Development of Optimization-Based Control Strategies for a Starved-Feed Semibatch Copolymerization Reactor

By

Luis F. T. Perea

A thesis submitted to the Department of Chemical Engineering
in conformity with the requirements for
the degree of Master of Science (Engineering)

Queen's University
Kingston, Ontario, Canada
April 2007

Library and
Archives Canada

Published Heritage
Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et
Archives Canada

Direction du
Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 978-0-494-26521-5
Our file Notre référence
ISBN: 978-0-494-26521-5

Abstract

Styrene and methacrylates are principal components of many automotive coatings resins. Acrylic resins for coatings production are produced using a "starved feed" strategy in which the monomer and initiator are fed at a fixed rate over a period of several hours to produce polymer with relatively constant composition and molecular weight with minimal on-line measurement. The resulting robust reactor operation comes at a price in terms of long batch time, and drift in both polymer molecular weight and composition still occurs in the early and late stages of the batch. Improved operation requires a combination of an optimal recipe with an effective on-line control strategy based on a mathematical model that capture the underlying physics of the process.

In this work, a real-time optimization scheme for nonlinear processes is implemented for this system. The system input and state trajectories are determined on-line to minimize a cost function defined in terms of deviation from target molecular weight and composition, and batch time. The performance of the proposed strategy is examined through numerical simulation for several study cases for the Butyl Methacrylate and Styrene (BMA/STY) free radical polymerization system which has been extensively studied in the past few years.[42,43] Results show that polymer quality can be assured in minimum batch time while achieving constraint satisfaction over the entire run.

Acknowledgments

I would like to thank Professor Robin Hutchinson, for his guidance, supervision and support throughout my research and in particular for his invaluable and wise suggestions at various stages of my project. I also acknowledge that the completion of this project would not have been by any means possible without the enormous contribution of Professor Martin Guay. I would also like to thank any other person or institution which in one way or another contributed to this effort, my parents Manuel and Consuelo, and my long time girlfriend Erika Ruth for their way long distance support and encouragement, Deheng Li for his academic contributions and willingness to help and Queen's University and DuPont de Nemours and Co. for financial support of this work.

Contents

List of Tables

List of Figures

Nomenclature

DE	reactor mixture density [Kg.L^{-1}]
f	initiator efficiency
f_i	mole fraction of monomer-i in the monomer mixture
f_R^j	mole fraction of polymer radicals ending in monomer unit-j
k_D	rate coefficient for initiation decomposition [s^{-1}]
k_i	rate coefficient for primary-radical initiation [L.mol^{-1}.s^{-1}]
k_p^{XY}	rate coefficient for addition of monomer Y to radical X [L.mol^{-1}.s^{-1}]
k_{tc}^{XY}	rate coefficient for termination by combination [L.mol^{-1}.s^{-1}]
k_{term}^{XX}	rate coefficient for homo-termination of species X [L.mol^{-1}.s^{-1}]
$k_{tr,M}^{XX}$	rate coefficient for transfer to monomer [L.mol^{-1}.s^{-1}]
$k_{tr,S}^{X}$	rate coefficient for transfer to solvent [L.mol^{-1}.s^{-1}]
k_t^{COP}	copolymer-averaged termination rate coefficient [L.mol^{-1}.s^{-1}]
m_{SOL}	mass of solvent pre-charged [Kg]
$\overline{M}_X$	molecular weight for species X [Kg.mol^{-1}]
M_n	number-average copolymer molecular weight [Kg.mol^{-1}]
Mr	mass reactor content [Kg]
R_{μ_0}	rate of production of new polymer chains [mol.L^{-1}.s^{-1}]
R_{init}	rate of radical generation from initiator [L.mol^{-1}.s^{-1}]

R_{tr}^{MON}	rate of transfer to monomer [L.mol^{-1}.s^{-1}]
R_{tc}	rate of termination by combination [L.mol^{-1}.s^{-1}]
R_{tr}^{SOL}	rate of transfer to solvent [L.mol^{-1}.s^{-1}]
r_i	monomer reactivity ratio for radical-i
u_1	mass flow rate of monomer BMA [Kg.s^{-1}]
u_2	mass flow rate of monomer STY [Kg.s^{-1}]
u_3	mass flow rate of initiator TBPA [Kg.s^{-1}]
V	mixture volume [L]
x_1	mass of unreacted monomer BMA in the reactor [Kg]
x_2	mass of unreacted monomer STY in the reactor [Kg]
x_3	mass of unreacted initiator in the reactor [Kg]
x_4	mass of polymer-BMA [Kg]
x_5	mass of polymer-STY [Kg]
x_6	mol of copolymer [mol]

Greek

λ_o^X	zeroth live moment of the polymer radicals X [mol.L^{-1}]
λ_o^{TOT}	total concentration of polymer radicals [mol.L^{-1}]
ρ^X	density of species X [Kg.L^{-1}]
α, β, γ	fraction of termination events that occur by disproportionation

Chapter 1 Introduction

1.1 Overview of Polymer Industry

The polymer industry, an important segment of the chemical processing industry with millions of tons per year produced worldwide, is facing many challenges to meet a rapidly changing and diversifying market environment in an ever competing global economy. The pressure for cost reductions and new product developments is creating a need for flexible production through multi-product facilities. In particular, batch and semibatch reactors are widely used in the chemical industry to produce fine chemicals, pigments, polymers and pharmaceuticals. Often the reactions show exothermic behaviour and tight quality specifications have to be met.[31]

As tighter environmental and safety regulations are imposed, competition constantly demands improved operation. However, control of semi-batch reactors is by no means a trivial task due to the unsteady state nature of the process, complex nonlinear behaviour and highly uncertain dynamics.[79,82] The nonlinear dynamics arise from the complexity of the physicochemical interactions and the kinetics of the polymerization reactions, and heat and mass transfer limitations.

Polymerization reactors face additional challenges, as process variables that affect important product quality properties need to be subject to tight control due to the irreversibility of the polymer quality. The actual customer specifications for end-use applications are often represented by non-molecular parameters (e.g., tensile strength, impact strength, color, crack resistance, thermal stability, melt index, density, etc.) that must be somehow related to fundamental polymer quality properties such as average molecular weight, copolymer composition, branching, crosslinking, and stereoregularity.[89] Properties are often difficult or even impossible to measure on-line; off-line measurements are usually available only at low sampling frequencies, introducing delays when tracking their evolution.

1.2 Motivation and Scope of the Work

Currently, acrylic resins for coatings production are produced using a "starved feed" feeding strategy in which the monomer and initiator are fed at a fixed rate over a period of several hours. This strategy is adopted due to the irreversibility of product quality. Operating at these conditions of high instantaneous conversion (low monomer concentrations) ensures the production of polymer with relatively constant quality properties requiring minimal on-line measurement. In particular, the result of this feeding strategy is that copolymer composition and average molecular weight remain uniform during the reaction.

However, the robust reactor operation comes at a price in terms of long batch time, and drift in both polymer molecular weight and composition still occurs in the early stages of the batch. Thus, there are significant incentives for improving reactor operation and the efficiency of quality monitoring which can translate to higher production rates, improved product quality, safer operation, and, as the ultimate consequence, improved profits.

In this work, a reduced dynamic model is developed to capture the essence of the starved-feed free-radical copolymerization process (Chapter 2). The model is used as a basis for on-line optimization of a butyl methacrylate and styrene (BMA/STY) system, reducing batch time while improving product uniformity. Chapter 3 provides an overview of the control strategies used for semi-batch systems and a description of the particular approach used in this work. Chapter 4 deals with the definition of the actual optimization problem; the problem is tailored to the optimization technique to be used through a suitable formulation. Several case studies are presented in Chapter 5 to illustrate the performance of the optimization approach, and the thesis concludes with a description of how the optimization strategy may be further improved and implemented.

Chapter 2 Model Development

2.1 Overview of Process Modeling

Modeling of polymerization processes, especially modeling of polymer architectural properties, is of enormous industrial importance because it plays a key role in achieving the industry's goal of speedy introduction of new products into markets.[89] Further, numerical simulation allows a critical analysis of the behaviour of the system under different operating conditions, improving the understanding of the kinetics and the influence of changing process variables on the process performance and product quality. A good process model can be used to predict the influence of operating conditions on reaction rate and polymer properties, guide the selection and optimization of standard operating conditions for existing and new polymer grades, and to guide process development from lab to pilot plant to full-scale production.[34]

The modeling approach and level of detail should be dictated by the application.[34] For design purposes, detailed mechanistic models to fully describe the process are important. However, for the purposes of control strategy specification, controller design and control system analysis, models that can replicate the dynamic trends of the target processes are usually sufficient.[86] It is acknowledged that the assumptions about the system on which a model is constructed does not entirely capture the full complexity of the system. Although no estimation of a real world system is an exact representation, a trade-off between accuracy and complexity should be made.

For example, to track polymer molecular weight, all mechanisms that include radical transfer must be included. Additional balances are needed to follow other molecular properties, such as the density of short-or long-chain branches, end-group functionality, and the creation and reaction of terminal double bonds.[34] The development of a model that can predict detailed polymer structure but that relies on measurements that either are not available or that introduce delays (in many cases the result of the sample analysis is only of current interest and too late for control decisions to be made[62]) may not be appropriate when operating and control of the process under disturbances is the ultimate objective.

A detailed description of how polymer architecture and polymerization rate depend on reaction conditions and species concentrations from a defined set of kinetic mechanisms has been developed for the BMA/STY system.[42] This model can be used for product development and off-line optimization of the current starved-feed operating strategy. In this work a reduced dynamic mathematical model has been built based on a reduced set of kinetic mechanisms embedded into overall material balances. Our aim is the development of a simplified model that captures the fundamental process behavior to design an efficient and robust system to control polymer properties while fully exploiting on-line optimization-based control techniques. Therefore, predictions on the intended controlled variables should be consistent and represent the dominant dynamics of the system. To this regard, the kinetics represented by the so called terminal model, which assumes that the reactivity of the polymer radical depends solely on the nature of its terminal monomer unit, was employed. Other mechanisms not considered in the modeling include styrene thermal initiation and methacrylate depropagation.

The accuracy of the reduced dynamic model simulated in Matlab is compared with the full detailed representation of the BMA/STY system built in Predici.[42] The Predici code has also been used to run a series of simulations with penultimate propagation kinetics, STY thermal initiation, and BMA depropagation mechanisms turned off. Therefore, this "reduced Predici model" is very similar to the model implemented in Matlab; small differences remain, such as the use of the quasi-steady state assumption on radical concentrations. Experimental data from the work of Deheng Li are also shown in the various plots.[43]

2.2 Development of Reduced Model

In this section the mathematical model of the free-radical co-polymerization system styrene (STY) and butyl methacrylate (BMA), using tert-butyl peroxyacetate (TBPA) as initiator, and xylene solvent is described. The reactions are carried out in a semibatch reactor under isothermal operation at 138 °C. Under standard assumptions such as perfect mixing, constant physical properties, quasi steady-state assumption (radical stationarity) and long chain hypothesis, mass balances on the initiator, solvent and monomers, and moment balances on the radical (live) and dead polymer chains yield the mathematical model summarized below. The model is based upon the set of kinetic mechanisms including initiation, propagation, transfer to monomer and solvent, and termination, as described in Scheme 2.1.

As mentioned previously, thermal initiation, penultimate propagation kinetics and depropagation, as well as chain-branching and scission are not considered for this reduced model. Table 2.1 summarizes the kinetic rate coefficients whereas the thermodynamical data are provided in Table 2.2. The primary references for these coefficients and parameters are contained in the work of Li.[42]

Initiator Decomposition	$I \xrightarrow{k_D} 2fI^*$
Chain Initiation	$I^* + M_j \xrightarrow{k_{i_j}} P_1^j$
Chain Propagation	$P_n^i + M_j \xrightarrow{k_p^{ij}} P_{n+1}^j$
Chain Termination	
Combination	$P_n^i + P_m^j \xrightarrow{k_{tc_{ij}}} D_{n+m}$
Disproportionation	$P_n^i + P_m^j \xrightarrow{k_{td_{ij}}} D_n + D_m$
Chain Transfer	
To Monomer	$P_n^i + M_j \xrightarrow{k_{tr,M}^{ij}} D_n + P_1^j$
To Solvent	$P_n^i + \mathrm{S} \xrightarrow{k_{tr,S}^i} D_n + \mathrm{S}^*$
	$\mathrm{S} + M_j \xrightarrow{k_{i_j}^{sol}} D_n + P_1^j$

Scheme 2.1 Set of kinetic mechanisms assuming terminal radical model.

In Scheme 2.1, the subscript n denotes the number of monomeric units in growing polymer radicals P_n and dead polymer chains D_n. Each reaction has an associated kinetic rate coefficient. The free radical initiator I unimolecularly decomposes (with rate coefficient k_D) to form two primary radicals I^* with efficiency f. Chain initiation occurs when the primary radical adds to monomer M_j to form a polymer radical of type-*j* (P^j). The relative amounts of dead polymer D_n and radical-i P_n^i, made up of a mixture of the monomer types present in the system, are determined by the copolymer composition. Chain growth occurs by addition of monomer M_j to radical-i P_n^i with the propagation rate coefficient k_p^{ij} dependent on both radical and monomer type. Dead chains are formed through bimolecular termination or chain transfer, as discussed later in the development of equations to track polymer molecular weight.

The resulting set of differential equations is:

$$\begin{aligned}
\frac{d(x_1(t))}{dt} &= u_1(t) - x_1(t)(k_p^{AA}\lambda_o^A(t) + k_p^{BA}\lambda_o^B(t)) \\
\frac{d(x_2(t))}{dt} &= u_2(t) - x_2(t)(k_p^{BB}\lambda_o^B(t) + k_p^{AB}\lambda_o^A(t)) \\
\frac{d(x_3(t))}{dt} &= u_3(t) - k_D x_3(t) \\
\frac{d(x_4(t))}{dt} &= x_1(t)(k_p^{AA}\lambda_o^A(t) + k_p^{BA}\lambda_o^B(t)) \\
\frac{d(x_5(t))}{dt} &= x_2(t)(k_p^{BB}\lambda_o^B(t) + k_p^{AB}\lambda_o^A(t)) \\
\frac{d(x_6(t))}{dt} &= V(t)(R_{\mu 0}(t))
\end{aligned} \tag{2.1}$$

$x_1(t)$ and $x_2(t)$ represent the total mass of unreacted BMA and STY in the reactor, while $x_3(t)$ is the mass of unreacted initiator. All the same, $x_4(t)$ and $x_5(t)$ represent the mass of BMA and STY in the polymer in the reactor and the number of moles of polymer is represented by $x_6(t)$.

The model is subject to a number of algebraic equations. The total concentration of radicals in the system is calculated assuming radical stationarity:

$$\lambda_o^{TOT}(t) = cc\left(\frac{DE(t)}{k_t^{COP}(t)}\right)^{1/2}\left(\frac{x_3(t)}{Mr(t)}\right)^{1/2} \tag{2.2}$$

with the constant cc defined as

$$cc = \left[\frac{2k_D f}{\overline{M}_I}\right]^{1/2} \tag{2.3}$$

System density is calculated assuming volume additivity:

$$DE(t) = Mr(t)\left[\frac{x_1(t)}{\rho^A} + \frac{x_2(t)}{\rho^B} + \frac{x_3(t)}{\rho^I} + \frac{m_{SOL}}{\rho^{SOL}} + \frac{(x_4(t)+x_5(t))}{\rho^{POL}}\right]^{-1} \tag{2.4}$$

The total mass in the reactor is calculated as:

$$Mr(t) = x_1(t) + x_2(t) + x_3(t) + x_4(t) + x_5(t) + m_{SOL} \tag{2.5}$$

where m_{SOL} is the initial charge of solvent in the reactor at the start of polymerization. $k_t^{COP}(t)$ is the termination rate coefficient in the copolymerization system, calculated as a function of monomer molar composition and the homo-termination rate coefficients k_{term}^{AA}, k_{term}^{BB} according to:

$$k_t^{COP}(t) = 10^{\left(f_1(t)\log(k_{term}^{AA}) + f_2(t)\log(k_{term}^{BB})\right)} \tag{2.6}$$

where $f_i(t)$ is the mole fraction of monomer-i in the unreacted monomer in the system:

$$f_1(t)=\frac{x_1(t)\overline{M}_B}{x_1(t)\overline{M}_B+x_2(t)\overline{M}_A} \tag{2.7}$$

$$f_2(t)=\frac{x_2(t)\overline{M}_B}{x_1(t)\overline{M}_B+x_2(t)\overline{M}_A} \tag{2.8}$$

The concentration of radicals of BMA (superscript "A") and STY (superscript ("B") is calculated as:

$$\lambda_o^A(t)=\lambda_o^{TOT}(t)f_R^A(t) \tag{2.9}$$

$$\lambda_o^B(t)=\lambda_o^{TOT}(t)f_R^B(t) \tag{2.10}$$

The radical fraction of each type is calculated according to the long-chain hypothesis:

$$f_R^A(t)=\frac{(k_p^{BA}\overline{M}_B)x_1(t)}{(k_p^{BA}\overline{M}_B)x_1(t)+(k_p^{AB}\overline{M}_A)x_2(t)} \tag{2.11}$$

$$f_R^B(t)=\frac{(k_p^{AB}\overline{M}_A)x_2(t)}{(k_p^{BA}\overline{M}_B)x_1(t)+(k_p^{AB}\overline{M}_A)x_2(t)} \tag{2.12}$$

and the volume of the reactor content is given by:

$$V(t)=\left[\frac{x_1(t)}{\rho^A}+\frac{x_2(t)}{\rho^B}+\frac{x_3(t)}{\rho^I}+\frac{m_{SOL}}{\rho^{SOL}}+\frac{(x_4(t)+x_5(t))}{\rho^{POL}}\right] \tag{2.13}$$

The rate of production of new polymer chains is required to calculate polymer molecular weight. It is defined by:

$$R_{\mu 0}(t)=R_{init}(t)+R_{tr}^{MON}(t)+R_{tr}^{SOL}(t)-\frac{R_{tc}(t)}{2} \tag{2.14}$$

where the rate of radical generation from initiator is:

$$R_{init}(t) = \frac{cc^2 DE(t) x_3(t)}{Mr(t)} \tag{2.15}$$

and the rate of transfer to monomer is:

$$R_{tr}^{MON}(t) = \left[m_1(t) \frac{x_1(t)}{Mr(t)} + m_2(t) \frac{x_2(t)}{Mr(t)} \right] \lambda_o^{TOT}(t) DE(t) \tag{2.16}$$

$$\begin{aligned} m_1(t) &= \frac{k_{tr,M}^{AA} f_R^A(t) + k_{tr,M}^{BA} f_R^B(t)}{\overline{M}_A} \\ m_2(t) &= \frac{k_{tr,M}^{BB} f_R^B(t) + k_{tr,M}^{AB} f_R^A(t)}{\overline{M}_B} \end{aligned} \tag{2.17}$$

The rate of transfer to solvent is:

$$R_{tr}^{SOL}(t) = \frac{s(t)}{Mr(t)} \lambda_o^{TOT}(t) DE(t) \tag{2.18}$$

$$s(t) = \frac{k_{tr,S}^{A} f_R^A(t) + k_{tr,S}^{B} f_R^B(t)}{\overline{M}_{SOL}} m_{SOL} \tag{2.19}$$

and finally the rate of termination by combination is:

$$R_{tc}(t) = k_{tc}^{AA}(t)(\lambda_o^A(t))^2 + k_{tc}^{BB}(t)(\lambda_o^B(t))^2 + 2K_{tc}^{AB}(t)(\lambda_o^A(t)\lambda_o^B(t)) \tag{2.20}$$

with the termination by combination defined as a function of the termination rate coefficient $k_t^{COP}(t)$, and the fraction of termination events that occur by disproportionation for AA (α), BB (β), and AB (γ) termination.

$$k_{tc}^{AA}(t) = (1-\alpha)k_t^{COP}(t) \tag{2.21}$$

$$k_{tc}^{BB}(t) = (1-\beta)k_t^{COP}(t) \tag{2.22}$$

$$k_{tc}^{BA}(t) = k_{tc}^{AB}(t) = (1-\gamma)k_t^{COP}(t) \tag{2.23}$$

Various quantities of interest can be calculated from the states. For the semi-batch reactor, monomer conversion is defined as the mass of polymer in the reactor divided by the mass of monomer and polymer:

$$\chi_m(t) = \frac{x_4(t) + x_5(t)}{x_1(t) + x_4(t) + x_2(t) + x_5(t)} \tag{2.24}$$

Number-average molecular weight and copolymer composition denoted by M_n and F are calculated as follows:

$$M_n(t) = \frac{x_4(t) + x_5(t)}{x_6(t)} \tag{2.25}$$

$$F(t) = \frac{\dfrac{x_4(t)}{\overline{\overline{M}}_A}}{\dfrac{x_4(t)}{\overline{\overline{M}}_A} + \dfrac{x_5(t)}{\overline{\overline{M}}_B}} \tag{2.26}$$

An objective of the starved-feed strategy is to keep these latter two quantities uniform throughout the course of the semi-batch reaction, while maintaining high conversion in the reactor.

Table 2.1 Rate coefficients and model parameters (A = BMA, B = STY)

Initiation [s^{-1}]	**T(K)**	**@ 138 °C**
k_D	$6.78\times10^{15}.\exp[-17714.T^{-1}]$	1.3185×10^{-3}
$f=5.155\times10^{-1}$		
Propagation [L.mol^{-1}.s^{-1}]		
k_p^{AA}	$3.80\times10^{6}.\exp[-2754.2.T^{-1}]$	4.69×10^{3}
k_p^{BB}	$4.266\times10^{7}.\exp[-3910.T^{-1}]$	3.162×10^{3}
$r_1=0.42,\ r_2=0.61$		
$k_p^{AB}=\dfrac{k_p^{AA}}{r_1}$		1.12×10^{4}
$k_p^{BA}=\dfrac{k_p^{BB}}{r_2}$		5.18×10^{3}
Termination [L.mol^{-1}.s^{-1}]		
k_{term}^{AA}	$7.10\times10^{7}.\exp[-830.T^{-1}]$	9.43×10^{6}
k_{term}^{BB}	$3.818\times10^{9}.\exp[-958.T^{-1}]$	3.09×10^{8}
$\alpha=0.65\ ,\ \beta=0.01\ ,\ \gamma=0.33$		
Transfer to monomer [L.mol^{-1}.s^{-1}]		
$k_{tr,M}^{AA}$	$1.56\times10^{2}.\exp[-2621.T^{-1}]$	2.658×10^{-1}
$k_{tr,M}^{BB}$	$2.31\times10^{6}.\exp[-6377.T^{-1}]$	4.243×10^{-1}
$k_{tr,M}^{BA}$	$k_{tr,M}^{AA}.k_p^{BB}(k_p^{AA}.r_2)^{-1}$	2.6678×10^{-1}
$k_{tr,M}^{AB}$	$k_{tr,M}^{BB}.k_p^{AA}(k_p^{BB}.r_1)^{-1}$	1.3918
Transfer to solvent [L.mol^{-1}.s^{-1}]		
$C_{S,A}$	$5.55\exp[-4590.T^{-1}]$	7.869×10^{-5}
$C_{S,B}$	$4.23\times10^{-8}\exp[3910.T^{-1}]$	5.707×10^{-4}
$k_{tr,S}^{A}$	$C_{S,A}.k_p^{AA}$	3.6889×10^{-1}
$k_{tr,S}^{B}$	$C_{S,B}.k_p^{BB}$	1.8116

Table 2.2 Thermodynamical data

Species-i	**Density [Kg.L^{-1}]**	**@ 138 °C**	**Molecular weight [Kg.mol^{-1}]**
A (BMA)	$\rho^{A} = 9.01\times10^{-1} - 8.35\times10^{-4}\times T$	7.858×10^{-1}	$\overline{M}_{A} = 142.2\times10^{-3}$
B (STY)	$\rho^{B} = 9.19\times10^{-1} - 6.65\times10^{-4}\times T$	8.273×10^{-1}	$\overline{M}_{B} = 104.15\times10^{-3}$
Sol (Xylene)	$\rho^{SOL} = 8.92\times10^{-1} - 1.3\times10^{-3}\times T$	7.126×10^{-1}	$\overline{M}_{SOL} = 106.17\times10^{-3}$
I (TBPA)	$\rho^{I} = 8.85\times10^{-1}$		$\overline{M}_{I} = 132.16\times10^{-3}$
Pol(Polymer)	$\rho^{POL} = 1.19 - 8.07\times10^{-4}\times T$	1.0786	

2.3 Comparison of Reduced and Full Models

The accuracy of the dynamic reactor model predictions have been tested against the full and the reduced kinetic model developed in Predici. Dynamic model simulations have been carried out for a set of several recipes in which the mass ratio of the two monomers in the feed have been varied, as done in the experimental study of Li.[42,43] The monomer mixture is fed at a constant mass flow into the initial xylene charge, which makes up 30% of total mass in the reactor at the end of the 6 hour feed period. TBPA initiator is also fed continuously, with the mass of initiator fed kept constant at 1.98 wt% relative to the mass of monomer fed.

