NTNU – Trondheim
Norwegian University of
Science and Technology

Model predictive control of a polyolefin reactor

Kasper Linnestad

Chemical Engineering and Biotechnology
Submission date: June 2015
Supervisor: Magne Hillestad, IKP
Co-supervisor: Svein Olav Hauger, Cybernetica AS

Norwegian University of Science and Technology
Department of Chemical Engineering

Abstract

Two models for the copolymerization of propylene and ethylene in a fluidized bed reactor (FBR) were developed. A simple model for control; and a more advanced model which served as both a replacement for logged industrial plant data, and as a structurally different simulator to test the robustness of the controller. To ensure the best possible consistency between the two models, a set of parameters in the control model (CM) was estimated by fitting it to the plant replacement model (PRM). Additionally, an unscented Kalman filter (UKF) was set up to further mitigate the discrepancies between the models.

A nonlinear model predictive control (NMPC) strategy was applied to transition between two different polypropylene (PP) grades. Hard constraints were imposed on the manipulated variables (MVS), while soft constraints were applied on the controlled variables (CVS).

The effect of utilizing inert feed as an MV was studied with the CM as the simulator, i.e., no mismatch between the model of the controller and the process simulator. By employing the inert feed, the control of pressure improved; however, more catalyst was required to keep the production at the desired level.

In order to demonstrate the effectiveness and robustness of NMPC, despite structural differences between the model and the plant, grade transitions were simulated with the PRM as the process simulator. The controller performed reasonably well, however, further tuning of both the controller and the UKF would result in more

effective grade transitions. Nevertheless, the controller proved to be robust and coped well, notwithstanding the model mismatch.

Sammendrag

To modeller for kopolymerisasjon av propylen og etylen i en fluidisert sjiktreaktor (FBR) ble utviklet. En forenklet modell for reguleringsformål (CM), og en mer avansert modell som ble brukt som erstatning for loggdata fra et industrielt anlegg (PRM). Den avanserte modellen er strukturelt forskjellig fra den enkle modellen, og dette ble benyttet for å vise robustheten til regulatoren. Parameterestimering ble brukt for å sikre best mulig overensstemmelse mellom de to modellene. I tillegg ble også et Kalmanfilter (UKF) satt opp for å begrense modellavvikene ytterligere.

En ulineær modellprediktiv reguleringsstrategi (NMPC) ble anvendt på overgangen mellom to typer polypropylen (PP). Harde beskrankninger ble pålagt de manipulerte variablene (MV), mens myke beskrankninger ble benyttet på de kontrollerte variablene (CV).

Effekten av å benytte inertføden som en MV ble studert med CM som prosessimulator, dvs., ingen uoverensstemmelse mellom regulatormodellen og simulatormodellen. Trykkreguleringen ble bedre av å benytte inertføden, men mengden katalysator måtte økes for å opprettholde produksjonen av PP.

For å demonstrere effektiviteten og robustheten av NMPC, selv med strukturelle forskjeller mellom modellen og industrianlegget, ble produktovergangen simulert med erstatningsmodellen (PRM) som prosessimulator. Regulatoren gjennomførte overgangene tilstrekkelig, til tross for modelluoverensstemmelsen. Fintuning av regulatoren og Kalmanfilteret er imidlertid nødvendig for å forbedre overgangene ytterligere. Likevel viste regulatoren seg å være robust og overkom modellavvikene godt.

Preface

This Master's thesis was written during the spring of 2015 in close collaboration with Cybernetica AS, who proposed the thesis. The thesis is the final part of the Chemical Engineering and Biotechnology study program at the Norwegian University of Science and Technology (NTNU), leading to the degree of M.Sc. in Chemical Engineering. During my final year at NTNU, I was a part of the Process Systems Engineering Group at the Department of Chemical Engineering at the Faculty of Natural Sciences and Technology.

This document has been typeset with LaTeX 2_ε and the Memoir Class. If you are reading the PDF-version of this document; all symbols, acronyms, URLS and cross-references have been made clickable for convenient navigation. It is worth noting that most modern PDF-readers have a back button that lets you navigate back to the previous view.

Matrices have been typeset as bold uppercase letters, while vectors have been typeset as bold lowercase letters. In addition, the following style conventions have been used; all variables signifying the flow of a quantity have been decorated with a caret; all transpositions, e.g., chemical reactions, are denoted with a tilde; lastly, all time-derivatives are indicated with an overdot.

Acknowledgements

I would first and foremost like to express my gratitude towards all the employees at Cybernetica, for giving me a warm welcome to their offices. Especially to my supervisor, Svein Olav Hauger, for helping me with all of Cybernetica's tools, in addition to his knowledge on estimation and model predictive control. At Cybernetica, I would also like to thank Peter Singstad for his constructive input during our meetings.

Furthermore, I would like to thank my supervisor at NTNU, Professor Magne Hillestad, for his dedication, encouragement and sound advice.

Lastly, I would like to thank all my friends who have shared study room with me the past year, for all the fun and fruitful discussions we have had.

Declaration of compliance

I declare that this is an independent work according to the exam regulations of the Norwegian University of Science and Technology (NTNU).

June 11, 2015
Trondheim, Norway

Kasper J. Linnestad

Contents

List of Figures

List of Tables

List of Code Snippets

List of Acronyms

Acronym	
CM	control model
CV	controlled variable
CW	cooling water
DD1	first-order divided difference
DD2	second-order divided difference
EKF	extended Kalman filter
FBR	fluidized bed reactor
FC	flow controller
GUI	graphical user interface

Acronym	
KF	Kalman filter
LC	level controller
MFI	melt flow index
MPC	model predictive control
MV	manipulated variable
NMPC	nonlinear model predictive control
OPC	open platform communcations
PC	pressure controller
PDI	polydispersity index
PFD	process flow diagram
PI	proportional integral
PID	proportional integral derivative
PP	polypropylene
PRM	plant replacement model
PSSA	pseudo-steady-state assumption
RHC	receding horizon control

Acronym

SQP	sequential quadratic programming
TC	temperature controller
UKF	unscented Kalman filter

List of latin symbols

Symbol	Description	Unit
A	area	m^2
A	parameter in the Redlich-Kwong equation of state	$\mathrm{Pa\,K}^{1/2}$
a	parameter in the Redlich-Kwong equation of state	$\mathrm{Pa\,m^6\,K^{0.5}\,kmol^{-2}}$
Ar	Archimedes number	–
B	parameter in the Redlich-Kwong equation of state	–
b	parameter in the Redlich-Kwong equation of state	$\mathrm{m^3\,kmol^{-1}}$
c	concentration	$\mathrm{kmol\,m^{-3}}$

Symbol	Description	Unit
C_p	total heat capacity	$\mathrm{J\,K^{-1}}$
c_p	heat capacity	$\mathrm{J\,kmol^{-1}\,K^{-1}}$
D	diffusivity	$\mathrm{m^2\,s^{-1}}$
d	diameter	m
$\dot{E}$	accumulation of total energy	$\mathrm{J\,s^{-1}}$
$\hat{E}$	flow of total energy	$\mathrm{J\,s^{-1}}$
E	total energy	J
e	offset	
f	monomer fraction	–
g	gravitational acceleration	$\mathrm{m\,s^{-2}}$
H	enthalpy	J
$\dot{H}$	accumulation of enthalpy	$\mathrm{J\,s^{-1}}$
$\hat{H}$	flow of enthalpy	$\mathrm{J\,s^{-1}}$
h	partial molar enthalpy	$\mathrm{J\,kmol^{-1}}$
h	level of solids	m
$\Delta_{\mathrm{rx}}h$	enthalpy of reaction	$\mathrm{J\,kmol^{-1}}$
$h_{\mathrm{b\vert c}}$	heat transfer resistance between the bubble phase and the cloud	$\mathrm{J\,m^{-3}\,K^{-1}\,s^{-1}}$

Symbol	Description	Unit
$h_{\mathrm{b\vert e}}$	heat transfer resistance between the bubble and the emulsion phase	$\mathrm{J\,m^{-3}\,K^{-1}\,s^{-1}}$
$h_{\mathrm{c\vert e}}$	heat transfer resistance between the the cloud and the emulsion phase	$\mathrm{J\,m^{-3}\,K^{-1}\,s^{-1}}$
$\mathbf{I}$	identity matrix	
J	cost function	–
$\mathbf{K}$	Kalman filter gain	
k	conductivity	$\mathrm{J\,m^{-1}\,s^{-1}\,K^{-1}}$
k	pressure driven flow constant	$\mathrm{m^{3}\,s^{-1}\,Pa^{-1}}$
k	sample number	–
$k_{\mathrm{b\vert e}}$	mass transfer resistance between bubble and emulsion phase	$\mathrm{s^{-1}}$
K_{c}	controller gain	
k_{u}	antiwindup parameter	
$k_{\mathrm{b\vert c}}$	mass transfer resistance between bubble phase and cloud	$\mathrm{s^{-1}}$
$k_{\mathrm{c\vert e}}$	mass transfer resistance between cloud and emulsion phase	$\mathrm{s^{-1}}$
kds	deactivation rate constants	$\mathrm{s^{-1}}$
kf	formation rate constants	$\mathrm{s^{-1}}$

Symbol	Description	Unit
kfh	transfer by reaction with hydrogen rate constants	$\mathrm{m^3\,kmol^{-1}\,s^{-1}}$
kfm	transfer to monomer rate constants	$\mathrm{m^3\,kmol^{-1}\,s^{-1}}$
kfr	transfer to cocatalyst rate constants	$\mathrm{m^3\,kmol^{-1}\,s^{-1}}$
kfs	spontaneous transfer rate constants	$\mathrm{s^{-1}}$
kh	reinitiation with monomer rate constants	$\mathrm{m^3\,kmol^{-1}\,s^{-1}}$
khr	reinitiation with cocatalyst rate constants	$\mathrm{m^3\,kmol^{-1}\,s^{-1}}$
ki	initiation rate constants	$\mathrm{m^3\,kmol^{-1}\,s^{-1}}$
kp	propagation reaction rate constants	$\mathrm{m^3\,kmol^{-1}\,s^{-1}}$
$\dot{m}$	accumulation of mass	$\mathrm{kg\,s^{-1}}$
$\hat{m}$	flow of mass	$\mathrm{kg\,s^{-1}}$
m	mass	kg
$\bar{M}_\mathrm{n}$	number average molecular mass	$\mathrm{kg\,kmol^{-1}}$
$\bar{M}_\mathrm{w}$	mass average molecular mass	$\mathrm{kg\,kmol^{-1}}$
MFI	melt flow index	$\mathrm{dg\,min^{-1}}$
M	molecular mass	$\mathrm{kg\,kmol^{-1}}$
$\tilde{N}$	extent of reaction	$\mathrm{kmol\,s^{-1}}$
$\mathcal{N}$	normal distribution	
n	number of moles	kmol
$\dot{n}$	accumulation of the number of moles	$\mathrm{kmol\,s^{-1}}$
$\hat{n}$	molar flow rate	$\mathrm{kmol\,s^{-1}}$

Symbol	Description	Unit
$\tilde{n}$	net reaction rate	$\mathrm{kmol\,s^{-1}}$
np	number of prediction samples	–
ns	total number of different sites on the catalyst	–
nu	number of input moves in the control horizon	–
nz	number of coincidence points in the prediction horizon	–
p	pressure	Pa
PDI	polydispersity index	–
$\hat{Q}$	flow of heat	W
$\mathbf{Q}$	covariance matrix of the measurement noise	
$\mathbf{Q}$	block matrix containing the weights of the outputs	
q	weight of an output	
$\mathbf{q}$	vector containing the square of the weights of the outputs	
R	molar gas constant	$\mathrm{J\,kmol^{-1}\,K^{-1}}$
$\mathbf{R}_{\upsilon}$	covariance matrix of the parameter noise	

Symbol	Description	Unit
$\mathbf{R}_\nu$	covariance matrix of the process noise	
$\mathbf{R}$	block matrix containing the weights for soft output constraints	
$\mathbf{r}$	block vector containing the weights for the linear soft output constraints	
Re	Reynolds number	–
$\mathbf{S}$	block matrix containing the weights of the inputs	
T	temperature	K
t	time	s
ΔT_{lm}	logarithmic mean temperature difference	K
$\dot{U}$	accumulation of internal energy	$\mathrm{J\,s^{-1}}$
$\hat{U}$	flow of internal energy	$\mathrm{J\,s^{-1}}$
U	overall heat transfer coefficient	$\mathrm{W\,m^{-2}\,K^{-1}}$
U	internal energy	J
$\mathbf{U}$	diagonal matrix with the span values for each output	
$\Delta\mathbf{u}$	input moves	
$\mathbf{u}$	inputs	
u	input	

Symbol	Description	Unit
u	velocity	$\mathrm{m\,s^{-1}}$
$\hat{V}$	volumetric flow rate	$\mathrm{m^3\,s^{-1}}$
V	volume	$\mathrm{m^3}$
$\hat{W}$	flow of work	W
X	productivity	$\mathrm{kg_{pol}/kg_{cat}}$
x	mole fraction	–
$\mathbf{x}$	states	
$\dot{\mathbf{x}}$	time-derivative of the states	
$\mathbf{y}$	measurements	
y	measurement/output	
Z	compressibility	–
$\mathbf{Z}$	diagonal matrix with the span values for each output	
$\mathbf{z}$	additional outputs	
z	derived output	

List of Greek Symbols

Symbol	Description	Unit
δ	bubble phase fraction	–
ε	slack variable for soft output constraints	
ε	void fraction	–
$\boldsymbol{\varepsilon}$	slack variables for soft output constraints	
$\boldsymbol{\theta}$	parameters and constants	
$\dot{\lambda}$	accumulation of the chain length distribution for living polymer	$\mathrm{kmol\,s^{-1}}$
$\hat{\lambda}$	flow of the chain length distribution for living polymer	$\mathrm{kmol\,s^{-1}}$

Symbol	Description	Unit
λ	moment of the chain length distribution for living polymer	kmol
$\tilde{\lambda}$	net reaction of the chain length distribution for living polymer	$\mathrm{kmol\,s^{-1}}$
μ	moment of the chain length distribution for living and dead polymer	kmol
$\tilde{\mu}$	net reaction rate of moment of the chain length distribution for living and dead polymer	$\mathrm{kmol\,s^{-1}}$
μ	dynamic viscosity	Pa s
$\boldsymbol{\nu}$	process noise through the parameters	
ξ	moment of the chain length distribution for dead polymer	kmol
$\dot{\xi}$	accumulation of the chain length distribution for dead polymer	$\mathrm{kmol\,s^{-1}}$
$\hat{\xi}$	flow of the chain length distribution for dead polymer	$\mathrm{kmol\,s^{-1}}$
$\tilde{\xi}$	net reaction of the chain length distribution for dead polymer	$\mathrm{kmol\,s^{-1}}$

Symbol	Description	Unit
ρ	density	$\mathrm{kg\,m^{-3}}$
τ_{D}	derivative time	s
τ_{I}	integral time	s
$\boldsymbol{\upsilon}$	parameter noise	
ϕ	sphericity	–
φ	terminal monomer fraction	–
ψ	reaction rate correction	–
ω	mass fraction	–
$\boldsymbol{\omega}$	measurement noise	

List of sub- and superscripts

Symbol	Description
$-$	*a priori*
$+$	*a posteriori*
$\star$	dimensionless
0	uninitiated
1	propylene
2	ethylene
a|b	transfer from a to b
avg	average
b	bubble phase
bed	fluidized bed

Symbol	Description
br	bubble rise
c	critical
$C_{2=}$	ethylene
$C_{3=}$	propylene
cat	catalyst
cm	control model
cw	cooling water
e	emulsion phase
ext	external
f	feed
fb	freeboard
g	gas
H_2	hydrogen
hex	heat exchanger
I	integral
j	catalyst site type

Symbol	Description
k	monomer type
k	sample number
lm	logarithmic mean
M	monomer
m	measurement
m	monomer type
max	maximum
mf	minimum fluidization
min	minimum
n	number
N_2	nitrogen
P	potential active site
p	particle
pol	polymer
prm	plant replacement model
pu	purge
r	reactor

Symbol	Description
rx	reacion
s	shaft
s	solid
sf	superficial
sp	setpoint
T	total
T	transpose
t	terminal
tot	total
w	weight
ν	moment number

Chapter 1

Introduction

> I want to say one word to you. Just one word. [...] Plastics. [...] There's a great future in plastics. Think about it. Will you think about it?
>
> — Mr. McGuire, 1967[1]

Synthetic polymers are a central part of the world we inhabit, and are used in numerous applications, ranging from drinking bottles to cars to consumer electronics [13, 68]. This thesis considers a polyolefin, namely polypropylene (PP), which has applications in pipes and fittings; packaging; furniture and parts for the automotive industry, to name a few (for a more complete list of applications, see Tripathi [104]).

The PP industry is highly competitive, and producers have to meet the diversified demands of the customers. These requirements are met by

[1] Nichols, M., Director, *The Graduate*, Embassy Pictures, 1967

producing several grades of PP [107]. Hence, the manufacturers must be able to transition smoothly between these grades, which has prompted the need to apply advanced control methods on the reactors.

This Master's thesis has been written in collaboration with Cybernetica, a small company that specializes in tailor made model based control systems. Cybernetica has implemented nonlinear model predictive control (NMPC) on multiple polyolefin plants with focus on smooth grade transitions, safer operation, increased production and more consistent product quality. In this work, Cybernetica's tools for parameter estimation and NMPC have been utilized.

The main objective of this thesis has been to demonstrate the effectiveness and robustness of NMPC applied to a PP fluidized bed reactor (FBR), notwithstanding the mismatch between the model used for predictions and the plant itself, i.e., to exhibit the robustness of NMPC. As a consequence of the lack of log data from an industrial PP plant, two models were developed. The control model (CM) used by the NMPC, and the structurally different plant replacement model (PRM), which served as a substitute for any log data. By using two structurally different models for control and simulation, the robustness of the controller can be corroborated.

1.1 Production of polypropylene

There exists different processes for the production of PP, such as the Himont Spheripol process, the Union Carbide Unipol process, the BASF vertical stirred bed process and the Amoco-Chisso horizontal stirred bed process [21]. In this thesis, a simplified version of the Unipol process has been considered.

According to Shi *et al.* [101], the polypropylene industry circled $15 billion with an average annual growth of five percent in 2006, and this industry exhibit the strongest growth pattern of all the bulk polymer industries [104]. According to a report by PlasticsEurope [80], the global demand of plastics in 2012 was 288 Mt, which was a 2.8 % increase compared to 2011; and in Europe, PP was the most demanded plastic at 18.8 %.

Polypropylene has several advantages over competing materials, e.g., higher stiffness, better temperature resistance, good fatigue resistance and good chemical resistance, especially when price is of concern [104]. However, PP suffers from disadvantages such as low transparency, higher thermal expansion and lower impact strength [104]. In this thesis, ethylene has been utilized as a comonomer for the production of copolymer. The properties of polypropylene are dependent on the amount of comonomer content of the polymer, thus different customers will require different compositions.

The aforementioned Unipol process utilizes two FBRS in series with different operating conditions. A simplified process flow diagram (PFD)

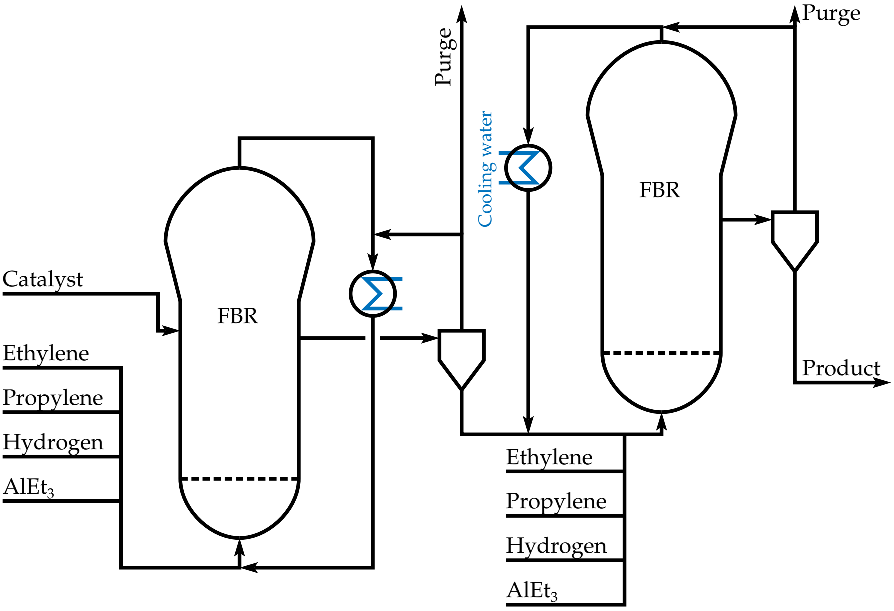


FIGURE 1.1 – A simplified process flow diagram (PFD) of the UNIPOL process. The process consists of two fluidized bed reactors (FBRS), with separate feeds and operating conditions. Adapted from Debling *et al.* [21].

of this multistage gas phase process is displayed in Figure 1.1. The two aforementioned models are, for simplicity, developed for the first reactor of the UNIPOL process only. The simplest model, the CM, lumps the whole process into one control volume, while the more complicated PRM utilizes four different control volumes. Hence, the structural difference between the two models will be significant. The CM and the PRM resemble the well-mixed and the two-phase models of McAuley, Talbot, *et al.* [67]. The development of the models is given in Chapter 2, where all the individual aspects of each model are elucidated in addition to a more complete process description.

1.2 Model predictive control

Model predictive control (MPC) is often referred to as a set of control methods based on prediction and optimization. The controller signals are obtained by minimization of an objective function, which usually contains the predicted future set point offsets and penalties for adjusting the inputs. This thesis has considered a subset of MPC, known as receding horizon control (RHC); these two terms will be used interchangeably hereafter. Receding horizon control is a control method that predicts the outputs and obtains the optimal sequence of future inputs based on these predictions, but only applies the calculated input at the current sample. This process is repeated at each sample, which yields a receding prediction horizon, from which the method has its name. The theory behind MPC is further clarified in Chapter 5.

A typical development process of the establishment of an MPC application is depicted in Figure 1.2. The procedure involves procurement of process knowledge and the establishment of a model of the plant. Logged process data from the plant are utilized to estimate unknown parameters in the model and to develop an estimator, e.g., a Kalman filter (KF). Once the model and the estimator has been properly validated against the data, simulations with the MPC are carried out to design the controller, i.e., tune the controller and select suitable constraints, until it passes a factory acceptance test. The final step is to implement the application on site with a subsequent site acceptance test.

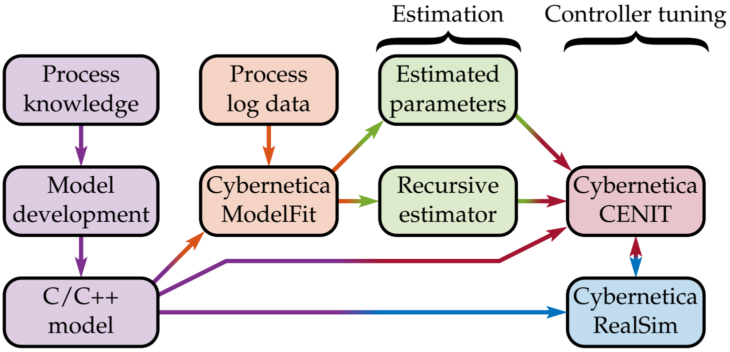


FIGURE 1.2 – Typical work flow of the development of a model predictive control (MPC) with a first-principles model. Process knowledge is gathered and a model is formulated based on physical and chemical principles. The model is then fitted against logged process data by estimating a set of unknown parameters. In addition, a recursive estimator, e.g., a Kalman filter, is set up. Finally, the controller is tuned in a simulation environment before it is implemented on a plant. ModelFit, CENIT and RealSim are Cybernetica's tools for parameter estimation, MPC and simulation, respectively.

This thesis follows a similar work flow as displayed in Figure 1.2; Chapter 2 presents the model development, while Chapters 3 and 4 explain the parameter estimation and the application of a recursive estimator, respectively. Finally, the design of the MPC is provided in Chapter 5. However, the thesis differs in the following ways; in a normal application, only one model would be developed; the controller would be tested in a simulation environment with model mismatch, but only parameter bias would be considered, not structural differences.

Chapter 2

Model development

> Modeling is a core activity in engineering and science: it provides insight, understanding and models are great sand-boxes – any game can be played.
>
> — H. A. Preisig, 2015[1]

Two models for the production of polypropylene (PP) in a fluidized bed reactor (FBR) have been developed, the control model (CM) and the plant replacement model (PRM). The reason for this duality is the lack of data from an industrial PP plant, but it also serves to demonstrate the robustness of the controller. The PRM has been developed as a substitute to the data and acts a process simulator, while the CM is used for control purposes.

[1] Preisig, H. A., "The ABC of process modelling, Lecture notes," Jan. 19, 2015, [Online]. Available: https://dl.dropboxusercontent.com/u/19261469/ABC_script.pdf (visited on May 13, 2015)

Modeling of FBRS is a much researched topic and the number of existing models is vast. Both computational fluid dynamics methodologies, e.g., discrete particle tracking, and simpler semi-empirical models have been utilized [47, 57]. In this thesis, the models have been developed for use in real-time applications which excludes the use of computational fluid dynamics due to the high computational load. Instead, the hydrodynamics of the FBR have been modeled similarly to the approach proposed by Kunii and Levenspiel [57] with the addition of the empirical correlations of Cui *et al.* [18].

The reactions involved in the production of PP have been modeled by the kinetic scheme offered by McAuley, MacGregor, and Hamielec [66]. This reaction scheme was originally developed for the production of polyethylene, but it has also been applied to polymerization of propylene. The modeling follows an approach similar to the one used by Shamiri, Wong, *et al.* [100], but the mole balances are formulated in extensive variables, e.g., the number of moles, instead of intensive variables, e.g., the concentrations. A method of moments has been applied to approximate the chain length distribution of the polymers. These modeling concepts will be elucidated in the succeeding sections.

2.1 PROCESS DESCRIPTION

The simplified process considered in this thesis consists of an FBR, a heat exchanger and a compressor as shown in the process flow diagram

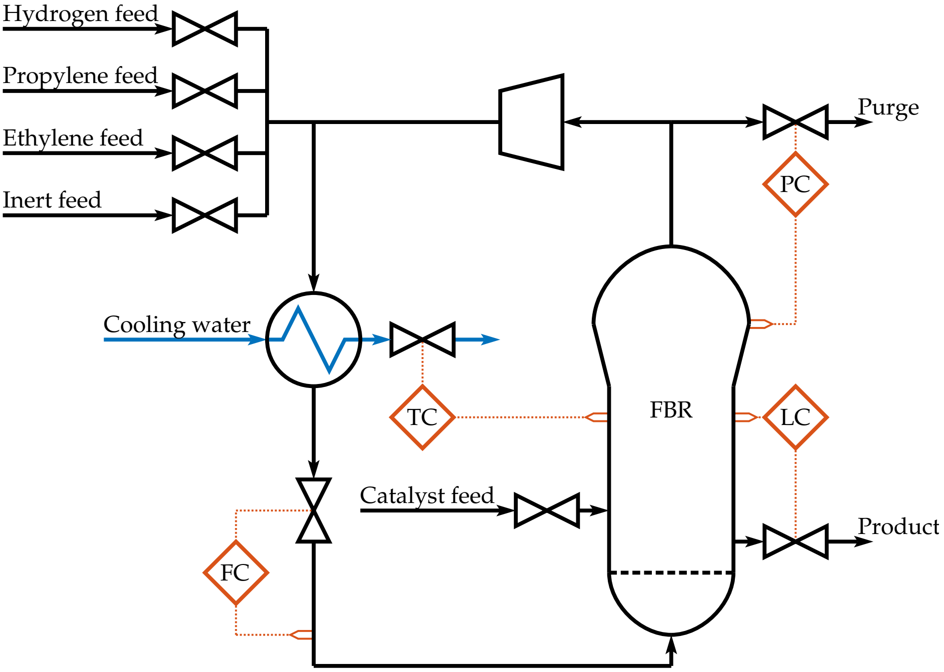


Figure 2.1 – The process flow diagram (PFD) of the first fluidized bed reactor (FBR) from the Unipol process given in Figure 1.1 in Chapter 1. Black lines denote mass flows in the system, blue lines are used for the cooling water (CW) while the orange dots indicate control signals.

(PFD) in Figure 2.1. The catalyst particles are fed directly into the reactor while the make-up gas is mixed with the recycle before it enters the heat exchanger. The reactor feed from the heat exchanger is controlled by a flow controller (FC) to give a constant gas velocity. The reactor pressure is controlled by a pressure controller (PC) which manipulates the purge rate from the reactor. The remainder of the gas from the reactor outlet is fed into a compressor before it is mixed with the fresh make-up streams. The production rate is given by a level controller (lc), while the temperature in the reactor is controlled by manipulating the flow rate of the cooling water (cw). This is a simplification of the temperature control present in an industrial reactor, where cascade control is often applied [85], e.g., by controlling the reactor temperature with the temperature of the cw (master), and controlling the temperature of the cw with the flow of cw (slave).

2.2 Model description

For simplicity, the effect of the compressor has been neglected in both the prm, and the cm. In addition, all fcs, e.g., the feed fcs and the recycle fc, have been assumed to be perfect controllers, i.e., event-dynamic.

2.2.1 Plant replacement model

The prm has four distinct control volumes, one for the heat exchanger and three for the reactor. These control volumes are assumed to be ideally mixed, i.e., no internal gradients. The three parts of the reactor are the bubble phase, the emulsion phase and the freeboard above the bed. The temperature controller (tc) is manipulating the temperature of the cw in order to keep the temperature in the reactor at the desired set point. Solids are presumed to be present in the bubble and emulsion phase volumes only; their distribution is assumed to occur instantly, and is calculated using the model of Cui *et al.* [18]. A physical topology of the model is presented in Figure 2.2. The model is further simplified by

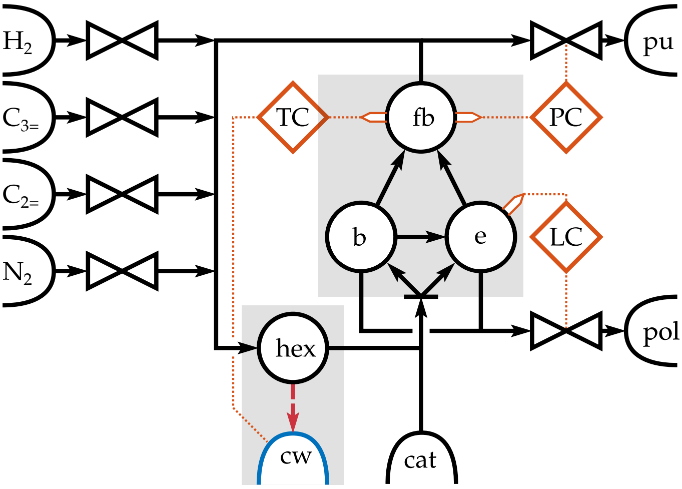


Figure 2.2 – A physical abstraction of the plant replacement model (PRM). The black lines, red dashes and orange dots denote mass flow, heat flow and control signals, respectively. $C_{3=}$ and $C_{2=}$ are abbreviations for propylene and ethylene, respectively. cat is the catalyst, pu is the purge outlet while pol is the product outlet. hex is the heat exchanger, while the reactor is split into three control volumes, freeboard (fb), emulsion phase (e) and bubble phase (b).

assuming that the TC can control the temperature of the CW instantly, i.e., no cascade control. The dynamics associated with the FC is assumed to be event-dynamic, hence a constant flow rate into the reactor is applied.

2.2.2 Control model

The CM considers the whole process as one ideally mixed volume; an abstraction of the CM is displayed in Figure 2.3. The controllers are identical to that of the PRM, but the FC for the recycle has not been considered, due to the fact that it resides within the control volume.

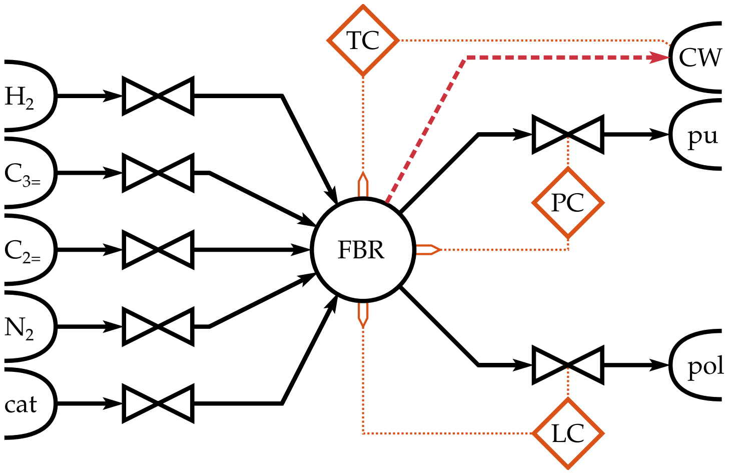


FIGURE 2.3 – A physical abstraction of the control model (CM). The black lines, red dashes and orange dots denote mass flow, heat flow and control signals, respectively. $C_{3=}$ and $C_{2=}$ are abbreviations for propylene and ethylene, respectively. cat is the catalyst, pu is the purge outlet while pol is the product outlet.

2.3 MODELING OF THE REACTION KINETICS

The model of the reaction kinetics used to simulate the polymerization process is taken from McAuley, MacGregor, and Hamielec [66], and is recited in this section. This model has been applied by several authors in applications related to polyolefin production [2, 43, 94–100]. The reactions included are shown in Table 2.1, and the corresponding rate constants are given in Table A.3 in Appendix A. Two important assumptions are made for the reaction rates; firstly that the concentration of the gaseous components adsorbed in the solid are in equilibrium; secondly, that all adsorption effects can be included in the reaction rate constants [66].

Table 2.1 – The reactions applied in the kinetic model [66, 100]. P is a potential site, $\mathrm{N_0}$ and $\mathrm{N_H}$ are uninitiated sites produced by formation and transfer, respectively. N and Q are used for living and dead polymer molecules. m and k denote the type of monomer, r is the length of the polymer chain and j is the catalyst site type.

DESCRIPTION	REACTION	RATE CONSTANT
Formation	$\mathrm{P}^j \longrightarrow \mathrm{N}^j_0$	kf^j
Initiation with monomers	$\mathrm{N}^j_0 + \mathrm{M}_m \longrightarrow \mathrm{N}^j_{m,1}$	ki^j_m
Propagation	$\mathrm{N}^j_{m,r} + \mathrm{M}_k \longrightarrow \mathrm{N}^j_{k,r+1}$	kp^j_{mk}
Transfer to hydrogen	$\mathrm{N}^j_{m,r} + \mathrm{H}_2 \longrightarrow \mathrm{N}^j_{\mathrm{H}} + \mathrm{Q}^j_r$	kfh^j_m
Transfer to cocatalyst	$\mathrm{N}^j_{m,r} \xrightarrow{\mathrm{AlEt_3}} \mathrm{N}^j_{m,1} + \mathrm{Q}^j_r$	kfr^j_m
Spontaneous transfer	$\mathrm{N}^j_{m,r} \longrightarrow \mathrm{N}^j_{\mathrm{H}} + \mathrm{Q}^j_r$	kfs^j
Reinitiation with monomer	$\mathrm{N}^j_{\mathrm{H}} + \mathrm{M}_m \longrightarrow \mathrm{N}^j_{m,1}$	kh^j_m
Reinitiation with cocatalyst	$\mathrm{N}^j_{\mathrm{H}} \xrightarrow{\mathrm{AlEt_3}} \mathrm{N}^j_{1,1}$	khr^j
Deactivation of living polymer	$\mathrm{N}^j_{m,r} \longrightarrow \mathrm{N}^j_{\mathrm{d}} + \mathrm{Q}^j_r$	kds^j
Deactivation of uninitiated site	$\mathrm{N}^j_0 \longrightarrow \mathrm{N}^j_{\mathrm{d}}$	kds^j
Deactivation of uninitiated site	$\mathrm{N}^j_{\mathrm{H}} \longrightarrow \mathrm{N}^j_{\mathrm{d}}$	kds^j

A mole balance for the number of potential catalyst sites is given in eq. (2.1)

$$\dot{n}^j_{\mathrm{P}} = \hat{n}^j_{\mathrm{P,f}} - \hat{n}^j_{\mathrm{P,pol}} - \tilde{n}^j_{\mathrm{P}}, \qquad \forall j \in \{1, \ldots, ns\} \tag{2.1}$$

in which the feed of potential active sites is proportional to the catalyst feed; the parameters for the catalyst used in the simulations are given in Table A.4 in Appendix A. The product flow can be calculated from:

$$\hat{n}^j_{\mathrm{P,pol}} = c^j_{\mathrm{P}} \hat{V}_{\mathrm{pol}} = \frac{n^j_{\mathrm{P}}}{V_{\mathrm{s}}} \hat{V}_{\mathrm{pol}} \tag{2.2}$$

The reaction term in eq. (2.1) is the formation of uninitiated active sites, which is calculated by:

$$\tilde{n}^j_{\mathrm{P}} = kf^j n^j_{\mathrm{P}} \tag{2.3}$$

Similar expressions can be obtained for uninitiated sites produced by formation and transfer, and are shown in eq. (2.4) in addition to the mole balance for polymer chains of unit length:

$$\dot{n}^j_{\mathrm{N_0}} = -\hat{n}^j_{\mathrm{N_0,pol}} + \tilde{n}^j_{\mathrm{N_0}} \tag{2.4a}$$

$$\dot{n}^j_{\mathrm{N_H}} = -\hat{n}^j_{\mathrm{N_H,pol}} + \tilde{n}^j_{\mathrm{N_H}} \tag{2.4b}$$

$$\dot{n}^j_{\mathrm{N}_{m,1}} = -\hat{n}^j_{\mathrm{N}_{m,1},\mathrm{pol}} + \tilde{n}^j_{\mathrm{N}_{m,1}}, \qquad \forall m \in \{1, 2\} \tag{2.4c}$$

The outflows via the product stream are obtained by eq. (2.2), while the reaction rates are recited below:

$$\tilde{n}^j_{\mathrm{N_0}} = \tilde{n}^j_{\mathrm{P}} - n^j_{\mathrm{N_0}} \left(c_{\mathrm{M_T}} ki^j_{\mathrm{T}} + kds^j \right) \tag{2.5a}$$

$$\tilde{n}^j_{\mathrm{N_H}} = \lambda^j_0 \left(c_{\mathrm{H_2}} kfh^j_{\mathrm{T}} + kfs^j_{\mathrm{T}} \right) - n^j_{\mathrm{N_H}} \left(c_{\mathrm{M_T}} kh^j_{\mathrm{T}} + kds^j + c_{\mathrm{AlEt_3}} khr^j \right) \tag{2.5b}$$

$$\begin{aligned} \tilde{n}^j_{\mathrm{N}_{m,1}} = {} & c_{\mathrm{M}_m} \left(ki^j_m n^j_{\mathrm{N_0}} + kh^j_m n^j_{\mathrm{N_H}} + kfm^j_{\mathrm{T}i} \lambda^j_0 \right) \\ & - n^j_{\mathrm{N}_{m,1}} \left(c_{\mathrm{M_T}} kp^j_{m\mathrm{T}} + c_{\mathrm{M_T}} kfm^j_{m\mathrm{T}} + c_{\mathrm{AlEt_3}} kfr^j_m \right. \\ & \left. + c_{\mathrm{H_2}} kfh^j_m + kfs^j_m + kds^j \right) \end{aligned} \tag{2.5c}$$

A pseudo-steady-state assumption (PSSA) is applied to eqs. (2.1) and (2.4) which yields the algebraic expressions in eq. (2.6).

$$n^j_{\mathrm{P}} = \frac{\hat{n}^j_{\mathrm{P,f}}}{\frac{\hat{V}_{\mathrm{pol}}}{V_{\mathrm{s}}} + kf^j} \tag{2.6a}$$

$$n^j_{\mathrm{N_0}} = \frac{\tilde{n}^j_{\mathrm{P}}}{\frac{\hat{V}_{\mathrm{pol}}}{V_{\mathrm{s}}} + c_{\mathrm{M_T}} ki^j_{\mathrm{T}} + kds^j} \tag{2.6b}$$

$$n^j_{\mathrm{N_H}} = \frac{\lambda^j_0 \left(c_{\mathrm{H_2}} kfh^j_{\mathrm{T}} + kfs^j_{\mathrm{T}} \right)}{\frac{\hat{V}_{\mathrm{pol}}}{V_{\mathrm{s}}} + c_{\mathrm{M_T}} kh^j_{\mathrm{T}} + kds^j + c_{\mathrm{AlEt_3}} khr^j} \tag{2.6c}$$

$$n^j_{\mathrm{N}_{m,1}} = \frac{c_{\mathrm{M}_m} \left(ki^j_m n^j_{\mathrm{N_0}} + kh^j_m n^j_{\mathrm{N_H}} + kfm^j_{\mathrm{T}i} \lambda^j_0 \right)}{\begin{aligned} \frac{\hat{V}_{\mathrm{pol}}}{V_{\mathrm{s}}} + c_{\mathrm{M_T}} kp^j_{m\mathrm{T}} + c_{\mathrm{M_T}} kfm^j_{m\mathrm{T}} + c_{\mathrm{AlEt_3}} kfr^j_m \\ + c_{\mathrm{H_2}} kfh^j_m + kfs^j_m + kds^j \end{aligned}} \tag{2.6d}$$

All the reaction rate constants that contain the subscript T are pseudo-kinetic rate constants, which are defined in eq. (2.8). The total monomer concentration, $c_{\mathrm{M_T}}$, is the sum of the individual monomer concentrations:

$$c_{\mathrm{M_T}} = \sum_{\forall m} c_{\mathrm{M}_m} \tag{2.7}$$

The pseudo-kinetic rate constants previously mentioned can be calculated as follows:

$$ki^j_{\mathrm{T}} = \sum_{\forall m} f_m ki^j_m \tag{2.8a}$$

$$kh^j_{\mathrm{T}} = \sum_{\forall m} f_m kh^j_m \tag{2.8b}$$

$$kfh^j_{\mathrm{T}} = \sum_{\forall m} \varphi^j_m kfh^j_m \tag{2.8c}$$

$$kfs^j_{\mathrm{T}} = \sum_{\forall m} \varphi^j_m kfs^j_m \tag{2.8d}$$

$$kfr^j_{\mathrm{T}} = \sum_{\forall m} \varphi^j_m kfr^j_m \tag{2.8e}$$

$$kfm^j_{k\mathrm{T}} = \sum_{\forall m} f_m kfm^j_{km} \tag{2.8f}$$

$$kfm^j_{\mathrm{T}k} = \sum_{\forall m} \varphi^j_m kfm^j_{mk} \tag{2.8g}$$

$$kfm^j_{\mathrm{TT}} = \sum_{\forall m} f_m kfm^j_{\mathrm{T}m} \tag{2.8h}$$

$$kp^j_{k\mathrm{T}} = \sum_{\forall m} f_m kp^j_{km} \tag{2.8i}$$

$$kp^j_{\mathrm{T}k} = \sum_{\forall m} \varphi^j_m kp^j_{mk} \tag{2.8j}$$

$$kp^j_{\mathrm{TT}} = \sum_{\forall m} f_m kp^j_{\mathrm{T}m} \tag{2.8k}$$

The monomer fraction f_m and the fraction of terminal monomer φ^j_m are given in eq. (2.9), while the expression for φ^j_m is obtained by making the long chain approximation for growing polymer chains [14, 66].

$$f_m = \frac{c_{\mathrm{M}_m}}{c_{\mathrm{M}_\mathrm{T}}} \tag{2.9a}$$

$$\varphi^j_m = \frac{f_m kp^j_{km}}{f_m kp^j_{km} + f_k kp^j_{mk}}, \qquad k \neq m \tag{2.9b}$$

Instead of having a mass balance of each chain length of living and dead polymer, balances for the moments of the chain length distributions are generated [37, 66, 97]. The balances for the zeroth, first and second moment for the chain length distribution of living polymer are given in eq. (2.10)

$$\dot{\lambda}^j_\nu = -\hat{\lambda}^j_{\nu,\mathrm{pol}} + \tilde{\lambda}^j_\nu, \qquad \forall \nu \in \{0, 1, 2\} \tag{2.10a}$$

$$\lambda^j_\nu = \sum_{r=1}^{\infty} r^\nu \sum_{m=1}^{2} n^j_{\mathrm{N}_{m,r}} \tag{2.10b}$$

while the corresponding moments for dead polymer are given in eq. (2.11).

$$\dot{\xi}^j_\nu = -\hat{\xi}^j_{\nu,\mathrm{pol}} + \tilde{\xi}^j_\nu, \qquad \forall \nu \in \{0, 1, 2\} \tag{2.11a}$$

$$\xi^j_\nu = \sum_{r=2}^{\infty} r^\nu \sum_{m=1}^{2} n^j_{\mathrm{Q}_{m,r}} \tag{2.11b}$$

λ^j_ν and ξ^j_ν denote the νth moment of the chain length distribution of living and dead respectively; j indicate the catalyst site. The net reaction rates for the moments of the chain length distributions are calculated by the expressions below [37, 66, 100]:

$$\begin{aligned}
\tilde{\lambda}^j_0 = {} & c_{\mathrm{M_T}} \left(ki^j_{\mathrm{T}} n^j_{\mathrm{N_0}} + kh^j_{\mathrm{T}} n^j_{\mathrm{N}_H} \right) + khr^j c_{\mathrm{AlEt_3}} n^j_{\mathrm{N}_H} \\
& - \lambda^j_0 \left(kfh^j_{\mathrm{T}} c_{\mathrm{H_2}} + kfs^j_{\mathrm{T}} + kds^j \right) && (2.12a) \\
\tilde{\lambda}^j_1 = {} & c_{\mathrm{M_T}} \left(ki^j_{\mathrm{T}} n^j_{\mathrm{N_0}} + kh^j_{\mathrm{T}} n^j_{\mathrm{N}_H} + kp^j_{\mathrm{TT}} \lambda^j_0 + kfm^j_{\mathrm{TT}} \lambda^j_0 \right) && (2.12b) \\
& + c_{\mathrm{AlEt_3}} \left(khr^j c_{\mathrm{H_2}} + kfr^j_{\mathrm{T}} \lambda^j_0 \right) \\
& - \lambda^j_1 \left(kfm^j_{\mathrm{TT}} c_{\mathrm{M_T}} + kfr^j_{\mathrm{T}} c_{\mathrm{AlEt_3}} + kfh^j_{\mathrm{T}} c_{\mathrm{H_2}} kfs^j_{\mathrm{T}} + kds^j \right) \\
\tilde{\lambda}^j_2 = {} & c_{\mathrm{M_T}} \left(ki^j_{\mathrm{T}} n^j_{\mathrm{N_0}} + kh^j_{\mathrm{T}} n^j_{\mathrm{N}_H} + kp^j_{\mathrm{TT}} \left(2\lambda^j_1 - \lambda^j_0 \right) + kfm^j_{\mathrm{TT}} \lambda^j_0 \right) && (2.12c) \\
& + c_{\mathrm{AlEt_3}} \left(khr^j c_{\mathrm{H_2}} + kfr^j_{\mathrm{T}} \lambda^j_0 \right) \\
& - \lambda^j_2 \left(kfm^j_{\mathrm{TT}} c_{\mathrm{M_T}} + kfr^j_{\mathrm{T}} c_{\mathrm{AlEt_3}} + kfh^j_{\mathrm{T}} c_{\mathrm{H_2}} kfs^j_{\mathrm{T}} + kds^j \right) \\
\tilde{\xi}^j_\nu = {} & \left(\lambda^j_\nu - \sum_{m=1}^{2} n^j_{\mathrm{N}_{m,1}} \right) && (2.12d) \\
& \cdot \left(kfm^j_{\mathrm{TT}} c_{\mathrm{M_T}} + kfr^j_{\mathrm{T}} c_{\mathrm{AlEt_3}} + kfh^j_{\mathrm{T}} c_{\mathrm{H_2}} + kfs^j_{\mathrm{T}} + kds^j \right)
\end{aligned}$$

It is worth noting that dead polymer of unit chain length is not considered a part of the polymer, hence the summation in eq. (2.11) starts at $r = 2$. The outflows of the moments of the chain length distributions are calculated by eq. (2.2). By inspecting eq. (2.12) it is evident that ξ^j_ν at each site can be lumped together as shown in eq. (2.13).

$$\xi_\nu = \sum_{j=1}^{ns} \xi_\nu^j \tag{2.13}$$

In addition, the second moments of the chain length distributions for dead and living polymer can be described by defining a bulk moment that includes both living and dead polymer chains. This bulk balance is defined in eq. (2.14).

$$\mu_2 = \sum_{j=1}^{ns} \lambda_2^j + \xi_2 \tag{2.14}$$

The reaction rates for these lump and bulk moments are calculated as:

$$\tilde{\xi}_\nu = \sum_{j=1}^{ns} \tilde{\xi}_\nu^j \tag{2.15a}$$

$$\begin{aligned}
\tilde{\mu}_2 &= \sum_{j=1}^{ns} \left(\tilde{\lambda}_2^j + \tilde{\xi}_2^j\right) \\
&= \sum_{j=1}^{ns} \Bigg[c_{\mathrm{M_T}} \left(ki_{\mathrm{T}}^j n_{\mathrm{N_0}}^j + kh_{\mathrm{T}}^j n_{\mathrm{N}_H}^j + kp_{\mathrm{TT}}^j \left(2\lambda_1^j - \lambda_0^j\right) \right. \\
&\qquad \left. + kfm_{\mathrm{TT}}^j \lambda_0^j \right) + c_{\mathrm{AlEt_3}} \left(khr^j c_{\mathrm{H_2}} + kfr_{\mathrm{T}}^j \lambda_0^j \right) \\
&\qquad - \sum_{m=1}^{2} n_{\mathrm{N}_{m,1}}^j \left(kfm_{\mathrm{TT}}^j c_{\mathrm{M_T}} + kfr_{\mathrm{T}}^j c_{\mathrm{AlEt_3}} \right. \\
&\qquad\qquad \left. + kfh_{\mathrm{T}}^j c_{\mathrm{H_2}} + kfs_{\mathrm{T}}^j + kds^j \right) \Bigg]
\end{aligned} \tag{2.15b}$$

In order to predict the composition of the polymer, additional balances of the bound monomers are needed [66]. These mole balances are recited in eq. (2.16).

$$\dot{n}_{\mathrm{B}_m} = -\hat{n}_{\mathrm{B}_m,\mathrm{pol}} + \tilde{n}_{\mathrm{B}_m}, \quad \forall m \in \{1, 2\} \tag{2.16a}$$

$$\tilde{n}_{\mathrm{B}_m} = -\tilde{n}_{\mathrm{M}_m} \tag{2.16b}$$

2.4 Mole balances of the gas components

The mole balances for each of the gaseous components, i.e., hydrogen, propylene, ethylene and nitrogen, are required for each of the control volumes. These balances will be presented for each of the two models in the subsequent subsections.

