the Process Systems Engineering group has been a truly inspiring experience for me. In addition, I send my thankful regards to my great friends and fellow master's students in the Process Systems Engineering group, for contributing to maintaining a fruitful environment both academically and socially. Thanks to my classmates and good friends in Trondheim, I can safely declare that the time spent to obtain my master's degree has been the best five years of my life!

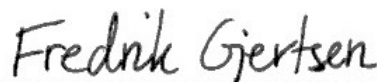
The work of this thesis has been conducted as a sidetrack to the research project COOPOL, which is a European Union collaboration project under the Seventh Framework Programme. I greet my fellow researchers and contributors to the project for all their direct and indirect help towards my thesis. The helpful behavior of the industrial partners to the project is also greatly appreciated.

Last, but not least, I acknowledge the invaluable support of my family through it all, without which my joy of life and motivation towards studying would be reduced.

Declaration of compliance

I hereby declare that this thesis is an independent work in agreement with the exam rules and regulations of the Norwegian University of Science and Technology.

June 23, 2014
Trondheim, Norway



Fredrik Gjertsen

Abstract

Polymer science is the underlying topic of this master's thesis, and the main scope is to develop and deploy models for on-line optimization and control for polymerization reactors. Specifically, free-radical emulsion copolymerization processes are studied, and the connection between lab-scale experiments to validate the models and the possible usage of the models for industrial-scale applications is maintained. In the work preceding this thesis, the author studied a semi-batch reactor setup for a specific free-radical emulsion copolymerization system, and the purpose of this thesis is to extend the previously established ideas to apply for continuous tubular reactors for a similar chemical system as well.

In going from batch type reactor setups, which are typically well-mixed tank reactors, to continuous type reactors with spatial distribution, the model equations are typically changed from ordinary differential equations to partial differential equations. A portion of the thesis is dedicated to establishing the model equations for a distributed system in general, and presenting various approaches to solve a system of partial differential equations. The thesis considers two approaches for this, where one is a spatial discretization method using finite differences to achieve a system of ordinary differential equations, while the other approach is based on a transformation of the variables in the coordinate system of the spatial problem. The respective residence time distributions for the developed models have been used to evaluate the mixing effects of the reactor models, in comparison with lab-scale experiments. Throughout the modeling work, the trade-off between accuracy and numerical efficiency for the models is central.

The mathematical models from the work are mainly formulated in the programming/modeling language Modelica. By using some of the in-house developed software of Cybernetica AS, several test cases are simulated to demonstrate how the models perform in on-line use. In order to mimic the behavior of a real plant, so-called plant replacement models are deployed. This is done to create a (slight) discrepancy between the "real" behavior of the reactor and the model on which the controller bases its calculations. In such cases, the need for an on-line estimator becomes evident, and for this thesis a Kalman filter type estimator is deployed. A theoretical foundation is established for both on-line estimation and control using model-based predictive control. The properties that are given the most attention in the context of control are the reactor outlet temperature and the conversion of monomer species through the reactor.

The results indicate that in order to encapture the mixing effects of a tubular reactor, as governed by the experiments, the models will require very refined discretization schemes in space, which are hard to obtain without compromising the numerical efficiency of the models. Despite of these observations, the tests show that models with a coarse spatial discretization tend to perform surprisingly well for the purpose of controlling the reactor temperature and the conversion of monomer. When plant replacement models are being deployed, the importance of an active and well-functioning estimator is demonstrated. Both the estimator and the controller are tuned in the simulations to yield as effective performance as possible for setpoint changes etc. In addition, test simulations are conducted in which the cooling jacket of the reactor is segmented to yield two separate degrees of freedom with respect to the cooling of the reactor. This provides a way to control the temperature dependent kinetics of the reactor indirectly, in addition to the actual reactor temperature.

Sammendrag

Det underliggende tema for denne masteravhandlingen er polymerkjemi. Hensikten er både å utvikle og bruke matematiske modeller til on-line optimalisering og styring av polymerisasjonsreaktorer. I denne oppgaven er det, mer spesifikt, fri-radikal emulsjonskopolymerisasjon som blir studert, og selv om modellene ofte er tilpasset eksperimenter i lab-skala er det av interesse å bruke modellene til industrielle anvendelser. Forfatteren har i et tidligere arbeid betraktet en semi-batchreaktor for et bestemt kopolymerisasjonssystem, og hensikten med denne masteravhandlingen er å overføre noen av de etablerte ideene til også å gjelde for en rørreaktor for et tilsvarende kjemisk system.

Når man betrakter kontinuerlige reaktorer med romlige blandedeffekter, slik som for eksempel temperatur- og konsentrasjonsgradienter gjennom reaktoren, vil likningssystemet som beskriver reaktoren (modellikningene) bli mer komplekst enn det man typisk oppnår når man betrakter typiske batchreaktorer. Modellikningene vil typisk gå fra å være ordinære differensiallikninger, som ofte er tilstrekkelige for å beskrive godt blandede batchreaktorer, til å bli partielle differensiallikninger, og en del av oppgaven er derfor bevilget til å diskutere hvordan man kan modellere et generelt system som er romlig distribuert, og fremfor alt hvilken type strategi man kan bruke for å løse det resulterende systemet av partielle differensiallikninger. Det blir presentert to fremgangsmåter i denne sammenheng, hvor den ene anvender endelige differanser (*eng.* finite differences) til å foreta en romlig diskretisering, mens den andre metoden er basert på en variabeltransformasjon av det romlige koordinatsystemet. Blandeeffektene i reaktoren blir studert ved å sammenligne teoretiske oppholdstidsfordelinger fra reaktormodelleringen med eksperimentelle resultater. Gjennom hele oppgaven hersker en underliggende bevissthet rundt avveilingen mellom nøyaktige modeller, det vil si modeller som samsvarer særdeles godt med virkeligheten, og numerisk effektive modeller, det vil si modeller som utviser kort beregningstid i ulike sammenhenger. Denne avveilingen er helt sentral når man skal designe et modellbasert regulatorssystem til on-line bruk.

Prosessmodellene som er utviklet i dette arbeidet er hovedsakelig formulert i programmerings-/modelleringsspråket Modelica. Ved å bruke den egenutviklede programvaren til Cybernetica AS for optimalisering og styring har det blitt gjennomført en rekke simuleringer for å demonstrere hvordan de utviklede modeller presterer. For å etterligne oppførselen til et ekte, realistisk reguleringsystem er det blitt innført en såkalt fabrikkertatter (*eng.* plant replacement model), hvis hensikt er å sørge for at det faktisk foreligger et avvik mellom den "ekte" reaktorens målte oppførsel og "spådommen" til modellen som regulatoren bruker i sine beregninger. På denne måten skapes det situasjoner hvor hensikten med å ha en estimator blir tydelig, og det blir benyttet en on-line estimator av typen Kalman filter i dette arbeidet. Et teoretisk grunnlag blir lagt for både on-line estimering og prosessstyring med modellbasert regulering. Temperaturregulering samt styring av omsetningsgrad av monomer gjennom reaktoren er de aktivitetene som blir viet mest oppmerksomhet i denne oppgaven, med hensyn til regulering.

Resultatene antyder at det, for å gi samsvar mellom de utviklede modellene og eksperimenter, når det gjelder blandeforholdene i reaktoren, er påkrevd å innføre en veldig fin romlig diskretisering. Dette er vanskelig å gjennomføre uten å gjøre modellen treg, dvs. numerisk ineffektiv i on-line beregninger. Til tross for denne indikasjonen viser testsimuleringer at reaktormod-

eller med ganske grov diskretisering gir overraskende gode resultater når det gjelder styring av reaktortemperatur og/eller omsetning av monomer gjennom reaktoren. Viktigheten av en velfungerende estimator blir demonstrert i simuleringene når det innføres en plant replacement-modell hvor det er avvik mellom regulatorens prediksjoner og den "ekte" reaktorens adferd. Det demonstreres også hvordan tilpasning (*eng.* tuning) av både estimatoren og regulatoren påvirker systemets oppførsel ved setpunktendringer og lignende. I tillegg blir det gjennomført regulerings-simuleringer hvor kjølekappen til reaktoren er oppdelt slik at flere forskjellige kjøletemperaturer utgjør separate frihetsgrader for systemet, og resultatene viser at man i tillegg til å kunne styre reaktortemperatur også kan styre omsetningsgraden av monomer indirekte gjennom den temperaturavhengige kinetikken i reaktoren.

Table of Contents

Preface	i
Abstract	iii
Sammendrag (Norwegian abstract)	v
List of Figures	xi
List of Tables	xii
List of Code snippets	xii
List of Acronyms and Abbreviations	xiii
List of Symbols	xiv
1 Introduction	1
1.1 Background	1
1.2 Topic: Industrial polymer chemistry	3
1.3 Challenge: Optimization and control of distributed systems	5
2 Fundamentals of free-radical emulsion copolymerization	7
2.1 Kinetics of free-radical copolymerization	7
2.2 Phase distribution in emulsions	16
2.3 Free-radical species modeling	19
2.4 Copolymer product quality	23
3 Concepts and methodology of tubular reactor modeling	25
3.1 Introduction: Problem formulation	25
3.2 Residence time distribution for flow behavior analysis	33
3.3 Solution using the numerical method of lines	35
3.4 Solution using an incremental formulation with change of variables	39
3.5 Solution using a formulation with mobile control volumes	44
3.6 Comparison of the various approaches	46
4 Software used in the work	49
4.1 Introduction to Modelica & Dymola	49
4.2 Cybernetica Modelfit	50
4.3 Cybernetica CENIT	54
5 On-line state and parameter estimation	57
5.1 Purpose and formulation of the estimator	57

5.2	Estimator tuning	61
6	Principles of model-based predictive control	63
6.1	Controller objective and formulation	63
6.2	Additional features of the controller	67
6.3	Controller tuning	68
6.4	The use of a separate model for plant representation	70
7	Simulations and results	73
7.1	Introductory case: Control of a semi-batch reactor	73
7.2	Preliminary task: Off-line parameter estimation	75
7.3	Recursive estimation simulations: Kalman filter testing	83
7.4	Control study I: Setpoint changes, conversion of monomer	85
7.5	Control study II: Setpoint changes, reactor temperature	87
7.6	Control study III: On-line estimator testing	92
7.7	Control study IV: Temperature changes with a segmented cooling jacket	94
7.8	Final control study: Control using a plant replacement model	101
7.9	Extensions to the control problem in an industrial perspective	110
8	Conclusions	115
8.1	Findings from the thesis work	115
8.2	Final conclusion	120
	References	121
A	Additional information for radical species modeling	I
B	Polymer moments for copolymer product calculations	III
C	Semi-batch reactor modeling	V
D	Shell balance derivation of balance principles, using cylindrical coordinates	IX
D.1	Transformation from cartesian type coordinates to cylindrical coordinates	IX
D.2	Shell balance derivation	XI
E	Example calculation using the numerical method of lines	XIII
E.1	Problem formulation	XIII
E.2	Deploying NMOL	XIV
F	Estimator derivation	XVII
F.1	Kalman filter estimator for linear time-discrete systems	XVII
F.2	Kalman filter estimator for linear continuous systems	XXII
F.3	Extended Kalman filter estimator for nonlinear continuous systems	XXV
G	A simplified example for software demonstration	XXIX

List of Figures

1.1	Typical work process in MPC development using mechanistic models.	2
2.1	Qualitative analogy for free-radical polymerization, using the lifetime of a typical biological cell culture in a closed-off environment (a) to illustrate the "life" of a typical system of growing free-radical polymer chains (b).	8
2.2	Mechanistic idea for initiator activation using an arbitrary peroxide compound in free-radical polymerization.	8
2.3	Mechanistic idea for initiator attack on a monomer double bond in free-radical polymerization.	9
2.4	Illustration of typical block copolymers (a) and random copolymers (b).	13
2.5	Mechanistic idea for polymer termination by chain recombination in a free-radical polymerization system.	14
2.6	Mechanistic idea for polymer termination by chain disproportionation in a free-radical system.	14
2.7	A classic illustration of emulsion polymerization in the context of phase distribution.	16
2.8	A plot showing the agreement between the three various approaches for radical species modeling, for a fictitious case. Curves show average number of radicals per particle versus dimensionless time.	21
3.1	A conceptual illustration of a tubular reactor with constant cross-sectional area.	26
3.2	An arbitrary control volume having a volume \mathcal{V} and a surface area \mathcal{A}	27
3.3	Illustration of typical RTD-behavior for a PFR.	35
3.4	Illustration of typical RTD-behavior for a dispersed model.	36
3.5	A drawing of a reactor tube which is divided into a series of discrete segments having finite extent.	38
3.6	Illustration of typical RTD-behavior for a tanks-in-series model.	40
3.7	Illustration of RTD-behavior for an incremental model with transformed variables.	43
3.8	Illustration of the effect of a smoothing factor for the differential equations at the (spatial) reactor outlet.	45
3.9	RTD-behavior for the modeling approach with well-mixed moving control volumes.	46
4.1	A figure showing the general idea of interconnecting Modelica units in a hierarchy.	51
4.2	Changes in the condition number for the scaled Hessian matrix in the model fitting calculation.	53
4.3	Changes in the identifiability ranking for the variables during model fitting calculation.	53
4.4	A block diagram showing the interconnection of the various components of the Cybernetica CENIT software.	54
5.1	A block diagram showing the interconnected components in an on-line estimator.	58
5.2	Illustrations of the various ways to constrain the updates in the Kalman filter type estimator algorithm in the Cybernetica CENIT software.	61
6.1	Two different suggestions for a typical control structure hierarchy for a plant. The role of MPC is indicated in both cases.	64

6.2	A conceptual block diagram showing the components of a typical MPC implementation with the required interconnection.	65
6.3	Typical trajectory for a process being controlled using MPC during a change in setpoint for one of the outputs (CVs).	68
6.4	Controller action illustrating the concepts of controller and prediction horizon as well as input blocking, including the corresponding system response.	69
6.5	Conceptual illustration for the distinction between the controller algorithms and the plant. In this thesis, all components of the structure are represented by computer models, but the scope is to mimic the real-life behavior as accurately as possible.	71
7.1	Controller action (MV) and process response (CV) for a temperature control case of a semi-batch reactor.	74
7.2	Simulation showing the conversion of monomer at four locations in the tubular reactor, before off-line parameter estimation has been performed.	77
7.3	Simulation showing the conversion of monomer at four locations in the tubular reactor, after off-line parameter estimation has been performed.	77
7.4	Figures showing the conversion of monomer through the reactor. The effect of modifying the partition coefficients for the phase equilibrium is illustrated for both modeling approaches of the thesis.	80
7.5	Simulation showing the conversion of monomer at four spatial locations in the tubular reactor, with optimal parameters. In this case, the solubility of monomer inside the growing polymer particles is a function of polymer particle size.	82
7.6	Reactor simulation, using an NMOL type model, without active on-line estimation and parameter estimation.	84
7.7	Reactor simulation, using an NMOL type model, with weak on-line estimation and parameter estimation due to moderate uncertainty in the measurements.	84
7.8	Reactor simulation, using an NMOL type model, with strong on-line estimation and parameter estimation due to high certainty in the measurements.	85
7.9	Controller action (MVs) for a tubular reactor with setpoint changes in the conversion of monomer. The setpoint for the reactor outlet temperature is kept constant.	86
7.10	System response (CVs) for a tubular reactor with setpoint changes in the conversion of monomer. The setpoint for the reactor outlet temperature is kept constant.	87
7.11	Input (MV) and output (CV) changes for a tubular reactor with setpoint changes in the reactor outlet temperature.	88
7.12	Input (MV) and output (CV) changes for a tubular reactor with setpoint changes in the reactor outlet temperature, with a constraint on the changes in the input.	89
7.13	Input (MV) and output (CV) changes for a tubular reactor with setpoint changes in the reactor outlet temperature, with constrained change in input. This simulation uses a model with an increased number of spatial discretization points.	90
7.14	Input (MV) and output (CV) changes for a tubular reactor with setpoint change in reactor outlet temperature, with constrained change in input. This simulation uses a reduced sampling time interval.	91
7.15	Simulated outputs (CVs) for a setpoint change in the reactor temperature with visible CV estimates and plant measurements. The estimator is tuned to deploy weak state updates.	93

7.16	Simulated outputs (CVs) for a setpoint change for the reactor temperature, showing visible CV estimates and plant measurements. The estimator is tuned to deploy strong state updates.	94
7.17	A drawing of a tubular reactor with a cooling jacket which is segmented into two individual segments. The temperatures of the cooling fluids in the respective segments are controllable.	95
7.18	Simulation results showing inputs (MVs) and outputs (CVs) for a tubular reactor with a segmented cooling jacket.	97
7.19	Simulation results showing inputs (MVs) and outputs (CVs) for a tubular reactor with a segmented cooling jacket. The tuning parameters of the controller are altered in comparison with Fig. 7.18.	98
7.20	Simulation investigating disturbance rejection for a tubular reactor with a segmented cooling jacket. The reactant feed to the reactor is disturbed while the setpoints are unadjusted.	100
7.21	Input (MV) and output (CV) changes for a tubular reactor with a setpoint change in the reactor outlet temperature. The conversion of monomer is not controlled.	104
7.22	Input (MV) and output (CV) changes for a tubular reactor with a setpoint change in the reactor outlet temperature. The tuning parameters are altered to demonstrate a smoother control trajectory than the case in Fig. 7.21.	104
7.23	Changes in controller action (MVs) and outputs (CVs) for a tubular reactor with a setpoint change in the reactor outlet temperature. The setpoint for the conversion of monomer through the reactor is not altered.	105
7.24	Changes in controller action (MVs) and outputs (CVs) for a tubular reactor with a setpoint change in the conversion of monomer. The setpoint for the reactor outlet temperature remains unchanged.	107
7.25	A simulated setpoint change for the reactor outlet temperature, using an MCV type model for the controller calculations.	108
7.26	A simulated setpoint change for the conversion of monomer through the reactor, using an MCV type model for the controller calculations.	109
7.27	Plots showing the idea for how the PSDs for two separate reactors can be combined to yield a wider distribution, if that is desired.	112
7.28	A simple demonstration showing the sensitivity of the polydispersity index to changes in the amount of chain transfer agent fed to the reactor.	113
C.1	Conceptual illustration of a semi-batch tank reactor with stirrer and continuous feeding.	VII
D.1	Coordinate system and corresponding equations for the transformation between cartesian coordinates and cylindrical coordinates.	IX
D.2	A cylindrical shell for the development of a shell balance	XI
E.1	A conceptual sketch of a one-dimensional, unsteady-state heat conduction problem	XIV
E.2	Resulting temperature profiles from solving the heat conduction problem.	XVI
G.1	A sketch of a simplified buffer tank.	XXIX
G.2	The GUI of the CENIT MMI, specifically for modifying the system variables.	XXXI
G.3	Changes in outputs (CVs) for a simulated setpoint change.	XXXII

List of Tables

3.1	Typical additional simplifications to the species mass balance for a continuous tubular reactor.	31
3.2	Main types of boundary conditions for PDEs.	33
3.3	Comparison between the different process models considered in the thesis.	48
4.1	Main components of the Cybernetica CENIT application.	56
5.1	Estimator action to noise in states, parameters and measurements.	62
7.1	Parameter modification for off-line parameter estimation for a tubular reactor using an MCV type model. The selection of parameters are chosen to affect the steady state behavior of the model.	78
7.2	Setpoint violation weights for 2x2-type control of a tubular reactor with a segmented cooling jacket.	99
7.3	Qualitative sensitivity matrix for the 2x2-type control system.	103

List of Code snippets

A.1	MATLAB code for radical species modeling.	I
E.1	MATLAB code for an unsteady-state heat transfer problem.	XV
E.2	Differential equations for the unsteady-state heat transfer problem.	XV
G.1	Modelica code for a simple buffer tank model.	XXIX

List of Acronyms and Abbreviations

BC	Boundary conditions
CSTR	Continuously Stirred Tank Reactor
CTA	Chain Transfer Agent
CV(s)	Controlled variable(s), system/controller output(s)
DAE	Differential Algebraic Equations
DV	Disturbance variable(s), system/controller disturbance(s)
EKF	Extended Kalman Filter
GUI	Graphical User Interface
HEKF	Hybrid Extended Kalman Filter
HSE	Health, safety and environment
IBVP	Initial boundary value problem
IEKF	Iterated Extended Kalman Filter
KF	Kalman Filter
LKF	Linearized Kalman Filter
LT	Laplace Transformation
LTI	Linear Time Independent (controller)
MCV	Moving Control Volume(s)
MHE	Moving Horizon Estimator
MIMO	Multiple inputs, multiple outputs
MMI	Man-machine interaction/interface
MPC	Model-based Predictive Control
MV(s)	Manipulated variable(s), system/controller input(s)
MWD	Molecular weight distribution
NMOL	(The) Numerical method of lines
NMPC	Nonlinear Model-based Predictive Control
ODE(s)	Ordinary Differential Equation(s)
OPC	Object linking and embedding for Process Control
PDE(s)	Partial Differential Equation(s)
PDI	Polydispersity Index
PFR	Plug Flow Reactor
PSD	Particle size distribution
PVC	Poly Vinyl Chloride
QP	Quadratic Programming
(D)RTO	(Dynamic) Real-Time Optimization
SBR	Styrene Butadiene Rubber
SQP	Sequential Quadratic Programming
TCP/IP	Transmission Control Protocol / Internet Protocol
UKF	Unscented Kalman Filter
VCM	Vinyl Chloride Monomer

List of Symbols

Roman letters:

A	System matrix for linear continuous systems, relating states with state changes
A	Matrix, equation system for full population balance for radical species modeling
A	Surface area for a tubular reactor
A	Surface area of an arbitrary body
a_p	Surface area of a (monodisperse) polymer particle
a_r	Surface area ratio between polymer particles and emulsifier molecules
a_s	Surface area of an emulsifier molecule
B	System matrix for linear continuous systems, relating inputs with state changes
C	System matrix for linear continuous systems, relating states with measurements
C	Termination-term for modeling free-radical species
C_p	Heat capacity
$c_{p,i}$	Specific heat capacity of component i
c_i	Concentration of component i
D	Dead polymer chain (polymer chain with a deactivated endgroup)
D_i	Effective mass diffusivity of species i
E	Energy content of a system
$E(t)$	Age distribution for residence time distribution considerations
$E[\dots]$	Statistical expectation value of a variable
$E_{A,i}$	Activation energy for species i
$F(t)$	Cumulative age distribution, RTD
\vec{F}	General vector function, (F_r, F_θ, F_z)
F_k	System matrix for linear discrete systems, relating states with state changes
F_i	Mass flow of component i
$F(t)$	Cumulative age distribution (integral of $E(t)$), for RTD considerations
f	General, arbitrary multi-variable function
f_I	Efficiency factor of free-radical initiation
G_k	System matrix for linear discrete systems, relating inputs with state changes
g	Gravitational constant
g	General, arbitrary, multi-variable function
h	Specific enthalpy
h	General, arbitrary multi-variable function
h	Height of a liquid fluid level
h	Step length (used in various numerical methods)
H	Enthalpy for a system
H_k	System matrix for linear discrete systems, relating states with measurements
J_k	Cost function for optimal estimation at time t_k
k	Desorption-term for modeling free-radical species
k_I	Rate constant for free-radical initiation
k_{ij}	Propagation rate constant for adding monomer type j to endgroup of type i

k_{ijk}	Propagation rate constant for adding monomer type k to endgroup of type j having a penultimate unit of type i
$k_{f,ij}$	Rate constant for chain transfer between a chain having an endgroup of type i and monomer type j
$k_{f,CTA,i}$	Rate constant for chain transfer from chain endgroup type i to CTA
k_{tc}	Rate constant for polymer chain termination by chain recombination
k_{td}	Rate constant for polymer chain termination by chain disproportionation
k_T	Combinated rate constant for polymer chain termination
$k_{reac,0}$	Initial rate constant for reaction $reac$
$k_{reac,adj}$	Adjustment factor for the rate constant of reaction $reac$
k_A	Heat transfer coefficient between reactor and cooling jacket
k_{ASR}	Heat transfer coefficient between surrounding environment and reactor
k_{ASJ}	Heat transfer coefficient between surrounding environment and cooling jacket
$k_{i,p1 \rightarrow p2}$	Mass transfer coefficient for component i between phases $p1$ and $p2$
k_i^p	Partition coefficient for monomer i in phase p (compared to water phase)
K_k	Estimator gain at time t_k
M_i	Monomer, type i
L	Length of the tubular reactor
M_{Mi}	Molar mass, monomer type i
M_n	Molar mass, number average
M_w	Molar mass, weight average
m	Total mass
m_i	Amount of component i , mass basis
\hat{m}_i	Flow rate of component i , mass basis
$\hat{m}_{i,p1 \rightarrow p2}$	Mass transfer of component i from phase $p1$ to phase $p2$
n_i	Molar amount of component i
n	Vector of respective molar amounts for particles carrying radicals
n	Number of discrete points in a spatially discrete tanks-in-series type reactor
\bar{n}	Average number of radicals per particle
\vec{n}	Normal vector to an arbitrary surface
n_r	Number of independent reactions in a chemical reactor
N	Prediction horizon for optimal control
N_T	Total number of (polymer) particles in an emulsion
$\mathcal{O}(\dots)$	Unspecified amount of quantity in the order of (\dots)
P	Vector of covariances for a corresponding state vector
P_k^-	Vector of <i>a priori</i> covariances for the state estimate at time t_k
P_k^+	Vector of <i>a posteriori</i> covariances for the state estimate at time t_k
P_i	Living polymer endgroup, type i
p	Pressure of a system
Q_k	Vector of covariances for process noise at time t_k
Q_c	Vector of covariances for process noise, continuous formulation
\tilde{Q}	Linearized vector of covariances for process noise
Q	Heat (energy) transfer
q	External heat transfer (used in balance equations)
q	Extra supporting variable for modelig free-radical species

r	Radial coordinate, cylindrical coordinates
r_{ij}	Relative reactivity between homo-propagation and cross-propagation for endgroup of type i in a copolymerization system
r_i	Reaction rate of component i
r_1	Linear weights for soft constraints (ϵ) in MPC outputs
r_2	Quadratic weights for soft constraints (ϵ) in MPC outputs
$r_{k,i}$	Molar rate of reaction for reaction no. k for species i .
R	Universal gas constant
R_k	Vector of covariances for measurement noise at time t_k
R_c	Vector of covariances for measurement noise, continuous formulation
\tilde{R}	Linearized vector of covariances for measurement noise
$R(t)$	Molar amount of reactant in a reactor at time t
s	Used to denote length of a polymer chain
T	Temperature
T	Small time interval (Δt) used to derive the continuous linear Kalman filter
t	Time
t^*	Time in a system of transformed variables
$Tr(A)$	Trace of a matrix A (linear algebra operation)
t_k	Time at point k
u	Vector of inputs for a system
u	Bulk fluid movement velocity
u_z	Bulk fluid movement velocity, axial direction
U	Internal energy of a system
U	Heat transfer coefficient
$\mathcal{U}(t - t_0)$	Heaviside Step function at time t_0
v	Measurement noise
v_k	Measurement noise at time t_k
\vec{v}	Vector field of fluid flow velocities, (v_r, v_θ, v_z)
v_r	Radial velocity
v_z	Axial velocity
v_d	Velocity contribution due to diffusive effects
V	Volume
\dot{V}	Volumetric flow rate
V_p	Volume of (polymer) particle phase
V_d	Volume of (monomer) droplet phase
V_w	Volume of (aqueous) water phase
\mathcal{V}	Volume of an arbitrary body
w	Process noise
w_k	Process noise at time t_k
w_i	Mass fraction of component i
W_s	Shaft work applied to a system
X	Degree of conversion
x	Relative monomer composition in system, used in the copolymer equation
x	Vector of states
x_k	Vector of states at time instant t_k

\dot{x}_k	Time derivative of (the vector of) states at time t_k
\hat{x}_k^-	Vector of <i>a priori</i> state estimates at time t_k
\hat{x}_k^+	Vector of <i>a posteriori</i> state estimates at time t_k
y	General, arbitrary, multi-variable function
y	Vector of measurements
y_m	Vector of measurements
y_k	Vector of measurements at time t_k
$y_{p,k}$	Vector of predicted outputs (measurements) at time t_k
y	Relative monomer representation in polymer, used in the copolymer equation
Y_i	Molar fraction of monomer type i in polymer
z	Axial coordinate, cylindrical coordinates
z^*	Spatial (axial) coordinate in transformed cylindrical coordinates
z	Model outputs (controlled variables)
z_k	Model outputs at time t_k

Greek letters:

α	Pre-factor, used for micellar nucleation
$\delta(t - t_0)$	Dirac Delta impulse function with "spike" at time t_0
Δt_k	Discrete time interval between time t_{k-1} and t_k
Δu_k	Change in input vector (for the MPC) between time t_{k-1} and t_k
Δz	Discrete step in space, axial direction
ϵ	State estimation error
ϵ	Soft constraint (slack) in MPC outputs
η	Subset of the vector of parameters (θ) which is unknown
θ	Vector of parameters for a system
θ	Azimuthal coordinate, cylindrical coordinates
μ_M^i	Consumption of monomer type i in the copolymerization process
μ_i^j	i 'th order moment of component j
ρ	Density
ρ_i	Density of component i
$\tilde{\rho}_I$	Generation of polymer particles due to micellar nucleation
σ	Generation term for modeling free-radical species
$\tilde{\sigma}$	General generation term used in balance equations
σ	Surface tension for growing polymer particles
τ	Used as a replacement for time (t) in integrals where time appears both in the integrand and the integration boundaries
τ	Mean residence time (used for various reactor types)
ϕ	Extra supporting variable for modeling free-radical species
ψ	A generalized, arbitrary quantity
ω	Used as a replacement for process noise (w) in integrals where the process noise is "transformed" from a continuous to a discrete formulation
ξ	General, arbitrary spatial coordinate

Introduction

1.1 Background

For the author, this work represents the finalization of the study program leading to the degree of M.Sc. in Chemical Engineering at the Norwegian University of Science and Technology. It also represents the conclusion of a one year long student internship in Cybernetica AS. Cybernetica AS is a small Trondheim-based company providing model-based niche software solutions to the process industry. The solutions can range from tailor-made process simulators to on-line optimization tools for specific processes, but the flagship of the business is the software for Nonlinear Model-based Predictive Control (NMPC) known as Cybernetica CENIT. MPC is a well-known technology which is being used in numerous plants around the world, and many companies develop solutions for MPC, but for nonlinear first principles state space models, Cybernetica is found to be one among very few companies that provide a complete well-functioning tool. The solutions of Cybernetica are applicable to the fields of chemical industry (e.g. polymer industry), petroleum technology (e.g. well technology and recovery of petroleum products) and metallurgic industry (e.g. electrolytic metal production processes), among others. Model-based approaches can, in other words, be deployed to control almost any kind of industrial process. In addition to this, as will be elaborated, the NMPC approach is generally considered to perform particularly well for complex processes with intricate couplings and significant nonlinearities, as compared to other methods for advanced process control. This quality makes the NMPC suited for treating processes like the tubular smart-scale reactor which is considered in this thesis.

The prelude to this thesis is a specialization project written during the fall of 2013, which in turn represents an extension to a summer internship during the summer of 2013 [1]. The topic of the student internship has been related to the European Union collaboration research project COOPOL¹ (Control and real time optimisation of intensive polymerisation processes) of the Seventh Framework Programme. The purpose of the project is to intensify copolymerization processes by addressing complex issues on several levels. The intensification is achieved through interdisciplinary work, e.g. by combining chemistry on the molecular level with sensor technology and methods for model-based control. Ultimately, the intensification must be optimized to yield as profitable and safe operation as possible for the process in mind.

To summarize the preceding work, the main scope has been related to modeling and optimization of a semi-batch reactor for free-radical emulsion copolymerization. First principles modeling and off-line parameter estimation using experimental data were among the key activities. Throughout this thesis, the established ideas for the semi-batch reactor are extended, and the purpose is to explore the behavior of a continuous tubular reactor for emulsion copolymerization processes which are similar to the system from the semi-batch studies. In the COOPOL project, this initiative is referred to as the smart-scale reactor setup, which is a suggested reactor structure with respect to process intensification of the already established semi-batch reactor setup. The step from batch reactor setups to continuous tubular reactor setups is expected to yield increased complexity in the process models, since the tubular reactor represents a spa-

¹Read more about the COOPOL project here: <http://www.coopol.eu>

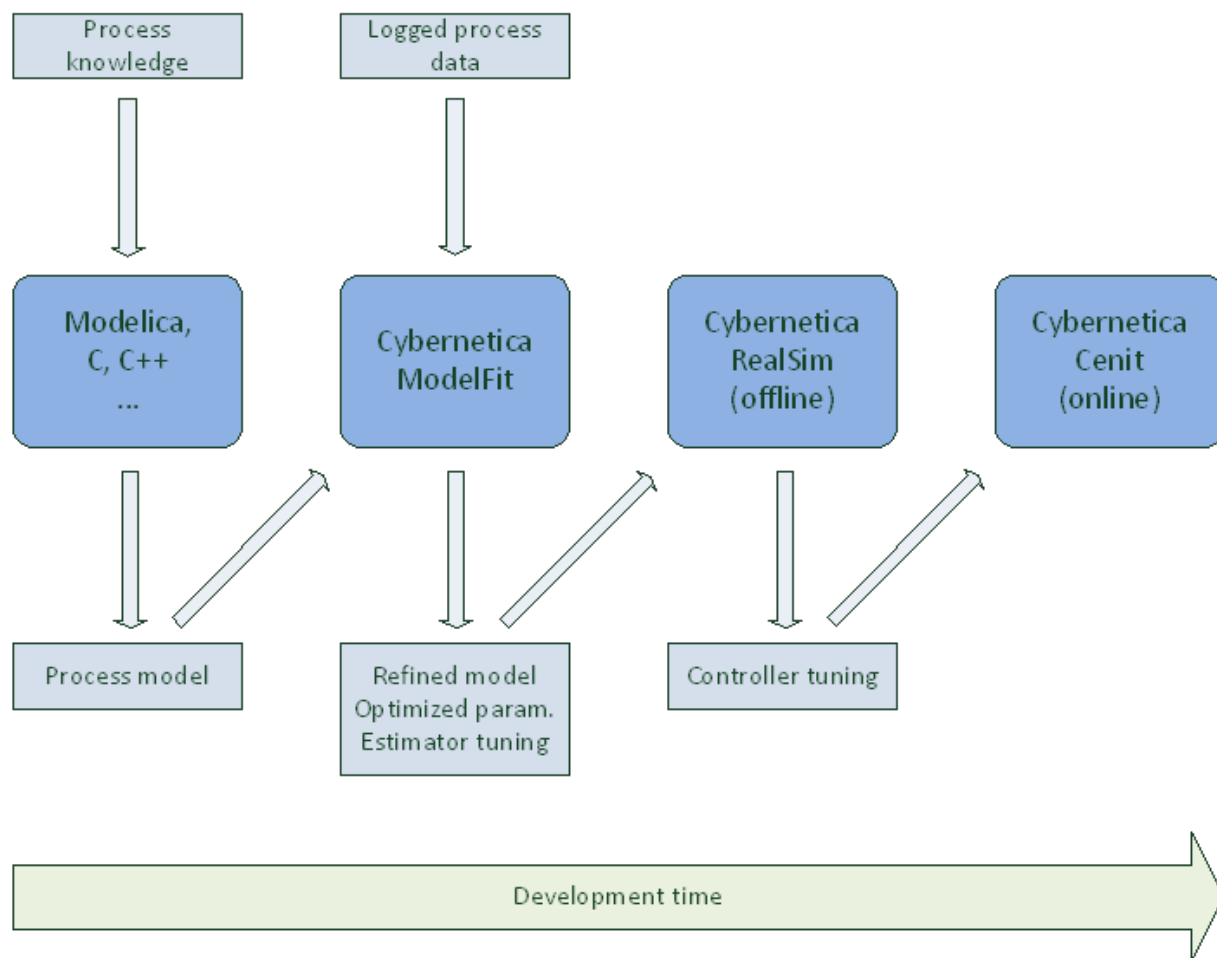


Figure 1.1: Typical work process in MPC development using mechanistic models.

tially distributed system. In on-line applications, this is expected to cause a larger demand for computational effort. As a consequence, it is desirable to investigate different strategies to construct efficient models for tubular copolymerization reactors, and to test the models for on-line optimization and control.

Fig. 1.1 shows the steps along the way in a typical development process of an MPC implementation for a process². The respective in-house developed software tools of Cybernetica, which are mentioned in the figure, are in turn introduced in Sec. 4, and deployed for all the simulations which are conducted in Sec. 7. For the specialization project which precedes this thesis [1], the scope was aimed at the left hand side of Fig. 1.1, i.e. at modeling and off-line optimization of the specific process, as mentioned. In the field of process control, the use of mechanistic first principles models represents an alternative to the use of empirical step change models from on-line considerations to design and tune the controllers. The development of first principles models for complicated processes is typically quite time consuming, but this is often regarded as time well

²This figure is printed with the permission of J. G. Dyrset, Cybernetica AS.

spent when the performance of the models in on-line applications become evident. Because of this, the left hand side of Fig. 1.1 has been prominent for this thesis as well, when a continuous tubular reactor has been considered. The development of the tubular reactor models, which are described in Sec. 3, has been a weighty part of the thesis, and improvements to the models have been added continually along the duration of the thesis work. The typical development process which is depicted in Fig. 1.1 is, in other words, not chronologically restricting in the sense that each stage necessarily must be completed before the succeeding stage can be considered. Despite the fact that process modeling has been one of the main activities of the work, it is emphasized that an underlying motivation of the thesis is to investigate the model behavior in on-line use with respect to optimization and control (i.e. the right hand side of Fig. 1.1).

1.2 Topic: Industrial polymer chemistry

The overall topic of the thesis, as for the entire student internship for which this thesis constitutes the main part, is related to industrial polymer chemistry. Although a severe portion of the focus in the thesis is aimed at modeling of distributed systems in general, which is applicable to numerous fields beyond polymer chemistry, the purpose is to explore how a specific copolymerization system can be modeled, optimized and controlled in an on-line application.

Polymer science and polymer industry are of huge importance for everyday life, in providing the basis for a vast variety of products. Examples can range from household products found in the kitchen and furniture, to plastics, rubber for car tires, health care products, agricultural applications, etc. The applications are numerous. Many industrial polymer processes also provide industrial chemicals for use in further processing. To exemplify the importance of polymer industry, the European consumption of plastics in 2011 was approximately 47 million tonnes, which is a 1.1% increase from 2010. A significant portion of this increase is associated with the automotive industry, which experienced an increase of almost 10% in the consumption of plastics [2]. The consumption is not expected to decrease significantly in the nearest future, and developing even better ways to produce the polymer products is both interesting and, above all, important.

Emulsion polymerization processes in particular, which are the focus of this work, are used for several different applications worldwide. The most significant ones are probably the industrial production processes leading to adhesives (glues) and various coatings. The preparation and use of adhesives is ancient, perhaps nearly as old as civilization itself. Speaking about modern industrial production, however, many of the industrial technologies were first established close to a hundred years ago, and they are still subject to research and improvements [3]. New ways to achieve desired reactor cooling, new monomers, new combination patterns of monomers in copolymerization, alternative reactor structures, etc. are just a few examples of factors that bring great variety and new possibilities to the science of emulsion (co)polymerization, as well as polymerization in general. The tubular smart-scale reactor, which is considered in this thesis, is merely one example of such a technology that can take the polymer industry one step further. To exemplify the importance of these types of processes, the world consumption of adhesives was approximately 12 million tonnes in 2010, having an average annual growth of 3.5%. The annual growth is predicted to be as high as 5% by 2015 [4]. In line with the increasing demands, higher production rates are sought by finding new solutions and by optimizing the operation, both with respect to production rates, product quality and safety.

In the pursuit of the best possible performance of chemical reactors used in polymerization

processes, with the ultimate goal being the most profitable operation under safe conditions, there are several challenges. One example of such a challenge is the fact that many polymerization systems are highly sensitive to temperature changes, due to the chemical reactions of the system being both exothermic and kinetically favorable at higher temperatures. This can lead to so-called run-away situations, and raises the need for proper strategies for temperature control for the reactor system. In a run-away situation the heat developed in the reactor system acts directly to speed up the chemical reaction rate. This may proceed in an uncontrollable manner, quickly establishing a vicious circle leading to plant failure if it is not counteracted. Such a scenario would endanger the employees of the plant, the ability to keep up the production as intended and possibly also the nearby environment. Because of this, temperature control is a crucial issue for most polymer reactors, and the use of sophisticated controllers can, with respect to this concern, contribute to yield safe operation of the plant. By using model-based controllers with adequate process models, the critical points in the operation of the reactor can be predicted in advance and be actively treated according to the desired objectives of the plant.

Another example of a typical challenge in polymerization processes is the time duration associated with making each batch of polymer product in a (semi-)batch reactor. For many industrial applications, the batch time may amount to several hours. Under given constraints for product quality, considerations with respect to health, safety and environment (HSE) etc., the batch time could be subject to optimization, with the goal being a larger production capacity in total as a consequence of shorter batch times. The reduction of batch time is a complex issue demanding extensive knowledge of all the aspects affecting the behavior of the reactor system. A suggestion to overcome the time demand in batch reactor applications is to explore the use of continuous reactors for polymer applications, for instance the use of tubular plug flow reactors (PFR). This is a less mature technology than batch reactor applications when it comes to polymer industry, and presents with challenges of its own. Flow patterns, mixing issues and appropriate cooling systems are among the challenges associated with continuous flow reactors, and they may be hard to incorporate effectively in a process model with strict requirements for numerical efficiency. The exploration of continuous tubular reactors for emulsion copolymerization processes is at the heart of this thesis, and several among the challenges which are mentioned will be addressed. It is emphasized that the semi-batch reactor setup is the most abundant setup in the industry today because it is more well-established than the tubular smart-scale reactor, and can provide a more profitable operation over time. Despite this, the tubular smart-scale reactor setup is believed to have a certain potential, and it is expected to be more abundant in industry in the future.

Common for both (semi-)batch reactors and continuous flow reactors, e.g. tubular smart-scale reactors, for emulsion copolymerization is that the evaluation and control of these systems require process models of high-end quality and extensive process understanding for the respective systems. In addition, these systems are often complex and time consuming to comprehend, since they are theoretically advanced. On the other hand, having an advanced and efficient controller is crucial for the success of such a reactor system. A process model of high quality is, in other words, the most important building block of a model-based controller of high quality. The NMPC approach is, as already mentioned, considered to perform well, generally speaking, for complex processes in comparison with other methods for advanced process control. Strong coupling of dynamic variables, prominent nonlinear effects and various constraints are among the issues which are considered to be resolved effectively by an NMPC approach, given an adequate process

model.

Some fundamental concepts of emulsion copolymerization are discussed in Sec. 2, with the purpose of properly establishing the system of interest from a chemical point of view. Although the scope of the reactor modeling part is fairly general, it is emphasized that the underlying topic of the thesis is a specific system for emulsion copolymerization of a system which includes two different monomers. It is of essence to encapture as much as possible of the details regarding the various phenomena of the system, from a chemical point of view, yet still be able to provide a model which is efficient to deploy in on-line use.

1.3 Challenge: Optimization and control of distributed systems

The extension in going from (semi-)batch type reactors, which were considered in the work preceding this thesis [1], to continuous tubular smart-scale reactors is, generally speaking, expected to lead to more complexity in the process models. A tubular smart-scale reactor represents a spatially distributed system, in contrast to the well-mixed batch type reactors without spatial gradients (by assumption). In practice, this means going from ordinary differential equations to partial differential equations for the model equations. This calls for different solution strategies, and the demand for computational effort is expected to increase. In brief, a common approach is to represent each partial differential equation by a system of ordinary differential equations, since dynamic equations in time are the most appropriate for on-line use of the model. Depending on the strategy for achieving this, i.e. the discretization of the problem, the system of model equations may become unsuitably large. In the context of on-line optimization and control, this raises the need for model simplifications and model reductions to achieve the required numerical efficiency of the model. At the same time it is important to realize that the model simplifications should not severely compromise the agreement between the process model and the real-life behavior of the system, i.e. the experimental behavior of the specific tubular reactor. Some concepts for the modeling of tubular reactors are discussed in Sec. 3, where various approaches are introduced to achieve adequate process models.

Experience shows that the main challenge to achieve efficient control does not necessarily lie in controller tuning, but rather in obtaining a proper process model. The importance of controller tuning is definitely not underestimated, and controller tuning for MPC is indeed investigated for some of the cases in the thesis, but there is a consensus that proper process models constitute the foundation of a well-functioning control system. One could claim that the strict requirement for a model of high quality is a drawback of the MPC approach, but the performance and flexibility of a well-functioning MPC implementation helps to outweigh this. [32]

The basis for the modeling work is one traditional approach, using spatial discretization by finite differences of lower order, and one alternative approach using transformation of coordinates, in order to solve the partial differential equations constituting the reactor model. The latter approach culminates in a model with moving control volumes which propagate through the reactor. There are several other approaches which could be deployed to model a distributed system, i.e. solve partial differential equations, in addition to the ones which are considered in this thesis. Higher order finite differences methods, finite volume methods, spectral methods and collocation methods are some of the other known methods which could be deployed alternatively. In establishing the models, experimental data for residence time distribution will be deployed to analyze how well the respective modeling approaches encapture the mixing conditions of the tubular reactor. This will provide valuable measures to evaluate the general quality of the models.

In the previous subsection, some concerns with respect to the safety of polymerization reactors were mentioned. For polymerization systems, the importance of safety is emphasized, but the focus on HSE remains prominent in all chemical industry on account of the tremendous consequences of plant failure. In this context, proper strategies for process control are crucial, and to develop well-formulated models is at the heart of this train of thought. Industrial process safety is indeed a matter to be addressed by system theory, as Leveson & Stephanopoulos state in their article of 2013 [5]. Among the conclusions of the article is the statement that while no process model can be complete in an absolute sense, the ability of the model to describe state changes and different operation modes in a correct way is closely correlated with safety. As a consequence, an incomplete process model may suggest ineffective, or even detrimental, action in an event where the safety of the plant is compromised. In agreement with this, spending considerable amounts of time on model development is justified. Developing efficient controllers³ is, in other words, an activity yielding both better (with respect to production rate and economics) and safer operation. In the specific context of controlling polymer reactors, situations where the model fails to predict dramatic temperature changes, and hence inhibits the effective control of temperature, are among the most dangerous. As a consequence, temperature control will be the most abundant type among the control studies which are investigated in this thesis.

In the context of a spatially distributed system, the cooling mechanism for a tubular smart-scale reactor is in itself of interest to consider. This type of reactor has a wide range of different cooling strategies which are possible to deploy, and they have different effects on the cooling performance. For the lab-scale experimental reactor, on which the work of this thesis is based, an oil bath is used, where the entire reactor is submerged in a homogeneous cooling fluid. This is the cooling setup which is investigated the most throughout this thesis. In addition, a typical setup for a tubular reactor is a cooling jacket, which encapsulates the reactor tube, where the cooling fluid flows countercurrent or concurrent to the reactor contents. As an extension, one could imagine a cooling jacket which is divided into several segments, for which the respective temperatures can be manipulated independently, covering the tubular reactor. This strategy is investigated in Sec. 7.7. It is emphasized that the ability of the process model to encapture the characteristics of the cooling jacket setup is crucial to achieve the desired reactor cooling, i.e. the desired temperature control.

³In the context of model-based predictive control, this is synonymous to developing adequate process models.

Section 2

Fundamentals of free-radical emulsion copolymerization

This section aims to give an introduction to some of the basic theoretical concepts of copolymerization, specifically for emulsion systems, i.e. the typical chemical behavior of such systems. The motivation for this is to establish a proper starting point for discussing the modeling of emulsion copolymerization reactors, which is a crucial achievement along the way to establish an on-line control system for such a reactor. The chemical features of the copolymerization reaction system, including possible reaction patterns, are discussed throughout this section. Sec. 2.1 introduces the various reactions, including a brief look at the respective mechanisms, that are expected to occur in a typical copolymerization system containing two monomer species, and the corresponding kinetic expression. A discussion on the phase behavior of copolymerization systems (Sec. 2.2) and the modeling of free-radical species (2.3) are both included, and some quality parameters for polymer products are introduced briefly at the end of the section (Sec. 2.4). For more details on emulsion (co-)polymerization, the reader is referred to more extensive publications on this subject. The theory of this section is, among others, based on the work of Painter & Coleman. [3] [6] [7]

2.1 Kinetics of free-radical copolymerization

The course of a free-radical polymerization reaction process can be separated/classified into several different stages. These are most commonly referred to as initiation, propagation and termination. In addition, so-called chain transfer processes will occur, in which activated polymer chains react with both free monomer units and chain transfer agent (CTA) in the reactor mixture. In free-radical polymerization, the growing polymer chains are often described as "living" (active) or "dead" (inactive/terminated) depending on whether they carry radicals on their end-groups, and the chemical behavior of the system is largely dependent on this. An analogy to the different stages (the "life") of the free-radical polymerization process, which helps to reinforce the active/terminated classification of polymer chains, is presented in Fig. 2.1. Here, the behavior of a rapidly growing biological system of cells in a closed-off environment with a limited amount of available nutrition is introduced, as seen in Fig. 2.1a. Initially, the cell population experiences excess of nutrition, and will multiply significantly in the so-called birth-dominated region until the births of cells balance the deaths of cells in the stationary region. When the source of nutrition is nearly depleted, the amount of deaths will outweigh the amount of births until the cell culture is extinct. The same behavior is found for the population of active polymer chains in a free-radical polymerization system. In Fig. 2.1b, the total polymer concentration, i.e. concentration of active and terminated chains combined, is shown. The concentration of activated chains is not illustrated, but it would be (qualitatively) similar to the living cell concentration in Fig. 2.1a. In this analogy, the monomers of the polymer system, which add to the growing chains, and hence act to increase the polymer concentration, are analogous to the nutrition for the biological cell culture. Once the nutrition (monomer) is drawing near to depletion, the cells

(active polymer chains) are more likely to die (deactivate) than to keep on growing. As an ending remark to this analogy, it is emphasized that the deactivation of polymer is not as fatal as the analogous term (death) suggests, given that they have already grown to a satisfying size. In certain cases, it may even be desirable to only let the polymer chains grow to a certain length, all depending on the specifications for the polymer product.

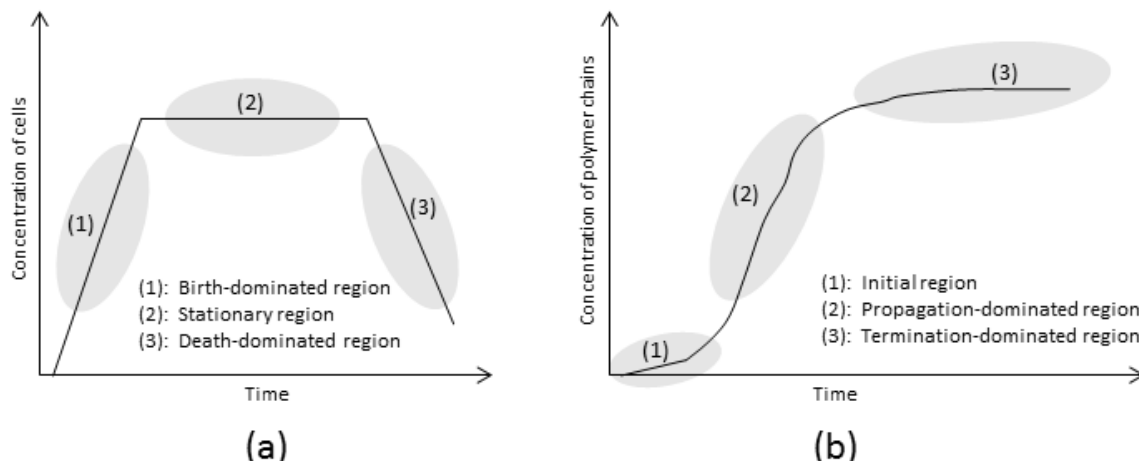


Figure 2.1: Qualitative analogy for free-radical polymerization, using the lifetime of a typical biological cell culture in a closed-off environment (a) to illustrate the "life" of a typical system of growing free-radical polymer chains (b).

2.1.1 Initiation

Initiation of radical species is a crucial step in a free-radical polymerization reaction system. Although the propagation reactions are the reactions that actually create and elongate the polymer chains, these reactions would never proceed in the first place without the activation/initiation of the radical species in the system. The modeling of the radical distribution in the system is itself challenging, and this is discussed briefly later (Sec. 2.3). Various types of peroxide compounds are commonly used in free-radical initiation processes, as peroxides easily decompose to form free radicals under the right conditions. In such situations, the peroxide compound is referred to as the *initiator* of the system, and it is important to recognize that the radicals may reside in either of the phases of the system. Fig. 2.2 is an illustration of the initiator activation for an arbitrary peroxide compound. Once the peroxide is activated it can start to "attack" the double bond(s)

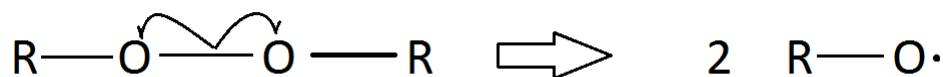


Figure 2.2: Mechanistic idea for initiator activation using an arbitrary peroxide compound in free-radical polymerization.

of the monomer units. This mechanism¹ is illustrated in Fig. 2.3. Notice that after the attack,

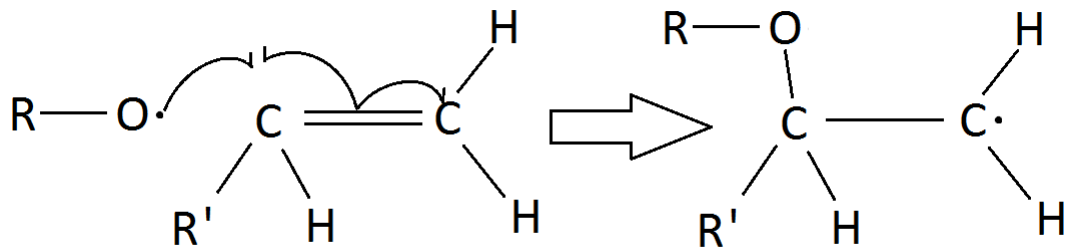


Figure 2.3: Mechanistic idea for initiator attack on a monomer double bond in free-radical polymerization.

the free radical (indicated by a small dot in the figures), which originally resided at the oxygen atom at the end of the decomposed peroxide, is now actually carried by one of the carbon atoms originally situated adjacent to the double bond of the monomer, which enables this carbon atom to attack a new double bond. In other words, the reaction proceeds towards propagation after this initial activation, and the polymer chain starts to grow in a "snowball"-manner². A polymer chain is, as mentioned, generally referred to as either living or dead, depending on whether the end group of the chain is activated by a radical species or not. In this sense, termination reactions, which are discussed later, tend to deactivate ("kill") living chains by neutralizing/removing the free-radical state of the growing polymer chain.

The rate of change for the amount of initiator compound is usually expressed as indicated in Eq. 2.1, where both initiator decomposition and external mass transfer of initiator are considered. In this expression, n_I is the amount of initiator in the reaction vessel and \hat{n}_I is the net external transport of initiator to the system, while f_I is the efficiency factor for the activation process. The kinetic rate constant (k_I) is calculated using a standard temperature dependent Arrhenius expression starting from an initial value for the rate constant ($k_{I,0}$), as shown in Eq. 2.2. Here, $E_{A,I}$ is the activation energy for the initiator decomposition, while R and T denote the universal gas constant and the system temperature, respectively. For reasons related to parameter estimation, which will be discussed later, a fictional adjustment factor ($k_{I,adj}$) is also included in this expression to ease parameter modifications in future treatment of the model.

$$\frac{dn_I}{dt} = -f_I k_I n_I + \hat{n}_I \quad (2.1)$$

$$k_I = \frac{k_{I,0} \exp\left(-\frac{E_{A,I}}{RT}\right)}{k_{I,adj}} \quad (2.2)$$

2.1.2 Propagation

The propagation reactions represent the connecting of monomer units to growing (living) polymer chains. This is the main part of the copolymerization reaction, and these are the dominant reactions in the propagation-dominated phase (2), as can be seen in Fig. 2.1b. Generally speaking,

¹In figures showing reaction mechanisms, the thin, curved arrows indicate the migration of electrons or radicals.

²Much like a snowball grows rapidly once it starts to roll downhill, a polymer chain grows rapidly once initiated/activated by a radical species. This is crucial to the polymerization process, yet important to control.

the system can consist of any number of various monomer types, which will affect the complexity of the system, and also the properties of the polymer product. The most usual cases are, however, one-monomer polymerization (homo-polymerization³) and copolymerization with two monomers⁴. The latter is the case for the process studied in this work, and hence this will govern the focus of this theoretical introduction. In the COOPOL project, there is a desire to account for as much as four separate monomers in a free-radical emulsion copolymerization system, but this is not considered in this thesis.

The notation used below, which is also used throughout the entire text, represents the various species of the chemical system as follows:

- M_i represents a monomer unit of type i . For a two-monomer copolymerization case, in other words, i can be either 1 or 2.
- $M_i(s)$ represents a polymer chain body consisting of a combination of s monomer units, which could be any combination of type 1 and type 2.
- P_i represents a living (active) endgroup of a polymer chain, which originally was a monomer unit of type i . This active endgroup is a key component in the free-radical reactions involved in the chemical reaction system. In the cell culture analogy introduced in Fig. 2.1, the cells are analogous to these living polymer chains with active endgroups.
- D_i represents a dead (deactivated) endgroup of a polymer chain, which originally was a monomer unit of type i .
- k_{ii} , k_{iji} , etc. are reaction rate constants for the various reactions involved in the propagation reactions in the polymerization process. For instance, k_{12} is the kinetic rate constant for the reaction where a monomer type 2 adds to a growing polymer chain having a living endgroup of type 1, k_{212} is the rate constant for the reaction where monomer type 2 adds to a growing polymer chain having an active endgroup of type 1, with a unit of type 2 neighboring the endgroup, and so on.

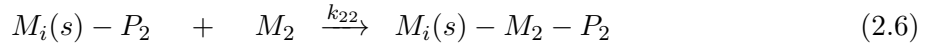
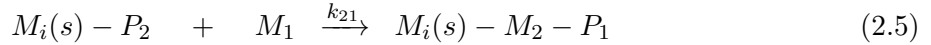
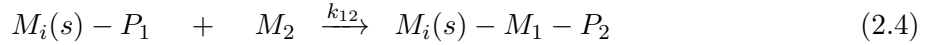
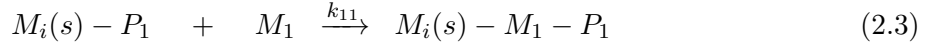
The monomers of the system can interact in various patterns, depending on the respective monomers reactive affinity for the other monomers in the system. This can be described by different modeling approaches, using varying degree of complexity. The Endgroup Kinetic Model, also known as The Terminal Model, for copolymerization involves four⁵ possible propagation reactions, as indicated in Eqs. 2.3 - 2.6. In this sense, the reactivity of the growing polymer

³A typical example of homo-polymerization: The production of polyvinyl chloride (PVC) uses vinyl chloride monomer (VCM), also known as chloroethene, to yield a very applicable plastic polymer. This polymer is mainly produced using chemical systems for free-radical suspension polymerization.

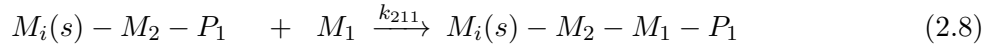
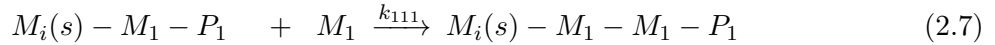
⁴A typical example of two-monomer copolymerization: The production of styrene-butadiene rubber (SBR) combines styrene monomer and butadiene monomer to yield a polymer product with desirable properties for rubber used in car tires, among other applications.

⁵This number is valid for a two-monomer system only, and will be higher in the event of introducing more monomer types, as this will allow for more possible combinations. Combinatorially speaking, the number of possible combinations is the amount of monomer types squared.

chain is determined solely by the characteristics of the terminal group.



A more complete, yet more complex, model for the reaction kinetics is known as The Penultimate Model. This approach also recognizes the effect of the penultimate unit in the growing polymer chain (that is, the unit neighboring the unit at the end of the chain). An example is shown in Eqs. 2.7 - 2.8.



In this case, $k_{111} \neq k_{211}$

Notice that both Eq. 2.7 and Eq. 2.8 in the penultimate model are accounted for as the same reaction in the terminal model, namely Eq. 2.3, although they are not identical. In other words, the penultimate model approach represents reality in a better way, but also introduces more complexity. In many situations, the terminal model is sufficient, and this model will hence be applied for the considerations in this thesis. The reaction rate constants are usually modeled in a way similar to the case for the initiator activation (Eq. 2.2), i.e. using an Arrhenius temperature dependency, as indicated in Eq. 2.9. For the propagation reactions, an adjustment factor ($k_{ij,adj}$) is included, like what was done for the initiation. Here, $E_{A,ij}$ is the activation energy for the propagation reaction where monomer type j adds to a growing chain with type 1 active endgroup.

$$k_{ij} = \frac{k_{ij,0} \exp\left(-\frac{E_{A,ij}}{RT}\right)}{k_{ij,adj}} \quad , \quad i, j \in [1, 2] \quad (2.9)$$

With the various possible combinations for chain propagation established, the composition of the reaction mixture as well as the composition of polymer chains are possible to model. This can provide valuable information about the state of the copolymer system, depending on what is already known for the system. For instance, this information can be used to achieve correct values for cross-propagation rate constants (i.e. k_{12} and k_{21}) from experimental data in a case where these are unknown. The algebraic combination of the various propagation reactions can yield the so-called copolymer equation, which is shown below (Eq. 2.13). The following derivation is based on three important assumptions.

1. The kinetics of each propagation step is first order with respect to the reactants, that is monomer and living/active chains, and the reactions are irreversible. These assumptions enable the establishment of simple kinetic expressions for the rate of change for each species.
2. The reaction kinetics assume a "pseudo steady-state" situation where the generation and consumption of each radical species balance each other. The mathematical interpretation of this is: $k_{12}[P_1][M_2] = k_{21}[P_2][M_1]$.

3. The reactor in mind is a batch reactor. For reactors having external mass transfer, the resulting expression will look slightly different. Considerations must be taken, for instance, if the equation is to be deployed for a semi-batch reactor system or a continuous tubular reactor.

Here, brackets indicate molar concentration. The change in the respective monomer concentrations can be expressed with respect to the proposed propagation reactions.

$$-\frac{d[M_1]}{dt} = k_{11}[P_1][M_1] + k_{21}[P_2][M_1] \quad (2.10)$$

$$-\frac{d[M_2]}{dt} = k_{12}[P_1][M_2] + k_{22}[P_2][M_2] \quad (2.11)$$

$$\implies \frac{d[M_1]}{d[M_2]} = \frac{k_{11}[P_1][M_1] + k_{21}[P_2][M_1]}{k_{12}[P_1][M_2] + k_{22}[P_2][M_2]} \quad (2.12)$$

Using the second assumption listed above, and introducing some extra variables for the purpose of overview, this expression can be rearranged in a comprehensible way for further applications. The introduced variables are the relative change in monomer concentration ($y = \frac{d[M_1]}{d[M_2]}$), the relative monomer concentration ($x = \frac{[M_1]}{[M_2]}$), the relative reactivity ratio between homo-propagation and cross-propagation for active endgroups of type 1 ($r_{12} = \frac{k_{11}}{k_{12}}$), and the relative reactivity ratio between homo-propagation and cross-propagation for active endgroups of type 2 ($r_{21} = \frac{k_{22}}{k_{21}}$). The final expression, after introducing the new variables, is given in Eq. 2.13, which is known as The Copolymer Equation⁶. In a case where the reaction mixture composition is well monitored, this expression can be used to determine the relative reactivity ratios, for instance using a Fineman-Ross plot. On the other hand, if the rate constants are well known, this equation can be used to adequately express the polymer composition as a function of the monomer blend composition.

$$y = \frac{1 + r_{12}x}{1 + \frac{r_{21}}{x}} \quad (2.13)$$

The values of the rate constant ratios (i.e. r_{12} and r_{21}) can also be used to qualitatively predict the polymer composition, seeing as they provide the ratios between the kinetic rate constants for homo-propagation and cross-propagation. High values for the relative ratios indicate higher affinity for homo-propagation than cross-propagation, and *vice versa* for low values. Some special cases for the relative reactivity ratios are represented below.

- $r_{12} \gg 1, r_{21} \gg 1$: In this case, the affinity for homo-propagation totally outweighs the affinity for cross-propagation. For an active endgroup of type 1, it is virtually no desire to let monomer of type 2 add to the chain, and *vice versa*. The result is expected to be two different types of homopolymers. For many copolymer applications, this is not desirable, as it is indeed the combination of monomers that gives rise to the wanted copolymer properties.