The Matlab code for the model is attached as Appendix A to this thesis. The system of equations was solved using Matlab solver "ODE15s". Simulink, the Matlab graphical interface, was used to control the execution of the simulations. Predictions of the reduced model implemented in Matlab are compared to the Predici models. The results presented in Figures 2.1 through 2.8 show the calculated M_n, [BMA], [STY], and [TBPA] profiles.

In addition the predicted copolymer composition is compared with the experimental data provided.

The predictions of the reduced Matlab model differ from those of the full Predici model because of the different set of kinetic mechanisms included. The differences are seen mainly for the BMA-rich recipes, and are caused by neglecting BMA depropagation. Lower predicted monomer concentrations also result from the use of the terminal model to describe chain propagation instead of penultimate kinetics. Note that in the work of Li, the fit of the full model to the experimental data was improved by changing k_t^{COP}. The predictions of the Matlab model are in agreement to those of the reduced Predici model and consistent with respect to the experimental data as can be observed from the plots. Predictions on copolymer composition are consistent with the experimental data in all cases showed.

Thus it can be concluded that the reduced model derived in this work is consistent with the previous work and experimental data, capturing the essential dynamics of the system. The model will be used as the basis for the on-line optimization scheme discussed in later chapters.

2.4 Summary

In this section a reactor model of reduced complexity has been developed. The reduced model predictions show to be in agreement with those of the reduced Predici model and consistent with the experimental data. Differences in the model fit in comparison to the full Predici model and to the data are due to the exclusion of BMA depropagation kinetics, the use of a terminal model kinetics and the approximation of the copolymer-averaged termination rate coefficient (k_t^{COP}) as mentioned earlier in this chapter. It is expected that the accuracy of the model predictions will improve as the excluded mechanisms are incorporated in the model. The reduced complexity and the acceptable accuracy of the reactor model developed in this section make it suitable for control studies.

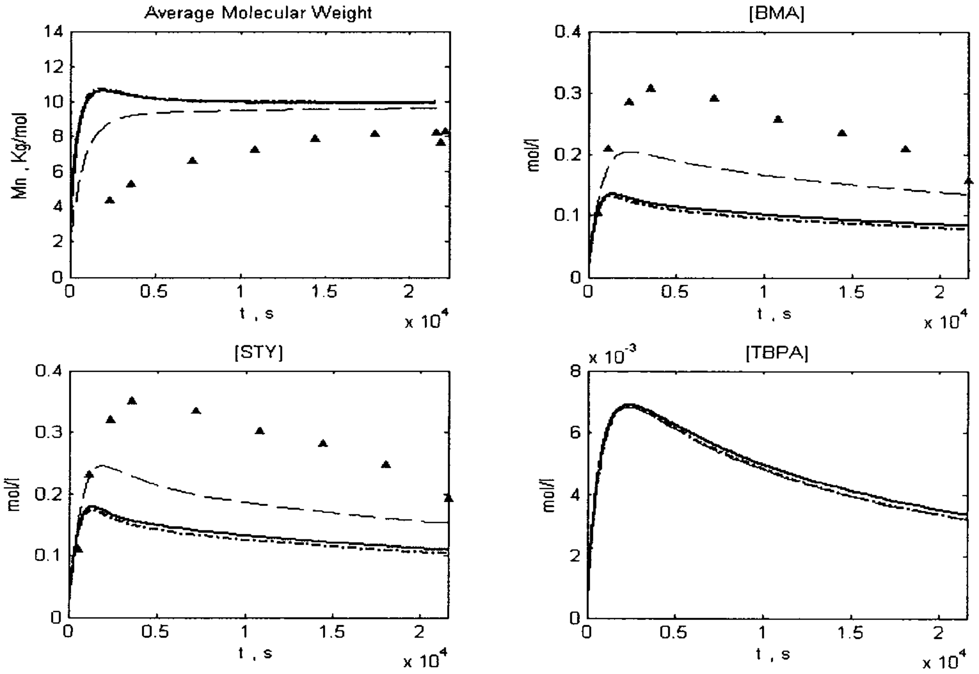


Figure 2.1 Predictions for BMA/STY system at 50/50 mass ratio. Reduced model (——); reduced Predici model (– · ·); full Predici model (– – –); experimental data (▲).

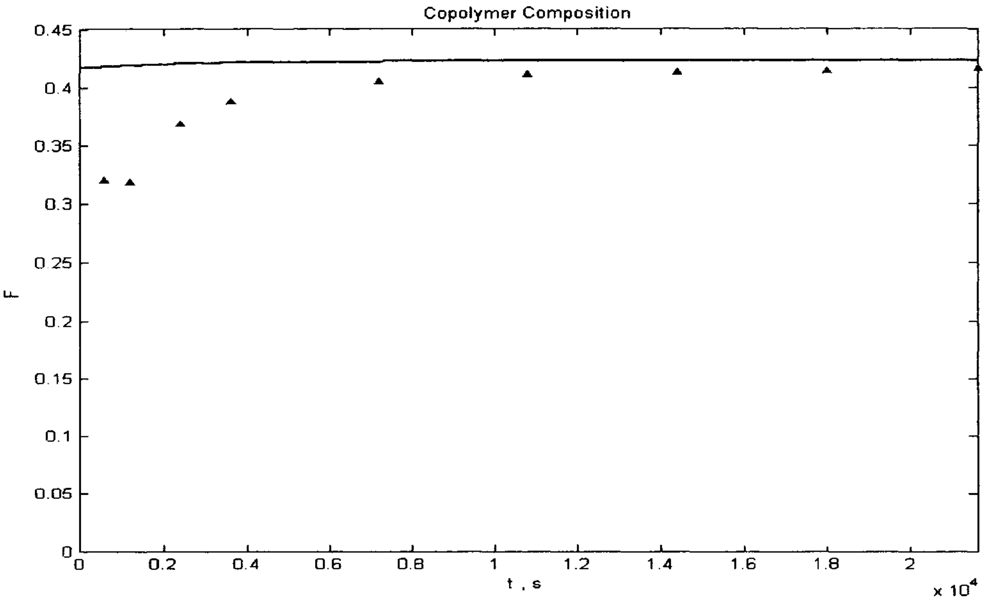


Figure 2.2 Copolymer composition for BMA/STY system at 50/50 mass ratio. Reduced model (——); experimental data (▲).

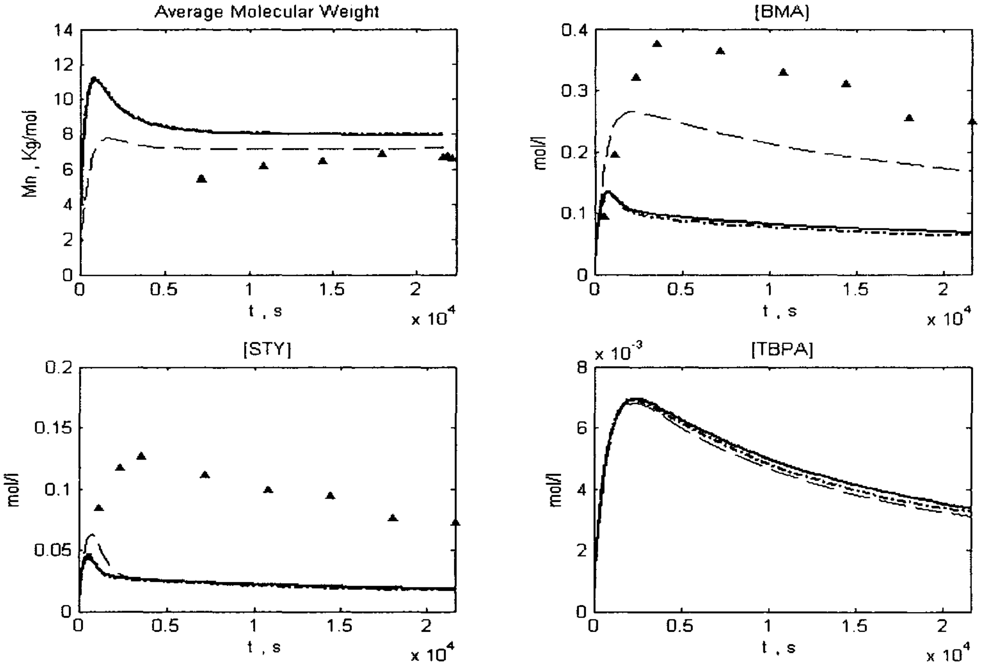


Figure 2.3 Predictions for BMA/STY system at 75/25 mass ratio
Reduced model (——); reduced Predici model (– · ·); full Predici model (– – –); experimental data (▲).

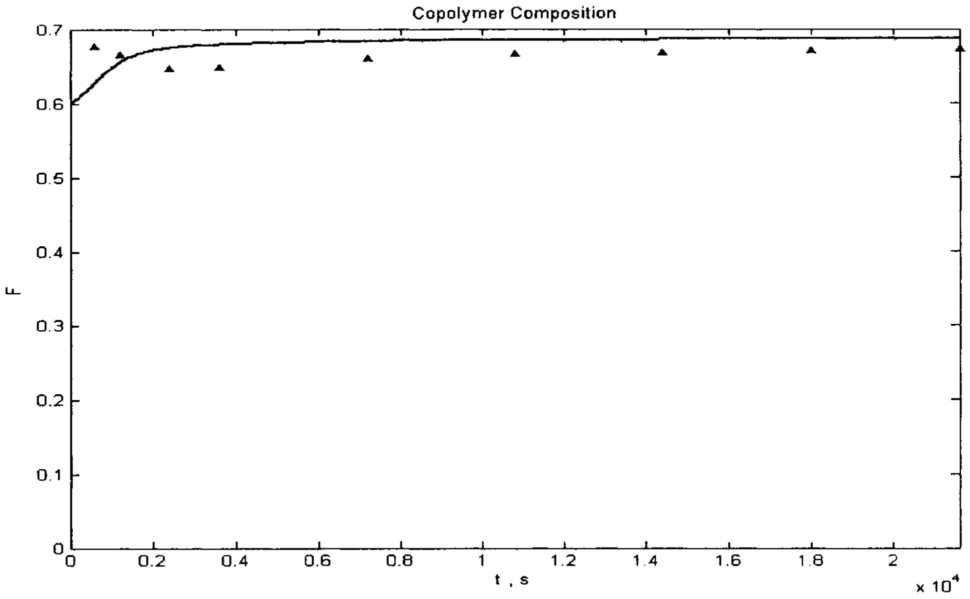


Figure 2.4 Copolymer composition for BMA/STY system at 75/25 mass ratio. Reduced model (——); experimental data (▲).

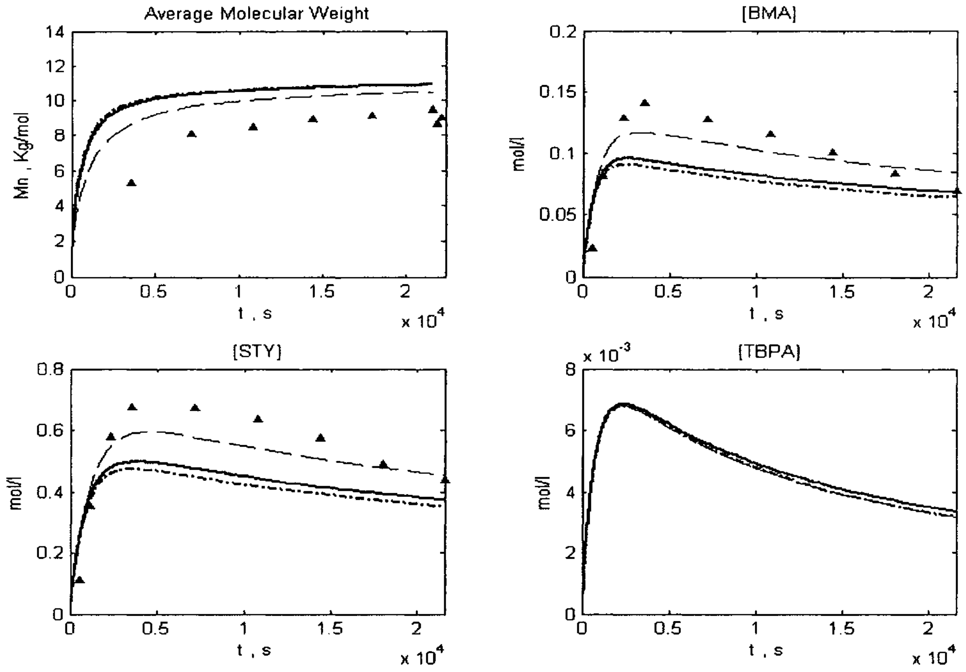


Figure 2.5 Predictions for BMA/STY system at 25/75 mass ratio Reduced model (——); reduced Predici model (– · ·); full Predici model (– – -); experimental data (▲).

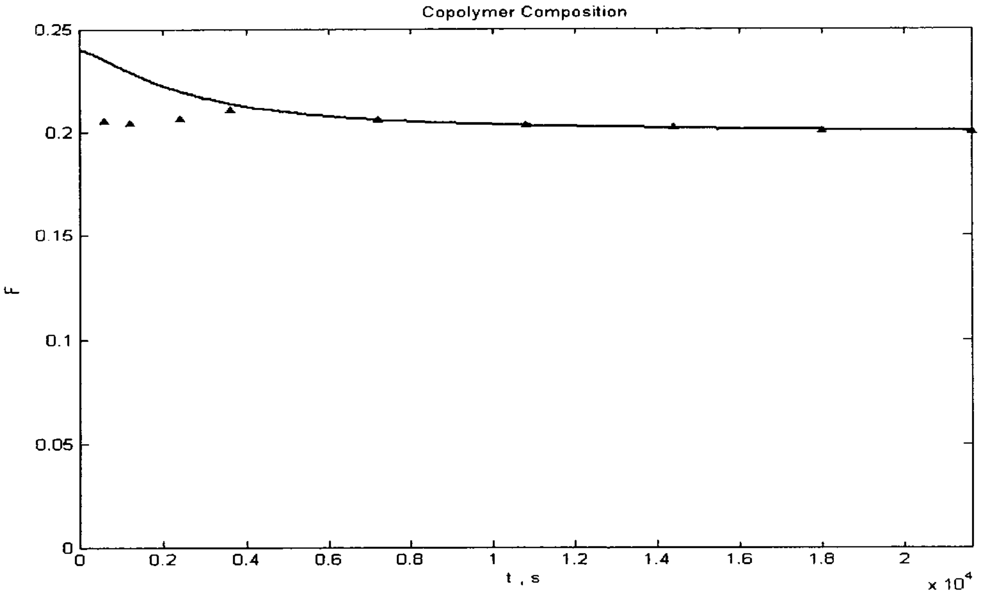


Figure 2.6 Copolymer composition for BMA/STY system at 25/75 mass ratio. Reduced model (——); experimental data (▲).

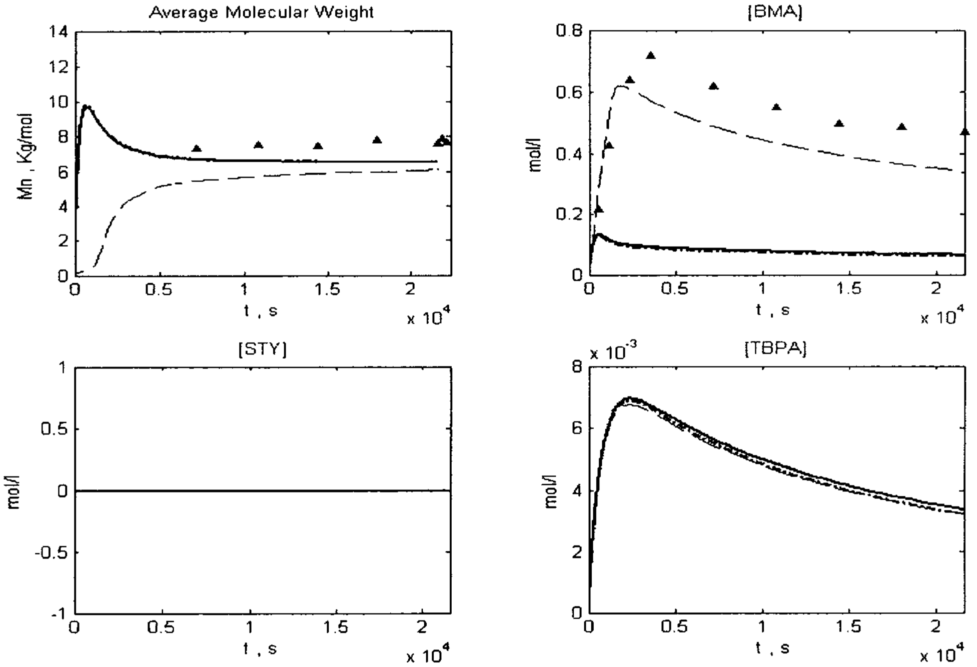


Figure 2.7 Predictions for BMA homopolymerization. Reduced model (——); reduced Predici model (– · ·); full Predici model (– – –); experimental data (▲).

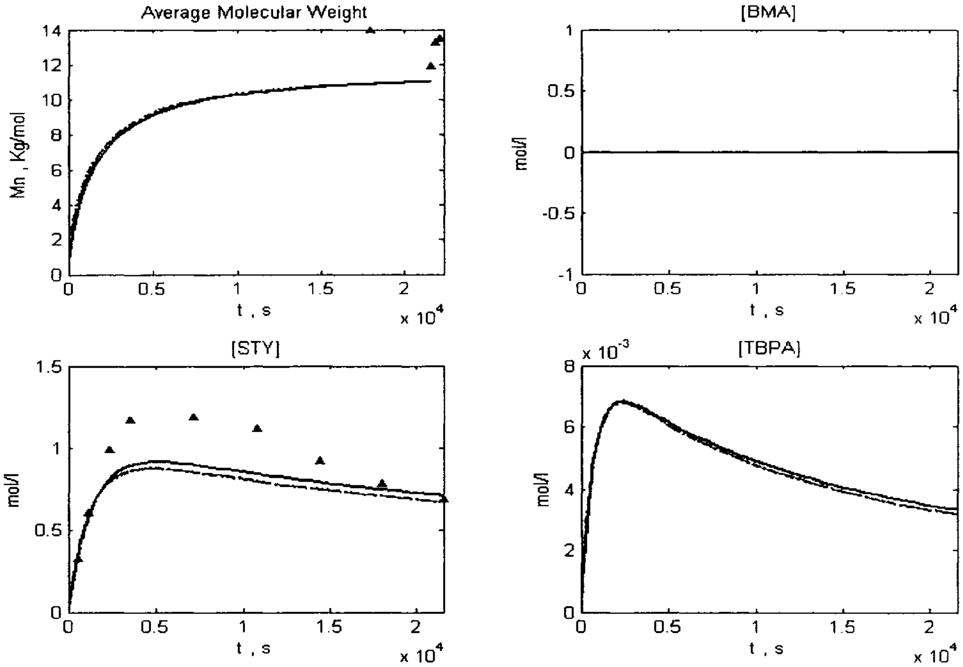


Figure 2.8 Predictions for STY homopolymerization.
Reduced model (——); reduced Predici model (– · ·); full Predici model (– – -); experimental data (▲).

Chapter 3 Control Strategies and System Optimization

3.1 Control Strategies for Semibatch Reactor Operation

Semibatch processes are similar to batch processes, except that reactant can be added and/or products removed during the polymerization.[34] Depending upon the feeding strategy the monomers are fed continuously or by following a pre-defined schedule to obtain required polymer quality specifications.

In a batch reaction, copolymer composition drifts according to the inherent reactivities of the monomers. One monomer is consumed preferentially, causing the monomer composition in the reactor to change as the overall monomer conversion increases. This change gives rise to a variation in copolymer composition with conversion. The BMA/STY system has an azeotropic composition such that BMA is incorporated preferentially into the copolymer at low BMA fraction in the reactor, while at high BMA fraction STY is incorporated preferentially.[42] However in semibatch operation, drift can be substantially reduced by maintaining a constant concentration ratio of the respective monomers in the reactor,[34] the objective of most copolymer production.

One of the most common strategies to achieve constant copolymer composition is to pre-charge the total amount of the least reactive (slower) comonomer into the reactor and feeding the other comonomers while maintaining a specified concentration ratio.[79] A similar approach consists of pre-charging a portion of the total reactant load at predetermined concentrations to obtain a copolymer with the desired composition, then feeding both monomers to the reactor with time varying flow rates to maintain such concentrations, and thus copolymer composition, constant with time.[73] Although these strategies are effective for copolymer composition control, they suffer from several drawbacks. They, are not necessarily time optimal, and the pre-charged amount has to be exactly calculated for a desired amount of polymer. A significant practical issue is that pre-charging one monomer may not be desired for safety reasons. Furthermore, the approach relies on measuring accurately the concentrations in the reactor to regulate the monomer feed rate; the measurement delay (e.g., using a gas chromatograph) makes the control problem more difficult.[25, 73]

Starved feed operation, the operating strategy currently used widely in industry, overcomes many of these problems. A starved feed reactor (SFR) is a semibatch polymerization reactor in which initiator and monomer are fed slowly into a fixed amount of solvent. The polymerization is carried out isothermally at elevated temperatures. The added initiator decomposes almost instantaneously, and the added monomer also polymerizes immediately. Thus, the monomer concentration in the SFR is very low, ideally approaching zero, such that polymer composition matches the composition of the fed monomers. The molecular weight of the product polymer can also be effectively controlled by the feed ratio of monomer to initiator.

Although this strategy is more robust than scheduled feeding described before (especially with the lack of on-line measurements), it is not time optimal and frequently requires a long batch time. As monomer concentration in the reactor is held low, the potential hazard for thermal runaway reaction scenario is minimized. A potential concern with operating in starved mode is that polymer concentrations are high, resulting in higher rates of transfer to polymer and branching reactions.[34] In addition, starved-feed operation does not ensure that the quality specification is met under process disturbances. In general the control system for a polymerization reactor must be sufficiently robust to handle unmeasured disturbances, which impact polymer reactor operation. These disturbances typically result either from trace amount of polymerization inhibitors left over after monomer purification prior to the polymerization reaction or from trace amounts of other compounds which may be present in a typical polymerization recipe and which may be affecting the reaction.[62] A well designed feeding strategy must overcome several pitfalls and meet quality and safety constraints. In addition, the control strategy can be time optimal and must provide a robust mode of operation to deal with disturbances and unknown physics to ensure polymer quality is achieved.