2.4.1 Control model

The mole balances are given in eq. (2.17), in which it has been assumed that the concentration of the components adsorbed in the solid is proportional to the concentration in the gas phase, and that the proportionality constant can be included in the rate constants in Table 2.1.

$$\dot{n}_{\mathrm{H_2}} = \hat{n}_{\mathrm{H_2,f}} - \hat{n}_{\mathrm{H_2,pu}} - \hat{n}_{\mathrm{H_2,pol}} + \tilde{n}_{\mathrm{H_2}} \tag{2.17a}$$

$$\dot{n}_{\mathrm{C_{3=}}} = \hat{n}_{\mathrm{C_{3=},f}} - \hat{n}_{\mathrm{C_{3=},pu}} - \hat{n}_{\mathrm{C_{3=},pol}} + \tilde{n}_{\mathrm{C_{3=}}} \tag{2.17b}$$

$$\dot{n}_{\mathrm{C_{2=}}} = \hat{n}_{\mathrm{C_{2=},f}} - \hat{n}_{\mathrm{C_{2=},pu}} - \hat{n}_{\mathrm{C_{2=},pol}} + \tilde{n}_{\mathrm{C_{2=}}} \tag{2.17c}$$

$$\dot{n}_{\mathrm{N_2}} = \hat{n}_{\mathrm{N_2,f}} - \hat{n}_{\mathrm{N_2,pu}} - \hat{n}_{\mathrm{N_2,pol}} \tag{2.17d}$$

The feed rates are modeled by

$$\hat{n}_{i,\mathrm{f}} = c_{\mathrm{g}} \hat{V}_{i,\mathrm{f}}, \qquad \forall i \in \{\mathrm{H_2}, \mathrm{C_{3=}}, \mathrm{C_{2=}}, \mathrm{N_2}\} \tag{2.18}$$

where c_{g} is the concentration of gas in the system; the gas concentration is obtained from eq. (2.19).

$$c_{\mathrm{g}} = \sum_i c_i = \sum_i \frac{n_i}{V_{\mathrm{g}}}, \qquad \forall i \in \{\mathrm{H_2}, \mathrm{C_{3=}}, \mathrm{C_{2=}}, \mathrm{N_2}\} \tag{2.19}$$

The purge rate of each component is calculated as

$$\hat{n}_{i,\mathrm{pu}} = c_i \hat{V}_{\mathrm{pu}}, \qquad \forall i \in \{\mathrm{H}_2, \mathrm{C}_{3=}, \mathrm{C}_{2=}, \mathrm{N}_2\} \tag{2.20}$$

where the volumetric purge rate is given by a PC, which will be elucidated in Section 2.8. The product stream contains mainly solids, but also a small quantity of the gaseous compounds. The amount of gas that leaves the reactor via the product valve is given in eq. (2.21)

$$\hat{n}_{i,\mathrm{pol}} = \varepsilon_{\mathrm{pol}} c_i \hat{V}_{\mathrm{pol}}, \qquad \forall i \in \{\mathrm{H}_2, \mathrm{C}_{3=}, \mathrm{C}_{2=}, \mathrm{N}_2\} \tag{2.21}$$

in which the void fraction in the product stream is dependent on the type of valve. The reaction rates are given as

$$\tilde{n}_{\mathrm{H}_2} = -c_{\mathrm{H}_2} \sum_{j=1}^{ns} kfh_{\mathrm{T}}^{j} \lambda_0^{j} \tag{2.22a}$$

$$\tilde{n}_{\mathrm{C}_{3=}} = -c_{\mathrm{C}_{3=}} \sum_{j=1}^{ns} kp_{\mathrm{T1}}^{j} \lambda_0^{j} \tag{2.22b}$$

$$\tilde{n}_{\mathrm{C}_{2=}} = -c_{\mathrm{C}_{2=}} \sum_{j=1}^{ns} kp_{\mathrm{T2}}^{j} \lambda_0^{j} \tag{2.22c}$$

where it has been assumed that the consumption of monomers by other reactions than propagation is negligible. The pseudo-reaction rate constants proposed by de Carvalho *et al.* [14] are used here, their definitions are shown in eq. (2.8). λ_0^j is the zeroth moment of the chain length distribution of living polymer produced at site j. In addition, it has been assumed that the concentration of the reactants at the active sites is proportional to the concentration in the gas phase; the proportionality constant has been included in the reaction rate constants.

2.4.2 Plant replacement model

Mole balances for the gaseous components are required for all the control volumes. The reactions are assumed to only occur in the bubble and the emulsion phase, which is equivalent to neglecting the elutriation of solids into the freeboard region. The mole balances for each of the control volumes are then given as

$$\dot{n}_{i,\mathrm{hex}} = \hat{n}_{i,\mathrm{f|hex}} + \hat{n}_{i,\mathrm{fb|hex}} - \hat{n}_{i,\mathrm{hex|e}} - \hat{n}_{i,\mathrm{hex|b}} \tag{2.23a}$$

$$\dot{n}_{i,\mathrm{fb}} = \hat{n}_{i,\mathrm{b|fb}} + \hat{n}_{i,\mathrm{e|fb}} - \hat{n}_{i,\mathrm{fb|hex}} - \hat{n}_{i,\mathrm{fb|pu}} \tag{2.23b}$$

$$\dot{n}_{i,\mathrm{e}} = \hat{n}_{i,\mathrm{hex|e}} + \hat{n}_{i,\mathrm{b|e}} - \hat{n}_{i,\mathrm{e|fb}} - \hat{n}_{i,\mathrm{e|pol}} + \tilde{n}_{i,\mathrm{e}} \tag{2.23c}$$

$$\dot{n}_{i,\mathrm{b}} = \hat{n}_{i,\mathrm{hex|b}} - \hat{n}_{i,\mathrm{b|e}} - \hat{n}_{i,\mathrm{b|fb}} - \hat{n}_{i,\mathrm{b|pol}} + \tilde{n}_{i,\mathrm{b}} \tag{2.23d}$$

for all $i \in \{H_2, C_{3=}, C_{2=}, N_2\}$. The feed flows into the heat exchanger are calculated by

$$\hat{n}_{i,\mathrm{f|hex}} = c_{\mathrm{g,hex}} \hat{V}_{i,\mathrm{f}} \tag{2.24}$$

where the concentration in the heat exchanger is

$$c_{\mathrm{g,hex}} = \sum_i c_{i,\mathrm{hex}} = \sum_i \frac{n_{i,\mathrm{hex}}}{V_{\mathrm{hex}}} \tag{2.25}$$

and the volumetric flow from the heat exchanger into the reactor is given by the superficial velocity into the reactor:

$$\hat{V}_{\mathrm{hex|r}} = A u_{\mathrm{sf}} \tag{2.26}$$

The molar flow of each component into the reactor can then be obtained:

$$\hat{n}_{i,\mathrm{hex|r}} = c_{i,\mathrm{hex}} \hat{V}_{\mathrm{hex|r}} \tag{2.27}$$

The flow into the reactor is split into the bubble phase and the emulsion phase, where the split factor is the bubble phase fraction defined in eq. (2.40).

$$\hat{n}_{i,\mathrm{hex|b}} = \delta \hat{n}_{i,\mathrm{hex|r}} \tag{2.28a}$$

$$\hat{n}_{i,\mathrm{hex|e}} = (1-\delta)\,\hat{n}_{i,\mathrm{hex|r}} \tag{2.28b}$$

By assuming that the total number of moles in the heat exchanger is constant, i.e., a PSSA for the number of moles in the heat exchanger, the recycle from the freeboard region to the heat exchanger can be calculated as:

$$\hat{n}_{\mathrm{fb|hex}} = \sum_i \hat{n}_{i,\mathrm{fb|hex}} = \sum_i \left(\hat{n}_{i,\mathrm{hex|r}} - \hat{n}_{i,\mathrm{f|hex}}\right) \tag{2.29}$$

The molar recycle flow of each component are then calculated:

$$\hat{n}_{i,\mathrm{fb|hex}} = \frac{c_{i,\mathrm{fb}}}{\sum_i c_{i,\mathrm{fb}}} \hat{n}_{\mathrm{fb|hex}} \tag{2.30}$$

The flow from the bubble and emulsion phases into the freeboard region are assumed to be proportional to the respective pressure differences

$$\hat{V}_{s|\mathrm{fb}} = -k_{s|\mathrm{fb}}\left(p_{\mathrm{fb}} - p_s\right), \qquad \forall s \in \{\mathrm{e}, \mathrm{b}\} \tag{2.31a}$$

$$\hat{n}_{i,s|\mathrm{fb}} = -c_{i,s}\hat{V}_{s|\mathrm{fb}} \tag{2.31b}$$

where it has been assumed that there is no backflow. The molar purge rate of each component is calculated as

$$\hat{n}_{i,\mathrm{fb|pu}} = c_{i,\mathrm{fb}}\hat{V}_{\mathrm{pu}} \tag{2.32}$$

where the volumetric purge rate is manipulated by a PC which is described in Section 2.8. The transfer rate between the emulsion and the bubble phase is modeled as proportional to the concentration difference in the phases [100] and the volume of the bubble phase, hence

$$\hat{n}_{i,\mathrm{b|e}} = -k_{\mathrm{b|e}} V_{\mathrm{b}} \left(c_{i,\mathrm{e}} - c_{i,\mathrm{b}} \right) \tag{2.33}$$

where the mass transfer resistance between bubble and emulsion phase is given in eq. (2.46). The outflow of gas in the product stream for each component in the bubble and the emulsion phase are given in eqs. (2.34a) and (2.34b), respectively.

$$\hat{n}_{i,\mathrm{b|pol}} = \delta \varepsilon_{\mathrm{pol}} c_{i,\mathrm{b}} \hat{V}_{\mathrm{pol}} \tag{2.34a}$$

$$\hat{n}_{i,\mathrm{e|pol}} = (1 - \delta)\, \varepsilon_{\mathrm{pol}} c_{i,\mathrm{e}} \hat{V}_{\mathrm{pol}} \tag{2.34b}$$

The volumetric product flow is controlled by an LC, which is given in Section 2.8.

The net reaction rates for the bubble and the emulsion phases are calculated as:

$$\tilde{n}_{\mathrm{H}_2,s} = -c_{\mathrm{H}_2,s} \sum_{j=1}^{ns} kfh_{\mathrm{T}}^{j} \lambda_{0,s}^{j}, \qquad \forall s \in \{\mathrm{e}, \mathrm{b}\} \tag{2.35a}$$

$$\tilde{n}_{\mathrm{C}_{3=},s} = -c_{\mathrm{C}_{3=},s} \sum_{j=1}^{ns} kp_{\mathrm{T1}}^{j} \lambda_{0,s}^{j} \tag{2.35b}$$

$$\tilde{n}_{\mathrm{C}_{2=},s} = -c_{\mathrm{C}_{2=},s} \sum_{j=1}^{ns} kp_{\mathrm{T2}}^{j} \lambda_{0,s}^{j} \tag{2.35c}$$

2.5 Hydrodynamic modeling

To obtain the amount of catalyst and cocatalyst in the reactor a mass balance is required for each of them. However, if the ratio between catalyst and cocatalyst is constant at all times, a single balance is required to describe the mass of both the catalyst and the cocatalyst. Thus, by assuming a constant mass fraction of cocatalyst in the catalyst, the following equations are valid:

$$\dot{m}_{\text{cat,tot}} = \hat{m}_{\text{cat,tot,f}} - \hat{m}_{\text{cat,tot,pol}} \tag{2.36a}$$

$$\hat{m}_{\text{cat,tot,pol}} = \frac{m_{\text{cat,tot}}}{V_{\text{s}}} \hat{V}_{\text{pol}} \tag{2.36b}$$

$$m_{\text{cat}} = \left(1 - \omega_{\text{AlEt}_3}\right) m_{\text{cat,tot}} \tag{2.36c}$$

$$n_{\text{AlEt}_3} = \omega_{\text{AlEt}_3} \frac{m_{\text{cat,tot}}}{M_{\text{AlEt}_3}} \tag{2.36d}$$

2.5.1 Control model

The void fraction of the CM is assumed to be constant, hence the level of solids is given by [99]

$$h = \left(A_{\text{r}} \left(1 - \varepsilon\right)\right)^{-1} V_{\text{s}} \tag{2.37}$$

where the volume of solids is:

$$V_{\text{s}} = \frac{m_{\text{pol}}}{\rho_{\text{pol}}} + m_{\text{cat,tot}} \left(\frac{\omega_{\text{AlEt}_3}}{\rho_{\text{AlEt}_3}} + \frac{1 - \omega_{\text{AlEt}_3}}{\rho_{\text{cat}}} \right) \tag{2.38}$$

The total mass of polymer in the reactor is given by amount of bound monomers [66]:

$$m_{\text{pol}} = \sum_{m=1}^{2} n_{\text{B}_m} M_{\text{M}_m} \tag{2.39}$$

2.5.2 Plant replacement model

The distribution of solids in the PRM is assumed to occur instantaneously, thus, only a total balance for each solid component is needed. The most important parameters are the bubble phase fraction and the void fractions defined in **??**, respectively.

The bubble phase fraction is defined as the volume of the bubbles in the bed divided by the total volume of the bed:

$$\delta = \frac{V_\mathrm{b}}{V_\mathrm{bed}} \tag{2.40}$$

The void fractions are the volumetric fractions of gas in each of the phase:

$$\varepsilon_\mathrm{e} = \frac{V_\mathrm{g,e}}{V_\mathrm{e}} \tag{2.41a}$$

$$\varepsilon_\mathrm{b} = \frac{V_\mathrm{g,b}}{V_\mathrm{b}} \tag{2.41b}$$

Cui *et al.* [18] obtained the expressions in eq. (2.42) for the bubble phase fraction and the void fractions for a fluidized bed of Geldart B particles, which the polymer resins are [100]:

$$\delta = 0.534 \left[1 - \exp\left(-\frac{u_\mathrm{sf} - u_\mathrm{mf}}{0.413}\right)\right] \tag{2.42a}$$

$$\varepsilon_\mathrm{e} = \varepsilon_\mathrm{mf} + 0.2 - 0.059 \exp\left(-\frac{u_\mathrm{sf} - u_\mathrm{mf}}{0.429}\right) \tag{2.42b}$$

$$\varepsilon_\mathrm{b} = 1 - 0.146 \exp\left(-\frac{u_\mathrm{sf} - u_\mathrm{mf}}{4.439}\right) \tag{2.42c}$$

these expressions have also been applied to fluidized bed polymerization by several other authors [2, 25, 43, 51, 52, 94, 95, 97, 98, 100]. u_sf and u_mf are the superficial velocity and the velocity at minimum fluidization, respectively.

The level of solids is calculated with an average void fraction in the bed, which is obtained from eq. (2.43). The level can then be calculated by eq. (2.37), and the volume of solids is calculated as in the CM.

$$\varepsilon_{\mathrm{avg}} = \delta \varepsilon_{\mathrm{b}} + (1-\delta)\,\varepsilon_{\mathrm{e}} \tag{2.43}$$

Equations (2.40) and (2.41) can be used to obtain an expression for the distribution of the solids between each of the two phases:

$$\frac{m_{\mathrm{b}}}{m_{\mathrm{tot}}} = \frac{\delta\,(1-\varepsilon_{\mathrm{b}})}{\delta\,(1-\varepsilon_{\mathrm{b}}) + (1-\delta)\,(1-\varepsilon_{\mathrm{e}})} \tag{2.44a}$$

$$\frac{m_{\mathrm{e}}}{m_{\mathrm{tot}}} = 1 - \frac{m_{\mathrm{b}}}{m_{\mathrm{tot}}} \tag{2.44b}$$

In eq. (2.42), the minimum fluidization velocity is obtained from the minimum fluidization Reynolds number. The minimum fluidization Reynolds number is commonly calculated by [37, 61, 100]

$$\mathrm{Re}_{\mathrm{mf}} = \frac{u_{\mathrm{mf}}\rho_{\mathrm{g}} d_{\mathrm{p}}}{\mu_{\mathrm{g}}} \tag{2.45a}$$

$$= \left(29.5^2 + 0.0357\mathrm{Ar}\right)^{1/2} - 29.5 \tag{2.45b}$$

$$\mathrm{Ar} = \frac{\rho_{\mathrm{g}}\left(\rho_{\mathrm{s}} - \rho_{\mathrm{g}}\right) g d_{\mathrm{p}}^3}{\mu_{\mathrm{g}}^2} \tag{2.45c}$$

where Ar is the Archimedes number; ρ_{g} and ρ_{s} are the densities of the gas and solid phases, respectively; d_{p} is the diameter of the solid particles, which is assumed to be constant; μ_{g} is the viscosity of the gas phase and g is the gravitational acceleration.

The transfer of mass between the emulsion phase and the bubble phase is given in eq. (2.33), in which the mass transfer coefficient, $k_{\mathrm{b|e}}$ is given by eq. (2.46), recited below [57].

$$k_{\mathrm{b|e}} = \left(\frac{1}{k_{\mathrm{b|c}}} + \frac{1}{k_{\mathrm{c|e}}} \right)^{-1} \tag{2.46}$$

$k_{\mathrm{b|c}}$ is the mass transfer resistance from the bubble phase to the cloud, i.e., the interfacial region between the bubble phase and the emulsion phase, while $k_{\mathrm{c|e}}$ is the resistance between the cloud and the emulsion phase. These resistances are calculated by the correlations of Kunii and Levenspiel [57]

$$k_{\mathrm{b|c}} = 4.5 \frac{u_{\mathrm{e}}}{d_{\mathrm{b}}} + 5.85 \frac{D_{\mathrm{g}}^{1/2} g^{1/4}}{d_{\mathrm{b}}^{5/4}} \tag{2.47a}$$

$$k_{\mathrm{c|e}} = 6.77 \left(\frac{\varepsilon_{\mathrm{e}} u_{\mathrm{br}} D_{\mathrm{g}}}{d_{\mathrm{b}}^{3}} \right)^{1/2} \tag{2.47b}$$

where u_{e} is the velocity of the emulsion phase, d_{b} is the bubble diameter, D_{g} is the diffusivity of the gas and u_{br} is the rise velocity of the bubbles. The bubble rise velocity is predicted with the correlation of Werther [108]:

$$u_{\mathrm{br}} = \varphi \sqrt{d_{\mathrm{b}} g}, \qquad \varphi = \begin{cases} 0.64 & \qquad d_{\mathrm{r}} \leq 0.1\,\mathrm{m} \\ 1.6 d_{\mathrm{r}}^{0.4} & 0.1\,\mathrm{m} < d_{\mathrm{r}} \leq 1\,\mathrm{m} \\ 1.6 & \qquad d_{\mathrm{r}} > 1\,\mathrm{m} \end{cases} \tag{2.48}$$

The diameter of the bubbles is calculated with the expression proposed by Mori and Wen [70], recited in eq. (2.49).

$$d_{\mathrm{b}} = d_{\mathrm{b,max}} - \left(d_{\mathrm{b,max}} - d_{\mathrm{b,0}} \right) \exp \left(-\frac{0.3h}{d_{\mathrm{r}}} \right) \tag{2.49a}$$

$$d_{\mathrm{b,0}} = 0.376 \left(u_{\mathrm{sf}} - u_{\mathrm{mf}} \right)^{2} \tag{2.49b}$$

Davidson and Harrison [20] proposed a maximum bubble diameter based on the terminal velocity of the particles, while Haider and Levenspiel [36] obtained a correlation for the terminal velocity of the particles. These correlations are recited in eq. (2.50) with the correction for the dimensionless particle diameter from Grace [35].

$$d_{\mathrm{b,max}} = 2\frac{u_{\mathrm{t}}}{g} \tag{2.50a}$$

$$u_{\mathrm{t}} = u_{\mathrm{t}}^{\star}\left(\frac{g\mu_{\mathrm{g}}\left(\rho_{\mathrm{s}}-\rho_{\mathrm{g}}\right)}{\rho_{\mathrm{g}}^{2}}\right)^{1/3} \tag{2.50b}$$

$$u_{\mathrm{t}}^{\star} = \left[\frac{18}{\left(d_{\mathrm{p}}^{\star}\right)^{2}} + \frac{2.335-1.744\phi_{\mathrm{p}}}{\sqrt{d_{\mathrm{p}}^{\star}}}\right]^{-1} \tag{2.50c}$$

$$d_{\mathrm{p}}^{\star} = 2.7 d_{\mathrm{p}}\mathrm{Ar}^{1/3} \tag{2.50d}$$

The velocity of the emulsion phase is approximated as the average between the velocity into the emulsion phase and the velocity out of the emulsion phase:

$$u_{\mathrm{e}} = \frac{1}{2}\left(u_{\mathrm{sf}} + \frac{\hat{V}_{\mathrm{e|fb}}}{\left(1-\delta\right)A_{\mathrm{r}}}\right) \tag{2.51}$$

In addition to mass transfer between the bubble and the emulsion phase, there is also heat transfer between the two phases. The heat transfer coefficients are calculated similarly to the mass transfer coefficients, and are given in eq. (2.52) [37, 57].

$$h_{\mathrm{b|e}} = \left(\frac{1}{h_{\mathrm{b|c}}} + \frac{1}{h_{\mathrm{c|e}}}\right)^{-1} \tag{2.52a}$$

$$h_{\mathrm{b|c}} = 4.5\frac{u_{\mathrm{e}}\rho_{\mathrm{g}}c_{p,\mathrm{g}}}{d_{\mathrm{b}}} + 5.85\frac{\left(\rho_{\mathrm{g}}c_{p,\mathrm{g}}k_{\mathrm{g}}\right)^{1/2} g^{1/4}}{d_{\mathrm{b}}^{5/4}} \tag{2.52b}$$

$$h_{\mathrm{c|e}} = 6.77\left(\frac{\varepsilon_{\mathrm{e}}u_{\mathrm{br}}\rho_{\mathrm{g}}c_{p,\mathrm{g}}k_{\mathrm{g}}}{d_{\mathrm{b}}^{3}}\right)^{1/2} \tag{2.52c}$$

2.6 Thermodynamics

The thermodynamics in the PRM and the CM differ slightly. These differences will be elucidated in the succeeding subsections.

2.6.1 Pressure

The PRM uses the Redlich-Kwong equation of state [38, 84], while the CM applies a compressibility factor. The Redlich-Kwong equation of state is given in eq. (2.53), and the definition of the compressibility factor is given in eq. (2.54).

$$p = \frac{RT\sum_{\forall i} c_i}{1-B} - \frac{A}{\sqrt{T}\left(1+B\right)}, \qquad \forall i \in \left\{\mathrm{H}_2, \mathrm{C}_{3=}, \mathrm{C}_{2=}, \mathrm{N}_2\right\} \tag{2.53a}$$

$$A = \sum_{\forall j}\sum_{\forall i} c_i c_j \sqrt{a_i a_j} \tag{2.53b}$$

$$B = \sum_{\forall i} c_i b_i \tag{2.53c}$$

$$a_i = \frac{R^2 T_{\mathrm{c},i}^{5/2}}{9\left(2^{1/3}-1\right) p_{\mathrm{c},i}} \tag{2.53d}$$

$$b_i = \frac{\left(2^{1/3}-1\right) RT_{\mathrm{c},i}}{3p_{\mathrm{c},i}} \tag{2.53e}$$

The subscript c denotes a critical constant; the critical constants for each of the components are given in Table 2.2.

Table 2.2 – Critical constants of the gaseous components.

component	critical temperature	critical pressure	source
Hydrogen	33.15 K	$12.964 \cdot 10^5$ Pa	[39]
Nitrogen	126.192 K	$33.9 \cdot 10^5$ Pa	[39]
Propylene	364.9 K	$45.9 \cdot 10^5$ Pa	[40]
Ethylene	282.35 K	$50.6 \cdot 10^5$ Pa	[40]

The compressibility factor Z, which is used to estimate the pressure of the CM is defined as [38]:

$$Z = \frac{p}{RT \sum_{\forall i} c_i}, \qquad \forall i \in \{\mathrm{H_2}, \mathrm{C_{3=}}, \mathrm{C_{2=}}, \mathrm{N_2}\} \tag{2.54}$$

2.6.2 Energy balance

The energy balance for a simple control volume, depicted in Figure 2.4, is given below [38]:

$$\dot{E} = \hat{E}_{\text{in}} - \hat{E}_{\text{out}} + \hat{Q} + \hat{W} \tag{2.55}$$

E is the total energy which includes internal energy, potential energy and kinetic energy. $\hat{Q}$ and $\hat{W}$ are the flow of heat to the system and the work performed by the surroundings on the system, respectively. By assuming that the system is nonmoving and that the flow of mechanical energy can be neglected, the energy balance can be written as:

$$\dot{U} = \hat{U}_{\text{in}} - \hat{U}_{\text{out}} + \hat{Q} + \hat{W} \tag{2.56}$$

The work performed on the system can be split into distinct contributions, the pressure-volume work associated with the flow of mass in and out of

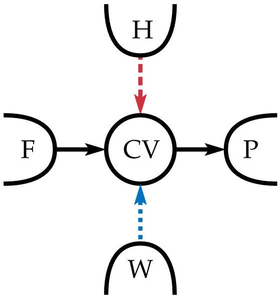


FIGURE 2.4 – A simple control volume. The black lines indicate flow of mass, while the red dashes and the blue dots denote heat flow and work flow, respectively.

the system, the pressure-volume work associated with change in volume of the system, the shaft work and all other contributions:

$$\hat{W} = \left(p\hat{V}\right)_{\text{in}} - \left(p\hat{V}\right)_{\text{out}} - p_{\text{ext}}\frac{\mathrm{d}V}{\mathrm{d}t} + \hat{W}_{\text{s}} + \hat{W}_{\text{other}} \tag{2.57}$$

By introducing the definition of enthalpy, $H = U + pV$, and neglecting all other work than pressure-volume work and shaft work, the following is obtained:

$$\dot{U} = \hat{H}_{\text{in}} - \hat{H}_{\text{out}} + \hat{Q} - p_{\text{ext}}\frac{\mathrm{d}V}{\mathrm{d}t} + \hat{W}_{\text{s}} \tag{2.58}$$

The left-hand side of this equation is usually rewritten in therms of enthalpy instead of internal energy, which yields:

$$\dot{H} = \hat{H}_{\text{in}} - \hat{H}_{\text{out}} + \hat{Q} + (p - p_{\text{ext}})\frac{\mathrm{d}V}{\mathrm{d}t} + V\frac{\mathrm{d}p}{\mathrm{d}t} + \hat{W}_{\text{s}} \tag{2.59}$$

By examining different differentials for the enthalpy and applying a set of appropriate Maxwell relations, the energy balance can be rewritten in terms of temperature [47]:

$$\begin{aligned} C_p \frac{\mathrm{d}T}{\mathrm{d}t} + \sum_{\forall c} h_c \dot{n}_c = {} & \hat{H}_{\text{in}} - \hat{H}_{\text{out}} + \hat{Q} \\ & + \left(p - p_{\text{ext}}\right) \frac{\mathrm{d}V}{\mathrm{d}t} + T \left(\frac{\partial V}{\partial T}\right)_p \frac{\mathrm{d}p}{\mathrm{d}t} + \hat{W}_{\text{s}} \end{aligned} \tag{2.60}$$

The motivation behind this transformation is that the temperature is required in many of the calculations, and solving the energy balance with respect to enthalpy would require an additional algebraic equation for the temperature. By utilizing the fact that enthalpy is Euler homogeneous of degree one [38], the flows of enthalpy can be rewritten as:

$$\hat{H} = \sum_{\forall c} h_c \hat{n}_c \tag{2.61}$$

Inserting the mole balances and eq. (2.61) into eq. (2.60) yields:

$$\begin{aligned} C_p \frac{\mathrm{d}T}{\mathrm{d}t} = {} & \sum_{\forall c} \left[\left(h_{c,\text{in}} - h_{c,\text{out}}\right) \hat{n}_{c,\text{in}} - h_{c,\text{out}} \tilde{n}_c\right] \\ & + \hat{Q} + \left(p - p_{\text{ext}}\right) \frac{\mathrm{d}V}{\mathrm{d}t} + T \left(\frac{\partial V}{\partial T}\right)_p \frac{\mathrm{d}p}{\mathrm{d}t} + \hat{W}_{\text{s}} \end{aligned} \tag{2.62}$$

By applying the following assumptions; the volume of the control volume is constant, neglecting any contributions by changes in pressure and shaft work; the well-known temperature equation is obtained [31];

$$C_p \frac{\mathrm{d}T}{\mathrm{d}t} = \sum_{\forall c} \left[\hat{n}_{c,\text{in}} \int_T^{T_{\text{in}}} c_{p,c} \,\mathrm{d}\tau\right] - \sum_{\forall r} \Delta_{\text{rx}} h_r \tilde{N}_r + \hat{Q} \tag{2.63}$$

where r indicates the different reactions. It is assumed that only the propagations contribute to the heat of reactions and that the heat capacities are constant, hence:

$$\sum_{\forall c} n_c c_{p,c} \frac{\mathrm{d}T}{\mathrm{d}t} = \sum_{\forall c} \left[c_{p,c} \hat{n}_{c,\text{in}} \left(T_{\text{in}} - T\right)\right] - \sum_{\forall r} \Delta_{\text{rx}} h_r \tilde{N}_r + \hat{Q} \tag{2.64}$$

Equation (2.64) is applied by the CM without any further manipulations, while the PRM requires an energy balance for each of the control volumes. The reactions are only taking place in the bubble and the emulsion phases, while it is assumed that the heat exchanger is the only control volume with heat transfer. Thus, rewriting eq. (2.64) for each of the control volumes:

$$\begin{aligned}\sum_{\forall c} n_{c,\mathrm{fb}} c_{p,c} \frac{\mathrm{d}T_{\mathrm{fb}}}{\mathrm{d}t} &= \sum_{\forall c} \left[c_{p,c} \hat{n}_{c,\mathrm{e|fb}} \left(T_{\mathrm{e}} - T_{\mathrm{fb}} \right) \right] \\ &\quad + \sum_{\forall c} \left[c_{p,c} \hat{n}_{c,\mathrm{b|fb}} \left(T_{\mathrm{b}} - T_{\mathrm{fb}} \right) \right]\end{aligned} \tag{2.65a}$$

$$\begin{aligned}\sum_{\forall c} n_{c,\mathrm{hex}} c_{p,c} \frac{\mathrm{d}T_{\mathrm{hex}}}{\mathrm{d}t} &= \sum_{\forall c} \left[c_{p,c} \hat{n}_{c,\mathrm{fb|hex}} \left(T_{\mathrm{fb}} - T_{\mathrm{hex}} \right) \right] \\ &\quad + \sum_{\forall c} \left[c_{p,c} \hat{n}_{c,\mathrm{f|hex}} \left(T_{\mathrm{f}} - T_{\mathrm{hex}} \right) \right] + \hat{Q}\end{aligned} \tag{2.65b}$$

$$\begin{aligned}\sum_{\forall c} n_{c,\mathrm{e}} c_{p,c} \frac{\mathrm{d}T_{\mathrm{e}}}{\mathrm{d}t} &= \sum_{\forall c} \left[c_{p,c} \hat{n}_{c,\mathrm{b|e}} \left(T_{\mathrm{b}} - T_{\mathrm{e}} \right) \right] \\ &\quad + \sum_{\forall c} \left[c_{p,c} \hat{n}_{c,\mathrm{f|e}} \left(T_{\mathrm{f}} - T_{\mathrm{e}} \right) \right] \\ &\quad + \sum_{\forall c} \left[c_{p,c} \hat{n}_{c,\mathrm{hex|e}} \left(T_{\mathrm{hex}} - T_{\mathrm{e}} \right) \right] \\ &\quad - \sum_{\forall r} \Delta_{\mathrm{rx}} h_r \tilde{N}_{r,\mathrm{e}}\end{aligned} \tag{2.65c}$$

$$\begin{aligned}\sum_{\forall c} n_{c,\mathrm{b}} c_{p,c} \frac{\mathrm{d}T_{\mathrm{b}}}{\mathrm{d}t} &= \sum_{\forall c} \left[c_{p,c} \hat{n}_{c,\mathrm{e|b}} \left(T_{\mathrm{e}} - T_{\mathrm{b}} \right) \right] \\ &\quad + \sum_{\forall c} \left[c_{p,c} \hat{n}_{c,\mathrm{f|b}} \left(T_{\mathrm{f}} - T_{\mathrm{b}} \right) \right] \\ &\quad + \sum_{\forall c} \left[c_{p,c} \hat{n}_{c,\mathrm{hex|b}} \left(T_{\mathrm{hex}} - T_{\mathrm{b}} \right) \right] \\ &\quad - \sum_{\forall r} \Delta_{\mathrm{rx}} h_r \tilde{N}_{r,\mathrm{b}}\end{aligned} \tag{2.65d}$$

2.6.3 Heat exchanger

Two distinct models for the heat exchanger are applied to the CM and the PRM. In the CM, the flow of heat is assumed to be proportional to the temperature difference between the CW and the reactor temperature, as displayed in eq. (2.66). A logarithmic mean temperature difference is applied to the heat transfer in the PRM, which is shown in eq. (2.67).

$$\hat{Q}_{\mathrm{cm}} = UA\left(T - T_{\mathrm{cw}}\right) \tag{2.66}$$

The simple heat transfer model in eq. (2.66) is valid for heat transfer between two control volumes with uniform temperature [34, 103].

$$\hat{Q}_{\mathrm{prm}} = UA\Delta T_{\mathrm{lm}} \tag{2.67a}$$

$$\Delta T_{\mathrm{lm}} = \frac{\left(T_{\mathrm{fb}} - T_{\mathrm{cw}}\right) - \left(T_{\mathrm{hex}} - T_{\mathrm{cw}}\right)}{\ln\left(\frac{T_{\mathrm{fb}} - T_{\mathrm{cw}}}{T_{\mathrm{hex}} - T_{\mathrm{cw}}}\right)} \tag{2.67b}$$

The model in eq. (2.67) is valid for a pure co or countercurrent heat-exchanger which exhibits plug flow characteristics [7, 34]. It is assumed that the temperature of the cold side, i.e., the CW, is constant.

2.7 State representation

The models are written in a state representation, in which the states are the variables for which a time-derivative is given. In addition, the models include estimation of the available measurements; the models can then be written as

$$\dot{\mathbf{x}} = \mathbf{f}\left(\mathbf{x}, \mathbf{u}; \boldsymbol{\theta}\right) \tag{2.68a}$$

$$\mathbf{y} = \mathbf{g}\left(\mathbf{x}, \mathbf{u}; \boldsymbol{\theta}\right) \tag{2.68b}$$

$$\mathbf{z} = \mathbf{h}\left(\mathbf{x}, \mathbf{u}; \boldsymbol{\theta}\right) \tag{2.68c}$$

where $\mathbf{x}$, $\mathbf{u}$, $\mathbf{y}$, $\mathbf{z}$ and $\boldsymbol{\theta}$ are vectors that contain the states, the inputs, the measurements, the outputs and the parameters, respectively. The vector of measurements and inputs are the same for both models, and are given in eqs. (2.69) and (2.72), respectively. In the PRM, the measurements of the composition of the gas phase, the pressure and the temperature are obtained from the freeboard section of the reactor, as indicated by Figure 2.2.

$$\mathbf{y} = \begin{bmatrix} x_{\mathrm{H}_2}, & x_{\mathrm{M}_1}, & x_{\mathrm{M}_2}, & p, & h, & T, & \hat{V}_{\mathrm{pu}}, & \hat{V}_{\mathrm{pol}}, & T_{\mathrm{cw}}, & \mathrm{MFI} \end{bmatrix}^{\mathsf{T}} \tag{2.69}$$

The mole fractions are calculated by eq. (2.70), while the melt flow index (MFI) is approximated by eq. (2.71) [66, 94, 97].

$$x_i = \frac{c_i}{\sum_{\forall c} c_c}, \qquad \forall c, i \in \left\{\mathrm{H}_2, \mathrm{C}_{3=}, \mathrm{C}_{2=}, \mathrm{N}_2\right\} \tag{2.70}$$

$$\mathrm{MFI} = 3.346 \cdot 10^{17} \bar{M}_{\mathrm{w}}^{-3.472} \tag{2.71}$$

$$\mathbf{u} = \begin{bmatrix} \hat{V}_{\mathrm{H}_2,\mathrm{f}}, & \hat{V}_{\mathrm{M}_1,\mathrm{f}}, & \hat{V}_{\mathrm{M}_2,\mathrm{f}}, & \hat{V}_{\mathrm{N}_2,\mathrm{f}}, & \hat{m}_{\mathrm{cat,f}}, & T_{\mathrm{sp}} \end{bmatrix}^{\mathsf{T}} \tag{2.72}$$

In polymer production it is often desirable to control variables that are not directly measured, such as the polydispersity index (PDI), the molecular weight of the polymer and its composition [16, 63, 65]. The additional outputs chosen in this thesis are collected in the vector $\mathbf{z}$, given in eq. (2.73).

$$\mathbf{z} = \begin{bmatrix} \mathrm{PDI}, & \bar{M}_{\mathrm{w}}, & x_{\mathrm{pol}}, & \hat{m}_{\mathrm{pol}}, & \hat{m}_{\mathrm{pu}}, & X \end{bmatrix}^{\mathsf{T}} \tag{2.73}$$

The PDI is the ratio between the mass average molecular mass and the number average molecular mass [37, 46, 66, 97]

$$\text{PDI} = \frac{\bar{M}_\mathrm{w}}{\bar{M}_\mathrm{n}} \tag{2.74}$$

where the average molecular masses are given in eq. (2.75).

$$\bar{M}_\mathrm{w} = M_\mathrm{M}\frac{\mu_2}{\mu_1} \tag{2.75a}$$

$$\bar{M}_\mathrm{n} = M_\mathrm{M}\frac{\mu_1}{\mu_0} \tag{2.75b}$$

$$M_\mathrm{M} = \sum_{m=1}^{2} x_{\mathrm{B}_m} M_{\mathrm{B}_m} \tag{2.75c}$$

The fraction of propylene bound in the polymer is obtained by:

$$x_\mathrm{pol} = \frac{n_{\mathrm{B}_1}}{\sum_{k=1}^{2} n_{\mathrm{B}_k}} \tag{2.76}$$

The production rate is given in eq. (2.77), while the purge rate is calculated by eq. (2.78).

$$\hat{m}_\mathrm{pol} = m_\mathrm{pol}\frac{\hat{V}_\mathrm{pol}}{V_\mathrm{s}} \tag{2.77}$$

$$\hat{m}_\mathrm{pu} = \rho_\mathrm{g}\hat{V}_\mathrm{pu} \tag{2.78}$$

The productivity of the reactor, X, is defined as the ratio between the mass of polymer produced and the mass of catalyst [105]:

$$X = \frac{m_\mathrm{pol}}{m_\mathrm{Ti}} \tag{2.79}$$

2.7.1 Control model

The states used in the CM are given in eq. (2.80);

$$\mathbf{x}_{\mathrm{cm}} = \begin{bmatrix} n_{\mathrm{H}_2} & n_{\mathrm{M}_1} & n_{\mathrm{M}_2} & n_{\mathrm{N}_2} & n_{\mathrm{B}_1} & n_{\mathrm{B}_2} & \lambda_0^1 & \lambda_0^2 & \lambda_1^1 & \lambda_1^2 \\ \xi_0 & \xi_1 & \mu_2 & m_{\mathrm{cat}} & T & \int_0^t e_h \,\mathrm{d}\tau & \int_0^t e_p \,\mathrm{d}\tau & \int_0^t e_T \,\mathrm{d}\tau \end{bmatrix}^{\mathsf{T}} \tag{2.80}$$

this includes the number of moles of the gaseous components, the moles of monomers bound in the polymer, the moments of the chain length distribution, the total mass of catalyst, the temperature and the integrals of the controller errors.

2.7.2 Plant replacement model

The states used in the plant replacement are given in eq. (2.81);

$$\mathbf{x}_{\mathrm{prm}} = \begin{bmatrix} n_{\mathrm{H}_2,\mathrm{hex}} & n_{\mathrm{M}_1,\mathrm{hex}} & n_{\mathrm{M}_2,\mathrm{hex}} & n_{\mathrm{N}_2,\mathrm{hex}} & T_{\mathrm{hex}} & n_{\mathrm{H}_2,\mathrm{fb}} & n_{\mathrm{M}_1,\mathrm{fb}} \\ n_{\mathrm{M}_2,\mathrm{fb}} & n_{\mathrm{N}_2,\mathrm{fb}} & T_{\mathrm{fb}} & n_{\mathrm{H}_2,\mathrm{b}} & n_{\mathrm{M}_1,\mathrm{b}} & n_{\mathrm{M}_2,\mathrm{b}} & n_{\mathrm{N}_2,\mathrm{b}} & T_{\mathrm{b}} \\ n_{\mathrm{H}_2,\mathrm{e}} & n_{\mathrm{M}_1,\mathrm{e}} & n_{\mathrm{M}_2,\mathrm{e}} & n_{\mathrm{N}_2,\mathrm{e}} & T_{\mathrm{e}} & n_{\mathrm{B}_1} & n_{\mathrm{B}_2} & \lambda_0^1 & \lambda_0^2 & \lambda_1^1 \\ \lambda_1^2 & \xi_0 & \xi_1 & \mu_2 & m_{\mathrm{cat}} & \int_0^t e_h \,\mathrm{d}\tau & \int_0^t e_p \,\mathrm{d}\tau & \int_0^t e_T \,\mathrm{d}\tau \end{bmatrix}^{\mathsf{T}} \tag{2.81}$$

which includes the number of moles of the gaseous components in each control volume, the temperature of each control volume, the moles of monomers bound in the polymer, the moments of the chain length distribution, the total mass of catalyst and the integrals of the controller errors.

2.8 CONTROLLERS

The general form of a proportional integral derivative (PID) controller is given in eq. (2.82) [3]

$$u(t) = u(0) + K_{\mathrm{c}} \left(e_{\mathrm{P}}(t) + \frac{1}{\tau_{\mathrm{I}}} \int_0^t e_{\mathrm{I}}(\tau)\,\mathrm{d}\tau + \tau_{\mathrm{D}} \frac{\mathrm{d}e_{\mathrm{D}}(t)}{\mathrm{d}t} \right) \tag{2.82}$$

in which K_{c} is the controller gain, τ_{I} and τ_{D} are the integral and derivative time, respectively. e is the set point offset defined in eq. (2.83), while u is the manipulated variable (MV), i.e., the input. The offset may be treated differently for each of the three terms. The offset for the integral term usually contains an additional term to prevent windup issues, while the offset applied in the derivative action is filtered to minimize the effect of measurement noise [3, 92].

$$e = y_{\mathrm{sp}} - y \tag{2.83}$$

y is the controlled variable (CV), i.e., the output or the measurement, and y_{sp} is its set point, i.e., the desired value. The input is usually limited, by physical constraints, between a maximum and a minimum value:

$$u_{\mathrm{min}} \leq u \leq u_{\mathrm{max}} \tag{2.84}$$

At the limits, the input is saturated which can cause reset windup of the integral action [92]. To remedy this problem, an antiwindup scheme was applied [3], which is illustrated in Figure 2.5.

The controllers were tuned sequentially, with a simplified version of the sequential algorithm of Hovd and Skogestad [45] (adapted from Seborg

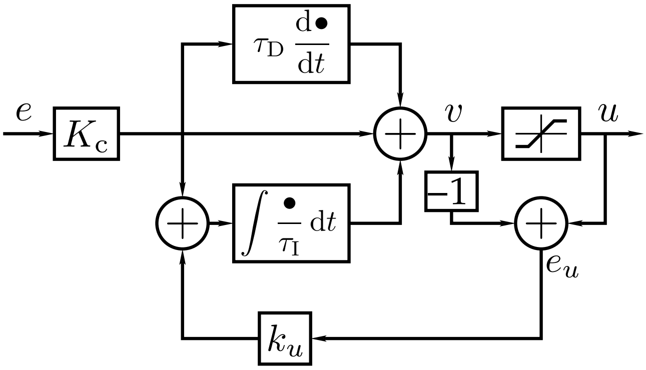

$\tau_{\mathrm{D}} \frac{\mathrm{d}\bullet}{\mathrm{d}t}$
e
K_{c}
v
u
-1
$\int \frac{\bullet}{\tau_{\mathrm{I}}} \mathrm{d}t$
e_u
k_u

Figure 2.5 – The structure of a simplified proportional integral derivative (PID) controller with the antiwindup scheme of Åström and Murray [3]. e is the set point offset, v is the desired input, u is the actual input. e_u is the input offset, which is fed back to the integrator.

et al. [92, ch. 18]). The linearizations were performed numerically with a central difference scheme.[2] Each controller was tuned by applying Matlab's PID toolbox[3] to the linearized model. This led to the following tuning algorithm:

1. Linearize the open-loop model.
2. Tune the LC, and close the controller loop.
3. Linearize the model with the LC loop closed.
4. Tune the PC and close the controller loop.
5. Linearize the model with both the LC and the PC loops closed.
6. Tune the TC and close the controller loop.

[2] http://www.iue.tuwien.ac.at/phd/heinzl/node27.html
[3] http://mathworks.com/help/control/pid-controller-design.html

Table 2.3 – The controller settings obtained with Matlab's pid toolbox.

controlled variable (y)	manipulated variable (u)	gain (K_c)	integral time (τ_I)
h	$\hat{V}_{pol}$	$-0.034\,99\,m^2\,s^{-1}$	440.1 s
p	$\hat{V}_{pu}$	$-3.451 \cdot 10^{-7}\,m^3\,s^{-1}\,Pa^{-1}$	907.7 s
T	T_{cw}	$2.347\,K\,K^{-1}$	1455 s

Table 2.4 – The controller constraints chosen in the simulations.

manipulated variable (u)	maximum (u_{max})	minimum (u_{min})
$\hat{V}_{pol}$	$0.02\,m^3\,s^{-1}$	$0\,m^3\,s^{-1}$
$\hat{V}_{pu}$	$0.05\,m^3\,s^{-1}$	$0\,m^3\,s^{-1}$
T_{cw}	60 °C	10 °C

The obtained tuning parameters are given in Table 2.3; the antiwindup tuning parameter, k_u was set equal to K_c^{-1} as recommended by Åström and Murray [3]. For simplicity, all the controllers were tuned as proportional integral (pi) controllers; the implementation of the pi controllers is shown in Code snippet d.1 in Appendix d. In addition, some physical constraints were imposed on the mvs of the controllers, these are given in Table 2.4.

2.9 Implementation

Both the prm and the cm were implemented both Matlab and C. The models were first implemented in Matlab for visualization and data generation purposes, and later in C to be able to interact with Cybernetica's programs. All additional parameters required for the simulations are given in Tables A.1 and A.2 in Appendix A.

The Matlab implementation of the plant replacement is displayed in Appendix B. The derivatives of the states are given in Code snippet B.1, while the measurements are shown in Code snippet B.2. An implicit solver, `ode15s`,[4] was applied for the integration of the model equations. A sample script which demonstrates how the model equations can be integrated is given in Code snippet B.3. A fourth-order Runge-Kutta scheme [56, p. 917] with a constant time-step was applied in the C implementation, this scheme is presented in Code snippet B.5. The remainder of the C source code is very similar to the Matlab code, and has been omitted for brevity.

The Matlab source code of the CM is given in Appendix C. The derivatives of the states are displayed in Code snippet C.1, while the measurements and outputs are shown in Code snippets C.2 and C.3, respectively. A second-order Runge-Kutta method (see Constantinides and Mostoufi [17, p. 290]) was used to integrate the model equations. The implementation in C of this model has also been excluded due to the many similarities with the Matlab code.

A set of input steps was applied to the models and a selection of the resulting step responses is plotted in Appendix E. These steps were applied after the offline parameter estimation in Chapter 3. Some discrep-

[4] http://mathworks.com/help/matlab/ref/ode15s.html

ancies between the CM and the PRM can be observed, these deviations are mitigated further by applying a recursive estimation technique, which is described in Chapter 4. These step responses served as a qualitative verification of the implementations, but due to an absence of data from an industrial scale reactor, the implementations have not been validated. However, similar models have been compared against experimental data by Shamiri, Hussain, Mjalli, and Mostoufi [95]; their results agreed reasonably well with the experimental data, even without estimation of any parameters.

Chapter 3

Offline parameter estimation

> The optimum value of any parameter (or set of parameters) is that value (or set of values) of which the likelihood is greatest.
>
> — R. A. Fisher, 1922[1]

To ensure the best possible consistency between the plant replacement model (PRM) and the control model (CM), a selection of parameters in the CM was fitted against the outputs from a series of step responses. This can be done by utilizing Cybernetica's tool for offline parameter estimation, ModelFit.[2] The parameters chosen for fitting were the heat transfer coefficient, the compressibility, the average void fraction and a set of reaction rate corrections. A brief explanation of the applied theory is provided in the subsequent section.

[1] Fisher, R. A., "On the mathematical foundations of theoretical statistics," *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 222, no. 594–604, pp. 309–368, 1922. DOI: 10.1098/rsta.1922.0009

[2] http://www.cybernetica.biz/v3/products/ModelFit/index.html

3.1 Theory

ModelFit calculated the parameters selected for estimation by solving the following optimization problem

$$\begin{aligned} &\min_{\boldsymbol{\theta}} J\left(\boldsymbol{\theta}\right) \\ \text{subject to } &\mathbf{x}_k = \mathbf{x}_{k-1} + \int_{t_{k-1}}^{t_k} \mathbf{f}\left(\mathbf{x}(\tau), \mathbf{u}(\tau); \boldsymbol{\theta}\right) \mathrm{d}\tau,\ \forall k \in \{1, \dots, n\} \\ &\mathbf{y}_k = \mathbf{g}\left(\mathbf{x}_k, \mathbf{u}_k; \boldsymbol{\theta}\right) \\ &\boldsymbol{\theta}_{\mathrm{min}} \leq \boldsymbol{\theta} \leq \boldsymbol{\theta}_{\mathrm{max}} \end{aligned} \tag{3.1}$$

in which the cost function, J, is the sum of squares of the difference between the estimated measurement and the actual measurement, given in eq. (3.2). It is worth noting that the cost function is only dependent on the parameters, the measurements are implicitly given by the parameters through the model equations. k denotes the sample number, while n is the total number of samples. Equation (3.1) is solved by applying a sequential quadratic programming (SQP) method as described by Nocedal and Wright [74, ch. 18]. The SQP algorithm works by approximating the cost function by a quadratic function and linearizing the constraints. This quadratic subproblem is solved at each iteration until the criteria of convergence has been met.

$$J\left(\boldsymbol{\theta}\right) = \left(\mathbf{y} - \mathbf{y}_{\mathrm{m}}\right)^{\mathsf{T}} \mathbf{Q} \left(\mathbf{y} - \mathbf{y}_{\mathrm{m}}\right) \tag{3.2}$$

The block vectors[3] $\mathbf{y}$ and $\mathbf{y}_{\mathrm{m}}$ contain all the variables from eq. (2.73) in Chapter 2 at every sample. The block matrix[2] $\mathbf{Q}$ contains the weight of

[3] A block vector/matrix is a vector/matrix that is composed of smaller vectors/matrices.

each variable, which is used to make the cost function dimensionless and it could be used to favor between different measurements and samples.

The weight matrix used to estimate the parameters was chosen as

$$\mathbf{Q} = \mathbf{I}_n \otimes \operatorname{diag}(\mathbf{q}) \tag{3.3a}$$

$$\mathbf{q} = \begin{bmatrix} q^2_{x_{\mathrm{H}_2}} & q^2_{x_{\mathrm{C}_{3=}}} & q^2_{x_{\mathrm{C}_{2=}}} & q^2_p & q^2_h & q^2_T & q^2_{\hat{V}_{\mathrm{pu}}} & q^2_{\hat{V}_{\mathrm{pol}}} & q^2_{T_{\mathrm{cw}}} & q^2_{\mathrm{MFI}} \end{bmatrix}^{\mathsf{T}} \tag{3.3b}$$

where the weights of the individual measurements are displayed in Table 3.1. **I** is the identity matrix[4] and ⊗ is the Kronecker product.[5] In this formulation, all the samples have been weighted equally; each measurement was sampled at regularly spaced intervals with a sample time of one minute. The measurements of pressure, level and temperature were not taken into account, because they are controlled by proportional integral (PI) controllers, hence they cannot be considered independent measurements.

