NTNU
Norwegian University of
Science and Technology

Nonlinear Model Predictive Control and Dynamic Real-Time Optimization of Semi-Batch Reactors

A Case Study of Expandable Polystyrene Production

Brittany Hall

Chemical Engineering
Submission date: June 2018
Supervisor: Johannes Jäschke, IKP
Co-supervisor: Peter Singstad, Cybernetica

Norwegian University of Science and Technology
Department of Chemical Engineering

Summary

The aim of this thesis was to develop an advanced two-layer control structure for the semi-batch polymerization of expandable polystyrene (EPS) that minimizes the batch time while still producing a set polymer quality and obeying other constraints.

While the use of advanced control structures in the chemical industry has become more common, their use is seen to a lesser extent in the polymer industry. This method of control requires an accurate process model, which can be an arduous task due to the nonlinearities and complex reactions that occur during polymerization. However, research has indicated that the use of such control structures can result in a reduced batch time and improved product purity; therefore, the effort to develop these control structures for polymerization should be considered.

Before diving into the development of the control structure for the production of EPS, some background information on polymerization and semi-batch reactor modeling is provided. Introductory concepts from optimization and control are then presented to highlight ideas that are necessary to understand before an advance control structure can be designed. Together, these two chapters provided the required background information to develop a two-level control structure for the production of EPS.

EPS was selected as the case study since it is one of the largest commodity polymers by production volume. The model equations are outlined along with the assumptions that were made in the model derivation. These equations were implemented in the programming language C using a template provided by Cybernetica AS; this allowed for the use of their software in the implementation and simulation of the advanced control structure. Offline optimization of the process was performed to identify a starting point for the optimal operating conditions. The two control layers were then constructed and validated. Attention is paid to how the two layers work together to calculate and realize the optimal operating conditions.

This work demonstrated that the objective can be achieved using a two-layer control structure where the DRTO level determines the optimal reactor temperature profile and the NMPC level follows the trajectory by minimizing the cooling water flow rate. To further motivate the development and use of advanced control structure, the potential economic advantages of this method over the current fixed recipe approach are discussed.

Preface

This thesis concludes the two year international Master's program within Industrial Chemistry and Biotechnology at the Norwegian University of Science and Technology (NTNU), resulting in an M.Sc. in Chemical Engineering. It was written during the spring of 2018 in collaboration with Cybernetica AS.

I wish to sincerely thank everyone at Cybernetica AS for giving me the opportunity to write my thesis for them and for creating a pleasant working environment. I would particularly like to thank Peter Singstad and Kasper Linnestad for their supervision. Kasper provided daily assistance that was invaluable in learning the software tools utilized in this project and provided helpful tips on how to design a well functioning advanced control structure. Marlene Louise Lund also deserves thanks for her work done on the development of the model and offline optimization formulation that was used in this thesis. Thanks are also due to my NTNU supervisor Associate Professor Johannes Jäschke for his assistance.

I would also like to thank my parents, Bob and Meg Hall, for always encouraging me; their support throughout my life has helped me to succeed in my endeavors. Last, but certainly not least, I want to thank my fiancè Sigve Karolius for being my inspiration to move to Norway and earn my Master's degree. He has been instrumental in my success at NTNU and my life would not be the same without him.

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).

Signature:

Place and Date: Trondheim - Gløshaugen, June 2018

Table of Contents

List of Figures

List of Tables

Latin Symbols

Symbol	Description	Unit
$A^{\bullet}$	Radical Diels-Alder adduct	–
A	Chemical component	–
A_l	Gel effect testing parameter	–
A	Gel effect tuning parameter	–
$\mathcal{A}$	Active set	–
a	Gel effect tuning parameter	–
[A]	Concentration of A	mol
A_{cr}	Gel effect critical point parameter	–
AH	Diels-Alder adduct	–
$\boldsymbol{a}_j$	$M \times n$-dimensional real matrix	–
B	Chemical component	–
[B]	Concentration of B	mol
B	Glass effect tuning parameter	–
$\boldsymbol{b}_j$	M-dimensional real vector	–
C	Cage effect tuning parameter	–
$\boldsymbol{c}_j$	Vector of constraints for j	–
$c_{p,c}$	Specific heat capacity of cooling fluid	$\mathrm{J\,kg^{-1}\,K^{-1}}$
$c_{p,i}$	Specific heat capacity of component i	$\mathrm{J\,kg^{-1}\,K^{-1}}$
$c_{p,R}$	Specific heat capacity of reactor contents	$\mathrm{J\,kg^{-1}\,K^{-1}}$
$c_{p,tot}$	Total heat capacity	$\mathrm{J\,K^{-1}}$
$c_{p,V}$	Specific heat capacity of reactor vessel	$\mathrm{J\,kg^{-1}\,K^{-1}}$
d	Linearized feasible direction	–
$\boldsymbol{d}$	$M \times n$-dimensional real vector	–
E	Total energy	J
$\mathcal{E}$	Equality constraints	–
E_{cr}	Gel effect critical point exponent parameter	$\mathrm{J\,mol^{-1}}$
$\frac{dE}{dt}$	Derivative of total energy	$\mathrm{J\,s^{-1}}$
E_{dm}	Thermal initiation activation energy	$\mathrm{J\,mol^{-1}}$
E_i	Activation energy	$\mathrm{J\,mol^{-1}}$
$\bar{E}_{in}$	Energy in from convection	$\mathrm{J\,kg^{-1}}$
E_p	Propagation activation energy	$\mathrm{J\,mol^{-1}}$
E_{tc}	Termination activation energy	$\mathrm{J\,mol^{-1}}$
E_{trM}	Transfer to monomer activation energy	$\mathrm{J\,mol^{-1}}$
f	Initiator efficiency	–

$\mathcal{F}$	Feasible set	–
f_{app}	Apparent initiator efficiency	–
ΔH_R	Reaction enthalpy	$\mathrm{J\,mol^{-1}}$
[I]	Initiator concentration	mol
$\mathcal{I}$	Inequality constraints	–
$\mathrm{I}^\bullet$	Initiator radical	–
I	Chemical initiator	–
i	Index of chemical components	–
$[i]$	Concentration of component i	mol
j	Index	–
$\boldsymbol{K}_k$	Kalman gain matrix	–
K	Test variable	–
k	Moment order/Discrete time step	–
k_{-1}	Backward thermal initiation rate constant	$\mathrm{m^3\,mol^{-1}\,s^{-1}}$
k_1	Forward thermal initiation rate constant	$\mathrm{m^3\,mol^{-1}\,s^{-1}}$
K_{cr}	Critical test variable	–
k_d	Decomposition of initiator rate constant	$\mathrm{s^{-1}}$
k_{dm}	Thermal initiation rate constant	$\mathrm{m^6\,mol^{-2}\,s^{-1}}$
k_e	Second order reaction rate constant	$\mathrm{m^3\,mol^{-1}\,s^{-1}}$
k_I	Monomer addition to initiator radical rate constant	$\mathrm{m^3\,mol^{-1}\,s^{-1}}$
$k_{i,0}$	Rate constant frequency factor	–
k_i	Second order rate constant for component i	$\mathrm{m^3\,mol^{-1}\,s^{-1}}$
k_p	Propagation rate constant	$\mathrm{m^3\,mol^{-1}\,s^{-1}}$
$k_{p,app}$	Apparent propagation rate constant	$\mathrm{m^3\,mol^{-1}\,s^{-1}}$
k_{tc}	Termination by combination rate constant	$\mathrm{m^3\,mol^{-1}\,s^{-1}}$
$k_{tc,app}$	Apparent termination rate constant	$\mathrm{m^3\,mol^{-1}\,s^{-1}}$
$k_{tc,rd,max}$	Maximum value of residual diffusion termination rate constant	$\mathrm{m^3\,mol^{-1}\,s^{-1}}$
$k_{tc,rd,min}$	Minimum value of residual diffusion termination rate constant	$\mathrm{m^3\,mol^{-1}\,s^{-1}}$
$k_{tc,rd}$	Residual diffusion termination rate constant	$\mathrm{m^3\,mol^{-1}\,s^{-1}}$
$k_{tc,seg}$	Segmental diffusion-controlled termination rate constant	$\mathrm{m^3\,mol^{-1}\,s^{-1}}$
$k_{tc,trans}$	Translational diffusion-controlled termination rate constant	$\mathrm{m^3\,mol^{-1}\,s^{-1}}$
k_{td}	Termination by disproportionation rate constant	$\mathrm{m^3\,mol^{-1}\,s^{-1}}$
k_{trM}	Transfer to monomer rate constant	$\mathrm{m^3\,mol^{-1}\,s^{-1}}$
k_{trT}	Transfer to chain transfer agent rate constant	$\mathrm{m^3\,mol^{-1}\,s^{-1}}$
$\mathcal{L}$	Lagrangian	–
l	Gel effect tuning parameter	–
[M]	Monomer concentration	mol

$M^{\bullet}$	Monomer radical	–
M	Monomer molecule	–
m_c	Mass of cooling fluid	kg
$\hat{m}_c$	Mass flow rate of cooling fluid	kgs^{-1}
$\hat{m}_{c,max}$	Maximum mass flow rate of cooling fluid	kgs^{-1}
$\hat{m}_{c,min}$	Minimum mass flow rate of cooling fluid	kgs^{-1}
$\hat{m}_{c,r}$	Required mass flow rate of cooling fluid	kgs^{-1}
$\hat{m}_i$	Mass flow rate into reactor	kgs^{-1}
$\overline{M}_n$	Number average molecular weight	$kgmol^{-1}$
$\overline{M}_{n,d}$	Desired number average molecular weight	$kgmol^{-1}$
$\overline{M}_{n,f}$	Final number average molecular weight	$kgmol^{-1}$
$\overline{M}_{n,max}$	Maximum number average molecular weight	$kgmol^{-1}$
$\overline{M}_{n,min}$	Minimum number average molecular weight	$kgmol^{-1}$
$\overline{M}_{n,p}$	Measured number average molecular weight	$kgmol^{-1}$
m_R	Mass of the reactor content	kg
m_V	Mass of the reactor vessel	kg
$\overline{M}_w$	Weight average molecular weight	$kgmol^{-1}$
$\overline{M}_{w,cr}$	Weight average molecular weight at the onset of the translational diffusion effect	$kgmol^{-1}$
$M_{w,M}$	Molecular weight of monomer	$kgmol^{-1}$
$\overline{M}_{w,p}$	Measured weight average molecular weight	$kgmol^{-1}$
N	Prediction horizon	–
n	Length of polymer chain	–
n	Number of variables	–
$\bar{n}$	Number average chain length	–
n_{C5}	Pentane molar mass	mol
$\frac{dn_{C5}}{dt}$	Rate of change of pentane molar mass	$mols^{-1}$
$\frac{dn_I}{dt}$	Rate of change of initiator molar mass	$mols^{-1}$
$\frac{dn_i}{dt}$	Rate of change of molar mass of component i	$mols^{-1}$
$\frac{dn_M}{dt}$	Rate of change of monomer molar mass	$mols^{-1}$
$\hat{n}_I$	Molar flow rate of initiator	$mols^{-1}$
$\hat{n}_i$	Molar flow rate of component i	$mols^{-1}$
$\hat{n}_{\text{in}}$	Molar flow rate into reactor	$mols^{-1}$
$\hat{n}_M$	Molar flow rate of monomer	$mols^{-1}$
$\hat{n}_{C5}$	Molar flow rate of pentane	$mols^{-1}$
n_I	Molar mass of initiator	mol
n_i	Molar mass of component i	mol
$n_{I,0}$	Molar mass of initiator initial loaded	mol
$n_{I,0,max}$	Maximum molar mass of initiator initial loaded	mol
$n_{I,0,min}$	Minimum molar mass of initiator initial loaded	mol
n_M	Monomer molar mass	mol

$n_{\mathrm{P,n}}$	Molar mass of polymer chains with length n	mol
$n_{\boldsymbol{r}}$	Number of decision variables	–
$n_{\boldsymbol{u}}$	Number of inputs	–
$n_{\boldsymbol{x}}$	Number of states	–
$n_{\boldsymbol{y}}$	Number of outputs	–
P	Inactive polymer chain	–
$\boldsymbol{P}$	$N \times n$-dimensional real symmetric matrix	–
$\boldsymbol{p}$	Search direction	–
$\boldsymbol{p}_k$	Search direction at k	–
$\mathrm{P_m}$	Inactive polymer chain of length m	–
$\mathrm{P_n}$	Inactive polymer chain of length n	–
$\mathrm{P_{n+m}}$	Inactive polymer chain of length n+m	–
Q	Added or removed heat	W
q	Penalty on deviation from reference	–
$\boldsymbol{q}$	Real valued n-dimensional vector	–
Q_{amb}	Heat transferred to surroundings	W
Q_J	Heat transferred to cooling jacket	W
$Q_{J,d}$	Cooling demand on jacket	W
$\boldsymbol{Q}_k$	Time-variant quadratic state weighting matrix at k	–
$\boldsymbol{Q}_{k+1}$	Time-variant quadratic state weighting matrix at k+1	–
R	Radical polymer chain	–
R	Gas constant	$\mathrm{J\,mol^{-1}\,K^{-1}}$
$\mathbb{R}$	Set of real numbers	–
r	Decision variable	–
$\boldsymbol{r}$	Vector of decision variables	–
$\mathrm{R_1}$	Radical polymer chain of length 1	–
r_1	Linear weights on output constraint violations	–
$\boldsymbol{r}_1$	Vector of linear weights on output constraint violations	–
$\boldsymbol{r}_2$	Vector of quadratic weights on output constraint violations	–
R_A	Reaction rate of A	$\mathrm{mol\,m^{-3}\,s^{-1}}$
$\boldsymbol{r}^*$	Optimal decision variables	–
R_B	Reaction rate of B	$\mathrm{mol\,m^{-3}\,s^{-1}}$
R_I	Reaction rate of initiator	$\mathrm{mol\,m^{-3}\,s^{-1}}$
R_i	Reaction rate of component i	$\mathrm{mol\,m^{-3}\,s^{-1}}$
$\boldsymbol{R}_k$	Time-variant input weighting matrix	–
$\boldsymbol{r}_k$	Vector of decision variables at iterate k	–
R_M	Reaction rate of monomer	$\mathrm{mol\,m^{-3}\,s^{-1}}$
$\mathrm{R_m}$	Radical polymer chain of length m	–

R_n	Radical polymer chain of length n	–
R_{n+1}	Radical polymer chain of length n+1	–
R_P	Reaction rate of propagation	$mol\,m^{-3}\,s^{-1}$
$\boldsymbol{S}$	Matrix of slack variable weights	–
$\mathcal{S}$	Domain	–
s	Slack variable weight	–
$\boldsymbol{S}_k$	Matrix of slack variable weights at k	–
$T^{\bullet}$	Radical transfer agent molecule	–
T	Transfer agent molecule	–
t	Time	s
T_0	Reference temperature	K
T_{amb}	Ambient temperature	K
Δt	Sampling time	s
Δt_{int}	Integration time step	s
dT_R	Derivative of reactor contents' temperature	Ks^{-1}
$dT_{R,max}$	Maximum derivative of reactor contents' temperature	Ks^{-1}
$dT_{R,min}$	Minimum derivative of reactor contents' temperature	Ks^{-1}
$\frac{d}{dt}T_{R,ref}$	Derivative of reference reactor temperature	Ks^{-1}
$\frac{dT_R}{dt}$	Derivative of reactor content temperature with respect to time	Ks^{-1}
T_{feed}	Feed temperature	K
t_f	Batch time	s
$t_{f,max}$	Maximum batch time	s
$t_{f,min}$	Minimum batch time	s
$T_{g,C5}$	Glass transition temperature of pentane	K
$T_{g,i}$	Glass transition temperature of i	K
$T_{g,M}$	Glass transition temperature of styrene	K
$T_{g,P}$	Glass transition temperature of polystyrene	K
T_J	Jacket temperature	K
$T_{J,i}$	Temperature of cooling fluid at the inlet	K
$T_{J,i,max}$	Maximum temperature of cooling fluid at the inlet	K
$T_{J,i,min}$	Minimum temperature of cooling fluid at the inlet	K
$T_{J,o}$	Temperature of cooling fluid at the outlet	K
$T_{J,p}$	Measured jacket temperature	K
T_R	Reactor temperature	K
$T_{R,0}$	Initial reactor temperature	K
$T_{R,d}$	Desired reactor temperature	K
$T_{R,max}$	Maximum reactor temperature	K
$T_{R,min}$	Minimum reactor temperature	K

$T_{R,p}$	Measured reactor temperature	K
$\boldsymbol{u}$	Vector of inputs	–
$\boldsymbol{u}_{-1}$	Vector of initial inputs	–
$(UA)_{amb}$	Overall heat transfer coefficient, heat loss	$\mathrm{W\,K^{-1}}$
$(UA)_J$	Overall heat transfer coefficient, cooling jacket	$\mathrm{W\,K^{-1}}$
$\boldsymbol{\Delta u}_k$	Input change rate at k	–
$\Delta \boldsymbol{u}^{ss}$	Composite vector of steady state inputs for each step of the DRTO	–
$\hat{\boldsymbol{u}}^{DRTO}$	Vector of DRTO inputs	–
$\hat{\boldsymbol{u}}_k^{DRTO}$	Vector of DRTO inputs at k	–
$\Delta\hat{\boldsymbol{u}}^{NMPC}$	Vector of NMPC input changes	–
$\Delta\hat{\boldsymbol{u}}_k^{NMPC}$	Vector of NMPC input changes at k	–
$\hat{\boldsymbol{u}}^{NMPC}$	Vector of NMPC inputs	–
$\hat{\boldsymbol{u}}_k^{NMPC}$	Vector of NMPC inputs at k	–
$\hat{\boldsymbol{u}}_{k-1}^{NMPC}$	Vector of NMPC inputs at k-1	–
$\boldsymbol{u}(t)$	Vector of time dependent inputs	–
$\boldsymbol{u}_k$	Vector of inputs at k	–
V	Polymer phase volume	$\mathrm{m^3}$
V_f	Total free volume	–
$V_{f,C5}$	Free volume of pentane	–
$V_{f,cr}$	Free volume of mixture for translational diffusion onset	–
$V_{f,cr,d}$	Critical free volume for cage effect onset	–
$V_{f,cr,p}$	Critical free volume for glass effect onset	–
$V_{f,M}$	Free volume of styrene	–
$V_{f,P}$	Free volume of polystyrene	–
$\boldsymbol{V}_{k-1}$	Process noise covariance at k-1	–
$\bar{\boldsymbol{v}}_{k-1}$	Mean process noise	–
$V_{m,C5}$	Molar volume of pentane	$\mathrm{m^3\,mol^{-1}}$
$V_{m,i}$	Molar volume of component i	$\mathrm{m^3\,mol^{-1}}$
$V_{m,M}$	Molar volume of styrene	$\mathrm{m^3\,mol^{-1}}$
$V_{m,P}$	Molar volume of polystyrene	$\mathrm{m^3\,mol^{-1}}$
$\boldsymbol{w}$	Directions inside the linearized feasible set	–
W_{ag}	Agitation work	$\mathrm{J\,s^{-1}}$
$\bar{\boldsymbol{w}}_k$	Mean measurement noise at k	–
$\boldsymbol{W}_k$	Measurement noise covariance at k	–
W_s	Shaft work	$\mathrm{J\,s^{-1}}$
X	Overall conversion	–
$\boldsymbol{x}$	Vector of states	–
$\boldsymbol{x}_0$	Vector of initial states	–
$\bar{\boldsymbol{x}}_k$	*A priori* state estimate at k	–
X_d	Desired overall conversion	–

$\dot{\boldsymbol{x}}(t)$	Time dependent process model	–
$\hat{\boldsymbol{x}}^{DRTO}$	Vector of DRTO states	–
$\hat{\boldsymbol{x}}_k^{DRTO}$	Vector of DRTO states at k	–
$\hat{\boldsymbol{x}}_k$	Vector of estimated states at k	–
$\hat{\boldsymbol{x}}_{k-1}$	Vector of predicted states at $k-1$	–
$\hat{\boldsymbol{x}}_{k+1}$	Vector of estimated states at $k+1$	–
$\hat{\boldsymbol{x}}_k^{NMPC}$	Vector of NMPC states at k	–
$\hat{\boldsymbol{x}}_{k+1}^{NMPC}$	Vector of NMPC states at $k+1$	–
X_{inst}	Instantaneous conversion	–
$\boldsymbol{x}_k$	Vector of states at k	–
$\overline{\boldsymbol{X}}_{k-1}$	*A priori* state covariance	–
$\boldsymbol{x}_{k+1}$	Vector of states at $k+1$	–
$\boldsymbol{x}_{k+1}^{ref}$	Vector of reference states at $k+1$	–
$\boldsymbol{x}(t)$	Vector of time dependent states	–
$\boldsymbol{y}$	Vector of plant measurements	–
$\boldsymbol{y}$	Vector of measurements	–
$\bar{\boldsymbol{y}}_k$	*A priori* measurements at k	–
$\hat{\boldsymbol{y}}_k$	Vector of estimated measurements at k	–
$\hat{\boldsymbol{y}}^{NMPC}$	Vector of NMPC outputs	–
$\hat{\boldsymbol{y}}_k^{NMPC}$	Vector of NMPC outputs at k	–
$\hat{\boldsymbol{y}}^{SP}$	Composite vector of NMPC set point trajectories	–
$\hat{\boldsymbol{y}}(t)$	Vector of time-dependent estimated measurements	–
$\boldsymbol{y}_k$	Vector of plant measurements at k	–
$\boldsymbol{y}(t)$	Vector of time-dependent measurements	–
$\bar{\boldsymbol{z}}_k$	Vector of predicted outputs at k	–
$\hat{\boldsymbol{z}}_k^{DRTO}$	Vector of DRTO outputs at k	–
$\hat{\boldsymbol{z}}_k$	Vector of updated outputs at k	–
$\boldsymbol{z}_k$	Vector of outputs at k	–
$\boldsymbol{z}$	Vector of outputs	–
$\boldsymbol{z}(t)$	Vector of time-dependent outputs	–

Greek Symbols

Symbol	Description	Unit
α	Convex function parameter	–
α_{C5}	Fractional free volume of pentane	–
α_i	Fractional free volume of component i	–
α_M	Fractional free volume of monomer	–
α_P	Fractional free volume of polymer	–
δ_c	Segmental diffusion parameter for styrene	–
ϵ	Slack variable	–
$\boldsymbol{\varepsilon}$	Deviation upper bound	–
$\boldsymbol{\epsilon}$	Vector of slack variables	–
$\boldsymbol{\eta}$	Decision variables in parameter estimation	–
∇	Gradient	–
$\nabla_{\boldsymbol{r}}$	Gradient with respect to $\boldsymbol{r}$	–
∇^2	Hessian	–
$\nabla_{\boldsymbol{rr}}$	Hessian with respect to $\boldsymbol{r}$	–
$\boldsymbol{\lambda}$	Vector of Lagrange multipliers	–
λ_0	Zeroth order moment, living chains	mol
λ_1	First order moment, living chains	mol
λ_2	Second order moment, living chains	mol
$\boldsymbol{\lambda}^*$	Vector of optimal of Lagrange multipliers	–
λ_j	Lagrange multiplier for constraint j	–
$\boldsymbol{\lambda}_{\boldsymbol{j}}^*$	Optimal Lagrange multiplier of constraint vector j	–
λ_k	K^{th} order moment, living chains	mol
μ_0	Zeroth order moment, dead chains	mol
μ_1	First order moment, dead chains	mol
μ_2	Second order moment, dead chains	mol
μ_k	K^{th} order moment, dead chains	mol
$\boldsymbol{\rho}$	MPC tuning parameter	–
ρ_{C5}	Density of pentane	$\mathrm{kg\,m^{-3}}$
ρ_I	Density of initiator	$\mathrm{kg\,m^{-3}}$
ρ_M	Density of monomer, styrene	$\mathrm{kg\,m^{-3}}$
ρ_P	Density of polymer, polystyrene	$\mathrm{kg\,m^{-3}}$
ρ_W	Density of water	$\mathrm{kg\,m^{-3}}$
$d\tau$	Derivative of generic integrator	–

τ	Generic integrator	–
$\boldsymbol{\theta}$	Time invariant parameters	–
$\bar{\boldsymbol{\theta}}_k$	Predicted parameters at k	–
$\hat{\boldsymbol{\theta}}_{\boldsymbol{k}}$	Updated parameter estimations	–
$\hat{\boldsymbol{\theta}}_{k-1}$	Estimated parameters at $k-1$	–
$\hat{\boldsymbol{\omega}}_{\boldsymbol{k-1}}$	Mean parameter noise at k-1	–

FUNCTIONS

Symbol	Description
$\mathcal{C}(\boldsymbol{r}^*, \boldsymbol{\lambda}^*)$	Critical cone
$\boldsymbol{c}_j(\boldsymbol{r})$	Constraint function of index j
$\boldsymbol{c}_{\mathcal{E}}(\boldsymbol{r})$	Equality constraint functions
$\boldsymbol{c}_{\mathcal{I}}(\boldsymbol{r})$	Inequality constraint functions
f	Scalar objective function
$\boldsymbol{f}$	State prediction function
$\widetilde{\boldsymbol{f}}^{DRTO}$	DRTO dynamic prediction model
$\boldsymbol{f}^{NMPC}$	NMPC dynamic prediction model
$\boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k)$	Plant model
$\boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\theta})$	Plant model with parameters
$\boldsymbol{f}(\hat{\boldsymbol{x}}_{k-1}, \hat{\boldsymbol{\theta}}_{k-1}, \boldsymbol{u}_{k-1}, \bar{\boldsymbol{v}}_{k-1})$	Augmented Kalman filter state prediction function
$\boldsymbol{f}(\hat{\boldsymbol{x}}_{k-1}, \boldsymbol{\theta}, \boldsymbol{u}_{k-1}, \bar{\boldsymbol{v}}_{k-1})$	Kalman filter state prediction function
$\boldsymbol{f}(t, \boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{\theta})$	Time dependent plant model
$\boldsymbol{f}(k, \boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\theta})$	Step dependent plant model
$f(\boldsymbol{r})$	Objective function n-dimensional vector
$f(\boldsymbol{r}^*)$	Objective function at optimal solution
$\boldsymbol{g}$	Measurement prediction function
$\boldsymbol{g}^{DRTO}$	DRTO output constraints
$\boldsymbol{g}^{NMPC}$	NMPC inequality input constraints
$\boldsymbol{g}^{SP}$	NMPC inequality setpoint trajectories
$\boldsymbol{g}(\bar{\boldsymbol{x}}_k, \boldsymbol{\theta}, \boldsymbol{u}_{k-1})$	Kalman filter measurement prediction function
$\boldsymbol{g}(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\theta})$	Predicted measurements model
$\boldsymbol{g}(, \bar{\boldsymbol{\theta}}_k, \boldsymbol{u}_{k-1})$	Measurement prediction function
$\boldsymbol{g}(t, \boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{\theta})$	Measurement prediction model
$\boldsymbol{h}(\boldsymbol{x}_k, \boldsymbol{u}_k)$	Output vector function
$\boldsymbol{h}^{DRTO}$	DRTO algebraic equations of the prediction model
$\boldsymbol{h}^{SP}$	NMPC equality setpoint trajectories
$\boldsymbol{h}^{NMPC}$	NMPC equality relations
$\boldsymbol{h}(\bar{\boldsymbol{x}}_k, \boldsymbol{\theta}, \boldsymbol{u}_{k-1})$	Kalman filter decision prediction function
$\boldsymbol{h}(\hat{\boldsymbol{x}}_k, \boldsymbol{\theta}, \boldsymbol{u}_{k-1})$	Kalman filter decision prediction function
$\boldsymbol{h}(t, \boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{\theta})$	Time dependent output model
Φ^{DRTO}	DRTO objective function
ϕ_{NMPC}	NMPC quadratic cost function

ACRONYMS

AKF augmented Kalman filter
APC advanced process control
CLD chain length distribution
CTA chain transfer agent
CV controlled variable
DAE differential algebraic equation
DCS distributed control system
DRTO dynamic real-time optimization
DV disturbance variable
EKF extended Kalman filter
EPS expandable polystyrene
GUI graphical user interface
KF Kalman filter
KKT Karush-Kuhn-Tucker conditions
LICQ linear independence constraint qualification
LP linear programming
MHE moving horizon estimator
MPC model predictive control
MV manipulated variable
MWD molecular weight distribution
NLP nonlinear programming
NMPC nonlinear model predictive control
ODE ordinary differential equation
OPC open platform communication
PI polydispersity index
PSSA pseudo-steady state assumption
QP quadratic programming
SQP sequential quadratic programming

Chapter 1

Introduction

1.1 Motivation

The favorable properties of polymer products has resulted in a soaring increase in their demand. In particular, one of the biggest commodity polymer products in the world is expandable polystyrene (EPS) [25]. The polymerization of styrene was discovered in 1839 but little commercial use was found until 1937 when an American chemist named Robert Dreisbach and others at Dow Chemical Company managed to obtain a purified styrene monomer and designed a pilot polymerization process [33]. Since then, it has become the fourth largest thermoplastic by production volume with applications in numerous markets such as packaging, consumer/institutional goods, electrical/electronic goods, building/construction, furniture, industrial/machinery and transportation [34]. The global market for EPS is forecasted to be worth more than US$18 billion by 2022 [26].

As the polymer industry becomes more competitive, manufacturers are facing mounting pressure to reduce production costs while simultaneously having to comply with increasingly stringent production constraints such as quality requirements and environmental regulations [16]. To deal with these constraints, a considerable effort has been put forth into improving the manufacturing process. One solution is to lower the production time and optimally distribute the available resources to ensure the desired polymer specifications are met [21]. These goals can be achieved through the application of advanced control structures to the process; the controller forces the system to stay near the optimal conditions that will result in the desired product quality in the shortest amount of time. However, given that polymer processes have highly nonlinear dynamics and involve complex reaction mechanisms, this is easier said than done [11].

The performance of a model-based controller is highly dependent on the quality of the model; therefore, it is essential that the model accurately capture the process's behavior [11]. These models are constructed using systems of differential algebraic equation (DAE)s, which consist of differential equations that describe the dynamic behavior of the system such as mass and energy balances, and algebraic equations that ensure physical and thermodynamic relations hold. Specifically for polymerization processes, models that are able to predict the polymer quality in terms of the

reactor operating conditions are the key to the efficient production of high quality, tailored polymers in addition to the improvement of plant operability and economics [16]. The development and validation of such dynamic models is time consuming and expensive; this is the main reason for the limited availability of such models for industrial applications [24].

Despite the work involved in constructing a valid model, it has been demonstrated that the use of advanced control techniques can result in a reduction in the batch time [21]. It is common for nonlinear systems to use an advanced process control algorithm known as nonlinear model predictive control (NMPC). This algorithm works to determine the optimal values for a selected objective function that is subject to the process model and constraints. The solution of this problem becomes the inputs to the process.

An upper control level can be added on top of this to further optimize the system to additional criteria; for dynamic systems, a dynamic real-time optimization (DRTO) algorithm is applied in this layer. Online measurements are used to calculate optimal control trajectories for the length of the batch; these trajectories are then passed to the NMPC layer below to be used as inputs. It has been shown in other works that a two-level strategy results in improved performance for a free-radical solution polymerization of styrene in a jacketed reactor [12]. Therefore, this work will consider the implementation of an two-level advanced control structure for the production of EPS.

1.2 Scope of Work

The aim of this work is to investigate the possibility of reducing the batch time of the production of EPS while still producing the desired polymer properties. Previous work was done by Marlene Lund in the development of the process model, which is utilized in the control structure developed in this thesis [21]. She also constructed the control structure that is used in the offline optimization performed in Chapter 4.

The specific goals for this thesis are the following:

a) Alter the process model to reflect the new objectives.

b) Determine the optimal monomer to initiator ratio, initial reactor temperature, and reactor temperature profile using offline optimization.

c) Implement a two-level control structure, consisting of a NMPC lower level and a DRTO upper level, in Cybernetica's software.

d) Determine what each level will control; define the manipulated variables, controlled variables, constraints and setpoints for each layer.

e) Demonstrate that the two-layers communicate and work together as desired.

f) Examine the effect that disturbances in the cooling water temperature and the system's cooling capacity at different points have on the batch time and produced polymer quality.

g) Perform a cost-benefit analysis to demonstrate the potential benefits of using an advanced control structure over the current fix recipe method.

1.3 Outline

Given the importance of the model in the construction of a good model-based controller, a short introduction to polymerization is given in Chapter 2 outlining the ideas that are utilized in the construction of the process model for EPS production. Polymers can be synthesized through a variety of reaction mechanisms such as addition and step growth reactions; in this work, emphasis is placed on a type of addition reaction known as free radical polymerization. The polymer quality is determined using an approach known as the method of moments. This is a statistical approach that tracks the moment balances of the living and dead chains throughout the batch. Free radical polymerization reactions experience a decrease in the reaction volume throughout the batch, which results in diffusion limiting the reaction rates; this can be modeled using free volume theory to calculate the reaction rates. Polymer production is typically carried out in semi-batch reactors so a brief introduction to semi-batch modeling is given.

Using the process model, the control structure can be designed utilizing the concepts covered in Chapter 3. Polymerization processes typically cannot be represented with linear models, so it is necessary to use a nonlinear formulation for the optimization problem. While there are many ways to solve such problems, the most common method is discussed in this chapter. Advanced process control combines optimization of dynamic systems with feedback where a process model is used to predict the process behavior; the general structure of such controllers is outlined here. The NMPC algorithm is emphasized because it works to optimize nonlinear process models. On top of this advanced control layer, another level known as supervisory control can be added to further optimize the process subject to additional goals. For dynamic models, a DRTO algorithm should be used in this level. This layer works to optimize the process in real time and provides new setpoints to the lower layer. Within these control structures, parameter estimation and state estimation can be included. The parameters initially selected may not fit the process exactly so online parameter updating can be added to the control structure that uses the online measurements to update the parameters used in the model. Often it is difficult to directly measure the states online so these values are estimated; there are two methods commonly used to estimate these values. The accuracy of these estimations

methods can impact the fit of the model so it is important to select them carefully.

Next the case study of the semi-batch polymerization of expandable polystyrene is introduced in Chapter 4. A first principles based model of this process is constructed and implemented in the programming language C where a numerical solution method is used to solve the differential equations. Offline optimization is performed on the model to determine the optimal monomer to initiator ratio, starting reactor temperature, and reactor temperature profile that results in the shortest batch time while still producing the desired product; these values are used in the development of the two-level control structure.

Chapter 5 explains how the NMPC lower layer is constructed and covers the problem formulation, what constraints are included, and the controller tunings. The aim of this level is to optimize the cooling water flow rate to keep the reactor temperature near an optimal profile that is determined by the upper supervisory layer. The development of this lower level is carried out in two main stages: 1) the controller is designed to follow a constant setpoint and 2) the controller is altered to track a fixed reference trajectory for the reactor temperature. For a constant setpoint, the selected controller tunings and input blocking are tested using different cooling water temperatures and changing the reactor temperature setpoint. The layer is edited to follow the optimal reactor temperature profile identified in the offline optimization. Assorted cooling water inlet temperatures are used to validate that the controller can handle possible seasonal changes.

The supervisory DRTO layer development is explained in Chapter 6. First a short introduction to the use of supervisory layers for semi-batch processes is given to motivate the use of an additional layer for this case study. The selected DRTO problem formulation, constraints, and tunings are then stated. The simulation results are presented, illustrating that the two-layers work together to achieve the desired outcome. The controller is then tested to examine the consequences of disturbances in the cooling capacity and the cooling water temperature at various points in the batch.

Chapter 7 provides a comparison of each of the stages of development plus a cost benefit analysis. Comparing the results of the different stages of development illustrates how similar the offline results are to the two-level control structure results under ideal conditions. A cost-benefit analysis is conducted to motivate the use of a two-layer control structure over the use of a fixed recipe, which is currently the common approach used for batch and semi-batch production.

Finally, Chapter 8 summarizes the findings of this work and briefly considers further work. This includes potential ways of improving the model and control structure, along with the next steps that should be done to implement this control structure in a real plant.

CHAPTER 2

POLYMERIZATION

Polymers are utilized extensively in applications from food packaging, clothing, home furnishing, transportation, medical devices, to information technology. Fibers like silk, wool, and cotton are examples of naturally occurring polymers that have been used for thousands of years [20]. Inspired by these natural polymers, scientists have designed synthetic polymers that can be manufactured to have specific properties. These favorable properties plus the relative ease of production has resulted in a rapid increase in the demand for polymers. These processes involve complex reaction mechanisms and highly nonlinear dynamics; consequently, from a control and optimization perspective, these systems are demanding [11].

In order to manufacture polymers in a safe and optimal manner, the controller requires a a model that accurately captures the process dynamics. Understanding the reaction mechanisms is a big part of this model development. Section 2.1 introduces basic polymer concepts and briefly discusses the formation process; free radical polymerization is the focus of the formation discussion. In this type of polymerization, diffusion-limitations arise as a result of the increased viscosity; these can have a large impact on the rate of reaction, as well as affecting the final product composition, so it is necessary to consider them when developing a model.

Most polymerization production is carried out in batch or semi-batch reactor; therefore, the model will include the mass and energy balance equations on a reactor, which are outlined in Section 2.2. The polymerization is modeled using a statistical approach known as the method of moments, which is discussed in Section 2.3. Here the living and dead chain moments are related to the average molecular weights which describe the molecular weight distribution of the product. Also considered in this section are the diffusional limitations, which are modeled using free volume theory.

2.1 Polymers and Polymerization

Polymers are defined as materials of very high molecular weight and consist of several structural units bound together by covalent bonds. They are created through the chemical reaction of smaller molecular compounds known as *monomers*. Polymers can be made up of hundreds, thousands, or even tens of thousands of monomers

[16]. In order to form polymers, monomers need to have reactive functional groups or bonds higher than single bonds [20]. For example, the polymer polystyrene is made up of repeating units of styrene, $[-C_8H_8-C_8H_8-]_n$, as illustrated below in Figure 2.1. Here the subscript n denotes the number of monomer units in the polymer and is referred to as the *chain length*.

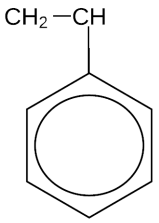


Figure 2.1: Styrene monomer [20]

Polymers are classified using different criteria but four possible options are: chemical nature, molecular structure, polymer chain growth mechanism, or the type of polymerization. Polymers can have straight chains, branched chains, or chain networks [21].

Polymerization refers to the process where monomers react to form polymers. There are two main types of kinetic mechanisms: *step-growth polymerization* or *chain polymerization*. Step-growth polymerization proceeds by reactions between two different functional groups. In chain polymerization, the polymer chains grow by the repeated rapid addition of monomer molecules to an active chain. This type of polymerization can proceed via free-radical, ionic, group-transfer or coordination mechanisms; free-radical polymerization is covered in more detail in Section 2.1.1. In this case, a chain initiator is required for the formation of primary active centers before polymerization can begin [16].

Polymerization processes can be classified into *homogeneous* and *heterogeneous* systems. In heterogeneous systems, the polymer is either insoluble in the monomer phase or the polymerization involves the presence of different phases [16]. This type of polymerization can be identified by the following process characteristics: suspension, emulsion, precipitation, or interfacial and solution polycondensations. In suspension polymerization, the process occurs in small beads in the continuous phase; emulsion polymerization has the reaction take place in micelles in a water phase. Homogenous polymerization processes are where all the reactants are mutually soluble and compatible with the produced polymer. Bulk and solution polymerization are the two main categories of homogenous systems. In *bulk polymerization*, the feed to the reactor consists of pure monomer with small amounts

of dissolved catalyst and molecular weight modifiers; this type of polymerization is typically exothermic. The production of polystrene is an example of this type of polymerization since styrene is miscible with its polymer [16]. Typical features of homogeneous bulk polymerization are the high viscosity of the reaction mixture and poor heat transfer characteristics; this is particularly true with homogeneous free-radical bulk polymerization where the viscosity increases dramatically [16]. This increase in viscosity can lead to diffusional factors playing a role in limiting the reaction rates, altering the product properties; these diffusional limitations are discussed in Section 2.1.2. To avoid associated problems with the high viscosity, a solvent can be added resulting in a *solution polymerization* process [16]. Both of these processes contain a single phase since the polymer is insoluble in its monomer.

2.1.1 Free-Radical Polymerization

Free-radical polymerization is driven by the addition of a monomer molecule to a radical active center. A free radical is an unpaired electron that gives rise to highly reactive chemical compounds [21]. In this type of polymerization, there are three basic reaction types that occur simultaneously: initiation, propagation, and termination. Initiation reactions continuously generate radicals. Propagation reactions are responsible for the growth of polymer chains by monomer addition to a radical center. Termination reactions occur between two radical centers and result in a net consumption of radicals. Another reaction type known as chain transfer may also occur where a free radical is transferred from one molecule to another [16]. This reaction type is not necessary for polymerization to occur.

Initiation is caused by a chemical compound that has the ability to form radical molecules. Commonly peroxides are utilized because they have a covalent oxygen bond separating two organic groups; the decomposition of this bond creates two radical molecules as shown in Reaction 2.1.

$$\mathrm{I} \xrightarrow{k_d} 2f\,\mathrm{I}^{\bullet} \tag{2.1}$$

Here I represents the chemical initiator, k_d is the reaction rate constant of the decomposition reaction, f is the initiator efficiency (a number between 0 and 1), and $\mathrm{I}^{\bullet}$ is the primary radical. A *mono-functional* initiator has one peroxide group and is the simplest form of peroxide initiators [21]. The primary radicals $\mathrm{I}^{\bullet}$ that are generated in the decomposition reaction combine with a monomer M producing a radical polymer chain of length one R_1 according to the following reaction [21]:

$$\mathrm{I}^{\bullet} + \mathrm{M} \xrightarrow{k_I} \mathrm{R}_1 \tag{2.2}$$

The rate constant k_I determines the rate of monomer addition to the initiator radical. The active chains produced in this initiation step will then undergo propagation.

Propagation occurs simultaneously to radical chains of all sizes and occurs at a rate k_p. This reaction is given by [21]:

$$R_n + M \xrightarrow{k_p} R_{n+1} \tag{2.3}$$

This results in a radical chain with a length increased by one monomer.

A radical chain R can be deactivated in several different ways to produce an inactive polymer chain P. *Termination by combination* occurs when two radical chains meet and form a covalent carbon-carbon bond from their radicals. This results in a dead chain where the chain length is the sum of the lengths of the two chains. This type of termination is shown in Reaction 2.4 [21].

$$R_n + R_m \xrightarrow{k_{tc}} P_{n+m} \tag{2.4}$$

Termination by disproportionation occurs when two radical chains meet and one of the chains transfers its radical to the other chain. This yields two dead polymer chains as given by [21]:

$$R_n + R_m \xrightarrow{k_{td}} P_n + P_m \tag{2.5}$$

Chain transfer is another way of ending a growing polymer chain. This type of reaction occurs between active radical chains and a chain transfer agent (CTA), represented by T. Many of the different components in the reaction mixture can serve as the CTA. Any impurities present in the reaction mixture may act as an unintended transfer agent. This effect can be exploited to control the chain length of the produced polymer through the use of a CTA [21]. Reaction 2.6 occurs at a rate of k_{trT}:

$$R_n + T \xrightarrow{k_{trT}} P_n + T^{\bullet} \tag{2.6}$$

This results in a dead chain polymer and a radical transfer agent.

2.1.2 Diffusion-Controlled Reactions

In bulk polymerization, the viscosity increases as the polymer chains grow, which can lead to a reduction in the reaction volume throughout the batch time. As a consequence, the polymerization reactions shown in Reactions 2.1-2.6 can become diffusion-controlled. There are three main side-effects of diffusion-controlled polymerization: gel effect, cage effect, and glass effect. Each of these effects is caused by the diffusion influencing a component of the polymerization process.

Diffusion-limitations have the largest impact on the termination reactions, resulting in what is known as the *gel effect*. The polymer chain mobility is reduced due to the increased density, causing a decrease in the termination rate constant. The drop in the reaction rate results in a broader molecular weight distribution; therefore, the final product has altered properties [8]. Sever gel effect can pose a safety risk

since it can lead to a dominant propagation reaction, which generates the majority of the heat in this reaction system. This generated heat could exceed the capacity of the cooling system and, consequently, cause the temperature inside the reactor to increase. This will in turn result in faster reaction rates and eventually, thermal runaway. Therefore it is critical that this effect is controlled to prevent unsafe conditions from resulting.

The *cage effect* is caused by the diffusional limitations affecting the chemical initiator's ability to polymerize monomers. After sufficient energy is supplied to the initiator molecules, the covalent bond is cleaved. The two fragments are surrounded by the reaction mass, forming a cage around them. This means that not all of the radicals can approach the monomer molecules to initiate reaction. Because of the close proximity of the generated radicals, only some of them can escape the cage and react with the monomer molecules. Inside this cage, an active initiator is likely to either self-terminate or react with other neighboring molecules as opposed to initiating new chains [8]. This leads to a decrease in the monomer addition to initiator radical rate constant; this limitation results in the introduction of an initiator's efficiency term to the reaction equations. The efficiency has to be considered when deciding how much initiator to add to the reactor because if insufficient initiator is added, the desired conversion and polymer properties cannot be achieved.

The final effect of the diffusion-limitation is on the *glass effect*, which alters the propagation rate constant. It is caused by the decrease in the mobility of monomer molecules. This only occurs at temperatures below the glass transition temperature of the polymer, which is the lower temperature limit that induces a structural change in the polymer [8]. If the temperature is close to the glass transition temperature, the probability of the collisions between active polymer chains and free monomer molecules can drastically decrease due to the reduction in monomer mobility. This causes the propagation rate to decrease and along with it, alter the polymer properties [21]. Consequently, this effect should be mitigated to achieve the desired polymer properties can be produced.

2.2 Semi-Batch Reactor Modeling

Batch and semi-batch reactors are typically used in the production of fine or specialty chemicals, polymers, and other high value products in industry [24]. The only difference between these two reactor types is that in a semi-batch reactor, components can be added throughout the batch whereas in a batch reactor all the components are added at the start of the batch. For the type of polymerization considered in this work, semi-batch reactors are more commonly used. The reactor is initially charged with some reactants at time zero and others are added throughout the reaction. No product is withdrawn until the entire reagent has been added and the reaction has proceeded to the required-extent. In many applications, semi-batch

reactors involve a substantial increase in the reaction mixture volume during a production cycle [18].

Semi-batch reactors have the advantage of good temperature control in addition to the ability of minimizing unwanted side reactions by maintaining a low concentration of one of the reactants. Another advantage is the concentration of a reactant can be kept relatively low or high to the advantage of suppressing side reactions, which improves the product yield. It is typical to use the mass and energy balances, along with the rate of conversion, to model these reactors.

2.2.1 Species and Energy Balances

For reactive processes where components are removed and generated by reactions within the mixture, it is convenient to write the mass balances in the species form shown here [21]. For a semi-batch reactor, the species balance for an arbitrary component i in the system and the energy balance are given by:

$$\text{Species balance:} \quad \frac{dn_i}{dt} = R_i V + \hat{n}_i \tag{2.7}$$

$$\text{Reactor energy balance:} \quad \frac{dE}{dt} = \hat{m}_i \bar{E}_{in} + Q + W_s \tag{2.8}$$

In the species balance, n_i is the amount of component i present, R_i is the reaction rate, $\hat{n}_i$ is the molar feed flow, and V is the reaction mixture volume. In the energy balance, E is the total energy, $\bar{E}_{in}$ is the energy per mass from the inlet, $\hat{m}_i$ is the mass flow rate of the inlet, Q is the added heat, and W_s is the shaft work.

The reaction rate R_i represents the frequency of formation and removal of a component in the reaction mixture. For a simple first order reaction $\text{A} \xrightarrow{k_e} \text{B}$, the reaction rates for each of the components using the law of mass action would be expressed as:

$$R_A = -k_e[A] \tag{2.9}$$

$$R_B = k_e[B] \tag{2.10}$$

Here k_e is the reaction rate constant and $[i]$ represents the concentration of component i. The overall reaction rate term includes all generated and consumed components in the reactor.

The energy balance can be transformed into a function of temperature by utilizing thermodynamic relations. This is commonly done because temperature is easier to measure and understand than enthalpy. The energy balance for the reactor in temperature form becomes:

$$\frac{dT_R}{dt} = \frac{\hat{m}_i \bar{E}_{in} + Q_J + Q_{amb} + W_s - \Delta H_R R_i V}{m_V c_{p,V} + m_R c_{p,R}} \tag{2.11}$$

Here m_V and $c_{p,V}$ are the mass and heat capacity of the reactor vessel; m_R and $c_{p,R}$ are the corresponding mass and heat capacity of the reactor contents. The energy introduced to the reactor through the feed stream is represented by $\hat{m}_i \bar{E}_{in}$. The added heat Q has been broken into the ambient contributions Q_{amb} and the cooling or heating system contribution Q_J [21]. The heat of reaction is represented by $\Delta H_R R_i V$; here ΔH_R is the change in the reaction enthalpy, R_i is the reaction rate, and V is the reaction volume. A full derivation of the reactor energy balance can be found in Appendix B Section B.3.

If a reactor should be controlled to a setpoint or trajectory, the temperature equation can be reformulated to calculate the heating or cooling demand $Q_{J,d}$.

$$Q_{J,d} = c_{p,tot} \frac{d}{dt} T_{R,ref} + \Delta H_R R_i V - \hat{m}_i \bar{E}_{in} - Q_{amb} - W_s \tag{2.12}$$

where $\frac{d}{dt} T_{R,ref}$ is the desired rate change of the reactor content's temperature and $c_{p,tot}$ is the total heat capacity. Note that in the case of *isothermal operation*, the temperature derivative will be zero.

2.2.2 Conversion

The conversion is defined as the ratio between the reacted amount and the fed amount of component i; in other words, it represents the amount of reactant that has been converted to product [6]. For semi-batch processes, two conversion types are calculated: the overall conversion and the *instantaneous conversion*. The two conversion types are formulated for a semi-batch reactor operated between time $t = 0$ and $t = t_f$ [9]:

$$\text{Overall conversion:} \qquad X = \frac{{n_i}^0 + \int_0^t \hat{n}_{\text{in}}(\tau) d\tau - n_i(t)}{{n_i}^0 + \int_0^{t_f} \hat{n}_{\text{in}}(\tau) d\tau} \tag{2.13}$$

$$\text{Instantaneous conversion:} \qquad X_{inst} = \frac{{n_i}^0 + \int_0^t \hat{n}_{\text{in}}(\tau) d\tau - n_i(t)}{\int_0^t \hat{n}_{\text{in}}(\tau) d\tau} \tag{2.14}$$

Here, ${n_i}^0$ is the amount of reactant present at time zero, $n_i(t)$ is the amount of reactant present in the system at time t, and $\hat{n}_{\text{in}}$ is the flow of the reactant into the system. The conversion is often used to determine when to terminate a batch and remove the product(s).

2.3 Polymerization Modeling

It is difficult to formulate deterministic models of polymerization because of the many reactions and stochastic features of growing polymer chains [21]. Polymerization models are typically simplified by using moment balances and average molecular weights; this approach is outlined in the sections below. How to include the diffusion-limitations in the reaction equations is also presented.

2.3.1 Method of Moments

The final product will consist of polymer chains of varying lengths. The most accurate method of describing the product quality is to use either the molecular weight distribution (MWD) or the chain length distribution (CLD). Controlling either of these distributions is an effective way of regulating the end product since the final properties of the polymer are highly dependent on the distributions.

Using a population balance approach would result in a complicated model so it is common practice to capture the important characteristics of the MWD or CLD by using the method of moments, which is a statistical approach [21]. For simplicity, only the MWD is used here to illustrate the concept of moment balances. A generic k^{th} order moment is given by:

$$\mu_k = \sum_{\mathrm{n}=1}^{\infty} \mathrm{n}^k \cdot n_{\mathrm{P,n}} \tag{2.15}$$

where $n_{\mathrm{P,n}}$ is the number of moles of polymer chains with length n.

Each of the three first moments has a physical representation associated with it. The zeroth order moment is the mean, which gives the concentration of polymer chains in the reactor. The first order moment is the standard deviation and represents the concentration of monomeric units that have been incorporated into the polymer chains. The zeroth and first moments can be related to the number-fraction and weight-fraction molecular weight distribution of the polymer chains as well [22]. The second order moment is the skewness, which indicates the width of the distribution, meaning that a high value of the second order moment is related to a heterogeneous mixture of chain lengths [21].

In free-radical polymerization, the different moment balances are applied to both living radical chains and terminated dead chains. A live chain can continue further chain growth (i.e. the chain still has a radical), whereas a dead chain is a polymer chain that will no longer grow (i.e. the chain has been terminated). Note that μ_k is used as the notation for dead chains and λ_k as the notation for living chains. Both of these moments are tracked throughout the process using moment balances.

2.3.2 Average Molecular Weights

Due to the kinetics of polymerization, all synthetic polymers and most natural polymers have a distribution of molecular weights [32]. Given that the molecular weight plays a key role in the physical and mechanical properties of the polymers, it is important to characterize these distributions using a variety of average quantities. The most commonly used averages are the *number average molecular weight* and the *weight average molecular weight*. These averages are utilized since they can be measured directly. Because these values give insight into the polymer's properties,

one or both the values are often used as a controlled variable in a polymerization process; therefore, it is important to understand what these two values represent.

Moment balances can be used to calculate both the number average molecular weight and the weight average molecular weight. The number average molecular weight is the ordinary arithmetic mean where the total polymer mass is divided by the number of chains. Using moments, it is calculated as:

$$\overline{M}_n = \frac{\lambda_1 + \mu_1}{\lambda_0 + \mu_0} M_{w,M} \tag{2.16}$$

where $\overline{M}_n$ is the number average molecular weight, λ_k is the k^{th} order moment of the living chains, μ_k is the k^{th} order moment of the dead chains, and $M_{w,M}$ is the molecular weight of the monomer. The fraction gives the number average chain length [21].

The weight average molecular weight calculation accounts for the fact that long chains contain a larger portion of the total polymer mass. The weight average molecular weight is calculated as the fraction between the sum of second order moments and the sum of first order moments multiplied by the monomer's molar mass. If a random chained monomer molecule is selected, the weight will on average be equal to the weight average molecular weight. This is calculated using:

$$\overline{M}_w = \frac{\lambda_2 + \mu_2}{\lambda_1 + \mu_1} M_{w,M} \tag{2.17}$$

The ratio between the weight average molecular weight and the number average molecular weight is used to classify the polymer quality. This ratio is known as the polydispersity index (PI) and is defined as [20]:

$$\text{PI} = \frac{\overline{M}_w}{\overline{M}_n} = \frac{\lambda_2 + \mu_2}{\lambda_0 + \mu_0} \tag{2.18}$$

This ratio gives a direct relationship to the standard deviation of the average weights and therefore can be used as a measure of the MWD [32]. A larger PI implies that the polymer product consists of chains of unequal lengths, whereas a PI equal to one means that all the polymer chains are equally long.

2.3.3 Free-Volume Theory

The diffusional-limitations on the initiation, propagation, and termination reactions can be modeled in many different ways but the most comprehensive methods are based on *free-volume theory*. Free volume theory relates the diffusion-limitations of polymer chains, initiator radicals and monomer molecules to the free volume surrounding these components [21].

The volume of a liquid has two parts: the volume occupied by the molecules and the empty space between the molecules. The empty space is known as the free volume

where only the portion that is continuously redistributed by thermal fluctuations is available for molecular transport. This part of the free volume is known as the *hole free volume* and the remaining space is known as the *interstitial free volume*. Diffusion and, consequently, all transport processes are controlled by the hole free volume [40]. As the polymer chains grow, they start to overlap and the polymer phase volume decreases; thus, the molecules have less space and the polymerization reactions become diffusion-controlled. The free volume available for diffusion can be increased through the addition of low viscosity additives.

Chapter 3
Optimization and Control

Automatic control systems have existed in a simple form as far back as two thousand years ago but the form commonly utilized in industry today was not seen until the 1950's and 1960's [35]. Control theory is the foundation on which these controllers are constructed and involves the control of continuously or semi-continuously operating dynamic systems in engineering processes [36]. Section 3.1 presents a brief introduction to these ideas by giving the definitions of key terms. Controllers work to regulate a system to produce the desired behavior. These ideas can be extended to determine the inputs for a given system to ensure that certain optimality conditions are met utilizing concepts from the field known as optimization.

Section 3.2 introduces the foundation of optimization. These ideas are used in industrial applications to determine the "least-worst" operating conditions that give the desired minimum. Physical and economical constraints are easily included in the optimization problem. First the components of an optimization problem are outlined with some properties of the solution. The general method of locating a solution is then discussed. The exact approach is dependent on the problem type; there are three main classifications of problem types which are introduced in this section. Solution methods for nonlinear programming problems specifically are covered since polymerization processes are typically nonlinear.

The two fields of optimization and control can be combined to create a control structure for a process as outlined in Section 3.3. This control structure can be divided into different layers based on the time scale in which they operate; together these layers make up the control hierarchy. This work focuses on the two top levels of this control hierarchy, which are known as the advanced process control layer and the optimization or supervisory layer. The advanced process layer uses a NMPC algorithm for processes whose nonlinearities cannot be neglected. The supervisory layer is an upper layer that has a different objective than the lower layers. For processes with large nonlinearities, it is common to utilize a DRTO formulation in the supervisory layer, which uses a dynamic model. These two levels work together to realize the desired outcome of the process. The solution from the supervisory layer becomes the input to the advance control layer below it.

3.1 Control

The use of control has become increasingly important in industrial applications due to increased competition, rapidly changing economic conditions, and stricter environmental and safety regulations. Together with process modeling, the use of a control structure is critical in operating manufacturing processes near the optimal conditions [31].

All control problems have three variable types: controlled variable (CV), manipulated variable (MV), and disturbance variable (DV). *Controlled variables* are the process variables that are being controlled (temperature, pressure, concentration, etc.); the desired value of a controlled variable is called a *setpoint*. *Manipulated variables* are process variables that can be adjusted to keep the controlled variables at or near their setpoints; commonly used manipulated variables are valve positions. They are often variables that can be easily measured and are paired based on how much effect they have on the CVs. *Disturbance variables* are process variables that affect the controlled variables but cannot be manipulated; typically, disturbances are things such as feed composition or the ambient temperature, which can be measured but not controlled. Selecting all of these variables correctly is a crucial step in the construction of a good control system. These variables are chosen from the process model variables and are selected based on process knowledge, experience, and the control objectives [31].

It is common to format the control structure in state space form, which is composed of four main variable types: states, inputs, outputs, and measurements. *States* $\boldsymbol{x}$ are the variables that describe the current physical state of the system. The states are typically the variables in the balance equations of the model. *Inputs* $\boldsymbol{u}$ describe the control inputs, which influence the controlled variables. The *outputs* $\boldsymbol{z}$ describe the calculated/estimated process output and often include the controlled variables. *Measurements* $\boldsymbol{y}$ are the actual measured process outputs, which are system parameters that can be physically measured such as concentration, temperature, pressure, etc.

3.2 Optimization

Problems that have constraints are known as constrained optimization problems and arise when models contain constraints. These constraints typically include physical limitations such as valve positions or safety limits. Constrained optimization problems have three main components: an objective function, decision variable(s) and constraint(s). The *objective function* f is a scalar function that describes a property to be minimized or maximized. The *decision variables* r can be real numbers, integers or even belong to other spaces such as function spaces; typically decision variables are limited to the Euclidean space, i.e, vectors of real variables.

The *constraints* $\boldsymbol{c}_j$ are a vector of scalar functions of the decision variables that define certain equations that the unknown variables must satisfy [23]. Constraints are broken up into two different categories: equality $\mathcal{E}$ and inequality $\mathcal{I}$ constraints [7].

For constrained problems, the search region is more limited than for unconstrained problems; the constraints mean that the solution is selected from a subset of $\mathbb{R}^{n_r}$. This subset is called the *feasible region*, which is the set of points satisfying all the constraints $\boldsymbol{c}_j$ [23]. The feasible region is formally defined as:

Definition 3.1 (Feasible Region)

$$\mathcal{F} = \{\boldsymbol{r} \in \mathbb{R}^{n_r} \mid (\boldsymbol{c}_j(\boldsymbol{r}) = 0,\ j \in \mathcal{E}) \wedge (\boldsymbol{c}_j(\boldsymbol{r}) \geq 0,\ j \in \mathcal{I})\} \tag{3.1}$$

Figure 3.1 illustrates the concept of a feasible region for a given objective function and constraints. The contours of the objective function are shown in the dashed lines (i.e., the set of points for which $f(\boldsymbol{r})$ has a constant value) along with the solution point $\boldsymbol{r}^*$; the two constraints are drawn with solid lines. The gray shaded area is the "infeasible side" of the inequality constraints so a solution will never be found here.

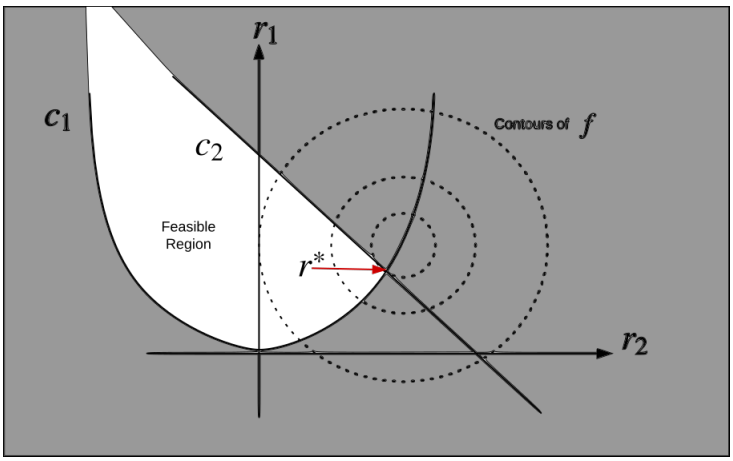


Figure 3.1: Geometrical representation of an example optimization problem [23].

When a potential solution value is inserted into the constraint, they are divided into active or inactive constraints. An *active constraint* is where $\boldsymbol{c}_j(\boldsymbol{r}) = 0$; therefore, all equality constraints by definition are active constraints. In comparison, an inequality constraint can be either active or inactive at a feasible point. When a constraint is active, the region where a solution may exist is smaller than when constraints are inactive.

For a multivariable control problem it is not always possible for all constraints to be fulfilled at all times. Therefore, the algorithm must know how to handle the constraints so that the problem does not become infeasible [28]. Constraints can be further categorized into hard, soft, or setpoint approximation. A *hard constraint* means that the constraint should never be violated in the future; constraints of this type are ranked in order of priority. *Soft constraints*, on the other hand, may be violated in the future but this violation is penalized in the objective function. *Setpoint approximation* of a constraint means that deviations above and below the constraint are penalized. Figure 3.2 shows these three constraint types where the shaded areas represent violations penalized in the dynamic optimization.

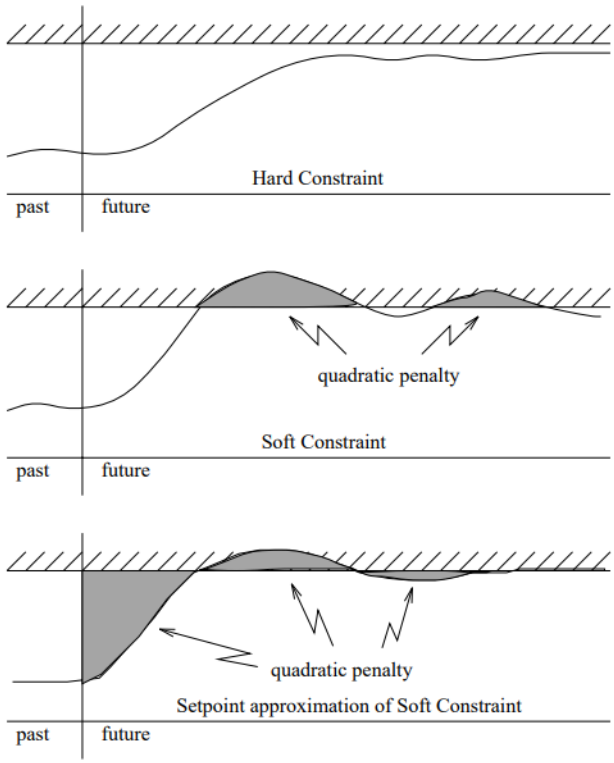


Figure 3.2: The three basic types of constraints [28].

3.2.1 Local versus Global Minimizers

The solution to an optimization problem is known as a *minimizer*; this is the point that gives the objective function the smallest value while also satisfying any constraints. It is ideal if a *global minimizer* is identified; this is a point where the function attains its absolute least value [23]. A global minimizer is defined as:

Definition 3.2 (Global Minimizer) *$\boldsymbol{r}^* \in \mathcal{F}$ is the solution to an optimization problem if $f(\boldsymbol{r}^*) \leq f(\boldsymbol{r})$, for all $\boldsymbol{r} \in \mathcal{F}$, where $\mathcal{F}$ is the feasible region.*

If there is only one global minimizer, then the global minimizer is a *strict global minimizer*. However, it is often difficult to find a global minimizer because the overall objective function shape is typically not well known; therefore, it is hard to determine that the algorithm used to solve the problem has not missed some region that may contain a minimizer. Therefore it is more common to say that the identified solution is a *local minimizer*, which is a point $\boldsymbol{r}^*$ that provides a minimum value in a neighborhood instead of the minimum in the whole feasible region. The formal definition is given by:

Definition 3.3 (Local Minimizer) *$f(\boldsymbol{r}^*) \leq f(\boldsymbol{r})$, for all $\boldsymbol{r} \in \|\boldsymbol{r} - \boldsymbol{r}^*\| < \boldsymbol{\varepsilon}$ where $\boldsymbol{r}$ can take values in a feasible open set about $\boldsymbol{r}^*$.*

A local minimizer may not even be the absolute least value in a neighborhood. but one that is the absolute least value in a given neighborhood is known as a *strict local minimizer*; this is true if $f(\boldsymbol{r}^*) < f(\boldsymbol{r})$, for all $\boldsymbol{r} \in \|\boldsymbol{r} - \boldsymbol{r}^*\| < \boldsymbol{\varepsilon}$. Examples of local and global minimizers are illustrated in Figure 3.3.