The system under study in this work is operated under starved feed conditions. Initiator and monomers are continuously fed in the desired mass ratio to provide composition and molecular weight control while maintaining low monomer concentration in the reactor. A representation of the process is given in Figure 3.1. Solution free-radical BMA/STY copolymerizations have been performed at 138 °C with monomer added at a constant rate over 6 hours at different mass ratio (feed compositions); initiator was fed in at a constant mass ratio of 1.98 wt% relative to monomer feed. The experimental study showed that the free STY and BMA monomer levels in the semibatch reactor remain low throughout the course of these reactions such that the monomer and polymer compositions remain relatively constant.[42]

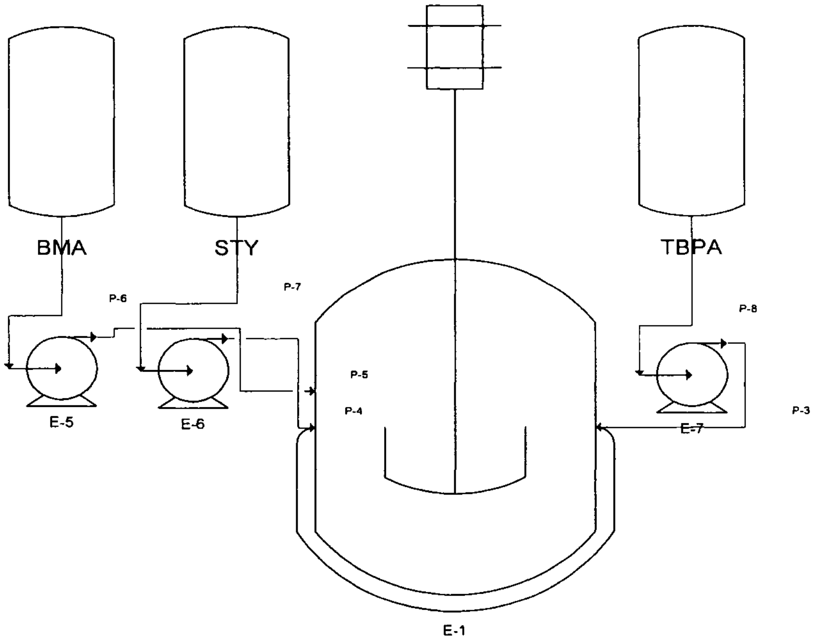


Figure 3.1 Semibatch reactor schematic

3.2 Optimization of Semibatch Reactor Operation

3.2.1 System Optimization Overview

Chemical industries and in particular those dedicated to the production of fine chemicals and polymers have come to recognize that the operation of the processes often provides far from optimum performance with regard to efficiency, yield, safety, or environmental compatibility. For batch and semibatch reactors the typical approach to improve performance has been divided into two parts. First, a recipe for the feed flowrate of one or more reaction components, typically based upon heuristics and experience, has been fixed. In a second step a controller has been designed in order to implement the determined feed flowrate in the presence of disturbances.[2] Potential for optimization comes from the fact that both steps can be executed simultaneously through the use of novel control algorithms. Indeed, many applications in engineering can be formulated as optimization problems whose objective and constraints are often dictated by cost criteria, safety, quality and environmental restrictions.

The objective function for the optimal operation of most processes is formulated so as to minimize the process time, cost of operation, and maximize the product yield and so on. The optimization then is possible through the manipulation of certain process variables such as flow rates, temperature, heat duty, etc. However, these process variables are not always adjustable and are often limited by physical bounds. Thus, the optimal operation of any process involves the determination of an extremum for the objective function by adjusting the process variables while satisfying the constraints on the process.[82]

By solving an optimization problem we obtain an open-loop optimal control (and corresponding state trajectories) that we may implement to control a plant. This standard approach discards model uncertainty (nominal optimization), and leads to an optimal input solution that may not correspond to optimal performance or may not even be feasible in the presence of uncertainty. Moreover,constraint satisfaction, highly important in the presence of safety or environmental constraints, may not be guaranteed. [23, 70]

Thus, the technical challenge is to optimize the system in light of uncertainties. Uncertainties arise from model mismatch, model parameter uncertainties and unmeasured process disturbances. In particular, kinetic parameter uncertainties arise from the fact that coefficients that are usually estimated from laboratory-scale experiments, may not be accurate in commercial-scale reactors due to differences in mixing, or heat and mass transfer characteristics.[70]

Conceptually, there are different approaches to deal with process uncertainties. The difference among them relates to whether or not measurements are used in the calculation of the optimal strategy. A robust optimization approach, for example, is used in the absence of process measurements, whereas measurement-based optimization approach is used in the opposite case. The robust optimization approach takes the uncertainty into account explicitly through the use of probability and the modeling of uncertainties. The input trajectories are then computed off-line once and utilized for all batches.[70] For measurement-based optimization, an online approach is used if measurements are available during the batch.When only offline measurements are available a run-to-run optimization approach is in order. In the latter, information from previous batches is used to update the operating strategy for the current operation. However, this approach does not account for the effect of process disturbances within the batch.

Mathematical optimization methods available in the literature address two major issues: solution of the system equations and formulation of optimization strategy. Most methods differ from one another in the approach employed to address these two issues.[82]

Direct and indirect methods are identified as the two most commonly used approaches to solve the system of differential equations in the optimization problem. The indirect method, based on optimal control theory or calculus of variations (Pontryagin minimum principle), results in TPBVPs (two boundary value problems) which are computationally intensive and whose solution is often a non-trivial task These techniques are not considered further in this work. In the direct method, discussed in more detail below, two different approaches have been reported: the sequential approach in which the process dynamic equations are integrated explicitly (control vector parameterization), and the simultaneous approach in which the equations are integrated implicitly.

Whether we choose to parameterize the control and/or the process states by a set of finite parameters (direct methods), the resulting system needs to be coupled with numerical optimization algorithms (typically nonlinear programming). In general, any numerical procedure for dynamic optimization involves the following steps: (i) choice of initial inputs; (ii) calculation of the system states, the performance index, and the constraints; and (iii) adaptation of the inputs towards the optimum (using for example gradient information). Steps (ii) and (iii) are then repeated until convergence.[70]

3.2.1.1 Literature review

In control vector parameterization (CVP), only the inputs are parameterized using a finite set of decision variables through, typically, a piecewise constant approximation over equally spaced time intervals.[4,70,80] The sensitivity equations are then integrated simultaneously with the process equations.[82] This sequential method has the advantage that the number of degrees of freedom in the nonlinear program remains relatively small.[4]

Several efficient optimization algorithms for sequential methods are available in the literature.[8,29,39,67,72] Most algorithms are similar in the strategies used for function evaluation, gradient evaluation, and formulation of the quadratic program.[82] The main drawbacks of this method are that the quality of the solution depends upon the parameterization of the control profile and that a fast convergence to the solution is obtained only if a feasible starting guess is provided. Nonetheless, finding this feasible initial solution is a non-trivial problem. Several CVP applications for nominal optimization have been reported in literature for batch distillation systems [27,51,58] and for semibatch and batch processes.[4,36]

In the simultaneous approach the explicit solution of the differential equations is avoided. Sensitivity equations for the dependent variables with respect to the parameters along with the system differential equations are replaced by an approximating set of algebraic equations and the optimization is performed in the full space of discretized inputs and states.[9,20,33,38,61,74,85]

Several batch process applications using this approach have been reported in literature.[21,24,44,65] For large scale systems, however, the total discretization of the system greatly increases the cost of computing and storage requirements, making its implementation in online optimization schemes difficult.

In sequential methods a system of differential equations is solved by using an optimization routine with an embedded differential equations solver. However, the original differential equation system is converted into a non-linear optimization problem in simultaneous methods, which is usually solved by nonlinear programming (NLP) routines such as Sequential Quadratic Programming.[82]

Thus, a large number of research studies are found concerning off-line optimization for process case studies and/or concerning closed loop controller design for certain specified trajectory or end-point problems, including state observer/estimator design.

Industry and academia have shown an increasing interest in using optimization based methods such as model predictive control (MPC) for trajectory tracking,[37,50,53,57] and on-line re-optimization strategies.[25,64,75] This tendency seems to go hand in hand with recent developments in sensor technology that are becoming prevalent in many industrial settings.

In model predictive control the optimization problem is solved over a prediction horizon at each sampling instant yielding a sequence of control moves from which only the first move is implemented. These control variables are manipulated to force the process variables to follow a pre-specified trajectory (reference tracking) from the current operating point to the target. The inputs trajectories are defined, typically in piecewise constant functions. Moreover, the optimization is solved using various techniques, including nonlinear programming or gradient methods.[56] A detailed description of model predictive control is available in several references. [28, 52, 63]

In particular, MPC has drawn the attention of some academic researchers focusing on the design of MPC controllers for regulation around a set point (in many cases for temperature control) and minimization of off-grade products in polymerization reactors.[11,18,48,54] MPC controller , state estimator design and implementation has also been carried out for fed-batch reactors.[1,17,22,32,69,77]

In order to deal with uncertainties online re-optimization schemes have been reported in literature. An online cascade optimization scheme was applied to a penicillin fed-batch fermentation process where the input was parameterized using a CVP approach in a piecewise fashion.[64] Cascade optimization combines the positive features of optimal operation and feedback control. This scheme is composed of a "high level" optimizer that repeatedly solves the optimization problem and a "low level" feedback controller that tracks the optimal trajectories calculated.

Also, a re-optimization based control for a semibatch emulsion copolymerization process to minimize the batch time while maintaining copolymer composition has been reported in literature.[25] For a desired copolymer composition, a specified reaction rate ratio for the comonomers has to be maintained while the reaction rates are maximized to achieve the shortest possible time. The re-optimization is triggered only if the ratio deviates from a tolerated band. The necessity to compute a new optimal trajectory is checked at every sampling interval up to the end of the batch. The measured or observed process states are used as initial conditions for the optimization of the remainder of the batch, and the calculated manipulated variables are implemented without the need of designing a tracking controller to follow the trajectory.

3.2.1.2 Shortcomings

Many industrial batch processes are operated through open-loop application of an off-line optimized input profile, such as feed or temperature; this approach does not account for process uncertainty and in particular, in polymerization processes, the irreversibility in polymer quality that requires fast response to existing disturbances will not be overcome. However, when on-line measurement information is available, on-line re-optimization promises considerable improvement.[75] One important precondition, however, is the availability of reliable and efficient (limited computational complexity) numerical optimal control algorithms.[5]

Particularly, the high computational requirements,[5] the dependence of the quality of the solution on the accuracy of the process model,[62] and the stability issue related to the prediction horizon approximation[16,49] are acknowledged to be the main problems arising from nonlinear MPC.

In addition to the prohibitve computing time for real time implementation, complexities arising from controller design in cascade optimization might become difficult to overcome.

To overcome those pitfalls, an on-line optimization technique [56] that addresses these shortcomings has been employed in the present work and it is described in more detail in the next section. This technique, unlike a cascade optimization scheme, incorporates the feedback measurements directly into the optimization procedure rather than into a low level tracking controller so it truly becomes an online optimization rather than an online re-optimization, thus avoiding the need for controller design and the related complexities.

The driving idea behind the method utilized in this work is that as measurements come in from the batch, new profiles are determined on-line and implemented simultaneously without the need of tracking the calculated profiles.[56] The major differences with other online strategies are in the numerical algorithm used to solve the optimization in comparison to other nonlinear optimization algorithms widely used in other optimization approaches. This improvement addresses the intensive computing time drawback without introducing any compromises for on-line implementation.

3.2.2 Online Optimization Technique Employed

3.2.2.1 Problem Statement

The basic control problem addressed here is to find the input trajectory that solves the dynamic optimization problem for some cost functional J (Eqs. 3.1 and 3.2). Mathematically, the general form of the constrained dynamic optimization problem can be expressed as:

$$\min_{u(t)} \; J(x,u) = \int_0^{t_f} q(x(t),u(t))dt \tag{3.1}$$

subject to:

$$\begin{aligned} &\dot{x}(t) = f(x(t),u(t)) \\ &w(x(t),u(t)) \geq 0 \;,\; \forall t \in [0,t_f] \\ &x(0) = x_0 \\ &x(t_f) = x_f \end{aligned} \tag{3.2}$$

In this formulation, $x \in \Re^n$ are the state variables, $u(t) = [u_1(t),...,u_m(t)]^T \in \Re^m$ is the vector of m input variables, $f : \Re^n \times \Re^m \rightarrow \Re^n$ is a smooth vector-valued function. The function $w : \Re^n \times \Re^m \rightarrow \Re^{nc}$ is also smooth vector-valued function. This represents the path constraints that limit the inputs and the states during the batch. The end-point variables that limit the outcome of the batch at final time is represented by $x(t_f)$.

The real-valued smooth cost function $q:\Re^n \times \Re^m \rightarrow \Re$ is assumed to be positive definite, i.e., $q(0,u)=0, \forall u \in \Re^m$ and $q(x,u)>0, \forall x \neq 0$. Similarly, $q(x,0)=0$, $\forall x \in \Re^n$ and $q(x,u)>0, \forall u \neq 0$.

In general, the particular choice of the problem formulation including the objective function and the set of constraints, the input parameterization and the complexity of the process dynamics have an effect on the problem's solution and the effort required to obtain it.

3.2.2.2 Parameterization

The input trajectories $u(t)=[u_1(t),...,u_m(t)]^T$ are approximated by a given parameterization defined as:

$$u_j(t)=\sum_{i=1}^{N}\theta_{ij}\phi_i(t)=\varphi(\theta,t) \tag{3.3}$$

where $\phi_i(t)$ are the basis functions and θ_{ij} are the parameters to be determined for $i=1,...,N$ and $j=1,...,m$, where N is the number of parameters for input j. θ_j represents the vector of parameters for input j.

3.2.2.3 Real Time Optimization Technique

The dynamic optimization problem is now expressed in terms of the parameters

$$\underset{\theta}{Min}\ J = \int_0^{t_f} q(x^p(t),\ \varphi(\theta,t))dt \tag{3.4}$$

subject to

$$\begin{aligned} &\dot{x}^p = f(x^p(t), \varphi(\theta,t)) \\ &w(x^p(t), \varphi(\theta,t)) \geq 0\ , \forall\, t \in [0, t_f] \\ &x^p(t) = x^m(t) \\ &x^p(t_f) = x_f \end{aligned} \tag{3.5}$$

The superscript *m* denotes a measured quantity, and the superscript *p* denotes a predicted quantity. Here, the predicted states at each time instant are initialized to the current state measurements starting from the current time

The constrained parameter set $\Omega = \{\theta \in \Re^N \mid w(x^p(t), \varphi(\theta,t)) \geq 0\}$ describe a convex subset of $\Re^N$ .It is assumed that the parameters evolve on a compact subset γ of $\Re^N$. The cost functional $J : \gamma \to \Re$ is assumed to be convex and continuously differentiable on γ. The assumptions made up to this point guarantees a local optimization of the constrained problem exists and that the gradient can be used to achieve that minimization.[56]

An interior point method with penalty function is used to include the constraint costs. The interior point method incorporating a log barrier function enforces the state and input constraints (essentially converting the constrained optimization problem into an unconstrained one) while the end-point constraints are incorporated through a terminal penalty function.[56] In the remaining equations obvious notation has been omitted.

Thus, let the path cost with barrier function be stated as follows:

$$L(x,u) = q(x,u) - \sum_{i=1}^{nc} \mu_1 \log(w_i(x,u) + \varepsilon) \tag{3.6}$$

To emphasize that the optimization is based on the current conditions, the new cost functional with interior point method and penalty function is stated as:

$$J_{ip} = \int_t^{t_f} L(x^p(\tau), \varphi(\theta,\tau))d\tau + M \| x^p(t_f) - x_f \|^2 \tag{3.7}$$

where $\tau \in [t, t_f]$ is the integration variable and nc is the number of inequality constraints. The parameter $\mu_1 > 0$ is the barrier parameter for the logarithm term, $M > 0$ is the penalty term and $\varepsilon > 0$ is a constraint relaxation factor (back-off) that prevents the barrier term from singularity.

The technique uses a gradient-based method for the solution of the dynamic optimization problems in real time. The method employs a straightforward diagonally scaled steepest descent parameter update law of the form:

$$\dot{\theta} = -\mathrm{Proj}(\Gamma \nabla_\theta J_{ip}, \Omega_w) \tag{3.8}$$

where $\Gamma > 0$ is the scaling matrix (referred as a gain matrix) to be defined. A discussion of the choice of Γ is given in Chapter 5. To avoid divergence of the update law a projection algorithm is used. This guarantees that the parameters remain in a convex set

$\Omega_w = \{\theta \in \Re^N \mid \| \theta \| \leq \eta\}$ for some $\eta > 0$ while assuring that the cost decreases until the optimal profile parameters are determined. A Lyapunov-based method was used to show convergence to the local minimizers of a user-defined cost functional.[30,56]

The projection algorithm is given by:[56]

$$\dot{\theta} = \begin{cases} \gamma, & \mathit{if} \; \| \theta \| < \eta \;\text{ or }\; (\| \theta \| = \eta \;\text{ and }\; \nabla P(\theta) \leq 0\,) \\ \\ \psi, & \mathit{otherwise} \end{cases} \tag{3.9}$$

where $\psi = \gamma - \gamma \dfrac{\nabla P(\theta) \nabla P(\theta)^T}{\| \nabla P(\theta) \|^2}$, γ is a compact subset of $\Re^n$ where the parameters are assumed to evolve, $P(\theta) = \theta^T \theta - \eta$, θ is the vector of input parameters and η is chosen such that $\| \theta \| \leq \eta$.

Full state measurements are assumed through simulation carried out in Matlab (simulation of the closed-loop system) whereas Odessa[41], a Fortran-based differential solver, was used to calculate the model predictions and the first order sensitivities. On each sampling time the process model is integrated while simultaneously obtaining sensitivity information for the objective function and the states with respect to the free optimization variables. This information is used to generate new values for the decision variables (inputs and time interval lengths) through the update law. The calculated inputs are then implemented until the next measurement is available.

The optimization procedure is nested within the process control scheme, with the optimization triggered when the next measurement is available (or prediction). The time for the solution of each optimization problem must be short enough to guarantee a sufficiently fast reaction to disturbances and should not be larger than the sampling time for the real time scheme to be implementable. The series of integration and optimization proceeds until reaching the last time interval.

Chapter 4 Problem Statement and On-line Optimization

4.1 Reactor Optimization Problem

The problem of real-time optimization of a system with nonlinear dynamics subject to control, states and endpoint constraints is considered in this chapter. A reduced model and an adequate objective function and set of constraints were formulated to fully exploit the on-line optimization technique used to develop an improved feed schedule.

The operation of the BMA/STY semi-batch reactor is subject to optimization. The objective is defined so as to minimize the batch time and maintain the copolymer molecular weight (M_n) and composition (F) at their targets values while satisfying path and terminal constrains on the states and/or the controls. Those operational, quality and safety related constraints have to be met during the batch and at its final time.

In order to find optimal trajectories for the operational degrees of freedom, a dynamic optimization problem is formulated describing the feed flow rates parameters and the time interval lengths as decision or optimizing variables. For tracking purposes the objectives for M_n(Eq 2.25) and F(Eq 2.26) have been aggregated into a single scalar function while defining the batch time as a terminal objective function

$$J = \int_0^{t_f} (\omega_1 (\frac{M_n(t)}{M_{sp}} - 1)^2 + \omega_2 (\frac{F(t)}{F_{sp}} - 1)^2) dt + \omega_3 t_f \qquad (4.1)$$

J is the objective function to be minimized, and M_{sp} and F_{sp} are the target values for molecular weight and composition respectively. The set of weights that properly scales the objectives are defined by $\omega_1, \omega_2, \omega_3$. The optimization is subject to process dynamics $\dot{x} = f(x,u)$ (Eq. 2.1), with the constraints described below.

State-dependent constraints typically result from safety and operability considerations, such as limits on temperature and concentrations. In particular, the monomer and initiator levels in the batch must not exceed certain maximum limits in order to meet safety constraints related to heat release. These constraints are described by:

$$\begin{gathered} 0 \le x_k(t) \le x_k^{\max}\ ,\ k \in \{1,2,3\} \\ x_{k+3}(t) \ge 0 \end{gathered} \tag{4.2}$$

Placing the constraints on unreacted monomer in the reactor is a convenient way to limit potential heat release, as this quantity is directly related to the maximum temperature rise in the reactor that could occur in the case of a complete loss of heat removal capacity in the system.

On the other hand, control constraints are often dictated by actuator limitations. For instance, non-negativity of flow rates is a common input constraint.

$$0 \le u_k(t) \le u_k^{\max}\ ,\ k \in \{1,2,3\} \tag{4.3}$$

Terminal constraints are commonly related to safety or productivity considerations. With regard to the latter, the desired mass of polymer at the batch end is specified as an endpoint constraint:

$$x_4(t_f) + x_5(t_f) = m_{pol}(t_f) \tag{4.4}$$

4.2 Parameterization

The control functions are parameterized using a single constant: $u_k(t) = u_k$ ($k = 1,2,3$) through the entire batch time horizon. Although the described optimization technique contemplates the use of a parameterization function of time to approximate the infinite-dimensional inputs, some compromises has been made in this work.

While it may be true that a constant parameterization might result in suboptimal profiles, it results in a manageable tuning process, for algorithm parameters and gain factors, and maintainability than otherwise (polynomial, exponential, sigmoidal parameterization, etc). As a consequence, in particular, reasonable response of the process outputs (molecular weight and composition) to gain changes has been observed in agreement with the physics of the process. This turns the tuning procedure into a relatively less complicated task.

The control parameterization is set to take advantage of the generally good control of polymer quality (molecular weight and composition) using starved-feed operation. Thus, the controls are expressed as follows:

$$\begin{aligned} u_1(t) &= \theta_1 \\ u_2(t) &= \theta_2\theta_1 \\ u_3(t) &= \theta_3\theta_1 \end{aligned} \tag{4.5}$$

The mass flowrate of STY (u_2) is related to the mass flowrate of BMA (θ_1) with proportionality factor θ_2. Similarly, θ_3 is the ratio of initiator to BMA mass flow.

On the other hand, since the batch end time of the problem is unspecified and also subject to minimization, the problem is converted to one with a specified end time through the use of the following transformation (time normalization):

$$\tau = t/t_f,\ \tau \in [0,1] \tag{4.6}$$

The independent variable is now τ, which varies from 0 to 1. Then, the final batch time is considered as an additional optimization parameter. However, an equivalent approach has been taken. Here, the batch time horizon t_f is decomposed into n equally spaced time intervals $[t_{ii}, t_{ii+1}]$ with $t_{ii} < t_{ii+1}$, $ii = 1,\ldots,n$, such that

$$t_f = \Delta t\, n \tag{4.7}$$

The length of the interval is considered as an optimization variable ($\theta_4 = \Delta t$) instead of the variable t_f for scaling purposes.

4.3 Modified Cost Functional

To emphasize that the optimization is based on the current conditions, the objective function and the process dynamics can be stated now as follows:

$$J(\tau)=\int_{\tau}^{1} t_f(\omega_1(\frac{M_n(\sigma)}{M_{sp}}-1)^2+\omega_2(\frac{F(\sigma)}{F_{sp}}-1)^2)d\sigma+\omega_3 t_f \qquad (4.11)$$

$$\dot{x}=\frac{dx}{d\tau}=t_f f_{(x,u)} \qquad (4.12)$$

where

$$t_f=\Delta t\, n \qquad (4.13)$$

The modified cost function then can be stated as follows

$$J_{ip}=\int_{\tau}^{1}\left[t_f(\omega_1(\frac{M_n(\sigma)}{M_{sp}}-1)^2+\omega_2(\frac{F(\sigma)}{F_{sp}}-1)^2)-\mu_1\sum_{j=1}^{z=m}\log(w_j+\varepsilon)\right]d\sigma+\omega_3 t_f+M(w_f)^2 \qquad (4.14)$$

where $\sigma\in[\tau,1]$, the m path constraints w_j are given by:

$$w_k=-x_k-x_k^{\max}+\varepsilon \qquad (4.15)$$

$$w_{k+1}=x_k+\varepsilon,\ k=1,2,3 \qquad (4.16)$$

For control constraints:

$$w_{k+6}=-u_k+u_k^{\max}+\varepsilon \qquad (4.17)$$

$$w_{k+9}=u_k+\varepsilon,\ k=1,2,3 \qquad (4.18)$$

and end point constraints:

$$w_f=x_4+x_5-m_{pol}(t_f) \qquad (4.19)$$

Numerical values for the bounds of the path constraints are given in Chapter 5 when several cases studies have been discussed. The integral part of the modified cost is considered as an additional state such that:

$$\dot{x}_7 = \frac{dx_7(\tau)}{d\tau} = t_f(\omega_1(\frac{M_n(\tau)}{M_{sp}} - 1)^2 + \omega_2(\frac{F(\tau)}{F_{sp}} - 1)^2) - \mu_1 \sum_{j=1}^{z=m} \log(w_j + \varepsilon) \tag{4.20}$$

Finding a locally optimal solution can often be achieved through appropriate initial guesses to the optimizer and the use of realistic constraints. However, to find a feasible set of initial guesses is sometimes not a trivial task due to the interior feasible point method (barrier function) used. The barrier method requires that the initial guess of the solution to be strictly feasible. In addition, ill conditioning inherent to the nonlinearity of the reactor system may play a role in the sensitivity of the solution to initial conditions.