The measurement data were generated by applying a series of steps in the inputs to the PRM. Each step was simulated for ten hours before a new step was applied. The initial input was:

$$\mathbf{u}_0 = \Big[4.5 \cdot 10^{-5}\,\mathrm{m^3\,s^{-1}} \quad 4.275 \cdot 10^{-2}\,\mathrm{m^3\,s^{-1}} \quad 2.205 \cdot 10^{-3}\,\mathrm{m^3\,s^{-1}} \quad 4.5 \cdot 10^{-6}\,\mathrm{m^3\,s^{-1}} \quad 3.385 \cdot 10^{-3}\,\mathrm{kg\,s^{-1}} \quad 353.15\,\mathrm{K} \Big]^{\mathsf{T}}$$

The steps were performed by perturbing the input vector by adding a positive and a negative step sequentially for each of the inputs. The

[4] http://mathworld.wolfram.com/IdentityMatrix.html
[5] http://mathworld.wolfram.com/KroneckerProduct.html

Table 3.1 – The weights for each measurement used in the parameter estimation.

MEASUREMENT	WEIGHT
Mole fraction of hydrogen	$q_{x_{\mathrm{H}_2}} = \frac{1}{\sqrt{5 \cdot 10^{-4}}}$
Mole fraction of propylene	$q_{x_{\mathrm{C}_{3=}}} = \frac{1}{1 \cdot 10^{-4}}$
Mole fraction of ethylene	$q_{x_{\mathrm{C}_{2=}}} = \frac{1}{1 \cdot 10^{-4}}$
Pressure	$q_p = 0$
Level	$q_h = 0$
Temperature	$q_T = 0$
Purge rate	$q_{\hat{V}_{\mathrm{pu}}} = \frac{1}{1 \cdot 10^{-3}\,\mathrm{m}^3\,\mathrm{s}^{-1}}$
Production rate rate	$q_{\hat{V}_{\mathrm{pol}}} = \frac{1}{1 \cdot 10^{-5}\,\mathrm{m}^3\,\mathrm{s}^{-1}}$
Cooling water temperature	$q_{T_{\mathrm{cw}}} = \frac{1}{1\,\mathrm{K}}$
Melt flow index	$q_{\mathrm{MFI}} = \frac{1}{1 \cdot 10^{-3}\,\mathrm{dg}\,\mathrm{min}^{-1}}$

steps of the feeds were set equal to 5 % of $\mathbf{u}(0)$, while the steps in the temperature set point were chosen to be ±3 K. To validate the estimated parameters, a new dataset was constructed similarly, but with the steps in a different order and with twice the magnitude.

3.2 Results and discussion

The estimated parameters are shown in Table 3.2 together with their respective initial guess. The corrections for the reaction rates are applied as:

$$kp^j_{km} = \psi^j_m kp^j_{km}, \quad k, m \in \{\mathrm{C}_{3=}, \mathrm{C}_{2=}\}, \quad j \in \{1, 2\} \tag{3.4a}$$

$$kfh^j_m = \psi_{\mathrm{H}_2} kfh^j_m \tag{3.4b}$$

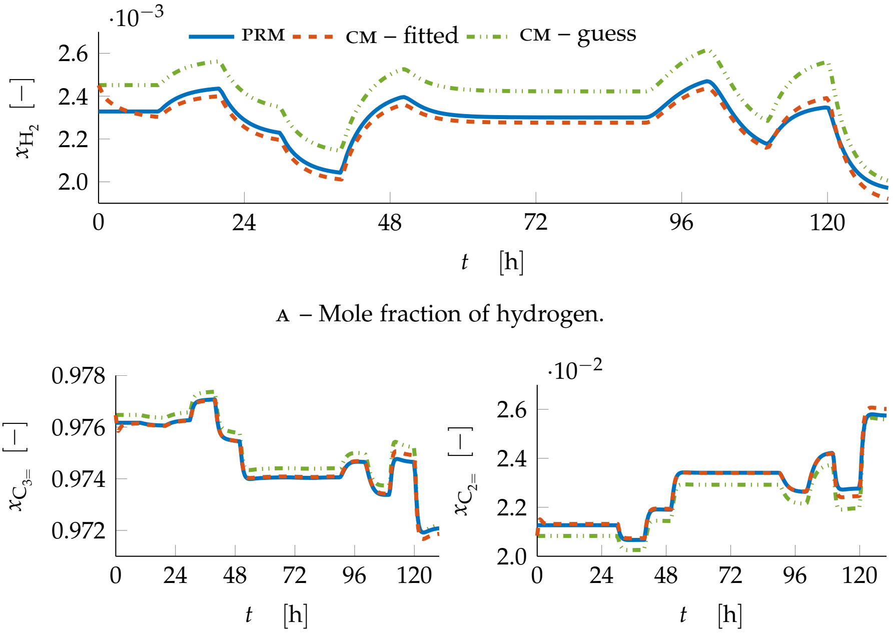
Table 3.2 – The results of the offline parameter estimation.

PARAMETER	INITIAL GUESS	ESTIMATED VALUE
Overal heat transfer coefficient, UA	$80\,\mathrm{kW\,K^{-1}}$	$83.58\,\mathrm{kW\,K^{-1}}$
Compressibility, Z	0.75	0.7222
Average void fraction, $\varepsilon_{\mathrm{avg}}$	0.7	0.7000
Correction for $C_{3=}$ at site 1, $\psi^1_{C_{3=}}$	1.0	0.9399
Correction for $C_{3=}$ at site 2, $\psi^2_{C_{3=}}$	1.0	1.006
Correction for $C_{2=}$ at site 1, $\psi^1_{C_{2=}}$	1.0	0.9149
Correction for $C_{2=}$ at site 2, $\psi^2_{C_{2=}}$	1.0	1.066
Correction for H_2, ψ_{H_2}	1.0	0.4570

These corrections are included to account for the effects of the temperature and concentration differences in the bubble and emulsion phases. A two-site correction for hydrogen was also carried out, but that resulted in one of the correction being equal to zero without producing a better fit to the PRM, hence a single correction was chosen. Several initial guesses were applied, but it was observed that the initial estimate for each of the parameters did not affect the results.

The graphical depictions of the results are shown in Figures 3.1 to 3.3. The validation mentioned in the previous section, for the mole fraction of hydrogen, is displayed in Figure 3.4, while the validation of the melt flow index (MFI) is presented in Figure 3.5. The remainder of the validations are left to Appendix F.

By inspecting Figures 3.1 to 3.3, it is evident that the estimation provides results which are in good agreement with the PRM. By examining



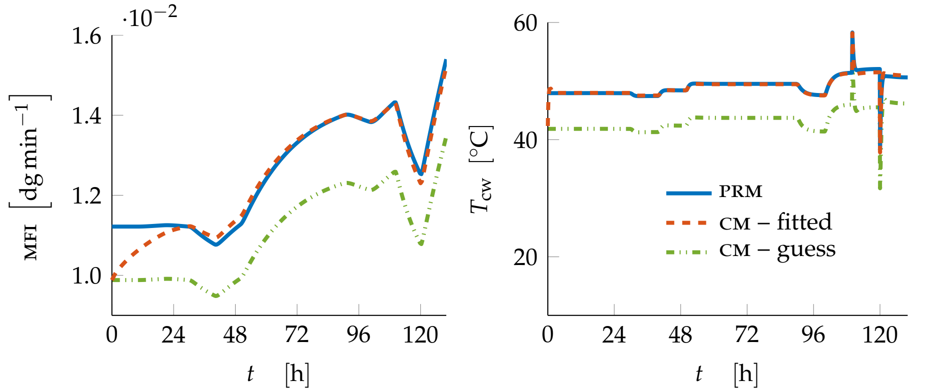
A – Mole fraction of hydrogen.

B – Mole fraction of propylene.

C – Mole fraction of ethylene.

Figure 3.1 – The results of the offline parameter estimation for the mole fractions. The blue line is the plant replacement model (PRM) while the red dashed line and the green dash-dotted line denote the control model (CM) before and after fitting, respectively.

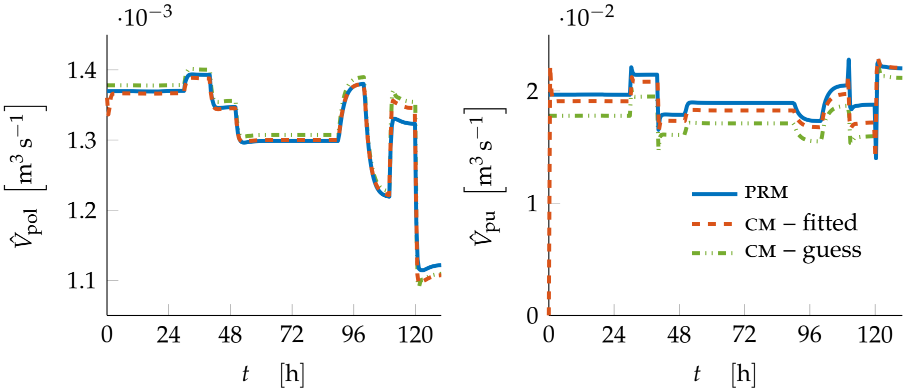
Figures 3.4 and 3.5 it apparent that the estimation provides accurate predictions for a doubling in the input steps. This is one of the advantages a nonlinear first-principles model possesses, compared to a linear heuristic or empirical model [90], i.e., being able to produce more precise predictions independent of operating conditions. However, the estimated parameters may not provide sufficiently accurate predictions for all operating conditions. To remedy this, a recursive estimation technique will be applied, which will be elucidated in Chapter 4.



A Melt flow index.

B – Temperature of the cooling water.

Figure 3.2 – The results of the offline parameter estimation for the melt flow index (MFI) and the temperature of the cooling water (CW). The blue line is the plant replacement model (PRM) while the red dashed line and the green dash-dotted line denote the control model (CM) before and after fitting, respectively.



A – Production rate.

B – Purge rate.

Figure 3.3 – The results of the offline parameter estimation for the product and purge rates. The blue line is the plant replacement model (PRM) while the red dashed line and the green dash-dotted line denote the control model (CM) before and after fitting, respectively.

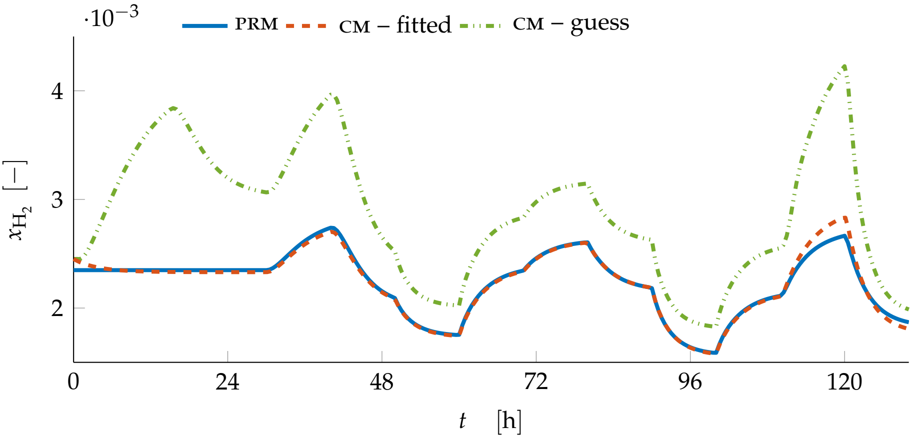


FIGURE 3.4 – The validation of the mole fraction of hydrogen. The blue line is the plant replacement model (PRM) while the red dashed line and the green dash-dotted line denote the control model (CM) before and after fitting, respectively.

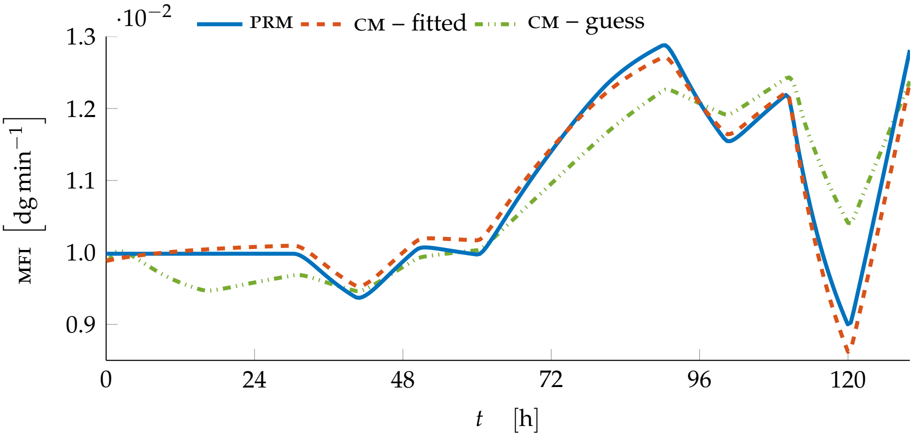


FIGURE 3.5 – The validation of the melt flow index (MFI). The blue line is the plant replacement model (PRM) while the red dashed line and the green dash-dotted line denote the control model (CM) before and after fitting, respectively.

Chapter 4

Online parameter estimation

The state is to be regarded always as an abstract quantity. Intuitively speaking, the state is the minimum amount of information about the past history of the system which suffices to predict the effect of the past upon the future.

— R. E. Kálmán, 1963[1]

A model will generally not be able to reproduce the actual plant measurements perfectly, hence there will be inconsistencies between the measurements and the measurements predicted by the model. This can, to some extent, be rectified by recursive state estimation techniques, in which the states are updated based on the deviations between the predicted measurements and the actual measurements. Several such estimation

[1] Kálmán, R. E., "Mathematical description of linear dynamical systems," *Journal of the Society for Industrial and Applied Mathematics Series A Control*, vol. 1, no. 2, pp. 152–192, 1963. doi: 10.1137/0301010

techniques are available, most notably the Kalman filter (KF) and its derivatives, the moving horizon estimator and the H_∞ filter [32, 90, 102].

In this thesis an augmented divided difference KF was chosen for online estimation. The filter was applied to the control model (CM) such that it was able to track the measurements from the plant replacement model (PRM) more precisely. A short review of the theory behind this type of estimator is given in the succeeding section.

4.1 Theory

The conventional Kalman filter was derived for linear state-space models[2], with additive noise in the both the states and the measurements [102]. Extensions of the filter to nonlinear models have also been developed, such as the extended Kalman filter (EKF), the unscented Kalman filter (UKF) and the divided difference filters of Nørgaard *et al.* [75]. The EKF works by linearizing the model by applying a Taylor expansion around the current state estimate, while the UKF propagates an ensemble of points, known as sigma points, through the nonlinear model equations [102]. The filters of Nørgaard *et al.* [75] are based on polynomial approximations rather than Taylor expansions, and works quite similarly to the UKF, which according to Simon [102], can be considered a special case of these filters.

In Cybernetica's tools, the first-order divided difference (DD1) and second-

[2] http://www.scholarpedia.org/article/State_space_model

order divided difference (DD2)[3] filters of Nørgaard *et al.* [75], which are generalizations of the filter proposed by Schei [89], are implemented. The DD2 filter was chosen for its improved accuracy compared to the DD1 filter. In addition, the filtered estimate is constrained by the applying the general projection approach to avoid violation of state constraints (see Simon [102, p. 216] or Kolås *et al.* [54, sec. 5]).

The applied noise modeling concept is similar to method 3 by Kolås *et al.* [55]. The noise is assumed to enter the model through a set of parameters, which ensures that the balance equations, e.g., mass, mole and energy, are not violated [90]. In addition, the parameters themselves are estimated and are allowed to vary with time, thus the state vector is augmented with the estimated parameters, which leads to the augmented KF. The parameters are thus modeled as integrated white noise; the model equations can be written as [54, 55, 102]:

$$\begin{aligned}
\mathbf{x}_k^- &= \mathbf{F}\left(\mathbf{x}_{k-1}^+, \mathbf{u}_{k-1}, \tilde{\boldsymbol{\theta}}_{k-1}\right) + \boldsymbol{\nu}_{\mathrm{I},k-1} && \textit{a priori} \text{ state estimate} \quad (4.1a)\\
\tilde{\boldsymbol{\theta}}_{k-1} &= \boldsymbol{\theta}_{k-1}^+ \left(1 + \boldsymbol{\nu}_{k-1}\right) && \text{process noise via parameters} \quad (4.1b)\\
\boldsymbol{\theta}_k^- &= \boldsymbol{\theta}_{k-1}^+ \left(1 + \boldsymbol{\upsilon}_{k-1}\right) && \textit{a priori} \text{ parameter estimate} \quad (4.1c)\\
\mathbf{y}_k^- &= \mathbf{g}\left(\mathbf{x}_k^-, \mathbf{u}_k, \boldsymbol{\theta}_k^-\right) + \boldsymbol{\omega}_k && \text{measurement estimate} \quad (4.1d)
\end{aligned}$$

$\mathbf{F}$ is given in eq. (4.2) and the superscripts $+$ and $-$ denote *a posteriori* and *a priori* estimates, respectively. Equation (4.1) is evaluated after the measurements from the previous sample $(k-1)$ has been processed

[3] Not to be confounded with the finite difference methods for numerical differentiation.

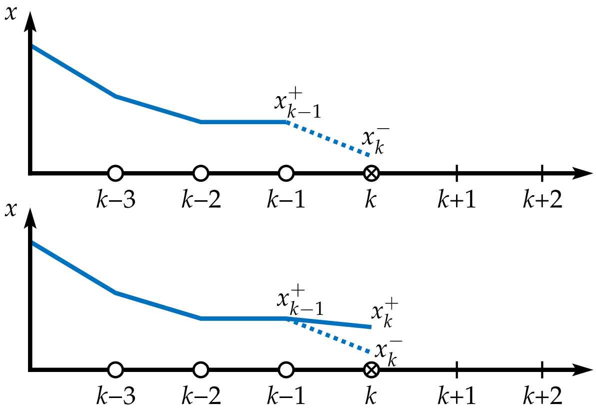


Figure 4.1 – Illustration of the Kalman filter (KF). x is the state, k is the sample number and the superscripts $-$ and $+$ denote the *a priori* and the *a posteriori* estimates, respectively. At each sample, the *a priori* estimate is obtained before the measurements are processed (top figure). After the measurements have been processed, the state is updated to yield the *a posteriori* estimate (bottom figure).

and yields the *a priori* estimates for the current sample (k). When the measurements from the current sample has been processed, eq. (4.4) is evaluated, and the *a posteriori* estimates are obtained. This process is illustrated in Figure 4.1. $\tilde{\boldsymbol{\theta}}$ is a supplementary variable through which the process noise is added. Due to estimation errors in the pressure, level and temperature, the integrated errors for the controllers may differ in the model compared to the real process. To remedy this, additive noise is assumed for the integral errors through the noise term $\boldsymbol{\nu}_{\mathrm{I}}$. The noise of the parameters is assumed to be multiplicative, while the measurement noise is purely additive. The noise is assumed to be uncorrelated and normal distributed, and is given in eq. (4.3).

The equation for the state estimate, given the previous estimate, is the previous state estimate plus the integral of the state derivatives from the previous sample to the current sample:

$$\mathbf{F}\left(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \boldsymbol{\theta}_{k-1}\right) = \mathbf{x}_{k-1} + \int_{t_{k-1}}^{t_k} \mathbf{f}\left(\mathbf{x}(\tau), \mathbf{u}(\tau), \boldsymbol{\theta}_{k-1}\right) \mathrm{d}\tau \qquad (4.2)$$

The noise of the process, parameters and measurements are assumed to be uncorrelated, i.e., their covariance is zero, and can be written as:

$$\boldsymbol{\nu}_k \sim \mathcal{N}(0, \mathbf{R}_{\boldsymbol{\nu}}) \qquad \text{Gaussian process noise} \qquad (4.3a)$$

$$\boldsymbol{\nu}_{\mathrm{I},k} \sim \mathcal{N}(0, \mathbf{R}_{\boldsymbol{\nu}_{\mathrm{I}}}) \qquad \begin{array}{l}\text{Gaussian process noise}\\ \text{on controller integrals}\end{array} \qquad (4.3b)$$

$$\boldsymbol{\upsilon}_k \sim \mathcal{N}(0, \mathbf{R}_{\boldsymbol{\upsilon}}) \qquad \text{Gaussian parameter noise} \qquad (4.3c)$$

$$\boldsymbol{\omega}_k \sim \mathcal{N}(0, \mathbf{Q}) \qquad \text{Gaussian measurement noise} \qquad (4.3d)$$

$\mathcal{N}(0, \sigma^2)$ indicates a normal distributed variable with zero mean and a variance of σ^2. Large variances in the process and parameter noise yield strong state and parameter updates, while large variances in the measurement noise yield weak updates. The standard deviation, σ, of each noise can then be considered a tuning parameter [19]. The assumption of uncorrelated noise yields diagonal covariance matrices; $\mathbf{R}_{\boldsymbol{\nu}}$, $\mathbf{R}_{\boldsymbol{\nu}\mathrm{I}}$, $\mathbf{R}_{\boldsymbol{\upsilon}}$ and $\mathbf{Q}$.

The *a posteriori* estimate of the augmented state containing both the states and the parameters is given by [55, 102]

$$\begin{bmatrix} \mathbf{x}_k^+ \\ \boldsymbol{\theta}_k^+ \end{bmatrix} = \begin{bmatrix} \mathbf{x}_k^- \\ \boldsymbol{\theta}_k^- \end{bmatrix} + \mathbf{K}_k \left(\mathbf{y}_k - \mathbf{y}_k^-\right) \qquad \begin{array}{r}\textit{a posteriori}\text{ state estimate}\\ \textit{a posteriori}\text{ parameter estimate}\end{array} \qquad (4.4)$$

TABLE 4.1 – The standard deviations of the measurement noise used in the simulations.

MEASUREMENT	STANDARD DEVIATION OF MEASUREMENT NOISE
Mole fraction of hydrogen, x_{H_2}	$1 \cdot 10^{-4}$
Mole fraction of propylene, $x_{C_{3=}}$	$1 \cdot 10^{-4}$
Mole fraction of ethylene, $x_{C_{2=}}$	$1 \cdot 10^{-4}$
Pressure, p	500 Pa
Level, h	0.1 m
Temperature, T	0.1 K
Purge rate, $\hat{V}_{pu}$	$1 \cdot 10^{-4}\,\mathrm{m^3\,s^{-1}}$
Production rate, $\hat{V}_{pu}$	$5 \cdot 10^{-6}\,\mathrm{m^3\,s^{-1}}$
Temperature of cooling water, T_{cw}	0.1 K
Melt flow index, MFI	$1.0 \cdot 10^{-3}\,\mathrm{dg\,min^{-1}}$

where **K** is the KF gain which is multiplied by the difference between the actual measurements and the estimated measurements. The KF gain is dependent on the covariances of the augmented state, which are estimated both *a priori* and *a posteriori* by the filter proposed by Nørgaard *et al.* [75] and Schei [89].

The compressibility, the heat transfer coefficient and the reaction rate corrections were chosen for online estimation. The filter was tuned manually in ModelFit by adjusting the standard deviations for the parameter noise and the process noise, including the noise on the controller integrals. The standard deviation in the measurement noise was assumed to be known, and is given in Table 4.1. The filter was tuned against a time series of outputs from the PRM. The outputs were generated by simulating with the two inputs given in eq. (4.5). The applied sequence of inputs was: $\mathbf{u}_1$ for 24 h, $\mathbf{u}_2$ for 24 h and then back to $\mathbf{u}_1$ for 24 h. Each of the measurements was sampled at regularly spaced intervals with a sample time of one minute.

$$\mathbf{u}_1 = \Big[1.813 \cdot 10^{-4}\,\mathrm{m^3\,s^{-1}} \quad 0.040\,46\,\mathrm{m^3\,s^{-1}} \quad 0.007\,04\,\mathrm{m^3\,s^{-1}} \quad 4.5 \cdot 10^{-6}\,\mathrm{m^3\,s^{-1}} \quad 0.012\,05\,\mathrm{kg\,s^{-1}} \quad 353.15\,\mathrm{K}\Big]^\mathsf{T} \tag{4.5a}$$

$$\mathbf{u}_2 = \Big[2.614 \cdot 10^{-4}\,\mathrm{m^3\,s^{-1}} \quad 0.042\,03\,\mathrm{m^3\,s^{-1}} \quad 0.004\,611\,\mathrm{m^3\,s^{-1}} \quad 4.5 \cdot 10^{-6}\,\mathrm{m^3\,s^{-1}} \quad 0.0100\,\mathrm{kg\,s^{-1}} \quad 353.15\,\mathrm{K}\Big]^\mathsf{T} \tag{4.5b}$$

The initial standard deviations of the process and the parameter noise were tuned by trial-and-error until a satisfactory filter was obtained. The philosophy of the tuning process was that the parameters should vary smoothly, and that the CM should be able to track the outputs from the PRM quite accurately. The first can be obtained by setting a low variance for the parameter noise, $\boldsymbol{\upsilon}$, while the latter was obtained by adjusting the variance of the process noise, $\boldsymbol{\nu}$. In addition the variance of the noise on the controller integrals, $\boldsymbol{\nu}_\mathrm{I}$, was increased until the model could track the cooling water (CW) temperature, purge rate and production rate without producing large deviations in temperature, pressure and level. Initially, the filter was tuned by looking at open-loop responses in ModelFit, and later retuned in closed-loop, i.e., with active control. This is explained further in Chapter 5. The results of the tuning process are presented in the next section.

Table 4.2 – The chosen standard deviations of the process and parameter noise for each of the estimated parameters.

parameter	standard deviation of process noise ν	standard deviation of parameter noise υ
Compressibility, Z	$5 \cdot 10^{-4}$	$1 \cdot 10^{-5}$
Heat transfer coefficient, UA	$5 \cdot 10^{-4}$	$5 \cdot 10^{-5}$
Site 1 correction for $C_{3=}$, $\psi^1_{C_{3=}}$	$5 \cdot 10^{-4}$	$2 \cdot 10^{-5}$
Site 2 correction for $C_{3=}$, $\psi^2_{C_{3=}}$	$5 \cdot 10^{-4}$	$2 \cdot 10^{-5}$
Site 1 correction for $C_{2=}$, $\psi^1_{C_{2=}}$	$5 \cdot 10^{-4}$	$2 \cdot 10^{-5}$
Site 2 correction for $C_{2=}$, $\psi^2_{C_{2=}}$	$5 \cdot 10^{-4}$	$2 \cdot 10^{-5}$
Correction for H_2, ψ_{H_2}	$5 \cdot 10^{-4}$	$5 \cdot 10^{-5}$

All the measurements were utilized in the KF despite not being independent; the measurements of the proportional integral (PI) controlled variables (CVs) can for instance not be considered independent of the other measurements at steady-state. However, they are dynamically independent, and it was observed that the filter performed better with these measurements included. Consequently, they are included hereafter in all the simulations with the KF.

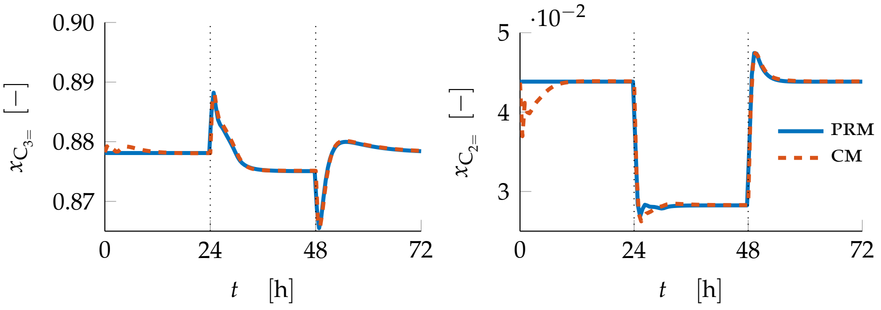
4.2 Results and discussion

The standard deviations for the noise of each of the parameter is displayed in Table 4.2, and the standard deviations for the noise of the controller integrals are presented in Table 4.3. The standard deviation of the multi-

Table 4.3 – The selected standard deviations of the noise on the controller integrals for the level controller (LC), the pressure controller (PC) and the temperature controller (TC).

CONTROLLER INTEGRAL	STANDARD DEVIATION ν_I
Level, $\int_0^t e_h \, d\tau$	$1 \cdot 10^{-2}$ m s
Pressure, $\int_0^t e_p \, d\tau$	$1 \cdot 10^{4}$ Pa s
Temperature, $\int_0^t e_T \, d\tau$	$5 \cdot 10^{-2}$ K s

plicative parameter noise is very small, which yielded smooth parameter variations. This is evident in Figures 4.8 and 4.9. The standard deviation of the process noise between ten and fifty times greater than the parameter noise, which allowed for accurate tracking of the measurements. This can be seen in Figures 4.2 to 4.7. The initial value of each parameter was the value obtained by the offline parameter estimation in Chapter 3.



A – Mole fraction of propylene.

B – Mole fraction of ethylene.

Figure 4.2 – The results with recursive parameter estimation of the mole fractions of the monomers. The blue line is the plant replacement model (PRM) while the red dashed line correspond to the control model (CM). The black dotted lines indicate when the each of the steps were applied.

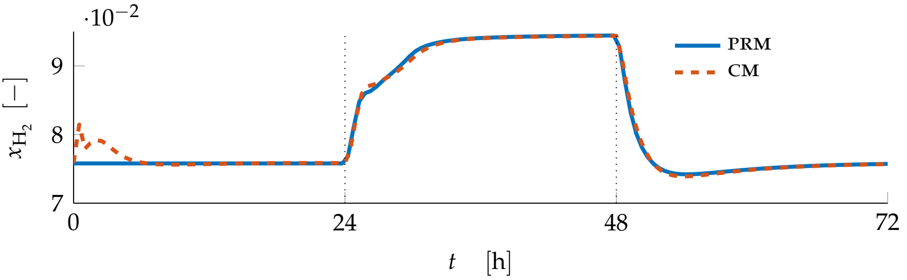


FIGURE 4.3 – The results with recursive parameter estimation of the mole fraction of hydrogen. The blue line is the plant replacement model (PRM) while the red dashed line correspond to the control model (CM). The black dotted lines indicate when the each of the steps were applied.

By inspecting Figures 4.8 and 4.9, rapid changes in the parameters at the beginning of the simulation can be observed. These changes are mainly due to the large measurement offsets at the beginning of the time series. However, after approximately twenty hours the measurement coincide more, and the parameters are stabilizing.

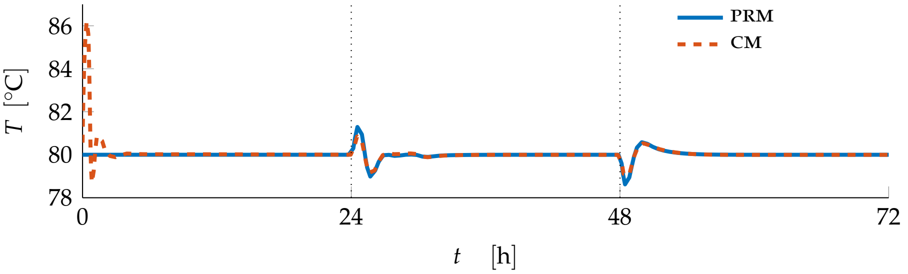
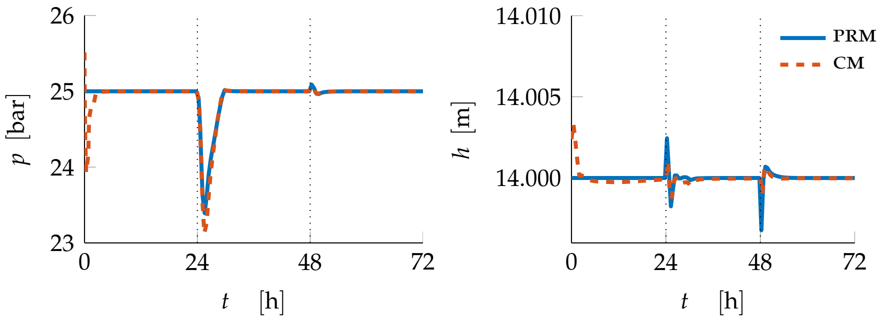


FIGURE 4.4 – The results with recursive parameter estimation for the temperature. The blue line is the plant replacement model (PRM) while the red dashed line correspond to the control model (CM). The black dotted lines indicate when the each of the steps were applied.



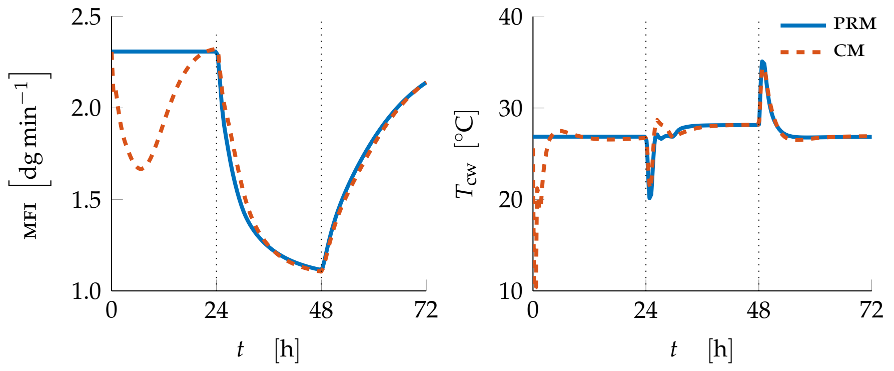
A – Pressure.

B – Level.

Figure 4.5 – The results with recursive parameter estimation for the $p$and level. The blue line is the plant replacement model (PRM) while the red dashed line correspond to the control model (CM). The black dotted lines indicate when the each of the steps were applied.

Figures 4.2 to 4.7 show that, after the initial stabilization, the CM is able to track the outputs from the PRM dynamically. Right after each step in the inputs, marked by the black dotted lines, large changes in the outputs occur. However, the outputs predicted by the CM coincide well with the outputs from the PRM.

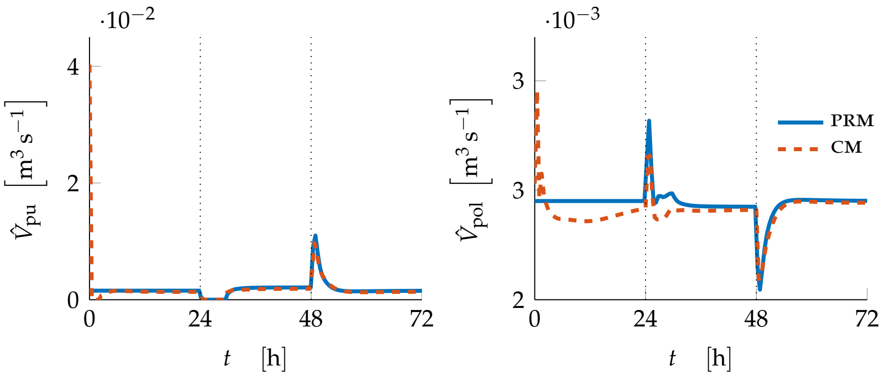
In Figures 4.8 and 4.9 it can be observed that the largest changes in the parameters occur right after each step change. This is probably due to an increase in the difference between the outputs of the two models, nevertheless, the modifications are relatively small. The fact that the measurements from the models are coinciding, and that the parameters are only experiencing minor changes, is an indication that the CM is able to capture the dynamic behavior of the PRM. Thus, the CM is a good approximation of the PRM, despite being structurally different.



a Melt flow index.

b – Temperature of the cooling water.

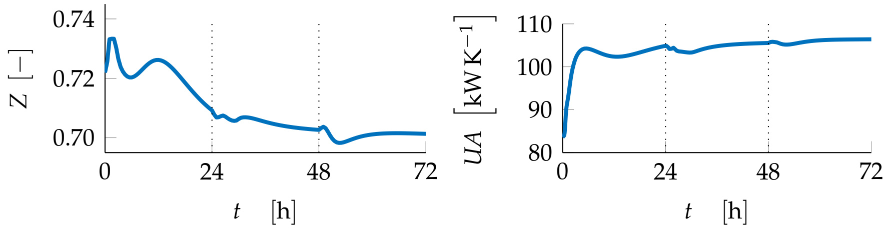
Figure 4.6 – The results of the recursive parameter estimation for the melt flow index (MFI) and the temperature of the cooling water (CW). The blue line is the plant replacement model (PRM) while the red dashed line correspond to the control model (CM). The black dotted lines indicate when the each of the steps were applied.



a – Purge rate.

b – Production rate.

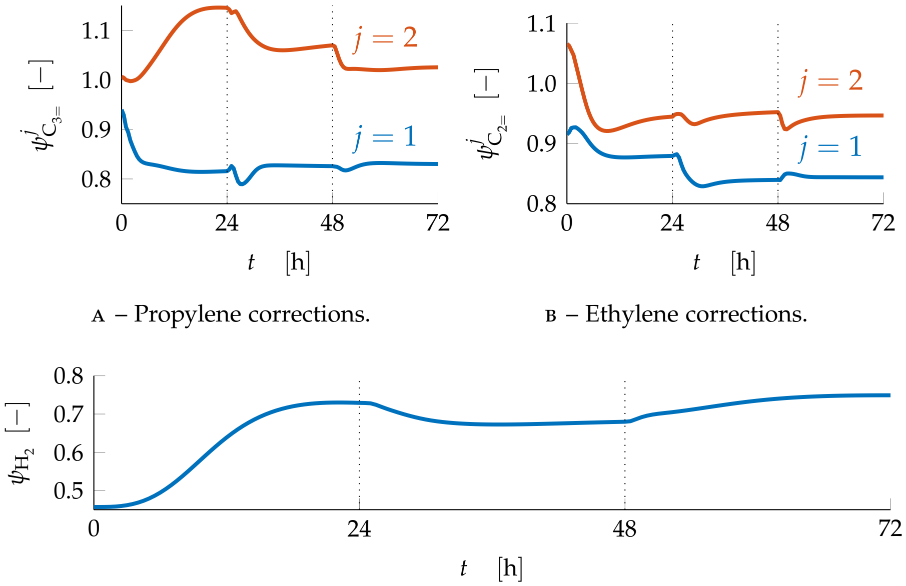
Figure 4.7 – The results of the recursive parameter estimation for the purge and production rates. The blue line is the plant replacement model (PRM) while the red dashed line correspond to the control model (CM). The black dotted lines indicate when the each of the steps were applied.



A – Compressibility.

B – Heat transfer coefficient.

Figure 4.8 – The results of the recursive parameter estimation for the compressibility and the heat transfer coefficient. The black dotted lines indicate when the each of the steps were applied.



A – Propylene corrections.

B – Ethylene corrections.

C – Hydrogen correction.

Figure 4.9 – The results of the recursive parameter estimation for the reaction rate corrections. The black dotted lines indicate when the each of the steps were applied.

Chapter 5

Nonlinear model predictive control

> The basic concept of MPC is to use a dynamic model to forecast system behavior, and optimize the forecast to produce the best decision – the control move at the current time.
>
> — J. B. Rawlings & D. Q. Mayne, 2015[1]

Model predictive control (MPC) is a set of control methods that obtains the controller signals by minimizing an objective function [11]. The objective function usually contains the future set point offsets for the controlled variables (CVs) and the future controller moves, i.e., the increments of the manipulated variables (MVs). The future outputs are predicted by a

[1] Rawlings, J. B. and Mayne, D. Q., *Model Predictive Control*, 5th ed. Madison, Wisconsin: Nob Hill Publishing, 2015, ISBN: 9780975937709 (print). [Online]. Available: http://jbrwww.che.wisc.edu/home/jbraw/mpc/electronic-book.pdf (visited on May 14, 2015)

model of the process in question, hence the term model predictive. In this thesis, the term MPC will be used for receding horizon control (RHC); the basic concept of RHC is to obtain the optimal inputs for the selected horizon, but only apply the inputs at the current sample. At each sample, the optimization problem is re-solved and a new set of inputs is applied. Since the length of the horizon is fixed it recedes as time proceeds, which has given this method its name [58].

The phrase nonlinear model predictive control (NMPC) is usually reserved for applications where the process model is nonlinear, not the optimization problem [12]. Nonlinear model predictive control has been applied in numerous applications for processes where the nonlinear dynamics are prominent, such as polymer production, power plant start-up, metal refining, aluminum electrolysis and fluidized catalytic cracking [4, 12, 32, 33, 71, 72, 86]; a survey of the available industrial MPC technologies has been performed by Qin and Badgwell [82].

In this thesis, Cybernetica's tool for NMPC, CENIT,[2] has been utilized in conjunction with their tool for simulation, RealSim. RealSim and CENIT communicate through open platform communcations (OPC) server; the setup of OPC, CENIT and RealSim is further explained in Appendix G. A diagram of the control scheme is displayed in Figure 5.1. The controller provides the inputs to the process and the model; the estimator updates the state and the parameters of the model, based on the difference between the measurements from the process and the measurements predicted by

[2] http://www.cybernetica.biz/v3/products/CENIT/index.html

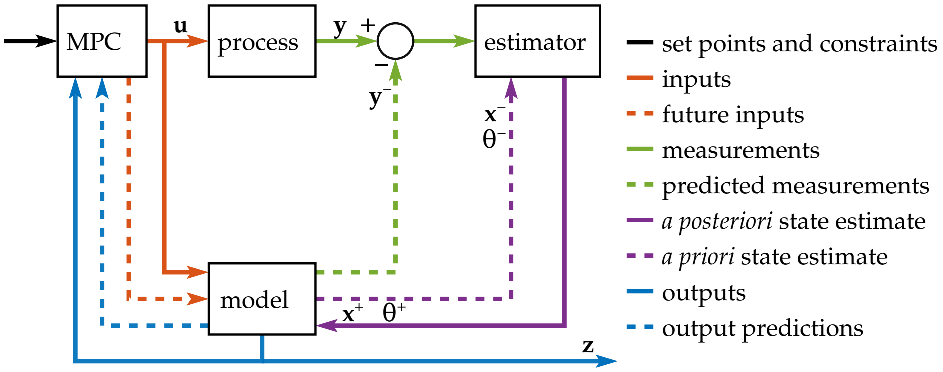

MPC
u
process
y
+
−
y⁻
estimator
x⁻
θ⁻
model
x⁺ θ⁺
z
— set points and constraints
— inputs
- - future inputs
— measurements
- - predicted measurements
— a posteriori state estimate
- - a priori state estimate
— outputs
- - output predictions

FIGURE 5.1 – A diagram of the model predictive control (MPC) scheme. The controller calculates the optimal sequence of future inputs by optimizing the response of the outputs from the model, and applies the inputs to the process at each sample. The estimator updates the states and the parameters of the model by utilizing the difference between the measurements and the predicted measurements.

the model. CENIT utilizes a sequential quadratic programming (SQP) algorithm to obtain the optimal controller action at each sample, in concurrence with an estimation technique for online state and parameter estimation, e.g., Kalman filter (KF). The implemented control algorithm is based on the works of Li and Biegler [59], Li, Biegler, *et al.* [60], and de Oliveira and Biegler [76, 77]. In the succeeding section, a brief review of the theory behind the principles of NMPC will be presented.

5.1 THEORY

A conceptual illustration of the RHC formulation is given in Figure 5.2. The prediction horizon gives the number of samples for which the outputs are predicted, while the control horizon determines the sample after which the inputs are held constant for the remainder of the prediction horizon, i.e., the control increments are zero beyond the control horizon. In addition, the inputs are usually held constant for several samples to reduce the number of optimization variables. This is known as blocking, and this particular blocking strategy is known as input blocking; more advanced blocking strategies have been developed (see Cagienard *et al.* [10]), but input blocking is usually sufficient for most applications. The input blocking is illustrated in Figure 5.2, in which the number of input moves has been reduced by blocking the input move at sample number $k+3$. The snapshot of the prediction and control horizon in Figure 5.5 is also depicting this.

The set point offsets of the outputs are included in the cost function, which is minimized by adjusting the increments of the inputs as illustrated in Figure 5.2. For slow or delayed outputs, the first samples after the current sample is usually omitted, this is shown for sample number $k+1$ in Figure 5.2. This is due to the large offset after set point changes which may lead to an overly aggressive controller [11]. Additionally, for outputs with slow dynamics, it is not necessary to include every sample in the cost function. This is due to the fact that the offset will be almost identical for two neighboring samples. The subset of the samples in the prediction horizon that is included, is known as coincidence points [82]; the coincidence points are marked with blue circles in Figure 5.2.

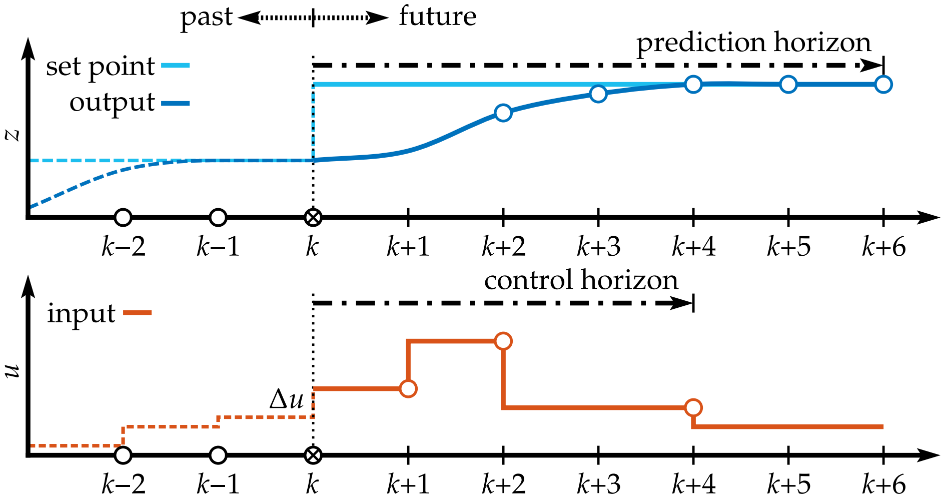


Figure 5.2 – Conceptual illustration of model predictive control (MPC). z is the controlled variable (CV) (top figure) and u is the manipulated variable (MV) (bottom figure), while the sample number is denoted by k. The blue circles denote where the output is weighted, and the orange circles indicate the samples where the input is changed. Adapted from Bemporad and Morari [5] and Findeisen, Imsland, *et al.* [28].

5.1.1 Optimization formulation

The optimization problem which is solved at each sample in CENIT is given in eq. (5.1). The solution of the optimization problem is the sequence of future inputs that yields the optimal response of the outputs [11]. The number of SQP iterations is adjustable, and may be less than the number required to reach convergence. However, the approximate solution at the previous sample is employed as the starting value for the next sample, thus the approximate solution will often become better at each sample [83]. This is due to the fact that the optimum will probably not differ significantly between subsequent samples [60].

$$
\begin{aligned}
\min_{\Delta\mathbf{u}} \quad & J\left(\Delta\mathbf{u}\right) \\
\text{subject to} \quad & \mathbf{x}_{k+j} = \mathbf{F}\left(\mathbf{x}_{k+j-1}, \mathbf{u}_{k+j-1}\right), \quad \forall j \in \left\{1, \ldots, np\right\} \\
& \mathbf{z}_{k+j} = \mathbf{h}\left(\mathbf{x}_{k+j}, \mathbf{u}_{k+j}\right) \\
& \mathbf{u}_{\text{min}} \leq \mathbf{u} \leq \mathbf{u}_{\text{max}} \\
& \Delta\mathbf{u}_{\text{min}} \leq \Delta\mathbf{u} \leq \Delta\mathbf{u}_{\text{max}} \\
& \mathbf{z}_{\text{min}} - \boldsymbol{\varepsilon} \leq \mathbf{z} \leq \mathbf{z}_{\text{max}} + \boldsymbol{\varepsilon} \\
& \mathbf{0} \leq \boldsymbol{\varepsilon} \leq \boldsymbol{\varepsilon}_{\text{max}}
\end{aligned}
\tag{5.1}
$$

$\Delta\mathbf{u}$ is a block vector[3] that contains the sequence of increments for all the MVs. The cost function, J, is presented in eq. (5.3); the first two constraints in eq. (5.1) are the model equations, which has to be satisfied for all the samples in the prediction horizon. k is the current sample number, while np is the number of samples in the prediction horizon. The function that propagates the state to the next sample, $\mathbf{F}$, is defined in eq. (4.2) in Chapter 4. The output vector, $\mathbf{z}$, contains the measurements from eq. (2.69) in addition to the outputs from eq. (2.73) from Chapter 2.

The constraints imposed on the inputs are usually physical limits, e.g., the feed rates must positive and less than the maximum value allowed by the valve [62]. Constraints have also been imposed to the input moves, $\Delta\mathbf{u}$, to ensure that the controller does not change the inputs faster than the physical limits of the actuators. However, tighter constraints than

[3] A block vector/matrix is a vector/matrix that is composed of smaller vectors/matrices.

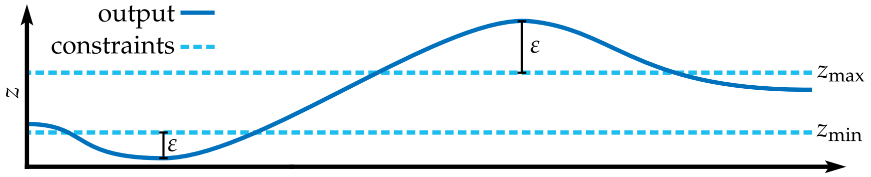


Figure 5.3 – An illustration of the slack variable that is used for the soft output constraints. z is the output, and ε is the slack variable.

the physical limits may also be applied to increase the smoothness of the input moves [106].