⁶The Copolymer Equation is also known as the Mayo-Lewis equation, attributed to Frank R. Mayo and Frederick M. Lewis.

- $r_{12} > 1$, $r_{21} > 1$: In this case, the active endgroups have an affinity for homo-propagation which is larger than the affinity for cross-propagation. This will promote copolymers with a block structure, with blocks of repeating units arising from the same type of monomer. The average length of the blocks will depend on the actual values for the relative ratios. This situation is illustrated in Fig. 2.4a.
- $r_{12} \approx 1$, $r_{21} \approx 1$: Here, the copolymer has the so-called "completely random" characteristic. This is intuitive, since this situation implies that an active endgroup has the same affinity for both monomer types in the mixture, regardless of whether the endgroup itself is type 1 or type 2. Hence, the general combination of monomers is hard to predict, and the specific monomer ordering becomes random. This situation is illustrated in Fig. 2.4b.
- $r_{12} \approx 0$, $r_{21} \approx 0$: In this case, the ratios are small. This indicates a low affinity to homo-propagation in comparison with cross-propagation. The result will be a so-called alternating copolymer, where every other unit is type 1 and every other unit is type 2.



Figure 2.4: Illustration of typical block copolymers (a) and random copolymers (b).

2.1.3 Termination and chain transfer

Growing (living) polymer chains are terminated by several different mechanisms, which are summarized in Eqs. 2.14 - 2.25 below. For the reactions referred to as chain transfer to monomer, monomer units act to terminate the growing polymer chain rather than to propagate (i.e. add to the growing chain). In this manner, the chain is terminated, and the monomer unit turns "living" itself in the sense that it now carries a radical. The monomer unit is, in other words, itself enabled for further propagation, thus establishing a new growing chain. Chain transfer to CTA proceeds in the same way, but instead of having a monomer unit terminate the growing chain, the chain transfer agent of the system is responsible. For termination by chain recombination, two growing chains meet and terminate each other by combining to a longer (dead) chain. The mechanistic idea of this phenomenon is drawn in Fig. 2.5. Chain termination by disproportionation is somewhat similar to termination by recombination, in that two living polymer chains meet and terminate each other. For chain disproportionation, however, the resulting products are two dead chains instead of one long chain. These two chains have different characteristics at the endgroup, with one being unsaturated (having a double bond in the chain). The mechanism for this effect is drawn in Fig. 2.6. [3]

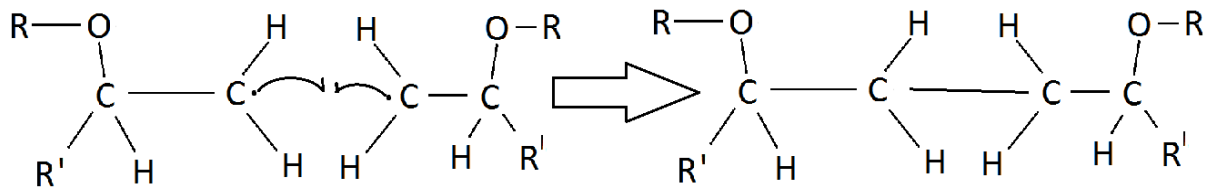


Figure 2.5: Mechanistic idea for polymer termination by chain recombination in a free-radical polymerization system.

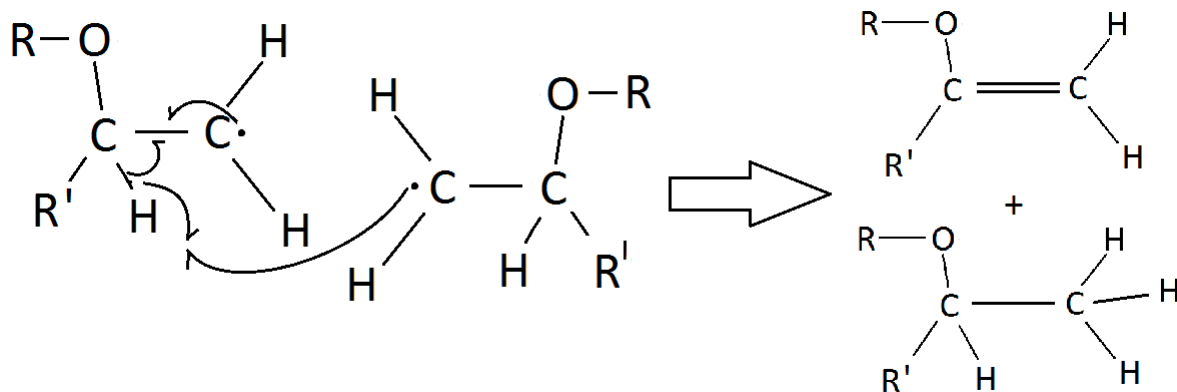
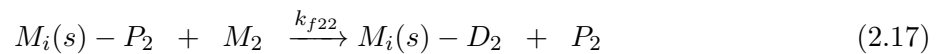
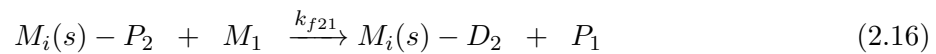
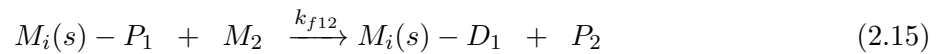
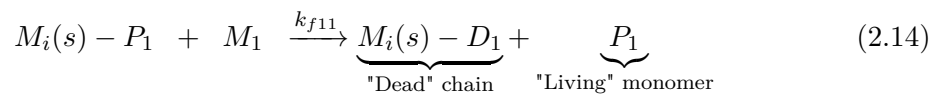
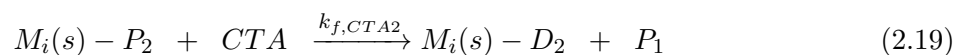
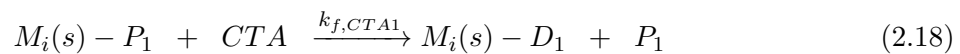


Figure 2.6: Mechanistic idea for polymer termination by chain disproportionation in a free-radical system.

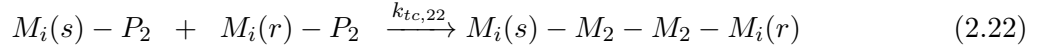
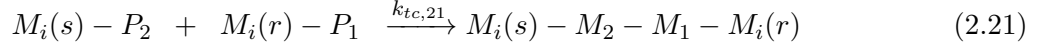
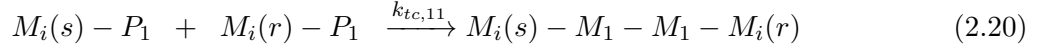
Chain transfer to monomer:



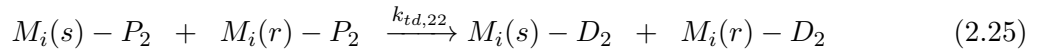
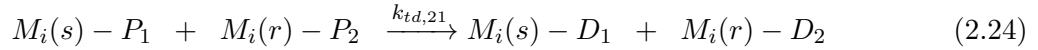
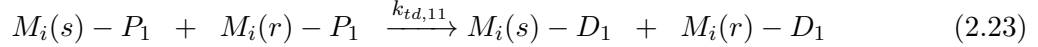
Chain transfer to CTA:



Termination by chain recombination:



Termination by chain disproportionation:



For modeling purposes, the large amount of kinetic rate constants are usually reduced by combining some of them to yield common rate constants. Cross-species chain transfer to monomer, for instance, is modeled as the geometric average of the homo-species transfer, as indicated in Eq. 2.26. The homo-species chain transfer to monomer, which is indicated in Eq. 2.27, is modeled in the same spirit as the Arrhenius type approaches for the previously modeled reaction rate constants. Adjustment factors are included for the purpose of parameter estimation later on.

$$k_{fij} = \sqrt{k_{fii}k_{fjj}} \quad , \quad i, j = 1, 2 \quad (2.26)$$

$$k_{fii} = \frac{k_{fii,0} \exp\left(-\frac{E_{fii}}{RT}\right)}{k_{fii,adj}} \quad (2.27)$$

The reaction rate constants for chain transfer to CTA can be modeled in a simplified way, by combining the contributions from the two types of endgroups, as indicated in Eq. 2.28. Here, an adjustment factor is included as well.

$$k_{f,CTA} = \frac{k_{f,CTA1} + k_{f,CTA2}}{k_{f,CTA,adj}} \quad (2.28)$$

For the termination reactions, the rate constants are combined into two rate constants, as indicated in Eqs. 2.29 & 2.30. These are, in turn, combined and adjusted as shown in Eq. 2.31. The result is one single parameter governing the total rate of chain termination. In comparison to having six individual parameters for chain termination, this may signal a reduction in the flexibility and accuracy of the model, but for the purpose of obtaining efficient models which perform well in the context of on-line optimization and control, these simplifications are appropriate.

$$k_{tc} = k_{tc,11} + k_{tc,21} + k_{tc,22} \quad (2.29)$$

$$k_{td} = k_{td,11} + k_{td,21} + k_{td,22} \quad (2.30)$$

$$k_T = \frac{k_{tc} + k_{td}}{k_{T,adj}} \quad (2.31)$$

This concludes the listing of the possible chemical reactions in the copolymerization system, which will constitute a crucial part of the complete process model.

2.2 Phase distribution in emulsions

A classic illustration of the most important components in a multi-phase emulsion copolymerization system is presented in Fig. 2.7, in order to give an overall representation of the behavior of these types of systems. This serves to sum up the preceding subsections on emulsion copolymerization, e.g. the fact that initiators reside in both water phase (represented by white space in the figure) and the polymer particle phase, and to motivate the discussion of phase distribution in such systems, which is of crucial importance when modeling multi-phase polymer systems.

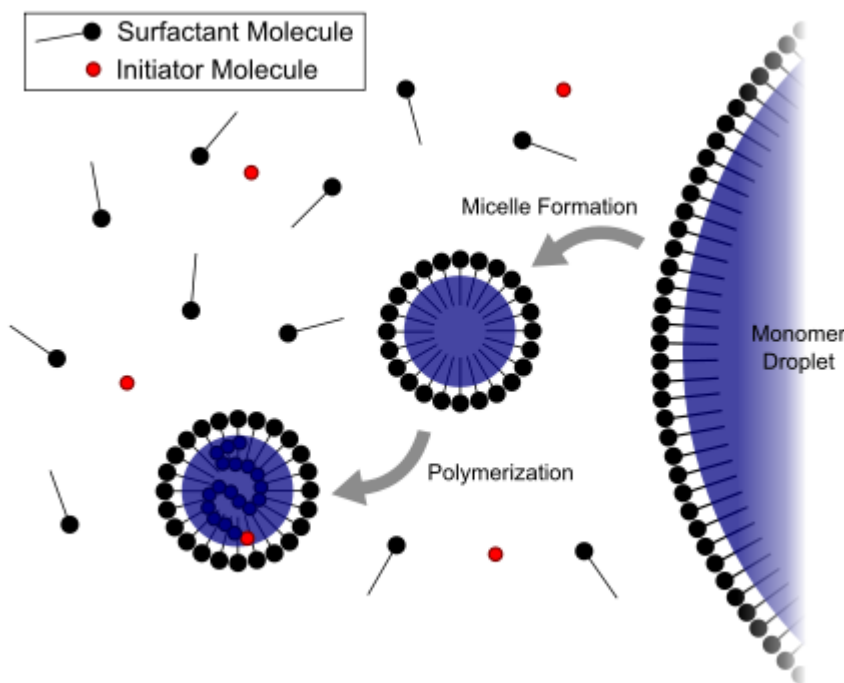


Figure 2.7: A classic illustration of emulsion polymerization in the context of phase distribution.

2.2.1 The different phases in a typical copolymer emulsion

There are three different phases typically considered in emulsion copolymerization processes, which are all included in Fig. 2.7. The aqueous (water) phase, represented by the white space in the figure, is largely made up of the water fed to the reactor, which acts as a bulk phase, dissolving the two other phases. Because of this, water is an important component within the system with respect to solubility issues and heat transfer properties, both between the various phases and with external components, such as a cooling jacket. The water phase is also the phase in which most of the initiator, CTA and emulgator/surfactant⁷ is dissolved. The aqueous phase can hold both

⁷Surfactant: **Surface active agent**. Surfactants are chemical species which align at the inter-phase boundaries in multi-phase systems such as emulsions, dispersions, etc., and act to either promote or prevent phase separation. Since surfactants most often are added in very small quantities, and don't participate in chemical reactions, the surfactants are often omitted from the modeling for simplicity.

monomer and polymer, but the solubility of these species are usually very low, sometimes even negligible, compared to the solubility in the (monomer) droplet phase and the (micellar) polymer particle phase. The micelles in which the polymer chains form and elongate are usually referred to as the particle phase of the emulsion system, and this will constitute the polymer product from the reactor. These micelles hold both initiator and CTA, in relatively small concentrations, while the main part of the micelles are growing polymer chains and unreacted monomer units. The third phase in the emulsion system is the monomer droplet phase, which is made up of unreacted monomer particles which have yet to travel to the particle phase (i.e. the micelles) to participate in polymer chain propagation. The behavior of the droplet phase depends on the nature of the monomer particles, but also on the operation of the reactor. In several applications, reactors are operated under so-called monomer starved conditions, in which the extent of the monomer droplet phase will be negligible. To achieve monomer starved conditions, the monomers are typically added carefully along the batch time in a typical semi-batch manner, such that a large excess of monomer is avoided at the beginning of the batch. The monomer is, in other words, transported (by molecular diffusion) to the polymer particle phase to participate in the chemical reactions of the system before the concentration of unreacted monomer is large enough to form a significant droplet phase. A common assumption is to rule out all possible chemical reactions from proceeding in the monomer droplet phase, as the extent of any reaction between the monomers will be negligible in this phase. As a remark to the discussion of the monomer droplet phase, it is emphasized that the presence of this phase is not believed to be negligible in the case of a continuous tubular reactor, and this phase must be accounted for in the models.

In certain copolymerization processes, the gas phase of the system must also be considered. Whether this is necessary or not usually depends on the nature of the monomers in the system in agreement with the operating conditions of the reactor. In several cases for emulsion copolymerization, both in industrial and lab-scale experiments, the modeling of the gas phase is not crucial for the successful modeling of the liquid part of the reactor system, which is where the chemical reactions of interest occur. In brief, the vapor pressure of the monomers under the operating conditions may, or may not, be large enough to provide a considerable contribution to the gas phase of the reactor. For the chemical system which is considered throughout this thesis, the gas phase is not included in the modeling work, and this simplification is assumed to be valid. This is in agreement with previous modeling work for the specific system. [1] [34]

2.2.2 Solubilities and internal transfer of mass and energy in the emulsion

When the behavior of the multi-phase emulsion copolymerization system is being formulated in the mathematical process model, both mass and energy transfer between the respective phases of the system must be considered. Although there are several different approaches to model this, depending on the knowledge of the emulsion system, the solubility of the components in the respective phases is usually an appropriate starting point. In some cases, so-called film models⁸, or even more complex models, can be used to describe the internal inter-phase mass transfer. For agitated (semi-)batch type reactors, the assumption of well-mixed reactor contents is usually valid, and this helps to support the use of solubility parameters to determine the inter-phase mass transfer. For a continuous tubular emulsion polymerization reactor however, the mixture is definitely not well-mixed, which will be duly discussed in Sec. 3.2. There will

⁸Film models are one, out of many, approach which can be deployed to describe the molecular diffusion of chemical species between different phases in a system. [19]

most likely be concentration gradients throughout the spatial domain of the reactor, but in the context of solving the model equations, the approach using the solubility of the monomers in the respective phases is useful, because it can be deployed to solve the phase distribution problem in each discrete point of the reactor in space. As discussed in Sec. 3, the spatially distributed reactor will typically be solved by some sort of discretization, and deploying the assumption of well-mixed conditions in each discrete piece of the reactor is believed to be a valid strategy.

In Eq. 2.32, the transfer of species i from phase w to phase p is formulated as the mass concentration difference between the equilibrium value (for instance given by the solubility of species i in phase p) and the actual value, multiplied with a mass transfer coefficient. In this formulation, both dynamics and steady-state behavior (governed by thermodynamics) are encapsulated. The difference in the concentrations will constitute the driving force of the mass transfer, and the mass transfer coefficient will govern the rate, i.e. the dynamics. The mass transfer coefficient ($k_{w \rightarrow p}$) and the equilibrium constant⁹ ($H_{i,eq}$) are, in a typical situation, obvious candidates for off-line parameter estimation in an extensive investigation of the model, at least in cases where detailed considerations with respect to mass transfer are desirable. An inescapable fact is that modifying the internal mass transfer in practice means modifying the extent of the chemical reactions in the reactor. This will in turn be demonstrated in Sec. 7.2.

$$\hat{m}_{i,w \rightarrow p} = k_{w \rightarrow p} (H_{i,eq} w_{i,w} - w_{i,p}) \quad (2.32)$$

In the specific modeling work of this thesis, so-called partition coefficients are introduced to describe the solubility of the monomers in the respective phases. The deployment of the respective partition coefficients are based on previous modeling work for the specific copolymerization system [34]. The partition coefficients set the solubility of the respective types of monomer in the respective phases. They govern, in other words, the steady-state solution to the phase equilibria in the system, but they also govern the driving forces in the dynamic inter-phase mass transfer formulations. A typical way to define a partition coefficient is given in Eq. 2.33. Here, k_i^p is the partition coefficient of monomer type i in the phase p , as compared to phase w . w_i^p and w_i^w indicate the weight fraction of monomer type i in phases p and w , respectively¹⁰.

$$k_i^p = \frac{w_i^p}{w_i^w} \quad (2.33)$$

The mass transfer formulation will be analogous to the one in Eq. 2.32, and the mass transfer rate constant ($k_{w \rightarrow p}$) will still govern the dynamics of this specific transport phenomenon.

The transfer of energy between the respective phases of the system can be modeled in an analogous manner. The driving force will, typically, be the temperature of the respective phases, and there will be a heat transfer coefficient governing the dynamics of the heat transport process. For considerations with respect to energy, equilibrium coefficients governing the state of equilibrium (e.g. partition coefficients), which are highly important in the context of mass transfer, are not introduced, as the state of equilibrium simply will be when the temperature in the two phases are the same. A slightly simplified way to consider the multi-phase system in the context of heat transfer, which have been done in this work, is to recognize that the aqueous phase, which

⁹This constant is denoted by H to emphasize the analogy of this mass transfer approach to the Henry's Law approach known for (dilute) vapor-liquid systems.

¹⁰The letters p and w are not chosen completely at random, but mean to indicate the polymer particle phase and the water phase, respectively.

mostly consists of water, makes up a large portion of the reaction mixture and encapsulates the two other phases of the system. It is assumed to be a valid strategy to model all external heat transfer going through the aqueous phase, with the aqueous phase distributing the heat on to the two other phases. This is also in agreement with the idea of a well-mixed reaction mixture.

2.3 Free-radical species modeling

In a free-radical copolymerization system, the behavior of the radical species is of key importance. In the previous sections, the important chemical reactions of the system are listed, and they all include active radical species. Because of this, the successful modeling of the radicals is crucial in order to properly describe the chemical behavior of the system. Several approaches are found in the literature for modeling the radicals, and a publication by Li & Brooks is of particular interest in this sense, which will be elaborated below. [10]

2.3.1 Number of radicals per polymer particle

For the notation used in modeling the radical species of the system, \bar{n} denotes the average number of radicals per particle. Furthermore, σ , k and C denote generation (radicals entering the particles), desorption (radicals exiting the particles) and decay (termination) of radicals from the particles, respectively. Extra parameters, defined to simplify the expressions, are presented in Eqs. 2.34 & 2.35.

$$\phi = \frac{2(2\sigma + k)}{2\sigma + k + C} \quad (2.34)$$

$$q = \sqrt{k^2 + 4\sigma\phi C} \quad (2.35)$$

The three approaches for radical species modeling which have been considered in the modeling work for the copolymerization cases of this thesis are listed and described in brief below.

1. Using a dynamic full population balance for particles carrying between 0 and i radicals, with i being a number chosen in advance. Choosing a high value for i gives more complexity, which in turn demands a higher computational effort, but it also increases the accuracy of the radical model. The question is whether this high accuracy is required or not. Using a full population balance, the radical model becomes as indicated in Eqs. 2.37 - 2.39. For this purpose, it is necessary to formulate n as indicated in Eq. 2.36, i.e. a vector containing the amounts of polymer carrying the respective numbers of radicals.

$$n = \begin{bmatrix} n_0 \\ n_1 \\ n_2 \\ \vdots \\ n_i \end{bmatrix} \quad (2.36)$$

Here, n_0 is the amount of polymer having zero radicals, n_1 has one radical, etc. The change in the polymer species will then be given as indicated below.

$$\frac{dn}{dt} = An \quad (2.37)$$

$$A = \begin{bmatrix} -\sigma & k & 2C & 0 & 0 & 0 & 0 & \dots \\ \sigma & -\sigma - k & 2k & 6C & 0 & 0 & 0 & \dots \\ 0 & \sigma & -\sigma - 2k - 2C & 3k & 12C & 0 & 0 & \dots \\ 0 & 0 & \sigma & -\sigma - 3k - 6C & 4k & 20C & 0 & \dots \\ 0 & 0 & 0 & \sigma & -\sigma - 4k - 12C & 5k & 30C & \dots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \end{bmatrix} \quad (2.38)$$

$$\frac{d\bar{n}}{dt} = \frac{1}{N_T} \begin{bmatrix} 0 & 1 & 2 & \dots & i \end{bmatrix} \frac{dn}{dt} \quad (2.39)$$

Here, N_T denotes the total number of particles in the system, which is assumed to be constant¹¹ for these equations to hold. This assumption is elaborated below. The complete strategy for constructing the A -matrix is given in App. A, where a MATLAB-script is provided to illustrate the general approach, depending on the chosen value for i .

- Using a steady state approximation for the average number of radicals per particle. Li & Brooks have written a comprehensible text where the topic is approximations to the full population balance approach [10]. One of the outcomes from the study is the steady-state approximate solution for the average number of radicals per particle, as indicated in Eq. 2.40. Using this approximation, the number of dynamic states for the system is reduced, but the model loses some of its accuracy.

$$\bar{n} = 2\sigma \frac{1 - \exp(-qt)}{(k + q) - (k - q) \exp(-qt)} \quad (2.40)$$

- The final suggestion is to use a dynamic Li-approximation for the average number of radicals per particle. The strategy in this case is similar to the steady state approximation, but an important difference is that the average number of radicals per particle is maintained as a dynamic state of the system. The formulation of this approximation is as indicated in Eq. 2.41. [10]

$$\frac{d\bar{n}}{dt} = \sigma - k\bar{n} - \phi C\bar{n}^2 \quad (2.41)$$

Among these different approaches, the latter is the one used in this work, because it provides a satisfying trade-off between accuracy and complexity (i.e. demand for computational effort). This leads to satisfying computational time without severely compromising the integrity of the model. A comparison between the three suggested approaches, illustrated for an entirely fictional case¹², is illustrated in Fig. 2.8. This example shows that the dynamic Li-approximation agrees very well with the full population balance approach. The steady-state approximation also shows decent agreement, apart from the behavior at the very start of the simulation. These results help to justify using any of the three approaches, but the dynamic Li-approximation is, as mentioned, a satisfying trade-off between the demand for computational and the accuracy of the model, and is hence the preferred approach for the purpose of this work.

¹¹A constant number of particles in an emulsion copolymerization system is usually valid when seed particles for polymerization is used. For a model operating with micellar nucleation, the total number of particles (N_T) will be a dynamic state of the system.

¹²Here, dimensionless time is used along the time axis.

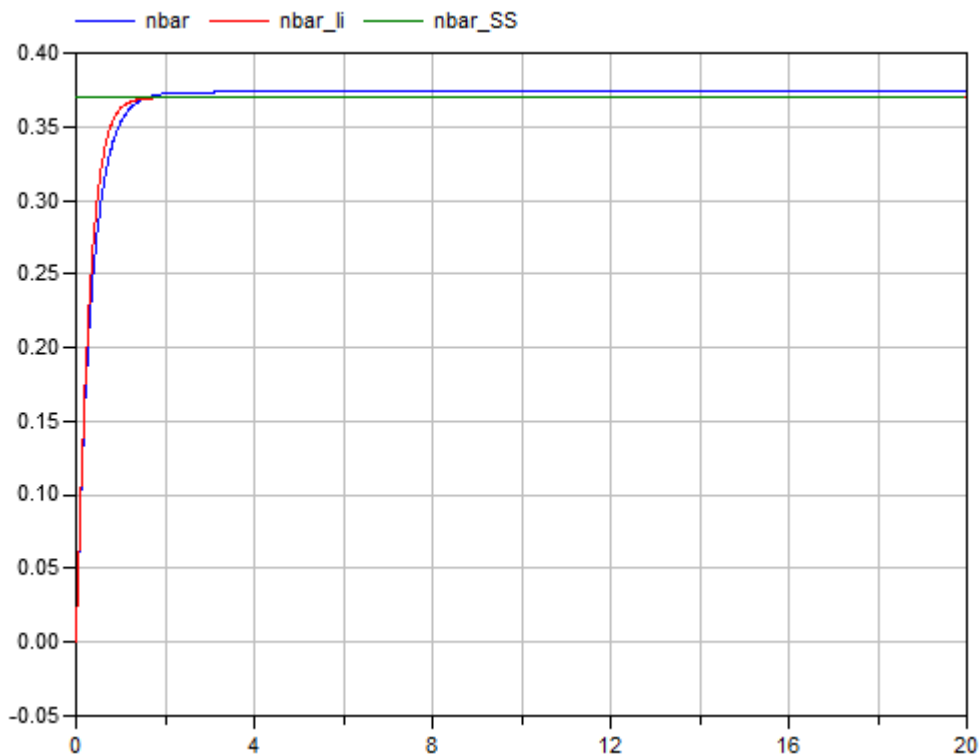


Figure 2.8: A plot showing the agreement between the three various approaches for radical species modeling, for a fictitious case. Curves show average number of radicals per particle versus dimensionless time.

2.3.2 Total number of polymer particles

Until now, the total number of growing polymer particles (N_T) has been assumed to be constant. This is a valid assumption for seeded polymerization processes, as mentioned, for which a known amount of polymer particles are fed to the reactor initially, and the formation of new particles are neglected. This approach is common for (semi-)batch reactor setups, in which all the seed particles can be added at the start of the batch. Generally speaking, seeded polymerization is possible for continuous reactor setups as well, but it requires a steady feed rate of seed particles, which may be hard to achieve. In certain cases, seeded polymerization is also found to cause clogging of the tubes in tubular reactors, yielding reduced reactor efficiency. In such cases, it would be desirable to operate the reactor without seed particles in the feed. Nucleation of polymer particles, i.e. the generation of growing radical polymer chains, which constitutes a negligible contribution in seeded polymerization, becomes highly important in unseeded polymerization. For emulsion polymerization, the phenomenon known as micellar nucleation is particularly evident¹³. This effect is important when the solubility of the respective monomers of the system

¹³The most important competing mechanisms to micellar nucleation are homogeneous nucleation (nucleation in the aqueous phase) and coagulative nucleation (nucleation in the droplet phase). None of these are considered in the work of this thesis, as they are negligible.

in water is low, meaning that chain propagation in the aqueous phase is negligible. For the specific copolymerization system studied in this thesis, this is the case indeed. For the unseeded approach, the total number of growing polymer particles will not be constant, but it will vary with the rate of (micellar) nucleation. To comprehend this phenomenon, the ideas of Nomura *et. al.* [11], which provide a valuable extension to the work of Gardon [12], are utilized. This approach is also discussed by Helgesen [7]. The total number of particles will now, rather than being a constant, be dynamic state of the system, described by a dynamic equation like the one shown in Eq. 2.42. Here, $\tilde{\rho}_I$ denotes the generation of growing polymer particles. This is related to the available amount of initiator (n_I), as the initiator molecules decompose to activate the polymer particles (recall Sec. 2.1). It is important to emphasize that this change in the particle number only proceeds as long as there are available emulsifier molecules in the system. The reason for this is that the emulsifier molecules of the reaction mixture facilitate the growth of the polymer particles by adsorbing to the surface of the polymer particles.

$$\frac{dN_T}{dt} = \begin{cases} \tilde{\rho}_I = 2f_I k_I n_I & \text{if vacant emulsifier molecules are present} \\ 0 & \text{else.} \end{cases} \quad (2.42)$$

The introduction of another state variable may impose a higher demand for computational effort in the equation system. In addition, the calculation of the total number of particles is expected to introduce a significant amount of stiffness to the model equations. In this context, stiffness is a property of the differential equation system which causes numerical instability for numerical solution of the equations. Unless extremely short step lengths are used for the numerical solvers, the equation solution will oscillate or sky-rocket¹⁴, because the magnitude of the gradient for the specific state is extremely large. The cause of this effect is related to time scales, in the sense that parts of the equation system can rapidly change the solution dramatically while other parts have relatively slow contributions. In many chemical systems, the kinetics of the various mechanisms vary a lot, and possible sources for stiffness in the differential equations are abundant. The strategy for solving stiff equation systems is to utilize sophisticated equation solvers, e.g. the stiff solvers of MATLAB (`ode15s`, `ode23s`, etc.) or CVODEs [13]. A key feature of these solvers is the adaptive, variable step length in the solution algorithms. They fight stiffness by deploying very short steps where it is required, yet they use sensibly large steps in the regions of the equation system where the stiffness is not evident. The result is an effective, yet correct, solution for the differential equation system. With that being said, it is emphasized that extreme stiffness can still constitute a problem, even to the more sophisticated solvers.

The calculation of the total number of particles is, as mentioned, a typical source of stiffness to the equation system, and because of this a proposed strategy to model the total number of particles is to deploy a steady-state solution to Eq. 2.42 and assume that it remains constant after a short time period at the very start of the simulation. This assumption is justified by the very fast dynamics of the micellar nucleation process. The steady-state solution to the equation, in agreement with the Smith-Ewart theory, is provided in Eq. 2.43. Here, the numerical pre-factor (α) can vary between a lower and an upper limit based on whether the particle growth is assumed to be proportional to the generation of radicals or proportional to the area of growing

¹⁴The term sky-rocket refers to a situation where a state is dramatically changed in a manner where it shoots out of bounds without returning to a sensible value, thus compromising the consistency of the equation system.

particles, respectively. [11]

$$N_T = \alpha \left(\frac{\tilde{\rho}_I}{\mu} \right)^{0.4} (a_s S_0)^{0.6} \quad \alpha \in [0.37, 0.53] \quad (2.43)$$

Furthermore, a_s is the specific surface area of coverage for each emulsifier molecule, while S_0 denotes the initial amount of emulsifier molecules. This solution is established by assuming that the volumetric growth rate of the particles (μ) is constant throughout the nucleation process. This assumption is not always easy to justify, and the fact that the actual growth is hard to determine points back to using a dynamic equation for the particle number. A suggestion to formulate this dynamic equation in an effective manner is based on the work of Helgesen & Singstad [7], in which the number of particles continues to increase until the free emulsifier molecules in the system are depleted, i.e. adsorbed to the surface of the polymer particles. A way to determine this critical point is to identify when the surface area of the polymer particles is equal to the total surface coverage area of all the initial emulsifier molecules. The introduced quantity a_r represents the ratio between the surface area of all the polymer particles and the covered surface area of all the emulsifier molecules. This ratio is defined in Eq. 2.45. The specific surface area of each polymer particle is calculated as shown in Eq. 2.46, where the known volume of the entire particle phase (V_p) is used to find the surface area of each particle, using the relationship between the surface area and the volume of a sphere. An important assumption in this context is that the polymer particles are monodisperse, i.e. that all particles are equally large.

$$\frac{dN_T}{dt} = \begin{cases} \tilde{\rho}_I (1 - a_r) & \text{if } a_r < 1 \\ 0 & \text{else.} \end{cases} \quad (2.44)$$

$$a_r = \frac{N_T a_p}{a_s S_0} \quad (2.45)$$

$$a_p = (4\pi)^{\frac{1}{3}} \left(\frac{3V_p}{N_T} \right)^{\frac{2}{3}} \quad (2.46)$$

2.4 Copolymer product quality

There are several quantities to calculate in order to investigate the quality/state of the copolymer product in a typical polymerization process. The most common quality identifier for a copolymer blend is the molecular weight distribution of the copolymer. Usually, two different formulations for molecular weight are used, as presented in Eqs. 2.47 & 2.48, which in turn can be used to formulate a quantity known as Polydispersity Index (PDI), as presented in Eq. 2.49. For industrial applications, the product needs to meet the product specification as closely as possible, and monitoring these quantities are crucial. One could imagine using other averages for the molecular weight distribution as well, for instance formulated using a higher order of polymer moments, but this is not of essence in this thesis, and has not been investigated. Particle size distributions are also sometimes introduced to identify the polymer product. This has not been granted much attention in this thesis. A comprehensible publication on molecular weight distributions for polymer systems is written by Crowley & Choi, made specifically for a free-radical polymerization system for control purposes [17]. The use of polymer moments for characterizing the copolymer

system is also discussed by Gao & Penlidis [18] and Wyman [20], among others.

$$\text{Number average molecular weight: } M_n = \sum_{i=1}^m (Y_i M_{Mi}) \frac{\mu_1^{P1} + \mu_1^{P2} + \mu_1^D}{\mu_0^{P1} + \mu_0^{P2} + \mu_0^D} \quad (2.47)$$

$$\text{Weight average molecular weight: } M_w = \sum_{i=1}^m (Y_i M_{Mi}) \frac{\mu_2^{P1} + \mu_2^{P2} + \mu_2^D}{\mu_1^{P1} + \mu_1^{P2} + \mu_1^D} \quad (2.48)$$

$$\text{Polydispersity index: } PDI = \frac{M_w}{M_n} \quad (2.49)$$

In these formulations, μ_i^j denotes the i 'th order moment with respect to component j as active component. $P1$ and $P2$ denote active chains where the endgroup is monomer type 1 and 2, respectively, while D denotes dead/deactivated polymer chains. The various polymer moments are formulated in a systematic manner in the appendix, App. B. Furthermore, Y_i denotes the abundance of monomer type i in the polymer, i.e. the relative consumption of monomer type i , while M_{Mi} is the molecular weight for each monomer unit of type i .

The degree of conversion of monomer in the reactor is another important quantity to monitor in a polymerization reactor. This is discussed briefly in App. C, where the theoretical backbone for semi-batch reactor modeling is established. Here, two alternative formulations for monomer conversion is presented in Eqs. C.10 & C.11. In most copolymerization processes, the conversion of monomer is desired to be as high as possible, and in the context of on-line control it is interesting to investigate how to achieve this while maintaining safe operation of the reactor. For reasons related to the kinetics of the system, this is not always achievable within the time limits of the batch, and strategies to optimize the conversion are very interesting. For a tubular reactor, the conversion of monomer through the reactor is calculated by comparing the concentration of the respective monomers at reactor outlet compared to reactor inlet. This is formulated mathematically in Eq. 2.50, where \hat{m}_i^0 and \hat{m}_i denote the total mass flow of monomer type i at reactor inlet and outlet, respectively.

$$X = \frac{\hat{m}_1^0 + \hat{m}_2^0 - \hat{m}_1 - \hat{m}_2}{\hat{m}_1^0 + \hat{m}_2^0} \quad (2.50)$$

The conversion of monomer is also related to the residence time distribution of the reactor, which is discussed in Sec. 3.2. In brief, shortening the mean residence time of the reactor means lowering the conversion of monomer, simply because the reactants have less time to react through the reactor. Altering the feed conditions, i.e. the respective ratios between the reactants, will also affect the conversion of monomer, but such a manipulation to the system must be done with extreme caution, as the polymer end product potentially can change its properties dramatically when the feed changes. The conversion of monomer, i.e. the extent of the chain propagation reactions, can also be influenced through controlling the reactor temperature, as the kinetics of the reactions are highly temperature dependent, as introduced in Sec. 2.1. These concepts are explored in some of the simulation studies of Sec. 7.

Section 3

Concepts and methodology of tubular reactor modeling

The purpose of this section is to establish the mathematical relationships and problems encountered when modeling distributed systems. A distributed system refers to a system which is not spatially uniform. In addition, the dynamic time-varying behavior of the system is essential for most systems. The behavior of the system when it is not operated at steady-state, for instance, is important to encapture in the context of on-line optimization and control. As a consequence, development and solution of partial differential equations are among the key activities for distributed systems. In this work, tubular reactors for emulsion copolymerization are considered. Emphasis is put, however, on not underestimating the value of the general approaches to model a distributed system which are presented in this section, which are applicable for a wide range of physical problems. For the purposes of on-line optimization and control of a distributed system, the numerical efficiency of the model is important. As a consequence, the numerical performance has been one of the main properties of interest throughout the development of the various models. In order to ensure agreement between the established reactor models and the mixing effects in a real reactor, residence time distributions are used to tune the models. This is explored in Sec. 3.2.

3.1 Introduction: Problem formulation

The specific problem in mind for this thesis is a tubular reactor for free-radical emulsion copolymerization. In Fig. 3.1, a generic tubular reactor is shown, having a perfect cylindrical geometry. A section of the reactor, having a width of ΔL , is highlighted together with flow lines indicating a flat and uniform velocity profile, for simplicity. Continuous flow reactors can have strongly varying geometries, but this thesis only concerns perfect cylinders, favoring the use of cylindrical coordinates for the mathematical formulation of the model equations. For a cylinder, a distinction is often made between the so-called "coin character" ($R \gg L$) and the so-called "pole character" ($L \gg R$). The type most often encountered in industrial scale chemical reactors is the pole character, having relatively small cross-sectional areas compared to the length of the reactor. The so-called smart-scale experimental reactor, on which some of the model validation in this thesis is based, has a geometry where the length of the reactor is approximately one hundred times as large as the diameter of the cross-section. The reactor has, in other words, an obvious pole character. In real tubular reactors, the fluid flow patterns can have very varying behavior, giving rise to complexity with respect to mixing effects in the reactor. Whether the flow regime of the reactor is turbulent or laminar, for instance, will affect the behavior of the reactor. The flow properties of the tubular reactor are closely related to the residence time distribution of the reactor, which is explored in Sec. 3.2. This provides a way to examine the degree of mixing effects within the reactor, which in turn will prove valuable when the separate approaches to solve the model equations for the distributed system are evaluated.

In the context of polymer industry, the amount of tubular reactors, and continuous processes

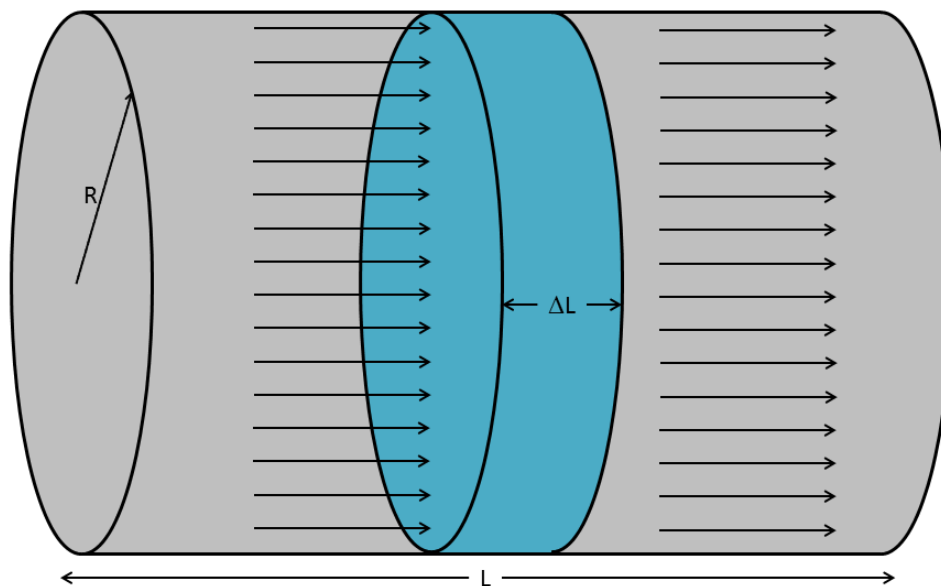


Figure 3.1: A conceptual illustration of a tubular reactor with constant cross-sectional area.

in general, is scarce in comparison to the amount of the more popular batch and semi-batch reactors. One of the benefits of a tubular reactor setup is the improved heat transfer properties in comparison with traditional batch and semi-batch reactors. One of the reasons for this is that the tubular reactor typically has a larger surface area between the reactor and the cooling mechanism, i.e. the cooling jacket encapsulating the reactor. As a consequence, the potential external heat transfer is larger. As mentioned in Sec. 2, reacting polymer systems are sensitive to temperature changes, and temperature control is one of the main challenges regarding such a reactor. Because of this, temperature control is a crucial activity in order to achieve operation in agreement with the safety regulations as well as optimized production rates of the desired product. It is worth noticing that in a typical batch type reactor setup, which is assumed to be well-mixed, the temperature is uniform for the entire reactor, while a tubular reactor has a spatial temperature profile which may vary significantly through the reactor.

3.1.1 Balance principle equations

For the purpose of establishing balance equations for a distributed system, a completely general and arbitrary control volume can be considered. Such a body is drawn in Fig. 3.2, where the body resides stationary in a field of moving fluid. For a generalized quantity, using words, the balance principle for the conservation of the quantity can be formulated as shown in Eq. 3.1. This approach towards formulating balance equations is sometimes referred to as the continuity approach.

$$\left(\begin{array}{c} \text{Rate of} \\ \text{accumulation} \\ \text{of quantity} \end{array} \right) = - \left(\begin{array}{c} \text{Net amount of quantity} \\ \text{leaving the boundary} \end{array} \right) + \left(\begin{array}{c} \text{Net amount of} \\ \text{quantity generated} \end{array} \right) \quad (3.1)$$

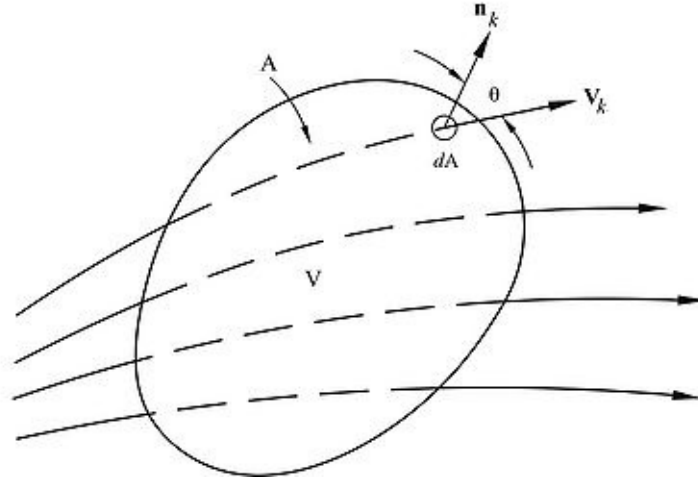


Figure 3.2: An arbitrary control volume having a volume \mathcal{V} and a surface area \mathcal{A} .

Expressing the balance principles using symbols, in which ψ denotes the arbitrary and general (intensive) quantity, the resulting equation becomes as shown in Eq. 3.2. The net generation of ψ is here denoted by $\tilde{\sigma}$. E.g. for a case where the component balance for a specific chemical compound is established, the generation term would represent the net generation of the specific species due to chemical reactions. For conserved quantities such as energy and total mass, the net generation is zero. The same would be the case in a species mass balance considering an inert (non-reacting) species, which typically is the case in studies of residence time distributions, where a so-called tracer compound is used. The negative sign of the surface integral is due to the choice of normal vector to the surface (\vec{n}), which is traditionally chosen such that the outward flow of the element is considered, in agreement with Eq. 3.1 and Fig. 3.2. The velocity field of the moving fluid is denoted by \vec{v} .

$$\frac{\partial}{\partial t} \left(\iiint_{\mathcal{V}} \psi dV \right) = - \oiint_{\mathcal{A}} (\psi \vec{v} \cdot \vec{n}) dA + \iiint_{\mathcal{V}} \tilde{\sigma} dV \quad (3.2)$$

Deploying the Gaussian Divergence Theorem¹, as provided in Eq. 3.3, the surface integral in Eq. 3.2 can be transformed to a volume integral.

$$\oiint_{\mathcal{A}} (\vec{F} \cdot \vec{n}) dA = \iiint_{\mathcal{V}} (\vec{\nabla} \cdot \vec{F}) dV \quad (3.3)$$

Introducing this modification, the equation is changed in the sense that all terms become volume integrals. For Eq. 3.4 to hold, the integrands must agree, i.e. Eq. 3.5 must also hold. Eq. 3.4 and Eq. 3.5 are referred to as the integral and differential forms of the balance principle equation,

¹This theorem is often referred to as Gauss' theorem (not to be confused with Gauss' law), Ostrogradsky's theorem, or simply The Divergence Theorem. The theorem represents a special case of Stokes' theorem in vector calculus. [23]

respectively. [19]

$$\frac{\partial}{\partial t} \left(\iiint_{\mathcal{V}} \psi dV \right) = - \iiint_{\mathcal{V}} \vec{\nabla} \cdot (\psi \vec{v}) dV + \iiint_{\mathcal{V}} \tilde{\sigma} dV \quad (3.4)$$

$$\implies \frac{\partial \psi}{\partial t} + \vec{\nabla} \cdot (\vec{v} \psi) = \tilde{\sigma} \quad (3.5)$$

This result (Eq. 3.5) is the PDE describing the changes in the quantity ψ in both time and three-dimensional space, which is a desirable result with respect to formulating the model equations for the distributed system. In going from Eq. 3.4 to Eq. 3.5, the Leibniz' Integral Rule was deployed to move the differential operator under the integral sign on the left hand side of the equation. It is emphasized that up until now, no assumptions have been made with respect to the geometry or behavior of the arbitrary body, and the equations are completely general. The equation can be deployed to describe the spatial distribution of any quantity, basically.

Turning the attention back towards the tubular reactor, which is the system for which the equations will be deployed, the equations are most appropriately treated using cylindrical coordinates, considering the geometry of the reactor tube. Formulated in cylindrical coordinates, the divergence of an arbitrary vector field (\vec{F}) is given as indicated in Eq. 3.6. In this notation, r , θ and z denote the radial, azimuthal and axial coordinates, respectively².

$$\vec{\nabla} \cdot \vec{F} = \frac{1}{r} \frac{\partial}{\partial r} (r F_r) + \frac{1}{r} \frac{\partial F_\theta}{\partial \theta} + \frac{\partial F_z}{\partial z} \quad (3.6)$$

For the tubular reactor system, circular symmetry is assumed. This is a valid assumption for many reactor systems, and the resulting effect is no azimuthal gradients. Another implication of this assumption is that gravitational contributions to the fluid flow pattern are neglected. By further manipulation of Eq. 3.5, and introducing the assumptions so far, a more explicit form of the expression (with respect to the coordinates) is achieved. This form of the equation is shown in Eq. 3.7.

$$\frac{\partial \psi}{\partial t} + \frac{1}{r} \frac{\partial}{\partial r} (r v_r \psi) + \frac{\partial}{\partial z} (v_z \psi) = \tilde{\sigma} \quad (3.7)$$

This result (Eq. 3.7) is an important achievement in the sense that it serves as a starting point for formulating the various model equations for the tubular reactor. The derivation which has been done in this section is a typical vector calculus approach to achieve the balance equation, and for the purpose of confirming the equation, this specific result is also derived using a so-called shell balance approach in the appendix (App. D). The resulting equation is the same in both approaches, which helps to confirm each of the derivations. The quantities of interest when making an efficient model of a tubular reactor are separate mass balances for the respective species of the reactor as well as an energy balance. Momentum equations, i.e. Navier-Stokes equations, have not been considered in these simplified reactor considerations, but including such equations are advised for in-depth studies, and the momentum equations become increasingly important when there is a considerable gas phase to model in the system. [19]

²The transformation between cartesian type coordinates and cylindrical coordinates is obtained in a fairly straight-forward manner, and it has been introduced in App. D.

3.1.2 The mass balances

The various species mass balances are of key importance when any chemical reactor is being modeled. By using the established general balance principle (Eq. 3.7), the desired species mass balance can be developed like indicated in Eq. 3.8. The main quantity of interest in this context is the concentration of each chemical component in the mixture, i.e. $\psi = c_i$ for component i . The net generation term is represented by the net generation of species i due to chemical reactions, i.e. $\tilde{\sigma}_i = \tilde{R}_i$. Here, \tilde{R}_i represents the sum of net generation of component i from all independent chemical reactions in the system (n_r) in agreement with Eq. 3.9.

$$\frac{\partial c_i}{\partial t} + \frac{1}{r} \frac{\partial}{\partial r} (r v_r c_i) + \frac{\partial}{\partial z} (v_z c_i) = \tilde{R}_i \quad (3.8)$$

$$\tilde{R}_i = \sum_{k=1}^{n_r} r_{k,i} \quad (3.9)$$

At this point, the dynamic equation (Eq. 3.8) describes the convective movement of the fluid in the system, i.e. the combination of advective transport (bulk movement) and transport due to mass diffusion. For the purpose of the species mass balances, these two terms are often separated since many physical cases often allow for neglecting one of them. The first step towards separating the two phenomena is to decompose the fluid velocity as indicated in Eq. 3.10, in which u denotes the fluid bulk velocity while v_d denotes the diffusive fluid velocity.

$$v = u + v_d \quad (3.10)$$

The product between the diffusive velocity (v_d) and the concentration of species i (c_i) equals the diffusive transport of species i , which is related to the chemical diffusivity (D_i) for species i through Ficks first law, as provided in Eq. 3.11. Here, an arbitrary spatial coordinate (ξ) is used to emphasize the general case.

$$\hat{n}_i = c_i v_{d,i} = -D_i \frac{\partial c_i}{\partial \xi} \quad (3.11)$$

Introducing Eq. 3.11 into Eq. 3.8 for the respective spatial coordinates, the distinction between advective motion and diffusive motion is achieved. The result is shown in Eq. 3.12, where the diffusive terms are moved to the right hand side by convention.

$$\frac{\partial c_i}{\partial t} + \frac{1}{r} \frac{\partial}{\partial r} (r u_r c_i) + \frac{\partial}{\partial z} (u_z c_i) = \frac{D_i}{r} \frac{\partial}{\partial r} \left(r \frac{\partial c_i}{\partial r} \right) + D_i \frac{\partial^2 c_i}{\partial z^2} + \tilde{R}_i \quad (3.12)$$

The result obtained in Eq. 3.12 could be subject to several possible simplifications, depending on the physical conditions of the specific reactor system in mind. For the further considerations in this work, the reactor tube will be assumed to be well-mixed in the radial direction, which implies no radial advection and no radial mass diffusion. In a way, this is represented in the simplified illustration provided at the very start of the section (Fig. 3.1). In addition, the diffusivity is assumed to be constant³, and the change in the bulk velocity with respect to the

³This is a very common assumption in many cases. The diffusion coefficient may be dependent on physical properties of the reaction mixture, such as density or temperature, but to use a constant value is often valid.

axial coordinate is assumed to be negligible⁴. The result is a dynamic partial differential equation (PDE) with one spatial dimension, as presented in Eq. 3.13. This equation is often referred to as the axial dispersion model, and constitutes the final result for the species mass balances for the purpose of this thesis.

$$\frac{\partial c_i}{\partial t} + u_z \frac{\partial c_i}{\partial z} - D_i \frac{\partial^2 c_i}{\partial z^2} = \tilde{R}_i \quad (3.13)$$

When treating physical systems, additional simplifications are applicable to the axial dispersion model depending on the operation mode of interest and the specific properties of the system. Some of the most common additional simplifications to the species mass balance are briefly presented in Tab. 3.1.

3.1.3 The energy balance

One of the main challenges in treating reacting polymer systems is, as already mentioned, the temperature control of the reactor. In the context of model-based control, it would seem excessive to state that effective temperature control is dependent on a successful process model, in particular with respect the energy balance of both the reactor itself and the cooling mechanism. Energy is a conserved quantity like total mass, but for the purpose of the reactor model the so-called temperature explicit form of the energy balance is often desired⁵. The result is a PDE describing changes in temperature, which often proves to be more appropriate than the changes in total energy [19]. The reason is that people tend to comprehend the state of a system more intuitively by the use of temperatures (e.g. 250 degrees Celsius) rather than the total content of energy (e.g. fifty million joule).

The strategy to simplify the energy balance equation, starting from the general balance principle (Eq. 3.1) is analogous to what was done for the species mass balance. The general balance principle equation in cylindrical coordinates from Eq. 3.7 is, in other words, used to account for the energy of the reactor as well. The diffusive flux of energy is expressed by Newtons law for heat diffusion, which is analogous to Ficks first law for mass diffusion, and the resulting expression is shown in Eq. 3.14. In this expression, q denotes the net external heat provided to the reactor vessel, as indicated in Eq. 3.17. Here, U denotes the specific heat transfer coefficient for the reactor wall, while A denotes the surface area of contact between the reactor itself and the engulfing cooling jacket. In most emulsion polymerization cases, cooling is required, yielding a negative value for q . This effect is consistent for the equation system when the cooling jacket temperature is lower than the reactor temperature ($T_J < T_R$), which is intuitive. The heat capacity of the reactor contents is denoted by C_p and modeled as indicated in Eq. 3.16, where the specific heat capacities of the chemical species ($c_{p,i}$) in the reaction mixture are weighed by their respective mass fractions (w_i) in the mixture. The heat developed from chemical reactions within the reactor is represented by the sum of separate heats of enthalpy from the respective

⁴The implication of this statement is that $\frac{\partial u_z}{\partial z} \approx 0$. This assumption is usually valid for the specific system studied in this thesis, mainly because there are only minor changes in the density of the reaction mixture through the reactor.

⁵The term temperature balance is sometimes used to refer to this type of equation, but this must be said with caution, since temperature is indeed an intensive quantity which is not conserved in the same manner as energy and mass.

Table 3.1: Typical additional simplifications to the species mass balance for a continuous tubular reactor.

Simplification	Effect
$\frac{\partial c_i}{\partial t} = 0$	The steady-state model. The time-varying behavior of the model is not considered, as the system is assumed to operate at steady-state. For continuous operation, the steady-state behavior of the system is of importance, but the time-varying behavior is crucial for the purpose of on-line optimization and control. As a consequence, this version of the balance equations are rarely considered directly in this thesis.
$u_z = 0$	No bulk movement. The only spatial mass transfer effects are due to diffusion. Because of the continuous feeding to the reactor, bulk movement is unavoidable for the tubular reactor, and this version of the balance equations is only of minor interest.
$D_i = 0$	Non-diffusing species model. For species with low diffusivity, this simplification may prove to be valid. Axial mass transport happens exclusively due to bulk movement, by assumption. This assumption is probably the most interesting with respect to simplifying the model equations for the tubular reactor in this thesis. By introducing this simplification, the terms with the second derivatives are eliminated, which is believed to ease the solution of the equations significantly.
$\tilde{R}_i = 0$	Non-reacting (inert) species. For an inert species, i.e. a tracer, moving through the reactor due to bulk movement and diffusion, there will be no net generation or consumption. This is interesting in the context of residence time distributions for the reactor, which will be investigated in Sec. 3.2, but not for the balance equations for the reacting species.
$\frac{\partial c_i}{\partial t} = 0, D_i = 0$	This combination of simplifications is known as the plug flow reactor (PFR) model , which is a popular reactor model for simplified steady-state systems. This is a non-diffusive reactor model which is only considered at steady-state operation.
$u_z = 0, \tilde{R}_i = 0$	This combination of simplifications yields an analogy to Ficks second law . For the tubular reactor system of this thesis, this is not particularly relevant.

reactions of the system, as shown in Eq. 3.15.

$$\rho C_p \left[\frac{\partial T}{\partial t} + \frac{1}{r} \frac{\partial}{\partial r} (r u_r T) + \frac{\partial}{\partial z} (u_z T) \right] = \rho C_p \left[\frac{k_e}{r} \frac{\partial}{\partial r} \left(r \frac{\partial T}{\partial r} \right) + k_e \frac{\partial^2 T}{\partial z^2} \right] - \Delta H_{rx} + q \quad (3.14)$$

$$\Delta H_{rx} = \sum_{k=1}^{n_r} \Delta H_{rx,k} \quad (3.15)$$

$$C_p = \sum_{i=1}^{n_c} w_i c_{p,i} \quad (3.16)$$

$$q = UA(T_J - T_R) \quad (3.17)$$

By using the same approach as for the species mass balance (which resulted in Eq. 3.13), the energy balance is simplified to yield the expression in Eq. 3.18. Among the most important simplifications are the assumption of no radial temperature gradients inside the reactor. This assumption is probably one of the most questionable ones of the entire thesis, since the external heat to the reactor, i.e. cooling, is added through the reactor wall. By assuming that the fluid inside the reactor is well-mixed, however, and that the reactor wall itself is limiting with respect to heat transfer, the assumption is believed to be valid. Another assumption is that the axial velocity of the bulk fluid is approximately constant, which was also assumed for the mass balances.

$$\rho C_p \left[\frac{\partial T}{\partial t} + u_z \frac{\partial T}{\partial z} - k_e \frac{\partial^2 T}{\partial z^2} \right] = -\Delta H_{rx} + q \quad (3.18)$$

The result is a dynamic heat balance equation, on the so-called temperature explicit form, for a one-dimensional spatial problem. This will constitute an important addition to the collection of species mass balances which are already established.

3.1.4 Boundary conditions for tubular reactor problems

The behavior of the system is governed by the boundary conditions (BC) of the system. With the already established balance equations for the reactor, the boundary conditions will affect the specific quantitative behavior of the system, and in the context of obtaining analytical solutions, the boundary conditions usually determine whether a PDE has a readily obtainable solution or no analytical solution at all. The slightest changes in the boundary conditions may sometimes be enough to dramatically worsen (or simplify) the road towards the solution of the equation. The purpose of the boundary conditions is to establish an agreement between the mathematical behavior of the PDEs and the physical interpretation of the system. Combining the established PDEs with proper boundary conditions will yield what is commonly referred to as an initial boundary value problem (IBVP). In such a problem, initial values (with respect to time) and boundary values (with respect to the spatial interpretation of the problem) are provided, and PDEs are solved to obey them. For a tubular reactor, for instance, the feeding conditions will dictate the boundary conditions at one spatial end of the reactor, and although the model equations of the reactor are unchanged, changing the boundary conditions will change the solution. By changing the feed composition of the tubular reactor, for instance, a new steady-state solution is achieved for the operation of the reactor.

Several types of boundary conditions are used for tubular reactors, depending on the nature of the operation mode. A common way of bounding a tubular reactor is known as the Danckwerts boundary conditions [19], in which both the conditions at the reactor inlet and *the changes* in the conditions at the reactor outlet is known at all times, as indicated in Eqs. 3.20 & 3.21. In addition to this, the initial distribution (at time $t = 0$) of the quantity in mind (ψ) in the system is given as indicated in Eq. 3.19.

$$t = 0 : \quad \psi(0, z) = f_0(z) \quad (3.19)$$

$$t > 0 : \quad \psi(t, 0) = \psi_0(t) \quad (3.20)$$

$$\left. \frac{\partial \psi}{\partial z} \right|_{z=L} = 0 \quad (3.21)$$

In the case where analytical solutions for the PDEs are desired, the boundary conditions should preferably be homogeneous. This is a crucial advantage when solving the equation system arising from the combination of the boundary conditions with the PDE. In certain cases, homogeneous boundary conditions can be achieved although they are initially inhomogeneous. A transformation to homogeneous boundary conditions is shown in Eq. 3.22. The idea is to define a slightly modified function for which the specific equation is solved, after which the function (solution) is transformed back to the solution of interest.

$$\begin{array}{ccc} \psi(t, 0) = \psi_0(t) & \xrightarrow{\tilde{\psi} = \psi - \psi_0(t)} & \tilde{\psi}(t, 0) = 0 \\ \frac{\partial \psi}{\partial z}(t, L) = 0 & & \frac{\partial \tilde{\psi}}{\partial z}(t, L) = 0 \end{array} \quad (3.22)$$

Boundary conditions for PDEs come in different forms, and they are typically separated into three main groups. The three main types of boundary conditions⁶ are presented in Tab. 3.2, together with their characteristics. For a tubular reactor model, the Robin type boundary conditions are the most commonly used, and Robin type boundary conditions are deployed in the work of this thesis. The BC in Eqs. 3.19 - 3.21 are typical Robin type BC, for instance.

Table 3.2: Main types of boundary conditions for PDEs.

Type	Characteristics
Type 1: Dirichlet	The Dirichlet type boundary conditions require the solution of the specific equation to be known at the boundaries of the problem. In a tubular reactor, this would imply knowing the temperature, concentrations, etc. at both the reactor inlet and the reactor outlet.
Type 2: Neumann	The Neumann boundary conditions require the derivatives of the solution, rather than the actual solution, to be known at the boundaries of the problem.
Type 3: Robin	In a Robin type boundary value problem, a combination of solutions and derivatives are provided at the boundaries of the problem. The Robin problem is, in other words, a hybrid between a Dirichlet type problem and a Neumann type problem. For a tubular reactor problem, this is very appropriate, because the conditions (i.e. the solutions) can be specified at the reactor inlet, while the changes (i.e. the derivatives of the solutions) can be specified at the reactor outlet. A usual way to handle this is to state that all changes cease at the reactor outlet, i.e. all derivatives are zero at the outlet.

3.2 Residence time distribution for flow behavior analysis

The residence time distribution (RTD) for a chemical reactor is a valuable concept to determine the flow properties of the system. In essence, investigating the RTD means tracing an inert

⁶The three main types of boundary conditions are attributed to the scientists Peter G. L. Dirichlet (1805 - 1859), Carl G. Neumann (1832 - 1925) and Victor G. Robin (1855 - 1897), respectively.

chemical species, usually referred to as a tracer compound, through the reactor and analyzing the concentration of the specific tracer at the reactor outlet over time. The various contributions from advective and diffusive terms in a convective flow system can be elegantly analyzed using a residence time consideration, and for the purpose of this thesis this concept has been used to evaluate the internal mixing effects in the developed reactor models. RTD for various reactor types have been subject to extensive analysis in the literature and the considerations in this context are inspired by Levenspiel as well as Scott Fogler. [30] [31]

The main quantities calculated as part of the residence time distribution are the so-called age distribution (denoted by $E(t)$) and the cumulative age distribution (denoted by $F(t)$), which is the integral of $E(t)$, as shown in Eq. 3.23. The E-distribution represents the instantaneous concentration of the tracer compound at the reactor outlet as a function of time. The F-distribution consequently indicates how much of the total amount of tracer fed to the reactor that have left the reactor as a function of time. Both distributions are usually normalized to provide a comprehensible scaling of the distributions.

$$F(t) = \int_{-\infty}^{\infty} E(t)dt \quad (3.23)$$

The idea behind the RTD experiment is, in essence, to feed the reactor a pulse of a tracer compound at a certain time (t_0). In a purely mathematical sense, this pulse has the characteristics of the Dirac Delta function. The characteristics of the Dirac Delta function and the Heaviside Step function, which is the integral of the Dirac Delta function, are presented in Eqs. 3.24 & 3.25, respectively.

$$\delta(t - t_0) = \begin{cases} 1 & t = t_0 \\ 0 & t \neq t_0 \end{cases} \quad \int_{-\infty}^{\infty} \delta(t - t_0) = 1 \quad (3.24)$$

$$\mathcal{U}(t - t_0) = \begin{cases} 1 & t \geq t_0 \\ 0 & t < t_0 \end{cases} \quad \int_{-\infty}^{\infty} \mathcal{U}(t - t_0) = 1 \quad (3.25)$$

The tracer compound is usually an inert chemical species which does not react throughout the reactor, and all changes in the concentration of the tracer are due to convective motion, that is either diffusion or bulk fluid movement. A special reactor case in the context of RTD is the plug flow reactor (PFR), which has no diffusive effects (recall Tab. 3.1). The result is that the spike/pulse of tracer fed to the reactor will propagate, in an unaffected manner, through the reactor at the pace of the bulk fluid. Consequently, the E-distribution will turn out to be a function with Dirac Delta character, while the F-distribution will have the characteristics of a step function (like e.g. the Heaviside step function). All the tracer fed to the reactor will, in other words, exit the reactor at the exact same time. This situation is illustrated in Figs. 3.3a & 3.3b, respectively.