Chapter 5 BMA/STY System Optimization

5.1 Optimization Results and Discussion

In this chapter, the performance of the proposed on-line optimization based control strategy is illustrated for several BMA/STY copolymerization case studies, including varying the initial feed rates and changing the copolymer molecular weight target value. Simultaneously, a comparison with the results obtained through simulation of the conventional starved feed operation at different feed composition (mass) ratios is carried out.

Cases 1-3 illustrate the optimization of the BMA/STY system relative to the conventional starved feed strategy for the 50:50 BMA: STY mass ratio. Case 1 follows experimental initial conditions (monomer and initiator feed rates) used in the experimental study of Li[43] and summarized in Table 5.1. Further improvement on these results is obtained through a change in the initial input parameters, as shown in Case 2. Case 3 presents the optimization under a different target value for molecular weight.

Finally, Case 4 deals with the optimization of the BMA/STY system relative to the conventional starved feed strategy at 75:25 BMA: STY mass ratio. Section 5.2 discusses the robustness of the proposed operation strategy under model parameter uncertainty.

The settings and results for the presented case studies are summarized in Tables 5.2 through Table 5.8, whereas the graphical outputs can be observed in Figures 5.1 through 5.10.

Table 5.1 Conventional starved feed strategy used in experiments of Li [43]

Results	**50:50 BMA:STY (wt ratio)**	**75:25 BMA STY (wt ratio)**
m_{TBPA}^{fed} , Kg	9.7E-03	9.7E-03
m_{BMA}^{fed} , Kg	2.46E-01	3.6E-01
m_{STY}^{fed} , Kg,	2.46E-01	1.23E-01
t_f , min	360	360
$m_{pol}(t_f)$, $(*)$	0.474	0.483
$(*)$ terminal constraint : mass of polymer produced (Kg)		

In Table 5.1, m_{TBPA}^{fed} , m_{BMA}^{fed}, m_{STY}^{fed} represent the mass of initiator , BMA and STY fed to reactor up to the final batch time, respectively. The final batch time is given by t_f. The mass of polymer produced at the final batch time is $m_{pol}(t_f)$. This will be taken as the terminal (end-point) constraint for the optimization cases to be treated later in this chapter.

Table 5.2 Settings for Cases 1-3

<table>
<tr><th>Initial Conditions</th><th>Case 1</th><th>Case 2</th><th>Case 3</th></tr>
<tr><td>$x_1(0)$, Kg</td><td colspan="3">0.215</td></tr>
<tr><td>$x_1(0) = x_2(0) = x_3(0) = x_4(0) = x_5(0)$,Kg</td><td colspan="3">0</td></tr>
<tr><td>$x_6(0)$, mol</td><td colspan="3">0</td></tr>
<tr><td><b>Initial control parameters</b></td><td colspan="3">$t_{ref} = 21600\,\text{s}$, $n = 1000$</td></tr>
<tr><td>$\theta_1(0)$, Kg.s^{-1}</td><td>2.46E-01/t_{ref}</td><td>1.7E-01/t_{ref}</td><td>2.4E-01/t_{ref}</td></tr>
<tr><td>$\theta_2(0)$</td><td>1</td><td>1</td><td>1</td></tr>
<tr><td>$\theta_3(0)$</td><td>0.039</td><td>0.074</td><td>0.086</td></tr>
<tr><td>$\theta_4(0)$, s</td><td colspan="3">t_{ref} / n</td></tr>
<tr><td colspan="4"><b>Target values</b></td></tr>
<tr><td>M_n , Kg.mol^{-1}</td><td colspan="2">9.96</td><td>8</td></tr>
<tr><td>F</td><td colspan="3">0.423</td></tr>
<tr><td colspan="4"><b>Constraint bounds</b></td></tr>
<tr><td>$m_{pol}^{\max}(t_f)$, Kg</td><td colspan="3">0.474</td></tr>
<tr><td>$x_k^{\min} \le x_k(t) \le x_k^{\max}$, $k \in \{1,2,3,4,5,6\}$,Kg</td><td colspan="2">$x_k^{\min}$</td><td>$x_k^{\max}$</td></tr>
<tr><td>$x_1(t)$</td><td colspan="2">0</td><td>0.01</td></tr>
<tr><td>$x_2(t)$</td><td colspan="2">0</td><td>0.01</td></tr>
<tr><td>$x_3(t)$</td><td colspan="2">0</td><td>5E-04</td></tr>
<tr><td>$x_4(t)$</td><td colspan="2">0</td><td>---</td></tr>
<tr><td>$x_5(t)$</td><td colspan="2">0</td><td>---</td></tr>
<tr><td>$x_6(t)$</td><td colspan="2">0</td><td>---</td></tr>
</table>

Table 5.3 Algorithm parameters for Cases 1-3

Algorithm Parameters	**Case 1**	**Case 2**	**Case 3**
M	2E+04	4E+04	6E+03
μ	1E-11	1E-11	1E-11
ε	1E-12	1E-12	1E-12
k_1	1E-12	1E-12	1E-12
k_2	1E-05	1E-05	1E-05
k_3	1E-05	1E-05	1E-05
k_4	1E-04	35	1.9

Table 5.4 Optimization results for Cases 1-3

Results	**Case 1**	**Case 2**	**Case 3**
m_{TBPA}^{fed} , Kg	9.8E-03	9.8E-03	12.3E-03
m_{BMA}^{fed} , Kg	2.46E-01	2.44E-01	2.45E-01
m_{STY}^{fed} , Kg	2.47E-01	2.44E-01	2.25E-01
$t_f^{ref} - t_f^{batch}$, min	59.0	77.2	69.4
$m_{pol}(t_f) - m_{pol}(t_f)$, Kg	2.2E-04	4.7E-04	4.6E-04
Computation time , s	11.02	12.83	10.52
Final Cost	155.27	227.73	172.04

Initial conditions and constraints for Cases 1-3 are summarized in Table 5.2, and the algorithm parameters in Table 5.3. The optimization results are summarized in Table 5.4. Table 5.3 shows the algorithm parameters chosen for a given case study. In particular, the adaptive gain (k_1, k_2, k_3, k_4) are scaling factors that play a role on controlling the rate of change of the parameters $(\theta_1, \theta_2, \theta_3, \theta_4)$ according to the steepest descent law which takes the form $\dot{\theta}_j = -\mathrm{Proj}(k_j \nabla_{\theta_j} J_{ip}, \Omega_w)$, $j = 1,\ldots,4$ used by the algorithm (Eq.3.8). In practice, it was found that increases in k_1 cause an increase in molecular weight while a decrease causes the opposite effect. All the same, an increase in k_2 results in a difficult copolymer composition control leading to negative deviations from its target. The opposite is also true, a decrease in k_2 increases the composition values. Similarly, k_3 has influence over the polymer quality specifications specifically on molecular weight showing a direct relationship with molecular weight variations, that is, a molecular weight increase is due to a rise in k_3 and vice versa.

Lastly, k_4 exerts a similar influence on the change of θ_4 (time interval length) and as a consequence on the batch time. An increase in k_4 emphasizes the time reduction task.

With regard to the other algorithm parameters, the barrier parameter (μ) is chosen small while the penalty factor (M) is chosen to be large. A larger M value may be needed to force terminal constraint satisfaction while a smaller value of μ may lead to numerical problems and erroneous constraint violation. Finally, small values of ε are to be used to avoid constraint singularity (in the log barrier term).

Finally, it is worth to point out that in all case studies the set of weights that properly scale the objectives are set to $\omega_1 = 1$, $\omega_2 = 1$ and $\omega_3 = 1/21600$. Increasing ω_3 leads to a greater emphasis on batch time reduction. However, those increments in time reduction were found to be negligible relative to that achieved with the base set of weights.

For Case 1, the initial flowrates of monomers and initiator are set to the experimental values used by Li. Fig 5.1 shows that although copolymer composition is kept at its target value, there is no improvement in the M_n profile after optimization; a transient overshoot still occurs in the first third of the batch. However, the algorithm is able to increase monomer and initiator flow rates (Fig 5.2) such that the batch is completed about 1 hr more quickly, compared to the 6 hr feed time of the original recipe. Although a decrease in the objective function values (see cost functional in Fig 5.1) is obtained, there is still an unexploited potential for further reduction on the objectives, mainly molecular weight deviation and batch time since good control on copolymer composition is achieved.

Free mass of TBPA is shown to be within its boundary while free mass of BMA and STY hit their boundary values at the batch end (Fig 5.1). All the same, it has been observed that the rest of process states show non negativity as defined in Table 5.2. The latter turns out to be the case for all runs carried out in this chapter.

In Fig.5.2, it is seen that the profiles initially show a slow rate of change and although the flow rates show a quick increase afterwards, the rate of adaptation of the flow rates is clearly not fast enough to achieve the objectives and constraint satisfaction at the batch start.

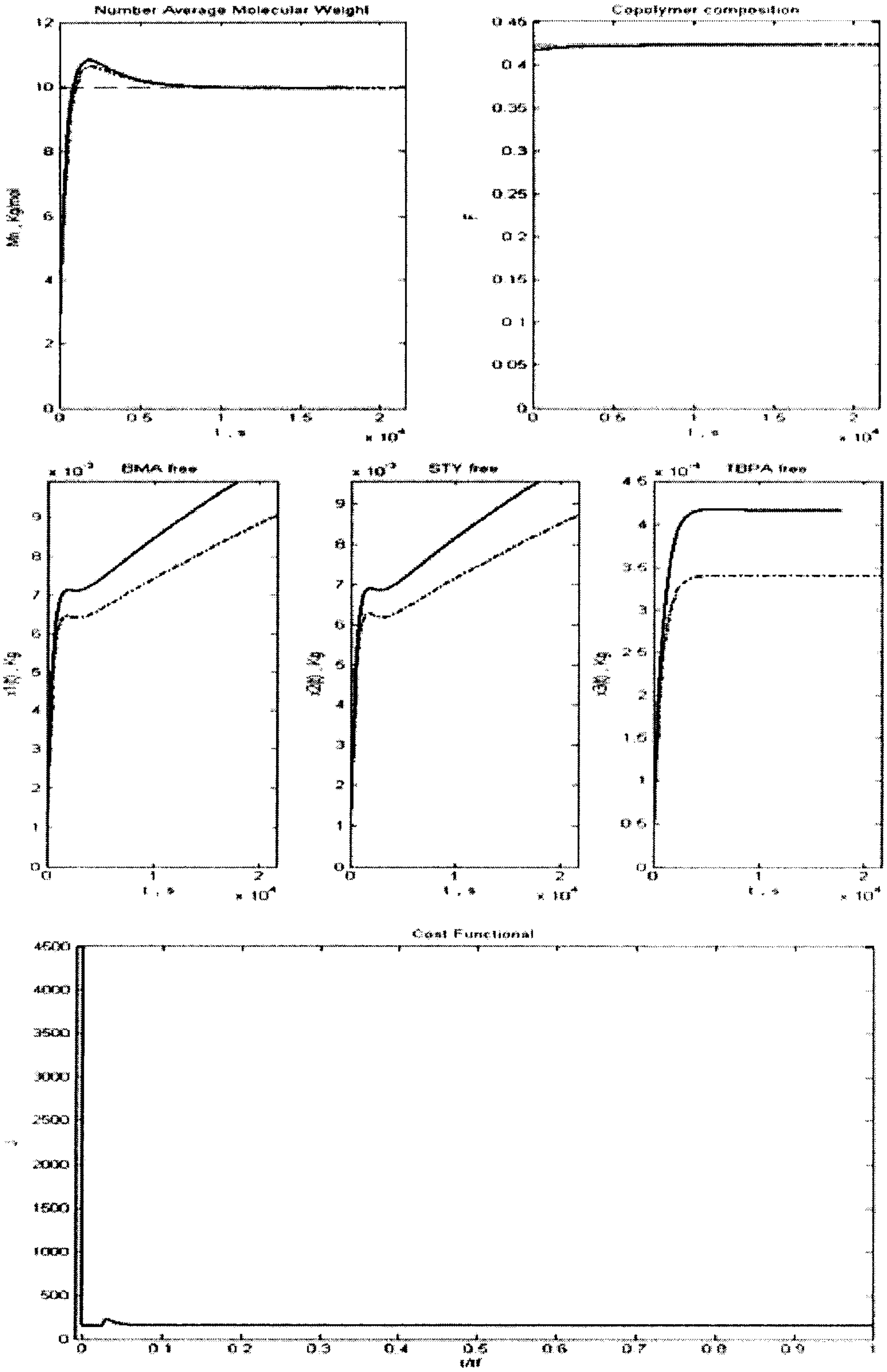


Fig.5.1 Optimization outputs for Case 1.
Optimization (——); conventional starved feed operation (– · ·); target (- - - -).

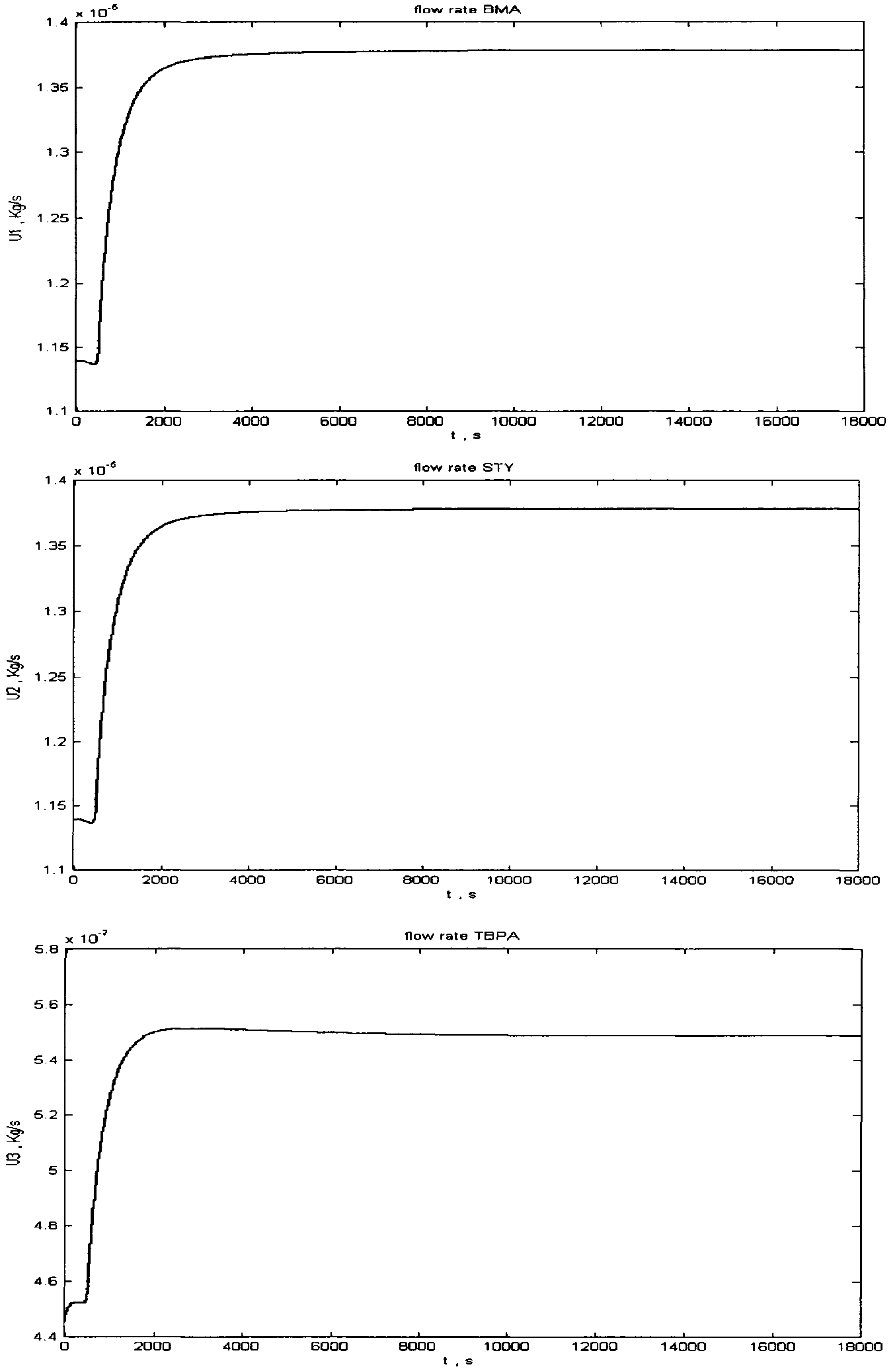


Fig.5.2 Feed profiles for Case 1

However, as a result of the posterior increased flow rates the objectives are finally fulfilled at roughly 10000 seconds, the point at which the policies adopt a flat shape to keep control towards the end of the process.

In addition, from Table 5.4 (Case 1), the load of monomers and initiator fed with the new feeding strategy is in accordance with the charge fed under the conventional starved feed operation (see Table 5.1). Although quality requirements are not fully met, the solution shows a reduction of batch time of around 16 % while approaching the terminal constraint on mass of polymer within 0.5 grams of tolerance.

The Case 1 results suggest that a further improvement on the objectives is in order. This is possible by adjusting the set of initial control parameters as reported in Table 5.2 (Case 2), with graphical output given by Fig.5.3 and Fig.5.4. The new set of initial conditions on the parameters was found by trial and error, based upon the structure of molecular weight and copolymer composition definitions. Decreasing initial monomer flow rates and increasing the initial initiator flow rate was found to eliminate molecular weight overshoot.

Fig 5.3 shows that the initial overshoot on molecular weight, as shown by simulation of the conventional operation, has been overcome while reducing the batch time by about 21% relative to the actual operation, as reported in Table 5.4 (Case 2). The improvement on the objectives is reflected by a large decrease on the cost functional as shown in Fig 5.3. Moreover, the final value of the cost functional as reported in Table 5.4 represents a reduction in 97.5 % relative to its initial value as inferred from Fig 5.3 showing the effectiveness of the optimization.

Fig 5.4 shows that both monomer feed flow rates are increased simultaneously with the initiator policy. The optimal initiator flow rate shows a large increase at the beginning of the batch and then the profile decreases as the limit on the free mass of initiator in the reactor is approached (Fig 5.3), followed by a slow increase with time as seen in Fig 5.4. Similarly, the monomer feed profiles climb quickly until they reach a period of relatively constant flow. Both monomer profiles are similar, suggesting that the mass ratio of monomers (θ_2) is being held close to unity, as shown by Fig 5.5. Similarly, the same can be deduced for Case 4 (75:25 BMA/ST mass ratio) from Figs 5.9 and 5.10. Moreover, Figs 5.5 and 5.10 show that the estimated parameters change more slowly as time increases (iterates). Also, it can be seen in Fig 5.5 that $\dot{\theta}$ (difference of successive parameters values) converges to zero as the parameters reach a stationary value as time approaches the batch end while the cost decreases at the same time as observed from Fig 5.3. This is expected as the cost will strictly decrease until the optimal profile parameters are determined (end of the batch) as discussed in the appropriate reference.[56] There, it was shown that the objective function or cost functional (J) will strictly decrease until the optimal profile parameters are determined, point at which the gradient with respect to the parameters is zero (parameter convergence).

Constraint satisfaction is achieved along the Case 2 batch while approaching the terminal constraint on mass of polymer within 0.5 grams of tolerance. The latter required a penalty factor of double magnitude relative to the first run (see Case 1 in Table 5.3). Keeping M at its former value will result in the terminal constraint on mass of polymer $m_{pol}(t_f)$ far less from expected. To force $m_{pol}(t_f)$ (as defined in Eq.4.6) within certain tolerance, M should be increased accordingly.

Moreover, the total consumption of raw material is in accordance to the charge fed under conventional starved feed as shown in Table 5.4 (Case 2). Additionally, it can be seen from Fig. 5.3 that the free masses of BMA and STY in the reactor reach values very near to the actual maximum bound of 0.01 kg at the batch end. Nonetheless, it has been observed that a relaxation on such constrained variables lead to a negligible improvement on the objective function.

A change in the molecular weight target value (decreased from 10,000 to 8000 g/mol) yields similar results, as seen in Table 5.4, Fig 5.6 and Fig 5.7 for Case 3. The initial conditions for Case 3 were obtained by adjusting the control parameters corresponding to flow rates of BMA and initiator (TBPA) from Case 2 to achieve a lower molecular weight. Since copolymer composition was not subject to changes, θ_2 was unaltered relative to Case 2. A batch time reduction of about 19% is achieved while reaching saturation on remaining mass of initiator in the reactor. Although a relaxation of the constraint may lead to further time reduction, it has been observed that a broader range leads to negligible improvement on batch time. A 26% increase (2.6 g) in the amount of initiator is required compared to the cases with higher molecular weight target values, as reported in Table 5.4 (Case 3). This increase is expected since low molecular weight resins are produced by increasing the amount of initiator fed to the reactor.

The optimization results summarized in Table 5.4 (Case 1-3) and Table 5.8 (Case 4) show a computing time on the order of ten seconds for each case. This makes the optimization procedure more appealing for on-line implementation. The time reported corresponds to the product of the averaged computing time for each control (flow rate) at each time step and the number of time intervals n. For most online optimization techniques, the heavy demand for computing time restricts the frequency of updates, and consequently these methods usually result in very discrete on-line changes.[56] However, the methodology used in this work enabled fast control calculation allowing for high number of updates as the number of intervals in which the process time was portioned is large (n). This results in relatively smooth control policies as seen throughout this section.

Fig 5.8 and Fig 5.9 show the outputs for the optimization of the BMA/STY system relative to a conventional starved feed operation at 75:25 BMA:STY feed mass ratio (Case 4). Results show that polymer quality can be assured in minimum time while having constraint satisfaction over the run. In particular, the time reduction achieved is about 32% relative to the conventional strategy, and the overall monomer load fed is in accordance to the open-loop starved feed operation. However, an initiator excess of about 3 % (equivalent to 0.3 grams) relative to the conventional satrategy is required, as reported in Table 5.8 as low molecular weight resin is being produced (similar to Case 3).