Soft constraints have been imposed on the outputs through the slack variable, $\boldsymbol{\varepsilon}$. The slack variable is a block vector with the all the constrained outputs for all the coincidence points, and it is defined in eq. (5.2). An illustration of the slack variable is given in Figure 5.3. The constraint enforced on the slack variable, $\boldsymbol{\varepsilon}_{\max}$, is usually set to infinity or a very large value to avoid infeasibility, i.e., an ill-posed optimization problem [48, 87].

$$\boldsymbol{\varepsilon}_k = \max\left(\mathbf{z}_k - \mathbf{z}_{\max},\, \mathbf{z}_{\min} - \mathbf{z}_k,\, 0\right) \tag{5.2}$$

The cost function implemented in CENIT is given below, and it includes four different terms. The first term is the quadratic set point offset, which becomes smaller the closer the outputs are to the desired set point. In order to penalize the controller moves, a quadratic cost is added to the increments of the inputs in the second term. The last two terms handles the soft output constraints imposed on the outputs via the slack variables; both a quadratic and a linear cost is included to minimize the slack variables. The linear cost is added because it can be shown that this may lead to the exact penalty method. In this method, the output constraints are not violated unless no other feasible solution exists [62]. Additionally, it enables a ranking between the different constraints [77]; the exact penalty method has, however, not been considered in this thesis.

$$J\left(\Delta\mathbf{u}\right) = \frac{1}{2}\left(\mathbf{z}^{\star} - \mathbf{z}_{\mathrm{sp}}^{\star}\right)^{\mathsf{T}}\mathbf{Q}\left(\mathbf{z}^{\star} - \mathbf{z}_{\mathrm{sp}}^{\star}\right) + \frac{1}{2}\left(\Delta\mathbf{u}^{\star}\right)^{\mathsf{T}}\mathbf{S}\left(\Delta\mathbf{u}^{\star}\right) + \frac{1}{2}\left(\boldsymbol{\varepsilon}^{\star}\right)^{\mathsf{T}}\mathbf{R}\left(\boldsymbol{\varepsilon}^{\star}\right) + \mathbf{r}^{\mathsf{T}}\boldsymbol{\varepsilon}^{\star} \tag{5.3}$$

The superscript $\star$ is used to denote dimensionless variables. The definition of the dimensionless variables are given in eq. (5.4). $\mathbf{Q}$ is the block matrix that contain the weight of each output at each coincidence point. The weights of all the input moves are contained in $\mathbf{S}$ for all the samples where the inputs are incremented. $\mathbf{R}$ includes the weights for the slack variables at each coincidence point for the quadratic constraint violation penalty, while $\mathbf{r}$ contains the weight for the linear constraint violation penalty. According to Qin and Badgwell [82], the weights are commonly constant throughout the whole prediction horizon; the weight matrices can thus be obtained by eq. (5.5). It is worth noting that the cost function is only dependent on the input moves due to the fact that both the outputs and the slack variables are implicitly given by the input moves via the model equations.

$$\mathbf{z}_k^{\star} = \mathbf{Z}^{-1}\mathbf{z}_k \tag{5.4a}$$

$$\mathbf{z}_{\mathrm{sp},k}^{\star} = \mathbf{Z}^{-1}\mathbf{z}_{\mathrm{sp},k} \tag{5.4b}$$

$$\boldsymbol{\varepsilon}_k^{\star} = \mathbf{Z}^{-1}\boldsymbol{\varepsilon}_k \tag{5.4c}$$

$$\Delta\mathbf{u}_k^\star = \mathbf{U}^{-1}\Delta\mathbf{u}_k \tag{5.4d}$$

Z and **U** are diagonal matrices that contain the span of each output and input, respectively; the span of a variable is defined hereafter as the typical range in which the variable varies. The introduction of the span variables ensures that the outputs, slack variables and input moves are dimensionless, and of the same order of magnitude. This makes the tuning of the weight matrices more intuitive, because the differences in the values of each variable has already been accounted for; the weight matrices will then better reflect the weight of each individual variable. The span of each of the variables is given in Section 5.1.4.

The weight matrices are

$$\mathbf{Q} = \mathbf{Q}_k \otimes \mathbf{I}_{nz} \tag{5.5a}$$

$$\mathbf{S} = \mathbf{S}_k \otimes \mathbf{I}_{nu} \tag{5.5b}$$

$$\mathbf{R} = \mathbf{R}_k \otimes \mathbf{I}_{nz} \tag{5.5c}$$

$$\mathbf{r} = \mathbf{r}_k \otimes \mathbf{I}_{nz} \tag{5.5d}$$

where $\otimes$ denotes the Kronecker product,[4] while $\mathbf{I}_n$ denotes the identity matrix[5] of size n; nu is the number of input moves in the control horizon, and nz is the number of coincidence points in the prediction horizon. The weight matrices at sample number k are diagonal matrices that contain the weight of each of their respective variables.

[4] http://mathworld.wolfram.com/KroneckerProduct.html

[5] http://mathworld.wolfram.com/IdentityMatrix.html

Equation (5.1) describes a nonlinear optimization problem that is solved in CENIT by an SQP algorithm. This algorithm involves linearization of the problem and sequentially solving quadratic problems [74]; the subproblems are solved by a line-search method as described by Li and Biegler [59]. Both the outputs and the slack variables are linearized with respect to the input moves to yield a quadratic problem; the linearizations are performed numerically by a central difference approximation at each sample in the prediction horizon along the predicted trajectory. For more details on the specific control algorithms, including constraint handling, see Li and Biegler [59], Li, Biegler, *et al.* [60], and de Oliveira and Biegler [76, 77].

5.1.2 Model predictive control of polyolefin reactors

For processes that operate over a narrow range of operating conditions, linear MPC is usually sufficient [1]. To the contrary, processes that exhibit highly nonlinear behavior or change operating conditions frequently, may require NMPC for acceptable control [6]. Nonlinear model predictive control have been applied to multiple polymerization processes by several authors with success, see for instance [1, 6, 8, 72, 91, 93, 107].

Eliçabe and Meira [24] described the highly nonlinear dynamics in polymer production processes, which stem from the mass and energy balances. These nonlinearities are part of why NMPC have been applied to polymerization reactors; most polymer grades are defined by properties such as chain length distribution, average molecular mass, comonomer composition and density which seldom are available for online measurement [1, 26, 65, 79]. Hence, estimation of these variables are required if they are to be controlled.

Customers of polymers have an increasing demand for diversified products, hence polymer producers must be able to meet these demands by producing several polymer grades [107]. The transition between these grades is therefore crucial for economical operation of a polymer-producing plant. Optimization of the grade transition problem have been researched extensively over the last decades [16, 21, 53, 63, 85], and several strategies for optimal grade transitions have been proposed (see Debling *et al.* [21] for an overview.). This has however, not been in the scope of this thesis; instead, the focus has been on set point tracking of the CVs.

The MVs were initially chosen as the feeds of hydrogen, monomers and catalyst, with the inert feed as a measured disturbance and a constant temperature set point. However, this proved to be insufficient to keep the pressure within the desired constraints given in Section 5.1.3. To mitigate this issue, the feed of nitrogen was also employed as an MV; a comparison between the pressure control with and without the extra input is presented Section 5.3.1. The inputs selected for the simulations are summarized in Table 5.1.

Initially, the concentrations of the gas phase were controlled in addition to the purge and production rates. The purpose of this was to test if the NMPC was able to track set point changes of the gas concentrations. The

TABLE 5.1 – The manipulated variables (MVs) and the controlled variables (CVs) used in the simulations.

MV	CV
Hydrogen feed, $\hat{V}_{\mathrm{H_2,f}}$	Melt flow index, MFI
Propylene feed, $\hat{V}_{\mathrm{C_{3=},f}}$	Polymer composition, x_{pol}
Ethylene feed, $\hat{V}_{\mathrm{C_{2=},f}}$	Purge rate, $\hat{m}_{\mathrm{pu}}$
Catalyst feed, $\hat{m}_{\mathrm{cat,f}}$	Production rate, $\hat{m}_{\mathrm{pol}}$
Nitrogen feed, $\hat{V}_{\mathrm{N_2,f}}$	Nitrogen feed, $\hat{V}_{\mathrm{N_2,f}}$

reason for this approach is that the MVs have a more direct impact on the gas concentrations than on the quality properties, e.g., melt flow index (MFI), hence they are easier to control. After verifying that the controller was able to track the set point changes of the concentrations, the CVs were chosen to be the MFI, the polymer composition, the purge rate and the production rate. Furthermore, with the extra input, i.e., the nitrogen feed, an additional output was chosen, namely the nitrogen feed itself. This creates an additional degree of freedom for the controller, and by setting the set point of the nitrogen feed to zero, the extra degree of freedom will only impact the dynamic responses and leave the steady-state values unaffected. The outputs chosen for the simulations are summarized in Table 5.1.

5.1.3 Applied constraints

Hard constraints have been imposed on the MVs and the input moves. These constraints represent the physical limitation of the actuators, e.g., the flow manipulators cannot open the valves more than 100 % or less

Table 5.2 – The hard constraints imposed on the manipulated variables (mvs). $\mathbf{u}$ contain the input variables, which has constraints on their maximum ($\mathbf{u}_{\text{max}}$) and minimum values ($\mathbf{u}_{\text{min}}$) in addition to their rate of change ($\Delta\mathbf{u}_{\text{max}}$).

input	maximum	minimum	increment
$\mathbf{u}$	$\mathbf{u}_{\text{max}}$	$\mathbf{u}_{\text{min}}$	$\Delta\mathbf{u}_{\text{max}}$
Hydrogen feed, $\hat{V}_{H_2,f}$	$0.01\,\text{m}^3\,\text{s}^{-1}$	$0\,\text{m}^3\,\text{s}^{-1}$	$0.0001\,\text{m}^3\,\text{s}^{-1}$
Propylene feed, $\hat{V}_{C_{3=},f}$	$0.1\,\text{m}^3\,\text{s}^{-1}$	$0\,\text{m}^3\,\text{s}^{-1}$	$0.001\,\text{m}^3\,\text{s}^{-1}$
Ethylene feed, $\hat{V}_{C_{2=},f}$	$0.1\,\text{m}^3\,\text{s}^{-1}$	$0\,\text{m}^3\,\text{s}^{-1}$	$0.001\,\text{m}^3\,\text{s}^{-1}$
Nitrogen feed, $\hat{V}_{N_2,f}$	$0.1\,\text{m}^3\,\text{s}^{-1}$	$4.5 \cdot 10^{-6}\,\text{m}^3\,\text{s}^{-1}$	$0.01\,\text{m}^3\,\text{s}^{-1}$
Catalyst feed, $\hat{m}_{\text{cat},f}$	$0.1\,\text{kg}\,\text{s}^{-1}$	$0\,\text{kg}\,\text{s}^{-1}$	$0.001\,\text{kg}\,\text{s}^{-1}$
Temperature set point, T_{sp}	90 °C	60 °C	0.5 °C

than 0 %. However, the constraints implemented on the temperature set point are not physical limits, but can be considered bounds that ensure a safe and efficient operation, i.e., to avoid runaway reactions and at the same time achieve high conversion rates. The constraints of the inputs chosen in the simulations are recited in Table 5.2. The constraints on the input increments are equal for both positive and negative increments, i.e., $|\Delta\mathbf{u}| \leq \Delta\mathbf{u}_{\text{max}}$. The minimum value of the nitrogen feed is nonzero to model the trace amounts of inert gases present in the other feed flows.

Soft constraints have been utilized for two of the output variables, the pressure and the temperature. These variables are controlled with proportional integral (pi) controllers, however, the simple controllers were not able to keep the cvs inside the specified limits. Both pressure control and temperature control are essential in polyolefin production, to ensure safe operations and correct product specifications [93, 107]. Consequently,

Table 5.3 – The soft constraints for pressure and temperature.

variable	maximum	minimum
Pressure, p	25.5 bar	24.5 bar
Temperature, T	90 °C	60 °C

the constraints were imposed in order to utilize the mvs actively, to help keep the pressure and the temperature inside their respective ranges. The soft constraints are listed in Table 5.3.

Y. Wang *et al.* [107] proposed to apply limits on the instantaneous polymer properties, because large variations in these variables may deteriorate the overall polymer quality, i.e., the polymer resins may become layered with polymer of highly different properties. This has however, not been considered in this thesis.

5.1.4 Span variables

The diagonal span matrices introduced in eq. (5.4) contain the span of each variable, i.e., the range in which the variable varies. As mentioned earlier, these matrices are introduced to make the tuning of the weight matrices more intuitive. By dividing each variable by their respective span, all the variables in eq. (5.3) will be of the same order of magnitude, hence the weight matrices will better reflect the individual weighting of each variable. The span of each of the outputs is presented in Table 5.4, while the span of each of the inputs is given in Table 5.5.

Table 5.4 – The chosen span of each of the controlled variables (cvs) that make up the span matrix in eq. (5.4), i.e., **Z**.

OUTPUT	SPAN
Mole fraction of propylene in polymer, x_{pol}	$1 \cdot 10^{-3}$
Melt flow index, mfi	$1 \cdot 10^{-2}\,\mathrm{dg\,min^{-1}}$
Production rate, $\hat{m}_{\mathrm{pol}}$	$1 \cdot 10^{-2}\,\mathrm{kg\,s^{-1}}$
Purge rate, $\hat{m}_{\mathrm{pu}}$	$5 \cdot 10^{-3}\,\mathrm{kg\,s^{-1}}$
Temperature, T	$5 \cdot 10^{-1}\,\mathrm{K}$
Pressure, p	$1 \cdot 10^{-1}\,\mathrm{bar}$
Nitrogen feed, $\hat{V}_{\mathrm{N_2,f}}$	$1 \cdot 10^{-3}\,\mathrm{m^3\,s^{-1}}$

Table 5.5 – The chosen span of each of the manipulated variables (mvs) that make up the span matrix in eq. (5.4), i.e., **U**.

INPUT	SPAN
Hydrogen feed, $\hat{V}_{\mathrm{H_2,f}}$	$1 \cdot 10^{-4}\,\mathrm{m^3\,s^{-1}}$
Propylene feed, $\hat{V}_{\mathrm{C_{3=},f}}$	$1 \cdot 10^{-3}\,\mathrm{m^3\,s^{-1}}$
Ethylene feed, $\hat{V}_{\mathrm{C_{2=},f}}$	$1 \cdot 10^{-3}\,\mathrm{m^3\,s^{-1}}$
Nitrogen feed, $\hat{V}_{\mathrm{N_2,f}}$	$1 \cdot 10^{-3}\,\mathrm{m^3\,s^{-1}}$
Catalyst feed, $\hat{m}_{\mathrm{cat,f}}$	$1 \cdot 10^{-3}\,\mathrm{kg\,s^{-1}}$

5.2 Tuning

The initial tuning of the controller was obtained by applying the control model (cm) as both the process simulator and as the predictor. This approach has the advantage that there is no mismatch between the predictions and the measurements, hence the focus can be on tuning of the controller instead of ensuring that the predictions are sufficiently accurate. After the initial tuning, the plant replacement model (prm) was used

TABLE 5.6 – The polypropylene grades used in the simulation of the grade transitions.

CV	GRADE A	GRADE B
Melt flow index, MFI	$2.6\,\mathrm{dg\,min^{-1}}$	$2.0\,\mathrm{dg\,min^{-1}}$
Polymer composition, x_{pol}	0.85	0.9
Purge rate, $\hat{m}_{\mathrm{pu}}$	$0.01\,\mathrm{kg\,s^{-1}}$	$0.01\,\mathrm{kg\,s^{-1}}$
Production rate, $\hat{m}_{\mathrm{pu}}$	$2.2\,\mathrm{kg\,s^{-1}}$	$2.2\,\mathrm{kg\,s^{-1}}$
Nitrogen feed, $\hat{V}_{\mathrm{N_2,f}}$	0	0

as the process simulator, and the controller was retuned with mismatch between the model of the controller and the model of the process simulator. The final tuning parameters will be presented in the subsequent subsections.

As previously outlined, the concentrations of the gas phase was initially chosen as the CVS, in addition to the purge and production rates. After verifying that the controller was able track set point changes in the gas concentrations, the outputs were chosen as listed in Table 5.1.

The controller was tuned by simulating a grade transition between two fictitious polypropylene (PP) grades. The specification of the two grades, grade A and grade B, are listed in Table 5.6. Mieras and van Rijn [69] studied a set of different polypropylene grades, and the MFIS were chosen based on their reported values. The purge rate was set to a low value to minimize flaring, while the production rate and the polymer composition were set to arbitrary, albeit realistic values.

5.2.1 Prediction horizon and coincidence points

A long prediction horizon is beneficial because it will allow all the cvs to stabilize within the horizon, even with large set point changes. However, increasing the prediction horizon increases the computational demand, thus choosing a very long prediction horizon may be infeasible in real-time applications. In this thesis a prediction horizon of 540 samples, i.e., nine hours, was found to be a reasonable trade-off.

The coincidence points of the outputs were chosen differently based on their dynamic responses. The pressure, temperature and the purge rate reacted fast, compared to the mfi, polymer composition and production rate, to any changes in the mvs. Based on this observed behavior, the coincidence points of the fastest variables were chosen more densely near the current sample number. The slower variables were, on the other hand, distributed more evenly throughout the prediction horizon. The last coincidence point of all the variables was selected as the prediction horizon itself, in order to deter steady-state offsets. Code snippet 5.1 displays the selection of the equispaced coincidence points for each output together with the selection of the last coincidence point.

5.2.2 Control horizon and input blocking

The control horizon was chosen to be a bit shorter than the prediction horizon, and the parametrization of the inputs, i.e., the samples at which the inputs are incremented, was chosen identically for all the mvs. The selection of the input parametrization is presented in Code snippet 5.2.

CODE SNIPPET 5.1 – The coincidence point selection for each of the controlled variables (CVS) written in C. The matrix containing the 36 coincidence points of each output is defined on line number 2, and line numbers 3-12 fills in the 35 first points at regularly spaced intervals. The last coincidence point is set equal to the prediction horizon on line numbers 14-15.

```c
 1  /* Initialize the prediction matrix */
 2  nrZerosIMat(Zpar, NZ, 36);             // 36 coincidence points, NZ outputs
 3  for (int j = 1; j <= 35; j++)                // 35 first coincidence points
 4  {
 5      Zpar->m[i_z_T][j]       = 5*j;          // Temperature every 5th sample
 6      Zpar->m[i_z_p][j]       = 3*j;             // Pressure every 3rd sample
 7      Zpar->m[i_z_f_m_pu][j]  = 5*j;                // Purge every 3rd sample
 8      Zpar->m[i_z_f_m_pol][j] = 10*j;          // Production every 5th sample
 9      Zpar->m[i_z_MFI][j]     = 15*j;                // MFI every 15th sample
10      Zpar->m[i_z_x_pol][j]   = 15*j; // Polymer composition every 15th sample
11      Zpar->m[i_z_f_V_I_in][j]= 15*j;         // Inert feed every 15th sample
12  }
13  /* Last coincidence for all the outputs is the prediction horizon */
14  for (int i = 1; i <= NZ; i++)
15      Zpar->[i][36]           = 540;      // Prediction horizon is 540 samples
```

Blocking of the inputs should be chosen such that the main characteristics of the optimal trajectory of the inputs are captured with the fewest number of increments; Figure 5.4 illustrates the parametrization of the inputs.

5.2.3 WEIGHT MATRICES OF THE CONTROLLER

The weights of the MVS and the CVS are given in Tables 5.7 and 5.8, respectively. The weights were initially chosen as unity, and adjusted after observing the closed-loop responses. Originally, the controller fed too much nitrogen, but by increasing the weights of the nitrogen feed, the amount of inert fed into the reactor was reduced. Large increments in the catalyst feed yielded large variations in the temperature of the reactor. In

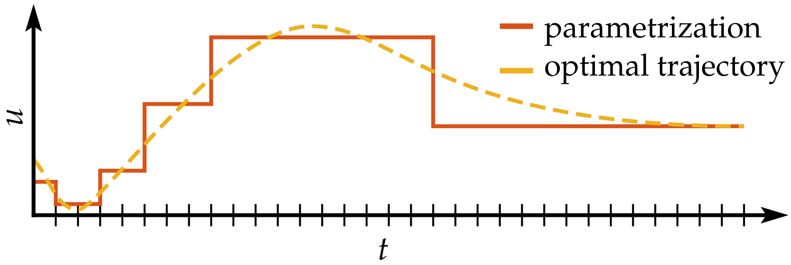


Figure 5.4 – An illustration of the selection of the input parametrization points. The yellow dashed line is the optimal trajectory of the input and the red line is the optimal parametrized input. The points where the input is incremented is select such that the discretized input is able to capture the shape of the optimal continuous input trajectory.

Table 5.7 – The weight of each of the manipulated variables (MVs) that constitutes the matrix **S** from eq. (5.5).

MV	WEIGHT
Hydrogen feed, $\hat{V}_{H_2,f}$	1
Propylene feed, $\hat{V}_{C_{3=},f}$	1
Ethylene feed, $\hat{V}_{C_{2=},f}$	1
Nitrogen feed, $\hat{V}_{N_2,f}$	2
Catalyst feed, $\hat{m}_{cat,f}$	5

Table 5.8 – The weight of each of the controlled variables (CVs) that constitutes the matrix **Q** from eq. (5.5).

CV	WEIGHT
Melt flow index, MFI	1
Polymer composition, x_{pol}	1
Purge rate, $\hat{m}_{pu}$	0.1
Production rate, $\hat{m}_{pol}$	0.1
Nitrogen feed, $\hat{V}_{N_2,f}$	2

CODE SNIPPET 5.2 – The parametrization for each of the manipulated variables (MVS) written in C. Each input has the same parametrization, and the selection of the points at which the MVS is displayed on line numbers 3-14.

```c
1  /* Initialize the input parametrization matrix */
2  nrZerosIMat(Upar, NU, 9);       // 9 parameters for the inputs (input moves)
3  for (int i = 1; i <= NU; i++)                  // Loop through all the inputs
4  {
5      Upar->m[i][1]    = 5;                      // Each input is parameterized
6      Upar->m[i][2]    = 10;                           // with nine parameters,
7      Upar->m[i][3]    = 15;                                // (input blocking)
8      Upar->m[i][4]    = 25;                      // which determine the input at
9      Upar->m[i][5]    = 45;                             // 0, 5, 10, 15, 25, 45
10     Upar->m[i][6]    = 75;                         // 75, 135, 230 and 400
11     Upar->m[i][7]    = 135;              // sampling intervals into the future
12     Upar->m[i][8]    = 230;
13     Upar->m[i][9]    = 400;
14 }
```

TABLE 5.9 – The weight of each of the slack variables for the soft constraints that constitutes the matrix **R** from eq. (5.5).

CONSTRAINED VARIABLE	WEIGHT
Pressure, p	$1 \cdot 10^6$
Temperature, T	$1 \cdot 10^6$

order to minimize these fluctuations in the temperature, the weight of the catalyst feed was increased. To emphasize the importance of the polymer quality outputs, i.e., MFI and polymer composition, the weights of these CVS were set to be significantly greater than the purge and production rates.

The weights of the slack variables for pressure and temperature are listed in Table 5.9, and were chosen to be orders of magnitude greater than the weights of the CVS and MVS to reduce the risk of violating the constraints.

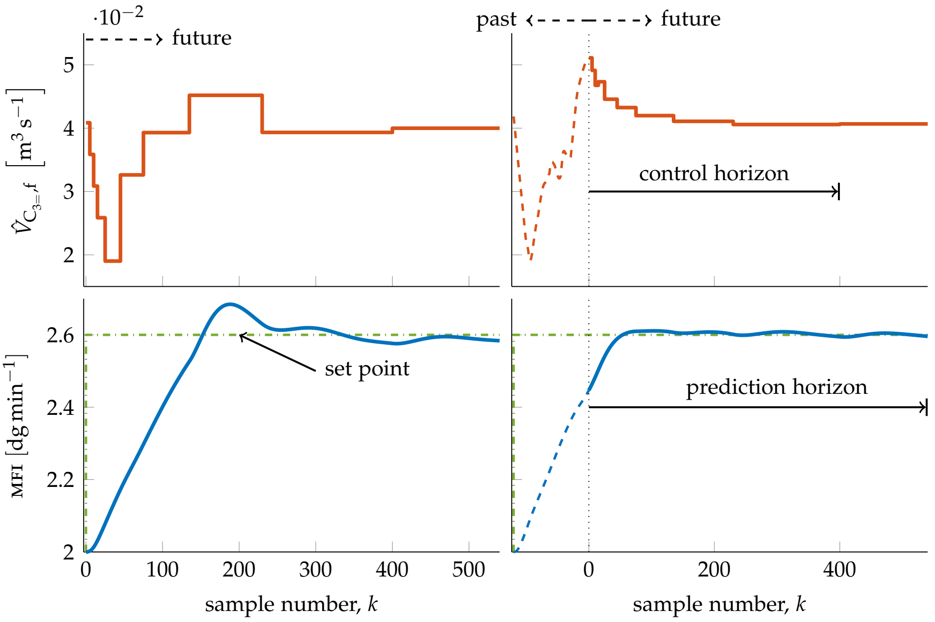


Figure 5.5 – A snapshot of the prediction and control horizons. The graphs on the left depict the predictions just after a set point change from grade B to grade A, while the graphs on the right depict the history and the predictions 120 samples later. The top row portrays the feed of propylene, while the bottom row displays the melt flow index (MFI).

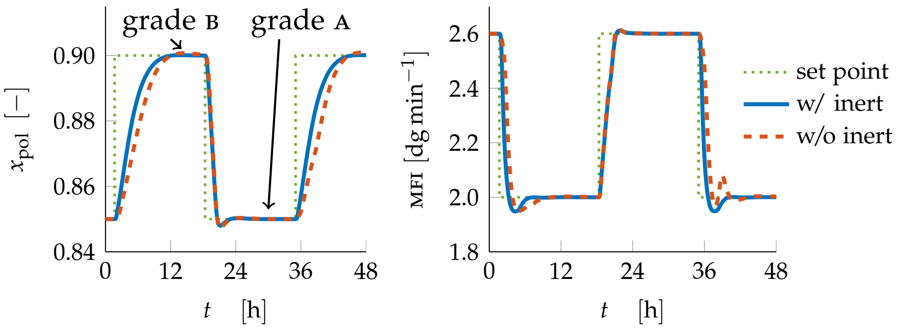
5.3 Results and discussion

Two sets of resulting grade transitions with the selected weights are presented in the succeeding subsections. Firstly, the effect of utilizing the inert feed as an input is elucidated with the CM as the process simulator, i.e., no mismatch of the model used by the controller and the process simulator. Secondly, the outcome of a model mismatch between the model of the simulator and the model of the controller is presented.

Figure 5.5 illustrates the future inputs and the predicted outputs just after a set point change, and two hours, i.e., 120 samples, later. The input blocking, with zeroth order hold, can be observed in the top row which shows the feed of propylene. By inspecting the top row, the input parametrization from Code snippet 5.2 appears to capture the shape of the applied input; i.e., the future input obtained initially, on the left; resembles the history of the applied input, on the right. This is an indication that the choice of input parametrization is acceptable.

By examining the left graph on the bottom row of Figure 5.5, it is discernible that the output is able to reach the set point within the prediction horizon. This suggests that the prediction horizon was chosen to be long enough. The response of the output is also improving with time, which is evident in the graph on the right where there no longer is a predicted overshoot. With this prediction horizon and five inputs, the NMPC used approximately two seconds on the calculations at each sample. This corresponds to thirty times faster than real-time, suitable for an industrial application.

Several issues present in a real PP plant have not been addressed, e.g., variable catalyst activity; input disturbances; impurities that cause deactivation of catalyst, such as carbon monoxide; measurement noise and delayed or offline measurements at irregularly spaced intervals. As previously mentioned, PP plants may consist of multiple reactors in series, and the model would have to be extended to include all the reactors. These problems would have to be resolved before implementation at a plant.



A – Polymer composition.

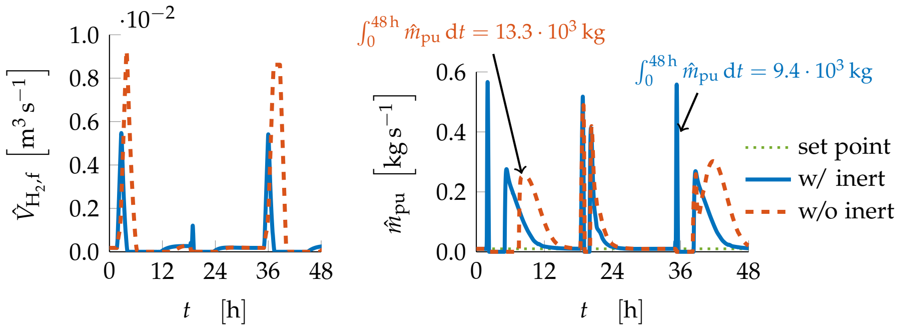
B – Melt flow index.

Figure 5.6 – Polymer composition and melt flow index (MFI) with and without inert feed as an input. The blue line and the dashed red line indicate with and without inert, respectively. The green dotted line denotes the set point.

5.3.1 Inert feed as an input

Figure 5.6 depicts the polymer composition and the MFI responses when transitioning between the two grades in Table 5.6. It can be observed that the transition is slightly faster with the nitrogen feed as an MV. This is probably due to the fact that the controller has to increase the feed of the other gases to keep the pressure within the desired limits, and is evident for the feed of hydrogen in Figure 5.7A. The other gas feeds are very similar for the two cases, and are given in Figure H.3 in Appendix H.

The total purge rate is lower with the inert feed as in input, which can be observed in Figure 5.7B. However, the use of more inert lowers the concentrations of the monomers, which in turn increases the demand of catalyst. Figure 5.9B compares the catalyst feed of the two cases, and it



A – Hydrogen feed.

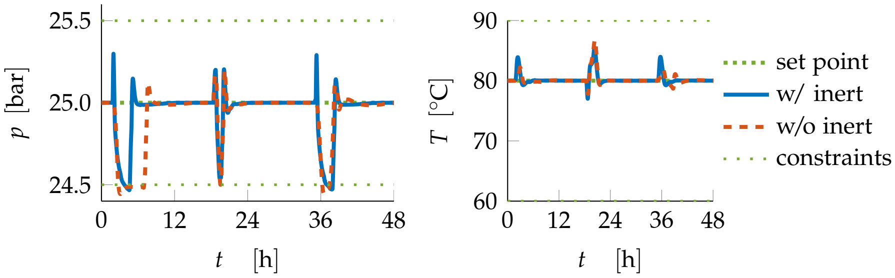
B – Purge rate.

FIGURE 5.7 – The feed of hydrogen and the purge rate with and without the feed of inert as an input. The blue line and the dashed red line indicate with and without inert, respectively.

is clear that more catalyst is required to keep the production rate at the desired set point with inert as an MV.

It is worth noting that increasing the fraction of propylene in the polymer, x_{pol}, is considerably slower than reducing it. This can be discerned in Figure 5.6A, and is due to the fact that ethylene polymerizes faster than propylene, and consequently makes it quicker to incorporate more ethylene in the polymer compared to propylene. The transition from grade A to grade B occurs in approximately eight hours, while the opposite transition occurs in about five hours. These differences in transition time are important when scheduling the transitions between multiple polymer grades.

Figure 5.8 displays the constrained outputs, pressure and temperature. The temperature fluctuates well within its constraints, while the pressure



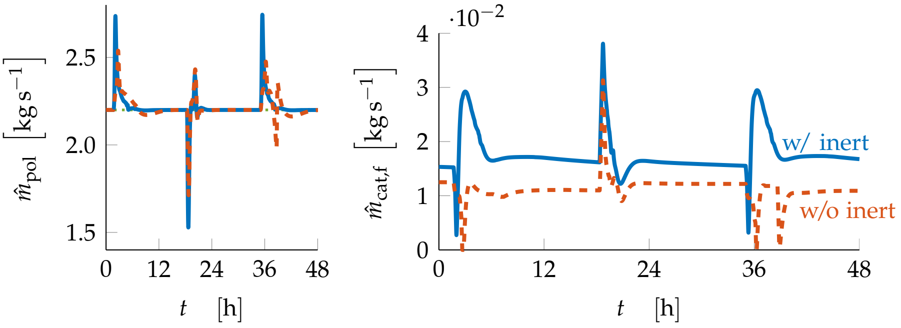
a – Pressure.

b – Temperature.

Figure 5.8 – Pressure and temperature with and without inert feed as an input. The blue line and the dashed red line indicate with and without inert, respectively. The green dotted lines denote the set point and the constraints.

violates its lower constraint when transitioning from grade a to grade b. In Figure 5.8a, it can be observed that the case with the inert as an mv recovers from the violation of the pressure constraint more rapidly than the case without.

The rate of production and the feed of catalyst is depicted in Figure 5.9; as previously mentioned, the case without the inert as an input requires less catalyst which is evident in Figure 5.9b. Thus, the productivity i.e., the ratio between polymer production and catalyst requirement, increases with lower amounts of inert, as expected. Figure 5.9a shows that the production rate experiences larger spikes, both positive and negative, when the inert feed is utilized as an mv. The larger positive spikes are probably due to the increased catalyst feed, while the more negative spike may be due to the lower concentration of monomers; the concentrations are displayed in Figure h.2 in Appendix h.



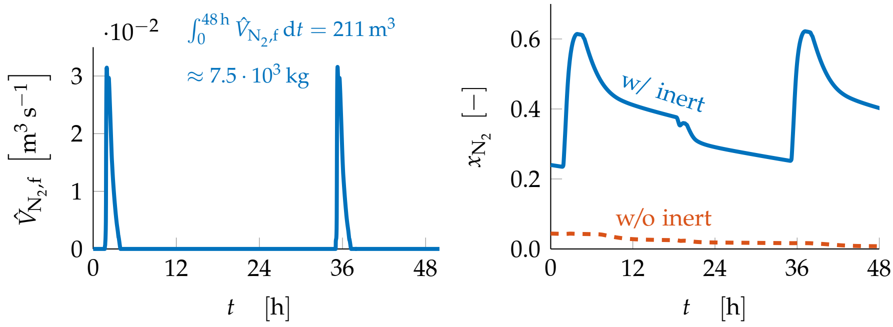
a – Production rate.

b – Catalyst feed.

Figure 5.9 – The feed of catalyst and the production rate with and without the feed of inert as an input. The blue line and the dashed red line indicate with and without inert, respectively.

The feed of nitrogen for the case with the nitrogen feed as an input is given in Figure 5.10a, while the concentration of nitrogen is displayed in Figure 5.10b. Figure 5.10a displays the total amount of nitrogen fed to the system during the simulation, and it as apparent that virtually all the nitrogen is added during the transition from grade a to grade b. The additional inert greatly impacts the concentration of the gaseous compounds, and can be discerned in Figure 5.10b. Due to the low purge rate, the inert stays in the system for an extended amount of time, which results in the increased requirement of catalyst (see Figure 5.9b).

To lower the amount of catalyst used in process, the productivity from eq. (2.79) could have been included in the optimization problem. However, this could have caused an infeasible optimization formulation. Alternatively, a decreased maximum constraint could have been imposed on the



A – Nitrogen feed.

B – Nitrogen mole fraction.

Figure 5.10 – The feed of nitrogen and the mole fraction of nitrogen with and without the nitrogen feed as an input. The blue line and the dashed red line indicate with and without inert, respectively.

nitrogen feed. A third option could have been to choose a greater weight in **Q** for the feed of nitrogen. The previously mentioned alternatives, would in all likelihood produce results that lie in between the two approaches taken in this section. Thus, the pressure control might improve compared to the case without inert feed as an MV, and the amount of catalyst might increase.

To achieve tighter control of pressure and temperature, the pressure controller (PC) and temperature controller (TC) could have been replaced by the NMPC; i.e., the pressure and the temperature could be added as CVs with the purge rate and the cooling water temperature as MVs. However, this would have increased the number of decision variables in the optimization formulation, and was not considered due to the limited time-frame of this work.

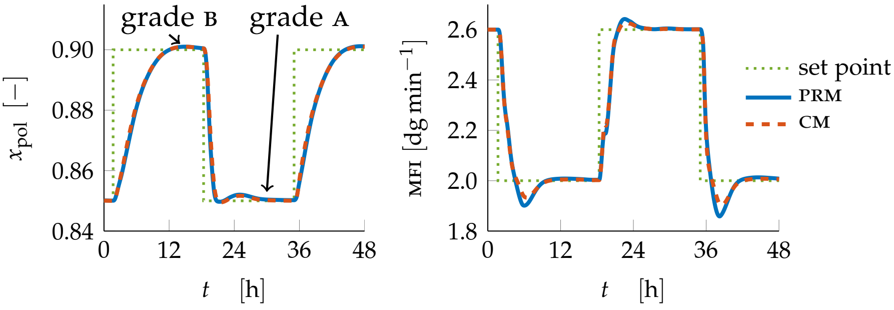
The large spikes in the production rate, which can be observed in Figure 5.9a, could have been mitigated by imposing soft constraints on the rate of production. Nevertheless, for the sake of simplicity, this has not been assessed in this thesis.

5.3.2 Model mismatch

To assess whether nmpc is suitable when there exists structural differences between the model and the plant, i.e., robust, the prm was set as the process simulator. The cm and the prm are structurally different, e.g., one control volume versus four control volumes, and are thus suitable for this assessment. The kf from Chapter 4 was utilized to minimize the deviations between the cm and the prm. The feed of nitrogen was included as an mv in the simulations to avoid problems with the control of the pressure.

Figure 5.11 displays the controlled polymer quality variables for series of grade transitions between grade a and grade b. The polymer composition is not measured, and has thus not been included in the estimator, despite this, the cm is able to accurately reproduce the composition calculated by the prm. This is evident in Figure 5.11a, in which the polymer composition calculated by the two models coincides with almost no observable discrepancies.

The mfi is given in Figure 5.11b, and it can be observed that the overshoots, especially when transitioning from grade a to grade b, are greater compared to the overshoots in Figure 5.6b. These overshoots might have



A – Polymer composition.

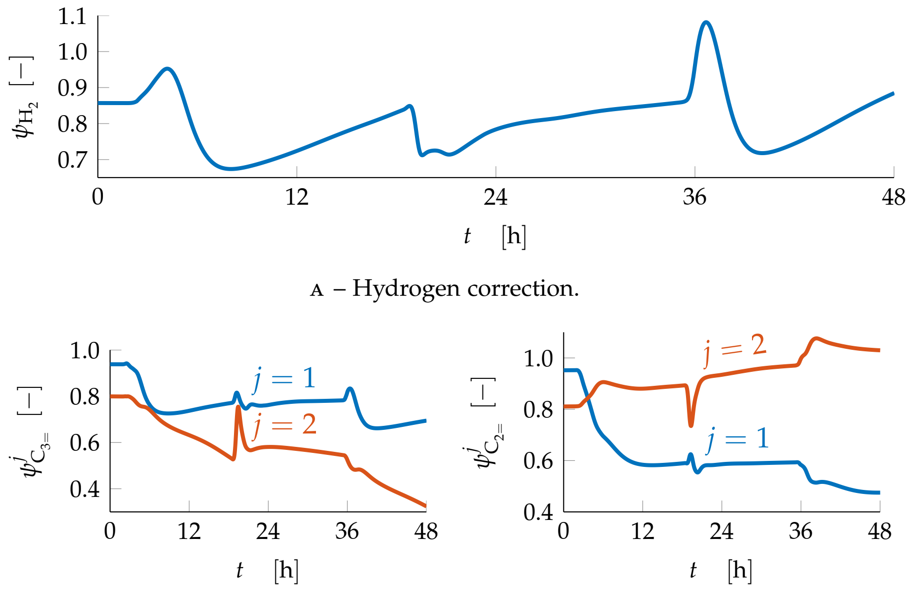
B – Melt flow index.

Figure 5.11 – Polymer composition and melt flow index (MFI) with model mismatch. The blue line and the dashed red line indicate the plant replacement model (PRM) and the control model (CM), respectively. The green dotted line denotes the set point.

been rectified by imposing constraints on the instantaneous MFI, i.e., the MFI of the polymer currently being produced.

The largest discrepancies between the MFI from the PRM, and the estimation by the CM occur at the overshoots. If less overshoot is predicted by CM compared to the PRM, the MFI of the PRM will overshoot and the KF will then adjust the parameters of the CM to minimize the model offset. This seems to have occurred at the transitions from grade A to grade B, and is reinforced by the fact that this coincides with the time at which the largest variations in the reaction rate correction for hydrogen take place (see Figure 5.12A).

Figure 5.12 demonstrates that the reaction rate corrections exhibit much larger variations than in the open-loop simulations from Figure 4.9 in Chapter 4. This is in all likelihood due to the fact that the transitions



A – Hydrogen correction.

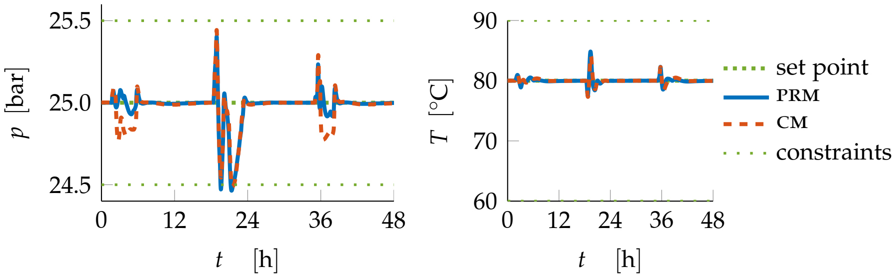
B – Propylene correction.

C – Ethylene correction.

Figure 5.12 – The reaction rate corrections for hydrogen, ethylene and propylene during the grade transitions.

occur much faster, and consequently, the parameters are prone to fluctuate more vigorously to maintain accurate measurement estimations. However, these fluctuations hint at model dissimilarities and the controller could be susceptible to poor predictions. This might have been mitigated by tuning the KF differently, e.g., lowering the standard deviation of the noise of the parameters and increasing the variance of the noise of the process.

The pressure and the temperature are presented in Figures 5.13A and 5.13B, respectively. The temperature of the reactor exhibits a very similar response to the simulations without model mismatch, which can be ascer-



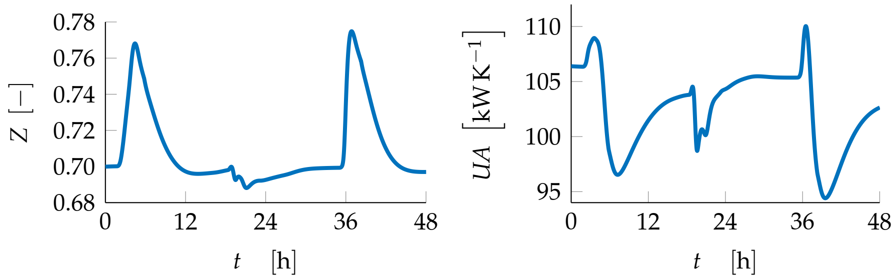
a – Pressure.

b – Temperature.

Figure 5.13 – Pressure and temperature with model mismatch. The blue line and the dashed red line indicate the control model (cm) and the plant replacement model (prm), respectively. The green dotted lines denote the set point and the constraints.

tained by comparing Figures 5.8b and 5.13b. On the contrary, Figures 5.8a and 5.13a evince quite different responses for the pressure. In the case with no model mismatch, the controller struggles most to contain the pressure within the constraints at the transitions from grade a to grade b, while Figure 5.13a displays the opposite behavior, i.e., the controller struggles most when transitioning from grade b to grade a. The reason for this behavior is unclear, and in view of the fact that the pressure is almost contained within the constraints, it has not been explored further.

Figure 5.13a shows minor deviations between the prm and the cm right after initiating the transitions from grade a to grade b. This coincides well with the largest changes in the compressibility, displayed in Figure 5.14a. The standard deviation of the compressibility was chosen to be relatively small, as seen in Table 4.2 in Chapter 4, and a larger variance might have allowed the controller to reproduce the measurements with a higher



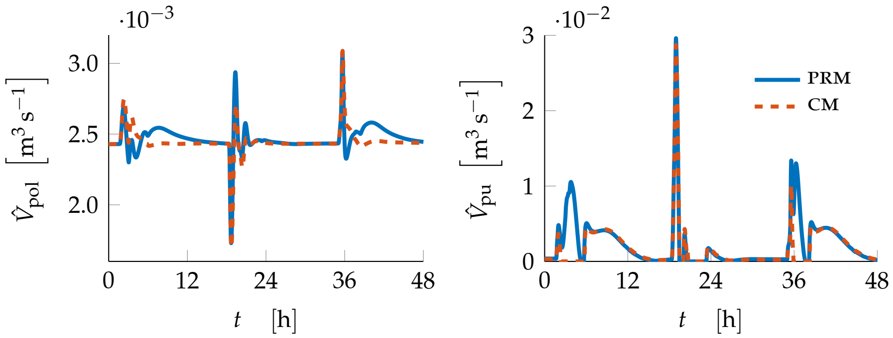
A – Compressibility.

B – Heat transfer coefficient.

FIGURE 5.14 – Compressibility and heat transfer coefficient during the grade transitions.

degree of accuracy. In spite of that, increasing the standard deviations of the compressibility would in all probability have induced more variations of the compressibility, which in turn could have led to unsatisfactory predictions. Increasing the standard deviation of the process noise via the compressibility could have been a satisfactory compromise, but these options have not been evaluated.

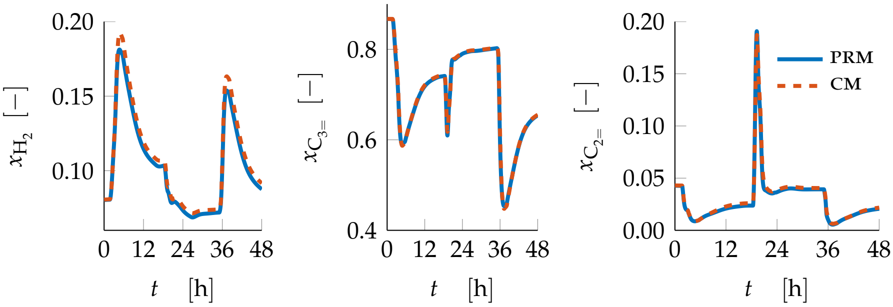
The volumetric purge and production rates are presented in Figure 5.15, in which some minor discrepancies between the CM and the PRM can be observed. These disagreements could easily have been rectified by including the flow of purge and production in the NMPC, i.e., setting the purge and production rates as MVS, and the level and the pressure as CVS. Nevertheless, if it is desirable to keep the level controller (LC) and the PC as PI controllers, other mitigations would have to be employed, e.g., better tuning of the KF. The flow rates are essential for achieving correct balance equations, e.g., mole and energy, and the deviations in Figure 5.15 would



a – Production rate.

b – Purge rate.

Figure 5.15 – Production and purge rates with model mismatch. The blue line and the dashed red line denote the control model (cm) and the plant replacement model (prm), respectively.



a – Hydrogen.

b – Propylene.

c – Ethylene.

Figure 5.16 – The mole fractions of hydrogen, propylene and ethylene. The blue line and the dashed red line indicate the plant replacement model (prm) and the control model (cm), respectively.

have to be mitigated in a real application. The mass based purge and production rates are omitted for the sake of brevity, but are displayed in Appendix h for completeness.

The mole fractions of hydrogen, ethylene and propylene are given in Figure 5.16, in which it can be ascertained that the mole fractions estimated by the CM coincides well with the PRM. This is in all likelihood due to two major factors; the low standard deviations of the measurements of the mole fractions, given in Table 4.1 in Chapter 4; in addition to the fact that the reaction scheme is identical for the two models.

If either of the models presented in this thesis were to be applied in the control of an industrial PP reactor, one option would be to utilize the PRM as the model of the NMPC. This would require offline estimation of several parameters of the PRM, e.g., flow rate coefficients, reaction rate corrections and heat transfer coefficient, against log data from the plant. Additionally, a recursive estimator would have to be set up and validated; the controller would also have to be tuned properly.

The controller was able to complete the grade transitions, despite the structural differences. As previously mentioned, the usual way to assess the robustness of the controller, is to vary the values of the parameters of the simulator slightly; but by testing the controller on a structurally different simulator, the test for robustness becomes more rigorous. These facts substantiate the claim that the controller is robust.

Chapter 6

Conclusion

In desperation I asked Fermi whether he was not impressed by the agreement between our calculated numbers and his measured numbers. He replied, "How many arbitrary parameters did you use for your calculations?" I thought for a moment about our cut-off procedures and said, "Four." He said, "I remember my friend Johnny von Neumann used to say, with four parameters I can fit an elephant, and with five I can make him wiggle his trunk."

— F. Dyson, 2004[1]

The purpose of this thesis was to demonstrate the applicability and robustness of nonlinear model predictive control (NMPC), despite structural differences between the model used by the controller and the plant, to a polypropylene (PP) fluidized bed reactor (FBR). Due to a lack of plant

[1] Dyson, F., "A meeting with Enrico Fermi," *Nature*, vol. 427, no. 297, 2004. DOI: 10.1038/427297a

data, two models were developed, a control model (CM) and a structurally different plant replacement model (PRM); the reason for this duality was to substantiate the robustness of NMPC. To ensure the best possible correspondence between the CM and the PRM, both offline and online parameter estimation techniques were carried out.

The controller was initially tuned by applying the CM as the model of the controller and the process simulator, i.e., no mismatch between the model of the controller and the simulator. In addition, the effect of utilizing the feed of an inert, e.g., nitrogen, as a manipulated variable (MV) was studied. Lastly, simulations were executed with the PRM as the process simulator, and a Kalman filter (KF) for online parameter estimation to demonstrate the robustness of the controller. In both scenarios, the performance of the controller was evaluated by simulating grade transitions between two PP grades.

6.1 Offline estimation

Offline parameter estimation was utilized to estimate eight parameters in CM. The fitting of the parameters was conducted by simulating a sequence of steps in the inputs, and minimizing the offset between the CM and the PRM. The objective function was chosen as the sum of squares of the deviation between the models at each sample, and the optimization was carried out with a sequential quadratic programming (SQP) algorithm. A validation of the obtained parameter set was performed by comparing the two models for a different sequence of inputs.

The fitting improved the agreement between the two models, and the resulting parameters were deemed to be reasonable; the parameters served as good initial values for the KF.

6.2 Online estimation

A second-order divided difference (DD2) KF was utilized for recursive parameter estimation for seven of the parameters of the CM. The standard deviation of the noise of the measurements was set to arbitrary, albeit realistic values, while standard deviations of the noise of the process and parameters were adjusted to tune the estimator. The KF was initially tuned against an open-loop simulation of a sequence of input steps, and then retuned after observing its behavior in a closed-loop grade transition.

The KF improved the correspondence between the two models further, but there were still some minor discrepancies. These deviations could most likely have been mitigated by fine-tuning the estimator, this was however, not in the scope of this thesis.

6.3 Model predictive control

A Newton-type NMPC algorithm was utilized for control of the reactor, and the optimization problem was solved using an SQP algorithm. Two polymer quality parameters were controlled, in addition to the purge and production rates. To test the performance of the controller, grade transitions between two pp grades were performed.

Two scenarios were studied, the first examined the effect of utilizing the feed of inert as an mv, while the second considered mismatch between the model used by the controller and the model used by the process simulator.

6.3.1 Inert feed as an input

Simulations of the grade transitions were carried out with and without the feed of inert as an input. The transitions occurred faster, with less use of purge and better control of pressure, when the inert feed was utilized. However, the use of catalyst increased considerably due to the decreased concentration of monomers.

6.3.2 Model mismatch

The controller was able to complete the grade transitions reasonable fast despite the mismatch between the model used for predictions, i.e., the cm, and the prm. However, the overshoots for the melt flow index (mfi) were found to be quite significant, and would have to be remedied in a real application.

The recursively estimated parameters experienced larger variations in closed-loop operations than in open-loop, especially the reaction rate corrections for the two monomers. Most of the measurements were tracked with a high degree of accuracy, and the most conspicuous mismatches between the two models were observed for the pressure, purge rate and production rate. Nevertheless, the estimator performed well, but further fine-tuning is needed mitigate the inconsistencies.

Overall, the controller and the estimator proved to be robust and were able to cope with the model mismatch. Testing of the controller in closed-loop with a structurally different process simulator is rarely accomplished, but has the significant advantage that it may corroborate the robustness of the controller before it is implemented on site.

Bibliography

[1] Ali, M. A.-H., Betlem, B., Weickert, G., and Roffel, B., "Non-linear model based control of a propylene polymerization reactor," *Chemical Engineering and Processing: Process Intensification*, vol. 46, no. 6, pp. 554–564, 2007. DOI: 10.1016/j.cep.2006.07.012.

[2] Alizadeh, M., Mostoufi, N., Pourmahdian, S., and Sotudeh-Gharebagh, R., "Modeling of fluidized bed reactor of ethylene polymerization," *Chemical Engineering Journal*, vol. 97, no. 1, pp. 27–35, 2004. DOI: 10.1016/S1385-8947(03)00133-5.

[3] Åström, K. J. and Murray, R. M., *Feedback Systems, An Introduction for Scientists and Engineers*, 2nd ed. Princeton University Press, 2014, ch. 10, ISBN: 9780691135762. [Online]. Available: http://www.cds.caltech.edu/~murray/books/AM08/pdf/fbs-pid_22Jun14.pdf (visited on Mar. 10, 2015).