It is emphasized that the RTD of the PFR is an extreme case which in reality is hard to reproduce, because it assumes absolutely no diffusive effects. In real applications, the fed tracer will leave the reactor at different times due to diffusion in the reactor. Many real reactors are well represented by the PFR model, however, because the diffusive effects (in comparison with the advective effects) are negligible. The more prominent the diffusive effects are, the wider the bell-shape of the E-distribution will be. The bell shape of the distribution curve is elaborated in Sec. 3.3, where it is illustrated in Fig. 3.6. In essence, the bell-shaped distribution governs

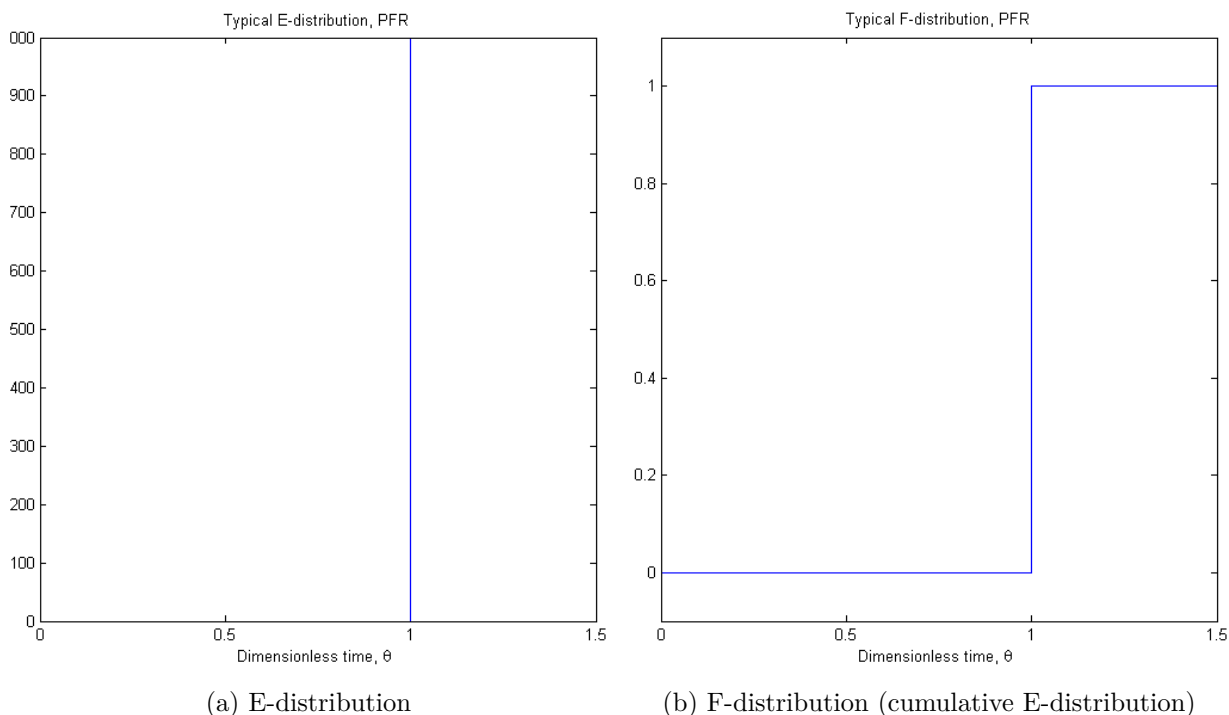


Figure 3.3: Illustration of typical RTD-behavior for a PFR.

how far away from the average residence time, and at what corresponding probability, tracer compound can be expected to exit the reactor.

Another approximate way of considering the RTD is in the context of moving control volumes with moving boundaries. This strategy for modeling a tubular reactor is discussed in Sec. 3.5. In essence, this situation represents small, but finite, volumes with the properties of a batch reactor, as introduced and discussed in App. C, propagating through the reactor. Such a volume will be well-mixed along the extent of the small (but finite) volume, and the residence time distribution will be modified accordingly. This situation is shown in Figs. 3.4a & 3.4b, which show the E-distribution and F-distribution in such a case, respectively. The larger the extent of the volume element is, the wider the block function (Fig. 3.4a) will be. Consequently, this yields a ramp function (Fig. 3.4b) which is less steep than what would be the case if the volume elements were smaller. This way of considering the RTD is sometimes referred to as the segregated flow model in the literature. [24]

3.3 Solution using the numerical method of lines

This section presents a method to reorganize a distributed system by using discretization, which is an important step in the right direction towards solving the model equations for the tubular reactor. The most common approach is to discretize the spatial coordinates, while leaving time to be a continuous variable. The effect is that the model equations are reduced from a handful of PDEs to a fairly large system (all depending on the resolution of the discretization) of ODEs in time, with each PDE being represented by a corresponding system of ODEs. The

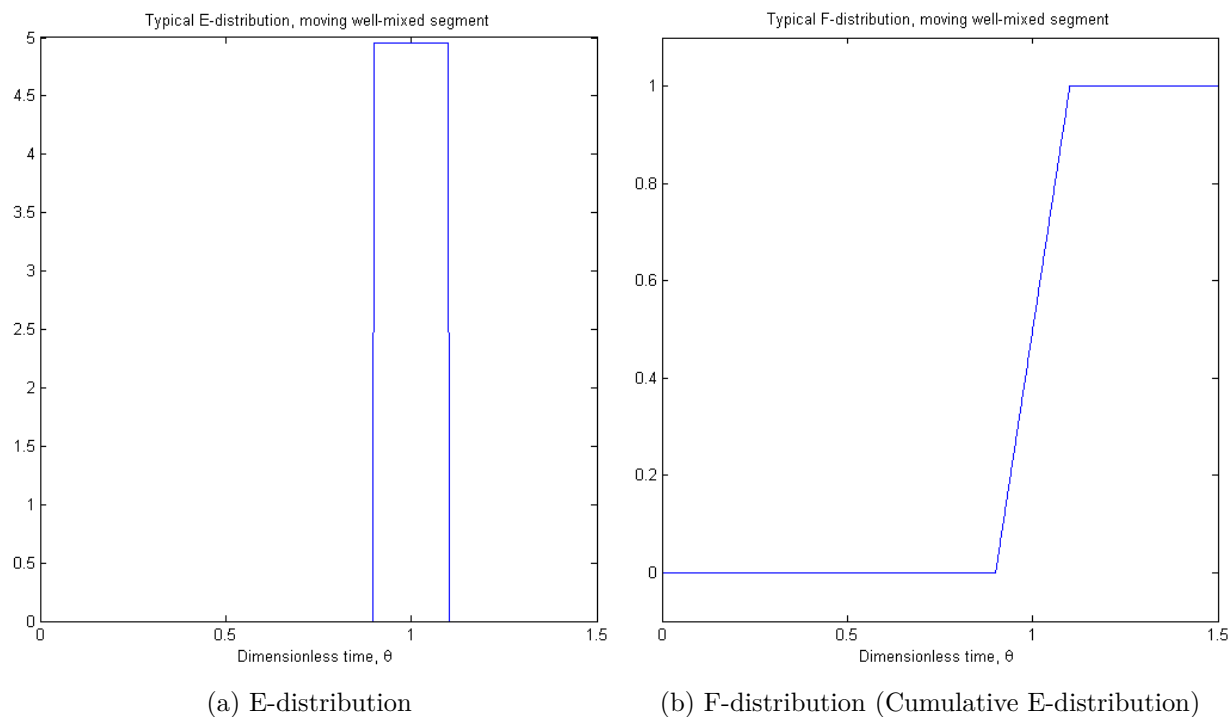


Figure 3.4: Illustration of typical RTD-behavior for a dispersed model.

discretization is based on finite differences, which is a traditional approach towards solving PDEs numerically⁷. [27]

Throughout this thesis, this solution strategy is being referred to as the numerical method of lines (NMOL). A simplified example on using NMOL, which is not related to tubular copolymerization reactors, is provided in App. E, in which a dynamic heat diffusion equation is solved with respect to time and one spatial coordinate. For this example, which has no specific relation to the modeling of a tubular reactor, the purpose is to demonstrate how this rather traditional method of spatial discretization can be deployed with satisfying results.

3.3.1 Solution strategy

The equation of interest is still the axial dispersion model, derived from the general balance principle (Eq. 3.7), which has been discussed in the previous sections. This equation is reprinted in Eq. 3.26, where the assumption of no radial gradients is already applied.

$$\frac{\partial \psi}{\partial t} + u_z \frac{\partial \psi}{\partial z} - D_e \frac{\partial^2 \psi}{\partial z^2} = \tilde{\sigma} \quad (3.26)$$

The key concept of the numerical method of lines is the ability to adequately approximate the various (spatial) derivatives of the system using finite differences. In this context, $f(x)$ is a

⁷The development of the NMOL type reactor model has been inspired by the modeling contributions to the COOPOL project provided by R. Pokorny & A. Zubov at VSCHT (Institute of Chemical Technology), Prague. Their help and guidance is gratefully acknowledged.

general function while x denotes an arbitrary independent variable which does not necessarily denote a dimension in space. Eqs. 3.27 & 3.28 are referred to as the forward and backward Taylor⁸ series approximation, respectively, to the function $f(x)$ with a perturbation step length of h . [23] [28]

$$f(x+h) = f(x) + h \frac{\partial f}{\partial x}(x) + \frac{h^2}{2} \frac{\partial^2 f}{\partial x^2}(x) + \dots = \sum_{n=0}^{\infty} \frac{h^n}{n!} \frac{\partial^n f}{\partial x^n}(x) \quad (3.27)$$

$$f(x-h) = f(x) - h \frac{\partial f}{\partial x}(x) + \frac{h^2}{2} \frac{\partial^2 f}{\partial x^2}(x) + \dots = \sum_{n=0}^{\infty} (-1)^n \frac{h^n}{n!} \frac{\partial^n f}{\partial x^n}(x) \quad (3.28)$$

A common application of Taylor series is the approximation of derivatives, often in the context of the linearization of nonlinear functions. In such situations, the Taylor series polynomials are often truncated at specific points to yield the specific approximations that satisfy the needs of the modeler. For the purpose of this work, only first and second order derivatives are of interest to obtain, and hence only the first three terms of each Taylor series polynomial are considered, as indicated in Eqs. 3.29 & 3.30. Here, $\mathcal{O}(h^3)$ represents the approximation error associated with discarding all terms with the order of 3 or higher. It is emphasized that the strategy of Taylor series truncation in finite differences yields approximations with error, but this error proves to be sufficiently small in many cases, justifying the strategy. This approach is also valid for achieving higher order derivatives, by using Taylor series polynomials with higher order and different combination patterns between the polynomials.

$$f(x \pm h) = f(x) \pm h \frac{\partial f}{\partial x}(x) + \frac{h^2}{2} \frac{\partial^2 f}{\partial x^2}(x) + \mathcal{O}(h^3) \quad (3.29)$$

$$\implies f(x \pm h) \approx f(x) \pm h \frac{\partial f}{\partial x}(x) + \frac{h^2}{2} \frac{\partial^2 f}{\partial x^2}(x) \quad (3.30)$$

By deploying these relationships, approximations for both first and second order derivatives are readily achieved. The first order derivative, given in Eq. 3.31, is the result of a simple algebraic shuffle of the forward approximation polynomial truncated after the second term. To achieve the second order derivative, which is presented in Eq. 3.32, the forward and the backward approximation polynomials (Eq. 3.30), both truncated after the third term, are combined and solved with respect to the second order derivative.

$$\frac{\partial f}{\partial x} \approx \frac{f(x+h) - f(x)}{h} \quad (3.31)$$

$$\frac{\partial^2 f}{\partial x^2} \approx \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} \quad (3.32)$$

The effects of the discretization become evident when Eqs. 3.31 & 3.32 are introduced into the balance principle equation of interest (Eq. 3.26). The result is shown in Eq. 3.33, which is the time-continuous ODE describing the change in the quantity ψ at the spatially discrete point k . In this expression, Δz denotes the spatial separation between the points in the discrete grid. The original PDE will thus be represented by n separate ODEs, depending on the discretization, each representing a discrete point in space. This strategy is applied to each quantity of interest

⁸The concept of Taylor series is attributed to the English mathematician Brook Taylor (1685 - 1731).

in the reactor, such as the respective species masses, the energy and the radical species of the reactor.

$$\frac{\partial \psi_k}{\partial t} + u_z \frac{\psi_k - \psi_{k-1}}{\Delta z} - D_e \frac{\psi_{k+1} - 2\psi_k + \psi_{k-1}}{(\Delta z)^2} = \tilde{\sigma}_k \quad (3.33)$$

When the number of discrete points in space (n) increases, the discrete system approaches the behavior of the distributed and spatially continuous system. It is worth mentioning that while a high amount of discrete points in space is believed to yield an accurate model, it will probably cause a large demand for computational effort, which is undesirable in the event of deploying the model in an on-line application.

In practice, the discrete system can also be interpreted as a series of well-mixed units (tanks). This idea is known as the tank-in-series model, which is a common way of approaching a distributed system. This way of considering the system reinforces the ideas from Sec. 2.2, where issues regarding the phase distribution of the emulsion system could be treated by assuming consecutive segments of the reactor being separately well-mixed. Fig. 3.5 shows a reactor tube in which the spatial coordinate is discretized into smaller elements of finite extent. The tube still has the properties of radial and angular symmetry and well-mixed conditions in all spatial coordinates but the axial. The behavior of the tubular reactor under this tanks-in-series consideration is explored further in the context of residence time distribution for the model arising from the NMOL approach.

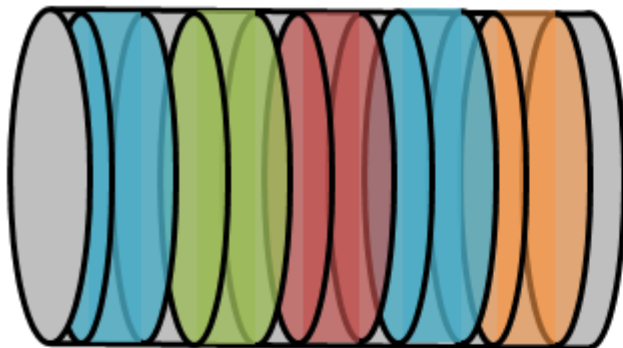


Figure 3.5: A drawing of a reactor tube which is divided into a series of discrete segments having finite extent.

3.3.2 Residence time distribution in the NMOL approach

Each discrete point in the distributed system can be considered to be a small (but finite) volume element behaving like a continuously stirred tank reactor (CSTR). This idea is illustrated in Fig. 3.5. In such a case, the normalized concentration of tracer (c_t) at the reactor outlet at a specific point in time is given by Eq. 3.37. In this expression n is the amount of equal imaginary tanks used to model the distributed system, while τ is the mean residence time of each tank. This expression originates from a recursive derivation using a convolution approach to connect the tanks together. For one CSTR, the normalized outlet concentration is given by Eq. 3.34. By consecutively attaching an increasing amount of small imaginary CSTRs in a step-wise manner,

Eq. 3.37 is achieved for n CSTRs. [30]

$$n = 1 : \quad c_{t,1}(t) = \frac{1}{\tau} e^{-\frac{t}{\tau}} \quad (3.34)$$

$$n = 2 : \quad c_{t,2}(t) = c_{t,1} * c_{t,1} = \int_0^t \left(\frac{1}{\tau}\right)^2 e^{-\frac{\xi}{\tau}} e^{-\frac{t-\xi}{\tau}} d\xi = \left(\frac{1}{\tau}\right)^2 e^{-\frac{t}{\tau}} t \quad (3.35)$$

$$n = 3 : \quad c_{t,3}(t) = c_{t,2} * c_{t,1} = \int_0^t \left(\frac{1}{\tau}\right)^3 e^{-\frac{\xi}{\tau}} \xi e^{-\frac{t-\xi}{\tau}} d\xi = \left(\frac{1}{\tau}\right)^2 e^{-\frac{t}{\tau}} \frac{t^2}{2} \quad (3.36)$$

⋮

$$\implies \quad c_{t,n}(t) = c_{t,n-1} * c_{t,1} = \int_0^t \left(\frac{1}{\tau}\right)^n e^{-\frac{\xi}{\tau}} \xi^{n-2} e^{-\frac{t-\xi}{\tau}} d\xi = \left(\frac{1}{\tau}\right)^n e^{-\frac{t}{\tau}} \frac{t^{n-1}}{(n-1)!} \quad (3.37)$$

The RTD for such a reactor model is represented in Fig. 3.6. Experimental data for an RTD experiment⁹ is also included to identify to what extent the model is expected to represent reality. The experiment is represented by the dotted line in Fig. 3.6, while the solid lines represent theoretical distributions with various degrees of discretization. For the NMOL approach, a large number of discretization points is required to achieve satisfying agreement between the model and the experiments, which is expected to cause problems with respect to the numerical efficiency of the model. Generally speaking, it is desirable to achieve an accurate model with as few states as possible. For a discrete spatial model obtained with the NMOL approach, the amount of states will increase linearly with the amount of discretization points, which means that there is a direct correlation between the spacing in the numerical discretization and the demand for computational effort for the model. If the amount of discretization points (n) is too high, the model may prove to be inappropriate for the purposes of on-line optimization and control. This issue is at the very heart of the thesis, and will be investigated extensively in the subsequent sections.

3.4 Solution using an incremental formulation with change of variables

This section explores a different strategy towards modeling the axially dispersed reactor model. Through introducing transformed variables, the model equations are modified to yield slightly different forms, allowing for alternative methods with respect to the solution of the problem. The residence time distribution of the reactor, in particular, is achieved in an elegant manner using an appropriate transformation of the variables in the problem.

The general equation of interest in this context, which motivates the transformation of the variables, is the axial dispersion model, as presented in Eq. 3.38 for a generalized quantity (ψ). Some assumptions have been introduced to arrive at this equation, as discussed in Sec. 3.1, but all terms of interest are maintained. Both diffusive effects and changes due to bulk movement are included, as well as the term for net generation of the quantity.

$$\frac{\partial \psi}{\partial t} + u_z \frac{\partial \psi}{\partial z} - D_e \frac{\partial^2 \psi}{\partial z^2} = \tilde{\sigma} \quad (3.38)$$

⁹The RTD experiments were conducted at The Technical University of Hamburg (TU Hamburg), Germany, at the Department of Macromolecular Chemistry. The work of F. Lüth is acknowledged.

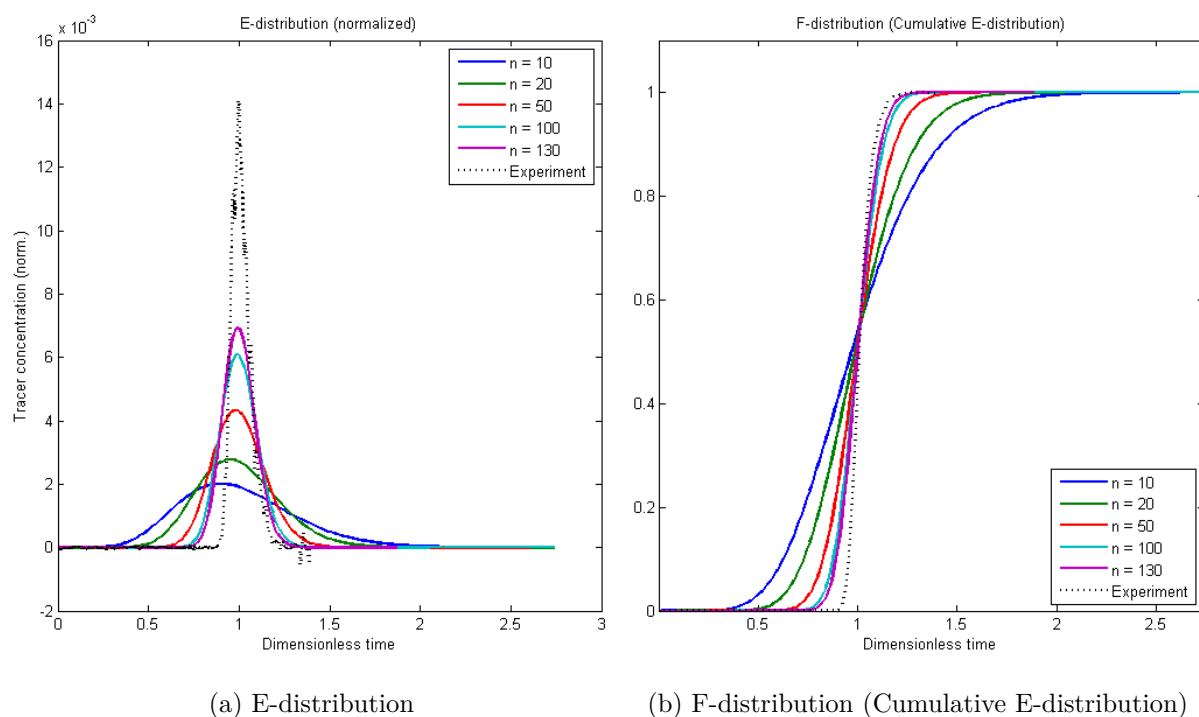


Figure 3.6: Illustration of typical RTD-behavior for a tanks-in-series model.

3.4.1 Model transformation (transformation of variables)

The first step towards formulating the model equations in a different set of coordinates is, obviously, to decide a set of coordinates which are believed to reduce the equation to a simplified form. This may not be trivial, indeed, and may require a certain degree of creativity. The transformation of variables is a technique which is fairly widespread¹⁰ in the algebraic solution of PDEs, but systematic and universal approaches are, in most cases, hard to deploy. For this specific work, the variable transformations in Eqs. 3.39 & 3.40 are chosen, carrying along a formal change in the functions of each of the quantities of interest in the system, as indicated in Eq. 3.41.

$$t^*(t, z) = t \quad (3.39)$$

$$z^*(t, z) = z - vt \quad (3.40)$$

$$\implies \psi(t, z) \rightarrow \psi(t^*, z^*) \quad (3.41)$$

The derivatives of interest for the balance equations in this work are the first order derivatives with respect to time as well as both first and second order derivatives with respect to the spatial coordinate. By deploying the variable transformations indicated in Eqs. 3.39 - 3.41, the derivatives are changed accordingly using the chain rule of differentiation, as shown in Eqs. 3.42 -

¹⁰Although this approach is described as fairly widespread, it is emphasized that the numerical method of lines, i.e. the method of spatial discretization, is far more common. Throughout this thesis, the NMOL approach is typically considered to be the traditional approach.

3.44.

$$\frac{\partial \psi}{\partial t} = \frac{\partial \psi}{\partial t^*} \frac{\partial t^*}{\partial t} + \frac{\partial \psi}{\partial z^*} \frac{\partial z^*}{\partial t} = \frac{\partial \psi}{\partial t^*} - v \frac{\partial \psi}{\partial z^*} \quad (3.42)$$

$$\frac{\partial \psi}{\partial z} = \frac{\partial \psi}{\partial t^*} \frac{\partial t^*}{\partial z} + \frac{\partial \psi}{\partial z^*} \frac{\partial z^*}{\partial z} = \frac{\partial \psi}{\partial z^*} \quad (3.43)$$

$$\implies \frac{\partial^2 \psi}{\partial z^2} = \frac{\partial^2 \psi}{\partial (z^*)^2} \quad (3.44)$$

When these transformations for the derivatives are introduced, Eq. 3.38 is effectively reduced to yield Eq. 3.45. Emphasis is kept on not forgetting that the variables have been transformed, and all the information of the problem is maintained through the transformation. The velocity of the fluid flow, for instance, is not lost just because it does not appear explicitly in the equation, but is accounted for by the respective coordinates.

$$\frac{\partial \psi}{\partial t^*} - D_e \frac{\partial^2 \psi}{\partial (z^*)^2} = \tilde{\sigma} \quad (3.45)$$

3.4.2 Analytical residence time distribution for a tracer compound

For an RTD experiment using an inert tracer compound, the diffusion equation is further reduced to a simpler form due to the fact that there is no net generation/consumption of tracer in the reactor, i.e. $\tilde{\sigma}$ is zero. This situation is illustrated in Eq. 3.46, where c_t denotes the concentration of tracer.

$$\frac{\partial c_t}{\partial t^*} - D_t \frac{\partial^2 c_t}{\partial (z^*)^2} = 0 \quad (3.46)$$

An equation like the one in Eq. 3.46 is a typical homogeneous PDE with the property of being separable. As a consequence, the possibility of obtaining analytical solutions is present, but the boundary conditions of the problem are expected to complicate this.

The first step towards finding an analytical solution to Eq. 3.46 is to proceed with the separation technique for PDEs. The solution method for a problem like this one is inspired by Kreyszig [23], but the strategy is widespread to the extent where it should be obtainable from most textbooks on engineering mathematics. The key assumption is that the tracer concentration solution function is a product of two contributions from the two respective independent variables, like indicated in Eq. 3.47.

$$c_t(t^*, z^*) = T(t^*)Z(z^*) \quad (3.47)$$

By introducing this change, Eq. 3.46 readily becomes as shown in Eq. 3.49, where the dot denotes differentiation with respect to time (i.e. t^*) while the apostrophe denotes differentiation with respect to the spatial coordinate (i.e. z^*).

$$Z(z^*)\dot{T}(t^*) = D_t T(t^*)Z''(z^*) \quad (3.48)$$

$$\implies \frac{\dot{T}(t^*)}{D_t T(t^*)} = \frac{Z''(z^*)}{Z(z^*)} \quad (3.49)$$

By examining this expression, it becomes clear that while left hand side only depends on t^* , the right hand side only depends on z^* , implying that each side is a constant, denoted by $-\lambda$ for

convenience. This gives rise to the system of ODEs which is provided in Eqs. 3.50 & 3.51, which have their respective general solutions as indicated in Eqs. 3.52 & 3.53. By combining the two general solutions in agreement with Eq. 3.47, the general solution for the tracer concentration is obtained, which is presented in Eq. 3.54. In this solution, A , B , C , k_1 , k_2 and λ are all undetermined coefficients which will be determined for the special solution of the problem.

$$\dot{T}(t^*) = -\lambda D_t T(t^*) \quad (3.50)$$

$$Z''(z^*) = -\lambda Z(z^*) \quad (3.51)$$

$$T(t^*) = A e^{-\lambda D_t t^*} \quad (3.52)$$

$$Z(z^*) = k_1 \cos(\sqrt{\lambda} z^*) + k_2 \sin(\sqrt{\lambda} z^*) \quad (3.53)$$

$$\Rightarrow c(t^*, z^*) = e^{-\lambda D_t t^*} \left(B \cos(\sqrt{\lambda} z^*) + C \sin(\sqrt{\lambda} z^*) \right) \quad (3.54)$$

The next step towards finding the solution is to deploy the appropriate boundary conditions of the problem. For the transformed variable system, the boundary conditions become as indicated in Eqs. 3.55 - 3.57. At the very beginning, the tracer concentration resembles a pulse fed to the reactor, represented by the Dirac Delta function ($\delta(z^*)$), while the change in tracer concentration is zero at the origin of the spatial coordinate. These boundary conditions may, at first glance, appear strange in comparison to the boundary conditions given in Eqs. 3.19 - 3.21, but the transformation of the variables must be kept in mind.

$$t^* = 0 : \quad c_t(0, z^*) = \delta(z^*) \quad (3.55)$$

$$t^* > 0 : \quad \left. \frac{\partial c_t}{\partial z^*} \right|_{z^*=0} = 0 \quad (3.56)$$

$$c_t(t^*, L) = 0 \quad (3.57)$$

The differential equation must obey the boundary conditions, and hence the boundary conditions help to determine the unknown coefficients of the general solution (i.e. B and C). The details of this calculation is omitted for simplicity, and the resulting Fourier series solution to the problem is shown in Eq. 3.58. For this solution, the pre-factor of the arguments (λ) becomes as indicated in Eq. 3.59.

$$c_t(t^*, z^*) = \sum_{n=1,3,5,\dots}^{\infty} \left\{ \frac{1}{2L} \cos(\sqrt{\lambda_n} z^*) e^{-\lambda_n D_t t^*} \right\} \quad (3.58)$$

$$\lambda_n = \left(\frac{n\pi}{2L} \right)^2 \quad (3.59)$$

The resulting residence time distribution for this approach to the model is presented in Fig. 3.7. The effective mass diffusivity for the model has been adjusted by trial-and-error to agree with the experimental RTD-behavior. In this consideration, the effective mass diffusivity is the sole parameter governing the internal mixing effects of the reactor, when both the fluid flow velocity and the reactor geometry are given. The assumption of no radial mass transport effects is emphasized, as this is one of the most important assumptions to achieve a simplified problem with respect to the spatial distribution.

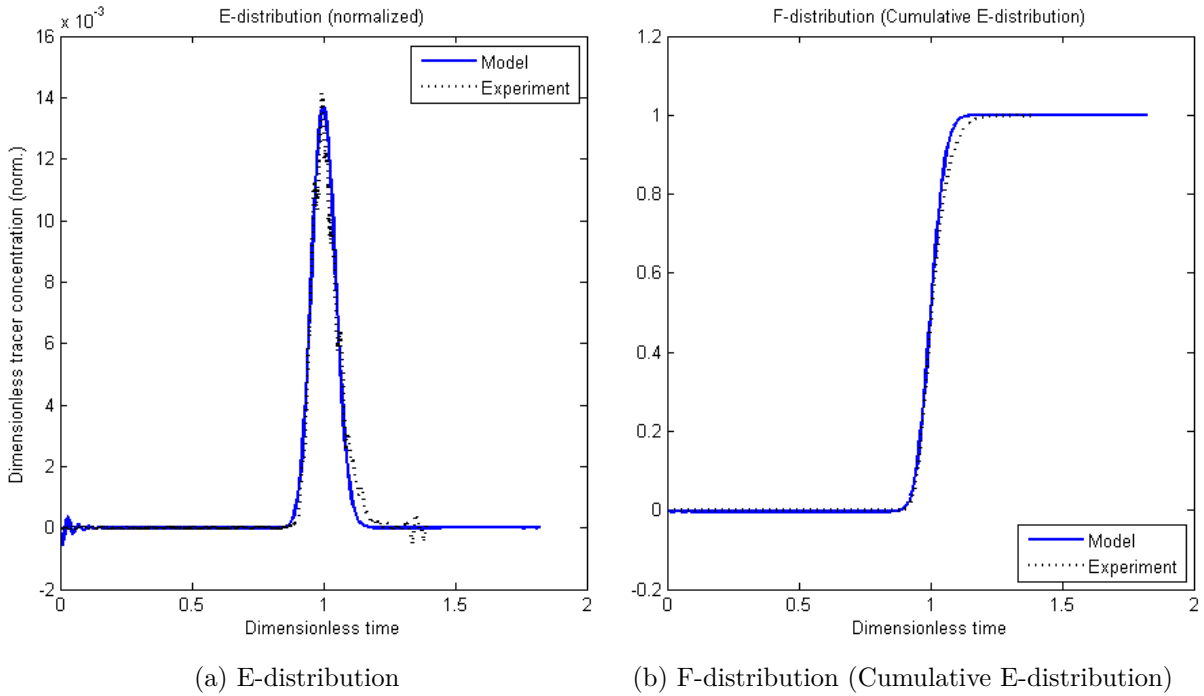


Figure 3.7: Illustration of RTD-behavior for an incremental model with transformed variables.

3.4.3 Model equations in transformed variables

When the species mass balances of the tubular reactor is considered, the concentration of the respective species is the quantity of interest ($\psi = c_i$). Since the chemical species in the reactor are participating in chemical reactions, the net generation term in the balance equation will no longer be equal to zero. Unlike what was the case for the residence time distribution, for which an analytical solution was obtainable, the equations are no longer separable, which leaves them more complicated to solve. A special case arises when the axial diffusivity is negligible, under which the axial balance equation (Eq. 3.13) is reduced to a particularly simple form, as shown in Eq. 3.60. In the same spirit, the temperature explicit energy balance is reduced from Eq. 3.18 to yield Eq. 3.61. It must be stressed that neglecting the diffusive effects, i.e. setting $D_i = 0$, may imply a violation of the mixing effects in the reactor.

$$\frac{\partial c_i}{\partial t^*} = \tilde{R}_i \quad (3.60)$$

$$\frac{\partial T}{\partial t^*} = \frac{1}{\rho C_p} (-\Delta H_{rx} + q) \quad (3.61)$$

Accidentally, these results are analogous to the species mass balances and energy balance for a simple, traditional batch type reactor, as described in App. C. In this comparison, the transformed variables are still kept in mind to ensure agreement between the modified model and the physical interpretation of the system. The transformation of the coordinates did not yield separable PDEs for the model equations, which leads to believe that a type of discretization is required anyways. Because of this, the strategy of transforming the coordinates may not consti-

tute a significant advantage over the traditional NMOL approach after all. The mathematical similarity between the transformed model and the batch model has, however, motivated a reactor model formulation using small (but finite) units with moving boundaries, which propagate through the reactor, to model the distributed system. This strategy is elaborated in Sec. 3.5.

3.5 Solution using a formulation with mobile control volumes

Another way to model the distributed system is essentially a hybrid between the two respective strategies of discretization in space and change of variables. In the NMOL approach (Sec. 3.3), the system was divided into a finite amount of parts having a small (but finite) extent, essentially establishing a series of (tiny) stationary tanks. In the other approach, where the variables were transformed (Sec. 3.5), the spatial domain was shifted linearly with time, yielding a moving spatial domain. In the final solution method, which is presented here and deployed in simulations later, these two strategies are combined to yield a modeling approach with moving control volumes. The ideas arising from this train of thought are throughout the thesis referred to as the moving control volume (MCV) approach towards modeling the distributed system. This may also be referred to as the segregated flow model, which is a term used in the literature when the residence time distributions for tubular reactors are considered. [24]

For each volume element propagating through the reactor, the individual behavior will be like that of a typical batch reactor. Like indicated in Eqs. 3.60 & 3.61 in Sec. 3.4, the model equations reinforce the idea of separate control volumes each acting like traditional batch reactors. The model equations for a batch reactor are derived and discussed in App. C. In this situation, the position of each moving element becomes a state in line with the fluid flow velocity, and the spatial extent of each element is important to keep track of. The effects of axial mixing due to diffusion and advection are both handled in this case, since each volume element is perfectly mixed within its finite boundaries while it propagates through the reactor. The mass transport between separate volume elements are excluded from the approach, however, giving rise to a possible source of error in the model. The belief is that this will be accounted for by the movement of the moving reactor elements.

Since, as mentioned, the (imaginary) reactor units move through the reactor, their respective positions become states of the system, as shown in Eq. 3.62. In this case, z_i denotes the spatial position of element no. i , with a velocity of v_i . In order to construct the model, the moving elements must be permitted to move beyond the spatial reactor outlet, and the properties of interest at the reactor outlet are calculated by interpolation between the two utmost moving elements. In this strategy, the state equations of the reactor are only active inside the reactor, and although the elements are allowed to continue moving, the other states are not changed. Mathematically, this is presented in Eq. 3.63.

$$\frac{dz_i}{dt} = v_i \quad (3.62)$$

$$\frac{dx}{dt} = \begin{cases} f(x, u, \theta, t) & \text{if } z \leq L \\ 0 & \text{else.} \end{cases} \quad (3.63)$$

The implementation of these concepts are straight-forward in the context of modeling the reactor, but when the model is being used in on-line estimation algorithms the states are perturbed. In order to ensure safe calculations, the model is smoothed to yield states which are twice

continuously differentiable. A way to achieve this is by multiplying each state equation by a corresponding smoothing term, typically consisting of a trigonometric function, which quickly (but smoothly) reduces the effect of the state equation to zero at a specific point. A suggestion for such a formulation is shown in Eq. 3.64, where the \arctan ¹¹ function is deployed.

$$\frac{dx}{dt} = f(x, u, \theta, t) \cdot \underbrace{\left(\frac{\pi}{2} - \arctan \left(\frac{z-L}{a} \right) \right)}_{\text{Smoothing term}} \cdot \frac{1}{\pi} \quad (3.64)$$

The quantity denoted by a is a constant referred to as the smoothing constant, which governs how narrow the window of change will be. The effect of the smoothing term is illustrated in Fig. 3.8. The smoothing term changes dramatically around $\frac{z}{L} = 1$, as intended, as this is the

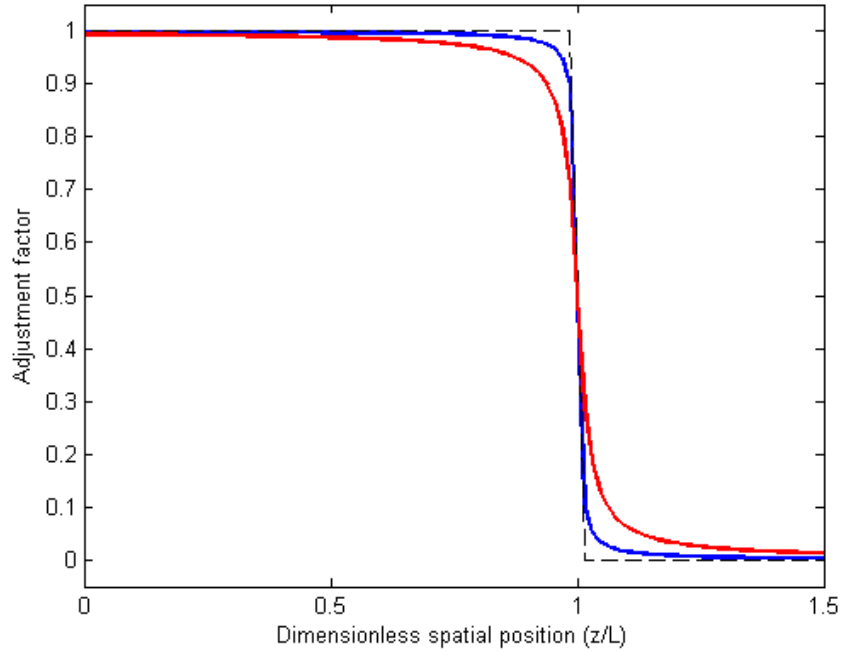


Figure 3.8: Illustration of the effect of a smoothing factor for the differential equations at the (spatial) reactor outlet.

axial length of the reactor and hence presents the reactor outlet. Fig. 3.8 shows three separate alternatives for the smoothing. The thin, dashed, black line shows the strict change from 1 to 0, while the red and blue curves show the behavior governed by Eq. 3.64, where a is four times larger for the red curve than for the blue curve.

The RTD associated with this approach is illustrated in Fig. 3.9. This is in agreement with the idea of a well-mixed volume element propagating through the reactor, which was initially

¹¹ $\text{Arctan}(x)$ is the inverse tangent function, also denoted by $\tan^{-1}(x)$, not to be confused with $\frac{1}{\tan(x)}$. The expression inside the parenthesis in Eq. 3.63, i.e. $\frac{\pi}{2} - \arctan \left(\frac{z-L}{a} \right)$, could also be denoted by $\text{arccot} \left(\frac{z-L}{a} \right)$, which is the corresponding inverse cotangent function. [23]

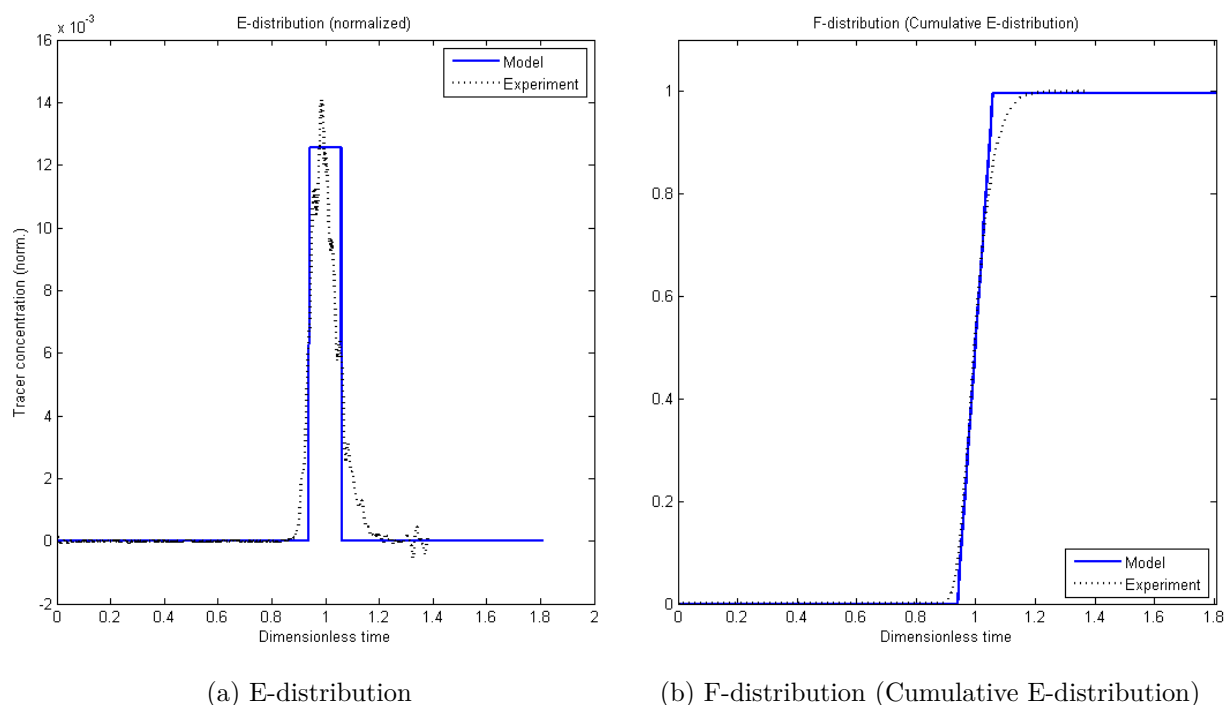


Figure 3.9: RTD-behavior for the modeling approach with well-mixed moving control volumes.

introduced in Sec. 3.2. Conceptually speaking, this is the same behavior as the one introduced in Fig. 3.4 in Sec. 3.2, for both the E-distribution and the F-distribution, and the experimental data from the previous RTDs has been included as well, represented by the thin dotted line. The purpose of comparing the Heaviside type (Eq. 3.25) E-distribution to the experimental data is to decide the required width of the moving control volumes. The larger the allowed width of the volume elements is, the lower the required amount of volume elements becomes, yielding a lower total amount of states. The Heaviside block character of the E-distribution does not agree particularly well with the serrated bell-curve of the experimental data, especially when the excellent fit of the analytical solution in Fig. 3.7a is recalled. In this case, however, where the allowed width of the volume element is adjusted in a trial-and-error manner, the fit becomes bearable. The F-distribution (Fig. 3.9b) also presents with certain deviations between the model and the experiments, but the general agreement is satisfying. A concluding remark is that with relatively few moving elements, i.e. a relatively coarse spatial discretization, a satisfying agreement between theory and experiments is achievable.

3.6 Comparison of the various approaches

Throughout this section, two separate strategies for treating a distributed system, i.e. solving partial differential equations, have been considered, also resulting in an intermediate suggestion for a space-discrete model with moving control volumes. In the context of on-line optimization and control, the numerical efficiency of the model is the most important property when it comes to deciding which model to implement, but the ability of the model to encapture the mixing effects of the reactor and ensure an agreement with reality must not be forgotten. This is the

quintessential trade-off for models which are desired to be deployed in on-line applications.

For the purpose of summarizing the preceding sections, a comparison between the models is given in Tab. 3.3. As a remark, the NMOL approach is found to be the safest approach, being the most traditional one, while the moving control volumes (MCV) model is believed to be the model with the most potential with respect to achieving an efficient control model while maintaining accuracy. It has been shown that the NMOL type reactor model needs a very fine discretization in space to yield a satisfying agreement with the RTD experiments, but it is believed that this modeling approach can provide valuable solutions, for instance for temperature control, using far less points of discretization than the considerations with respect to RTD indicate. This will be investigated in Sec. 7.

When constructing discrete schemes in space, the concept of numerical diffusion is an inescapable subject to address. Numerical diffusion is, in brief, the discrepancy between the real diffusive nature of a system and the diffusive behavior predicted by the corresponding computer simulation [25]. This concept becomes particularly evident for spatial discretization using finite differences. The discretization scheme which is developed for the NMOL approach in Sec. 3.3 has a low order, in the sense that the spatial derivatives are only functions of a very limited number of adjacent points. Standard forward schemes have been used, but more sophisticated ways of obtaining a discrete grid is possible, for instance using a higher order in the finite differences approach, or using differently directed schemes, such as for instance backward schemes, with more points. Generally speaking, a high order method will yield less numerical diffusion than a low order method will, because the derivatives will be calculated more accurately. Another way to counteract numerical diffusion is to deploy another method of spatial discretization, e.g. finite volume methods, numerical collocation methods or spectral methods. These methods are generally known to have a higher order than the standard finite differences methods. These alternative methods have not been considered in depth throughout this thesis, but they remain interesting candidates for further investigation if this work should be extended, or if the models are being revised thoroughly. If an alternative way of discretizing the spatial problem provides more accurate derivatives without contributing significantly to the demand for computational effort, deploying it will definitely constitute a step in the right direction with respect to developing efficient models for on-line optimization and control.

An appropriate strategy in the context of simulating an on-line control system is to use the numerically most demanding model as a plant replacement model to perform plant calculations and simulate real-life behavior, while the numerically most efficient model is deployed for the purposes of on-line optimization and control. This separate use of different models is elaborated in Sec. 4.3, where the NMPC software (Cybernetica CENIT) is introduced, and in Sec. 6.4, where the concept of plant representation models is discussed. The desire for models which are numerically efficient is also elaborated and emphasized in Secs. 5 & 6, where the concepts of on-line estimation and NMPC are discussed, respectively.

Table 3.3: Comparison between the different process models considered in the thesis.

Model type	Concluding remarks
The numerical method of lines	<p>This strategy towards solving PDEs is a safe and relatively straightforward method of performing spatial discretization. The method is, generally speaking, both valid and effective. A weakness of the NMOL approach for the specific system is the need for a large amount of discretization points to yield a satisfying accuracy in the model. This results in a large amount of states and consequently a large demand for computational effort. The need for a large amount of discretization points was evident from the comparison between the experimental RTD and the RTD predicted by the model. The question remains, however, how high the accuracy of the model needs to be in order to provide an efficient model for the purpose of controlling the process in mind. The trade-off between numerical efficiency and model accuracy remains an inescapable topic.</p>
Incremental model (transformation of variables)	<p>The transformation of variables strategy is an effective way to reduce the balance equations for the tubular reactor case. In the context of RTD, the reduction is particularly effective, yielding a separable PDE. This equation has an analytical solution which agrees particularly well with experimental data for RTD when the diffusivity is adjusted accordingly. For the species mass balances for reacting species and energy balance, however, this is not the case, and a numerical solution is needed. Because of this, the incremental model approach is, by itself, not significantly better than the NMOL approach. The main conclusion to be drawn from this modeling approach, apart from the details regarding RTD, is the additional reduction achieved under certain assumptions, i.e. negligible diffusive effects, which yields the moving control volume approach.</p>
Mobile control volume model	<p>The mobile control volume approach is inspired by the transformation of variables approach, in the sense that having a collection of moving volume elements with finite extent will represent the tubular reactor in a satisfying way. This model presents with a relatively low demand for computational effort, and the results from the simulations appear sensible and promising. Properties of interest inside the reactor are calculated by interpolation between the moving elements of the reactor, and the moving elements are reinitialized at the spatial beginning of the reactor once they have exited the spatial reactor outlet.</p>

Section 4

Software used in the work

The purpose of this section is to provide an overview of the various software that have been utilized in the work. In Sec. 4.1, the programming language Modelica is introduced along with the Dymola tool, which has been used to formulate the reactor models introduced in the thesis. Secs. 4.2 & 4.3 present the in-house Cybernetica software tools Modelfit and CENIT, respectively, which are used for model verification, estimator tuning and on-line control.

In order to demonstrate the behavior of the software, a super-simplified process model has been developed to constitute an example of how a process can be modeled in Modelica and implemented in the Cybernetica CENIT software. This model and an example simulation for an on-line control case is provided in App. G.

4.1 Introduction to Modelica & Dymola

Modelica is a programming language for object-oriented programming which was first released in 1997. Since then, the language has been developed to treat a range of applications within different fields, and although Modelica may not be the most abundant programming language with respect to world-wide use, the applications are numerous, for instance in the automotive industry. Modelica also provides effective tools and packages to model multi-phase flow systems in a systematic manner. In being object-oriented, Modelica has similarities to classic programming languages like *C++* or Java, but it also has differences, and Modelica is often referred to as a *modeling language* rather than a *programming language*. Among the strengths are easy declaration and treatment of variables, as well as strong performance with respect to numerical efficiency and computational time. An example of a Modelica script is presented in Cs. G.1 in App. G, in which the dynamic model for a super-simplified buffer tank is implemented as an example.

The Modelica language is excellent for solving DAE-systems¹. The mathematical representation of such a problem is formulated in Eqs. 4.1 & 4.2. Eq. 4.3 shows how measurements from the process are *predicted* from the state of the system, which will be valuable when the model predictions are compared to the actual measurements from a real process in the context of on-line estimation (Sec. 5).

$$\dot{x} = f(x, u, \theta, t) \quad (4.1)$$

$$0 = g(x, u, \theta, t) \quad (4.2)$$

$$y_p = h(x, u, \theta, t) \quad (4.3)$$

In this formulation, x is the vector of states for the system, y_p is the vector of predicted system outputs corresponding to a vector of measurements from the real system (y_m), u is the vector of inputs to the system, and θ is the vector of parameters for the system. The functions f , g and h indicate arbitrary functions for the system, governing the dynamic nature of the system.

¹DAEs: A system of differential and algebraic equations, in contrast to ODE-system, which only contains ordinary differential equations. In this thesis, the most typically encountered type of system are the so-called semi-explicit DAEs.

Dymola² is a proprietary software for implementation of Modelica code. In that sense, Dymola supports both editing, compilation and simulation (i.e. numerical integration etc.) of Modelica code. When solving problems like the one in Eqs. 4.1 & 4.2, Dymola deploys powerful algorithms which account for stiffness problems (discussed briefly in Sec. 2.3), etc. This provides the modeler with an efficient way to construct process models, even for highly nonlinear and computational demanding processes. In addition to this, Dymola is able to produce simulation plots and data files for use in other applications, which makes multi-platform treating of process models and corresponding data very effective. A particularly valuable feature of the Modelica/Dymola combination, which is elaborated in Sec. 4.2, is the possibility of exporting the whole process model component, and not only the simulation results, to other applications using the so-called Model Exchange interface. This allows for constructing process models using the flexibility and powerful environment of the Modelica language, which in turn can be deployed in another compatible software. Good examples of this are the Cybernetica Modelfit and CENIT software, which are both compatible with the Model Exchange interface, allowing for the use of models built in Modelica in on-line control implementations.

A special treat of the Dymola software is the opportunity to use a graphical user interface (GUI), which enables the user to perform graphical drag-and-drop modeling of processes, given the desired components/subunits of the process³. This promotes the use and development of modeling libraries for various types of processes, and represents a large potential with respect to reusability and modifiability of models, which can prove to be a huge advantage to the modeler. With these possibilities, Dymola can be used as a dynamic process simulator, like Aspen Plus or Aspen HYSYS, with the option of manipulating each unit⁴ as desired using the Modelica language. This facilitates a large degree of customization in the modeling work. Using a modular and hierarchical approach to modeling also enables the modeler to interchange and/or modify certain parts of the model without compromising the respective other components or the overall structure of the model. This strategy is illustrated in Fig. 4.1. Figs. 4.1a & 4.1b show two different approaches for modeling a specific system, whereas Fig. 4.1b is the modular approach. The two approaches yield the same overall model, and the simulation results are expected to be similar, but the advantage of the modular approach is, as already mentioned, the possibility of easily exchanging certain parts/components without compromising the total structure of the model. Imagine, for instance, a reactor model where it is of interest to change the cooling mechanism. In a modular approach, changing the cooling jacket model would not include tempering with the energy balance of the reactor itself.

4.2 Cybernetica Modelfit

Cybernetica Modelfit is a software developed by Cybernetica AS for off-line state and parameter estimation. The Modelfit software is designed to work alongside the software for nonlinear model-

²Dymola: **D**ynamic **M**odeling **L**aboratory, a software developed by Dassault Systemes AB, Sweden. Non-proprietary software for Modelica also exist, for instance the OpenModelica software.

³The approach of designing subunits which are combined to yield the complete process model, rather than creating one single large model script, is referred to as the modular approach. This philosophy in the modeling work has been prominent throughout the entire work of this thesis.

⁴One of the main drawbacks of Modelica/Dymola in comparison with the mentioned commercial process simulator tools is the fact that apart from the fields of electrical circuit modeling, mechanical modeling and fluid flow modeling, the standard libraries of Dymola are scarce. Because of this, the modeler is sometimes required to make components "from scratch" or adopt components made by a third-party contributor.

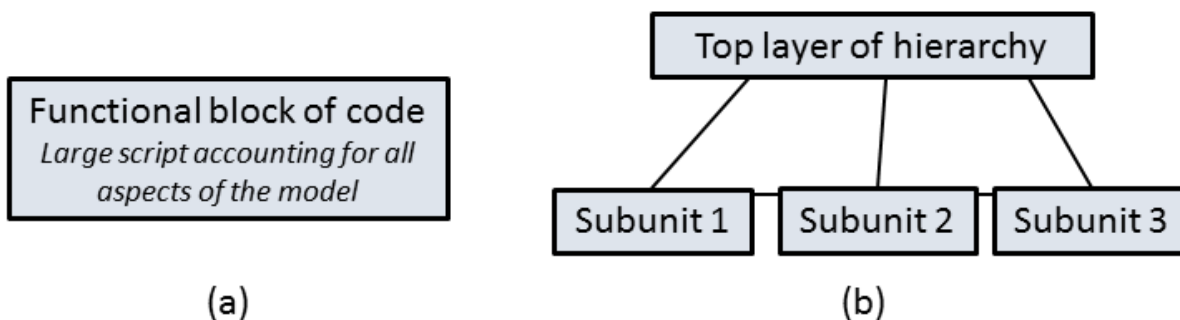


Figure 4.1: A figure showing the general idea of interconnecting Modelica units in a hierarchy.

based predictive control (NMPC), known as Cybernetica CENIT which is introduced in Sec. 4.3. Modelfit is, however, in itself a process model simulator tool which can be used to run ballistic model simulations and perform off-line parameter fitting. This provides a valuable tool for off-line treatment of a developed model before on-line implementation. Off-line parameter estimation is a key activity in order to ensure agreement between a first principles state space model and experimental data from the real process, which will be elaborated for the tubular reactor case in Sec. 7.2. Before a model is deployed in on-line use, it must be tested in an off-line environment, and Modelfit provides a way to tune the on-line estimator, i.e. establish the noises of the states, parameters and measurements of the system. Some concepts of on-line estimation of states and parameters are discussed in Sec. 5, and the performance of the estimator is demonstrated for a tubular reactor case in Sec. 7.3.

To utilize the Modelfit software, the traditional approach is to formulate the specific process model in the programming language C, as this provides a suitable basis for importing the model into the Modelfit software. Modelfit is, in other words, programmed to recognize the various features of the model from the formulation of the model script. For models formulated in other environments than C, a technique⁵ for so-called model exchange exists, which is referred to as the Functional Mock-up Interface (FMI). The FMI is a tool to establish a platform which enables the communication between models and applications initially formulated in different languages. One of the supported languages in the FMI is Modelica, and this motivates the use of Modelica for these kinds of applications, in agreement with the introduction given in Sec. 4.1. Modelica models can, in other words, easily be exported to the Modelfit software, and the Modelfit software will even recognize which variables are inputs and which variables are outputs, given that the Modelica model is formulated correctly. The FMI translation tool, which is relatively easy to administer, is implemented in the Dymola software.

Given an imported model, e.g. using the model exchange strategy as introduced above, Modelfit can be used to test models, and perform off-line parameter fitting using given experimental measurements. Modelfit has built-in routines for treating DAEs, where the user can choose among several alternatives. Simple numerical Euler-integration is available, where the user is inclined to set the step length of the method, and a mid-point Euler method is also implemented. The most

⁵The FMI is developed as a part of the MODELISAR European project, which had the German automotive company Daimler AG among its main initiators and contributors. During the work of this thesis, a new version of the interface, FMI 2.0, is being developed, which is believed to have less bugs than the former version.

sophisticated tool available to the user for solving the system equations is the CVODEs tool⁶. The choice of solution method depends on the complexity of the system, regarding nonlinearity, stiffness, etc. It is emphasized that custom solvers could also be implemented by the modeler. The trade-off between computational time and accuracy/convergence of the solution may also play a part in deciding the solution method. For on-line implementations, this consideration is of key importance, and the CVODEs solver is believed to be a good choice in this context.

In an off-line parameter estimation procedure, where the purpose is to calculate the optimal parameter values for a model to fit experimental measurements, the Modelfit software solves a minimization problem with respect to a chosen set of parameters, in which the sum of squares for the deviations between the model predictions and the experimental measurements is desired to be as low as possible. This is formulated mathematically in Eq. 4.4.

$$\min_{\eta} \sum_{k=1}^{n_{ky}} (y_{p,k} - y_{m,k})^2 \quad (4.4)$$

$$\eta \in \theta$$

In this formulation, $y_{p,k}$ and $y_{m,k}$ are model predictions and measurements, respectively, for the sample point k . The number of valid measurements is represented by n_{ky} , while η is the selection of parameters among all the parameters (θ) which are chosen for optimization.

For setting up and solving these kinds of optimization problems, the Modelfit software has several additional features. The software is constructed to provide a user-friendly way to decide what parameters to estimate, i.e. to decide η from θ . If the user also wishes to estimate initial values for the states of the system, these variables are accessed and activated for estimation in the same simple manner as for the parameters. This may be particularly interesting in cases where the initial states are fictional in the modeling work, but need to be estimated once the performance of the model is evaluated in comparison with experimental data. In addition to this, Modelfit provides valuable supporting calculations during the course of the optimization procedure, making it easier for the user to evaluate the quality of the solution. One of these calculated quantities is the scaled Hessian condition number, for which an example⁷ is illustrated in Fig. 4.2. Generally speaking, the condition number indicates how the output value of a function will respond to a change in the input value(s). For the case of parameter fitting, the function in mind is the cost function as introduced in Eq. 4.4. A high value for the condition number for the *scaled* Hessian matrix is usually an indication that some of the parameters in the set has low individual sensitivities towards the cost function. Another possible cause of a high value for the condition number is linear dependence between some of the optimization variables. A way of resolving this could be to re-evaluate the combination of optimization variables (η) to yield a smoother iteration procedure. Another quantity calculated for the respective optimization variables is the identifiability ranking, for which an example is shown in Fig. 4.3. The identifiability ranking can be interpreted in somewhat of the same way as the condition number for the Hessian matrix, in the sense that it indicates the sensitivity of the respective optimization variables towards the cost function. In this consideration, however, the variables are ranked, and the user is able to better

⁶CVODEs is a tool developed under the SUNDIALS (Suite of Nonlinear and Differential/Algebraic equation Solvers) project at the Center for Applied Scientific Computing of the Lawrence Livermore National Laboratory, designed for numerical solution of stiff DAEs. [13]

⁷This example originates from a batch simulation of an emulsion copolymerization system. This simulation has no particular relevance to the cases simulated in Sec. 7.

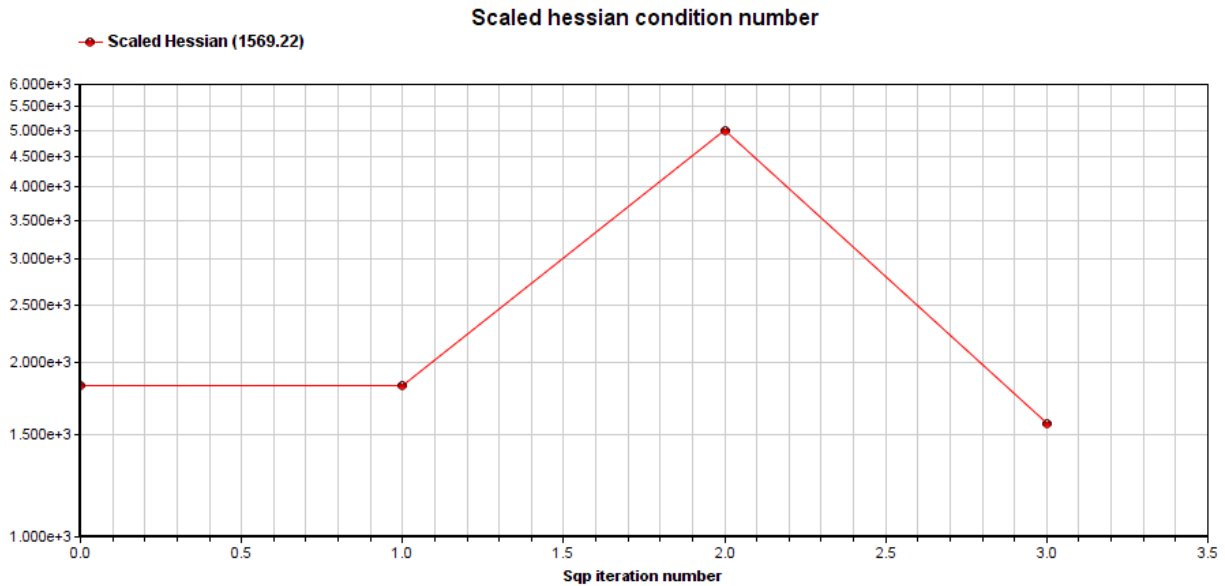


Figure 4.2: Changes in the condition number for the scaled Hessian matrix in the model fitting calculation.

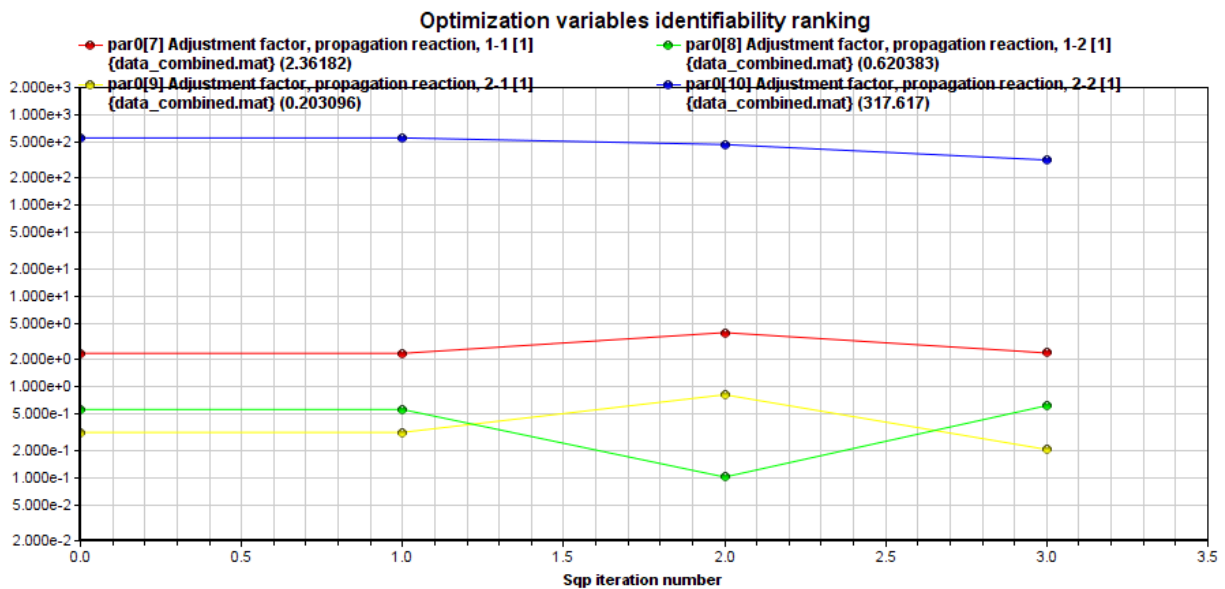


Figure 4.3: Changes in the identifiability ranking for the variables during model fitting calculation.

decide which variables to dismiss when re-choosing variables for optimization. The desire of the modeler is usually to achieve a significant effect by making slight changes in the parameters, which will require a high identifiability ranking. A low identifiability ranking usually implies a low sensitivity to the cost function for that specific variable or a strong linear dependence to one of the other variables. In other words, the variables with the lowest identifiability rankings are the most sensible candidates for being dismissed from the optimization procedure. [39]

4.3 Cybernetica CENIT

The Cybernetica CENIT software is the utility used for on-line control in the thesis. It provides a complete solution for NMPC with appropriate estimator algorithms. Details on these concepts are given in Secs. 6 & 5, respectively. The CENIT software is a bundle of various components interacting to yield an effective controller. The interconnection of the units of the CENIT application is illustrated in Fig. 4.4. In a real implementation, the CENIT Kernel is installed on a

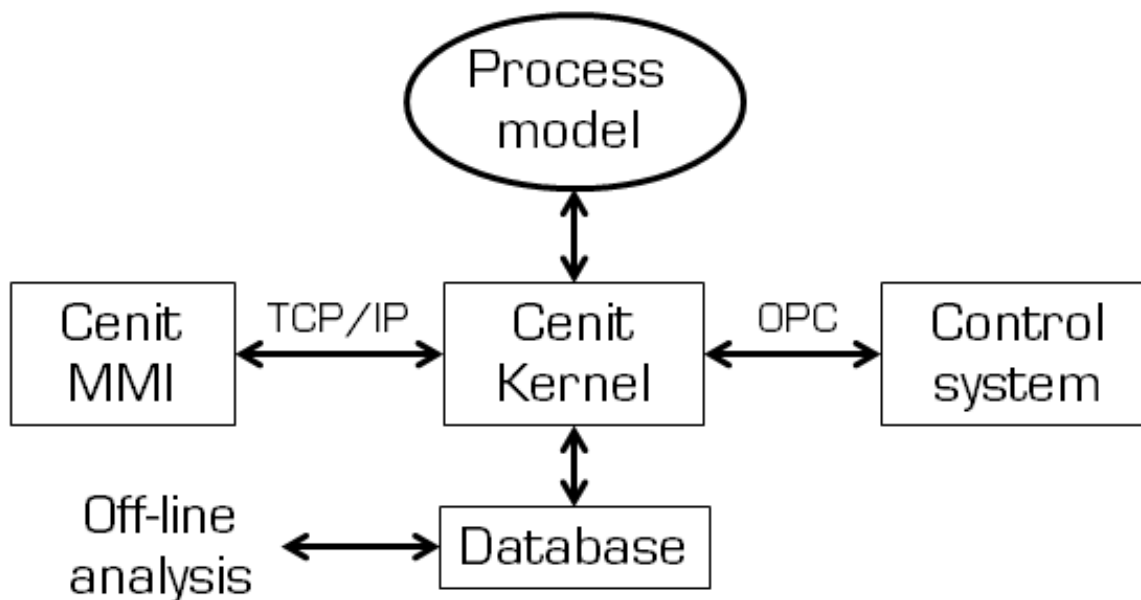


Figure 4.4: A block diagram showing the interconnection of the various components of the Cybernetica CENIT software.

computer at the plant, which in turn communicates with the low-level control system of the plant through the OPC⁸ standard. The MPC can be used to control inputs to the process directly, or it can be used to set the setpoints for other controllers at a lower level in the hierarchy. This distinction is introduced in Fig. 6.1, which is discussed briefly in Sec. 6.1. The CENIT Kernel is closely connected to the process model, which is the component which is specific for each application. The model governs the calculations for the estimator and the controller which are

⁸The OPC platform is a double abbreviation denoting Object linking and embedding for Process Control or Open Platform Communication.

conducted in the CENIT Kernel. In addition, the CENIT Kernel communicates with the (optional) database component which stores data from the operation, allowing for off-line analysis and trending of parameters. This can prove to be valuable, both with respect to the maintenance of the controller application as well as detection of bottle necks in the plant, etc. The CENIT MMI (Man-Machine Interface) is an interface to govern the CENIT Kernel, and can be administered from any computer through TCP/IP, given the right authentication. This is also a valuable feature with respect to maintenance, as engineers can control the control system without visiting the control room of the specific plant. In typical test simulations for a fictional implementation, which has been done in this thesis, all the components of the application are usually run on the same computer. In such a case, the various protocols and platforms connecting the components may seem excessive, but they are crucial for the success of a real-life implementation such a control system. A brief summary, explaining each of the components, is given in Tab. 4.1.