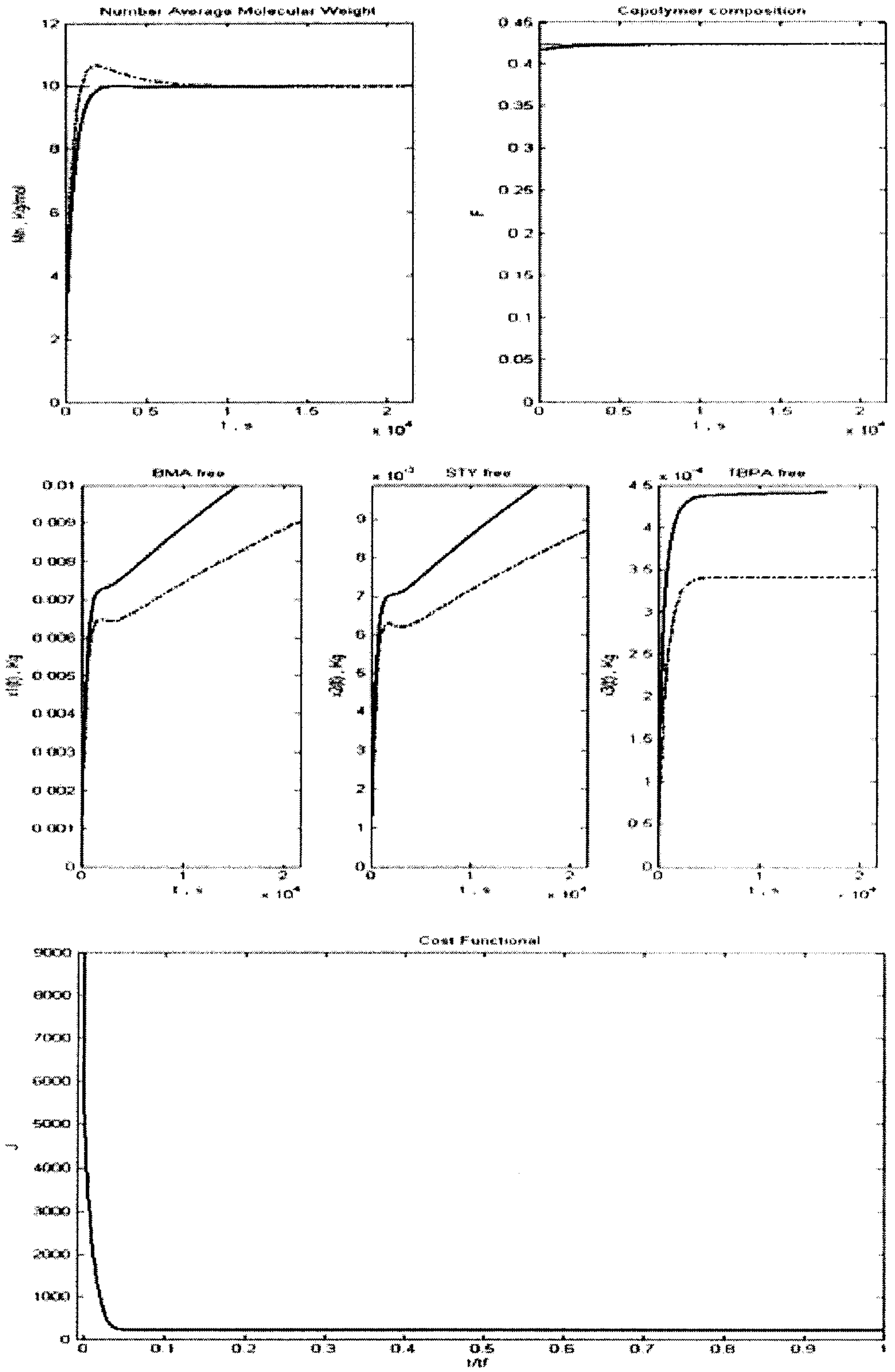


Fig.5.3 Optimization outputs for Case 2.
Optimization (——); conventional starved feed operation (– · ·); target (- - - -).

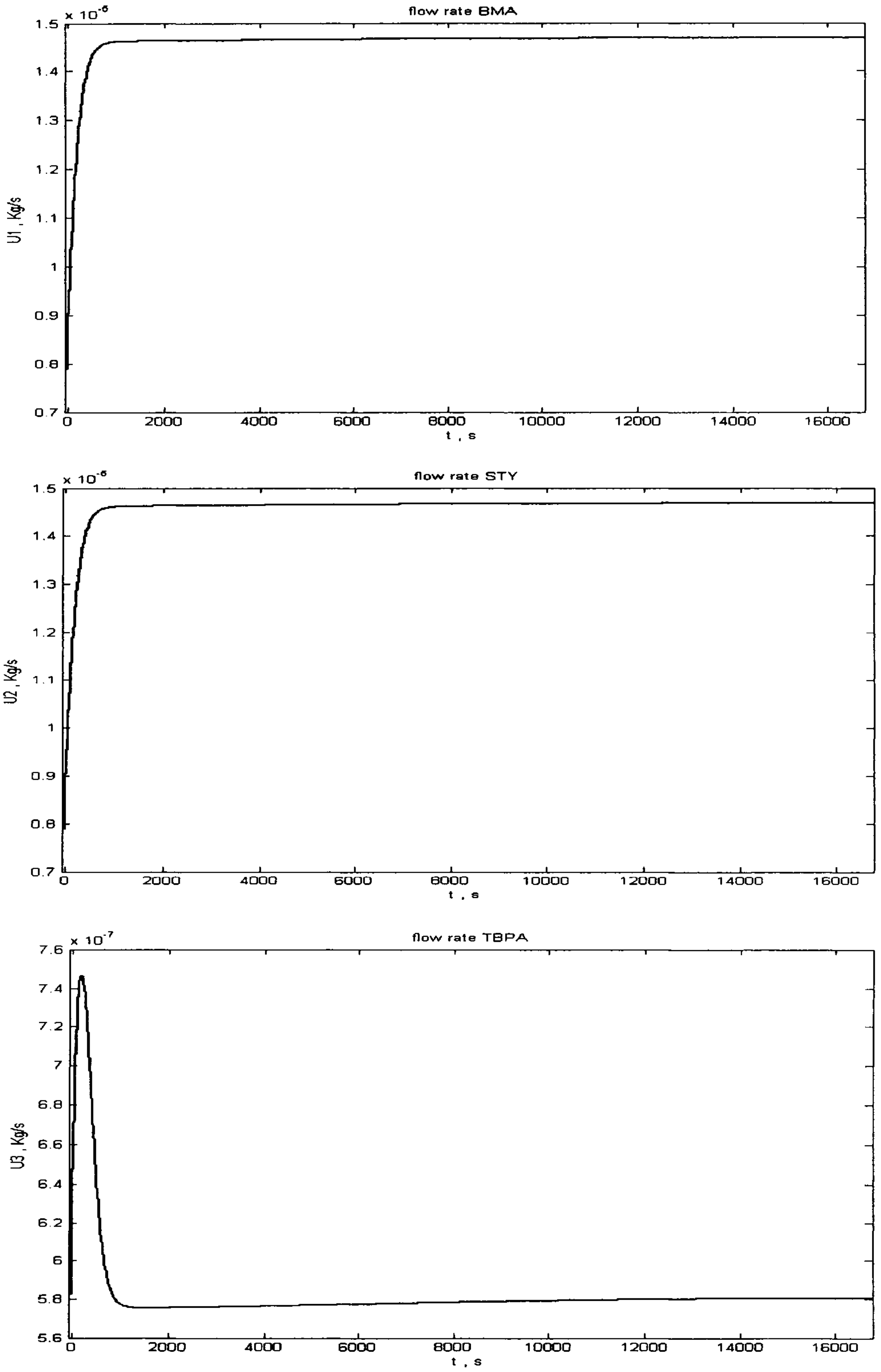


Fig.5.4 Feed profiles for Case 2

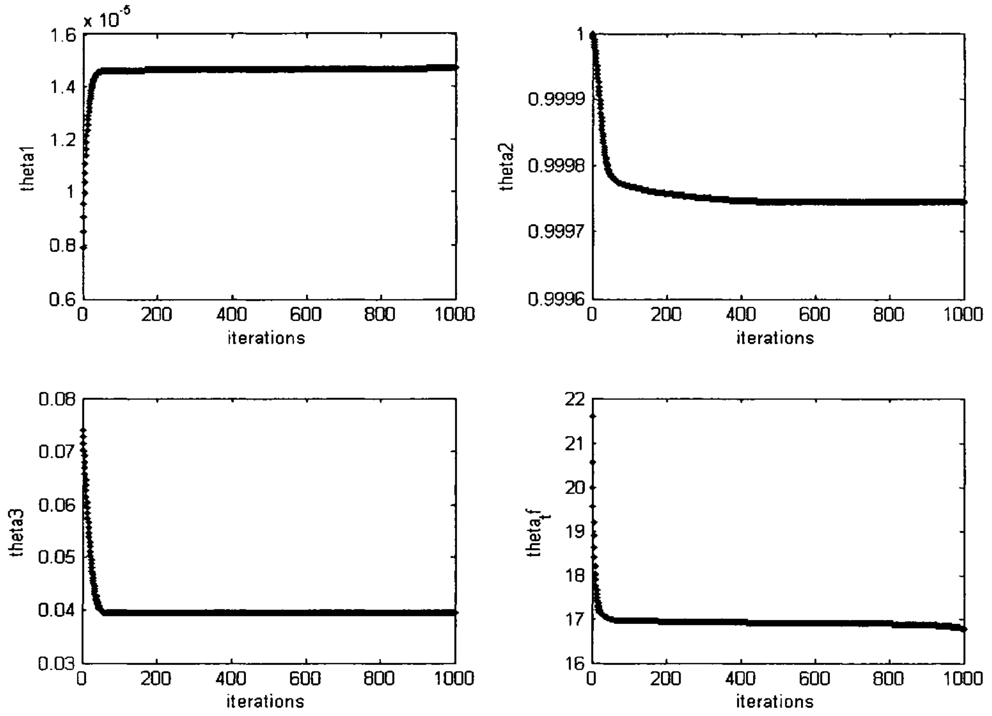


Fig.5.5 Parameter convergence for Case 2

Note that no solution to the control problem was obtained when running the optimization procedure using the experimental initial conditions of Li (see Table 5.1): results were similar to the conventional operation and no time minimization was achieved. Following the same practical rules employed to adjust the initial conditions on the control parameters (from Table 5.5), the initial guess for this case was obtained (see Table 5.6). The results of the subsequent optimization carried out under this new initial guess set are reported in Table 5.8, whereas the algorithm parameters are found in Table 5.7.

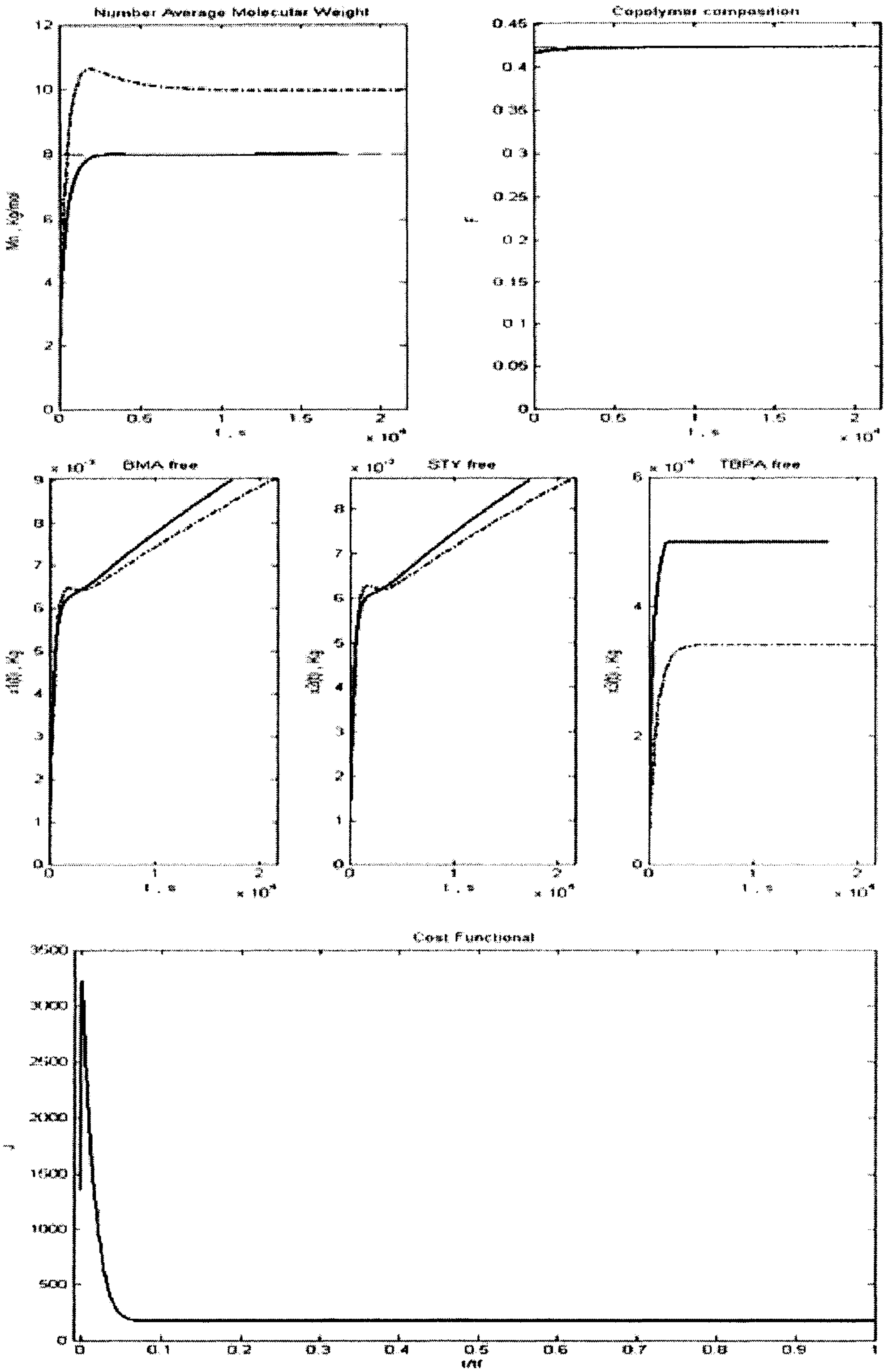


Fig.5.6 Optimization outputs for Case 3.
Optimization (——); conventional starved feed operation (– · ·); target (- - - -).

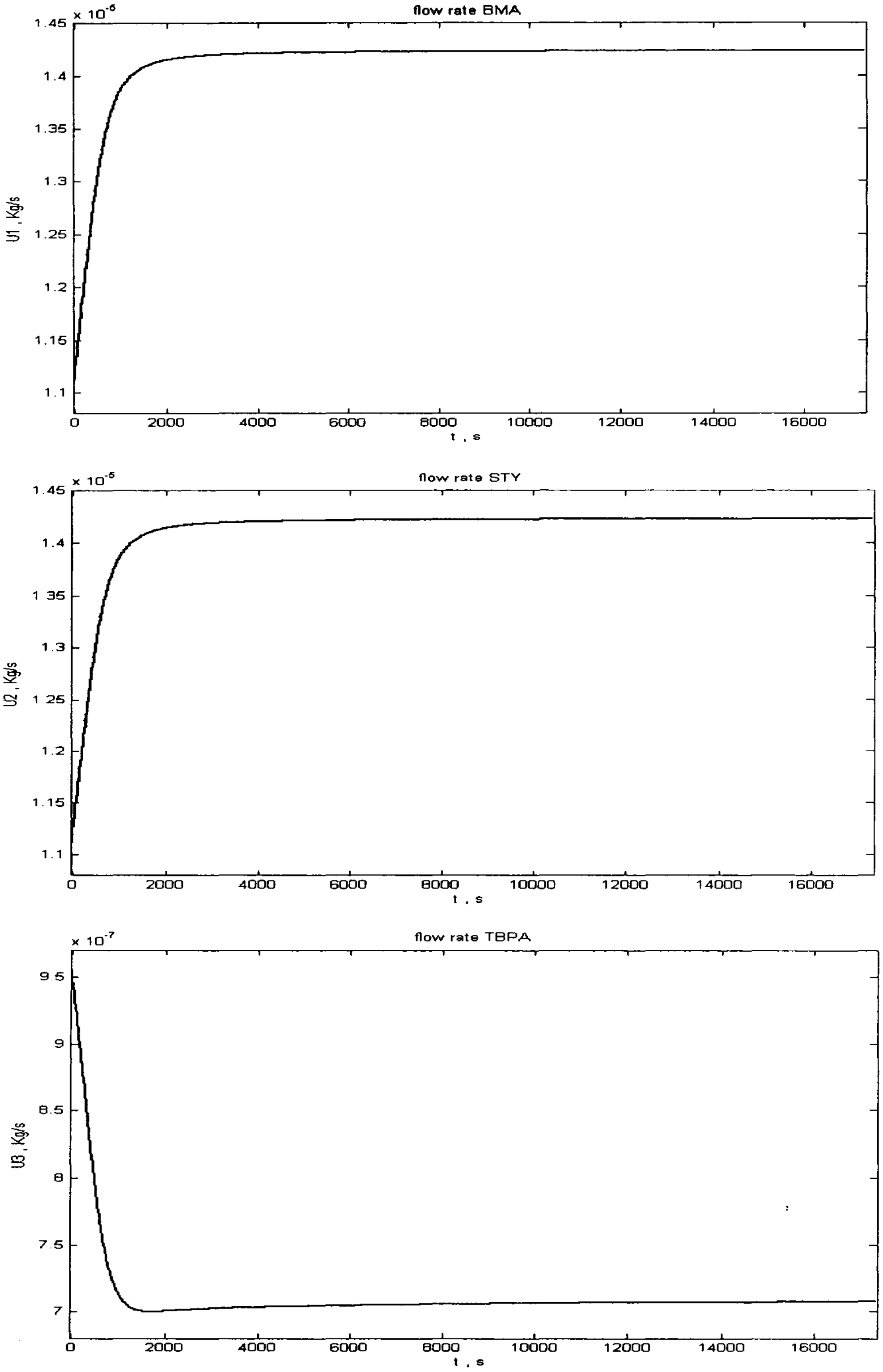


Fig.5.7 Feed profiles for Case 3

A large decrease in final batch time is reflected by a large decrease of the cost functional once polymer quality specifications are met as shown in Fig 5.8 and Table 5.8. The reduction in the objective function value is about 99.6 % relative to its initial cost as inferred from Fig.5.8 confirming the effectiveness of the optimization procedure.

As we have seen, most of the optimization work is carried out at the first portion of the batch where quick changes on the flow rates are needed in order to achieve the objectives and meet constraints. Fig.5.9 shows that the rate of adaptation of BMA and STY flow rates are fast enough for the optimization procedure to succeed. In addition, the initiator flow rate suffers a steep reduction (followed by a period of slight flat adjustments) as the TBPA free mass in the reactor reach saturation. Table 5.8 shows the total amount of raw material fed into the reactor, in reasonable agreement with the experimental values (Table 5.1). However, a significant improvement in polymer quality (molecular weight and composition control) was achieved, and the batch time was reduced by close to two hours.

Table 5.5 Initial control parameters for Case 4 corresponding to experimental conditions summarized in Table 5.1

Experimental Conditions	**75:25 BMA:STY (wt ratio)**
$\theta_1(0)$ (Kg.s^{-1})	3.69E-01/t_{ref}
$\theta_2(0)$	0.333
$\theta_3(0)$	0.0263

Table 5.6 Settings for Case 4

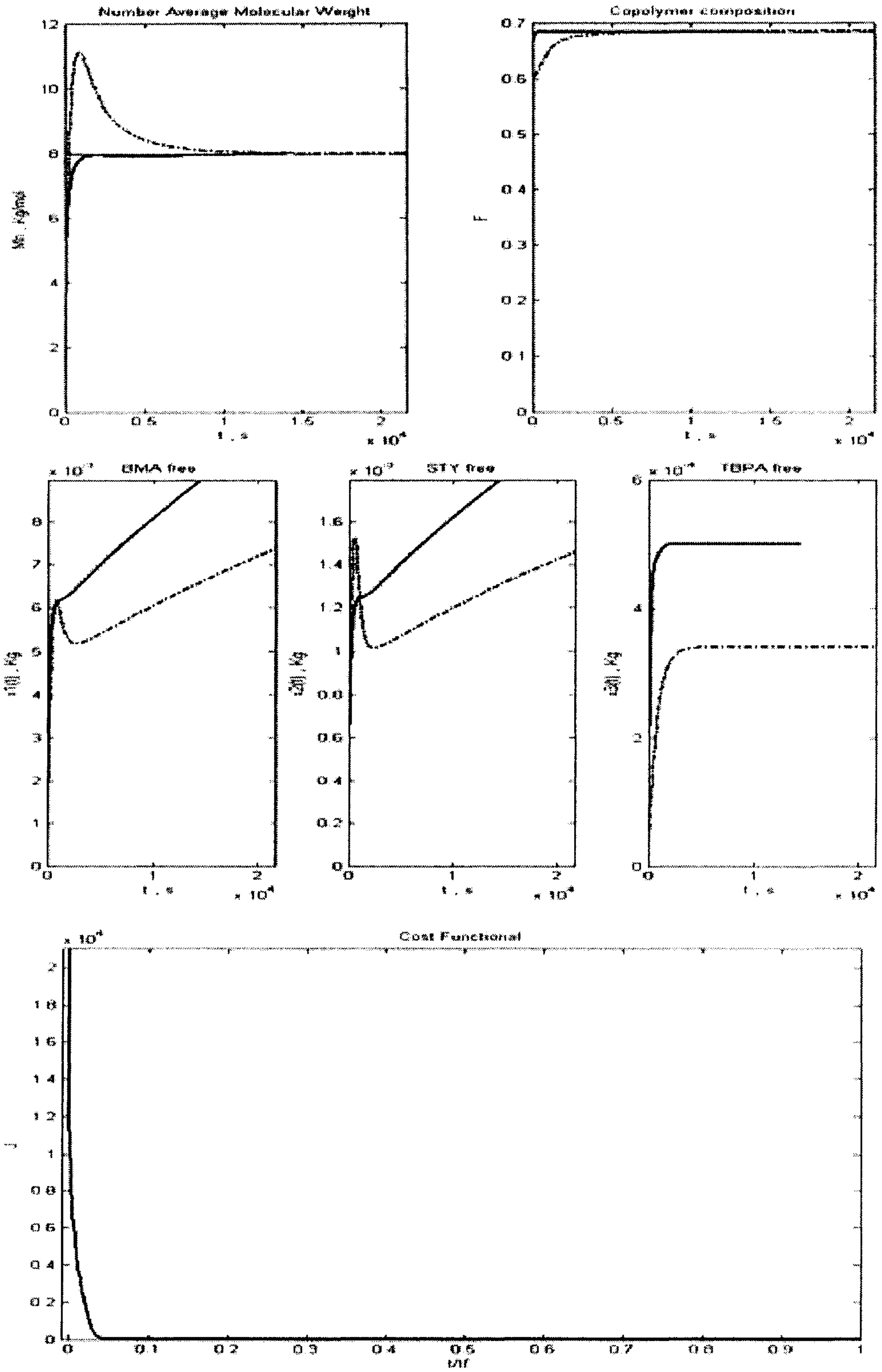
Initial Conditions	**Case 4**	
$x_1(0)$, Kg	0.215	
$x_1(0) = x_2(0) = x_3(0) = x_4(0) = x_5(0)$, Kg	0	
$x_6(0)$, mol	0	
Initial control parameters	$t_{ref} = 21600\,\text{s}$, $n = 1000$	
$\theta_1(0)$, Kg.s^{-1}	5.5E-01/t_{ref}	
$\theta_2(0)$	0.24	
$\theta_3(0)$	0.069	
$\theta_4(0)$, s	t_{ref}/n	
Target values		
M_n , Kg.mol^{-1}	7.98	
F	0.685	
Constraint bounds		
$m_{pol}^{\max}(t_f)$, Kg	0.483	
$x_k^{\min} \le x_k(t) \le x_k^{\max}$, $k \in \{1,2,3,4,5,6\}$, Kg	$x_k^{\min}$	$x_k^{\max}$
$x_1(t)$	0	0.01
$x_2(t)$	0	0.01
$x_3(t)$	0	5E-04
$x_4(t)$	0	---
$x_5(t)$	0	---
$x_6(t)$	0	---

Table 5.7 Algorithm parameters for Case 4

Algorithm Parameters	**Values**
M	9E+04
μ	1E-11
ε	1E-12
k_j	{1e-14,1e-03,1e-05,0.6}

Table 5.8 Optimization results for Case 4

Results	**Values**
m_{TBPA}^{fed} , Kg	10E-03
m_{BMA}^{fed} , Kg	3.69E-01
m_{STY}^{fed} , Kg	1.23E-01
$t_f^{ref} - t_f^{batch}$, min	116.9
$m_{pol}(t_f) - m_{pol}(t_f)$, Kg	2E-04
Computation time , s	10.58
Final Cost	79.60



Number Average Molecular Weight

Copolymer composition

BMA free

STY free

TBPA free

Cost Functional

Fig.5.8 Optimization output for case 4.
Optimization (——); conventional starved feed operation (– · ·); target (- - - -).