[4] Bartusiak, R. D., "NLMPC: a platform for optimal control of feed- or product-flexible manufacturing," in *Assessment and Future Directions of Nonlinear Model Predictive Control*, ser. Lecture Notes in Control and Information Sciences, Findeisen, R., Allgöwer, F., and Biegler, L. T., Eds., vol. 358, Springer Berlin Heidelberg, 2007, pp. 367–381, ISBN: 9783540726982. DOI: 10.1007/978-3-540-72699-9_30.

[5] Bemporad, A. and Morari, M., "Robust model predictive control: a survey," in *Robustness in identification and control*, ser. Lecture Notes in Control and Information Sciences, Garulli, A. and Tesi, A., Eds., vol. 245, Springer London, 1999, pp. 207–226, ISBN: 9781852331795. DOI: 10.1007/BFb0109870.

[6] BenAmor, S., Doyle III, F. J., and McFarlane, R., "Polymer grade transition control using advanced real-time optimization software," *Journal of Process Control*, vol. 14, no. 4, pp. 349–364, 2004. DOI: 10.1016/j.jprocont.2003.06.001.

[7] Bergman, T. L., Lavine, A. S., Incropera, F. P., and Dewitt, D. P., *Fundamentals of Heat and Mass Transfer*, 7th ed. Hoboken, New Jersey: John Wiley & Sons Limited, 2011, ISBN: 9780470501979.

[8] Bindlish, R., "Nonlinear model predictive control of an industrial polymerization process," *Computers & Chemical Engineering*, vol. 73, pp. 43–48, 2015. DOI: 10.1016/j.compchemeng.2014.11.001.

[10] Cagienard, R., Grieder, P., Kerrigan, E. C., and Morari, M., "Move blocking strategies in receding horizon control," *Journal of Process*

Control, vol. 17, no. 6, pp. 563–570, 2007. DOI: 10.1016/j.jprocont.2007.01.001.

[11] Camacho, E. F. and Bordons, C., *Model Predictive Control*, 2nd ed., Grimble, M. J. and Johnson, M. A., Eds., ser. Advanced Textbooks in Control and Signal Processing. London, England: Springer London, 2007, ISBN: 9780857293985. DOI: 10.1007/978-0-85729-398-5.

[12] Camacho, E. F. and Bordons, C., "Nonlinear model predictive control: an introductory review," in *Assessment and Future Directions of Nonlinear Model Predictive Control*, ser. Lecture Notes in Control and Information Sciences, Findeisen, R., Allgöwer, F., and Biegler, L. T., Eds., vol. 358, Springer Berlin Heidelberg, 2007, pp. 1–16, ISBN: 9783540726982. DOI: 10.1007/978-3-540-72699-9_1.

[13] Carraher Jr., C. E., *Carraher's Polymer Chemistry*, 9th ed. Boca Raton, Florida: CRC Press/Taylor & Francis, 2014, ISBN: 9781466552036.

[14] Carvalho, A. B. de, Gloor, P. E., and Hamielec, A. E., "A kinetic mathematical model for heterogeneous Ziegler-Natta copolymerization," *Polymer*, vol. 30, no. 2, pp. 280–196, 1989. DOI: 10.1016/0032-3861(89)90118-3.

[15] Chase Jr., M. W., Ed., *NIST-JANAF Thermochemical Tables*, 4th ed., American Institute of Physics, 1998, ISBN: 9781563968310.

[16] Chatzidoukas, C., Perkins, J. D., Pistikopoulos, E. N., and Kiparissides, C., "Optimal grade transition and selection of closed-loop

controllers in a gas-phase olefin polymerization fluidized bed reactor," *Chemical Engineering Science*, vol. 58, no. 16, pp. 3643–3658, 2003. DOI: 10.1016/S0009-2509(03)00223-9.

[17] Constantinides, A. and Mostoufi, N., *Numerical Methods for Chemical Engineers with* MATLAB *Applications*, 1st ed., ser. Prentice Hall International Series in the Physical and Chemical Engineering Sciences. Upper Saddle River, New Jersey: Prentice Hall PTR, 1999, ISBN: 0130138517.

[18] Cui, H., Mostoufi, N., and Chaouki, J., "Characterization of dynamic gas–solid distribution in fluidized beds," *Chemical Engineering Journal*, vol. 79, no. 2, pp. 133–143, 2000. DOI: 10.1016/S1385-8947(00)00178-9.

[19] Dalle Molle, D. T. and Himmelblau, D. M., "Fault detection in a single-stage evaporator via parameter estimation using the kalman filter," *Industrial & Engineering Chemistry Research*, vol. 26, no. 12, pp. 2482–2489, 1987. DOI: 10.1021/ie00072a016.

[20] Davidson, J. F. and Harrison, D., *Fluidised particles*. Cambridge University Press, 1963, ISBN: 0521047897.

[21] Debling, J. A., Han, G. C., Kuijpers, F., Verburg, J., Zacca, J. J., and Ray, W. H., "Dynamic modeling of product grade transitions for olefin polymerization processes," *AIChE Journal*, vol. 40, no. 3, pp. 506–520, 1994. DOI: 10.1002/aic.690400312.

[24] Eliçabe, G. E. and Meira, G. R., "Estimation and control in polymerization reactors. a review," *Polymer Engineering & Science*, vol. 28, no. 3, pp. 121–135, 1988, ISSN: 1548-2634. DOI: 10.1002/pen.760280302.

[25] Farag, H., Ossman, M., Mansour, M., and Farid, Y., "Modeling of fluidized bed reactor for ethylene polymerization: effect of parameters on the single-pass ethylene conversion," *International Journal of Industrial Chemistry*, vol. 4, no. 1, pp. 1–10, 2013. DOI: 10.1186/2228-5547-4-20.

[26] Feather, D., Harrell, D., Lieberman, R., and Doyle III, F. J., "Hybrid approach to polymer grade transition control," *AIChE Journal*, vol. 50, no. 10, pp. 2502–2513, 2004. DOI: 10.1002/aic.10250.

[27] Findeisen, R., Allgöwer, F., and Biegler, L. T., Eds., vol. 358, ser. Lecture Notes in Control and Information Sciences, Springer Berlin Heidelberg, 2007, ISBN: 9783540726982. DOI: 10.1007/978-3-540-72699-9.

[28] Findeisen, R., Imsland, L., Allgöwer, F., and Foss, B. A., "State and output feedback nonlinear model predictive control: an overview," *European Journal of Control*, vol. 9, no. 2–3, pp. 190–206, 2003. DOI: 10.3166/ejc.9.190-206.

[30] Floyd, S., Heiskanen, T., Taylor, T. W., Mann, G. E., and Ray, W. H., "Polymerization of olefins through heterogeneous catalysis. VI. effect of particle heat and mass transfer on polymerization behavior

and polymer properties," *Journal of Applied Polymer Science*, vol. 33, no. 4, pp. 1021–1065, 1987. DOI: 10.1002/app.1987.070330402.

[31] Fogler, H. S., *Elements of Chemical Reaction Engineering*, 4th ed., Amundson, N. R., Ed., ser. Prentice Hall PTR International Series in the Physical and Chemical Engineering Sciences. Upper Saddle River, New Jersey: Pearson Education Inc., 2006, ISBN: 9780130473943.

[32] Foss, B. A. and Schei, T. S., "Putting nonlinear model predictive control into use," in *Assessment and Future Directions of Nonlinear Model Predictive Control*, ser. Lecture Notes in Control and Information Sciences, Findeisen, R., Allgöwer, F., and Biegler, L. T., Eds., vol. 358, Springer Berlin Heidelberg, 2007, pp. 407–417, ISBN: 9783540726982. DOI: 10.1007/978-3-540-72699-9_33.

[33] Franke, R. and Doppelhamer, J., "Integration of advanced model based control with industrial IT," in *Assessment and Future Directions of Nonlinear Model Predictive Control*, ser. Lecture Notes in Control and Information Sciences, Findeisen, R., Allgöwer, F., and Biegler, L. T., Eds., vol. 358, Springer Berlin Heidelberg, 2007, pp. 399–406, ISBN: 9783540726982. DOI: 10.1007/978-3-540-72699-9_32.

[34] Geankoplis, C. J., *Transport Processes and Separation Process Principles, (includes Unit Operations)*, 4th ed. Upper Saddle River, New Jersey: Pearson Education Limited, 2009, ISBN: 013101367X.

[35] Grace, J. R., “Modelling and simulation of two-phase fluidized bed reactors,” in *Chemical Reactor Design and Technology*, ser. NATO ASI Series, Lasa, H. I. de, Ed., vol. 110, Springer Netherlands, 1986, pp. 245–289, ISBN: 9789401084574. DOI: 10.1007/978-94-009-4400-8_7.

[36] Haider, A. and Levenspiel, O., “Drag coefficient and terminal velocity of spherical and nonspherical particles,” *Powder Technology*, vol. 58, no. 1, pp. 63–70, 1989. DOI: 10.1016/0032-5910(89)80008-7.

[37] Hatzantonis, H., Yiannoulakis, H., Yiagopoulos, A., and Kiparissides, C., “Recent developments in modeling gas-phase catalyzed olefin polymerization fluidized-bed reactors: the effect of bubble size variation on the reactor’s performance,” *Chemical Engineering Science*, vol. 55, no. 16, pp. 3237–3259, 2000. DOI: 10.1016/S0009-2509(99)00565-5.

[38] Haug-Warberg, T., *Den termodynamiske arbeidsboken, for de to hundre hjem*, Norwegian, 2nd ed. Trondheim, Norway: Kolofon Forlag AS, 2006, ISBN: 9788230002056.

[39] Haynes, W. M., Ed., *CRC Handbook of Chemistry and Physics, Critical Constants of Inorganic Compounds*, 95th ed., Boca Ration, Florida: CRC Press/Taylor & Francis, 2015, ISBN: 9781482208672.

[40] Haynes, W. M., Ed., *CRC Handbook of Chemistry and Physics, Critical Constants of Organic Compounds*, 95th ed., Boca Ration, Florida: CRC Press/Taylor & Francis, 2015, ISBN: 9781482208672.

[41] Haynes, W. M., Ed., *CRC Handbook of Chemistry and Physics, Standard Atomic Weights*, 95th ed., Boca Ration, Florida: CRC Press/Taylor & Francis, 2015, ISBN: 9781482208672.

[42] Haynes, W. M., Ed., *CRC Handbook of Chemistry and Physics, CODATA Recommended Values of the Fundemental Physical Constants*, 95th ed., Boca Ration, Florida: CRC Press/Taylor & Francis, 2015, ISBN: 9781482208672.

[43] Ho, Y. K., Shamiri, A., Mjalli, F. S., and Hussain, M. A., "Control of industrial gas phase propylene polymerization in fluidized bed reactors," *Journal of Process Control*, vol. 22, no. 6, pp. 947–958, 2012. DOI: 10.1016/j.jprocont.2012.04.003.

[45] Hovd, M. and Skogestad, S., "Sequential design of decentralized controllers," *Automatica*, vol. 30, no. 10, pp. 1601–1607, 1994. DOI: 10.1016/0005-1098(94)90099-X.

[46] Hutchinson, R. A., Chen, C. M., and Ray, W. H., "Polymerization of olefins through heterogeneous catalysis X: modeling of particle growth and morphology," *Journal of Applied Polymer Science*, vol. 44, no. 8, pp. 1389–1414, 1992. DOI: 10.1002/app.1992.070440811.

[47] Jakobsen, H. A., *Chemical Reactor Modeling, Multiphase Reactive Flows*, 2nd ed. Trondheim, Norway: Springer International Publishing, 2014, ISBN: 9783319050928. DOI: 10.1007/978-3-319-05092-8.

[48] Johansen, T. A., "Introduction to nonlinear model predictive control and moving horizon estimation," in *Selected Topics on Con-*

strained and Nonlinear Control, Huba, M., Skogestad, S., Fikar, M., Hovd, M., Johansen, T. A., and Rohal'-Ilkiv, B., Eds., Bratislava, Slovakia: Miloslav Roubal ROSA, Dolný Kubín and Tlačiareň Vrábel, Dolný Kubín, 2011, ch. 5, pp. 187–239, ISBN: 9788096862740.

[51] Kiashemshaki, A., Mostoufi, N., and Sotudeh-Gharebagh, R., “Two-phase modeling of a gas phase polyethylene fluidized bed reactor,” *Chemical Engineering Science*, vol. 61, no. 12, pp. 3997–4006, 2006. DOI: 10.1016/j.ces.2006.01.042.

[52] Kiashemshaki, A., Mostoufi, N., Sotudeh-Gharebagh, R., and Pourmahdian, S., “Reactor modeling of gas-phase polymerization of ethylene,” *Chemical Engineering & Technology*, vol. 27, no. 11, pp. 1227–1232, 2004. DOI: 10.1002/ceat.200401964.

[53] Kiparissides, C., “Polymerization reactor modeling: a review of recent developments and future directions,” *Chemical Engineering Science*, vol. 51, no. 10, pp. 1637–1659, 1996. DOI: 10.1016/0009-2509(96)00024-3.

[54] Kolås, S., Foss, B. A., and Schei, T. S., “Constrained nonlinear state estimation based on the UKF approach,” *Computers & Chemical Engineering*, vol. 33, no. 8, pp. 1386–1401, 2009. DOI: 10.1016/j.compchemeng.2009.01.012.

[55] Kolås, S., Foss, B. A., and Schei, T. S., “Noise modeling concepts in nonlinear state estimation,” *Journal of Process Control*, vol. 19, no. 7, pp. 1111–1125, 2009. DOI: 10.1016/j.jprocont.2009.03.002.

[56] Kreyszig, E., *Advanced Engineering Mathematics*, 10th ed. Hoboken, New Jersey: John Wiley & Sons, 2010, ISBN: 9780470458365.

[57] Kunii, D. and Levenspiel, O., *Fluidization Engineering*, 2nd ed., ser. Butterworth-Heinemann series in chemical engineering. Boston, Massachusetts: Butterworth-Heinemann, 1991, ISBN: 9780409902334.

[58] Kwon, W. H. and Han, S. H., *Receding Horizon Control, Model Predictive Control for State Models*, 1st ed., ser. Advanced Textbooks in Control and Signal Processing. London, England: Springer-Verlag London, 2005, ISBN: 9781846280177. DOI: 10.1007/b136204.

[59] Li, W. C. and Biegler, L. T., "A multistep, Newton-type control strategy for constrained, nonlinear processes," *Chemical Engineering Research and Design*, vol. 67, pp. 562–577, 1989.

[60] Li, W. C., Biegler, L. T., Economou, C. G., and Morari, M., "A constrained pseudo-Newton control strategy for nonlinear systems," *Computers & Chemical Engineering*, vol. 14, no. 4–5, pp. 451–468, 1990. DOI: 10.1016/0098-1354(90)87020-P.

[61] Lucas, A., Arnaldos, J., Casal, J., and Puigjaner, L., "Improved equation for the calculation of minimum fluidization velocity," *Industrial & Engineering Chemistry Process Design and Development*, vol. 25, no. 2, pp. 426–429, 1986. DOI: 10.1021/i200033a013.

[62] Maciejowski, J. M., *Predictive Control, with Constraints*, 1st ed. Essex, England: Pearson Education Limited, 2002, ISBN: 0201398230.

[63] Mahadevan, R., Doyle III, F. J., and Allcock, A. C., "Control-relevant scheduling of polymer grade transitions," *AIChE Journal*, vol. 48, no. 8, pp. 1754–1764, 2002. DOI: 10.1002/aic.690480816.

[64] Mawatari, Y., Tatemoto, Y., and Noda, K., "Prediction of minimum fluidization velocity for vibrated fluidized bed," vol. 131, no. 1, pp. 66–70, 2003. DOI: 10.1016/S0032-5910(02)00323-6.

[65] McAuley, K. B. and MacGregor, J. F., "On-line inference of polymer properties in an industrial polyethylene reactor," *AIChE Journal*, vol. 37, no. 6, pp. 825–835, 1991. DOI: 10.1002/aic.690370605.

[66] McAuley, K. B., MacGregor, J. F., and Hamielec, A. E., "A kinetic model for industrial gas-phase ethylene copolymerization," *AIChE Journal*, vol. 36, no. 6, pp. 837–850, 1990. DOI: 10.1002/aic.690360605.

[67] McAuley, K. B., Talbot, J. P., and Harris, T. J., "A comparison of two-phase and well-mixed models for fluidized-bed polyethylene reactors," *Chemical Engineering Science*, vol. 49, no. 13, pp. 2035–2045, 1994. DOI: 10.1016/0009-2509(94)E0030-T.

[68] McMurry, J., *Organic Chemistry*, 8th ed. Belmont, California: Brooks/Cole, Cengage Learning, 2012, ISBN: 9780840054449.

[69] Mieras, H. J. M. A. and Rijn, C. F. H. van, "Influence of molecular weight distribution on the elasticity and processing properties of polypropylene melts," *Journal of Applied Polymer Science*, vol. 13, no. 2, pp. 309–322, 1969. DOI: 10.1002/app.1969.070130204.

[70] Mori, S. and Wen, C. Y., “Estimation of bubble diameter in gaseous fluidized beds,” *AIChE Journal*, vol. 21, no. 1, pp. 109–115, 1975. DOI: 10.1002/aic.690210114.

[71] Nagy, Z. K., Mahn, B., Franke, R., and Allgöwer, F., “Real-time implementation of nonlinear model predictive control of batch processes in an industrial framework,” in *Assessment and Future Directions of Nonlinear Model Predictive Control*, ser. Lecture Notes in Control and Information Sciences, Findeisen, R., Allgöwer, F., and Biegler, L. T., Eds., vol. 358, Springer Berlin Heidelberg, 2007, pp. 465–472, ISBN: 9783540726982. DOI: 10.1007/978-3-540-72699-9_38.

[72] Naidoo, K., Guiver, J., Turner, P., Keenan, M., and Harmse, M., “Experiences with nonlinear MPC in polymer manufacturing,” in *Assessment and Future Directions of Nonlinear Model Predictive Control*, ser. Lecture Notes in Control and Information Sciences, Findeisen, R., Allgöwer, F., and Biegler, L. T., Eds., vol. 358, Springer Berlin Heidelberg, 2007, pp. 383–398, ISBN: 9783540726982. DOI: 10.1007/978-3-540-72699-9_31.

[74] Nocedal, J. and Wright, S. J., *Numerical Optimization*, 2nd ed., ser. Springer Series in Operations Research and Financial Engineering. New York City, New York: Springer-Verlag New York, 2006, ISBN: 9780387400655. DOI: 10.1007/978-0-387-40065-5.

[75] Nørgaard, M., Poulsen, N. K., and Ravn, O., "New developments in state estimation for nonlinear systems," *Automatica*, vol. 36, no. 11, pp. 1627–1638, 2000. DOI: 10.1016/S0005-1098(00)00089-3.

[76] Oliveira, N. M. C. de and Biegler, L. T., "An extension of Newton-type algorithms for nonlinear process control," *Automatica*, vol. 31, no. 2, pp. 281–286, 1995. DOI: 10.1016/0005-1098(94)00086-X.

[77] Oliveira, N. M. C. de and Biegler, L. T., "Constraint handing and stability properties of model-predictive control," *AIChE Journal*, vol. 40, no. 7, pp. 1138–1155, 1994. DOI: 10.1002/aic.690400706.

[78] O'Neill, M. and Budavari, S., Eds., *The Merck Index, An Encyclopedia of Chemicals, Drugs, and Biologicals*, 13th ed., Whitehouse Station, New Jersey: Merck and Co., Inc., 2001, p. 60, ISBN: 9780911910131.

[79] Özkan, L., Kothare, M. V., and Georgakis, C., "Control of a solution copolymerization reactor using multi-model predictive control," *Chemical Engineering Science*, vol. 58, no. 7, pp. 1207–1221, 2003. DOI: 10.1016/S0009-2509(02)00559-6.

[80] PlasticsEurope. (2013). Plastics – the facts 2013, An analysis of European latest plastics production, demand and waste data, PlasticsEurope, European Association of Plastics Recycling, and Recovery Organisations, [Online]. Available: http://www.plasticseurope.org/documents/document/20131014095824-final_plastics_the_facts_2013_published_october2013.pdf (visited on Jun. 7, 2015).

[82] Qin, S. and Badgwell, T. A., "A survey of industrial model predictive control technology," *Control Engineering Practice*, vol. 11, no. 7, pp. 733–764, 2003. DOI: 10.1016/S0967-0661(02)00186-7.

[83] Rawlings, J. B. and Mayne, D. Q., *Model Predictive Control*, 5th ed. Madison, Wisconsin: Nob Hill Publishing, 2015, ISBN: 9780975937709 (print). [Online]. Available: http://jbrwww.che.wisc.edu/home/jbraw/mpc/electronic-book.pdf (visited on May 14, 2015).

[84] Redlich, O. and Kwong, J. N. S., "On the thermodynamics of solutions. V: an equation of state. fugacities of gaseous solutions," *Chemical Reviews*, vol. 44, no. 1, pp. 233–244, 1949. DOI: 10.1021/cr60137a013.

[85] Richards, J. R. and Congalidis, J. P., "Measurement and control of polymerization reactors," *Computers & Chemical Engineering*, vol. 30, no. 10–12, pp. 1447–1463, 2006. DOI: 10.1016/j.compchemeng.2006.05.021.

[86] Roman, R., Nagy, Z. K., Cristea, M. V., and Agachi, S. P., "Dynamic modelling and nonlinear model predictive control of a fluid catalytic cracking unit," *Computers & Chemical Engineering*, vol. 33, no. 3, pp. 605–617, 2009. DOI: 10.1016/j.compchemeng.2008.08.007.

[87] Rossiter, J. A., *Model-Based Predictive Control, A Practical Approach*, 1st ed., Bishop, R. H., Ed., ser. Control Series. Boca Raton, Florida: CRC Press, 2003, ISBN: 0849312914.

[88] Rowley, R. L., Wilding, W. V., and Oscarson, J. L., Eds., *DIPPR Project 801 Data Compilation of Pure Compound Properties*, Design Institute for Physical Properties, 2011.

[89] Schei, T. S., "A finite-difference method for linearization in nonlinear estimation algorithms," *Automatica*, vol. 33, no. 11, pp. 2053–2058, 1997. DOI: 10.1016/S0005-1098(97)00127-1.

[90] Schei, T. S., "On-line estimation for process control and optimization applications," *Journal of Process Control*, vol. 18, no. 9, pp. 821–828, 2008. DOI: 10.1016/j.jprocont.2008.06.014.

[91] Schei, T. S. and Singstad, P., "Nonlinear model predictive control of a batch polymerization process," in *Proceedings of the American Control Conference 1998*, vol. 6, Jun. 1998, pp. 3381–3385. DOI: 10.1109/ACC.1998.703205.

[92] Seborg, D. E., Edgar, T. F., Mellichamp, D. A., and Doyle III, F. J., *Process dynamics and control*, 3rd ed. Hoboken, New Jersey: John Wiley & Sons, 2011, ISBN: 9780470646106.

[93] Seki, H., Ogawa, M., Ooyama, S., Akamatsu, K., Ohshima, M., and Yang, W., "Industrial application of a nonlinear model predictive control to polymerization reactors," *Control Engineering Practice*, vol. 9, no. 8, pp. 819–828, 2001, Advanced Control of Chemical Processes. DOI: 10.1016/S0967-0661(01)00046-6.

[94] Shamiri, A., Hussain, M. A., Mjalli, F. S., and Mostoufi, N., "Comparative simulation study of gas-phase propylene polymerization

in fluidized bed reactors using aspen polymers and two-phase models," *Chemical Industry & Chemical Engineering Quarterly*, vol. 19, no. 1, pp. 13–24, 2013. DOI: 10.2298/CICEQ111214038S.

[95] Shamiri, A., Hussain, M. A., Mjalli, F. S., and Mostoufi, N., "Experimental and modeling analysis of propylene polymerization in a pilot-scale fluidized bed reactor," *Industrial & Engineering Chemistry Research*, vol. 53, no. 21, pp. 8694–8705, 2014. DOI: 10.1021/ie501155h.

[96] Shamiri, A., Hussain, M. A., Mjalli, F. S., and Mostoufi, N., "Improved single phase modeling of propylene polymerization in a fluidized bed reactor," *Computers & Chemical Engineering*, vol. 36, pp. 35–47, 2012. DOI: 10.1016/j.compchemeng.2011.07.015.

[97] Shamiri, A., Hussain, M. A., Mjalli, F. S., and Mostoufi, N., "Kinetic modeling of propylene homopolymerization in a gas-phase fluidized-bed reactor," *Chemical Engineering Journal*, vol. 161, no. 1–2, pp. 240–249, 2010. DOI: 10.1016/j.cej.2010.04.037.

[98] Shamiri, A., Hussain, M. A., Mjalli, F. S., Mostoufi, N., and Hajimolana, S., "Dynamics and predictive control of gas phase propylene polymerization in fluidized bed reactors," *Chinese Journal of Chemical Engineering*, vol. 21, no. 9, pp. 1015–1029, 2013. DOI: 10.1016/S1004-9541(13)60565-0.

[99] Shamiri, A., Hussain, M. A., Mjalli, F. S., Mostoufi, N., and Shafeeyan, M. S., "Dynamic modeling of gas phase propylene homopolymerization in fluidized bed reactors," *Chemical Engineering*

Science, vol. 66, no. 6, pp. 1189–1199, 2011. DOI: 10.1016/j.ces.2010.12.030.

[100] Shamiri, A., Wong, S. W., Zanil, M. F., Hussain, M. A., and Mostoufi, N., “Modified two-phase model with hybrid control for gas phase propylene copolymerization in fluidized bed reactors,” *Chemical Engineering Journal*, vol. 264, pp. 706–719, 2015. DOI: 10.1016/j.cej.2014.11.104.

[101] Shi, J., Liu, X., and Sun, Y., “Melt index prediction by neural networks based on independent component analysis and multi-scale analysis,” *Neurocomputing*, vol. 70, no. 1–3, pp. 280–287, 2006. DOI: 10.1016/j.neucom.2006.01.029.

[102] Simon, D., *Optimal State Esimation, Kalman, H_∞, and Nonlinear Approaches*, 1st ed. Hoboken, New Jersey: John Wiley & Sons, 2006, ISBN: 9780471708582.

[103] Smith, R., *Chemical Process Design and Integration*, 1st ed. Chichester, England: John Wiley & Sons Limited, 2005, ISBN: 0471486809.

[104] Tripathi, D., *Practical Guide to Polypropylene*, ser. Rapra practical guide series. Smithers Rapra Technology, 2002, ISBN: 9781859572825. [Online]. Available: http://app.knovel.com/hotlink/toc/id:kpPGP00002/practical-guide-polypropylene/practical-guide-polypropylene (visited on Jun. 2, 2015).

[105] Tullock, C. W., Tebbe, F. N., Mulhaupt, R., Ovenall, D. W., Setterquist, R. A., and Ittel, S. D., “Polyethylene and elastomeric

polypropylene using alumina-supported bis(arene) titanium, zirconium, and hafnium catalysts," *Journal of Polymer Science Part A: Polymer Chemistry*, vol. 27, no. 9, pp. 3063–3081, 1989. DOI: 10.1002/pola.1989.080270916.

[106] Wang, L., *Model Predictive Control System Design and Implementation Using* MATLAB®, 1st ed., Grimble, M. J. and Johnson, M. A., Eds., ser. Advances in Industrial Control. London, England: Springer London, 2009, ISBN: 9781848823310. DOI: 10.1007/978-1-84882-331-0.

[107] Wang, Y., Seki, H., Ohyama, S., akamatsu, K., Ogawa, M., and Ohshima, M., "Optimal grade transition control for polymerization reactors," *Computers & Chemical Engineering*, vol. 24, no. 2–7, pp. 1555–1561, 2000. DOI: 10.1016/S0098-1354(00)00550-0.

[108] Werther, J., "Hydrodynamics and mass transfer between the bubble and emulsion phases in fluidized beds of sand and cracking catalyst," in *Proceedings of the 4th International Conference On Fluidization*, Kunii, D. and Ryozo, T., Eds., American Institute of Chemical Engineers, New York, 1983, pp. 93–101.

Appendix A

Additional parameters

> It is a capital mistake to theorize before one has data. Insensibly one begins to twist facts to suit theories, instead of theories to suit facts.
>
> — Sherlock Holmes, 1892[1]

A.1 Reactor design

The parameters of the reactor are given in Table A.1, including the dimensions of the reactor, the heat transfer coefficient and the flow rate constants.

[1] Doyle, Sir. A. C., "A Scandal in Bohemia," in *The Adventures of Sherlock Holmes*. 1892

TABLE A.1 – Reactor design parameters.

VARIABLE	VALUE	SOURCE
Reactor diameter, d_r	$5\,\mathrm{m}$	[2]
Reactor volume, V_{tot}	$350\,\mathrm{m}^3$	chosen
Heat exchanger volume, V_{hex}	$25\,\mathrm{m}^3$	chosen
Particle diameter, d_p	$500 \cdot 10^{-6}\,\mathrm{m}$	[100]
Particle sphericity, ϕ_p	1	[100]
Heat transfer coefficient, UA	$60\,\mathrm{kW\,K^{-1}}$	chosen
Flow constant for emulsion phase, k_e	$2 \cdot 10^{-3}\,\mathrm{m^3\,s^{-1}\,Pa^{-1}}$	chosen
Flow constant for bubble phase, k_b	$5 \cdot 10^{-3}\,\mathrm{m^3\,s^{-1}\,Pa^{-1}}$	chosen

A.2 PHYSICAL PARAMETERS AND CONSTANTS

All the physical parameters and constants are displayed in Table A.2.

TABLE A.2 – Physical parameters and constants.

VARIABLE	VALUE	SOURCE
Catalyst density, ρ_{cat}	$2370\,\mathrm{kg\,m^{-3}}$	[95]
Polymer density, ρ_{pol}	$910\,\mathrm{kg\,m^{-3}}$	[100]
$AlEt_3$ density, ρ_{AlEt3}	$832\,\mathrm{kg\,m^{-3}}$	[78]
Minimum fluidization void fraction, ε_{mf}	0.45	[66, 100]
Product stream void fraction, ε_{pol}	0.3	[64]
Molecular mass of H_2, M_{H_2}	$2.016\,\mathrm{kg\,kmol^{-1}}$	[41]
Molecular mass of $C_{3=}$, $M_{C_{3=}}$	$42.081\,\mathrm{kg\,kmol^{-1}}$	[41]

Continued on next page

Table A.2 – Continued from previous page

VARIABLE	VALUE	SOURCE
Molecular mass of $C_{2=}$, $M_{C_{2=}}$	$28.054\,\mathrm{kg\,kmol^{-1}}$	[41]
Molecular mass of N_2, M_{N_2}	$28.014\,\mathrm{kg\,kmol^{-1}}$	[41]
Molecular mass of $AlEt_3$, M_{AlEt_3}	$114.17\,\mathrm{kg\,kmol^{-1}}$	[41]
Molecular mass of Ti, M_{Ti}	$47.867\,\mathrm{kg\,kmol^{-1}}$	[41]
Gas constant, R	$8.314 \cdot 10^3\,\mathrm{J\,kmol^{-1}\,K^{-1}}$	[42]
Heat capacity of H_2, c_{p,H_2}	$29.02 \cdot 10^3\,\mathrm{J\,kmol^{-1}\,K^{-1}}$	[88]
Heat capacity of $C_{3=}$, $c_{p,C_{3=}}$	$71.89 \cdot 10^3\,\mathrm{J\,kmol^{-1}\,K^{-1}}$	[88]
Heat capacity of $C_{2=}$, $c_{p,C_{2=}}$	$47.46 \cdot 10^3\,\mathrm{J\,kmol^{-1}\,K^{-1}}$	[88]
Heat capacity of N_2, c_{p,N_2}	$29.16 \cdot 10^3\,\mathrm{J\,kmol^{-1}\,K^{-1}}$	[88]
Heat capacity of $AlEt_3$, $c_{p,AlEt_3}$	$239 \cdot 10^3\,\mathrm{J\,kmol^{-1}\,K^{-1}}$	[15]
Heat capacity of polymer, $c_{p,\mathrm{pol}}$	$2.25 \cdot 10^3\,\mathrm{J\,kg^{-1}\,K^{-1}}$	[1]
Heat capacity of catalyst, $c_{p,\mathrm{cat}}$	$0.75 \cdot 10^3\,\mathrm{J\,kg^{-1}\,K^{-1}}$	[15]
Heat of propagation of $C_{3=}$, $\Delta_{\mathrm{rx}} h_{C_{3=}}$	$-1.0376 \cdot 10^8\,\mathrm{J\,kmol^{-1}}$	[30]
Heat of propagation of $C_{2=}$, $\Delta_{\mathrm{rx}} h_{C_{2=}}$	$-1.0753 \cdot 10^8\,\mathrm{J\,kmol^{-1}}$	[30]
Gravitational acceleration, g	$9.81\,\mathrm{m\,s^{-2}}$	[42]
Gas viscosity, μ_{g}	$1.14 \cdot 10^{-4}\,\mathrm{Pa\,s}$	[100]
Gas diffusivity, D_{g}	$4 \cdot 10^{-7}\,\mathrm{m^2\,s^{-1}}$	[37]
Gas conductivity, k_{g}	$0.0318\,\mathrm{J\,m^{-1}\,s^{-1}\,K^{-1}}$	[37]

A.3 Reaction rate constants

The reaction rate constants used in Table 2.1 in Chapter 2 is presented in Table A.3.

TABLE A.3 – Reaction rate constants for a two-site model of the reactions in Table 2.1 in Chapter 2 [100].

VARIABLE	SITE TYPE 1	SITE TYPE 2
Formation $\left[\mathrm{s}^{-1}\right]$		
kf	1	1
Initiation $\left[\mathrm{m}^3\,\mathrm{kmol}^{-1}\,\mathrm{s}^{-1}\right]$		
ki_1	9.8	9.8
ki_2	14.6	14.6
kh_1	1	1
kh_2	0.1	0.1
khr	20	20
Propagation $\left[\mathrm{m}^3\,\mathrm{kmol}^{-1}\,\mathrm{s}^{-1}\right]$		
kp_{11}	220.477	22.047
kp_{12}	591.1098	130.783
kp_{21}	1.701	376.396
kp_{22}	4.561	6.698
Transfer $\left[\mathrm{m}^3\,\mathrm{kmol}^{-1}\,\mathrm{s}^{-1}\right]$		
kfm_{11}	0.006	0.006
kfm_{12}	0.0021	0.0021
kfm_{21}	0.006	0.006
kfm_{22}	0.0021	0.0021
kfh_1	0.088	0.37
kfh_2	0.088	0.37
kfr_1	0.024	0.12
kfr_2	0.048	0.24
kfs_1	$0.0001\,\mathrm{s}^{-1}$	$0.0001\,\mathrm{s}^{-1}$
kfs_2	$0.0001\,\mathrm{s}^{-1}$	$0.0001\,\mathrm{s}^{-1}$
Deactivation $\left[\mathrm{s}^{-1}\right]$		

Continued on next page

Table A.3 – Continued from previous page

VARIABLE	SITE TYPE 1	SITE TYPE 2
kds	0.0001	0.0001

A.4 Catalyst properties

The properties of the catalyst used in the simulations are given in Table A.4.

Table A.4 – Catalyst properties [46].

VARIABLE	VALUE
Mass fraction of titanium on catalyst, ω_{Ti}	0.02
Fraction of titanium that are potential sites, x_{P}	0.4
Fraction of sites that are of type 1, x_{P}^{1}	0.8064
Fraction of sites that are of type 2, x_{P}^{2}	0.1936

Appendix B

Implementation of the plant replacement model

> Always code as if the guy who ends up maintaining your code will be a violent psychopath who knows where you live. Code for readability.
>
> — J. F. Woods, 1991[1]

B.1 Model equations

The implementation of the state derivatives is displayed in Code snippet B.1.