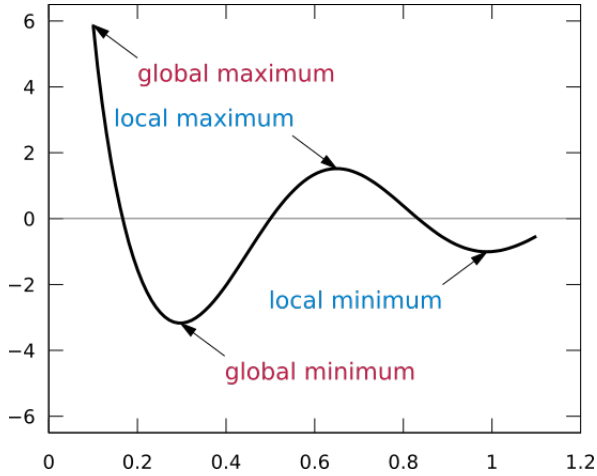


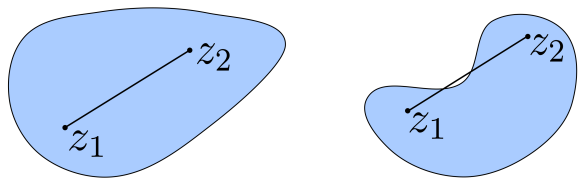
Figure 3.3: Global versus local extrema[1]

[1] http://commons.wikimedia.org/wiki/File:Extrema_example.svg

For functions that have numerous peaks and valleys, it can be challenging for algorithms to find the global minimums because the algorithm can get "stuck" in a valley at a local minimum. Most solution methods therefore do not find global minimizers but rather local ones. In the case that additional global information about the objective function is available, the global minima are more likely to be identified; one special case is when the objective function is convex [23].

3.2.2 Convex and Nonconvex Problems

Convex and nonconvex problems are two important subclasses of optimization problems. Illustrations of a convex and a nonconvex set are shown in Figure 3.4.



a) A convex set. b) A nonconvex set.

Figure 3.4: Comparison of a convex and a nonconvex set.

In a convex set it is possible to draw a line between any two points in the set and all points on that line will also be included in the set; this is not the case for a nonconvex set.

Convex problems have the following useful property that is exploited by many solvers making them easier to solve:

> *A local minimizer of a convex optimization problem is also a global minimizer.*

An optimization problem is convex if it satisfies the following two conditions:

- *The objective function is a convex function (Definition A.4)*
- *The feasible set $\mathcal{F}$ is a convex set (Definition A.3)*

If a problem is *strictly convex*, there will exist only one global solution. Nonconvex problems may have multiple feasible regions and locally optimal points within each region; consequently, it is much more difficult to identify if a feasible solution exists.

3.2.3 Identifying Minimizers

For a smooth objective function there are efficient ways of identifying local minima; in particular, if a function is twice continuously differentiable, it can be determined if a point is a local minimizer by examining the gradient $\nabla f(\boldsymbol{r}^*)$ and the Hessian $\nabla^2 f(\boldsymbol{r}^*)$ [23]. This exploits the well known Taylor's Theorem, which is defined in Definition A.6. From this theorem, *necessary conditions* and *sufficient conditions* can be derived by assuming that $\boldsymbol{r}$ is a local minimizer and proving facts about the gradient and the Hessian [23]. Necessary conditions must be met for the point to

potentially be a local minimizer but does not guarantee that the point is a local minimizer. For a constrained optimization problem, the necessary conditions are referred to as the first-order necessary conditions or Karush-Kuhn-Tucker conditions (KKT) conditions. To guarantee that a point is a local minimizer, the sufficient conditions must be met; these are known as second-order sufficient conditions.

3.2.3.1 First-Order Conditions

The first-order conditions explain how the first derivatives of the objective function and the active constraints are related to each other at a solution; if these conditions are satisfied, then a move in any direction along the function will result in either an increase in the first-order approximation of the objective function or will stay the same [23]. Minimizers may only be found within a certain region that is known as the *active set*; the full definition is given in Definition A.8. This helps to limit the number of potential solutions.

An important property used in the definition of the first-order conditions is linear independence constraint qualification (LICQ), which states that the gradients are linearly independent at the examined point; the formal definition for LICQ can be found in Definition A.9. This property must be true in order for the first order necessary conditions defined below to hold. When these KKT conditions are satisfied, the optimal Lagrange multiplier $\boldsymbol{\lambda}_j^*$ is unique for the solution point $\boldsymbol{r}^*$.

Theorem 3.1 (First order necessary conditions) *Assume that $\boldsymbol{r}^*$ is a local solution of Problem 3.9. Assume as well that $f(\boldsymbol{r})$ and all $\boldsymbol{c}_j$ are differentiable and their derivatives are continuous. Further, assume that all the active constraint gradients are linearly independent at $\boldsymbol{r}^*$ (meaning that LICQ holds). Then there exists Lagrange multipliers $\boldsymbol{\lambda}^*$ for $j \in \mathcal{E} \cup \mathcal{I}$ such that the following conditions (known as the KKT conditions) hold at $(\boldsymbol{r}^*, \boldsymbol{\lambda}^*)$ [7]:*

$$\nabla_{\boldsymbol{r}} \mathcal{L}(\boldsymbol{r}^*, \boldsymbol{\lambda}^*) = 0 \tag{3.2}$$

$$\boldsymbol{c}_j(\boldsymbol{r}^*) = 0, \qquad j \in \mathcal{E} \tag{3.3}$$

$$\boldsymbol{c}_j(\boldsymbol{r}^*) \geq 0, \qquad j \in \mathcal{I} \tag{3.4}$$

$$\boldsymbol{\lambda}_j^* \geq 0, \qquad j \in \mathcal{I} \tag{3.5}$$

$$\boldsymbol{\lambda}_j^* \boldsymbol{c}_j(\boldsymbol{r}^*) = 0, \qquad j \in \mathcal{I} \tag{3.6}$$

The KKT conditions are only necessary conditions for a local solution; therefore, any minimizer that satisfies these conditions may be a local minimum but it is not guaranteed. To prove that the point is a local minimizer, sufficient conditions must be used; these are derived through second order optimality conditions [7].

3.2.3.2 Second-Order Conditions

The second derivatives also play a role in the optimality conditions and serve in a "tiebreaking" capacity. Second-order conditions help to determine if a move results in an increase or decrease in the objective function. Note that since second derivatives are being used, a stronger smoothness assumption is required here than previously; therefore, it is assumed that the objective function and constraints are all twice continuously differentiable [23]. The second derivatives are used to develop sufficient conditions, which are conditions on the objective function and the constraints, that ensure that a point is a local solution of the problem [23].

The solutions that satisfy the second order conditions are limited by the *critical cone*, which contains all the directions that result in an adherence to the active inequality constraints [23]; the formal definition is presented in Definition A.2. This helps to narrow down the number of possible moves that can be made in the search for the optimal point.

Theorem 3.2 (Second order sufficient conditions) *Suppose that for some feasible point $\boldsymbol{r}^* \in \mathbb{R}$ there exists Lagrange multipliers $\boldsymbol{\lambda}^*$ such that the KKT conditions (Equations 3.2-3.6) are satisfied, and that $f(\boldsymbol{r})$ and all $\boldsymbol{c}_j$ are twice differentiable and their derivatives are continuous. Assume also that:*

$$\nabla_{\boldsymbol{rr}} \mathcal{L}(\boldsymbol{r}^*, \boldsymbol{\lambda}^*) > 0 \tag{3.7}$$

Then $\boldsymbol{r}^*$ is a strict local solution of Problem 3.9.

Note that $\nabla_{\boldsymbol{rr}} \mathcal{L}(\boldsymbol{r}^*, \boldsymbol{\lambda}^*)$ needs to only be positive in the directions defined by the critical cone.

3.2.4 Problem Types

Constrained optimization problems are typically divided into three categories based on the form of the objective function and the constraints; they are either linear programming (LP), quadratic programming (QP), or nonlinear programming (NLP) problems. The definition of a LP is provided in Definition A.10, with the other two defined in the sections below. QP and NLP problems are emphasized in this work because these problem types are utilized later in this work. Polymerization processes are nonlinear so a NLP problem is the appropriate optimization problem type for these systems. Some algorithms convert NLP problems into QP problems to make them easier to solve so it is necessary to also be familiar with this problem format.

3.2.4.1 Quadratic Program (QP)

A QP problem has a quadratic objective function $f(\boldsymbol{r})$ and linear constraints $\boldsymbol{c}_j$. This can be generically written in the form:

$$\begin{aligned} \min_{\boldsymbol{r} \in \mathbb{R}^{n_r}} \quad & \frac{1}{2}\boldsymbol{r}^T \boldsymbol{P} \boldsymbol{r} + \boldsymbol{q}^T \boldsymbol{r} \\ \text{s.t.} \quad & \boldsymbol{c}_j(\boldsymbol{r}) = \boldsymbol{a}_j{}^T \boldsymbol{r} - \boldsymbol{b}_j = 0, \qquad j \in \mathcal{E}, \\ & \boldsymbol{c}_j(\boldsymbol{r}) = \boldsymbol{a}_j{}^T \boldsymbol{r} - \boldsymbol{b}_j \geq 0, \qquad j \in \mathcal{I} \end{aligned} \tag{3.8}$$

Since all the constraints are linear, the feasible set is convex; however, this does not imply that the problem itself is convex. QP problems may or may not be convex depending on the quadratic component of the objective function ($\boldsymbol{r}^T \boldsymbol{P} \boldsymbol{r}$) and the constraints. The quadratic weight matrix $\boldsymbol{P}$ is an $n_{\boldsymbol{r}}$-dimensional symmetric matrix and is known as the Hessian matrix. If $\boldsymbol{P}$ is positive semidefinite ($\boldsymbol{P} \geq 0$), then Problem 3.8 is a convex problem. For a convex QP problem, the KKT conditions serve as necessary and sufficient conditions, meaning that in this case, the KKT conditions give a global solution.

3.2.4.2 Nonlinear program (NLP)

A NLP problem has a linear or quadratic objective function and nonlinear constraints. A general NLP can be written as:

$$\begin{aligned} \min_{\boldsymbol{r} \in \mathbb{R}^{n_r}} \quad & f(\boldsymbol{r}) \\ \text{s.t.} \quad & \boldsymbol{c}_j(\boldsymbol{r}) = 0, \qquad j \in \mathcal{E}, \\ & \boldsymbol{c}_j(\boldsymbol{r}) \geq 0, \qquad j \in \mathcal{I} \end{aligned} \tag{3.9}$$

where $f(\boldsymbol{r})$ is an n-dimensional vector and the decision variables $\boldsymbol{r}$ are defined in the Euclidean space; $\mathcal{E}$ and $\mathcal{I}$ are disjunct index sets such that $\mathcal{E} \cap \mathcal{I} = \emptyset$ for the constraints.

3.2.5 Solution Methods

Optimization algorithms are iterative and typically begin with an initial guess of the solution. The algorithm then generates a sequence of improved solution estimates (known as *iterates*) until the algorithm terminates. The method that is used to move from one iterate to another is what distinguishes the algorithms from one another; many strategies use the objective function values, the constraint functions, and the first and second derivatives of the functions [23].

NLPs problems can be solved using a variety of methods and which one is selected depends on the properties of the problem. The main algorithms for solving a constrained NLP problem are: interior-point, sequential quadratic programming (SQP), active-set, and trust-region methods. Here attention is only given to the most commonly utilized method known as SQP.

3.2.5.1 Sequential Quadratic Programming (SQP)

One of the most effective method families for solving nonlinear, constrained optimization problems is through the generation of steps by solving quadratic sub-problems, resulting in the method known as SQP [23]. Note that this approach only works for problems that have a twice continuously differentiable objective function and constraints.

These methods solve general nonlinear problems while honoring the bounds at all iterations. Their strength lies in solving problems with significant nonlinearities in the constraints, which is the case for many polymerization processes. Many optimization algorithms require a feasible starting point to initiate the algorithm, the SQP method does not; instead it penalizes constraint violations as a part of the objective function. This is an advantage of this method because selecting an initial guess can often be difficult.

The essence of this method is that a sequence of optimization sub-problems is solved where a quadratic model of the objective function is used subject to a linearization of the constraints [37]. Each sub-problem is solved with the use of either an *active set method* or an *interior-point method* [21]. The sub-problems are formulated in a manner that ensures convergence to a local minimum of the NLP as the number of iterates reaches infinity [23].

For a nonlinear programming problem of the form shown in Equation 3.9, the Lagrangian will be given by:

$$\mathcal{L}(\boldsymbol{r}, \boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2) = f(\boldsymbol{r}) - \boldsymbol{\lambda}_1^T \boldsymbol{c}_{\mathcal{E}}(\boldsymbol{r}) - \boldsymbol{\lambda}_2^T \boldsymbol{c}_{\mathcal{I}}(\boldsymbol{r}) \tag{3.10}$$

where $\boldsymbol{\lambda}_1$ is a vector of the Lagrange multipliers for the equality constraints and $\boldsymbol{\lambda}_2$ is a vector of the Lagrange multipliers for the inequality constraints. At an iterate $\boldsymbol{r}_k$, the algorithm defines an appropriate search direction $\boldsymbol{p}_k$ as the solution to the QP sub-problem given by:

$$\min_{\boldsymbol{p}} \quad f(\boldsymbol{r}) + \nabla f(\boldsymbol{r})^T \boldsymbol{p} + \frac{1}{2} \boldsymbol{p}^T \nabla^2 \mathcal{L}(\boldsymbol{r}_k, \boldsymbol{\lambda}_{1,k}, \boldsymbol{\lambda}_{2,k}) \boldsymbol{p} \tag{3.11a}$$

$$\text{s.t.} \quad \boldsymbol{c}_{\mathcal{E}}(\boldsymbol{r}) + \nabla \boldsymbol{c}_{\mathcal{E}}(\boldsymbol{r})^T \boldsymbol{p} = 0, \tag{3.11b}$$

$$\boldsymbol{c}_{\mathcal{I}}(\boldsymbol{r}) + \nabla \boldsymbol{c}_{\mathcal{I}}(\boldsymbol{r})^T \boldsymbol{p} \geq 0 \tag{3.11c}$$

This subproblem is then solved multiple times until a solution is identified. A full explanation and derivation of the SQP method as well as more information on other solution methods can be found in Nocedal and Wright [23].

3.3 Combining Optimization and Control

Advanced control layers that combine optimization and control are becoming more common in industrial applications as costs and regulations increase, resulting in

the need to operate the plants at more optimum conditions. Figure 3.5 shows an example of how optimization and control can be utilized together in a control structure to operate a process in an optimal manner.

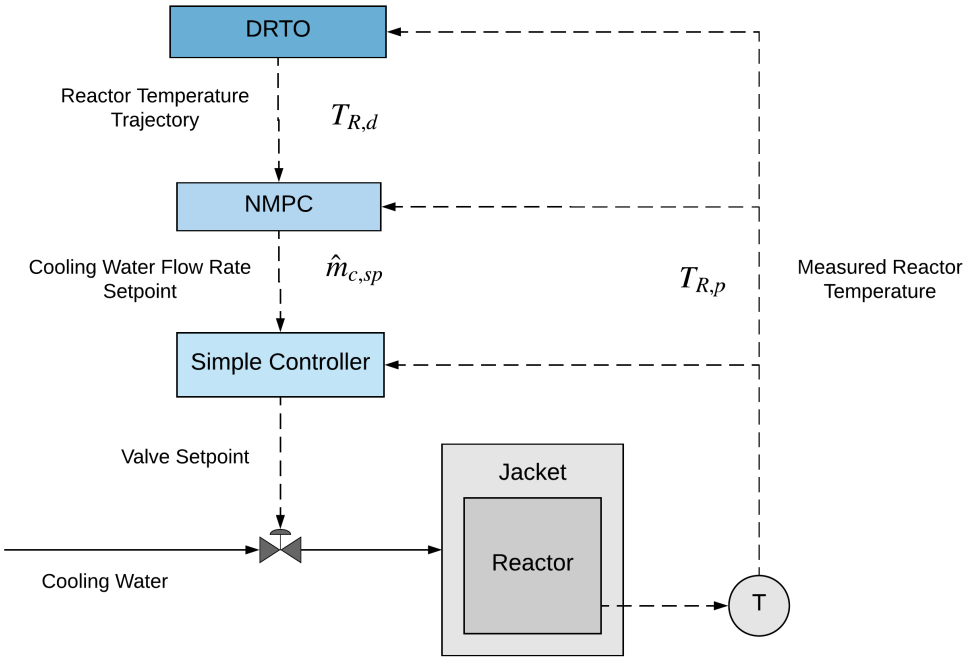


Figure 3.5: Illustration of how optimization and control are combined in a process.

Here the DRTO layer calculates the optimal reactor temperature trajectory. This is sent to the NMPC layer, which then determines the optimal cooling water flow rate needed to track the trajectory from the upper level. The cooling water flow rate is passed to the simple controller, which is typically some version of a P, PI, or PID controller. This controller then realizes this flow rate by adjusting the valve position. All of these layers receive the reactor temperature measurement from the process since they operate on different time scales; therefore, it is important that they each know the latest state of the reactor. Each layer makes control decisions at its own frequency with the simple controller operating on the shortest time scale; note that the inputs from the above layer always trump what the layer itself decides. These ideas are further explained in Section 3.3.1.

Attention is given to the two top layers of the control hierarchy, since that is the focus of this thesis. The advanced process control (APC) level is outlined in Section 3.3.2 emphasizing the algorithm for nonlinear models known as NMPC. The upper level known as supervisory control is then discussed in Section 3.3.3 with emphasis on the DRTO method. Finally how to estimate to incorporate parameter and state

estimation into the control structure is discussed.

3.3.1 Control Hierarchy

In industry, a plant's control structure is typically composed of several different layers that together make up a *control hierarchy* such as the one illustrated in Figure 3.6. The layers are separated based on the time scale in which it operates. The top layer typically operates on a time scale of days to months, whereas the bottom layer operates on a time scale of milliseconds to seconds. Each layer has a specific goal it wishes to accomplish such as following safety requirements or maximizing the process's profitability.

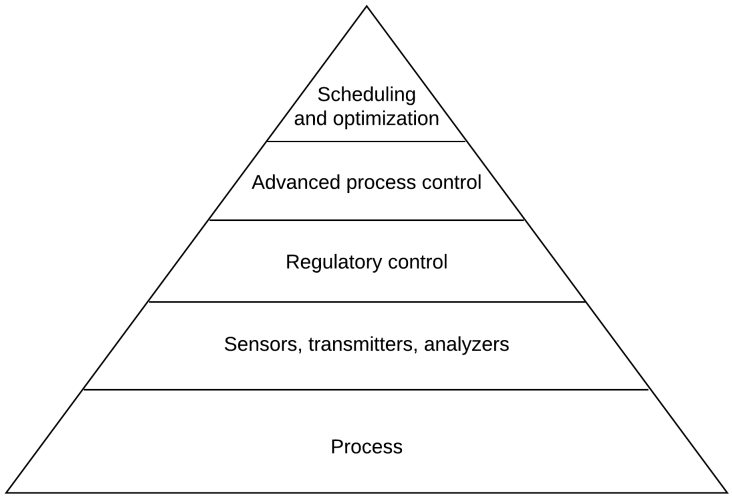


Figure 3.6: Common control hierarchy [7]

The bottom three layers are necessary for all process plants and the top two levels are optional but whose addition helps to increase the profitability. The decision to implement the top two levels is based on the particular company's financial priorities [2]. The lower layers are executed at a higher frequency than the upper layers; this is because the computational demand increases substantially from the bottom layer to top layer.

At the bottom are the physical components of the *process* itself such as pipes, reactors, heaters, distillation columns, etc. The process values (e.g. temperature, concentration, pressure) are measured through the use of *sensors, transmitters, and analyzers*; the data is passed in real-time to controllers in the *regulatory control layer* using communication networks. The regulatory layer consists of traditional controllers such as proportional, proportional-integral, or proportional-

integral-derivative controllers, which are used to control things on a fast time scale (millisecond-second); these controllers are embedded in a distributed control system (DCS). Typically variables that should be tightly controlled, meaning the deviation from the desired value should be very small, are controlled in this layer. The setpoints for the controlled variables in this level are typically supplied by the level above. The APC layer controls the process through the regulatory layer and typically communicates with the DCS through the use of open platform communication (OPC). However, this alone is not sufficient to meet industry requirements for reliability and availability so safety critical controllers are also placed in regulatory layer; this is particularly required for unstable processes such as an exothermic reactor [7].

APC is an umbrella term that covers a broad range of techniques and control methodologies [2]. The basic purpose of this layer is manage the interactions within the process such that process variability is reduced and the plant can run as close to the operating constraints as possible. The APC works to help reduce the variation in process variables in the industrial application; in turn, this leads to an increased throughput and higher profit. A dynamic model of the process is utilized along with a history of past control actions, and an optimization cost function over the receding prediction horizon to calculate the optimum control need [2].

The top layer contains the scheduling and optimization, which includes production plans of different products or as an output for an optimization application [7]. While the optimum operating conditions for a plant are determined as part of the initial process design, changes in equipment availability, economic conditions and process disturbances can mean that the optimum operating conditions need to be recalculated. The solution to this optimiziation problem then becomes the input or the new setpoints to the APC layer below. On top of this, further plantwide optimization could be performed that optimially distributes resources between different systems in the plant.

3.3.2 Advanced Control Layer

Advanced control layers typically utilize model predictive control (MPC), which for a nonlinear process is known as NMPC. This is a form of control where the current control action is obtained by solving at each sampling instant a finite horizon open loop optimal control problem using the current state of the plant as the initial state; the optimization produces the optimal control sequence from which the first value in this sequence is taken and applied to the plant [7]. This type of control predicts what the process behavior will be and updates this prediction using the feedback from the process.

NMPC couples open loop optimization with feedback control since at each time step a new solution of the dynamic optimization problem is required; in other words, NMPC essentially solves a similar optimization problem over and over at each

time step. Hence, a *moving horizon* approach is used where the prediction horizon changes from $k,\ldots,k+N$ to $k+1,\ldots,k+N+1$ from one time step to the next as illustrated in Figure 3.7 [23].

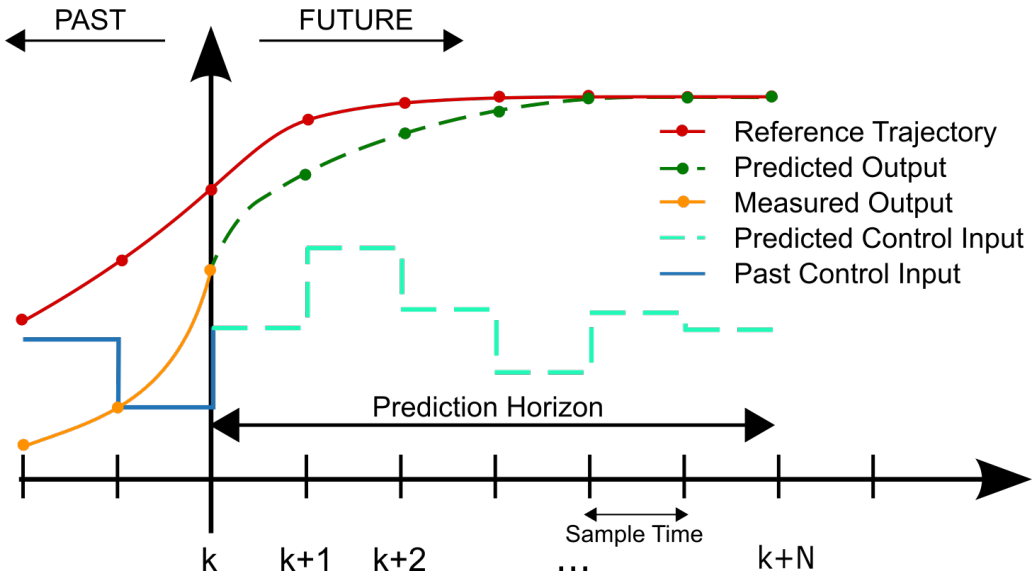


Figure 3.7: Simple illustration of the MPC concept. [1]

[1] By Martin Behrendt [CC BY-SA 3.0 (https://creativecommons.org/licenses/by-sa/3.0)

NMPC brings a structure approach to solutions where the main aim is to minimize a performance criterion in the future, which is subject to constraints on the manipulated inputs and outputs; the future behavior is computed according to a model of the plant [2].

The first generation of controllers that used MPC algorithms were developed in the 1970's [17]. MPC has been successful in industry due to its increased performance and easy to understand concept. Modern processing plants incorporate these model predictive control algorithms as a part of a multi-level control hierarchy. The usage of these algorithms has increased rapidly with over 4,500 applications reported worldwide in a multitude of areas such as chemicals, food processing, automotive and aerospace applications [28]. However, application is not seen as often in the polymer industry where only 17 applications report using it [28]. NMPC is seen somewhat more frequently in the polymer industry with 21 applications being reported [28]. The success of these algorithms are highly dependent on the accuracy of the process model in predicting the dynamic behavior of the process. The use of MPC and NMPC can have the following advantages [17]:

a) The process model captures the dynamic and static interactions between the variables.

b) The constraints are considered systematically.

c) The optimal setpoints are calculated together with the control calculations.

d) The predictions can provide early warnings of potential problems.

3.3.2.1 NMPC Problem Formulation

A NMPC problem following a reference trajectory can be formulated as:

$$
\begin{aligned}
\min_{\boldsymbol{r}\in\mathbb{R}^{n_r}} \quad & f(\boldsymbol{r}) = \frac{1}{2}(\boldsymbol{x}_{k+1}-\boldsymbol{x}_{k+1}^{ref})\boldsymbol{Q}_{k+1}(\boldsymbol{x}_{k+1}-\boldsymbol{x}_{k+1}^{ref}) + \boldsymbol{d}_{\boldsymbol{x}_{k+1}}(\boldsymbol{x}_{k+1}-\boldsymbol{x}_{k+1}^{ref}) + \frac{1}{2}\boldsymbol{u}_k{}^T\boldsymbol{R}_k\boldsymbol{u}_k \\
& + \boldsymbol{d}_{\boldsymbol{u}_k}\boldsymbol{u}_k + \frac{1}{2}\Delta\boldsymbol{u}_k{}^T\boldsymbol{S}_k\Delta\boldsymbol{u}_k + \boldsymbol{d}_{\Delta\boldsymbol{u}_k}\Delta\boldsymbol{u}_k + \boldsymbol{r}_1{}^T\boldsymbol{\epsilon} + \frac{1}{2}\boldsymbol{\epsilon}^T\operatorname{diag}(\boldsymbol{r}_2)\boldsymbol{\epsilon} \\
\text{s.t.} \quad & \boldsymbol{x}_{k+1} = \boldsymbol{f}(\boldsymbol{x}_k,\boldsymbol{u}_k) && k=0,\dots,N-1, \\
& \boldsymbol{z}_k = \boldsymbol{h}(\boldsymbol{x}_k,\boldsymbol{u}_k) && k=0,\dots,N-1, \\
& \boldsymbol{x}_0,\boldsymbol{u}_{-1} = \text{given}, \\
& x^{low}-\boldsymbol{\epsilon} \le \boldsymbol{x}_k \le x^{high}+\boldsymbol{\epsilon} && k=1,\dots,N, \\
& u^{low} \le \boldsymbol{u}_k \le u^{high} && k=0,\dots,N-1, \\
& \Delta u^{low} \le \Delta\boldsymbol{u}_k \le \Delta u^{high} && k=0,\dots,N-1, \\
& \boldsymbol{Q}_k \ge 0 && k=1,\dots,N, \\
& \boldsymbol{R}_k \ge 0 && k=0,\dots,N-1, \\
& \boldsymbol{S}_k \ge 0 && k=0,\dots,N-1
\end{aligned}
\tag{3.12}
$$

Since a feasible point may not always exist, and therefore a control input would be unavailable, a solution is found through the use of *slack variables*. This softens the state constraints and alters the objective function.

The objective function penalizes state deviations from the desired reference $\boldsymbol{x}_{k+1}^{ref}$, the usage of inputs, the input change rate $\Delta\boldsymbol{u}_k$ and the slack variables $\boldsymbol{\epsilon}$, which represent deviation from hard state constraints. It also contains quadratic and linear weights for each of the penalized variables. The value of the linear weighted term increases instantly when the variable moves away from the reference value. The quadratic weights give a low value gradient in the interval around the reference value and therefore, provide a smaller penalty for a small deviation from the reference. If there is a large deviation from the optimality, the quadratic weights penalize this more than the linear weights [21].

The quadratic state weighting matrix $\boldsymbol{Q}_{k+1}$ and the linear state weighting vector $\boldsymbol{d}_{\boldsymbol{x}_{k+1}}$ allows individual states to be given different importance in the solution. For example, it may be more important that some states converge more quickly to the reference value than others. The state weighting also provides the ability to weight the importance of state convergence against the importance of other variables in the objective function. For easy to measure and access inputs, the input weights

would be small compared to the state weights for instance. The quadratic state weights $\boldsymbol{R}_k$ and linear weights $\boldsymbol{d}_{\boldsymbol{u}_k}$ for inputs could also be used for tuning of the input priority.

The rate of change in the input is penalized to minimize the wear and tear on equipment. Larger weights on these variables contributes to less aggressive control moves. The slack variables are added to the state constraints to ensure that the solution of the optimization problem is feasible at all times [23]. Larger weights $\boldsymbol{\rho}$ and $\boldsymbol{S}$ on the slack variables means a larger penalization of state bound violations.

The constraints on the states, inputs, and input rate change serve as lower and upper limits. The state constraints are often things such as temperature, concentration, or pressure limits while input constraints are typically physical constraints such as valve openings. The input rate change constraints work to limit how quickly a physical change can occur; these constraints help to limit the wear and tear on the physical process equipment. The last constraints simply require that all the weights be positive values.

3.3.2.2 NMPC Algorithm

It is common to use feedback on the outputs of the systems since state feedback is often not realistic; this is because an exact measure of the state at each time step is required for state feedback and this is typically not possible. Instead, a state estimate based on available measurements (i.e., available output data) is typically used instead. This is referred to as *output feedback NMPC*. A generalized output feedback NMPC algorithm is outlined in Algorithm 3.1.

Algorithm 3.1: NMPC with output feedback [7].

1 **for** $k = 0, 1, 2, \ldots$ **do**
2 Compute an estimate of the current state $\boldsymbol{x}_k$ based on the measured data up until time k.
3 Solve the NLP problem 3.12 on the prediction horizon from k to $k+N$ with $\boldsymbol{x}_k$ as the initial condition.
4 Apply the first control move $\boldsymbol{u}_k$ from the solution above.
5 **end**

The optimization problem requires an initial value and this initial value is typically found by computing a state estimate $\hat{\boldsymbol{x}}_{k+1}$ which relies on the latest available measurements [23]. State estimation is discussed further in Section 3.3.5. This form of NMPC which uses available measurements, i.e., available output data, is called *output feedback NMPC*.

3.3.2.3 Output and Input Trajectories

Industrial NMPC controllers use four different options to specify future CV behavior: a setpoint, zone, reference trajectory or funnel [28]. Here attention is given to the *reference trajectory* approach where a desired future path for each CV is specified. A curve is drawn from the current CV value to the setpoint, creating a reference trajectory. Any deviations from this trajectory are penalized as illustrated in Figure 3.8 where the gray area demonstrates the penalization. One drawback of this method is that it penalizes the output when it drifts too quickly towards the setpoint. However, in the case of model mismatch, the reference trajectory is beneficial in that it will slow down the CV and minimize the overshoot [28].

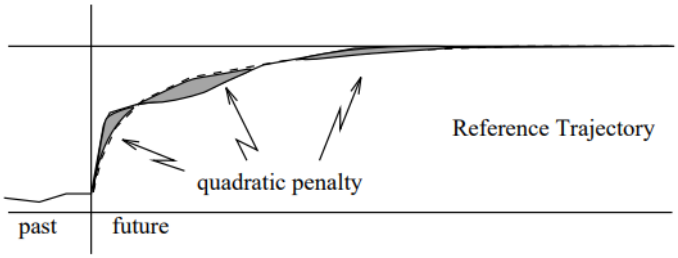


Figure 3.8: Illustration of how a reference trajectory is used for a controlled variable [28].

3.3.3 Supervisory Control Layer

The supervisory control layer involves recalculating the optimal trajectories in real time using updated values from the process; here emphasis is placed on the use of a DRTO algorithm in this level. DRTO is a closed-loop economic optimizer that computes setpoint targets for the lower level; this is done by solving a dynamic model at each step. Previously only steady-state models could be used for real-time optimization, which required the process to reach steady state before the updated trajectories could be calculated. Using a dynamic model means that the calculations can be performed more frequently since it is not necessary for the process to reach a steady before a new calculation is performed [12]. A problem of this type can be solved by following these six steps [2]:

1.) Identify the Process Variables

2.) Select the Objective Function

3.) Develop the Process Model and Constraints

4.) Simplify the Model and Objective Function

5.) Compute the Optimum

6.) Perform sensitivity studies

The process variables identified in this step are used in the process model and the objective function. An objective function is selected based on operating profit, product qualities and quantities, as well as the plant configuration. Process models are formulated and any operating limits are identified for the process variables. The model is constructed to be as simple as possible so that it can be solved rapidly enough to be used in an online, real-time application. Optimal setpoints are then determined using a selected optimization technique. Finally, the most sensitive parameters are identified by varying the model and cost parameters [2].

3.3.3.1 DRTO Problem Formulation

There are many different ways of formulating a DRTO problem. One option is to structure the DRTO problem in the form of a multilevel dynamic optimization problem with embedded NMPC optimization sub-problems. The two-level closed-loop DRTO problem formulation consists of a primary DRTO optimization problem using a nonlinear dynamic system (Problem 3.13) that predicts the closed-loop response dynamics, and an inner NMPC optimization sub-problem based on the dynamic system that calculates the optimal control input trajectories (Problem 3.14). Here the controller set-point trajectories $\hat{\boldsymbol{y}}^{SP}$ are the decision variables for the outer problem and the control input trajectories $\hat{\boldsymbol{u}}_k^{NMPC}$ are the decision variables for the inner problem [12].

$$
\begin{aligned}
&\min_{\hat{\boldsymbol{y}}^{SP}} \quad \Phi^{DRTO} \\
&\text{s.t.} \quad 0 = \tilde{\boldsymbol{f}}^{DRTO}(\hat{\boldsymbol{x}}_k^{DRTO}, \hat{\boldsymbol{z}}_k^{DRTO}, \hat{\boldsymbol{u}}_k^{DRTO}), \quad \text{for } k = 0,\ldots,N-1, \\
&\qquad 0 = \boldsymbol{h}^{DRTO}(\hat{\boldsymbol{x}}_k^{DRTO}, \hat{\boldsymbol{z}}_k^{DRTO}, \hat{\boldsymbol{u}}_k^{DRTO}), \\
&\qquad 0 \le \boldsymbol{g}^{DRTO}(\hat{\boldsymbol{x}}_k^{DRTO}, \hat{\boldsymbol{z}}_k^{DRTO}, \hat{\boldsymbol{u}}_k^{DRTO}), \\
&\qquad 0 = \boldsymbol{h}^{SP}(\hat{\boldsymbol{y}}^{SP}), \\
&\qquad 0 \le \boldsymbol{g}^{SP}(\hat{\boldsymbol{y}}^{SP})
\end{aligned}
\tag{3.13}
$$

$$
\begin{aligned}
(\hat{\boldsymbol{u}}^{DRTO} - \Delta\boldsymbol{u}^{ss}) = \min_{\hat{\boldsymbol{u}}^{NMPC}} \quad & \phi_{NMPC}(\hat{\boldsymbol{y}}^{NMPC}, \hat{\boldsymbol{y}}^{SP}, \Delta\hat{\boldsymbol{u}}^{NMPC}) \\
\text{s.t.} \quad & \hat{\boldsymbol{x}}_{k+1}^{NMPC} = \boldsymbol{f}^{NMPC}(\hat{\boldsymbol{x}}_k^{NMPC}, \hat{\boldsymbol{u}}_k^{NMPC}), \quad \text{for } k = 0,\ldots,N-1, \\
& 0 = \boldsymbol{h}^{NMPC}(\hat{\boldsymbol{x}}_k^{NMPC}, \hat{\boldsymbol{y}}_k^{NMPC}, \hat{\boldsymbol{u}}_k^{NMPC}, \hat{\boldsymbol{u}}_{k-1}^{NMPC}, \Delta\hat{\boldsymbol{u}}_k^{NMPC}), \\
& 0 \le \boldsymbol{g}^{NMPC}(\hat{\boldsymbol{u}}_k^{NMPC})
\end{aligned}
\tag{3.14}
$$

Where $\hat{\boldsymbol{u}}^{NMPC}, \hat{\boldsymbol{x}}^{DRTO}, \hat{\boldsymbol{y}}^{NMPC}, \Delta\hat{\boldsymbol{u}}^{NMPC}, \hat{\boldsymbol{y}}^{SP}$, and $\hat{\boldsymbol{u}}^{DRTO}$ are defined using the following notation:

$$
\hat{\boldsymbol{u}}^{DRTO} = \begin{bmatrix} (\hat{\boldsymbol{u}}^{DRTO}{}_0)^T & (\hat{\boldsymbol{u}}^{DRTO}{}_1)^T & \ldots & (\hat{\boldsymbol{u}}^{DRTO}{}_{N-1})^T \end{bmatrix}^T
$$

$\Delta\boldsymbol{u}^{ss}$ is a composite vector of the steady-state inputs repeated for each step of the DRTO prediction horizon for compatibility with $\hat{\boldsymbol{u}}^{DRTO}$.

Φ^{DRTO} is an objective function where $\widetilde{\boldsymbol{f}}^{DRTO}$ is the dynamic model used for DRTO prediction at step $k \in [0,\ldots,N-1]$. The equality constraints $\boldsymbol{h}^{DRTO}$ contain the algebraic equations of the prediction model. The inequality constraints $\boldsymbol{g}^{DRTO}$ contain the output constraints, which could comprise a subset of the states. The equality $\boldsymbol{h}^{SP}$ enforces the NMPC setpoint trajectories so that they are constant over each DRTO interval within each optimization horizon, which is an integer multiple of the NMPC sampling interval; the inequality constraints $\boldsymbol{g}^{SP}$ gives an upper and lower bound of these setpoint trajectories. Note that input constraints are not imposed here but are instead taken care of by the NMPC optimization subproblem [12].

The NMPC sub-problem utilizes the state-space formulation of a standard input-constrained NMPC controller with a quadratic objective function. Here only a single NMPC optimization problem is required within the DRTO calculation; the prediction and control horizons are extended to match that of the DRTO optimization horizon. ϕ_{NMPC} is a quadratic cost function and $\boldsymbol{f}^{NMPC}$ is the NMPC prediction model at time step $k \in [0,\ldots,N-1]$. The equality constraints $\boldsymbol{h}^{NMPC}$ contain the relations between the states and the inputs as well as between the inputs and input changes. The equality constraints $\boldsymbol{g}^{NMPC}$ consist of the input constraints. $\hat{\boldsymbol{x}}_k^{NMPC} \in \mathbb{R}^{n_{\boldsymbol{x}}}$ is a vector of the NMPC model states, $\hat{\boldsymbol{y}}_k^{NMPC} \in \mathbb{R}^{n_{\boldsymbol{y}}}$ is a vector of the corresponding NMPC model outputs, and $\hat{\boldsymbol{u}}_k^{NMPC} \in \mathbb{R}^{n_{\boldsymbol{u}}}$ is a vector of NMPC inputs at each prediction step. $\hat{\boldsymbol{y}}^{SP} \in \mathbb{R}^{N.n_{\boldsymbol{y}}}$ is a composite vector of the NMPC setpoint trajectories for the controlled outputs over the DRTO optimization horizon N.

The inputs to the plant model correspond to the optimal input trajectory of the NMPC sub-problem plus the nominal steady state inputs. The closed-loop DRTO strategy optimizes the setpoint trajectories directly and passes them to the NMPC controller at the level below. These setpoint trajectories are then shifted in time to account for the moving horizon of the NMPC controller [12].

This supervisory control structure can be used to further optimize a single process or a whole plant. Here, this control structure will be used to optimize a single process. Therefore, the construction of the DRTO objective function is not based in economics; further discussion of the application of such control structures is outlined in Chapter 6.

3.3.4 Problem Size Reduction

The computational effort required to solve a dynamic optimization problem is dependent on the number of decision variables, which is equal to the sum of the states and inputs. This can be mathematical expressed as:

$$n = N(n_{\boldsymbol{x}} + n_{\boldsymbol{u}}) \tag{3.15}$$

Here N is the prediction horizon, $n_{\boldsymbol{x}}$ is the number of state variables, and $n_{\boldsymbol{u}}$ is the number of input variables.

For large systems, it is often desirable to reduce the size of the problem to decrease this computational effort; this can be done using *control input blocking*. The idea is illustrated below in Figure 3.9. Here, the manipulated variables are forced to take constant values over intervals spanning one or more time steps [7]. Typically, input variables are allowed to move frequently at the beginning of the prediction horizon while towards the end, the block lengths increase since the states typically have settled to a steady value by this point. It has been shown that input blocking is able to reduce computational effort with little consequences in control quality [7].

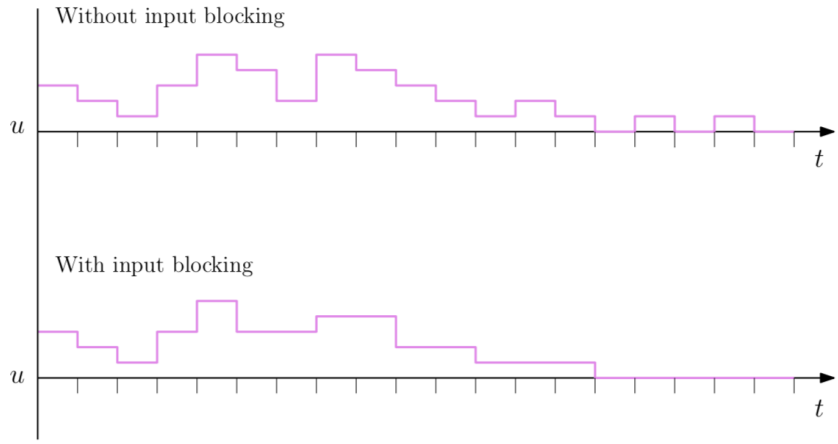


Figure 3.9: Illustration of how the use of input blocking alters the predicted inputs [21].

3.3.5 Parameter and State Estimation

Parameter and state estimation accuracy is critical for the development of good process models and control algorithms, respectively. Here, attention is given to one offline parameter estimation method and one method of state estimation. Parameter estimation is done to help improve the parameters used in the model based on process data; this can be done online or offline. State estimation is done for states that cannot be directly measured and is only done online.

Typically, estimator blocks are included in close connection with the process model block. The estimator works by continuously reinforcing the process model based on the outputs from the real process [9]. Figure 3.10 shows how the process model is updated using online state and parameter estimations that are calculated using the model deviation from the plant measurements. The input is injected to the process, the model, and the estimator; the process and the model also receive any distur-

bances. The model calculates the states and predicts the measurements. For the current time step, the model predicted measurements and the actual measurements from the process are sent to the estimator. The estimator uses the input value and the difference between the measurements to calculate the states and parameters for the next time step. These states and parameters are feed back to the model to improve its accuracy.

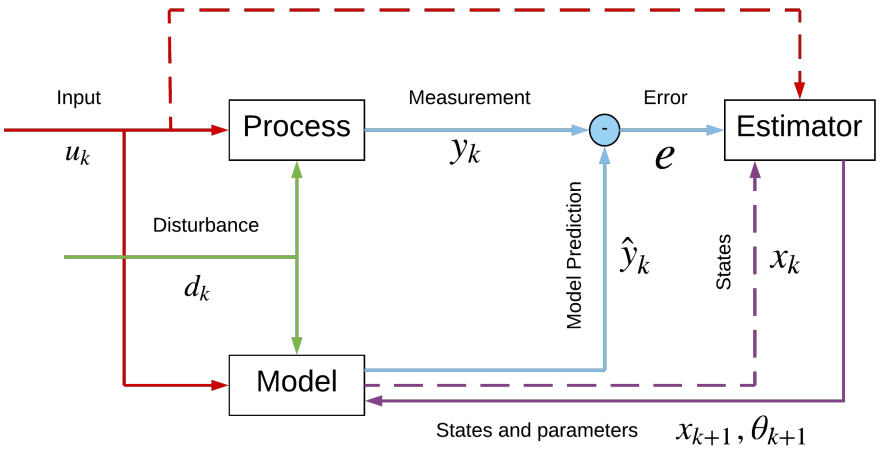


Figure 3.10: Block diagram of online state and parameter estimation.

3.3.5.1 Parameter Estimation

When a model is first developed, the parameter values are selected from literature data, experiments, and/or calculations; however, there is often uncertainty in these parameters that can result in model mismatch, causing poor performance in model based controllers. The goal is to estimate the parameters such that they describe the real process behavior as precisely as possible relative to the plant measurements [27].

Parameter estimation can be done either online or offline. Offline parameter estimation can be used to adjust time-invariant parameters that have uncertainty. A common method of offline parameter estimation is done by solving a least squares optimization problem, where the squared difference between the plant measurements and model predicted measurements is minimized. Such a formulation is shown in Equation 3.16 [21].

$$\begin{aligned} \min_{\boldsymbol{\eta}} \quad & \sum_{k=1}^{N} (\hat{\boldsymbol{y}}_k - \boldsymbol{y}_k)^2 \\ \text{s.t.} \quad & \boldsymbol{x}_{k+1} = \boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\theta}), \\ & \hat{\boldsymbol{y}}_k = \boldsymbol{g}(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\theta}), \\ & \boldsymbol{\eta} \subseteq \boldsymbol{\theta} \end{aligned} \tag{3.16}$$

This is a simple regression method where the solution is selected such that the estimated parameters $\boldsymbol{\eta}$ minimize the squared difference between the predicted measurements $\hat{\boldsymbol{y}}_k$ and the plant measurements $\boldsymbol{y}_k$, where k refers to a point in time. The decision variables $\boldsymbol{\eta}$ are selected from a subset of the entire parameter set $\boldsymbol{\theta}$ to decrease the problem size; parameters with the largest uncertainty are typically chosen as decision variables. Since the problem is nonlinear due to the nonlinearities in the model constraints, it must therefore be solved using a nonlinear method. It is common to utilize the SQP algorithm that was briefly discussed in Section 3.2.5.1 to solve these problems.

Online parameter estimation is used in the event the parameters are uncertain and time varying. In this case, the model parameters are continually adjusted according to real-time measurements from the plant. This results in an improved fit between the model and the plant; hence, improving the performance of the model based control optimization [29]. Online parameter estimation is often combined with state estimation. In this work parameter estimation is not considered further but it is important to understand the concept nonetheless.

3.3.5.2 State Estimation

State estimators give information about the process that cannot be measured directly or if the measurements are too expensive to obtain or too noisy. For polymer processes, credible measurements are rarely available, which causes issues in online parameter estimation; in this case, state estimators are essential in providing insight into how to apply optimal control [21].

Online estimation methods that can be considered for the problem type of interest are Kalman filter (KF) based estimators and moving horizon estimator (MHE) [29]. The KF is the more common approach to state estimation and is the type utilized in this work. This state estimator produces an optimal estimate in the sense that the mean value of the sum of the estimation errors gets a minimal value [10]. This algorithm requires that the states that are to be estimated must be observable. A state is *observable* if the state can be determined from the system model, along with its inputs and outputs for a finite number of steps [7]. The algorithm assumes that the system in which the states are estimated is excited by random disturbances (or process noise) and that the measurements contain random measurement noise [10].

The KF works in two steps: 1) predicting the state and the estimates' covariance and 2) updating the measurement and covariance of measurement residuals. The Kalman gain is used to update the state estimate and the covariance of the state estimate [27]. This is referred to as a recursive estimation method, which can be formulated generically as:

$$\hat{\boldsymbol{x}}_k = \bar{\boldsymbol{x}}_k + \boldsymbol{K}_k(\boldsymbol{y}_k - \bar{\boldsymbol{y}}_k) \tag{3.17}$$

Here $\bar{\boldsymbol{x}}_k$ is the vector of estimated states at step k, and $\boldsymbol{K}_k$ is the KF gain matrix, which determines how much the deviation from the plant measurements affects the updated parameters. Recursive estimation methods are ideal since they are efficient in terms of memory, and the new parameter estimates are based on both the measurement history and the previous parameter estimates [21].

The KF is derived for linear systems but can be extended to nonlinear systems, which is known as the extended Kalman filter (EKF). This state estimator is then written as:

Prediction part:

$$\bar{\boldsymbol{x}}_k = \boldsymbol{f}(\hat{\boldsymbol{x}}_{k-1}, \boldsymbol{\theta}, \boldsymbol{u}_{k-1}, \bar{\boldsymbol{v}}_{k-1}) \tag{3.18}$$

$$\bar{\boldsymbol{y}}_k = \boldsymbol{g}(\bar{\boldsymbol{x}}_k, \boldsymbol{\theta}, \boldsymbol{u}_{k-1}) + \bar{\boldsymbol{w}}_k \tag{3.19}$$

$$\bar{\boldsymbol{z}}_k = \boldsymbol{h}(\bar{\boldsymbol{x}}_k, \boldsymbol{\theta}, \boldsymbol{u}_{k-1}) \tag{3.20}$$

Measurement correction:

$$\hat{\boldsymbol{x}}_k = \bar{\boldsymbol{x}}_k + \boldsymbol{K}_k(\boldsymbol{y}_k - \bar{\boldsymbol{y}}_k) \tag{3.21}$$

$$\hat{\boldsymbol{z}}_k = \boldsymbol{h}(\hat{\boldsymbol{x}}_k, \boldsymbol{\theta}, \boldsymbol{u}_{k-1}) \tag{3.22}$$

where $\bar{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}(t_{k|k-1})$ is the *a priori* state estimate, $\bar{\boldsymbol{y}}_k$ is the predicted measurement, $\bar{\boldsymbol{z}}_k$ is the predicted output, $\hat{\boldsymbol{x}}_{k-1} = \hat{\boldsymbol{x}}(t_{k-1|k-1})$ is the state estimate at t_{k-1}, $\bar{\boldsymbol{v}}_{k-1}$ is the mean process noise and $\bar{\boldsymbol{w}}_k$ is the mean measurement noise.

The KF gain matrix $\boldsymbol{K}_k$ can either be a constant or, more commonly, it is calculated from the process noise covariance $\boldsymbol{V}_{k-1}$, the measurement noise covariance $\boldsymbol{W}_k$, *a priori* state covariance $\overline{\boldsymbol{X}}_{k-1}$ and the partial derivatives of the state prediction function $\boldsymbol{f}$ and the measurement prediction function $\boldsymbol{g}$. The process noise covariance is calculated using the standard deviation of the process noise, which is treated as an estimator parameter. The measurement noise covariance is calculated in a similar manner. The *a priori* state covariance's initial value is an estimator parameter defined by the initial state standard deviations.

3.3.5.3 Parameter and State Estimation Combined

In many applications, the KF can be used to estimate parameters and/or disturbances in addition to the state variables [10]. To include parameter estimation in these algorithms, the state vector is replaced by an augmented state vector that contains both the states and parameters. This is known as the augmented Kalman filter (AKF) and is given by [21]:

Prediction part:

$$\bar{\boldsymbol{x}}_k = \boldsymbol{f}(\hat{\boldsymbol{x}}_{k-1}, \hat{\boldsymbol{\theta}}_{k-1}, \boldsymbol{u}_{k-1}, \bar{\boldsymbol{v}}_{k-1}) \tag{3.23}$$

$$\bar{\boldsymbol{\theta}}_k = \hat{\boldsymbol{\theta}}_{k-1} + \hat{\boldsymbol{\omega}}_{\boldsymbol{k-1}} \tag{3.24}$$

$$\bar{\boldsymbol{y}}_k = \boldsymbol{g}(, \bar{\boldsymbol{\theta}}_k, \boldsymbol{u}_{k-1}) + \bar{\boldsymbol{v}}_{k-1} \tag{3.25}$$

Measurement correction:

$$\begin{bmatrix} \hat{\boldsymbol{x}}_k \\ \hat{\boldsymbol{\theta}}_{\boldsymbol{k}} \end{bmatrix} = \begin{bmatrix} \bar{\boldsymbol{x}}_k \\ \bar{\boldsymbol{\theta}}_k \end{bmatrix} + \boldsymbol{K}_k(\boldsymbol{y}_k - \bar{\boldsymbol{y}}_k) \tag{3.26}$$

CHAPTER 4

SEMI-BATCH STYRENE POLYMERIZATION CASE STUDY

The concepts outlined in the previous chapters are now applied to the semi-batch polymerization of EPS. A description of the process outlining how EPS is typically manufactured is given in Section 4.1. The equations that make up the process model are given in Section 4.2; this includes first principle equations (mass and energy balances) along with the reaction equations (moment balances and reaction rates). Section 4.3 explains how the model is implemented and introduces the software that is utilized throughout this work. The numerical method that is used to solve the DAEs is briefly covered along with how parameter and state estimation is done.

Using the constructed model, offline optimization is performed in Section 4.4 to find some base line optimal operating values that will be used to develop the advanced control structure. The offline optimization variable selection is given along with the resulting offline optimization problem formulation. Details of how the offline optimization problem is solved is discussed in Section 4.5. The offline optimization results are then presented.

4.1 Process Description

EPS is typically produced by suspension, free radical polymerization of styrene in a semi-batch reactor. The reactor is initially charged with liquid styrene that is distributed finely in water. When sufficient amounts of initiator and heat are added, the styrene polymerizes to form harder polystyrene beads.

A generic process schematic is shown in Figure 4.1. There is one inlet point where the monomer, initiator, and additives are added. The reactor is preloaded with monomer and water with the addition of the initiator indicating the start of the batch; the additives are added at some point during the batch. Since it is a semi-batch reactor, the outlet is only utilized at the end of the batch to extract the product. A blade is used to stir the reactor contents, thereby limiting the concentration and temperature gradients. A jacket surrounds the reactor and is used to regulate the temperature by using cooling water to remove any excess heat generated by the reactions. This helps maintain the reactor temperature between the necessary

range of 100 – 140 °C to produce the desired polymer properties [21].

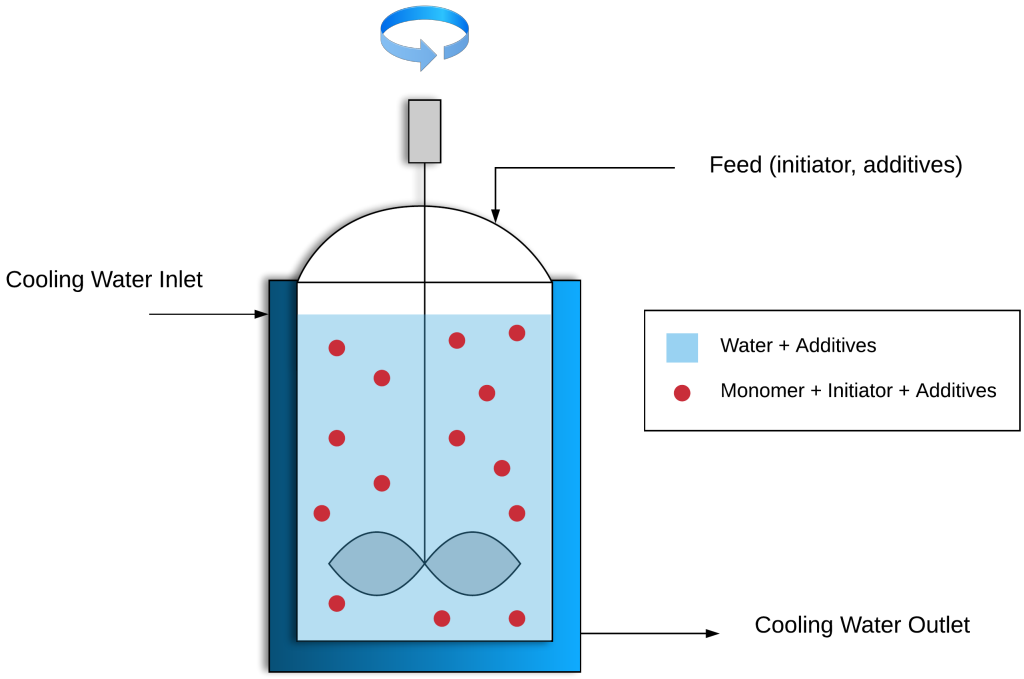


Figure 4.1: A generalized semi-batch reactor for suspension polymerization of expandable polystyrene.

Additives are occasionally needed in polymerization reactions to help produce the desired product; two types of additives known as *bead stabilizers* and *blowing agents* are commonly used in the production of EPS. Bead stabilizers are added to avoid aggregation of the dispersed phase; aggregates that sink to the reactor bottom can cause poor heat transfer conditions, which may eventually lead to thermal runaway [21]. Blowing agents are oil soluble chemicals that swell into beads whose addition as the reaction occurs helps in the reduction of the reaction volume density; for EPS production, pentane is commonly used. The addition of the blowing agent causes the particles to swell, resulting in the expandability property of the polystyrene beads; as a consequence, the droplet viscosity of the liquid decreases. Therefore, the amount of pentane added, and the time at which it is added, has a large effect on the final products properties [21].

The chemical reactions begin when initiator and enough heat is supplied. A radical chemical initiator dissociates and reacts with monomer droplets at temperatures above 100 °C. The initiator radical fragments take up styrene units creating growing radical chains before eventually becoming dead polystyrene molecules. Initiators

are typically expensive chemicals and therefore, the amount of initiator used should be the correct amount for the fixed monomer mass so that little is wasted.

Once polymerization is complete, the added blowing agent needs to be removed so it is not present in the final product. When the process has reached a high monomer conversion, the EPS beads are heated with steam causing the blowing agent to boil off. This softens the polymer and causes the beads to expand to eight times their initial volume. The final average bead diameter is approximately 0.5 to 1 mm [15].

Once the reaction has reached a set conversion, the product is removed from the reactor. The point at which this conversion is reached is known as the *batch time*. The shorter this is, the more product that can be manufactured annually and consequently, the larger the profits of the plant. Therefore, the batch length is of critical importance when trying to improve such processes.

4.1.1 Reaction Kinetics

Polystyrene is made by free-radical polymerization of styrene monomers; this type of polymerization has three main reaction types that occur simultaneously: initiation, propagation, and termination. In this section, these reactions and corresponding reaction rates for the production of EPS specifically are covered.

Both chemical and thermal initiation occur during the polymerization of polystyrene. Chemical initiation was previously defined in Chapter 2 Section 2.1.1 so is not discussed again here. Thermal initiation only becomes substantial once the temperature is sufficiently high (> 100°C); styrene molecules form radical components independent of the additives. This process is illustrated below in Reactions 4.1-4.4 [21]. During thermal initiation, two styrene molecules undergo a Diels-Alder reaction to form Diels-Alder adduct 1-phenyltetralin (AH) [21]. When this molecule reacts with a styrene molecule M, two radical molecules $M^{\bullet}, A^{\bullet}$ are formed that can further initiate a polymer chain.

$$2\,M \underset{k_{-1}}{\overset{k_1}{\longleftrightarrow}} AH \tag{4.1}$$

$$M + AH \xrightarrow{k_{dm}} M^{\bullet} + A^{\bullet} \tag{4.2}$$

$$M + M^{\bullet} \xrightarrow{k_p} R_1 \tag{4.3}$$

$$M + A^{\bullet} \xrightarrow{k_p} R_1 \tag{4.4}$$

Activation of a styrene chain by a radical component and the propagation reaction of a growing chain are shown in Figure 4.2. The free radical of the molecule forms a single bond with one of the electrons from the double bond of styrene; this results in the addition of a styrene molecule to either the initiator radical or growing chain. The phenyl group's position depends on the direction which the styrene molecule

approaches the radical; normal polystyrene is atactic meaning that the phenyl group direction is random [21].

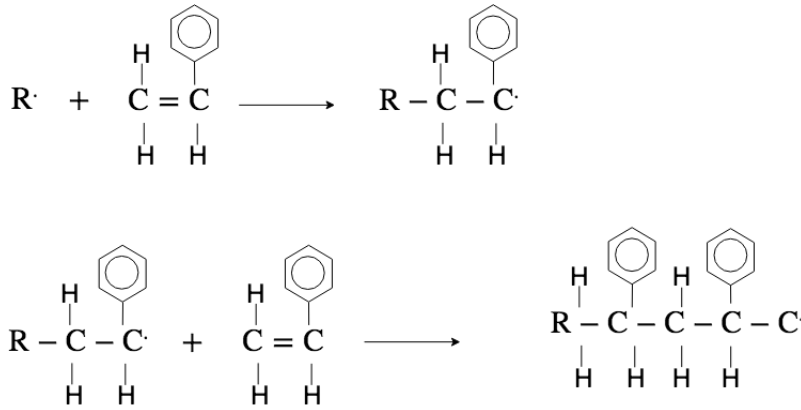


Figure 4.2: Activation of a new growing chain polystyrene and the propagation process [21].

Radical polystyrene chains are primarily terminated through combination; therefore, termination by disproportion is negligible in the polymerization of styrene [1]. Another termination possibility is chain transfer, which is where activated polymer chains react with CTA or monomers. In chain transfer to a monomer, a monomer unit terminates the growing chain and the monomer unit then carries the radical. Chain transfer to CTA proceeds in the same way but instead of a monomer unit, the CTA is responsible [9]. In some cases, CTA is added to the reactor to control the chain length of the polymer product [21].

4.2 Model

While the development of the mathematical model was conducted by Lund, a brief summary is given here since the model is crucial to the development of a good control structure [21]. The assumptions listed below were made to create a simplified model that could be used in optimization and control.

- The chemical initiator is mono-functional.
- The reactor is charged with monomer and water such that the monomer is dispersed into tiny organic phase droplets in the continuous water phase.
- The reaction is initiated by adding all the chemical initiator at time $t = 0$.
- Pentane is added at some unknown point during the reaction and is fed to the reactor over a short period of time.
- The delay between the addition of initiator and the onset of the initiation reaction, as well as for the swelling of pentane into the particles, are all

neglected.

- The solubilities of styrene, initiator, polystyrene, and pentane in water, and vice versa, are assumed to be negligible. In reality, styrene and pentane are slightly soluble in water, as shown in Table 4.1. This solubility could result in a small amount of the monomer not being available for reaction. In addition, some pentane will be lost to the water phase so less pentane will contribute to reduction of the diffusional limitation and the expansion properties of the polymer beads. On the other hand, water can also dissolve in the polymer phase and increase its total and free volumes.

Table 4.1: Solubility of compounds in water and vice versa at 20 °C [21].

Compound	**Solubility** [$\mathrm{kg\,m^{-3}}$]	**Water solubility** [$\mathrm{kg\,m^{-3}}$]
Styrene	0.29	0.54
Polystyrene	0.00	0.00
Pentane	0.40	0.09
Initiator	0.00-0.01	0.00

- All reactions are treated as elementary and irreversible.
- The pseudo-steady state assumption (PSSA) can be applied to the moment balances of living chains; this eliminates the stiff condition of the differential equations originating from the radical dynamics. This assumption is valid to apply since radical chains are extremely reactive intermediate species so they have short lives.
- Mass transfer limitations between the polymer phase and the water phase are neglected; thus, the process is modeled without temperature gradients in the reactor content. This is also known as a perfect mixing assumption.
- The pressure is assumed constant.

4.2.1 Dynamic Model Equations

A dynamic model was constructed consisting of component balances for the reactor contents plus energy balances on the reactor vessel and cooling jacket. The parameter values and other constants used in the model are provided in Appendix B.

4.2.1.1 Species Balances, Moment Balances, and Reaction Rates

The species and moment balances for the polymerization of polystyrene are outlined below. Also included in this section are the equations for the reaction rates and reaction rate constants.

Species Balances:

The monomer is pre-loaded along with water and the initiator is added at the beginning of the batch ($t = 0$); more reactants can be fed throughout the batch reaction ($t > 0$) if required. Before the initiator is added, the reactor is heated to an optimal initial temperature.

Both styrene monomer and initiator are consumed in the reaction process. The dynamic species balances for monomer and initiator are:

$$\frac{dn_M}{dt} = R_M V + \hat{n}_M \tag{4.5}$$

$$\frac{dn_I}{dt} = R_I V + \hat{n}_I \tag{4.6}$$

where n_i is the amount of reactant i present in the system and $\hat{n}_i$ is the corresponding molar flow of the reactants. R_M and R_I are the reaction rates of the monomer and initiator, respectively. The reaction rates are calculated using:

$$R_M = (-k_p - k_{trM})[\mathrm{M}][\lambda_0] - 2fk_d[\mathrm{I}] - 2k_{dm}[\mathrm{M}]^3 \tag{4.7}$$

$$R_I = -2fk_d[\mathrm{I}] \tag{4.8}$$

Here k_p, k_{trM}, k_d and k_{dm} are the rate constants for propagation, chain transfer to monomer, chemical initiation, and thermal initiation, respectively. The efficiency of the chemical initiator is expressed by f and is a number between zero and one. The concentrations of initiator and monomer are represented by [I] and [M]. Finally, λ_0 is the zeroth order moment of active polymer chains on a concentration basis.

Pentane is fed to the reactor at some point but does not contribute to any of the reactions itself. The balance equation for pentane is given by:

$$\frac{dn_{C5}}{dt} = \hat{n}_{C5} \tag{4.9}$$

where the amount $\hat{n}_{C5}$ added to the system has a direct effect on the volume and viscosity of the reactive phase.