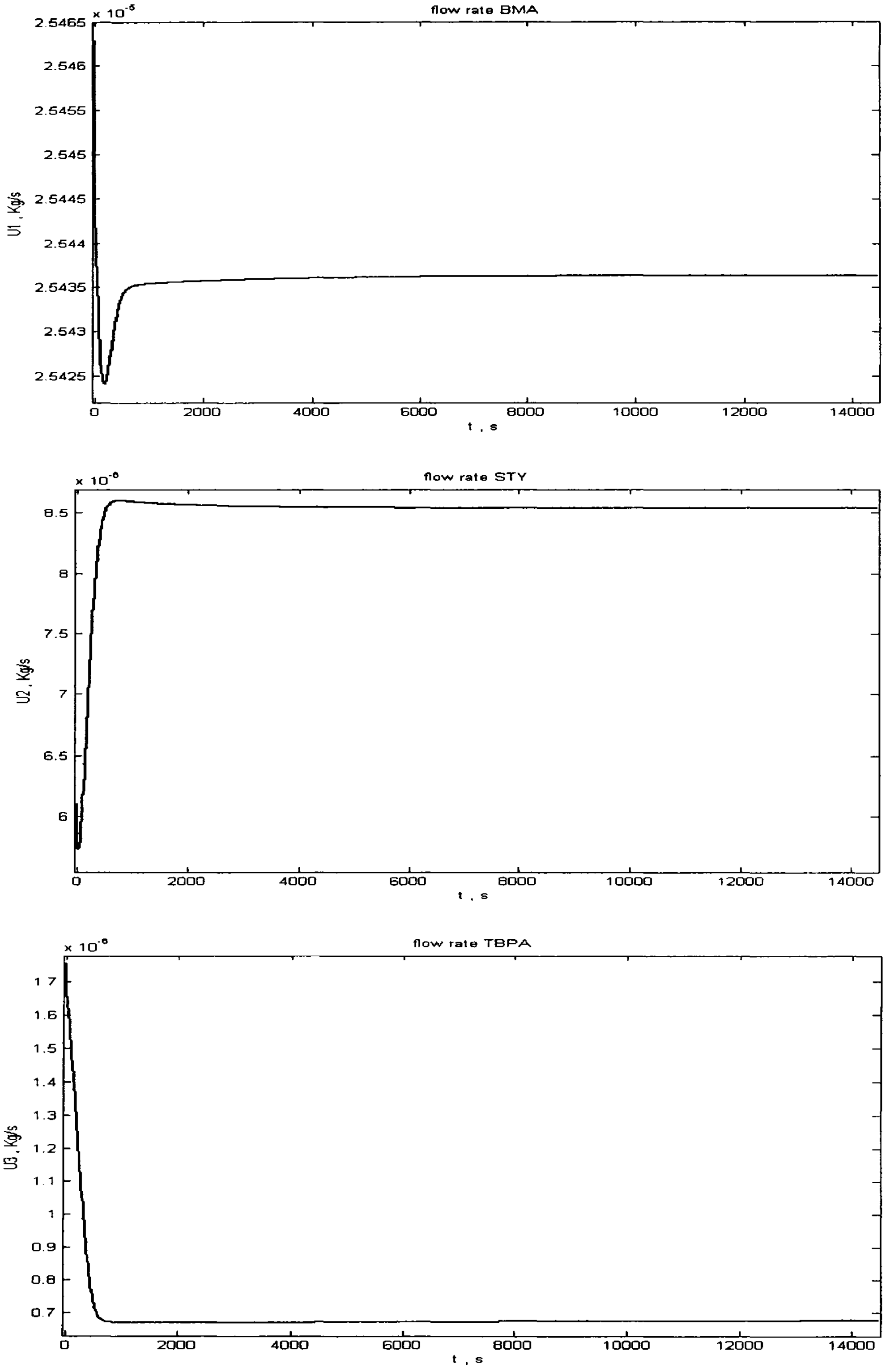


Fig.5.9 Feed profiles for Case 4

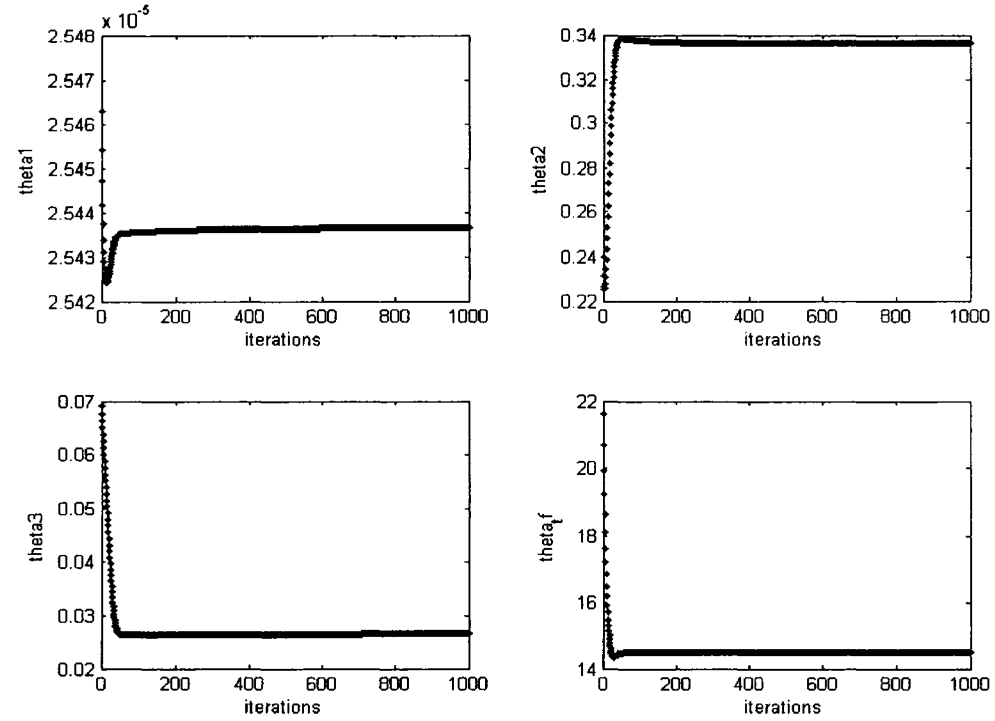


Fig.5.10 Parameter convergence for Case 4

5.2 Robustness under Model Parameter Uncertainty

Robustness under model parameters uncertainty has been analyzed and presented for Case 2 only. The robustness with respect to the homopropagation rate coefficient (k_p^{AA}) and the reactivity ratio (r_1) have been selected for study, as these parameters have a strong effect on the rate of polymerization and thus on molecular weight and copolymer composition control, respectively. Moreover, it is of interest to have a sense of the degree of model parameter uncertainty (model mismatch) under which the solution to the optimization problem is still viable.

The simulations set in which the parameter k_p^{AA} is at its nominal value (no uncertainty) is indicated in solid line and labeled as optimal. Also, the curve labeled as simulation represents the open-loop conventional starved feed policy under no uncertainty.

The effect of uncertainty in k_p^{AA} and r_l is summarized in Table 5.9 along with the reference Case 2 optimal results. Those results correspond to a variation in k_p^{AA} and r_l of ± 20 % and ± 30 % relative to their nominal values. The results are presently graphically in Fig 5.11 through Fig 5.16.

The constrained variables both in presence of -20% and +20% uncertainties in model parameter k_p^{AA} are shown in Fig 5.12. Similarly, constrained state trajectories were obtained for uncertainties in r_l and can be found in Fig 5.15. Deviation on monomer constrained trajectories are no more than ± 0.001 Kg for variations in k_p^{AA} and no more than ± 0.002 Kg for uncertainties in r_l whereas initiator constrained trajectory shows a better agreement with that of the optimal solution for both model parameters.

The flow rates of monomers and initiator corresponding to the optimal strategies in presence of $\pm 20\%$ uncertainty in k_p^{AA} are shown in Fig 5.13, and the control policies under r_l uncertainty ($\pm 30\%$ nominal value) are presented in Fig 5.16. From Table 5.9, it can be seen that the load of raw material fed to the reactor is consistent with the optimal solution (Case 2) while the final batch time achieved does not differ by more than 10 % relative to Case 2 for both model parameters.

Table 5.9 Optimization results under model parameter uncertainty

Results	120 % k_p^{AA}	80 % k_p^{AA}	130 % r_l	70 % r_l	**Case 2 (optimal)**
m_{TBPA}^{fed} , Kg	9.9E-03	9.6E-03	9.6E-03	10E-03	9.8E-03
m_{BMA}^{fed} , Kg	2.44E-01	2.46E-01	2.46E-01	2.44E-01	2.44E-01
m_{STY}^{fed} , Kg	2.44E-01	2.46E-01	2.46E-01	2.44E-01	2.44E-01
$t_f^{ref} - t_f^{batch}$, min	73.7	81.8	83.5	69.5	77.2

The growth rate of polymer chains is increased as the propagation rate constant is increased, which results in the molecular weight slightly drifting away from the optimal curve momentarily until the control is recovered. The decrease in k_p^{AA} shows the opposite effect, a temporary negative deviation on molecular weight, as seen in Fig 5.11. An uncertainty of up to $\pm 20\%$ in k_p^{AA} is well tolerated while still achieving good polymer quality control, constraint satisfaction (see Fig 5.12) and a reduction in batch time. Time reduction is favored by feeding the monomers and initiator at higher rates as it can be observed from Fig 5.13 for 80 % k_p^{AA} policies. The same turned out to be true for higher values of the reactivity ratio (+ 30 % uncertainty) as seen in Fig 5.16. The opposite is also true as shown by the corresponding curves of 120% k_p^{AA} and 70 % r_l uncertainties (see also Table 5.9).

On the other hand, the reactivity ratios reflect the inherent tendencies of a radical to react with its own monomer relative to the comonomer. It is expected that a change on the reactivity ratios will give rise to a variation in copolymer composition. Indeed, Fig 5.14

shows the copolymer composition drifting away from the optimal curve (and from the target value) as the uncertainty in r_l is introduced. In particular, an uncertainty of +30 % in r_l causes a temporary positive deviation from its target value while -30 % in r_l yields the opposite effect. Uncertainty of 30 % in r_l is also well tolerated while achieving the objectives and meeting constraints (see Fig 5.15).

5.3 Summary

In this section a real-time optimization technique to optimize the control profiles over the remaining batch as described in Chapter 4 has been used. The results of the optimization show that considerable improvements in polymer quality (less drift in molecular weight and copolymer composition) and productivity (batch time reduction) can be obtained while satisfying the constraints imposed to the system. Moreover, this novel technique and the use of a reduced order model result in low computational effort which can be further exploited through the use of a more complex model that improves the accuracy of the model predictions.

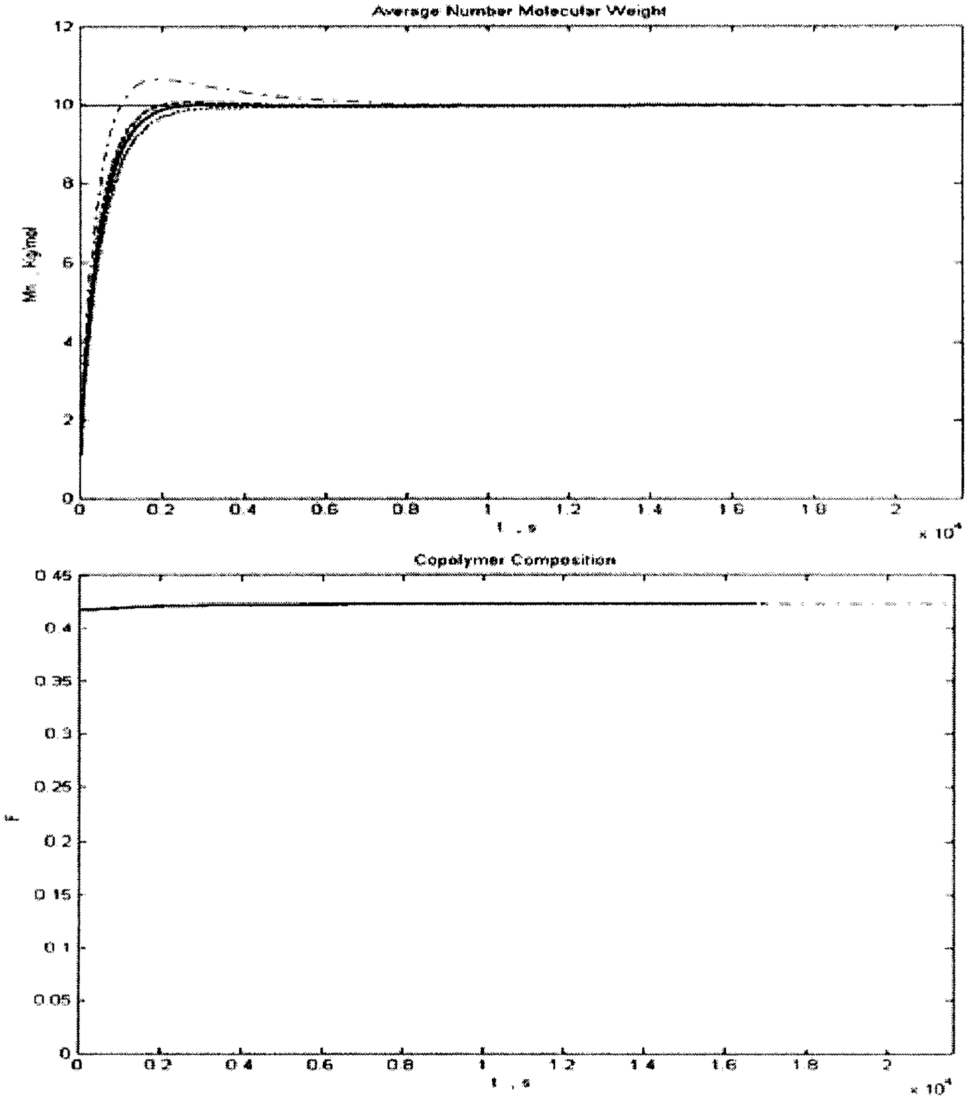


Fig.5.11 Optimization under uncertainty in k_p^{AA}.

Optimal (——); 80 % k_p^{AA} (……); 120 % k_p^{AA} (– –); simulation (-·-·-).

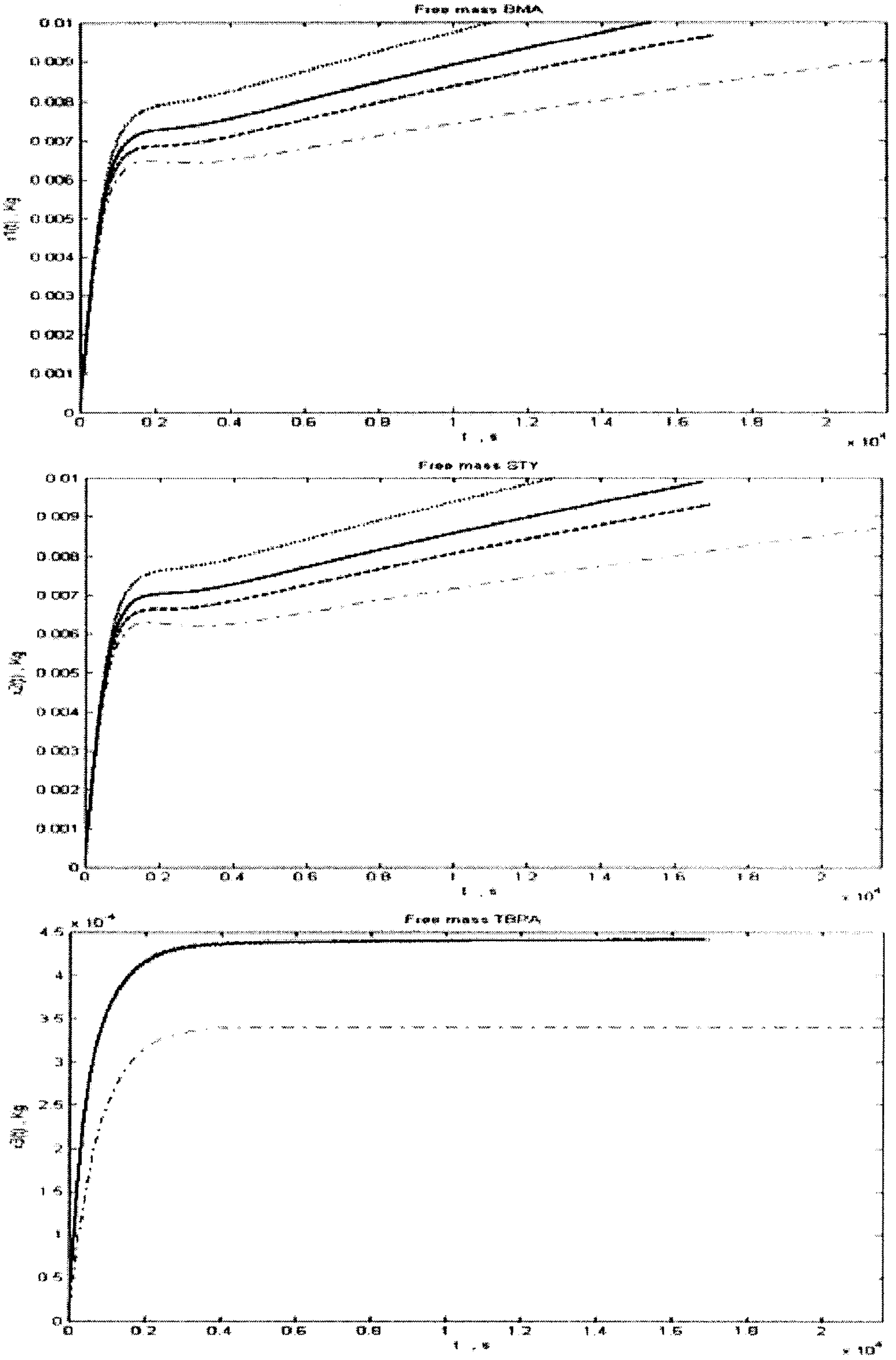


Fig.5.12 State trajectory under uncertainty in k_p^{AA}.

Optimal (——); 80 % k_p^{AA} (……); 120 % k_p^{AA} (– –); simulation (-·-·-).

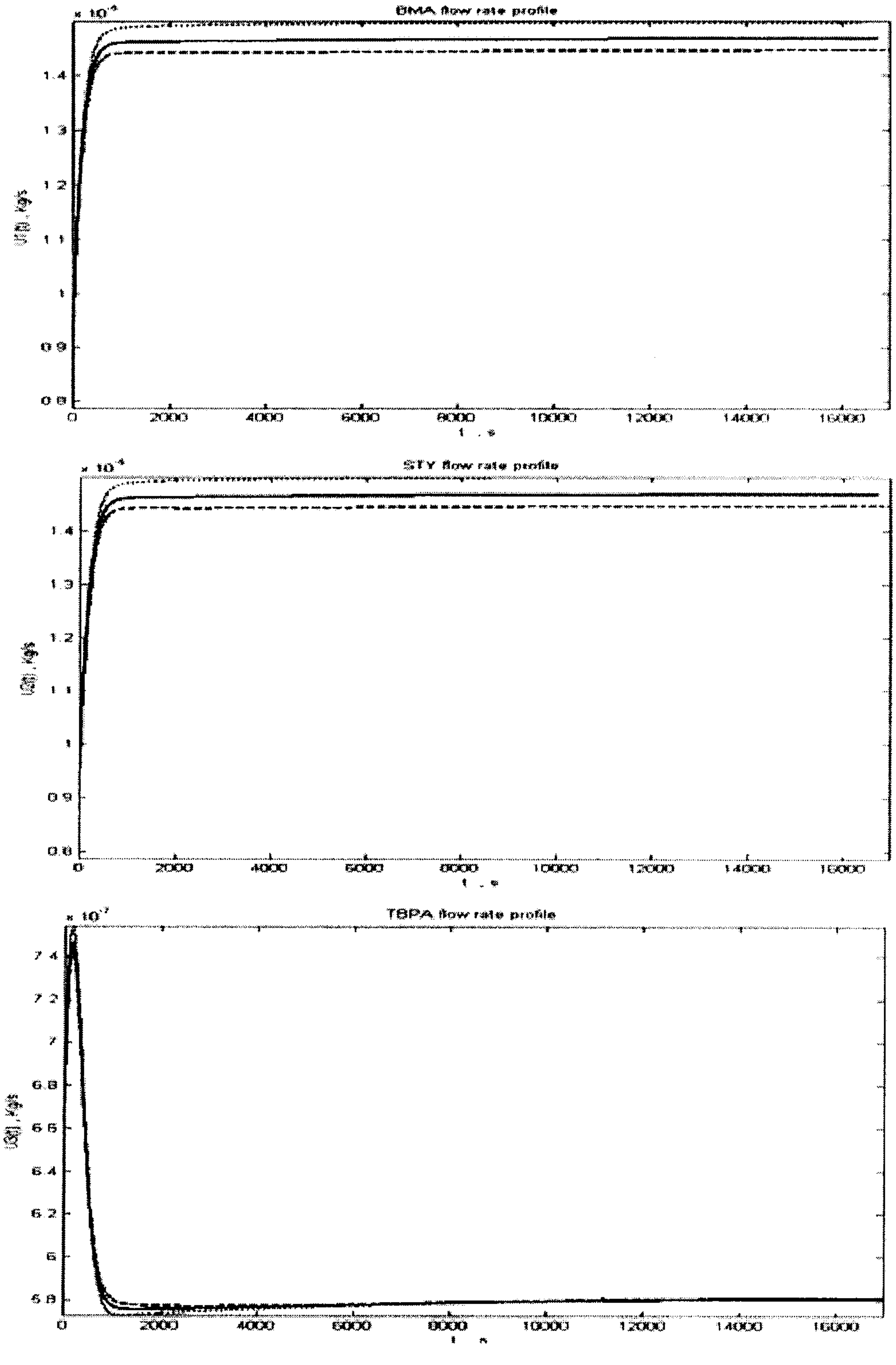


Fig.5.13 Control policies under uncertainty in k_p^{AA}.

Optimal (——); 80 % k_p^{AA} (……); 120 % k_p^{AA} (– –).