An important addition to the CENIT software is the Cybernetica RealSim software. In a real implementation, the RealSim software will log the behavior of the real plant. In a fictional implementation however, like the ones considered in this thesis, the RealSim software will simulate the actual behavior of the real system. The RealSim software will, in other words, return the response, according to the assigned process model, to the inputs suggested by the controller. For the purpose of this thesis, this introduces the concept of using different process models for different tasks. The computationally least demanding model is suited for on-line optimization and control due to relative low computational costs in optimization problems. This model may prove, however, to not be as accurate as more demanding models. The solution is to use the computationally more demanding model with better accuracy as a plant replacement, i.e. to simulate the plant behavior in correspondence to the controller action. It is emphasized that the model which is implemented in the controller preferably should be the fastest and computationally least demanding. Using two separate models in this manner introduces the effect of incompleteness of the model in the controller, which is important to obtain to mimic a real-life situation. There is, in other words, a certain deviation between the behavior predicted by the model in the controller calculations and the behavior of the real (in this case "real") system. If the exact same models were used for both control and plant simulations, there would exist an exact relationship between the model and the "plant", which is unachievable in reality. These ideas are elaborated in Sec. 6.4, where the concept of deploying a separate plant replacement model for the simulations is explained.

Table 4.1: Main components of the Cybernetica CENIT application.

Component	Description
CENIT Kernel	This is the heart of the CENIT application. It interconnects the model and control system with algorithms for estimation and control. The CENIT Kernel usually operates from a computer at the specific plant which is being controlled, and it can be controlled through the CENIT MMI.
CENIT MMI	This is the engineering interface used to govern the CENIT Kernel. This enables the engineer to configure the controller and extract valuable information. Among the most important features of the CENIT MMI are tuning of both the estimator and the controller, as well as trending and monitoring of the various quantities of interest for the system, i.e. states and measurements.
Process model	This is the application-specific part of the CENIT application. This component is usually a first principles state space model, i.e. the mathematical formulation, for the process model, and hence it governs the dynamic behavior of the system. Generally speaking, the model component can be formulated in any programming language, as long as it communicates with the CENIT Kernel properly.
Database	The purpose of the database is to store historical data from the process for trending variables of interest in the process and performing off-line analysis for the operation.
Control system	This is the process control system, which handles the low-level communication with the process. It must not, in other words, be confused with the actual MPC controller. This component is, strictly speaking, not a part of the CENIT application, but rather a part of the process. The interconnection between the real process and the controller is handled by this system. This component decides, for instance, whether the MPC controller will control the inputs directly or indirectly. This concept is shown in Fig. 6.1.

Section 5

On-line state and parameter estimation

This section is written to include an introduction to the concept of on-line estimation of states and parameters, which is a crucial part of an MPC implementation for a system. Several approaches to optimal state estimation are available, among which the Kalman filter estimator and the Moving Horizon Estimator (MHE) are possibly the most celebrated and commonly deployed approaches. For the purpose of this thesis, a Kalman filter type estimator has been appointed for the controller implementation. While different algorithms for on-line estimation indeed have their respective strengths and weaknesses, making the choice of estimator dependent on the characteristics of the studied process, alternative estimator algorithms to the Kalman filter type estimator have not been explored in the work of this thesis.

A detailed derivation of the Kalman filter estimator algorithms for several types of systems is given in App. F. The starting point is a linear time-discrete system, but the systems of interest in the context of this thesis are nonlinear time-continuous systems, and the estimator is extended to cover this as well. It is emphasized that while this entire section is aimed at providing an introduction to on-line estimation, and contains the conceptual strategy of the estimation, it does not encapture the complete detailed nature of the rather sophisticated estimation algorithms of the Cybernetica software (introduced in Sec. 4).

5.1 Purpose and formulation of the estimator

The quintessential purpose of the estimator is to establish a link between the measured system behavior from the plant and the model-based calculations in the controller. The important blocks of an on-line estimator implementation are shown¹ in Fig. 5.1, where the respective components are interconnected to illustrate the flow of information in the on-line implementation of the specific system. This part of the system, as will be elaborated in Sec. 6, is the core of the entire block diagram for the complete controller implementation (Fig. 6.2). The Kalman filter estimator works to neutralize the effects arising from the fact that the process model, on which the controller action depends, is incomplete. A process model can, as previously stated (Sec. 1, [5]), never be complete in an absolute sense, and the need for an estimator will always be present in an on-line controller implementation in the real world. A model of high-end quality will generally require less dramatic intervention from the estimator than a less adequate model will, as the high-end controller model will be better able to predict the behavior of the system on its own. The success of the estimator is always important, in any case. In theoretical studies such as this thesis, where computer simulations are constructed to mimic the plant behavior, the possibility to have a 1-to-1 exact agreement between the controller model and the plant behavior is present. In order to demonstrate process control, this may be a fruitful starting point, but this will make the estimator unnecessary. In order to recreate realistic situations, a deviation must be established between the controller model and the plant. This deviation is, as duly mentioned, inescapable in real-life applications, and a way to introduce such a deviation in computer simulations is to assign different values for certain corresponding parameters in the two models. Another strategy

¹This figure is printed with the permission of S. O. Hauger, Cybernetica AS.

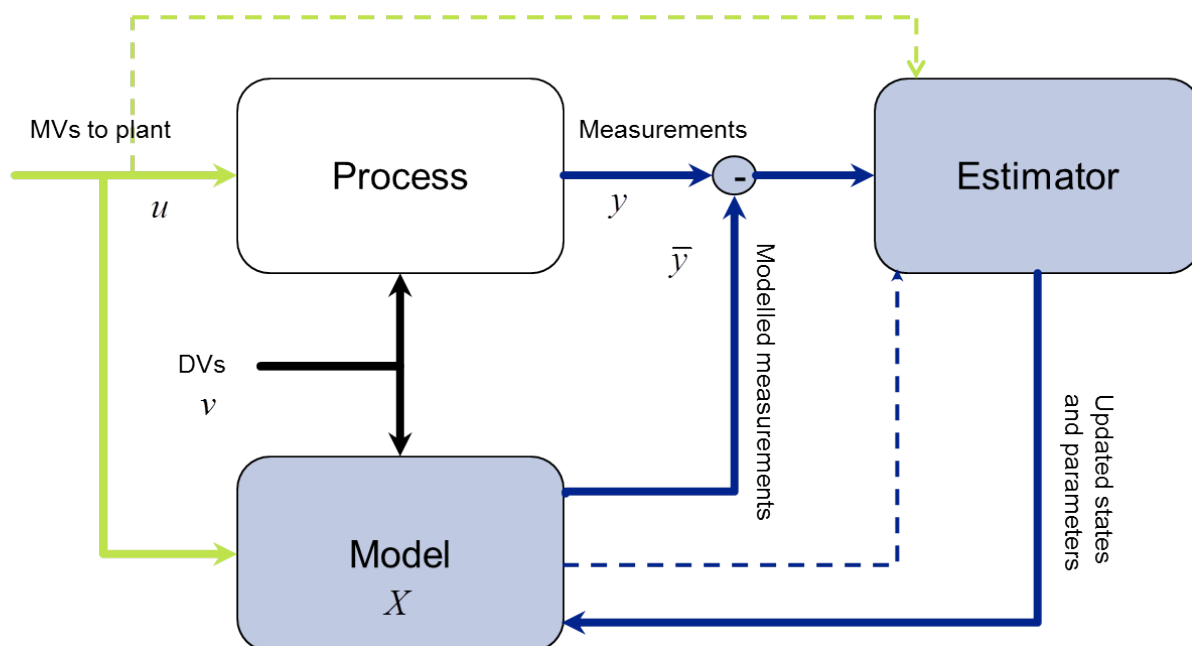


Figure 5.1: A block diagram showing the interconnected components in an on-line estimator.

is to deploy two different types of models in the controller and the plant, respectively, where the behavior of the two differ even with identical parameters. These ideas are elaborated further under the concept known as plant replacement, which is discussed in Sec. 6.4.

The formulation for the state estimates are provided in Eqs. 5.1 - 5.5. These equations hold for the so-called Hybrid Extended Kalman filter (HEKF) estimator, in which a continuous process model is used with time-discrete measurements. This result is obtained from the derivation in App. F. The specific estimation algorithms deployed in the Cybernetica software tools (introduced in Sec. 4) use a Divided Difference approach, which is closely related to the Hybrid and Unscented Kalman Filter approaches. The important thing to recognize is that while the integration of the model between each sample in time² is continuous, the algorithm has a discretization feature where it jumps from one sample to the next, only deploying estimator updates at each sample point in time. This is a sensible way to operate the system, given that the time sampling interval agrees with the rate at which the plant can conduct measurements. Temperature measurements, for instance, can typically be collected rapidly, while measurements of fluid composition etc. are harder and slower to obtain.

$$\text{Between } t_k^+ \text{ and } t_{k+1}^-: \quad \dot{\hat{x}} = f(\hat{x}, u, w_*, t) \quad (5.1)$$

$$\text{(between samples)} \quad \dot{P} = AP + PA^T + \tilde{Q} \quad (5.2)$$

²Time samples (and sampling time interval) in the controller algorithm is a concept which is elaborated in the discussion of the MPC controller in Sec. 6.

$$\text{Between } t_k^- \text{ and } t_k^+: \quad K_k = P_k^- H_k^T (H_k P_k^- H_k^T + \tilde{R}_k)^{-1} \quad (5.3)$$

$$\text{(at each sample)} \quad \hat{x}_k^+ = \hat{x}_k^- + K_k (y_k - h_k(\hat{x}_k^-, v_*, t_k)) \quad (5.4)$$

$$P_k^+ = (I - K_k H_k) P_k^- (I - K_k H_k)^T + K_k \tilde{R}_k K_k^T \quad (5.5)$$

In these equations, \hat{x} denotes state estimates while y_k denotes measurements at sample number k . Superscripted minus and plus signs indicate *a priori* and *a posteriori* estimates, respectively. \tilde{Q} and \tilde{R} are the process noise and measurement noise matrices, respectively, K_k is the Kalman filter gain at sample number k . H_k and h_k represent the part of the model which predicts the outputs (specifically the measurements).

The importance of state estimation is once again emphasized, as this is the number one purpose of the estimator. In addition, the possibility of performing on-line parameter estimation is an interesting feature, especially in the context of on-line control using first principles models. The reason for this is that the parameters of the model are each known to a specific degrees of accuracy, which may vary a lot between the various parameters. Some parameters are known with high certainty, while others are unknown and may even be weakly time-variant for the system. Undoubtedly, off-line parameter estimation prior to an on-line implementation, where the model is tuned to fit experimental data, is crucial for the success of the entire implementation, but the time-variant behavior of certain parameters can be accounted for by the estimator. In this strategy, a so-called *augmented* state vector (denoted by x') is constructed, which contains both original states and parameters of choice. From the entire collection of parameters for the system (θ), a set of uncertain parameters are chosen (η) for which some specific parameters (λ) are subject to on-line parameter estimation³. The reformulations are, mathematically speaking, as indicated in Eqs. 5.6 & 5.7. The augmented state vector is then valid for the same estimator algorithms as the original state vector.

$$x' = \begin{bmatrix} x \\ \lambda \end{bmatrix} \quad (5.6)$$

$$\dot{x}' = f'(x', u, \theta, w, w_\lambda) \quad (5.7)$$

The estimation of state variables in an on-line implementation, in addition to state estimation, has an extremely important purpose in addition to modifying slightly varying parameters during the operation. By including noise on parameters, integral action is achieved in the controller. The reason for this, is that the estimator then manages to predict offset free outputs. This property is very valuable with respect to the performance of the controller. In an MPC implementation, integral action is a necessity just like for more simple control setups. In a PI-controller, for instance, the integral action represents an improvement of the P-controller, giving offset free control. When noise for the respective process parameters is included, two different approaches are usually deployed. In one case, shown in Eq. 5.8, the noise is included in an additive manner to the respective parameters. The other case, which is presented in Eq. 5.9, uses a relative

³A prime example of a class of parameters which typically have weakly time-varying behavior are heat transfer coefficients. Due to conditions in the reactor, fouling etc., this parameter may change slightly, and the controller system is believed to perform smoother if the estimator manages to encapture this change.

formulation where the noise is some fraction of the parameter value itself.

$$\tilde{\lambda}_{k-1} = \hat{\lambda}_{k-1} + \bar{v}_{k-1} \quad (5.8)$$

$$\tilde{\lambda}_{k-1} = \hat{\lambda}_{k-1} (1 + \bar{v}_{k-1}) \quad (5.9)$$

The Cybernetica software uses the latter of the two, i.e. Eq. 5.9, as this makes it slightly easier to impose the same relative noise on the parameters of choice, even though they may be in significantly different ranges. The model predictions (here represented in a time-discrete form) will, conversely, become as indicated in Eq. 5.10. In this expression, the separation between the non-varying parameters (θ), which may be the result of an off-line parameter estimation, and the time-varying parameters ($\tilde{\lambda}_{k-1}$), which are part of the on-line estimation problem, is emphasized. In the simulations in Sec. 7, both of these ways to handle the parameters of the system, i.e. off-line and on-line parameter estimation, are considered in turn.

$$\bar{x}_k = f(\hat{x}_{k-1}, \theta, \tilde{\lambda}_{k-1}, u_{k-1}) \quad (5.10)$$

In essence, a common way to achieve efficient on-line estimation is to carefully select a number of parameters for on-line modification, which is equal to the number of independent measurements the process (plant) provides.

An important argument for choosing to add noise through some of the parameters is related to the fact that the estimator perturbs the model in the calculations. The reason for perturbing the model is to discover the sensitivities of the independent variables in the calculations. In the Cybernetica software tools, which are introduced in Sec. 4, the typical amount of perturbations is $2n_s + 1$, where n_s is the number of states in the model. That is, one perturbation in each direction for each state and one unperturbed simulation. From this perspective, it is obvious that an increasing amount of states will yield an increased demand for computational effort. An important thing to realize is that the perturbations may lead to the violation of the balance principles on which the consistency of the model is dependent. To exemplify this, consider the mole balance of a chemical species reacting to another chemical species. In independently perturbing, through the process noise, the separate amounts of moles of the two species, the risk of violating the mass balance is present. This is not the case, however, if the strategy instead is to perturb the parameter(s) governing the reaction rate, i.e. the kinetic rate expression, for the two species. The same type of argument applies to heat (energy) balances, for instance, which should not be violated in the calculations, preferably.

Another important concept to consider, which is related to both perturbation within the estimator algorithm and the violation of balance principles, is the risk of the estimator breaking sensible limits when it performs state updates. This may cause convergence problems in the model and it may also lead to inconsistencies which can, in a worst case scenario, lead to the failure of the implementation. Among the typical inconsistency errors are negative volumes, masses or pressures, mass fractions or mole fractions which are above 1, etc. Mathematically speaking, this is no concern to the estimator itself, but to the consistency of the process model, it may be fatal. Depending on the robustness of the model, this can rapidly overturn the model and start producing model predictions with severe errors. In Fig. 5.2, three different scenarios are illustrated, in which the perturbations of the estimator suggest an infeasible solution. For simplicity, and for the purpose of illustration, the figures are drawn in two-dimensional space, but it is emphasized that the concepts apply to any state vector in n -space. The feasible area

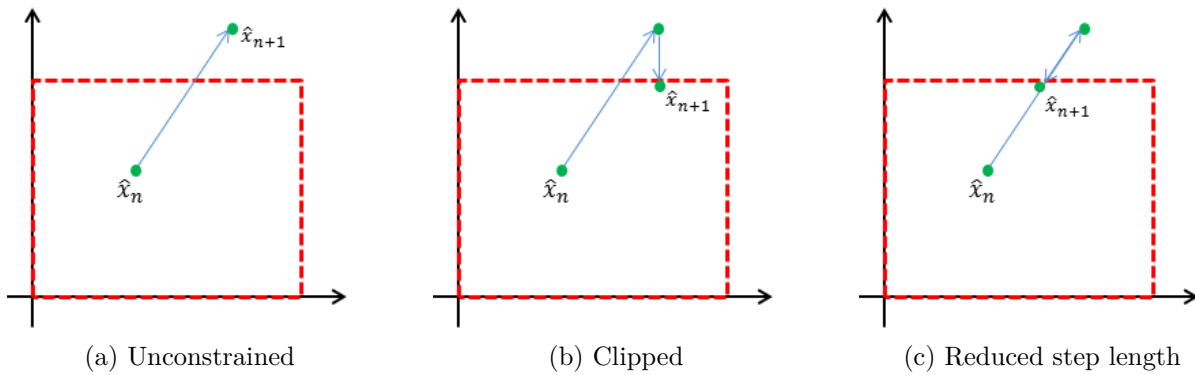


Figure 5.2: Illustrations of the various ways to constrain the updates in the Kalman filter type estimator algorithm in the Cybernetica CENIT software.

is indicated by a red, dashed square in each of the figures, while the green dots connected by blue arrows indicate the steps made in the state updating procedure. Fig. 5.2a shows the unconstrained case, in which the infeasibility of the updated solution is ignored. Throughout the work of this thesis, experience indicates that this is a very risky way to run the application, and estimator breakdown may occur. Fig. 5.2b, on the other hand, illustrates the so-called clipped way of updating the states. In this case, the estimator jumps from the infeasible solution which was first suggested to the closest feasible solution possible. In other words, it assumes the one feasible solution which is closest, in space, to the original infeasible solution. The last alternative is the so-called step length reduction method⁴, which is illustrated in Fig. 5.2c. The idea in this case is to step carefully, in the event of an infeasible state update, along the trajectory of the update (i.e. the blue arrow in Fig. 5.2) until the boundary of the feasible area is reached. It is worth mentioning that while both the clipped and the reduced step length methods yield feasible solutions, thus representing safer alternatives than the unconstrained method, they do not give the same solution. This is evident from Fig. 5.2, and one method may have advantage over the other depending on the specific system for which it is being deployed. It is important to recognize that, depending on the respective values for the previous solution (\hat{x}_n) and the infeasible solution, the difference between the final solution found by clipping and the final solution found by step length reduction may be quite significant. For the purpose of this thesis, the clipped method has been deployed the most, as this provides the one feasible solution which is "closest" to the unfeasible solution originally proposed by the estimator.

5.2 Estimator tuning

As indicated in Eqs. 5.1 - 5.5, the calculations of the estimation algorithm are dependent on both the states and the measurements of the system, but also the noise in the system. These established equations are based on the derivations of App. F. In order to tune the estimator, the

⁴Step length reduction is a common concept in many applications of numerical mathematics. The most intuitive way of stepping carefully in the right direction is to evaluate whether each step is feasible or not, and halve the step length if the step is infeasible, until a convergence criterion is achieved. Figuratively speaking, this is like stepping closer and closer to the edge of a cliff, while constantly evaluating how long the next step can be without falling off.

noise of both the measurements and the process states are set by the modeler. As discussed for on-line parameter estimation in the previous section, noise in parameters yields integral action, and this calls for tuning the estimator through deciding the noise for the respective parameters in addition to the states and the measurements.

To decide the noise of the states, parameters and measurements of the system is not at all a trivial task. As will be discussed for the tuning of the controller, in Sec. 6.3, experience is useful when designing implementations, and systematic approaches for tuning are not easily obtained. Some of the tuning parameters, i.e. the noise, of the estimator can be deduced, however. The measurement noise can usually be obtained from the instrumentation in the plant. Temperature measurements from thermometers, for instance, may have a known measurement error of 0.5 K . A concentration measurement using gas chromatography (GC), on the other hand, may have a known measurement error of 0.1 %. Determining state error is less trivial, and the model itself should be "consulted" to evaluate the certainty of the various states. The same argument applies for parameters which are expected to vary. The error could, for instance, be deduced by considering the corresponding parameter in similar systems.

The typical action of the estimator based on the noise of the respective system variables are summarized in Tab. 5.1. This behavior is shown for a specific series of reactor test simulations in Sec. 7.3. These considerations will also be highly important for the simulations in Secs. 7.6, 7.7 & 7.8.

Table 5.1: Estimator action to noise in states, parameters and measurements.

Noise variable	Estimator action
Noisy (uncertain) measurement	The estimator uses a weak update from the specific measurement. Since the measurement is uncertain, the estimator does not want to base the model updates largely on that specific measurement.
Certain measurement	The estimator deploys a strong model update from the specific measurement, since the measurement is considered to be relatively certain.
Noisy (uncertain) state	The estimator uses a strong update of the state, since the state is uncertain, and the state prediction made by the process model probably needs to be corrected significantly.
Certain state	The accuracy of this state prediction is probably good, and the need for a strong update from the estimator to this state is not crucial.
Noisy (uncertain) parameter	If the parameter is activated for on-line estimation, the parameter is included in the augmented state vector as shown in Eq. 5.6. Since the parameter has large uncertainty, the estimator will seek to modify this parameter to achieve a better fit between the model predictions and the plant behavior.
Certain parameter	The parameter can be considered to be constant throughout the operation of the plant, and is typically not included in the on-line parameter estimation problem.

Section 6

Principles of model-based predictive control

The purpose of this section is to provide an introduction to the concepts of model-based predictive control (MPC) in the context of mechanistic, first principles process models. Optimal state estimation was introduced in Sec. 5, based on the theoretical background provided in App. F, and will pose a crucial contribution to the complete MPC formulation. These theoretical considerations will be important for the simulations and results of the thesis, which are presented in Sec. 7.

The nature and introductory theory of MPC is presented in Sec. 6.1, while some of the features of MPC are elaborated in Sec. 6.2. Issues related to controller tuning are presented in Sec. 6.3, while the idea of deploying two separate process models for controller calculations and plant representation, respectively, is elaborated in Sec. 6.4. The explored theory on MPC, which is presented in this section, is inspired by Rawlings & Mayne as well as Hauger, Schei and Dyrset. [21] [32] [40]

The scope of this thesis is aimed at the individual modeling and treatment of a specific unit in a plant, i.e. tubular reactors, and considerations regarding plant-wide control structures are sparse with respect to details. The main scope is to achieve efficient control of specific reactors in an isolated point of view, and no concerns have been made with respect to the other process units of a typical polymerization plant. The cooling fluid, for instance, is assumed to be readily available in a reservoir for this work, at the desired temperature, while a more complete approach would require a model to describe the treatment of cooling fluid. Both the flexibility and applicability of MPC in a plant-wide context remain emphasized, however. As will be indicated later, the strengths of the MPC approach are even clearer when the complexity of the system, and the number of variables, increases.

6.1 Controller objective and formulation

Fig. 6.1 shows two versions of a typical control hierarchy for a process plant. The distinction between the two suggested control structures is inspired by Skogestad [36]. Here, the dotted regions are the respective control layers. Generally speaking, MPC can be used to manipulate inputs (valves etc.) directly, like indicated in Fig. 6.1b. A setup where the MPC governs the behavior of a series of simpler control loops, however, like the one shown in Fig. 6.1a, is more common. As mentioned in Fig. 4.4 in Sec. 4.3, the control system component is dependent on the control layer structure of the plant. One could imagine, for instance, the MPC controller taking a supervisory role where it governs the respective setpoints for several other low-level controllers, e.g. PID type controllers¹.

¹A PID (proportional-integral-derivative) controller is low-level feedback controller which is extremely widespread with respect to applications in the process industry. This type of controller is not investigated in this thesis, although the value of controlling a process indirectly through PID type of controllers using MPC is mentioned. The setup and, specifically, the tuning of PID controllers are discussed by Skogestad, among others. [37] [38]

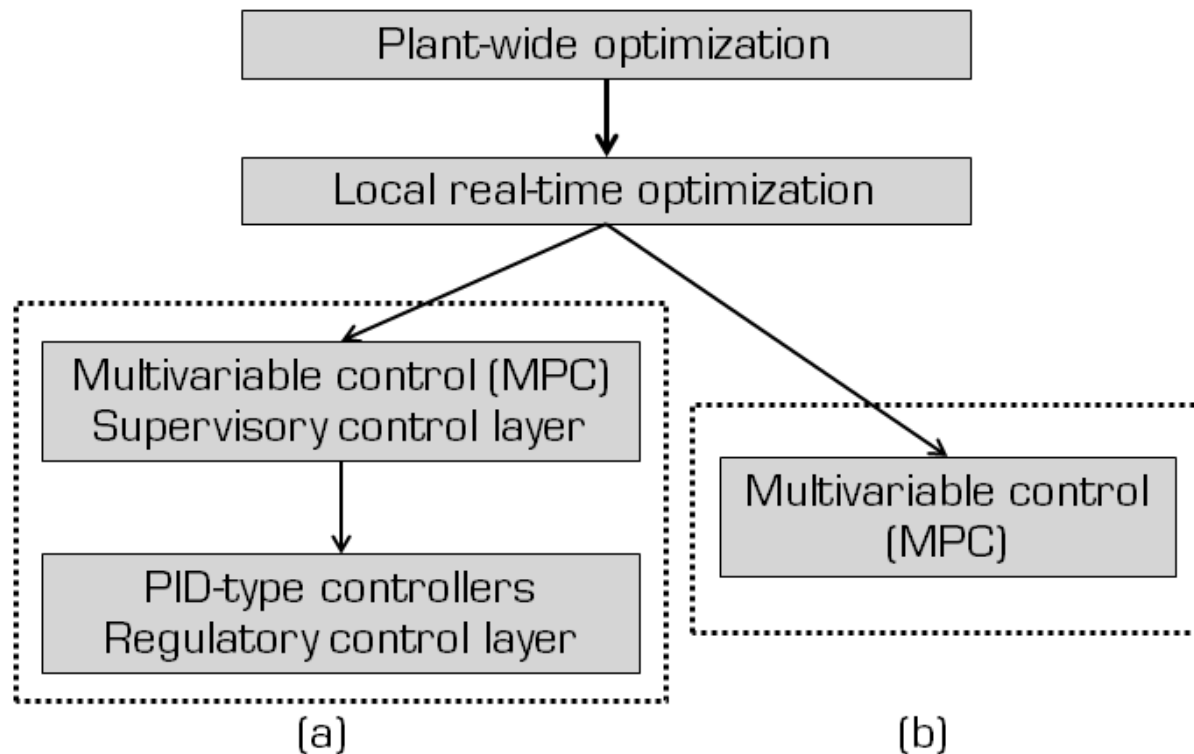


Figure 6.1: Two different suggestions for a typical control structure hierarchy for a plant. The role of MPC is indicated in both cases.

Fig. 6.2 is an extension² to Fig. 5.1 from Sec. 5, where the block diagram for an on-line estimator was illustrated. The controller block is now included in the diagram as well, and the interconnection between the controller and the other blocks (primarily the process itself and the process model) is shown. Just like for the estimator, the success of the controller is strongly dependent on the quality of the process model. This is in agreement with what was mentioned in the introduction to this thesis (Sec. 1), where it was stated that the ability of the controller to properly predict changes in the process is crucial in the context of safe and stable operation. [5]

The MPC is an optimal controller in the sense that the suggested controller action is the result (solution) of an optimization problem. In addition to being a mathematically optimal approach towards controlling a process, the strengths of MPC include effective treatment of constraints and easy application to MIMO (multiple inputs, multiple outputs) systems. Intricate and complex interconnections of the variables are, generally speaking, no problem for the MPC as long as this is properly accounted for in the process model.

The mathematical formulation of an MPC problem is given in Eqs. 6.3 - 6.8. The vector of model outputs (CVs) at time t_k is denoted by z_k , which is related to the states of the system through a measurement function (g) like indicated in Eq. 6.1. The vector of inputs at time t_k is

²This figure is printed with the permission of S. O. Hauger, Cybernetica AS.

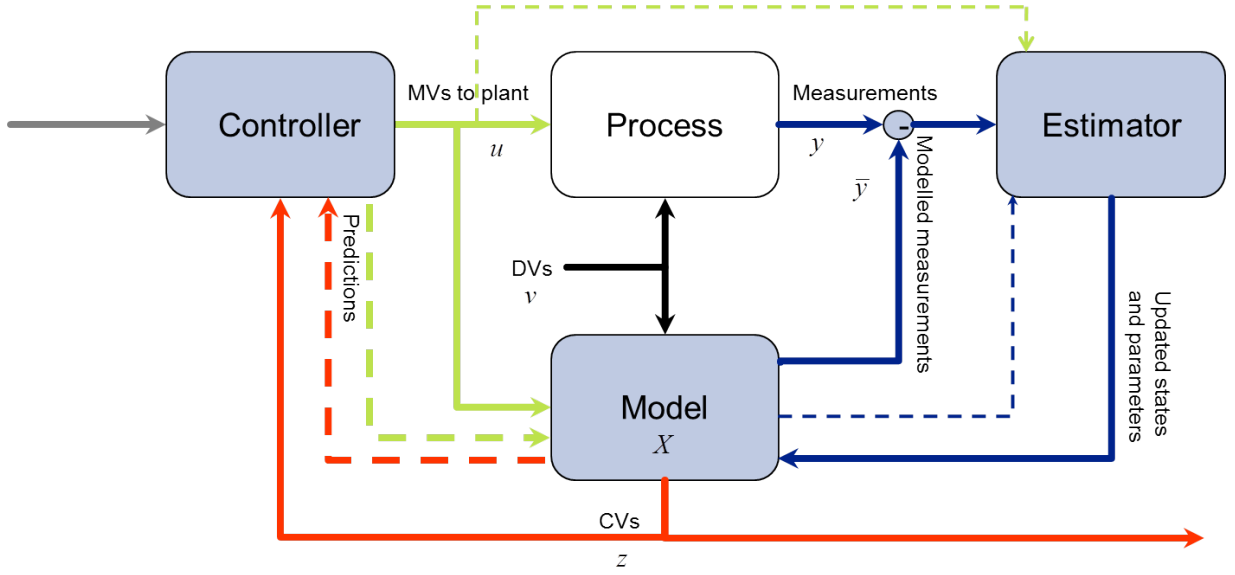


Figure 6.2: A conceptual block diagram showing the components of a typical MPC implementation with the required interconnection.

denoted by u_k , while the changes in the inputs are denoted by Δu_k in agreement with Eq. 6.2.

$$z_k = g(x_k, \theta) \quad (6.1)$$

$$\Delta u_k = u_k - u_{k-1} \quad (6.2)$$

Effort is made to emphasize that deviation variables are being used in the formulation, implying that the values of the variables reflect the deviation between the actual variable values and the respective set points. This means that z_k and Z in the following equations actually refer to the differences ($z_k - z_{k,ref}$) and ($Z - Z_{ref}$), respectively.

$$\min_U \quad \frac{1}{2} \left[\sum_{k=0}^{N-1} \left(z_k^T Q_k z_k + u_k^T R_k u_k + (\Delta u)_k^T S_k (\Delta u)_k \right) + z_N^T Q_N z_N \right] \quad (6.3)$$

$$\text{s.t.} \quad \dot{x} = f(x, u, \theta) \quad (6.4)$$

$$z_{min} \leq z \leq z_{max} \quad (6.5)$$

$$u_{min} \leq u \leq u_{max} \quad (6.6)$$

$$\Delta u_{min} \leq \Delta u \leq \Delta u_{max} \quad (6.7)$$

$$z_0 = g(x_0, \theta) \quad , \quad \text{given.} \quad (6.8)$$

The entries Q_k , R_k and S_k are penalty weighing matrices for prioritizing certain elements of the objective function at time t_k . In the formulation in Eq. 6.3, they represent the punishment of deviation from output setpoints, inputs and input changes, respectively. The desire for a strict controller with respect to obeying output setpoint would require weighing Q_k heavy, while a controller ensuring smooth and careful changes in inputs would have a heavily weighted S_k , for instance. It is emphasized that the *relative* weighing of the quantities govern the effective

penalty towards the cost function. These quantities, i.e. the entries of the respective matrices, are referred to as the tuning parameters of the controller, and will be elaborated briefly in Sec. 6.3.

Eqs. 6.4 - 6.8 are referred to as the constraints of the optimization problem. They are included to ensure agreement between the mathematical solution of the problem and the physical interpretation of the system. A volume should never be negative, and an output source (e.g. an electrical engine) should never produce 110% of the possible maximum, for instance. Constraints are not only set by the physical nature of the system in mind, but could also be set by the modeler. Consider, for instance, a swimming pool with a fairly strong heater. Physically speaking, it would be no problem to heat the water to 80 °C, but this would eliminate all of the comfort and safety associated with the pool, and the upper limit for the temperature of the pool would typically be set significantly lower³. The difference between the mathematical optimum and the constrained solution obeying the physical interpretation of the system is emphasized, and to achieve the optimal feasible solution, rather than the solution which is optimal in a strictly mathematical sense, is of importance for the performance of the controller. Forcing the controller to achieve unfeasible setpoints may in fact critically compromise the performance of the controller.

A way to avoid that the controller finds mathematically infeasible solutions for the controller action, is to introduce so-called soft constraints. In this sense, no solution for the outputs (CVs) are infeasible, in a mathematical sense, but these solutions are penalized harder towards the cost function than the truly feasible solutions are. For a soft-constrained control problem, the MPC equations (from Eqs. 6.3 - 6.8) become as shown in Eqs. 6.9 - 6.15.

$$\min_U \frac{1}{2} \left[\sum_{k=0}^{N-1} \left(z_k^T Q_k z_k + u_k^T R_k u_k + (\Delta u)_k^T S_k (\Delta u)_k + 2r_{1,k} \epsilon_k + r_{2,k} \epsilon_k^2 \right) + z_N^T Q_N z_N \right] \quad (6.9)$$

$$\text{s.t.} \quad \dot{x} = f(x, u, \theta) \quad (6.10)$$

$$z_{min} - \epsilon \leq z \leq z_{max} + \epsilon \quad (6.11)$$

$$0 < \epsilon \leq \epsilon_{max} \quad (\epsilon_{max} \rightarrow \infty, \text{ usually}) \quad (6.12)$$

$$u_{min} \leq u \leq u_{max} \quad (6.13)$$

$$\Delta u_{min} \leq \Delta u \leq \Delta u_{max} \quad (6.14)$$

$$z_0 = g(x_0, \theta) \quad , \quad \text{given.} \quad (6.15)$$

In Eq. 6.16, the cost function from Eq. 6.9 is shown in standard vector notation. In this case, the respective quantities of the control problem are denoted by capital letters, indicating that they are matrices. Q , R and S are now weighing *matrices*, containing the tuning weight of each respective quantity at each sampling time instant in the control horizon. Conversely, Z , U and ΔU represent every output deviation from the setpoints, every input and every input change, respectively, at all time instants through the control horizon. As a consequence, the variables are not indicated by k for the sample at time t_k . In addition, r_1 , r_2 and ϵ are now vectors. This notation is often preferred over the one in Eq. 6.9 because of its compact form. By deploying standard matrix operations from linear algebra, this formulation is often easier to work with, as

³At this point, the difference between the desired setpoint and the upper limit of the output variable is emphasized. Although the temperature of the swimming pool is controlled to a setpoint of e.g. 35 °C, it can increase beyond this, all depending on how the other variables of the system are weighted towards the cost function, and how the couplings for the respective variables are.

well. The constraints are treated accordingly.

$$\min_U \frac{1}{2} Z^T Q Z + \frac{1}{2} U^T R U + \frac{1}{2} (\Delta U)^T S (\Delta U) + r_1^T \epsilon + \frac{1}{2} \epsilon^T \text{diag}(r_2) \epsilon \quad (6.16)$$

The MPC problem (presented in Eqs. 6.3 - 6.16) is solved for every time instant separated by an interval, usually referred to as the sampling time, for N steps into the future. Here, N denotes the prediction horizon, governing how far into the future the controller will predict the behavior of the process. Although each step in the MPC calculates a sequence of controller actions into the future ($[u_0^T, u_1^T, \dots, u_{N-1}^T]^T$), it is emphasized that only the very first controller action (u_0) in the sequence is applied to the process before a new optimization problem is solved, and the sequence of controller actions is recalculated. This happens, as already mentioned, at every time instant of the controller, where the time separation between each calculation is governed by the sampling time, i.e. the controller horizon, which is discussed briefly in Sec. 6.3. It is crucial to realize that although the controller suggests an optimal way of controlling the process for several steps into the future, only the very first step of the controller action is deployed in each sample. A typical chain of events for an MPC is illustrated in Fig. 6.3, where a setpoint change for a CV is shown. The corresponding suggested control action and predicted model output response is also shown. Having a numerically efficient process model is of key importance, considering the amount of iterations involved in the solution of the optimization problem. For a controller with a short sampling time, i.e. a short control horizon, this requirement becomes particularly evident, because the time between each new optimization problem will be short in such as case. If the controller algorithms consistently fail to converge between each time sample, the risk of the controller falling permanently behind is prominent. A model which is *too* numerically efficient, on the other hand, in the sense that it does not accurately describe the process to a satisfying degree, may fail to suggest the proper controller actions. Once again, the trade-off between numerical efficiency and accuracy is stressed.

6.2 Additional features of the controller

In addition to providing controller actions which are optimal in a mathematical sense and handling constraints in an effective way, the typical MPC controller has certain features to strengthen the on-line performance.

Input blocking⁴ is an important concept in order to ensure the desired performance of the controller. The purpose of input blocking is to reduce the number of independent variables, i.e. degrees of freedom, for the optimization problem in the MPC, which in turn yields a lower demand for computational power. To exemplify the need for this action, consider a system having 5 separate inputs (MVs), for which the control horizon is 20 samples into the future. This will yield an optimization problem with 100 unknowns, with corresponding demands for computational effort. In the optimization algorithm, the free variables are investigated in a manner where perturbations to the variables are made, and this is computationally demanding. If these individual controller actions are blocked into (for instance) 5 blocks of increasing size, for instance 1, 2, 3, 6 and 8 samples, respectively, the problem will already be reduced to 25 unknowns, easing the computation. Further simplifications may also be possible without severely

⁴Blocking here refers to the concept of gathering several units in the same block (unit). It should not be confused with the concept of preventing/obstructing something (blockage).

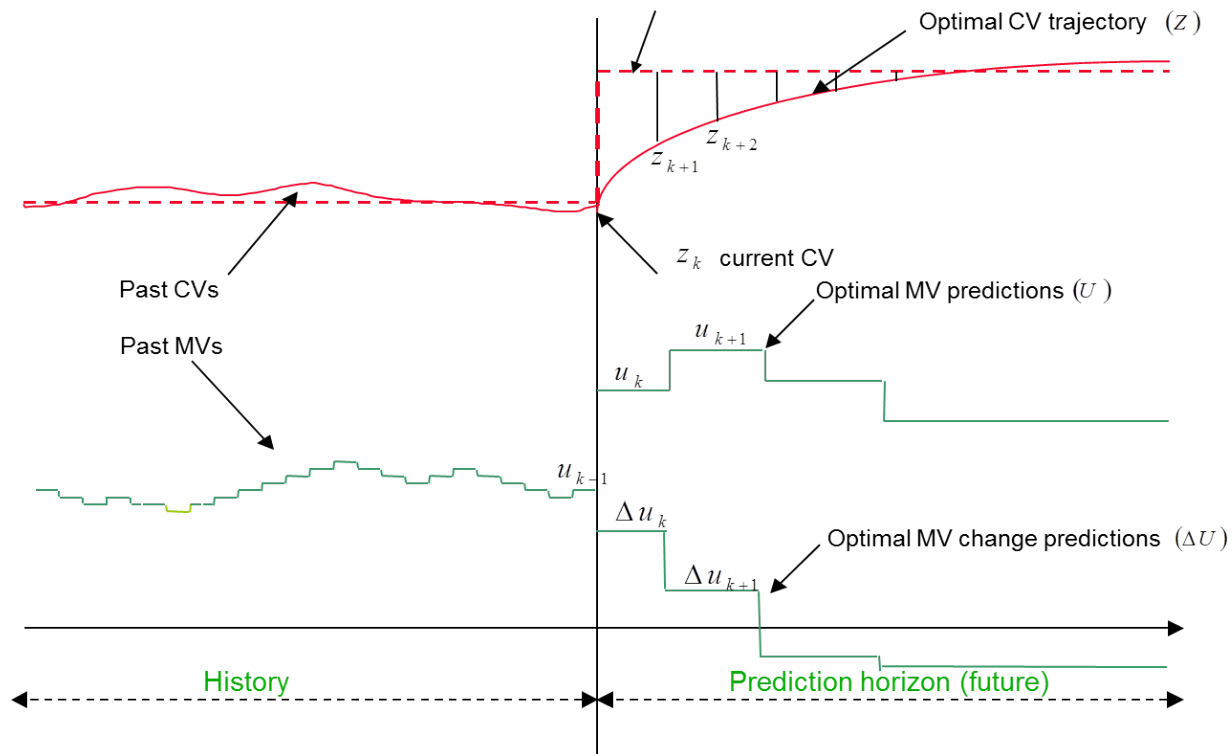


Figure 6.3: Typical trajectory for a process being controlled using MPC during a change in setpoint for one of the outputs (CVs).

compromising the performance of the controller. The concept of input blocking was illustrated in Fig. 6.3 in Sec. 6.1, where the future controller action was blocked in separate blocks of increasing size as the horizon moves farther from the instant. Close to the time of the actual sample, the next few steps into the future should be calculated with quite high accuracy, but the controller action far into the future may not be as important to determine to the same accuracy. The reason for this is simply that it is the very first step of the controller action which will actually be deployed, as explained in Sec. 6.1. The suggested controller action at the end of the control horizon will, as a matter of fact, never actually be deployed, but will be recalculated when that point in time is reached. This situation is highlighted in Fig. 6.4 as well⁵. Emphasis is put on the fact that the controller horizon and prediction horizon may be different, but the prediction horizon must be longer than the control horizon.

6.3 Controller tuning

In the introduction to this thesis (Sec. 1), it was stated that the key to achieve efficient process control may lie in obtaining an adequate process model, as this will lay the proper foundation for the model-based control system, rather than (fine)tuning the controller. Despite of this, it was emphasized that the importance of controller tuning is not neglected, and this section will

⁵Figs. 6.3 & 6.4 are printed with the permission of S. O. Hauger, Cybernetica AS.

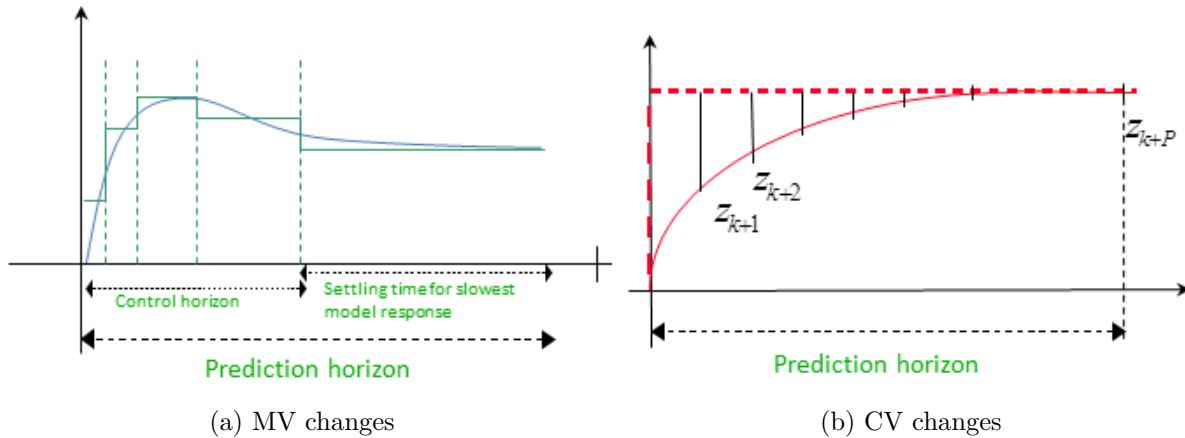


Figure 6.4: Controller action illustrating the concepts of controller and prediction horizon as well as input blocking, including the corresponding system response.

discuss the possibilities and advantages of tuning the MPC type controller sensibly.

In Eq. 6.9 in Sec. 6.1, the optimization problem, also referred to as the cost function, of the MPC was introduced. The compact equivalent was provided in Eq. 6.16. This formulation includes the weighing matrices (Q_k , R_k , S_k) governing the relative punishment, towards the cost function, when the respective outputs deviate from their designated setpoints at time t_k . In being tuning parameters, these matrices are left for the modeler to decide, and this activity is rarely trivial to perform adequately [33]. Indeed, the desire to control some outputs more strictly than others may give a qualitative indication towards the relative weighing of the deviations, but quantitatively deciding the weighing, i.e. the tuning parameters, is more complex than that, because it affects the stability and robustness of the controller. The performance of the controller may change in seemingly unpredictable ways when the tuning of the controller, or the estimator for that matter, is changed. Systematic approaches towards tuning MPC is hard to achieve, especially in MIMO systems with strongly coupled dynamics, and experience from tuning other similar processes sometimes proves to be more valuable than initiating a fixed systematic approach. Some people highlight this as one of the drawbacks of MPC technology, but the performance of a successfully tuned controller helps to strengthen the general reputation of MPC.

The R -matrix in the MPC formulation is usually not utilized, and it will not be tuned in the work of this thesis. The actual values of the MVs (or the deviation of the MVs from nominal values to be precise) are, in other words, usually not punished towards the cost function. A hypothetical case where it would be of interest to punish inputs (MV) deviating from nominal values would be if, for instance, an input is an energy demanding source for which the nominal value is very low. If it is undesirable to use this input extensively, but it can not be strictly constrained due to considerations with respect to safety, it would be sensible to punish the MV deviation for this variable using the R -matrix. The two other penalty matrices, i.e. Q and S will be investigated and tuned in this work.

The respective CV deviations from the assigned setpoints (Z) are obviously punished (through Q), as we want the controller to drive the process as close to the desired setpoints as possible. For

cases where soft constraints are used, which is advised (and deployed in the work of this thesis), the two vectors of tuning parameters r_1 and r_2 must be set as well. As mentioned in Sec. 6.1, these tuning parameters govern the linear and quadratic penalties, respectively, for violating the soft constraints. Changes in MVs (ΔU) are also commonly punished towards the cost function. A reason for this is that abrupt changes in controller action is undesired for many industrial processes, and may violate the physical conditions of the system. Because of this, the changes in the MVs are usually subject to constraints in addition to the penalty in the cost function. These constraints are shown in Eq. 6.14, and are considered to be hard constraints which are impossible to violate in order to secure the physical validity of the system. For some of the simulations in this thesis, presented in Sec. 7, these concepts become important. It is not necessarily realistic, for instance, that a cooling jacket for a tubular reactor can change its temperature in the range of 20 °C in one short time sample (e.g. 30 seconds), and this change needs to be constrained. In other cases this effect may be completely feasible. Consider, for instance, an efficient valve which could easily go from completely closed to fully open in a very short matter of time. In such cases, it may not be necessary to penalize the changes in this input. The Cybernetica CENIT software, which was introduced in Sec. 4.3, supports all these features with respect to constraints and controller tuning.

In addition, the parameters known as prediction horizon and control horizon (illustrated in Fig. 6.4a) are also considered to be tuning parameters for the controller. The determination of these two parameters are, like the weighing matrices, not trivial. The longer these horizons are, the longer into the future the controller predicts the process behavior, which obviously is a good thing. It is of interest to predict as far into the future as possible, but this also requires more computational effort, which is crucial in the on-line implementation. The best choice for these parameters is, in other words, a classic trade-off. In Sec. 7.8, it will become evident that these consideration are crucial in order to properly control the conversion of monomer through the reactor, for instance. Part of the reason that a long prediction horizon is needed is the time delay which is associated with the tubular reactor, which may constitute an obstacle to the effective control of the reactor.

6.4 The use of a separate model for plant representation

The very idea of process modeling is to formulate the behavior of a specific process in a mathematical framework, and for the purpose of process control, models are deployed in on-line use with estimator algorithms updating the model, as previously discussed (Sec. 5), and controller algorithms deciding the appropriate controller action for the next step of the process. For the work in this thesis, no real process for which the model can be tested on-line exists, and the concept known as plant replacement is introduced to simulate the on-line behavior of the developed process models. The idea is to use two separate models in the controller calculations (optimization problems, etc.) and the plant representation (plant simulation), respectively. The Cybernetica RealSim software, which is introduced briefly in Sec. 4.3, undertakes the task to represent the plant behavior, using one edition of the process model. The Cybernetica CENIT software on the other hand, which is governing the NMPC, uses another version of the process model. A conceptual drawing of these ideas is given in Fig. 6.5. In this manner, the testing environment for the performance of the models (in the context of control) is created artificially. Like indicated in Fig. 6.5, the two separate models can have differing characteristics, although they represent the same process. The purpose of the plant replacement is to simulate the plant

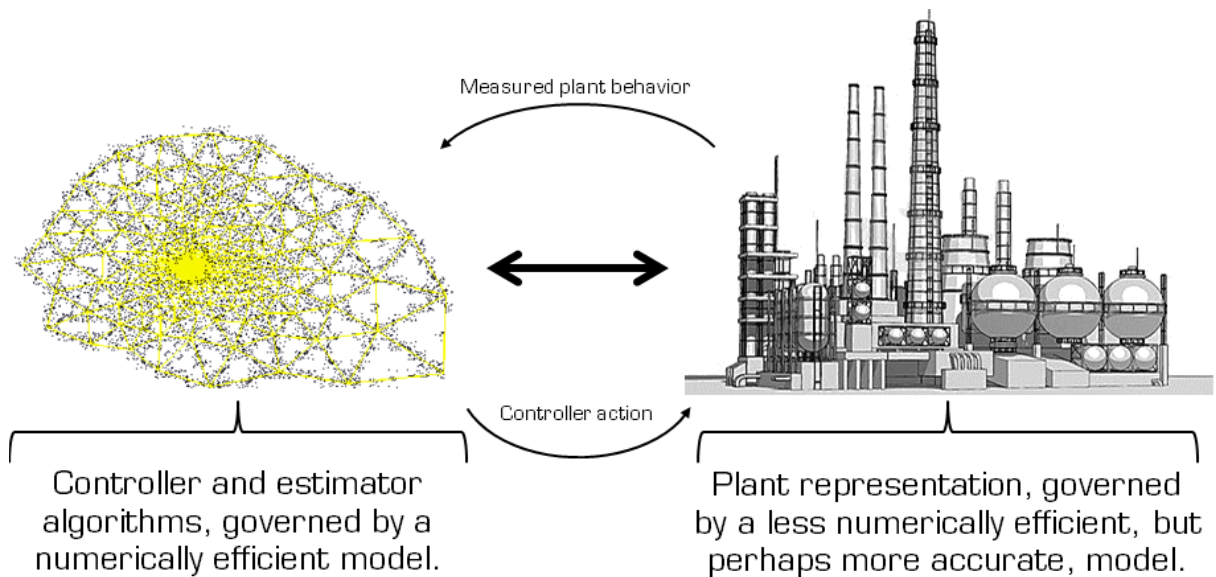


Figure 6.5: Conceptual illustration for the distinction between the controller algorithms and the plant. In this thesis, all components of the structure are represented by computer models, but the scope is to mimic the real-life behavior as accurately as possible.

behavior according to the inputs (MVs) decided by the controller and return measurements of interest back to the controller. The desire for numerical efficiency is not the most important concern with respect to this specific task, and the ability to return as accurate process behavior as possible is more important. The plant replacement should, in other words, be governed by the most accurate process model available. For the controller algorithms, the situation is the opposite. Because the model is used for iterations, optimization problems, etc., the numerical efficiency is crucial. The accuracy of the model is not as important as long as there is a good qualitative agreement with the plant replacement, but the performance of the controller is expected to improve with increasing model accuracy. In the context of spatial discretization for the process models, which has been prominent throughout this thesis, the sensible approach is to deploy a model with a refined/tight discretization in the process simulator, while the controller uses a coarsely discretized model, generally speaking.

Emphasis is put on the fact that to have differing models in the controller unit and the plant representation is more realistic than having the same model handling both tasks. The reason is that in real-life practice, a process model very rarely manages to encapture the entire detailed complexity of a given process, and deviations between the model-predicted behavior and the real-life process behavior is unavoidable. In the (very improbable) event where the process model is a perfect representation of the real plant, the need for on-line estimation (Kalman filter or similar) would actually not be needed, as the predictions made by the controller model would be identical to the actual measured process behavior. The whole purpose of the estimator, which is elaborated in Sec. 5, is to accommodate the fact that real plant measurements differ from predicted model outputs, and to reinforce the model based on this. This is an extremely important step along the way to achieve effective model-based control of a process, which will be demonstrated in Sec. 7.

Section 7

Simulations and results

The purpose of this section is to provide the main results from the thesis work. This part of the thesis is initiated with a recapitulation from the specialization project on which this thesis is based [1]. In Sec. 7.1, the model components used in the modeling work is used to construct a test case with a semi-batch reactor, in order to demonstrate a simple temperature control problem for this reactor. Starting to consider the tubular reactor, Sec. 7.2 introduces the behavior of the established models by discussing off-line parameter estimation, which is an important step along the way to implement a first principles process model for on-line use. In this activity, both kinetic and thermodynamic properties are modified to validate the models. Sec. 7.3 represents the first approach towards the Kalman filter, and illustrates how the Kalman filter can be tuned by first considering the filter in off-line simulations without controller action. In Secs. 7.4 - 7.8, the behavior of the established models in closed loop simulations are explored, using the models with software for NMPC. The Cybernetica CENIT software, which is utilized for the simulations in this section, is also demonstrated for a simplified example case in App. G, where the temperature and the fluid level in a simple buffer/storage tank is simulated and controlled. The established theory of Secs. 5 & 6, i.e. the fundamentals of on-line estimation and model-based predictive control, respectively, constitutes the foundation of this part of the thesis. By the end of the section, in Sec. 7.9, a discussion is provided where possible modifications to the reactor in an industrial perspective are evaluated briefly.

7.1 Introductory case: Control of a semi-batch reactor

The first reactor type of interest in the series of control cases is actually a semi-batch reactor. A brief introduction to the model equations for a semi-batch reactor is provided in App. C. As mentioned in previous considerations, the semi-batch reactor is a well explored reactor setup for emulsion copolymerization. Another justification for including this case in this thesis, for which the focus lies heavily on continuous tubular reactors, is the fact that while the preceding specialization project treated the semi-batch reactor setup extensively with respect to modeling and off-line parameter estimation, simulation cases involving control were never conducted [1]. Because of this, it is interesting to investigate how the reactor model behaves in the context of control.

The reactor in mind is a small (lab-scale) tank reactor, with an efficient mixing mechanism, having a batch time in the range of a few hours. The feeding strategy is to add reactants to the reactor until the batch is halfway through, after which all feeds to the reactor are stopped. The reactor feed has approximately 56 w% monomer, of which approximately 70 w% is monomer type 1. There are several valid candidates when choosing outputs (CVs) of interest for the semi-batch reactor, like conversion of monomer or molecular weight distribution, but this has not been considered in this, rather simplified, introductory case. Instead, the main focus in this case is to control the reactor temperature to a desired setpoint. The importance of adequate temperature control for polymer reactors has already been emphasized on several occasions, and remains one of the clearest benefits of a successful controller.

Fig. 7.1a shows the controllable jacket temperature of the reactor. There exist several strate-

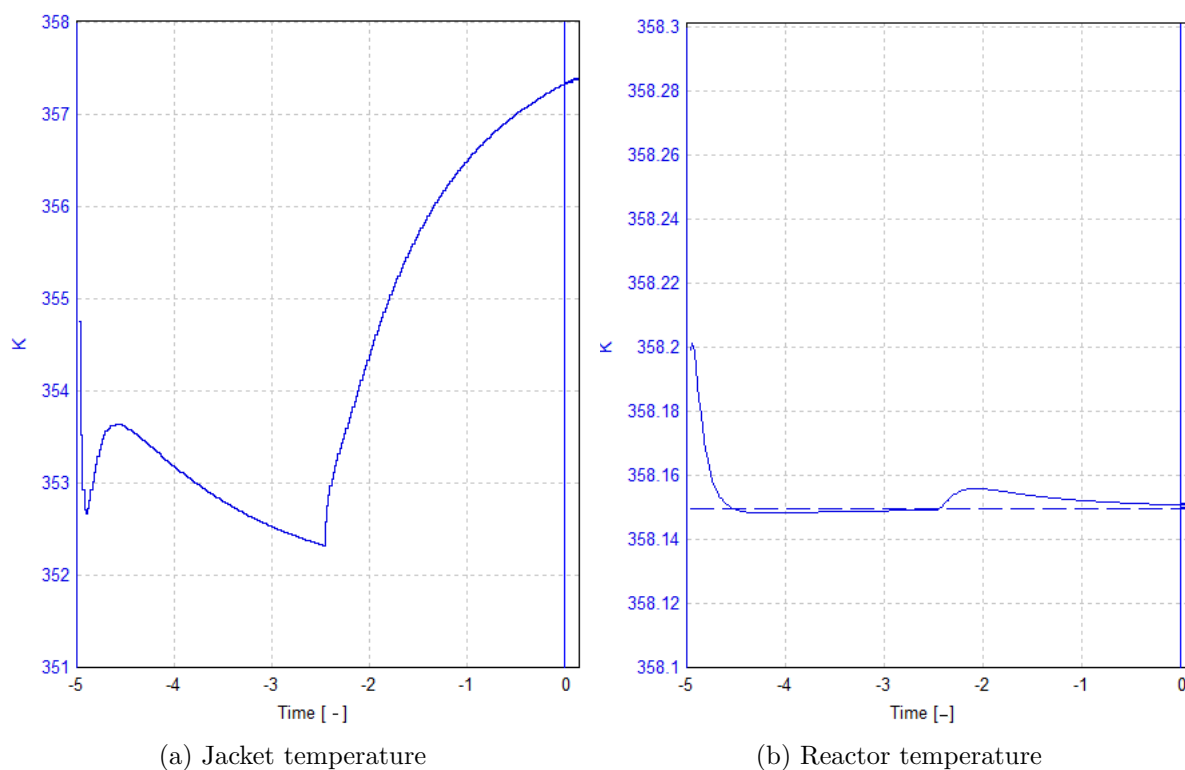


Figure 7.1: Controller action (MV) and process response (CV) for a temperature control case of a semi-batch reactor.

gies to achieve the desired reactor cooling. Among them are the physical cooling jacket encapsulating the reactor, and an oil/coolant bath in which the reactor is submerged, but several other setups for reactor cooling are possible. The jacket temperature is, in any case, considered to be a controllable quantity, i.e. an input (MV) to the process. Fig. 7.1b portrays the corresponding reactor temperature to the changing jacket temperature, which is an output variable (CV) from the process. From Fig. 7.1, it is apparent that the temperature control for the reactor is satisfying with respect to staying close to the desired setpoint of 358.15 K^1 . An interesting observation is that the cooling temperature has a clear monotonic increase through the latter half of the batch. The reason for this is that the progress of the chemical reactions, and thus the development of heat, is much more evident in the first half, where new reactants are continuously fed to the reactor. In the latter half, the main purpose is to ensure that the reactions proceed safely towards completion, preferably as close to completion as possible. In an industrial application, it might not be strictly necessary to maintain the temperature setpoint through the second half of the batch, as long as the temperature is not significantly reduced or increased. One of the reasons for this is that the development of hotspots and temperature spikes in the reactor are more common to arise in the first half of the batch.

The controller action, i.e. the adjusting of the jacket temperature, is not aggressive with

¹Notice the scaling of the ordinate axis in Fig. 7.1b, which only shows a narrow band around the setpoint temperature. The temperature is virtually on setpoint throughout the batch.

respect to changing dramatically, but shows a rather smooth behavior. For this specific test case, which is not affected by disturbances during the simulation, this result agrees with the concept of having an estimator predicting the changes in temperature into the future, and a controller choosing effective yet sensible actions as a consequence. The change in the cooling fluid temperature is not constrained in the controller, and the system shows no need for a dramatic change in the cooling strategy during the duration of the batch.

An important aspect of temperature control in the context of polymerization reactors, which has been discussed earlier [1], is the fact that effective temperature control is often easier to achieve in lab-scale experiments than in industrial scale reactors. In an industrial scale batch type reactor, the heat transfer properties may differ significantly from what is the case for the lab-scale reactor, raising the need for more aggressive temperature control to stay on setpoint. For most cases which have been considered in this thesis, lab-scale experiments are performed using a cooling batch, in which the entire reactor is submerged. This setup is, in practice, impossible to achieve in a full-scale industrial reactor. In addition, industrial scale reactors can experience mixing problems to a larger extent than what is the case for lab-scale reactors. In the case of poor mixing, hot spots can arise in the reactor, which disturbs the reactor temperature and thus affects the efficient control of the reactor temperature. Most batch type reactors have proper agitators that ensure well-mixed conditions, but for tubular reactors the fluid flow patterns will affect the generation of hotspots. Assumptions and simplifications are usually introduced to describe the fluid flow, while the actual fluid flow in real reactors may be very intricate.

7.2 Preliminary task: Off-line parameter estimation

Moving on from Sec. 7.1, the next step is to consider a continuous tubular reactor setup. When a process model has been established using first principles, it should be verified and validated using experimental data to ensure a satisfying agreement between the behavior of the model and reality. The term reality refers, throughout this thesis, to the actual behavior of the reactor, as revealed by experimental investigations. Off-line parameter estimation was among the main topics of the work preceding this thesis [1], and the importance of performing off-line parameter estimation was emphasized. For the tubular reactor model, model fitting remains equally important. In this section, such a procedure is demonstrated for an MCV type process model, as introduced in Sec. 3.5. In later sections, Sec. 7.8 in particular, this model type will be deployed as a plant replacement model, and hence the ability of the model to adequately represent reality is important.

Temperature control is, as emphatically mentioned, an important activity for polymerization reactors, and has, as a consequence, been included in many of the simulations of this thesis, but for this specific part the temperature is kept constant. The main focus is instead aimed at the conversion of monomer through the reactor, i.e. the kinetic and thermodynamic factors governing the extent of the respective chemical reactions and the phase equilibrium of the chemical system. One reason for this is the desire to ensure agreement between the simulated conditions and the operating conditions under which the experiments were conducted. Another reason is that the heat transfer properties will be accounted for in on-line considerations, rather than this preliminary off-line parameter estimation.