Moment Balances of Living and Dead Chains:

The moment balances are based on work by Wu et al. and were extended to include chemical initiation terms [38]. The zeroth order moment is given by Equation 4.10 and represents the rate of change of the living chains. Active chains are produced by two initiation processes and are consumed by termination; these reactions are illustrated by the three terms in Equation 4.10. The first order moment of living chains is given in Equation 4.11 and is related to the total chain length of the living

chains. This equation shows that initiation and propagation increase the chain length and termination stops the growth. The transfer to monomer ends the growth of one chain and initiates the growth of another. The second order moment is shown in Equation 4.12 and is almost equivalent to the balance for the first order moments. The PSSA means that all three equations can be solved directly for each of the moments for the living chains [21].

$$\frac{d\lambda_0}{dt} = (2fk_d[\mathrm{I}] + 2k_{dm}[\mathrm{M}]^3 - k_{tc}[\lambda_0]^2)V = 0 \tag{4.10}$$

$$\begin{aligned}\frac{d\lambda_1}{dt} &= (2fk_d[\mathrm{I}] + 2k_{dm}[\mathrm{M}]^3 + k_{trM}[\mathrm{M}][\lambda_0] + k_p[\mathrm{M}][\lambda_0] \\ &\quad - k_{trM}[\mathrm{M}][\lambda_1] - k_{tc}[\lambda_0][\lambda_1])V = 0\end{aligned} \tag{4.11}$$

$$\begin{aligned}\frac{d\lambda_2}{dt} &= (2fk_p[\mathrm{I}] + 2k_{dm}[\mathrm{M}]^3 + k_{trM}[\mathrm{M}][\lambda_1] + k_p[\mathrm{M}]([\lambda_0] + 2[\lambda_2]) \\ &\quad - k_{trM}[\mathrm{M}][\lambda_2] - k_{tc}[\lambda_0][\lambda_2])V = 0\end{aligned} \tag{4.12}$$

The moment balances for the dead chains are given below in Equations 4.13-4.15. The zeroth order moment increases as the active chains are terminated either by radical transfer to a monomer or termination by combination. The first order moment, related to the number of monomers tied in died chains, increases by the same termination processes. The second order moment is similar to the first order moment balance with an additional termination term [21].

$$\frac{d\mu_0}{dt} = \left(k_{trM}[\mathrm{M}][\lambda_0] + k_{tc}\frac{[\lambda_0]^2}{2}\right)V \tag{4.13}$$

$$\frac{d\mu_1}{dt} = \left(k_{trM}[\mathrm{M}][\lambda_1] + k_{tc}[\lambda_1][\lambda_0]\right)V \tag{4.14}$$

$$\frac{d\mu_2}{dt} = \left(k_{trM}[\mathrm{M}][\lambda_2] + k_{tc}\left([\lambda_2][\lambda_0] + [\lambda_1]^2\right)\right)V \tag{4.15}$$

Free Volumes:

The free volumes of styrene, polystyrene and pentane are calculated using:

$$V_{f,M} = \left(0.025 + \alpha_M\left(T_R - T_{g,M}\right)\right)\frac{n_M V_{m,M}}{V} \tag{4.16}$$

$$V_{f,P} = \left(0.025 + \alpha_P\left(T_R - T_{g,P}\right)\right)\frac{\mu_0 V_{m,P}}{V} \tag{4.17}$$

$$V_{f,C5} = \left(0.025 + \alpha_{C5}\left(T_R - T_{g,C5}\right)\right)\frac{n_{C5} V_{m,C5}}{V} \tag{4.18}$$

where α_i is the fractional free volume, $T_{g,i}$ is the glass transition temperature, n_i is the molar mass, and $V_{m,i}$ is the molar volume. The index i represents either the

monomer, polymer or pentane. The total free volume is the sum of the contributions of each of the components.

$$V_f = V_{f,M} + V_{f,P} + V_{f,C5} \tag{4.19}$$

The free volume is used in the calculation of some of the reaction rates.

Rate Constants:

The reaction rate constants are calculated using the Arrhenius expression:

$$k_i = k_{i,0} \exp\left(-\frac{E_i}{RT_R}\right), \qquad i = \{d, dm, p, tc, trM\} \tag{4.20}$$

where $k_{i,0}$ is the frequency factor, E_i is the activation energy, R is the gas constant, and T_R is the reactor temperature.

Reaction Rates:

During the polymerization process, different reactions control the termination rate; therefore, there are several different reaction rates that make up the overall termination rate. The reaction rate equations for each of the termination reaction types are stated below, along with the cage and glass effect contributions to the overall reaction rate.

At low monomer conversion and low viscosity, *segmental diffusion* controls the diffusional limitations. The segmental diffusion-controlled termination rate constant is given by:

$$k_{tc,seg} = \left[1 + \frac{\delta_c \mu_0 M_{w,M}}{V}\right] k_{tc} \tag{4.21}$$

Here the segmental diffusion parameter for styrene δ_c is close to zero; therefore, this effect is very small and could safely be neglected with little impact on the model's accuracy.

At higher conversions, the *translational diffusion* begins to control the termination rate. The rate constant is given by:

$$k_{tc,trans} = \left(\frac{\overline{M}_w}{\overline{M}_{w,cr}}\right)^a \exp\left[-A\left(\frac{1}{V_f} - \frac{1}{V_{f,cr}}\right)\right] k_{tc} \tag{4.22}$$

Here, $\overline{M}_{w,cr}$ and $V_{f,cr}$ denote the weight average molecular weight and free volume of the mixture at the onset of the translational diffusion effect, respectively; A and a are both tuning parameters. The onset of this regime is indicated by a critical point, where a test variable K becomes equal to a critical variable K_{cr}. The test

variable can be tuned by changing two parameters: l and A_l. The critical variable can be tuned by changing A_{cr}.

$$K = \overline{M}_w{}^l \exp\left(\frac{A_l}{V_f}\right) \tag{4.23}$$

$$K_{cr} = A_{cr} \exp\left(\frac{E_{cr}}{RT_R}\right) \tag{4.24}$$

The *reaction* or *residual diffusion* contribution always acts on the system regardless of the conversion. The size of this effect is determined by the following linear conversion function between an upper and lower bound:

$$k_{tc,rd} = k_{tc,rd,min}X + k_{tc,rd,max}(1-X) \tag{4.25}$$

The diffusional effects are gathered together into one expression for the overall termination rate:

$$k_{tc,app} = \left(\frac{1}{k_{tc,seg}} + \frac{1}{k_{tc,trans}}\right)^{-1} + k_{tc,rd} \tag{4.26}$$

For the cage and glass effects, the following expressions are used:

$$f_{app} = \exp\left[-B\left(\frac{1}{V_f} - \frac{1}{V_{f,cr,d}}\right)\right] f \tag{4.27}$$

$$k_{p,app} = \exp\left[-C\left(\frac{1}{V_f} - \frac{1}{V_{f,cr,p}}\right)\right] k_p \tag{4.28}$$

where f_{app} is the apparent initiator efficiency and $k_{p,app}$ is the apparent propagation rate constant. These reaction rate equations should be inserted into the model equations discussed Chapter 2.

4.2.1.2 Energy Balances

The energy balances for the reactor and the jacket are constructed in the form of temperature equations. This is done because temperature is an easily measured quantity compared to enthalpy. The full derivation of the energy balances can be found in Appendix B.

The temperature equation for the reactor is given by:

$$\frac{dT_R}{dt} = \frac{\sum_i c_{p,i}\hat{n}_i(T_{feed} - T_R) - \Delta H_R R_P V - (UA)_J(T_R - T_J) - (UA)_{amb}(T_R - T_{amb}) + W_{ag}}{\sum_i n_i c_{p,i} + m_V c_{p,V}} \tag{4.29}$$

The first term represents the enthalpy from the inlet where T_{feed} is the feed temperature. For free-radical polymerization, the bulk of the reaction heat generated comes from the propagation reaction; thus the propagation rate R_P is used in the heat of

reaction term. The propagation rate depends on the concentrations of monomer and living polymer chains and is calculated using:

$$R_P = k_p[\mathrm{M}][\lambda_0] \tag{4.30}$$

The terms $(UA)_J$ and $(UA)_{amb}$ are the overall heat transfer coefficients for heat transfer to the cooling water and surroundings. T_J and T_{amb} are the cooling and ambient temperatures. In this case, the shaft work done on the system is from the agitation W_{ag}. Currently, the model code does not consider the agitation work or the heat loss to the surrounding environment since these contributions are deemed minor in comparison with the other energy contributions. However, the code can be easily altered to include nonzero values for these terms.

The temperature equation for the cooling jacket is given by:

$$\frac{dT_J}{dt} = \frac{(UA)_J(T_R - T_J) + \hat{m}_c c_{p,c}(T_{J,i} - T_{J,o})}{m_c c_{p,c}} \tag{4.31}$$

where $\hat{m}_c$ is the mass flow rate of the cooling fluid, $c_{p,c}$ is the heat capacity of the cooling fluid, $T_{J,i}$ and $T_{J,o}$ are the inlet and outlet temperatures of the jacket, and m_c is the mass of cooling fluid contained in the jacket [21].

4.3 Model Implementation

The model is formulated in the programming language C using a model template provided by Cybernetica AS that facilitates integration into Cybernetica's software. The following model structure is used in this template:

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(t, \boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{\theta}) \tag{4.32}$$

$$\boldsymbol{x}_0 = \text{given} \tag{4.33}$$

$$\hat{\boldsymbol{y}}(t) = \boldsymbol{g}(t, \boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{\theta}) \tag{4.34}$$

$$\boldsymbol{z}(t) = \boldsymbol{h}(t, \boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{\theta}) \tag{4.35}$$

where $\boldsymbol{x}(t)$ is the state vector, $\boldsymbol{u}(t)$ is the input vector, $\boldsymbol{\theta}$ is the parameter vector, $\hat{\boldsymbol{y}}(t)$ is the (predicted) measurement vector, and $\boldsymbol{z}(t)$ is the output vector. The model equations are constructed as functions in the template and the numerical method to solve the model is selected here.

To help with the implementation of the control structure, the states, inputs, measurements, outputs, parameters and constants are declared explicitly in the template. Interfaces to other useful applications such as Cybernetica ModelFit and Cybernetica CENIT are also supplied.

4.3.1 Software

Cybernetica AS is a company that focuses on developing custom advanced model-based control structures for nonlinear processes. They have developed their own

suite of software designed for this purpose, which consists of ModelFit, CENIT, and RealSim. ModelFit is designed for parameter estimation and model verification; this tool is not utilized in this work so it is not discussed further. RealSim is used to simulate an instance of the model to behave as if it were the plant and CENIT is the nonlinear optimization tool; RealSim and CENIT communicate through the use of an OPC server. All three pieces of software are designed to work with a process model implemented in C using Cybernetica's template structure. This model typically extends beyond the balance and reaction equations mentioned in the previous sections of this chapter to include the information necessary for the control structure; this is covered in more detail in the relevant chapters and sections. The work in this thesis is conducted using CENIT and RealSim to implement the two control levels and perform simulations, respectively.

Cybernetica CENIT is designed as an NMPC controller software using nonlinear first principles based models; it can be implemented to control continuous or batch processes in an optimal manner. This software utilizes the custom developed model. CENIT can compensate for any model mismatch by using one of its two integrated estimators: the KF or the MHE. This software employs a SQP algorithm using a line-search method to solve dynamic optimization problems; small perturbations are introduced to the control inputs producing a sensitivity matrix from which the best solution is identified [17]. CENIT has a GUI known as CENIT MMI, which allows users to easily change settings in CENIT such as turn on/off constraints, change variable types, set new setpoints, or alter the controller weights. In the CENIT MMI different variables can be plotted allowing users to see both the current and the predicted states. While CENIT is typically utilized to perform online optimization, it can also be used for offline optimization where the model is used to simulate the real plant.

Cybernetica RealSim serves as a simulation tool and a test-bench for online control applications [17]. The controller can be tested offline and tuned prior to installation in addition to allowing for open-loop testing and debugging of the controller. Users can even simulate real operation where they can change parameters as a real operator would. Together with CENIT, RealSim can serve as a real-time simulator for testing various controller settings on the plant. A separate instance of the same model that is used by CENIT is run in RealSim where the values are synchronized through an OPC server; this allows the simulation to be executed more rapidly than a real-time process [17]. The main motivation to use an OPC server for offline simulations is that it ensures that the simulated conditions are as similar as possible to online conditions; this also negates the need to alter anything in the model if it is to used online or offline. RealSim determines the "measurements" which are then passed to CENIT to be used in the control layer(s). CENIT calculates and executes the appropriate control action based on these measured values. The simulation can

be run infinitely, paused, or simulated for just one sample. RealSim displays the current data values in the main window. In this GUI, the user can modify inputs and parameters when the simulation is paused.

4.3.1.1 Numerical Solving Method

The framework provides several different numerical solution methods with the simplest method being Euler integration. Since this model is a non-stiff system due to the PSSA assumption on the living chain moments, the Euler method is an adequate solver. Euler's method works to solve initial value first-order ordinary differential equation (ODE) systems by calculating approximate solution values at equidistant points using the initial value [19]. Specifically, explicit Euler integration is used, which is the simplest Runge-Kutta method.

Applying Euler's method results in the following system:

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(t, \boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{\theta}) \tag{4.36}$$

$$\boldsymbol{x}_0 = \text{given} \tag{4.37}$$

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \Delta t_{int} \boldsymbol{f}(k, \boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\theta}) \tag{4.38}$$

The sampling time Δt is selected to be 15 seconds and the integration time step Δt_{int} is equal to the sampling time. Lund found that shorter step lengths and the use of more complex solvers did not improve the accuracy and increased the computation time [21].

4.3.1.2 Parameter and State Estimation

While parameter estimation is important in the accuracy of the model, offline parameter estimation requires the use of real plant data and online parameter estimation requires a real plant. Since neither of these two were available, parameter estimation could not be performed. This work uses a hypothetical lab-scale reactor so there is little benefit in performing online parameter estimation because these parameters will have to be changed if the control structure developed in this thesis were to be implemented on a real process. Previously, Lund performed offline parameter estimation using data from a manufacturer of EPS but there was insufficient data to conclude that the identified parameters improved the model accuracy [21]. Therefore, the original parameters selected by Lund are used here and parameter estimation is left for future work.

State estimation is not considered in this thesis. Instead, the estimated states were set equal to the calculated states by setting the Kalman gain to a constant matrix of zeros. This was not done because the measured states in this process are all values that could be directly measured. In addition, measurement and process noise were neglected so there was no need to apply a correction to the predicted states since there was likely to be little mismatch between the predicted states and the measured states.

4.4 Offline Optimization Problem Development

Offline optimization is performed to identify the optimal monomer to initiator ratio, starting reactor temperature, and the reactor temperature profile throughout the batch. This stage is the first step in the development of the advanced control structure since the results from the offline optimization will be used in the construction of the two-layers.

4.4.1 Variable Selection

The states are the balanced variables in the equations: the monomer, initiator, and pentane quantities, the moments for dead chains, and the reactor and jacket temperatures. For the purpose of optimization, the batch time t_f is included as a state variable as well.

The input variables are the molar flow rates of the monomer, initiator, and pentane into the reactor; the cooling water mass flow rate and cooling water inlet temperatures are also inputs. An extra input variable of the temperature derivative dT_R is added to ensure the possibility of perfect temperature control. Additionally the initial temperature T_0 is included as an input to find the optimal starting reactor temperature. This is a decision variable in the first sample of the prediction horizon. After the first sample, it is replaced by the reactor temperature derivative dT_R for the remaining samples in the prediction horizon. To optimize the initiator amount a different approach is used where an additional input variable $n_{I,0}$ is added along with a state variable that keeps track of the amount of initiator fed to the reactor. The change rate is calculated using the following equation [21]:

$$\frac{dn_I}{dt} = R_I V + \frac{(n_{I,0} - n_{I,fed})}{\tau_I} \tag{4.39}$$

If τ_I is selected small enough, the system will behave as if the initiator was initially loaded into the reactor; here the time constant was set to 60 seconds.

Since the reactor system in this work is a hypothetical semi-batch polystyrene reactor, the selected measurements are the essential ones. Here, the reactor and jacket temperatures, plus the weight and number average molecular weights are assumed to be measurable quantities. The outputs include the moments of living chains, the average molecular weights, number average chain length, the conversion, and the polydispersity index. These variables are selected since they are calculated from the states or inputs; in addition they provide useful information about the reaction progress or the product.

The reactor temperature derivative was chosen for the manipulated variable so perfect temperature control could be achieved; this allows for a smoother temperature trajectory in comparison to direct manipulation of the temperature since a new temperature is only determined at each sampling point.

The choice of state, input, measurement and output variables for offline optimization are summarized in their vector form below.

$$\boldsymbol{x} = \begin{bmatrix} n_M \\ n_I \\ n_{C5} \\ \mu_0 \\ \mu_1 \\ \mu_2 \\ T_R \\ T_J \\ t_f \end{bmatrix} \qquad \boldsymbol{u} = \begin{bmatrix} \hat{n}_M \\ \hat{n}_I \\ \hat{n}_{C5} \\ \hat{m}_c \\ dT_R \\ T_{J,i} \end{bmatrix} \qquad \boldsymbol{y} = \begin{bmatrix} T_{R,p} \\ T_{J,p} \\ \overline{M}_{n,p} \\ \overline{M}_{w,p} \end{bmatrix} \qquad \boldsymbol{z} = \begin{bmatrix} \lambda_0 \\ \lambda_1 \\ \lambda_2 \\ \overline{M}_n \\ \overline{M}_w \\ \bar{\text{n}} \\ X \\ \text{PI} \end{bmatrix} \tag{4.40}$$

where $\overline{M}_{n,p}$ is the measured number average molecular weight of the polymer and that $\overline{M}_n$ is the model predicted value; the same holds for the weight average molecular weights.

4.4.2 Problem Formulation

The offline optimization is performed with the aim of minimizing the batch time while producing the desired product quality subject to other process constraints. The decision variables are selected to be the initial ratio between the loaded monomer and initiator plus the dynamic temperature trajectory. The problem formulation is given by:

$$\min_{T_{R,0}, dT_R, n_{I,0}, \epsilon} \quad q(\overline{M}_{n,f} - \overline{M}_{n,d})^2 + r_1\epsilon + s_{dT_R}\Delta dT_R{}^2 \tag{4.41a}$$

$$\text{s.t.} \quad \dot{\boldsymbol{x}}(t) = \boldsymbol{f}(t, \boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{\theta}), \tag{4.41b}$$

$$\boldsymbol{y}(t) = \boldsymbol{g}(t, \boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{\theta}), \tag{4.41c}$$

$$\boldsymbol{z}(t) = \boldsymbol{h}(t, \boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{\theta}), \tag{4.41d}$$

$$t_{f,min} - \epsilon \leq t_f \leq t_{f,max} + \epsilon \quad , \tag{4.41e}$$

$$X \geq X_d \quad , \tag{4.41f}$$

$$\hat{m}_{c,min} \leq \hat{m}_c \leq \hat{m}_{c,max} \quad , \tag{4.41g}$$

$$T_{J,i,min} \leq T_{J,i} \leq T_{J,i,max} \ , \tag{4.41h}$$

$$T_{R,min} \leq T_R \leq T_{R,max} \quad , \tag{4.41i}$$

$$dT_{R,min} \leq dT_R \leq dT_{R,max}, \tag{4.41j}$$

$$n_{I,0,min} \leq n_{I,0} \leq n_{I,0,max} \ , \tag{4.41k}$$

$$\overline{M}_{n,min} \leq \overline{M}_n \leq \overline{M}_{n,max} \tag{4.41l}$$

The objective function consists of a quadratic weighting on the deviation between the final number average molecular weight $\overline{M}_{n,f}$ and the desired value $\overline{M}_{n,d}$, the linearly weighted slack variable on the batch time constraint, and a quadratic term

penalizing the change rate of the temperature derivative. A quadratic weighting is chosen for the average molecular weight deviation so that a small deviation from the desired value is acceptable but a large deviation is highly penalized. A special solution is required for the batch time term as explained by Lund [21]. Because of the format of the framework used in this thesis, it is easiest to constrain the batch time to a small region close to 0 and add a slack variable to avoid infeasibility arising from increasing the batch time. The slack variable is linearly weighted meaning that large time usages are strongly penalized. The final objective function term is added to ensure a smooth temperature profile. The weights are selected by trial and error with the selected weights displayed in Table 4.2; these values used in the offline optimization were originally selected by Lund [21]. These weights are scaled by the specified span of the input move or output; the *span* is the range of values that can be had by the input move or output. This is done to normalize the output and input move optimization weights since the spans for different variables can be substantially different. The span is selected based on physical or safety limitations present in the system; for example, the span on the reactor temperature derivative is 0.001.

Table 4.2: Normalized weights in the offline optimization of the monomer to initiator ratio and reactor temperature profile [21].

Weight	Value
q	450.0
r_{1,T_R}	0.01
s_{dT_R}	1.0

Weights s are placed on $T_{R,0}$ and $n_{I,0}$ but they are negligible so as to not penalize the calculation; in addition, both variables are parameterized so that they only appear in the objective function for the first calculation. Therefore, these terms are not included in the overall problem formulation.

Constraints 4.41b-4.41d represent the process model [21]. The constraint on the batch time is a soft constraint plus a slack variable, as shown in Constraint 4.41e. Constraint 4.41f on the conversion guarantees that the desired conversion is achieved. Constraints 4.41g-4.41i are safety constraints. The rate of change of the temperature is bounded in Constraint 4.41j to ensure that physical limitations on temperature change are taken into account [21]. The amount of initiator is limited to a fixed range to decrease the feasible region, making the problem easier to solve; this is done in Constraint 4.41k. Finally, a constraint is placed on the number average molecular weight in Constraint 4.41l to constrict the potential final number average molecular weight of the produced polymer. The input and output constraint values are shown in Table 4.3.

Table 4.3: Offline optimization input and output constraints.

Description	Symbol	Value	Unit
Desired number average molecular weight	$\overline{M}_{n,d}$	100.0	$\mathrm{kg\,mol^{-1}}$
Minimum number average molecular weight	$\overline{M}_{n,min}$	60.0	$\mathrm{kg\,mol^{-1}}$
Maximum number average molecular weight	$\overline{M}_{n,max}$	150.0	$\mathrm{kg\,mol^{-1}}$
Lower bound on batch time	$t_{f,min}$	0	min
Upper bound on batch time	$t_{f,max}$	0.1	min
Desired conversion	X_d	80	%
Lower bound on reactor temperature	$T_{R,min}$	50	°C
Upper bound on reactor temperature	$T_{R,max}$	140	°C
Lower bound on cooling water temperature	$T_{J,i,min}$	10	°C
Upper bound on cooling water temperature	$T_{J,i,max}$	50	°C
Lower bound on temperature derivative	$dT_{R,min}$	-0.5	$\mathrm{K\,s^{-1}}$
Upper bound on temperature derivative	$dT_{R,max}$	0.5	$\mathrm{K\,s^{-1}}$
Lower bound on total initiator loaded	$n_{I,0,min}$	0	mol
Upper bound on total initiator loaded	$n_{I,0,max}$	1000	mol

4.4.2.1 Prediction Horizon and Input Blocking

The prediction horizon is chosen to be 240 minutes so it is long enough to capture the end of the batch. The sampling interval is set to 15 seconds since it is expected one simulation step should solve within this time and so that it solves sufficiently fast for use online; this translates into having 960 samples for a full simulation. The prediction horizon is divided into 185 blocks for the temperature derivative input variable. This results in 186 decision variables to calculate for the temperature profile since the calculations start at time zero, which is a considerable decrease compared to the 961 variables without input blocking. The blocking structure is constructed so that it is finer at the beginning and decreases throughout the prediction horizon. This means that the temperature can change more frequently at the beginning of the horizon. A new temperature derivative is calculated every sample for the first 50 samples, every fifth sample up to sample 725, and then a single block is used for the rest of the prediction horizon. At this point, the batch is expected to be finished so the temperature derivative should no longer need to change.

4.5 Solving the Offline Optimization Problem

For offline optimization, one sample step is simulated using CENIT and RealSim to determine the predicted states and inputs for the entire batch. This is why the prediction horizon is selected to be long enough to ensure that the batch finishes within the simulated time. Since it is desired for the optimal solution to be determined for one step of the SQP algorithm, complete convergence is needed; therefore, the number of SQP iterations is set to 20. Because the two goals of the optimization problem pose a trade-off problem, an acceptable deviation from the product quality at the end of the batch is defined to be ± 1.0 of the desired final number average molecular weight of 100 $\mathrm{kg\,mol^{-1}}$.

To replicate real operation where the product is removed once a set conversion is reached, the model is set to stop calculating all time derivatives once 80% conversion is reached [21]. The average molecular weight deviation and the batch time should be weighted at the exact point the batch is finished; however, this point is unknown *a priori*. The batch can reach 80% conversion at any point, even between two samples, so the derivatives are set to zero from the next sample regardless of where the exact 80% conversion is reached. This logic is coded into the model, taking into account that CENIT makes perturbations in the input variables to determine the sensitivity in the output variables. If the perturbations are too small to move the 80% mark before the current sample or after the next sample, the sensitivity of the batch time variable will be zero for all or some of the input variables [21]. Linear interpolation can be used to avoid this issue by setting the derivatives to zero at the exact point where 80% conversion is reached. Figure 4.3 illustrates how the interpolated results compare to the results without interpolation.

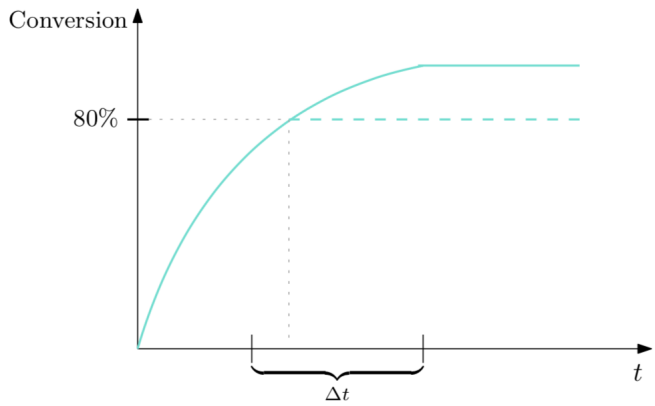


Figure 4.3: Conversion as a function of time with interpolation (dashed) and without (solid) [21].

This is implemented using relational and logical operators to determine if the conversion will reach 80% between the current sample and the next [21]. If this occurs, the derivatives are scaled by a factor $\frac{\Delta\tau}{\Delta t_{int}}$. $\Delta\tau$ is the time from the current sample to the time where 80% conversion is reached and is calculated using [21]:

$$\Delta\tau = \frac{(0.8 - X_k)\Delta t_{int}}{X_{k+1} - X_k} \tag{4.42}$$

where X_k and X_{k+1} represent the conversion at the current and the next sample, respectively. This is done for numerical reasons because when the initiator is added all at once the system becomes very stiff.

Perfect temperature control is assumed here. The reactor temperature derivative is therefore equal to the calculated optimal reactor temperature derivative. The required cooling demand $Q_{J,d}$ is then calculated using Equation 2.12.

The desired final number average molecular weight $\overline{M}_n$ is 100.0 $\mathrm{kg\,mol^{-1}}$ and the batch is terminated once the conversion reaches 80%. An initial monomer amount of 108290 mol or 11344 kg is assumed; this value is determined based on the reactor parameters, which are described in further detail in Appendix B.

4.5.1 Results

The optimal monomer to initiator ratio is found to be 667.25, which corresponds to 163.2 mol or 44.14 kg of initiator. Figure 4.4 shows the initiator consumption throughout the batch, which has a profile similar to that in Lund's work.

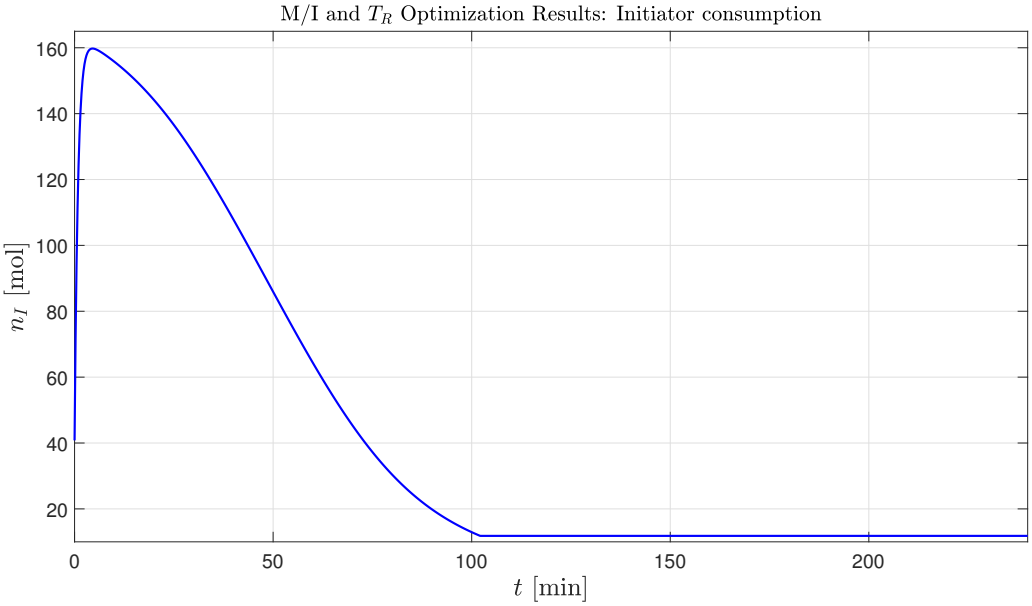


Figure 4.4: Initiator consumption rate for the optimization of the monomer to initiator ratio and temperature profile.

The optimal ratio is larger than the one identified by Lund, meaning that less initiator is now needed; the decrease in required initiator mass is likely due to the model changes, particularly the removal of the constraint on the heat demand. The initiator amount in the reactor settles to 11.81 mol so 93.3% of the initiator is consumed. Given that initiator is expensive, it is important that little remains by the end of the batch to limit the waste. It should be stated that this model does not account for any initiator lost to the continuous phase, which would not be the case for a real process [21].

The optimal initial reactor temperature $T_{R,0}$ is calculated to be 120 °C, which is approximately the same as the starting temperature found by Lund. Figure 4.5 shows the optimal temperature profile, which increases throughout the batch until the upper bound is reached at 91.25 minutes. Therefore, the reactor temperature constraint is active for the last 11 min of the process. The optimized temperature trajectory suggests that it is best to start the reactor at a lower temperature and increase the reactor temperature to the maximum until the batch is terminated. This temperature profile is similar to the one found by Lund [21]. In addition, a similar reactor temperature response for a batch reactor of polymer grafting reactions has been found in another work, where the reactor temperature increases regularly to achieve the desired monomer conversion rate and grafting efficiency [4].

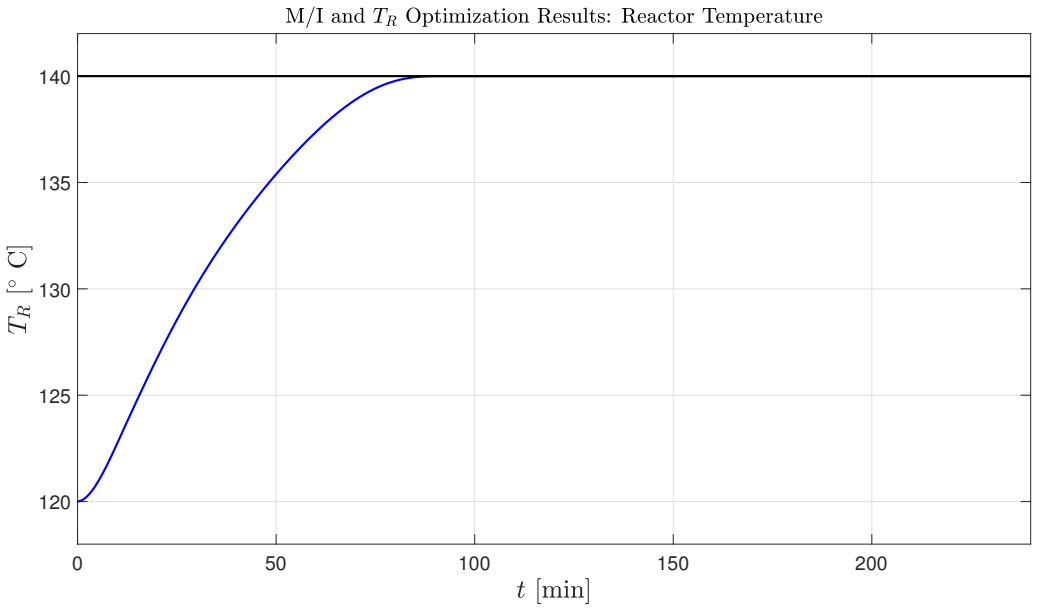


Figure 4.5: Reactor temperature profiles for the optimization of the monomer to initiator ratio and temperature profile.

The jacket temperature is initialized at 60 °C and from there it increases until it

matches the reactor temperature, as illustrated in Figure 4.6. The jacket temperature stabilizes out to 140 °C towards the end of the batch. Lund's model code did not include an energy balance on the jacket temperature so one is added in this work. Here, perfect temperature control is assumed so the cooling fluid flow rate is not calculated. Consequently, when the initial reactor temperature and the rate of change of the reactor temperature are optimized, the jacket temperature tracks the reactor temperature since the cooling water mass flow rate is zero. Since the jacket temperature is initially lower than the reactor temperature, the temperature gradient is in the direction of the jacket; therefore, heat is transferred to the jacket increasing the temperature. This would not be the case for a real reactor where the cooling water flow rate is used to regulate the reactor temperature; therefore, the jacket temperature will likely no longer match this profile when the cooling water is introduced later in this thesis. But for the case where the cooling water flow rate is zero, the jacket behaves as anticipated.

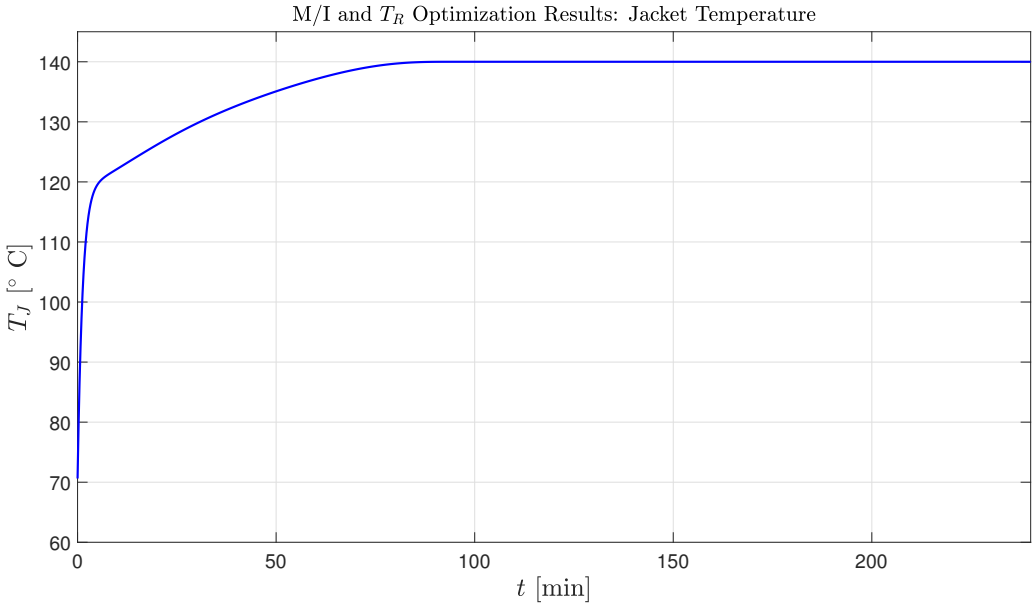


Figure 4.6: Jacket temperature profiles for the optimization of the monomer to initiator ratio and temperature profile.

Figure 4.7 illustrates how the number average molecular weight changes throughout the batch. The final number average molecular weight is 99.71 $\mathrm{kg\,mol^{-1}}$, which is 0.29% away from the desired value and within the allowed range. The corresponding weight average molecular weight is 180.40 $\mathrm{kg\,mol^{-1}}$. This gives a polydispersity index of 1.81, which is within 0.02 units of the value found by Lund; these deviations are within the numerical accuracy of the optimization algorithm so the PI can be considered identical.

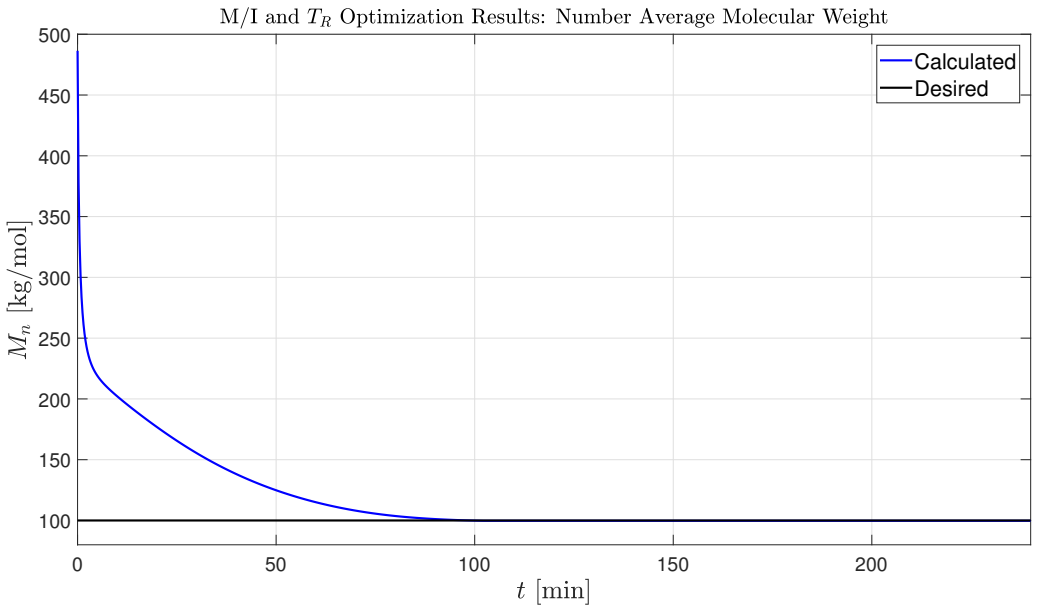


Figure 4.7: Number molecular weight for the optimization of the monomer to initiator ratio and temperature profile.

Figure 4.8 shows the conversion of the monomer to the polymer throughout the batch.

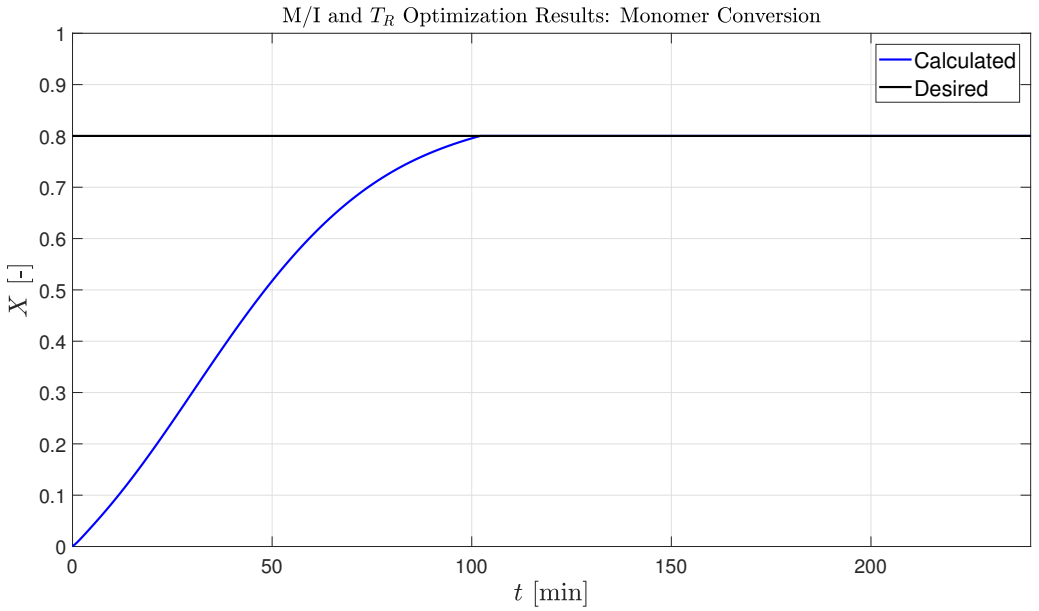


Figure 4.8: Conversion rate for optimization of the monomer to initiator ratio and temperature profile.

The desired conversion of 80% is reached in 102.3 min; this value is 1.6 minutes shorter than the batch time identified using the old model, which is likely due to the removal of the constraint on the cooling demand. The cooling demand places another constraint that must be satisfied and its removal means that the system can use whatever cooling capacity it needs to realize the optimal reactor trajectory. Therefore, the desired conversion can be reached a little bit faster. The conversion rate is the most rapid at the beginning of the batch when the initiator concentration is the highest. It then slows down toward the end of the batch as the amount of available initiator and monomer for reactions decreases. When there is less initiator available to form radicals, the probability of a new radical chains forming is decreased. This slow down can also be explained by the diffusional limitations; these factors are most likely to play a role in limiting the monomer conversion towards the end of the batch since the density of the reactor mass has increased meaning that transport is diffusion controlled.

The optimized monomer to initiator ratio and the optimal temperature profile found here are utilized later in this thesis to construct the NMPC and DRTO layers. While the initiator amount is used in both layers, the temperature profile is only used in the construction of the NMPC layer. The offline optimized reactor temperature trajectory is needed because the NMPC, which must be designed first, requires a reference profile. Once the DRTO layer is completed, this temperature trajectory will be continually re-optimized using measurements from the process; this updated trajectory will then be passed to the NMPC layer as setpoints. The development of these two layers is covered in the next two chapters.

Chapter 5
NMPC Level for Expandable Polystyrene Production

The challenges encountered in the polymer industry are due to the complexities in the kinetics and reaction mechanisms, physical changes, transport effects, non-ideal mixing and conveying, and inherent process nonlinearities. In addition, the variables that affect the product quality are difficult to measure online, making the quality monitoring, and therefore the control, complex [39].

For batch and semi-batch polymerization processes, the design of the control structure consists of two stages. The first stage is the offline design of a control trajectory and the second stage is the implementation and execution of this control trajectory [39]. Once the control trajectory is designed, it needs to be executed as closely as possible; this is can be done utilizing a NMPC algorithm since it has been shown that it performs reasonably for polymerization production [39]. In this chapter, the implementation and execution of the control trajectory found in the offline optimization is realized using a NMPC algorithm. Section 5.1 introduces the variable selection for the NMPC layer. The resulting problem formulation, along with the selected weights, is stated here; the prediction horizon and input blocking are then chosen.

The NMPC level is developed using two main steps: first the reactor temperature is controlled to a constant setpoint, and then the reactor temperature is controlled to an optimized reference temperature profile. The constant reactor temperature setpoint results are used to exam the effect the cooling water temperature has on the cooling water demands. A setpoint change in the reactor temperature is also conducted to test the selected controller tunings. The layer is then altered to follow a reference trajectory for the reactor temperature; this trajectory is the one found in the offline optimization stage. The results for both development stages are presented in Section 5.2.

5.1 NMPC Problem Development

The aim of the NMPC layer is to take the optimal reactor temperature trajectory determined by the DRTO layer and follow the optimal path as closely as possible.

The NMPC level will manipulate the cooling water flow rate to realize this trajectory. The selection of the states, inputs, output and measurement variables should be done with this goal in mind. The problem formulation also considers the aim of this level when choosing the objective function and the constraints. The weights are picked carefully such that certain states and inputs are prioritized over others.

5.1.1 Variable Selection

The states are the balanced variables in the model equations: the monomer, initiator, and pentane amounts, the moments of the dead chains, plus the reactor and jacket temperatures. The mass flow rate of the cooling fluid $\hat{m}_c$ is selected as the decision variable; this also serves as the manipulated variable to the desired controlled variable in this level, the reactor temperature. The reactor temperature needs to be controlled in the NMPC level because fast response times are required so that the reactor is kept at the desired temperature; this cannot be achieved in the DRTO level, which typically operates on a longer time scale.

The measurement variables are the same four selected in the offline optimization stage. The cooling water flow rate should also be included as a measurement variable when this control structure is implemented in a real process but for now during the development stage it is excluded. The choice of output variables are the living chain moments, average molecular weights, and other variables that are calculated from the state and input values that provide information about the semi-batch progress.

The choice of state, input, measurement and output variables are summarized in the vectors below.

$$\boldsymbol{x} = \begin{bmatrix} n_M \\ n_I \\ n_{C5} \\ \mu_0 \\ \mu_1 \\ \mu_2 \\ T_R \\ T_J \end{bmatrix} \qquad \boldsymbol{u} = \begin{bmatrix} \hat{m}_c \\ dT_R \end{bmatrix} \qquad \boldsymbol{y} = \begin{bmatrix} T_{R,p} \\ T_{J,p} \\ \overline{M}_{n,p} \\ \overline{M}_{w,p} \end{bmatrix} \qquad \boldsymbol{z} = \begin{bmatrix} \lambda_0 \\ \lambda_1 \\ \lambda_2 \\ \overline{M}_n \\ \overline{M}_w \\ \bar{\text{n}} \\ X \\ \text{PI} \end{bmatrix} \tag{5.1}$$

5.1.2 Problem Formulation

The chosen formulation aims to optimize the cooling water flow rate to maintain the desired temperature derivative and minimizing the amount of wear and tear on the cooling water flow rate valve. Here, $\hat{m}_c$ is the mass flow rate of the cooling water and is the only decision variable in this level. The problem formulation is chosen with the knowledge that the upper level will be responsible for minimizing the batch time and ensuring that the desired quality is achieved.

$$\min_{\hat{m}_c,\epsilon} \quad q(T_R - T_{R,d})^2 + r_1\epsilon + s_{\hat{m}_c}\Delta\hat{m}_c{}^2 \tag{5.2a}$$

$$\text{s.t.} \quad \dot{\boldsymbol{x}}(t) = \boldsymbol{f}(t,\boldsymbol{x}(t),\boldsymbol{u}(t),\boldsymbol{\theta}), \tag{5.2b}$$

$$\boldsymbol{y}(t) = \boldsymbol{g}(t,\boldsymbol{x}(t),\boldsymbol{u}(t),\boldsymbol{\theta}), \tag{5.2c}$$

$$\boldsymbol{z}(t) = \boldsymbol{h}(t,\boldsymbol{x}(t),\boldsymbol{u}(t),\boldsymbol{\theta}), \tag{5.2d}$$

$$T_{R,min} - \epsilon \leq T_R \leq T_{R,max} + \epsilon, \tag{5.2e}$$

$$\hat{m}_{c,min} \leq \hat{m}_c \leq \hat{m}_{c,max} \quad , \tag{5.2f}$$

$$T_{J,i,min} \leq T_{J,i} \leq T_{J,i,max} \tag{5.2g}$$

The objective function is composed of the quadratic deviation between the actual reactor temperature T_R and the desired reactor temperature $T_{R,d}$, the linearly weighted slack variable on the reactor temperature, and a quadratic term penalizing changes in the cooling water mass flow rate. Quadratic weighting is selected for the deviation in the reactor temperature because small deviations from the desired value are acceptable but large deviations should be highly penalized. The reactor temperature should follow a set trajectory but it is possible that under certain circumstances, there may be large deviations from this trajectory so slack variable ϵ is added to avoid infeasibility arising due to constraint violations. This slack variable is weighted with a linear weight r_1, which heavily penalizes large violations. The final term in the objective function is the penalty on the change in the cooling water mass flow rate between two samples; it heavily penalizes large changes in the mass flow rate so there are smooth changes in the mass flow rate. This limits the amount of wear and tear on the valve that is used to control the cooling water flow rate. In addition, weighting the change of the cooling water flow rate makes the controller less aggressive, thereby preventing oscillations in the process. The weights are found using trial and error; the values are summarized in Table 5.1. As is the case for the offline optimization, these weights are normalized by the corresponding spans.

Table 5.1: Normalized weights in the optimization of the mass flow rate of cooling fluid.

Weight	Value
q	30
r_{1,T_R}	0.5
$s_{\hat{m}_c}$	10

Once again, the first three constraints given by Constraints 5.2b-5.2d represent the process model. Constraint 5.2e states that the reactor temperature must remain within a specified range for both safety reasons and to achieve the required product

quality. In addition, the cooling water inlet temperature is limited to a certain temperature range since a minimum amount of cooling capacity is required in order to achieve the desired results; this is shown in Constraint 5.2g. The cooling fluid flow rate should also be within a maximum and a minimum value as stated in Constraint 5.2f; this decreases the feasible set and provides for the possibility that a plantwide control layer may limit the cooling capacity. The upper and lower limits are summarized in Table 5.2.

Table 5.2: Input and output constraints

Description	Symbol	Value	Unit
Lower bound on reactor temperature	$T_{R,min}$	50	°C
Upper bound on reactor temperature	$T_{R,max}$	140	°C
Lower bound on mass flow rate of cooling water	$\hat{m}_{c,min}$	0	$\mathrm{kg\,s^{-1}}$
Upper bound on mass flow rate of cooling water	$\hat{m}_{c,max}$	100	$\mathrm{kg\,s^{-1}}$
Lower bound on cooling water inlet temperature	$T_{J,i,min}$	10	°C
Upper bound on cooling water inlet temperature	$T_{J,i,max}$	50	°C

Note that the upper bound on the mass flow rate of the cooling water is large; this is because the cooling capacity is unknown for a hypothetical system. For a real reactor, the cooling capacity is based on the available water mass and flow rates, which are based on the jacket volume and the size of the pipes used to carry the cooling water to the system. However, for a hypothetical system, these pieces of information are arbitrary or unknown; therefore, it is reasonable to determine the systems required cooling capacity by calculating the optimal flow rate under the assumption of an “infinite” cooling capacity. A more realistic cooling constraint will be introduced later on in the development of the control structure after simulations reveal what the required cooling water flow rate is for this system; from these results, a reasonable cooling capacity constraints for this process will be selected. The lower bound is set to zero for now but a safety margin should be implemented on both ends when an appropriate range is identified.

5.1.2.1 Prediction Horizon and Input Blocking

The prediction horizon is selected to be 15 minutes with sample intervals of 15 s, giving a total of 60 sample points for a full simulation. For open loop unstable systems like this process, the prediction horizon has to be relatively short. This is because if a long prediction horizon is used, the reactor will “explode” in the simulation; this explosion cannot be linearized and therefore causes numerical issues in the controller. Consequently, it is important to select short prediction horizons for open loop unstable processes to avoid these numerical issues.

To reduce the number of decision variables, input blocking is used. The prediction

horizon is divided into 5 blocks for the mass flow rate of the cooling fluid. This means that there are only 6 decision variables to calculate for the optimal mass flow rate of cooling fluid, compared to 61 variables without blocking. At the beginning of the prediction horizon the blocking structure is finer, meaning that the mass flow rate can vary frequently in this region; the resolution then decreases such that the mass flow rate is not allowed to change as much. A new mass flow rate is calculated on samples 0, 2, 8, 12, 20, and 28; a single block is then used for the remainder of the prediction horizon. This input blocking was found by trial and error by running a simulation with different blockings until the quality of the results were found to be similar to the results without input blocking.

5.2 Solving the NMPC Problem

Unlike the offline optimization where CENIT and RealSim are only run for one sample step, the NMPC application requires the optimization algorithm to be run for the full length of the batch. Since the prediction horizon is only 15 minutes long, the predicted results will only be for a fraction of the batch. For the application of NMPC to a system, complete convergence to the optimal solution is not necessary since it is only required that the system moves toward the desired state from sample to sample [21]. Therefore, the SQP algorithm can be limited further than for the offline optimization; a limit of 5 iterations is selected to ensure that the problem solves rapidly enough for real time application.

As was done for the offline optimization, to replicate real operation where the product is removed once the a set conversion is reached, the model is set to stop calculating all time derivatives once 80% conversion is reached. Perfect temperature control is no longer assumed so the reactor temperature is calculated using Equation B.40. This differs from how the calculation is performed in the offline optimization; this change had to be implemented in this level so that the optimal cooling water mass flow rate would be determined. The optimal monomer to initiator ratio and the initial reactor temperature identified in the offline optimization stage are used in the construction of this level.

5.2.1 Results

As previously state, the NMPC layer is first constructed to track a constant setpoint to validate the selected controller tunings; it is then changed to track the reference trajectory. The isothermal operation shows the ability of this layer to maintain a constant reactor temperature. Setpoint changes and different jacket temperatures are simulated to validate the selected weights. Once the weights are approved, the level is then altered to track the optimal reactor temperature profile identified in the offline optimization. Different cooling water temperatures are tested to see the effect this has on the controller's ability to follow the fixed trajectory.

5.2.1.1 Isothermal Operation

A setpoint of 120 °C is used for the reactor temperature. Three different inlet cooling water temperatures between 10-20 °C are simulated to test that the controller can handle seasonal fluctuations that may occur in the cooling water source.

The reactor temperatures for the three different temperatures are nearly indistinguishable, as illustrated in Figure 5.1. All of them have a tiny undershoot, with a cooling water temperature of 10 °C also having a tiny overshoot. The controller is able to quickly achieve the desired setpoint for any cooling water temperature within this range. The simulation is run for longer than the expected batch time (based on the offline optimization results) to show how the controller behaves over this time period. This results show that the controller is able to perfectly maintain the desired setpoint for a significant period of time.

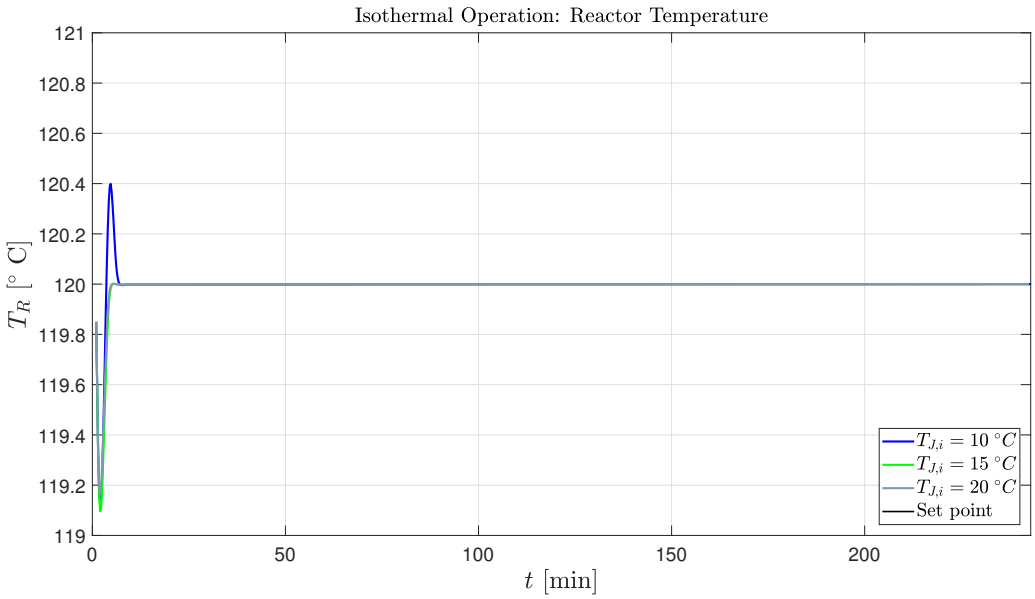


Figure 5.1: Reactor temperature for different cooling water temperatures during isothermal operation.

Figure 5.2 shows that the jacket temperature is unaffected by the different cooling water temperatures, which is explained by the corresponding cooling water flow rates. This profile illustrates how the jacket behaves when a non-zero cooling water flow rate is used The temperature rapidly increases from the starting temperature, producing a peak at the beginning; this peak is caused by the cooling water flow rate increasing slowly. The jacket temperature then increases for the remainder of the batch and ends just under 120 °C.

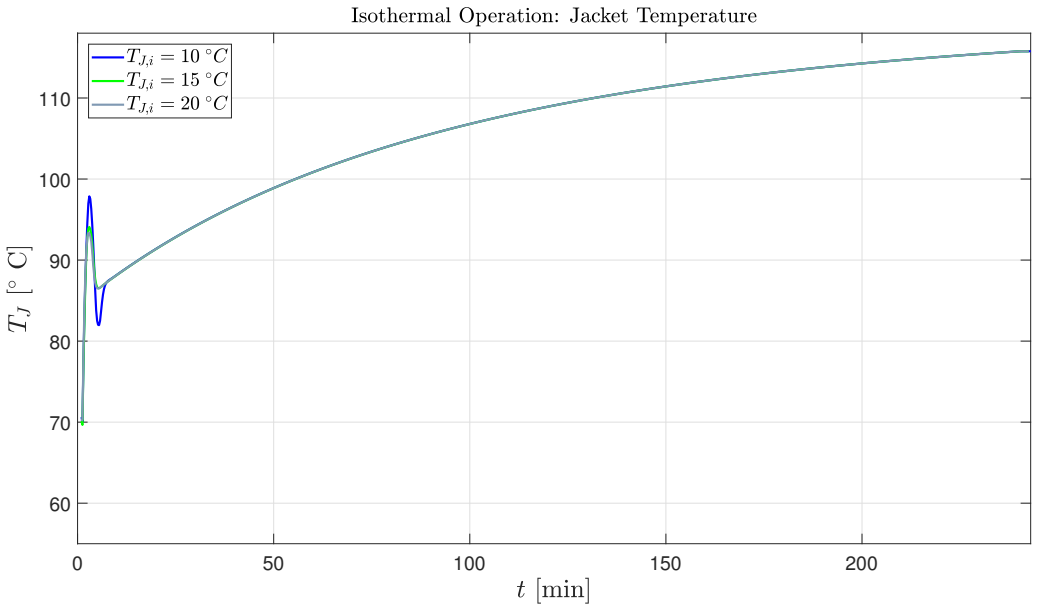


Figure 5.2: Jacket temperature for different cooling water temperatures during isothermal operation.

The cooling water flow rate increased with the cooling water temperature as shown in Figure 5.3.

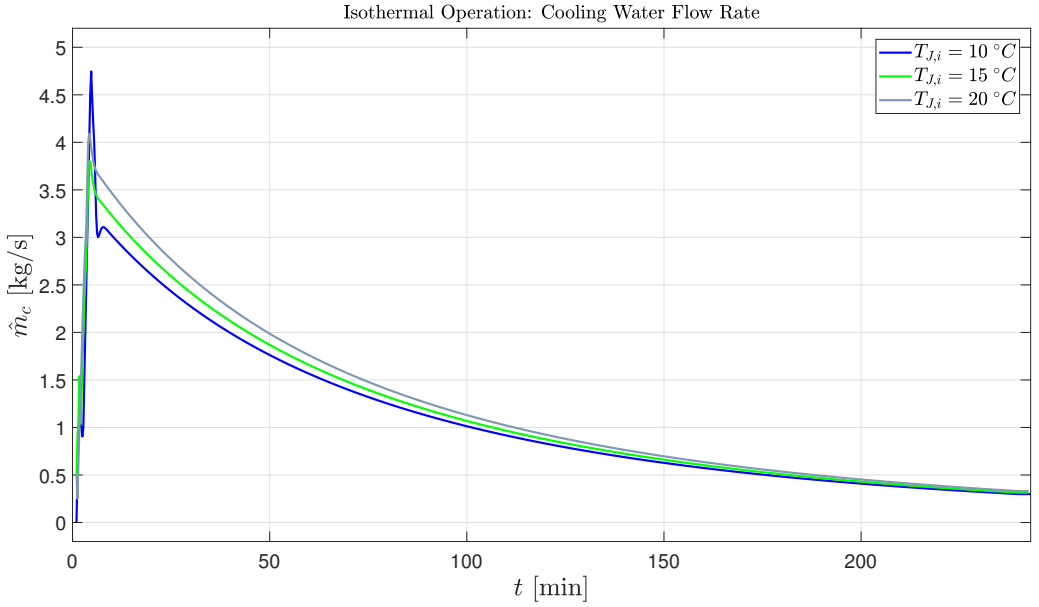


Figure 5.3: Cooling water flow rate for different cooling water temperature during isothermal operation.

Given that more mass is required to remove the same amount of heat from the reactor since the temperature gradient decreases as the cooling water temperature increases, this is expected. More cooling fluid must go through the system in order to maintain a constant reactor temperature. It is interesting to see that initially the coldest cooling water has the highest flow rate; this can be explained by the determined reactor temperature for this temperature. This may illustrate that the optimization algorithm struggles to locate a solution for these operating conditions; this may be indicative of a flat objective function.

Table 5.3 summarizes the results of the isothermal simulations.

Table 5.3: Comparison of different cooling water temperatures affect on polymer properties and batch time for isothermal operation.

$T_{J,i}$ [°C]	$\overline{M}_n$ [kg mol^{-1}]	$\overline{M}_w$ [kg mol^{-1}]	PI [-]	t_f [min]
10	149.61	246.94	1.65	240.3
15	149.64	247.06	1.65	240.4
20	149.64	247.03	1.65	240.4

The final number average molecular weight is far from the desired value in all instances, which is unsurprising because the monomer to initiator ratio used here is for non-isothermal operation so it is not possible to achieve the desired value under these conditions. These simulations are not meant to show that the controller can achieve the desired product but rather the desired setpoint.

Setpoint Changes in the Reactor Temperature:

A step change in the reactor setpoint is done to test the controller tunings. The reactor setpoint is increased from 120 to 140 °C; once a new steady state temperature is reached, the setpoint is then decreased from 140 to 130 °C.

The resulting reactor temperatures are illustrated in Figures 5.4 where the setpoint changes are all made at the same sample number. The temperature profiles show that the higher the cooling water temperature, the slower the response time. For the lowest cooling water temperature, the controller is able to bring the reactor to the new setpoint of 140 °C in 15.75 min; when the cooling water temperature is 20 °C, it takes the controller 17.75 min to reach the new setpoint. In contrast, when the setpoint decreases from 140 °C to 130 °C, the controller is quicker to bring the reactor to its new setpoint, taking 7.25 min and 8 min, respectively. These simulations show that the controller is able to handle setpoint changes both directions for various cooling water temperature.

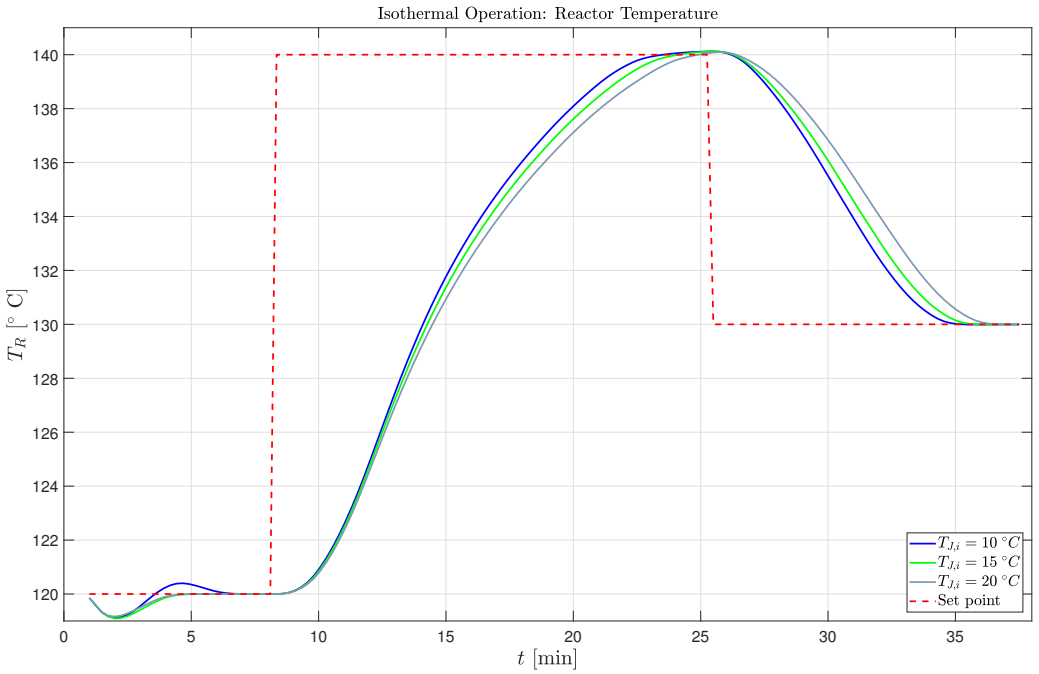


Figure 5.4: Reactor temperature response to reactor temperature setpoint changes during isothermal operation.

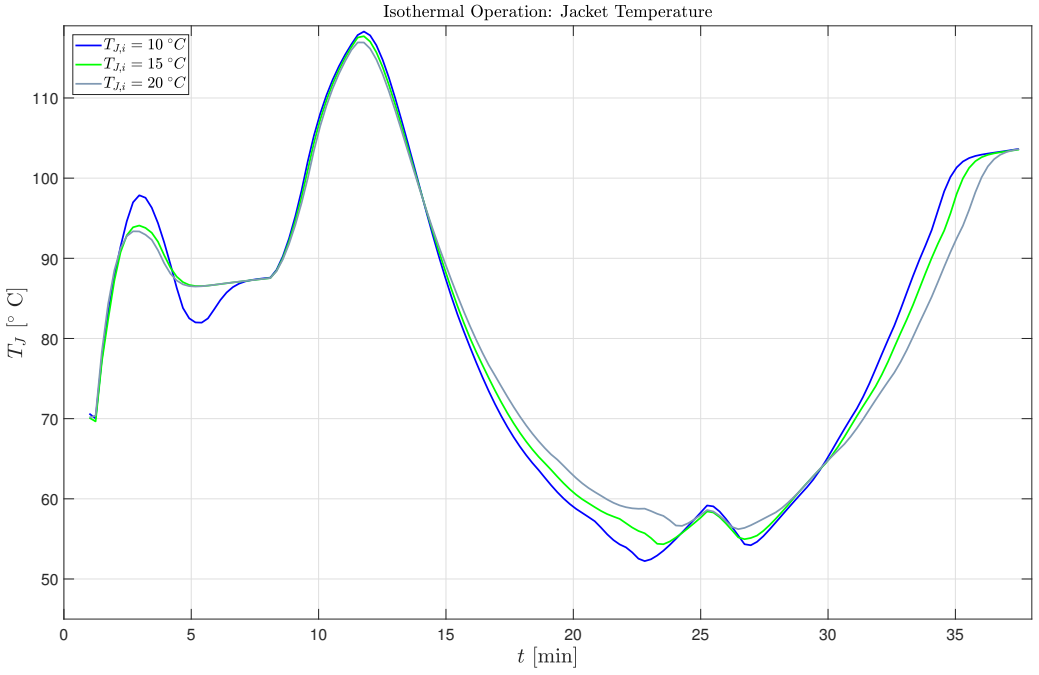


Figure 5.5: Jacket temperature response to reactor temperature setpoint changes during isothermal operation.

Figure 5.5 shows the jacket temperature response to the setpoint changes. When the reactor temperature setpoint is changed to 140 °C, the jacket temperature increases rapidly until it reached 116.2 °C; it then decreases to 50.6 °C as the reactor temperature reaches the new setpoint. The decrease in jacket temperature is caused by the rapid increase in the cooling water flow rate from 0 to around 18 $\mathrm{kg s^{-1}}$. When the setpoint is changed to 130 °C, the jacket temperature increased to 108.3 °C; again, this is due to the cooling water flow rate decreasing.

The cooling water flow rate has a similar profile for all of the different cooling water temperatures as shown in Figure 5.6; the key difference being in the average and maximum cooling fluid flow rate. For a cooling water temperature of 10 °C, the maximum cooling fluid flow rate is 18.9 $\mathrm{kg s^{-1}}$ and the average flow rate is 7.4 $\mathrm{kg s^{-1}}$. In comparison, a cooling water temperature of 20 °C gives a maximum flow rate of 21.9 $\mathrm{kg s^{-1}}$ and an average flow rate of 8.5 $\mathrm{kg s^{-1}}$. This shows that when the cooling water temperature increases by 10 °C, the cooling fluid flow rate increases by a maximum of 4.2 $\mathrm{kg s^{-1}}$. Unsurprisingly, the higher the cooling water temperature, the more cooling fluid required.