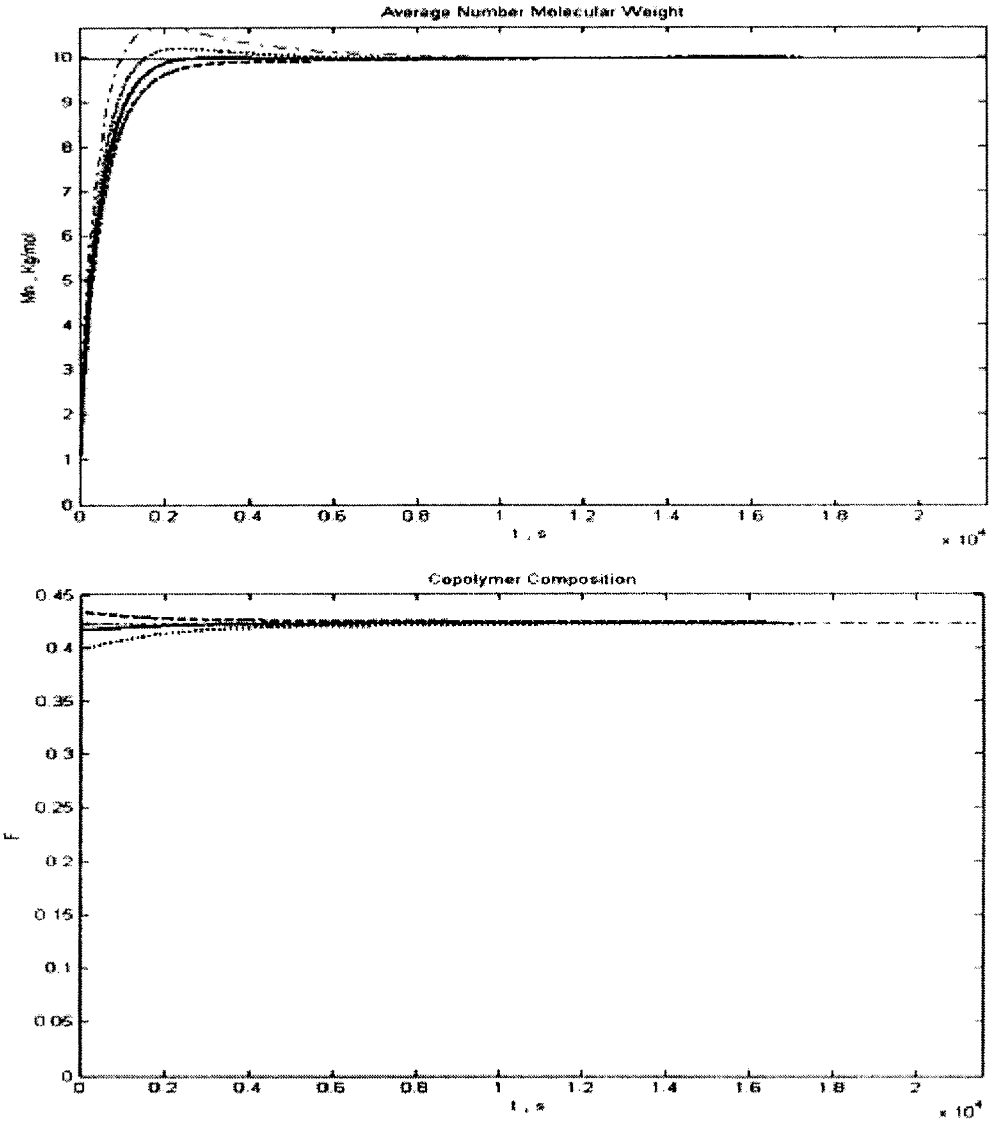


Fig.5.14 Optimization under uncertainty in r_l.
Optimal (——); 70 %r_1(……); 130 %r_1(– –); simulation (-·-·-).

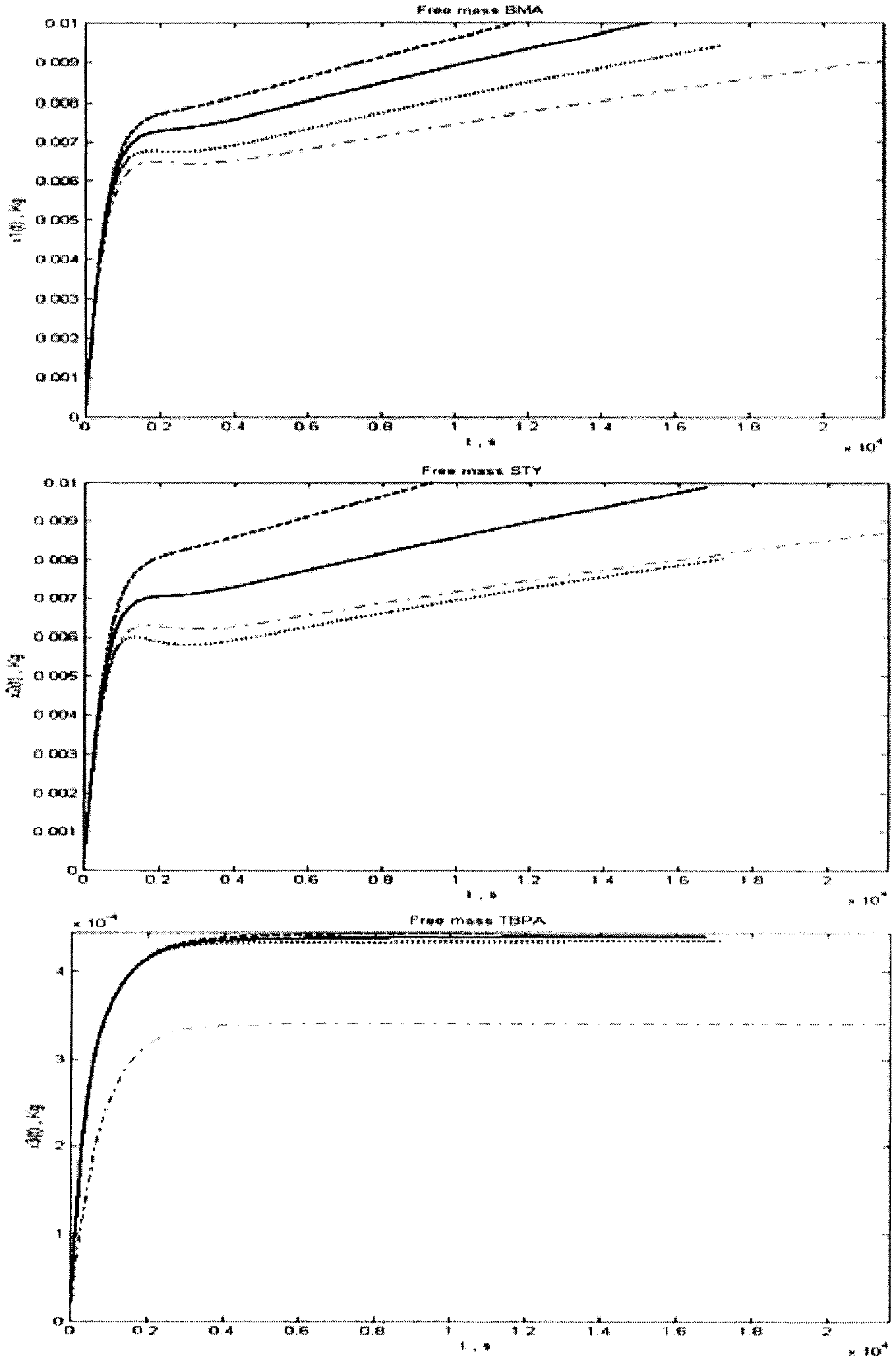


Fig.5.15 State trajectory under uncertainty in r_l.

Optimal (——); 70 %r_1(……); 130 %r_1(– –); simulation (-·-·-).

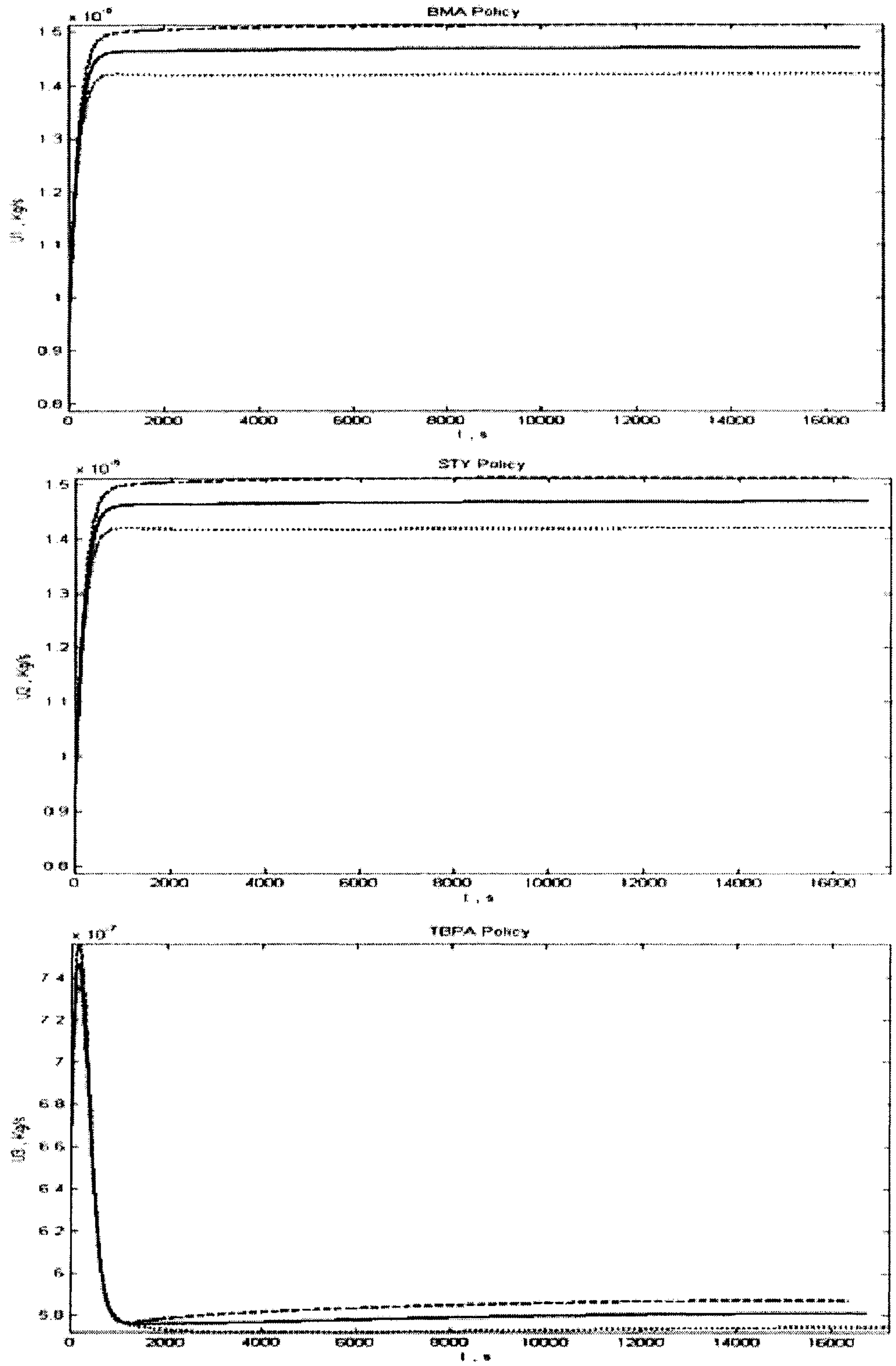


Fig.5.16 Control policies under uncertainty in r_l.
Optimal (——); 70 %r_1 (······); 130 %r_1 (– –).

Chapter 6 Conclusions and Future Work

6.1 Conclusions

In this study, the development of an optimization based control strategy for a semi-batch polymerization reactor was investigated. A reduced mechanistic model was developed based on a proposed set of reaction mechanisms of high-temperature solution polymerization of BMA and STY. It was shown that the model can capture the underlying dynamics of the system making it suitable for control studies.

The actual process is being operated under starved-feed conditions, a strategy adopted for safety reasons and to maintain good composition and molecular weight control in the absence of on-line measurement. Such a scenario is taken as the basis for the study of real-time optimization in this work through the application of a novel technique [56] as desicribed in Chapter 3.

Optimal control policies are calculated based on minimization of a performance index that is formulated based on desired quality and productivity requirements such as minimization of batch time, molecular weight and copolymer composition deviation along the batch. The applicability of the proposed methodology was presented for various practical examples.

The results of the on-line optimization based control strategy show that considerable improvements in polymer quality and in productivity can be achieved. In particular, for Case 2 (50:50 BMA:STY mass ratio) the batch time can be reduced by 21% and for Case

4 (75:25 BMA:STY mass ratio) by 32% compared with the conventional starved-feed operation. Moreover, the control strategy holds the free monomer and initiator in the reactor low throughout the batch. This reduction in batch time is achieved while also improving product quality, achieved by manipulating the initial feed rate of monomers and initiator.

Optimization results reveal implementable control policies that allow for the production of higher quality polymer resins at lower costs. Also, the obtained constrained trajectories for all the variables of interest are within their expected range as determined by the path constraints. Furthermore, the production of a desired amount of polymer at the batch end is assured while requiring roughly the same load of monomers and initiator as used under the conventional starved-feed operation for the same molecular weight and composition target values; as shown by Case 3 a change in molecular weight target results in a change in the amount of initiator required.

The results for Case 2, the main example in this work, show that the calculated feed flow rates are relatively high at the beginning of the batch and then level off or decreases for monomers and initiator feed, respectively, as the end of the batch is approached. Not surprisingly, adding feed more quickly allows for the reduction of the batch time as the limits on the constrained state trajectories (free BMA, STY and TBPA in the reactor) are approached. Allowing the feed flow rates to be higher by increasing the limits on the constrained variables is expected to result in a decrease of batch time. However, increasing the ultimate limit values chosen did not result in significant improvement.

Nonetheless, it was noticed that an improvement on batch time reduction can be obtained by increasing the barrier parameter value (μ_l) until a point at which any further improvement comes at the expense of the polymer quality (reflected by unacceptable deviations from targets and an increase on the final cost function value). Also, it was seen that lower values of μ_l result in less batch time reduction for the same constraint bounds. This is expected as decreases in the barrier parameter diminish the effect of the barrier term, so that the iterates can gradually approach the boundary of the feasible region as determined by the set of constraints (which may result in constraint saturation) restricting the objective value improvement and therefore the batch time reduction.

With regard to the extent of deviation from the optimal solution in the presence of uncertainty, this varies from parameter to parameter. Some parameters do not influence the optimal solution under uncertainty, while uncertainty in others may lead to deviations from optimal solution. No important deviation in polymer quality and productivity was found for the model parameters chosen for study in this work.

Furthermore, the maintenance of the program, that is changes in the process model, control problem formulation (performance index and set of constraints) and changes in quality specifications, turns out to be of relatively reduced difficulty. It does not require complete redesign, as opposed to control input parameterization using other functional forms.

6.2 Future Work

A natural extension of this work will be the experimental validation of the optimization results in laboratory and production scale. That is, the strategy implemented in the actual batch operation and a comparison of the experimental results with those obtained from the computer simulations presented in this work.

Due to the efficiency of the real-time optimization scheme, in terms of fast calculation of the optimal profile at any sampling instant, a more complex model would still be feasible for on-line implementation. This unused capacity of the real-time strategy may lead to consider the inclusion of additional kinetic mechanisms such as BMA depropagation or penultimate chain-growth kinetics to improve the accuracy of the model predictions especially for BMA-rich recipes. In addition, it would be interesting to analyze the structure of optimal solution with a model that includes an energy balance, and to extend the control strategy to the calculation of an optimal temperature profile. Further constraints on the total amount of raw material used in a run (e.g., initiator) or constraints including cooling system limitations against cooling failure could also be considered.

Another important direction for further study to be explored is the estimation problem. Currently, the closed-loop control strategy design is based only on state predictions and no actual plant measurements. What measurements are available versus what are needed for updating the model needs to be further evaluated. Moreover, when process operation is limited by state constraints, it is essential to have accurate state estimates or measurements for the process to be operated as close as possible to the restrictions for optimal performance. Thus, it would also be of interest to study the impact of measurement noise on the transient and stationary behaviors of optimal profiles and the overall control performance.

References

[1] Abdennour, R., Ksouri, M., M'sahli, F.(2002), “Experimental Nonlinear Model Based Predictive Control for a Class of Semi-Batch Chemical Reactors”, The International Journal of Advanced Manufacturing Technology, Computer Science and Engineering, 20(6), 459-463.

[2] Abel, O. and Marquardt, W.(1998), “A model predictive control scheme for save and optimal operation of exothermic semi–batch reactors”, in ‘Proceedings of IFAC Symposium Dycops–5’, 761–766.

[3] Abel, O., Helbig, A., Marquardt, W. (1998), “Optimization approaches to control–integrated design of industrial batch reactors”, Nonlinear Model Based Process Control, Nato Asi Series, Kluwer academic publishers, pp. 513–551.

[4] Abel, O., Helbig, A., Marquardt, W., Zwick, H., Daszkowski, T. (2000), “Productivity optimization of an industrial semi–batch polymerization reactor under safety constraints”, Journal of Process Control, 10, 351–362.

[5] Abel, O. and Marquardt, W. (2003), “Scenario-integrated on-line optimization of batch reactors”, Journal of Process Control, 13, 703–715.

[6] Batina, I. (2004), “Model predictive control for stochastic systems by randomized algorithms”, PhD thesis, Eindhoven: Technische Universiteit Eindhoven.

[7] Biegler, L.T., “Nonlinear programming: concepts, algorithms and applications lecture series”, Chemical Engineering Department, Carnegie Mellon University.

[8] Biegler, L.T., and R.R. Hughes (1983). “Process optimization: A comparative case study”, Computers and Chemical Engineering, 7(5), 645-661.

[9] Biegler, L. T. (1984), “Solution of dynamic optimization problems by successive quadratic programming and orthogonal collocation”, Computers and Chemical Engineering, 8, 243-248.

[10] Biegler, L. T., Cervantes, A. M., Wächter, M. A.(2002), “Advances in simultaneous strategies for dynamic process optimization”, Chemical Engineering Science, 57, 575-593.

[11] Bindlish, R. and Rawlings, J. B. (2003), “Target linearization and model predictive control of polymerization processes”, AIChE Journal, 49, 2885.

[12] Bock, H.G., and Platt, K.J. (1984), "A multiple shooting algorithm for direct solution of optimal control problems", In 'the 9th IFAC World Congress', 242-247, Budapest.

[13] Bonvin, D., Srinivasan, B., Ruppen, D.(2001), "Dynamic optimization in the batch chemical industry", in 'Proceedings of the CPC-VI Conference, American Institute of Chemical Engineers, Symposium Series', 32(98), 255-273.

[14] Cao, G., Zhu, Z., Zhang, M., Le, H., Yuan W. (2001), "Kinetics of NMBA polymerization in a starved feed reactor", Journal of Applied Polymer Science , 81(9), 2068-2075.

[15] Cervantes, A. and Biegler, L. T. (1998), "Large-scale DAE optimization using a simultaneous NLP formulation", American Institute of Chemical Engineers Journal, 44(5), 1038-1050.

[16] Chen, H. and Allgöwer, F.(1998), "Nonlinear model predictive control schemes with guaranteed stability", Nonlinear Model Based Process Control, Nato-Asi Series kluwer academic publishers, Dordrecht, The Netherlands, 465–494.

[17] Chylla, R. W. and Haase, D. (1993), "Temperature control of semibatch polymerization Reactors", Computers and Chemical Engineering, 17, 257.

[18] Congalidis, J.P., J.R. Richard, and W.H. Ray (1989), "Feedforward and Feedback Control of a Solution Copolymerization Reactor", AIChE Journal, 35, 891-907.

[19] Cueli, J.R. and Bordons, C. (2005), "Iterative nonlinear control of a semibatch reactor. Stability analysis", in 'Proceedings of the 44th IEEE Conference on Decision and Control, and The European Control Conference', Seville, Spain.

[20] Cuthrell, J.E. and Biegler, L.T. (1987), "On the optimization of differential-algebraic process systems", AIChE Journal, 33, 1257–1270.

[21] Cuthrell, J. E. and Biegler, L. T.(1989), "Simultaneous optimization methods for batch reactor control profiles", Computers and Chemical Engineering ,13, 49-62.

[22] De Oliveira, N.M.C. and Biegler, L.T.(1994), "Constraint handling and stability properties of model predictive control", AIChE Journal, 40(7), 1138–1155.

[23] Diehla, M., Bocka, H.J., Schlödera, J.P, Findeisenb, R., Nagyc, Z., Allgöwerb, F.(2002), "Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations", Journal of Process Control, 12, 577–585.

[24] Eaton, J. W., and J. B. Rawlings (1990), "Feedback Control of Chemical Processes Using On-Line Optimization Techniques", Computers and Chemical Engineering, 14, 469.

[25] Fiber, S., Gesthuisen, R., Engell, S. (2002), "Re-optimization based control of copolymer quality in semi-continuous emulsion polymerization process", in 'Proceedings of the American Control Conference Anchorage', AK May 8-10.

[26] Flores-Tlacuahuac, A. and Biegler, L.T. (2004), "Dynamic optimization of HIPS open-loop unstable polymerization reactors", Department of Chemical Engineering, Carnegie-Mellon University.

[27] Furlonge, H. I., Pantelides, C. C., Sorensen, E. (1999), "Optimal operation of multivessel batch distillation columns", American Institute of Chemical Engineers Journal, 45(4), 781-800.

[28] Garcia, C.E., Prett, D.M., Morari, M., (1989), "Model predictive control: theory and practice—a survey", Automatica, 25(3), 335–348.

[29] Goh, C. J. and Teo, K. L. (1988), "Control parameterization: a unified approach to optimal control problems with general constraints", Automatica, 24, 3-18.

[30] Guay, M. and Zhang, T.(2003), "Adaptive extremum-seeking control of nonlinear systems with parametric uncertainties", Automatica, 30, 1283-1293.

[31] Helbig, A., Abel, O., M'hamdi, A., Marquardt, W. (1996), "Analysis and nonlinear model predictive control of the chylla–haase benchmark problem", in 'Proceedings of CONTROL'96', Exeter, UK, 1172–1177.

[32] Helbig, A., Abel, O., Marquardt, W. (1998), "Model predictive control for on-line optimization of semi-batch reactors", in 'Proceedings of the American Control Conference Philadelphia', Pennsylvania.

[33] Hertzberg, T., and Asbjornsen, O.A. (1977), "Parameter estimation in nonlinear differential equations", Computer Applications in the Analysis of Data and Plants, Science Press, Princeton.

[34] Hutchinson, R.A. (2005), "Free-radical polymerization: homogeneous", Handbook of polymer reaction engineering, 153-211.

[35] Ioannou, P. and Sun, J.(1996), Robust Adaptive Control , Prentice Hall, Inc,electronic copy at http://www-rcf.usc.edu/~ioannou/Robust_Adaptive_Control.htm.

[36] Ishikawa, T., Liberis, L., Natori, Y., Pantelides, C.C.(1997), "Modeling and optimisation of an industrial batch process for the production of dioctyl phthalate", Computers and Chemical Engineering, 21(suppl.), 1239-1244.

[37] Jarupintusophon, P., Lelann, M.V., Cabassud, M., Casamatta, G. (1995), "Realistic model-based predictive and adaptive control of batch reactor", Computers and Chemical Engineering, 18, s445-s449.

[38] Kraft, D. (1985), "On converting optimal control problems into nonlinear programming problems", in Computational Mathematical Programming, K. Schittkowski, ed., NATO ASI Series F: Computer and Systems Sciences 15, Springer Verlag, Berlin and Heidelberg, 261-280

[39] Lau, K.H., and Ulrichson, D.L. (1992), "Effects of local constraints on the convergence behavior of sequential modular simulators", Computers and Chemical Engineering, 15(9), 887-892.

[40] Lehtonen, J., Salmi, T., Vuori, A., Haario, H. (1997), "Optimization of the reaction conditions for complex kinetics in a semibatch reactor", Industrial and Engineering Chemistry Research, 36(12), 5196-5206.

[41] Leis, K. and Kramer, M.(1988), "Algorithm 658 odessa – and ordinary differential equation solver with explicit simultaneous sensitivity equations "

[42] Li, D. and Hutchinson, R.A.(2006), "High temperature semibatch free radical copolymerization of butyl methacrylate and styrene", Macromolecular Symposium, 243, 24–34.

[43] Li, D. (2006), "Modeling of Kinetic Complexities in High Temperature Free Radical Polymerization for Production of Acrylic Coatings Resins", PhD thesis, Queen's University.

[44] Logsdon, J.S. and L.T. Biegler (1989), "On the Accurate Solution of Differential-Algebraic Optimization Problems", I&EC Research, 28, 1628.

[45] Luus, R., Iterative dynamic programming, Chapman & hall monographs and surveys in pure and applied mathematics.

[46] Luus, R. and Rosen, O. (1991), "Application of dynamic programming to final state constrained optimal control problems", Industrial and Engineering Chemistry Research, 30, 1525-1530.

[47] Luus, R. (1994), "Optimal control of batch reactors by iterative dynamic programming", Journal of Process Control, 4, 218-226.

[48] Maner, Bryon R. and Doyle, Francis J. (1997) , "Polymerization Reactor Control Using Autoregressive-Plus Volterra-Based MPC", AIChE Journal, 43(7), 1763-1784.

[49] Mayne, D.Q., Rawlings, J.B., Rao, C.V., Scokaert, P.(2000), "Constrained model predictive control: stability and optimality", Automatica, 36 (6) ,789–814.

[50] M'hamdi, A., Helbig, A., Abel, O., Marquardt,W.(1996), "Newton-type receding horizon control and state estimation - a case study' , preprints of the 13th World Congress of IFAC, volume m, 121-126.