The conversion of monomer is calculated at four spatial locations along the reactor and

compared with experimental data from a lab-scale reactor². The four spatial locations are the reactor outlet (indicated by blue curves in the following figures), which is typically the point of highest interest, as well as points at 75% (red curves), 50% (black curves) and 25% (pink/magenta curves) of the total reactor length. The simulations used a smart-scale tubular reactor operating with 30 w% monomer in the continuous feed, of which 76.5 w% was monomer type 1. The temperature of the reactor was controlled using an oil bath in which the coiled reactor tube was entirely submerged. This setup provides an effective way to maintain a stable reactor temperature. The experimental behavior of the reactor is measured at steady-state conditions, but for the purpose of illustration, the following simulations include the transient zones of the simulations as well, to represent the start-up of the reactor in each case. For the specific conditions of these simulations, steady-state conditions are achieved no sooner than 11 minutes after the reactor initiation, and the measurements are indicated by dashed lines in Figs. 7.2, 7.3 & 7.5, where the coloring corresponds to the coloring for the model predictions. As an addition, the steady-state profiles in space are included in Fig. 7.4, where the conversion of monomer through the reactor is shown before and after the parameters have been adjusted, for both the MCV type reactor model and the NMOL type reactor model.

For the off-line optimization procedure, in which the specific parameters are determined, the Cybernetica Modelfit tool was utilized, as described in Sec. 4.2. By having access to experimental data for the conversion of monomer under isothermal reactor conditions, the parameters governing chain propagation of polymer as well as the inter-phase mass transfer, i.e. factors governing the phase distribution, were chosen for modification. In this sense, both the dynamic behavior of the model and the steady-state solution are perturbed in the parameter estimation procedure, but since the recorded experimental data is obtained from steady-state operation of the reactor, the objective of the optimization procedure is to achieve a satisfying fit between the model and reality at steady-state operation. The thermodynamic properties, i.e. the partition coefficients in this case, govern the *actual* steady-state solution, while the dynamic properties³, in this case the kinetic adjustment factors, ensure that the proper steady-state solution is reached *in the right time*. The initial behavior of the model, before any parameters were modified, is shown in Fig. 7.2. After the chosen parameters of the model are optimized, the agreement between the model predictions and the experimental behavior becomes as shown in Fig. 7.3.

The model does not present with a perfect fit between the model predictions and reality after the parameters have been adjusted, but the improvement is noticeable, particularly with respect to the overall conversion of monomer through the reactor, i.e. the achieved conversion at the spatial end of the reactor. This point in the reactor is usually the one of highest interest for the application of the process model, as the terminal product qualities often are desired to match certain predefined requirements. In other words, being able to control the conditions at the reactor outlet is important, but measuring and controlling quantities at internal points of the reactor is also useful. A remark in this context is that while the optimization problem of the off-line parameter estimation procedure works to eliminate all deviations between the measured behavior and the model predictions, it would be sensible to weigh the deviations for the reactor outlet slightly heavier than the deviations at the internal points. The resulting effect is that the

²The experiments were conducted at TU Hamburg, Department of Macromolecular Chemistry (Institut für Makromolekulare Chemie). The work of F. Lüth, in conducting experimental studies of the smart-scale reactor, is acknowledged and greatly appreciated.

³The kinetic adjustment factors were introduced in Eq. 2.9 in Sec. 2.1. The monomer partition coefficients are discussed in Sec. 2.2.

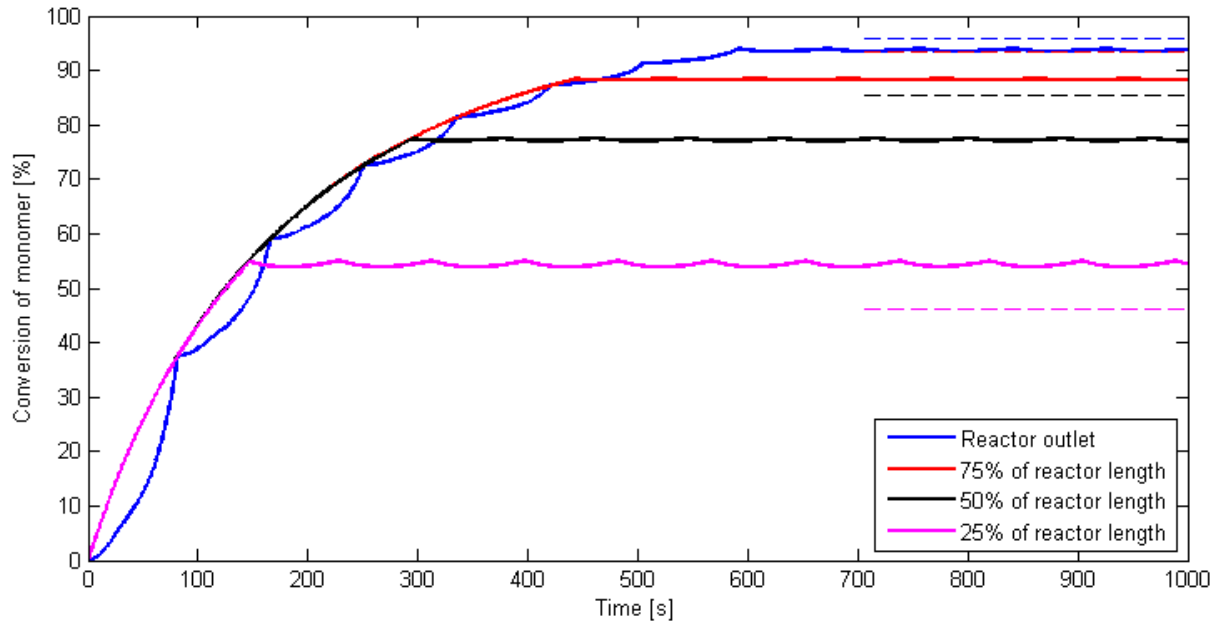


Figure 7.2: Simulation showing the conversion of monomer at four locations in the tubular reactor, before off-line parameter estimation has been performed.

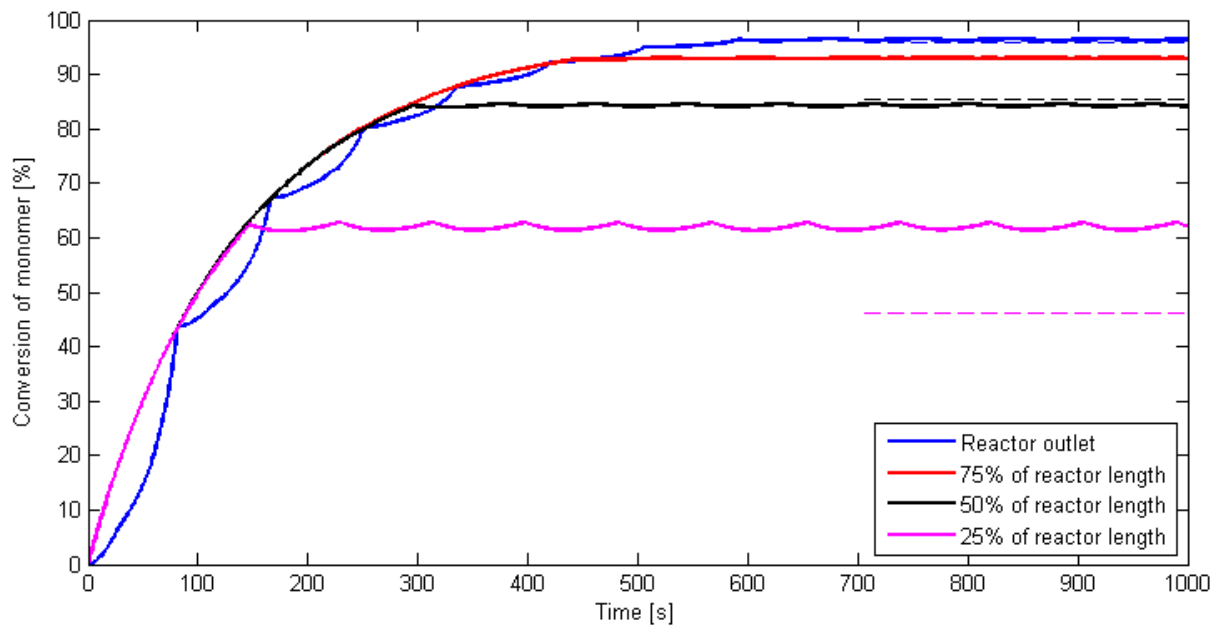


Figure 7.3: Simulation showing the conversion of monomer at four locations in the tubular reactor, after off-line parameter estimation has been performed.

conditions at the reactor outlet are forced to agree with the experiments at the expense of the deviations at the internal points. This effect is evident in this case, and while the agreement between the model and the experiments become virtually inseparable at the reactor outlet, there is a severe deviation at the point 25% into the reactor. The changes in the respective parameters which were chosen for modification in going from Fig. 7.2 to Fig. 7.3 are reproduced in Tab. 7.1.

Table 7.1: Parameter modification for off-line parameter estimation for a tubular reactor using an MCV type model. The selection of parameters are chosen to affect the steady state behavior of the model.

Parameter	Old value		New value
Adjustment factor, type 1-1 chain propagation	50	→	43.75
Adjustment factor, type 1-2 chain propagation	22	→	87.32
Adjustment factor, type 2-1 chain propagation	50	→	43.75
Adjustment factor, type 2-2 chain propagation	22	→	63.52
Partition coefficient, monomer type 1 in droplet phase	2 618	→	2 290.75
Partition coefficient, monomer type 2 in droplet phase	705	→	714.63
Partition coefficient, monomer type 1 in particle phase	1 570	→	2 395.09
Partition coefficient, monomer type 2 in particle phase	460	→	402.50

The general trends in the results from the off-line parameter estimation procedure are summarized below.

- The kinetic reactivity of growing polymer chains with type 1 endgroup is increased for the homo-propagation case (1-1 propagation), while it is increased for the cross-propagation (1-2 propagation) reaction. It is emphasized that when the adjustment factor is decreased, the actual rate is increased and *vice versa*, in agreement with Eq. 2.9.
- The change in the kinetic reactivity of growing polymer chains with type 2 endgroup shows the opposite behavior as the reactivity of type 1 endgroups. The rate of the cross-propagation reaction (2-1 propagation) is slightly increased, while the rate of the homo-propagation (2-2 propagation) is decreased.
- The solubility of monomer (i.e. the specific partition coefficients) in the monomer droplet phase relative to the water phase is adjusted. If the partition coefficients are decreased, the effect is more monomer in the water phase, and indirectly more monomer in the particle phase. As a consequence, a decrease in the partition coefficient for the monomer phase will indirectly give a higher extent of chain propagation in the particle phase. It is interesting to notice that while the partition coefficient for monomer type 1 decreases, the partition coefficient for monomer type 2 increases slightly. This is possibly because monomer type 1 is more reactive with respect to chain propagation in the particle phase.
- The solubility of monomer in the particle phase relative to the water phase is adjusted in agreement with what was observed for the corresponding partition coefficients for the monomer droplet phase. The partition coefficient for monomer type 1 is increased slightly

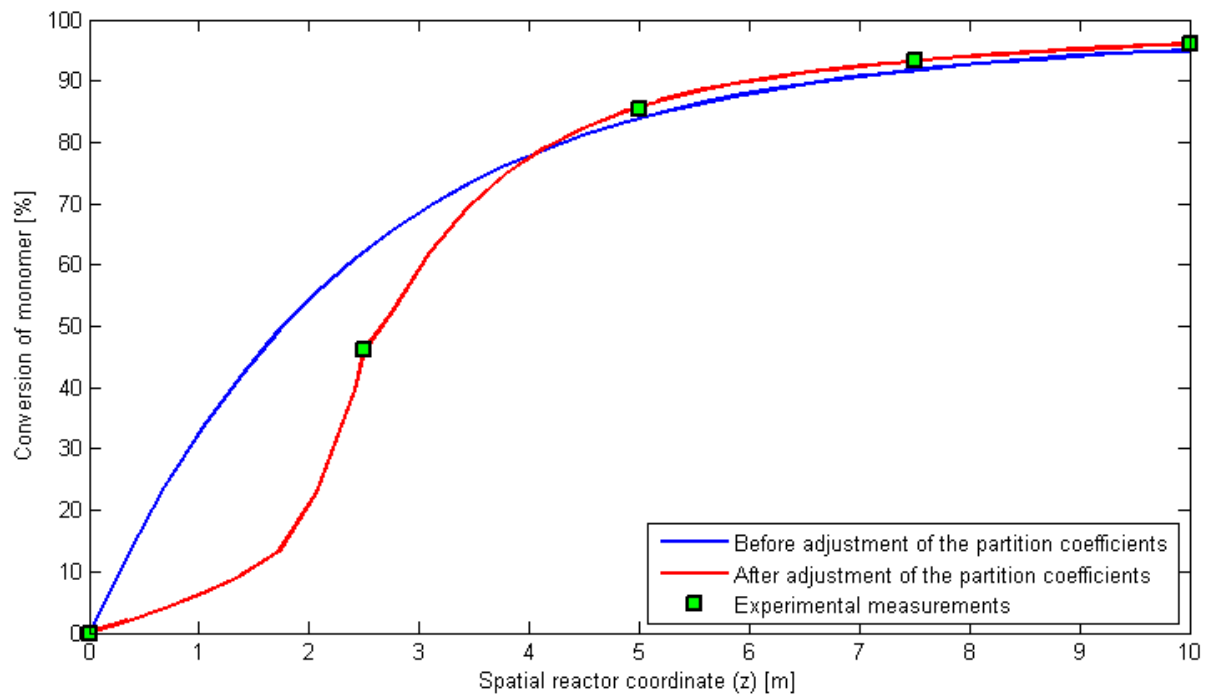
to yield more monomer 1 in the particle phase and thus promote the consumption. Conversely, the partition coefficient for monomer type 2 is reduced slightly.

There is still a disagreement between the model and the experiments, with respect to the conversion of monomer, which the parameter modification procedure does not seem to correct (Fig. 7.3). Although the conversion of monomer halfway through the reactor (indicated by black curves in the respective figures) is slightly lower in the model predictions than in the experiments, the conversion one quarter into the reactor (represented by pink/magenta in the figures) is actually found to be significantly lower in the experiments than what the model suggests. The model will, in other words, predict a conversion of monomer in the first part of the reactor which is too high, while the corresponding predictions will be slightly lower than what the experiments indicate later in the reactor. Because the modification of a set of parameters, as shown in Tab. 7.1 when going from Fig. 7.2 to Fig. 7.3, does not account for this error in the model, it is necessary to dig deeper. This undesired effect is believed to originate from the inability of the model to encapture the fact that the solubilities of the respective monomers in the growing polymer particles are indeed functions of the polymer particle size⁴. To accommodate for this, a modification to the model must be added, in which the partition coefficients, which until now have been constant, are allowed to vary in agreement with the state of the reactor mixture. By adding a correcting term to the respective partition coefficients, which is dependent on the size of the growing polymer particles, the extents of the specific chemical reactions can be modified in a desired manner, possibly without performing all of the changes in the parameters which were required when going from Fig. 7.2 to Fig. 7.3. The dependency of monomer solubility on polymer particle size has been investigated by Pokorny [35].

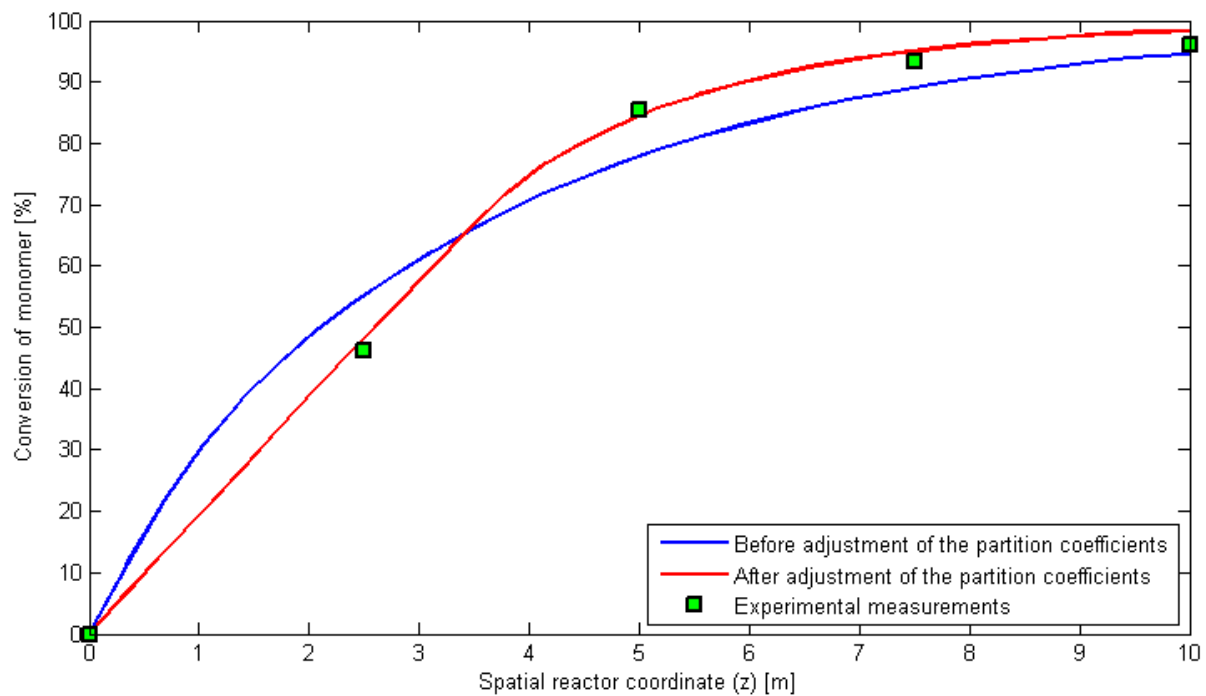
Before any modifications have been done, the partition coefficients (as introduced briefly in Sec. 2.2) are believed to overestimate the solubility of the monomers in the spatial beginning of the reactor, and they are, conversely, believed to slightly underestimate the solubility of the monomers through the rest of the reactor. This behavior is confirmed by steady-state simulations which were conducted before and after the phase equilibrium of the models were revised. These simulations are shown in Fig. 7.4, where the conversion of monomer through the reactor is shown for both of the modeling approaches which have been considered in this thesis, i.e. the numerical method of lines approach and the moving control volumes approach, which were introduced in Secs. 3.3 & 3.5, respectively. These figures are different from Figs. 7.2, 7.3 & 7.5 in that they provide the spatial profiles of the conversion of monomer at steady-state, at a specific time rather than the transient behavior. The fact that there is a difference in the observed behavior of the NMOL type model and the MCV type model in Fig. 7.4 is due to a combination of reasons. On one hand, they might be governed by slightly different parameters to yield the optimal fit to the experimental data. On the other hand, the phase equilibrium is treated differently in the two modeling approaches, resulting in a slight difference with respect to how the partition coefficients affect the conversion of monomer.

The structure of the mentioned correction term to the respective partition coefficients is indicated in Eq. 7.1. The complete set of equations to determine the solubility of the respective monomers in the particle phase is based on the work of Morton *et. al.*, who did a comprehensive

⁴It may appear intuitive that when the particles of the polymer phase grows, their ability to hold monomer increases. This observation is explored and elaborated in COOPOL D4.4 (internal COOPOL report), where theoretical investigations are conducted to determine the solubilities of the respective monomers in the polymer particle phase. [35]



(a) Reactor model using the numerical method of lines (NMOL)



(b) Reactor model using the approach of moving control volumes (MCV)

Figure 7.4: Figures showing the conversion of monomer through the reactor. The effect of modifying the partition coefficients for the phase equilibrium is illustrated for both modeling approaches of the thesis.

work on the swelling of latex particles, which is based on the Flory-Huggins theory for multi-phase systems [41]. This approach is rich in details and accuracy, but one of the major concerns of this approach is the increased demand for computational effort which is associated with the increased complexity of the equation system. In order to bypass this obstacle for the purposes of this work, an empirical correction term is constructed to yield the desired effect without severely compromising the numerical efficiency of the model. When this term is introduced, the new functions for the partition coefficients become as indicated in Eq. 7.1.

$$k_i^p = a_{i,1} \left[a_{i,2}\sigma + \arctan\left(\frac{r_p}{a_{i,3}}\right) \right] \tilde{k}_i^p, \quad i \in [M_1, M_2] \quad (7.1)$$

In this expression, $a_{i,1}$, $a_{i,2}$ and $a_{i,3}$ are parameters which are introduced and tuned to fit the experiments, while σ and r_p denote the surface tension and the particle radius of the growing polymer particles, respectively. The surface tension for the growing polymer particles has not been investigated in detail, and the product $a_{i,2}\sigma$ has been fitted as one parameter. The varying partition coefficients are denoted by k_i^p while the constant ("steady-state") coefficients are indicated by \tilde{k}_i^p . The respective partition coefficients are modified for each of the types of monomers in the system, and it is emphasized that this change is only introduced for the partition coefficients for the polymer particle phase, while the partition coefficients of the respective monomers in the monomer droplet phase are not dependent on the size of the growing polymer particles.

The implementation of the modification to the partition coefficients must be done in agreement with the already established strategy for solving the phase equilibrium of the system. Like discussed in Sec. 2.2, the phase equilibrium calculations may introduce a certain demand for computational effort, and it is desirable that the correction terms do not act to further complicate these calculations significantly. For the NMOL modeling approach (Sec. 3.3), for instance, the phase equilibrium problem is solved by assuming that equilibrium is achieved in each discrete step, for which the volumes of the respective phases are determined in an iterative manner. The reason for this is that the species mass balances for each species are formulated to include all the phases of the reactor rather than constructing one balance equation for each phase with corresponding terms accounting for the inter-phase mass transfer. Such an approach is expected to yield a large demand for computational effort, as the number of states in the model would be nearly tripled. It is emphasized, however, that the iteration scheme of the phase distribution also contributes with a certain demand for computational effort. As a consequence, r_p from Eq. 7.1 must be included as part of the iterative scheme in the NMOL approach. In the MCV modeling approach (Sec. 3.5), on the other hand, the phase equilibrium problem is treated differently. In this case, the inter-phase mass transfer is accounted for in the dynamic model equations. The inter-phase mass transfer is then determined by a combination of a dynamic rate constant and a driving force based on the thermodynamic equilibrium condition, i.e. the partition coefficients, like indicated in Eq. 2.32. The phase equilibrium is, in other words, not solved in an iterative manner, and the volumes of all the phases, as well as the particle size of the growing polymer, are known explicitly. This allows for a more direct use of the expression from Eq. 7.1. Other alternative empirical relationships can also be introduced to account for the varying behavior of the partition coefficients. The conversion of monomer, for instance, is closely correlated to the size of the growing polymer particles, and could be deployed instead of the particle size to construct similar empirical correction terms to the partition coefficients, for which the parameters would have to be fitted accordingly.

The final result from the off-line parameter estimation procedure is shown in Fig. 7.5, which can be compared to Figs. 7.2 & 7.3, where the latter provides a better fit to the experiments than the former. The improvement from Fig. 7.3 to Fig. 7.5 however, is definitely worth mentioning. To clarify, Fig. 7.5 represents the entire dynamic simulation for the start-up of the reactor, for which Fig. 7.4b illustrates the spatial profile at steady-state. From this result, the steady-state behavior of the reactor model, with respect to conversion of monomer, is considered to provide a satisfying agreement with reality.

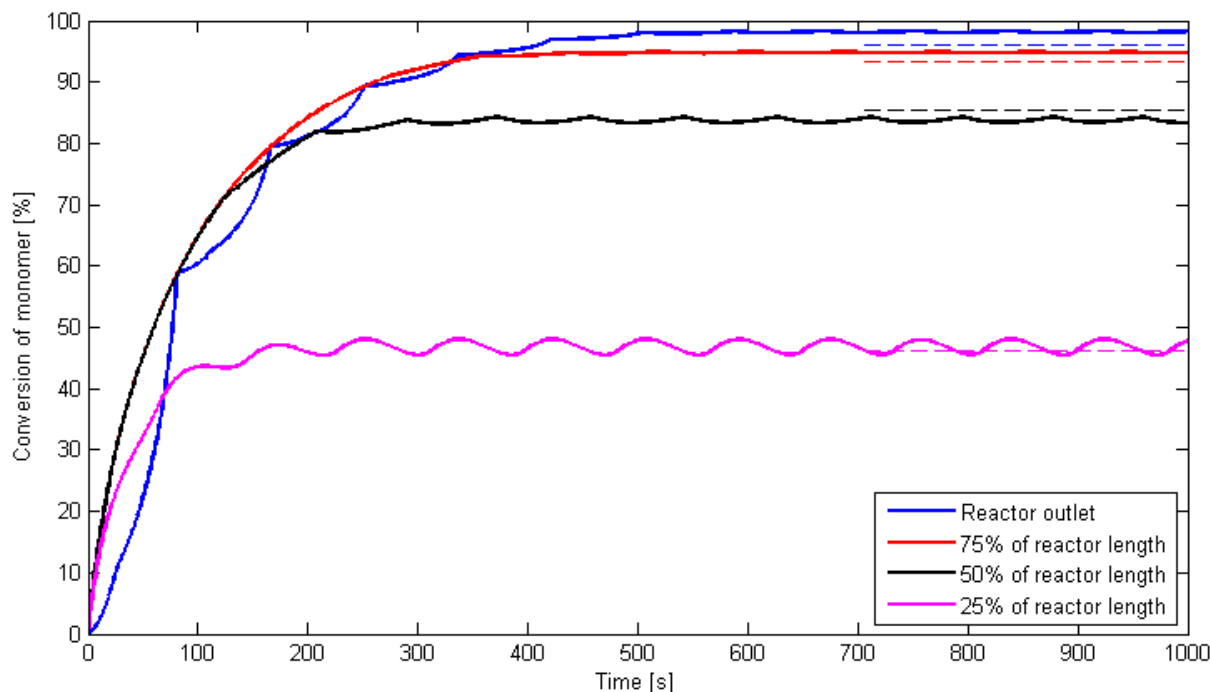


Figure 7.5: Simulation showing the conversion of monomer at four spatial locations in the tubular reactor, with optimal parameters. In this case, the solubility of monomer inside the growing polymer particles is a function of polymer particle size.

In a retrospective consideration, it would appear obvious that the rather mediocre model fit that was achieved in the specialization project preceding this thesis suffered from the effect of the constant partition coefficients [1]. In that case, the kinetic parameters of the model were adjusted to yield agreement between the model predictions and the experiments, but very few concerns were made with respect to the phase equilibrium of the emulsion system, and the partition coefficients remained untouched. As a consequence, the solubility of monomer in the particle phase was probably overestimated in the beginning and underestimated towards the end. The characteristic behavior where the conversion of monomer increases too fast in the beginning and slows down towards the end, in comparison with experimental data, was observed. From a qualitative point of view, the similarities are striking. It is emphasized that a semi-batch reactor was considered in that case, which was operated under so-called monomer starved conditions⁵.

⁵Monomer starved conditions are achievable in semi-batch reactor setups when monomer is added cautiously

There are differences between the two reactor setups, but the observed effect with respect to the conversion of monomer is found to be qualitatively similar for a tubular reactor, and adjusting the partition coefficients is believed to cause a similar effect for the semi-batch reactor.

7.3 Recursive estimation simulations: Kalman filter testing

In Sec. 5, the purpose of on-line estimation was introduced and discussed. The purpose of this section is to conduct simulations to explore the performance of the Kalman filter estimator without the intervention of the complete NMPC, i.e. without actual controller action. For a complete implementation, both the estimator and the controller will work alongside each other, but an important step along the way towards setting up the various units of the controller implementation is to test the estimator without controller action. This is sometimes referred to as recursive filtering simulations, which can be performed using the Cybernetica Modelfit tool (introduced in Sec. 4.2). The results from the simulations are somewhat related to the parameter estimation procedure, which was conducted in Sec. 7.2, in the sense that the model is altered to fit the available measurements, but emphasis is put on the distinction between off-line and on-line use. In the off-line estimation problem, a collection of parameters is modified to fulfill an optimization problem in which the difference between the predicted model outputs and the measured behavior is minimized for a specific experiment. In the on-line case, the Kalman filter will update the states (and possibly some of the parameters) of the model along the way to account for the differences between the predicted model outputs and the measured system behavior. This action will proceed in a continuous manner. The uncertainties in the various states and measurements (and possibly some of the parameters) of the system, which actually constitute the tuning of the estimator⁶, govern how aggressive the updates from the estimator to the model will be. In Fig. 7.6, a simulation is conducted without active estimation (this is sometimes referred to as a ballistic simulation). In this figure, and the following two (Figs. 7.7 & 7.8), the blue curves represent measurements while the green curves indicate the model predicted behavior.

For this specific case, an NMOL type model with few discretization points was deployed. The measurements of these simulations are fictional and generated just to demonstrate the qualitative behavior of the estimator based on the tuning. From the first simulation, there are undeniably deviations between the measurements and the model predicted behavior with respect to both conversion of monomer and reactor outlet temperature, which are the two available measurements in this case. These deviations are unavoidable in this ballistic simulation because, as mentioned, the measurements are fictional. The deviations can be reduced by including an on-line estimator, however, which is very desirable. In order to achieve this, the Kalman filter estimator is activated. For all the simulations in this section, the same white noise was used on all states, and this noise was set to be relatively small. This will, intuitively, require a low degree of state updating, but the level of certainty for the measurements will also affect the estimator action, which will be evident in the two following simulations. The action of the estimator based on the respective

along the duration of the batch. The effect is that the concentration of monomer never exceeds the amount which is accounted for by the solubility of monomer in the aqueous phase and the particle phase. As a consequence, the extent of the monomer droplet phase is negligible.

⁶The successful tuning of the estimator is, like the tuning of the controller, not trivial to achieve. In many cases, the uncertainties of the various states, measurements and parameters are hard to determine, and the process noise must be set by the modeler based on experience or based on the desired estimator performance. Some claim that estimator tuning is just as complex a task as controller tuning. Estimator tuning is discussed in Sec. 5.2.

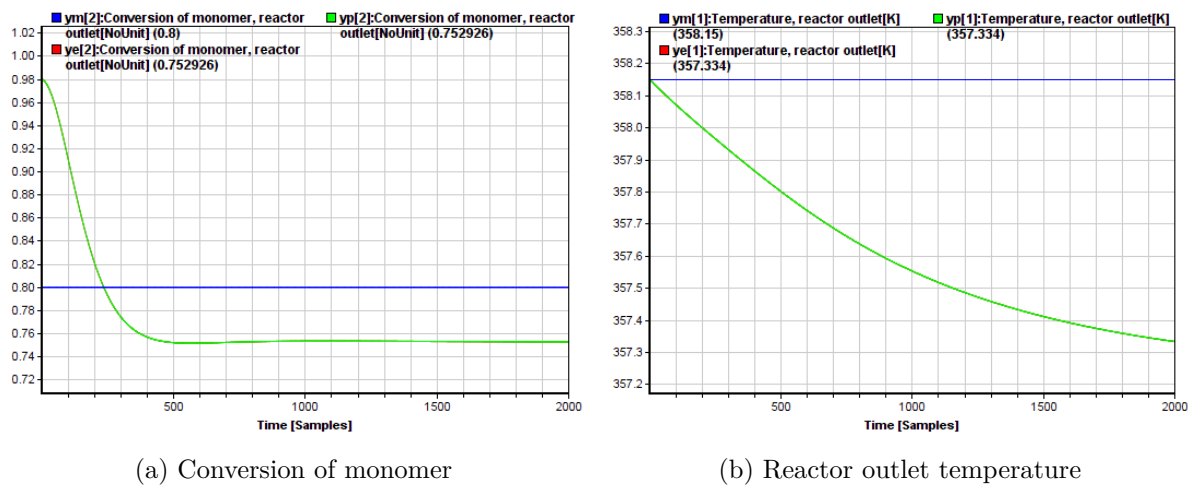


Figure 7.6: Reactor simulation, using an NMOL type model, without active on-line estimation and parameter estimation.

uncertainties for the states and the measurements are summarized in Tab. 5.1. Fig. 7.7 shows a recursive filtering simulation where the state updates are categorized as weak. The reason for the estimator to return weak state updates is that the measurements are fairly uncertain. The estimator action is visible compared to the ballistic simulation (Fig. 7.6), but the model updates from the estimator is considered to be weak because the filter is unable to effectively counteract the deviations between the measured behavior and the model predicted behavior.

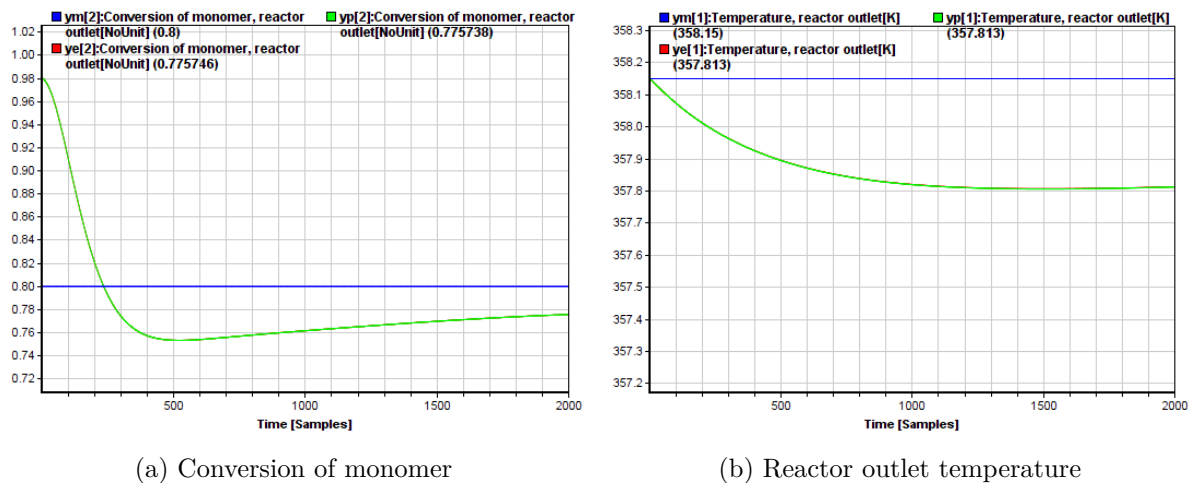


Figure 7.7: Reactor simulation, using an NMOL type model, with weak on-line estimation and parameter estimation due to moderate uncertainty in the measurements.

The final step of this short study is to investigate the qualitative behavior of the estimator when the measurement noise is very low, i.e. when the estimator is led to provide strong up-

dates. This is the case for Fig. 7.8, which shows a simulation case similar to the previous two (Figs. 7.6 & 7.7), but with strong estimator updates. The strong updates are achieved because the measurements in this case are set to have a very high certainty, and the Kalman filter adjusts the states accordingly⁷. Keeping in mind the considerations with respect to the estimator action from Tab. 5.1 in Sec. 5, all these results agree with the expected behavior. The Kalman filter estimator is, as emphatically mentioned, a crucial part of the complete NMPC implementation, and it will be investigated further in Sec. 7.6.

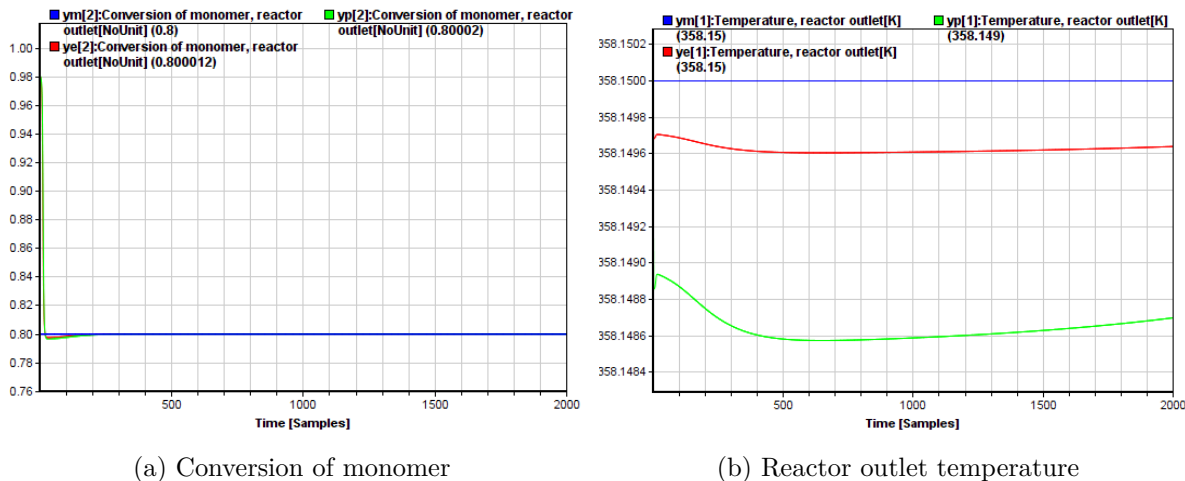


Figure 7.8: Reactor simulation, using an NMOL type model, with strong on-line estimation and parameter estimation due to high certainty in the measurements.

7.4 Control study I: Setpoint changes, conversion of monomer

The purpose of this first control study of the tubular copolymerization reactor is aimed at keeping the temperature at a desired setpoint while changing the setpoint for the conversion of monomer through the reactor. The model which is utilized in these simulations is a simplified model based on the NMOL approach, using five spatial discretization points. This is well below the required number to represent the mixing effects of the reactor in a satisfying way, as discussed for the RTD of the NMOL type reactor model in Sec. 3.3, but this rather coarse discretization proves to be useful for control purposes.

This test case uses a liquid feed containing 30 w% monomer, of which 76.5 w% is monomer type 1. This specific feed composition will also be deployed for the following cases throughout the remaining part of this thesis. It is emphasized that it is also of interest, in an industrial context, to investigate how the reactor behaves if the feed has a higher monomer content, e.g. 40 w%. The inputs (MVs) which were chosen for this specific reactor test simulation are the total mass feed rate and the cooling jacket temperature. Several other candidates for appropriate inputs exist, for instance the fraction of monomer in the feed (this option is explored in Sec. 7.8) and

⁷Observe the scaling of the ordinate axis in Fig. 7.8 to realize how close to the measurement the model predictions are adjusted! For the simulation with strong estimator action, the estimated behavior and the measured behavior are, practically speaking, inseparable.

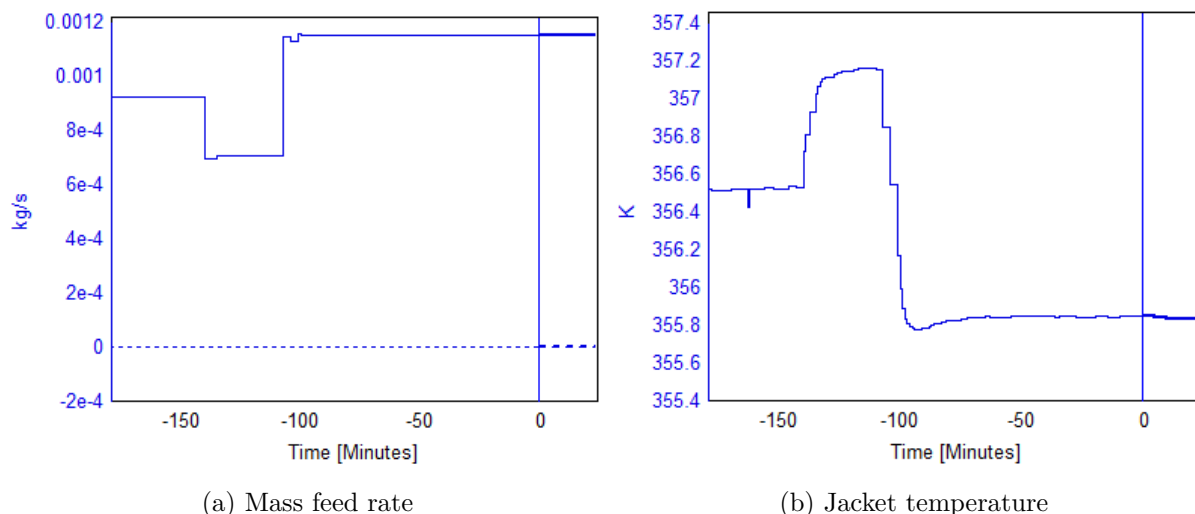


Figure 7.9: Controller action (MVs) for a tubular reactor with setpoint changes in the conversion of monomer. The setpoint for the reactor outlet temperature is kept constant.

the respective distribution of various monomer types in the feed. These quantities are treated with caution however, because altering the feeding conditions is believed to endanger the quality parameters of the polymer product⁸. It is emphasized, for this case, that while the total mass flow rate of the feed is manipulated, the respective ratios between the monomers and the other reactants of the feed are left untampered with.

In this reactor test case, the setpoint for the conversion of monomer is altered two times to investigate if the controller manages to effectively reach the new setpoint in each case. At the same time, the temperature is controlled to stay on the desired setpoint of 358.15 K. In this system, the temperature and conversion are coupled in the sense that a higher temperature will yield faster kinetics and *vice versa*, but with the chosen inputs it should be possible to control both of the outputs of interest to their respective setpoints. This coupling of the variables of the system will be investigated and discussed further in Sec. 7.8, which represents a finalization of this series of control cases. Figs. 7.9a & 7.9b show the controller action (MVs) of the process for two separate setpoint changes. The corresponding behavior of the process outputs are displayed in Fig. 7.10, where both the conversion of monomer and the reactor outlet temperature are shown. In these figures, the red dashed lines indicate the setpoints, while the solid blue lines indicate the measured process behavior. From Fig. 7.10a, it is evident that the conversion of monomer changes effectively according to the changes in the setpoint. Fig. 7.10b may lead the reader to believe that the temperature is fluctuating strongly when the setpoint in conversion of monomer is changed, but a careful examination of the ordinate axis confirms that these fluctuations are insignificant in magnitude. A remark in this context is that the temperature setpoint, which is constant throughout this simulation, is efficiently maintained. The observant reader will notice that neither of the two controlled outputs appear to reach their exact respective setpoints, and it

⁸The validity of this statement is approved by O. Naeem, Polymer Reaction Engineering, BASF. From experience, it is not desirable to tamper significantly with the composition of the feed, as this is expected to cause fluctuations in the product quality.

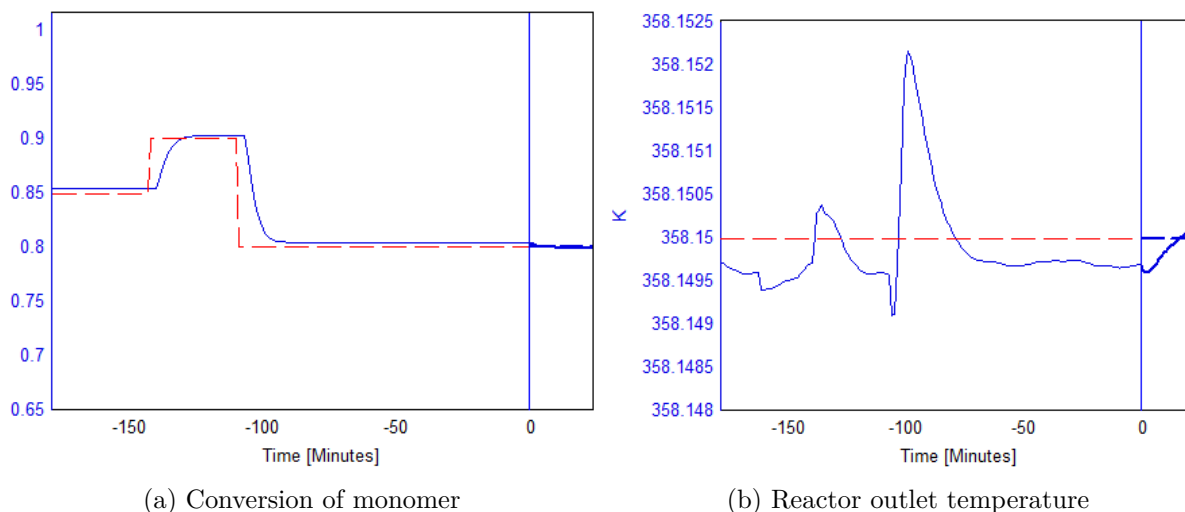


Figure 7.10: System response (CVs) for a tubular reactor with setpoint changes in the conversion of monomer. The setpoint for the reactor outlet temperature is kept constant.

seems to be a deliberate offset in each of the outputs. This effect is not due to a faulty controller, but rather a consequence of the chosen controller tuning. In this simulation, no specific concerns have been made with respect to altering the tuning parameters. Tuning of MPC is discussed in Sec. 6.3. This observed situation at steady-state operation, with the offset from the assigned setpoints, represents the optimal solution from a mathematical point of view, i.e. the solution that minimizes the cost function of the control problem, under the given conditions.

The time delay through the tubular reactor is evident when the respective setpoint changes are simulated. The reactor has an expected mean residence time of just over 10 minutes, and this agrees well with the observed time delay in the simulation results. It is important to recognize that the residence time of approximately 10 minutes is obtained from simulations using nominal variable values, and will change when the mass flow of the feed stream is manipulated, which is the case in the simulations of this section. In essence, changing the mass flow of the feed to the reactor means changing the fluid flow velocity and thus the residence time through the reactor. The longer the mixture stays in the reactor tube, the longer the chemical reactions proceed, and the resulting effect is that the mass flow of the feed can be used to control the conversion of monomer through the reactor.

7.5 Control study II: Setpoint changes, reactor temperature

The purpose of this next control study is to simulate the behavior of the system to setpoint changes in the reactor outlet temperature. The reactor model which is deployed is the same as the one that was used in Sec. 7.4, which is an NMOL type model with very few spatial discretization points. The feeding conditions are the same as the ones that were used in the previous case (Sec. 7.4). The temperature at the reactor outlet is believed to be among the quantities of the system which are readily controlled, due to the effective cooling mechanisms of the reactor setup. The reactor is assumed to be submerged in an oil bath, for which the temperature can be

rapidly and accurately adjusted⁹. The temperature of the feed entering the reactor is constant at 358.15 K, and the composition of the feed remains untampered with throughout the simulation. Fig. 7.11a shows the changes in the jacket temperature for the reactor, while Fig. 7.11b shows the actual setpoint changes and the corresponding change in the reactor outlet temperature. The control appears to be relatively rapid and smooth. The change in temperature from Fig. 7.11 is a typical first order response with respect to the characteristics of the change. Predicting and obtaining model characteristics from simulations are discussed, among others, by Skogestad [37]. It is worth mentioning that a significant branch of control engineering is based on the use of empirical models and simplified dynamic models from experimental step responses in the plant. In this thesis, first principles state space models are at the heart of the controller development, but in a different case where the models were unknown, these ideas could be deployed to identify adequate tuning etc. for the controllers. An important remark in this context is the already mentioned fact that the temperature changes behave nicely in the manner of a first order type of process.

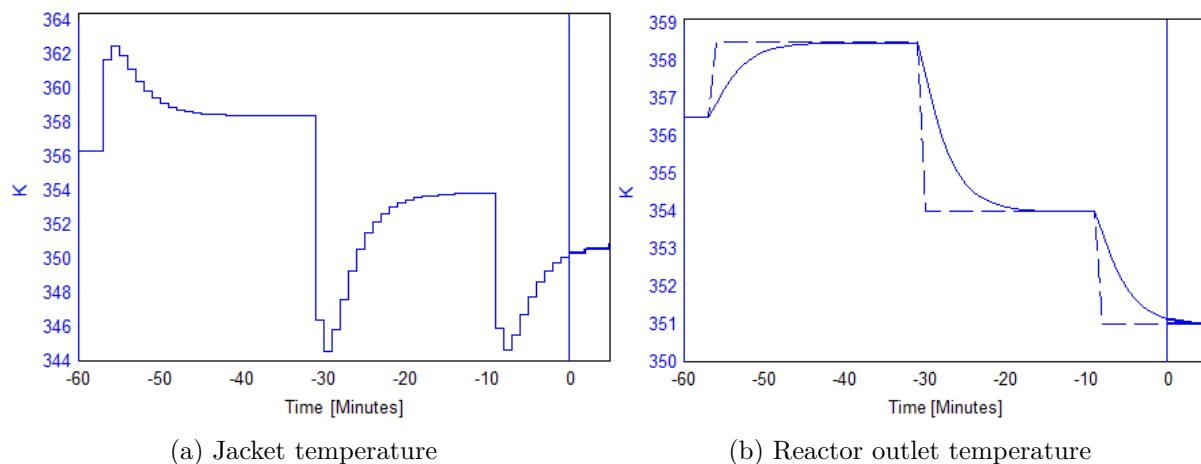


Figure 7.11: Input (MV) and output (CV) changes for a tubular reactor with setpoint changes in the reactor outlet temperature.

One unrealistic aspect of this simulation (Fig. 7.11) is the fact that the cooling jacket temperature changes several degrees in a leap when the setpoint for the reactor outlet temperature changes. Although the oil bath is assumed to have a readily controlled temperature itself, the sudden leaps do not represent reality, i.e. the physical interpretation of the system. A way to accommodate for this, which is included in the next simulation, is to include a constraint on how much the inputs can change from sample to sample. In the controller formulation in Sec. 6.1, this is introduced as Δu_k (Eq. 6.7). As explained in Sec. 6.1, the MPC allows for constraining the changes in the inputs in two different ways. On one hand, a soft constraint where the changes in the inputs are weighted into the cost function of the MPC problem is possible, while a hard constraint, on the other hand, allows for strictly limiting the magnitude of change in the respec-

⁹Please notice that the term cooling jacket is used in a wide sense, including "all" cooling mechanisms embracing the reactor tube. The oil bath is referred to as cooling jacket, although this may appear somewhat strange in a strictly physical sense.

tive inputs. In the following simulation, the latter type of constraint is deployed. The cooling jacket temperature, for instance, is limited to only change (increase or decrease) with a magnitude within a certain interval. To obey the physical representation of the system, the change in the cooling jacket temperature is limited to one degree per minute for the following simulations. It is emphasized that one does not necessarily have to choose between these two types of input constraints, and both types can easily be utilized simultaneously. In such a situation, the changes in the specific input would be restricted within a specific interval, and the magnitude of the change would pose a contribution to the cost function of the MPC. For the simulation in Fig. 7.12, the input change constraint is introduced halfway through the simulation, while the first half of the simulation is unconstrained in the context of input changes. This is evident for the first of the two setpoint changes, where the jacket temperature bursts as much as twenty degrees up over one sample, which is unrealistic for the specific oil bath cooling mechanism¹⁰. For the latter of the two setpoint changes, however, the constraint has become active, and the changes for the jacket temperature shows the characteristic "staircase" behavior. In this case, the input is saturated in each sample point along the declining staircase, meaning that the input has maximized its efforts. Emphasis is put on the fact that while the latter of the two setpoint changes (the constrained case) achieves the desired setpoint slower than the unconstrained, ideal case, this is a far more realistic representation of the systems behavior.

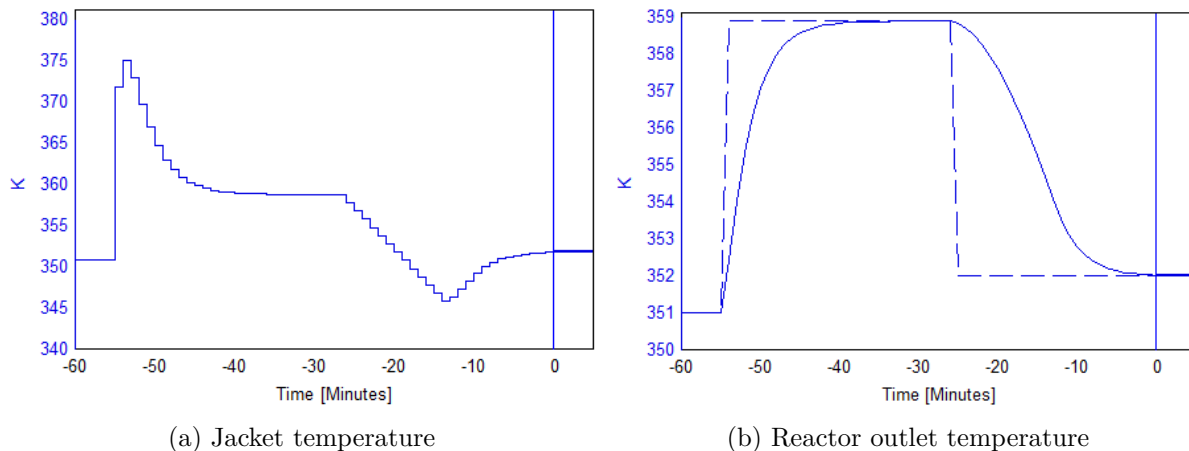


Figure 7.12: Input (MV) and output (CV) changes for a tubular reactor with setpoint changes in the reactor outlet temperature, with a constraint on the changes in the input.

As an extension to this test case, an NMOL type model with more discretization points is tested for the same use, i.e. temperature control. The purpose is to explore whether a model with a higher amount of discretization points is needed to achieve efficient process control or not, and to what extent a more detailed model may improve the performance of the controller. It is known, from the considerations regarding RTD in Sec. 3.2, that a high number of discretization points in space is required to adequately represent the mixing effects of the tubular copolymerization reactor, but in the context of temperature control, this requirement may not be that strict. One

¹⁰For other cooling mechanisms, this rapid change in cooling fluid temperature may be achievable, and the constraints must be adjusted accordingly.

could easily imagine a case where the controller and estimator algorithms are governed by a relatively simple model having few discretization points, while the plant is represented by a more complex model having a better representation of the real-life mixing effects. The idea of using separate models for controller algorithms and plant simulations is elaborated in Sec. 6.4, and in Sec. 7.6 these ideas are deployed directly to investigate the performance of an on-line estimator. The slightly more complex model, as compared to the one which was used in the simulations of Figs. 7.11 & 7.12, uses 25 discretization points in space, which is a significant increase from the previously tested models until now. A setpoint change in the reactor outlet temperature is simulated, and the resulting system behavior is shown in Fig. 7.13. The constraint on the changes in the cooling jacket temperature, which was introduced for the simulations in Fig. 7.12, is maintained for this simulation as well. The performance of the controller is more or less the same, which is evident from comparing Fig. 7.13b to the last half (i.e. the latter of the two setpoint changes) of Fig. 7.12b. This result helps to confirm what has already been suspected, namely that increasing the spatial discretization may not necessarily provide a significant improvement in the context of efficient process control. It is emphasized that this remark is based solely on the observations from the cases on temperature control, and may indeed not be a universally valid statement for all the quantities of the reactor.

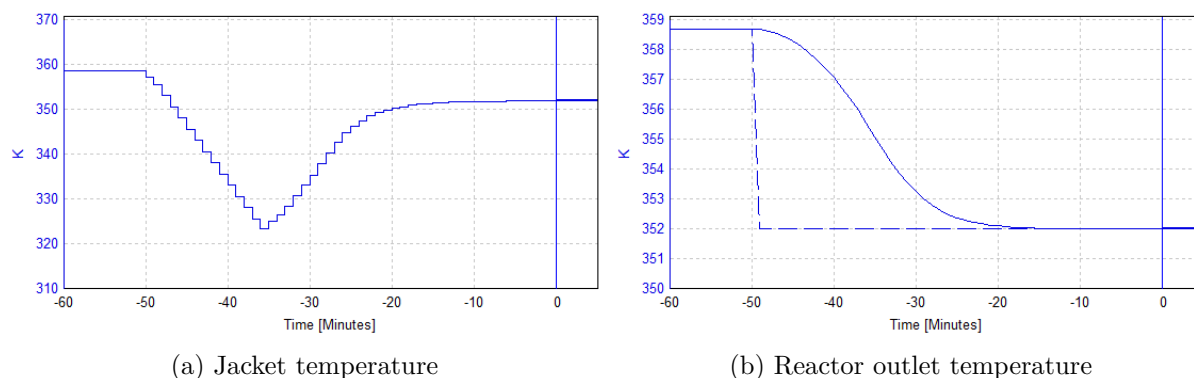


Figure 7.13: Input (MV) and output (CV) changes for a tubular reactor with setpoint changes in the reactor outlet temperature, with constrained change in input. This simulation uses a model with an increased number of spatial discretization points.

It is important to recognize that this approach, as a consequence of a finer spatial discretization, will be more computationally demanding than the previously deployed models, which were simpler. Each sample in time is found to require an average calculation time of approximately 8 seconds in the CENIT software¹¹. This is well below the sampling time interval itself, which in this case is 60 seconds (which is a quite high value for this kind of system). This represents a stamp of approval towards deploying the model in an on-line implementation, since the controller calculations is expected to converge, with a safety margin, in time before each new sample. Because of this, it is of interest to investigate if a reduced sampling time interval will yield an improved control structure with respect to reaching new setpoints efficiently, in comparison to the case where samples are drawn every 60 seconds. In Fig. 7.14, test case which is similar to the

¹¹All the calculations and simulations that were conducted as part of this thesis work were performed using a standard laptop computer.

one in Fig. 7.13 has been conducted, with the only difference being the sampling time interval, which is reduced from 60 seconds to 30 seconds. The resulting system behavior (Fig. 7.14) is hard to separate from the previous one (Fig. 7.13), which is sensible. Comparing Fig. 7.14b to Fig. 7.13b confirms that the temperature change is the same in the two cases. A comparison between Fig. 7.14a and Fig. 7.13a indicates that although the characteristic "staircase" change in the first case has more steps (due to more samples), the net change through the setpoint change is more or less identical. The reason for this is the constraint which has been put on the changes in the input, which is the same in both cases.

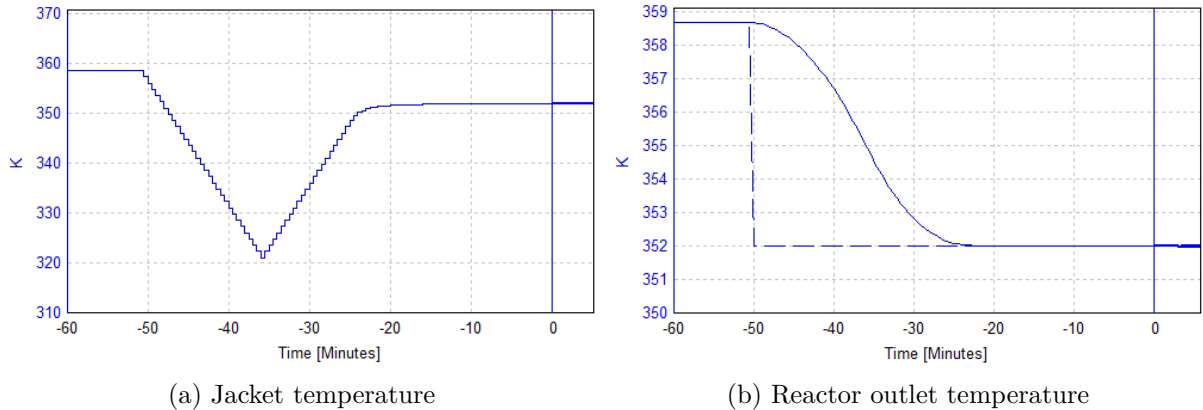


Figure 7.14: Input (MV) and output (CV) changes for a tubular reactor with setpoint change in reactor outlet temperature, with constrained change in input. This simulation uses a reduced sampling time interval.

In this new, modified case, where the sampling time interval has been reduced from 60 seconds to 30 seconds, the computational time becomes slightly larger due to more calculations, but it does not endanger the validity of the model for on-line use. One sample calculation in the CENIT software requires approximately 11 seconds in this case, on average, as compared to 8 seconds in the case with poorer discretization. An important remark in the context of computational time is based on the observation that the efficiency of the control does not seem to increase much when the degree of spatial discretization increases, yet the computational time increases significantly. In the crucial trade-off between computational time and model performance, these observations encourage the use of models with relatively poor spatial discretizations.

The required time to conduct the change in temperature is quite long compared to the mean residence time of the reactor. One of the main reasons for the long time required to change the temperature is the constraint which is put on changing the cooling fluid temperature. As discussed before, the idea of changing the cooling fluid temperature significantly in a very short matter of time (in a large leap) is unrealistic, and the maximum change in temperature of the cooling fluid should be limited to a few degrees per minute¹², maximum. This constraint is based on the physical interpretation of the system, since the reactor tube is submerged in an oil bath. The resulting behavior is that a setpoint change in the bath temperature of about 8

¹²In the experimental smart-scale reactor at TU Hamburg, from which most of the experimental data to this thesis is obtained, the temperature of the cooling fluid is rarely controlled at all, mainly because this is hard to accomplish. For the purpose of this thesis, the cooling fluid temperature is assumed to be controllable, nevertheless.

degrees will take fairly long time (which is evident from the simulations), while smaller changes will be effectuated more rapidly. A quick recapitulation of Fig. 7.11, for instance, illustrates how effective the temperature control potentially can be when the cooling fluid is allowed to change its temperature dramatically over a short period of time. In such cases, the sampling time interval is desired to be as short as possible, still staying above sensible limits, to encapture the rapid changes.

7.6 Control study III: On-line estimator testing

The purpose of this section is to explore the role of the on-line estimator. In a sense, the investigations in this section are very similar to what was done in Sec. 7.3, where the Kalman filter estimator was first tested, and it represents, chronologically speaking, a step back from the control cases in Secs. 7.4 & 7.5. The main difference between Sec. 7.3 and this one, however, is that while the former considers the so-called off-line recursive simulations where the Kalman filter estimator is tuned before being deployed in on-line applications, the latter considers the actual performance of the estimator in an on-line environment where the controller is active as well. The purpose is, in other words, to illustrate the role played by the on-line estimator as part of the block diagram in Fig. 6.2 in agreement with the theory of Sec. 5.

The tubular reactor which is considered in this case has two outputs (CVs) of interest, which are the reactor outlet temperature and the conversion of monomer through the reactor. For these simulations, a setpoint change in the reactor outlet temperature is introduced, and the role of the on-line estimator throughout response to the setpoint change is investigated. In order to construct a situation where the need for an on-line estimator is evident¹³, the ideas of a plant replacement model from Sec. 6.4 are deployed. It would be sufficient to deploy two identical models and alter some of the parameters in one of them, for instance in the simulator model, to achieve the desired discrepancy between the controller model and the representation of the plant. In this case, however, the controller uses an NMOL type model with few (5) discrete points in space to achieve numerical efficiency, like the one used in Figs. 7.11 & 7.12, while the simulator has more discrete points in space (25) to achieve a decent model accuracy, like the one used in Fig. 7.14. In addition to this, the main parameter governing external heat transfer, i.e. the heat transfer coefficient through the reactor wall, has been adjusted in the plant replacement model, to yield a slightly slower heat transfer. The consequence is that the model predictions which are calculated by the controller will deviate from the simulated behavior of the plant replacement model. This is evident from Fig. 7.15, where the estimator state updates are weak due to a very defensive tuning of the estimator. Specifically, the uncertainties of the measurements are set to have fairly high values. Estimator tuning is discussed briefly in Sec. 5.2, where Tab. 5.1 summarizes the typical action of the estimator based on the tuning. In the following figures, the solid blue curves represent the measured process behavior from the plant, while the red curves represent the internally calculated variables of the controller, i.e. the model predictions. The dashed red line is the *a priori* model prediction while the dotted red line is the *a posteriori* prediction where the measurements are taken into account by the estimator. It is crucial to realize that while the blue lines provide the actual behavior of the system, the red lines are the

¹³In the event where both the controller algorithms and the process simulator use the same model, the estimator action is not needed at all, as the controller will be able to predict the exact behavior of the plant without the need of being corrected based on measured plant behavior. The fact that no process model can represent a real process in an absolute sense implies that the use of an estimator will never be excessive in model-based control.

information that the controller assumes for the internal calculations, e.g. the minimization of the cost function. One could say, essentially, that if the model predictions (the red curves in Figs. 7.15 & 7.16) obey the setpoints of the controller, the controller will not strive to further manipulate the inputs of the system although the actual system (indicated by blue lines) has not reached the desired setpoint. As a consequence, it is important to ensure agreement between the two to ensure the desired action of the controller, and this is the purpose of the on-line estimator. As a matter of fact, the two red lines coincide in this first case (Fig. 7.15), on account of the tuning of the estimator.