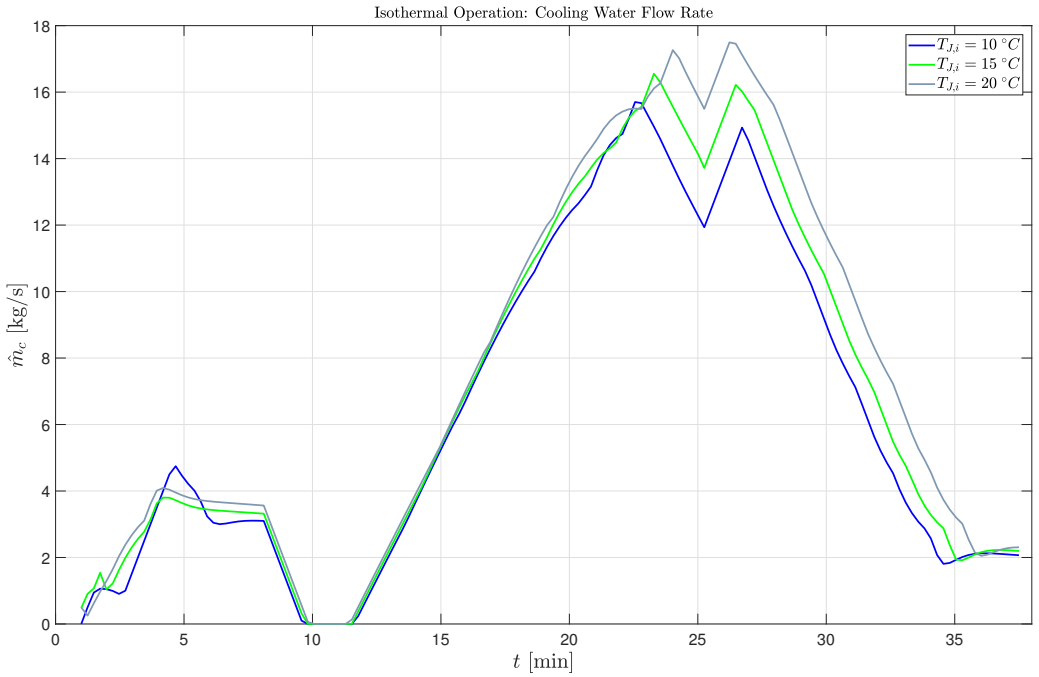


Figure 5.6: Cooling water flow rate response to reactor temperature setpoint changes during isothermal operation.

These results validate the selected tunings by demonstrating that the controller is able to handle disturbances in the cooling water temperature as well as setpoint changes. In addition, the controller's performance is smooth and rapid, which are

both desirable properties.

5.2.1.2 Reactor Temperature Trajectory Simulations

The next step in the construction of the NMPC layer is to track the optimal temperature trajectory found in the offline optimization stage; this is done by feeding the profile to the model as a moving setpoint for the reactor temperature.

Figure 5.7 shows the calculated reactor temperature in solid blue and the optimal trajectory in dashed red. At the beginning of batch, there is a minor deviation between the temperature and the setpoint; otherwise, the controller is able to tightly control the temperature to the setpoint. The reactor begins at 120 °C and dips slightly as the cooling water flow rate increases. The smooth temperature profile demonstrates how well the controller is able to track the setpoint.

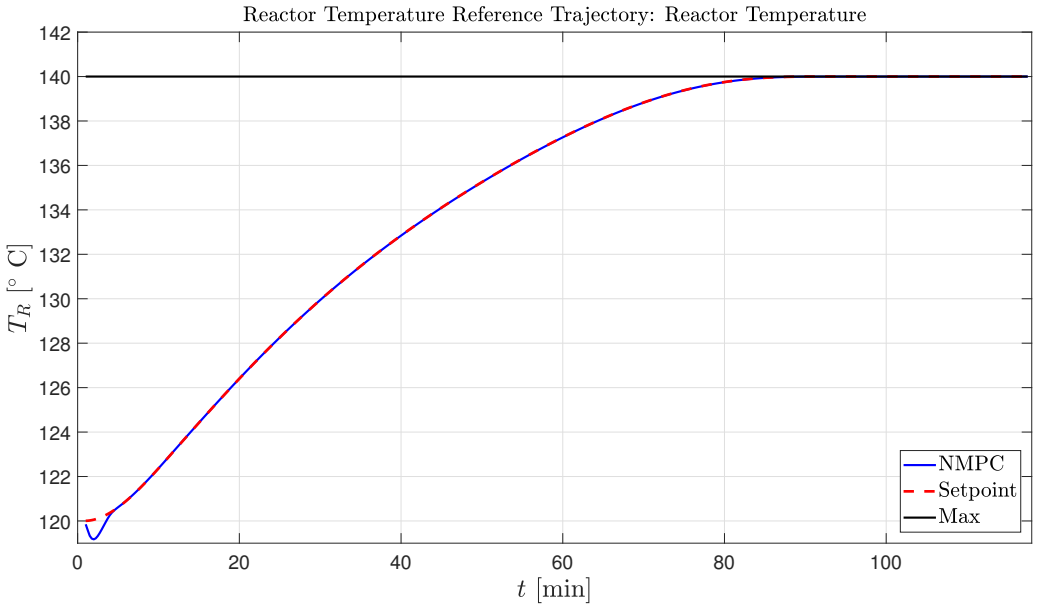


Figure 5.7: Reactor temperature for the optimization of the cooling water flow rate following a fixed reactor temperature trajectory.

The jacket temperature is shown in Figure 5.8 and provides insight into what the actual jacket temperature would look like when this control structure is implemented on a real process. At the beginning of the batch, the peak reflects the slow increase of the cooling water flow rate. Once the cooling water flow rate is sufficient enough to remove heat from the system, the jacket temperature increases at a more gradual rate. The jacket temperature then steadily rises until it reaches 130 °C by the end of the batch.

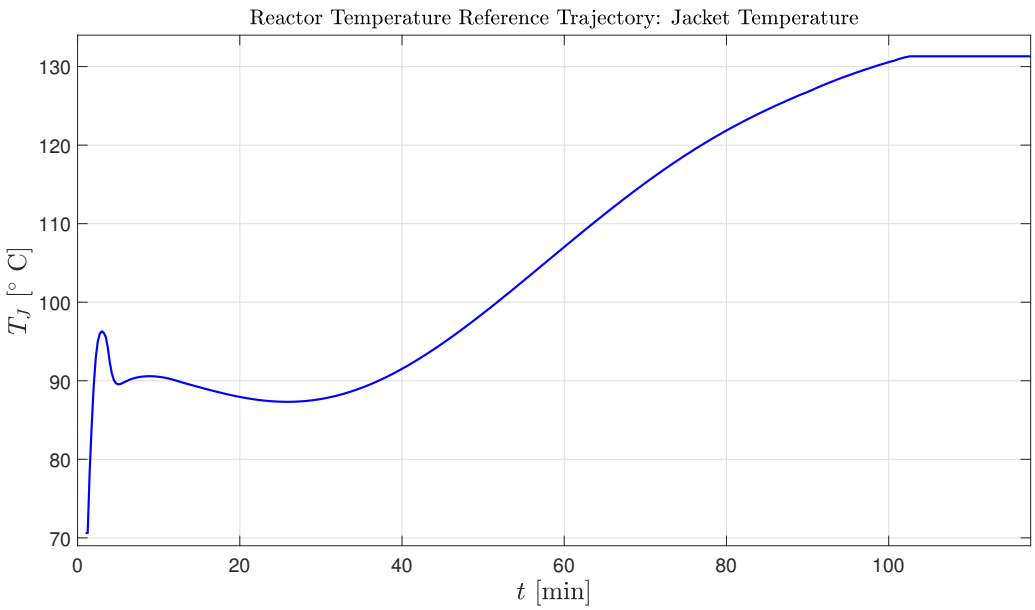


Figure 5.8: Jacket temperature for the optimization of the cooling water flow rate following a fixed reactor temperature trajectory.

The optimized cooling water flow rate is shown in Figure 5.9.

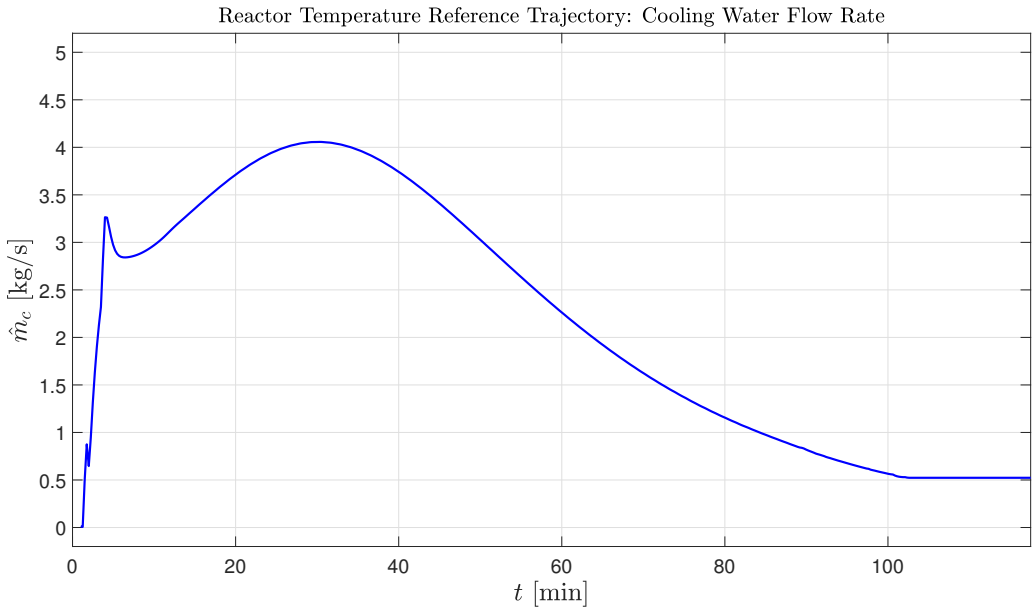


Figure 5.9: Cooling water flow rate for the optimization of the cooling water flow rate following a fixed reactor temperature trajectory.

The controller struggles to find the optimum value at the start of the batch resulting in a spike; however, after this point, the cooling water flow rate has a smooth profile which indicates that the selected controller weights are sufficient. This also means that the valve will not experience excessive wear and tear due to frequent changes in the position. The cooling water flow rate is maximum during the middle of the batch, this implies that this is when the reactions generate the most heat. The highest cooling water flow rate is slightly above 4 $\mathrm{kg s^{-1}}$ suggesting that a reasonable upper bound for this system should be somewhere near this number. However, as simulated the upper limit is never active, which suggests that the cooling capacity is essentially infinite.

The number average molecular weight profile throughout the batch is shown in Figure 5.10. The final number average molecular weight is 99.67 kg/mol; the corresponding weight average molecular weight is 179.05. The number and weight average molecular weights give a polydispersity index of 1.80. All of these values are extremely close to the offline optimization results and the deviations can be justified by the deviation of the reactor temperature at the start of the batch.

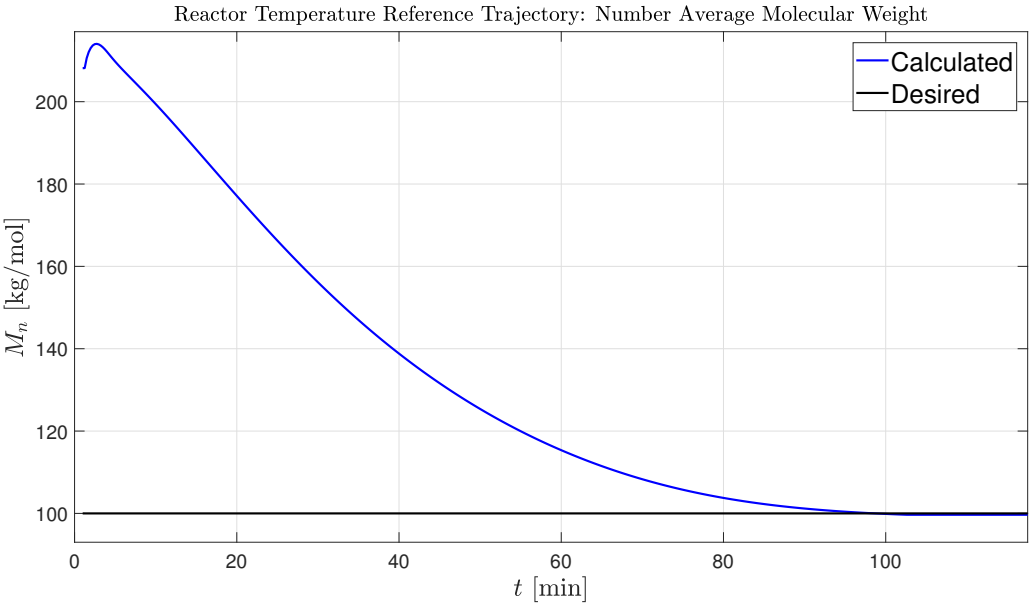


Figure 5.10: Number average molecular weight for the optimization of the cooling water flow rate following a fixed reactor temperature trajectory.

A conversion of 80% is reached at 102.3 min as shown in Figure 5.11, which means that the batch time is identical to the predicted offline optimized batch time. This is expected since the controller is able to track the reactor temperature profile identified in the offline optimization; therefore, they should have identical batch times. This demonstrates that the NMPC controller designed in this chapter works

as desired.

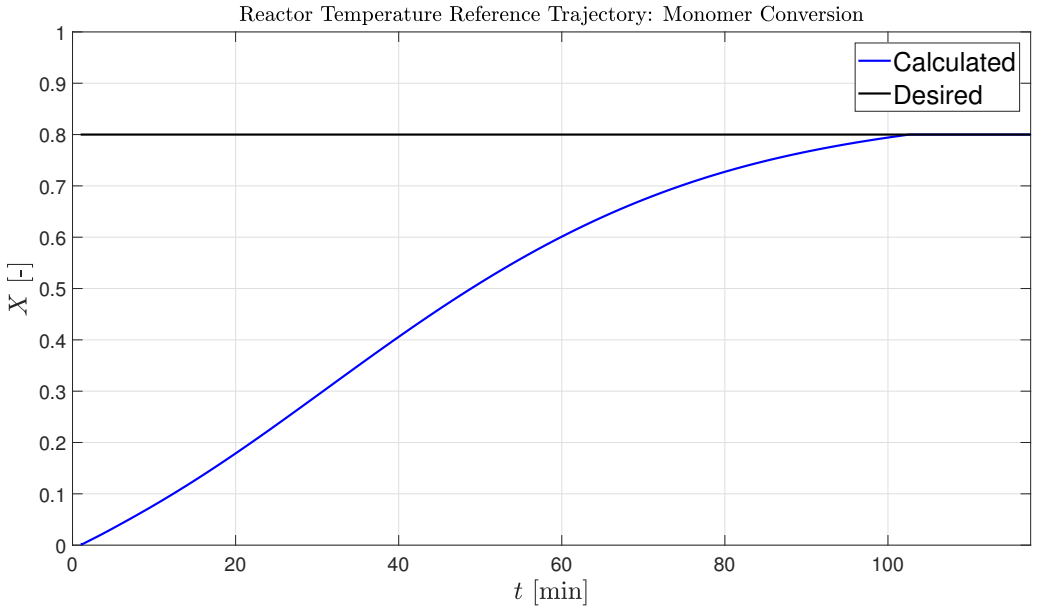


Figure 5.11: Conversion rate for the optimization of the cooling water flow rate following a fixed reactor temperature trajectory.

The initiator consumption is shown in Figure 5.12.

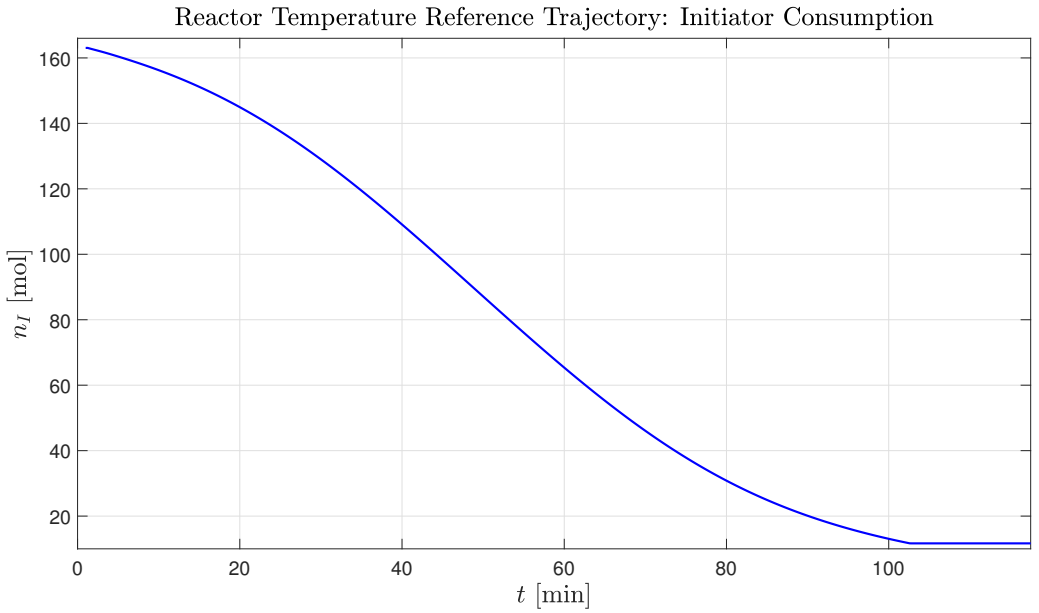


Figure 5.12: Initiator consumption for the optimization of the cooling water flow rate following a fixed reactor temperature trajectory.

At the end of the batch only 15.6 moles of initiator remaining, meaning that 91.1% of the initiator has been used. Ideally, the consumption rate would be higher since initiator is often expensive and any excess initiator is essentially wasted. This is less than what is predicted by the offline optimization suggesting that the deviation from the temperature reference trajectory at the beginning of the batch may be the cause.

These results show that this controller can successfully follow a reference trajectory for the reactor temperature through the optimization of the cooling water flow rate. Tracking the optimized reactor temperature profile gives a polystyrene product quality near the desired value with a shortened batch time. The results imply that the reactor temperature trajectory determined by the offline optimization is either the optimal trajectory for the selected operating conditions or very close to it. However, using a fixed profile can potentially prove problematic if any of the operating conditions are not near the conditions used to produce the profile; this is why online updates of this profile can be beneficial. In the next stage of development, the DRTO level will determine the reactor temperature trajectory for this layer in real time using process measurements to update the trajectory. But before this layer is developed, the NMPC controller is tested for different operating conditions.

Disturbance in Cooling Water Temperature:

To test the controller's response to a disturbance when trying to follow a reference trajectory, the cooling water inlet temperature is varied; temperatures ranging between 10-20 °C are simulated.

Figure 5.13 shows that the three different cooling water temperatures produce reactor temperature profiles that are virtually indistinguishable from one another. The setpoint is again presented as a dashed red line, with the simulated temperature profiles being presented with solid lines. The black line is the maximum allowed reactor temperature. All three cooling water temperatures give a small deviation from the optimal trajectory but quickly recover to match the desired setpoint. These results illustrate that the controller is able to handle different cooling water temperatures and suggest that when the cooling capacity is unlimited, the system can track this profile for any cooling water temperature.

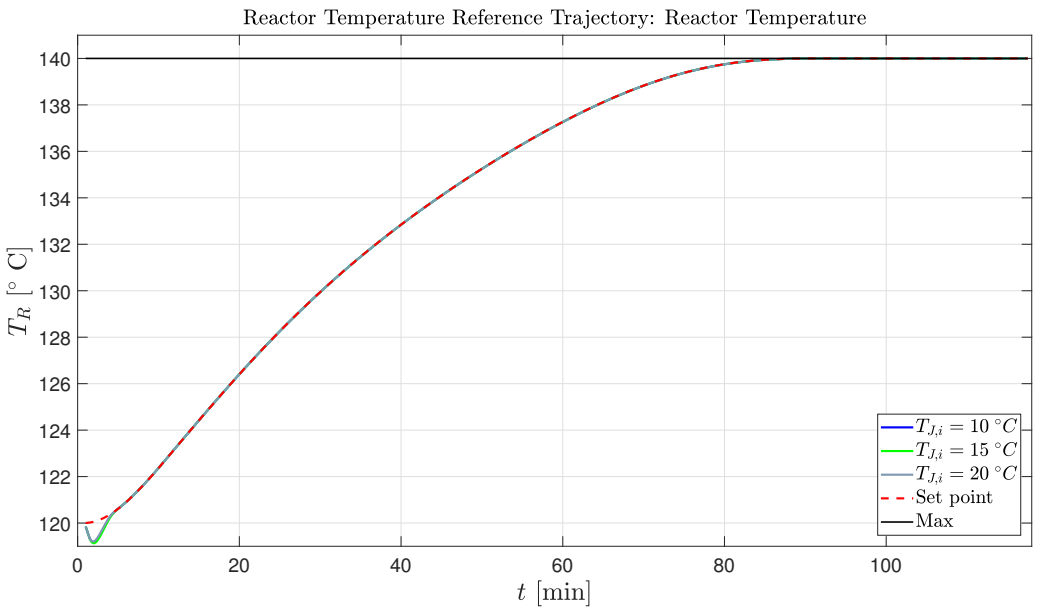


Figure 5.13: Reactor temperature at various cooling water temperatures for the optimization of the cooling water flow rate.

The jacket temperature is shown in Figure 5.14 and shows that the cooling water temperature does not affect the jacket temperature.

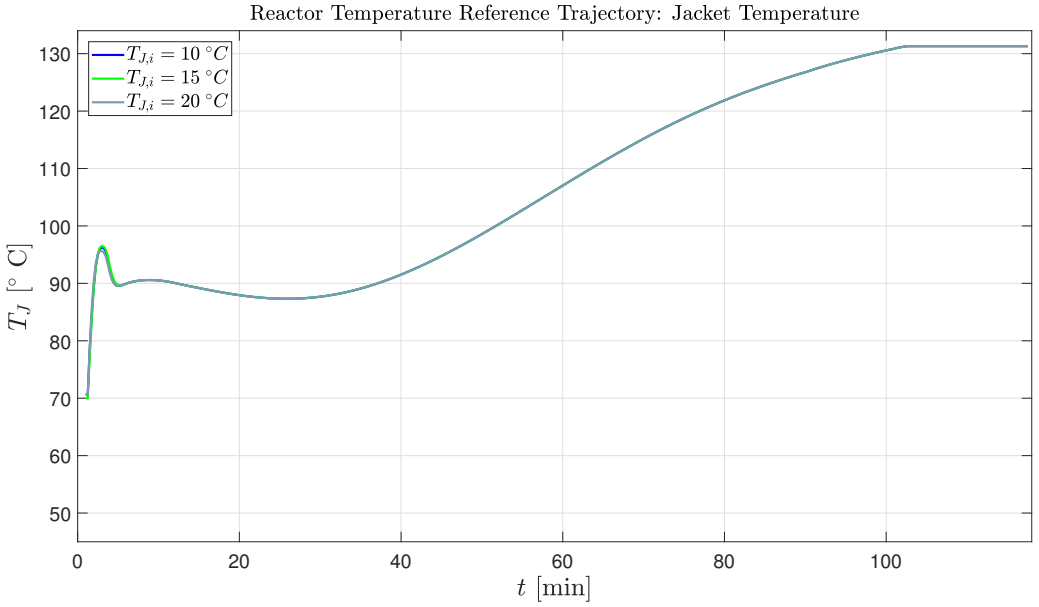


Figure 5.14: Jacket temperature at various cooling water temperatures for the optimization of the cooling water flow rate.

The jacket temperature is the same for all the different cooling water temperatures because the cooling water flow rate is able to increase to compensate for the increased temperature. For non-isothermal operation, the jacket temperature reaches a maximum just under 130 °C; this is about 15 °C degrees higher than for isothermal operation. Given that the reactor temperature is 20 °C higher by the end of the batch for non-isothermal operation, it is unsurprising that the jacket temperature increases to a higher value in comparison to isothermal operation.

Figure 5.15 shows that the cooling water flow rate increases as the cooling water temperature increases. The differences in the cooling water flow rate between the coldest water and the warmest are small so unless the process is operating in a location where cooling water is extremely expensive or limited, this increase is likely insignificant to the overall cost of operation. The ability of the cooling water flow rate to increase explains why the reactor and jacket temperature can remain the same even for different cooling water temperatures. Again in all cases the controller produces a smooth profile for the cooling water flow rate.

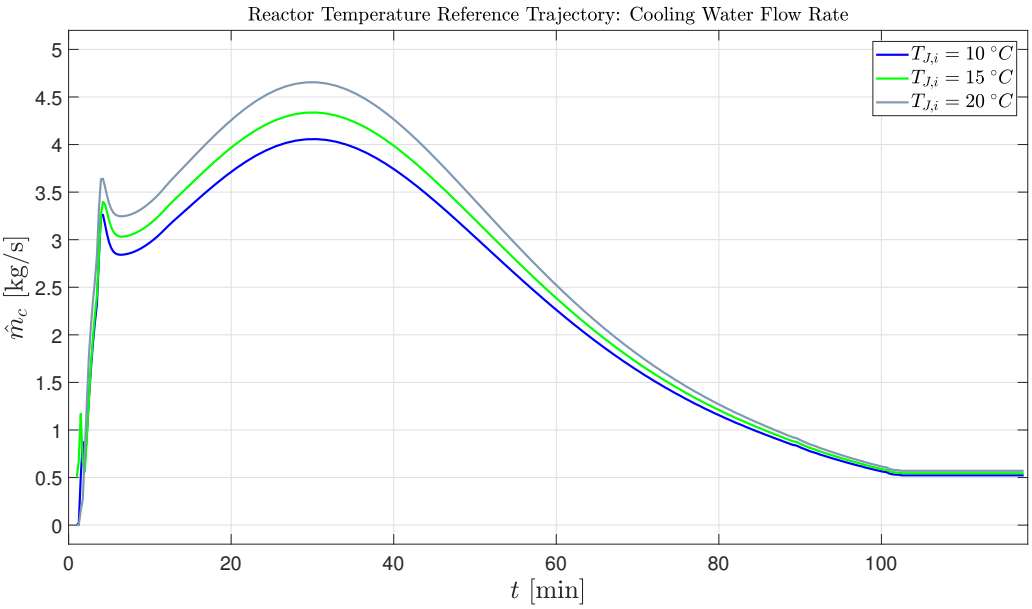


Figure 5.15: Cooling water flow rate at various cooling water temperatures for the optimization of the cooling water flow rate.

Table 5.4 summarizes the results of the three different cooling water temperatures. The batch time, final number and weight average molecular weights were unaffected by the cooling water temperature because the controller is able to maintain the desired reactor temperature profile for all of the tested cooling water temperatures; since the reactor temperature affects the product properties the most, obviously

the final product will be about the same if the reactor profile is the same. This demonstrates that the selected weights and input blocking are sufficient for the simulated disturbances.

Table 5.4: Comparison of different cooling water inlet temperatures effect on polymer properties and batch time for non-isothermal operation.

$T_{J,i}$ [°C]	$\overline{M}_n$ [kg mol^{-1}]	$\overline{M}_w$ [kg mol^{-1}]	PI [-]	t_f [min]
10	99.67	179.05	1.80	102.3
15	99.67	179.06	1.80	102.3
20	99.67	179.05	1.80	102.3

CHAPTER 6

DRTO LEVEL FOR EXPANDABLE POLYSTYRENE PRODUCTION

DRTO levels are commonly used in batch and semi-batch processes to determine the optimal reference trajectories for the states and inputs in real time using plant measurements to update predictions. The objective function is solved at each point subject to the dynamic process model. This method makes use of online measurements to maximize process performance, while still satisfying environmental and operating constraints [4]. In this chapter the development of a DRTO layer for the production of EPS is examined.

A brief introduction to the use of DRTO layers in semi-batch processes in general is covered in Section 6.1 before proceeding to discuss the specific application of DRTO to the production of EPS. The problem construction is outlined Section 6.2, where the variable selection for this layer is stated along with the resulting problem formulation. Finally the chosen prediction horizon and input blocking for the DRTO layer are stated.

Section 6.3 covers how the DRTO optimization problem is solved. Since this layer has to work with the NMPC developed in the previous chapter, the two layers must communicate properly for the desired objectives to be achieved; how this communication is done is explained in this section.

Simulation results using the full control structure are then presented. Finally, the controller is tested by studying the effects that changes in the cooling capacity of the system and the cooling water inlet temperature have on the batch time and the polymer properties.

6.1 Dynamic Real Time Optimization in Semi-Batch Processes

The DRTO layer reoptimizes the reference trajectories throughout the batch at set intervals by using the measured states. The aim of this layer is to optimize such things as feed rates of raw materials, temperature profiles and other operating conditions for the process in real time based on feedback from the process. This

ultimately is for the purpose of [5]:

- Controlling the quality of the final product by predicting throughout the whole batch and complying with constraints on the product quality.
- Maximizing productivity by minimizing batch cycle times.
- Maintaining safe margins in the limits in the cooling capacity, temperature, pressure, etc.

This layer is often based on economic criterion where the batch time is weighted with the cost of materials. Product quality, specifications, safety considerations, and other limitations can be implemented as constraints at this level [5].

On top of the DRTO layer can be another optimization algorithm which performs plantwide optimization. The output of this optimization problem becomes the inputs to each individual control structure to help distribute the available resources between the reactors. As an example, Figure 6.1 shows a possible hierarchical structure for the EPS plant considered in this work.

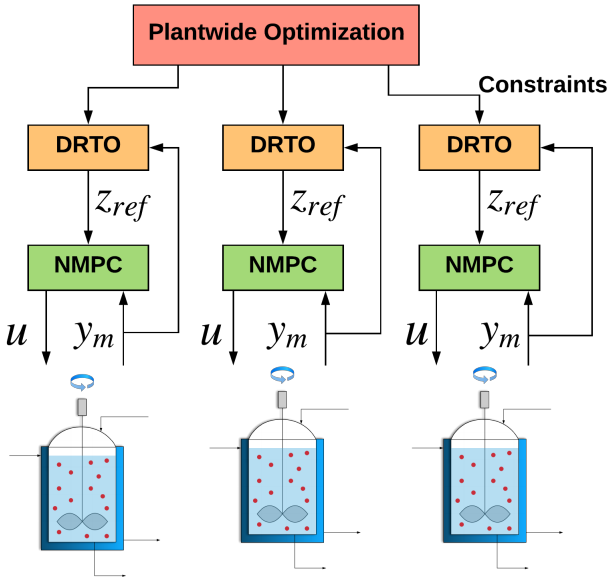


Figure 6.1: Plantwide optimization of an EPS polymerization plant.

On the top level, plantwide optimization occurs where decisions based on market demands and resource availability are made. This layer distributes the resources optimally throughout the plant and is passed to the DRTO layer as constraints.

The DRTO level solves an optimization problem based on these constraints and measurements from the process; the output from this calculation is a set of operating reference trajectories [13]. The use of a dynamic model over a steady state model benefits processes that experience frequent transitions such as those seen in polymerization industries [13]. The setpoints for the NMPC come from the DRTO trajectories; the NMPC layer then aims to control the process close to the optimal paths. In this work, the offline optimization served as a pre-stage of the NMPC and DRTO applications.

6.2 Problem Development

The DRTO layer for the production of EPS is constructed with the goal of reducing the batch time required to reach the desired product quality and conversion. Specifically, this is done by optimizing the reactor temperature derivative under the constraint of the desired product properties and minimizing the batch time. The output from this level will be the input to the NMPC level, thereby providing a new desired reactor temperature each time the DRTO layer is run. To achieve these objectives, the variables and the objective function must be carefully determined. The variable selection is discussed first, followed by the full problem formulation for the DRTO layer.

6.2.1 Variable Selection

Given that the aim of this level is nearly identical to the offline optimization, the selected variables will be similar. The states are the balanced variables: monomer, initiator, and pentane mass, along with the three dead chain moments, and the reactor and jacket temperatures. Additionally, she batch time t_f is included as a state variable. The decision variable in this layer is the derivative of the reactor temperature dT_R.

The molar flow rates of the monomer, initiator, and pentane, plus the cooling fluid mass flow rate and the derivative of the reactor temperature are selected as the input variables. The cooling water inlet temperature is also included. The derivative of the reactor temperature is selected as the manipulated variable for perfect temperature control and optimization. This allows for a smooth temperature trajectory since manipulating the temperature derivative means that a new temperature can only be calculated at each sampling point. The process measurements are the same as before: the reactor and jacket temperatures plus the measured average molecular weights.

The output vector is comprised of variables that are calculated from the states and inputs; this includes the living chain moments, plus both average molecular weights, the conversion, the average number chain length, and the polydispersity index. An additional output variable is added to determine the required cooling water flow

rate based on the reactor energy balance $\hat{m}_{c,r}$; this allows for the optimal cooling water flow rate determined in the NMPC layer to be compared with the flow rate that the DRTO calculates is required to follow the optimal reactor temperature profile. While both values are subject to the same constraint, having the cooling water flow rate as a state means that it softly constrained as opposed to the outputs, which have hard constraints.

The choice of variables is summarized in the vectors below.

$$\boldsymbol{x} = \begin{bmatrix} n_M \\ n_I \\ n_{C5} \\ \mu_0 \\ \mu_1 \\ \mu_2 \\ T_R \\ T_J \end{bmatrix} \qquad \boldsymbol{u} = \begin{bmatrix} \hat{n}_M \\ \hat{n}_I \\ \hat{n}_{C5} \\ dT_R \\ T_{J,i} \end{bmatrix} \qquad \boldsymbol{y} = \begin{bmatrix} T_{R,p} \\ T_{J,p} \\ \overline{M}_{n,p} \\ \overline{M}_{w,p} \end{bmatrix} \qquad \boldsymbol{z} = \begin{bmatrix} \lambda_0 \\ \lambda_1 \\ \lambda_2 \\ \overline{M}_n \\ \overline{M}_w \\ \bar{\mathrm{n}} \\ X \\ PI \\ \hat{m}_{c,r} \end{bmatrix} \tag{6.1}$$

6.2.2 DRTO Problem

To fulfill the aim of this supervisory layer, the objective function must be carefully constructed. At the same time, certain constraints must not be violated. As previously stated, the decision variable is the dynamic temperature trajectory during the batch. The following formulation is selected to prioritize between the different goals and penalize unwanted deviations that are likely to occur.

$$\min_{dT_R, \boldsymbol{\epsilon}} \quad q(\overline{M}_{n,f} - \overline{M}_{n,d})^2 + \boldsymbol{r}_1 \boldsymbol{\epsilon} + s_{dT_R} \Delta dT_R{}^2 \tag{6.2a}$$

$$\text{s.t.} \qquad \dot{\boldsymbol{x}}(t) = \boldsymbol{f}(t, \boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{\theta}), \tag{6.2b}$$

$$\boldsymbol{y}(t) = \boldsymbol{g}(t, \boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{\theta}), \tag{6.2c}$$

$$\boldsymbol{z}(t) = \boldsymbol{h}(t, \boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{\theta}), \tag{6.2d}$$

$$t_{f,min} - \epsilon \leq t_f \leq t_{f,max} + \epsilon \quad , \tag{6.2e}$$

$$T_{R,min} \leq T_R \leq T_{R,max} \quad , \tag{6.2f}$$

$$X \geq X_d \quad , \tag{6.2g}$$

$$dT_{R,min} \leq dT_R \leq dT_{R,max} \quad , \tag{6.2h}$$

$$\hat{m}_{c,min} - \epsilon \leq \hat{m}_{c,r} \leq \hat{m}_{c,max} + \epsilon \quad , \tag{6.2i}$$

$$T_{J,i,min} \leq T_{J,i} \leq T_{J,i,max} \quad , \tag{6.2j}$$

$$\overline{M}_{n,min} \leq \overline{M}_n \leq \overline{M}_{n,max} \tag{6.2k}$$

The objective function consists of the quadratic deviation between the final product number average molecular weight $\overline{M}_{n,f}$ and the desired value $\overline{M}_{n,d}$, the linearly weighted slack variable of the batch time constraint, the linearly weighted slack

variable of the mass flow rate, and a quadratic term providing penalization of the change rate of the temperature derivative. Quadratic weighting is chosen for the average molecular weight deviation since a small deviation is acceptable but a large deviation should be highly penalized. The batch time term is the treated the same way it is in the offline optimization; it is constrained to a region near zero and a slack variable is introduced to avoid infeasibilities arising. A slack variable is also introduced on the required cooling water flow rate to avoid infeasibilities arising from constraint violations. The final term is the penalty of the change in dT_R between two samples and is added to ensure a smooth temperature profile. The weights are summarized in Table 6.1. Note that the weight on the slack variable for the required cooling water flow rate is extremely large. This weighting proved necessary to enforce the upper limit on the output; even still in some simulations this is violated.

Table 6.1: Normalized weights used in the optimization of the reactor temperature profile.

Weight	Values
q	450.0
r_{1,t_f}	0.01
$r_{1,\hat{m}_{c,r}}$	2×10^7
s_{dT_R}	0.01

Constraints 6.2b-6.2d represent the process model. The batch time is penalized using a soft constraint and a slack variable as shown in Constraint 6.2e. Constraint 6.2f guarantees that the desired conversion is reached. To ensure that the operation conditions are safe, the reactor temperature is bounded by minimum and maximum values, and the reactor temperature derivative is limited to ensure that physical limits on the temperature change are considered as given by Constraints 6.2g and 6.2h, respectively. A constraint on the required cooling water flow rate is also introduced in Constraint 6.2i to reflect the cooling capacity limits. The cooling water inlet temperature is also constrained in Constraint 6.2j. Constraint 6.2k places a constraint on the number average molecular weight throughout the batch. The values of the bounds and other parameters used in the formulation are given in Table 6.2.

Table 6.2: Upper and lower bound values in the DRTO optimization problem.

Description	Symbol	Value	Unit
Desired number average molecular weight	$\overline{M}_{n,d}$	100.0	$\mathrm{kg\,mol^{-1}}$
Minimum number average molecular weight	$\overline{M}_{n,min}$	60.0	$\mathrm{kg\,mol^{-1}}$
Maximum number average molecular weight	$\overline{M}_{n,max}$	150.0	$\mathrm{kg\,mol^{-1}}$
Lower bound on batch time	$t_{f,min}$	0	min
Upper bound on batch time	$t_{f,max}$	0.1	min
Desired conversion	X_d	80	%
Lower bound on temperature	$T_{R,min}$	50	°C
Upper bound on temperature	$T_{R,max}$	140	°C
Lower bound on temperature derivative	$dT_{R,min}$	-0.5	$\mathrm{K\,s^{-1}}$
Upper bound on temperature derivative	$dT_{R,max}$	0.5	$\mathrm{K\,s^{-1}}$
Lower bound on mass flow rate of cooling fluid	$\hat{m}_{c,min}$	0	$\mathrm{kg\,s^{-1}}$
Upper bound on mass flow rate of cooling fluid	$\hat{m}_{c,max}$	4.0	$\mathrm{kg\,s^{-1}}$
Lower bound on cooling water temperature	$T_{J,i,min}$	10	°C
Lower bound on cooling water temperature	$T_{J,i,max}$	60	°C

Note that the upper bound of the cooling water flow rate has been changed in this layer compared to the NMPC layer. This is because it is desired to have a cooling constraint that would be active for at least part of the batch so a value of 4 $\mathrm{kg\,s^{-1}}$ is selected. For a real process, this bound would be based on the pipe diameter and the velocity of the cooling water; however, since this is a hypothetical reactor, a reasonable estimate is used. Different cooling fluid flow rate upper bounds and their effect on the process are examined in Section 6.3.3.1.

6.2.2.1 Prediction Horizon and Input Blocking

It is important that the prediction horizon for the DRTO layer be longer than the NMPC layer prediction horizon; this separation is important to meet real time requirements. In addition, the prediction horizon should be slightly longer than the expected batch time. Therefore the prediction horizon is set to be 4 hours (240 minutes) to be sufficiently long to ensure that the batch can reach the desired conversion; the sample time is set to 15 seconds, meaning that there are 960 samples for the full horizon. It is important that the DRTO prediction horizon be sufficiently long to capture process dynamics that effect the process long term [13].

To reduce the number of decision variables, input blocking is implemented for the temperature derivative. The prediction horizon is divided into 185 blocks; this means that there are only 186 decision variables to calculate for the temperature profile in comparison to the 961 variables without input blocking. The input blocking is constructed such that it is finer at the beginning so that the temperature can

vary in this region and the resolution decreases throughout the prediction horizon. A new temperature derivative is calculated every sample for the first 50 samples and then every fifth sample until sample 725 (181.25 minutes); for the remainder of the prediction horizon, a single block is used. This is because the batch is expected to be close to finishing at this point and therefore the temperature in this region does not affect the polymer product [21].

6.3 Solving the DRTO Problem

Cybernetica CENIT and RealSim are used again to solve and simulate the optimization problem. Similar to the NMPC layer, the DRTO application requires the optimization algorithm to be run at every time step. While often it is common to run the DRTO level less frequently than the NMPC level, here the DRTO level is run at each sample; this is done simply for ease when performing the simulations. Having the same sampling time in both the layers makes it easier to setup the simulations in RealSim. The simulator does not care if the time it takes to run a sample matches the set sample time in each layer; therefore, setting the sample time to a time shorter than what it actually takes for each layer to solve the optimization problem, does not affect the results. Some tests should be run to determine what the actual sampling time for the DRTO layer should be before this control structure is implemented on a real process; this value should be selected based on how long it takes the DRTO layer to solve the problem.

Since this is online optimization, complete convergence to the optimal solution in one step is not required. Therefore to reduce the computational effort, the number of SQP iterations is limited to 5 to guarantee that the problem solves rapidly enough for real time application.

As is done in the NMPC layer, the model is set to stop calculating all time derivatives once 80% conversion is reached. The model used in this level optimizes the reactor temperature derivative so it no longer calculates it the same way as is done in the NMPC layer. Perfect temperature control in CENIT is assumed in this level so the reactor temperature derivative is set equal to the calculated optimal reactor temperature derivative and not calculated using the energy balance; this is the same approach that is used in offline optimization.

6.3.1 Communication Between the Layers

The two levels need to be connected to create the full control structure, which is the goal of this thesis. These two layers have been constructed to operate on different time scales; these time scales are specified by the prediction horizon and the sampling time. The upper DRTO layer has a prediction horizon that is long enough to predict past the end of the batch. This layer is constructed to calculate the optimal reactor temperature profile. The lower layer has a shorter prediction

horizon so it does not see the end of the batch until the end of the simulation. Because the top layer has a longer prediction horizon, there are more variables to calculate and thus the computational effort for the top layer is greater than for the bottom layer. While it is common to alter the sample time for the top layer to decrease the computational effort, it is not done in this instance for ease of simulation as previously explained. The NMPC layer functions as the advanced control level for the process by operating on a shorter time scale and responding quickly to reject any key disturbances in the process to maintain safe operation. The DRTO works on top of this on a longer time scale to respond to slow-varying disturbances.

This communication requires two different instances of CENIT to operate simultaneously; both layers also communicate with RealSim. RealSim provides the measurements to both the DRTO layer and the NMPC layer. The DRTO level calculates the optimal reactor temperature trajectory and then passes this to the NMPC level. In CENIT, this is done through a text file (Tref.txt); the DRTO layer writes out the reactor temperature profile data for the length of the NMPC prediction horizon. The NMPC layer then optimizes the cooling water flow rate so that it can follow the trajectory. The calculated states from the NMPC level are then sent back to the DRTO level so it can recalculate the optimal reactor temperature trajectory based on the new measurements; this is done via a text file (States.txt). In this instance, OPC is used to pass the cooling water inlet temperature and the initial reactor temperature to both layers; however, in a real application, the two layers would not communicate through the OPC but instead these values would be written to a file as was done for the reactor temperature profile and the states. Figure 6.2 illustrates the communication flow used in this work.

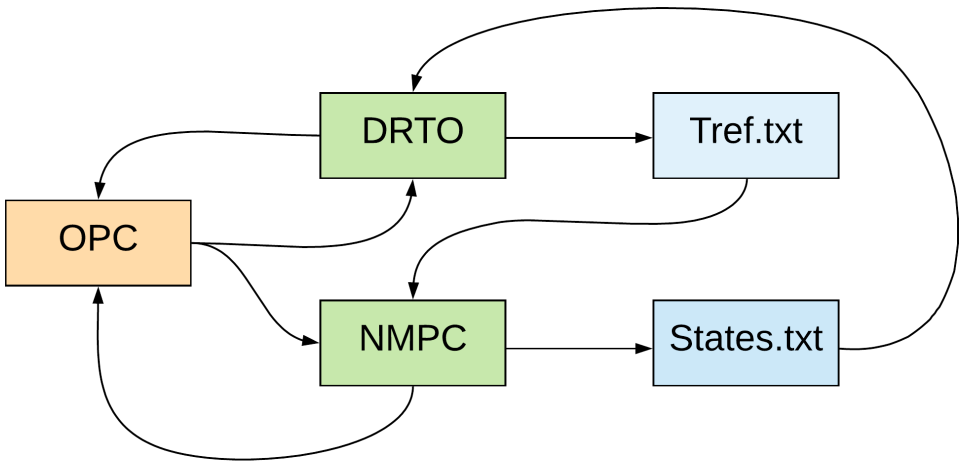


Figure 6.2: Illustration of the communication flow between the layers.

6.3.2 Alterations

When the two layers are connected, the advanced control layer may require some further tuning in order for the layers to work together as desired. Sometimes the two layers may compete since they have different goals so together they have to be tuned to work together. The upper bound on the cooling water flow rate in the NMPC is changed to match the constraint on the required cooling water flow rate in the DRTO layer. Even though the DRTO layer does not send the cooling water flow rate values to the NMPC layer, the value of the required cooling water flow rate calculated in the supervisory layer affects the calculated reactor temperature derivative. This affects the reactor temperature, which is sent to the NMPC layer and can have an impact on the calculated optimal cooling flow rate.

6.3.3 Results

Simulations are run to confirm that the two-level control structure operates as desired. A cooling water inlet temperature of 10 °C is used with all the other values the same as previous simulations. All the plots included the initially predicted trajectories at $t = 0$ from the DRTO versus the actual simulated trajectories from both layers; the trajectories from the NMPC layer are not included because the prediction horizon for this layer is so short compared to the full batch.

The reactor temperature begins at 120 °C and increases throughout the batch before settling just under the maximum allowable temperature as seen Figure 6.3.

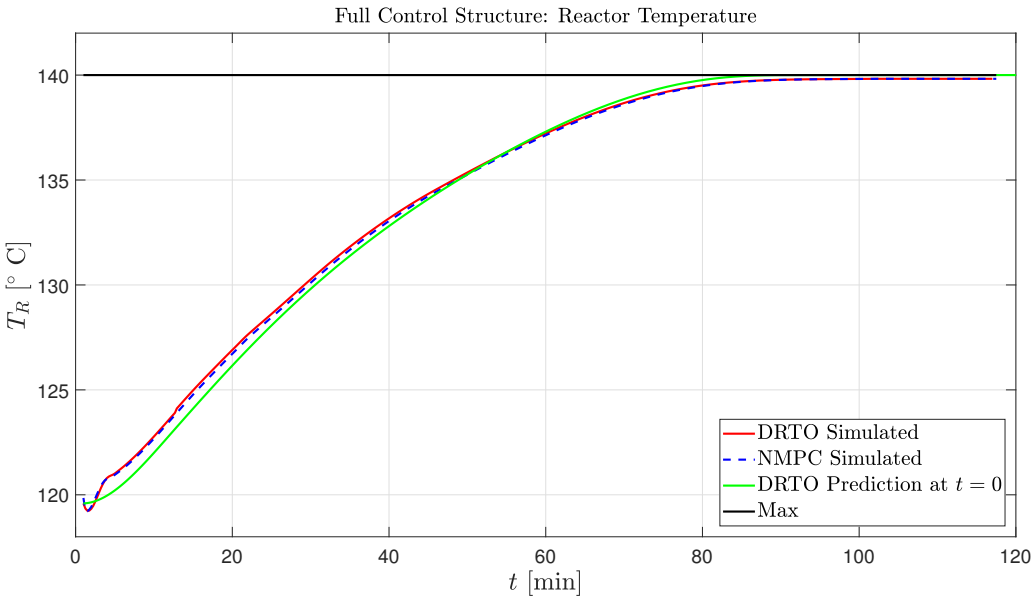


Figure 6.3: Reactor temperature using a two-level advanced control structure.

The initially predicted temperature profile by the DRTO layer differs marginally from the simulated result; the predicted temperature profile reaches the maximum value just before the simulated profile. For the simulated results, the reactor temperature decreases as the cooling water flow rate increases rapidly and carries away the heat produced by the reactions; the reactor temperature increases to match the shape of the predicted profile after just a few minutes. The simulated results illustrate that the two layers are able work together to further optimize the reactor temperature throughout the batch. It is a positive that the simulated profile is similar to the predicted profile since the same model and operating conditions are used and no process or measurement noise is considered.

As seen in Figure 6.4, the predicted and simulated jacket temperatures differ considerablely. The DRTO predicted jacket temperature is similar to the one seen in the offline optimization stage because the predicted cooling flow rate at this point is zero/extremely low. The simulated results show how the jacket temperature profile looks when there is a cooling water flow rate. The jacket temperature spikes at the beginning as the cooling water flow rate slowly increases to work on removing the generated heat. As the reactions proceeded, the jacket temperature increases as it works to remove the heat generated in the reactor. By the end of the batch, the jacket temperature reaches 130 °C.

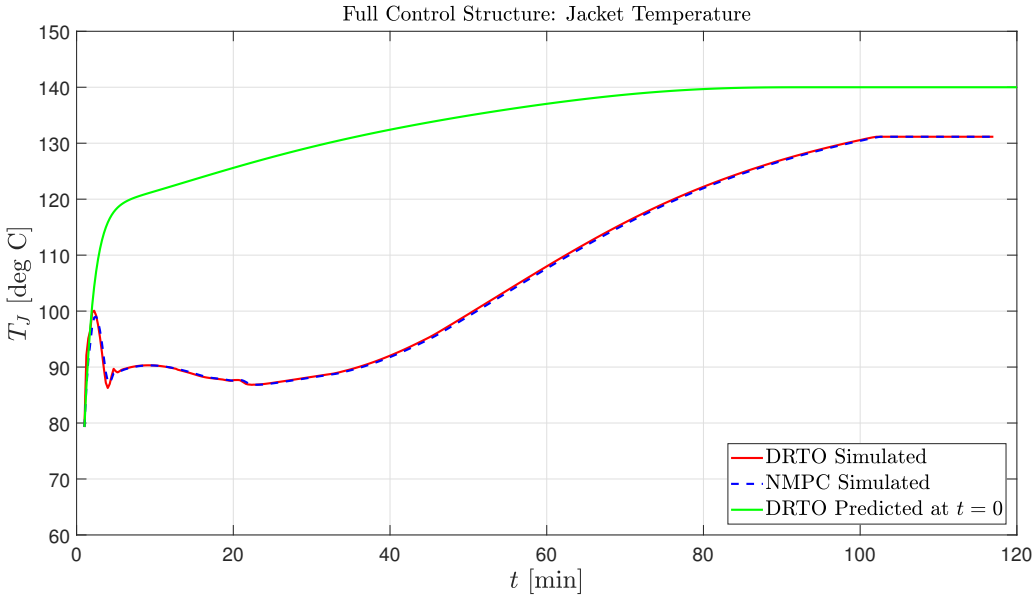


Figure 6.4: Jacket temperature using a two-level advanced control structure.

The cooling water flow rates determined by the DRTO and NMPC layers are shown by Figure 6.5. The DRTO calculates the required flow rate to follow the optimal reactor temperature derivative; the constraint on this calculation is a soft constraint

and therefore, it is possible to violate this constraint. The NMPC layer determines the optimal flow rate based on the hard constraint and is the actual flow rate that is implemented in the process. The predicted cooling water flow rate is lower than the realized flow rate. This is because it takes a few samples before the controller realizes the true amount of cooling water required to realize the reactor temperature trajectory; consequently the DRTO level is poor at predicting the cooling water flow rate for the whole prediction horizon at the start of the batch. The NMPC level is better at predicting the cooling water flow rate but only predicts for 15 minutes. The two simulated profiles are nearly identical for a cooling constraint of 4 kgs^{-1}. The constraint is active for a brief period of time between minute 20 and 40 into the batch; for the remaining parts of the batch, the flow rate is always under the maximum allowed value. These results suggest that a cooling constraint of 4.0 kgs^{-1} provides sufficient cooling capacity for a cooling water temperature of 10 °C as expected. However, this constraint may prove too constricting for other cooling water temperatures.

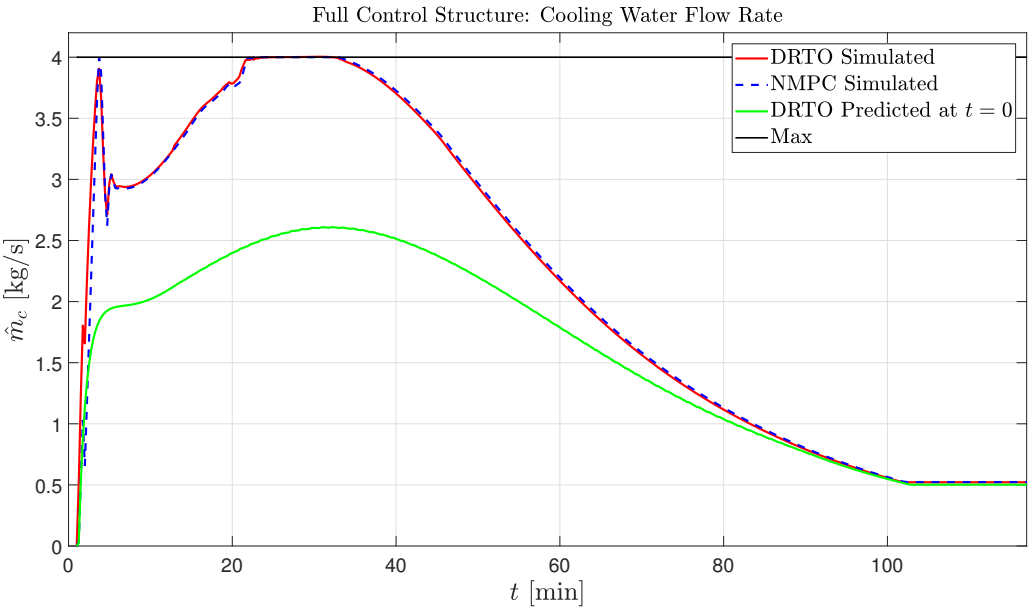


Figure 6.5: Cooling water flow rate using a two-level advanced control structure.

The initiator consumption throughout the batch time is shown in Figure 6.6. By the end of the batch only 11.99 mol of initiator remain, meaning that 92.7% of the initiator has been consumed. In reality since some initiator will likely be lost to the continuous phase, little initiator will be wasted. The simulated consumption profiles are nearly identical to the predicted value, which is good.

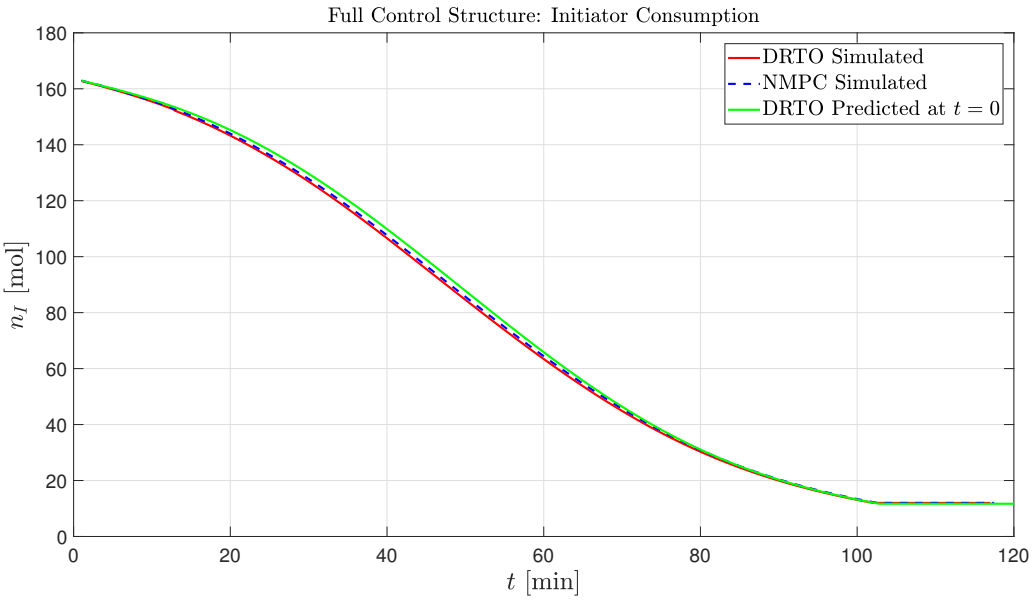


Figure 6.6: Initiator consumption rate using a two-level advanced control structure.

Figure 6.7 illustrates the number average molecular weight throughout the batch.

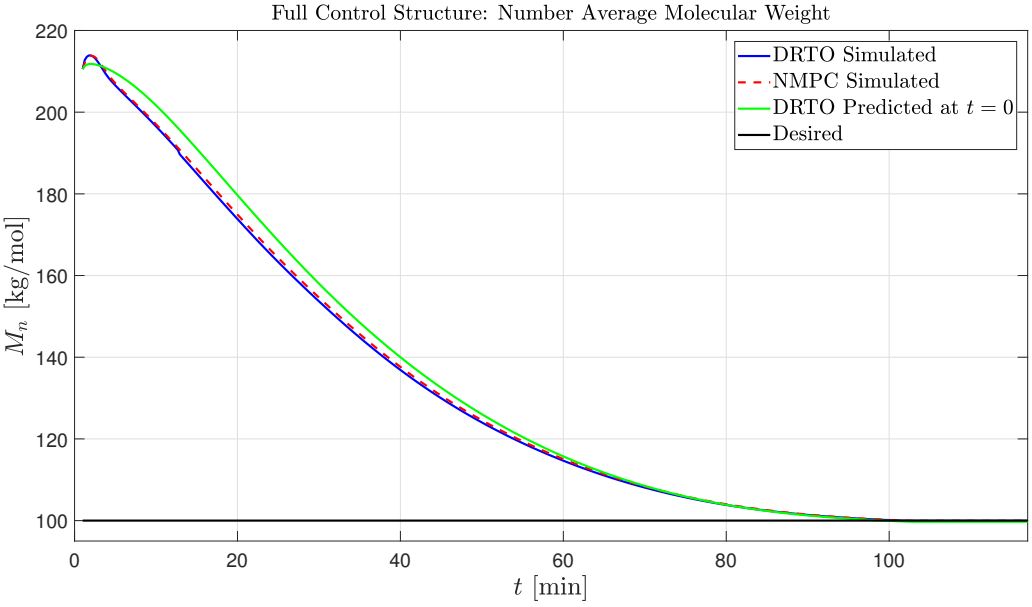


Figure 6.7: Number average molecular weights using a two-level advanced control structure.

The final number average molecular weight is 99.95 kg/mol, which is within 0.05%

of the desired value. This is a slightly better value than is found for the NMPC following a fixed reactor trajectory. This illustrates how the use of a DRTO layer can help further improve the process. While the improvement is minor, it should be recalled that there is no cooling constraint considered in the offline optimization, which has an impact on the reactor behavior and product. The corresponding weight average molecular weight is 179.07 kg/mol, resulting in a PI of 1.79.

The conversion throughout the batch is shown in Figure 6.8. The batch time is 102.7 min, which is only 24 s longer than the offline optimization predicted time and the NMPC level following a reference trajectory. The simulated batch time is identical to the predicted batch time. This suggests that the addition of a cooling constraint of 4 $\mathrm{kg\,s^{-1}}$ has little impact on the overall batch time in these circumstances.

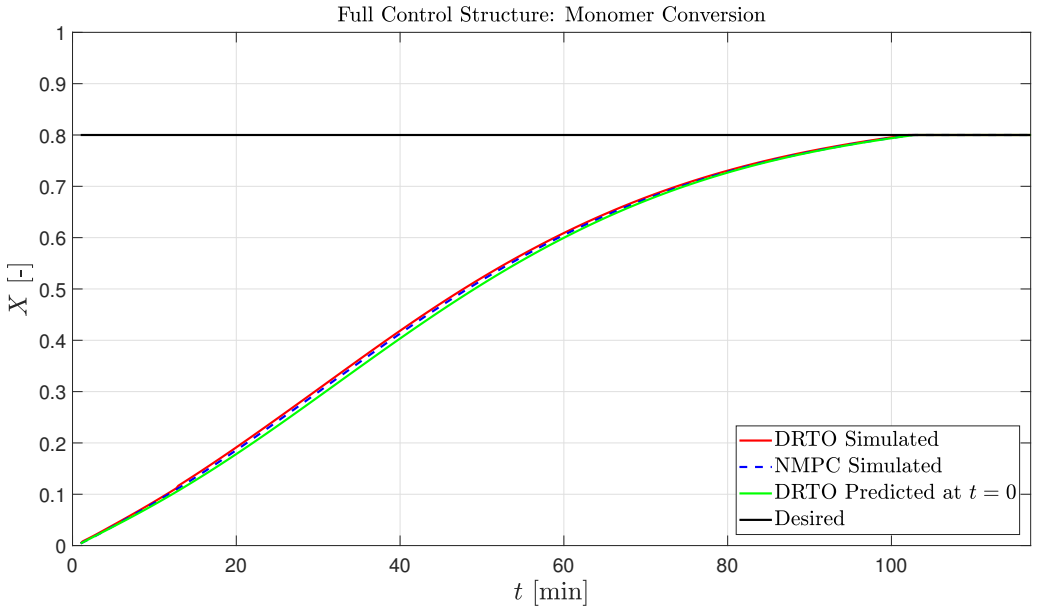


Figure 6.8: Conversion rate of monomer to polymer using a two-level advanced control structure.

6.3.3.1 Controller Testing

The control structure is now tested to determine where the system is most sensitive to disturbances and to identify the operating limits of the controller. The cooling capacity of the system and the cooling water inlet temperature are changed to test the controller. First, the upper bound on the cooling water flow rate is varied to examine how the cooling capacity of the system affects the batch time and the product quality. A constant upper bound on the cooling water flow rate is simulated first; subsequently, the constraint is varied at different points throughout the simulation to discover where the process is most sensitive to the cooling water flow

rate. The results of these simulations are presented in Section 6.3.3.1.

The cooling water inlet temperature is then varied to capture any seasonal changes that may cause the cooling water source to change. The first simulations use a constant cooling water temperature, with the following ones having a disturbance of ±5 °C at some point during the batch time. These simulations will reveal how sensitive the process is to this disturbance at different times in the production process. Section 6.3.3.1 examines the outcome of these simulations.

Disturbance One: Cooling Capacity

A system's cooling capacity is determined by several physical constraints but in this hypothetical system, the upper bound is selected somewhat arbitrarily; therefore, to examine the controller's ability to handle a different cooling capacity, several different cooling water flow rate upper bounds are examined. As seen previously, a cooling constraint of 4 kgs^{-1} provides sufficient heat removal for this process when the cooling water temperature is 10 °C. It is desired to discover at what mass flow rate the system can no longer achieve the desired results. Simulations are run with a constant maximum constraint throughout the whole batch ranging between 3.75-4.5 kgs^{-1}. Simulations of this nature illustrate the effects of having a smaller or larger cooling capacity and reveal how sensitive the system is to the cooling capacity.

What happens when the constraint decreases at some point during the batch is then examined; the exact point where the decrease occurs is changed to determine where the system is most vulnerable to disturbances in this constraint. This study is done to mimic the possibly real scenario where a plantwide optimizer decides that the optimal cooling water allocation should be different. These simulations demonstrate the effect this would have on this process.

Constant Constraint:

The following five different maximum cooling water flow rates are simulated: 3.75, 4.0, 4.25, 4.3, and 4.5 kgs^{-1}. In the figures below, the dashed lines represent the values determined by the NMPC layer and the solid lines represent the values determined by the DRTO layer; the values determined by the NMPC layer are the ones that would actually be implemented in the plant.

Figure 6.9 shows that the cooling constraint has a large impact on the reactor temperature profile. A cooling capacity of 3.75 kgs^{-1} results in a reactor temperature profile that substantially violates the upper bound; this suggests that the cooling

constraint is too tight for this cooling water inlet temperature. This temperature profile is similar to the others until thirty minutes into the batch where it begins to deviate. The other four cooling limits have similar reactor temperature profiles beginning at 120 °C and ending just below the upper limit. These four profiles become indistinguishable after about fifty minutes into the batch. The differences in the reactor temperature trajectories can be explained by examining the corresponding cooling water mass flow rates.

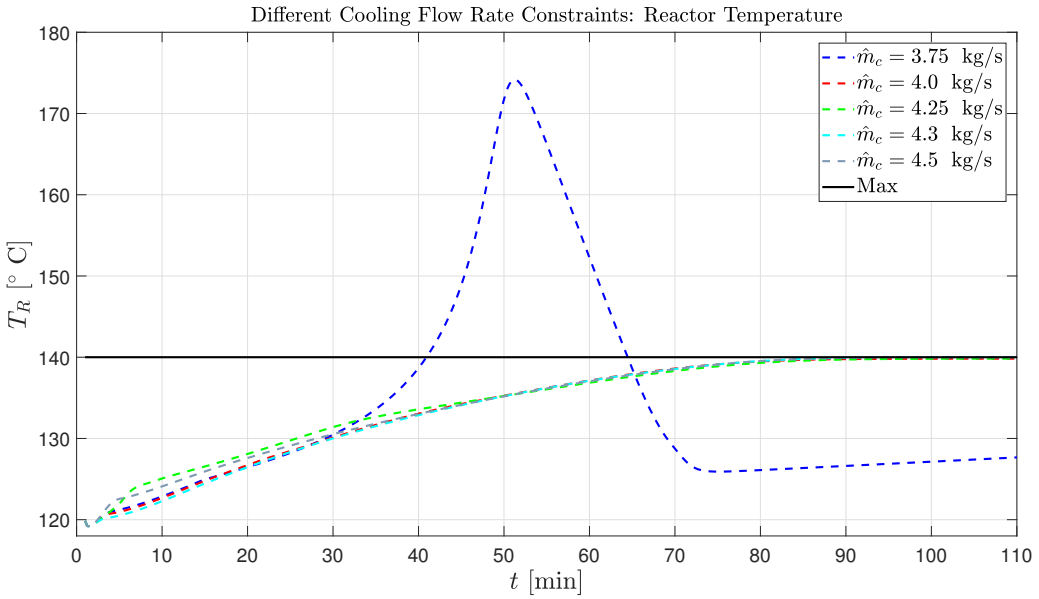


Figure 6.9: Reactor temperatures for different cooling constraints using a two-level advanced control structure.