[1] https://groups.google.com/d/msg/comp.lang.c++/rYCO5yn4lXw/oITtSkZOtoUJ

Code snippet b.1 – The implementation of the state derivatives, `model_dx.m`.

```
function xdot = model_dx(~, x, u, par)
    % Model for a polypropylene copolymerization in fluidized bed reactor.
    % A two-phase model is employed for the reactor.
    % 4 Control volumes:
    %   Heat exchanger tube side
    %   Emulsion phase
    %   Bubble phase
    %   Disengagement zone
    %
    % Abbreviations:
    %   hex     Heat exchanger
    %   r       Reactor
    %   fb      Freeboard
    %   e       Emulsion phase
    %   b       Bubble phase
    % Strategy:
    %   Calculate solid distribution
    %   Calculate flows
    %   Calculate reactions
    %   Energy calculations
    %   Apply balance equations
    % States:
    %   n_H2_hex    Moles of hydrogen in the heat exchanger
    %   n_M1_hex    Moles of propene in the heat exchanger
    %   n_M2_hex    Moles of ethene in the heat exchanger
    %   n_I_hex     Moles of inert in the heat exchanger
    %   T_hex       Temperature of the tube side in the heat exchanger
    %   n_H2_fb     Moles of hydrogen in the disengagement zone
    %   n_M1_fb     Moles of propene in the disengagement zone
    %   n_M2_fb     Moles of ethene in the disengagement zone
    %   n_I_fb      Moles of inert in the disengagement zone
    %   T_fb        Temperature in the bubble phase 1
    %   n_H2_b      Moles of hydrogen in the bubble phase
    %   n_M1_b      Moles of propene in the bubble phase
    %   n_M2_b      Moles of ethene in the bubble phase
    %   n_I_b       Moles of inert in the bubble phase
    %   T_b         Temperature in the bubble phase
    %   n_H2_e      Moles of hydrogen in the emulsion phase
    %   n_M1_e      Moles of propene in the emulsion phase
    %   n_M2_e      Moles of ethene in the emulsion phase
    %   n_I_e       Moles of inert in the emulsion phase
    %   T_e         Temperature in the emulsion phase
    %   n_B1        Moles of propene bound in the polymer
    %   n_B2        Moles of ethene bound in the polymer
    %   Y_0_1       0th moment of living polymer produced at site 1
    %   Y_0_2       0th moment of living polymer produced at site 2
    %   Y_1_1       1st moment of living polymer produced at site 1
    %   Y_1_2       1st moment of living polymer produced at site 2
    %   X_0         0th moment of dead polymer produced at both sites
    %   X_1         1st moment of dead polymer produced at both sites
    %   YX_2        2nd moment of living and dead polymer produced at both
    %               sites
    %   m_cat       Total mass of catalyst
    %   i_e_h       Integral of the level offset
    %   i_e_p       Integral of the pressure offset
    %   i_e_T       Integral of the temperature offset
    % Inputs:
    %   f_V_H2_in   Feed rate of hydrogen
    %   f_V_M1_in   Feed rate of propene
    %   f_V_M2_in   Feed rate of ethene
    %   f_V_I_in    Feed rate of inert
    %   f_m_cat_in  Feed rate of catalyst and cocatalyst
    %   T_s         Temperature setpoint
    % Parameters:
    %   g           Gravitational acceleration
    %   V_hex       Volume of heat exchanger
    %   V_t         Total reactor volume
    %   A_r         Cross sectional area of reactor
    %   D_r         Diameter of reactor
    %   eps_mf      Void fraction at minimum fluidization
    %   eps_prod    Void fraction of product stream
    %   k_dp_e      Pressure driven flow rate constant for emulsion phase
    %   k_dp_b      Pressure driven flow rate constant for bubble phase
    %   d_p         Particle diameter
    %   u_0         Superficial velocity
    %   sphericity  Particle sphericity
    %   h_s         Level setpoint
    %   p_s         Pressure setpoint
    %   T_f         Feed temperature
    %   T_ref       Reference temperature
    %   Z           Compressibility factor
    %   rho_cat     Catalyst density
    %   rho_pol     Polymer density
    %   rho_TEAL    TEAL density
    %   w_TEAL      Mass fraction TEAL in catalyst feed
    %   Mw_H2       Molecular mass of hydrogen
    %   Mw_M1       Molecular mass of propylene
    %   Mw_M2       Molecular mass of ethylene
    %   Mw_I        Molecular mass of inert
    %   mu_g        Dynamic viscosity of the gas
    %   rho_g       Density of the gas
    %   D_g         Diffusivity of the gas
    %   k_g         Conductivity of the gas{-1} s^{-1} K^{-1}]
    %   n_site_1    Moles of potential active site 1 per gram catalyst
    %   n_site_2    Moles of potential active site 2 per gram catalyst
    %   R           Universal gas constant
    %   f_V_prod_0  Steady-state product flow
    %   f_V_purge_0 Steady-state purge flow
    %   T_cw_0      Steady-state cooling water temperature
    %   K_h         Level controller gain
    %   K_p         Pressure controller gain
    %   K_T         Temperature controller gain
    %   tau_I_h     Level controller integral time
    %   tau_I_p     Pressure controller integral time
    %   tau_I_T     Temperature controller integral time
    %   c_p_H2      Heat capacity of hydrogen
    %   c_p_M1      Heat capacity of propylene
    %   c_p_M2      Heat capacity of ethylene
    %   c_p_I       Heat capacity of inert
    %   c_p_pol     Specific heat capacity of polymer
    %   c_p_cat     Specific heat capacity of catalyst
    %   c_p_TEAL    Specific heat capacity of TEAL
    %   r_h_1       Heat of reaction for propylene polymerization at T_ref
    %   r_h_2       Heat of reaction for ethylene polymerization at T_ref
    %   r_cp_1      Heat capacity of reaction for propylene polymerization
    %   r_cp_2      Heat capacity of reaction for ethylene polymerization
    %% Extractions
    % Heat exchanger
    n_H2_hex    = x(1);         % [kmol]
    n_M1_hex    = x(2);         % [kmol]
    n_M2_hex    = x(3);         % [kmol]
    n_I_hex     = x(4);         % [kmol]
    T_hex       = x(5);         % [K]
    % Disengagement zone (freeboard/overhead)
    n_H2_fb     = x(6);         % [kmol]
    n_M1_fb     = x(7);         % [kmol]
    n_M2_fb     = x(8);         % [kmol]
    n_I_fb      = x(9);         % [kmol]
    T_fb        = x(10);        % [K]
    % Bubble phase
    n_H2_b      = x(11);        % [kmol]
    n_M1_b      = x(12);        % [kmol]
    n_M2_b      = x(13);        % [kmol]
```

```
n_I_b       = x(14);        % [kmol]
T_b         = x(15);        % [K]
% Emulsion phase
n_H2_e      = x(16);        % [kmol]
n_M1_e      = x(17);        % [kmol]
n_M2_e      = x(18);        % [kmol]
n_I_e       = x(19);        % [kmol]
T_e         = x(20);        % [K]
% Solids in reactor
n_B1        = x(21);        % [kmol]
n_B2        = x(22);        % [kmol]
Y_0_1       = x(23);        % [kmol]
Y_0_2       = x(24);        % [kmol]
Y_1_1       = x(25);        % [kmol]
Y_1_2       = x(26);        % [kmol]
X_0         = x(27);        % [kmol]
X_1         = x(28);        % [kmol]
YX_2        = x(29);        % [kmol]
m_cat       = x(30);        % [kg]
% Integral errors for controllers
i_e_h       = x(31);        % [m s]
i_e_p       = x(32);        % [Pa s]
i_e_T       = x(33);        % [K s]

% Inputs
f_V_H2_in   = u(1);         % [m^{3} s^{-1}]
f_V_M1_in   = u(2);         % [m^{3} s^{-1}]
f_V_M2_in   = u(3);         % [m^{3} s^{-1}]
f_V_I_in    = u(4);         % [m^{3} s^{-1}]
f_m_cat_in  = u(5);         % [kg s^{-1}]
T_s         = u(6);         % [K]

% Parameters
% Universal constants
g           = par.g;            % [m s^{-2}]
% Reactor parameters
V_hex       = par.V_hex;        % [m^{3}]
V_t         = par.V;            % [m^{3}]
A_r         = par.area_cs;      % [m^{2}]
D_r         = par.D_r;          % [m]
eps_mf      = par.eps_mf;       % [-]
eps_prod    = par.eps_prod;     % [-]
k_dp_e      = par.k_dp_e;       % [m^{3} s^{-1} Pa^{-1}]
k_dp_b      = par.k_dp_b;       % [m^{3} s^{-1} Pa^{-1}]
p_s         = par.p_s;          % [Pa]
```

```
    h_s        = par.h_s;           % [m]
    w_TEAL     = par.w_TEAL;        % [-]
    d_p        = par.d_p;           % [m]
    u_0        = par.u_0;           % [m s^{-1}]
    sphericity = par.sphericity;    % [-]
    n_site_1   = par.n_site_1;      % [kmol kg^{-1}]
    n_site_2   = par.n_site_2;      % [kmol kg^{-1}]
    T_f        = par.T_f;           % [K]
    T_ref      = par.T_ref;         % [K]
    % Molecular weights
    Mw_H2      = par.Mw_H2;         % [kg kmol^{-1}]
    Mw_M1      = par.Mw_M1;         % [kg kmol^{-1}]
    Mw_M2      = par.Mw_M2;         % [kg kmol^{-1}]
    Mw_I       = par.Mw_I;          % [kg kmol^{-1}]
    % Densities
    rho_pol    = par.rho_pol;       % [kg m^{-3}]
    rho_cat    = par.rho_cat;       % [kg m^{-3}]
    rho_TEAL   = par.rho_TEAL;      % [kg m^{-3}]
    % Gas parameters
    mu_g       = par.mu_g;          % [Pa s]
    rho_g      = par.rho_g;         % [kg m^{-3}]
    D_g        = par.D_g;           % [m^{2} s^{-1}]
    k_g        = par.k_g;           % [J m^{-1} s^{-1} K^{-1}]
    % Controller parameters
    f_V_prod_0     = par.u0_h;          % [m^{3} s^{-1}]
    K_h            = par.K_h;           % [m^{2} s^{-1}]
    tau_I_h        = par.tau_I_h;       % [s]
    f_V_prod_min   = par.u_h_min;       % [m^{3} s^{-1}]
    f_V_prod_max   = par.u_h_max;       % [m^{3} s^{-1}]
    f_V_purge_0    = par.u0_p;          % [m^{3} s^{-1}]
    K_p            = par.K_p;           % [m^{3} s^{-1} Pa^{-1}]
    tau_I_p        = par.tau_I_p;       % [s]
    f_V_purge_min  = par.u_p_min;       % [m^{3} s^{-1}]
    f_V_purge_max  = par.u_p_max;       % [m^{3} s^{-1}]
    T_cw_0         = par.u0_T;          % [K]
    K_T            = par.K_T;           % [-]
    tau_I_T        = par.tau_I_T;       % [s]
    T_cw_min       = par.u_T_min;       % [K]
    T_cw_max       = par.u_T_max;       % [K]
    % Heat capacities
    c_p_H2  = par.c_p_H2;           % [J kmol^{-1} K^{-1}]
    c_p_M1  = par.c_p_M1;           % [J kmol^{-1} K^{-1}]
    c_p_M2  = par.c_p_M2;           % [J kmol^{-1} K^{-1}]
    c_p_I   = par.c_p_I;            % [J kmol^{-1} K^{-1}]
    c_p_pol = par.c_p_pol;          % [J kg^{-1} K^{-1}]
    c_p_cat = par.c_p_cat;          % [J kg^{-1} K^{-1}]
    c_p_TEAL= par.c_p_TEAL;         % [J kg^{-1} K^{-1}]
    % Heat of reactions
    r_h_1   = par.r_h_1;            % [J kmol^{-1}]
    r_h_2   = par.r_h_2;            % [J kmol^{-1}]
    r_cp_1  = par.r_cp_1;           % [J kmol^{-1} K^{-1}]
    r_cp_2  = par.r_cp_2;           % [J kmol^{-1} K^{-1}]
    %% Calculation of solid distribution
    % Mass of solids
    m_pol   = Mw_M1*n_B1 + Mw_M2*n_B2;
    m_TEAL  = w_TEAL*m_cat;
    m_c     = m_cat - m_TEAL;
    % Volume of solids
    V_pol   = m_pol/rho_pol;
    V_cat   = m_c/rho_cat;
    V_TEAL  = m_TEAL/rho_TEAL;
    V_s     = V_pol + V_cat + V_TEAL;
    % Average density of solids
    rho_s   = (m_pol + m_cat)/V_s;
    % Minimum fluidization velocity
    Ar      = rho_g*(rho_s - rho_g)*g*d_p^3/mu_g^2;
    Re_mf   = sqrt(870.25 + 0.375*Ar) - 29.5;
    u_mf    = mu_g/(rho_g*d_p)*Re_mf;

    eps_b   = 1 - 0.146*exp(-(u_0 - u_mf)/4.439);
    eps_e   = eps_mf + 0.2 - 0.059*exp(-(u_0-u_mf)/0.429);
    delta   = 0.534*(1 - exp(-(u_0 - u_mf)/0.413));
    % Average void fraction
    eps_avg = delta*eps_b + (1-delta)*eps_e;

    % Height of bed
    h   = V_s/((1-eps_avg)*A_r);
    % Volume of bed
    V_bed   = h*A_r;
    % Emulsion phase volume
    V_e     = (1 - delta)*V_bed;
    % Bubble phase volume
    V_b     = V_bed - V_e;

    % Distribution
    [Y_0_1_e, Y_0_1_b]  = distributeSolids(delta, eps_b, eps_e, Y_0_1);
    [Y_0_2_e, Y_0_2_b]  = distributeSolids(delta, eps_b, eps_e, Y_0_2);
    [Y_1_1_e, Y_1_1_b]  = distributeSolids(delta, eps_b, eps_e, Y_1_1);
    [Y_1_2_e, Y_1_2_b]  = distributeSolids(delta, eps_b, eps_e, Y_1_2);
    [m_cat_e, m_cat_b]  = distributeSolids(delta, eps_b, eps_e, m_cat);
```

```
[m_pol_e, m_pol_b]  = distributeSolids(delta, eps_b, eps_e, m_pol);
% Inflow of catalyst to each control volume
[f_m_cat_f_e, f_m_cat_f_b] = distributeSolids(delta, eps_b, ...
                                              eps_e, f_m_cat_in);
% Inflow of potential sites on catalyst
f_n_P_1_in    = f_m_cat_in*(1-w_TEAL)*n_site_1;  % site 1 [kmol s^{-1}]
f_n_P_2_in    = f_m_cat_in*(1-w_TEAL)*n_site_2;  % site 2 [kmol s^{-1}]
[f_n_P_1_in_e, f_n_P_1_in_b] = distributeSolids(delta, eps_b, ...
                                                eps_e, f_n_P_1_in);
[f_n_P_2_in_e, f_n_P_2_in_b] = distributeSolids(delta, eps_b, ...
                                                eps_e, f_n_P_2_in);
%% Gas concentration and pressure calculations
% Gas concentrations in disengagement zone
V_fb = V_t - V_hex - V_bed;      % Volume of disengagement zone [m^{3}]
[c_M1_fb, c_M2_fb, c_H2_fb, c_I_fb, c_g_fb] = getGasConc(n_M1_fb, ...
                                                         n_M2_fb, ...
                                                         n_H2_fb, ...
                                                         n_I_fb, ...
                                                         V_fb);
% Gas concentrations in heat exchanger
[c_M1_hex, c_M2_hex, c_H2_hex, c_I_hex, ...
c_g_hex] = getGasConc(n_M1_hex, n_M2_hex, n_H2_hex, n_I_hex, V_hex);

% Gas concentrations in emulsion phase
[c_M1_e, c_M2_e, c_H2_e, c_I_e, ~] = getGasConc(n_M1_e, ...
                                                n_M2_e, ...
                                                n_H2_e, ...
                                                n_I_e, ...
                                                V_e*eps_e);
% Gas concentrations in bubble phase
[c_M1_b, c_M2_b, c_H2_b, c_I_b, ~] = getGasConc(n_M1_b, ...
                                                n_M2_b, ...
                                                n_H2_b, ...
                                                n_I_b, ...
                                                V_b*eps_b);
% Pressure in reactor
[p_fb,~]= getPressure(c_H2_fb, c_M1_fb, c_M2_fb, c_I_fb, T_fb, par);
   [p_b, ~]= getPressure(c_H2_b, c_M1_b, c_M2_b, c_I_b, T_b, par);
[p_e, ~]= getPressure(c_H2_e, c_M1_e, c_M2_e, c_I_e, T_e, par);
%% Flow calculations
% Outflow from heat exchanger is given by the superficial velocity
f_V_hex_r       = u_0*A_r;
f_n_H2_hex_r    = c_H2_hex*f_V_hex_r;
f_n_M1_hex_r    = c_M1_hex*f_V_hex_r;
f_n_M2_hex_r    = c_M2_hex*f_V_hex_r;
```

```
f_n_I_hex_r     = c_I_hex*f_V_hex_r;
% Gas feed into the heat exchanger
f_n_H2_f_hex    = c_g_hex*f_V_H2_in;
f_n_M1_f_hex    = c_g_hex*f_V_M1_in;
f_n_M2_f_hex    = c_g_hex*f_V_M2_in;
f_n_I_f_hex     = c_g_hex*f_V_I_in;
% Recycle stream is given by a stationary total mass balance for hex
% Total flow from heat exchanger to reactor
f_n_hex_r   = f_n_H2_hex_r + f_n_M1_hex_r ...
            + f_n_M2_hex_r + f_n_I_hex_r;               % [kmol s^{-1}]
% Total flow from feed to heat exchanger
f_n_f_hex   = f_n_H2_f_hex + f_n_M1_f_hex ...
            + f_n_M2_f_hex + f_n_I_f_hex;               % [kmol s^{-1}]

% Mole flow of recycle from disengagement zone to heat exchanger
f_n_fb_hex      = f_n_hex_r - f_n_f_hex;            % [kg s^{-1}]
f_V_fb_hex      = f_n_fb_hex/c_g_fb;                % [m^{3} s^{-1}]
f_n_H2_fb_hex   = c_H2_fb*f_V_fb_hex;               % [kmol s^{-1}]
f_n_M1_fb_hex   = c_M1_fb*f_V_fb_hex;               % [kmol s^{-1}]
f_n_M2_fb_hex   = c_M2_fb*f_V_fb_hex;               % [kmol s^{-1}]
f_n_I_fb_hex    = c_I_fb*f_V_fb_hex;                % [kmol s^{-1}]

% Calculation of reactor inlet flows
f_n_H2_hex_e    = (1-delta)*f_n_H2_hex_r;
f_n_M1_hex_e    = (1-delta)*f_n_M1_hex_r;
f_n_M2_hex_e    = (1-delta)*f_n_M2_hex_r;
f_n_I_hex_e     = (1-delta)*f_n_I_hex_r;
f_n_H2_hex_b    = f_n_H2_hex_r - f_n_H2_hex_e;
f_n_M1_hex_b    = f_n_M1_hex_r - f_n_M1_hex_e;
f_n_M2_hex_b    = f_n_M2_hex_r - f_n_M2_hex_e;
f_n_I_hex_b     = f_n_I_hex_r - f_n_I_hex_e;

% Internal reactor flow calculations, given by pressure differences
[f_n_H2_b_fb, f_n_M1_b_fb, ...
 f_n_M2_b_fb, f_n_I_b_fb, ...
~] = getReactorFlows(c_H2_b, c_M1_b, c_M2_b, c_I_b, p_b, ...
                     c_H2_fb, c_M1_fb, c_M2_fb, c_I_fb, p_fb, k_dp_b);
[f_n_H2_e_fb, f_n_M1_e_fb, ...
 f_n_M2_e_fb, f_n_I_e_fb, ...
 f_V_e_fb] = getReactorFlows(c_H2_e, c_M1_e, c_M2_e, ...
                             c_I_e, p_e, c_H2_fb, ...
                             c_M1_fb, c_M2_fb, ...
                             c_I_fb, p_fb, k_dp_e);
%% Flow controllers
% Pressure controller
```

```
    [f_V_fb_purge, e_p] = PIcontrol(p_fb, p_s, f_V_purge_0, K_p, ...
                                    tau_I_p, i_e_p, ...
                                    f_V_purge_min, f_V_purge_max);
    % Purge flows
    f_n_H2_fb_purge   = c_H2_fb*f_V_fb_purge;             % [kmol s^{-1}]
    f_n_M1_fb_purge   = c_M1_fb*f_V_fb_purge;             % [kmol s^{-1}]
    f_n_M2_fb_purge   = c_M2_fb*f_V_fb_purge;             % [kmol s^{-1}]
    f_n_I_fb_purge    = c_I_fb*f_V_fb_purge;              % [kmol s^{-1}]
    % Level controller
    [f_V_r_prod, e_h] = PIcontrol(h, h_s, f_V_prod_0, K_h, ...
                                  tau_I_h, i_e_h, ...
                                  f_V_prod_min, f_V_prod_max);
    % Emulsion gas in product stream
    f_n_H2_e_prod   = eps_prod*c_H2_e*(1-delta)*f_V_r_prod;
    f_n_M1_e_prod   = eps_prod*c_M1_e*(1-delta)*f_V_r_prod;
    f_n_M2_e_prod   = eps_prod*c_M2_e*(1-delta)*f_V_r_prod;
    f_n_I_e_prod    = eps_prod*c_I_e*(1-delta)*f_V_r_prod;
    % Bubble gas in product stream
    f_n_H2_b_prod   = eps_prod*c_H2_b*delta*f_V_r_prod;
    f_n_M1_b_prod   = eps_prod*c_M1_b*delta*f_V_r_prod;
    f_n_M2_b_prod   = eps_prod*c_M2_b*delta*f_V_r_prod;
    f_n_I_b_prod    = eps_prod*c_I_b*delta*f_V_r_prod;
    % Solids in product stream
    tau_prod    = V_s/f_V_r_prod;   % Product flow "Time constant" [s^{-1}]
    f_n_B1_prod = n_B1/tau_prod;    % Bound C3= product flow [kmol s^{-1}]
    f_n_B2_prod = n_B2/tau_prod;    % Bound C2= product flow [kmol s^{-1}]
    f_Y_0_1_prod= Y_0_1/tau_prod;   % Y_0_1 product flow [kmol s^{-1}]
    f_Y_0_2_prod= Y_0_2/tau_prod;   % Y_0_2 product flow [kmol s^{-1}]
    f_Y_1_1_prod= Y_1_1/tau_prod;   % Y_1_1 product flow [kmol s^{-1}]
    f_Y_1_2_prod= Y_1_2/tau_prod;   % Y_1_2 product flow [kmol s^{-1}]
    f_X_0_prod  = X_0/tau_prod;     % X_0 product flow [kmol s^{-1}]
    f_X_1_prod  = X_1/tau_prod;     % X_1 product flow [kmol s^{-1}]
    f_YX_2_prod = YX_2/tau_prod;    % YX_2 product flow [kmol s^{-1}]
    f_m_cat_prod= m_cat/tau_prod;   % Catalyst product flow [kg s^{-1}]

    % Temperature controller
    [T_cw, e_T] = PIcontrol(T_fb, T_s, T_cw_0, K_T, tau_I_T, ...
                            i_e_T, T_cw_min, T_cw_max);

    %% Hydrodynamic calculations
    % Densities
    rho_g_b     = getGasDensity(c_H2_b, c_M1_b, c_M2_b, c_I_b, ...
                                Mw_H2, Mw_M1, Mw_M2, Mw_I);
    % Bubble diameters
    d_b     = getBubbleDiameter(0.5*h, mu_g, rho_g_b, rho_s, g, ...
                                d_p, sphericity, u_0, u_mf, D_r);
    % Bubble rise velocities
    u_br        = 1.6*sqrt(g*d_b);
    % Emulsion phase velocity as average between in and out velocities
    u_e         = 0.5*(u_0 + f_V_e_fb/((1-delta)*A_r));

    % Mass transfer coefficients
    K_be        = getMassTransferCoeff(u_e, d_b, D_g, g, u_br, eps_e);
    % Heat transfer coefficients
    c_p_g_b     = convertHeatCapacity(rho_g_b, c_H2_b, c_M1_b, c_M2_b, ...
                                      c_I_b, c_p_H2, c_p_M1, c_p_M2, ...
                                      c_p_I);
    H_be        = getHeatTransferCoeff(u_e, d_b, rho_g_b,...
                                       c_p_g_b, k_g, g, u_br, eps_e);
    % Molar flows between emulsion and bubble phases
    f_n_H2_b_e  = getInterfacialFlow(c_H2_e, c_H2_b, K_be, V_b);
    f_n_M1_b_e  = getInterfacialFlow(c_M1_e, c_M1_b, K_be, V_b);
    f_n_M2_b_e  = getInterfacialFlow(c_M2_e, c_M2_b, K_be, V_b);
    f_n_I_b_e   = getInterfacialFlow(c_I_e, c_I_b, K_be, V_b);
    %% Reaction calculations
    [r_n_H2_e, r_n_M1_e, r_n_M2_e, ...
     r_n_B1_e, r_n_B2_e, r_Y_0_1_e, ...
     r_Y_0_2_e, r_Y_1_1_e, r_Y_1_2_e, ...
     r_X_0_e, r_X_1_e, r_YX_2_e] = getReactionRates(c_H2_e, c_M1_e, ...
                                                    c_M2_e, Y_0_1_e, ...
                                                    Y_0_2_e, Y_1_1_e, ...
                                                    Y_1_2_e, T_e, ...
                                                    V_s, m_cat, ...
                                                    tau_prod, ...
                                                    f_n_P_1_in_e, ...
                                                    f_n_P_2_in_e, par);
    [r_n_H2_b, r_n_M1_b, r_n_M2_b, ...
     r_n_B1_b, r_n_B2_b, r_Y_0_1_b, ...
     r_Y_0_2_b, r_Y_1_1_b, r_Y_1_2_b, ...
     r_X_0_b, r_X_1_b, r_YX_2_b] = getReactionRates(c_H2_b, c_M1_b, ...
                                                    c_M2_b, Y_0_1_b, ...
                                                    Y_0_2_b, Y_1_1_b, ...
                                                    Y_1_2_b, T_b, ...
                                                    V_s, ...
                                                    m_cat, tau_prod, ...
                                                    f_n_P_1_in_b, ...
                                                    f_n_P_2_in_b, par);
    %% Energy calculations
    % Heat exchanger
    % Heat capacity of heat exchanger contents [J K^{-1}]
    C_p_hex = n_H2_hex*c_p_H2 + n_M1_hex*c_p_M1 ...
            + n_M2_hex*c_p_M2 + n_I_hex*c_p_I;
    % Heating of feed [J s^{-1}]
    f_H_f_hex   = (f_n_H2_f_hex*c_p_H2 + f_n_M1_f_hex*c_p_M1 ...
                   + f_n_M2_f_hex*c_p_M2 + f_n_I_f_hex*c_p_I) ...
                *(T_f - T_hex);
    % Heating of recycle [J s^{-1}]
    f_H_fb_hex  = (f_n_H2_fb_hex*c_p_H2 + f_n_M1_fb_hex*c_p_M1 ...
                   + f_n_M2_fb_hex*c_p_M2 + f_n_I_fb_hex*c_p_I) ...
                *(T_fb - T_hex);

    % Heat transfer
    q_hex_cw = getHeatTransfer(T_hex, T_fb, T_cw, f_H_f_hex, ...
                               f_H_fb_hex, par);

    % Disengagement zone
    % Heat capacity of freeboard contents
    C_p_fb      = n_H2_fb*c_p_H2 + n_M1_fb*c_p_M1 ...
                + n_M2_fb*c_p_M2 + n_I_fb*c_p_I;
    % Heating of flow from bubble phase
    f_H_b_fb    = (f_n_H2_b_fb*c_p_H2 + f_n_M1_b_fb*c_p_M1 ...
                  + f_n_M2_b_fb*c_p_M2 + f_n_I_b_fb*c_p_I)*(T_b - T_fb);
    % Heating of flow from emulsion phase
    f_H_e_fb    = (f_n_H2_e_fb*c_p_H2 + f_n_M1_e_fb*c_p_M1 ...
                  + f_n_M2_e_fb*c_p_M2 + f_n_I_e_fb*c_p_I)*(T_e - T_fb);

    % Bubble phase
    % Heat capacity of bubble phase 1 contents
    C_p_b   = n_H2_b*c_p_H2 + n_M1_b*c_p_M1 + n_M2_b*c_p_M2 ...
            + n_I_b*c_p_I + m_pol_b*c_p_pol ...
            + m_cat_b*((1-w_TEAL)*c_p_cat + w_TEAL*c_p_TEAL);
    % Heating of gas from heat exchanger [J s^{-1}]
    f_H_hex_b   = (f_n_H2_hex_b*c_p_H2 + f_n_M1_hex_b*c_p_M1 ...
                + f_n_M2_hex_b*c_p_M2 + f_n_I_hex_b*c_p_I)*(T_hex - T_b);
    % Heating of catalyst feed [J s^{-1}]
    f_H_f_b     = f_m_cat_f_b*((1-w_TEAL)*c_p_cat + w_TEAL*c_p_TEAL) ...
                * (T_f - T_b);
    % Heat of reactions [J s^{-1}]
    r_H_1_b    = (r_h_1 + r_cp_1*(T_b - T_ref))*r_n_B1_b;
    r_H_2_b    = (r_h_2 + r_cp_2*(T_b - T_ref))*r_n_B2_b;

    % Emulsion phase
    % Heat capacity of emulsion phase contents
    C_p_e   = n_H2_e*c_p_H2 + n_M1_e*c_p_M1 + n_M2_e*c_p_M2 ...
            + n_I_e*c_p_I + m_pol_e*c_p_pol ...
            + m_cat_e*((1-w_TEAL)*c_p_cat + w_TEAL*c_p_TEAL);
    % Heating of gas from heat exchanger [J s^{-1}]
    f_H_hex_e   = (f_n_H2_hex_e*c_p_H2 + f_n_M1_hex_e*c_p_M1 ...
                + f_n_M2_hex_e*c_p_M2 + f_n_I_hex_e*c_p_I)*(T_hex - T_e);
    % Heating of gas from bubble phases
    f_H_b_e     = getInterfacialFlow(T_e, T_b, H_be, V_b);
    % Heating of catalyst feed [J s^{-1}]
    f_H_f_e = f_m_cat_f_e*((1-w_TEAL)*c_p_cat + w_TEAL*c_p_TEAL) ...
            * (T_f - T_e);
    % Heat of reactions [J s^{-1}]
    r_H_1_e = (r_h_1 + r_cp_1*(T_e - T_ref))*r_n_B1_e;
    r_H_2_e = (r_h_2 + r_cp_2*(T_e - T_ref))*r_n_B2_e;
    %% Gas phase mole balances
    % Heat exchanger
    d_n_H2_hex  = f_n_H2_fb_hex + f_n_H2_f_hex - f_n_H2_hex_r;
    d_n_M1_hex  = f_n_M1_fb_hex + f_n_M1_f_hex - f_n_M1_hex_r;
    d_n_M2_hex  = f_n_M2_fb_hex + f_n_M2_f_hex - f_n_M2_hex_r;
    d_n_I_hex   = f_n_I_fb_hex + f_n_I_f_hex - f_n_I_hex_r;
    % Disengagement zone
    d_n_H2_fb   = f_n_H2_b_fb + f_n_H2_e_fb ...
                - f_n_H2_fb_hex - f_n_H2_fb_purge;
    d_n_M1_fb   = f_n_M1_b_fb + f_n_M1_e_fb ...
                - f_n_M1_fb_hex - f_n_M1_fb_purge;
    d_n_M2_fb   = f_n_M2_b_fb + f_n_M2_e_fb ...
                - f_n_M2_fb_hex - f_n_M2_fb_purge;
    d_n_I_fb    = f_n_I_b_fb + f_n_I_e_fb ...
                - f_n_I_fb_hex - f_n_I_fb_purge;
    % Bubble phase
    d_n_H2_b    = f_n_H2_hex_b - f_n_H2_b_e - f_n_H2_b_fb ...
                - f_n_H2_b_prod + r_n_H2_b;
    d_n_M1_b    = f_n_M1_hex_b - f_n_M1_b_e - f_n_M1_b_fb ...
                - f_n_M1_b_prod + r_n_M1_b;
    d_n_M2_b    = f_n_M2_hex_b - f_n_M2_b_e - f_n_M2_b_fb ...
                - f_n_M2_b_prod + r_n_M2_b;
    d_n_I_b     = f_n_I_hex_b - f_n_I_b_e - f_n_I_b_fb - f_n_I_b_prod;
    % Emulsion phase
    d_n_H2_e    = f_n_H2_hex_e + f_n_H2_b_e - f_n_H2_e_fb ...
                - f_n_H2_e_prod + r_n_H2_e;
    d_n_M1_e    = f_n_M1_hex_e + f_n_M1_b_e - f_n_M1_e_fb ...
                - f_n_M1_e_prod + r_n_M1_e;
    d_n_M2_e    = f_n_M2_hex_e + f_n_M2_b_e - f_n_M2_e_fb ...
                - f_n_M2_e_prod + r_n_M2_e;
    d_n_I_e     = f_n_I_hex_e + f_n_I_b_e - f_n_I_e_fb - f_n_I_e_prod;
    %% Energy balances
    % Heat exchanger
    d_T_hex = (f_H_fb_hex + f_H_f_hex - q_hex_cw)/C_p_hex;
    % Disengagement zone
    d_T_fb  = (f_H_b_fb + f_H_e_fb)/C_p_fb;
    % Bubble phase 1
    d_T_b  = (f_H_f_b + f_H_hex_b - r_H_1_b - r_H_2_b)/C_p_b;
    % Bubble phase 2
    % Emulsion phase
    d_T_e   = (f_H_f_e + f_H_hex_e + f_H_b_e  - r_H_1_e - r_H_2_e)/C_p_e;
    %% Solid balances
    % Catalyst mass balance
    d_m_cat = f_m_cat_in - f_m_cat_prod;
    % Bound monomers balance
    d_n_B1  = r_n_B1_b  + r_n_B1_e - f_n_B1_prod;
    d_n_B2  = r_n_B2_b  + r_n_B2_e - f_n_B2_prod;
    % Moments equations
    d_Y_0_1 = r_Y_0_1_b + r_Y_0_1_e - f_Y_0_1_prod;
    d_Y_0_2 = r_Y_0_2_b + r_Y_0_2_e - f_Y_0_2_prod;
    d_Y_1_1 = r_Y_1_1_b + r_Y_1_1_e - f_Y_1_1_prod;
    d_Y_1_2 = r_Y_1_2_b + r_Y_1_2_e - f_Y_1_2_prod;
    d_X_0   = r_X_0_b + r_X_0_e - f_X_0_prod;
    d_X_1   = r_X_1_b + r_X_1_e - f_X_1_prod;
    d_YX_2  = r_YX_2_b + r_YX_2_e - f_YX_2_prod;
    %% Insertions
    xdot    = zeros(size(x));
    % Heat exchanger
    xdot(1) = d_n_H2_hex;
    xdot(2) = d_n_M1_hex;
    xdot(3) = d_n_M2_hex;
    xdot(4) = d_n_I_hex;
    xdot(5) = d_T_hex;
    % Disengagement zone
    xdot(6) = d_n_H2_fb;
    xdot(7) = d_n_M1_fb;
    xdot(8) = d_n_M2_fb;
    xdot(9) = d_n_I_fb;
    xdot(10)= d_T_fb;
    % Bubble phase
    xdot(11)    = d_n_H2_b;
    xdot(12)    = d_n_M1_b;
    xdot(13)    = d_n_M2_b;
    xdot(14)    = d_n_I_b;
    xdot(15)    = d_T_b;
    % Emulsion phase
    xdot(16)    = d_n_H2_e;
    xdot(17)    = d_n_M1_e;
    xdot(18)    = d_n_M2_e;
    xdot(19)    = d_n_I_e;
    xdot(20)    = d_T_e;
    % Solids
    xdot(21)    = d_n_B1;
    xdot(22)    = d_n_B2;
    xdot(23)    = d_Y_0_1;
    xdot(24)    = d_Y_0_2;
    xdot(25)    = d_Y_1_1;
    xdot(26)    = d_Y_1_2;
    xdot(27)    = d_X_0;
    xdot(28)    = d_X_1;
    xdot(29)    = d_YX_2;
    xdot(30)    = d_m_cat;
    % Integral errors
    xdot(31)    = e_h;
    xdot(32)    = e_p;
    xdot(33)    = e_T;
end
```

The implementation of the measurements is presented in Code snippet B.2.

Code snippet B.2 – The implementation of the measurements, model_y.m.

```
function y = model_y(x, u, par)
    %% Measurements
    % States:
    %   n_H2_hex    Moles of hydrogen in the heat exchanger
    %   n_M1_hex    Moles of propene in the heat exchanger
    %   n_M2_hex    Moles of ethene in the heat exchanger
    %   n_I_hex     Moles of inert in the heat exchanger
    %   T_hex       Temperature of the tube side in the heat exchanger
    %   n_H2_fb     Moles of hydrogen in the disengagement zone
    %   n_M1_fb     Moles of propene in the disengagement zone
    %   n_M2_fb     Moles of ethene in the disengagement zone
    %   n_I_fb      Moles of inert in the disengagement zone
    %   T_fb        Temperature in the bubble phase 1
    %   n_H2_b      Moles of hydrogen in the bubble phase
    %   n_M1_b      Moles of propene in the bubble phase
    %   n_M2_b      Moles of ethene in the bubble phase
    %   n_I_b       Moles of inert in the bubble phase
    %   T_b         Temperature in the bubble phase
    %   n_H2_e      Moles of hydrogen in the emulsion phase
    %   n_M1_e      Moles of propene in the emulsion phase
    %   n_M2_e      Moles of ethene in the emulsion phase
    %   n_I_e       Moles of inert in the emulsion phase
    %   T_e         Temperature in the emulsion phase
    %   n_B1        Moles of propene bound in the polymer
    %   n_B2        Moles of ethene bound in the polymer
    %   Y_0_1       0th moment of living polymer produced at site 1
    %   Y_0_2       0th moment of living polymer produced at site 2
    %   Y_1_1       1st moment of living polymer produced at site 1
    %   Y_1_2       1st moment of living polymer produced at site 2
    %   X_0         0th moment of dead polymer produced at both sites
    %   X_1         1st moment of dead polymer produced at both sites
    %   YX_2        2nd moment of living and dead polymer produced at both
    %               sites
    %   m_cat       Total mass of catalyst
    %   i_e_h       Integral of the level offset
    %   i_e_p       Integral of the pressure offset
    %   i_e_T       Integral of the temperature offset
    % Inputs:
    %   f_V_H2_in   Feed rate of hydrogen
    %   f_V_M1_in   Feed rate of propene
    %   f_V_M2_in   Feed rate of ethene
    %   f_V_I_in    Feed rate of inert
    %   f_m_cat_in  Feed rate of catalyst and cocatalyst
    %   T_s         Temperature setpoint
    % Parameters:
    %   g           Gravitational acceleration
    %   V_hex       Volume of heat exchanger
    %   V_t         Total reactor volume
    %   A_r         Cross sectional area of reactor
    %   D_r            Diameter of reactor
    %   eps_mf      Void fraction at minimum fluidization
    %   eps_prod    Void fraction of product stream
    %   k_dp_e      Pressure driven flow rate constant for emulsion phase
    %   k_dp_b      Pressure driven flow rate constant for bubble phase
    %   d_p         Particle diameter
    %   u_0         Superficial velocity
    %   sphericity  Particle sphericity
    %   h_s         Level setpoint
    %   p_s         Pressure setpoint
    %   T_f         Feed temperature
    %   T_ref       Reference temperature
    %   rho_cat     Catalyst density
    %   rho_pol     Polymer density
    %   rho_TEAL    TEAL density
    %   w_TEAL      Mass fraction TEAL in catalyst feed
    %   Mw_M1       Molecular mass of propylene
    %   Mw_M2       Molecular mass of ethylene
    %   mu_g        Dynamic viscosity of the gas
    %   rho_g       Density of the gas
    %   f_V_prod_0  Steady-state product flow
    %   f_V_purge_0 Steady-state purge flow
    %   T_cw_0      Steady-state cooling water temperature
    %   K_h         Level controller gain
    %   K_p         Pressure controller gain
    %   K_T         Temperature controller gain
    %   tau_I_h     Level controller integral time
    %   tau_I_p     Pressure controller integral time
    %   tau_I_T     Temperature controller integral time
    % Measurements:
    %   x_H2_fb     Mole fraction of hydrogen in the disengagement zone
    %   x_M1_fb     Mole fraction of propylene in the disengagement zone
    %   x_M2_fb     Mole fraction of ethylene in the disengagement zone
    %   x_I_fb      Mole fraction of nitrogen in the disengagement zone
    %   p_fb        Pressure in the disengagement zone
    %   h           Level
    %   T_fb        Temperature in the disengagement zone
    %   f_V_purge   Purge rate
    %   f_V_prod    Product rate
    %   T_cw        Cooling water temperature
    %   MI          Melt index
    %% Extractions
    % Disengagement zone (freeboard/overhead)
    n_H2_fb     = x(6);         % [kmol]
    n_M1_fb     = x(7);         % [kmol]
    n_M2_fb     = x(8);         % [kmol]
    n_I_fb      = x(9);         % [kmol]
    T_fb        = x(10);        % [K]
    % Solids in reactor
    n_B1        = x(21);        % [kmol]
    n_B2        = x(22);        % [kmol]
    Y_1_1       = x(25);          % [kmol]
    Y_1_2       = x(26);        % [kmol]
    X_1         = x(28);        % [kmol]
    YX_2        = x(29);        % [kmol]
    m_cat       = x(30);        % [kg]
    % Integral errors for controllers
    i_e_h       = x(31);        % [m s]
    i_e_p       = x(32);        % [Pa s]
    i_e_T       = x(33);        % [K s]
```

```

    % Inputs
    T_s         = u(6);         % [K]

    % Parameters
    % Universal constants
    g           = par.g;            % [m s^{-2}]
    % Reactor parameters
    V_hex       = par.V_hex;        % [m^{3}]
    V_t         = par.V;            % [m^{3}]
    A_r         = par.area_cs;      % [m^{2}]
    eps_mf      = par.eps_mf;       % [-]
    p_s         = par.p_s;          % [Pa]
    h_s         = par.h_s;          % [m]
    w_TEAL      = par.w_TEAL;       % [-]
    d_p         = par.d_p;          % [m]
    u_0         = par.u_0;          % [m s^{-1}]
    % Molecular weights
    Mw_M1       = par.Mw_M1;        % [kg kmol^{-1}]
    Mw_M2       = par.Mw_M2;        % [kg kmol^{-1}]
    % Densities
    rho_pol     = par.rho_pol;      % [kg m^{-3}]
    rho_cat     = par.rho_cat;      % [kg m^{-3}]
    rho_TEAL    = par.rho_TEAL;     % [kg m^{-3}]
    % Gas parameters
    mu_g        = par.mu_g;         % [Pa s]
    rho_g       = par.rho_g;        % [kg m^{-3}]
    % Controller parameters
    f_V_prod_0      = par.u0_h;         % [m^{3} s^{-1}]
    K_h             = par.K_h;          % [m^{2} s^{-1}]
    tau_I_h         = par.tau_I_h;      % [s]
    f_V_prod_min    = par.u_h_min;      % [m^{3} s^{-1}]
    f_V_prod_max    = par.u_h_max;      % [m^{3} s^{-1}]
    f_V_purge_0     = par.u0_p;         % [m^{3} s^{-1}]
    K_p             = par.K_p;          % [m^{3} s^{-1} Pa^{-1}]
    tau_I_p         = par.tau_I_p;      % [s]
    f_V_purge_min   = par.u_p_min;      % [m^{3} s^{-1}]
    f_V_purge_max   = par.u_p_max;      % [m^{3} s^{-1}]
    T_cw_0          = par.u0_T;         % [K]
    K_T             = par.K_T;          % [-]
    tau_I_T         = par.tau_I_T;      % [s]
    T_cw_min        = par.u_T_min;      % [K]
    T_cw_max        = par.u_T_max;      % [K]
    %% Calculation of solid distribution
    % Mass of solids
```

```
m_pol   = Mw_M1*n_B1 + Mw_M2*n_B2;
m_TEAL  = w_TEAL*m_cat;
m_c     = m_cat - m_TEAL;
% Volume of solids
V_pol   = m_pol/rho_pol;
V_cat   = m_c/rho_cat;
V_TEAL  = m_TEAL/rho_TEAL;
V_s     = V_pol + V_cat + V_TEAL;
% Average density of solids
rho_s   = (m_pol + m_cat)/V_s;
% Minimum fluidization velocity
Ar      = rho_g*(rho_s - rho_g)*g*d_p^3/mu_g^2;
Re_mf   = sqrt(870.25 + 0.375*Ar) - 29.5;
u_mf    = mu_g/(rho_g*d_p)*Re_mf;

eps_b   = 1 - 0.146*exp(-(u_0 - u_mf)/4.439);
eps_e   = eps_mf + 0.2 - 0.059*exp(-(u_0-u_mf)/0.429);
delta   = 0.534*(1 - exp(-(u_0 - u_mf)/0.413));
% Average void fraction
eps_avg = delta*eps_b + (1-delta)*eps_e;

% Height of bed
h   = V_s/((1-eps_avg)*A_r);
% Volume of bed
V_bed   = h*A_r;

%% Gas concentration and pressure calculations
% Gas concentrations in disengagement zone
V_fb = V_t - V_hex - V_bed;      % Volume of disengagement zone [m^{3}]
[c_M1_fb, c_M2_fb, c_H2_fb, c_I_fb, c_g_fb] = getGasConc(n_M1_fb, ...
                                                         n_M2_fb, ...
                                                         n_H2_fb, ...
                                                         n_I_fb, ...
                                                         V_fb);
[x_M1_fb, x_M2_fb, x_H2_fb, ~, ~] = getGasConc(c_M1_fb, ...
                                                    c_M2_fb, ...
                                                    c_H2_fb, ...
                                                    c_I_fb, ...
                                                    c_g_fb);

% Pressure in disengagement zone
[p_fb, ~]   = getPressure(c_H2_fb, c_M1_fb, c_M2_fb, ...
                          c_I_fb, T_fb, par);


```

```
    %% Controllers
    % Pressure controller
    [f_V_purge, ~] = PIcontrol(p_fb, p_s, f_V_purge_0,  K_p, tau_I_p, ...
                               i_e_p, f_V_purge_min, f_V_purge_max);
    % Level controller
    [f_V_prod, ~] = PIcontrol(h, h_s, f_V_prod_0, K_h, tau_I_h, ...
                              i_e_h, f_V_prod_min, f_V_prod_max);
    % Temperature controller
    [T_cw, ~] = PIcontrol(T_fb, T_s, T_cw_0, K_T, tau_I_T, ...
                          i_e_T, T_cw_min, T_cw_max);

    % Molecular masses
    % Average monomer molecular weight
    Mw_M = (Mw_M1*n_B1 + Mw_M2*n_B2)/(n_B1 + n_B2);        % [kg kmol^{-1}]
    % Mass average polymer molecular weight
    Mw_w = Mw_M*(YX_2/(X_1 + Y_1_1 + Y_1_2));              % [kg kmol^{-1}]

    MI   = 3.3542e17*Mw_w^-3.472;                   % Melt index [dg / min]
    %% Insertions
    y       = zeros(10,1);
    y(1)    = x_H2_fb;
    y(2)    = x_M1_fb;
    y(3)    = x_M2_fb;
    y(4)    = p_fb;
    y(5)    = h;
    y(6)    = T_fb;
    y(7)    = f_V_purge;
    y(8)    = f_V_prod;
    y(9)    = T_cw;
    y(10)   = MI;
end
```

B.2 Integration of the model equations

To demonstrate how the model equations can be integrated in Matlab, a sample script has been provided in Code snippet B.3.

Code snippet B.3 – The main script for integrating the model equations, main.m.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  Name     : main
%  Function : Sample main script to integrate the models in Matlab
%  Method   : ode15s (backwards difference)
%  Author   : Kasper J. Linnestad
%  Modified : 25.5.2015
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

[x0, u0, par, y0, z0]= init_model();                            % Initialize
n_y                  = length(y0);                 % Number of measurements
n_z                  = length(z0);                      % Number of outputs
[x0, y0, z0, par]    = steady_state(x0, u0, par, 5e6);% Obtain steady-state

% Apply step in hydrogen feed
u   = u0;
u(1)= 1.1*u0(1);
% Integrate for 24 hours
[t, x] = ode15s(@(t,x) model_dx(t, x, u, par), [0, 24*3600], x0)
% Calculate measurements and outputs
[y, z] = calculate_y_z(t, x, u, par, n_y, n_z);
```

The initialization of the plant replacement model (PRM) is shown in Code snippet B.4.

Code snippet B.4 – The function that returns the initial values for the plant replacement model (PRM), init_model.m.

```
function [x0, u0, par, y0, z0] = init_model(u0)
    cwd = cd;
    cd('../matlab_well-mixed');
    if nargin < 1
        [x0, u0, par, ~, ~] = init_model;
    else
        [x0, ~, par, ~, ~] = init_model;
    end
    [x_ss, ~, ~, ~] = steady_state(x0, u0, par, 1e10);
    cd(cwd);
    par = init_par(par);
    x0  = init_x(x_ss, par);
    y0  = model_y(x0, u0, par);
    z0  = model_z(x0, u0, y0, par);
    cd(cwd);
end

function par = init_par(par)
    par.g               = 9.81;     % Gravitational acceleration [m s^{-2}]
    par.V_hex           = 25;       % Heat exchanger volume [m^{3}]
    par.D_r             = 5;        % Reactor radius [m]
    par.k_dp_e          = 2e-4;     % "Valve const" [m^{3} s^{-1} Pa^{-1}]
    par.k_dp_b          = 1e-4;     % "Valve const" [m^{3} s^{-1} Pa^{-1}]
    par.d_p             = 1000e-6;  % Particle diameter [m]
    par.sphericity      = 1;        % Particle sphericity [-]
    par.mu_g            = 1.14e-4;  % Gas viscosity [Pa s]
    par.rho_g           = 45;       % Gas density
    par.D_g             = 4e-7;     % Gas diffusivity [m^{2} s^{-1}]
    par.k_g             = 0.0318;   % Gas conductivity [J m^{-1}s^{-1}K^{-1}]
    par.UA              = 80000;    % Heat exchanger coefficient [W K^{-1}]
    par.u_0             = 0.57;     % Superficial gas velocity
    % Redlich-Kwong parameters
    par.a_H2            = par.R^2*33.15^(5/2)/(9*(2^(1/3)-1)*1.2964e6);
    par.a_M1            = par.R^2*364.9^(5/2)/(9*(2^(1/3)-1)*4.59e6);
    par.a_M2            = par.R^2*282.35^(5/2)/(9*(2^(1/3)-1)*5.06e6);
    par.a_I             = par.R^2*126.192^(5/2)/(9*(2^(1/3)-1)*3.39e6);
    par.b_H2            = (2^(1/3)-1)*par.R*33.15/(3*1.2964e6);
    par.b_M1            = (2^(1/3)-1)*par.R*364.9/(3*4.59e6);
    par.b_M2            = (2^(1/3)-1)*par.R*282.35/(3*5.06e6);
    par.b_I             = (2^(1/3)-1)*par.R*126.192/(3*3.39e6);
    par.chi_H2          = 1.0;
    par.chi_M1_1        = 1.0;
    par.chi_M1_2        = 1.0;
    par.chi_M2_1        = 1.0;
    par.chi_M2_2        = 1.0;
end

function x = init_x(x0, par)
    % Extract states from well-mixed model
    n_H2    = x0(1);
    n_M1    = x0(2);
    n_M2    = x0(3);
    n_I     = x0(4);
    n_B1    = x0(5);
    n_B2    = x0(6);
    Y_0_1   = x0(7);
    Y_0_2   = x0(8);
    Y_1_1   = x0(9);
    Y_1_2   = x0(10);
    X_0     = x0(11);
    X_1     = x0(12);
    YX_2    = x0(13);
    m_cat   = x0(14);
    T       = x0(15);
    % Calculate volumes
    V_hex   = par.V_hex;            % Heat exchanger volume
    V_t     = par.V;                % Total volume
    V_bed   = par.area_cs*par.h_s;  % Bed volume
    V_fb    = V_t - V_hex - V_bed;  % Freeboard volume
    delta   = 0.33;                 % Bubble phase fraction
    V_e     = (1-delta)*V_bed;      % Emulsion phase volume
    V_b     = delta*V_bed;          % Bubble phases volume
    w_hex   = 1.35*V_hex/V_t;       % Heat exchanger fraction
                                    % (adjusted for correct pressure)
    w_fb    = V_fb/V_t;             % Freeboard fraction
    w_e     = V_e/V_t;              % Emulsion phase fraction
    w_b     = V_b/V_t;              % Bubble phase fraction
    % Heat exchanger
    n_H2_hex    = n_H2*w_hex;
    n_M1_hex    = n_M1*w_hex;
    n_M2_hex    = n_M2*w_hex;
    n_I_hex     = n_I*w_hex;
    T_hex       = T;
    % Disengagement zone (freeboard/overhead)
    n_H2_fb     = n_H2*w_fb;
    n_M1_fb     = n_M1*w_fb;
    n_M2_fb     = n_M2*w_fb;
    n_I_fb      = n_I*w_fb;
    T_fb        = T;
    % Bubble phase
    n_H2_b     = n_H2*w_b;
    n_M1_b     = n_M1*w_b;
    n_M2_b     = n_M2*w_b;
    n_I_b      = n_I*w_b;
    T_b        = T;
    % Emulsion phase
    n_H2_e      = n_H2*w_e;
    n_M1_e      = n_M1*w_e;
    n_M2_e      = n_M2*w_e;
    n_I_e       = n_I*w_e;
    T_e         = T;
    x = zeros(33,1);
    % Heat exchanger
    x(1) = n_H2_hex;
    x(2) = n_M1_hex;
    x(3) = n_M2_hex;
    x(4) = n_I_hex;
    x(5) = T_hex;
    % Disengagement zone
    x(6) = n_H2_fb;
    x(7) = n_M1_fb;
    x(8) = n_M2_fb;
    x(9) = n_I_fb;
    x(10)= T_fb;
    % Bubble phase
    x(11)    = n_H2_b;
    x(12)    = n_M1_b;
    x(13)    = n_M2_b;
    x(14)    = n_I_b;
    x(15)    = T_b;
    % Emulsion phase
    x(16)    = n_H2_e;
    x(17)    = n_M1_e;
    x(18)    = n_M2_e;
    x(19)    = n_I_e;
    x(20)    = T_e;
    % Solids
    x(21)    = n_B1;
    x(22)    = n_B2;
    x(23)    = Y_0_1;
    x(24)    = Y_0_2;
    x(25)    = Y_1_1;
    x(26)    = Y_1_2;
    x(27)    = X_0;
    x(28)    = X_1;
    x(29)    = YX_2;
    x(30)    = m_cat;
    % Integral errors
    x(31)    = 0;
    x(32)    = 0;
    x(33)    = 0;
end
```

B.3 Runge-Kutta implementation

The implementation of the fourth-order Runge-Kutte integration scheme is presented in Code snippet B.5.

Code snippet B.5 – The implementation of a fourth-order Runge-Kutta integration scheme [56], ode4.C.

```
/***************************************************************************
*  Name     : model_x_rk4
*  Function : Integration of the state vector
*  Method   : Runge-Kutta 4th order method
*  Author   : Kasper J. Linnestad
*  Modified : 28.4.2015
**************************************************************************/

int model_x_rk4             // Out: Error code (0 = OK!)
(
    double *xs,             // Out: New state
    double *xs_old,         // In: Old state
    double *par,            // In: Parameter vector
    double *con,            // In: Constant vector
    double *u,              // In: Process inputs
    double *dcalcvar,       // In: Internal variable calculated by model
    int    *icalcvar,       // In: Internal variable calculated by model
    double dt               // In: Sampling time
)
{
    int    i, n, k, r = 0;
    double dxs1[NS], dxs2[NS], dxs3[NS], dxs4[NS];
    double xs_tmp[NS];
    double dtn;
    n = (int) ceil(dt/DT_INT);
    dtn   = dt/n;
    for (i = 0; i < NS; i++)
        xs[i] = xs_old[i];
    for (k = 0; k < n; k++)
    {
        if (!r)
            r = model_dx
            (
                dxs1,
                xs,
                par,
                con,
                u,
                dcalcvar,
                icalcvar,
                dtn
            );
        for (i = 0; i < NS; i++)
            xs_tmp[i] = xs[i] + dxs1[i] * dtn / 2.0;
        if (!r)
            r = model_dx
            (
                dxs2,
                xs_tmp,
                par,
                con,
                u,
                dcalcvar,
                icalcvar,
                dtn
            );
        for (i = 0; i < NS; i++)
            xs_tmp[i] = xs[i] + dxs2[i] * dtn / 2.0;
        if (!r)
            r = model_dx
            (
                dxs3,
                xs_tmp,
                par,
                con,
                u,
                dcalcvar,
                icalcvar,
                dtn
            );
        for (i = 0; i < NS; i++)
            xs_tmp[i] = xs[i] + dxs3[i] * dtn;
        if (!r)
            r = model_dx
            (
                dxs4,
                xs_tmp,
                par,
                con,
                u,
                dcalcvar,
                icalcvar,
                dtn
            );
        for (i = 0; i < NS; i++)
            xs[i] += (dxs1[i] + 2.0*dxs2[i] + 2.0*dxs3[i] + dxs4[i])
                   * dtn / 6.0;
    }
    return r;
}
```

APPENDIX C

IMPLEMENTATION OF THE CONTROL MODEL

> Everyone knows that debugging is twice as hard as writing a program in the first place. So if you're as clever as you can be when you write it, how will you ever debug it?
>
> — B. W. KERNIGHAN & P. J. PLAUGER, 1978[1]

C.1 MODEL EQUATIONS

The implementation of the state derivatives is displayed in Code snippet C.1.

[1] Kernighan, B. W. and Plauger, P. J., *The Elements of Programming Style*. 1978

Code snippet c.1 – The implementation of the state derivatives, `model_dx.m`.

```
function xdot = model_dx(~, x, u, par)
    %% Model for a fluidized bed polypropylene reactor
    % The pseudo steady-state assumption (PSSA) is used for the moles of
    % potential active sites, uninitiated sites produced by monomers and
    % uninitiated sites produced by H2 transfer, n_P, n_0 and n_H
    % respectively.
    % States:
    %   n_H2    Moles of hydrogen
    %   n_M1    Moles of propylene (C3=, monomer 1)
    %   n_M2    Moles of ethylene (C2=, monomer 2)
    %   n_I     Moles of inert (N2)
    %   n_B1    Moles of reacted propylene bound in the polymer
    %   n_B2    Moles of reacted ethylene bound in the polymer
    %   Y_0_1   0th moment of chain length distribution for living polymer
    %           produced at site 1
    %   Y_0_2   0th moment of chain length distribution for living polymer
    %           produced at site 2
    %   Y_1_1   1st moment of chain length distribution for living polymer
    %           produced at site 1
    %   Y_1_2   1st moment of chain length distribution for living polymer
    %           produced at site 2
    %   X_0     0th moment of chain length distribution for dead polymer
    %           produced at site 1 and 2
    %   X_1     1st moment of chain length distribution for dead polymer
    %           produced at site 1 and 2
    %   YX_2    2nd moment of chain length distribution for living and
    %           dead polymer produced at site 1 and 2
    %   m_cat   Mass of catalyst and cocatalyst
    %   T       Reactor temperature
    %   i_e_h   Integral of the setpoint error in the level
    %   i_e_p   Integral of the setpoint error in the pressure
    %   i_e_T   Integral of the setpoint error in the temperature
    % Inputs:
    %   f_V_H2_in   Hydrogen feed
    %   f_V_M1_in   Propylene feed
    %   f_V_M2_in   Ethylene feed
    %   f_V_I_in    Inert feed
    %   f_m_cat_in  Catalyst feed
    % Parameters:
    %   h_s         Level setpoint
    %   p_s         Pressure setpoint
    %   T_f         Feed temperature
    %   T_ref       Reference temperature
    %   Z           Compressibility factor
    %   rho_cat     Catalyst density
    %   rho_pol     Polymer density
    %   rho_TEAL    TEAL density
    %   eps_avg     Average void fraction
    %   eps_prod    Void fraction of product stream
    %   area_cs     Cross sectional area
    %   w_TEAL      Mass fraction TEAL in catalyst feed
    %   Mw_M1       Molecular mass of propylene
    %   Mw_M2       Molecular mass of ethylene
    %   n_site_1    Moles of potential active site 1 per gram catalyst
    %   n_site_2    Moles of potential active site 2 per gram catalyst
    %   V           Volume
    %   R            Universal gas constant
    %   UA          Heat transfer coefficient times heat transfer area
    %   f_V_prod_0  Steady-state product flow
    %   f_V_purge_0 Steady-state purge flow
    %   T_cw_0      Steady-state cooling water temperature
    %   K_h         Level controller gain
    %   K_p         Pressure controller gain
    %   K_T         Temperature controller gain
    %   tau_I_h     Level controller integral time
    %   tau_I_p     Pressure controller integral time
    %   tau_I_T     Temperature controller integral time
    %   c_p_H2      Heat capacity of hydrogen
    %   c_p_M1      Heat capacity of propylene
    %   c_p_M2      Heat capacity of ethylene
    %   c_p_I       Heat capacity of inert
    %   c_p_pol     Specific heat capacity of polymer
    %   c_p_cat     Specific heat capacity of catalyst
    %   c_p_TEAL    Specific heat capacity of TEAL
    %   r_h_1       Heat of reaction for propylene polymerization at T_ref
    %   r_h_2       Heat of reaction for ethylene polymerization at T_ref
    %   r_cp_1      Heat capacity of reaction for propylene polymerization
    %   r_cp_2      Heat capacity of reaction for ethylene polymerization
    %% Extractions
    % States
    n_H2    = x(1);         % [kmol]
    n_M1    = x(2);         % [kmol]
    n_M2    = x(3);         % [kmol]
    n_I     = x(4);         % [kmol]
    n_B1    = x(5);         % [kmol]
    n_B2    = x(6);         % [kmol]
    Y_0_1   = x(7);         % [kmol]
    Y_0_2   = x(8);         % [kmol]
    Y_1_1   = x(9);         % [kmol]
```

```
    Y_1_2   = x(10);        % [kmol]
    X_0     = x(11);        % [kmol]
    X_1     = x(12);        % [kmol]
    YX_2    = x(13);        % [kmol]
    m_cat   = x(14);        % [kg]
    T       = x(15);        % [K]
    i_e_h   = x(16);        % [m s]
    i_e_p   = x(17);        % [Pa s]
    i_e_T   = x(18);        % [K s]

    % Inputs:
    f_V_H2_in   = u(1);         % [m^{3} s^{-1}]
    f_V_M1_in   = u(2);         % [m^{3} s^{-1}]
    f_V_M2_in   = u(3);         % [m^{3} s^{-1}]
    f_V_I_in    = u(4);         % [m^{3} s^{-1}]
    f_m_cat_in  = u(5);         % [kg s^{-1}]
    T_s         = u(6);         % [K]

    % Parameters
    % General
    h_s         = par.h_s;          % [m]
    p_s         = par.p_s;          % [Pa]
    T_f         = par.T_f;          % [K]
    T_ref       = par.T_ref;        % [K]
    Z           = par.Z;            % [-]
    rho_cat     = par.rho_cat;      % [kg m^{-3}]
    rho_pol     = par.rho_pol;      % [kg m^{-3}]
    rho_TEAL    = par.rho_TEAL;     % [kg m^{-3}]
    eps_avg     = par.eps_avg;      % [-]
    eps_prod    = par.eps_prod;     % [-]
    area_cs     = par.area_cs;      % [m^{2}]
    w_TEAL      = par.w_TEAL;       % [-]
    Mw_M1       = par.Mw_M1;        % [kg kmol^{-1}]
    Mw_M2       = par.Mw_M2;        % [kg kmol^{-1}]
    n_site_1    = par.n_site_1;     % [kmol kg^{-1}]
    n_site_2    = par.n_site_2;     % [kmol kg^{-1}]
    V           = par.V;            % [m^{3}]
    R           = par.R;            % [kJ kmol^{-1} K^{-1}]
    UA          = par.UA_wm;        % [W K^{-1}]

    % Controllers
    f_V_prod_0      = par.u0_h;         % [m^{3} s^{-1}]
    K_h             = par.K_h;          % [m^{2} s^{-1}]
    tau_I_h         = par.tau_I_h;      % [s]
    f_V_prod_min    = par.u_h_min;      % [m^{3} s^{-1}]
```

```
f_V_prod_max    = par.u_h_max;      % [m^{3} s^{-1}]
f_V_purge_0     = par.u0_p;         % [m^{3} s^{-1}]
K_p             = par.K_p;          % [m^{3} s^{-1} Pa^{-1}]
tau_I_p         = par.tau_I_p;      % [s]
f_V_purge_min   = par.u_p_min;      % [m^{3} s^{-1}]
f_V_purge_max   = par.u_p_max;      % [m^{3} s^{-1}]
T_cw_0          = par.u0_T;         % [K]
K_T             = par.K_T;          % [-]
tau_I_T         = par.tau_I_T;      % [s]
T_cw_min        = par.u_T_min;      % [K]
T_cw_max        = par.u_T_max;      % [K]

% Heat capacities
c_p_H2  = par.c_p_H2;   % [J kmol^{-1} K^{-1}]
c_p_M1  = par.c_p_M1;   % [J kmol^{-1} K^{-1}]
c_p_M2  = par.c_p_M2;   % [J kmol^{-1} K^{-1}]
c_p_I   = par.c_p_I;    % [J kmol^{-1} K^{-1}]
c_p_pol = par.c_p_pol;  % [J kg^{-1} K^{-1}]
c_p_cat = par.c_p_cat;  % [J kg^{-1} K^{-1}]
c_p_TEAL= par.c_p_TEAL; % [J kg^{-1} K^{-1}]

% Heat of reactions
r_h_1   = par.r_h_1;    % [J kmol^{-1}]
r_h_2   = par.r_h_2;    % [J kmol^{-1}]
r_cp_1  = par.r_cp_1;   % [J kmol^{-1} K^{-1}]
r_cp_2  = par.r_cp_2;   % [J kmol^{-1} K^{-1}]

%% Calculations
% Outflow of polymer product is controlled by a PI level controller
m_pol           = Mw_M1*n_B1 + Mw_M2*n_B2;         % Polymer mass [kg]
V_s             = m_pol/rho_pol ...             % Solid volume [m^{3}]
                + m_cat*((1-w_TEAL)/rho_cat + w_TEAL/rho_TEAL);
h               = V_s/((1 - eps_avg)*area_cs);             % Level [m]
% Level controller
[f_V_prod, e_h] = PIcontrol(h, h_s, f_V_prod_0, K_h, ...
                            tau_I_h, i_e_h, ...
                            f_V_prod_min, f_V_prod_max);
% Gas concentrations [kmol m^{-3}]
[c_M1, c_M2, c_H2, c_I, c_g] = getGasConc(n_M1, n_M2, n_H2, ...
                                          n_I, V - V_s);

% Inflow of potential sites on catalyst
f_n_P_1_in    = f_m_cat_in*(1-w_TEAL)*n_site_1; % site 1 [kmol s^{-1}]
f_n_P_2_in    = f_m_cat_in*(1-w_TEAL)*n_site_2; % site 2 [kmol s^{-1}]
% Inflow of gases
```

```
f_n_H2_in   = c_g*f_V_H2_in;                        % H2 [kmol s^{-1}]
f_n_M1_in   = c_g*f_V_M1_in;                       % C3= [kmol s^{-1}]
f_n_M2_in   = c_g*f_V_M2_in;                       % C2= [kmol s^{-1}]
f_n_I_in    = c_g*f_V_I_in;                          % I [kmol s^{-1}]

% Gas in product flow
f_n_M1_prod = eps_prod*c_M1*f_V_prod; % C3= product flow [kmol s^{-1}]
f_n_M2_prod = eps_prod*c_M2*f_V_prod; % C2= product flow [kmol s^{-1}]
f_n_H2_prod = eps_prod*c_H2*f_V_prod;  % H2 product flow [kmol s^{-1}]
f_n_I_prod  = eps_prod*c_I*f_V_prod;% Inert product flow [kmol s^{-1}]

% Solids in product flow
tau_prod        = V_s/f_V_prod;% Product flow "Time constant" [s^{-1}]
f_n_B1_prod     = n_B1/tau_prod;% Bound C3= product flow [kmol s^{-1}]
f_n_B2_prod     = n_B2/tau_prod;% Bound C2= product flow [kmol s^{-1}]
f_Y_0_1_prod    = Y_0_1/tau_prod;   % Y_0_1 product flow [kmol s^{-1}]
f_Y_0_2_prod    = Y_0_2/tau_prod;   % Y_0_2 product flow [kmol s^{-1}]
f_Y_1_1_prod    = Y_1_1/tau_prod;   % Y_1_1 product flow [kmol s^{-1}]
f_Y_1_2_prod    = Y_1_2/tau_prod;   % Y_1_2 product flow [kmol s^{-1}]
f_X_0_prod      = X_0/tau_prod;       % X_0 product flow [kmol s^{-1}]
f_X_1_prod      = X_1/tau_prod;       % X_1 product flow [kmol s^{-1}]
f_YX_2_prod     = YX_2/tau_prod;     % YX_2 product flow [kmol s^{-1}]
f_m_cat_prod    = m_cat/tau_prod;  % Catalyst product flow [kg s^{-1}]

% Purge is controlled by a PI pressure controller
p    = Z*c_g*R*T;                                       % Pressure [Pa]
[f_V_purge, e_p] = PIcontrol(p, p_s, f_V_purge_0, K_p, ...
                             tau_I_p, i_e_p, ...
                             f_V_purge_min, f_V_purge_max);
f_n_H2_purge    = c_H2*f_V_purge;             % H2 purge [kmol s^{-1}]
f_n_M1_purge    = c_M1*f_V_purge;            % C3= purge [kmol s^{-1}]
f_n_M2_purge    = c_M2*f_V_purge;            % C2= purge [kmol s^{-1}]
f_n_I_purge     = c_I*f_V_purge;           % Inert purge [kmol s^{-1}]

%% Reaction rates
[r_n_H2, r_n_M1, r_n_M2, ...
 r_n_B1, r_n_B2, r_Y_0_1, ...
 r_Y_0_2, r_Y_1_1, r_Y_1_2, ...
 r_X_0, r_X_1, r_YX_2] = getReactionRates(c_H2, c_M1, ...
                                          c_M2, Y_0_1, ...
                                          Y_0_2, Y_1_1, ...
                                          Y_1_2, T, ...
                                          V_s, ...
                                          m_cat, tau_prod, ...
                                          f_n_P_1_in, ...
```

```
                                              f_n_P_2_in, par);

    % Temperature control
    [T_cw, e_T] = PIcontrol(T, T_s, T_cw_0, K_T, tau_I_T, ...
                            i_e_T, T_cw_min, T_cw_max);
    q           = UA*(T_cw - T);                % Heat transfer [J s^{-1}]
    % Energy balance
    % Heat capacity of reactor contents [J K^{-1}]
    C_p     = n_H2*c_p_H2 + n_M1*c_p_M1 + n_M2*c_p_M2 + n_I*c_p_I ...
            + m_pol*c_p_pol ...
            + m_cat*((1-w_TEAL)*c_p_cat + w_TEAL*c_p_TEAL);
    % Heating of feed [J s^{-1}]
    f_H_in  = (f_n_H2_in*c_p_H2 + f_n_M1_in*c_p_M1 + f_n_M2_in*c_p_M2 ...
               + f_n_I_in*c_p_I ...
               + f_m_cat_in*((1-w_TEAL)*c_p_cat + w_TEAL*c_p_TEAL))...
            *(T_f - T);
    % Heat of reactions [J s^{-1}]
    r_H_1   = (r_h_1 + r_cp_1*(T - T_ref))*r_n_B1;
    r_H_2   = (r_h_2 + r_cp_2*(T - T_ref))*r_n_B2;
    %% Derivatives
    % Mole balances for gases (reaction in emulsion phase only)
    d_n_H2  = f_n_H2_in - f_n_H2_purge - f_n_H2_prod + r_n_H2;
    d_n_M1  = f_n_M1_in - f_n_M1_purge - f_n_M1_prod + r_n_M1;
    d_n_M2  = f_n_M2_in - f_n_M2_purge - f_n_M2_prod + r_n_M2;
    d_n_I   = f_n_I_in - f_n_I_purge - f_n_I_prod;
    % Mole balances for bound monomer
    d_n_B1  = -f_n_B1_prod + r_n_B1;
    d_n_B2  = -f_n_B2_prod + r_n_B2;
    % Mass balance for catalyst and cocatalyst
    d_m_cat = f_m_cat_in - f_m_cat_prod;
    % Distribution balances
    d_Y_0_1 = -f_Y_0_1_prod + r_Y_0_1;
    d_Y_0_2 = -f_Y_0_2_prod + r_Y_0_2;
    d_Y_1_1 = -f_Y_1_1_prod + r_Y_1_1;
    d_Y_1_2 = -f_Y_1_2_prod + r_Y_1_2;
    d_X_0   = -f_X_0_prod + r_X_0;
    d_X_1   = -f_X_1_prod + r_X_1;
    d_YX_2  = -f_YX_2_prod + r_YX_2;
    % Energy balance
    d_T     = (f_H_in - r_H_1 - r_H_2 + q)/C_p;
    %% Insertions
    xdot        = zeros(size(x));
    xdot(1)     = d_n_H2;
    xdot(2)     = d_n_M1;
    xdot(3)     = d_n_M2;
    xdot(4)     = d_n_I;
    xdot(5)     = d_n_B1;
    xdot(6)     = d_n_B2;
    xdot(7)     = d_Y_0_1;
    xdot(8)     = d_Y_0_2;
    xdot(9)     = d_Y_1_1;
    xdot(10)    = d_Y_1_2;
    xdot(11)    = d_X_0;
    xdot(12)    = d_X_1;
    xdot(13)    = d_YX_2;
    xdot(14)    = d_m_cat;
    xdot(15)    = d_T;
    xdot(16)    = e_h;
    xdot(17)    = e_p;
    xdot(18)    = e_T;
end
```

The implementation of the measurements is presented in Code snippet c.2.

Code snippet c.2 – The implementation of the measurements, model_y.m.