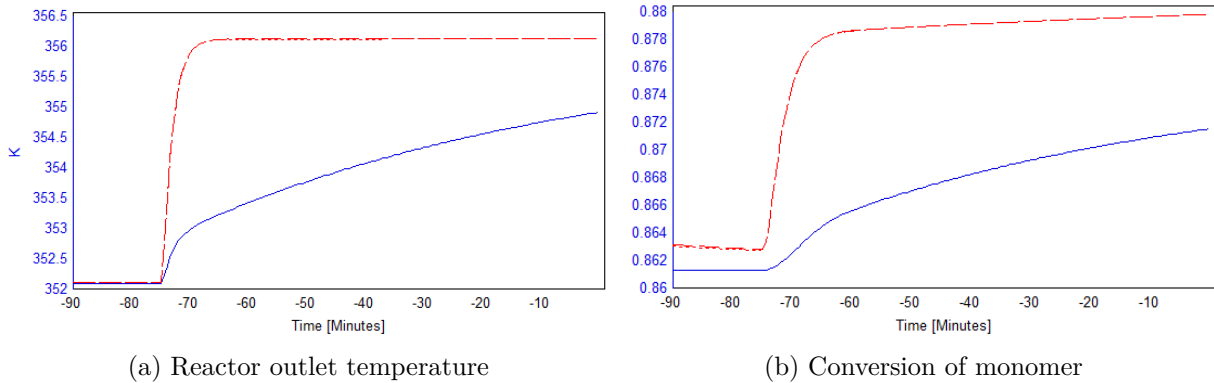


Figure 7.15: Simulated outputs (CVs) for a setpoint change in the reactor temperature with visible CV estimates and plant measurements. The estimator is tuned to deploy weak state updates.

Fig. 7.16 shows a quite different situation, although it is the same step in the setpoint (of the reactor outlet temperature) which is conducted. In this case, the estimator is tuned to be significantly more aggressive with respect to the state updates than what was the case in Fig. 7.15. As a consequence, the setpoint change for the reactor outlet temperature is obeyed much more effectively, and the agreement between the model predictions of the controller algorithm and the measured plant behavior is significantly improved.

The difference between Fig. 7.15 and Fig. 7.16 illustrates the very essence of having an on-line estimator. Although the respective behavior of the blue and red lines in Fig. 7.15 agree from a qualitative point of view, the plant (i.e. the blue lines) is significantly slower than the predictions of the controller (the red lines). Because the estimator is tuned in a defensive way, the resulting controller action is consequently much more defensive than required to reach the setpoint in an effective manner. In a way, one could say that the controller does not even know that it is acting too defensive, because the updating of information from the plant measurements, i.e. the estimator action, is too weak.

Emphasis is put on the fact that the simulations of this section are purely aimed at demonstrating the effect of the estimator. The simulated steps for the setpoint of the reactor outlet temperature are conducted without any specific concerns towards the controller action itself, apart from the concluding remark that the effectiveness of the controller action is significantly improved when the estimator is tuned appropriately. The behavior and effect of a well-functioning estimator is of great importance for the simulations in Sec. 7.8, among others. In such control

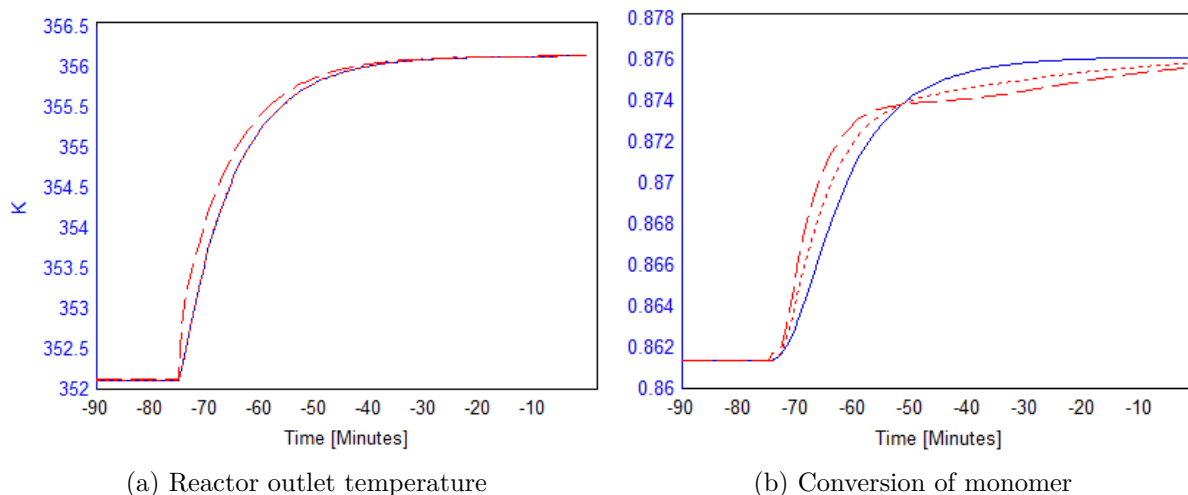


Figure 7.16: Simulated outputs (CVs) for a setpoint change for the reactor temperature, showing visible CV estimates and plant measurements. The estimator is tuned to deploy strong state updates.

cases, the performance of the controller is at the risk of being severely compromised without the work of the estimator¹⁴, as duly mentioned before.

7.7 Control study IV: Temperature changes with a segmented cooling jacket

The simulations presented in this section represent an extension to the cases in Secs. 7.4 & 7.5 in the sense that the model of the cooling jacket has been improved. This improvement has been introduced to achieve a more realistic representation of the reactor cooling system with respect to typical industrial polymer reactors. The cooling oil bath, which is deployed for lab-scale experiments of the smart-scale tubular reactor, is well represented by a single controllable cooling fluid temperature for the entire reactor tube, but having several segments of the cooling jacket covering their respective parts of the reactor tube, with individually controllable cooling fluid temperatures, is believed to yield more effective temperature control and a more realistic representation of an industrial case. This idea is illustrated in Fig. 7.17, where a tubular reactor is drawn with two separate cooling jackets covering one half of the reactor each. In principle, it would be possible to imagine that one half of the reactor is being heated with steam while the other is cooled with cold water, for instance. The model which has is deployed in the following simulations is an NMOL type model with ten points of spatial discretization. This is quite similar to what was tested in Sec. 7.5, where NMOL type models with both 5 and 25 steps were tested.

It would be possible to imagine a tubular reactor having a cooling jacket which is segmented into more than just two parts as well, e.g. five consecutive segments with their respective fluid

¹⁴In a humorous and somewhat far-fetched consideration, it would be possible to compare the teamwork of the controller and the estimator to the brilliant 60's pop duo Simon & Garfunkel. In such a comparison, the celebrated, self-declared leader (Simon) would be the controller, while the role of the ever so important, yet perhaps not that prominent part (Garfunkel) would be played by the estimator. Without either, the function of the duo would be compromised, and getting the two to cooperate properly is crucial.

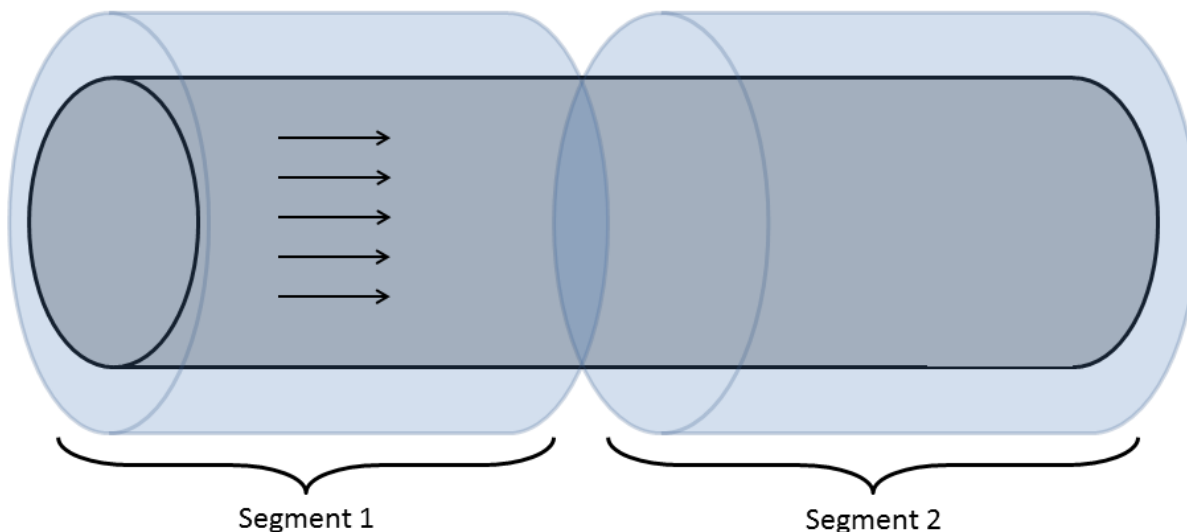


Figure 7.17: A drawing of a tubular reactor with a cooling jacket which is segmented into two individual segments. The temperatures of the cooling fluids in the respective segments are controllable.

temperatures representing a degree of freedom. An issue in such a situation is the desire to have a corresponding number of inputs (MVs) and outputs (CVs) for the system. As will be demonstrated, using two separate cooling jacket segments to control the reactor outlet temperature and the conversion of monomer (the kinetics of the chemical reactions are temperature dependent) constitutes a typical 2x2-type control structure. In such a control system, two inputs are used to control two outputs, but it is emphasized that the control is not conducted in a 1-to-1 manner where the variables are decoupled, but in a coupled manner using mathematical optimization based on the model predictions. As a consequence, it is desired to have a certain sensitivity between the inputs and the outputs of the system to achieve effective control. To emphasize this idea, it would be close to impossible, hypothetically speaking, to efficiently control the residence time distribution using only the respective temperatures of the cooling jacket segments. The reason for this is that variations in the reactor temperature, within the safe and sensible limits, only cause negligible contributions toward the fluid flow conditions of the reactor¹⁵.

In Fig. 7.18, the behavior of the system is shown for separate setpoint changes in the reactor outlet temperature and the conversion of monomer through the reactor. These simulations are, in other words, very similar to the case in Sec. 7.4. Instead of controlling the conversion of monomer by manipulating the feed conditions however, the feed is left untampered with, and two separate degrees of freedom for the cooling jacket are used to control both the conversion of monomer and the reactor outlet temperature. It is desirable to investigate if this strategy appears to be effective.

In the following figures, the red dashed and the red dotted lines indicate the information that

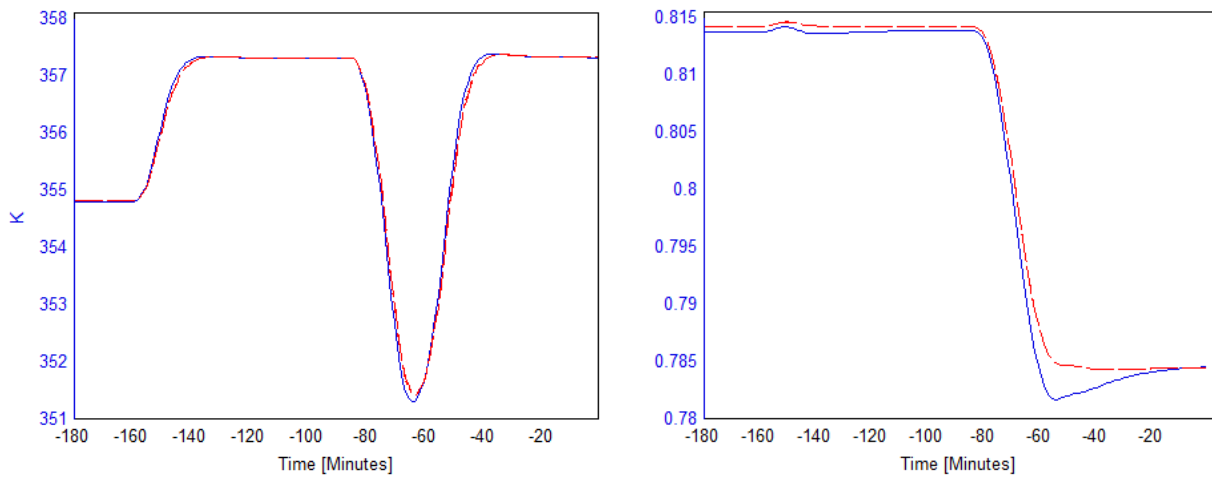
¹⁵The temperature will affect the density of the reaction mixture, both directly and indirectly through the progression of the chemical reactions. This will in turn affect the fluid flow conditions, but for the operation conditions of the specific chemical system considered in this thesis, these effects are not evident.

is available to the estimator and the controller calculations when it comes to the outputs (CVs). The blue lines indicate the measured plant behavior, which is provided by the process simulator model, on which the estimator determines the appropriate state and parameter updates to the model (recall Fig. 5.1). For the inputs (MVs), which in this case are the respective cooling fluid temperatures of the two segments of the cooling jacket, the blue lines indicate segment number one (the first half of the reactor) while the red lines indicate segment number two (the second half of the reactor). In order to obtain a high conversion of monomer, which is usually desirable, the temperature should be as high as possible through the reactor to speed up the copolymerization reactions (recall the temperature dependent kinetics that were introduced in Sec. 2.1), but it is important to stay within the limits of safe operation. As discussed in previous parts of the thesis (Sec. 1), elevated temperatures can cause hotspots and run-away situations in the reactor, which severely endanger the safe operation of the reactor and the entire plant. As a consequence, the constraints in the controller system has been set with caution. In addition, the cooling jacket temperatures are constrained to only change one degree per minute, which gives the characteristic staircase change in the input which was first observed in Fig. 7.12. Without this constraint, the setpoint changes is expected to be obeyed even faster, but it is emphasized that this constraint is adjusted to construct a case which is realistic in the sense that some cooling systems are unable to change the cooling temperature in a leap, as previously discussed (Sec. 7.5).

The expected behavior of the two inputs in this system is that, although the variables are coupled, cooling segment number 1 will largely act to control the conversion of monomer, in the sense that it heats¹⁶ the first half of the reactor length up, in a more aggressive manner than usual, to ensure that the propagation reactions of the reactor proceed as fast as desired, while cooling segment number two acts to ensure that the desired reactor outlet temperature is reached by cooling the last half of the reactor accordingly. This behavior is observed for both temperature setpoints in Fig. 7.18c, where the blue curve is definitely above the red curve. An interesting effect to observe from this first simulation is that when the setpoint for the conversion of monomer is reduced, the need to heat the reactor intensely during the first half is not required, and cooling segment number one is actually colder than cooling segment number two. Another observation from Fig. 7.18 is that the discrepancies between the measured plant behavior and the predictions made by the controller are small, which is evident from comparing the blue curves to the red curves. This is achieved by deploying almost identical models for the respective purposes of estimation and control and plant replacement, which eases the work of the on-line estimator (recall Sec. 7.6).

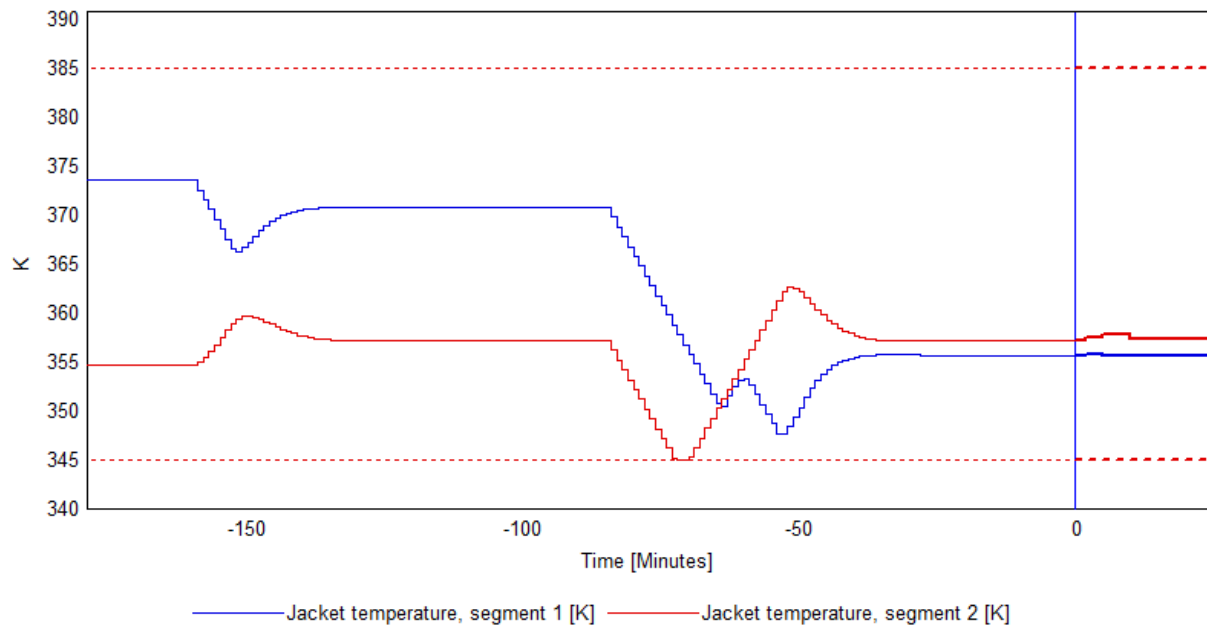
Fig. 7.19 shows the same steps in the setpoints for the reactor outlet temperature and the conversion of monomer through the reactor as the previous simulation (Fig. 7.18) does, but the respective weights for setpoint violation have been modified. The motivation for changing the weights is the observed flight from setpoint for the reactor outlet temperature (Fig. 7.18a) when the setpoint for the conversion of monomer is changed. The weights for violating deviations in the setpoints are the entries of the Q -matrix from Eq. 6.16 in Sec. 6.1. A higher (relative) weight for a CV will yield a higher contribution of the specific corresponding deviation towards the cost function, and the controller will strive to prioritize the reduction of the deviation of this variable rather than the other output variable, for which the respective deviations weigh less

¹⁶Once again, the term cooling jacket is somewhat misleading. For most of the modes of operation for the reactor which are considered in this thesis, the cooling jacket will indeed act to cool the reactor, but there are also situations where the cooling jacket will be warmer than the reactor itself, and hence work as a heating device.



(a) Reactor outlet temperature

(b) Conversion of monomer



(c) Cooling jacket temperatures

Figure 7.18: Simulation results showing inputs (MVs) and outputs (CVs) for a tubular reactor with a segmented cooling jacket.

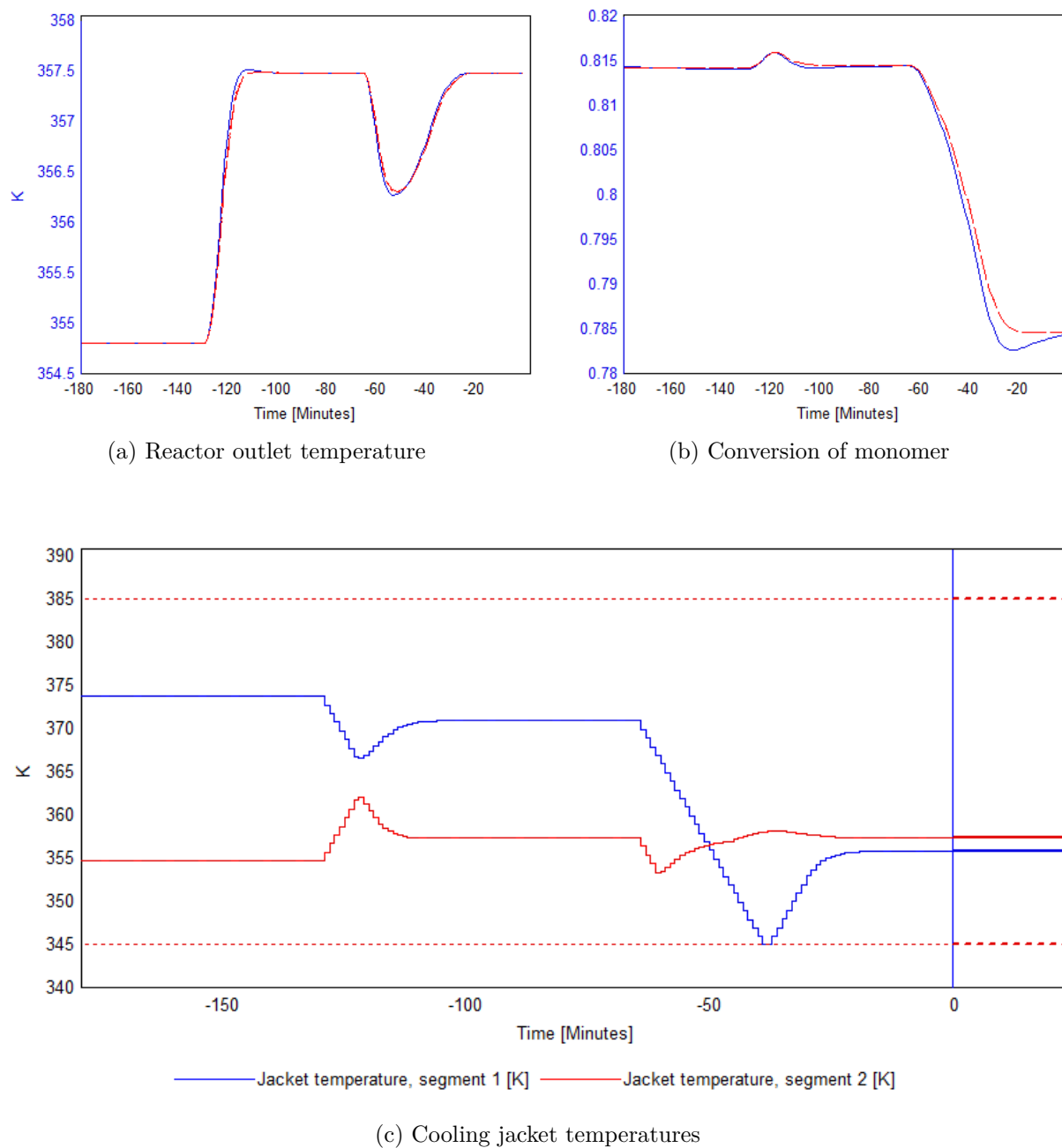


Figure 7.19: Simulation results showing inputs (MVs) and outputs (CVs) for a tubular reactor with a segmented cooling jacket. The tuning parameters of the controller are altered in comparison with Fig. 7.18.

towards the cost function. The resulting behavior of the system in this simulation has several qualitative similarities to the simulation from Fig. 7.18. At the nominal conversion of monomer, the first half of the reactor is heated more intensely than the second half, but this tendency is changed when the setpoint for the conversion of monomer is reduced. In addition, the changes in the respective setpoints are obeyed in a manner which is, qualitatively speaking, similar in Fig. 7.18 and Fig. 7.19. At first glance, there are not many effects from the changing of the tuning parameters, but there are certain subtle improvements to address.

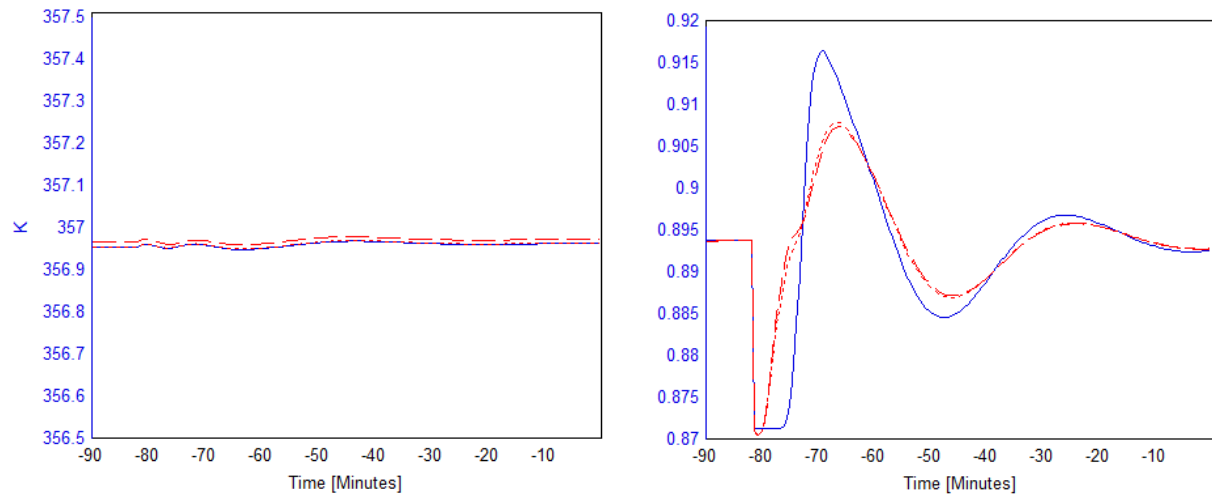
The perhaps most obvious effect after the changes in the weights have been done is that the setpoint change in the conversion of monomer is handled significantly better with respect to the reactor outlet temperature. The conversion of monomer is virtually the same in both cases, which is evident from comparing Fig. 7.18b to Fig. 7.19b. The comparison of Figs. 7.18a & 7.19a tells a different story, however. The latter has a far better ability to keep the reactor outlet temperature at the desired setpoint when the setpoint for the conversion of monomer changes. It is an inescapable fact that the reactor outlet temperature has a significant undesirable change when the setpoint for the conversion of monomer changes, which the changes in the tuning parameters were unable to eliminate completely. The change in the reactor outlet temperature has the characteristic behavior of a typical disturbance in both of the simulated cases, and one possible action would be to weigh the violation of the setpoint for the reactor outlet temperature even heavier to try to avoid this. The change in the setpoint for the conversion of monomer does not, as mentioned, seem to be greatly affected by the changes in the weights, and comparing Fig. 7.18b to Fig. 7.19b does not reveal any significant difference.

Tab. 7.2 provides the controller weights which were used to punish setpoint violations for the respective outputs (CVs) in the two previous simulations, i.e. Figs. 7.18 & 7.19. The weights were not subject to optimization or in-depth studies in this case, but were chosen in a diametrically opposite manner to illustrate the qualitative effects from changing the tuning parameters for the two cases.

Table 7.2: Setpoint violation weights for 2x2-type control of a tubular reactor with a segmented cooling jacket.

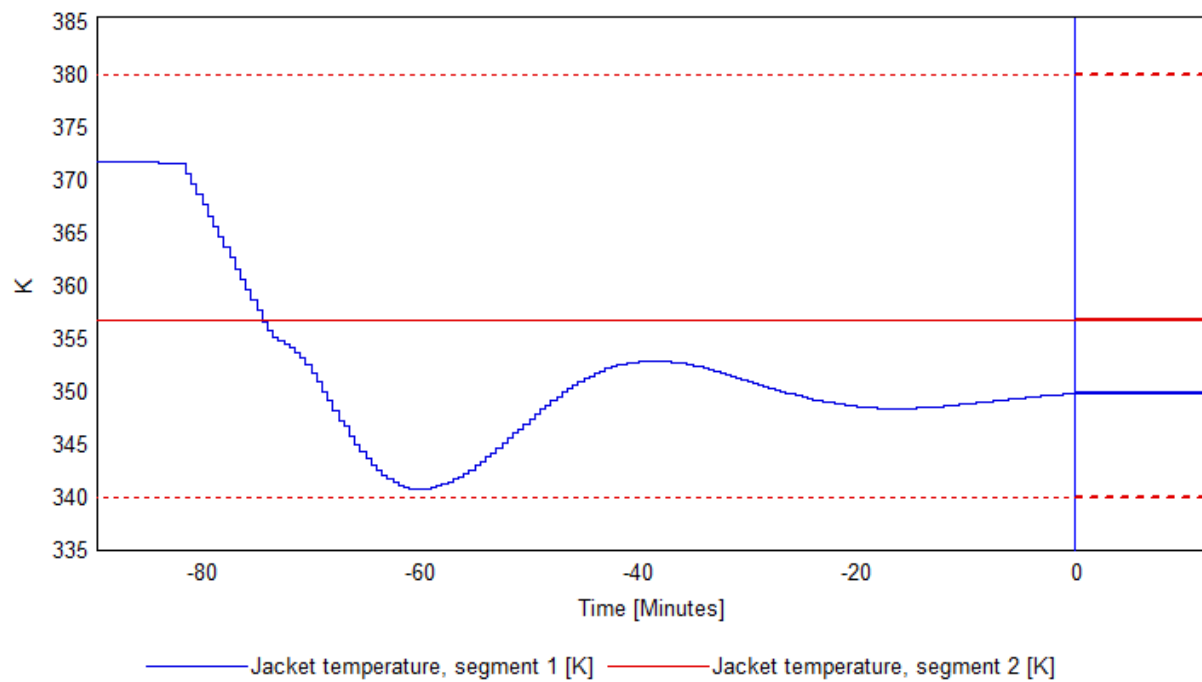
	Weight, reactor outlet temperature	Weight, conversion of monomer
Case 1 (Fig. 7.18)	1	20
Case 2 (Fig. 7.19)	12	1

As a final simulation for the tubular reactor with the segmented cooling jacket, the systems ability to reject a disturbance is investigated. In Fig. 7.20, a disturbance is introduced for the reactant feed to the reactor, and the corresponding controller action is simulated. Specifically, the monomer content of the feed is reduced with about 15%. Similar to the simulation in Fig. 7.19, the first segment of the cooling jacket is responsible most of the work, while the



(a) Reactor outlet temperature

(b) Conversion of monomer



(c) Cooling jacket temperatures

Figure 7.20: Simulation investigating disturbance rejection for a tubular reactor with a segmented cooling jacket. The reactant feed to the reactor is disturbed while the setpoints are unadjusted.

latter of the two segments is rather passive with respect to changing the temperature. For the last simulation (Fig. 7.20), this effect is particularly evident, where the temperature of the last cooling segment (the red curve in Fig. 7.20c) remains virtually unchanged throughout the simulation. This is in agreement with the observed behavior of the reactor outlet temperature (Fig. 7.20a), for which the effects of the disturbance are negligible, and the setpoint is obeyed in a satisfying manner throughout the simulation. The effects of the disturbance on the conversion of monomer (Fig. 7.20b) are more evident than for the reactor outlet temperature, and the inverse response from altering the feed conditions is visible. This simulation is yet another indication that the time delay of the reactor is hard to accommodate in a swift manner, but the conversion is returned to the setpoint eventually. A fine tuning of the controller is believed to cause a smoother trajectory with less evident overshoots, but the setpoint is then expected to be reached even slower. Conversely, the controller can be tuned to reach the setpoint faster, but with a more oscillatory behavior. In either case, the setpoint will be regained after the effects of the disturbance. At this point, it is valuable to recall that a feature of the NMPC is feedforward action. This effect is associated with the controller's ability to "predict the future" in the sense that the model will suggest future trajectories for the outputs, based on the inputs, including the disturbances. The effect of a disturbance can, in other words, be effectively counteracted if an adequate model governs the controller action. As a concluding remark to this simulation, the control structure with the segmented cooling jacket appears to perform well with respect to rejection of disturbances.

In conclusion, the use of a segmented cooling jacket appears to be an interesting strategy to control a tubular copolymerization reactor. This statement arises from the fact that most aspects of the reactor, e.g. the kinetics, are strongly dependent on the temperature profile of the reactor, which is observed for all the simulations. This means that being able to control the temperature effectively for the cooling mechanism, possibly with several separate degrees of freedom, is an effective way to control the basic properties of interest for the reactor in total. This is also believed to enhance the operation of the reactor in the context of safety, because the generation of hotspots can be counteracted more effectively. At this point, it is important to realize that hotspots may be difficult to discover without extensive use of temperature measurements along the reactor. It is important to tune the controller correctly such that the various segments of the cooling jacket cooperate effectively. In addition, the cooling jacket temperatures must be constrained in a sensible manner such that the controller does not suggest control actions which endanger the safe operation of the plant to achieve the desired setpoints for the outputs. In theory, there are no specific limitations to how many segments the cooling jacket can have, but there must be a match between the available measurements, the variables to manipulate and the variables to control, as mentioned. In practice, the physical design of the cooling jacket will limit the possible number of separate cooling segments, but for an industrial-scale reactor, this number can supposedly be significantly larger than just two.

7.8 Final control study: Control using a plant replacement model

This section contains the final control studies for the tubular polymerization reactor which is considered in this thesis. The two main quantities of interest to control are, as before, the reactor outlet temperature and the conversion of monomer through the reactor. Since two separate modeling approaches have been considered in the thesis, these are considered in turn for the purpose of on-line control. To make further use of the concept of plant replacement models

(from Sec. 6.4), each model type has been tested by deploying it to control a process which is governed by the other modeling approach, and *vice versa*. In Sec. 7.8.1, an NMOL type model (introduced in Sec. 3.3) is used to control a process which is represented by an MCV type model (introduced in Sec. 7.8.2). Conversely, this situation is rearranged in Sec. 7.8.2 such that an MCV type model is deployed in the controller while the plant behavior is governed by an NMOL type model. At this point, it is important to emphasize that while both models have had their parameters adjusted to fit an experiment, they are not fitted against each other in an optimal sense, and discrepancies between the two models may exist on account of the parameters of the models as well as the differences in the methods which have been used in the modeling work.

In the previous control studies of this thesis, the two outputs of interest (the reactor outlet temperature and the conversion of monomer through reactor) have mainly been controlled individually in separate cases without the interconnection of the two. For this final step, it is of interest to investigate a multi-variable case where they are considered simultaneously. The ready achievement of effective multi-variable control is one of the benefits from using MPC over other other types of advanced process control, and it is desirable to investigate this using a so-called 2x2-type control structure. The cooling jacket temperature remains an interesting candidate for the temperature control, and for the second input the weight fraction of monomer in the feed stream was chosen. This is different from what was done in Sec. 7.4, where the conversion of monomer was controlled by manipulating the mass flow rate of the feed stream. In that case, the conversion of monomer was controlled by manipulating the residence time of the reactor, essentially, while the following simulations uses a constant mass feed, hence it has a (close to) constant¹⁷ flow velocity through the reactor. The conversion of monomer is, in other words, not controlled by (indirectly) adjusting the residence time, but rather by carefully adjusting the amount of monomer which is fed to the reactor.

It is important to recognize that the inputs and outputs are coupled in the sense that changing one input (while keeping the other constant) may affect both outputs. A qualitative process matrix, relating the sensitivity of the inputs (MVs) and the outputs (CVs) of the process, is given in Tab. 7.3. The purpose of this array is to identify in what direction the respective outputs are expected to change when an increase is made in a specific input¹⁸. Increasing the cooling jacket temperature, for instance, yields an increase in both the reactor outlet temperature and the conversion in monomer, which is indicated by plus signs in the respective entries. The qualitative characteristics of the system are based on observations from previous simulations and the fact that the kinetics of the chemical reactions, which were introduced in Sec. 2.1, are strongly dependent on the reactor temperature.

In the following simulations, the coloring of the curves are adapted from Sec. 7.6. The red lines (dotted and dashed) represent the information on which the estimator and controller make their decisions, while the blue curves are the plant behavior as simulated by the plant replacement model based on the controller action.

¹⁷As the copolymerization reactions proceed, the density of the reactor mixture will change. This will affect the fluid flow velocity slightly, but for the system considered in this thesis, the changes in density are close to negligible.

¹⁸The parentheses in the array indicate a relatively weak dependence.

Table 7.3: Qualitative sensitivity matrix for the 2x2-type control system.

	Reactor outlet temperature	Conversion of monomer
Cooling jacket temperature	+	+
Weight fraction, monomer in feed	(+)	-

7.8.1 Controlling an MCV type process with an NMOL type model

The first case to investigate is to let an NMOL type process model govern the controller block of the NMPC, based on plant measurements provided by an MCV type model (recall Fig. 6.5). The system of mathematical models for this case is constructed by deploying an NMOL type model with 10 discrete points in space, as introduced in Sec. 3.3 and deployed for on-line control in Secs. 7.4, 7.5 & 7.7. This (rather low) amount of spatial discretization points may not, according to the discoveries in Sec. 3.3, be sufficient to completely represent the mixing effects of the reactor, but simulations in previous sections (Secs. 7.4 - 7.6) indicate that a fairly low amount of discrete points in space is adequate in the context of process control. This NMOL model will, in other words, be well suited to handle the controller and estimator algorithms of the system (recall Fig. 6.5). The other model which is deployed is an MCV type model with 15 separate moving units. This model was first introduced in Sec. 3.5, and validated under the off-line parameter estimation procedure in Sec. 7.2. As a consequence, this model is believed to represent reality in a satisfying way, and hence constitute an adequate candidate for playing the plant replacement role, as discussed in Sec. 6.4.

Fig. 7.21 illustrates how a simple temperature change is conducted for the reactor. In this first simulation, the conversion of monomer is not controlled, and the reactor outlet temperature has all the attention. The resulting controller action shows slight tendencies towards overshooting in the controlled variable (Fig. 7.21b), which is a different behavior than what was observed for the simplified temperature control cases from Sec. 7.5. This issue is believed to be resolved by fine tuning of the controller, possibly through altering the tuning of the estimator specifically. It is emphasized that while a different tuning is expected to yield a smoother control for the setpoint change, it is also expected to give a slower control. To demonstrate this, a simulation is shown in Fig. 7.22, where the tuning parameters are changed to give a far smoother setpoint change in the reactor outlet temperature, with no overshoot. All in all, the temperature control for this case is considered to be effective and satisfyingly rapid in comparison to the previous cases.

The next step in this study is to enforce a setpoint for the conversion of monomer as well, in order to explore the 2x2-type control structure from Tab. 7.3. In Fig. 7.23, the inputs (MVs) and outputs (CVs) of the system are presented for a setpoint change in the reactor outlet temperature. An interesting observation from this simulation is how the conversion of monomer, which is set to be controlled to a specific setpoint, changes slightly when the temperature changes, but is efficiently controlled back to the setpoint. This change is expected, because the chain propagation

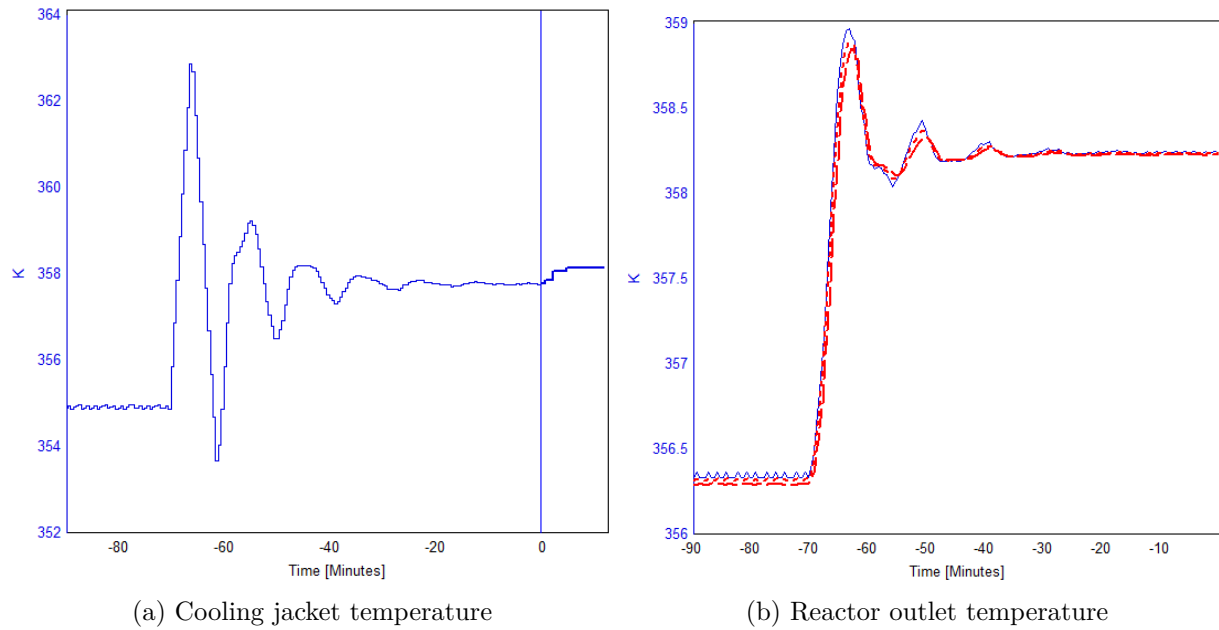


Figure 7.21: Input (MV) and output (CV) changes for a tubular reactor with a setpoint change in the reactor outlet temperature. The conversion of monomer is not controlled.

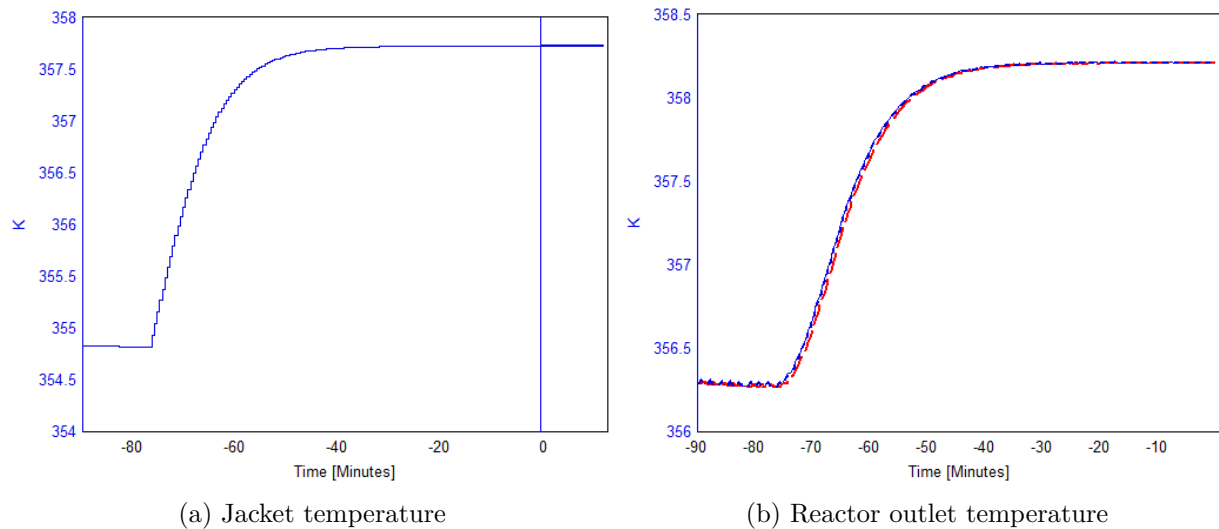


Figure 7.22: Input (MV) and output (CV) changes for a tubular reactor with a setpoint change in the reactor outlet temperature. The tuning parameters are altered to demonstrate a smoother control trajectory than the case in Fig. 7.21.

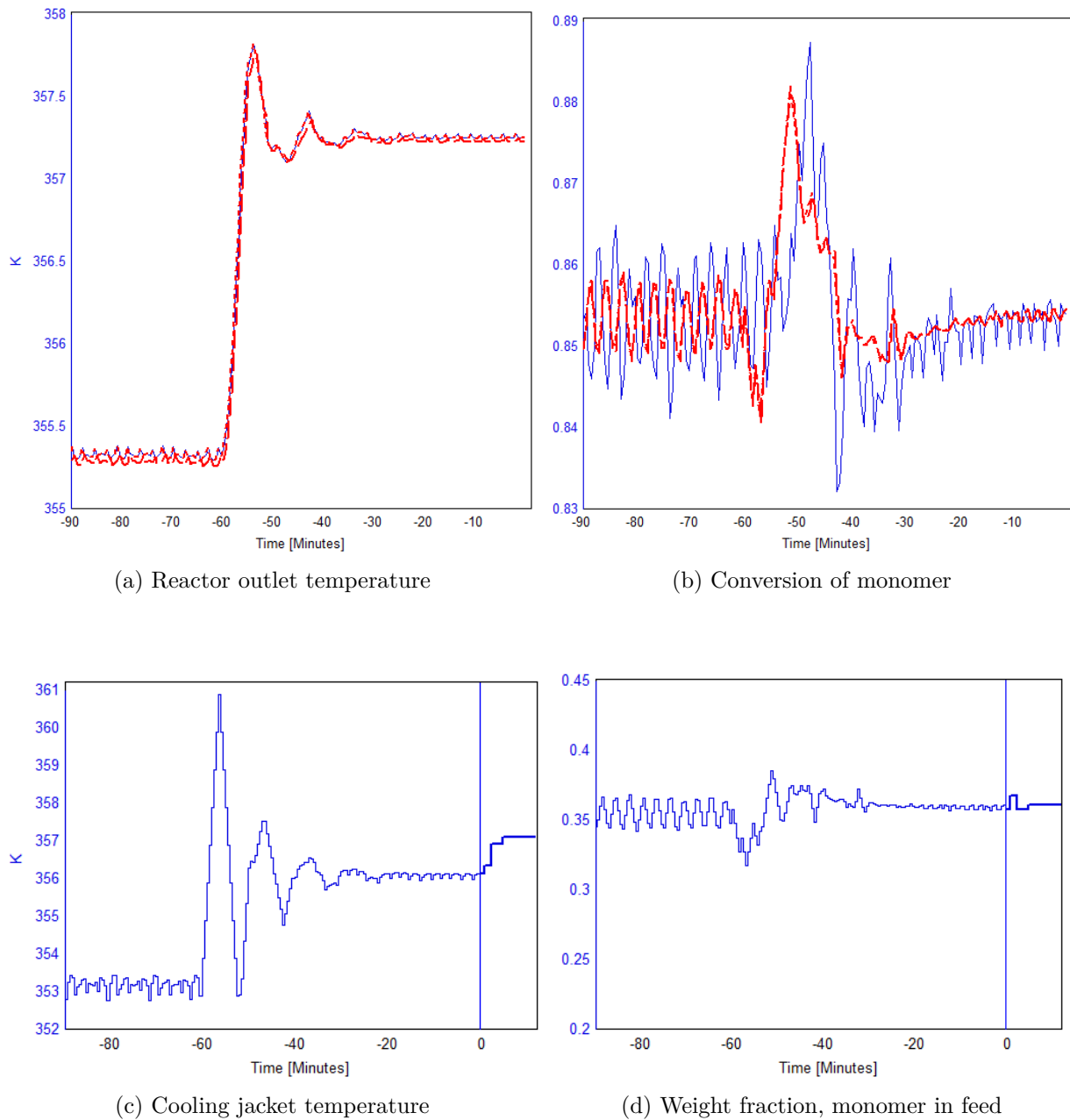


Figure 7.23: Changes in controller action (MVs) and outputs (CVs) for a tubular reactor with a setpoint change in the reactor outlet temperature. The setpoint for the conversion of monomer through the reactor is not altered.

reactions of the copolymerization system are highly temperature dependent, as mentioned in Sec. 2.1. The seemingly successful behavior of the system in this setpoint change is due to the complementing relationship between the two inputs, and provides a good indication with respect to the control structure.

In Fig. 7.24, the inputs and outputs of the system are presented for a setpoint change in the conversion of monomer through the reactor. The setpoint change is obeyed in a satisfyingly efficient manner, but just like for the simulation in Fig. 7.23, the output for which the setpoint is adjusted has a slight overshoot in the control. Like for the reactor outlet temperature, this is believed to be resolvable by performing a fine tuning of the controller. The reactor outlet temperature is virtually unaffected by the change in the conversion of monomer, which is evident from Fig. 7.24a. This is a very desirable situation, and it should be considered in comparison with the simulations from Figs. 7.18 & 7.19 in Sec. 7.7, where the reactor outlet temperature experienced a significant change, with the characteristics of a typical disturbance to the process, when the setpoint for the conversion of monomer was changed.

For the last simulation (Fig. 7.24), the controller action governing the cooling jacket temperature may, at first glance, appear to be uneven and hectic. By carefully examining the scale of the ordinate axis (in Fig. 7.24c), this suspicion is discarded. The changes in the feed monomer content (Fig. 7.24d) are also undramatic in the sense that the input varies over a fairly small range of values, which is well within the interval (with respect to the monomer content in the feed) for which experiments are conducted. For both inputs of this simulation, this is in agreement with what was discussed in Sec. 7.5, among other parts of this report, regarding the fact that performing (large) leap changes in the input may be unsafe and even impossible to accomplish in agreement with the physical interpretation of the reactor system. The small variations in the feed monomer content is not believed to strongly violate what was briefly discussed in Sec. 7.4, where it was stated that tampering with the feed composition must be done with extreme caution. The reason for this is the experienced opinion that variations in the feed composition may cause significant changes in product quality, i.e. molecular weight distributions (MWD), particle size distributions (PSD), etc. A justification for manipulating the monomer content of the feed in spite of these considerations, is the fact that while the amount of monomer is carefully adjusted, the relative ratios of the reactants (apart from the ratio between monomer and water) remain the same, and the two types of monomer are still fed in the same ratio to each other. This means that although the feed composition is adjusted, these modifications are not believed to severely compromise the resulting product quality.

In conclusion, the NMOL type models are found to be particularly well-suited to control a tubular smart-scale reactor, even when there are significant differences between the model predictions and the measured plant behavior. The spatial discretization of the model governs the numerical efficiency of the controller, but the controller is considered to perform satisfyingly well, even for a surprisingly coarse spatial discretization.

7.8.2 Controlling an NMOL type process with an MCV type model

For this second study of the plant replacement concept, the purpose is to invert the case from Sec. 7.8.1, in the sense that the models are switched such that an NMOL type process now represents the plant, while the controller is being governed by an MCV type model. The amount of discrete elements in the MCV model is reduced from 15 to 8 from the previous case, to yield a better numerical efficiency, while the number of spatial discretization points in the NMOL

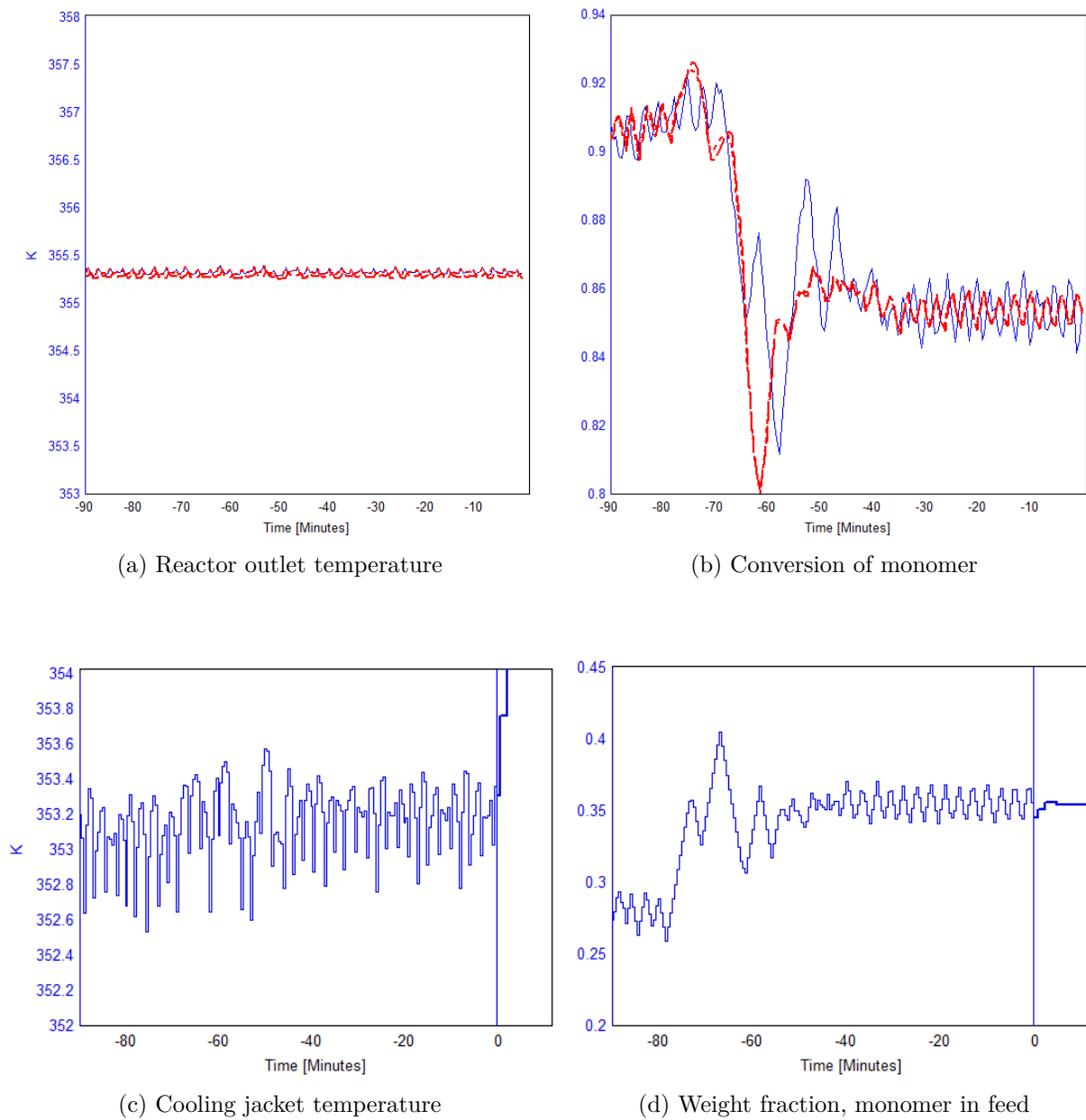


Figure 7.24: Changes in controller action (MVs) and outputs (CVs) for a tubular reactor with a setpoint change in the conversion of monomer. The setpoint for the reactor outlet temperature remains unchanged.

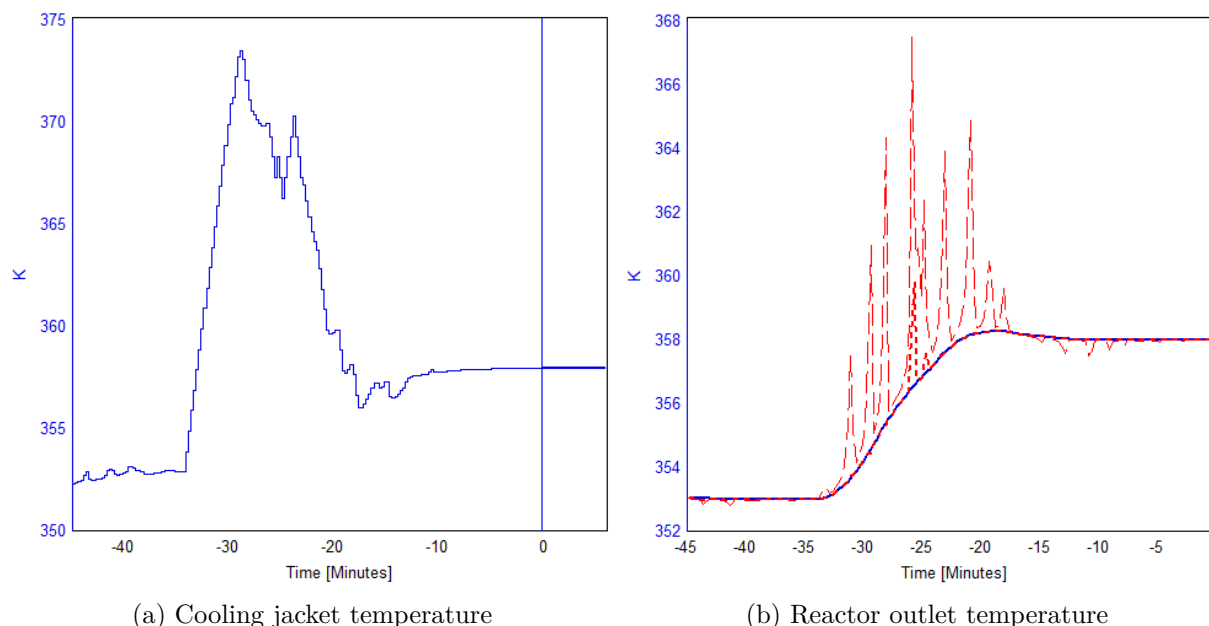


Figure 7.25: A simulated setpoint change for the reactor outlet temperature, using an MCV type model for the controller calculations.

model has been increased from 10 to 30, to achieve a better representation of reality. There is a deliberate compliance between the various simulated cases and the ideas regarding plant replacement models (from Sec. 6.4), in other words.

A simple setpoint change for the reactor outlet temperature is simulated in Fig. 7.25. For this simulation, each sample in the CENIT software requires a computational time¹⁹ of approximately 5 seconds, on average, which is well below the sampling time interval.

The shapes of the respective curves should definitely be addressed for the simulation in Fig. 7.25. The serrated, dashed red line from Fig. 7.25b in particular, which represents the model predictions of the controller with respect to reactor outlet temperature, appears to disagree with the measured behavior of the plant throughout the setpoint change. The red and blue curves coincide in a satisfying manner at steady-state operation, but the temperature predictions from the controller model is significantly more sensitive to changes in the cooling jacket temperature than the plant is. Despite this discrepancy, the temperature of the reactor outlet, for the plant, is changed smoothly and correctly, thus validating the use of MCV type models for temperature control. At this point, it is appropriate to emphasize that there are clearly some quantitative deviations between the two models, but these are effectively accounted for and counteracted by the adequately tuned estimator.

Similar to what was done for the reactor outlet temperature, a setpoint change is made for the conversion of monomer as well. These are, in other words, the same investigations that were conducted for the reversed case from Sec. 7.8.1, where the NMOL type model governed

¹⁹As mentioned before, these simulations are conducted on a standard laptop computer. For this thesis, it would not necessarily be more valuable to run the simulations on a particularly fast computer, e.g. a super-computer, because industrial control systems are often found to be operated by standard work stations at the plant.

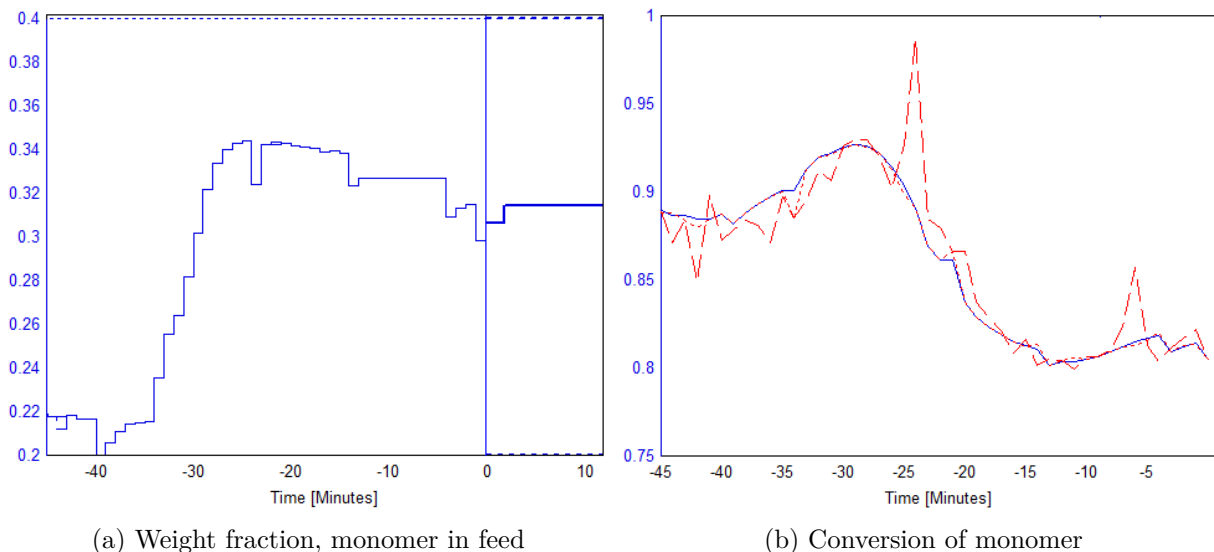


Figure 7.26: A simulated setpoint change for the conversion of monomer through the reactor, using an MCV type model for the controller calculations.

the controller. Fig. 7.26 shows a step change in the setpoint for the conversion of monomer through the reactor. As before, red curves denote the information on which the controller bases its calculations, while the blue curves denote the measured plant behavior. The CENIT calculations average at around 9 seconds of computational time for each sample. It is definitely worth mentioning that this demand is already expected to be slightly higher than what is needed, strictly speaking, because some of the parameters of the model, governing the kinetics and thermodynamics²⁰, are enabled for slight modifications through on-line parameter estimation. This is in agreement with the issues and ideas that were discussed in Secs. 5.1 & 7.2 regarding parameter estimation.

The results from the setpoint change (Fig. 7.26) show the expected behavior. An increase in the monomer content of the feed is required to reduce the conversion of monomer when the mean residence time (i.e. the flow velocity through the reactor) is left unchanged. Conversely, a decrease in the monomer content of the feed will yield an increase in the conversion of monomer. It is important to recognize that the input (MV) is the weight fraction of monomer, i.e. a concentration (on mass basis), and the counterweight to maintain a constant mean residence time is pure water in the feed. This setup can definitely be tweaked to yield unfavorable situations which are undesirable despite that the conversion of monomer is high, and the need for appropriate constraints in both inputs and outputs is evident. Consider, for instance, a situation where the weight fraction of monomer in the feed is very low. In such a case, it would be easy to achieve very close to 100% conversion through the reactor, but the actual amount of produced polymer product would be unreasonably low. In order to avoid this, the monomer content of the feed is constrained to reside between 20% and 40%, which is also believed to agree with the

²⁰Specifically, these parameters are the kinetic adjustment factors and the partition coefficients. These are, in other words, the same parameters that were considered for the off-line parameter estimation problems from Sec. 7.2.

statement from before where it was claimed that tampering with the feed conditions must be performed with caution. Another valid constraint for this problem is to demand that the total mass of polymer product per unit of time must not fall under a predefined lower limit, to ensure a profitable rate of production. The inverse response effect, that is the indubitable increase before the decrease towards the new steady-state, for the conversion of monomer is visible after the setpoint has been changed, but the controller is able to suggest the correct action and achieve the new setpoint in a fairly smooth manner. The key to this success is related to the ideas of the prediction horizon from Sec. 6.3, where both the prediction horizon and the control horizon were introduced as tuning parameters for the NMPC. If the prediction horizon is too short, the controller may not be able to predict the next steady-state properly. Instead, it is caught up with the initial inverse response, and ends up suggesting the wrong action. Conversely, using a long prediction horizon will lead the controller to predict the next steady-state properly, given an adequate process model, and the controller will ignore the inverse response effect. It is definitely worth mentioning that an increase in the prediction horizon implies a potential increase in the computational time associated with each sample for the controller. These concepts are illustrated in Fig. 6.4b.

In summary, the MCV type model appears to master the same tasks as the NMOL type model does when it comes to using NMPC for these control cases. This is a good sign with respect to the overall scope of the thesis, which is to investigate and compare these two various approaches to model a tubular copolymerization reactor. It is found that the MCV type model must have a fairly coarse spatial discretization, i.e. a fairly low number of moving elements, in order to meet the requirements for numerical efficiency in on-line use. As a consequence, the search for valid simplifications and modifications/reformulations to the equation system is not closed permanently.

7.9 Extensions to the control problem in an industrial perspective

In addition to the various cases that have been investigated throughout this entire section, there are several other aspects of the tubular smart-scale reactor which are interesting to consider in an industrial perspective²¹. Some of these cases are discussed briefly here, to provide suggestions for possible extensions to this work, but they are not simulated or studied in depth as they are considered to reside outside the immediate scope of this thesis. In Sec. 7.7 a reactor model was tested, for which the cooling jacket was divided into two separate segments with individually manipulable temperatures. This is one of the suggested modifications to the reactor which are believed to be readily achievable for an industrial tubular reactor. Among the suggestions are also the respective investigations of the particle size distribution (PSD) and the molecular weight distribution (MWD), which are two important advanced properties of interest with respect to controlling a tubular reactor for emulsion copolymerization.

In some cases, it may be difficult to control properties that suffer from a significant time delay in an effective manner. One such quantity is the conversion of monomer through the reactor, for instance, when it is being controlled by manipulating the feed to the reactor. As observed (Fig. 7.26b), the conversion of monomer through the reactor actually has a so-called inverse response effect, which may prove to complicate the control even more. To comprehend this behavior, the definition of the conversion of monomer (Eq. 2.50) must be investigated.

²¹The help of O. Naeem, Polymer Reaction Engineering, BASF, in providing insightful considerations regarding the realization of industrial-scale tubular reactors for emulsion copolymerization, is acknowledged and appreciated.

Because the conversion of monomer is calculated by comparing quantities at the reactor inlet and the reactor outlet, the time delay (i.e. the residence time of the reactor) will impose a discrepancy between the information which is available at the reactor inlet and the reactor outlet, respectively. As a consequence, a fairly odd behavior is observed, where the transfer from one mode of steady-state operation to another starts off by causing a change in the conversion of monomer (for which the inverse response applies) in the wrong direction before it converges to the right steady-state solution. The inverse response effect may also be prominent for some of the other properties of the reactor, and this imposes the requirement that the model must be able to predict the future accurately. One of the reasons that the conversion of monomer is controlled in a fairly efficient manner by using a segmented cooling jacket (Sec. 7.7) is because the time delay in this case is shorter than when the feed conditions are manipulated. This statement is comprehended by considering the physical interpretation of the reactor, where the cooling jacket encapsulates the reactor and hence affects the entire axial extent of the reactor simultaneously, while manipulations done to the feed actually has to propagate through the entire reactor before it can affect the conditions at the reactor outlet. At this point, it is important to recall that the reactor has a mean residence time of about ten minutes at nominal operation. Other tubular reactors may have different RTDs, but the time delay is an extremely important issue in any case. Alternative control structures and/or modifications to the reactor which provide degrees of freedom for which the obstacle of time delay is partially/completely avoided are of particular interest in order to control the tubular reactor more effectively.