Figure 6.10 reveals that the cooling water upper limit has the most noticeable influence at the beginning of the batch. This is the point in the production where the cooling water flow rates are increasing so the jacket temperatures quickly rise, producing a spike; once the cooling water flow rates become high enough to cool the jacket, the jacket temperature profiles are smooth. By the end of the batch, all of the jacket temperatures have settled to 130 °C, with the exception of the lowest constraint; this is aligned with the similar cooling water flow rates for this period of the batch. A cooling constraint of 3.75 $\mathrm{kg s^{-1}}$ has a similar jacket temperature profile until around 45 minutes into the batch; here it forms a peak, followed by a dip before increasing to 128 °C. The end jacket temperature is lower for this constraint since the final reactor temperature is significantly lower compared to the alternative constraints. The jacket temperature is higher than the reactor temperature because the cooling water flow rate towards the end of the batch is nearly zero, which is much lower than for the other cooling constraints so heat is

still being transferred to the jacket.

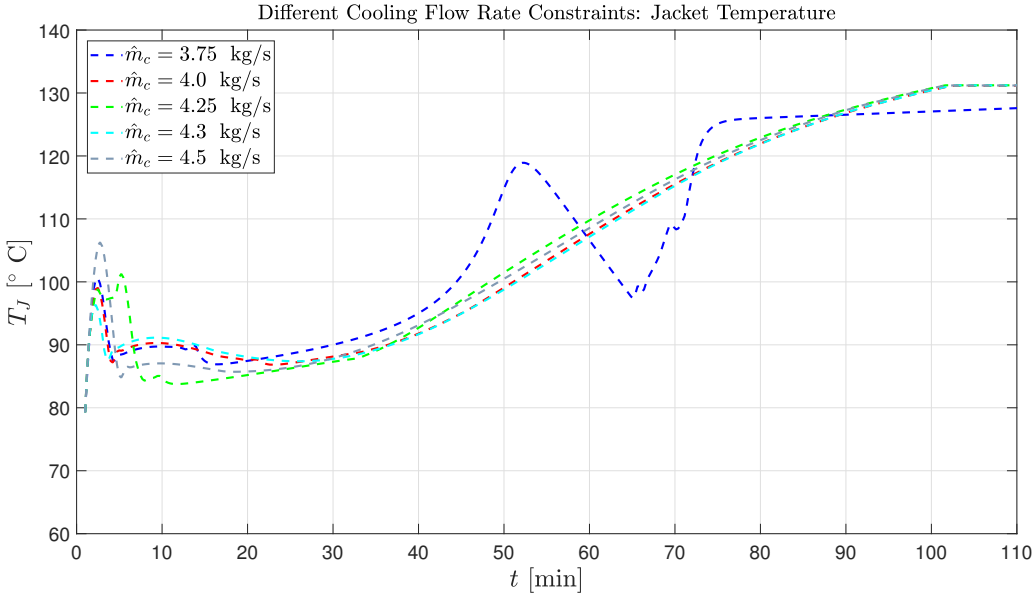


Figure 6.10: Jacket temperatures for different cooling constraints using a two-level advanced control structure.

Figure 6.11 illustrates the cooling water flow rate for each of the tested constraints.

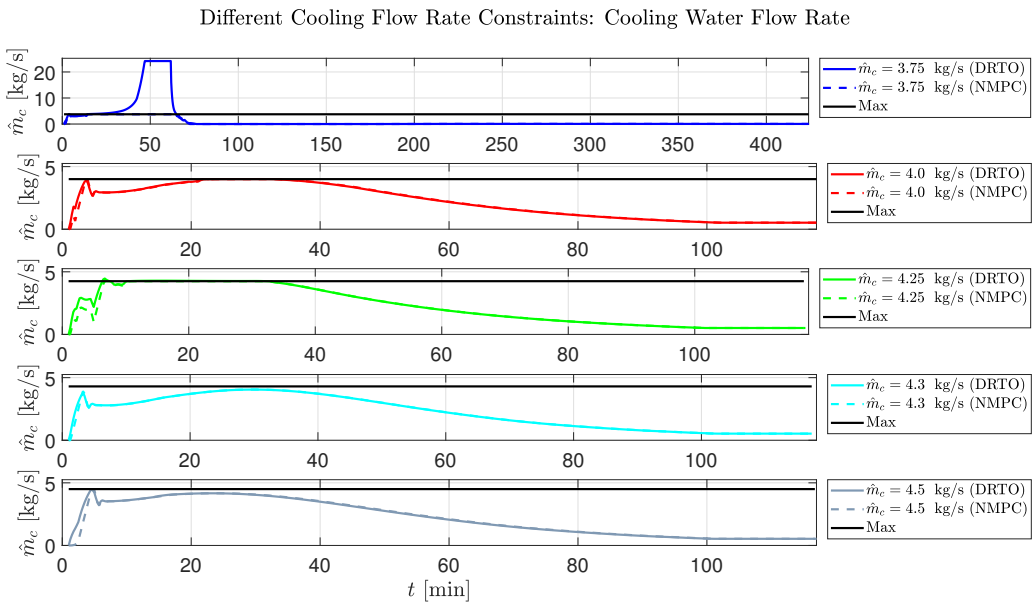


Figure 6.11: Cooling water flow rates for different cooling flow rate constraints using a two-level advanced control structure.

The dashed lines represent the required cooling flow rate based on the energy balances of the jacket and the reactor calculated in the DRTO layer. The solid lines represent the optimal cooling flow rate calculated in the NMPC layer; this is the actual cooling water flow rate that would be implemented in the process. When the upper limit is 3.75 $\mathrm{kg s^{-1}}$, the cooling water flow rate in both layers violates the upper constraint; this suggests that this cooling capacity is insufficient to provide the heat removal the process needs. The cooling water flow rate determined by the DRTO and NMPC layers are nearly identical for the other cooling constraints; a 4.25 $\mathrm{kg s^{-1}}$ cooling constraint has a somewhat different initial cooling profile but matches the others after approximately ten minutes. When the cooling capacity is under 4.3 $\mathrm{kg s^{-1}}$ the constraint is active for a portion of the batch; the smaller the flow rate, the longer the constraint is active for. The constraint is inactive for a limit at or above 4.3 $\mathrm{kg s^{-1}}$.

The final number average molecular weight is only noticeable altered for the lowest simulated cooling capacity as seen in Figure 6.12. The deviation from the setpoint in this instance is 33.51 %. All the other cooling water flow rate upper bounds are able to produce a final number average molecular weight within the allowed range. The final number average molecular weights for the other constraints are nearly identical.

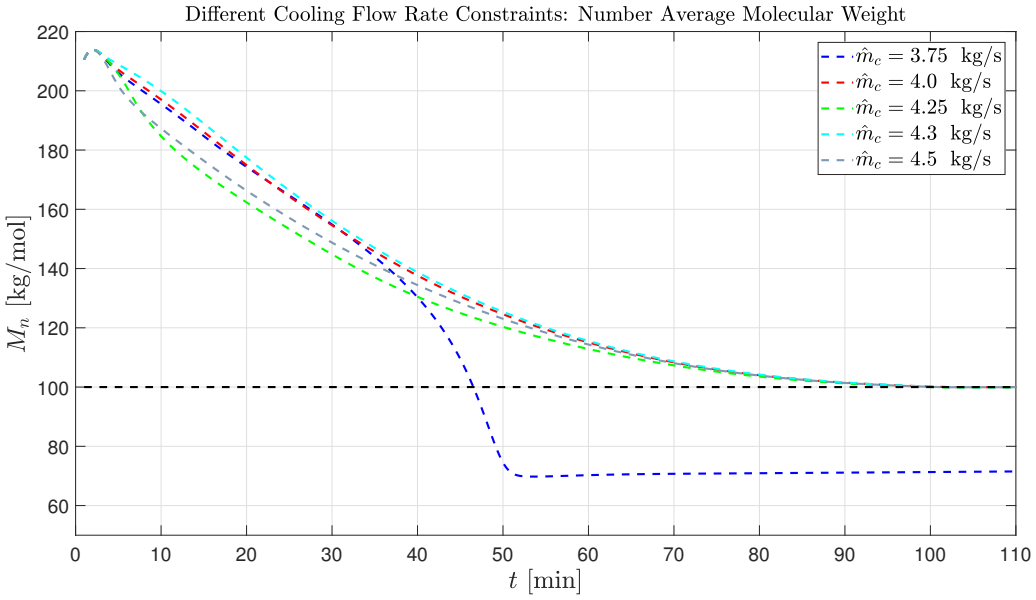


Figure 6.12: Number average molecular weight for different cooling flow rate constraints using a two-level advanced control structure.

The batch time for each of the different constraints is shown in Figure 6.13. The lowest cooling constraint of 3.75 $\mathrm{kg s^{-1}}$ has a batch time four times the other limits.

All of the other cooling constraints give approximately the same batch time of 102-103 minutes. A constraint of 4.25 kgs^{-1} gives the shortest time but also has the polymer with the "worst" final number average molecular weight of the four that can achieve the desired value. These results suggest that for a constraint at or above 4 kgs^{-1}, the batch time is not substantially altered. However, if the constraint is below 4 kgs^{-1}, the system can achieve the desired conversion but not the required polymer quality; in addition, it takes four times longer to reach the 80% conversion.

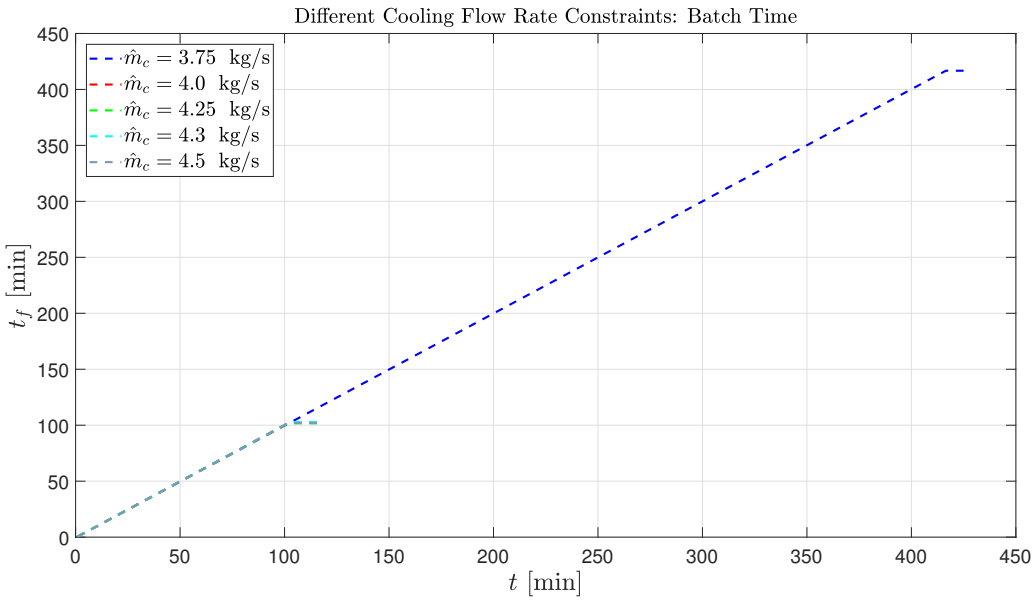


Figure 6.13: Batch times for different cooling flow rate constraints using a two-level advanced control structure.

Figure 6.14 shows the initiator consumption for the different cooling water flow rate constraints. The four highest constraints result in approximately the same amount of initiator being consumed during the batch. Since they all have the same consumption, this suggests that even though the operating conditions are slightly different the amount of initiator used is near the optimal value. However, it would be beneficial to include the amount on initiator as a decision variable similar to what was done in the offline optimization so that if the operating conditions are further changed, the amount of initiator will update accordingly. For a upper bound of 3.75 kgs^{-1}, all of the initiator is consumed within about fifty minutes into the batch; this is due to the high reactor temperature.

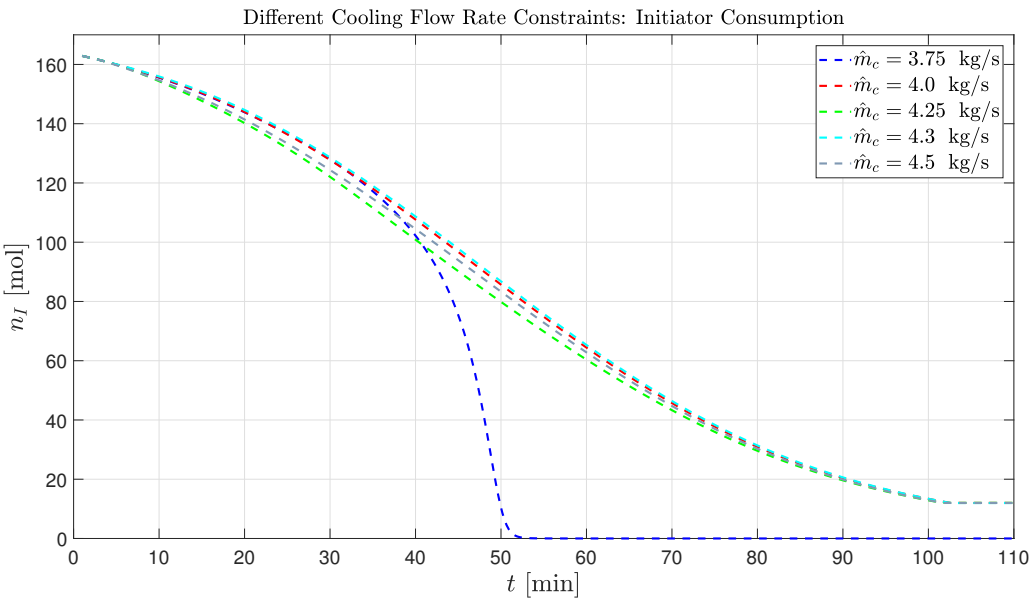


Figure 6.14: Rate of initiator consumption for different cooling flow rate constraints using a two-level advanced control structure.

The results for each of the simulated cooling flow rate constraints are summarized in Table 6.3.

Table 6.3: Simulation results for different cooling flow rate constraints.

$\hat{m}_{c,max}$ [kgs^{-1}]	$\overline{M}_n$ [kg/mol]	$\overline{M}_w$ [kg/mol]	PI [-]	t_f [min]
3.75	77.49	240.20	3.10	416.7
4.0	99.95	179.07	1.79	102.7
4.25	99.89	176.04	1.76	101.8
4.3	99.95	179.60	1.80	102.8
4.5	99.94	177.33	1.77	101.9

The results indicate that for a cooling constraint between 4-4.5 kgs^{-1}, the affect on the polymer properties and the batch time is small. While the range of cooling flow rates tested here is relatively small, the purpose of these simulations is to determine the minimum maximum constraint that will still result in a shortened batch time and the required polymer quality. Cooling water flow rates below 4.0 kgs^{-1} plus a cooling water inlet temperature of 10 °C are unable to produce the desired results. This suggests that a cooling capacity below 4.0 kgs^{-1} is inadequate to provide the necessary cooling for the system to maintain the reactor temperature that is required to obtain the desired polymeric properties at this cooling water temperature.

Changing Constraint:

The system has an initial cooling water flow rate of 4.3 kgs^{-1} and at some point during the production, the cooling capacity is decreased to 4 kgs^{-1}. An initial cooling flow rate of 4.3 kgs^{-1} is selected because a decrease from 4 kgs^{-1} would not guarantee that the needed product quality is achieved.

The reactor temperature profiles for each of the different constraint changes are shown in Figure 6.15. The largest impact on the reactor temperature profile is seen when the change occurs at sample 100; in this case, the temperature exceeds the upper bound by almost ten degrees. The controller is able to bring the reactor temperature below the limit after about twenty minutes with the temperature being around 138 °C before dropping to 130 °C at 100 minutes into the batch. This large of a deviation from the optimal profile means that the desired product quality will not be possible and that it will take longer for 80% conversion to be reached. It is interesting to see that the beginning of the trajectories are the same for sample 100 and 150 and similarly for sample 20 and 250; this can be explained by examining the corresponding cooling water flow rates which show similar behavior. After 30 minutes though, sample 150 and 250 trajectories become identical. These results suggest that the system is most sensitive to a decrease in the cooling water capacity near sample 100 when the cooling water flow rate is maximum.

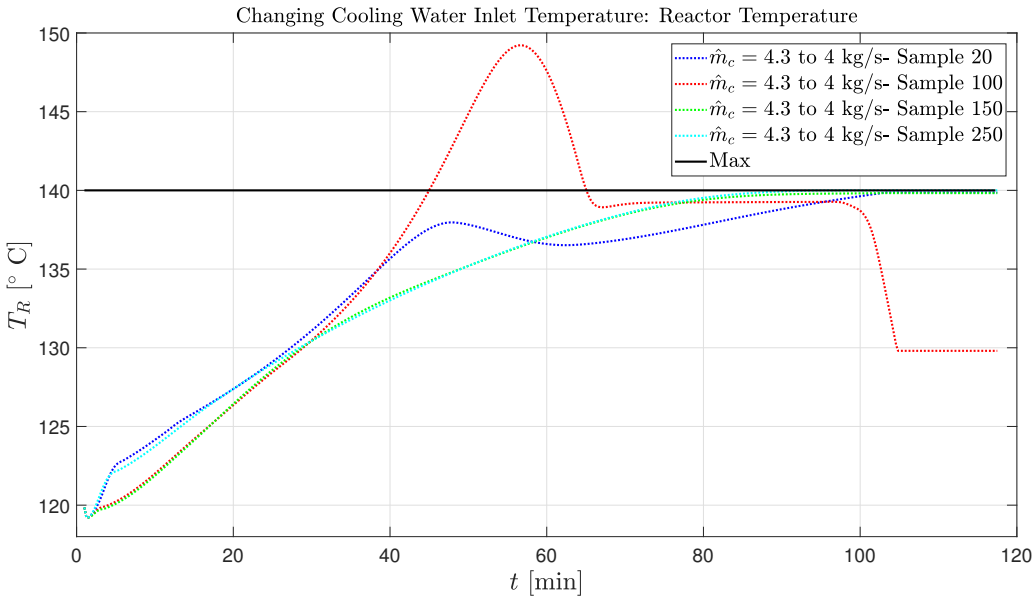


Figure 6.15: Reactor temperatures for a varying cooling capacity using a two-level advanced control structure.

The optimal cooling water flow rates shown in Figure 6.16 explain the reactor temperature profiles. If the constraint is not changed before sample 150, the cooling constraint is inactive. When the change is implemented at sample 20, the cooling constraint becomes active shortly after; a similar response is seen when the change is made at sample 100. Sample 100 produces a unique cooling water flow rate since around 65 minutes, the cooling water flow rate rapidly decreases and then remains at a low value for a period of time before increasing again around minute 100, corresponding to where the reactor temperature drops. This odd behavior is a result of the controller in the NMPC layer struggling to track the reactor temperature profile; the cooling water flow rate drops when the reactor temperature is below the upper bound. The NMPC controller then decides that the cooling water flow rate should be higher, which results in the reactor temperature dropping.

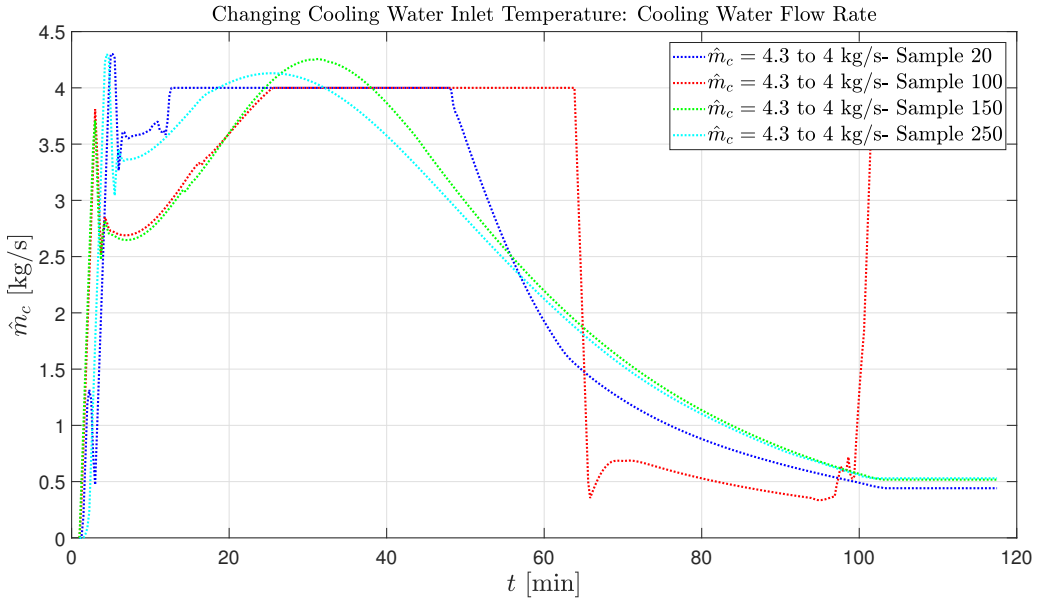


Figure 6.16: Cooling water flow rates for varying cooling capacity using a two-level advanced control structure.

The jacket temperature for each of the cooling constraints is illustrated in Figure 6.17. The largest differences are seen when the change is made at sample 100. Sample 100 has an odd increase in the jacket temperature around the 60 minute mark, reflecting the decrease in the cooling water flow rate. The jacket temperature decreases when the cooling water flow rate increases toward the end of the batch. As is the case for the reactor temperature profiles, sample 20 and 250 have similar jacket temperatures as do sample 100 and 150. Three of the simulations end with the jacket temperatures near 130 °C, whereas the jacket temperature for sample 100 ends around 90 °C.

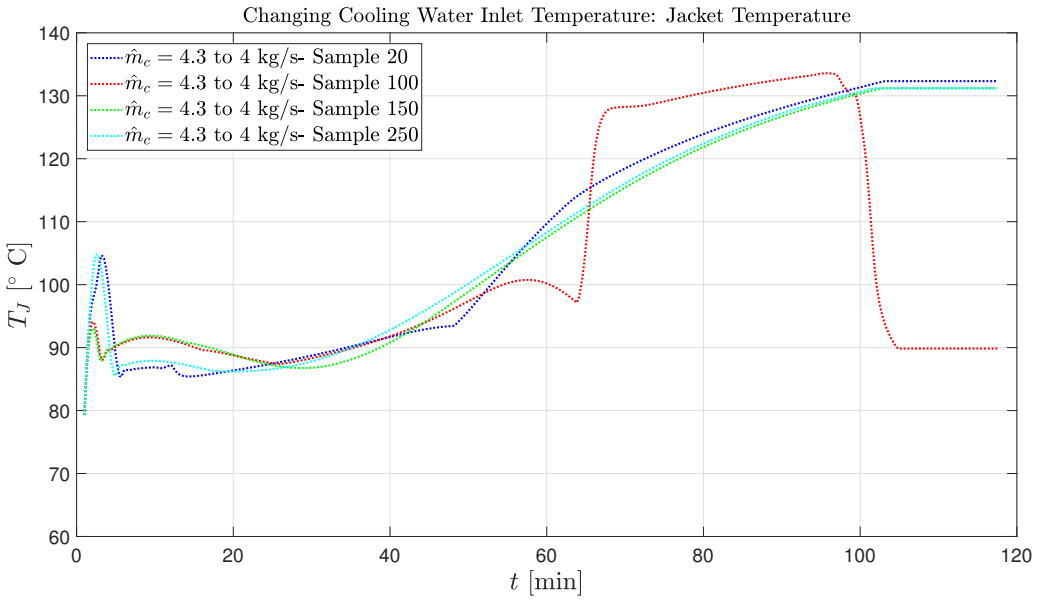


Figure 6.17: Jacket temperature profiles for varying cooling capacity using a two-level advanced control structure.

The number average molecular weights during the batch time are shown in Figure 6.18.

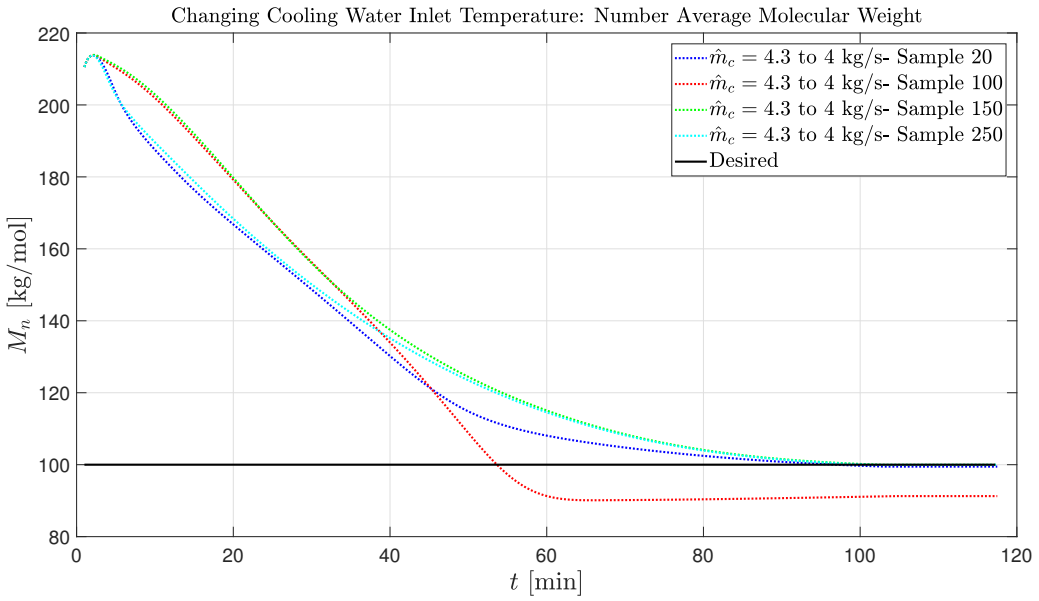


Figure 6.18: Number average molecular weights for varying cooling capacity using a two-level advanced control structure.

When the cooling capacity is changed at sample 100, the needed final number average molecular weight is not achieved; the value is 7.73% away from the desired final number average molecular weight. In all other circumstances, the required product purity is achieved.

Table 6.4 summarizes the outcome of all the simulations to illustrate the consequence that decreasing the cooling capacity at various points in the batch has on the final product and the batch time. The results indicate that the system is most sensitive to a decrease in the cooling capacity towards the beginning of the batch, particularly if the change is implemented at or near sample 100. While the batch time is not significantly extended, the final number average molecular is outside the desired range. These four different simulations reveal that the batch is most sensitive to a change in the cooling water flow rate upper limit where the cooling water flow rate is at its highest. Therefore, it is recommended to either introduce a safety margin to the cooling water flow rate upper bound or to place a restriction on when in the batch the constraint can be altered.

Table 6.4: Affect of cooling capacity changes at various points in the batch.

$\hat{m}_{c,max}$ [kg s^{-1}]	**Sample**	$\overline{M}_n$ [kg mol^{-1}]	$\overline{M}_w$ [kg mol^{-1}]	PI [-]	t_f [min]
4.3 to 4.0	20	99.47	175.99	1.77	103.3
4.3 to 4.0	100	91.27	174.86	1.92	104.9
4.3 to 4.0	150	99.95	179.62	1.80	103.1
4.3 to 4.0	250	99.94	177.75	1.78	102.1

Disturbance Two: Cooling Water Inlet Temperature

Two different simulation types are run in this section: 1) constant cooling water inlet temperature, and 2) changing cooling water inlet temperature. For the constant cooling water inlet temperature simulations, three different cooling water temperatures are tested for a cooling capacity of 4.3 kg s^{-1}. The original cooling capacity of 4 kg s^{-1} is not used because that system cannot handle higher cooling water temperatures due to the insufficient cooling capacity; therefore, it is deemed prudent to run the simulations with a cooling capacity that could handle higher cooling water temperatures which would provide some further insight into the system.

Constant Cooling Water Inlet Temperature:

The cooling water inlet temperature has a large impact on the reactor temperature

profiles as seen in Figure 6.19, where a cooling water temperature of 20 °C has the greatest influence. The reactor temperature violates the upper bound by almost 35 °C for this cooling water temperature. This points out a limitation in the system itself rather than indicating a construction flaw in the controller. These results show that the system cannot handle a cooling water inlet temperature much above 15 °C for this cooling constraint; a larger cooling capacity is therefore necessary to realize the optimal reactor temperature profile at this temperature. The reactor temperature profiles for the other two cooling water temperatures have the largest deviations from one another at the beginning of the batch; these differences are unsurprising based on the corresponding cooling water profiles. Despite the discrepancies at the start of the batch, after about minute forty, the profiles are the same with the final reactor temperature ending at just under the limit.

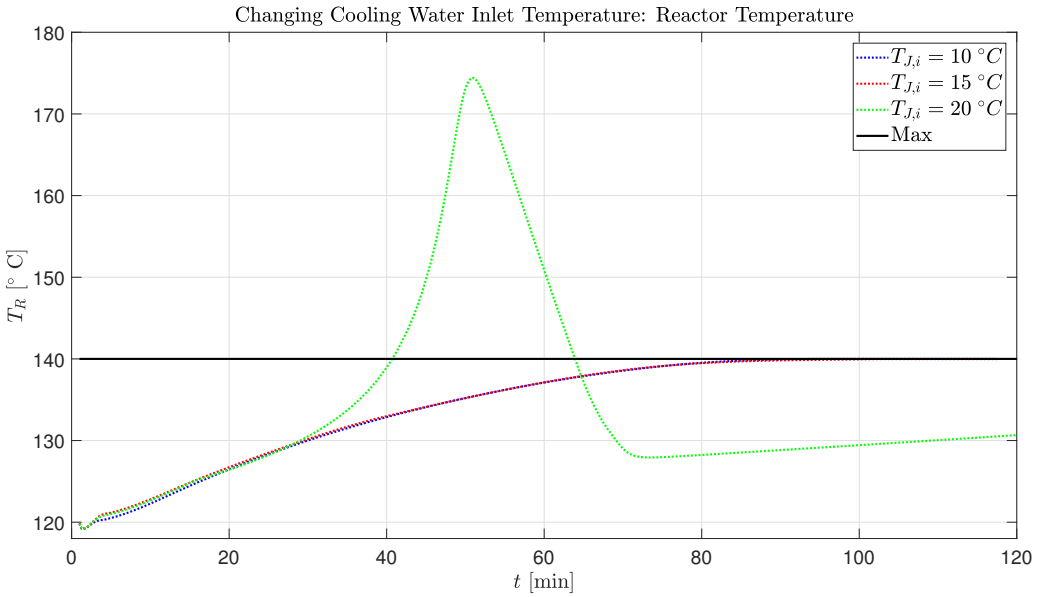


Figure 6.19: Reactor temperature profiles for various cooling water temperatures using a two-level advanced control structure.

Figure 6.20 shows the corresponding cooling water flow rates for each cooling water inlet temperature. The cooling constraint is not active for the lowest cooling water temperature; it is active for the middle cooling water temperature for a short period of time. In comparison, for the highest simulated temperature, the constraint is severely violated by the DRTO layer. The controller brings the cooling water flow rate under the upper limit again, where it remains quite low for the remainder of the batch. This might seem counter-intuitive but the reactor temperature decreases so rapidly that the cooling water must be turned off so that the reactor temperature can try to reach the desired value.

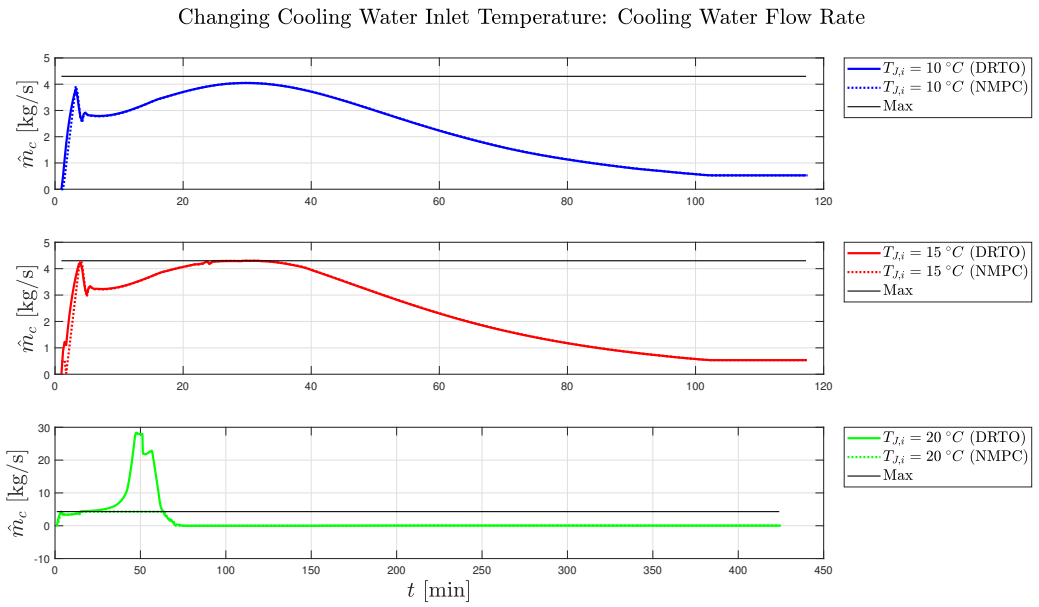


Figure 6.20: Cooling water flow rates for various cooling water temperatures using a two-level advanced control structure.

To handle the cooling water constraint violation in the DRTO layer for temperature of 20 °C, the controller compensates by decreasing the reactor temperature derivative as shown in Figure 6.21.

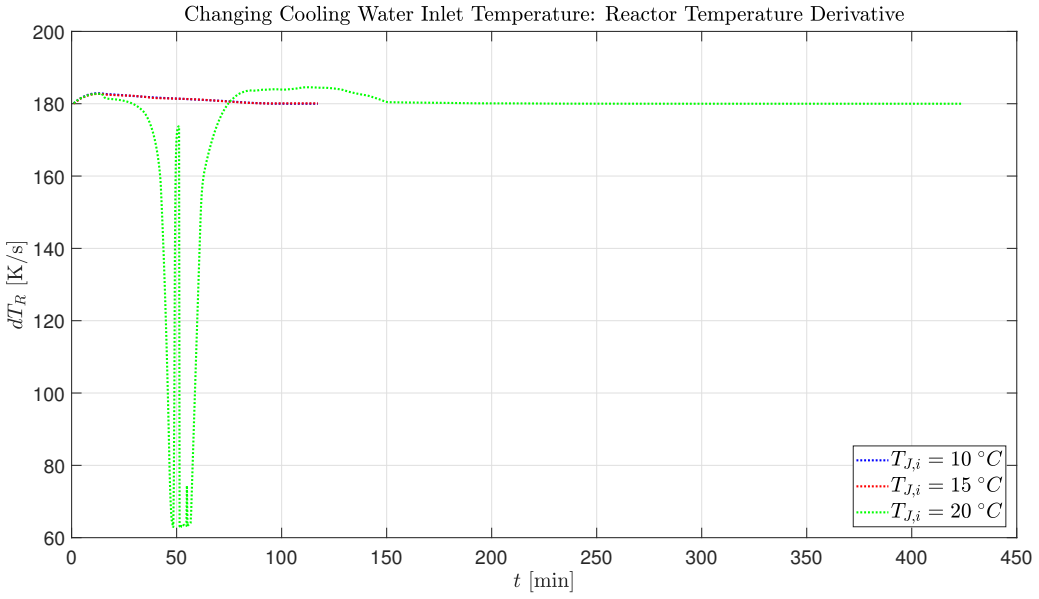


Figure 6.21: Reactor temperature derivatives for various cooling water temperatures using a two-level advanced control structure.

The derivative decreases considerably more for the highest cooling water temperature compared to the other temperatures. The calculated reactor temperature derivatives track with the changes in the reactor temperature and the cooling water flow rate as expected. The derivative for the other two cooling water temperatures are identical and illustrate how small the changes are if the optimal reactor temperature trajectory is achieved. The change is much smoother for the lower temperatures, which is unsurprising since the related reactor temperature profiles are smooth.

The jacket temperature profiles for each of the feed temperatures are plotted in Figure 6.22. For higher cooling water temperatures, the jacket temperature is increased. However, for the two lowest temperatures, the profiles are nearly identical; this is expected given that the reactor temperature profiles are nearly indistinguishable for these feed temperatures. In the case of a 20 °C cooling water temperature, the jacket temperature does not have as smooth of a profile as the others, which is a product of the reactor temperature at these points.

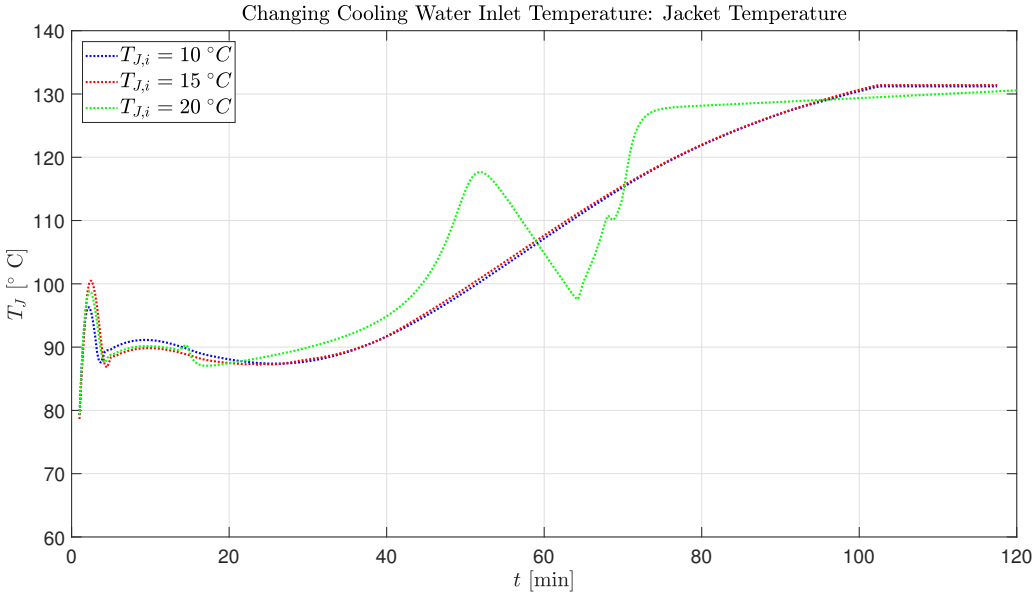


Figure 6.22: Jacket temperatures for various cooling water temperatures using a two-level advanced control structure.

The number average molecular weights for the various cooling water temperatures are presented in Figure 6.23. For the highest cooling water temperature, the controller is unable to produce the desired final number average molecular weight; in this instance the final number average molecular weight is 22.75% away from the desired value. This is unsurprising considering how far away the reactor temperature profile is from the offline optimized results in both instances. A cooling

water temperature of 10 and 15 °C can result in the necessary final number average molecular weight. This suggests that temperatures below 10 °C and between 10 and 15 °C will also lead to the production of the desired polymeric properties.

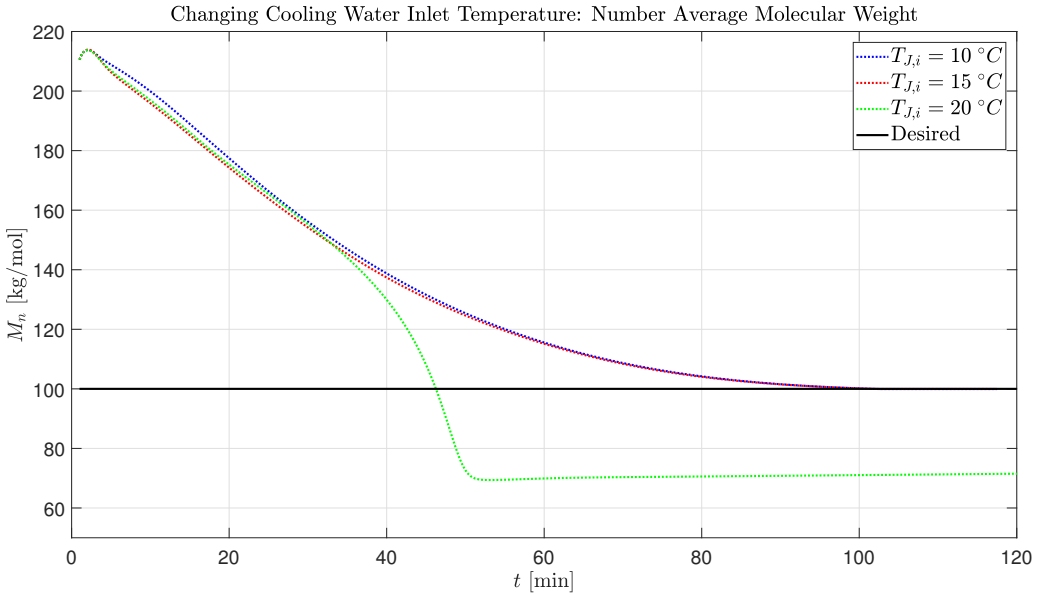


Figure 6.23: Number average molecular weights for various cooling water temperatures using a two-level advanced control structure.

The results of different cooling water temperature simulations are summarized in Table 6.5.

Table 6.5: Effect of cooling water temperature on the process outcome when the cooling constraint is 4.3 $\mathrm{kg\,s^{-1}}$.

$T_{J,i}$ [°C]	$\overline{M}_n$ [$\mathrm{kg\,mol^{-1}}$]	$\overline{M}_w$ [$\mathrm{kg\,mol^{-1}}$]	PI [-]	t_f [min]
10	99.95	179.60	1.80	102.8
15	99.96	178.95	1.79	102.6
20	77.25	236.28	2.96	388.2

The results indicate that the cooling water temperature has a meaningful impact on the system when the cooling water flow rate is near the system's limit. If the cooling capacity were unconstrained, the flow rate could compensate for this increase in temperature; however, when it is constrained too much, the system cannot track the optimal reactor temperature because the cooling capacity is insufficient. Since the optimal reactor temperature cannot be tracked, the necessary final number average molecular weight cannot be realized. The results show that the higher the cooling

water temperature for this cooling water flow rate constraint, the longer the batch time and the larger the deviation from the desired final number average molecular weight. Somewhere between 15 and 20 °C exists a limit where the system can no longer achieve optimal results for this cooling capacity.

Varying Cooling Water Inlet Temperature:

The cooling water temperature is varied at four different times throughout the process to determine where in the batch the process is most sensitive to a feed temperature change. The changes are implemented at sample 20, 100, 150, and 250; these correspond to minute 5, 37.5, 62.5, and 87.5 in the batch, respectively.

Both a step up and a step down in the cooling water temperature are tried. The plots shown here are for a maximum of 4 $\mathrm{kg\,s^{-1}}$; the plots for a cooling constraint of 4.3 $\mathrm{kg\,s^{-1}}$ can be found in Appendix C. These results show how the controller handles an increase or a decrease in the cooling water temperature for a system with a cooling constraint that is strongly active for most of the batch. First presented are the plots for a decrease in the cooling water temperature followed by the plots for an increase of the same magnitude.

Decrease in Cooling Water Temperature:

Figure 6.24 shows that the reactor temperature profiles are insignificantly affected by a -5°C step change in the cooling water temperature. This is anticipated because this system is highly exothermic so a decrease in the cooling water temperature is actually beneficial since it means less cooling fluid is needed to remove the same amount of heat due to the increased temperature gradient between the reactor and the jacket. The only noticeable deviations in the temperature profiles are seen in the middle of the batch which matches where the cooling water flow rate is at its highest and pushes against the upper limit; therefore, a decrease in the cooling water temperature translates into a decrease in the cooling water flow rate and consequently a small change in the reactor temperature. All four simulations end with the reactor temperature just under the maximum temperature limit. These results demonstrate that decreasing the cooling water temperature has a minor impact on the ability of the controller to follow the optimal reactor temperature trajectory.

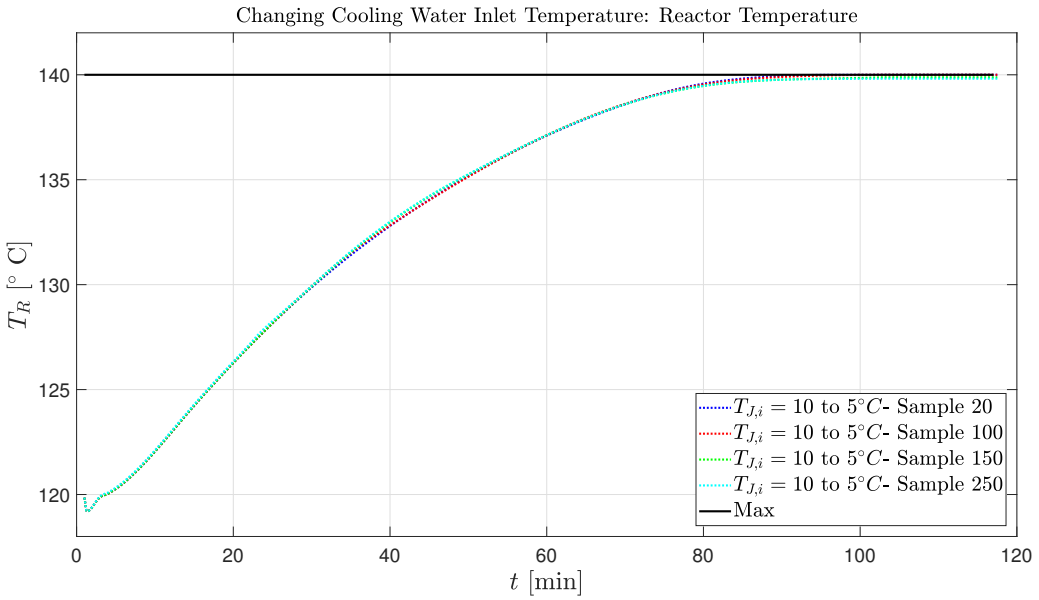


Figure 6.24: Reactor temperature profiles for a decreasing cooling water temperature using a two-level advanced control structure.

The corresponding cooling water flow rates are plotted in Figure 6.25.

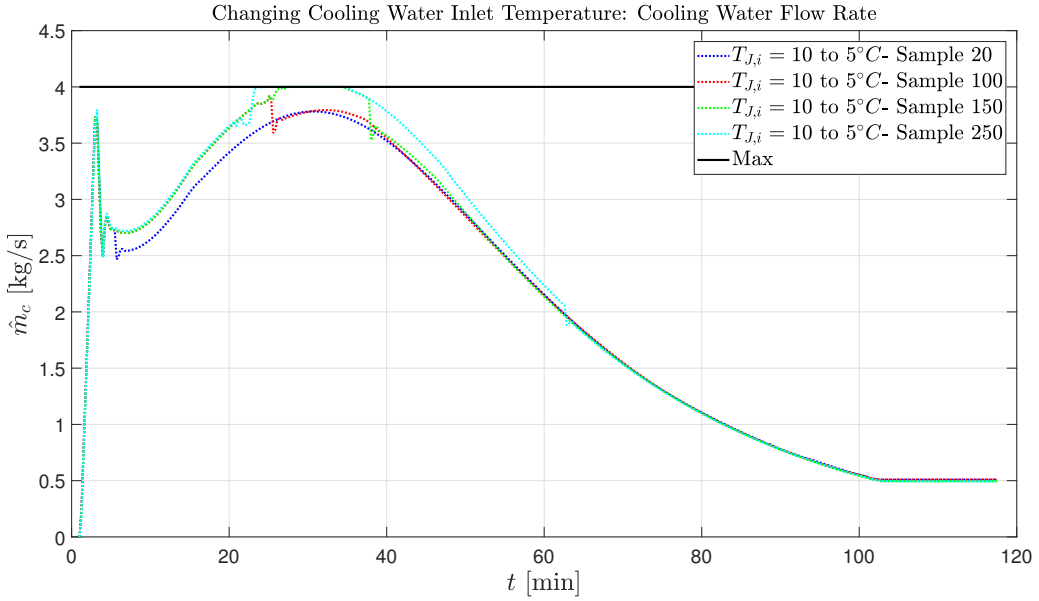


Figure 6.25: Cooling water flow rates for a decreasing cooling water temperature using a two-level advanced control structure.

The decrease in the cooling water temperature is marked by an immediate step

down in the cooling water flow rate in all cases demonstrating that the controller is quick to respond to changes in the cooling water temperature. When the change happens early in the batch, the cooling constraint is never active; for changes later in the batch, the cooling constraint is active for ten to fifteen minutes during the first half of the batch. All four profiles are identical after approximately sixty five minutes. As expected, a decrease in the cooling water temperature means that less cooling fluid is required.

Figure 6.26 shows the jacket temperature for each of the simulations. It is not impacted by the change in cooling water temperature because the controller decreases the flow rate through the jacket; consequently, the jacket temperature is essentially identical in all instances. This is unsurprising since the reactor temperature profile is hardly impacted by the change in the feed temperature. As a consequence the amount of heat transferred to the jacket will be the same; therefore, the jacket temperature should also be roughly the same.

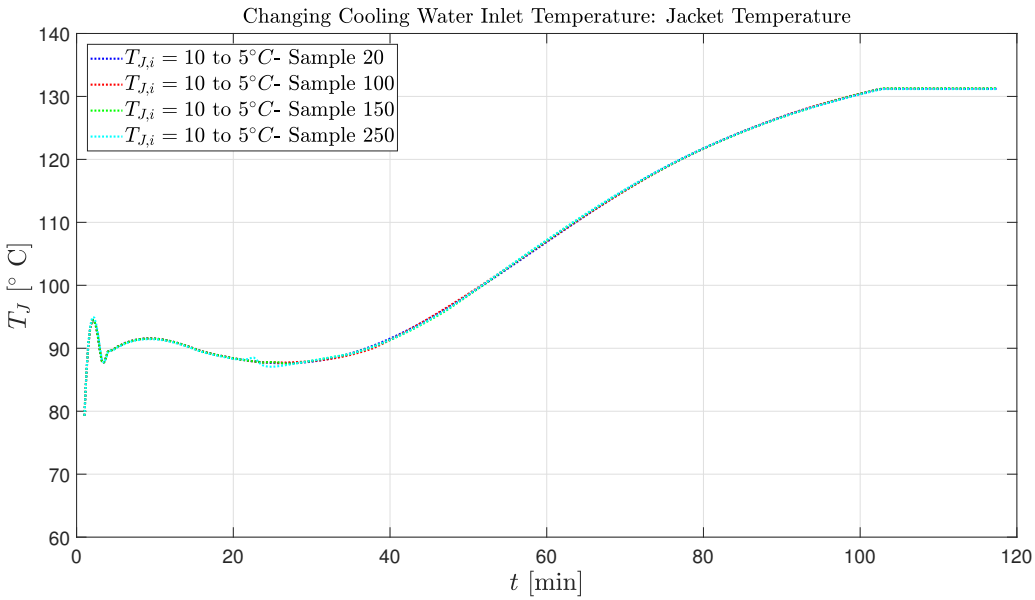


Figure 6.26: Jacket temperatures for a decreasing cooling water temperature using a two-level advanced control structure.

Figure 6.27 indicates that the number average molecular weights are unaffected by the cooling water temperature throughout the batch. All the simulations result in the essentially the same final number average molecular weight. This final number average value is nearly identical to when a constant cooling water temperature of 10 °C is used. This suggests that as long as the cooling constraint is not too restrictive, the controller can handle a decrease in the cooling water inlet temperature without impacting the polymer quality.

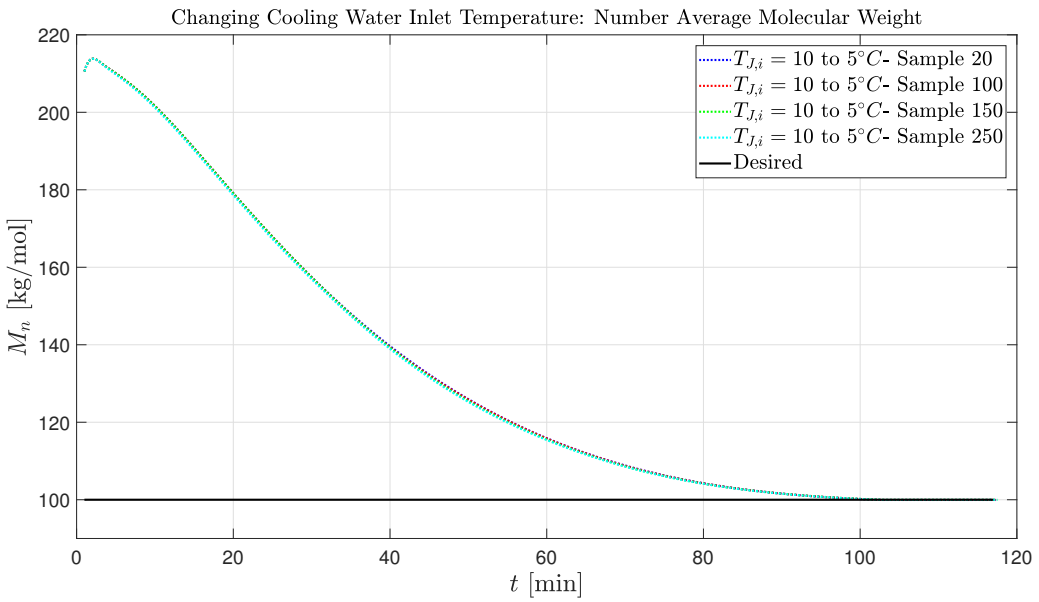


Figure 6.27: Number average molecular weights for a decreasing cooling water temperature using a two-level advanced control structure.

Similarly the batch time is unaltered by the disturbance in the cooling water temperature as illustrated in Figure 6.28. All four simulations reach the desired 80% conversion at about 103 minutes.

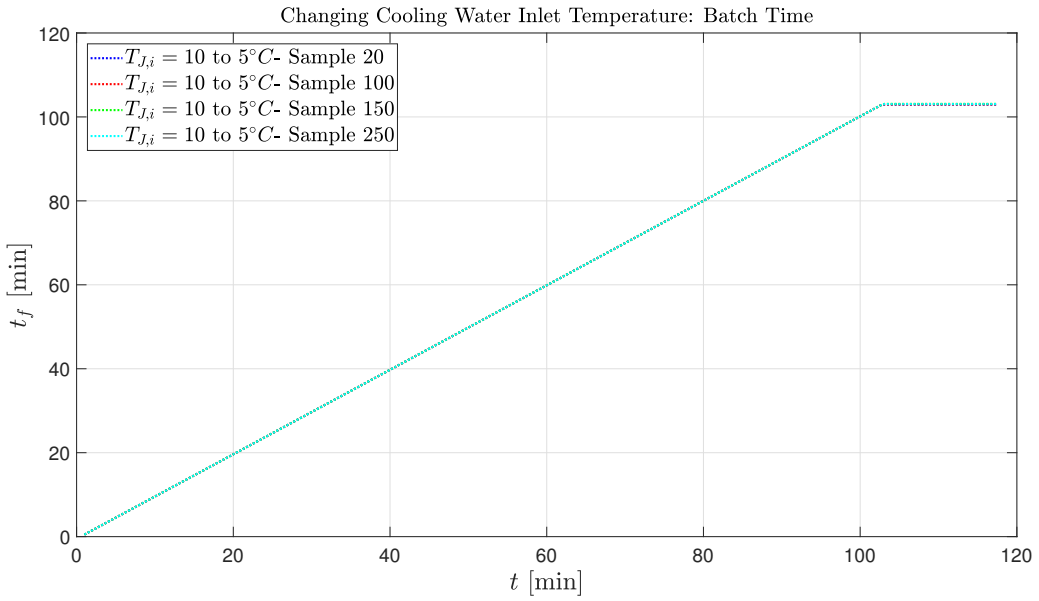


Figure 6.28: Batch times for a decreasing cooling water temperature using a two-level advanced control structure.

These simulations demonstrate that a five degree decrease in the cooling water temperature does not have a large influence on the product quality or the batch length. This suggests that the system is not very sensitive to a decrease in the cooling water temperature. It is predictable that a lower feed temperature would not have a noticeable impact on the batch outcome since a lower feed temperature helps to increase the system's cooling capacity. This does not indicate that the cooling water temperature is not a key component in determining the optimal operating conditions if the system's cooling capacity is adequate. Rather it shows that if the cooling capacity is sufficient, the system is insensitive to changes in the cooling water temperature.

Increase in Cooling Water Temperature:

The reactor temperature profiles for a +5°C step change are shown in Figure 6.29.

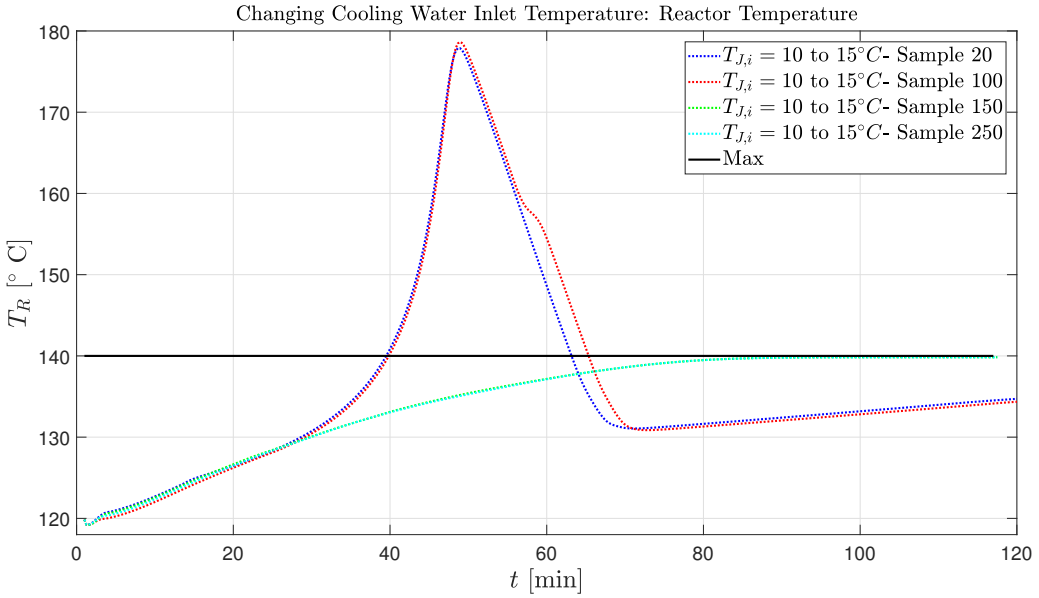


Figure 6.29: Reactor temperatures for an increasing cooling water temperature using a two-level advanced control structure.

The controller struggles to achieve the optimal reactor temperature when the cooling water temperature is increased before sample 150; the upper bound is violated by over forty degrees when the change is made at sample 20 and 100. If the cooling water temperature is increased after this, the affect is minimal. For sample 20 and 100, the profiles are comparable. When the change is made at sample 20, the cooling flow rate can increase so it isn't until the cooling capacity proves inadequate,

the reactor temperature increases; this occurs at the same point in the batch for sample 100 and causes the reactor temperature to increase swiftly. The constrained cooling capacity cannot increase to the necessary flow rates to remove the heat as needed to track the optimal reactor temperature profile. So these results do not show a failure of the controller but rather a limit of the system.

The increase in the cooling water temperature means that the required cooling water flow rates are notably higher for the first two simulated samples as illustrated in Figure 6.30. The DRTO controller calculates that the flow rate would have to be above the upper limit when the cooling water temperature changes at sample 20 or 100. When the change is made later in the batch, the effect is insignificant since the constraint is inactive and the cooling water flow rate can therefore increase to compensate.

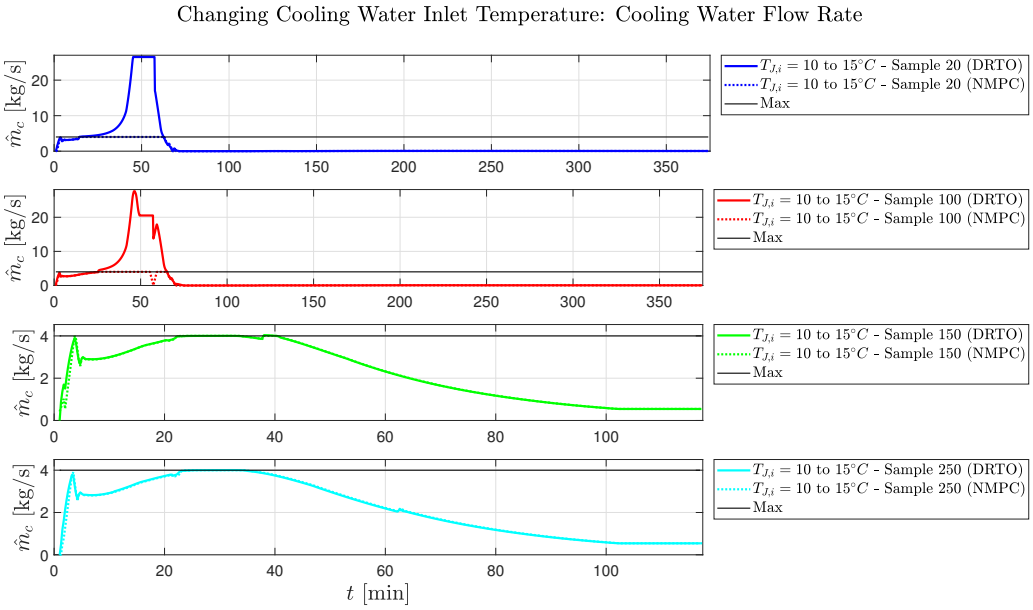


Figure 6.30: Cooling water flow rates for an increasing cooling water temperature using a two-level advanced control structure.

The jacket temperatures reflect the increased feed temperature as demonstrated in Figure 6.31. Despite the reactor temperature reaching almost 180 °C, the jacket temperature remains around 130 °C, which shows that the additional heat in the reactor is removed. The pikes in the jacket temperature for sample 20 and 100 are a result of the reactor temperature changes.

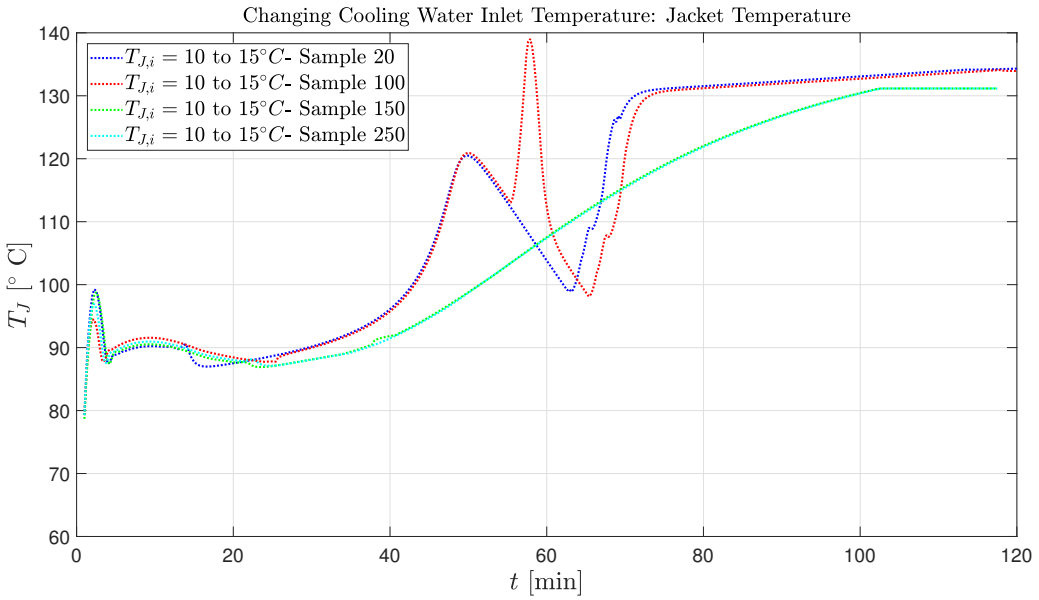


Figure 6.31: Jacket temperatures for an increasing cooling water temperature using a two-level advanced control structure.

Figure 6.32 illustrates that the number average molecular weights are affected when the disturbance occurs early in the batch.

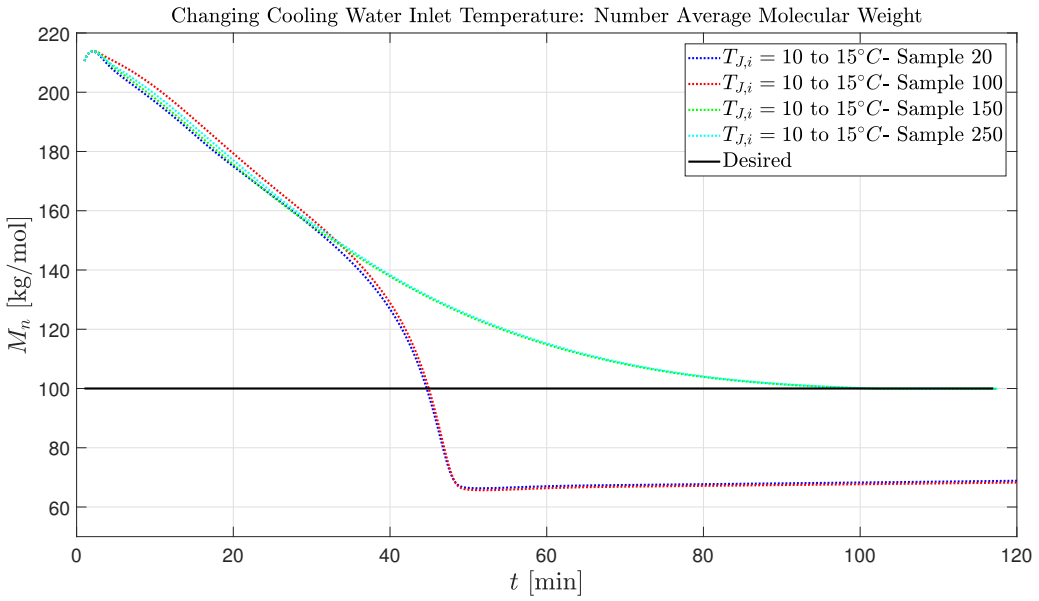


Figure 6.32: Number average for an increasing cooling water temperature at various points in the batch.

When the cooling water temperature increases early in the batch, the controller is unable to achieve the desired product quality. In fact, the final number average is almost at the minimal allowed value of 60 kg/mol in these two cases. If the increase occurs at sample 150 or 250, the desired final number average molecular weight is within the allowed range.