```
function y = model_y(x, u, par)
    %% Measurement estimation
    % States:
    %   n_H2    Moles of hydrogen
    %   n_M1    Moles of propylene (C3=, monomer 1)
    %   n_M2    Moles of ethylene (C2=, monomer 2)
    %   n_I     Moles of inert (N2)
    %   n_B1    Moles of reacted propylene bound in the polymer
    %   n_B2    Moles of reacted ethylene bound in the polymer
    %   Y_1_1   1st moment of chain length distribution for living polymer
    %           produced at site 1
    %   Y_1_2   1st moment of chain length distribution for living polymer
    %           produced at site 2
    %   X_1     1st moment of chain length distribution for dead polymer
    %           produced at site 1 and 2
    %   YX_2    2nd moment of chain length distribution for living and
    %           dead polymer produced at site 1 and 2
    %   m_cat   Mass of catalyst and cocatalyst
    %   T       Temperature
    %   i_e_h   Integral of the setpoint error in the level
    %   i_e_p   Integral of the setpoint error in the pressure
    %   i_e_T   Integral of the setpoint error in the temperature
    % Inputs:
    %   T_s     Temperature setpoint
    % Parameters:
    %   h_s         Level setpoint
    %   p_s         Pressure setpoint
    %   Z           Compressibility factor
    %   rho_cat     Catalyst density
    %   rho_pol     Polymer density
    %   rho_TEAL    TEAL density
    %   eps_avg     Average void fraction
    %   area_cs     Cross sectional area
    %   w_TEAL      Mass fraction TEAL in catalyst feed
    %   Mw_M1       Molecular mass of propylene
    %   Mw_M2       Molecular mass of ethylene
    %   V           Volume
    %   R           Universal gas constant
    %   f_V_prod_0  Steady-state product flow
    %   f_V_purge_0 Steady-state purge flow
    %   T_cw_0      Steady-state cooling water temperature
    %   K_h         Level controller gain
    %   K_p         Pressure controller gain
    %   K_T         Temperature controller gain
    %   tau_I_h     Level controller integral time
    %   tau_I_p     Pressure controller integral time
    %   tau_I_T     Temperature controller integral time
    % Measurements:
    %   x_H2        Mole fractions of hydrogen
    %   x_M1        Mole fractions of propylene
    %   x_M2        Mole fractions of ethylene
    %   x_I         Mole fractions of nitrogen
    %   p           Pressure
    %   h           Level
    %   T           Temperature
    %   f_V_purge   Purge rate
    %   f_V_prod    Product rate
    %   T_cw        Cooling water temperature
    %   MI          Melt index
    %% Extractions
    % States
    n_H2    = x(1);         % [kmol]
    n_M1    = x(2);         % [kmol]
    n_M2    = x(3);         % [kmol]
    n_I     = x(4);         % [kmol]
```

```
    n_B1    = x(5);         % [kmol]
    n_B2    = x(6);         % [kmol]
    Y_1_1   = x(9);         % [kmol]
    Y_1_2   = x(10);        % [kmol]
    X_1     = x(12);        % [kmol]
    YX_2    = x(13);        % [kmol]
    m_cat   = x(14);        % [kg]
    T       = x(15);        % [K]
    i_e_h   = x(16);        % [m s]
    i_e_p   = x(17);        % [Pa s]
    i_e_T   = x(18);        % [K s]

    % Inputs
    T_s     = u(6);         % [K]

    % Parameters
    h_s         = par.h_s;          % [m]
    p_s         = par.p_s;          % [Pa]
    Z           = par.Z;            % [-]
    rho_cat     = par.rho_cat;      % [kg m^{-3}]
    rho_pol     = par.rho_pol;      % [kg m^{-3}]
    rho_TEAL    = par.rho_TEAL;     % [kg m^{-3}]
    eps_avg     = par.eps_avg;      % [-]
    area_cs     = par.area_cs;      % [m^{2}]
    w_TEAL      = par.w_TEAL;       % [-]
    Mw_M1       = par.Mw_M1;        % [kg kmol^{-1}]
    Mw_M2       = par.Mw_M2;        % [kg kmol^{-1}]
    V           = par.V;            % [m^{3}]
    R           = par.R;            % [kJ kmol^{-1} K^{-1}]
    % Controller settings
    f_V_prod_0      = par.u0_h;         % [m^{3} s^{-1}]
    K_h             = par.K_h;          % [m^{2} s^{-1}]
    tau_I_h         = par.tau_I_h;      % [s]
    f_V_prod_min    = par.u_h_min;      % [m^{3} s^{-1}]
    f_V_prod_max    = par.u_h_max;      % [m^{3} s^{-1}]
    f_V_purge_0     = par.u0_p;         % [m^{3} s^{-1}]
    K_p             = par.K_p;          % [m^{3} s^{-1} Pa^{-1}]
    tau_I_p         = par.tau_I_p;      % [s]
    f_V_purge_min   = par.u_p_min;      % [m^{3} s^{-1}]
    f_V_purge_max   = par.u_p_max;      % [m^{3} s^{-1}]
    T_cw_0          = par.u0_T;         % [K]
    K_T             = par.K_T;          % [-]
    tau_I_T         = par.tau_I_T;      % [s]
    T_cw_min        = par.u_T_min;      % [K]
    T_cw_max        = par.u_T_max;      % [K]
```

```
%% Calculations
% Level
m_pol           = Mw_M1*n_B1 + Mw_M2*n_B2;          % Polymer mass [kg]
V_s             = m_pol/rho_pol ...              % Solid volume [m^{3}]
                + m_cat*((1-w_TEAL)/rho_cat + w_TEAL/rho_TEAL);
h               = V_s/((1 - eps_avg)*area_cs);      % Level [m]
% Level controller
[f_V_prod, ~] = PIcontrol(h, h_s, f_V_prod_0, K_h, ...
                          tau_I_h, i_e_h, f_V_prod_min, f_V_prod_max);

% Gas concentrations [kmol m^{-3}]
[c_M1, c_M2, c_H2, c_I, c_g] = getGasConc(n_M1, n_M2, n_H2, n_I, ...
                                          V - V_s);
% Mole fractions
x_H2                         = c_H2/c_g;
x_M1                         = c_M1/c_g;
x_M2                         = c_M2/c_g;
x_I                          = c_I/c_g;

% Pressure [Pa]
p              = Z*c_g*R*T;
[f_V_purge, ~] = PIcontrol(p, p_s, f_V_purge_0, K_p, ...
                           tau_I_p, i_e_p, ...
                           f_V_purge_min, f_V_purge_max);

% Temperature
[T_cw, ~] = PIcontrol(T, T_s, T_cw_0, K_T, tau_I_T, ...
                      i_e_T, T_cw_min, T_cw_max);

% Average monomer molecular weight in polymer
Mw_M = (Mw_M1*n_B1 + Mw_M2*n_B2)/(n_B1 + n_B2);        % [kg kmol^{-1}]
% Mass average polymer molecular weight
Mw_w = Mw_M*(YX_2/(X_1 + Y_1_1 + Y_1_2));              % [kg kmol^{-1}]
MI   = 3.3542e17*Mw_w^-3.472;                    % Melt index[dg / min]
%% Insertions
y       = zeros(10,1);
y(1)    = x_H2;             % [-]
y(2)    = x_M1;             % [-]
y(3)    = x_M2;             % [-]
y(4)    = p;                % [Pa]
y(5)    = h;                % [m]
y(6)    = T;                % [K]
y(7)    = f_V_purge;        % [m^{3} s^{-1}]
y(8)    = f_V_prod;         % [m^{3} s^{-1}]
y(9)    = T_cw;             % [K]
```

```
    y(10)   = MI;               % [g / 10 min]
end
```

The implementation of the derived outputs is presented in Code snippet c.3.

Code snippet c.3 – The implementation of the derived outputs, `model_z.m`.

```
function z = model_z(x, ~, ~, par)
    %% Measurement estimation
    % States:
    %   n_H2    Moles of hydrogen
    %   n_M1    Moles of propylene (C3=, monomer 1)
    %   n_M2    Moles of ethylene (C2=, monomer 2)
    %   n_I     Moles of inert (N2)
    %   n_B1    Moles of reacted propylene bound in the polymer
    %   n_B2    Moles of reacted ethylene bound in the polymer
    %   Y_1_1   1st moment of chain length distribution for living polymer
    %           produced at site 1
    %   Y_1_2   1st moment of chain length distribution for living polymer
    %           produced at site 2
    %   X_1     1st moment of chain length distribution for dead polymer
    %           produced at site 1 and 2
    %   YX_2    2nd moment of chain length distribution for living and
    %           dead polymer produced at site 1 and 2
    %   m_cat   Mass of catalyst and cocatalyst
    %   T       Temperature
    %   i_e_h   Integral of the setpoint error in the level
    %   i_e_p   Integral of the setpoint error in the pressure
    %   i_e_T   Integral of the setpoint error in the temperature
    % Inputs:
    %   T_s     Temperature setpoint
    % Parameters:
    %   h_s         Level setpoint
    %   p_s         Pressure setpoint
    %   Z           Compressibility factor
    %   rho_cat     Catalyst density
    %   rho_pol     Polymer density
    %   rho_TEAL    TEAL density
    %   eps_avg     Average void fraction
    %   area_cs     Cross sectional area
    %   w_TEAL      Mass fraction TEAL in catalyst feed
    %   Mw_M1       Molecular mass of propylene
    %   Mw_M2       Molecular mass of ethylene
    %   V           Volume
    %   R            Universal gas constant
    %   f_V_prod_0  Steady-state product flow
    %   f_V_purge_0 Steady-state purge flow
    %   T_cw_0      Steady-state cooling water temperature
    %   K_h         Level controller gain
    %   K_p         Pressure controller gain
    %   K_T         Temperature controller gain
    %   tau_I_h     Level controller integral time
    %   tau_I_p     Pressure controller integral time
    %   tau_I_T     Temperature controller integral time
    % Measurements:
    %   x_H2        Mole fractions of hydrogen
    %   x_M1        Mole fractions of propylene
    %   x_M2        Mole fractions of ethylene
    %   x_I         Mole fractions of nitrogen
    %   p           Pressure
    %   h           Level
    %   T           Temperature
    %   f_V_purge   Purge rate
    %   f_V_prod    Product rate
    %   T_cw        Cooling water temperature
    %   MI          Melt index
    %% Extractions
    % States
    n_H2    = x(1);         % [kmol]
    n_M1    = x(2);         % [kmol]
    n_M2    = x(3);         % [kmol]
    n_I     = x(4);         % [kmol]
    n_B1    = x(5);         % [kmol]
    n_B2    = x(6);         % [kmol]
    Y_0_1   = x(7);         % [kmol]
    Y_0_2   = x(8);         % [kmol]
    Y_1_1   = x(9);         % [kmol]
    Y_1_2   = x(10);        % [kmol]
    X_0     = x(11);        % [kmol]
    X_1     = x(12);        % [kmol]
    YX_2    = x(13);        % [kmol]
    m_cat   = x(14);        % [kg]
    T       = x(15);        % [K]
    i_e_h   = x(16);        % [m s]
    i_e_p   = x(17);        % [Pa s]

```

```
    % Parameters
    h_s         = par.h_s;          % [m]
    p_s         = par.p_s;          % [Pa]
    Z           = par.Z;            % [-]
    rho_cat     = par.rho_cat;      % [kg m^{-3}]
    rho_pol     = par.rho_pol;      % [kg m^{-3}]
    rho_TEAL    = par.rho_TEAL;     % [kg m^{-3}]
    eps_avg     = par.eps_avg;      % [-]
    area_cs     = par.area_cs;      % [m^{2}]
    w_TEAL      = par.w_TEAL;       % [-]
    Mw_M1       = par.Mw_M1;        % [kg kmol^{-1}]
    Mw_M2       = par.Mw_M2;        % [kg kmol^{-1}]
    Mw_H2       = par.Mw_H2;        % [kg kmol^{-1}]
    Mw_I        = par.Mw_I;         % [kg kmol^{-1}]
    V           = par.V;            % [m^{3}]
    R           = par.R;            % [kJ kmol^{-1} K^{-1}]
    % Controller settings
    f_V_prod_0      = par.u0_h;         % [m^{3} s^{-1}]
    K_h             = par.K_h;          % [m^{2} s^{-1}]
    tau_I_h         = par.tau_I_h;      % [s]
    f_V_prod_min    = par.u_h_min;      % [m^{3} s^{-1}]
    f_V_prod_max    = par.u_h_max;      % [m^{3} s^{-1}]
    f_V_purge_0     = par.u0_p;         % [m^{3} s^{-1}]
    K_p             = par.K_p;          % [m^{3} s^{-1} Pa^{-1}]
    tau_I_p         = par.tau_I_p;      % [s]
    f_V_purge_min   = par.u_p_min;      % [m^{3} s^{-1}]
    f_V_purge_max   = par.u_p_max;      % [m^{3} s^{-1}]
    %% Calculations
    % Level
    m_pol           = Mw_M1*n_B1 + Mw_M2*n_B2;          % Polymer mass [kg]
    V_s             = m_pol/rho_pol ...              % Solid volume [m^{3}]
                    + m_cat*((1-w_TEAL)/rho_cat + w_TEAL/rho_TEAL);
    h               = V_s/((1 - eps_avg)*area_cs);              % Level [m]
    % Level controller
    [f_V_prod, ~]   = PIcontrol(h, h_s, f_V_prod_0, K_h, ...
                                tau_I_h, i_e_h, ...
                                f_V_prod_min, f_V_prod_max);
    f_m_pol         = m_pol*f_V_prod/V_s;
    % Purge rate
    [c_M1, c_M2, c_H2, c_I, c_g] = getGasConc(n_M1, n_M2, n_H2, n_I, ...
                                              V - V_s);
    p               = Z*c_g*R*T;
    [f_V_purge, ~]  = PIcontrol(p, p_s, f_V_purge_0, K_p, ...
                                tau_I_p, i_e_p, ...
                                f_V_purge_min, f_V_purge_max);
    rho_gas         = c_H2*Mw_H2 + c_M1*Mw_M1 + c_M2*Mw_M2 + c_I*Mw_I;
    f_m_pu          = rho_gas*f_V_purge;

    % Polymer properties
    x_pol= n_B1/(n_B1 + n_B2);
    Mw_M = x_pol*Mw_M1 + (1-x_pol)*Mw_M2;                  % [kg kmol^{-1}]
    Mw_w = Mw_M*(YX_2/(X_1 + Y_1_1 + Y_1_2));              % [kg kmol^{-1}]
    Mw_n = Mw_M*((X_1 + Y_1_1 + Y_1_2)/(X_0 + Y_0_1 + Y_0_2));
    PDI  = Mw_w/Mw_n;

    prod = m_pol/((1-w_TEAL)*m_cat*0.02);
    %% Insertions
    z       = zeros(6,1);
    z(1)    = PDI;                  % [-]
    z(2)    = Mw_w;                 % [kg kmol^{-1}]
    z(3)    = x_pol;                % [-]
    z(4)    = f_m_pol;              % [kg s^{-1}]
    z(5)    = f_m_pu;               % [kg s^{-1}]
    z(6)    = prod;                 % [-]
end
```

C.2 Integration of the model equations

The main script is identical for the control model (cm) and the plant replacement model (prm), it is displayed in Code snippet b.3 in Appendix b. The initialization of the model is provided in Code snippet c.4.

Code snippet c.4 – The function that returns the initial values for the control model, init_model.m.

```
function [x0, u0, par, y0, z0] = init_model()
    par             = init_par;
    [x0, u0]          = init_x_u(par);
    y0                 = model_y(x0, u0, par);
    z0                 = model_z(x0, u0, y0, par);
end

function [x0, u0] = init_x_u(par)
    x0      = zeros(18,1);
    u0      = zeros(6,1);
```

```
% Initial conditions:
% Feed composition
x_H2     = 0.001;
x_M1     = 0.95;
x_M2     = 0.049;
x_I      = 1e-4;

p        = par.p_s;
Z        = par.Z;
R        = par.R;
V        = par.V;
T        = par.T_f;
rho_pol  = par.rho_pol;
Mw_M1    = par.Mw_M1;
Mw_M2    = par.Mw_M2;

% Solid and gas volumes
V_s      = par.h_s*par.area_cs*(1-par.eps_avg);
V_g      = V - V_s;
% Moles of gas
n_g      = p*V_g/(Z*R*T);
n_H2     = n_g*x_H2;
n_M1     = n_g*x_M1;
n_M2     = n_g*x_M2;
n_I      = n_g*x_I;
% Polymer
x_M1_s   = x_M1/(x_M1 + x_M2);
x_M2_s   = x_M2/(x_M1 + x_M2);
w_pol    = 0.999;
m_pol    = rho_pol*V_s*w_pol;
Mw_M     = x_M1_s*Mw_M1 + x_M2_s*Mw_M2;
n_pol    = m_pol/Mw_M;
n_B1     = n_pol*x_M1_s;
n_B2     = n_pol*x_M2_s;
% Insert states
% Gas phase
x0(1)    = n_H2;
x0(2)    = n_M1;
x0(3)    = n_M2;
x0(4)    = n_I;
% Bound monomers
x0(5)    = n_B1;
x0(6)    = n_B2;
% Catalyst
x0(14)   = (1 - w_pol)/w_pol*m_pol;
```

```
    % Temperature
    x0(15)  = T;
    % Volumetric feed rate
    f_V_tot = 0.045;
    f_m_ZN  = 1e-3;
    f_m_cat = f_m_ZN/(1-par.w_TEAL);
    u0(1)   = f_V_tot*x_H2;
    u0(2)   = f_V_tot*x_M1;
    u0(3)   = f_V_tot*x_M2;
    u0(4)   = f_V_tot*x_I;
    u0(5)   = f_m_cat;
    u0(6)   = 353.15;
end

function par = init_par
    par.h_s     = 14;           % Level setpoint [m]
    par.p_s     = 25e5;         % Pressure setpoint [Pa]
    par.T_f     = 325.15;       % Feed temperature [K]
    par.T_ref   = 342.15;       % Reference temperature for reaction rates
    par.UA_wm   = 83577;        % Heat exchanger coefficient [W K^{-1}]
    par.rho_cat = 2345;         % Catalyst density [kg m^{-3}]
    par.rho_pol = 910;          % Polymer density [kg m^{-3}]
    par.rho_TEAL= 832.4;        % TEAL density [kg m^{-3}]
    par.eps_mf  = 0.45;         % Minimum fluidization void fraction
    par.eps_avg = 0.7           % Average void fraction
    par.eps_prod= 0.3;          % Void fraction in product stream
    d           = 5;            % Reactor diameter [m]
    par.area_cs = pi/4*d^2;     % Cross sectional area [m^{2}]
    par.Mw_TEAL = 114.17;       % Molecular mass of TEAL [kg kmol^{-1}]
    par.Mw_M1   = 42.08;        % Molecular mass of C3= [kg kmol^{-1}]
    par.Mw_M2   = 28.05;        % Molecular mass of C2= [kg kmol^{-1}]
    par.Mw_H2   = 2*1.008;      % Molecular mass of H2 [kg kmol^{-1}]
    par.Mw_I    = 2*14.007;     % Molecular mass of N2 [kg kmol^{-1}]
    par.V       = 350;          % Volume [m^{3}]
    par.R       = 8.314e3;      % Gas constant [J kmol^{-1} K^{-1}]
    par.Z       = 0.7222;       % Compressibility factor
    % Calculation of active sites per kg catalyst
    w_Ti        = 0.02;             % Mass fraction Ti in catalyst
    x_P         = 0.2;              % Potential site fraction
    Mw_Ti       = 47.867;           % Molecular mass of Ti
    x_P_1       = 0.8064;           % Fraction of type 1 site
    x_P_2       = 0.1936;           % Fraction of type 2 site
    n_site_T    = w_Ti/Mw_Ti*x_P;   % Moles of active sites [kmol kg^{-1}]
    par.n_site_1= n_site_T*x_P_1;   % site 1
    par.n_site_2= n_site_T*x_P_2;   % site 1

    r_AlTi      = 50;              % Molar ratio of aluminium to titanium
    % Mass fraction of TEAL in catalyst feed
    par.w_TEAL  = 1/(Mw_Ti/(r_AlTi*w_Ti*par.Mw_TEAL) + 1);

    % Heat capacities
    par.c_p_H2  = 29.04e3;              % [J kmol^{-1} K^{-1}]
    par.c_p_M1  = 71.89e3;              % [J kmol^{-1} K^{-1}]
    par.c_p_M2  = 47.46e3;              % [J kmol^{-1} K^{-1}]
    par.c_p_I   = 29.16e3;              % [J kmol^{-1} K^{-1}]
    par.c_p_pol = 2.25e3;               % [J kg^{-1} K^{-1}]
    par.c_p_cat = 0.77e3;               % [J kg^{-1} K^{-1}]
    par.c_p_TEAL= 239e3/par.Mw_TEAL;    % [J kg^{-1} K^{-1}]
    % Heat of reactions
    par.r_h_1   = -103763200;           % [J kmol^{-1}]
    par.r_h_2   = -107528800;           % [J kmol^{-1}]
    par.r_cp_1  = 29288;                % [J kmol^{-1} K^{-1}]
    par.r_cp_2  = 46900;                % [J kmol^{-1} K^{-1}]
    % Activation energies
    par.Ea_1    = 7200*4.184e3;          % [J kmol^{-1}]
    par.Ea_2    = 9000*4.184e3;          % [J kmol^{-1}]
    % Level controller parameters
    par.K_h     = -0.0349863;   % Gain [m^2 s^{-1}]
    par.tau_I_h = 440.113;      % Integral time [s]
    par.u0_h    = 0;            % Initial product rate
    par.u_h_min = 0;            % Minimum product flow [m^{3} s^{-1}]
    par.u_h_max = 0.02;         % Maximum product flow [m^{3} s^{-1}]
    % Pressure controller parameters
    par.K_p     = -3.45115e-7;  % Gain [m^{3} s^{-1} Pa^{-1}]
    par.tau_I_p = 907.686;      % Integral time [s]
    par.u0_p    = 0;            % Initial purge rate [m^{3} s^{-1}]
    par.u_p_min = 0;            % Minimum product flow [m^{3} s^{-1}]
    par.u_p_max = 0.05;         % Maximum product flow [m^{3} s^{-1}]
    % Temperature controller parameters
    par.K_T     = 2.34731;      % Gain [-]
    par.tau_I_T = 1454.81;      % Integral time [s]
    par.u0_T    = 303.15;       % Initial cooling water temperature [K]
    par.u_T_min = 283.15;       % Minimum cooling water temperature [K]
    par.u_T_max = 333.15;       % Maximum cooling water temperature [K]
    % Reaction rates (1)
    % Formation
    par.k_f_1   = 1;        % site 1 [s^{-1}]
    par.k_f_2   = 1;        % site 2 [s^{-1}]
    % Initiations
    par.k_i_1_1 = 9.8;      % C3= site 1 [m^{3} kmol^{-1} s^{-1}]
```

```
par.k_i_1_2 = 9.8;      % C3= site 2 [m^{3} kmol^{-1} s^{-1}]
par.k_i_2_1 = 14.6;     % C2= site 1 [m^{3} kmol^{-1} s^{-1}]
par.k_i_2_2 = 14.6;     % C2= site 2 [m^{3} kmol^{-1} s^{-1}]

par.k_h_1_1 = 1;    % C3= site 1 [m^{3} kmol^{-1} s^{-1}]
par.k_h_1_2 = 1;    % C3= site 2 [m^{3} kmol^{-1} s^{-1}]
par.k_h_2_1 = 0.1;  % C2= site 1 [m^{3} kmol^{-1} s^{-1}]
par.k_h_2_2 = 0.1;  % C2= site 2 [m^{3} kmol^{-1} s^{-1}]

par.k_hr_1  = 20;   % site 1 [m^{3} kmol^{-1} s^{-1}]
par.k_hr_2  = 20;   % site 2 [m^{3} kmol^{-1} s^{-1}]
% Propagations
T_corr = exp(-10000*4.184/par.R*(1/par.T_ref - 1/333.15));
par.k_p_11_1    = 220.477*T_corr; % C3= C3= s1 [m^{3} kmol^{-1} s^{-1}]
par.k_p_11_2    = 22.047*T_corr;  % C3= C3= s2 [m^{3} kmol^{-1} s^{-1}]
par.k_p_12_1    = 591.1098*T_corr;% C3= C2= s1 [m^{3} kmol^{-1} s^{-1}]
par.k_p_12_2    = 130.783*T_corr; % C3= C2= s2 [m^{3} kmol^{-1} s^{-1}]
par.k_p_21_1    = 1.701*T_corr;   % C2= C3= s1 [m^{3} kmol^{-1} s^{-1}]
par.k_p_21_2    = 376.396*T_corr; % C2= C3= s2 [m^{3} kmol^{-1} s^{-1}]
par.k_p_22_1    = 4.561*T_corr;   % C2= C2= s1 [m^{3} kmol^{-1} s^{-1}]
par.k_p_22_2    = 6.698*T_corr;   % C2= C2= s2 [m^{3} kmol^{-1} s^{-1}]
% Transfers
par.k_fm_11_1   = 0.006;    % C3= C3= site 1 [m^{3} kmol^{-1} s^{-1}]
par.k_fm_11_2   = 0.006;    % C3= C3= site 2 [m^{3} kmol^{-1} s^{-1}]
par.k_fm_12_1   = 0.0021;   % C3= C2= site 1 [m^{3} kmol^{-1} s^{-1}]
par.k_fm_12_2   = 0.0021;   % C3= C2= site 2 [m^{3} kmol^{-1} s^{-1}]
par.k_fm_21_1   = 0.006;    % C2= C3= site 1 [m^{3} kmol^{-1} s^{-1}]
par.k_fm_21_2   = 0.006;    % C2= C3= site 2 [m^{3} kmol^{-1} s^{-1}]
par.k_fm_22_1   = 0.0021;   % C2= C2= site 1 [m^{3} kmol^{-1} s^{-1}]
par.k_fm_22_2   = 0.0021;   % C2= C2= site 2 [m^{3} kmol^{-1} s^{-1}]

par.k_fh_1_1    = 0.088;    % C3= site 1 [m^{3} kmol^{-1} s^{-1}]
par.k_fh_1_2    = 0.37;     % C3= site 2 [m^{3} kmol^{-1} s^{-1}]
par.k_fh_2_1    = 0.088;    % C2= site 1 [m^{3} kmol^{-1} s^{-1}]
par.k_fh_2_2    = 0.37;     % C2= site 2 [m^{3} kmol^{-1} s^{-1}]

par.k_fr_1_1    = 0.024;    % C3= site 1 [m^{3} kmol^{-1} s^{-1}]
par.k_fr_1_2    = 0.12;     % C3= site 2 [m^{3} kmol^{-1} s^{-1}]
par.k_fr_2_1    = 0.048;    % C2= site 1 [m^{3} kmol^{-1} s^{-1}]
par.k_fr_2_2    = 0.24;     % C2= site 2 [m^{3} kmol^{-1} s^{-1}]

par.k_fs_1_1    = 0.0001;     % C3= site 1 [s^{-1}]
par.k_fs_1_2    = 0.0001;     % C3= site 2 [s^{-1}]
par.k_fs_2_1    = 0.0001;     % C2= site 1 [s^{-1}]
par.k_fs_2_2    = 0.0001;     % C2= site 2 [s^{-1}]
```

```
    % Deactivation
    par.k_ds_1  = 0.0001;   % site 1 [s^{-1}]
    par.k_ds_2  = 0.0001;   % site 2 [s^{-1}]

    % Reaction rate corrections (estimated)
    par.chi_M1_1    = 0.9399;    % C3= site 1
    par.chi_M1_2    = 1.0064;    % C3= site 2
    par.chi_M2_1    = 0.9149;    % C2= site 1
    par.chi_M2_2    = 1.0655;    % C2= site 2
    par.chi_H2      = 0.4570;    % H2 site 1
end
```

Appendix D

Auxiliary Matlab functions

> Hofstadter's Law: It always takes longer than you expect, even when you take into account Hofstadter's Law.
>
> — D. Hofstadter, 1979[1]

D.1 Common

This section contains the functions utilized by both the plant replacement model (PRM) and the control model (CM). The implementation of a proportional integral (PI) controller is displayed in Code snippet D.1.

[1] Hofstadter, D., *Gödel, Escher, Bach: An Eternal Golden Braid*. 1979

Code snippet D.1 – The implementation of a proportional integral (PI) controller, PIcontrol.m.

```
function [u, e] = PIcontrol(y, y_s, u_0, K, tau_I, i_e, u_min, u_max)
    %% PI Controller
    % Calculates the applied input and the error for integration for a PI
    % controller. Includes anti-windup where the reset time has been set
    % equal to the integral time.
    % Inputs:                                   Pseudo-units
    %   y       Current output/measurement      [y]
    %   y_s     Setpoint                        [y]
    %   u_0     Initial input                   [u]
    %   K       Controller gain                 [u/y]
    %   tau_I   Controller integral time        [s]
    %   i_e     Integral of the offset          [y s]
    %   u_min   Minimum input constraint        [u]
    %   u_max   Maximum input constraint        [u]
    % Outputs:
    %   u       Applied input                   [u]
    %   e       Error to be integrated          [y]
    e = y_s - y;                    % Setpoint offset           [y]
    v = u_0 + K*(e + 1/tau_I*i_e);  % Desired input             [u]
    u = max(u_min, min(u_max, v));  % Constraints               [u]
    e_u = u - v;                    % Input offset              [u]
    e   = e + e_u/K;                % Error to be integrated    [y]
end
```

The calculation of the gas concentrations is given in Code snippet D.2.

Code snippet D.2 – The implementation of the calculation of the gas phase concentrations, getGasConc.m.

```
function [c_M1, c_M2, c_H2, c_I, c_g] = getGasConc(n_M1, n_M2, ...
                                                   n_H2, n_I, V)
    c_M1    = n_M1/V;                   % Propylene [kmol m^{-3}]
    c_M2    = n_M2/V;                   % Ethylene [kmol m^{-3}]
    c_H2    = n_H2/V;                   % Hydrogen [kmol m^{-3}]
    c_I     = n_I/V;                    % Inert [kmol m^{-3}]
    c_g     = c_M1 + c_M2 + c_H2 + c_I; % Gas concentration [kmol m^{-3}]
end
```

The reaction rates is obtained with Code snippet D.3.

Code snippet D.3 – The implementation of the calculation of the reaction rates, getReactionRates.m.

```
function [r_n_H2, r_n_M1, ...
          r_n_M2, r_n_B1, ...
          r_n_B2, r_Y_0_1, ...
          r_Y_0_2, r_Y_1_1, ...
          r_Y_1_2, r_X_0, ...
          r_X_1, r_YX_2] = getReactionRates(c_H2, c_M1, c_M2, Y_0_1, ...
                                            Y_0_2, Y_1_1, Y_1_2, T, ...
                                            V_s, m_cat, tau_prod, ...
                                            f_n_P_1_in, f_n_P_2_in, par)
    %% Calculate the reaction rates
    % Input:
    %   c_H2        Hydrogen concentration
    %   c_M1        Propylene concentration
    %   c_M2        Ethylene concentration
    %   c_I         Inert concentration
    %   Y_0_1       0th moment of chain length distribution for living
    %               polymer produced at site 1
    %   Y_0_2       0th moment of chain length distribution for living
    %               polymer produced at site 2
    %   Y_1_1       1st moment of chain length distribution for living
    %               polymer produced at site 1
    %   Y_1_2       1st moment of chain length distribution for living
    %               polymer produced at site 2
    %   T           Temperature
    %   V_s         Volume of solids
    %   m_cat       Mass of catalyst and cocatalyst
    %   tau_prod    Product flow rate divided by solid volume
    %   f_n_P_1_in  Feed rate of potential active sites of type 1
    %   f_n_P_2_in  Feed rate of potential active sites of type 2
    % Parameters:
    %   T_ref       Reference temperature
    %   Mw_TEAL     Molecular mass of TEAL
    %   w_TEAL      Mass fraction TEAL in catalyst feed
    %   R            Universal gas constant
    %   k_f_1       Formation rate constant for site 1
    %   k_f_2       Formation rate constant for site 2
    %   k_i_1_1     Initiation from n_0 and C3= on site 1
    %   k_i_1_2     Initiation from n_0 and C3= on site 2
    %   k_i_2_1     Initiation from n_0 and C2= on site 1
    %   k_i_2_2     Initiation from n_0 and C2= on site 2
    %   k_h_1_1     Initiation from n_H and C3= on site 1
    %   k_h_1_2     Initiation from n_H and C3= on site 2
    %   k_h_2_1     Initiation from n_H and C2= on site 1
    %   k_h_2_2     Initiation from n_H and C2= on site 2
    %   k_hr_1      Initiation from n_H and TEAL on site 1
    %   k_hr_2      Initiation from n_H and TEAL on site 2
    %   k_p_11_1    Propagation with terminal C3= with C3= on site 1
    %   k_p_11_2    Propagation with terminal C3= with C3= on site 2
    %   k_p_12_1    Propagation with terminal C3= with C2= on site 1
    %   k_p_12_2    Propagation with terminal C3= with C2= on site 2
    %   k_p_21_1    Propagation with terminal C2= with C3= on site 1
    %   k_p_21_2    Propagation with terminal C2= with C3= on site 2
    %   k_p_22_1    Propagation with terminal C2= with C2= on site 1
    %   k_p_22_2    Propagation with terminal C2= with C2= on site 2
    %   k_fm_11_1   Transfer with terminal C3= with C3= on site 1
    %   k_fm_11_2   Transfer with terminal C3= with C3= on site 2
    %   k_fm_12_1   Transfer with terminal C3= with C2= on site 1
    %   k_fm_12_2   Transfer with terminal C3= with C2= on site 2
    %   k_fm_21_1   Transfer with terminal C2= with C3= on site 1
    %   k_fm_21_2   Transfer with terminal C2= with C3= on site 2
    %   k_fm_22_1   Transfer with terminal C2= with C2= on site 1
    %   k_fm_22_2   Transfer with terminal C2= with C2= on site 2
    %   k_fh_1_1    Transfer with terminal C3= with H2 on site 1
    %   k_fh_1_2    Transfer with terminal C3= with H2 on site 2
    %   k_fh_2_1    Transfer with terminal C2= with H2 on site 1
    %   k_fh_2_2    Transfer with terminal C2= with H2 on site 2
    %   k_fr_1_1    Transfer with terminal C3= with TEAL on site 1
    %   k_fr_1_2    Transfer with terminal C3= with TEAL on site 2
    %   k_fr_2_1    Transfer with terminal C2= with TEAL on site 1
    %   k_fr_2_2    Transfer with terminal C2= with TEAL on site 2
    %   k_fs_1_1    Spontaneous transfer with terminal C3= on site 1
    %   k_fs_1_2    Spontaneous transfer with terminal C3= on site 2
    %   k_fs_2_1    Spontaneous transfer with terminal C2= on site 1
    %   k_fs_2_2    Spontaneous transfer with terminal C2= on site 2
    %   k_ds_1      Deactivation on site 1
    %   k_ds_2      Deactivation on site 2
    % Output:
    %   r_n_H2      Reaction rate of hydrogen
    %   r_n_M1      Reaction rate of propylene
    %   r_n_M2      Reaction rate of ethylene
    %   r_n_B1      Reaction rate of bound propylene
    %   r_n_B2      Reaction rate of bound ethylene
    %   r_Y_0_1     Reaction rate of 0th moment of living at site 1
    %   r_Y_0_2     Reaction rate of 0th moment of living at site 2
    %   r_Y_1_1     Reaction rate of 1st moment of living at site 1
    %   r_Y_1_2     Reaction rate of 1st moment of living at site 2
    %   r_X_0       Reaction rate of 0th moment of dead polymer
    %   r_X_1       Reaction rate of 1st moment of dead polymer
```

```
%   r_YX_2      Reaction rate of 2nd moment of polymer
%% Extractions
w_TEAL      = par.w_TEAL;       % [-]
Mw_TEAL     = par.Mw_TEAL;      % [kg kmol^{-1}]
T_ref       = par.T_ref;        % [K]
R           = par.R;            % [J kmol^{-1} K^{-1}]
% Rate constants
% Formation
k_f_1   = par.k_f_1;    % site 1 [s^{-1}]
k_f_2   = par.k_f_2;    % site 2 [s^{-1}]
% Initiations
k_i_1_1 = par.k_i_1_1;  % C3= site 1 [m^{3} kmol^{-1} s^{-1}]
k_i_1_2 = par.k_i_1_2;  % C3= site 2 [m^{3} kmol^{-1} s^{-1}]
k_i_2_1 = par.k_i_2_1;  % C2= site 1 [m^{3} kmol^{-1} s^{-1}]
k_i_2_2 = par.k_i_2_2;  % C2= site 2 [m^{3} kmol^{-1} s^{-1}]

k_h_1_1 = par.k_h_1_1;  % C3= site 1 [m^{3} kmol^{-1} s^{-1}]
k_h_1_2 = par.k_h_1_2;  % C3= site 2 [m^{3} kmol^{-1} s^{-1}]
k_h_2_1 = par.k_h_2_1;  % C2= site 1 [m^{3} kmol^{-1} s^{-1}]
k_h_2_2 = par.k_h_2_2;  % C2= site 2 [m^{3} kmol^{-1} s^{-1}]

k_hr_1  = par.k_hr_1;   % site 1 [m^{3} kmol^{-1} s^{-1}]
k_hr_2  = par.k_hr_2;   % site 2[m^{3} kmol^{-1} s^{-1}]
% Propagations
k_p_11_1    = par.k_p_11_1; % C3= C3= site 1 [m^{3} kmol^{-1} s^{-1}]
k_p_11_2    = par.k_p_11_2; % C3= C3= site 2 [m^{3} kmol^{-1} s^{-1}]
k_p_12_1    = par.k_p_12_1; % C3= C2= site 1 [m^{3} kmol^{-1} s^{-1}]
k_p_12_2    = par.k_p_12_2; % C3= C2= site 2 [m^{3} kmol^{-1} s^{-1}]
k_p_21_1    = par.k_p_21_1; % C2= C3= site 1 [m^{3} kmol^{-1} s^{-1}]
k_p_21_2    = par.k_p_21_2; % C2= C3= site 2 [m^{3} kmol^{-1} s^{-1}]
k_p_22_1    = par.k_p_22_1; % C2= C2= site 1 [m^{3} kmol^{-1} s^{-1}]
k_p_22_2    = par.k_p_22_2; % C2= C2= site 2 [m^{3} kmol^{-1} s^{-1}]
% Transfers
k_fm_11_1   = par.k_fm_11_1;% C3= C3= site 1 [m^{3} kmol^{-1} s^{-1}]
k_fm_11_2   = par.k_fm_11_2;% C3= C3= site 2 [m^{3} kmol^{-1} s^{-1}]
k_fm_12_1   = par.k_fm_12_1;% C3= C2= site 1 [m^{3} kmol^{-1} s^{-1}]
k_fm_12_2   = par.k_fm_12_2;% C3= C2= site 2 [m^{3} kmol^{-1} s^{-1}]
k_fm_21_1   = par.k_fm_21_1;% C2= C3= site 1 [m^{3} kmol^{-1} s^{-1}]
k_fm_21_2   = par.k_fm_21_2;% C2= C3= site 2 [m^{3} kmol^{-1} s^{-1}]
k_fm_22_1   = par.k_fm_22_1;% C2= C2= site 1 [m^{3} kmol^{-1} s^{-1}]
k_fm_22_2   = par.k_fm_22_2;% C2= C2= site 2 [m^{3} kmol^{-1} s^{-1}]

k_fh_1_1    = par.k_fh_1_1;     % C3= site 1 [m^{3} kmol^{-1} s^{-1}]
k_fh_1_2    = par.k_fh_1_2;     % C3= site 2 [m^{3} kmol^{-1} s^{-1}]
k_fh_2_1    = par.k_fh_2_1;     % C2= site 1 [m^{3} kmol^{-1} s^{-1}]
```

```
    k_fh_2_2    = par.k_fh_2_2;     % C2= site 2 [m^{3} kmol^{-1} s^{-1}]

    k_fr_1_1    = par.k_fr_1_1;     % C3= site 1 [m^{3} kmol^{-1} s^{-1}]
    k_fr_1_2    = par.k_fr_1_2;     % C3= site 2 [m^{3} kmol^{-1} s^{-1}]
    k_fr_2_1    = par.k_fr_2_1;     % C2= site 1 [m^{3} kmol^{-1} s^{-1}]
    k_fr_2_2    = par.k_fr_2_2;     % C2= site 2 [m^{3} kmol^{-1} s^{-1}]

    k_fs_1_1    = par.k_fs_1_1;     % C3= site 1 [s^{-1}]
    k_fs_1_2    = par.k_fs_1_2;     % C3= site 2 [s^{-1}]
    k_fs_2_1    = par.k_fs_2_1;     % C2= site 1 [s^{-1}]
    k_fs_2_2    = par.k_fs_2_2;     % C2= site 2 [s^{-1}]
    % Deactivation
    k_ds_1  = par.k_ds_1;    % site 1 [s^{-1}]
    k_ds_2  = par.k_ds_2;   % site 2 [s^{-1}]

    % Activation energies for propagations
    Ea_1    = par.Ea_1;     % C3= [J kmol^{-1}]
    Ea_2    = par.Ea_2;     % C2= [J kmol^{-1}]

    % Corrections
    chi_M1_1    = par.chi_M1_1;     % C3= site 1
    chi_M1_2    = par.chi_M1_2;     % C3= site 2
    chi_M2_1    = par.chi_M2_1;     % C2= site 1
    chi_M2_2    = par.chi_M2_2;     % C2= site 2
    chi_H2      = par.chi_H2;       % H2

    %% Calculations
    % Correct reaction rate constants
    k_fh_1_1 = chi_H2*k_fh_1_1;
    k_fh_2_1 = chi_H2*k_fh_2_1;
    k_fh_1_2 = chi_H2*k_fh_1_2;
    k_fh_2_2 = chi_H2*k_fh_2_2;

    k_p_11_1 = chi_M1_1*k_p_11_1;
    k_p_21_1 = chi_M1_1*k_p_21_1;
    k_p_11_2 = chi_M1_2*k_p_11_2;
    k_p_21_2 = chi_M1_2*k_p_21_2;

    k_p_12_1 = chi_M2_1*k_p_12_1;
    k_p_22_1 = chi_M2_1*k_p_22_1;
    k_p_12_2 = chi_M2_2*k_p_12_2;
    k_p_22_2 = chi_M2_2*k_p_22_2;

    % Update temperature dependence

    k_p_11_1 = k_p_11_1*exp(-Ea_1/R*(1/T - 1/T_ref));
    k_p_11_2 = k_p_11_2*exp(-Ea_1/R*(1/T - 1/T_ref));
    k_p_21_1 = k_p_21_1*exp(-Ea_1/R*(1/T - 1/T_ref));
    k_p_21_2 = k_p_21_2*exp(-Ea_1/R*(1/T - 1/T_ref));

    k_p_12_1 = k_p_12_1*exp(-Ea_2/R*(1/T - 1/T_ref));
    k_p_12_2 = k_p_12_2*exp(-Ea_2/R*(1/T - 1/T_ref));
    k_p_22_1 = k_p_22_1*exp(-Ea_2/R*(1/T - 1/T_ref));
    k_p_22_2 = k_p_22_2*exp(-Ea_2/R*(1/T - 1/T_ref));


    % Cocatalyst concentration
    n_TEAL  = m_cat*w_TEAL/Mw_TEAL;     % Moles of TEAL [kmol]
    c_TEAL  = n_TEAL/V_s;               % TEAL [kmol m^{-3}]

    % Fractions for pseudo rate constants
    c_M_T   = c_M1 + c_M2;  % Total monomer concentration [kmol m^{-3}]
    f_1     = c_M1/c_M_T;   % Fraction of C3=
    f_2     = c_M2/c_M_T;   % Fraction of C2=
    % Fractions of active sites of type j which have terminal monomer i
    psi_1   = f_1*k_p_21_1 + f_2*k_p_12_1;  % Denominator for site 1
    psi_2   = f_1*k_p_21_2 + f_2*k_p_12_2;  % Denominator for site 2
    phi_1_1 = f_1*k_p_21_1/psi_1;           % C3= site 1
    phi_1_2 = f_1*k_p_21_2/psi_2;           % C3= site 2
    phi_2_1 = f_2*k_p_12_1/psi_1;           % C2= site 1
    phi_2_2 = f_2*k_p_12_2/psi_2;           % C2= site 2