The particle size distribution for a continuous tubular copolymerization reactor is, among others, described by Bouaswaig *et. al.* [42], who modeled a detailed approach to continuous emulsion copolymerization reactors which encaptures phenomena such as phase equilibria, micellar nucleation, etc. (some of which were introduced in Sec. 2). The PSD is, without doubt, a complex aspect of a polymerization reactor, and a sophisticated model is required in order to properly account for all the phenomena governing the PSD. Some of the phenomena may give rise to significant complexity in the equation system, and they may cause stiffness, etc. As a consequence, the desire to construct a highly sophisticated model may not be consistent with the desire to design a model which has the numerical efficiency to be suited for on-line optimization and control. This trade-off has been emphatically discussed throughout the thesis, and it is no less important when more complex features of the reactor are considered. One of the indications to be extracted from the work is that the PSD is strongly correlated with the residence time distribution of the reactor [42]. The mass flow of the feed stream was manipulated in some of the simulations in Sec. 7.4, which in practice means a change in the RTD. With a more advanced process model, which does not assume monodispersity for the growing polymer particles, the PSD is believed to be controllable by manipulating the feed conditions. Another way to achieve a desired PSD in an industrial application is to construct a series of tubular reactors, working in parallel, which have different RTDs. In practice, this means that the respective reactors must have different cross-sectional areas or different lengths if they are given the same amount of feed. Conversely, one could deploy a structure where several identical reactors are fed different feed streams. These ideas are illustrated in Fig. 7.27, where the PSDs from two separate reactors, having different RTDs, are combined to yield a wider distribution with respect to particle size. It is emphasized that these plots are fictional and merely state the qualitative idea of obtaining a desired PSD by combining several reactors in parallel. The distribution of the particle size for each reactor is assumed to be normally distributed for the purpose of illustrating the idea of

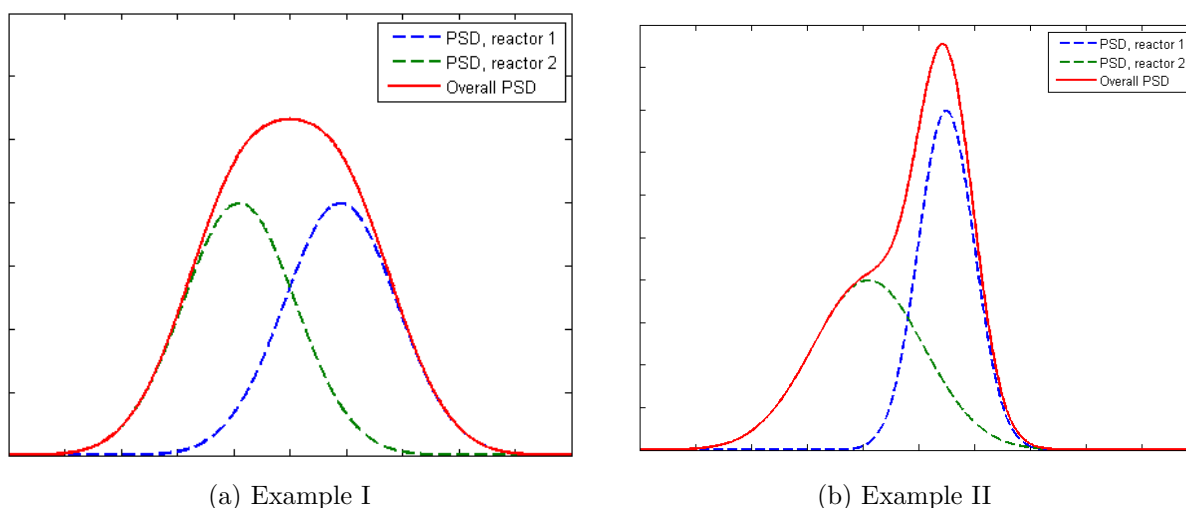


Figure 7.27: Plots showing the idea for how the PSDs for two separate reactors can be combined to yield a wider distribution, if that is desired.

these figures, but that does not necessarily have to be the case. It is emphasized that whether the PSD should be wide or narrow depends wholly on the product specification of the polymer product. The point is, in any case, that the combination of several reactors can be deployed to achieve a desired overall PSD which one single reactor may be unable to achieve. In the context of process control, the simultaneous control of several reactors in parallel may constitute an interesting case.

Another property of interest, which is related to the PSD, is the molecular weight distribution (MWD) for the reactor. The properties that constitute the MWD are the respective molecular weights that were introduced briefly in Sec. 2.4, and they have in common that they are based on molecular weight moments for the growing polymer phase [17]. For the work of this thesis, the molecular weight moments are analogous to the moment equations for the semi-batch reactor from the work preceding this thesis [1], and hence they are formulated as indicated in App. B. It is emphasized that the collection of moment equations should be subject to a revision before the model is suited to control these quantities efficiently in on-line applications. It is expected, just like for the PSD, that to adequately encapture these effects will require a sophisticated model, and this requirement may compromise the numerical efficiency which is crucial for on-line optimization and control.

A way to influence the MWD is to manipulate the amount of chain transfer agent (CTA) which is fed to the reactor, because this has a direct effect on the termination rate for the growing polymer chains. The details of this effect were elaborated, from a chemical point of view, in Sec. 2.1, where the kinetics for the various chemical reactions of the copolymerization system were introduced. Specifically, Eqs. 2.18 & 2.19 indicate how CTA react with growing polymer chains to terminate the chain growth. An interesting case would be to control the polydispersity index (PDI), as defined in Eq. 2.49, by manipulating the amount of CTA in the feed stream. The motivation for doing this is that the PDI is a typical indicator for product quality, and being able to control it is of interest. There is definitely a relationship between the amount of CTA fed to

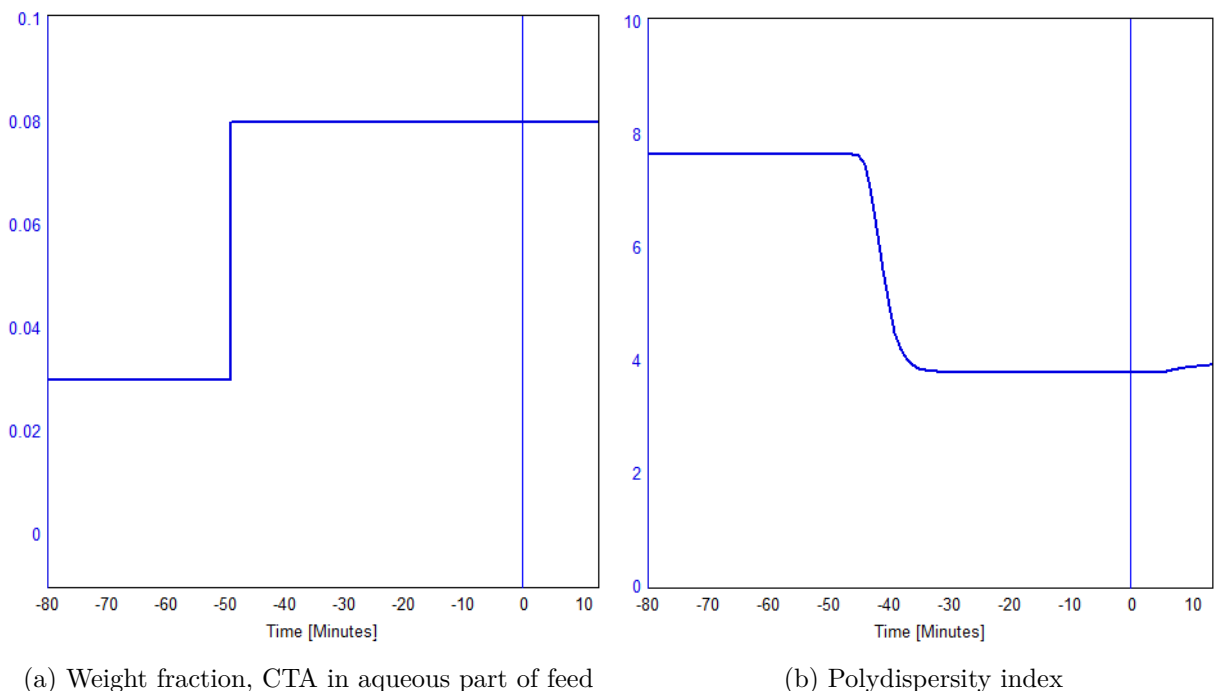


Figure 7.28: A simple demonstration showing the sensitivity of the polydispersity index to changes in the amount of chain transfer agent fed to the reactor.

the reactor and the PDI, as illustrated by the simple step response simulation in Fig. 7.28. This is a cursory simulation to indicate that there is, qualitatively speaking, a sensitivity between the variables that can be used for control.

Instead of manipulating the feeding conditions at the reactor inlet, another suggestion is to include a side stream of CTA which is added somewhere along the axial length of the tubular reactor, as this introduces a new degree of freedom. A feedback loop where the MWD is monitored at the reactor outlet, and the addition of CTA in the side stream is manipulated accordingly, poses an interesting structure with respect to an industrial implementation of such a copolymerization reactor. This is a good suggestion because the main feed of the reactor is not tampered with, and only extra CTA is added to control the MWD. Another good reason to include a side stream is the fact that the time delay associated with the residence time of the reactor is, to some extent, avoided. For this specific suggestion, the time delay for the feedback loop will increase linearly with the distance from the reactor outlet to the side stream. It is important to emphasize that there might be issues, related to the kinetics of the system, which prevent the placement of the side stream very close to the reactor outlet. When the residence time is reduced, so is the time for the extra CTA to take effect, and there is certainly a trade-off to take into account. One could probably find and use an optimal placement of the side stream under the given constraints and objectives of the reactor.

For some modes of operation, the conversion of monomer through the reactor may not be complete, i.e. lower than 100%, which is indeed observed for most of the simulations that are shown through Secs. 7.4 - 7.8. As a consequence, the polymer product blend will contain a certain

amount of unreacted monomer, which should be regained and preferably reused to maintain a high profitability for the operation. Throughout the thesis, the attention has mainly been aimed at the tubular reactor itself, and no specific considerations have been made with respect to other process units in the plant that support the reactor. One could imagine, for instance, a separator situated posterior to the reactor, which is able to separate the phases of the emulsion and possibly separate certain components within the phases. In such a case, the concept of recycling becomes interesting. If a portion of the unreacted monomer is available for being fed back to the reactor, a higher overall conversion of monomer is expected to be achievable. In such a case, the mass flow of the recycle loop would constitute a degree of freedom for the control structure. It is worth mentioning that the same ideas that were discussed for the addition of CTA in the previous paragraph apply to this case as well, and the recycle feed should not necessarily be introduced at the inlet end of the reactor, but possibly somewhere along the axial length. In any case, it is emphasized that such modifications should be studied in-depth before they are applied for on-line use.

To sum up there are, without doubt, several possible modifications to the tubular smart-scale reactor, which are interesting to investigate, that were not considered in depth in this thesis. One could claim that this thesis is a mere step in the right direction with respect to completely exploring the nature of the tubular smart-scale reactor, and further development of the models is suggested for future extensions from this work if the desire is to implement NMPC for a real-life industrial-scale tubular copolymerization reactor.

Conclusions

The purpose of the thesis was to develop two separate models for a tubular smart-scale reactor for emulsion copolymerization. One modeling approach, which is referred to as the numerical method of lines (NMOL) approach, uses finite differences to solve the reactor equations, while the other approach, which is referred to as the moving control volumes (MCV) approach, exploits certain simplifications in the model equations to construct a model which has conceptual similarities to a so-called segregated flow approach. Both of the models have been validated through off-line parameter estimation, using logged experimental data, and simulated for specific control cases using NMPC with a Kalman filter type on-line estimator.

This concluding section aims to summarize the remarks from Sec. 7, based on the established models from Sec. 3, to portray the results of the thesis work and the main challenges to be considered in any possible continuation of this work. While the scope of Sec. 8.1 is to provide a thorough summary of the findings from the work, Sec. 8.2 provides a condensed conclusion for the work.

8.1 Findings from the thesis work

The discoveries of Sec. 3 are by themselves interesting, in showing that alternative approaches for modeling spatially distributed systems, i.e. solving partial differential equations, may constitute efficient alternatives to more traditional strategies. This statement is reinforced by the behavior of the MCV type model in ballistic simulations, which is considered to perform well. Among the results are the indication that a relatively coarse spatial discretization for the MCV approach can provide a proper trade-off between the demand for computational effort and model accuracy. The validity of the traditional approaches are emphasized, however, and the NMOL type reactor models are found to perform well in a wide range of cases, and adjusting the resolution of the spatial discretization gives a flexibility to the model. In conclusion, both of the model types which were considered and developed throughout this thesis, i.e. the NMOL approach and the MCV approach, have their respective strengths and weaknesses, but they both achieve satisfying agreement with reality as well as fairly efficient numerical behavior under the appropriate spatial discretization.

A significant portion of Sec. 3 was dedicated to the investigation of residence time distributions for the respective reactor models which were developed in the thesis. The main motivation for this is the fact that an RTD-investigation is a widespread tool to address the mixing effects of a chemical reactor. In the context of temperature control, under the given assumptions and simplifications of this work, the mixing effects of the reactor are not found to be crucial. The reason for this is that the tubular reactor, on account of its geometry²², is assumed to have well-mixed radial profiles, i.e. no radial gradients. Since the external cooling is exclusively introduced through the reactor wall, the axial mixing effects are not vital for the cooling. On the other hand, the axial distribution of heat is coupled to the axial mixing effects of the reactor. The

²²Recall that the smart-scale reactor is a tube which has a typical "pole" character, meaning that the length of the reactor is significantly larger than the diameter of the cross section.

mixing effects are important with respect to several other quantities of the reactor, however, such as the concentration profiles for the various chemical species, for instance. If the quality of the polymer product is to be controlled by manipulating the feeding conditions, for instance, the mixing effects of the reactor are highly important in order to predict the delayed changes through the tubular reactor. With these ideas in mind, the models were compared to corresponding RTD experiments, after which the parameters governing diffusive mass transport were investigated and adjusted, in order to achieve a better agreement between the models and the experimental results.

At this point it is emphatically mentioned that the considerations for the reactor modeling have been fairly arbitrary, and the fact that emulsion copolymerization reactors are treated in the thesis is not stressed excessively in this context, thus providing a basis for the treatment of spatially distributed reactors in general. An important remark with respect to these ideas is that the successful model-based control of a tubular smart-scale reactor provides a framework to treat other similar reactor systems in the context of on-line optimization and control.

An important issue for the tubular copolymerization reactor is whether the diffusive effects are prominent or not. The system of balance equations for the reactor is simplified significantly when the mass diffusion, i.e. typical Fickian diffusion, is neglected, and although this simplification introduces a slight discrepancy between the model and the real nature of the system, this assumption is considered to be valid in agreement with the RTD experiments. An important observation from the RTD experiments is that the mixing effects of the smart-scale reactor are weak in the sense that mass diffusional effects are inconspicuous. This indication is particularly interesting because neglecting the mass diffusion will simplify the system of equations significantly, as mentioned, and this has consequently been exploited for the models. For the MCV approach, it was discovered that a fairly low amount of discrete elements, namely around ten elements, are sufficient to yield an adequate model, but the NMOL approach requires a high amount of discretization points in space to agree with the RTD experiment, namely in the range of several hundred. It is emphasized, however, that the NMOL model has a qualitative behavior which appears to be surprisingly decent, even at a very coarse spatial discretization, and this leads to believe that this approach has a certain potential with respect to model-based process control. This was investigated and confirmed for some of the control cases which were conducted in Sec. 7.

Other methods for the treatment of spatially distributed systems are possible to deploy as well, but alternatives have not been considered in depth in this thesis. Among the widespread techniques for solving partial differential equations are finite volume methods, collocation methods and spectral methods, to mention some. Variable transformation methods, such as the Laplace Transform or the Fourier Transform, may also provide effective strategies to treat the equations. One could certainly claim that some of these methods are more sophisticated than the finite differences approach which was deployed in this thesis. The possibility is not ruled out, that an alternative method for treating the system of partial differential equations describing the tubular reactor indeed could result in an equation system which would be more efficient than the methods which were deployed for this thesis. Specifically, the numerical scheme arising from the spatial discretization of the problem may be better suited when alternative approaches are deployed. Considerations like these are left for future in-depth studies extending from this work, but it is emphasized that investigations of alternative methods are of interest.

The first test case with respect to process control was conducted for a semi-batch reactor

which, apart from Sec. 7.1, has been granted very little attention throughout the thesis. The semi-batch reactor was introduced and treated in a previous work made by the author of the thesis, and the simplified control case was investigated to provide a finalization to the considerations of this work. It is emphasized that the semi-batch reactor which was considered represents the same chemical system as the emulsion copolymerization for which the tubular smart-scale reactor is treated, and hence provides an interesting system for the purpose of comparison. With that being said, the cooling mechanisms for a semi-batch tank reactor and a typical tubular reactor (having the "pole"-character) may be quite different, and there are other evident differences between the two reactor setups. The results show that temperature control for a semi-batch reactor is effective when NMPC is used. The reactor temperature is kept at a desired setpoint to avoid spikes in the reactor temperature and potential run-away situations. The control is desired to be achieved without aggressive action from the cooling jacket, and the simulation indicates that this is manageable for an undisturbed reactor which operates at nominal conditions. The general trend is that the jacket acts to cool the reactor more actively during the first half of the batch time than the last half. This is the expected behavior indeed, since the chemical reactions of the system proceed more extensively in the first half as a consequence of the continuous feed to the reactor. The semi-batch reactor is a more well-established reactor setup for industrial applications of emulsion copolymerization, and there are several other quantities of interest, in addition to the temperature, which can be controlled, e.g. the product quality. This has not been granted further attention in this thesis, but the investigation of the semi-batch reactor setup is definitely a useful parallel to the tubular smart-scale reactor when it comes to the nature of the emulsion copolymerization system.

Before the various control cases were conducted for the tubular smart-scale reactor, the models were adjusted to fit experiments using off-line parameter estimation. In Sec. 7.2, the Cybernetica Modelfit tool was used to modify certain parameters of the developed models in order to fit the model predictions to an experiment, in which the conversion of monomer through the reactor was measured at four equidistant points, including the reactor outlet. Both kinetic parameters for the chemical reactions, i.e. the dynamics, and partition coefficients governing the phase equilibria, i.e. thermodynamics, were included in the off-line parameter estimation problem. An immediate result from this procedure was that although certain key parameter of the models were adjusted optimally, a satisfying agreement between the models and the experiment was not achieved with respect to the conversion of monomer through the reactor. To account for this, the models were modified in the sense that the solubilities of the respective monomers in the polymer particle phase were adjusted. The various partition coefficients, which govern the solubility of the monomers in the respective phases, were originally constants, but they were modified to vary with the size of the growing polymer particles. This modification is justified by the intuitive suspicion that the polymer particles are able to hold an increasing amount of unreacted monomer as they grow larger, but this modification also agrees with the work of Morton *et. al.* on monomer partitioning [41]. The resulting effect from the modification was that the spatial profile for the conversion of monomer through the reactor, as indicated by the experiments, are obeyed more closely by the models. The terminal quantities of the reactor, i.e. the properties at the spatial reactor outlet, are usually of highest interest, but this consideration helps to emphasize that properties at internal points of the reactor and the spatial profiles are also important.

Micellar nucleation is one of the phenomena in the emulsion copolymerization system which

have been investigated more closely than what was done for the semi-batch system from the preceding work [1]. The explanation for this is that semi-batch reactors almost exclusively operate with seeded polymerization, where a known number of polymer particles, which in practice stays constant, is fed to the reactor initially. For continuous reactors, this is not as common, and the generation/initiation of the polymer particles, i.e. the phenomenon known as micellar nucleation, must be accounted for. The theory for micellar nucleation was established in Sec. 2.3 and introduced into the models, but this is found to contribute with a significant portion of stiffness to the equation system. The micellar nucleation process is extremely rapid compared to most other processes in the reactor, and it has been suggested to deploy steady-state solutions instead of dynamic equations to account for the micellar nucleation. This is considered to be a valid strategy with respect to the performance of the models, at least when the motivation is to use the models for on-line purposes. These findings have a ripple effect because using steady-state solutions to avoid stiffness might be relevant for other parts of the process models as well, e.g. in the other equations for the radical calculations. This may decrease the demand for computational effort for the model, without significantly deteriorating the quality of the outputs from the model.

The various test cases for the tubular smart-scale reactor with respect to process control show that effective control of the reactor outlet temperature is readily achieved in most cases. The successful temperature control of the smart-scale reactor is the result of the encapsulating effect of the cooling jacket for the entire reactor, combined with the assumption of effective heat transfer properties deployed for the models. It is worth mentioning that the experimental data, for which the models have been validated, suffer from the fact that the experiments are conducted under isothermal operating conditions, where the oil bath enclosing the reactor maintains a constant temperature. While these experiments allow for off-line parameter estimation of kinetics properties, solubility parameters, etc., which indeed have been performed, the heat transfer properties of the models are hard to verify. A suggestion for further work is to establish a new lab-scale reactor, or even a pilot plant, which has a different cooling system, where the heat transfer properties of the reactor can be investigated in depth. An actual cooling jacket represents a more realistic situation, from an industrial point of view, compared to the oil bath cooling strategy. Recall from the simulations that, in order to honor the nature of the oil bath cooling system, the temperature for the cooling jacket has been constrained to change relatively slowly throughout the simulations of this thesis. With a different cooling strategy, where the cooling jacket can change its temperature more rapidly, the temperature control is believed to be even more effective than what has been demonstrated in this work.

The control of the conversion of monomer through the reactor by manipulating the feed is a slightly more involved matter, but this is also found to be achievable in an effective manner using NMPC. To achieve this, the model must be able to correctly predict the changes at the reactor outlet which correspond to the specific manipulation in the feed. For a typical tubular reactor, there is a significant time delay associated with such a control strategy. To exemplify, the tubular smart-scale reactor which is considered in this thesis has a mean residence time of about ten minutes at nominal operating conditions. The conversion of monomer is actually found to have a so-called inverse response where the immediate effect (on the conversion) from manipulating the amount of monomer in the feed will go in the opposite direction of the actual steady-state change for a short time after the manipulation. The reason for this is related to how the conversion of monomer is formulated in the process models, and it is utterly crucial

that the model is able to suggest the correct controller action to achieve the desired conversion despite these effects. The key to overcome the obstacle of prominent time delays is to tune the controller, specifically to elongate the prediction horizon of the controller, such that the controller will properly predict the changes from one steady-state solution to another, rather than to get caught up in the inverse response behavior. A longer prediction horizon will typically yield a slight increase in the demand for computational effort, but the alternative is to have a controller that can potentially suggest controller actions which are diametrically opposite of the sensible action to suggest.

Performing test cases where the number of spatial discretization points is varied shows that a low degree of discretization, i.e. a relatively coarse discretization scheme, is favorable for effective process control. At first glance this statement may appear surprising, and it is emphasized that this situation constitutes a contrast to many other situations when it comes to developing and deploying mathematical models. Indeed, it was found that in order to completely represent the mixing effects of the tubular smart-scale reactor, the spatial discretization must be very tight. This requirement is found to be more dire for the NMOL approach than for the MCV approach, on account of the discrete elements of the MCV model actually propagating through the reactor. The conclusion to be drawn from these considerations is that coarsely discretized models possess an advantage over the tightly discretized models on account of the demand for computational effort which makes the tightly discretized models undesirable for on-line use. At this point, it is emphasized that the need for models of high-end quality does not necessarily collide with the statement that coarse discretizations are favorable, and *clever* ways to formulate the models are still important to ensure the desired performance.

A special test case was constructed where the cooling jacket of the smart-scale reactor was segmented into two separate parts, each covering their respective half of the tubular reactor. The purpose of this was to investigate to what extent the properties of interest for the tubular reactor could be controlled by the cooling jacket alone instead of manipulating the feeding conditions at the reactor inlet. This is believed to reduce the significance of the time delay through the reactor with respect to achieving effective control of the properties at the reactor outlet. Most importantly, this strategy is believed to portray an interesting setup with respect to the industrial realization of the emulsion copolymerization reactor in mind. The results show that with the 2x2-type control structure, for which both of the two degrees of freedom are found in the cooling jacket, both the reactor outlet temperature and the conversion of monomer through the reactor can be controlled effectively. The simulations show that the first half of the cooling jacket, chronologically speaking, is the most active contributor to the control of the conversion of monomer, which is sensible since the reactor temperature affects the extent of the propagation reactions in the reactor, due to the temperature dependency of the kinetics. Correspondingly, the last half of the cooling jacket acts to ensure that the reactor outlet temperature meets the desired setpoint. In addition, the ability of this structure to reject disturbances was tested, when a change was made in the feed composition while the respective setpoints for the properties of interest were kept constant. The results show that the disturbances are quite effectively counteracted within the constraints of the inputs. A suggestion for further investigations is to use more than just two segments in the cooling jacket, and discover whether there are other properties of interest in the reactor that can be controlled effectively through the cooling strategy.

The purpose of a well-functioning on-line estimator was demonstrated when the concept of plant replacement models were shown in the final control study (Sec. 7.8). In this case, the

two modeling approaches used in the thesis, between which there are certain discrepancies, were tested in turn for controller calculations while the other model played the role of the "real" plant. Based on these results, the models are considered to be adequate for on-line use. A suggestion for further work is to reproduce the control cases for an actual real-life reactor to validate the performance of the NMPC. Such investigations could even be performed for a pilot plant.

From the results of this thesis, it would be sensible to conclude that the use of model-based approaches, such as NMPC, for polymerization processes is an appropriate strategy to achieve effective control. The performance of the controller is, as emphatically mentioned before, largely dependent on the quality of the model on which it bases its calculations, but a proper model will yield both safe and effective operation of the reactor at hand. MPC methods are, generally speaking, considered to be increasingly effective when the complexity of the system for which it is deployed increases, as compared to other methods for advanced process control. The typical copolymerization reactors which have been considered in this thesis are found to reinforce this statement, as complex elements of the models such as nonlinearities, phase equilibria, various constraints, strongly coupled dynamics etc. are effectively treated by the NMPC.

Despite what has been discovered with respect to numerical efficiency and model accuracy in the context of process control, it is believed that in order to control more complex features of the reactor, e.g. the molecular weight distribution or the partial size distribution, the models with a very coarse spatial discretization may not be sufficient. The concluding remarks with respect to spatial discretization must be somewhat revised, in other words, if other properties of the reactor are investigated in the context of process control. This was discussed briefly in Sec. 7.9, where possible extensions to the control problem in an industrial context were considered. It is also suggested that certain modifications to the physical structure of the reactor may provide ways to control such properties more effectively.

It is appropriate to include a concluding remark regarding the use of Modelica to construct these kind of models. Although the possibility of using Modelica code through the FMI has been demonstrated in the work of this thesis, the FMI has certain weaknesses which complicate the use of models written in Modelica in external software. As a consequence, an advice is to rewrite the models in the programming language C, preferably, in the event of a significant revision of the models, as this is believed to increase the compatibility of the models and also provide a slight increase in the numerical efficiency of the models. With that being said, a release of a revised version of the FMI is imminent, which is believed to strengthen the compatibility of the Modelica models. The strengths of Modelica, together with the various built-in functions, provide a powerful platform for the modeler, but they need to be completely encapsulated by the FMI as well.

8.2 Final conclusion

The MCV approach is a valid strategy to obtain a functioning and efficient model for a spatially distributed system. This approach is advised for on-line control of tubular liquid-phase reactors in general, based on the performance which is observed for the case study on the emulsion copolymerization reactor. The NMOL approach is certainly also a useful method to treat these kinds of systems, and performs well for the simulated cases. The use of NMPC together with an appropriate on-line estimator is considered to be a promising strategy for a process as complex as the tubular smart-scale reactor, and further investigations should be aimed at both fine adjustments of the models as well as more on-line testing in lab-scale or even for a pilot plant.

References

- [1] F. Gjertsen, Models for on-line control of batch polymerization processes: State and parameter estimation for semi-batch free-radical emulsion copolymerization processes, Technical report, Specialization project, Dept. Chem. Eng., NTNU, 2013
- [2] PlasticsEurope, Plastics - the Facts 2012, An analysis of European Plastics production, demand and waste data for 2011, Belgium, 2012
- [3] P. C. Painter & M. M. Coleman, Fundamentals of Polymer Science, 2nd Ed., CRC Press, 1997
- [4] I. K. Khairullin, Adhesive-Melts - The Most Dynamically Developing Area in World Production and Consumption of Adhesives, Pol. Sc. Series D, Glues and Sealing Materials, 2013, Vol. 6, No. 1
- [5] N. Leveson & G. Stephanopoulos, A System-Theoretic, Control-Inspired View and Approach to Process Safety, AIChE J., Vol. 60, Iss. 1, 2013
- [6] H. Ghodke, S. Raman & B. E. Ydstie, Modeling and Control of Free Radical Copolymerization, Department of Chemical Engineering, Carnegie Mellon University, Pittsburgh, USA
- [7] A. H. Helgesen, Toolbox for generation of nonlinear control models for semi-batch emulsion polymerization reactors, Master thesis, Department of Chemical Engineering, Norwegian University of Science and Technology, 2011
- [8] A. Nyström, Modeling and Simulation of a Multi Phase Semi-batch Reactor, Master thesis, Department of Mathematical Sciences, Chalmers University of Technology & Göteborg University, Gothenburg, Sweden, 2007
- [9] H. S. Fogler & N. Gurmen, University of Michigan, Note on modeling of semi-batch reactors, 2007, URL: http://www.umich.edu/~essen/html/06chap/html/prs_cstr.htm
- [10] B. Li & B. W. Brooks, Prediction of the Average Number of Radicals per Particle for Emulsion Polymerization, J. Pol. Sc., Vol. 31, Iss. 9, Aug. 1993
- [11] M. Nomura, H. Tobita & K. Suzuki, Emulsion Polymerization: Kinetic and Mechanistic Aspects, Adv. Polym. Sci. 175, 2005, pp. 1 - 128, Springer Verlag
- [12] J. L. Gardon, Emulsion Polymerization. I. Recalculation and Extension of the Smith-Ewart Theory, J. Pol. Sci. Part A-1, Vol. 6, pp. 623 - 641, 1968
- [13] A. C. Hindmarsh & R. Serban, User Documentation for cvodes v2.7.0, March 2012, Center for Applied Scientific Computing, Lawrence Livermore National Laboratory
URL: http://computation.llnl.gov/casc/sundials/documentation/cvs_guide.pdf

- [14] H. Olsson, H. Elmqvist & M. Otter, Modelica - A Unified Object-Oriented Language for Systems Modeling, Language Specification, Version 3.3, May 9, 2012
URL: <https://www.modelica.org/documents/ModelicaSpec33.pdf>
- [15] J. Nocedal & S. J. Wright, Numerical Optimization, 2nd Ed., Springer, 2006
- [16] T. S. Schei, On-line estimation for process control and optimization applications, 8th International IFAC Symposium on Dynamics and Control of Process Systems, Preprints Vol. 2, June 6-8, 2007
- [17] T. J. Crowley & K. Y. Choi, Calculation of Molecular Weight Distribution from Molecular Weight Moments in Free Radical Polymerization, Ind. Eng. Chem. Res. 1997, 36, pp. 1419 - 1423
- [18] J. Gao & A. Penlidis, Mathematical modeling and computer simulator/database for emulsion polymerizations, Prog. Polym. Sci. 27 (2002), pp. 403 - 535, Elsevier
- [19] H. A. Jakobsen, Chemical Reactor Modeling: Multiphase Reactive Flows, Springer, 2008
- [20] C. E. Wyman, Polymer Moment Equations for Distributed Parameter Systems, AIChE J., Vol. 21, No. 2, 1975
- [21] J.B. Rawlings & D. Q. Mayne, Model Predictive Control: Theory and Design, 2013, Nob Hill Publishing, LLC
- [22] D. Simon, Optimal State Estimation: Kalman, H_∞ and Nonlinear approaches, 1st Ed., John Wiley & Sons, 2006
- [23] E. O. Kreyszig, Advanced Engineering Mathematics, 9th Ed., Wiley, 2005
- [24] M. P. Dudukovic & R. M. Felder, Mixing Effects in Chemical Reactors - V - Micromixing and the Segregated Flow Model, Module E4.8, Modular Instruction Series, American Institute of Chemical Engineers
- [25] R. Hollemann, O. Fringer & M. Stacey, Numerical diffusion for flow-aligned unstructured grids with application to estuarine modeling, Int. J. Meth. Fluids, 2013
- [26] R. E. Walpole, R. H. Myers, S. L. Myers & K. Ye, Probability and Statistics for Engineers and Scientists, 8th Ed., Pearson Education, 2007
- [27] M. B. Cutlip & M. Shacham, The Numerical Method of Lines for Partial Differential Equations, as adapted from: M. B. Cutlip & M. Shacham, Problem Solving in Chemical Engineering with Numerical Methods, Upper Saddle River, Prentice Hall, 1999
- [28] D. Eberly, Derivative Approximation by Finite Differences, Geometric Tools, LLC, 2008
- [29] M. Hovd, Lecture notes for the course Advanced Control of Industrial Processes, Department of Engineering Cybernetics, 2009
- [30] O. Levenspiel, Chemical Reaction Engineering, 2nd Ed., Wiley International Edition, John Wiley & Sons, 1972

- [31] H. Scott Fogler, Elements of Chemical Reaction Engineering, 4th Ed., 2005, Prentice Hall Publishing
- [32] S. O. Hauger, MPC: Model Predictive Control, Presentation slides, TTK16, NTNU & Cybernetica AS, 2013
- [33] G. Shah & S. Engell, Tuning MPC for Desired Output Closed-Loop Performance for MIMO Systems, 2011 American Control Conference, USA 2011
- [34] COOPOL, Deliverable 4.6: Control model for industrial scale semi-batch polymerisation reactor, Internal COOPOL report, March 2013
- [35] COOPOL, Deliverable 4.4: Utilities for partitioning of species, emulsion rheology and coagulation/fouling, Internal COOPOL report, April 2014
- [36] S. Skogestad, Control structure design for complete chemical plants, Computers and Chemical Engineering 28, pp. 219 - 234, Elsevier, 2004
- [37] S. Skogestad, Simple analytic rules for model reduction and PID controller tuning, J. Process Control, Vol. 13, Iss. 4, June 2003, pp. 291 - 309
- [38] S. Skogestad, Probably the best simple PID tuning rules in the world, J. Process Control, September 2001
- [39] Cybernetica ModelFit User Manual, v. 1.20, Cybernetica AS
- [40] T. S. Schei & J. G. Dyrset, Newton-type MPC algorithm for nonlinear state-space models - Revision 2, Internal Report, Cybernetica AS, 2004
- [41] M. Morton, S. Kaizerman & M. W. Altier, Swelling of latex particles, J. Colloid Sci., Iss. 9, pp. 300 - 312, 1954
- [42] A. E. Bouaswaig, W. Mauntz & S. Engell, Modeling and simulation of the particle size distribution for emulsion polymerization in a tubular reactor, ESCAPE 18, 2008, Elsevier Ltd.

Appendix A

Additional information for radical species modeling

When applying a full population balance to describe the radicals of the copolymerization system, as introduced in Sec. 2, the calculations use a system of equations involving a so-called A -matrix. The purpose of this matrix is to govern the respective changes in particles carrying between 0 and N radicals. The A -matrix is constructed as indicated in the MATLAB script¹ in Cs. A.1. The full population balance approach to model the radical species of the reactor has, in most simulations in Sec. 7, not been deployed because the dynamic Li-approximation (introduced in Sec. 2) is more efficient. In this script, σ , k and C represent the corresponding quantities from Sec. 2, i.e. generation, desorption and decay of radicals, respectively, while N denotes the maximum number of radicals per particle considered in the approach.

```
1 function [ dn ] = ndot_full( t, n )
2 % ndot_full
3 % function to calculate the time derivative of relative
4 % radical frequency vector for emulsion polymerization
5
6 Li_parameters;
7 N = max(size(n))-1; % Max. number of radicals per particle
8
9 A = [ -sigma k 2*C 0 zeros(1,N-3 );
10       sigma -(sigma + k ) 2*k 6*C zeros(1,N-3 );
11       zeros( N-1, N+1 ) ];
12
13 for i = 2:N,
14     A(i+1,i) = sigma;
15     A(i+1,i+1) = - sigma - i * k - i * (i-1) * C;
16     if i < N,
17         A(i+1,i+2) = (i+1) * k;
18     end
19     if i < N-1,
20         A(i+1,i+3) = (i+2)*(i+1) * C;
21     end
22 end
23
24 dn = A * n ;
25 end
```

Code snippet A.1: MATLAB code for generating the A -matrix used in the full population balance, used for radical species modeling in Sec. 2.

¹This script is printed with the permission of P. Singstad, Cybernetica AS.

Appendix B

Polymer moments for copolymer product calculations

This section presents the polymer moment equations which are used in the calculations for product quality. The polymer moments were introduced in Sec. 2 for calculating the molecular weight distribution for the copolymer product, as described by Crowley and Choi [17]. The use of moments for describing the molecular weight distribution is also discussed by Gao and Penlidis. [18]

The definition of the polymer moments is given in Eqs. B.1 - B.3. Here, i denotes the order of the moment, while $[P_j(n)]$ denotes the concentration of living polymer med endgroup j , having a length of n units, etc. The resulting system of equations for a copolymerization system with two different monomers is given in Eqs. B.4 - B.12.

$$\mu_i^{P1} = \sum_{n=1}^{\infty} n^i [P_1(n)] \quad (\text{B.1})$$

$$\mu_i^{P2} = \sum_{n=1}^{\infty} n^i [P_2(n)] \quad (\text{B.2})$$

$$\mu_i^D = \sum_{n=1}^{\infty} n^i [D(n)] \quad (\text{B.3})$$

Zeroth, first and second order moments, with respect to living chains with type 1 endgroup:

$$\begin{aligned} \frac{d\mu_0^{P1}}{dt} = & \frac{[M_1]}{[M_1] + [M_2]} \frac{2f_I k_I n_I}{V_p} - k_{p12}[M_2]\mu_0^{P1} + k_{p21}[M_1]\mu_0^{P2} \\ & - k_{f12}[M_2]\mu_0^{P1} + k_{f21}[M_1]\mu_0^{P2} - \phi(k_{tc} + k_{td}) \left((\mu_0^{P1})^2 + \mu_0^{P1}\mu_0^{P2} \right) \end{aligned} \quad (\text{B.4})$$

$$\begin{aligned} \frac{d\mu_1^{P1}}{dt} = & \frac{[M_1]}{[M_1] + [M_2]} \frac{2f_I k_I n_I}{V_p} + k_{p11}[M_1]\mu_0^{P1} + k_{p21}[M_1] (\mu_0^{P2} + \mu_1^{P2}) \\ & - k_{p12}[M_2]\mu_1^{P1} - k_{f,CTA,1}[CTA] (\mu_1^{P1} - \mu_0^{P1}) - k_{f11}[M_1] (\mu_1^{P1} - \mu_0^{P1}) \\ & + k_{f21}[M_1]\mu_0^{P2} - k_{f12}[M_2]\mu_1^{P1} - \phi(k_{tc} + k_{td}) (\mu_0^{P1}\mu_1^{P1} + \mu_1^{P1}\mu_0^{P2}) \end{aligned} \quad (\text{B.5})$$

$$\begin{aligned} \frac{d\mu_2^{P1}}{dt} = & \frac{[M_1]}{[M_1] + [M_2]} \frac{2f_I k_I n_I}{V_p} + k_{p11}[M_1] (\mu_0^{P1} + 2\mu_1^{P1}) \\ & + k_{p21}[M_1] (\mu_0^{P2} + 2\mu_1^{P2} + \mu_2^{P2}) - k_{p12}[M_2]\mu_2^{P1} \\ & - k_{f,CTA,1}[CTA] (\mu_2^{P1} - \mu_0^{P1}) - k_{f11}[M_1] (\mu_2^{P1} - \mu_0^{P1}) + k_{f21}[M_1]\mu_0^{P2} \\ & - k_{f12}[M_2]\mu_2^{P1} - \phi(k_{tc} + k_{td}) (\mu_0^{P1}\mu_2^{P1} + \mu_2^{P1}\mu_0^{P2}) \end{aligned} \quad (\text{B.6})$$

Zeroth, first and second order moments, with respect to living chains with type 2 endgroup:

$$\begin{aligned} \frac{d\mu_0^{P2}}{dt} = & \frac{[M_2]}{[M_1] + [M_2]} \frac{2f_I k_I n_I}{V_p} + k_{p12}[M_2]\mu_0^{P1} - k_{p21}[M_1]\mu_0^{P2} \\ & + k_{f12}[M_2]\mu_0^{P1} - k_{f21}[M_1]\mu_0^{P2} - \phi(k_{tc} + k_{td}) \left((\mu_0^{P2})^2 + \mu_0^{P1}\mu_0^{P2} \right) \end{aligned} \quad (\text{B.7})$$

$$\begin{aligned} \frac{d\mu_1^{P2}}{dt} = & \frac{[M_2]}{[M_1] + [M_2]} \frac{2f_I k_I n_I}{V_p} + k_{p22}[M_2]\mu_0^{P2} - k_{p21}[M_1]\mu_1^{P2} + k_{p12}[M_2] (\mu_0^{P1} + \mu_1^{P1}) \\ & - k_{f,CTA,2}[CTA] (\mu_1^{P2} - \mu_0^{P2}) - k_{f22}[M_2] (\mu_1^{P2} - \mu_0^{P2}) \\ & - k_{f21}[M_1]\mu_1^{P2} + k_{f12}[M_2]\mu_0^{P1} - \phi(k_{tc} + k_{td}) (\mu_0^{P1}\mu_1^{P2} + \mu_1^{P2}\mu_0^{P2}) \end{aligned} \quad (\text{B.8})$$

$$\begin{aligned} \frac{d\mu_2^{P2}}{dt} = & \frac{[M_2]}{[M_1] + [M_2]} \frac{2f_I k_I n_I}{V_p} + k_{p22}[M_2] (\mu_0^{P2} + 2\mu_1^{P2}) - k_{p21}[M_1]\mu_2^{P2} \\ & + k_{p12}[M_2] (\mu_0^{P1} + 2\mu_1^{P1} + \mu_2^{P1}) - k_{f,CTA,2}[CTA] (\mu_2^{P2} - \mu_0^{P2}) \\ & - k_{f22}[M_2] (\mu_2^{P2} - \mu_0^{P2}) - k_{f21}[M_1]\mu_2^{P2} + k_{f12}[M_2]\mu_0^{P1} \\ & - \phi(k_{tc} + k_{td}) (\mu_0^{P1}\mu_2^{P2} + \mu_2^{P2}\mu_0^{P2}) \end{aligned} \quad (\text{B.9})$$

Zeroth, first and second order moments, with respect to dead chains:

$$\begin{aligned} \frac{d\mu_0^D}{dt} = & k_{f,CTA,1}[CTA]\mu_0^{P1} + k_{f,CTA,2}[CTA]\mu_0^{P2} + k_{f11}[M_1]\mu_0^{P1} \\ & + k_{f22}[M_2]\mu_0^{P2} + k_{f12}[M_2]\mu_0^{P1} + k_{f21}[M_1]\mu_0^{P2} \\ & + \phi k_{tc} \left(\frac{1}{2}\mu_0^{P1}\mu_0^{P1} + \mu_0^{P2}\mu_0^{P1} + \frac{1}{2}\mu_0^{P2}\mu_0^{P2} \right) \\ & + \phi k_{td} \left(\mu_0^{P1}\mu_0^{P1} + 2\mu_0^{P2}\mu_0^{P1} + \mu_0^{P2}\mu_0^{P2} \right) \end{aligned} \quad (\text{B.10})$$

$$\begin{aligned} \frac{d\mu_1^D}{dt} = & k_{f,CTA,1}[CTA]\mu_1^{P1} + k_{f,CTA,2}[CTA]\mu_1^{P2} + k_{f11}[M_1]\mu_1^{P1} \\ & + k_{f22}[M_2]\mu_1^{P2} + k_{f12}[M_2]\mu_1^{P1} + k_{f21}[M_1]\mu_1^{P2} \\ & + \phi(k_{tc} + k_{td}) (\mu_0^{P1}\mu_1^{P1} + \mu_0^{P2}\mu_1^{P1} + \mu_1^{P2}\mu_0^{P1} + \mu_0^{P2}\mu_1^{P2}) \end{aligned} \quad (\text{B.11})$$

$$\begin{aligned} \frac{d\mu_2^D}{dt} = & k_{f,CTA,1}[CTA]\mu_2^{P1} + k_{f,CTA,2}[CTA]\mu_2^{P2} + k_{f11}[M_1]\mu_2^{P1} \\ & + k_{f22}[M_2]\mu_2^{P2} + k_{f12}[M_2]\mu_2^{P1} + k_{f21}[M_1]\mu_2^{P2} \\ & + \phi k_{tc} \left(\mu_0^{P1}\mu_2^{P1} + \mu_1^{P1}\mu_1^{P1} + \mu_0^{P2}\mu_2^{P1} + 2\mu_1^{P2}\mu_1^{P1} + \mu_2^{P2}\mu_0^{P1} + \mu_0^{P2}\mu_2^{P2} + \mu_1^{P2}\mu_1^{P2} \right) \\ & + \phi k_{td} \left(\mu_2^{P1}\mu_0^{P1} + \mu_0^{P2}\mu_2^{P1} + \mu_2^{P2}\mu_0^{P1} + \mu_2^{P2}\mu_0^{P2} \right) \end{aligned} \quad (\text{B.12})$$

This system of differential equations provides a significant contribution to the total number of states for the model. This acts to increase the demand for computational power.

Appendix C

Semi-batch reactor modeling

This additional section introduces some of the basic concepts of semi-batch reactor modeling. Semi-batch reactors are, strictly speaking, not the main reactor type of interest for this thesis, but the modeling of a semi-batch reactor can be utilized when modeling a distributed tubular reactor. The strategy introduced in Sec. 3.5, in particular, is essentially a collection of batch reactor which move in space. Because of this, modeling concepts for the semi-batch reactor has been included. [8][9]

The theory in this section is adopted from a specialization project, written by the author of this thesis, in which semi-batch reactor modeling was the main purpose. It has been included as an appendix to this thesis to provide a theoretical backbone to the strategies of tubular reactor modeling, in which the batch reactor equations are valuable. [1]

The semi-batch reactor has several similarities to the regular batch reactor which, by assumption, is a perfectly mixed tank reactor with no gradients, neither with respect to concentration, temperature or any other intensive quantity, in spatial coordinates. Under this assumption, the modeling is simplified, but the reactor design accounts for time variation. It is important to note that a regular batch reactor does not allow for convective fluid transport across the reactor boundary during the course of the reaction. For a batch reactor, in its most simple form, the design equations (which are the component mass balances and energy balance, respectively) can hence be written as indicated in Eqs. C.1 & C.2.

$$\text{Mass balances: } \frac{dN_i}{dt} = r_i V \quad i = 1, \dots, n \quad (\text{C.1})$$

$$\text{Energy balance: } \frac{dU}{dt} = Q + W_s \quad (\text{C.2})$$

In this formulation, N_i is the amount of species i , U is the internal energy of the reactor, $V(t)$ is the (time dependent) volume of the reactor contents, r_i is the reaction rate of component i , W_s is the shaft work applied to the system, which is usually negligible, while Q is the energy change associated with external heat transfer (heating, cooling, heat loss, etc.). This model describes a system with n chemical species, each with its independent mass balance. The reaction rate of each component in the system (r_i) can be modeled in several ways, depending on the complexity and the nature of the chemical reactions. A common approach is to calculate the reaction rate using a rate law, in which the reaction rate is assumed to vary with the concentration of the respective component to some order, like indicated in Eq. C.3. In this expression, k indicates the order of the rate law, and k_i is a reaction rate constant, for example calculated as indicated in Eq. 2.9 for the copolymerization case. In Eq. 2.9, the Arrhenius temperature dependency of the reaction rate constant is used, and this is a widespread approach for modeling kinetics of this kind.

$$r_i = k_i c_i^k \quad (\text{C.3})$$

In some applications, assumptions can be made to simplify the model equations further. For instance, a so-called temperature explicit energy equation can be achieved under the assumption

that the heat capacity (indicated by C_p in Eq. C.4) of the reactor content is time independent. This assumption often holds for aqueous solutions, among other systems. In cases where this assumption does not hold, the heat capacity should be modeled dynamically as a temperature function in order to achieve the temperature explicit form of the energy balance. The motivation for obtaining a temperature explicit energy balance is that the intensive property temperature appears much more tangible than energy. The fact that the measuring of temperature can be performed in a straight-forward manner for many applications also makes temperature a preferred quantity over energy. Another simplification for the model equations comes along when the total volume of the reactor content can be assumed to be constant as time passes in the reactor. Using $c_i = \frac{N_i}{V}$ (where c_i denotes the concentration of species i), this simplifies the molar mass balances. These common simplifying cases are given in Eqs. C.4 & C.5.

$$\text{Mass balances:} \quad \frac{dc_i}{dt} = r_i \quad , \quad i = 1, \dots, n \quad (\text{C.4})$$

$$\text{Energy balance:} \quad mC_p \frac{dT}{dt} = W_s + Q \quad (\text{C.5})$$

The model equations for the batch reactor are the basis for the approach towards a semi-batch reactor model which, in addition to "just" being a time dependent reaction vessel, also allow for external mass transfer across the reactor boundary. In the COOPOL copolymerization reactor process, which is the basis for this work, for instance, the reactants are added gradually during the reaction time, instead of adding them all at the start of the reaction (which would be the regular batch reactor approach). Another strategy of interest could be to extract product gradually during the course of the batch, which is also allowed in the semi-batch reactor approach. The reactor design now approaches the nature of a continuous reactor, in which CSTR² probably is the closest relative. Effects from the flowing fluids, both with respect to species mass balances and the energy balance must now be added, and the volume of the fluid contents must be given attention as well.

$$\begin{aligned} \text{Mass balances:} \quad \frac{dN_i}{dt} = r_i V(t) + \hat{n}_i &\implies \frac{d}{dt} (c_i V(t)) = r_i V(t) + \hat{n}_i \\ \implies V(t) \frac{dc_i}{dt} + c_i \frac{dV}{dt} &= r_i V(t) + \hat{n}_i \end{aligned} \quad (\text{C.6})$$

In Eq. C.6, \hat{n}_i is the net flow rate of species i to the reactor, i.e. the difference between the inflow and the outflow. In this general approach, effort is made to emphasize that the species mass balances hold for formulations on both molar basis and mass basis, as long as the corresponding reaction rates etc. are treated in agreement with this, i.e. using the correct units. The mass basis is perhaps the most established and intuitive approach, and this is mainly due to the principle of conservation of mass. This enables the development of useful equations to describe the system, e.g. volume balances³ or dynamic equations for the composition of the respective species in the reactor. In the expression below (Eq. C.8), ρ is the density of the fluid mixture residing in the

²Continuously stirred tank reactor: A time-invariant, perfectly mixed tank reactor.

³The expression "volume balance" should be used with care, since the volume itself is not a conserved quantity. In the same way an energy balance can *manifest* itself as an "enthalpy balance" in certain cases, the mass balance can be reformulated to a "volume balance" if this is desired by the modeler.

reactor.

$$\frac{dm_{tot}}{dt} = \hat{m}_{in} - \hat{m}_{out} \quad (C.7)$$

$$\implies \frac{d}{dt}(\rho V) = \hat{m}_{in} - \hat{m}_{out}$$

$$\implies \rho \frac{dV}{dt} + V \frac{d\rho}{dt} = \hat{m}_{in} - \hat{m}_{out} \quad (C.8)$$

This equation can be combined with the species mass balances, Eq. C.6, to eliminate the derivative of the volume from the mass balance if desired, but that also raises the need for a differential equation in time expressing the density variation. In many cases, simplifications can be introduced to make the model easier to treat, yet still represent reality in a good way. For instance, for a fluid with constant density and no fluid flow out of the reactor, this equation intuitively reduces to Eq. C.9, which represents a special simplified case for the semi-batch reactor case. A conceptual drawing of a semi-batch tank reactor is given in Fig. C.1.

$$\frac{dV}{dt} = \hat{V}_{in} \quad (C.9)$$

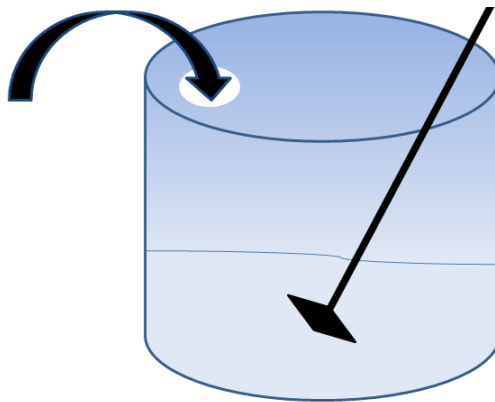


Figure C.1: Conceptual illustration of a semi-batch tank reactor with stirrer and continuous feeding.

The conversion of reactant in a semi-batch reactor can be formulated in several ways. In Eqs. C.10 & C.11, two different formulations are shown for a batch simulated from $t = t_0$ to $t = t_1$ in time. Here, $R(t)$ denotes the molar amount of reactant in the reactor at time t and $\hat{n}_R(\tau)$ denotes the net flow of reactant at time τ ⁴. The two expressions have striking similarities, but while the "continuous" (also referred to as instantaneous conversion) formulation considers the conversion with respect to accumulated monomer at time t , the "total" (also referred to as global conversion) formulation considers the conversion with respect to accumulated monomer

⁴The greek letter τ is here introduced to be used as an integration variable, with the main purpose being to avoid confusion when it comes to the variable t denoting time.

at time t_1 . That is, the total amount of monomer for the entire batch time.

$$\text{"Total" formulation: } X(t) = \frac{R(t_0) + \int_{t_0}^t \hat{n}_R(\tau) d\tau - R(t)}{R(t_0) + \int_{t_0}^{t_1} \hat{n}_R(\tau) d\tau} \quad (\text{C.10})$$

$$\text{"Continuous" formulation: } X(t) = \frac{R(t_0) + \int_{t_0}^t \hat{n}_R(\tau) d\tau - R(t)}{R(t_0) + \int_{t_0}^t \hat{n}_R(\tau) d\tau} \quad (\text{C.11})$$

For a batch reactor, the total and continuous formulations will be identical. For a semi-batch reactor, the two formulations will be different, but they will approach the same values if the feeding to the semi-batch reactor ends.

With this in mind, attention can be given to the energy balance of the reactor. The semi-batch reactor is an open system, and the energy balance will take effect from the mass flow dynamics of the semi-batch reactor. This represents a difference from the regular batch-reactor, which is a closed system without external convective flows. In the following expression, h and h_0 denote the specific enthalpy of the fluid in the reactor and the fluid feed, respectively. Note that due to the assumption of a perfectly mixed reactor vessel, the intensive properties will be the same in the entire reactor, and the specific enthalpy of the exiting fluid will hence be that of the uniform fluid residing in the reactor. The simplified energy balance, where external shaft work applied to the reactor is neglected⁵, is proposed in Eq. C.12. In addition, the enthalpy changes due to chemical reactions in the reactor system is important to consider in the modeling work.

$$\frac{dU}{dt} = Q + \hat{m}_{in} \cdot h_0 - \hat{m}_{out} \cdot h \quad (\text{C.12})$$

For many reactor applications, the very purpose of modeling the energy balance is to achieve information about the temperature, which is a quantity of key importance in most cases, with respect to monitoring the behavior of the system. For a typical free-radical copolymerization reaction system for instance, as described in Sec. 2, the reactor temperature is of key importance for both the performance and safety considerations of the chemical reactor. Another reason for bothering with having the temperature as a key process variable is the fact that temperature, in contradiction to energy and enthalpy, is an intuitive and tangible quantity. For many reactor applications, the temperature is also easy to measure without having to deal with a significant time delay, thus providing a safe and reasonably accurate indirect measure of the state of the reactor. By modifying the energy balance (Eq. C.12), the expression in Eq. C.13 is achieved, thus allowing to specifically monitor the change in temperature over time for the reactor.

$$\begin{aligned} \frac{d}{dt} \left(\underbrace{m_{RCp,R}(T - T_{ref})}_{\text{Reactor vessel}} + \underbrace{mc_p(T - T_{ref})}_{\text{Reactor contents}} \right) &= Q + \hat{m}_{in} \cdot h_0 - \hat{m}_{out} \cdot h \\ \implies \frac{dT}{dt} &= \frac{Q + \hat{m}_{in} \cdot h_0 - \hat{m}_{out} \cdot h}{m_{RCp,R} + mc_p} \end{aligned} \quad (\text{C.13})$$

In typical semi-batch reactor cases, the collection of balance equations add to yield a system of ordinary differential equations which must be solved simultaneously. This concludes the brief discussion on first principles modeling of semi-batch reactors.

⁵This assumption does not generally hold. For many tank reactors, however, it does hold despite the fact that the tank reactor is agitated, because the agitator contribution is small compared to the contributions from chemical reactions and cooling/heating.

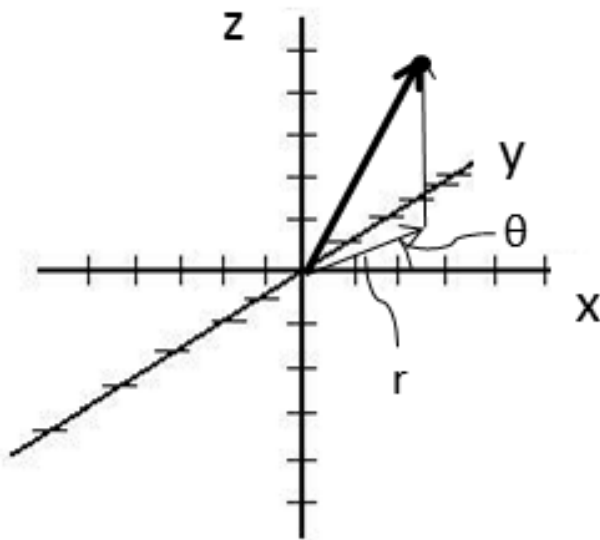
Appendix D

Shell balance derivation of balance principles, using cylindrical coordinates

This additional section is added to the thesis to provide an alternative derivation of the balance principle equation (Eq. 3.7) which was established in Sec. 3.1. Here, a shell balance approach is considered in two dimensions. It is emphasized that the third spatial dimension, which for cylindrical coordinates is the azimuthal (angular) coordinate, is omitted because all the cases which are considered in this thesis has the property of centric symmetry and negligible gravitational effects. Under these assumptions, only the radial (r) and axial (z) coordinates are considered.

D.1 Transformation from cartesian type coordinates to cylindrical coordinates

First of all, the transformation from cartesian type coordinates to cylindrical coordinates is shown. The three-dimensional spatial coordinate system is illustrated in Fig. D.1a, while the corresponding equations are given in Fig. D.1b. In this transformation, the xy -plane represents



(a) Coordinate system

$$r = \sqrt{x^2 + y^2} \quad (\text{D.1})$$

$$\theta = \arctan\left(\frac{y}{x}\right) \quad (\text{D.2})$$

$$z = z \quad (\text{D.3})$$

$$x = r \cos(\theta) \quad (\text{D.4})$$

$$y = r \sin(\theta) \quad (\text{D.5})$$

$$z = z \quad (\text{D.6})$$

(b) Corresponding equations

Figure D.1: Coordinate system and corresponding equations for the transformation between cartesian coordinates and cylindrical coordinates.

the cross-section plane of the typical reactor, while the axial z -coordinate is untampered with. The length of the coordinate component in the xy -plane is denoted by r while the angle between the x -axis and the coordinate component in the xy -plane is denoted by the azimuthal angle, θ .