The batch times are shown in Figure 6.33 and illustrate that when the reactor temperature profile is extremely different from the optimal one, the batch time is extended considerably. For an increase in the cooling water temperature early in the batch, the batch time three times longer than the predicted time. When the increase occurs later in the batch, there is enough cooling capacity for the system to handle the disturbance so the batch time is unaffected and is around the 103 minutes.

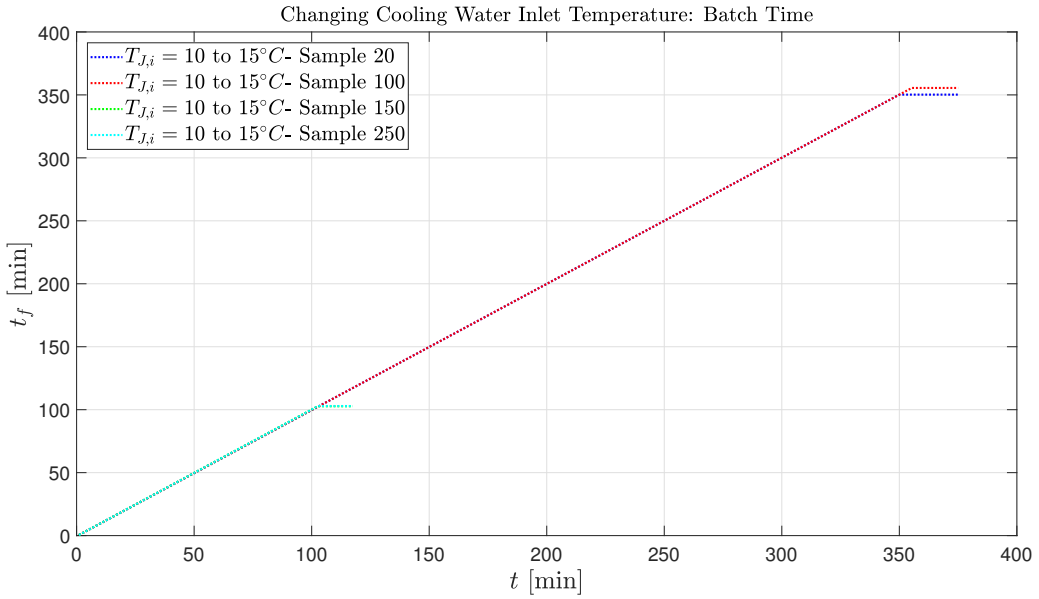


Figure 6.33: Batch times for an increasing cooling water temperature using a two-level advanced control structure.

The results suggest that an increase in cooling water temperature by 5 °C, has a large impact on the system if this change occurs early in the batch. If the disturbance occurs at sample 20 or 100 the desired results cannot be achieved, illustrating that the system is most sensitive to a temperature increase early in the production; this is where the most cooling water is required to track the optimal reactor temperature and the system cannot compensate for the increased cooling water temperature because of the constraint. Therefore, before implementing this control structure on a real system, the cooling constraint upper limit should be increased to include a safety margin so that the controller can handle disturbances such as this at any

point in the process.

Comparison of Results:

The results for the simulations are summarized in Table 6.6 and includes the values for a cooling constraint of 4.3 $\mathrm{kg\,s^{-1}}$. The plots for a cooling constraint of 4.3 $\mathrm{kg\,s^{-1}}$ can be found in Appendix C Section C.2.1.1.

For a cooling constraint of 4.0 $\mathrm{kg\,s^{-1}}$, the system is extremely sensitive to temperature increases at the beginning of the batch as seen by the increased batch times and the large deviation from the desired final number average molecular weight. In comparison, when the constraint is 4.3 $\mathrm{kg\,s^{-1}}$, the system is able to handle the temperature increase at any point without a notable impact on the batch time and without sacrificing the product quality. When the cooling water temperature decreases, there is little effect on the batch outcome for either simulated cooling constraint, which is unsurprising.

Table 6.6: Step change in the cooling water temperature at various points in the batch.

$\hat{m}_c$ [$\mathrm{kg\,s^{-1}}$]	$T_{J,i}$ [°C]	**Sample**	$\overline{M}_n$ [$\mathrm{kg\,mol^{-1}}$]	$\overline{M}_w$ [$\mathrm{kg\,mol^{-1}}$]	PI [-]	t_f [min]
4.0	$10 \rightarrow 5$	20	99.94	179.98	1.80	102.9
4.0	$10 \rightarrow 5$	100	99.94	179.94	1.80	103.0
4.0	$10 \rightarrow 5$	150	99.97	179.90	1.80	103.1
4.0	$10 \rightarrow 5$	250	99.96	179.76	1.80	103.1
4.0	$10 \rightarrow 15$	20	74.59	230.05	3.08	350.2
4.0	$10 \rightarrow 15$	100	73.93	231.02	3.12	355.5
4.0	$10 \rightarrow 15$	150	99.89	179.11	1.79	102.7
4.0	$10 \rightarrow 15$	250	99.94	179.44	1.80	102.9
4.3	$10 \rightarrow 5$	20	99.94	179.24	1.79	102.6
4.3	$10 \rightarrow 5$	100	99.95	179.60	1.80	102.8
4.3	$10 \rightarrow 5$	150	99.95	179.60	1.80	102.8
4.3	$10 \rightarrow 5$	250	99.94	179.60	1.80	102.8
4.3	$10 \rightarrow 15$	20	99.94	179.44	1.80	102.9
4.3	$10 \rightarrow 15$	100	99.86	177.33	1.78	102.1
4.3	$10 \rightarrow 15$	150	99.91	179.54	1.80	103.0
4.3	$10 \rightarrow 15$	250	99.95	178.46	1.79	102.3

Chapter 7
Results and Discussion

The goal of this work was to develop a two-level advanced control structure consisting of a NMPC level and a DRTO level for the semi-batch production of EPS. The control structure was constructed in the previous two chapters and tested for two types of disturbances. In this chapter, the three different stages of development are compared in Section 7.1 to illustrate how the reactor temperature profile, batch times and final number average molecular weights are similar for each stage.

The construction of a process model and a two-level advanced control structure is a time consuming and costly process, which is why many companies in the polymer industry have not bothered to change from the current approach. In industry the common approach for batch or semi-batch reactors is to perform offline optimization for a worst case scenario to develop a fixed recipe that is used in the production of every single batch. This fixed recipe approach means that the reactor conditions are not altered even if operating conditions are changed. Consequently, this method has its limitations in its ability to handle disturbances in the operating conditions. This is where the two-level online control structure has an advantage. To motivate the development and use of an advanced control structure over the use of a fixed recipe, a cost benefit analysis is performed in Section 7.2.

7.1 Result Comparison

The offline optimization, the NMPC layer following a fixed reference reactor temperature, and the two-level control structure are compared. Recall that the three different stages of development have some differences. For instance, the offline optimization assumes perfect temperature control so the cooling water flow rate is not calculated and consequently, the cooling capacity is assumed infinite. While perfect temperature is not assumed in the NMPC layer, the cooling capacity is still essentially infinite since the constraint is set to a large number such that it is never active. A restrictive limit on the cooling capacity is added when the NMPC and DRTO layers are combined.

Figure 7.1 shows that the offline optimization and the NMPC favor a more rapid increase in the reactor temperature than the full controller. The offline and NMPC only stage having practically identical profiles is expected since the NMPC only

layer tracks the optimal reactor temperature trajectory determined by the offline stage. The two layer controller produces a reactor temperature profile with some small deviation from the NMPC only layer. This demonstrates that for comparable operating settings the three stages produce nearly identical reactor temperature profiles. It is interesting that the two layer controller has a tighter cooling constraint than the other layers, but is still able to follow the optimal trajectory. This shows that if the cooling water flow rate is not constricted to the point where the cooling capacity is insufficient, the controller can realize the optimal reactor temperature profile.

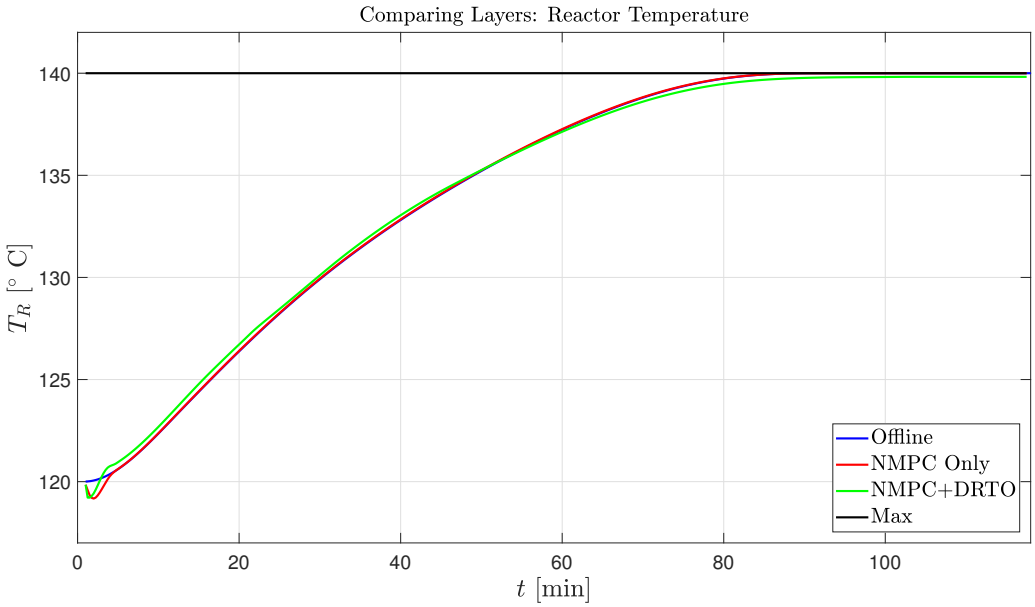


Figure 7.1: Comparison of the reactor temperature profiles for the three different design stages.

The jacket temperature is substantially different for the offline optimization compared to the other two stages as Figure 7.2 illustrates; this is because there is no cooling fluid present in the offline optimization. Therefore, the jacket follows the reactor temperature since no heat is removed from the system. For the other two stages, the cooling water flow rate is an optimized variable so the jacket temperature is cooled as heat is removed. The NMPC only and the two level control structure have analogous jacket temperature profiles, which is unsurprising considering the similarities between the reactor temperature and cooling water flow rate profiles.

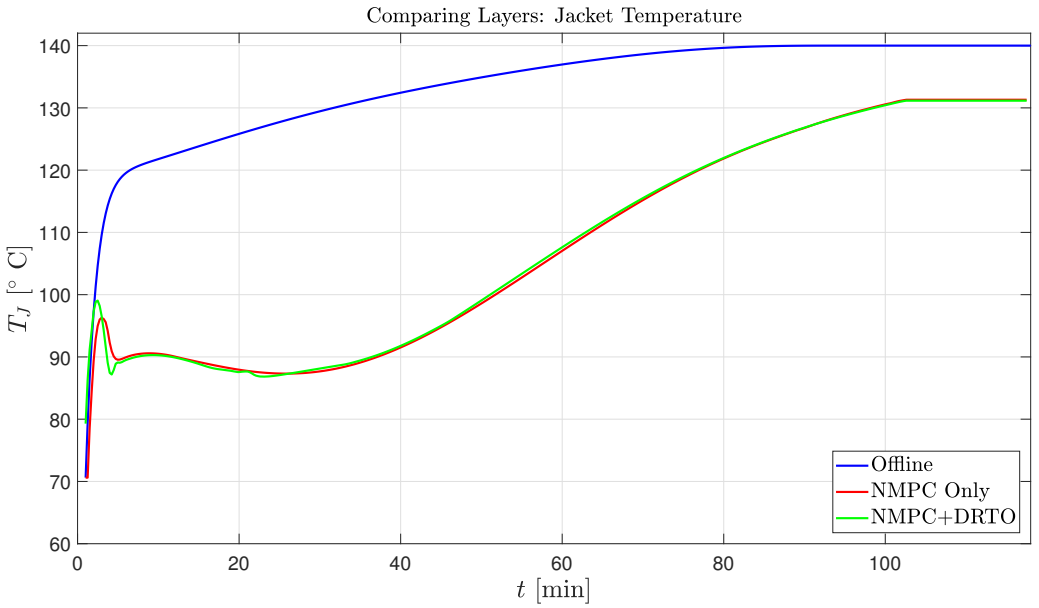


Figure 7.2: Comparison of the jacket temperature profiles for the three different design stages.

Figure 7.3 shows the cooling water flow rates for two of the stages.

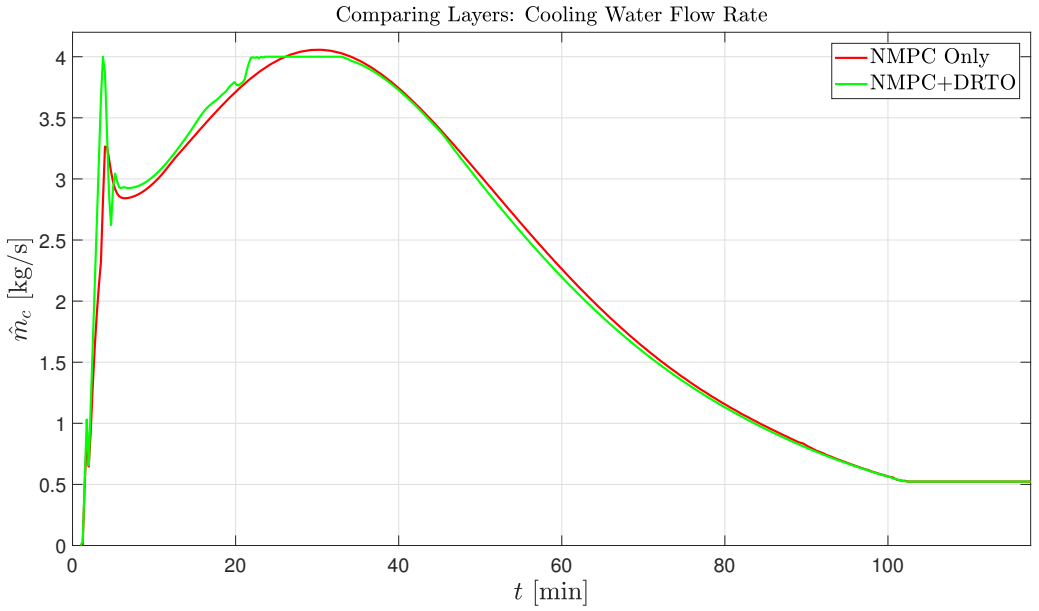


Figure 7.3: Comparison of the cooling water flow rates for the two of the design stages.

Recall that the offline optimization does not calculate the optimal cooling water flow rate so there is nothing to plot for this stage. This plot shows the cooling water flow rates calculated and executed by the NMPC layer. The NMPC only layer has a higher flow rate than the combination stage because the constraint on the two-level cooling water flow rate is more restrictive. It is interesting to notice that the system dynamics change slightly when a tighter cooling constraint is implemented as demonstrated by the differences in the initial cooling water flow rates. There is a larger spike for the combination stage at the beginning of the batch in comparison to the NMPC only layer.

The number average molecular weights throughout the batch are shown in Figure 7.4. Despite the differences in the controller in each of the three stages, the final number average molecular weights are within the desired range of 100 ± 1.0 $\mathrm{kg\,mol^{-1}}$.

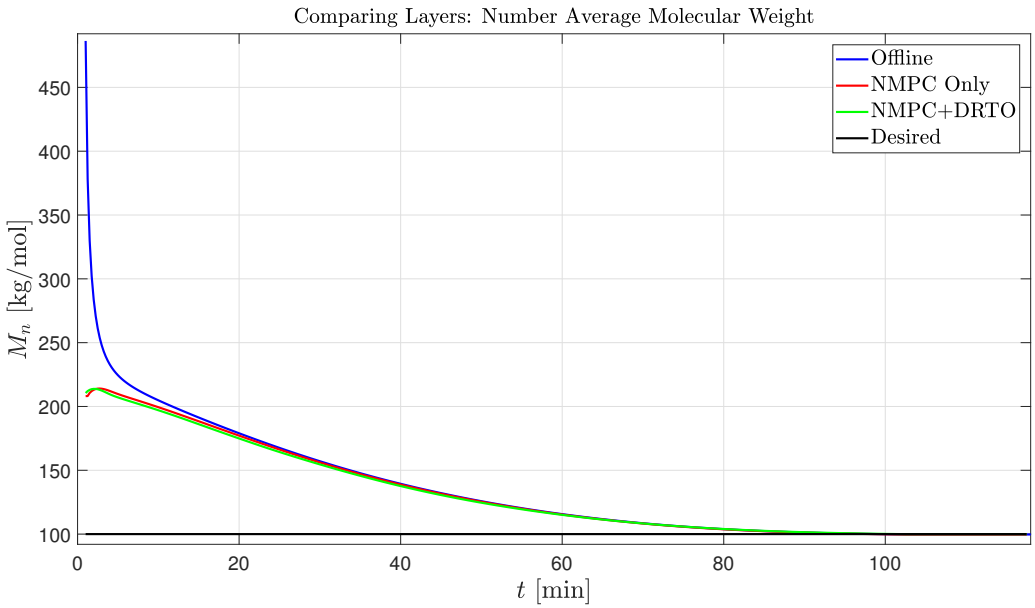


Figure 7.4: Comparison of the number average molecular weight distributions for the three different design stages.

Figure 7.5 illustrates the consumption rates of the initiator throughout the batch. The rates of consumption are similar for all three stages, consequently the amount of initiator consumed in each instance is almost indistinguishable. This means that little initiator is wasted and since initiator can be one of the most expensive components in polymer production, this translates into savings.

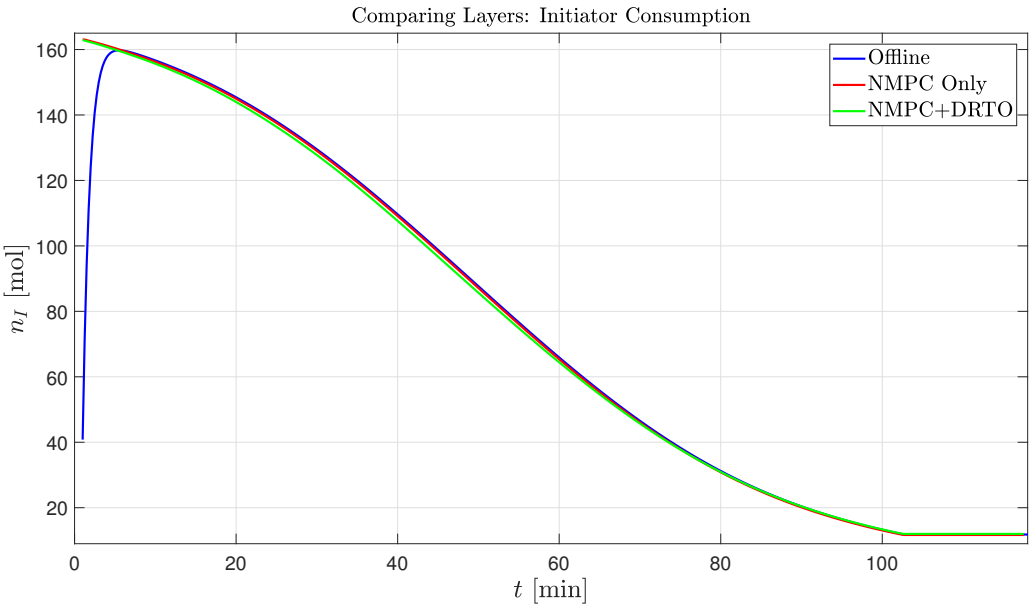


Figure 7.5: Comparison of the initiator consumption rate for the three different design stages.

Table 7.1 presents the values for a cooling water inlet temperature of 10 °C from all the different stages of development for comparison. The results illustrate that the offline optimization stage predicts an outcome not far from the optimal operating conditions. The full control structure demonstrates the effect of adding a tight constraint on the cooling water flow rate; when the optimal reactor temperature trajectory is calculated online, the batch time is a little over two minutes longer than the offline results predict. It is positive that the results are similar because the same model is used in each case and therefore, the outcome should be comparable.

Table 7.1: Comparison of results for each stage of development.

Simulation	t_f [min]	$\overline{M}_n$ [kgmol^{-1}]	$\overline{M}_w$ [kgmol^{-1}]	PI [-]
Offline Optimization	102.3	99.71	180.40	1.81
NMPC	102.3	99.67	179.05	1.80
DRTO + NMPC	102.7	99.95	179.07	1.79

Comparing the three stages of development shows that for similar operating conditions the offline results do not measurably differ from the trajectories realized using online optimization. However, this does not suggest that that there is no additional benefit to using online optimization, it simply means that for ideal cases where the offline and actual operating conditions are identical, the benefit is small. It is

uncommon for operating conditions to be constant and there are often disturbances that have to be accounted for in a real process such as cooling water temperature or the cooling capacity of the system. This scenario is explored in the next section where a cost benefit analysis is performed to show how online optimization is beneficial when the operating conditions differ from those used in the offline optimization. What the values in the table illustrate is that the addition of the cooling constraint in this case has a minor impact on the batch outcome since the offline results are unconstrained but the full structure has an active constraint. This suggests that as long as the cooling capacity is sufficient, a constraint on the cooling capacity does not have a large impact on the batch time or product properties. On the other hand, if the constraint is severe enough that the system can no longer cool the reactor fast enough to track the optimal reactor temperature trajectory, the constraint has a huge effect on the outcome.

7.2 Cost-Benefit Analysis

In industry it is common practice to use a fixed recipe for batch and semi-batch production; this recipe is determined by performing offline optimization for worst case scenarios so that the process is always operated with a safety margin. However, this fixed recipe does not account for changes in the operating conditions. This translates into the batch being run with non-ideal operating conditions. Therefore, using a fixed recipe approach can lead to a loss in profit. This is where an advanced control structure like the one developed in this work can prove beneficial.

The offline recipe is now compared to the use of a full control structure for different operating conditions. The simulation results are intended to demonstrate how the two-levels work together to handle any disturbances to the system; these values can then be compared to the fixed recipe. Based on previous simulation results, a cooling flow rate upper bound for the offline optimization is selected to be 4.5 $\mathrm{kg\,s^{-1}}$; this value is chosen so that the system can handle the maximum allowed cooling water inlet temperature of 20 °C. A heat transfer coefficient of 800 $\mathrm{W\,m^{-1}\,K^{-2}}$ is used for a dirty reactor, the average reactor represents the more common operation condition with a heat transfer coefficient of 900 $\mathrm{W\,m^{-1}\,K^{-2}}$, and a clean reactor uses a value of 1000 $\mathrm{W\,m^{-1}\,K^{-2}}$. Three different cooling capacities are tested for the full control structure: 4.0 $\mathrm{kg\,s^{-1}}$ (Low), 4.5 $\mathrm{kg\,s^{-1}}$ (Medium), and 5.0 $\mathrm{kg\,s^{-1}}$ (High). Two different cooling water temperatures are simulated as well: 10 °C (Cold) and 20 °C (Hot). A figure illustrating the full list of different simulations that are run is shown in Appendix C Section C.3.

The DRTO level used here is altered to determine the optimal initial starting reactor temperature and the optimal monomer to initiator ratio in addition to the optimal reactor temperature profile. The calculation of the optimal initial starting reactor temperature and monomer to initiator ratio is only performed at the first

sample point; this means that $T_{R,0}$ and n_I are only decision variables in the first optimization step. This is similar to what was done in the online optimization problem formulation.

7.2.1 Offline vs Online Optimization Results

Here all of cold cooling water inlet temperature simulations are compared to the offline recipe; the other results are shown in Appendix C. These simulations are selected because the results best illustrate the benefit of using an advanced control structure over a fixed recipe.

7.2.1.1 High Cooling Capacity and Cold Cooling Water Temperature

The three different reactor conditions (clean, average, and dirty) are now tested using a high cooling capacity, meaning that the upper limit of the cooling water flow rate is 5.0 $\mathrm{kg s^{-1}}$, and a cold cooling water temperature of 10 °C. The results are juxtaposed to the offline recipe, which is determined for a dirty reactor with a cooling flow rate of 4.5 $\mathrm{kg s^{-1}}$ and a cooling water temperature of 20 °C.

Figure 7.6 shows the initiator consumption for each of the simulations.

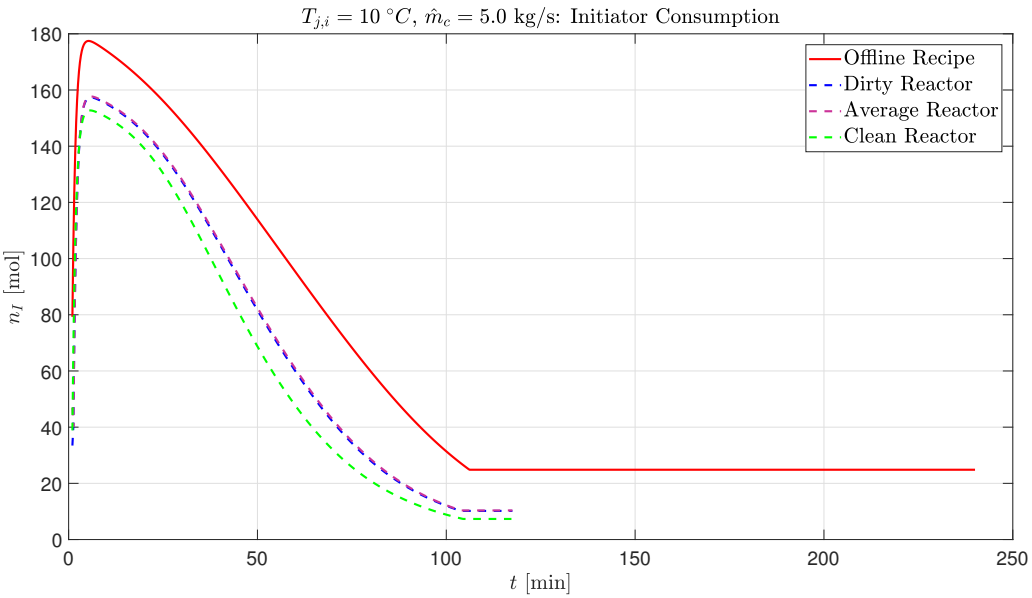


Figure 7.6: Initiator profiles for a high cooling capacity and cold cooling water temperature.

The optimal initiator to monomer ratios for a clean, average, and dirty reactor are 700.7, 678.5, and 679.7, respectively. All three online simulations have larger ratios than the offline recipe which has a ratio of 600.82. For a clean reactor the ratio is 116% larger, meaning that a great deal of initiator would be wasted; a dirty and

average reactor have an optimal ratio about 113% larger than the offline recipe. Using the online optimal monomer to initiator ratios for this scenario could result in a considerable annual savings in initiator cost.

For these operating conditions, the three online simulations generate reactor temperature profiles that are distinct from the offline recipe as illustrated in Figure 7.7. The optimal initial reactor temperatures are: 132.4 (clean), 128.7 (average), and 132.5 °C (dirty); these are all higher than the offline recipe which has an optimal starting temperature of 125.6 °C. From these initial optimal temperatures, the reactor cools down to be more similar to the offline results. This suggests that the current controller settings for calculating the optimal initial temperature may require further tuning. However, this value does not affect the reactor temperature profile overall or the batch outcome. All three profiles increase at a more rapid rate than the predicted offline results. In this scenario, the dirty and average reactor have comparable profiles; the clean reactor has a similar trend but increases at a more rapid rate. By the end of the batch, all three profiles have reached the same temperature just below the upper limit. In contrast, the offline recipe ends at a lower reactor temperature.

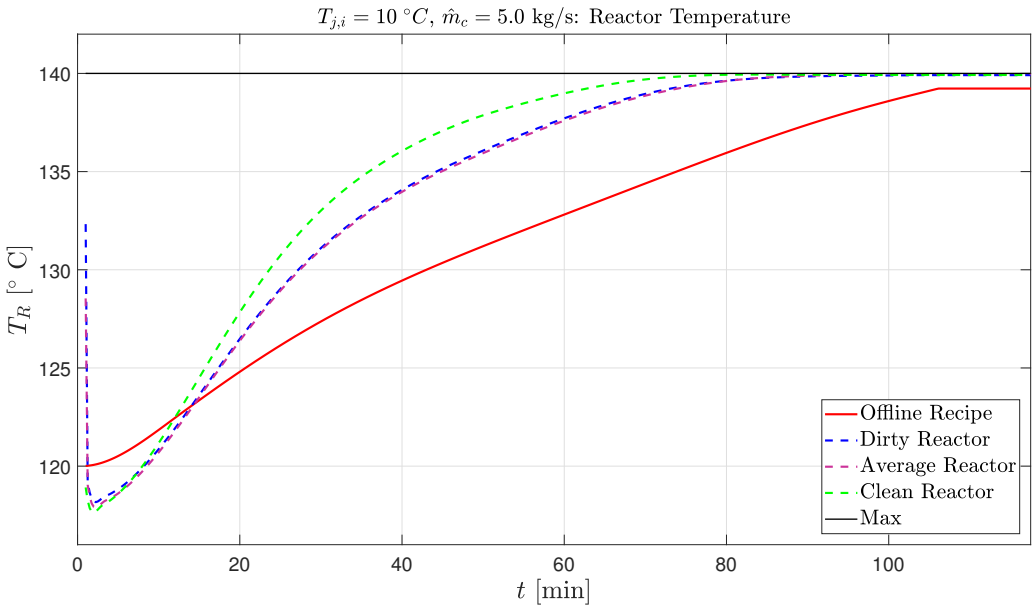


Figure 7.7: Reactor temperature profiles for a high cooling capacity and cold cooling water temperature.

Figure 7.8 reveals that a clean reactor does not always have the smallest flow rate. Between minutes 20 and 40, the flow rate for the clean reactor increases above the flow rate for the average reactor; this is due to the quicker increase in the reactor

temperature for the clean reactor. Overall the three online profiles have a similar shape but differ from the offline recipe, which again under predict the cooling water flow rate.

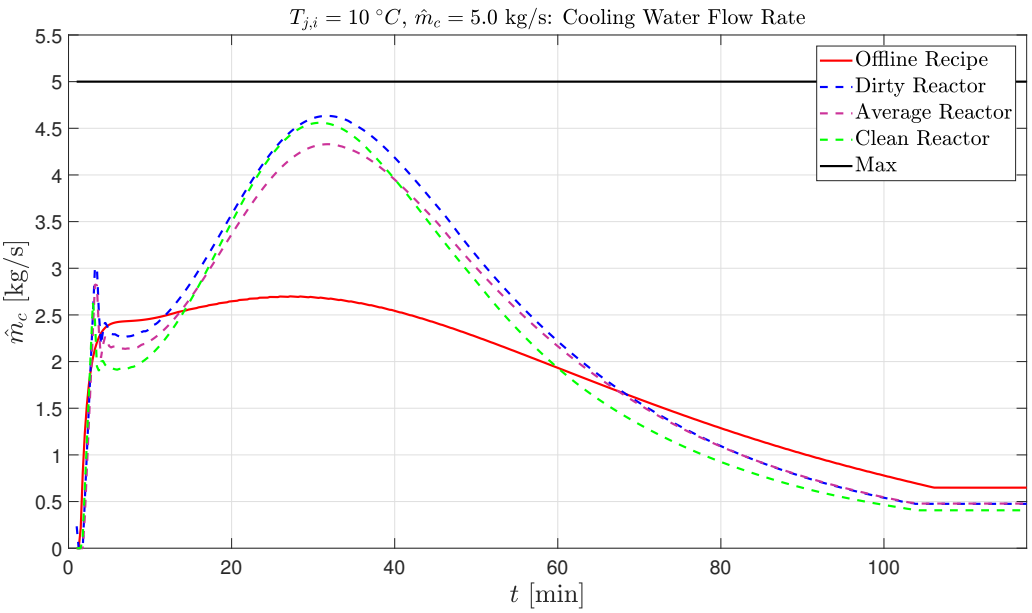


Figure 7.8: Cooling water flow rates for a high cooling capacity and cold cooling water temperature.

The final number average molecular weight for the three online cases are indistinguishable and improved compared to the offline recipe as shown in Figure 7.9. The offline recipe predicts a final number average molecular weight of 99.53 $\mathrm{kg\,mol^{-1}}$ whereas the clean, average, and dirty online simulations give a final number average molecular weight of 99.97, 99.96, and 99.96 $\mathrm{kg\,mol^{-1}}$, respectively. All four cases produce a polymer within the accepted range for the final number average molecular weight.

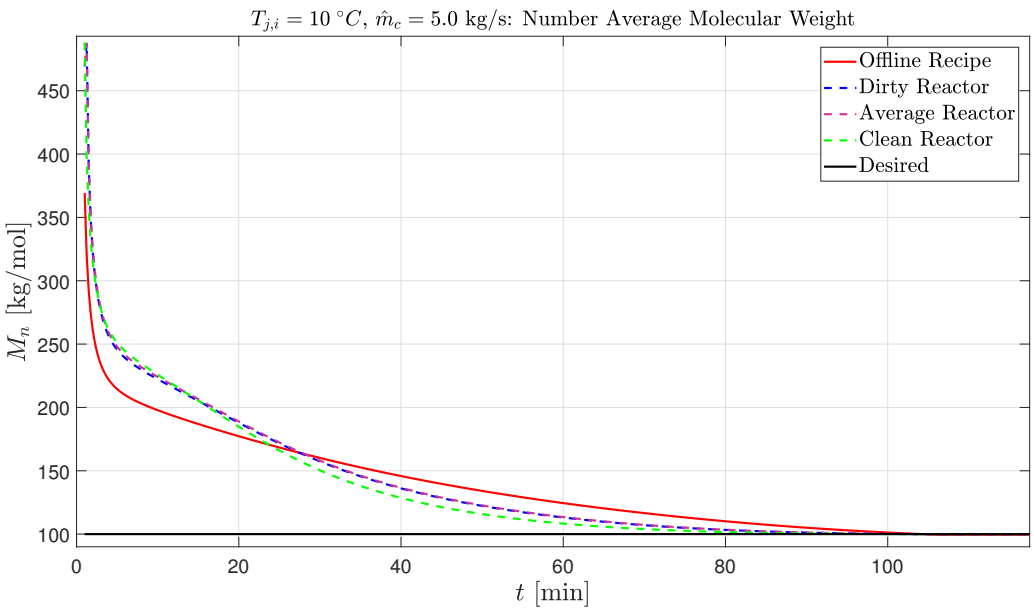


Figure 7.9: Number average molecular weights for a high cooling capacity and cold cooling water temperature.

Figure 7.10 shows the batch times for each scenario with the longest batch being the offline recipe at 106 minutes.

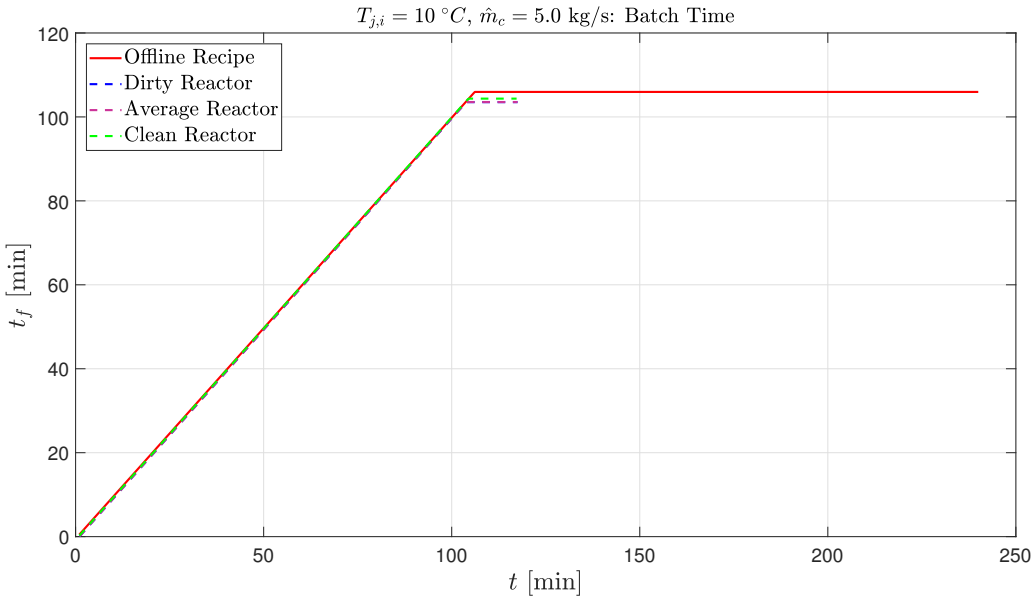


Figure 7.10: Batch times for a high cooling capacity and cold cooling water temperature.

The shortest batch is actually for the dirty reactor with the average reactor being second and the clean reactor being third; the differences between the batch times are all less than one minute. Compared to the offline recipe, the online optimization would result in up to an additional 120 batches per year per reactor. If there are several batch reactors being run in series, this could result in a noticeable increase in profit.

For the considered case of a high cooling capacity and a cold cooling water temperature, the full control structure produces an improved polymer quality and gives shorter batch times. Therefore, using the fixed offline recipe in this instance would cause a loss in product purity and give an increased batch time which would result in a loss in profit.

7.2.1.2 *Medium Cooling Capacity and Cold Cooling Water Temperature*

The three different reactor conditions (clean, average, and dirty) are now tested for a medium cooling capacity, meaning that the upper limit of the cooling water flow rate is 4.5 $\mathrm{kg s^{-1}}$, and a cold cooling water temperature of 10 °C. The results are juxtaposed to the offline recipe which is determined for a dirty reactor with a larger cooling flow rate of 4.5 $\mathrm{kg s^{-1}}$ and a cooling water temperature of 20 °C.

Figure 7.11 illustrates how the initiator consumption and optimal amounts differ for the four different scenarios.

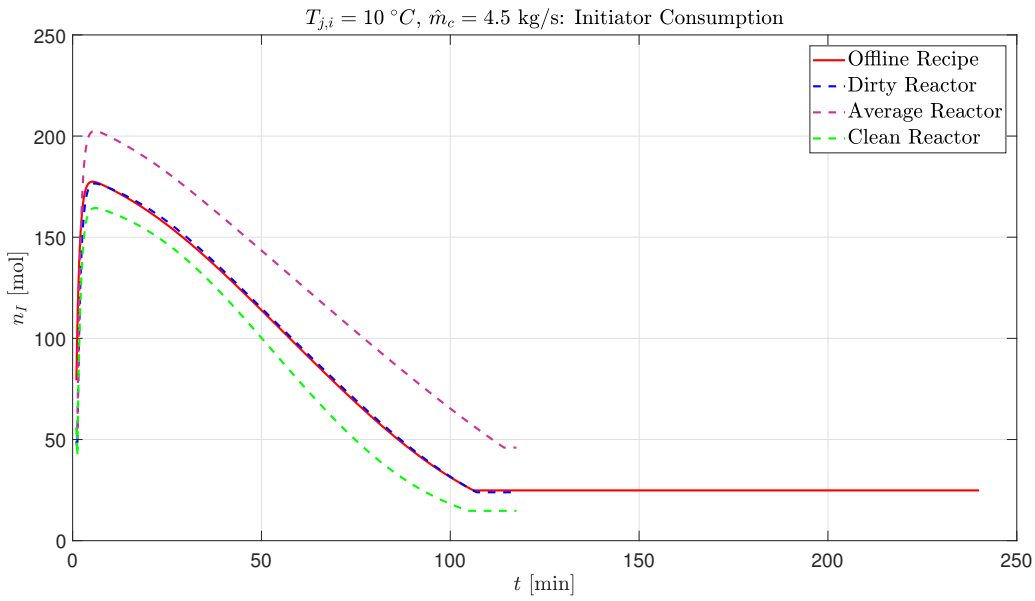


Figure 7.11: Initiator consumption for a medium cooling capacity and cold cooling water temperature.

The optimal monomer to initiator ratios are calculated to be: 651.30 (clean), 527.81 (average), and 605.76 (dirty). The clean and dirty reactors have larger monomer to initiator ratios compared to the offline recipe; therefore, in these two circumstances, following the offline recipe would result in an excess of initiator and ergo, money being wasted. On the other hand, the average reactor has a lower monomer to initiator ratio so insufficient initiator would be used in this instance if the offline recipe is followed; for this reason, the desired conversion will not be achieved and the batch would have to be thrown out.

The reactor temperature profiles for these operating conditions are surprising since the average reactor temperature profile is not in between the clean and dirty reactor temperature profiles as shown in Figure 7.12.

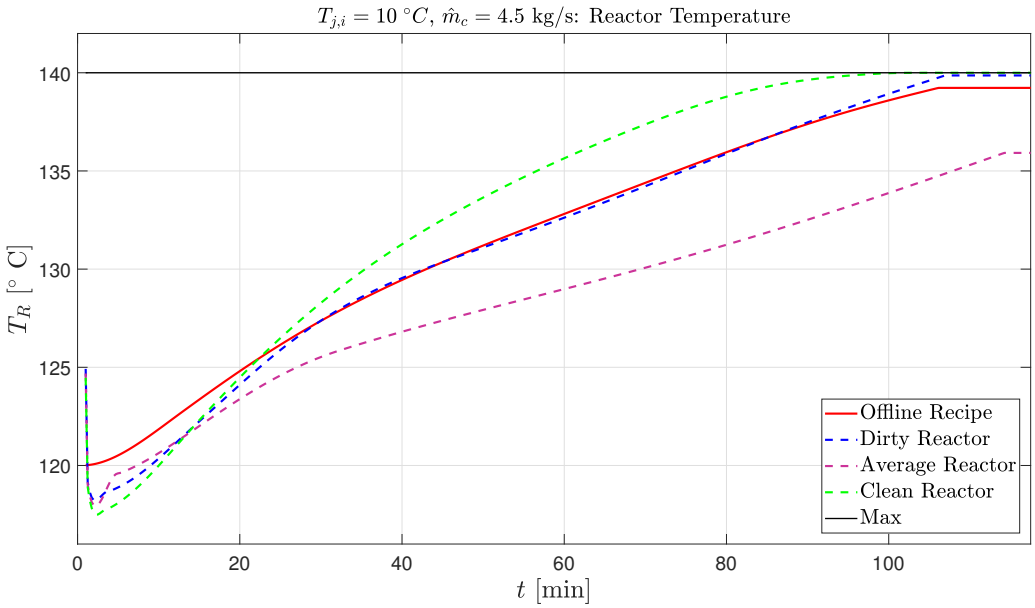


Figure 7.12: Reactor temperatures for a medium cooling capacity and cold cooling water temperature.

The optimal starting reactor temperatures for the online simulations are 124.5 (clean), 124.7 (average), and 124.9 °C (dirty). The offline recipe has an optimal starting temperature of 125.6 °C so all of the online optimal starting temperatures are lower. The average reactor has a lower temperature profile than the other reactor states; it is possible that these operating conditions result in an optimization problem has a flat area where the algorithm struggles to locate the true local minimum. This could explain why the profile is not in between the clean and dirty reactor as expected. The dirty reactor has an almost indistinguishable profile from the offline optimization profile after thirty minutes, which suggests that it would be

fine to use the offline recipe but would not be optimal for a clean or average reactor A clean reactor has a faster increasing reactor temperature and reaches the upper constraint around 90 minutes into the batch.

The reactor temperature profiles' relationship is reflected in the cooling water flow rates, as shown in Figure 7.13. A clean reactor has the lowest cooling water flow rate, which is unsurprising because the heat transfer coefficient is the highest for a clean reactor so less cooling water is required to remove the same amount of heat. A dirty reactor has a higher cooling water flow rate than a clean reactor because the heat transfer coefficient is lower. It is surprising to see that the cooling water flow rate profiles for a clean and dirty reactor are identical after about forty five minutes since the reactor temperature profiles are different. For the average reactor the initial flow rate is noticeably higher than the dirty and clean reactor flow rates, which explains why the reactor temperature is lower. These results further demonstrate how the operating conditions for an average reactor are unexpected. As is the case in when the cooling capacity is high, the offline recipe predicts a low required cooling water flow rate; this is due to the fact that the DRTO layer is not constructed to optimize the flow rate so its prediction for the flow rate is rather poor.

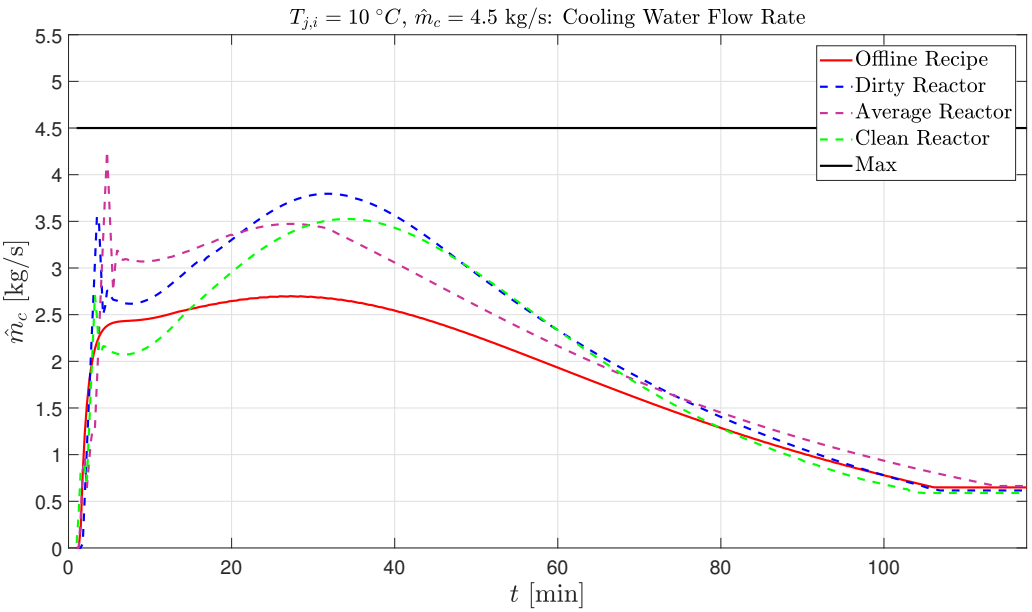


Figure 7.13: Cooling water flow rates for a medium cooling capacity and cold cooling water temperature.

Figure 7.14 illustrates that the best final number average molecular weight is achieved for the clean reactor with a final number average molecular weight of 99.94 $\mathrm{kg\,mol^{-1}}$. The offline recipe results in the "worst" final number average

molecular weight at 99.53 $\mathrm{kg\,mol^{-1}}$ but even this would be an acceptable result. However, all four conditions produce a final number average molecular weight within the allowed range. This results show that all of these operating conditions are able to produce the required polymer quality.

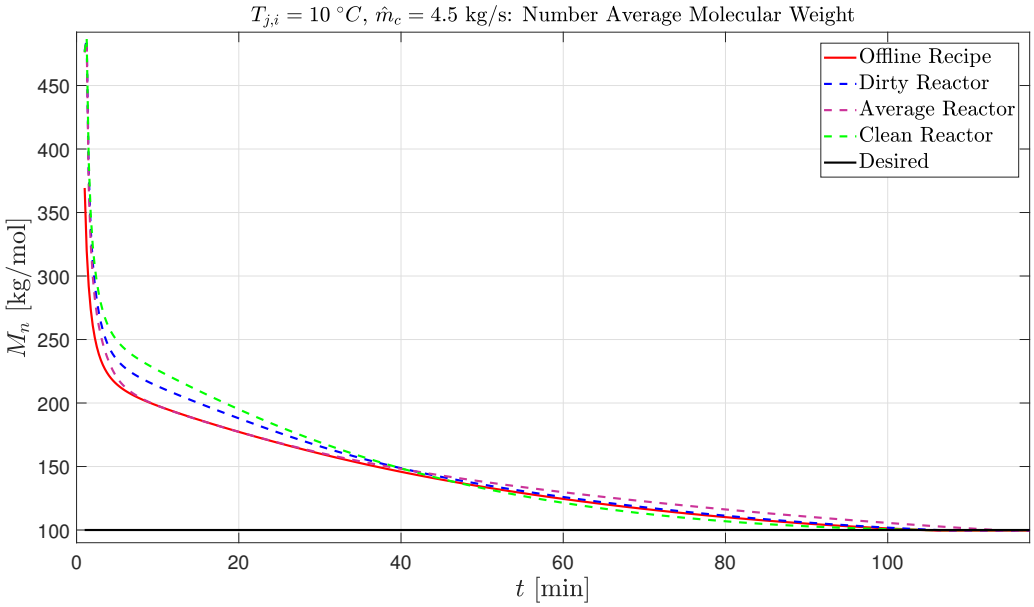


Figure 7.14: Number average molecular weights for a medium cooling capacity and cold cooling water temperature.

The shortest batch time is for the clean reactor at 104.5 minutes, followed by the offline recipe at 106 minutes, the dirty reactor at 106.6 minutes, and finally the average reactor at 113.9 minutes, as shown in Figure 7.15. The average reactor has the longest batch time because the calculated optimal reactor temperature is so different from the others. Because it is colder than the other reactors, it takes longer to reach the desired conversion. If the offline recipe is used for the average scenario, the batch would be terminated about eight minutes prematurely before the desired conversion of 80% is reached. The product would therefore not be at the required number average molecular weight and the batch would have to thrown out. Using the offline recipe for the dirty reactor would be okay since the batch times are nearly identical but for the clean reactor, the batch time is extended by almost two minutes; this would result in a loss of up to 71 batches per year per reactor.

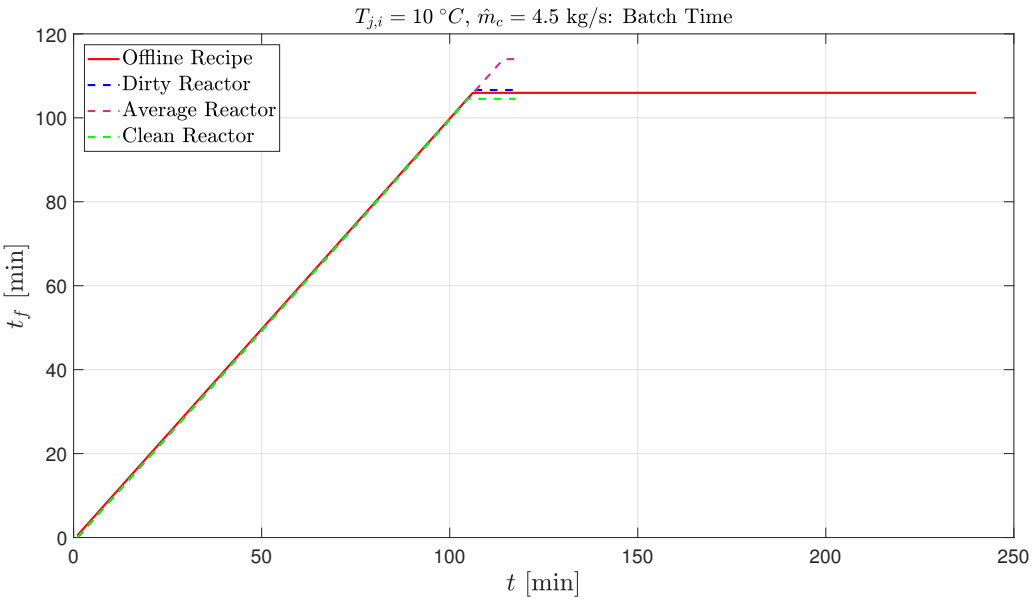


Figure 7.15: Batch times for a medium cooling capacity and cold cooling water temperature.

For a medium cooling capacity and a low cooling water temperature, the full control structure is an improvement in all cases except a dirty reactor. Because the controller struggles with the average reactor conditions, it may seem beneficial to use the offline recipe in this instance; however, if any disturbances were to occur, without the use of an online controller, the system could not respond. Therefore, in this scenario, the full control structure proves superior in its ability to adapt to the different operating conditions.

7.2.1.3 Low Cooling Capacity and Cold Cooling Water Temperature

Using a low cooling capacity, meaning that the upper limit of the cooling water flow rate is 4.0 $\mathrm{kg\,s^{-1}}$, and a cold cooling water temperature of 10 °C, the three different reactor conditions are now simulated. The results are then juxtaposed to the offline recipe.

Figure 7.16 shows the optimal monomer to initiator ratios for this operating conditions: 646.72 (clean), 648.72 (average), and 458.54 (dirty). Compared to the offline recipe ratio of 600.82, the clean and average reactors have a higher ratio; therefore, initiator would be wasted if the offline recipe is followed. For a dirty reactor, the ratio is lower so insufficient initiator would be added if the offline recipe is used. As a by product, the desired conversion could not be reached for a dirty reactor and the final number average molecular weight would not be within the allowed range. Therefore, this batch would have to be thrown out and money would be lost.

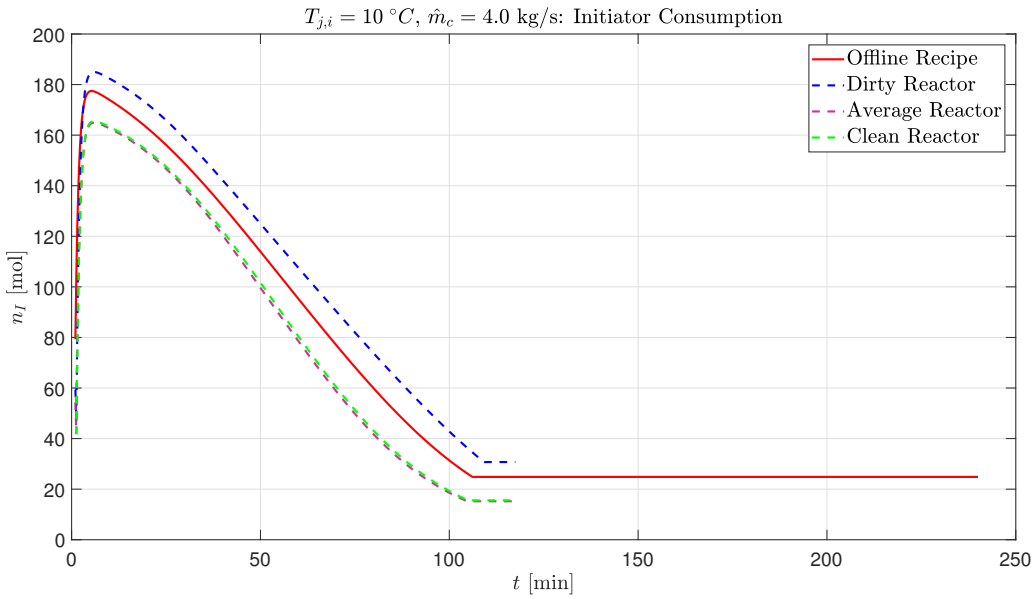


Figure 7.16: Initiator for a low cooling capacity and cold cooling water temperature.

Figure 7.17 shows the reactor temperature profile for each reactor condition.

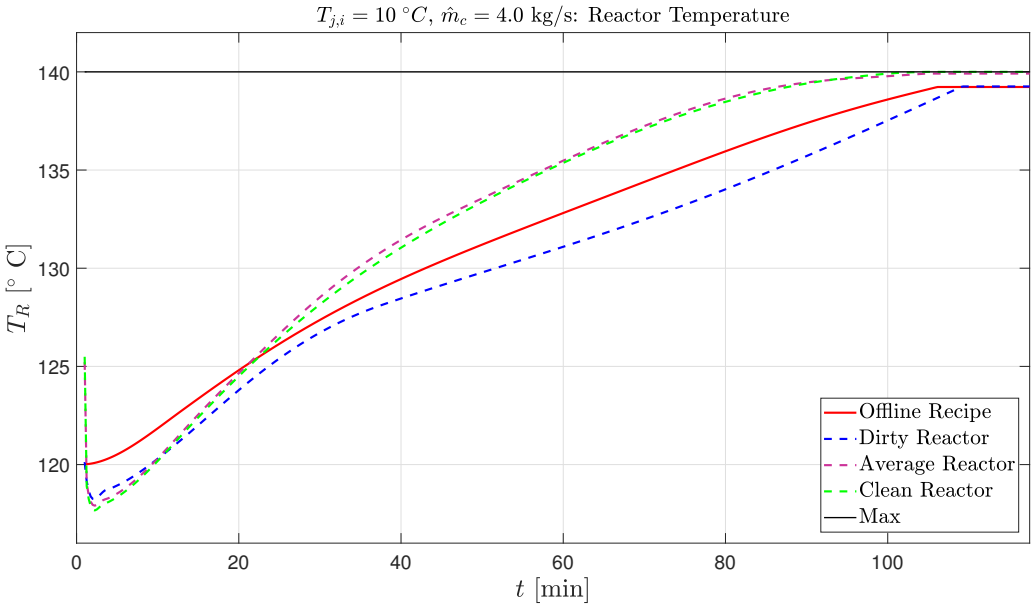


Figure 7.17: Reactor temperature for a low cooling capacity and cold cooling water temperature.

The clean and average reactors have indistinguishable temperature profiles, while

the dirty reactor temperature profile is similar in shape to the offline recipe. The dirty reactor should have a profile similar to the offline recipe with the differences resulting from the upper bound of the cooling constraint and the cooling water temperatures. The fact that the clean and average reactor give analogous reactor temperature profiles demonstrates that for these particular operating conditions, the reactor temperature profile is unaffected by a 10% decrease in the heat transfer coefficient.

Figure 7.18 shows that the cooling water flow rates behave as expected with the dirty reactor having the highest flow rate and the clean reactor has the lowest flow rate. It is expected that the clean reactor would have the lowest cooling flow rate and the dirty reactor would have the largest due to the value of the coefficient of heat transfer; consequently, the clean reactor needs less cooling water to remove the same amount of heat than is required for a dirty reactor. The average reactor has a cooling water flow rate approximately between the dirty and clean reactors. It is interesting to observe that the average reactor has a higher flow rate than the clean reactor even though the reactor temperatures are nearly identical; this demonstrates how the coefficient of heat transfer can impacts the required cooling water flow rate but not the corresponding reactor temperature.

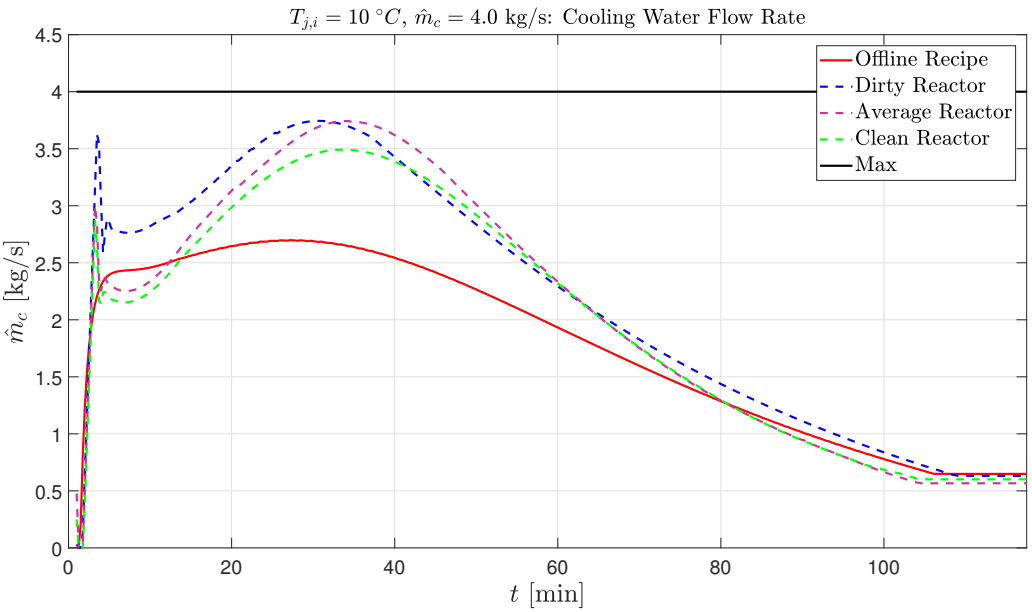


Figure 7.18: Cooling water flow rates for a low cooling capacity and cold cooling water temperature.

A clean reactor produces a polymer with a final number average molecular weight of 99.95 $\mathrm{kg\,mol^{-1}}$, which is the closest to the desired value as seen in Figure 7.19.

The average and dirty reactors produces a polymer with a final number average molecular weight of 99.93 and 99.67 $\mathrm{kg\,mol^{-1}}$, respectively. In all three cases an acceptable polymer is produced that is within ±1 of the required value. The offline recipe predicts a final number average molecular weight of 99.53 $\mathrm{kg\,mol^{-1}}$, which is worse than the online values but still within the allowed range.

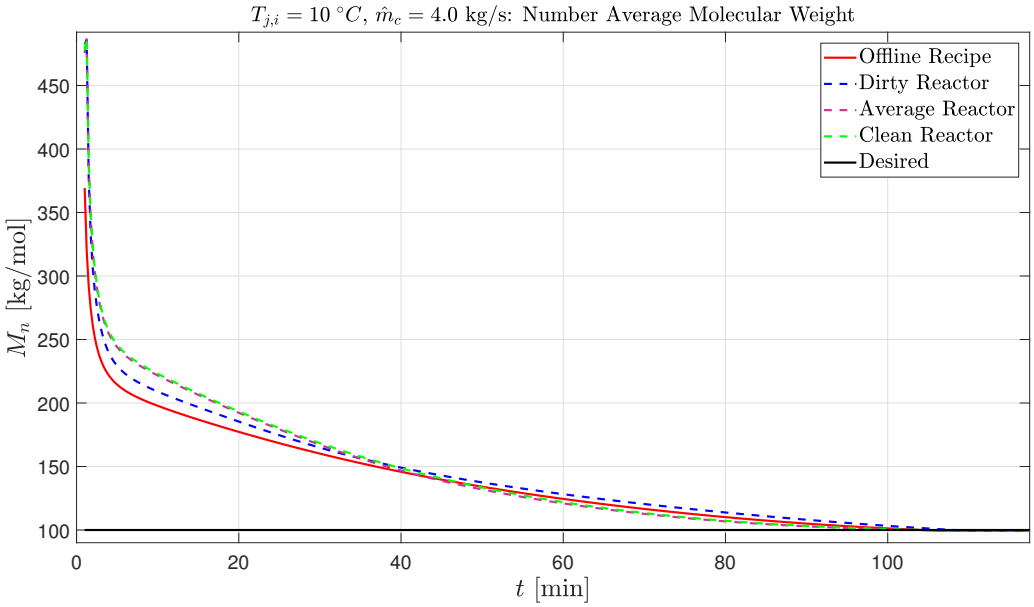


Figure 7.19: Number average molecular weights for a low cooling capacity and cold cooling water temperature.

Figure 7.20 shows that the shortest batch time is for the average reactor at 104.4 minutes; this is only 0.2 minutes shorter than the batch time for the clean reactor, which may be within the accuracy of the optimization algorithm. The dirty reactor gives a longer batch time at 108.9 minutes, which is expected since the reactor has a worse heat transfer coefficient which means that it is more difficult to regulate the reactor temperature and therefore, it takes longer to reached the desired conversion. The offline recipe predicts a batch time of 106.0 minutes, which is longer than for the clean or average reactors. These results show that under these conditions, using an online optimization approach will result in a shorter batch time in most instances. For a clean or average reactor, the batch time is shortened by a minute, which could result in up to 76 more batches per year per reactor; if there are several batch reactors being run in parallel in the plant, this could result in a consequential increase in production if online optimization is used.

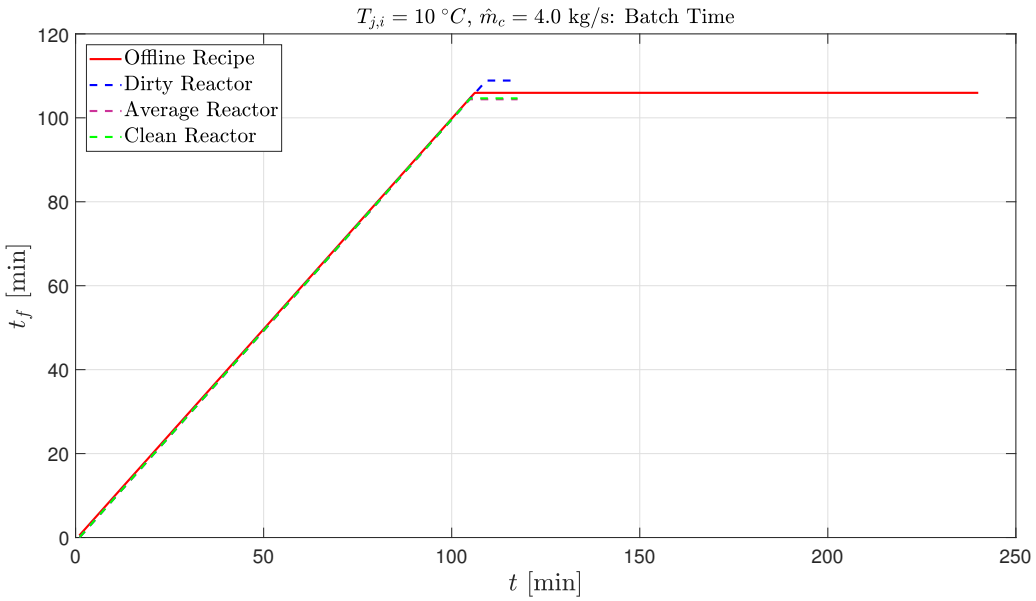


Figure 7.20: Batch times for a low cooling capacity and cold cooling water temperature.

These simulations indicate that even for a lower cooling capacity, the full control structure is able to produce a better quality product and in most cases, a shorter batch time in comparison to the offline recipe. When the reactor is dirty, the full control structure has a longer batch time but the product purity is better than for the offline recipe; the longer batch time suggests that if the offline recipe is used in this case, the batch would be terminated prematurely before a conversion of 80% were reached.

7.2.2 Summary of Results

The simulation results are summarized in Table 7.2 for ease of comparison, were M/I represents the monomer to initiator ratio. These results indicate that there is a benefit of using online optimization versus following a fixed recipe in the majority of the cases. For scenarios where the operating conditions are similar to those used in the offline recipe, the advantage is limited. In the event on disturbances that occur during a batch, the offline recipe would not adjust where as the online approach could alter its operating conditions to account for this disturbance. Therefore, it is recommended that a two-level online optimization control structure be implemented for use in production of EPS despite the time and cost it may take to develop such a control structure. There is a high probability that this could lead to an increase in profit due to a reduction in batch time and a savings in initiator since the amount of initiator would be customized to each batch.