    % Initiations
    k_i_T_1 = f_1*k_i_1_1 + f_2*k_i_2_1; % site 1
    k_i_T_2 = f_1*k_i_1_2 + f_2*k_i_2_2; % site 2

    k_h_T_1 = f_1*k_h_1_1 + f_2*k_h_2_1; % site 1
    k_h_T_2 = f_1*k_h_1_2 + f_2*k_h_2_2; % site 2

    % Transfers
    k_fh_T_1 = phi_1_1*k_fh_1_1 + phi_2_1*k_fh_2_1;
    k_fh_T_2 = phi_1_2*k_fh_1_2 + phi_2_1*k_fh_2_2;

    k_fs_T_1 = phi_1_1*k_fs_1_1 + phi_2_1*k_fs_2_1;
    k_fs_T_2 = phi_1_2*k_fs_1_2 + phi_2_2*k_fs_2_2;

    k_fm_1T_1 = f_1*k_fm_11_1 + f_2*k_fm_12_1;
    k_fm_1T_2 = f_1*k_fm_11_2 + f_2*k_fm_12_2;

    k_fm_2T_1 = f_1*k_fm_21_1 + f_2*k_fm_22_1;
    k_fm_2T_2 = f_1*k_fm_21_2 + f_2*k_fm_22_2;

    k_fm_T1_1 = phi_1_1*k_fm_11_1 + phi_2_1*k_fm_21_1;
    k_fm_T1_2 = phi_1_2*k_fm_11_2 + phi_2_2*k_fm_21_2;

    k_fm_T2_1 = phi_1_1*k_fm_12_1 + phi_2_1*k_fm_22_1;
    k_fm_T2_2 = phi_1_2*k_fm_12_2 + phi_2_2*k_fm_22_2;

    k_fm_TT_1 = f_1*k_fm_T1_1 + f_2*k_fm_T2_1;
    k_fm_TT_2 = f_1*k_fm_T1_2 + f_2*k_fm_T2_2;

    k_fr_T_1 = phi_1_1*k_fr_1_1 + phi_2_1*k_fr_2_1;
    k_fr_T_2 = phi_1_2*k_fr_1_2 + phi_2_2*k_fr_2_2;

    % Propagations

    k_p_1T_1 = f_1*k_p_11_1 + f_2*k_p_12_1;
    k_p_1T_2 = f_1*k_p_11_2 + f_2*k_p_12_2;

    k_p_2T_1 = f_1*k_p_21_1 + f_2*k_p_22_1;
    k_p_2T_2 = f_1*k_p_21_2 + f_2*k_p_22_2;

    k_p_T1_1 = phi_1_1*k_p_11_1 + phi_2_1*k_p_21_1;
    k_p_T1_2 = phi_1_2*k_p_11_2 + phi_2_2*k_p_21_2;

    k_p_T2_1 = phi_1_1*k_p_12_1 + phi_2_1*k_p_22_1;
    k_p_T2_2 = phi_1_2*k_p_12_2 + phi_2_2*k_p_22_2;

    k_p_TT_1 = f_1*k_p_T1_1 + f_2*k_p_T2_1;
    k_p_TT_2 = f_1*k_p_T1_2 + f_2*k_p_T2_2;

    % PSSA
    % Potential active sites
    n_P_1 = f_n_P_1_in/(k_f_1 + 1/tau_prod);    % site 1 [kmol]
    n_P_2 = f_n_P_2_in/(k_f_2 + 1/tau_prod);    % site 2 [kmol]
    % Initiation sites with monomer
    n_0_1 = k_f_1*n_P_1/(k_i_T_1*c_M_T + k_ds_1 + 1/tau_prod); % s1 [kmol]
    n_0_2 = k_f_2*n_P_2/(k_i_T_2*c_M_T + k_ds_2 + 1/tau_prod); % s2 [kmol]
    % Initiation sites with hydrogen
    n_H_1 = Y_0_1*(k_fh_T_1*c_H2 + k_fs_T_1)... % site 1 [kmol]
          / (k_h_T_1*c_M_T + k_ds_1 + k_hr_1*c_TEAL + 1/tau_prod);
    n_H_2 = Y_0_2*(k_fh_T_2*c_H2 + k_fs_T_2)... % site 2 [kmol]
          / (k_h_T_2*c_M_T + k_ds_2 + k_hr_2*c_TEAL + 1/tau_prod);
    % Initiated polymer chains of length 1
    % C3= for site 1 [kmol]
```

```
n_1_1_1 = c_M1*(k_i_1_1*n_0_1 + k_h_1_1*n_H_1 + Y_0_1*k_fm_T1_1) ...
        / (c_M_T*(k_p_1T_1 + k_fm_1T_1) + c_H2*k_fh_1_1 ...
           + c_TEAL*k_fr_1_1 + k_fs_1_1 + k_ds_1 + 1/tau_prod);
% C3= for site 2 [kmol]
n_1_1_2 = c_M1*(k_i_1_2*n_0_2 + k_h_1_2*n_H_2 + Y_0_2*k_fm_T1_2) ...
        / (c_M_T*(k_p_1T_2 + k_fm_1T_2) + c_H2*k_fh_1_2 ...
           + c_TEAL*k_fr_1_2 + k_fs_1_2 + k_ds_2 + 1/tau_prod);
% C2= for site 1 [kmol]
n_1_2_1 = c_M2*(k_i_2_1*n_0_1 + k_h_2_1*n_H_1 + Y_0_1*k_fm_T2_1) ...
        / (c_M_T*(k_p_2T_1 + k_fm_2T_1) + c_H2*k_fh_2_1 ...
           + c_TEAL*k_fr_2_1 + k_fs_2_1 + k_ds_1 + 1/tau_prod);
% C2= for site 2 [kmol]
n_1_2_2 = c_M2*(k_i_2_2*n_0_2 + k_h_2_2*n_H_2 + Y_0_2*k_fm_T2_2) ...
        / (c_M_T*(k_p_2T_2 + k_fm_2T_2) + c_H2*k_fh_2_2 ...
           + c_TEAL*k_fr_2_2 + k_fs_2_2 + k_ds_2 + 1/tau_prod);
% Totals
n_1_T_1 = n_1_1_1 + n_1_2_1; % site 1 [kmol]
n_1_T_2 = n_1_1_2 + n_1_2_2; % site 2 [kmol]

% Reactions
% Initiation with monomers and n_0
r_i_T_1 = c_M_T*k_i_T_1*n_0_1;  % site 1 [kmol s^{-1}]
r_i_T_2 = c_M_T*k_i_T_2*n_0_2;  % site 1 [kmol s^{-1}]
% Inititation with monomers and n_H
r_h_T_1 = c_M_T*k_h_T_1*n_H_1;  % site 1 [kmol s^{-1}]
r_h_T_2 = c_M_T*k_h_T_2*n_H_2;  % site 1 [kmol s^{-1}]
% Inititation with cocatalyst and n_H
r_hr_1  = c_TEAL*k_hr_1*n_H_1;  % site 1 [kmol s^{-1}]
r_hr_2  = c_TEAL*k_hr_2*n_H_2;  % site 1 [kmol s^{-1}]
% Total living polymer on site 1
r_Y_0_1 = r_i_T_1 + r_h_T_1 + r_hr_1 ...
        - Y_0_1*(k_fh_T_1*c_H2 + k_fs_T_1 + k_ds_1);
% Total living polymer on site 2
r_Y_0_2 = r_i_T_2 + r_h_T_2 + r_hr_2 ...
        - Y_0_2*(k_fh_T_2*c_H2 + k_fs_T_2 + k_ds_2);
% 1st moment of living polymer on site 1
r_Y_1_1 = r_i_T_1 + r_h_T_1 + r_hr_1 ...
        + Y_0_1*(c_M_T*(k_p_TT_1 + k_fm_TT_1) + c_TEAL*k_fr_T_1) ...
        - Y_1_1*(c_M_T*k_fm_TT_1 + c_TEAL*k_fr_T_1 + ...
                 c_H2*k_fh_T_1 + k_fs_T_1 + k_ds_1);
% 1st moment of living polymer on site 2
r_Y_1_2 = r_i_T_2 + r_h_T_2 + r_hr_2 ...
        + Y_0_2*(c_M_T*(k_p_TT_2 + k_fm_TT_2) + c_TEAL*k_fr_T_2) ...
        - Y_1_2*(c_M_T*k_fm_TT_2 + c_TEAL*k_fr_T_2 ...
                 + c_H2*k_fh_T_2 + k_fs_T_2 + k_ds_2);
```

```
    % Total dead polymer
    r_X_0   = (Y_0_1 - n_1_T_1)*(c_M_T*k_fm_TT_1 + c_TEAL*k_fr_T_1 ...
                     + c_H2*k_fh_T_1 + k_fs_T_1 + k_ds_1) ...
            + (Y_0_2 - n_1_T_2)*(c_M_T*k_fm_TT_2 + c_TEAL*k_fr_T_2 ...
                     + c_H2*k_fh_T_2 + k_fs_T_2 + k_ds_2);
    % 1st moment of dead polymer
    r_X_1   = (Y_1_1 - n_1_T_1)*(c_M_T*k_fm_TT_1 + c_TEAL*k_fr_T_1 ...
                     + c_H2*k_fh_T_1 + k_fs_T_1 + k_ds_1) ...
            + (Y_1_2 - n_1_T_2)*(c_M_T*k_fm_TT_2 + c_TEAL*k_fr_T_2 ...
                     + c_H2*k_fh_T_2 + k_fs_T_2 + k_ds_2);
    % 2nd moment of living and dead polymer
    r_YX_2  = r_i_T_1 + r_h_T_1 + r_hr_1 ...
            + r_i_T_2 + r_h_T_2 + r_hr_2 ...
            + Y_0_1*(c_M_T*(k_fm_TT_1 - k_p_TT_1) + c_TEAL*k_fr_T_1) ...
            + Y_0_2*(c_M_T*(k_fm_TT_2 - k_p_TT_2) + c_TEAL*k_fr_T_2) ...
            + 2*Y_1_1*c_M_T*k_p_TT_1 ...
            + 2*Y_1_2*c_M_T*k_p_TT_2 ...
            - n_1_T_1*(c_M_T*k_fm_TT_1 + c_TEAL*k_fr_T_1 ...
                       + c_H2*k_fh_T_1 + k_fs_T_1 + k_ds_1) ...
            - n_1_T_2*(c_M_T*k_fm_TT_2 + c_TEAL*k_fr_T_2 ...
                       + c_H2*k_fh_T_2 + k_fs_T_2 + k_ds_2);
    % Gas components
    r_n_H2  = -c_H2*(Y_0_1*k_fh_T_1 + Y_0_2*k_fh_T_2);  % H2
    r_n_M1  = -c_M1*(Y_0_1*k_p_T1_1 + Y_0_2*k_p_T1_2);  % C3=
    r_n_M2  = -c_M2*(Y_0_1*k_p_T2_1 + Y_0_2*k_p_T2_2);  % C2=
    % Bound monomers
    r_n_B1  = -r_n_M1;  % C3=
    r_n_B2  = -r_n_M2;  % C2=
end
```

The steady-state is obtained by integrating for a long time, and can be found by utilizing Code snippet D.4

Code snippet D.4 – The implementation of the calculation of the steady-state, steady_state.m.

```
function [x, y, z, par] = steady_state(x0, u0, par, t_end)
    if length(x0) < 30
        nSel = 13;
    else
        nSel = 29;
    end
    options = odeset('AbsTol', eps, ...
                     'RelTol', 1e-13, ...
                     'NormControl', 'on', ...
                     'Stats', 'off', ...
                     'OutputFcn', @odeplot, ...
                     'OutputSel', nSel);
    [t, x]  = ode15s(@(t,x) model_dx(t,x,u0,par), [0 t_end], x0, options);
    [y, ~] = calculate_y_z(t, x, u0, par, 10);
    x       = x(end,:)';
    y       = model_y(x, u0, par);
    z       = model_z(x, u0, y, par);
    par.u0_p= y(7);
    par.u0_h= y(8);
    par.u0_T= y(9);
    x(end-2:end) = 0;
end
```

To calculate the measurements and the derived outputs, Code snippet D.5 has been utilized.

Code snippet D.5 – The implementation of the calculation of the measurements and the outputs given the state at each time-step, `calculate_y_z.m`.

```
function [x, y, z, par] = steady_state(x0, u0, par, t_end)
    if length(x0) < 30
        nSel = 13;
    else
        nSel = 29;
    end
    options = odeset('AbsTol', eps, ...
                     'RelTol', 1e-13, ...
                     'NormControl', 'on', ...
                     'Stats', 'off', ...
                     'OutputFcn', @odeplot, ...
                     'OutputSel', nSel);
    [t, x]  = ode15s(@(t,x) model_dx(t,x,u0,par), [0 t_end], x0, options);
    [y, ~] = calculate_y_z(t, x, u0, par, 10);
    x       = x(end,:)';
    y       = model_y(x, u0, par);
    z       = model_z(x, u0, y, par);
    par.u0_p= y(7);
    par.u0_h= y(8);
    par.u0_T= y(9);
    x(end-2:end) = 0;
end
```

D.2 PLANT REPLACEMENT MODEL

This section contains the functions used exclusively in the implementation of the PRM.

The conversion of the heat capacity of the gas phase from mole based to mass based is given in Code snippet D.6.

CODE SNIPPET D.6 – The implementation of the conversion of the heat capacity from mole-based to mass-based, convertHeatCapacity.m.

```
function c_p_g = convertHeatCapacity(rho_g, c_H2, c_M1, c_M2, c_I, ...
                                     c_p_H2, c_p_M1, c_p_M2, c_p_I)
    c_p_g = (c_H2*c_p_H2 + c_M1*c_p_M1 + c_M2*c_p_M2 + c_I*c_p_I)/rho_g;
end
```

The distribution of solids between the bubble and the emulsion phase has been obtained from Code snippet D.7.

CODE SNIPPET D.7 – The implementation of the distribution of solids between the bubble phase and the emulsion phase, distributeSolids.m.

```
function [e, b] = distributeSolids(delta, eps_b, eps_e, tot)
    b   = (delta*(1-eps_b))/(delta*(1-eps_b) + (1-delta)*(1-eps_e)) * tot;
    e   = tot - b;
end
```

To calculate the diameter of the bubbles, Code snippet D.8 was applied.

CODE SNIPPET D.8 – The implementation of the diameter of the bubbles, getBubbleDiameter.m.

```
function d_b = getBubbleDiameter(z, mu_g, rho_g, rho_s, g, ...
                                 d_p, sphericity, u_0, u_mf, D)
```

```
    d_p_s   = 2.7*d_p*(mu_g^(-2)*rho_g*(rho_s - rho_g)*g)^(1/3);
    u_T_s   = (18*d_p_s^(-2) ...
            + (2.335 - 1.744*sphericity)*d_p_s^(-0.5))^(-1);
    u_T     = u_T_s*(mu_g*rho_g^(-2)*(rho_s-rho_g)*g)^(1/3);
    d_b_max = 2*u_T^2/g;
    d_b_0   = 0.376*(u_0 - u_mf)^2;
    d_b     = d_b_max - (d_b_max - d_b_0)*exp(-0.3*z/D);
end
```

The density of the gas phase has been calculated by Code snippet D.9.

Code snippet D.9 – The implementation of the density of the gas phase, getGasDensity.m.

```
function rho_g = getGasDensity(c_H2, c_M1, c_M2, c_I, ...
                               Mw_H2, Mw_M1, Mw_M2, Mw_I)
    rho_g = c_H2*Mw_H2 + c_M1*Mw_M1 + c_M2*Mw_M2 + c_I*Mw_I;
end
```

The heat transfer in the heat exchanger was calculated by Code snippet D.10.

Code snippet D.10 – The implementation of the heat transfer in the heat exchanger, getHeatTransfer.m.

```
function q = getHeatTransfer(T_hex, T_fb, T_cw, f_H_f_hex, ...
                             f_H_fb_hex, par)
    %% Calculates the heat flow from the heat exchanger
    % Inputs:
    %   T_hex       Heat exchanger temperature
    %   T_fb        Freeboard temperature
    %   T_cw        Cooling water temperature
    %   f_H_f_hex   Enthalpy required to heat the feed flow
    %   f_H_fb_hex  Enthalpy required to heat the recycle flow
    % Parameters:
    %   UA          Heat transfer coefficient times heat transfer area
    %   T_f         Feed temperature
    % Output:
    %   q           Heat transfer


    UA  = par.UA;
    T_f = par.T_f;
    % Inlet temperature is the weighted average of the recycle and feed
    if abs(T_fb - T_hex) < 1e-2     % Avoid division by zero
        T_in = T_fb;
    else
        T_in = (f_H_f_hex + f_H_fb_hex)...
             / (f_H_f_hex/(T_f - T_hex) + f_H_fb_hex/(T_fb - T_hex)) ...
             + T_hex;
    end
    dT_1    = T_in - T_cw;      % Temperature difference at inlet
    dT_2    = T_hex - T_cw;     % Temperature difference at outlet

    if abs(dT_1 - dT_2) < 1e-2;     % Avoid division by zero
        dT_lm = dT_1;
    else
        if dT_1/dT_2 > 0
            dT_lm   = (dT_1 - dT_2)/log(dT_1/dT_2);
        else
            dT_lm   = abs(max(dT_1,dT_2));
        end
    end
    q = UA*dT_lm;
end
```

To obtain the heat transfer coefficient, Code snippet D.11 has been utilized.

Code snippet D.11 – The implementation of the heat transfer coefficient, getHeatTransferCoeff.m.

```
function H_be = getHeatTransferCoeff(u_e, d_b, rho_g, c_p_g, ...
                                     k_g, g, u_br, eps_e)
    H_bc = 4.5*(u_e*rho_g*c_p_g/d_b) ...
         + 5.85*(sqrt(k_g*rho_g*c_p_g*sqrt(g))/d_b^(5/4));
    H_ce = 6.77*sqrt(k_g*rho_g*c_p_g*eps_e*u_br/d_b^3);
    H_be = 1/(1/H_bc + 1/H_ce);
end
```

The interfacial flow between the bubble and the emulsion phase is given in Code snippet D.12.

Code snippet D.12 – The implementation of the interfacial flows between the bubble phase and the emulsion phase, getInterfacialFlow.m.

```
function f_n_bi_e = getInterfacialFlow(c_e, c_bi, K_be, V_bi)
    f_n_bi_e = -K_be*V_bi*(c_e - c_bi);
end
```

To obtain the mass transfer coefficient, Code snippet D.13 has been utilized.

Code snippet D.13 – The implementation of the mass transfer coefficient, getMassTransferCoeff.m.

```
function K_be = getMassTransferCoeff(u_e, d_b, D_g, g, u_br, eps_e)
    K_ce = 6.77*sqrt(D_g*eps_e*u_br/d_b^3);
    K_bc = 4.5*u_e/d_b + 5.85*sqrt(D_g*sqrt(g))/d_b^(5/4);
    K_be = 1/(1/K_ce + 1/K_bc);
end
```

The implementation of the Redlich-Kwong equation of state is presented in Code snippet D.14.

Code snippet D.14 – The implementation of the Redlich-Kwong equation of state, getPressure.m.

```
function [p, Z] = getPressure(c_H2, c_M1, c_M2, c_I, T, par)
    % Calculates the pressure and the compressibility of a mixture using
    % the Redlich-Kwong equation of state
    % Volume corrections
    b_H2 = par.b_H2;
    b_M1 = par.b_M1;
    b_M2 = par.b_M2;
    b_I  = par.b_I;
    % Attractive potential corrections
    a_H2 = par.a_H2;
    a_M1 = par.a_M1;
    a_M2 = par.a_M2;
    a_I  = par.a_I;
    % Gas constant
    R = par.R;
    % Mixture volume correction
    B = c_H2*b_H2 + c_M1*b_M1 + c_M2*b_M2 + c_I*b_I;
    % Mixture attractive potential correction
    A = (c_H2*sqrt(a_H2) + c_M1*sqrt(a_M1) ...
      + c_M2*sqrt(a_M2) + c_I*sqrt(a_I))^2;
    % Total gas concentration
    c = c_H2 + c_M1 + c_M2 + c_I;
    % Pressure
    p = c*R*T/(1 - B) - A/(sqrt(T)*(1 + B));
    Z = p/(c*R*T);
end
```

Code snippet D.15 has been applied to calculate the internal flow in the reactor.

CODE SNIPPET D.15 – The implementation of the internal reactor flows from the bubble and emulsion phases to the freeboard region, getReactorFlows.m.

```
function [f_n_H2, f_n_M1, f_n_M2, f_n_I, ...
          f_V] = getReactorFlows(c_H2_1, c_M1_1, c_M2_1, c_I_1, p1,...
                                 c_H2_2, c_M1_2, c_M2_2, c_I_2, p2, k)
    if p1 > p2
        f_V     = -k*(p2 - p1);
        f_n_H2  = f_V*c_H2_1;
        f_n_M1  = f_V*c_M1_1;
        f_n_M2  = f_V*c_M2_1;
        f_n_I   = f_V*c_I_1;
    else
        f_V     = -k*(p2 - p1);
        f_n_H2  = f_V*c_H2_2;
        f_n_M1  = f_V*c_M1_2;
        f_n_M2  = f_V*c_M2_2;
        f_n_I   = f_V*c_I_2;
    end
end
```

Appendix E

Step responses

That's one small step for a man, one giant leap for mankind.

— N. Armstrong, 1969

The steps were performed by performing perturbations on the inputs around a steady-state. The nominal inputs were

$$\mathbf{u}_0 = \begin{bmatrix} 1.813 \cdot 10^{-4}\,\mathrm{m^3\,s^{-1}} & 0.040\,46\,\mathrm{m^3\,s^{-1}} & 0.007\,04\,\mathrm{m^3\,s^{-1}} \\ 4.5 \cdot 10^{-6}\,\mathrm{m^3\,s^{-1}} & 0.012\,05\,\mathrm{kg\,s^{-1}} & 353.15\,\mathrm{K} \end{bmatrix}^\mathsf{T}$$

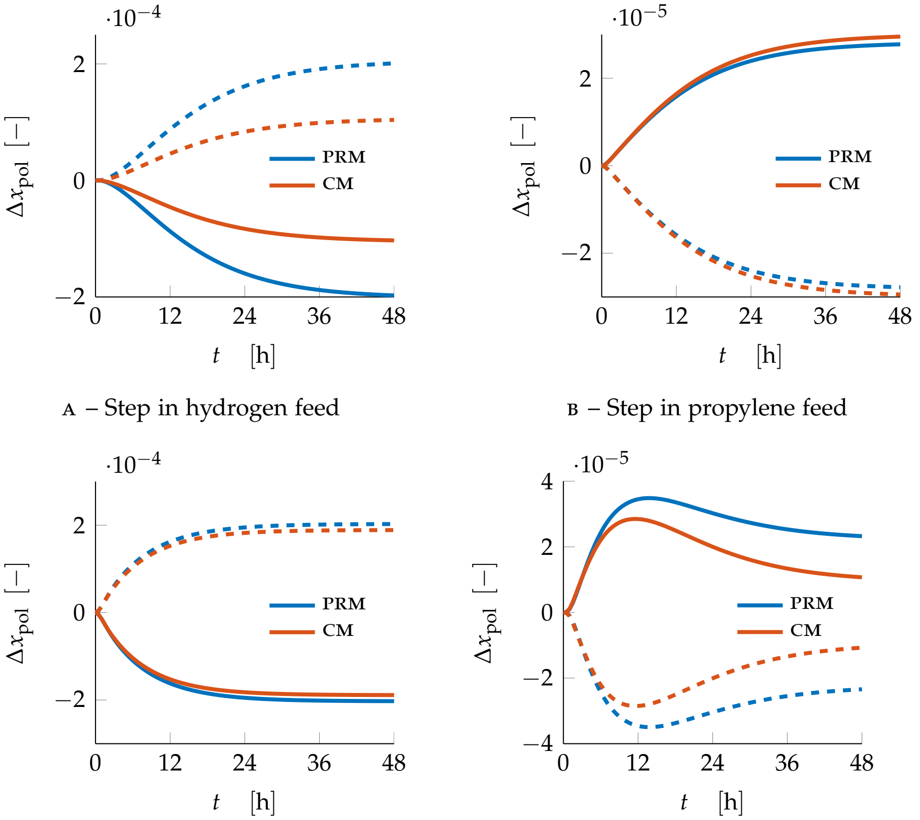
and the perturbations were chosen as

$$\Delta\mathbf{u} = \begin{bmatrix} 1 \cdot 10^{-5}\,\mathrm{m^3\,s^{-1}} & 1 \cdot 10^{-5}\,\mathrm{m^3\,s^{-1}} & 1 \cdot 10^{-5}\,\mathrm{m^3\,s^{-1}} \\ 0\,\mathrm{m^3\,s^{-1}} & 1 \cdot 10^{-5}\,\mathrm{kg\,s^{-1}} & 0\,\mathrm{K} \end{bmatrix}^\mathsf{T}$$

The resulting step responses are presented in the succeeding sections.

E.1 Polymer composition

The perturbations of the polymer composition, i.e., the percentage of propylene in the polymer, is depicted in Figure E.1.



A – Step in hydrogen feed

B – Step in propylene feed

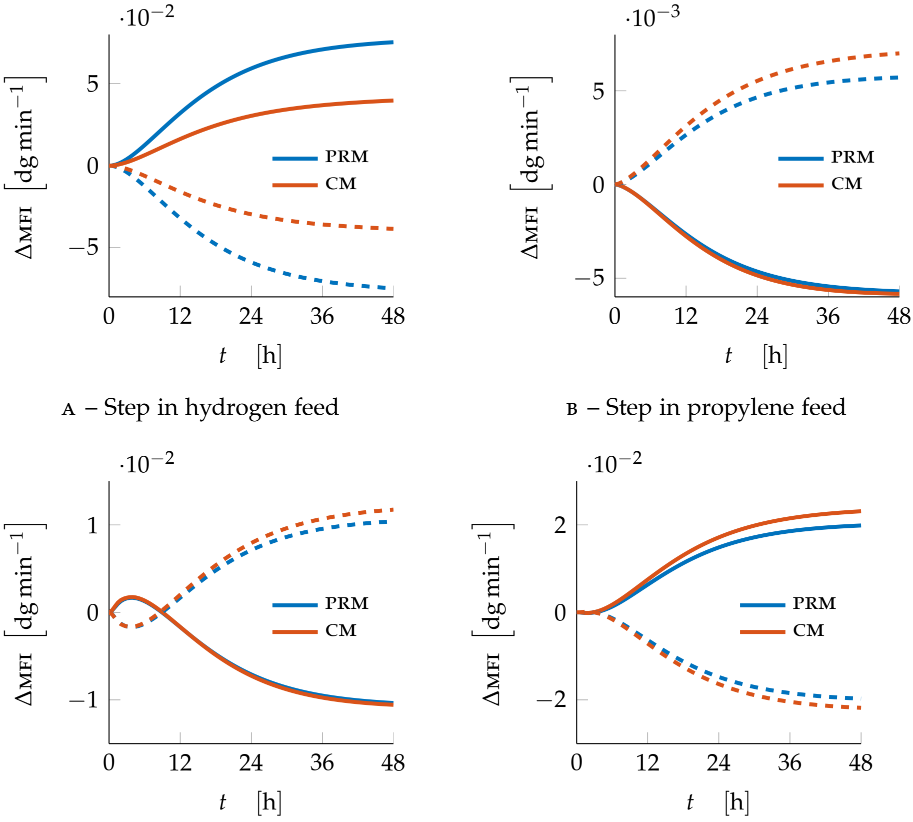
C – Step in ethylene feed

D – Step in catalyst feed

Figure E.1 – The resulting step responses of the polymer composition (x_{pol}) when perturbing the feeds. The blue line and the blue dashed line correspond to a positive and negative perturbation respectively, applied to the plant replacement model (PRM). The red line and the red dashed line correspond to a positive and negative perturbation respectively, applied to the control model (CM).

E.2 Melt flow index

The step responses of the melt flow index (MFI) are portrayed by Figure E.2.



A – Step in hydrogen feed

B – Step in propylene feed

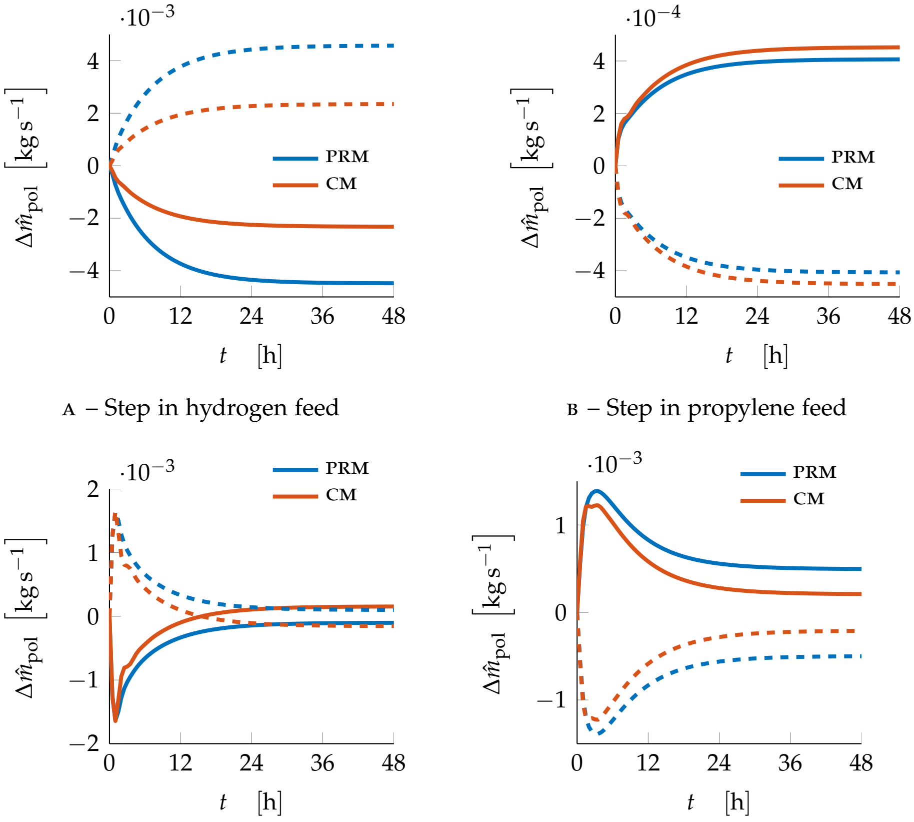
C – Step in ethylene feed

D – Step in catalyst feed

Figure E.2 – The resulting step responses of the melt flow index (MFI) when perturbing the feeds. The blue line and the blue dashed line correspond to a positive and negative perturbation respectively, applied to the plant replacement model (PRM). The red line and the red dashed line correspond to a positive and negative perturbation respectively, applied to the control model (CM).

E.3 Production rate

The sensitivity of the production rate is displayed in Figure E.3.



A – Step in hydrogen feed

B – Step in propylene feed

C – Step in ethylene feed

D – Step in catalyst feed

Figure E.3 – The resulting step responses of the production rate ($\hat{\dot{m}}_{\text{pol}}$) when perturbing the feeds. The blue line and the blue dashed line correspond to a positive and negative perturbation respectively, applied to the plant replacement model (PRM). The red line and the red dashed line correspond to a positive and negative perturbation respectively, applied to the control model (CM).

Appendix F

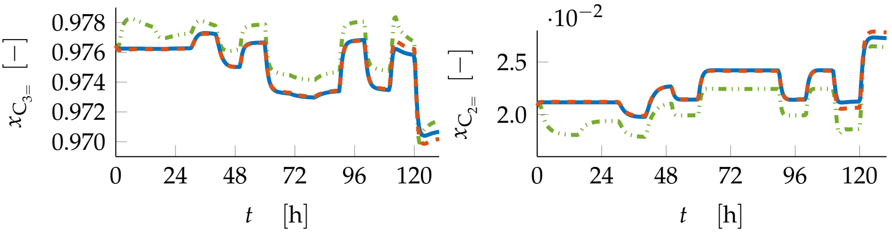
Validation of the offline parameter estimation

All models are wrong, but some are useful.

— G. E. P. Box & N. R. Draper, 1987[1]

To validate the parameters obtained by the offline parameter estimation from Chapter 3, a simulation of a series of steps in the manipulated variables (MVs) was performed. The resulting monomer mole fractions are given in Figure F.1, while the other measurements can be examined in Figure F.2.

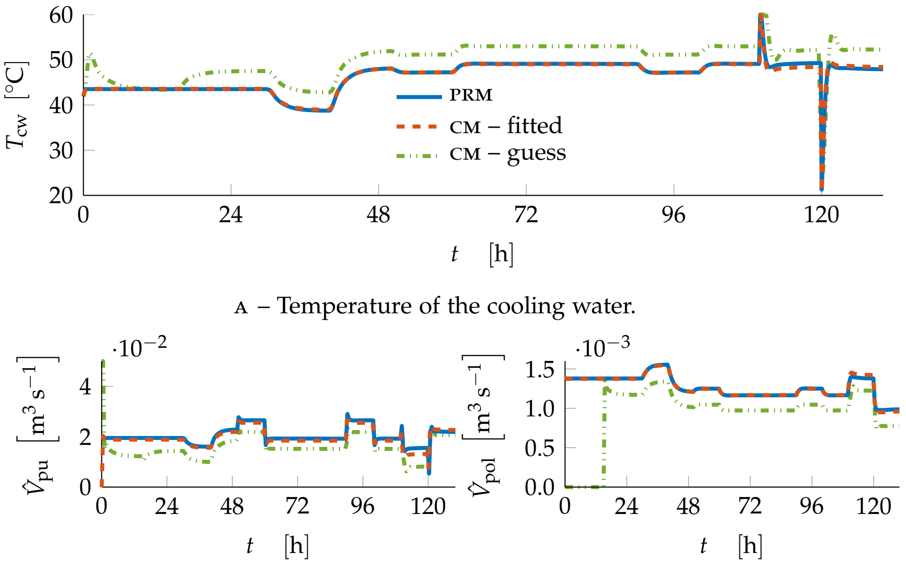
[1] Box, G. E. P. and Draper, N. R., *Empirical model-building and response surfaces*. 1987



A – Propylene.

B – Ethylene.

FIGURE F.1 – The validation of the offline parameter estimation for the mole fraction of the monomers. The blue line is the plant replacement model (PRM) while the red dashed line and the green dash-dotted line are the control model (CM) before and after fitting respectively.



A – Temperature of the cooling water.

B – Purge rate.

C – Production rate.

FIGURE F.2 – The validation of the offline parameter estimation for the temperature of the cooling water, melt flow index (MFI), purge and production rates. The blue line is the plant replacement model (PRM) while the red dashed line and the green dash-dotted line are the control model (CM) before and after fitting respectively.

Appendix G

Setup of the simulation tools

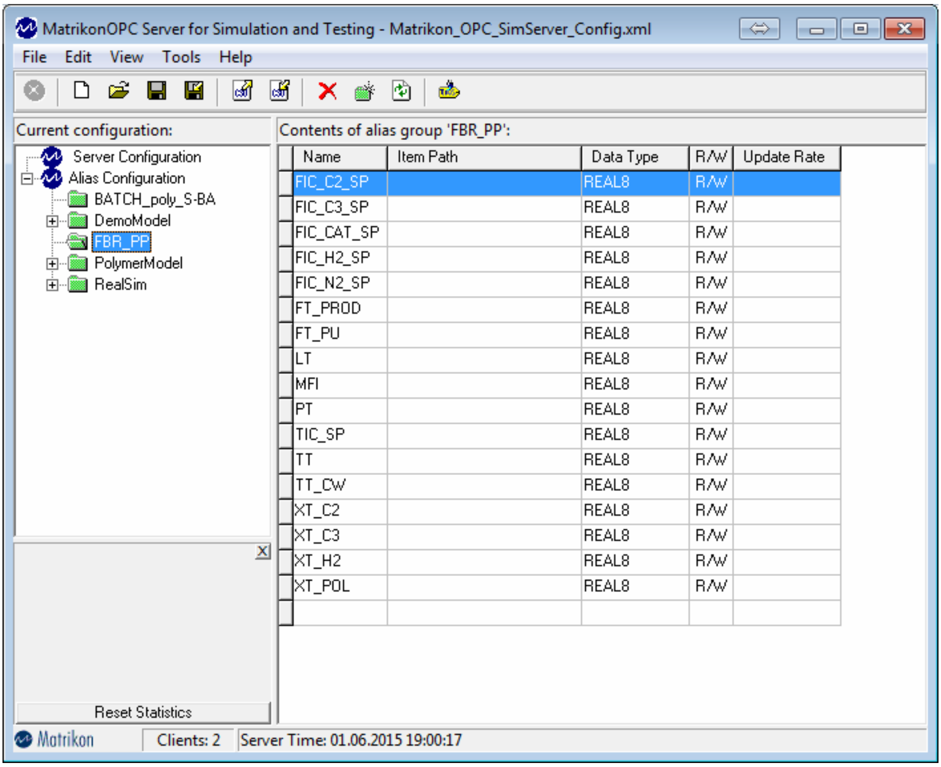
Cybernetica's tool for nonlinear model predictive control (NMPC), CENIT,[1] together with their tool for carrying out simulations, RealSim.[2] These two tools communicate through tags via an open platform communcations (OPC) server. Tags were created for all the manipulated variables (MVs) and the measurements in the Matrikon OPC server.[3] The measurements were then set to be written by the simulator, RealSim and read by CENIT. The inputs on the other hand, were set up oppositely, thus they were written to the server by CENIT and read by RealSim. An illustration of the tags in Matrikon is given in Figure G.1.

The graphical user interface (GUI) for RealSim is displayed in Figure G.2, where the simulations can be set to run for a given number of samples, to

[1] http://www.cybernetica.biz/v3/products/CENIT/index.html

[2] http://www.cybernetica.no/v3/products/RealSim/RealSim.html

[3] http://www.matrikonopc.com/



Contents of alias group 'FBR_PP':

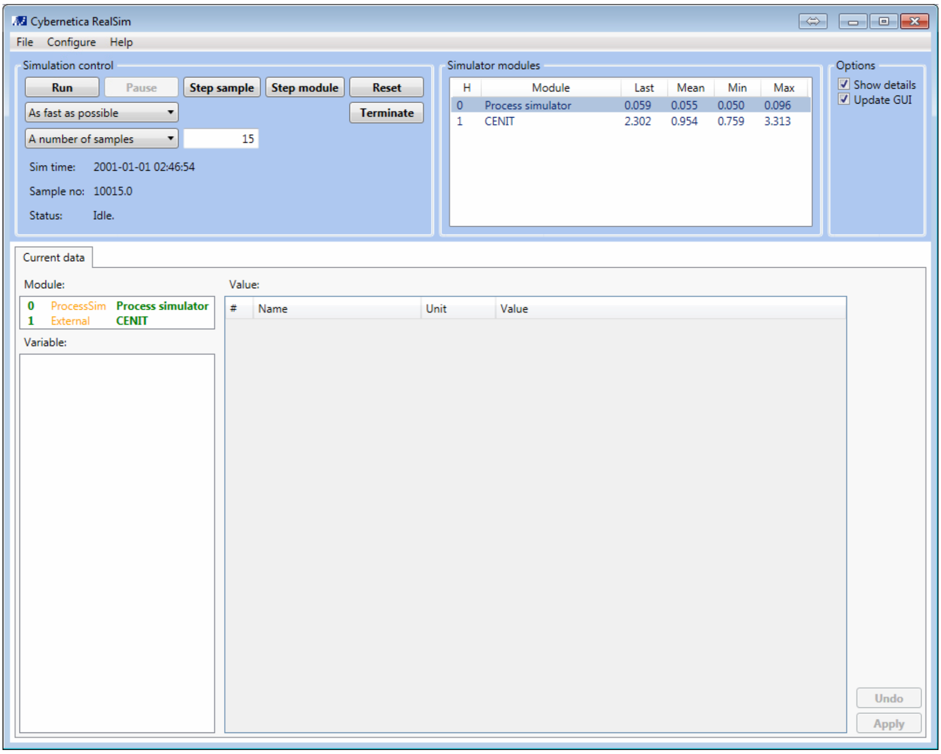
Name	Item Path	Data Type	R/W	Update Rate
FIC_C2_SP		REAL8	R/W	
FIC_C3_SP		REAL8	R/W	
FIC_CAT_SP		REAL8	R/W	
FIC_H2_SP		REAL8	R/W	
FIC_N2_SP		REAL8	R/W	
FT_PROD		REAL8	R/W	
FT_PU		REAL8	R/W	
LT		REAL8	R/W	
MFI		REAL8	R/W	
PT		REAL8	R/W	
TIC_SP		REAL8	R/W	
TT		REAL8	R/W	
TT_CW		REAL8	R/W	
XT_C2		REAL8	R/W	
XT_C3		REAL8	R/W	
XT_H2		REAL8	R/W	
XT_POL		REAL8	R/W	

Clients: 2 Server Time: 01.06.2015 19:00:17

FIGURE G.1 – The tags of the inputs and the measurements in the Matrikon open platform communcations (OPC) server.

a given number of samples, or indefinitely.

The tool for NMPC, CENIT has a command line interface to the kernel in addition to a GUI. They are presented in Figures G.3 and G.4 respectively. The GUI, CENITMMI, allows the user to interactively adjust the set points of the controlled variables (CVS), the weights, the constraints in addition to viewing the predictions and the history of every calculated variable.



As fast as possible

A number of samples 15

Sim time: 2001-01-01 02:46:54

Sample no: 10015.0

Status: Idle.

H	Module	Last	Mean	Min	Max
0	Process simulator	0.059	0.055	0.050	0.096
1	CENIT	2.302	0.954	0.759	3.313

☑ Show details
☑ Update GUI

Module:

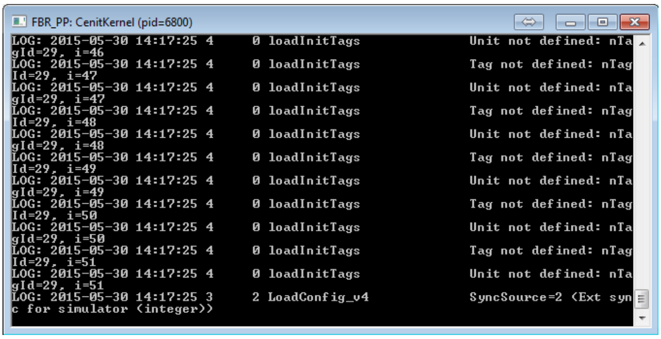
0	ProcessSim	Process simulator
1	External	CENIT

Variable:

Value:

#	Name	Unit	Value

Figure G.2 – The graphical user interface (GUI) of RealSim.



```
LOG: 2015-05-30 14:17:25 4     0 loadInitTags              Unit not defined: nTa
gId=29, i=46
LOG: 2015-05-30 14:17:25 4     0 loadInitTags              Tag not defined: nTag
Id=29, i=47
LOG: 2015-05-30 14:17:25 4     0 loadInitTags              Unit not defined: nTa
gId=29, i=47
LOG: 2015-05-30 14:17:25 4     0 loadInitTags              Tag not defined: nTag
Id=29, i=48
LOG: 2015-05-30 14:17:25 4     0 loadInitTags              Unit not defined: nTa
gId=29, i=48
LOG: 2015-05-30 14:17:25 4     0 loadInitTags              Tag not defined: nTag
Id=29, i=49
LOG: 2015-05-30 14:17:25 4     0 loadInitTags              Unit not defined: nTa
gId=29, i=49
LOG: 2015-05-30 14:17:25 4     0 loadInitTags              Tag not defined: nTag
Id=29, i=50
LOG: 2015-05-30 14:17:25 4     0 loadInitTags              Unit not defined: nTa
gId=29, i=50
LOG: 2015-05-30 14:17:25 4     0 loadInitTags              Tag not defined: nTag
Id=29, i=51
LOG: 2015-05-30 14:17:25 4     0 loadInitTags              Unit not defined: nTa
gId=29, i=51
LOG: 2015-05-30 14:17:25 3     2 LoadConfig_v4             SyncSource=2 (Ext syn
c for simulator (integer))
```

Figure G.3 – The command line interface of CENIT.

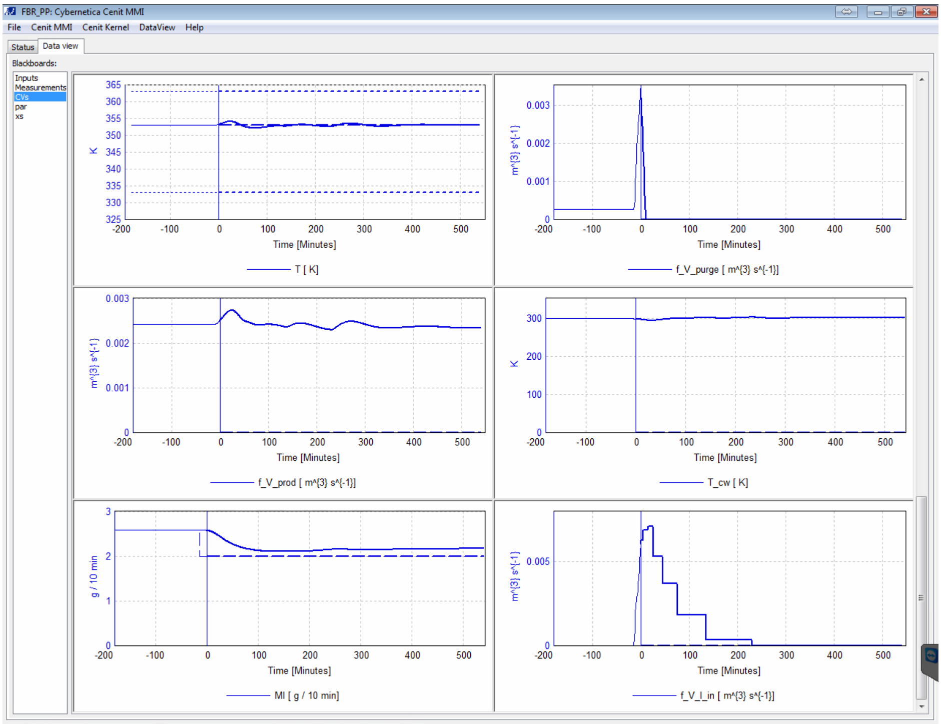


FIGURE G.4 – The graphical user interface (GUI) of CENIT, CENITMMI.

Appendix H

Grade transitions

All the additional results of the nonlinear model predictive control (NMPC) grade transitions from Chapter 5 are presented in the subsequent sections.

H.1 Inert feed as an input

The mole fractions of the components are displayed in Figures H.1 and H.2, while the feed of the monomers are given in Figure H.3.

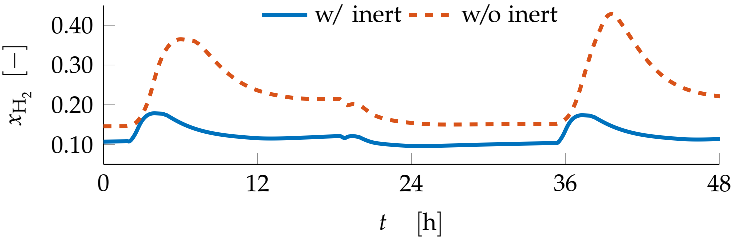
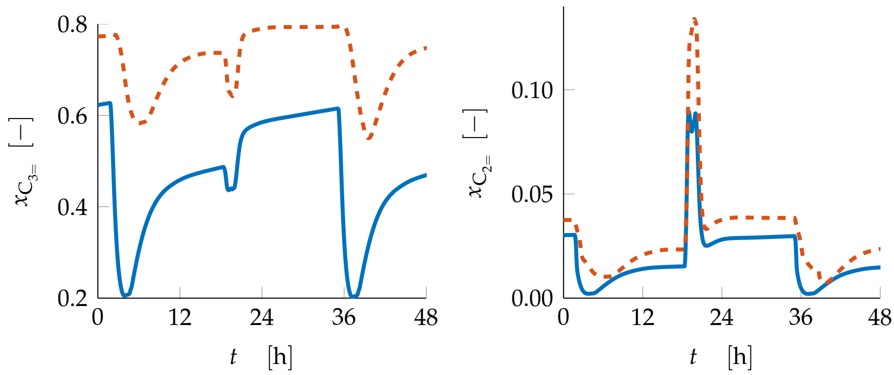


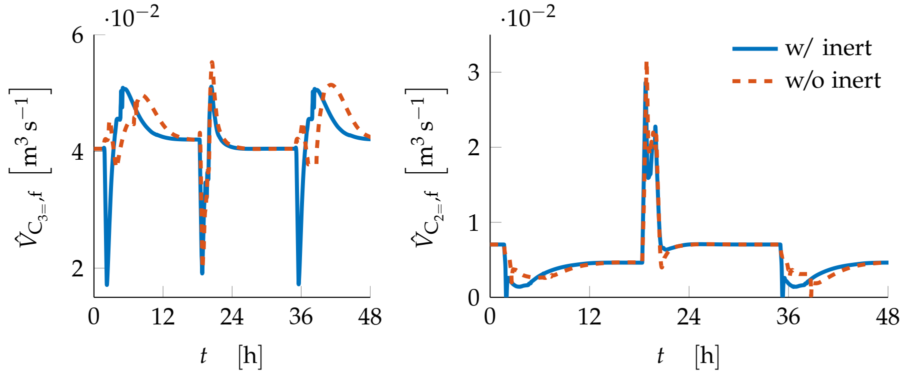
Figure H.1 – The mole fraction of hydrogen with and without the feed of inert as an input.



A – Propylene mole fraction.

B – Ethylene mole fraction.

Figure H.2 – The mole fractions of propylene and ethylene with and without the feed of inert as an input. The blue line and the dashed red line indicate with and without inert respectively.



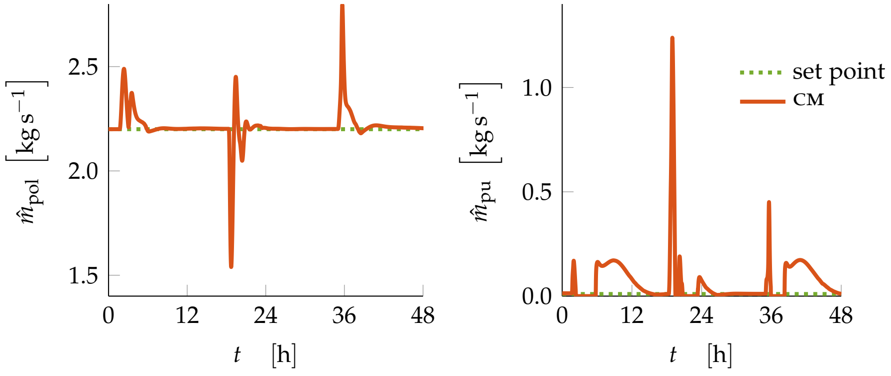
A – Propylene feed.

B – Ethylene feed.

Figure H.3 – The feed of propylene and ethylene with and without the feed of inert as an input. The blue line and the dashed red line indicate with and without inert respectively.

H.2 MODEL MISMATCH

The purge and the production rate of the control model (CM) is presented in Figure H.4.



A – Production rate.

B – Purge rate.

FIGURE H.4 – Production and purge rates with model mismatch. The red line and the dotted green line denote the control model (CM) and the set point respectively.