These relationships are summarized in Eqs. D.1 - D.6. The unit vectors (denoted by a hat) of the cylindrical system are given in Eqs. D.7 - D.9. An interesting observation is that the unit vectors of both the radial and the azimuthal (angular) coordinates are functions of the angular coordinate itself.

$$\hat{r} = \hat{x}\cos(\theta) \quad (\text{D.7})$$

$$\hat{\theta} = -\hat{x}\sin(\theta) + \hat{y}\cos(\theta) \quad (\text{D.8})$$

$$\hat{z} = \hat{z} \quad (\text{D.9})$$

The partial derivatives of this transformed system is given in Eq. D.10, where Eqs. D.1 - D.3 are differentiated with respect to x , y and z , respectively. In addition, the derivatives of the unit vectors from Eqs. D.7 - D.9 are differentiated to yield the result in Eq. D.11.

$$\frac{\partial(r, \theta, z)}{\partial(x, y, z)} = \begin{pmatrix} \frac{x}{\sqrt{x^2+y^2}} & \frac{y}{\sqrt{x^2+y^2}} & 0 \\ \frac{-y}{\sqrt{x^2+y^2}} & \frac{x}{\sqrt{x^2+y^2}} & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (\text{D.10})$$

$$\frac{\partial(\hat{r}, \hat{\theta}, \hat{z})}{\partial(r, \theta, z)} = \begin{pmatrix} 0 & 0 & 0 \\ \hat{\theta} & -\hat{r} & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad (\text{D.11})$$

From this result, the *del* operator, i.e. the operator for the spatial derivatives of a function, can be developed, like shown in Eq. D.17. In order to arrive at this result however, an incremental step in the cylindrical coordinate system must first be established. This is formulated in Eq. D.13.

$$d\vec{s} = d(r\hat{r} + z\hat{z}) \quad (\text{D.12})$$

$$\implies d\vec{s} = \hat{r}dr + \hat{\theta}rd\theta + \hat{z}dz \quad (\text{D.13})$$

The del operator for cartesian coordinates are shown in Eq. D.14.

$$\vec{\nabla} = \left[\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z} \right] = \hat{x}\frac{\partial}{\partial x} + \hat{y}\frac{\partial}{\partial y} + \hat{z}\frac{\partial}{\partial z} \quad (\text{D.14})$$

A way to find the del operator in cylindrical coordinates is to formulate the change in an arbitrary function (f) in two different ways. On one hand, the change is expressed by the total differential (Eq. D.15) while the other uses the incremental step from Eq. D.13 (Eq. D.16).

$$df = \frac{\partial f}{\partial r}dr + \frac{\partial f}{\partial \theta}d\theta + \frac{\partial f}{\partial z}dz \quad (\text{D.15})$$

$$df = \vec{\nabla}f \cdot d\vec{s} = \left(\vec{\nabla}f\right)_r dr + \left(\vec{\nabla}f\right)_\theta rd\theta + \left(\vec{\nabla}f\right)_z dz \quad (\text{D.16})$$

By equating the two expressions and collecting the terms in their respective coordinates, the del operator for the cylindrical coordinate system becomes as indicated in Eq. D.17.

$$\vec{\nabla} = \left[\frac{\partial}{\partial r}, \frac{1}{r}\frac{\partial}{\partial \theta}, \frac{\partial}{\partial z} \right] = \hat{r}\frac{\partial}{\partial r} + \frac{\hat{\theta}}{r}\frac{\partial}{\partial \theta} + \hat{z}\frac{\partial}{\partial z} \quad (\text{D.17})$$

The last step of this section is to establish the divergence of a vector field in cylindrical coordinates. This is the operation which is performed in the vector calculus approach to the balance

principle equation in Sec. 3.1. The established del operator from Eq. D.17 will be deployed directly in this case. Here, \vec{F} is an arbitrary vector field. The result is shown in Eq. D.23.

$$\vec{\nabla} \cdot \vec{F} = \left[\hat{r} \frac{\partial}{\partial r}, \frac{\hat{\theta}}{r} \frac{\partial}{\partial \theta}, \hat{z} \frac{\partial}{\partial z} \right] \cdot \vec{F} \quad (\text{D.18})$$

$$= \left[\hat{r}, \frac{\hat{\theta}}{r}, \hat{z} \right] \cdot \left[\frac{\partial F}{\partial r}, \frac{\partial F}{\partial \theta}, \frac{\partial F}{\partial z} \right] \quad (\text{D.19})$$

$$= \hat{r} \left(\frac{\partial F_r}{\partial r} \hat{r} + \frac{\partial F_\theta}{\partial r} \hat{\theta} + \frac{\partial F_z}{\partial r} \hat{z} + \frac{\partial \hat{r}}{\partial r} F_r + \frac{\partial \hat{\theta}}{\partial r} F_\theta + \frac{\partial \hat{z}}{\partial r} F_z \right) \quad (\text{D.20})$$

$$+ \hat{\theta} \left(\frac{\partial F_r}{\partial \theta} \hat{r} + \frac{\partial F_\theta}{\partial \theta} \hat{\theta} + \frac{\partial F_z}{\partial \theta} \hat{z} + \frac{\partial \hat{r}}{\partial \theta} F_r + \frac{\partial \hat{\theta}}{\partial \theta} F_\theta + \frac{\partial \hat{z}}{\partial \theta} F_z \right) \quad (\text{D.21})$$

$$+ \hat{z} \left(\frac{\partial F_r}{\partial z} \hat{r} + \frac{\partial F_\theta}{\partial z} \hat{\theta} + \frac{\partial F_z}{\partial z} \hat{z} + \frac{\partial \hat{r}}{\partial z} F_r + \frac{\partial \hat{\theta}}{\partial z} F_\theta + \frac{\partial \hat{z}}{\partial z} F_z \right) \quad (\text{D.22})$$

By introducing the respective partial derivatives from Eq. D.11 and collecting the terms, recalling that the unit vectors of the terms must agree to produce a contribution other than zero to the divergence⁶, the final results is obtained. This expression is shown in Eq. D.23.

$$\vec{\nabla} \cdot \vec{F} = \frac{1}{r} \frac{\partial}{\partial r} (r F_r) + \frac{1}{r} \frac{\partial F_\theta}{\partial \theta} + \frac{\partial F_z}{\partial z} \quad (\text{D.23})$$

D.2 Shell balance derivation

A typical cylindrical shell is presented in Fig. D.2, having an inner radius r and an outer radius $r + dr$. The length of the shell stretches from z to $z + dz$. As in Sec. 3.1, the general quantity

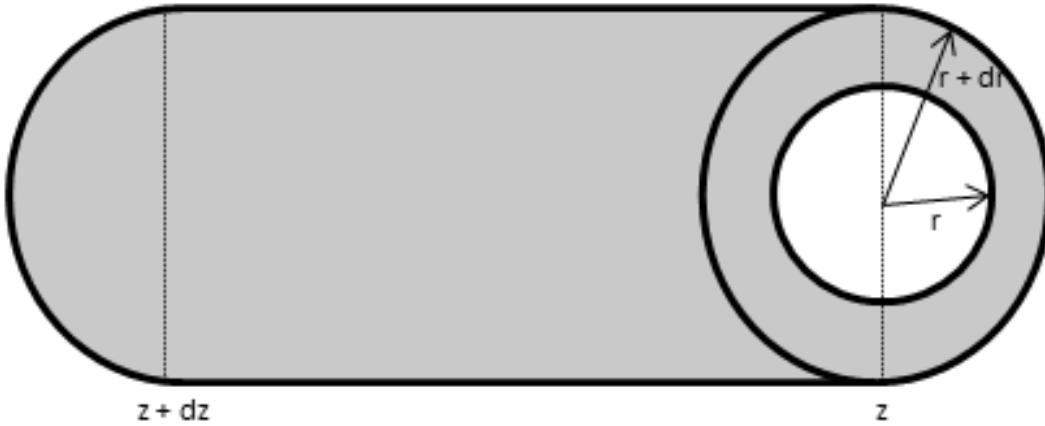


Figure D.2: A cylindrical shell for the development of a shell balance

for which the balance equation is derived is denoted by ψ and the net generation of ψ is denoted

⁶To exemplify, $\hat{r}\hat{r} = 1$, while $\hat{r}\hat{z} = 0$.

by σ . Assuming positive fluid flow along the positive direction of the respective coordinate axes, the accumulation of the quantity (ψ) in the interior of the shell is given by Eq. D.24. In these formulations, V denotes the volume of the shell, A_r denotes the surface area of the shell in the r -direction, while A_z denotes the surface area in the z -direction.

$$\frac{\partial}{\partial t} (\psi V) = (A_r v_r \psi) \Big|_r - (A_r v_r \psi) \Big|_{r+dr} + (A_z v_z \psi) \Big|_z - (A_z v_z \psi) \Big|_{z+dz} + \sigma \quad (\text{D.24})$$

$$V = \left(\pi(r + dr)^2 - \pi r^2 \right) dz \approx 2\pi r dr dz \quad (\text{D.25})$$

$$A_r \Big|_r = 2\pi r dz \quad (\text{D.26})$$

$$A_r \Big|_{r+dr} = 2\pi(r + dr) dz \quad (\text{D.27})$$

$$A_z \Big|_z = \left(\pi(r + dr)^2 - \pi r^2 \right) dz \approx 2\pi r dr \quad (\text{D.28})$$

$$A_z \Big|_{z+dz} = \left(\pi(r + dr)^2 - \pi r^2 \right) dz \approx 2\pi r dr \quad (\text{D.29})$$

$$\implies \frac{\partial \psi}{\partial t} = \frac{(rv_r \psi) \Big|_r - (rv_r \psi) \Big|_{r+dr}}{r dr} + \frac{(v_z \psi) \Big|_z - (v_z \psi) \Big|_{z+dz}}{dz} + \sigma \quad (\text{D.30})$$

By making the shell infinitesimally small, the spatial derivatives are recognized after careful examination. The definition for the differentiation of a general function is given in Eq. D.31. Using this definition, the balance equation can be rewritten to yield Eq. D.33.

$$f'(x) = \lim_{dx \rightarrow 0} \frac{f(x + dx) - f(x)}{dx} \quad (\text{D.31})$$

$$\frac{\partial \psi}{\partial t} = \lim_{dr \rightarrow 0} \frac{(rv_r \psi) \Big|_r - (rv_r \psi) \Big|_{r+dr}}{r dr} + \lim_{dz \rightarrow 0} \frac{(v_z \psi) \Big|_z - (v_z \psi) \Big|_{z+dz}}{dz} + \sigma \quad (\text{D.32})$$

$$\implies \frac{\partial \psi}{\partial t} = -\frac{1}{r} \frac{\partial}{\partial r} (rv_r \psi) - \frac{\partial}{\partial z} (v_z \psi) + \sigma \quad (\text{D.33})$$

By comparison, this expression is equal to Eq. 3.7 in Sec. 3.1. This concludes the derivation of the balance equation for a generalized quantity using the shell balance approach. As mentioned in Sec. 3.1, this approach gives an expression which is similar to the one found by the vector field calculus approach.

Appendix E

Example calculation using the numerical method of lines

The purpose of this additional section is to include an example on how the numerical method of lines (NMOL), which is introduced in Sec. 3.3, can be used to solve a specific partial differential equation. The NMOL is a numerical method based on approximation of derivatives by the use of finite differences, as introduced in Sec. 3.3. This specific example was inspired by Cutlip & Shacham [27], who conducted a similar calculation for a heat transfer problem.

E.1 Problem formulation

The equation to be solved is the one-dimensional equation for unsteady-state heat transfer, which is presented in Eq. E.1. This represents a rod or a wall conducting heat in the x -direction only. The thermal diffusivity in the equation is calculated as shown in Eq. E.2. Here, k denotes the thermal conductivity, ρ is the density and c_p is the heat capacity of the medium through which the heat travels. For this specific example, the thermal diffusivity is set to a fictional value of $5e^{-5} \text{ m}^2/\text{s}$.

$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2} \quad (\text{E.1})$$

$$\alpha = \frac{k}{\rho c_p} \quad (\text{E.2})$$

The problem in this specific example is presented with the boundary conditions given in Eqs. E.3 - E.5. These boundary conditions are of the Robin type, often referred to as type 3 boundary conditions (after Dirichlet and von Neumann types). For Robin boundary conditions, a mix of solutions and derivative values constitute the set of boundary conditions, whereas the Dirichlet type only contains solutions and the von Neumann type only contains derivatives. The classification of boundary conditions are elaborated in Sec. 3.1, and the boundary conditions for this example are in the same class as the boundary conditions which are deployed for the tubular reactor in the main part of the thesis. The boundary conditions correspond to the physical interpretation of the system, as illustrated in Fig. E.1, where a one-dimensional wall with a total length of 1.00 m , leaning on another non-conducting wall, is discretized into ten separate slabs. At the beginning of the simulation ($t = 0$), the interior of the wall holds a constant temperature of 298.15 K (Eq. E.3). The surrounding environment holds a constant temperature of 600 K throughout the simulation (Eq. E.4), while the heat transfer through the innermost discrete element of the wall is zero, on account of the neighboring wall being completely insulated (Eq. E.5).

$$t = 0 : \quad T_i = 298.15 \quad , \quad i = 2, \dots, n + 1 \quad (\text{E.3})$$

$$t \leq 0 : \quad T_1 = 600 \quad (\text{E.4})$$

$$\left. \frac{\partial T}{\partial x} \right|_{n+1} = 0 \quad (\text{E.5})$$

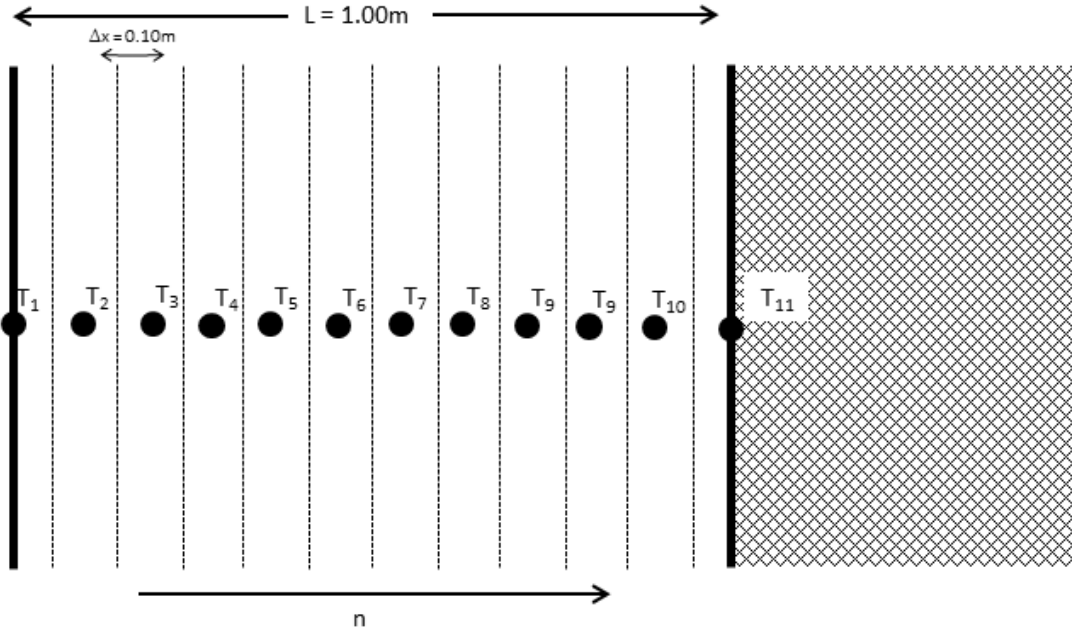


Figure E.1: A conceptual sketch of a one-dimensional, unsteady-state heat conduction problem.

E.2 Deploying NMOL

The strategy of the NMOL is to discretize the spatial dimension of the problem using finite differences, as indicated in Fig. E.1, hence reducing the partial differential equation to an ordinary differential equation (in time) to be solved with a suitable ODE solver. For the purpose of this example, the built-in solver `ode15s` of MATLAB was utilized. Using central differences throughout the wall to express the second derivative with respect to the spatial coordinate, the expression in Eq. E.6 is achieved. In the same spirit, the first derivative with respect to the spatial coordinate is found at end of the wall (to be used in conjunction with the boundary condition in Eq. E.5) using a backwards difference scheme as depicted in Eq. E.7. [28]

$$\frac{\partial^2 T_i}{\partial x^2} \approx \frac{T_{i+1} - 2T_i + T_{i-1}}{(\Delta x)^2}, \quad i = 2, \dots, n \quad (\text{E.6})$$

$$\frac{\partial T_i}{\partial x} \approx \frac{3T_i - 4T_{i-1} + T_{i-2}}{2\Delta x}, \quad i = n + 1 \quad (\text{E.7})$$

This formulation effectively reduces the partial differential equation to a set of ordinary differential equations representing the change in temperature for each of the discrete points in the wall. This system of equations is indicated in Eqs. E.8 - E.10.

$$T_1 = 600 \quad (\text{E.8})$$

$$\frac{\partial T_i}{\partial t} = \frac{T_{i+1} - 2T_i + T_{i-1}}{(\Delta x)^2}, \quad i = 2, \dots, n \quad (\text{E.9})$$

$$T_{11} = \frac{4T_{i-1} - T_{i-2}}{3} \quad (\text{E.10})$$

The solution to this system of ODEs is formulated in MATLAB, as indicated in Cs. E.1 & Cs. E.2. The resulting temperature profiles from the simulations are shown in Fig. E.2, in which a 3D mesh plot is provided. As dictated by the boundary conditions, the temperature of the outermost discrete point in the wall (T_1) is constant, and the interior points of the wall expels the expected time-varying behavior.

```

1 % Solution of unsteady-state heat transfer equation
2 % Written by: Fredrik Gjertsen, Cybernetica AS
3
4 clc
5 close all
6
7 L = 1; % Total length of the slab [m]
8 n = 10; % Number of discrete elements
9 dx = L/(n); % Thickness of each element
10 xspan = 0:dx:L; % Spatial x-span
11 tspan = [0 3000]; % Timespan for solution
12 T0 = 298.15; % Initial slab temperature
13 T1 = 600; % Outside temperature
14 a = 5e-5; % Thermal diffusivity
15
16 MassMatrix = diag([0,ones(1,n-1),0]); % Defining a mass matrix
17 M = sparse(MassMatrix); % (exploiting sparsity)
18 opts = odeset('RelTol',1e-2,'Mass',M); % Options for the ODE solver
19 [t y] = ode15s(@heatTransFun,tspan,[T1;T0.*ones(n,1)],opts,T1,dx,a);
20
21 figure(2) % 3D mesh plot for temperature development
22 set(figure(2),'Position',[300 300 800 800]);
23 mesh(xspan,t(:,1)./60,y);
24 title('Temperature variation (in time and space)');
25 xlabel('Spatial coordinate, x [m]');
26 ylabel('Time [min]');
27 zlabel('Temperature [K]');
28 colormap(cool);
29 colorbar('EastOutside');

```

Code snippet E.1: The main program for the unsteady-state heat transfer problem, declaring variables, solving the set of ODEs and plotting results.

This concludes the simplified example on the numerical method of lines.

```

1 function dT = heatTransFun(t,T,T1,dx,alpha)
2     dT = zeros(length(T),1);
3     dT(1) = T(1) - T1;           % T1 does not change
4     for i = 2:length(T)-1
5         dT(i) = alpha*(T(i+1) - 2*T(i) + T(i-1))/dx^2;
6     end
7     dT(length(T)) = 3*T(length(T)) - 4*T(length(T)-1) + T(length(T)-2);
8 end

```

Code snippet E.2: A function defining the differential equations for the unsteady-state heat transfer problem.

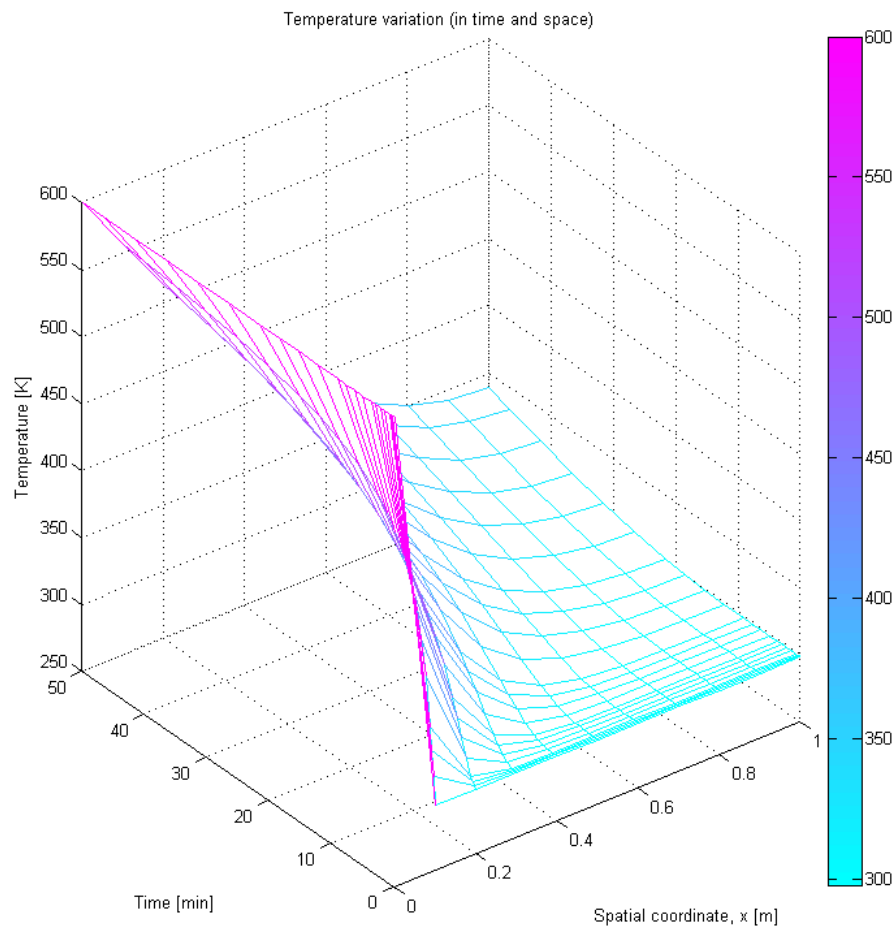


Figure E.2: Resulting temperature profiles from solving the heat conduction problem.

Appendix F

Estimator derivation

This additional section is written to include details on the derivation of the Kalman filter (KF) estimator equations used for on-line estimation, as discussed in Sec. 5. The theory presented in this appendix is adopted from the preliminary specialization project written by the author of this thesis. The Kalman filter algorithm equations are readily derived for off-line considerations, but they become truly valuable for the complete on-line MPC implementation. [1]

The definition of an estimator is of key importance when designing an on-line controller implementation, and the tuning of the estimator may sometimes prove to be harder than designing the controller itself. In Sec. F.1, the Kalman filter for a linear time-discrete system is established as a starting point, and this is extended to apply for continuous linear systems as well in Sec. F.2. In Sec. F.3, the (Extended) Kalman filter (EKF) for nonlinear systems is introduced. The following derivation adopts the approach, and hence the notation, of Simon. [22]

F.1 Kalman filter estimator for linear time-discrete systems

The starting point for the derivation of the Kalman filter is a general linear time-discrete system. This system is represented, in a general formulation, in Eqs. F.3 & F.4. In the case where the linear system is not time-discrete but continuous, which indeed is the case for most dynamic systems, the linear continuous system is converted to a discrete system. The formulation of the continuous system is shown in Eq. F.1, where the dotted notation indicates differentiation with respect to time.

$$\dot{x} = Ax + Bu + \omega \quad (\text{F.1})$$

The discretized solution⁷ to such a system is indicated in Eq. F.2, in which x_k is the vector of states and u_k is the vector of inputs at time t_k . The discrete time interval $t_k - t_{k-1}$ is denoted by Δt_k .

$$x_k = e^{A\Delta t_k} x_{k-1} + \int_{t_{k-1}}^{t_k} e^{A(t_k-\tau)} [B(\tau)u(\tau) + \omega(\tau)] d\tau \quad (\text{F.2})$$

The general formulation for the linear time-discrete system will then be as presented in Eqs. F.3 & F.4.

$$x_k = F_{k-1}x_{k-1} + G_{k-1}u_{k-1} + w_{k-1} \quad (\text{F.3})$$

$$y_k = H_k x_k + v_k \quad (\text{F.4})$$

In this system, x_k is the vector of states at the discrete point t_k in time. The corresponding vector of measurements, y_k , is represented by the sum of the state measurements ($H_k x_k$) and the measurement noise (v_k). Here, H_k is a matrix deciding which of the states that go into the measurement vector. F_k and G_k are the system matrices, governing how the states and inputs

⁷These expressions contain mathematical terms such as matrix exponentials, convolution integrals, and other complex concepts. These concepts are not elaborated in this text, and the curious reader is referred to more extensive texts for a walkthrough, e.g. Kreyszig. [23]

(MVs), respectively, at time t_k affect the state at time t_{k+1} . In special situations where F and G are time independent (constant) matrices, the system is said to be linear time-independent (LTI). The process noise at time t_k is represented by w_k . Both noise terms (w_k and v_k) are assumed to be zero-mean white noise with covariances⁸ denoted by Q_k and R_k , respectively. The zero-mean assumption is justified by the requirement that systematic mean-contributions from the noise are accounted for in the model, i.e. the system matrices. By comparison between the solution for the discretized continuous system (Eq. F.2) and the general discrete system (Eq. F.3), the system matrices must be as indicated in Eqs. F.5 - F.7.

$$F_{k-1} = e^{A\Delta t_k} \quad (\text{F.5})$$

$$G_{k-1} = \int_{t_{k-1}}^{t_k} e^{A(t_k-\tau)} B d\tau \quad (\text{F.6})$$

$$w_{k-1} = \int_{t_{k-1}}^{t_k} e^{A(t_k-\tau)} \omega(\tau) d\tau \quad (\text{F.7})$$

The purpose of the estimator is to propose an optimal estimate of the state at time t_k (x_k), and this estimate is denoted by \hat{x}_k . A distinction is made between so-called *a priori* estimates and *a posteriori*⁹ estimates, which are denoted \hat{x}_k^- and \hat{x}_k^+ , respectively. In this sense, whether an estimate at time t_k is *a priori* or *a posteriori* is a question of whether the measurement information at time t_k is used or not. Mathematically, this can be formulated as shown in Eqs. F.8 & F.9 for the *a priori* and *a posteriori* estimates, respectively. It is emphasized that \hat{x}_k^- and \hat{x}_k^+ are estimates of the same quantity, i.e. the state vector at time t_k (x_k).

$$\hat{x}_k^- = E \left[x_k \mid y_1, y_2, \dots, y_{k-1} \right] \quad (\text{F.8})$$

$$\hat{x}_k^+ = E \left[x_k \mid y_1, y_2, \dots, y_k \right] \quad (\text{F.9})$$

Here, $E[\dots]$ denotes the (statistically) expected value. As is evident from Eqs. F.8 & F.9, the two estimates use the same previous measurements (y_1, y_2, \dots, y_{k-1}). The only difference is that the *a posteriori* estimate also utilizes the measurement at time t_k (y_k).

From this point on, P_k is introduced to denote the covariance of the state estimate error at time t_k . Notice that the state covariance also has a distinction between *a priori* and *a posteriori* information, as indicated in Eqs. F.10 & F.11.

$$P_k^- = E \left[\left(x_k - \hat{x}_k^- \right) \left(x_k - \hat{x}_k^- \right)^T \right] \quad (\text{F.10})$$

$$P_k^+ = E \left[\left(x_k - \hat{x}_k^+ \right) \left(x_k - \hat{x}_k^+ \right)^T \right] \quad (\text{F.11})$$

The system is initiated by an initial state, x_0^+ , having a covariance P_0^+ . In the case where x_0^+ is well known, the covariance (i.e. uncertainty) is zero, but in the case where x_0^+ is (very) uncertain,

⁸Details on theoretical concepts from the field of statistics are omitted from this text. For a theoretical background on statistics deployed in the thesis, Walpole *et. al.* was consulted. [26]

⁹These terms are originally known from philosophy. *A priori* and *a posteriori* mean "from the earlier" and "from the later", respectively, and these terms are used to describe the "quality" of the knowledge.

P_0^+ approaches ∞I . The next step is to analyze how the state estimates and covariances propagate through time. Using the established definition in Eq. F.8 in combination with the proposed model in Eq. F.3, the general propagation of state estimates becomes as indicated in Eq. F.12. A justification for this expression is that the process noise (w_k) has a zero-mean probability distribution.

$$\begin{aligned} \hat{x}_1^- &= F_0 \hat{x}_0^+ + G_0 u_0 \\ \text{General expression: } \hat{x}_k^- &= F_{k-1} \hat{x}_{k-1}^+ + G_{k-1} u_{k-1} \end{aligned} \quad (\text{F.12})$$

The propagation of the covariances is slightly more complicated. The approach starts from the definition in Eq. F.10, the model in Eq. F.3 and the result in Eq. F.12. The result, showing how the covariances propagate in time, is provided in Eq. F.13.

$$\begin{aligned} P_k^- &= E \left[\left(x_k - \hat{x}_k^- \right) \left(x_k - \hat{x}_k^- \right)^T \right] \\ \left(x_k - \hat{x}_k^- \right) &= F_{k-1} x_{k-1} + G_{k-1} u_{k-1} + w_{k-1} - \hat{x}_k^- \\ \hat{x}_k^- &= F_{k-1} \hat{x}_{k-1}^+ + G_{k-1} u_{k-1} \\ \implies \left(x_k - \hat{x}_k^- \right) &= F_{k-1} \left(x_{k-1} - \hat{x}_{k-1}^+ \right) + w_{k-1} \\ \left(x_k - \hat{x}_k^- \right) \left(x_k - \hat{x}_k^- \right)^T &= F_{k-1} \left(x_{k-1} - \hat{x}_{k-1}^+ \right) \left(x_{k-1} - \hat{x}_{k-1}^+ \right)^T F_{k-1}^T + w_{k-1} w_{k-1}^T \\ &\quad + F_{k-1} \left(x_{k-1} - \hat{x}_{k-1}^+ \right) w_{k-1}^T + w_{k-1} \left(x_{k-1} - \hat{x}_{k-1}^+ \right)^T F_{k-1}^T \\ \implies P_k^- &= F_{k-1} P_{k-1}^+ F_{k-1}^T + Q_{k-1} \end{aligned} \quad (\text{F.13})$$

This result was found by recognizing the expected value of each of the terms in the expression for $\left(x_k - \hat{x}_k^- \right) \left(x_k - \hat{x}_k^- \right)^T$, once it was written out. This is illustrated in Eq. F.14. Eq. F.15 is in agreement with the proposed covariance of the process noise, while Eqs. F.16 & F.17 agrees with the fact that the estimation error $\left(x_k - \hat{x}_k^- \right)$ is uncorrelated with the process noise (w_k).

$$\begin{aligned} E \left[F_{k-1} \left(x_{k-1} - \hat{x}_{k-1}^+ \right) \left(x_{k-1} - \hat{x}_{k-1}^+ \right)^T \right] &= F_{k-1} E \left[\left(x_{k-1} - \hat{x}_{k-1}^+ \right) \left(x_{k-1} - \hat{x}_{k-1}^+ \right)^T \right] F_{k-1}^T \\ &= F_{k-1} P_{k-1}^+ F_{k-1}^T \end{aligned} \quad (\text{F.14})$$

$$E \left[w_{k-1} w_{k-1}^T \right] = Q_{k-1} \quad (\text{F.15})$$

$$E \left[w_{k-1} \left(x_{k-1} - \hat{x}_{k-1}^+ \right)^T F_{k-1}^T \right] = 0 \quad (\text{F.16})$$

$$E \left[F_{k-1} \left(x_{k-1} - \hat{x}_{k-1}^+ \right) w_{k-1}^T \right] = 0 \quad (\text{F.17})$$

It has now been investigated how both state estimates and covariances propagate through time for a linear time-discrete system, as indicated by Eqs. F.12 & F.13, respectively. These equations give the relationships for the transitions $\hat{x}_k^+ \rightarrow \hat{x}_{k+1}^-$ and $P_k^+ \rightarrow P_{k+1}^-$, i.e. the propagation in time. The next step is to formulate expressions for the transitions $\hat{x}_k^- \rightarrow \hat{x}_k^+$ and $P_k^- \rightarrow P_k^+$, i.e. the transition from *a priori* estimates to *a posteriori* estimates at time t_k . This change is done by using the so-called approach of recursive least squares estimation. By having a measurement (y_k) as indicated in Eq. F.4, the measurement correction is formulated as indicated in Eq. F.18.

Here, K_k is the estimator gain matrix, which becomes the Kalman filter gain when the errors between the states and the state estimates are minimized.

$$\hat{x}_k^+ = \hat{x}_k^- + K_k (y_k - H_k \hat{x}_k^-) \quad (\text{F.18})$$

To minimize the estimation errors, a proper cost function¹⁰ is defined as indicated in Eq. F.19, in which the error ϵ is defined as shown in Eq. F.20.

$$\begin{aligned} J_k &= E \left[(x_1 - \hat{x}_1^+)^2 \right] + E \left[(x_2 - \hat{x}_2^+)^2 \right] + \cdots + E \left[(x_n - \hat{x}_n^+)^2 \right] \\ &= E \left[\epsilon_{x1,k}^2 + \epsilon_{x2,k}^2 + \cdots + \epsilon_{xn,k}^2 \right] \\ &= E \left[\epsilon_{x,k}^T \epsilon_{x,k} \right] \\ &= \text{Tr} \left(E \left[\epsilon_{x,k} \epsilon_{x,k}^T \right] \right) \end{aligned} \quad (\text{F.19})$$

$$\epsilon_{xi} = x_i - \hat{x}_i^+ \quad (\text{F.20})$$

The expected value for the vector of errors is needed, and this expression can be manipulated by combining the error definition (Eq. F.20), the measurement definition (Eq. F.4) and the estimator measurement correction (Eq. F.18). The result is portrayed in Eq. F.21.

$$\begin{aligned} E[\epsilon_{x,k}] &= E \left[x - \hat{x}_k^+ \right] \\ &= E \left[x - \hat{x}_k^- - K_k (y_k - H_k \hat{x}_k^-) \right] \\ &= E \left[x - \hat{x}_k^- - K_k (H_k x + v_k - H_k \hat{x}_k^-) \right] \\ &= E \left[x - \hat{x}_k^- - K_k H_k (x - \hat{x}_k^-) - K_k v_k \right] \\ &= (I - K_k H_k) E \left[x - \hat{x}_k^- \right] - K_k E[v_k] \end{aligned} \quad (\text{F.21})$$

This result is possible to expand to give the expected value for the square of the error, as shown in Eq. F.22.

$$\begin{aligned} E \left[\epsilon_{x,k} \epsilon_{x,k}^T \right] &= (I - K_k H_k) E \left[(x - \hat{x}_k^-) (x - \hat{x}_k^-)^T \right] (I - K_k H_k)^T \\ &\quad - K_k E \left[v_k (x - \hat{x}_k^-)^T \right] + K_k E \left[v_k v_k^T \right] K_k^T \\ &\quad - (I - K_k H_k) E \left[(x - \hat{x}_k^-) v_k^T \right] K_k^T \\ \implies E \left[\epsilon_{x,k} \epsilon_{x,k}^T \right] &= (I - K_k H_k) P_k^- (I - K_k H_k)^T + K_k R_k K_k^T \end{aligned} \quad (\text{F.22})$$

¹⁰In this formulation, Tr denotes the Trace of the matrix. This is a matrix operation which consists of calculating the sum of the elements on the diagonal of the matrix.

This is achieved by recognizing the definition of P_k^- , as well as remembering that the noises are uncorrelated with the error. R_k is, as postulated earlier, the covariance of the measurement noise at time t_k . These requirements are summarized in Eqs. F.23 - F.25.

$$P_k^- = E \left[\left(x_k - \hat{x}_k^- \right) \left(x_k - \hat{x}_k^- \right)^T \right] \quad (\text{F.23})$$

$$E \left[v_k \left(x - \hat{x}_k^- \right) \right] = 0 \quad (\text{F.24})$$

$$E \left[v_k v_k^T \right] = R_k \quad (\text{F.25})$$

By remembering the definition of P_k^+ from Eq. F.11, these previous results also enable the formulation of P_k^+ , as represented in Eq. F.26.

$$\begin{aligned} P_k^+ &= E \left[\epsilon_{x,k} \epsilon_{x,k}^T \right] \\ \implies P_k^+ &= (I - K_k H_k) P_k^- (I - K_k H_k)^T + K_k R_k K_k^T \end{aligned} \quad (\text{F.26})$$

The purpose of the next steps is to solve an optimization problem to uncover the optimal estimator gain, K_k , which is called the Kalman filter gain. To do this, the cost function (accounting for the estimator error), is differentiated with respect to K_k and set equal to zero. This procedure is shown in Eqs. F.27 - F.31. The step between Eq. F.30 and Eq. F.31 is performed by postulating that $\frac{\partial(\text{Tr}(ABA^T))}{\partial A} = 2AB$, given that B is symmetric. In this case, this is assumed to be fulfilled.

$$\frac{\partial J}{\partial K_k} = 0 \quad (\text{F.27})$$

$$\frac{\partial \left(\text{Tr} \left(E \left[\epsilon_{x,k} \epsilon_{x,k}^T \right] \right) \right)}{\partial K_k} = 0 \quad (\text{F.28})$$

$$\frac{\partial}{\partial K_k} \left(\text{Tr} \left((I - K_k H_k) P_k^- (I - K_k H_k)^T + K_k R_k K_k^T \right) \right) = 0 \quad (\text{F.29})$$

$$\frac{\partial \left(\text{Tr} \left((I - K_k H_k) P_k^- (I - K_k H_k)^T \right) \right)}{\partial K_k} + \frac{\partial \left(\text{Tr} \left(K_k R_k K_k^T \right) \right)}{\partial K_k} = 0 \quad (\text{F.30})$$

$$2(I - K_k H_k) P_k^- (-H_k^T) + 2K_k R_k = 0 \quad (\text{F.31})$$

Eq. F.31 is reorganized to yield the final expression for the Kalman filter gain, which is shown in Eq. F.32.

$$\implies K_k = P_k^- H_k^T \left(H_k P_k^- H_k^T + R_k \right)^{-1} \quad (\text{F.32})$$

Finally, the resulting equations for the Kalman filter algorithm are listed in Eqs. F.33 - F.37.

This system of equations is usually referred to as Riccati-equations.

$$P_k^- = F_{k-1} P_{k-1}^+ F_{k-1}^T + Q_{k-1} \quad (\text{F.33})$$

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} \quad (\text{F.34})$$

$$\hat{x}_k^- = F_{k-1} \hat{x}_{k-1}^+ + G_{k-1} u_{k-1} \quad (\text{F.35})$$

$$\hat{x}_k^+ = \hat{x}_k^- + K_k (y_k - H_k \hat{x}_k^-) \quad (\text{F.36})$$

$$P_k^+ = (I - K_k H_k) P_k^- (I - K_k H_k)^T + K_k R_k K_k^T \quad (\text{F.37})$$

In some cases, it is desirable to combine the measurement updates and the estimation propagation in time to yield one expression including both. In other words, the *a priori* state estimate update from time t_{k-1} to time t_k is combined with the *a posteriori* update at time t_k , and the same strategy is applied to the covariances (P_k). This is illustrated in Eqs. F.38 & F.39.

$$\left. \begin{array}{l} \hat{x}_{k-1}^+ \rightarrow \hat{x}_k^- \\ \hat{x}_k^- \rightarrow \hat{x}_k^+ \end{array} \right\} \implies \hat{x}_{k-1}^+ \rightarrow \hat{x}_k^+ \quad (\text{F.38})$$

$$\implies \hat{x}_k^+ = (I - K_k H_k) (F_{k-1} \hat{x}_{k-1}^+ + G_{k-1} u_{k-1}) + K_k y_k$$

$$\left. \begin{array}{l} P_{k-1}^+ \rightarrow P_k^- \\ P_k^- \rightarrow P_k^+ \end{array} \right\} \implies P_{k-1}^+ \rightarrow P_k^+ \quad (\text{F.39})$$

$$\implies P_k^+ = (I - K_k H_k) (F_{k-1} P_{k-1}^+ F_{k-1}^T + Q_{k-1})$$

This concludes the establishment of the Kalman filter equations for a linear time-discrete system.

F.2 Kalman filter estimator for linear continuous systems

The purpose of this section is to extend the established state estimation theory for linear time-discrete systems (Sec. F.1) to also account for linear continuous systems. Continuous systems are often encountered in dynamic modeling of systems from real applications, and this is an important class of systems, considering both the linear and the nonlinear cases. In this section, the linear approach is discussed, and the idea is to eventually extend this to also apply for the nonlinear case. The linear continuous system is equal to the one introduced in Sec. F.1, and the measurement dynamics are also included, as indicated in Eqs. F.40 & F.41. As for the linear case, the system is initiated by the initial state estimates (\hat{x}_0^+) which is known to some accuracy, reflected in the covariances of the initial states (P_0^+).

$$\dot{x} = Ax + Bu + w \quad (\text{F.40})$$

$$y = Cx + v \quad (\text{F.41})$$

The strategy for extending the Kalman filter equations to the continuous system is to examine and compare the system matrices for the two system formulations. The system matrices for the discrete system, as introduced in Eqs. F.5 & F.6 in Sec. F.1, are reproduced in Eqs. F.42 & F.43. Notice that the G -matrix has been rewritten¹¹ for the purpose of this section, and that the time indices have been omitted to emphasize the transfer from the discrete formulation to the

¹¹Careful examination between Eq. F.6 and Eq. F.43 confirms that the reformulation is valid.

continuous formulation. Here, T is included to denote a time interval (Δt), which is assumed to be small. Under the assumption that T is small (and later to be infinitesimally small), the matrix exponential¹² expressions can be approximated to yield far simpler expressions.

$$F = e^{AT} \approx (I + AT) \quad (\text{F.42})$$

$$G = e^{AT} [I - e^{-AT}] A^{-1} B \approx BT \quad (\text{F.43})$$

The covariances for the noises of the system must also be considered. For the discrete case, the process noise (w_k) and measurement noise (v_k) at time t_k have covariances Q_k and R_k , respectively, while the continuous case has instantaneous covariances of Q_c and R_c , respectively. For the continuous noise functions, this is formulated mathematically in Eqs. F.44 & F.45.

$$E[w(t)w(\tau)] = Q_c \delta(t - \tau) \quad (\text{F.44})$$

$$E[v(t)v(\tau)] = R_c \delta(t - \tau) \quad (\text{F.45})$$

For this to hold for a small time interval T , it can be shown that the relationships between Q and Q_c and R and R_c must be as indicated in Eqs. F.46 & F.47, respectively. This is related to the total covariance associated with the entire time interval, i.e. sum of time intervals.

$$Q = Q_c \Delta t \quad (\text{F.46})$$

$$R = \frac{R_c}{\Delta t} \quad (\text{F.47})$$

Starting with the Kalman filter gain equation in Eq. F.34 from Sec. F.1, the corresponding terms for the continuous system are inserted to yield the continuous Kalman filter gain, as represented in Eq. F.48.

$$\begin{aligned} K_k &= P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} \\ K_k &= P_k^- C^T \left(C P_k^- C^T + \frac{R_c}{T} \right)^{-1} \\ \frac{K_k}{T} &= P_k^- C^T (C P_k^- C^T T + R_c)^{-1} \\ \lim_{T \rightarrow 0} \frac{K_k}{T} &= P_k^- C^T R_c^{-1} \end{aligned} \quad (\text{F.48})$$

The covariances for the states of the system also need attention. The covariance formulations from Eqs. F.33 & F.37 are reproduced in Eqs. F.49 & F.50, respectively. For the purpose of the continuous formulation, the indices of the system matrices have been omitted, and Eq. F.33 has been reorganized¹³.

$$P_{k+1}^- = F P_k^+ F^T + Q \quad (\text{F.49})$$

$$P_k^+ = (I - K_k C) P_k^- \quad (\text{F.50})$$

¹²The matrix exponential is defined as an infinite series: $e^X = \sum_{k=0}^{\infty} \frac{1}{k!} X^k = I + X + \frac{1}{2!} X^2 + \frac{1}{3!} X^3 + \dots$. It has many properties in common with the traditional exponential function (i.e. the base of the natural logarithm).

¹³The algebraic manipulation of the expression has been omitted from this text, but careful analysis shows that the two expressions are equal. [22]

For small values of T , the approximation in Eq. F.42 is valid, and the equation for the covariances are developed as indicated below. The final result, known as the differential Riccati equation is presented in Eq. F.51.

$$\begin{aligned}
P_{k+1}^- &= FP_k^+ F^T + Q \\
&= (I + AT) P_k^+ (I + AT)^T + Q_c T \\
&= P_k^+ + (AP_k^+ + P_k^+ A^T + Q_c) T + AP_k^+ A^T T^2 \\
&= (I - K_k C) P_k^- + A (I - K_k C) P_k^- A^T T^2 \\
&\quad + [A (I - K_k C) P_k^- p (I - K_k C) P_k^- A^T + Q_c] T \\
\implies \frac{P_{k+1}^- - P_k^-}{T} &= -\frac{K_k C P_k^-}{T} + AP_k^+ A^T T \\
&\quad + (AP_k^- A K_k C P_k^- + P_k^- A^T - K_k C P_k^- A^T + Q_c) \\
\implies \dot{P} = \lim_{T \rightarrow 0} \frac{P_{k+1}^- - P_k^-}{T} &= -PC^T R_c^{-1} CP + AP + PA^T + Q_c \tag{F.51}
\end{aligned}$$

In this operation, the already established expression for the continuous Kalman filter gain, which is presented in Eq. F.48, was applied.

The next step is to establish a differential formulation for the state estimate propagation in time. The strategy is completely analogous to what was done for the covariances in Eq. F.51. The state estimate equations, which are indicated in Eqs. F.35 & F.36 for the discrete case, are rewritten for the continuous case in Eqs. F.52 & F.53.

$$\hat{x}_{k+1}^- = F \hat{x}_k^+ + G u_k \tag{F.52}$$

$$\hat{x}_k^+ = \hat{x}_k^- + K_k (y_k - H \hat{x}_k^-) \tag{F.53}$$

For small values of T , the approximations in Eqs. F.42 & F.43 are valid, and Eqs. F.52 & F.53 are combined to yield the final result presented in Eq. F.54 as shown below.

$$\begin{aligned}
\hat{x}_{k+1}^+ &= F \hat{x}_k^+ + G u_k + K_k (y_k - H F \hat{x}_k^+ - H G u_k) \\
&= (I + AT) \hat{x}_k^+ + B T u_k \\
&\quad + K_k (y_k - C (I + AT) \hat{x}_k^+ - C B T u_k) \\
&= \hat{x}_k^+ + AT \hat{x}_k^- + B T u_k \\
&\quad + PC^T R_c^{-1} T (y_k - C \hat{x}_k^+ - C A T \hat{x}_k^+ - C B T u_k) \\
\implies \frac{\hat{x}_{k+1}^+ - \hat{x}_k^+}{T} &= A \hat{x}_k + B u_k \\
&\quad + PC^T R_c^{-1} (y_k - C \hat{x}_k^+ - C A T \hat{x}_k^+ - C B T u_k) \\
\implies \dot{\hat{x}} = \lim_{T \rightarrow 0} \frac{\hat{x}_{k+1}^+ - \hat{x}_k^+}{T} &= A \hat{x} + B u + K (y - C \hat{x}) \tag{F.54}
\end{aligned}$$

In this operation, the already established expression for the continuous Kalman filter gain, which is presented in Eq. F.48, was applied.

Finally, the resulting equations for the linear continuous system are achieved, as presented in Eqs. F.55 - F.57.

$$\dot{\hat{x}} = A\hat{x} + Bu + K(y - C\hat{x}) \quad (\text{F.55})$$

$$K = PC^T R_c^{-1} \quad (\text{F.56})$$

$$\dot{P} = -PC^T R_c^{-1} CP + AP + PA^T + Q_c \quad (\text{F.57})$$

This system of equations is often referred to as the differential Riccati equations for the Kalman filter estimator.

F.3 Extended Kalman filter estimator for nonlinear continuous systems

The purpose of this section is to briefly describe how Kalman filter estimators can be applied for nonlinear continuous systems. In previous sections, the filtering equations for linear systems have been established. These results provide a useful basis for further considerations, but emphasis is put on the fact that real-life systems which are entirely linear are non-existent. Making linear models to describe a real-life process may sometimes be a valid approach, and the theory for linear estimation is in that sense quite useful, but it is definitely valuable to establish an estimator for nonlinear systems as well.

In some cases, it may be sufficient to approximate the nonlinear model as a linearized model. The procedure of linearization is illustrated in Eqs. F.58 - F.60. Here, Φ is an arbitrary continuous multi-variable function, chosen to illustrate the procedure. The linearization procedure has its basis in the theory of Taylor series, in which a linearized function can be considered to be the first order Taylor series polynomial of the original function, when all higher order terms are neglected. In Eqs. F.58 - F.60, the star (*) indicates nominal trajectory values, from which the deviations in the linearized function are considered. The nominal trajectory is defined as indicated in Eqs. F.61 & F.62.

$$\Phi = f(x_1, x_2, \dots, x_n) \quad (\text{F.58})$$

$$\implies \Phi \approx \Phi_* + \left. \frac{\partial f}{\partial x_1} \right|_* (x_1 - x_1^*) + \left. \frac{\partial f}{\partial x_2} \right|_* (x_2 - x_2^*) + \dots + \left. \frac{\partial f}{\partial x_n} \right|_* (x_n - x_n^*) \quad (\text{F.59})$$

$$\implies \Delta\Phi \approx k_1\Delta x_1 + k_2\Delta x_2 + \dots + k_n\Delta x_n \quad (\text{F.60})$$

$$\dot{x}_* = f(x_*, u_*, w_*, t) \quad (\text{F.61})$$

$$y_* = h(x_*, v_*, t) \quad (\text{F.62})$$

Using this strategy, which (necessarily) also applies for differential equations, a nonlinear dynamic

model can be linearized as indicated in Eqs. F.63 - F.66.

$$\dot{x} = f(x, u, w, t) \quad (\text{F.63})$$

$$y = h(x, v, t) \quad (\text{F.64})$$

$$\Delta \dot{x} = \underbrace{\frac{\partial f}{\partial x} \Big|_*}_{A} \Delta x + \underbrace{\frac{\partial f}{\partial u} \Big|_*}_{B} \Delta u + \underbrace{\frac{\partial f}{\partial w} \Big|_*}_{D} \Delta w$$

$$\Rightarrow \Delta \dot{x} = A\Delta x + B\Delta u + D\Delta w \quad (\text{F.65})$$

$$\Delta y = \underbrace{\frac{\partial h}{\partial x} \Big|_*}_{C} \Delta x + \underbrace{\frac{\partial h}{\partial v} \Big|_*}_{E} \Delta v$$

$$\Rightarrow \Delta y = C\Delta x + E\Delta v \quad (\text{F.66})$$

At this point, a so-called Linearized Kalman filter (LKF) estimator can be formulated for the linearized version of the nonlinear model in agreement with the results from the discussion on linear continuous estimators (Sec. F.2). In that case, the results would be as indicated in Eqs. F.67 - F.69, in which $\tilde{Q} = DQD^T$ and $\tilde{R} = ERE^T$ represents the covariance of the process noise and measurement noise, respectively.

$$\Delta \dot{\hat{x}} = A\Delta \hat{x} + K(\Delta y - C\Delta \hat{x}) \quad (\text{F.67})$$

$$K = PC^T \tilde{R}^{-1} \quad (\text{F.68})$$

$$\dot{P} = AP + PA^T + \tilde{Q} - PC^T \tilde{R}^{-1} CP \quad (\text{F.69})$$

Another approach for nonlinear state estimation, which is slightly more sophisticated, is the so-called Extended Kalman filter (EKF). This is believed to be one of the most abundant approaches utilized in the world when it comes to nonlinear estimation implementations [22]. In the EKF, one of the most apparent problems associated with the Kalman filter for the linearized process is addressed directly, namely the fact that the optimal trajectory (from which the deviations are calculated) may not be straight-forward to decide. In this sense, the definition of the nominal trajectory (Eqs. F.61 & F.62) is combined with the result from the linearized state estimation procedure (Eq. F.67) to yield the expression in Eq. F.70.

$$\dot{x}_* + \Delta \dot{\hat{x}} = f(x_*, u_*, w_*, t) + A\Delta \hat{x} + K(y - y_* - C(\hat{x} - x_*)) \quad (\text{F.70})$$

The next step in the strategy, which is the key maneuver of the EKF, is to set x_* equal to \hat{x} , i.e. to use the state estimates as the nominal trajectory. In this operation, it is kept in mind that $\Delta \hat{x} = \hat{x} - x_*$. This will change the differential equation for the state estimates, while the expressions for the estimator gain as well as covariances are the same. Finally, the system of equations for the EKF is presented in Eqs. F.71 - F.73.

$$\dot{\hat{x}} = f(\hat{x}, u, w_*, t) + K(y - h(\hat{x}, v_*, t)) \quad (\text{F.71})$$

$$K = PC^T \tilde{R}^{-1} \quad (\text{F.72})$$

$$\dot{P} = AP + PA^T + \tilde{Q} - PC^T \tilde{R}^{-1} CP \quad (\text{F.73})$$

In most real-life engineering cases, continuous models are used for the dynamic systems, while the measurements are discrete, i.e. sampled at specific points in time, as indicated in Eqs. F.74 & F.75.

Temperatures can, for instance, be measured every ten seconds depending on the equipment, while measurements related to fluid composition may have a slightly longer time interval between each measurement. The point is that the measurements are not necessarily continuously updated from the process.

$$\dot{x} = f(x, u, w, t) \quad (\text{F.74})$$

$$y_k = h_k(x_k, v_k) \quad (\text{F.75})$$

In such cases, a so-called Hybrid EKF (HEKF) can be deployed. Here, the purpose is to use the nonlinear continuous dynamics to propagate the state estimates in time without the use of measurements between the points of measurement, yet include a discrete step from *a priori* to *a posteriori* at each time t_k where a measurement is available. The set of equations for this case is presented in Eqs. F.76 - F.80, in agreement with previous results. Here, the process noise (Q) is a continuous quantity, while the measurement noise (R_k) is discrete at each time instant of valid measurement.

$$\text{Between } t_k^+ \text{ and } t_{k+1}^-: \quad \dot{\hat{x}} = f(\hat{x}, u, w_*, t) \quad (\text{F.76})$$

$$\dot{P} = AP + PA^T + \tilde{Q} \quad (\text{F.77})$$

$$\text{Between } t_k^- \text{ and } t_k^+: \quad K_k = P_k^- H_k^T (H_k P_k^- H_k^T + \tilde{R}_k)^{-1} \quad (\text{F.78})$$

$$\hat{x}_k^+ = \hat{x}_k^- + K_k (y_k - h_k(\hat{x}_k^-, v_*, t_k)) \quad (\text{F.79})$$

$$P_k^+ = (I - K_k H_k) P_k^- (I - K_k H_k)^T + K_k \tilde{R}_k K_k^T \quad (\text{F.80})$$

In addition to what these sections have been considering, there are more sophisticated approaches available to be deployed in the context of on-line estimation. The Iterated Extended Kalman filter (IEKF) is one example, in which the Taylor series expansion, i.e. the linearization, of the model functions are re-iterated at each point using the *a posteriori* state estimates. There also exist higher order methods¹⁴ for implementing the EKF, none of which are considered in this work. The Unscented Kalman Filter (UKF) is another example of a more complicated estimator, different from the EKF in that it reduces the error originating from the linearization in the EKF. Possible extensions to the EKF, like the UKF, are scarce in detail in this text on behalf of the choice of estimator in the thesis.

When considering which estimator to implement, it must be recognized that while some filters are more accurate than others, they demand more computational power. This can prove to be a major concern for on-line applications, and the performance of the estimator must be evaluated in close comparison with the actual system for which the estimator will be implemented. Generally speaking, the performance of the estimator with respect to computational time is closely correlated with the performance of the process model itself (i.e. the complexity of the process model).

This concludes the discussion on Kalman filter estimators for dynamic systems, for the purposes of this thesis work. The estimator is a key feature of the complete model-based predictive controller, which constitutes one of the main topics of interest in this thesis.

¹⁴The strategy for the higher order methods is to use higher order Taylor series expansion polynomials than the first order method which was deployed for the LKF and the EKF. This will act to reduce the error originating from the linearization, but it will lead to more complex calculations and hence a larger demand for computational power. [22]

Appendix G

A simplified example for software demonstration

The purpose of this additional section is to introduce a super-simplified first principles state space model to represent a buffer tank, for which the software tools utilized in the thesis will be demonstrated. These tools include the modeling language Modelica (and the Dymola tool) as well as the Cybernetica software for model-based predictive control. An illustration of the simple buffer tank is provided in Fig. G.1. This is a tank with two liquid inward flows and one liquid outward flow. Flows 2 and 3 are controllable (MVs) while flow 1 is regarded as a disturbance (DV). The fluid in mind is chosen to be pure water, for simplicity.

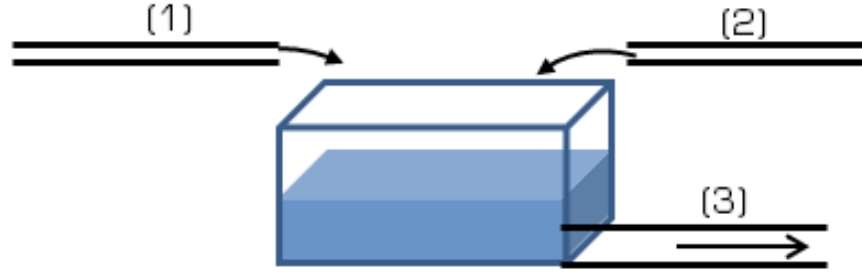


Figure G.1: A sketch of a simplified buffer tank.

The material balance for the tank is given in Eq. G.1 while the energy balance in temperature explicit form is given in Eq. G.2. Here, h denotes the height of the liquid level in the tank while A denotes the constant base area of the tank. The mass flows are denoted by F_i for pipe i . For the fluid properties, the heat capacity and density are denoted by C_p and ρ , respectively. These properties are, for simplicity, assumed to be constant over the specific temperature range.

$$\begin{aligned} \frac{d}{dt}(\rho Ah) &= F_1 + F_2 - F_3 \\ \implies \frac{dh}{dt} &= \frac{1}{\rho A} (F_1 + F_2 - F_3) \end{aligned} \quad (\text{G.1})$$

$$\begin{aligned} \frac{d}{dt}(\rho C_p Ah(T - T_{ref})) &= C_p (F_1(T_1 - T_{ref}) + F_2(T_2 - T_{ref}) - F_3(T - T_{ref})) \\ \implies \frac{d}{dt}(h(T - T_{ref})) &= \frac{1}{\rho A} (F_1(T_1 - T_{ref}) + F_2(T_2 - T_{ref}) - F_3(T - T_{ref})) \end{aligned} \quad (\text{G.2})$$

The temperature of the water in the tank and the height of the liquid level are, in other words, the two states of the system. In Modelica, the implementation is like indicated in Cs. G.1.

In most cases, a developed model should be subject to off-line parameter estimation prior to the on-line implementation of the model for the purpose of optimization and control. The reason

```

1 within ;
2 model bufferTank "Model of a buffer tank"
3
4     import blox = Modelica.Blocks.Interfaces;
5
6     Real h(unit="m", max=10,min=0,start=3) "Fluid level in tank";
7     Real T(unit="K",max=500,min=200,start=300) "Temperature in tank";
8
9     parameter Real A(unit="m2") = 0.5;
10    parameter Real T0(unit="K") = 298.15;
11    parameter Real T1(unit="K") = 450;
12    parameter Real T2(unit="K") = 200;
13    blox.RealInput par1_rho(unit="kg/m3",min=800,max=2000,nominal=1000)
14    "Fluid density";
15
16    blox.RealInput u1_F1(unit="kg/s",min=0,max=10,nominal=1)
17    "Disturbing feed stream";
18    blox.RealInput u2_F2(unit="kg/s",min=0,max=10,nominal=1)
19    "Controllable cooling fluid stream";
20    blox.RealInput u3_F3(unit="kg/s",min=0,max=10,nominal=2)
21    "Controllable outflow";
22
23    blox.RealOutput y1_h = h "Tank level";
24    blox.RealOutput y2_T = T "Tank temperature";
25
26    blox.RealOutput z1_h = h "Tank level";
27    blox.RealOutput z2_T = T "Tank temperature";
28
29    equation
30    der(h) = (u1_F1 + u2_F2 - u3_F3)/par1_rho/A;
31    par1_rho*A*der(h*(T-T0)) = u1_F1*(T1 - T0) + u2_F2*(T2-T0) - u3_F3*(T-T0);
32
33    annotation (uses(Modelica(version="3.2")));
34 end bufferTank;

```

Code snippet G.1: Modelica code for a simple buffer tank model.

for this is to ensure that the physical parameters of the model agree with the measured behavior of the model, and to modify the parameters when this is not the case. The Cybernetica Modelfit software is excellent for this task, as described in Sec. 4.2. For this simplified system however, which only contains water, the physical parameters are well known, and this step is skipped.

The simplified model is tested in on-line use with the Cybernetica CENIT software, as introduced in Sec. 4.3. The software allows for readily changing the setpoints of the CVs, the weighing of the setpoint violations, etc. An important part of the graphical user interface (GUI) of the CENIT man-machine interface (MMI) is shown in Fig. G.2, in which the configuration of CVs, MVs and DVs are shown. The user-friendliness of the software is emphasized, and the software presents with a lot of flexibility.

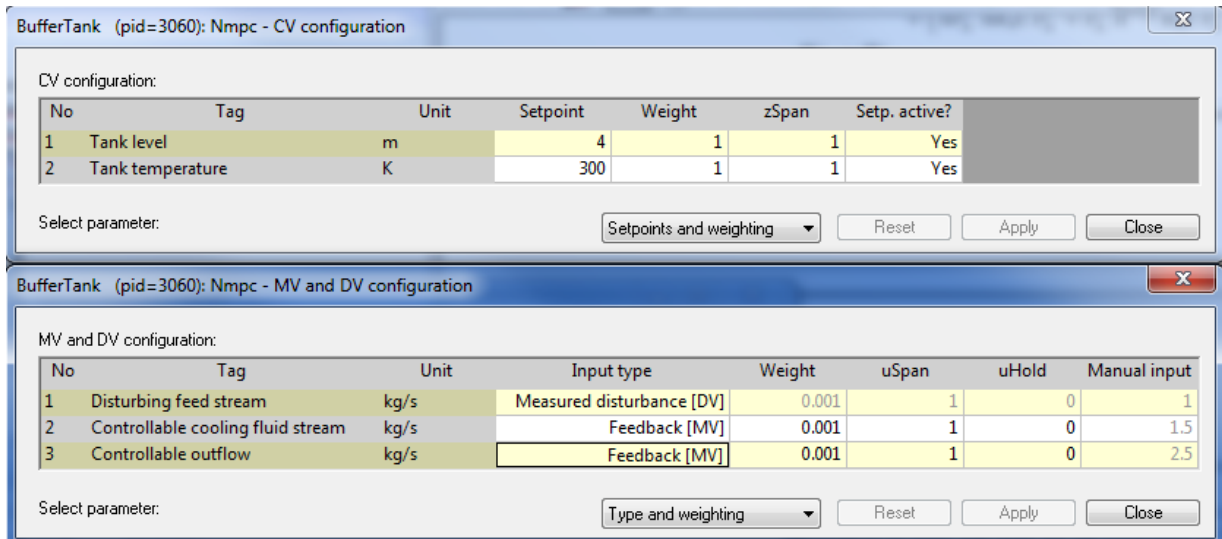
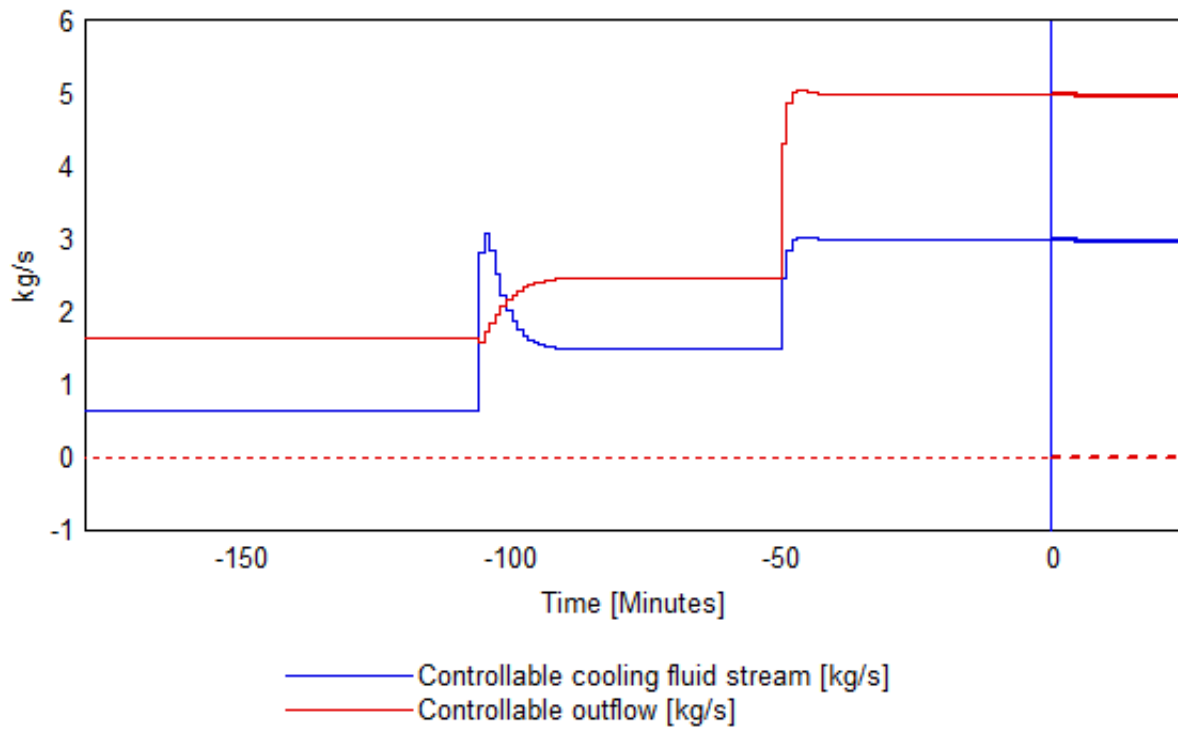


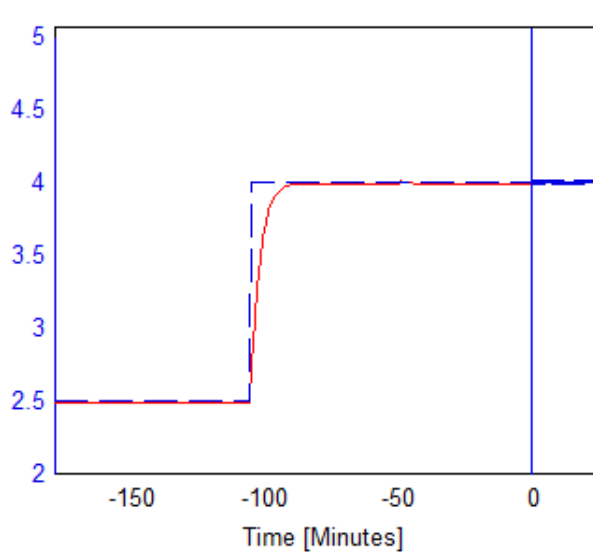
Figure G.2: The GUI of the CENIT MMI, specifically for modifying the system variables.

For the purpose of this short demonstration, a test run in which both setpoint changes and changes in disturbances are considered is simulated. The setpoint for the tank level (h_s) is changed from 2.5 m to 4 m , while the setpoint for the temperature is changed from 350 K to 300 K . A while later, the disturbance, i.e. the mass flow of stream 1, is changed from 1 kg/s to 2 kg/s . The resulting behavior of the controller is shown in Fig. G.3. In Fig. G.3a, the controller actions with respect to inputs (MVs) are shown, while Figs. G.3b & G.3c show the corresponding changes in the respective outputs (CVs).

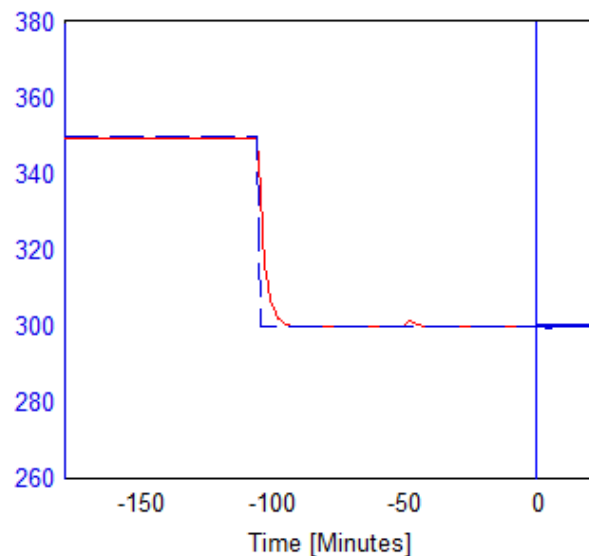
Although this model is simple, a concluding remark is to emphasize the effect of the controller. Changes in setpoints are effectively and quickly obeyed, and changes in the disturbance are virtually ineffective in the sense that the controller counteracts the disturbances very effectively. This concludes the simplified example on using the Cybernetica software for MPC.



(a) Changes in input. This is the controller action.



(b) Liquid level height



(c) Tank temperature

Figure G.3: Changes in outputs (CVs) for a simulated setpoint change.