Table 7.2: Cost Benefit Analysis

Reactor Condition	$\hat{m}_{c,max}$ [kg s^{-1}]	$T_{J,i}$ [°C]	M/I [-]	$T_{R,0}$ [°C]	t_f [min]	$\overline{M}_{n,f}$ [kg mol^{-1}]
Offline	4.5	20	600.8	125.6	106.0	99.53
Clean	5.0	10	700.7	132.4	104.4	99.97
Clean	4.5	10	651.3	140.3	104.5	99.94
Clean	4.0	10	646.7	125.6	104.6	99.95
Average	5.0	10	678.5	128.7	103.6	99.96
Average	4.5	10	527.8	124.8	113.9	99.67
Average	4.0	10	648.7	125.4	104.4	99.92
Dirty	5.0	10	679.7	132.5	103.5	99.97
Dirty	4.5	10	605.8	125.1	106.6	99.75
Dirty	4.0	10	458.5	120.3	108.9	99.67
Clean	5.0	20	644.5	119.2	104.7	99.92
Clean	4.5	20	643.6	125.3	104.8	99.91
Clean	4.0	20	714.9	130.0	394.2	82.83
Average	5.0	20	555.4	126.1	111.3	99.66
Average	4.5	20	639.8	125.4	104.8	99.88
Average	4.0	20	651.6	120.9	395.5	77.29
Dirty	5.0	20	600.8	125.5	106.9	99.72
Dirty	4.5	20	615.9	129.4	105.9	99.77
Dirty	4.0	20	681.8	127.1	349.6	64.16

Using an estimated cost for the initiator of 2.11 US$/kg, economic calculations are performed to illustrate the potential savings on initiator for the offline recipe versus the online optimization [30]. The amount of initiator required per batch is calculated along with the number of batches that can be run per reactor per year assuming 24/7 and 365 days a year operation. Then the total annual cost of initiator at each operating condition is determined. These calculations are summarized in Appendix C. These values are for production in one reactor while in reality, plants are likely to have multiple reactors running in parallel; therefore, the total plantwide initiator expenses would be greater so any savings would also be larger. Figures 7.21-7.23 show the annual cost of initiator versus the batch time for the offline recipe and the three reactor conditions at the different cooling capacities and cooling water inlet temperatures. Bear in mind that the initiator masses are based on a lab-scale reactor so the annual cost of initiator for a full size production will be remarkably higher.

Figure 7.21 shows the annual amount cost of initiator for the offline recipe and for a dirty reactor. While it appears that a dirty reactor has the smallest annual initiator

cost for a low cooling capacity and a hot cooling water temperature, these conditions result in the fewest number of batches per year and the polymer quality is not near the required final number average molecular weight; therefore, it is recommended to not operate under these conditions. The annual initiator cost for a dirty reactor with a low cooling capacity and cold cooling water is the largest of all the scenarios for a dirty reactor. This is because the optimized amount of initiator in this instance is considerably more in comparison to the others; this may be that the simulation conditions result in a flat optimization problem making it challenging to locate the minimum, which is why the value differs from that of an average or clean reactor. The other four online simulations have a lower annual cost of initiator than the offline recipe, with the greatest savings being for the combination of a high cooling capacity and cold cooling water. Therefore, for all the operating conditions that produce in the required final number average molecular, the use of an online control structure results in a reduction in the annual initiator expense for a dirty reactor.

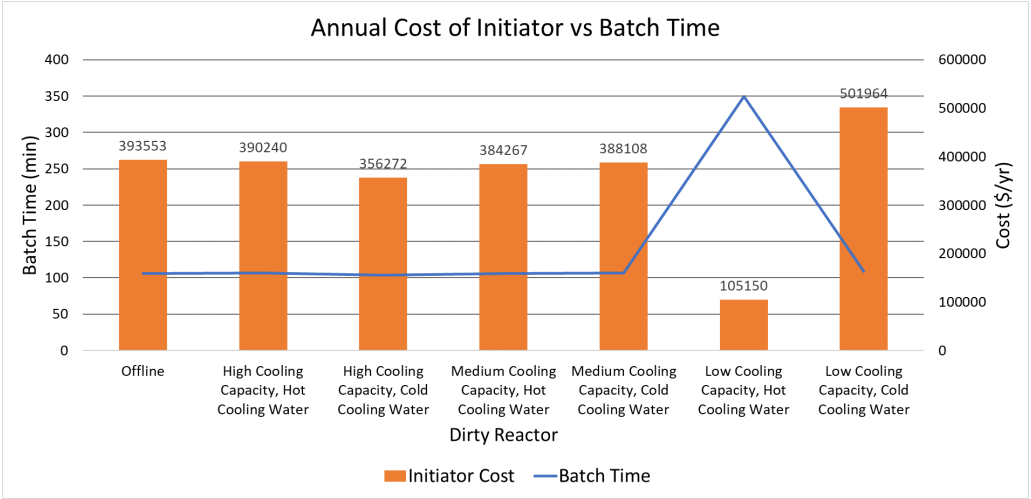


Figure 7.21: Comparison of annual cost of initiator versus batch time for different reactor conditions in a dirty reactor.

An average reactor shows somewhat contrasting results since two of the online simulations have larger annual expenses, as seen in Figure 7.22. However, as was explained previously, this are a caused by these operating conditions causing the optimization problem to have a flat spot, so the identified optimum is not the true local optimum. Therefore, this particular results should be treated as outliers. For two of the other scenarios, the savings on initiator is hefty.

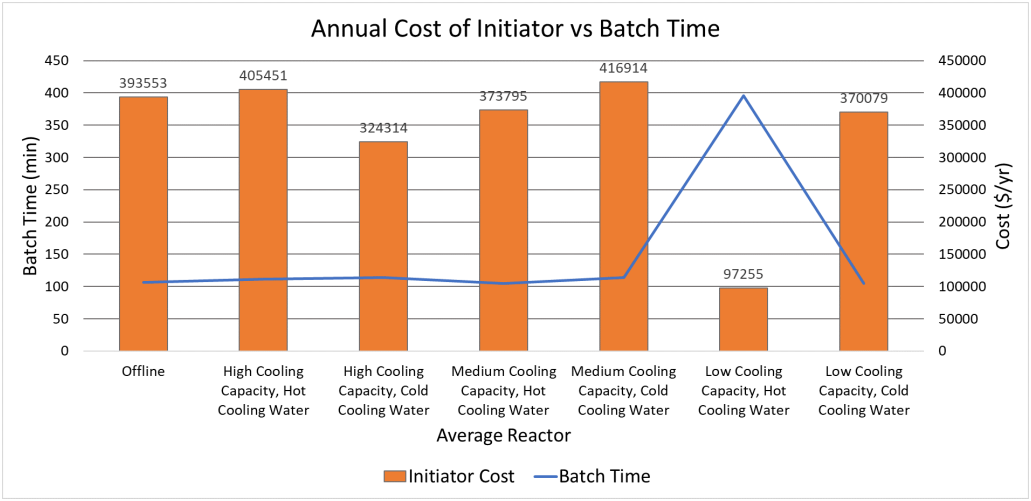


Figure 7.22: Comparison of annual cost of initiator versus batch time for different reactor conditions in an average reactor.

A clean reactor gives the clearest picture of the potential annual savings, as illustrated in Figure 7.23.

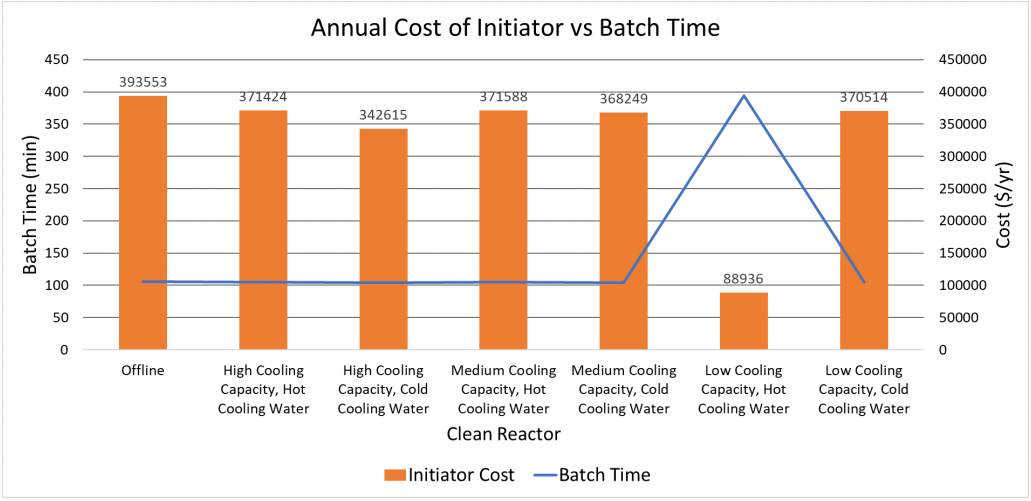


Figure 7.23: Comparison of annual cost of initiator versus batch time for different reactor conditions in a clean reactor.

On average a clean reactor requires less initiator so the annual expense will be lower than for an average reactor or dirty reactor. This could be used as motivation for increasing the amount of reactor cleanings done. Since all of the simulations using an online control structure require less initiator, the annual savings on average would be US$74,665 per reactor in comparison to using a fixed recipe.

For all the different scenarios, the online optimization will prove advantageous over the fixed recipe in both the controller's ability to handle disturbances and the economic savings. If the offline recipe is followed the annual initiator cost is US$393,553. In comparison, the online optimization control structure has a range of annual costs between US$88,935-501,963, depending on the simulated operating conditions. The true operating cost when using the online approach would be somewhere in between these two values since the reactor conditions will change between cleanings and seasonally. Taking the average annual operating cost for the online simulations suggests that the annual savings for one reactor could be upwards of US$98,000. If the plant has several reactors, this savings would be substantial.

CHAPTER 8

CONCLUSION

8.1 Conclusion

This report summarized theoretical concepts of modeling polymerization processes and semi-batch reactors, which served as the foundation for the first principles model developed for the production of EPS. Several key assumptions were made so that the resulting model is simple and numerically robust, which means that the computational effort to solve the model is low. This property is important since the model was to be used in the construction of an advanced control structure.

Background information on advanced control structures was provided to establish a foundation from which a two-layer control structure could be built for the case study used in this work. First offline optimization of the EPS model was performed before beginning constructing the control structure. This was done to determine the optimal monomer to initiator ratio, initial reactor temperature, and the optimal reactor temperature profile throughout the batch. These values were required before the development of the full control structure could start.

A NMPC lower level was chosen because the polymerization process is highly nonlinear and the nonlinearities could not be neglected. This layer is constructed to follow the reactor temperature trajectory found in the offline optimization stage. The cooling water flow rate is optimized to track this profile. The prediction horizon for this layer was selected to be short since the process is open loop unstable so a long prediction horizon would result in poor control. In addition, it is important that this level have a shorter prediction horizon than the top level so that there is sufficient time scale separation between the two layers.

Once the bottom layer could track the optimal reactor temperature profile, the upper layer was developed. The top level uses a DRTO approach to recalculate the optimal reactor temperature trajectory based on process measurements. The new optimal trajectories are sent to the NMPC below using an OPC server. The DRTO layer was constructed to calculate the optimal reactor temperature derivative rather than the reactor temperature itself; this allows for smoother control of the reactor temperature. This level has a prediction horizon that exceeds the anticipated batch time; it is important that this level has a prediction horizon that is at least the

length of the batch time. Together the two-layers work to calculate and realize the optimal operating conditions for the production of EPS to minimize the batch time while still producing the required final number average molecular weight.

The two layers were tested using various cooling water temperatures and cooling capacity constraints. The upper bound on the cooling water flow rate plays a large role in the resulting batch time; it was also found that below 4.0 kg/s and with a cooling water temperature of 10 °C, the system could not produce the desired product. This suggested that for the simulated system, a cooling capacity below this provided insufficient cooling to realize the optimal reactor temperature profile. The cooling water temperatures that the system could handle were dependent upon the selected upper bound for the cooling water flow rate. If the limit was sufficient, then a higher cooling water temperature had a small impact on the batch time and the resulting polymer product; however, if the cooling flow rate was too small, then the higher cooling water temperatures would lead to a batch time that was up to three times longer than before and a polymer with a final number average molecular weight notable far from the desired value. These results show the importance of selecting either a moving cooling upper limit or choosing a fixed upper bound that includes a sufficient safety margin.

While the two layers were often able to work together to achieve the desired results, even with a perfect model there were instances when the NMPC level had problems following the reference trajectory. These differences are due to the variations in the models that are solved in the two different layers. The different prediction horizons, objective functions and manipulated variables, along with the different temperature equations contribute to the mismatch between the two layers. While extensive tuning was done for both layers, there were still some operating conditions that the system could not handle. Rather than revealing problems with the controller or the model, it is thought that this shows system limitations, which should be considered when selecting the bounds on the cooling water temperature and the cooling water flow rates.

Each stage of development was then compared to show how the full control structure trajectories differ from the offline optimization. These results were positive since the trajectories were fairly similar despite the addition of a cooling constraint to the full control structure. This suggests that in an ideal case, the offline optimization and the advanced control structure perform comparably; however, if any of the conditions change, the advanced control structure is able to handle these disturbances whereas the offline optimization is fixed so it cannot. A cost-benefit analysis was performed to illustrate the advantages of using a two-level advanced control structure versus using a fixed offline recipe for the production of EPS. In industry, it is common to use a fixed recipe approach for batch production because it is easier and cheaper to determine this recipe than it is to develop an online optimization control structure.

However, this work showed that the use of a two-level advanced control structure can result in shortened batch times and improved product quality. Therefore, it is recommended that advanced control structures be considered for use in the production of EPS.

8.2 Further Work

The next step would be to alter the model to fit a real pilot reactor or a production plant. The first task in this would be change the sample times of the two layers. It was roughly timed to take the DRTO layer about twenty seconds and the NMPC two seconds to solve their respective optimization problem for one sample step. This suggests that the selected sample time of fifteen seconds for the DRTO level is too short by approximately five seconds; the NMPC level had the same sample time so it could be shortened if desired. The total solve time for one sample step is approximately twenty-two seconds. This is sufficiently quick for an online application but should be confirmed.

Some constants in the model representing the reactor size and properties should be altered to match the process. The initiator type should also be adjusted as required to match the one used in the actual process. The cooling constraint limits and jacket feed temperature limits need to match the physical limitations of the process. Further tuning of the controller weights may also prove necessary when working with a different reactor.

When a real process is identified, further economic evaluation can be added to the cost benefit analysis. The current annual cost of the initiator and the current annual profit for a fixed recipe approach can be compared to what the cost would be using the two-level control structure. It was attempted to do so in this work but the costs of chemicals and other operating costs are difficult to find in public information. Therefore, the only option was to use an estimate based on a value found online for the import value of the initiator to India so it is a rough estimate.

This work did not consider the addition of the blowing agent pentane so future work could be done to investigate the effect that this component would have on the batch time and the product properties. Lund implied in her work that the amount of pentane added affects the viscosity of the reaction mixture and the final number average molecular weight [21]. She also showed that the conversion is not very sensitive to the addition of pentane, which suggests that the batch time might not be affected by the amount of pentane added. Including the pentane in the model would improve the model accuracy so it is worth investigating.

Considering that the polymer industry is predicted to continue its growth, continuation of this work could have a large pay off. The model is constructed in such a way as to make it easy to quickly change the reactor parameters to fit any new

process; therefore, it would be possible to utilize the same foundation to construct an advanced control structure for many different EPS production facilities.

APPENDIX A
THEORETICAL SUPPLEMENTS

A.1 Optimization Definitions

Definition A.1 (Feasible directions) *Given a feasible point $\boldsymbol{r}$ and the active constraint set $\mathcal{A}(\boldsymbol{r})$ from Definition A.8, the set of linearized feasible directions $\mathcal{F}(\boldsymbol{r})$ is*

$$\mathcal{F}(\boldsymbol{r}) = \left\{ d \,\middle|\, \begin{array}{ll} d^T \nabla \boldsymbol{c}_j(\boldsymbol{r}) = 0, & \text{for all } j \in \mathcal{E}, \\ d^T \nabla \boldsymbol{c}_j(\boldsymbol{r}) \geq 0, & \text{for all } j \in \mathcal{A}(\boldsymbol{r}) \cap \mathcal{I} \end{array} \right\}$$

where $\mathcal{F}(\boldsymbol{r})$ is a cone and is dependent on the definition of the constraint functions $\boldsymbol{c}_j, j \in \mathcal{E} \cup \mathcal{I}$ [23].

Definition A.2 (Critical Cone) *The critical cone $\mathcal{C}(\boldsymbol{r}^*, \boldsymbol{\lambda}^*)$ is defined as follows [23]:*

$$\mathcal{C}(\boldsymbol{r}^*, \boldsymbol{\lambda}^*) = \{\boldsymbol{w} \in \mathcal{F}(\boldsymbol{r}^*) \,|\, \nabla \boldsymbol{c}_j(\boldsymbol{r}^*)^T \boldsymbol{w} = 0, \text{ all } j \in \mathcal{A}(\boldsymbol{r}^*) \cap \mathcal{I} \text{ with } \boldsymbol{\lambda}_j^* > 0\}$$

which can equivalently be written:

$$\boldsymbol{w} \in \mathcal{C}(\boldsymbol{r}^*, \boldsymbol{\lambda}^*) \Leftrightarrow \begin{cases} \nabla \boldsymbol{c}_j(\boldsymbol{r}^*)^T \boldsymbol{w} = 0, & \text{for all } j \in \mathcal{E} \\ \nabla \boldsymbol{c}_j(\boldsymbol{r}^*)^T \boldsymbol{w} = 0, & \text{for all } j \in \mathcal{A}(\boldsymbol{r}^*) \cap \mathcal{I} \text{ with } \boldsymbol{\lambda}_j^* > 0 \\ \nabla \boldsymbol{c}_j(\boldsymbol{r}^*) \geq 0, & \text{for all } j \in \mathcal{A}(\boldsymbol{r}^*) \text{ with } \boldsymbol{\lambda}_j^* = 0 \end{cases} \tag{A.1}$$

where $\boldsymbol{w} \in \mathcal{F}(\boldsymbol{r}^*)$ such that $\boldsymbol{w}^T \nabla f(\boldsymbol{r}^*) = 0$.

Definition A.3 (Convex Set) *A set $\mathcal{S} \in \mathbb{R}^n$ is a convex set if the straight line segment connecting any two points in $\mathcal{S}$ lies entirely inside $\mathcal{S}$ [23].*

Definition A.4 (Convex Function) *The function f is a convex function if its domain $\mathcal{S}$ is a convex set and if for any two points $\boldsymbol{x}$ and $\boldsymbol{y}$ in $\mathcal{S}$, the following property is satisfied [23]:*

$$f(\alpha \boldsymbol{x} + (1-\alpha)\boldsymbol{y}) \leq \alpha f(\boldsymbol{x}) + (1-\alpha) f(\boldsymbol{y}), \qquad \text{for all } \alpha \in [0,1] \tag{A.2}$$

Definition A.5 (Convex Programming) *The term convex programming is used to describe a special case of the general constrained optimization shown in Problem 3.9 in which [23]:*

- *the objective function is convex,*
- *the equality constraint functions $\boldsymbol{c}_j, j \in \mathcal{E}$ are linear, and*
- *the inequality constraint functions $\boldsymbol{c}_j, j \in \mathcal{I}$, are concave.*

Definition A.6 (Taylor's Theorem) *Suppose that $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable and that $p \in \mathbb{R}^n$. Then we have that*

$$f(r+p) = f(r) + \nabla f(r+tp)^T p \tag{A.3}$$

for some $t \in (0,1)$. Moreover, if f is twice continuously differentiable, we have that

$$\nabla f(r+p) = \nabla f(r) + \int_0^1 \nabla^2 f(r+tp) p \, dt \tag{A.4}$$

and that

$$f(r+p) = f(r) + \nabla f(r)^T p + \frac{1}{2} p^T \nabla^2 f(r+tp) p \tag{A.5}$$

for some $t \in (0,1)$.

Definition A.7 (Lagrange Function) *For a generic constrained optimization problem, the Lagrange function can be written as:*

$$\mathcal{L}(\boldsymbol{r}, \boldsymbol{\lambda}) = f(\boldsymbol{r}) - \sum_{j \in \mathcal{E} \cup \mathcal{I}} \lambda_j \boldsymbol{c}_j(\boldsymbol{r}) \tag{A.6}$$

where λ_j are the Lagrange multipliers for the corresponding constraint $\boldsymbol{c}_j(\boldsymbol{r})$.

The Lagrange multiplier reveals how sensitive the optimal objective value function is to the presence of the constraint. In other words, it indicates how hard the objective function pushes or pulls the solution against the particular constraint [23].

Definition A.8 (Active Set) *The active set $\mathcal{A}(\boldsymbol{r})$ at any feasible $\boldsymbol{r}$ consists of the equality constraint indices from $\mathcal{E}$ together with the indices of the inequality constraints j for which $\boldsymbol{c}_j(\boldsymbol{r}) = 0$; that is,*

$$\mathcal{A}(\boldsymbol{r}) = \mathcal{E} \cup \{ j \in \mathcal{I} \mid \boldsymbol{c}_j(\boldsymbol{r}) = 0 \}$$

Definition A.9 (LICQ) *Given the point $\boldsymbol{r}$ and the active set $\mathcal{A}(\boldsymbol{r})$ defined in Definition A.8, we say that the LICQ holds if the set of active constraint gradients $\{\nabla \boldsymbol{c}_j(\boldsymbol{r}),\ j \in \mathcal{A}(\boldsymbol{r})\}$ is linearly independent [23].*

Definition A.10 (Linear Programming) *If the objective function f is linear at all constraints $\boldsymbol{c}_j$ are linear, then the problem will be a linear program (LP) which is of the form [7]:*

$$\min_{\boldsymbol{r} \in \mathbb{R}^{n_r}} \quad \boldsymbol{d}^T \boldsymbol{r} \tag{A.7a}$$

$$\text{s.t.} \quad \boldsymbol{c}_j(\boldsymbol{r}) = \boldsymbol{a}_j{}^T \boldsymbol{r} - \boldsymbol{b}_j = 0, \qquad j \in \mathcal{E}, \tag{A.7b}$$

$$\boldsymbol{c}_j(\boldsymbol{r}) = \boldsymbol{a}_j{}^T \boldsymbol{r} - \boldsymbol{b}_j \geq 0, \qquad j \in \mathcal{I} \tag{A.7c}$$

A.2 Collocation

Collocation is one method utilized in dynamic optimization to discretize a continuous system. This method belongs to the more general method of weighted residuals, which were used to find approximate solutions to differential equations before finite element method was developed [14]. The collocation method takes the weight functions from the family of the Dirac functions in a set domain. The integration of this weighted residual results in the residuals being forced to zero at set points in the domain [14]. As the number of collocation points increases, the better the approximation of the differential equation is and the approximate solution begins to approach the true solution. Since this method was originally created, a variety of sub-methods of collocation have been developed as well; the difference between these methods is how the collocation points are selected. One such method is known as orthogonal collocation, which works by selecting the collocation points as the roots of orthogonal polynomials [14].

A.2.1 Orthogonal collocation

Orthogonal collocation is applied to the system of differential equations and converts them into algebraic ones. Orthogonal polynomials are selected because they have a variety of useful properties such as: recurrence relation, existence of real roots, and interlacing roots [14]. These properties provide a way to solve, expand and understand many different common differential equations. This discretization method can then be paired with an optimization method such that the problem will simultaneously converge to an optimum and solve the differential equations [3]. A polynomial approximation is applied to the differential equation and requires satisfaction of the equation at the zeros of the orthogonal polynomials, which are the discrete collocation points.

While any orthogonal polynomial can be used, it has been shown that the use of Lagrange polynomials pairs well with the optimization method of sequential

quadratic programming [3]. The Lagrange interpolation polynomials are given by:

$$y_i(x) = \sum_{k=1}^{N+1} c_k L_k(x) \tag{A.8}$$

where N is the order of the polynomial and $L_k(x)$ is the Lagrange polynomial function that is defined by:

$$L_k(x_j) = \prod_{j=1, j\neq k}^{N+1} \frac{x - x_j}{x_k - x_j} \tag{A.9}$$

where $\{x_j \mid j = 1, 2, \ldots, N+1\}$ are the interpolation nodes. The Lagrange polynomial has the property that:

$$L_k(x_j) = \begin{cases} 1, & j = k \\ 0, & j \neq k \end{cases}$$

Legendre polynomials are another common choice. However, when orthogonal collocation is used in optimization, it is best to use Lagrange polynomials. The reason for this is expanded upon in Section A.2.1.1.

A.2.1.1 Orthogonal Collocation in Optimization

Legendre polynomials require bounds and starting points to find the coefficients; however, given that they have no physical significance and thus no *a priori* estimated ranges, this is difficult to do [3]. Using Lagrange polynomials remedies this since $t_0 = 0$ and $t_i, i = 1, \ldots, n$ are zeros of an n^{th} order Legendre polynomial defined from 0 to t_f. It is then easy to apply meaningful bounds and starting points from physical insight if $y_i = y_n(t_i)$ is selected as the decision variables. Other decision variables include the constant parameters and the coefficients of the polynomial approximation to the control profiles [3].

The ODE can then be written as algebraic equalities at the collocation points. This can be generically written as:

$$\begin{aligned} \min_x \quad & F(x) \\ \text{s.t.} \quad & r(x) = 0, \\ & h(x) = 0, \\ & g(x) \leq 0, \\ & x_l \leq x \leq x_u \end{aligned} \tag{A.10}$$

The SQP method can then be trivially applied to Problem A.10. Together, this gives an implicit orthogonal collocation solution to the ODE and is easy to apply and converges to the optimum superlinearly [3].

APPENDIX B
FURTHER MODEL INFORMATION

B.1 Reactor Calculations

The following calculations show how certain values for the model were determined. All the calculations performed here resulted in changes to the original model taken from Lund [21]. These changes were deemed necessary to refine the model to better represent a real reactor. It should be noted, however, that the way the model was previously formulated and utilized, changing these values would have little impact on the results; therefore, the results presented in [21] are still relevant.

The model was designed to handle both a lab scale reactor and a real process scale reactor. All of the results presented in this work used the lab scale reactor, which has a volume of 24 m^3. Other parameters were determined based on this assumption. In addition, a more realistic mass of monomer was selected using the reactor size; previously, only 1 mol of monomer was used for convenience and does not represent a realistic amount.

Reactor diameter:

It was assumed that the reactor is a cylindrical shape with half spheres at either end and that the height of the cylindrical portion is equal to the diameter of the spherical ends. A simple sketch illustrates this assumption in Figure B.1. Using the reactor volume, these values were then determined to be:

$$V_R = V_{sphere} + V_{cylinder} \tag{B.1}$$

$$= \pi r^2 + \frac{4}{3}\pi r^3 \tag{B.2}$$

$$= \pi(\frac{d}{2})^2 + \frac{4}{3}\pi(\frac{d}{2})^3 \tag{B.3}$$

$$\Rightarrow \boxed{d = 2.64\,\text{m}} \tag{B.4}$$

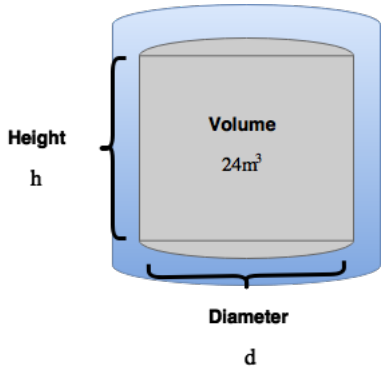


Figure B.1: Simple illustration of reactor geometry

Heat Exchange Surface Area:

The reactor was assumed to be 90% covered by the jacket; therefore, the heat exchange surface area is given by:

$$A_{total} = 0.9 * (A_{cylinder} + A_{sphere}) \tag{B.5}$$

$$= 0.9 * (2\pi rh + \cancel{2\pi r^2} + 4\pi r^2) \tag{B.6}$$

$$= 0.9 * (2\pi \frac{d}{2} d + 4\pi \frac{d}{2}^2) \tag{B.7}$$

$$= 0.9 * (2\pi d^2) \tag{B.8}$$

$$= \boxed{39.32\,\mathrm{m}^2} \tag{B.9}$$

Mass of Liquid in Cooling Circuit:

It was assumed that the jacket holds a width of 2 cm of cooling fluid; the thickness of the jacket walls are neglected.

$$V_{jacket} = V_{liquid} - V_{reactor} \tag{B.10}$$

$$= \pi r_0^2 d_0 + \frac{4}{3}\pi r_0^3 - 24 \tag{B.11}$$

$$= \pi \frac{d_0}{2}^2 d_0 + \frac{4}{3}\pi \frac{d_0}{2}^3 - 24 \tag{B.12}$$

$$= \pi \frac{d + 0.02}{2}^2 (d + 0.02) + \frac{4}{3}\pi \frac{d + 0.02}{2}^3 - 24 \tag{B.13}$$

$$= 0.64\mathrm{m}^3 \times \frac{1000\,\mathrm{L}}{1\,\mathrm{m}^3} \times \frac{1\,\mathrm{kg}}{1\,\mathrm{L}} \tag{B.14}$$

$$= \boxed{637\,\mathrm{kg}} \tag{B.15}$$

Initial Monomer and Water Mass:

The monomer and water mass were calculated assuming that the reactor is 80% full at the beginning of the batch and that 35% of this volume is water.

$$V_{water} = 0.8 * 0.35 * V_{total} \tag{B.16}$$

$$= 6.72\,\mathrm{m}^3 \times \frac{1000\,\mathrm{kg}}{1\,\mathrm{m}^3} \times \frac{1000\,\mathrm{g}}{1\,\mathrm{kg}} \times \frac{1\,\mathrm{mol}}{18.015\,\mathrm{g}} \tag{B.17}$$

$$= \boxed{3.7302 \times 10^5\,\mathrm{mol}} \tag{B.18}$$

$$V_{monomer} = 24 - 6.72\,\mathrm{m}^3 \tag{B.19}$$

$$= 12.48\,\mathrm{m}^3 \times \frac{909\,\mathrm{kg}}{1\,\mathrm{m}^3} \times \frac{1000\,\mathrm{g}}{1\,\mathrm{kg}} \times \frac{1\,\mathrm{mol}}{104.15\,\mathrm{g}} \tag{B.20}$$

$$= \boxed{1.0892 \times 10^5\,\mathrm{mol}} \tag{B.21}$$

B.2 Model Parameters

This table provides a summary of the parameters used in the model. Several sources were used to find these values but this not discussed here; the details and sources can be found in [21].

Parameter	Description	Value	Unit
a	Gel effect tuning parameter	1.75	-
A	Gel effect tuning parameter	0.465	-
A_{cr}	Gel effect critical point parameter	9.44	$(\mathrm{kg\,mol^{-1}})^{0.5}$
A_l	Gel effect testing parameter	0.348	-
B	Glass effect tuning parameter	1.00	-
C	Cage effect tuning parameter	1.00	-
E_{cr}	Gel effect critical point exponent parameter	1.60×10^4	$\mathrm{J\,mol^{-1}}$
E_{dm}	Thermal initiation activation energy	1.16×10^5	$\mathrm{J\,mol^{-1}}$
E_p	Propagation activation energy	3.25×10^4	$\mathrm{J\,mol^{-1}}$
E_{tc}	Termination activation energy	1.40×10^4	$\mathrm{J\,mol^{-1}}$
E_{trM}	Transfer to monomer activation energy	1.27×10^5	$\mathrm{J\,mol^{-1}}$
f	Initiator efficiency	0.85	-
k_{d0}	Decomposition of initiator rate constant	9.24×10^{15}	$\mathrm{s^{-1}}$
k_{dm0}	Thermal initiation rate constant	0.219	$\mathrm{m^6\,mol^{-2}\,s^{-1}}$
k_{p0}	Propagation rate constant	4.27×10^4	$\mathrm{m^3\,mol^{-1}\,s^{-1}}$

k_{tc0}	Termination by combination rate constant	1.47×10^7	$m^3\,mol^{-1}\,s^{-1}$
$k_{tc,rd,max}$	Maximum value of residual diffusion termination rate constant	1.74×10^{-27}	-
$k_{tc,rd,min}$	Minimum value of residual diffusion termination rate constant	2.34×10^{-26}	-
k_{trM}	Transfer to monomer rate constant	6.05×10^{12}	$m^3\,mol^{-1}\,s^{-1}$
R	Gas constant	8.314	$J\,mol^{-1}\,K^{-1}$
$T_{g,C5}$	Glass transition temperature of pentane	123	K
$T_{g,M}$	Glass transition temperature of styrene	185	K
$T_{g,P}$	Glass transition temperature of polystyrene	370	K
$V_{f,cr,d}$	Critical free volume for cage effect onset	0.069	-
$V_{f,cr,p}$	Critical free volume for glass effect onset	0.0465	-
α_{C5}	Fractional free volume of pentane	0.00079	-
α_M	Fractional free volume of monomer	0.001	-
α_P	Fractional free volume of polymer	0.00028	-
δ_c	Segmental diffusion parameter for styrene	0.001	-
ΔH_R	Reaction enthalpy	-7.11×10^4	$J\,mol^{-1}$
ρ_{C5}	Density of pentane	$649 - 1.15(T_R - T_0)$	$kg\,m^{-3}$
ρ_I	Density of initiator	1560.00	$kg\,m^{-3}$
ρ_M	Density of monomer, styrene	$924.0 - 0.918(T_R - T_0)$	$kg\,m^{-3}$
ρ_P	Density of polymer, polystyrene	$1084.8 - 0.605(T_R - T_0)$	$kg\,m^{-3}$
ρ_W	Density of water	1000.0	$kg\,m^{-3}$

B.3 Derivation of Energy Balances

B.3.1 Reactor

The general law of energy conservation is expressed as [31]:

$$\left\{\begin{array}{c}\text{rate of energy}\\ \text{accumulation}\end{array}\right\} = \left\{\begin{array}{c}\text{rate of energy in}\\ \text{by convection}\end{array}\right\} - \left\{\begin{array}{c}\text{rate of energy out}\\ \text{by convection}\end{array}\right\} + \left\{\begin{array}{c}\text{net rate of heat addition}\\ \text{to the system from}\\ \text{the surroundings}\end{array}\right\} + \left\{\begin{array}{c}\text{net rate of work}\\ \text{performed on the system}\\ \text{by the surroundings}\end{array}\right\}$$

The equation can be written explicitly as:

$$\frac{dE}{dt} = \hat{m}_{in}\bar{E}_{in} - \hat{m}_{out}\bar{E}_{out} + Q + W \tag{B.22}$$

where the hats indicate mass per unit time and the bar indicates energy per unit mass [17]; the units for the energy balance are J/s. Since it is a semi-batch reactor there is no outflow so the second flow term is zero and the equation becomes:

$$\frac{dE}{dt} = \hat{m}_{in}\bar{E}_{in} + Q + W \tag{B.23}$$

The work term can be split into three different contributing factors: the flow work at the inlet W_f, the agitation work W_{ag}, and the work done by volume change W_v. There is no flow work at the outlet since this is a semi-batch reactor and therefore, there is no outflow during production.

$$W = W_f + W_{ag} + W_v \tag{B.24}$$

$$= \hat{m}_{in}\frac{P_{in}}{\rho_{in}} + W_{ag} - P\frac{dV}{dt} \tag{B.25}$$

The energy term is composed of the internal U_I, kinetic U_{KE}, and potential energy U_{PE} and can be written as:

$$E = U_I + U_{KE} + U_{PE} \tag{B.26}$$

The energy balance can then be written as:

$$\frac{d}{dt}(U_I + U_{KE} + U_{PE}) = \hat{m}_{in}(U_{int} + U_{KE} + U_{PE}) + Q + W_{ag} \tag{B.27}$$

Since the potential and kinetic energy are typically small compared to the internal energy, these terms can be neglected.

It is common to write the energy balance in terms of enthalpy using the definition of enthalpy: $H = U + PV$. Inserting this definition along with the individual work components gives:

$$\frac{dH}{dt} - P\frac{dV}{dt} - V\frac{dP}{dt} = \hat{m}_{in}H_{in} + Q + \hat{m}_{in}\frac{P_{in}}{\rho_{in}} + W_{ag} - P\frac{dV}{dt} \tag{B.28}$$

Assuming constant pressure and simplifing results in the following expression:

$$\frac{dH}{dt} = \hat{m}_{in}H_{in} + Q + W_{ag} \tag{B.29}$$

The agitation work is assumed small and can therefore be neglected, giving:

$$\frac{dH}{dt} = \hat{m}_{in}H_{in} + Q \tag{B.30}$$

In control it is common to convert the enthalpy balance to a function of temperature since this is easier to measure and therefore control. This conversion can be accomplished using the following thermodynamic relation that relates enthalpy to temperature, pressure, and number of moles:

$$dH = \left(\frac{\partial H}{\partial T}\right)_{P,n_i} dT + \left(\frac{\partial H}{\partial P}\right)_{T,n_i} dP + \left(\frac{\partial H}{\partial n_i}\right)_{P,T,n_k} dn_i \tag{B.31}$$

Note that the first partial derivative is the definition of the heat capacity $C_p = mc_p$ and since the pressure is assumed constant, the second partial derivative is zero. The third partial derivative represents the partial molar enthalpies $\overline{H}_i$. Inserting this into the energy balances gives:

$$\frac{dmc_pT_R}{dt} + \sum_i \overline{H}_i \frac{dn_i}{dt} = \hat{m}_{in}H_{in} + Q \tag{B.32}$$

If constant mass and heat capacity is assumed, which is a valid assumption for liquid systems, the equation becomes:

$$mc_p\frac{T_R}{dt} + \sum_i \overline{H}_i \frac{dn_i}{dt} = \hat{m}_{in}H_{in} + Q \tag{B.33}$$

The heat Q can be broken down into its individual components: the ambient contribution Q_{amb} and the heating/cooling system contribution Q_J. In this work the ambient contribution is neglected. The contribution from the jacket can be written as:

$$Q_J = -(UA)_J(T_R - T_J) \tag{B.34}$$

where $(UA)_j$ is the jacket's coefficient of heat transfer, T_R is the reactor temperature, and T_J is the jacket temperature. The temperature equation then becomes:

$$mc_p\frac{T_R}{dt} + \sum_j \overline{H}_i \frac{dn_i}{dt} = \hat{m}_{in}H_{in} + (UA)_J(T_R - T_J) \tag{B.35}$$

Rearranging then gives:

$$\frac{dT_R}{dt} = \frac{\hat{m}_{in}H_{in} - (UA)_J(T_R - T_J) - \sum_i \overline{H}_i \frac{dn_i}{dt}}{mc_p} \tag{B.36}$$

Since mc_p is the total mass and total heat capacity, it can be broken down into its components (reactor contents + vessel) as:

$$mc_p = \sum_i \underbrace{n_i c_{p,i}}_{\text{reactor contents}} + \underbrace{m_V c_{p,V}}_{\text{vessel}} \tag{B.37}$$

Since the propagation reaction generates the most heat of the three polymerization reaction types, the heat contribution from the components can be written as:

$$\sum_i \overline{H}_i \frac{dn_i}{dt} = \Delta H_R R_P V \tag{B.38}$$

The heat from the feed can be broken down into the individual component contributions and written as:

$$\hat{m}_{in}H_{in} = \sum_i c_{p,i}\hat{n}_i(T_{feed} - T_R) \tag{B.39}$$

Inserting this into the temperature equation gives the equation seen in Chapter 4.

$$\frac{dT_R}{dt} = \frac{\sum_i c_{p,i}\hat{n}_i(T_{feed} - T_R) - \Delta H_R R_P V - (UA)_J(T_R - T_J))}{\sum_i n_i c_{p,i} + m_V c_{p,V}} \tag{B.40}$$

B.3.2 Jacket

The jacket energy balance can be derived in a similar manner.

$$\left\{\begin{array}{c}\text{rate of energy}\\ \text{accumulation}\end{array}\right\} = \left\{\begin{array}{c}\text{rate of energy in}\\ \text{by flow}\end{array}\right\} + \left\{\begin{array}{c}\text{rate of energy out}\\ \text{by heat transfer}\end{array}\right\} - \left\{\begin{array}{c}\text{rate of energy out}\\ \text{by flow}\end{array}\right\}$$

Some of the derivation is now skipped since it was shown in the reactor temperature equation derivation. If constant density, volume and heat capacity are assumed for the jacket, the temperature equation then becomes:

$$\rho V c_{p,c}\frac{d(T_J - T_{ref})}{dt} = F\rho c_{p,c}[(T_{J,i} - T_{ref}) - (T_J - T_{ref})] - \dot{Q} \tag{B.41}$$

Since T_{ref} is a constant:

$$\frac{d(T_J - T_{ref})}{dt} = \frac{dT_J}{dt} \tag{B.42}$$

Combing this with the fact that $m_c = V\rho$ and $\hat{m}_c = F\rho$, the equation then becomes:

$$\frac{dT_J}{dt} = \frac{\hat{m}_c c_{p,c}(T_{J,i} - T_J) - \dot{Q}}{m_c c_{p,c}} \tag{B.43}$$

The heat transfer from the reactor can be written as

$$Q = -(UA)_J(T_R - T_J) \tag{B.44}$$

where $(UA)_J$ is the jacket's coefficient of heat transfer.

The equation is then easily transformed into the form seen in Chapter 4:

$$\frac{dT_J}{dt} = \frac{(UA)_J(T_R - T_J) + \hat{m}_c c_{p,c}(T_{J,i} - T_{J,o})}{m_c c_{p,c}} \tag{B.45}$$

Appendix C
Additional Simulation Results

C.1 Offline Optimization Results

The results of the offline optimization to calculate the optimal monomer to initiator ratio, starting reactor temperature, and the dynamic temperature trajectory to produce the shortest batch time are now examined for a cooling water inlet temperature of 20 °C.

The consumption rate of the initiator is shown in Figure C.1 where the optimal monomer to initiator ratio is found to be 667.25, which corresponds to 163.2 mol of initiator. This ratio is the same as for a cooling water temperature of 10 °C, suggesting that the cooling water temperature does not impact the optimal ratio under these conditions.

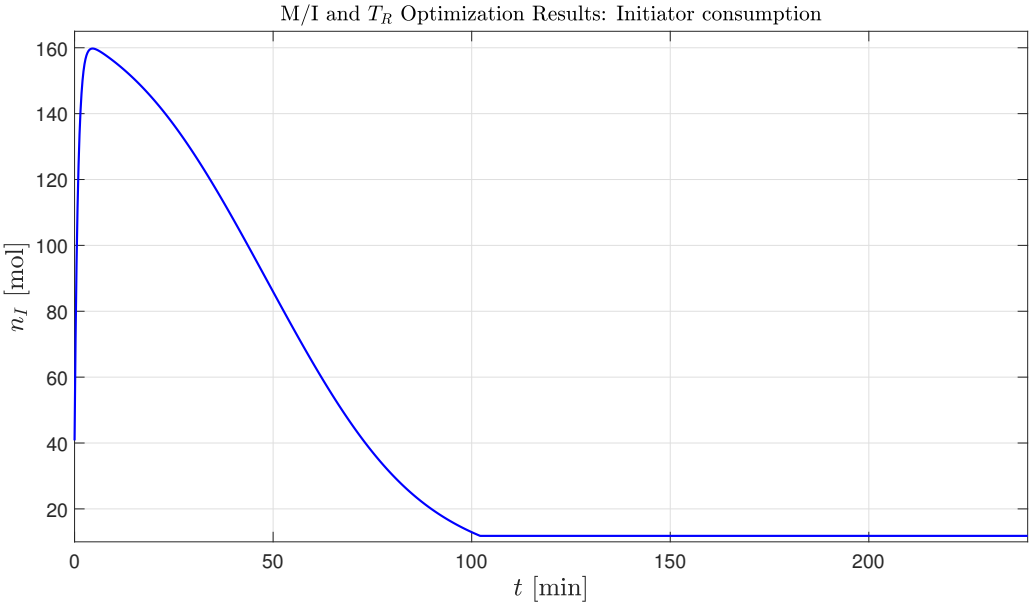


Figure C.1: Initiator consumption for the optimization of the monomer to initiator ratio and temperature profile.

Figure 4.5 shows the optimal temperature profile. The optimal initial reactor temperature $T_{R,0}$ was calculated to be 120 °C. From here, the temperature increased throughout the batch time until the upper bound was reached at 91.25 min. This means that the reactor temperature constraint was active for the last 11 min of the process, as was the case for a lower cooling water inlet temperature. This is the same reactor temperature profile that was identified for a cooling water temperature of 10 °C.

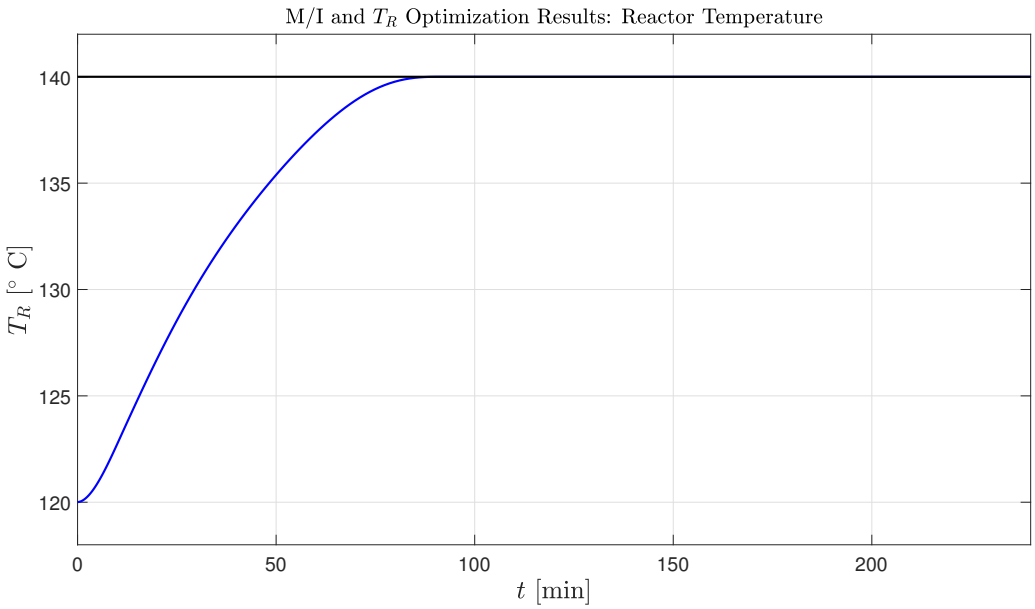


Figure C.2: Reactor temperature profiles for the optimization of the monomer to initiator ratio and temperature profile.

The jacket temperature is initialized at 60 °C and then increases until it ends at 140 °C to match the reactor temperature, as illustrated in Figure C.3. Since no cooling water flow rate is calculated, no heat is removed from the system so the jacket heats up as the reactor transfer heats to it. This profile is identical to the one found for the lower cooling water temperature.

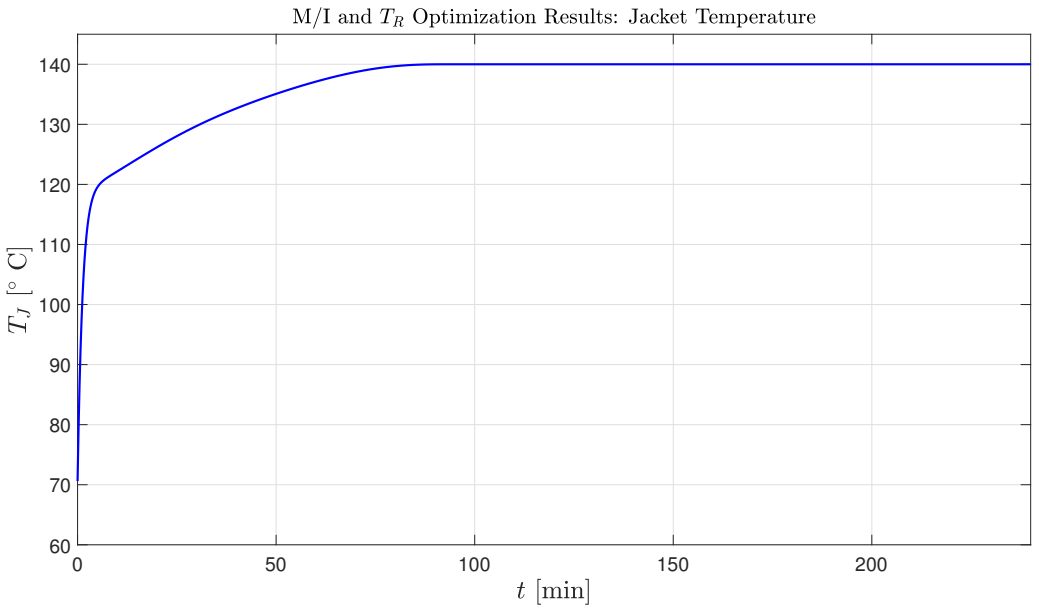


Figure C.3: Jacket temperature profiles for monomer to initiator ratio and temperature profile optimization

Figure C.4 illustrates how the number average molecular weight changes throughout the batch time.

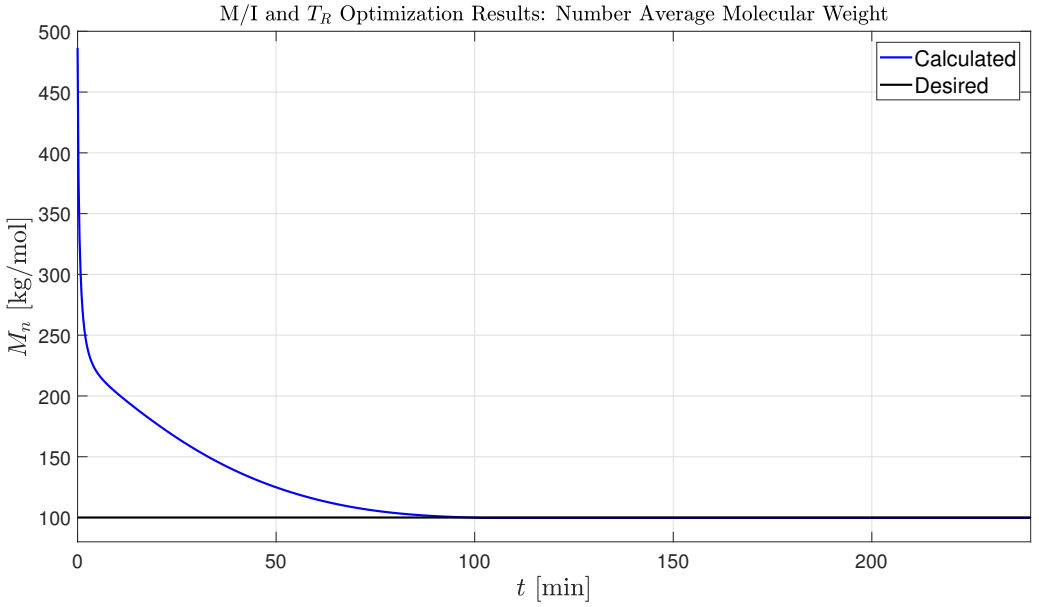


Figure C.4: Number molecular weight for monomer to initiator ratio and temperature profile optimization.

The final number average molecular weight is 99.71 kg/mol, which is within 0.29% of the required value; this is identical to the results for a lower cooling water inlet temperature. The corresponding weight average molecular weight is 180.4 kg/mol. The resulting polydispersity index is 1.81, which is the same as before.

The optimal batch time is 102.3 min, suggesting that the cooling water inlet temperature does not have a large impact on the batch time when perfect temperature control is assumed. Figure C.5 shows the conversion of the monomer to the polymer throughout the batch.

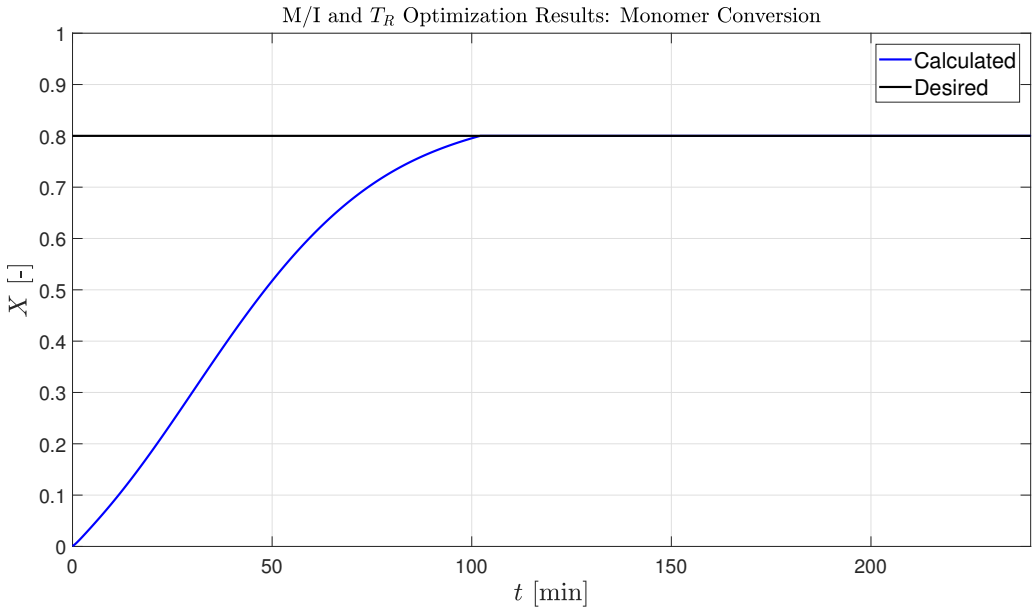


Figure C.5: Conversion rate for monomer to initiator ratio and temperature profile optimization.

These results suggest that for the problem formulation used in the offline optimization the cooling water inlet temperature has little effect on the batch time or the product quality. This is unsurprising because of the assumptions made here, namely perfect temperature control, which means that the cooling water flow rate is not calculated; consequently, what temperature the cooling water is will have no impact on the outcome. Perfect temperature control aslo implies that regardless of what occurs in the process, the reactor temperature will be maintained at the desired value. This means that the system is essential insensitive to disturbances in the temperature (either in the cooling water inlet or the reactor feed). Therefore, it makes sense that a higher cooling water inlet temperature does not impact the results.

C.2 DRTO and NMPC Layers Results

The following sections present the plots of the results of other simulations conducted using the DRTO and NMPC layers. The results for varying cooling water temperatures at different points in the batch for a cooling water flow rate upper limit of 4.3 $kgs^{-1}$are given.

C.2.1 Disturbance 2: Cooling Water Temperature

C.2.1.1 Varying Cooling Water Temperature

First a decrease of -5 °C in the cooling water inlet temperature is simulated followed by a +5 °C increase in the cooling water inlet temperature. These changes are implemented at various points throughout the batch to find where the process is most sensitive to a disturbance in the cooling water inlet temperature. Here a cooling flow rate upper bound of 4.3 kgs^{-1} is used. The exact values from these simulations are reported in Chapter 6.

Decrease in Temperature:

For this cooling constraint, a decrease in the cooling water inlet temperature has minimal impact on the reactor temperature profile as shown in Figure C.6.

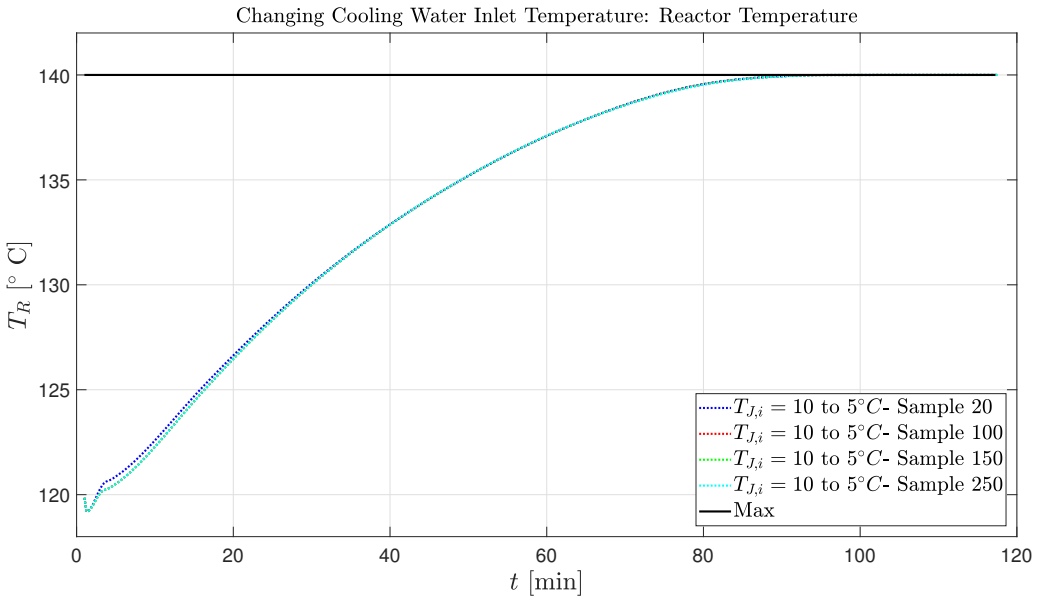


Figure C.6: Affect of decreasing the cooling water temperature at different times on the reactor temperature.

It is difficult to differentiate the four profiles from one another, other than during the first twenty minutes of the batch. This demonstrates that when the cooling water flow rate is sufficient, the impact of a decrease in the cooling water inlet temperature is negligible. When the disturbance occurs at sample 20, the reactor temperature increases a tiny bit faster than the the other temperature profiles. This is caused by the cooling water flow rate being lower in this simulation in comparison to the other simulations. The other cases experience a change in the cooling water flow rate later in the batch, which does not affect the reactor temperature profile.

Figure C.7 shows that the jacket temperature is virtually the same throughout the batch. As is the case for the reactor temperature profiles, the only noticeable deviation is between the sample 20 simulation and the others. In this instance, the jacket temperature for sample 20 is higher at the start of the batch than for the other simulations. This is due to the decreased cooling water flow rate, which causes the jacket temperature to heat up to a higher temperature.

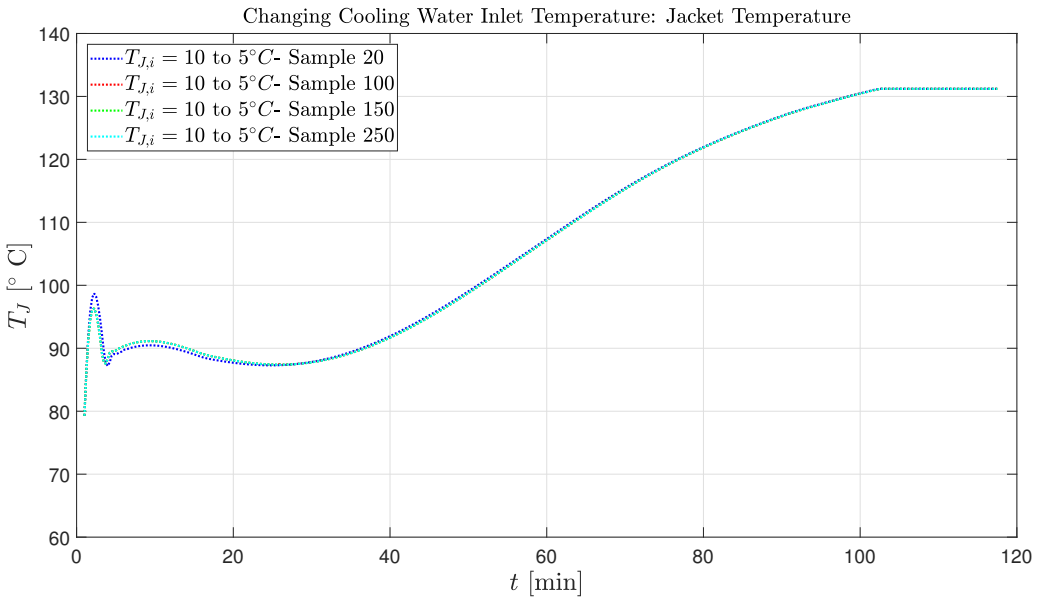


Figure C.7: Affect of decreasing the cooling water temperature at different times on the jacket temperature.

The cooling water flow rate is shown for each of the simulations in Figure C.8. The cooling water flow rate decreases immediately after the cooling water temperature decreases; this demonstrates that the controller is quick to respond to a change in the cooling water inlet temperature. For all of these simulations, the actual and the required cooling water flow rates are nearly identical; this is because the cooling constraint is sufficiently large so the constraint is never active for the tested cooling water inlet temperature. After 64 minutes into the batch, the cooling water flow

rates for all simulations are identical.

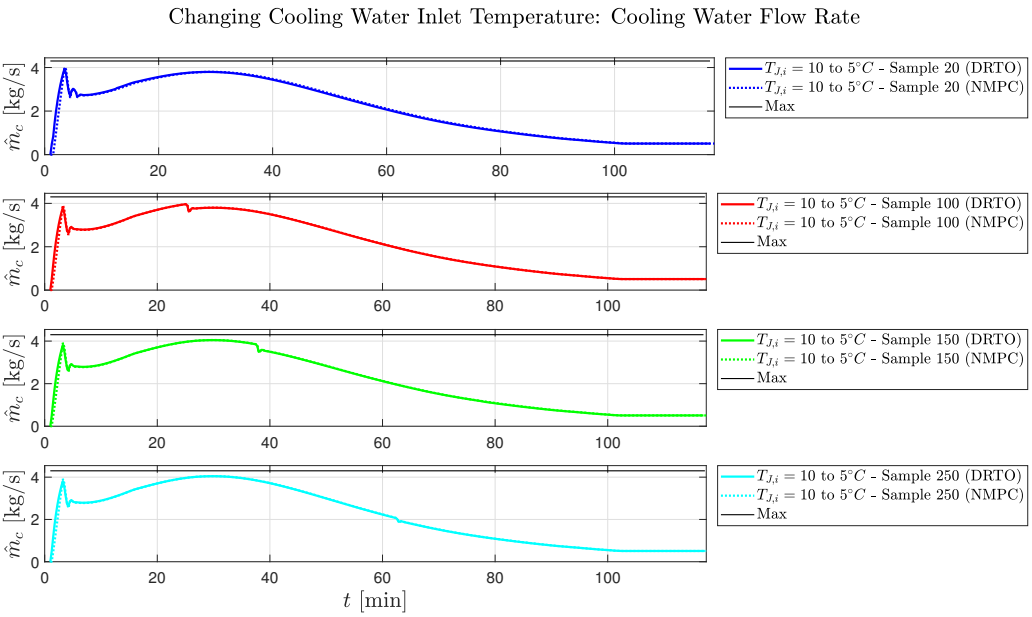


Figure C.8: Affect of decreasing the cooling water temperature at different times on the cooling water flow rate.

The batch time is the same in all cases as Figure C.9 demonstrates.

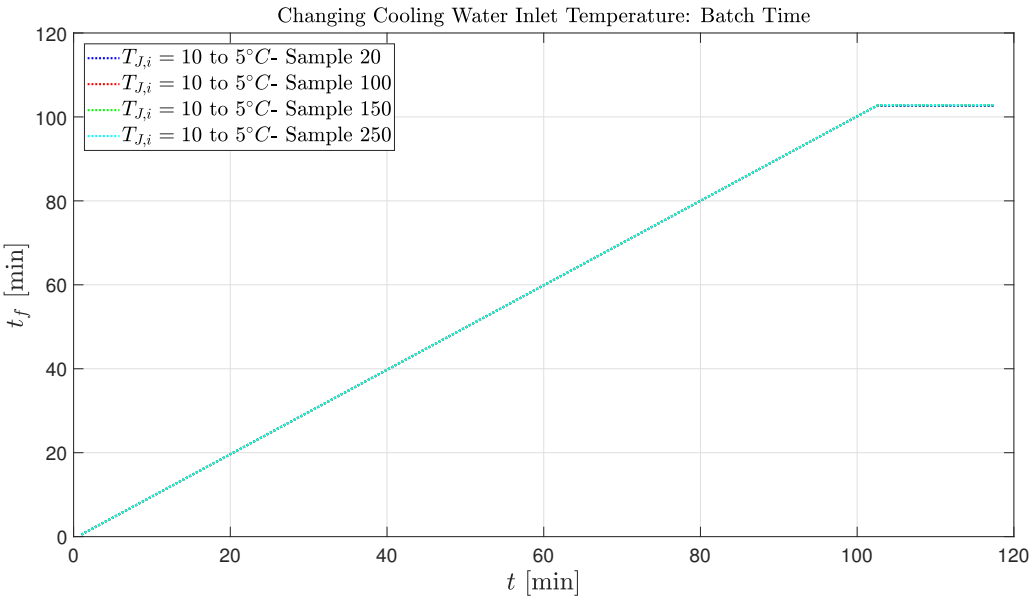


Figure C.9: Affect of decreasing the cooling water temperature at different times on the batch time.

The reactor temperature profiles are essentially unaltered by the decrease in the cooling water inlet temperature, so it is logical that the batch times are also unaffected. Compared to a constant cooling water inlet temperature, the batch time is about the same. This shows that a 5 °C cooler stream of cooling water does not help to shorten the batch time. The process instead decreases the cooling water flow rate in response to the colder cooling water inlet temperature.

Since the reactor temperature profiles and batch times are unchanged, the final number average molecular weights are indistinguishable as shown in Figure C.10. In all four simulations, the final number average molecular weight is within the desired range.

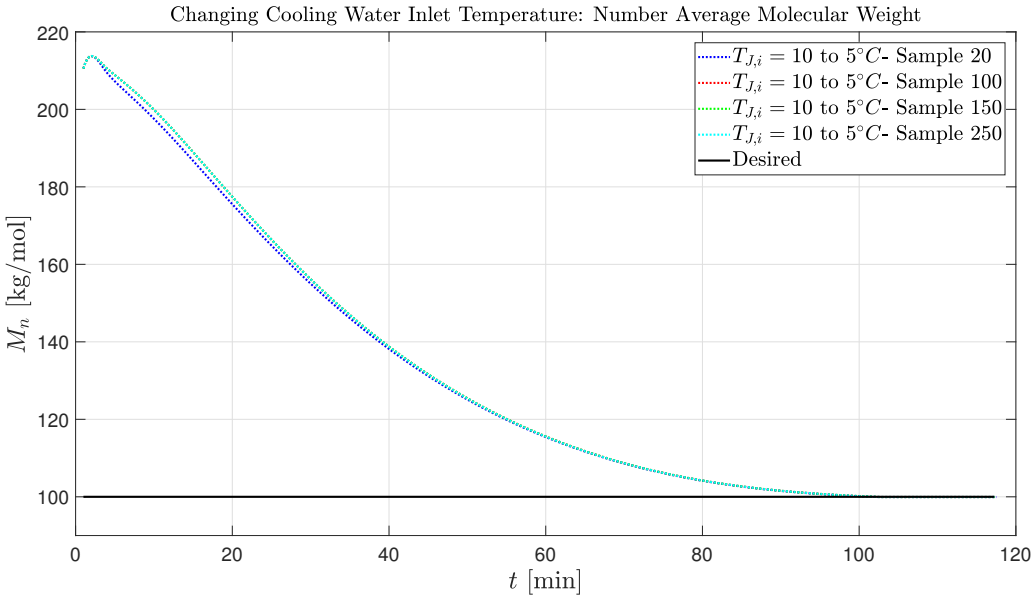


Figure C.10: Affect of decreasing the cooling water temperature at different times on the number average molecular weight.

These simulations illustrate that a decrease in the cooling water inlet temperature has less impact on the system when the cooling constraint is not active in comparison to when the cooling constraint is 4.0 kgs^{-1}. It is expected that a decrease in the cooling water inlet temperature for this cooling constraint would not have a large impact. However, if the initial cooling water inlet temperature were higher, say 20 °C, a decrease of 5 °C would be the difference between the batch producing the desired product and finishing in a reasonable time versus the batch producing a bad product and taking three times longer.

Increase in Temperature:

Figure C.11 shows that an increase of 5 °C in the cooling water inlet temperature has a larger impact on the temperature profile than does a decrease. The reactor temperature profiles differ the most from one another at the beginning of the batch. By approximately minute 60 in the batch, all the profiles are interchangeable with the final temperature ending just below the upper limit. When the disturbance occurs at sample 20 or 100 the effect on the temperature profile is the largest, suggesting that an increase in the cooling water inlet temperature early in the batch has a larger effect.

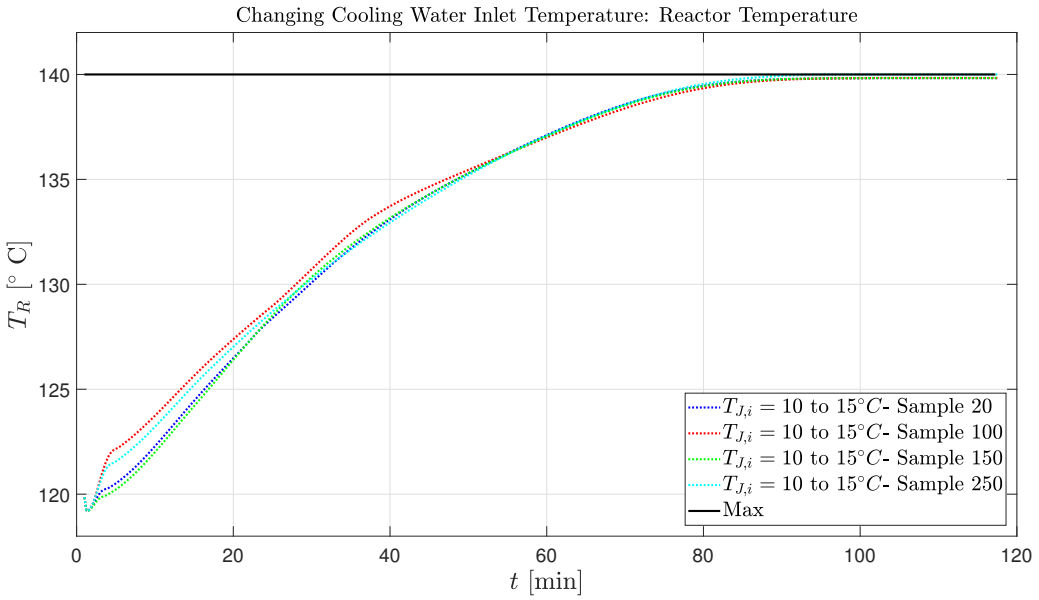


Figure C.11: Affect of increasing the cooling water temperature at different times on the reactor temperature.

Figure C.12 illustrates that the jacket temperature responds similarly to the reactor temperature profile. The final jacket temperature ends at about 130 °C in all cases, which is the same final temperature for a cooling water inlet temperature of 10 °C; this suggests that the system is able to compensate for the increased cooling water inlet temperature by increasing the cooling water flow rate.

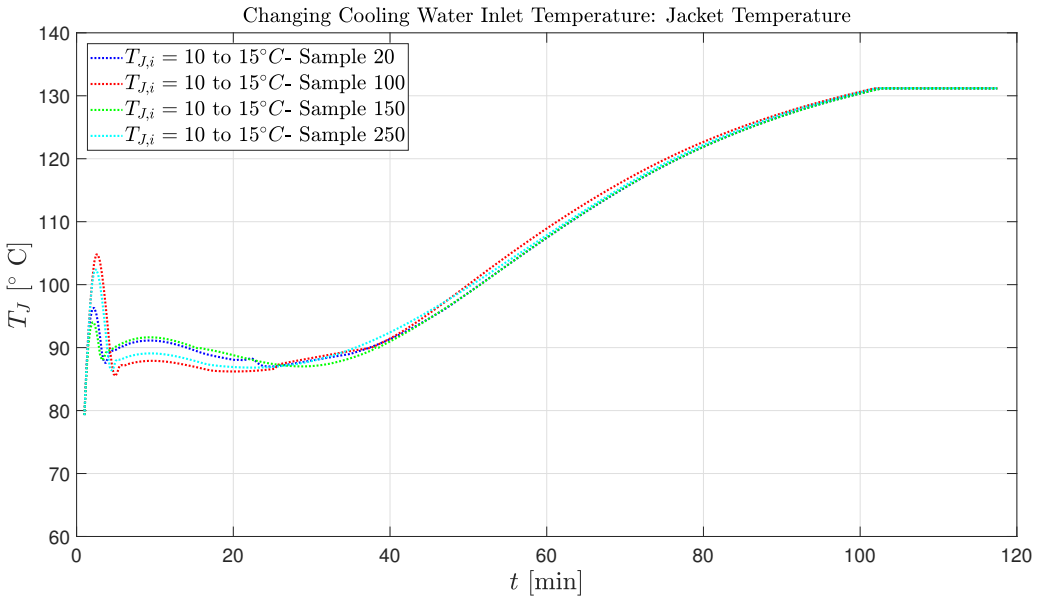


Figure C.12: Affect of increasing the cooling water temperature at different times on the jacket temperature.

The cooling capacity of the system is able to handle the increased cooling water inlet temperature as shown in Figure C.13.

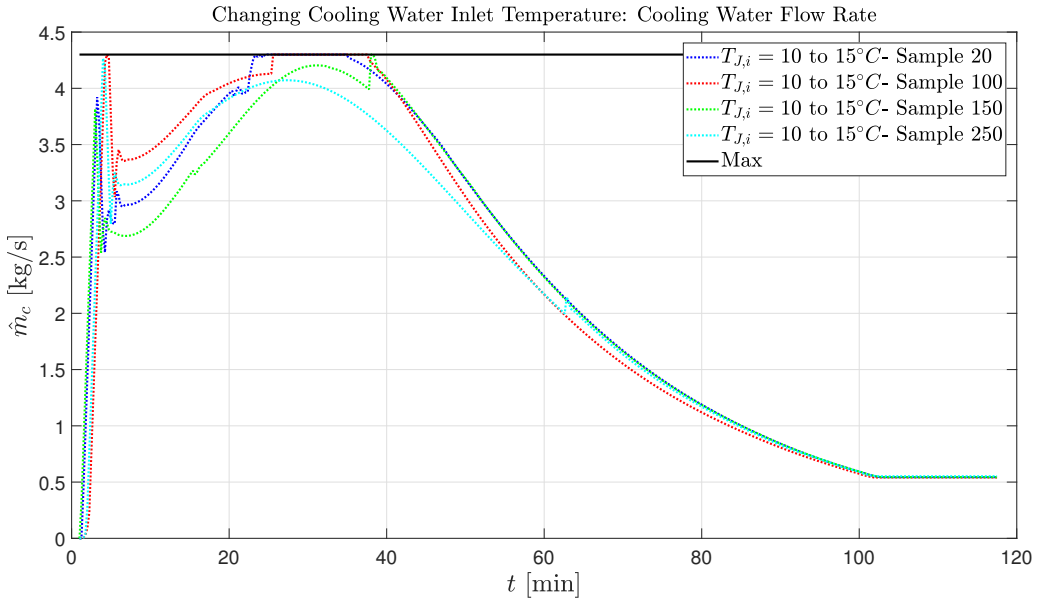


Figure C.13: Affect of increasing the cooling constraint at different times in the batch on the cooling water flow rate.

This is unsurprising considering that previous simulations showed that this constraint could handle a cooling water inlet temperature of 15 °C. When the cooling water inlet temperature increases at sample 20 or 100, the constraint is active for part of the batch. This is because the cooling flow rate is already near the upper limit at these points in the batch so when the cooling water temperature increases, the cooling flow rate has to increase to track the optimal reactor temperature. When the increase occurs later in the batch the cooling water flow rate is far enough below the limit that the constraint is never active.

Despite the impact on the reactor and the jacket temperature profile, the batch time is approximately the same as when the cooling water inlet temperature is a constant, as demonstrated in Figure C.14.

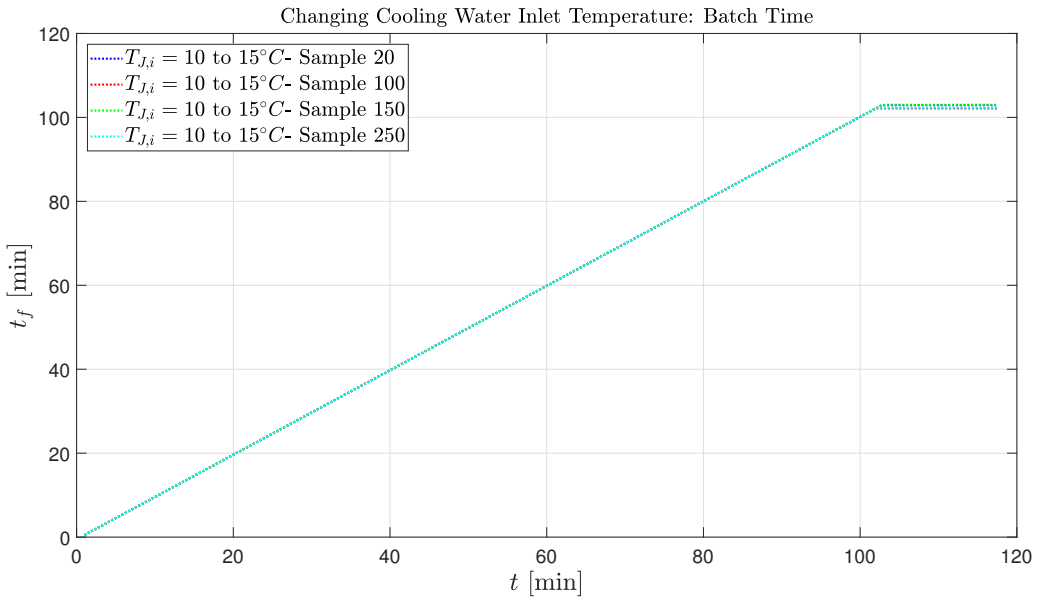


Figure C.14: Affect of increasing the cooling constraint at different times in the batch on the batch time.

The final number average molecular weight is unaffected as shown in Figure C.15. All four simulations produce a final number average molecular within the allowable range.

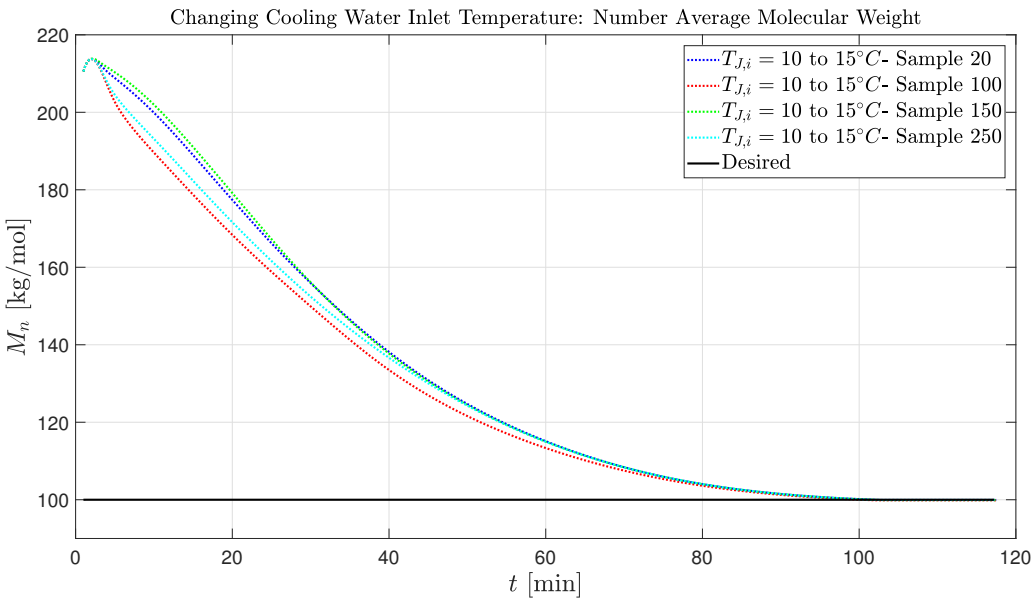


Figure C.15: Affect of increasing the cooling constraint at different times in the batch on the number average molecular weight.

The results suggest that if the cooling capacity is sufficient, the system can handle a increase in the cooling water inlet temperature at any point throughout the batch. In addition, the batch time and produced polymer will not be greatly affected.

C.3 Cost Benefit Analysis

A cost benefit analysis is performed to motivate the construction and implementation of a two-level advanced control structure over the current approach of using a fixed recipe. A recipe is developed using offline optimization for a worst case scenario so that the process will always operate within safe margins; this recipe is then followed for every batch produced. While this approach may work for most of the batches, it has a huge disadvantage in that the operating conditions do not change to respond to any disturbances.

An offline recipe is constructed for a worst case scenario for the case study used in this work; these operating conditions are then compared to a series of online simulations to illustrate the potential benefits of using a two-level advanced control structure. An illustration of the different simulations run for the cost benefit analysis is shown in Figure C.16.

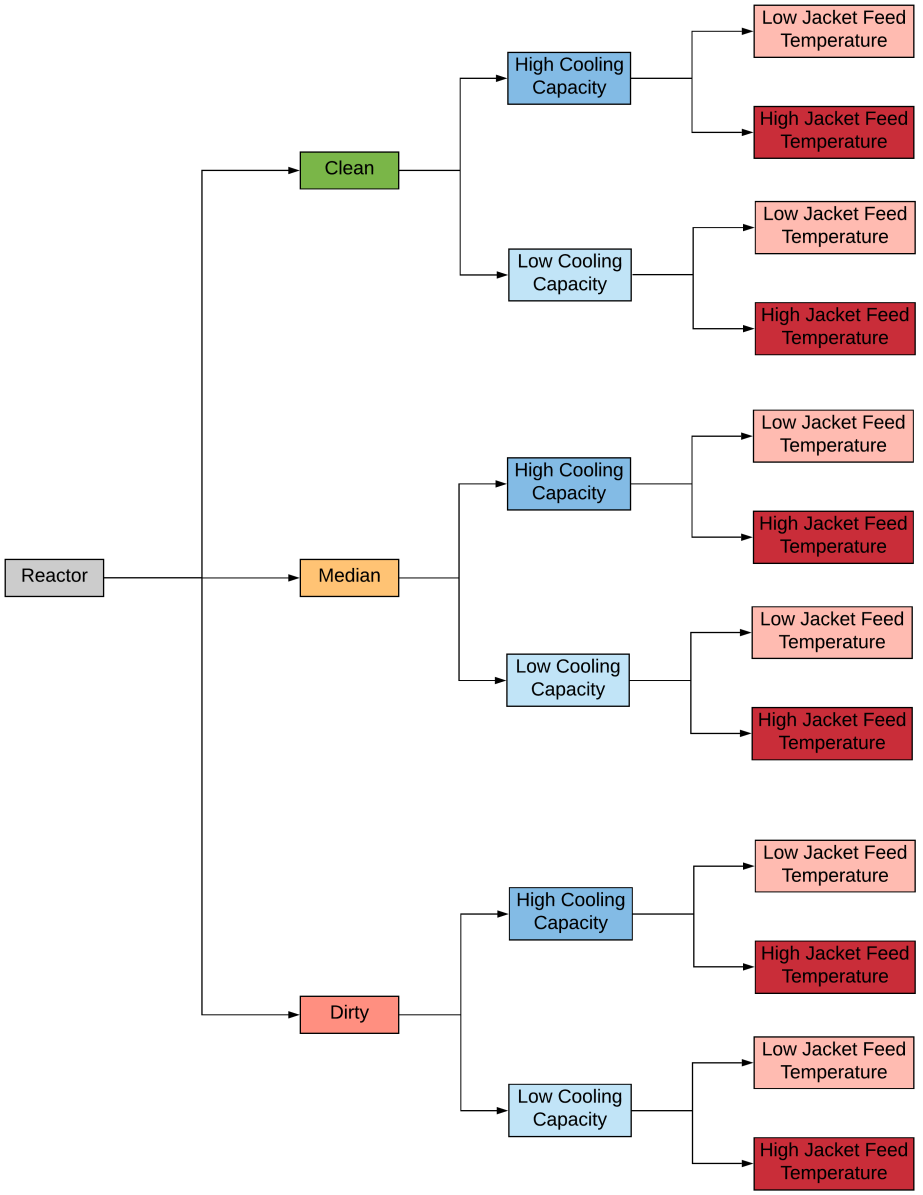


Figure C.16: Representation of all the simulations run using the full control structure.

C.3.1 Offline vs Online Results

C.3.1.1 High Cooling Capacity and Hot Cooling Water Temperature

The offline recipe is now compared to simulation results for a high cooling capacity of 5.0 $\mathrm{kg\,s^{-1}}$ and a hot cooling water temperature of 20 °C. This is a higher cooling capacity but the same cooling water temperature that is used to calculate the offline recipe.

The optimal monomer to initiator ratios for a clean, average, and dirty reactor are 644.4, 555.4, and 600.8, respectively. Figure C.17 shows the initiator consumption for each of the simulations. A clean reactor has a larger monomer to initiator ratio compared to the offline recipe which has a ratio of 600.8; therefore, initiator would be wasted in this instance. An average reactor has a lower monomer to initiator ratio suggesting that insufficient initiator would be used if the offline recipe is followed. A dirty reactor has the same ratio as the offline recipe, so in this instance there would be little benefit to online optimization unless a disturbance occurs.

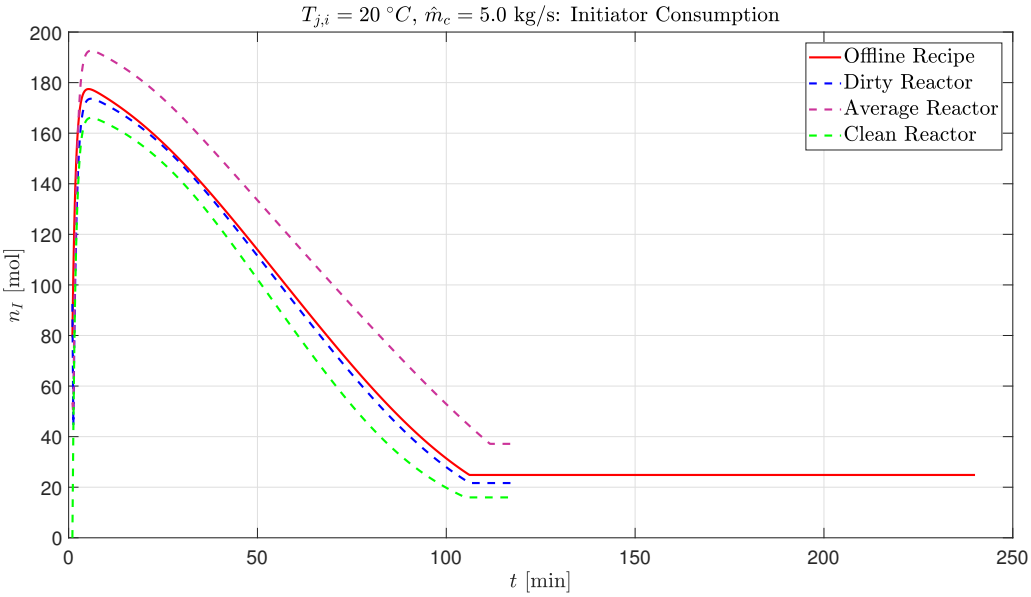


Figure C.17: Comparison of the initiator consumption for a high cooling capacity and hot cooling water inlet temperature.

The optimal initial reactor temperatures are 119.2, 126.1, 125.5 °C as seen in Figure C.18. For this scenario, the dirty reactor has a similar reactor temperature profile to the offline recipe after approximately thirty minutes; therefore, the offline reactor temperature profile could be used and still have good results. However, for a clean reactor or an average reactor, the reactor temperature profiles differ considerably from the offline recipe. The average reactor has a surprising profile compared to a

dirty or clean reactor; this is likely caused by the operating conditions producing an optimization problem that has a flat objective function making it challenging for the algorithm to locate a minimum. Therefore, the located minimum may not be the actual minimum, resulting in this odd reactor temperature profile for the average reactor.

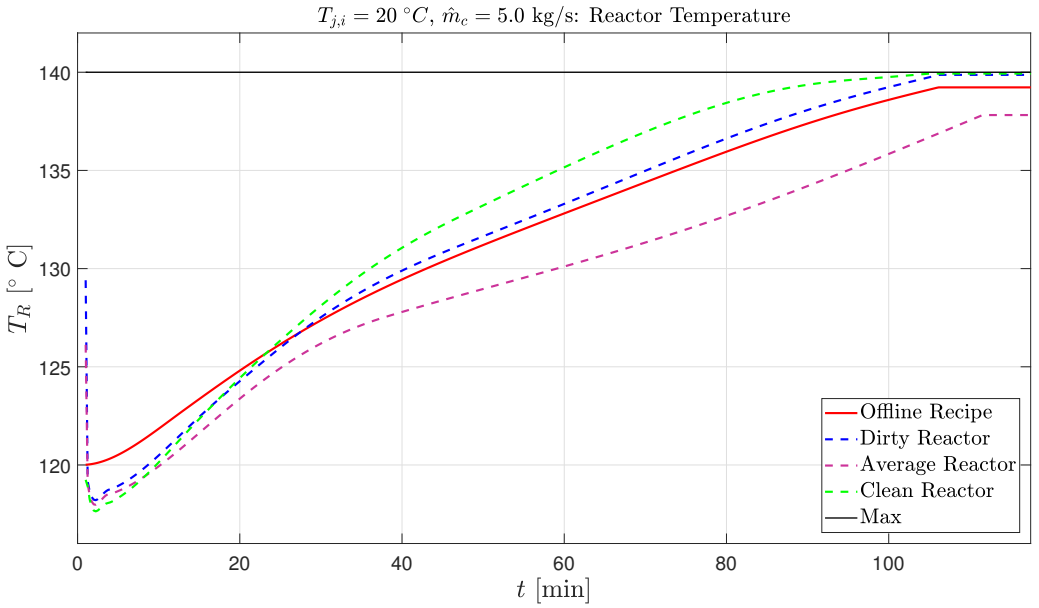


Figure C.18: Comparison of the reactor temperature for a high cooling capacity and hot cooling water inlet temperature.

The cooling water flow rates are shown in Figure C.19 and help explain the reactor temperature profiles. The clean reactor has the lowest cooling water flow rate, which is expected since the heat transfer coefficient is higher so less cooling water is required to track the reactor temperature profile. Despite the reactor temperature profile not falling between the clean and dirty profiles, the average reactor cooling water flow rate falls roughly between the clean and dirty cooling water flow rates.

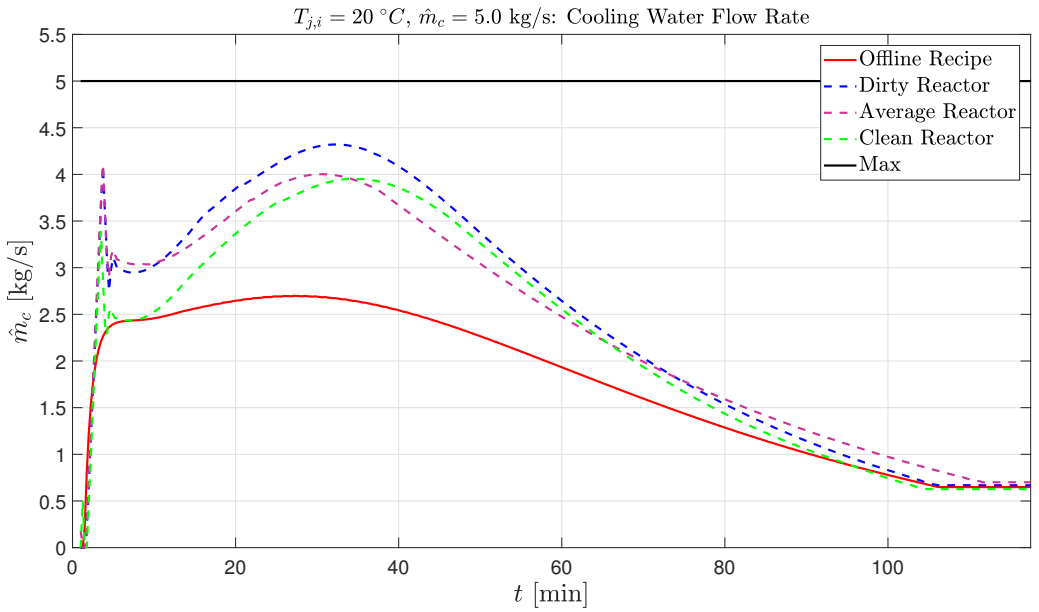


Figure C.19: Comparison of the cooling water flow rate for a high cooling capacity and high cooling water inlet temperature.

All four simulations are able to produce the required final number average molecular weight as demonstrated in Figure C.20.

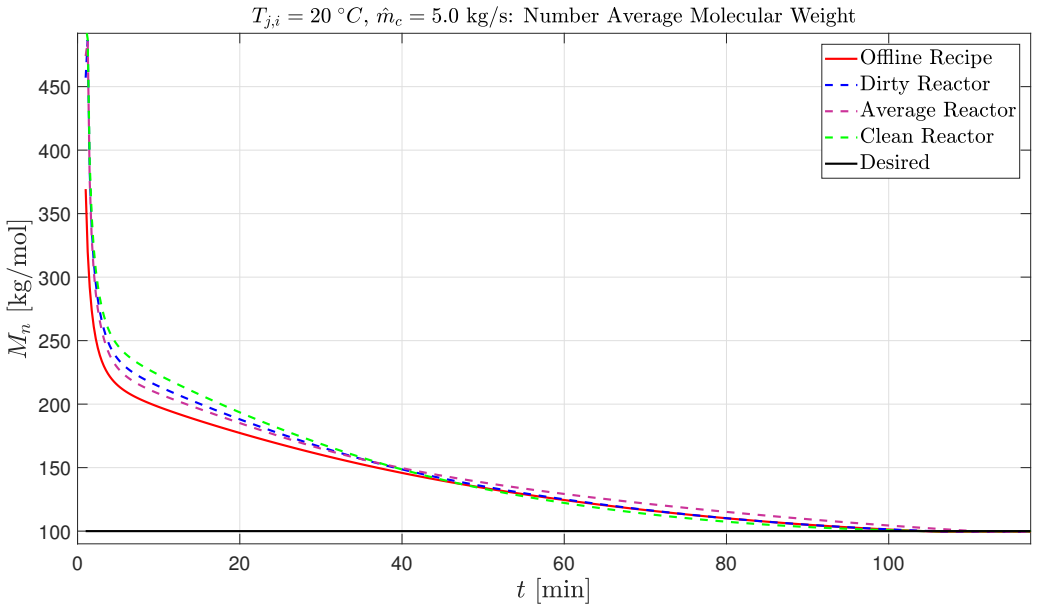


Figure C.20: Comparison of the number average molecular weight for a high cooling capacity and high cooling water inlet temperature.

This is the only cooling water constraint tested that was able to produce a polymer product within the allowable range for this cooling water inlet temperature. The final number average molecular weight closest to the desired value comes from a clean reactor in this instance.

The batch times for the four simulations are shown in Figure C.21 and reveal that the shortest batch time is for a clean reactor. The longest batch time is for the average reactor, this is unexpected but given the calculated reactor temperature profile deviates from the others not surprising. This is likely a product of the operating conditions resulting in a flat objective function making it difficult for the algorithm to locate the optimum. The dirty reactor has an identical batch time to the offline predicted batch time, further illustrating that the use of the offline recipe for this scenario would be okay. The clean reactor gives a batch time 1.3 minutes shorter than the offline recipe, which would translate into a loss of 62 batches per year per reactor if the offline recipe is used.

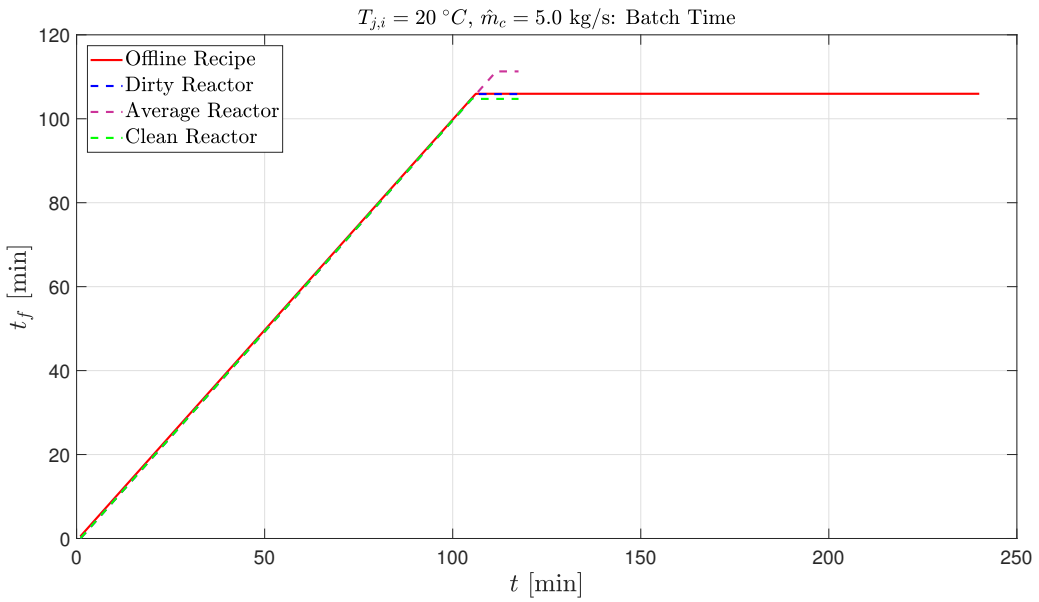


Figure C.21: Comparison of the offline recipe to the full control structure batch time for a high cooling capacity and high cooling water inlet temperature for three different reactor conditions.

These results indicate that in some circumstances the use of online optimization would be beneficial but in others it would make little difference. However, these simulations do not account for any disturbances occurring during the batch.

C.3.1.2 Medium Cooling Capacity and Hot Cooling Water Inlet Temperature

The offline recipe is now compared to simulation results for a medium cooling capacity of 4.5 $\mathrm{kg\,s^{-1}}$ and a hot cooling water temperature of 20 °C. This is the same cooling capacity and cooling water temperature that was used to calculate the offline recipe.

Figure C.22 shows the initiator consumption rate for each scenario. The optimal monomer to initiator ratios are found to be 643.6, 639.8, and 615.9 for a clean, average and dirty reactor, respectively. This ratios are all larger than the offline recipe of 600.8 so initiator would be wasted if the offline recipe is used for any of these conditions. However, for a dirty reactor, the initiator consumption is indistinguishable from the offline recipe.

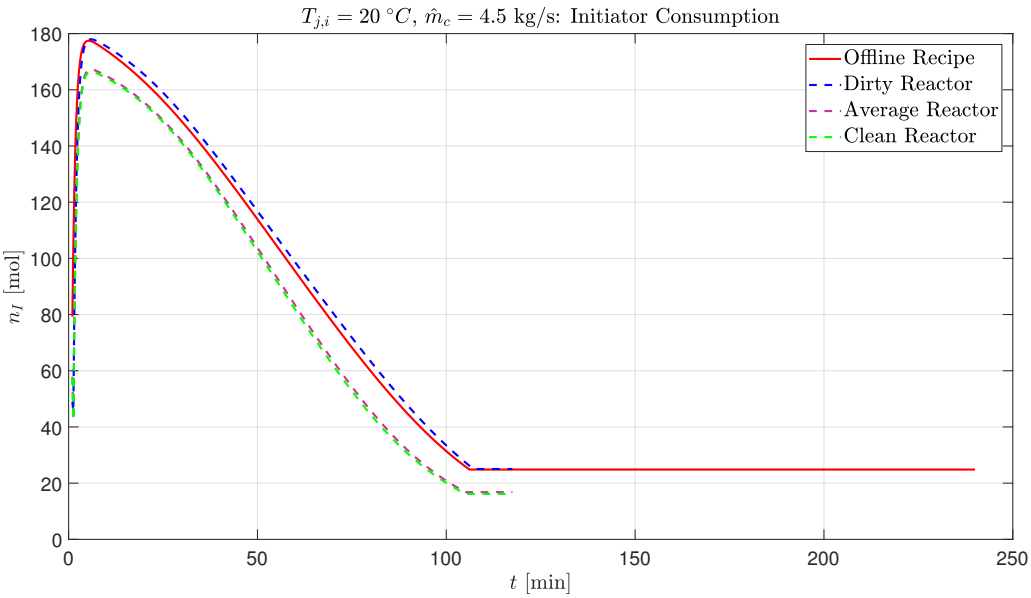


Figure C.22: Comparison of the initiator consumption for a medium cooling capacity and hot cooling water inlet temperature.

The reactor temperature profiles are plotted in Figure C.23. A clean and average reactor have nearly identical reactor temperature profiles. The dirty reactor temperature profile is similar to the offline recipe after about thirty minutes with some minor differences. This is not surprising considering the simulation conditions are nearly identical.

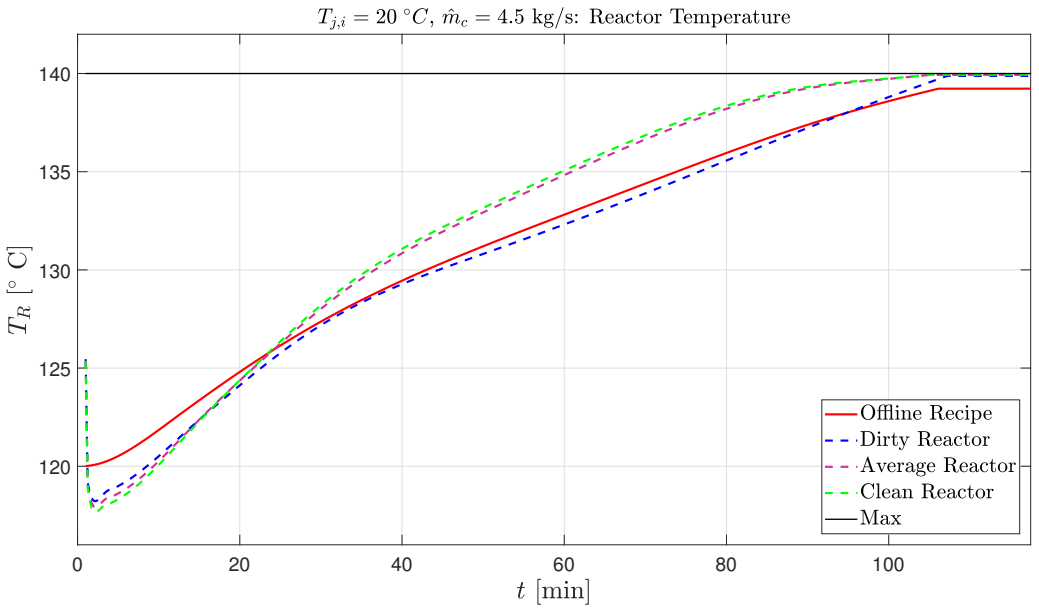


Figure C.23: Comparison of the reactor temperature for a medium cooling capacity and hot cooling water inlet temperature.

Figure C.24 illustrates the cooling water flow rates.

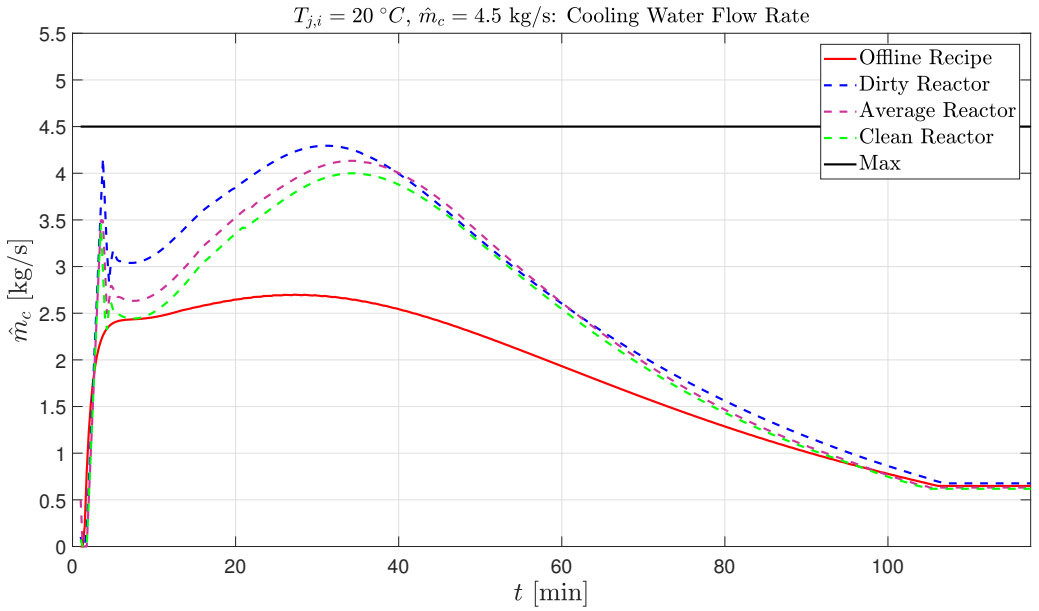


Figure C.24: Comparison of the cooling water flow rate for a medium cooling capacity and high cooling water inlet temperature.

In this instance, the cooling water flow rates behave as expected with the dirty reactor having the highest flow rate and the average reactor falling in between the dirty and clean reactor flow rates. The dirty reactor has the lowest coefficient of heat transfer meaning that more cooling water is required to remove the same amount of heat than for a clean reactor.

The number average molecular weights throughout the batch are shown in Figure C.25. All four simulations result in a final number average molecular weight within the desired range. A clean reactor results in the best product quality.

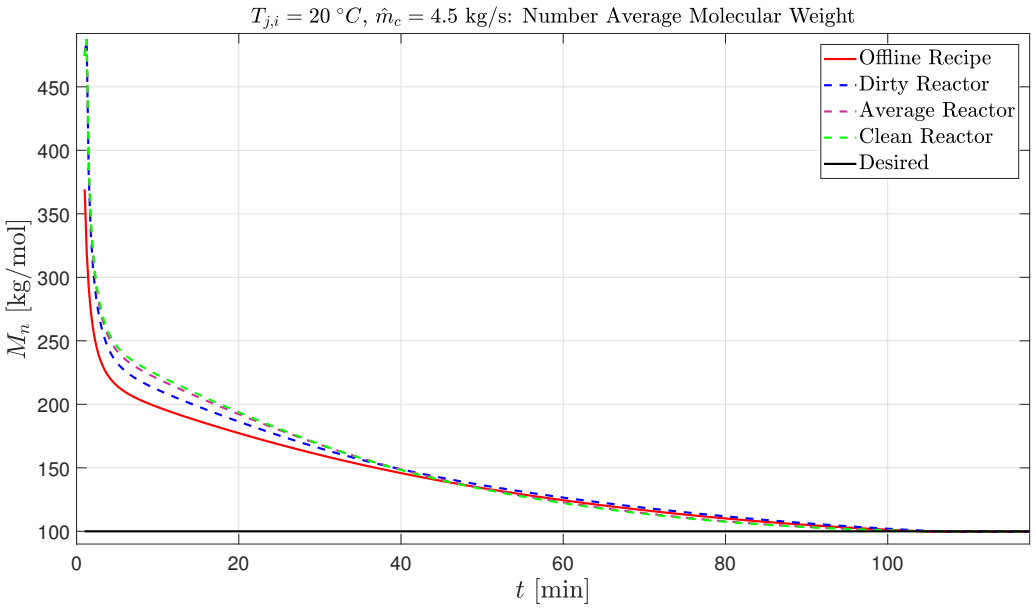


Figure C.25: Comparison of the number average molecular weight for a medium cooling capacity and high cooling water inlet temperature.

For this scenario the batch times are as expected, with the shortest being for a clean reactor and the longest being for a dirty reactor, as shown in Figure C.26. The clean reactor has a batch time 1.2 minutes shorter than the offline recipe, which means that 61 more batches could be run annually for each reactor. For a dirty reactor, the batch would be terminated prematurely before an 80% conversion is reached, resulting in a wasted batch since the product would be out of specification.

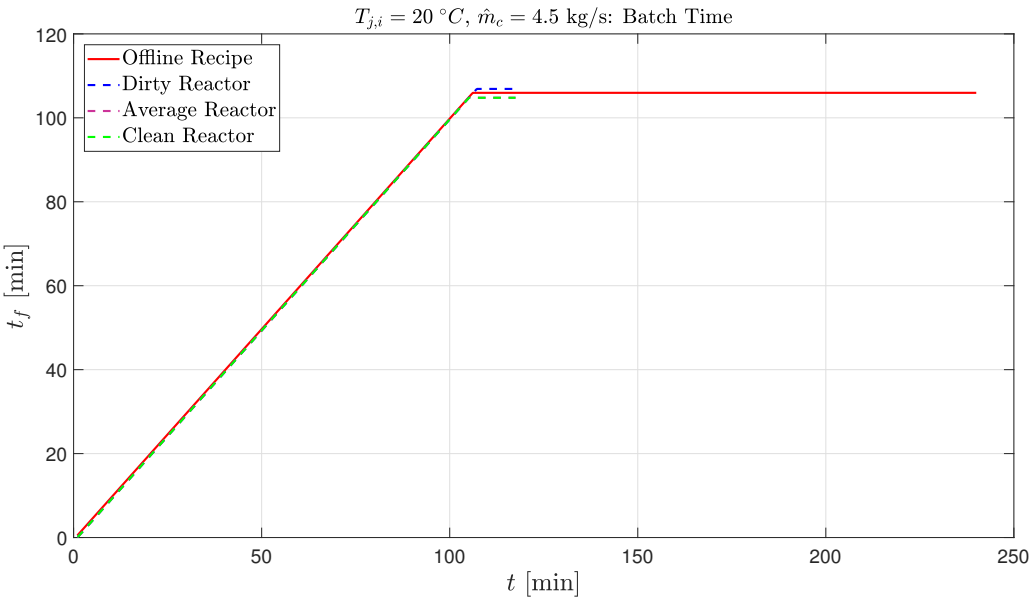


Figure C.26: Comparison of the batch time for a medium cooling capacity and high cooling water inlet temperature.

C.3.2 Low Cooling Capacity and High cooling water inlet Temperature

The offline recipe is now compared to simulation results for a low cooling capacity of 4.0 $\mathrm{kg s^{-1}}$ and a hot cooling water temperature of 20 °C. This is a lower cooling capacity than used in the offline recipe but the same cooling water temperature.

Figure C.27 shows the initiator consumption rate for the different conditions. The optimal monomer to initiator ratios for a clean, average, and dirty reactor are found to be 714.9, 651.6, and 681.8, respectively. This ratios are all significantly higher than the offline ratio and would result in a lot of wasted initiator if the offline recipe were followed. As can be seen in the plots, the online simulations result in all the initiator being consumed whereas the online recipe predicts leftover initiator.

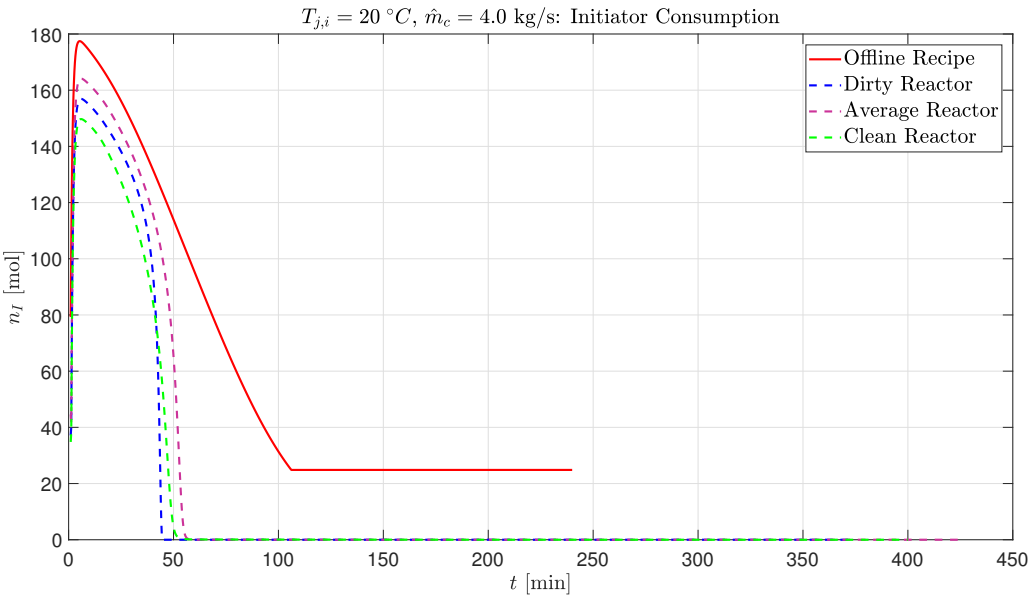


Figure C.27: Comparison of the initiator consumption for a low cooling capacity and high cooling water inlet temperature.

The reactor temperature profiles are plotted in Figure C.28 and show that the online simulations all result in a violation in the upper bound.

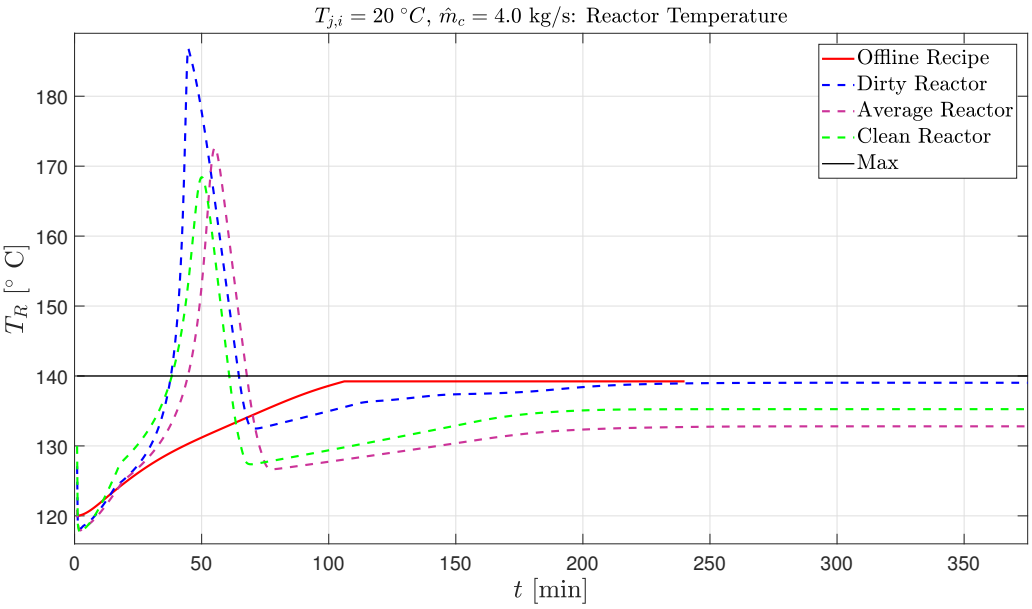


Figure C.28: Comparison of the reactor temperature for a low cooling capacity and high cooling water inlet temperature.

In comparison, the offline recipe does not predict this. This suggests that the offline recipe is not able to fully account for the limit on the system's cooling capacity. This is not surprising because the full predictions are done by the DRTO layer, which is not constructed to determine the optimal cooling water flow rate. Therefore, the DRTOs predictions for the required cooling water flow rate are not the best. This is why the offline recipe does not predict a reactor temperature profile anywhere near the online simulations.

The corresponding cooling water flow rates are shown in Figure C.29. All of the online cooling water flow rates calculated in the NMPC level are actively constrained at the beginning of the batch. The required cooling water flow rates calculated by the DRTO all violate the upper limit; this suggests that for this cooling constraint upper limit, the system has inadequate cooling to track the reactor temperature profiles. This explains why the reactor temperature profiles for the simulated results are so far from the optimal trajectory. The offline recipe again under predicts the required cooling water flow rate.

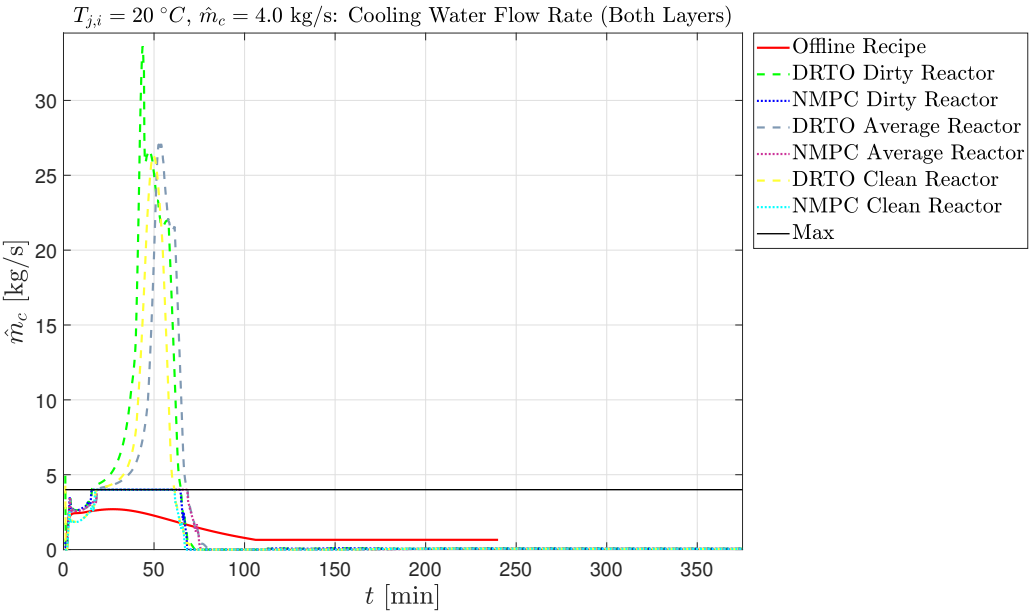


Figure C.29: Comparison of the cooling water flow rate for a low cooling capacity and high cooling water inlet temperature.

Because the reactor temperature profiles deviate from the trajectory, the polymer quality is outside the necessary range as shown in Figure C.30. The clean reactor gives a polymer product 17.2% away from the desired value, with the average and dirty reactors being even further away.

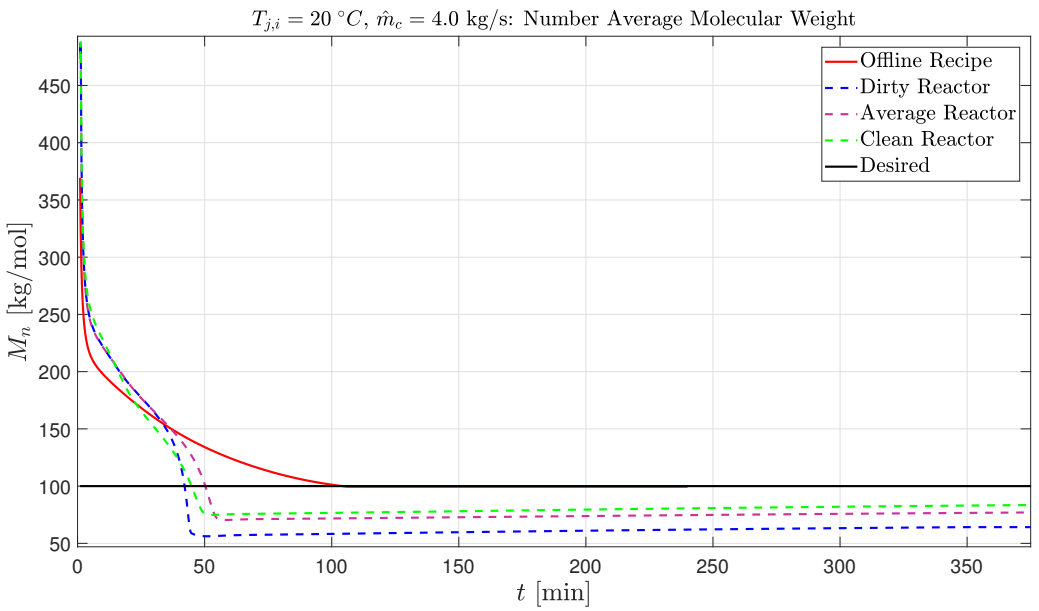


Figure C.30: Comparison of the number average molecular weight for a low cooling capacity and high cooling water inlet temperature.

In this case, the simulated batch times are nearly three to four times longer than the offline recipe, as seen in Figure C.31.

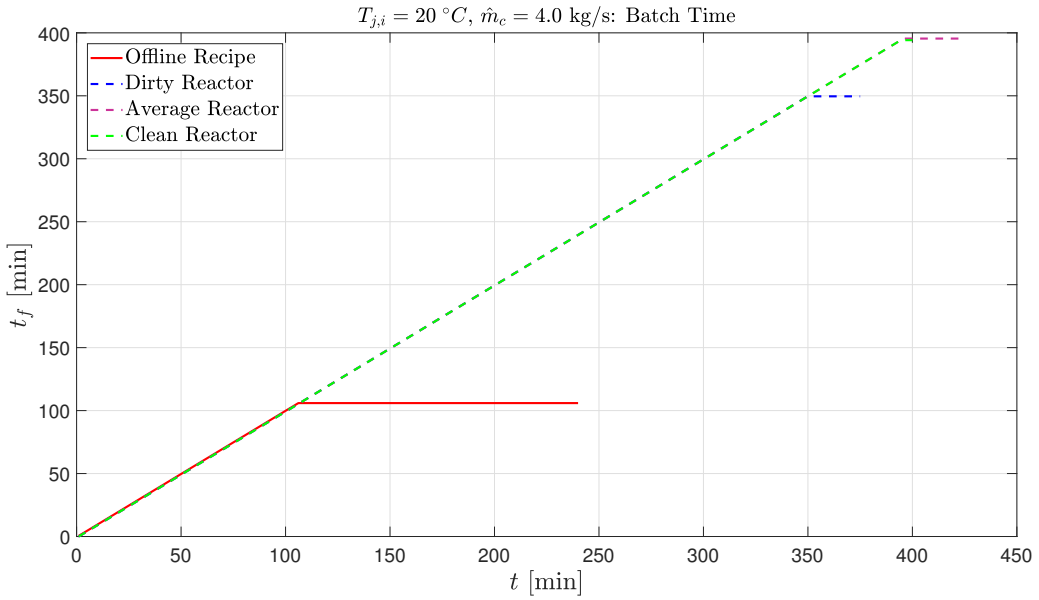


Figure C.31: Comparison of the offline recipe to the full control structure batch for a low cooling capacity and high cooling water inlet temperature for three different reactor conditions.

This is caused by the reactor temperature deviation from the optimal trajectory.

These results show that these operating conditions are infeasible for this system. A cooling limit of 4.0 $kg s^{-1}$ is too tight to allow the system the required cooling water for a cooling water inlet temperature of 20 °C. So regardless of the method of control used here, the system will not produce the product. These simulations show the system's limitations and serve as a reminder of the importance of including safety constraints in a controller.

C.3.2.1 Results Summary

The annual cost of initiator for the different scenarios is calculated here. First the required mass of initiator from each simulation is found; this was an optimized input value calculated by the DRTO layer when each simulation was run. This values are summarized in Table C.1. The cost of initiator per batch is then calculated assuming an initiator cost of 2.11$/kg. These values are summarized in Table C.2. Next the number of batches that can be run per year are calculated based on the batch time and the assumption that the plant operates 24/7 and 365 days a year. These batch times are summarized in Table C.3 and the number of batches per year are presented in Table C.4. Finally, the annual cost is calculated using these values, which are shown in Table C.5.

Table C.1: Amount of Initiator Used Per Batch for Different Operating Conditions

	Offline (kg/batch)	High Cooling Capacity, Hot Cooling Water (kg/batch)	High Cooling Capacity, Cold Cooling Water (kg/batch)	Medium Cooling Capacity, Hot Cooling Water (kg/batch)	Medium Cooling Capacity, Cold Cooling Water (kg/batch)	Low Cooling Capacity, Hot Cooling Water (kg/batch)	Low Cooling Capacity, Cold Cooling Water (kg/batch)
Dirty	37.60	37.60	33.24	36.68	37.29	33.13	49.27
Average		40.67	33.29	35.31	42.80	34.67	34.82
Clean		35.05	32.24	35.10	34.68	31.60	34.93

Table C.2: Cost of Initiator Per Batch for Different Operating Conditions

	Offline ($/batch)	**High Cooling Capacity, Hot Cooling Water ($/batch)**	**High Cooling Capacity, Cold Cooling Water ($/batch)**	**Medium Cooling Capacity, Hot Cooling Water ($/batch)**	**Medium Cooling Capacity, Cold Cooling Water ($/batch)**	**Low Cooling Capacity, Hot Cooling Water ($/batch)**	**Low Cooling Capacity, Cold Cooling Water ($/batch)**
Dirty	79.37	79.37	70.16	77.42	78.71	69.94	104.00
Average		85.86	70.28	74.53	90.35	73.18	73.51
Clean		73.99	68.05	74.09	73.22	66.70	73.74

Table C.3: Batch Times for Different Operating Conditions

	Offline (min/batch)	**High Cooling Capacity, Hot Cooling Water (min/batch)**	**High Cooling Capacity, Cold Cooling Water (min/batch)**	**Medium Cooling Capacity, Hot Cooling Water (min/batch)**	**Medium Cooling Capacity, Cold Cooling Water (min/batch)**	**Low Cooling Capacity, Hot Cooling Water (min/batch)**	**Low Cooling Capacity, Cold Cooling Water (min/batch)**
Dirty	106.0	106.9	103.5	105.9	106.6	349.6	108.9
Average		111.3	113.9	104.8	113.9	395.5	104.4
Clean		104.7	104.4	104.8	104.5	394.2	104.6

Table C.4: Total Number of Batches Per Year for Different Operating Conditions

	Offline (batches/yr)	**High Cooling Capacity, Hot Cooling Water (batches/yr)**	**High Cooling Capacity, Cold Cooling Water (batches/yr)**	**Medium Cooling Capacity, Hot Cooling Water (batches/yr)**	**Medium Cooling Capacity, Cold Cooling Water (batches/yr)**	**Low Cooling Capacity, Hot Cooling Water (batches/yr)**	**Low Cooling Capacity, Cold Cooling Water (batches/yr)**
Dirty	4958	4917	5078	4963	4931	1503	4826
Average		4722	4615	5015	4615	1329	5034
Clean		5020	5034	5015	5030	1333	5025

Table C.5: Annual Cost of Initiator for Different Operating Conditions

	Offline ($/yr)	**High Cooling Capacity, Hot Cooling Water ($/yr)**	**High Cooling Capacity, Cold Cooling Water ($/yr)**	**Medium Cooling Capacity, Hot Cooling Water ($/yr)**	**Medium Cooling Capacity, Cold Cooling Water ($/yr)**	**Low Cooling Capacity, Hot Cooling Water ($/yr)**	**Low Cooling Capacity, Cold Cooling Water ($/yr)**
Dirty	393553	390240	356272	384267	388108	105150	501964
Average		405451	324314	373795	416914	97255	370079
Clean		371424	342615	371588	368249	88936	370514

BIBLIOGRAPHY

[1] Almeida, A., Wada, K., and Secchi, A. (2008). Simulation of styrene polymerization reactors: Kinetic and thermodynamic modeling. *Brazilian Journal of Chemical Engineering*, 25(2):337–349.

[2] Amin, A. A. (2016). Advanced Process Control and Real-Time Optimization. online. https://www.controleng.com/single-article/advanced-process-control-and-real-time-optimization/0e7ad7a15a267fed65902439aeefb959.html.

[3] Biegler, L. T. (1984). Solution of dynamic optimization program by successive quadratic programming and orthogonal collocation. *Computers and Chemical Engineering*, 8(3):243–248.

[4] Bousbia-Salah, R., Lesage, F., Hu, G.-H., and Latifi, A. (2017). Dynamic real-time optimization of a batch polymerization process. In *27th European Symposium on Computer Aided Process Engineering*.

[5] Cybernetica AS (2018). Batch Optimization. online. http://cybernetica.no/technology/batch-optimization/.

[6] Fogler, H. S. (2006). *Elements of Chemical Reaction Engineering*. Prentice Hall, fourth edition.

[7] Foss, B. and Heirung, T. A. N. (2013). Merging optimization and control. Trondheim, Norway.

[8] Frounchi, M., Farhadi, F., and Mohammadi, R. P. (2002). Simulation of styrene radical polymerization in batch reactor: A modified kinetic model for high conversion. *Scientia Iranica*, 9(1):86–92.

[9] Gjertsen, F. (2013). Models for on-line control of batch polymerization processes: State and parameter estimation for semi-batch free-radical emulsion copolymerization processes. Specialization project, Norwegian University of Science and Technology, Trondheim, Norway.

[10] Haugen, F. (2008). Kompendium for kybernetics 2 ved høgskolen i oslo. online. http://techteach.no/fag/seky3322/0708/kalmanfilter/kalmanfilter.pdf.

[11] Hosen, M. A., Hussain, M. A., and Mjalli, F. S. (2011). Control of polystyrene batch reactors using neural network based model predictive control (nnmpc): An experimental investigation. *Control Engineering Practice*, 19:454–467.

[12] Jamalduin, M. Z. and Swartz, C. L. (2016). Closed-loop formulation for nonlinear dynamic real-time optimization. *International Federation of Automatic Control*, 49(7):406–411.

[13] Jamaludin, M. Z. and Swartz, C. L. E. (2017). Dynamic real-time optimization with closed-loop prediction. *AIChE Journal*, 63(9):3896–3911.

[14] Kajotoni, M. M. (2008). *A Comparative Study of Collocation Methods for the Numerical Solution of Differential Equations*. PhD thesis, University of KwaZulu-Natal, Durban.

[15] Keer, L. D., Steenberge, P. H. M. V., Reyniers, M.-F., Marin, G. B., Hungenberg, K.-D., Seda, L., and D'hooge, D. R. (2017). A complete understanding of the reaction kinetics for the industrial production process of expandable polystyrene. *AIChE Journal*, 63(6):2043–2059.

[16] Kiparissides, C. (1996). Polymerization reactor modeling: A review of recent developments and future directions. *Chemical Engineering Science*, 51(10):1637–1659.

[17] Kjetsá, E. (2017). Bilevel model predictive control of a semi-batch emulsion copolymerisation. Master's thesis, Norwegian University of Science and Technology.

[18] Koker, A. K. (2001). *Modeling of Chemical Kinetics and Reactor Design*. Gulf Professional Publishing, Texas, USA.

[19] Kreyszig, E. (2011). *Advanced Engineering Mathematics*. Wiley, tenth edition.

[20] Kumar, A. and Gupta, R. K. (2003). *Fundamentals of Polymer Engineering*. Marcel Dekker, Inc., second edition.

[21] Lund, M. L. (2016). Modeling and optimization of polymerization semi-batch reactors for expandable polystyrene. Specialization project, Norwegian University of Science and Technology, Trondheim, Norway.

[22] Mastan, E. and Zhu, S. (2015). Method of moments: A versatile tool for deterministic modeling of polymerization kinetics. *European Polymer Journal*, 68:139–160.

[23] Nocedal, J. and Wright, S. J. (2000). *Numerical Optimization*. Springer, second edition.

[24] Nyström, A. (2007). Modeling and simulation of a multi phase semi-batch reactor. In *The 48th Scandinavian Conference on Simulation and Modeling*, pages 173–182.

[25] Plastics Europe (2018). Expanded Polystyrene. online. https://www.plasticseurope.org/en/about-plastics/what-are-plastics/large-family.

[26] Plastics Insight (2017). Expanded Polystyrene (EPS): Production, Price, Market and its Properties. online. https://www.plasticsinsight.com/resin-intelligence/resin-prices/expanded-polystyrene/.

[27] Preisig, H. A. (2016). *The ABC of Modelling*. Second edition.

[28] Qin, S. J. and Badgwell, T. A. (2003). A survey of industrial model predictive control technology. *Control Engineering Practice*, 11:733–764.

[29] Schei, T. S. (2008). On-line estimation for process control and optimization applications. *Journal of Process Control*, 18:821–828.

[30] Seair Info Solutions Private Limited (2018). Details of Indian Import Data of Dicuyml Peroxide. online. https://www.seair.co.in/dicuyml-peroxide-import-data.aspx.

[31] Seborg, D. E., Edgar, T. F., Mellichamp, D. A., and III, F. J. D. (2011). *Process Dynamics and Control*. Wiley, third edition.

[32] Sheu, W.-S. (2001). Molecular weight averages and polydispersity of polymers. *Journal of Chemical Education*, 78(4):554–555.

[33] The Editors of Encyclopedia Britannica (2018). Polystyrene: Chemical Compound. online. https://www.britannica.com/science/polystyrene.

[34] U.S. Environmental Protection Agency (2015). AP-42:Compilation of Air Emissions Factors. Environmental report, Environmental Protection Agency, USA.

[35] Wikipedia (2018a). Control Engineering. online. https://en.wikipedia.org/wiki/Control_engineering#History.

[36] Wikipedia (2018b). Control Theory. online. https://en.wikipedia.org/wiki/Control_theory.

[37] Wikipedia (2018c). Sequential Quadratic Programming. online. https://en.wikipedia.org/wiki/Sequential_quadratic_programming.

[38] Wu, G. Z. A., Denton, L. A., and Laurence, R. L. (1982). Batch polymerization of styrene-optimal temperature histories. *Polymer Engineering and Science*, 22(1):1–8.

[39] Yoon, W. J., Kim, Y. S., Kim, I. S., and Choi, K. Y. (2004). Recent advances in polymer reaction engineering: Modeling and control of polymer properties. *Korean Journal of Chemical Engineering*, 21(1):147–167.

[40] Zielinski, J. M. and Duda, J. L. (1992). Predicting polymer/solvent diffusion coefficients using free-volume theory. *AIChE Journal*, 38(3):405–415.

INDEX