[51] Mujtaba, M. and Macchietto, S.(1997), "Efficient optimization of batch distillation with chemical reaction using polynomial curve fitting technique", Industrial and Engineering Chemistry Research, 36(6), 2287-2295.

[52] Muske, K.R. and Rawlings, J.B. (1993), "Model predictive control with linear models", AIChE Journal, 39(2), 262–287.

[53] Nagy, Z. and Agachi, S. (1997), "Model predictive control of a PVC batch reactor", Computers and Chemical Engineering, 21(6), 571-591.

[54] Özkan, L., Kothare, M. V., & Georgakis, C. (2003), "Control of a solution copolymerization reactor using multi-model predictive control", Chemical Engineering Science, 58, 1207.

[55] Palanki, S. and Vemuri, J. (2005), "Optimal operation of semi-batch processes with a single reaction", International Journal Chemical Reactor Engineering, Vol 3, Article A17.

[56] Peters, N.J. (2005), "Real-Time Dynamic Optimization of Nonlinear Batch Systems", MScE thesis, Queen's University.

[57] Peterson, T., Hernandez, E., Arkun, Y., Schork, F.J. (1992), "A nonlinear DMC algorithm and its applications to a semibatch polymerization reactor", Chemical Engineering Science, 47(4), 737-753.

[58] Pollard, G. P. and Sargent, R. W. H.(1970), "Off line computation of optimum controls for a plate distillation column", Automatica, 6 ,59-76.

[59] Rantow, F.S., Soroush, M., Grady, M.C. (2005), "Optimal control of a high temperature semi-batch solution polymerization reactor", American Control Conference June 8-10, Portland, OR, USA.

[60] Rawlings, J.B. and Muske, K.R.(1993), "The stabitity of constrained receding horizon control", IEEE Transactions on Automatic Control, 38, 1512–1516.

[61] Renfro, J.G., Morshedi, A.M. and Asbjornsen, O.A. (1987), "Simultaneous optimization and solution of systems described by differential / algebraic equations", Computers and Chemical Engineering. 11, 503–507.

[62] Richards, J.R. and Congalidis, J.P. (2006), "Measurement and control of polymerization reactors", Computers and Chemical Engineering, 30, 1447–1463.

[63] Ricker, N.L. (1991), "Model Predictive Control: State of the Art, Chemical Process Control", 'Fourth International Conference on Chemical Process Control', 271-296, Amsterdam.

[64] Ruppen, D.(1994), A contribution to the implementation of adaptive optimal operation for discontinuous chemical reactors, PhD thesis, ETH Zürich.

[65] Ruppen, D., Benthack, C. and Bonvin, D. (1995), "Optimization of batch reactor operation under parametric uncertainty - Computational aspects", Process Control, 5(4), 235-240.

[66] Ryali, V. and Moudgalya, K.M. (1998), "Transfer function modeling and robust control of chylla-haase challenge control problem", Systems and Control Group, Dept. of Chemical Engineering, Iit Bombay.

[67] Sargent R.W.H. and Sullivan, G.R. (1977), "The development of an efficient optimal control package", in 'Proc. of the 8th IFIP Conference on Optimization Technologies', 158-168.

[68] Schlegel, M. and Marquardt, W. (2006), "Detection and exploitation of the control switching structure in the solution of dynamic optimization problems", Journal of Process Control, 16, 275–290.

[69] Seki H., Ogawa M., Ooyama S., Akamatsu K.; Ohshima M.; Yang W.(2001), "Industrial application of a nonlinear model predictive control to polymerization reactors", Control Engineering Practice, 9(8), 819-828.

[70] Srinivasan, B., Bonvin, D., Visser, E., Palanki, S. (2003), "Dynamic optimization of batch processes: role of measurements in handling uncertainty", Computers and Chemical Engineering, 27(1), 27-44.

[71] Srinivasan, B. and Bonvin, D.(2004), "Dynamic optimization under uncertainty via NCO tracking: a solution model approach" , In 'BatchPro Symposium' , 17-35.

[72] Storen, S. and Hertzberg, T. (1995), "The sequential linear quadratic programming algorithm for solving dynamic optimization problems - a review", Computers & Chemical Engineering, Vol. 19, s495-s500.

[73] Tabash, R. and Teymour, F. (2004), "Analyzing compositional drift transients in copolymerization systems using digital encoding", Chemical Engineering Science, 59, 5129–5137.

[74] Tanartkit, P. and L. T. Biegler (1995), "Stable Decomposition for Dynamic Optimization", I &EC Research, 34, 1253.

[75] Terwiesch, P.(1995), "Cautious on-line correction of batch process operation", AIChE Journal, 41(5), 1337-1340.

[76] Toulouse, C., Cezerac, J., Cabassud, M., Lelann, V., Casamatta, G. (1996), "Optimisation and scale-up of batch chemical reactors: impact of safety constraints", Chemical Engineering Science, 51(10), 2243-2252.

[77] Tsen, A.Y., Jang, S.S., Wong, D.S.H., Joseph, B.(1996), "Predictive control of quality in batch polymerization using hybrid ANN models", AIChE Journal, 42(2), 455–465.

[78] Tyner, D., Soroush, M., Grady, M.C., Richards, J., Congalidis, J.P.(2000), "Mathematical modeling and optimization of a semi-batch polymerization reactor" in 'Proceedings of ADCHEM International Symposium on Advanced Control of Chemical Processes'.

[79] Urretabizkaia, A., Arzamendi,G., Unzue,M.J., Asua,J.M. (1994), "High solids content emulsion terpolymerization of vinyl acetate, methyl methacrylate and butyl acrylate, I. open loop composition control", Journal of polymer science, part A: Polymer Chemistry, 32(9), 1779-1788.

[80] Vassiliadis, V. S., Sargent, R. W. H., Pantelides, C. C.(1994a), "Solution of a class of multistage dynamic optimization problems without path constraints", Industrial and Engineering Chemistry Research, 33(9), 2111-2122.

[81] Vassiliadis, V. S., Sargent, R. W. H., Pantelides, C. C. (1994b), "Solution of a class of multistage dynamic optimization problems: problems with path constraints", Industrial and Engineering Chemistry Research, 33(9), 2123-2133.

[82] Vemuri, Y.J. (2004), Real-time optimization of semi-batch reactors, PhD thesis, The Florida State University.

[83] Villadsen, J. and Michelsen, M. L.(1978), "Solution of differential equation models by polynomial approximation" . Englewood Cliffs: Prentice-Hall.

[84] Visser, E., Srinivasan, B., Palanki, S., Bonvin, D. (2000), "A feedback-based implementation scheme for batch process optimization", Journal of Process Control, 10, 399-410.

[85] Vlassenbroeck, J. (1988), "A chebyskev polynomial method for optimal control with state constraints", Automatica, 24, 499–506.

[86] Willis, M. aqnd Tham M.T., Chemical advanced process control, Engineering and advanced materials, University of Newcastle upon Tyne.
http://lorien.ncl.ac.uk/ming/advcontrl/sect3.htm

[87] Wulkow, M., Predici simulation package for polyreactions, software manual.

[88] Xiong Z. and Zhang, J. (2005), "optimal control of fed-batch processes based on multiple neural networks", Applied Intelligence Journal, 22, 149–161.

[89] Yoon, W.J., Kim, Y.S., Kim, I.S., Choi, K.Y. (2004), "Recent advances in polymer reaction engineering: modeling and control of polymer properties", Korean Journal on Chemical Engineering, 21(1), 147-167.

[90] Zheng, A. and Morari, M. (1995), "Stability of model predictive control with mixed constraints", IEEE Transactions on Automatic Control, 40 (10), 1818–1823.

Appendices

Appendix A: BMA/STY Reduced Process Model

A.1 Matlab Reduced Process Model

new_model2.m

In the model , A , B, I stand for BMA , STY and TBPA respectively

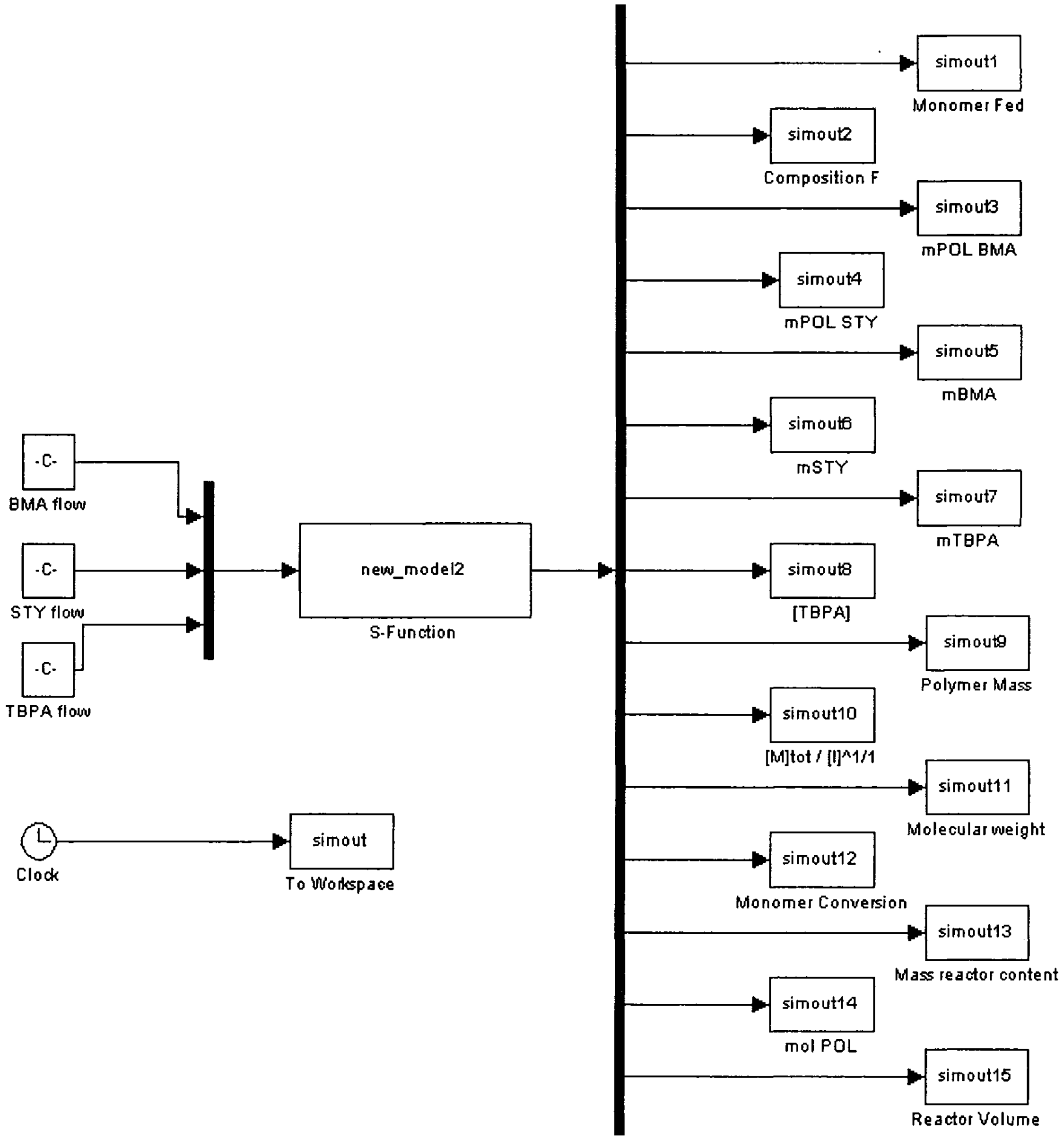
```
function [sys, y0,str,ts] = Reactor_Sim(t,y,u,flag,o1,o2,o3,o4,o5,o6)

fac=1;
switch flag

case 0
str =[];
ts =[0 0];
s = simsizes;
s.NumContStates  = 6;
s.NumDiscStates  = 0;
s.NumOutputs     = 15;
s.NumInputs      = 3;
s.DirFeedthrough = 0;
s.NumSampleTimes = 1;

sys = simsizes(s);

y0 = [o1,o2,o3,o4,o5,o6];

case 1

T=411.15 ;
Td = T - 273.15 ; % C
msol=fac*0.215;
f= 5.1548e-01;

% reactivity ratios
r1 = 0.42 ;
r2 = 0.61;
```

```
% initiation coefficient < s^-1 >
kd = 6.78e+15*exp(-1.7714e+04/T) ;


% propagation rates < l/(mol.s) >
kpaa = 3.802e+06*exp(-2.754e+03/T);
kpbb = 4.266e+07*exp(-3.910e+03/T);
kpab = kpaa/r1
kpba = kpbb/r2 ;

% termination coefficients < l/(mol.s) >
kterm_aa = 7.10e+07*exp(-8.30e+02/T);
kterm_bb = 3.818e+09*exp(-9.58e+02/T);

% transfer to monomer < l/(mol.s) >
ktrm_aa = 1.56e+02*exp(-2.621e+03/T) ;
ktrm_bb = 2.31e+06*exp(-6.377e+03/T) ;
ktrm_ba = ktrm_aa*kpbb/(kpaa*r2); % ktrm_aa*kpba/kpaa;
ktrm_ab = ktrm_bb*kpaa/(kpbb*r1); % ktrm_bb*kpab/kpbb;

% transfer to solvent < l/(mol.s) >
Cs_a = 5.55*exp(-4.590e+03/T) ;
Cs_b = 1e-04;

ktrs_a = Cs_a*kpaa ;
ktrs_b = Cs_b*kpbb ;

% molecular weights (Kg/mol)
Mi = 1.3210e-01;
Ma = 1.42e-01;
Mb = 1.0415e-01;
Ms = 1.0617e-01;

% densities (Kg/l)
pA = 9.01e-01 - 8.35e-04*Td;
pB = 9.19e-01 - 6.65e-04*Td;
pI = 8.85e-01 ;
pSol = 8.92e-01 - 1.3e-03*Td ;
pPol = 1.19 - 8.07e-04*Td;


% defining inputs
u1 = u(1);
u2 = u(2);
u3 = u(3);
```

```
% defining state vector
x1p=y(1); % mA(t)
x2p=y(2); % mB(t)
x3p=y(3); % mI(t)
x4p=y(4); % mAPol(t)
x5p=y(5); % mBPol(t)
x6p=y(6); % total mol of polymer

Mr=x1p+x2p+x3p+x4p+x5p+msol;
D = Mr/( x1p/pA + x2p/pB  + x3p/pI + msol/pSol + x4p/pPol + x5p/pPol );
C = sqrt(2*f*kd/Mi) ;

f2 = x1p*Mb / ( x1p*Mb + x2p*Ma+eps ) ;
f3 = x2p*Ma / ( x1p*Mb + x2p*Ma+eps ) ;
frA = kpba*Mb*x1p/( kpba*Mb*x1p + kpab*Ma*x2p+eps );
frB = kpab*Ma*x2p/( kpba*Mb*x1p + kpab*Ma*x2p+eps );

Ktcop = 10^( f2*log10(kterm_aa) + f3*log10(kterm_bb));
Ltot = C*sqrt(D*x3p/( (Ktcop)*Mr )) ;
Mn=(x4p+x5p)/(x6p+eps);

La = Ltot*frA ;
Lb = Ltot*frB ;
V = Mr/D ;

alpha = 0.65 ;
betha = 0.01 ;
gamma = 0.33 ;

ktc_aa = (1- alpha)*Ktcop ;
ktc_bb = (1- betha)*Ktcop ;
ktc_ab = (1- gamma)*Ktcop ;
ktc_ba = ktc_ab ;

Rinit = (C^2)*D*x3p/Mr ;
Rtrm = ( ( ktrm_aa*frA + ktrm_ba*frB )*x1p/(Mr*Ma) + (ktrm_bb*frB
            +  ktrm_ab*frA)*x2p/(Mr*Mb) )*Ltot*D
Rtrs = ( ktrs_a*frA + ktrs_b*frB )*msol*Ltot*D/(Mr*Ms);
Rtc = ktc_aa*La^2 + ktc_bb*Lb^2 + 2*ktc_ab*La*Lb;

Ruo = Rinit + Rtrm + Rtrs -Rtc/2 ;

% RHS system
yprime(1,1) = u1 - x1p*( kpaa*La + kpba*Lb );
yprime(2,1) = u2 - x2p*( kpbb*Lb + kpab*La );
yprime(3,1) = u3 - kd*x3p;
yprime(4,1) = x1p*( kpaa*La + kpba*Lb );
yprime(5,1) = x2p*( kpbb*Lb + kpab*La );
yprime(6,1) = V*Ruo;

sys = [yprime];


case 3

 % defining outputs
x1p=y(1); % mA(t)
x2p=y(2); % mB(t)
x3p=y(3); % mI(t)
x4p=y(4); % mAPol(t)
x5p=y(5); % mBPol(t)
x6p=y(6);

T = 411.15;     % K
Td = T -273.15; % C
pA = 9.01e-01 - 8.35e-04*Td;
pB = 9.19e-01 - 6.65e-04*Td;
pI = 8.85e-01 ;
pSol = 8.92e-01 - 1.3e-03*Td ;
pPol = 1.19 - 8.07e-04*Td;

% termination coefficients < l/(mol.s) >
kterm_aa = 7.10e+07*exp(-830/T);
kterm_bb = 3.818e+09*exp(-958/T);

Mi = 1.3210e-01;
Ma = 1.42e-01;
Mb = 1.0415e-01;
Ms = 1.0617e-01;
msol=fac*0.215;

f2 = x1p*Mb / ( x1p*Mb + x2p*Ma+eps ) ;
f3 = x2p*Ma / ( x1p*Mb + x2p*Ma+eps ) ;
Ktcop = 10^( f2*log10(kterm_aa) + f3*log10(kterm_bb));

% composition
Fnum=x4p/Ma;
Fden=x4p/Ma + x5p/Mb;
F=Fnum/(Fden+eps);

% average molecular weight:
```

```
Mnum=x4p+x5p;
Mden=x6p;
Mn=Mnum/(Mden+eps);

Mr=x1p+x2p+x3p+x4p+x5p+msol;

D = Mr/( x1p/pA + x2p/pB  + x3p/pI + msol/pSol + x4p/pPol + x5p/pPol );
V =(Mr/D);

CI=D*x3p/(Mi*Mr);
Rc=(D/Mr)*(x1p/Ma+x2p/Mb)/((CI/Ktcop)^0.5+eps);

C_BMA_reactor = x1p/(V*Ma);
C_Sty_reactor = x2p/(V*Mb);
C_I_reactor = x3p/(V*Mi);
PolymerMass = x4p+x5p;
MonomerFed = x1p+x2p+x4p+x5p;
xc=PolymerMass/(MonomerFed+eps); %instantaneous conversion

mAPOL = x4p;   % mA,POL(t)
mA = x1p;    % mA(t)
mBPOL = x5p; % mB,POL(t);
mB = x2p;    % mB(t)
mI = x3p;    % mI(t)
molpol = x6p;

CBMA = x1p/(V*Ma);
CSty = x2p/(V*Mb);
CI = x3p/(V*Mi);
xc_abs=PolymerMass/(MonomerFed+mI+eps);
rel_M_mpol = PolymerMass/(MonomerFed+mI+msol);

sys = [MonomerFed,F,mAPOL ,mBPOL ,
mA,mB,mI,CI,PolymerMass,Rc,Mn,xc_abs,MassR,molpol,V]';

case{2,4,9}

   sys = [];

otherwise
  error('unhandled flag')
end
```

A.2 Simulink Layout



Appendix B: Optimization Routines

B.1 Matlab Optimization Routine

```
function[Clock,tout,xs,vec,tm,X,Mn,F,Mnsim,Fsim,U1,U2,U3] =ReactorRunSF(x0);

mex odef.f90  ODESSA.f
tref=21600;
ni=1000 ; %  120 <=ni <= 1000 works fine ! , analysis done for CASE 2

%% CASE 1

% M_sp=9.96;
% F_sp=4.23e-01;
% mpol=0.474;
% theta0_u1 = [0.246/tref]';
% theta0_u2 = [1.0]';
% theta0_u3 = [0.0391]';
% theta0_tf = [21600/ni]';
% kk=diag([1e-12,1e-05,1e-05,1e-04]);
% M=2e+04;
% mu1=1e-11;
% epsilon1=1e-12;
% epsilon2=1e-12;
% epsilon3=1e-12;
% x1b=[0 0.01];      % mBMA(t)
% x2b=[0 0.01];      % mSty(t)
% x3b=[0 5e-04];


%%% CASE 2 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
M_sp=9.96;
F_sp=4.23e-01;
mpol=0.474;
theta0_u1 = [1.70e-01/tref]';
theta0_u2 = [1.0]';
theta0_u3 = [0.074]';
theta0_tf = [21600/ni]';
kk=diag([1e-12,1e-05,1e-05,35]);  % k3=0.2e-05 and k3=1.5e-05 with k4=1e-01
M=4e+04;
```

```
mu1=1e-11;
epsilon1=1e-12;
epsilon2=1e-12;
epsilon3=1e-12;
x1b=[0 0.01];     % mBMA(t)
x2b=[0 0.01];     % mSty(t)
x3b=[0 5e-04];

%%%% CASE 3 %%%%%%%%%%%%%%%%%%%%%%%%%%%

% M_sp=8;
% F_sp=4.23e-01;
% mpol=0.474;
% theta0_u1 = [2.40e-01/tref]';
% theta0_u2 = [1.0]';
% theta0_u3 = [0.086]';
% theta0_tf = [21600/ni]';
% kk=diag([1e-12,1e-05,1e-05,1.9]);
% M=6000;
% mu1=1e-11;
% epsilon1=1e-12;
% epsilon2=1e-12;
% epsilon3=1e-12;
% x1b=[0 0.01];     % mBMA(t)
% x2b=[0 0.01];     % mSty(t)
% x3b=[0 5e-04];


% CASE 4

% M_sp=7.98;
% F_sp=6.85e-01;
% mpol=4.83e-001; % 75 25
% theta0_u1 = [5.5e-01/tref]';
% theta0_u2 = [0.24]';
% theta0_u3 = [0.069]';
% theta0_tf = [21600/ni]';
% kk=diag([1e-14,1e-03,1e-05,0.6]);
% M=9e+04;
% mu1=1e-11;
% epsilon1=1e-12;
% epsilon2=1e-12;
% epsilon3=1e-12;
% x1b=[0 0.01];     % mBMA(t)
% x2b=[0 0.01];     % mSty(t)
% x3b=[0 5e-04];
```

```
Tem=411.15e+00; % K
r1 = 0.42e+00;
r2 = 0.61e+00;
f=0.515e+00;
Mi = 1.3210e-01;
Ma = 1.42e-01;
Mb = 1.0415e-01;
Ms = 1.0617e-01;
alpha = 0.65e+00;
betha = 0.01e+00;
gamma = 0.33e+00;
    facc=1.0e+00;
msol = facc*0.215;
EEPS=eps;


per=0.99;
Vf=1;


% input bounds
u1b =[0 5];     % feed flow rate BMA
u2b =[0 5];     % feed flow rate Sty
u3b =[0 5];     % feed flow rate Init.

% weighting factors
wff1=1/1; %1/1.5;
wff2=1/1; %1/1.5;
wff3=1/21600; %1/10600;

x1ub=x1b(2);
x2ub=x2b(2);
x3ub=x3b(2);
u1ub=u1b(2);
u2ub=u2b(2);
u3ub=u3b(2);

vec=[M_sp,F_sp,Tem,msol,wff1,wff2,x1ub,x2ub,x3ub,u1ub,u2ub,u3ub,ni,mpol,per,Vf,
mu1,epsilon1,epsilon2,epsilon3]; % for passing values to odef.f90

NS=7;  number of states (original states + extra state due to cost function)

NPAR_u1=1;
NPAR_u2=1;
```

```
NPAR_u3=1;
NPAR_tfpar=1;
N=NPAR_u1 + NPAR_u2 + NPAR_u3+NPAR_tfpar;  % total number of parameters

x10 = x0(1); % mBMA
x20 = x0(2); % mSty
x30 = x0(3); % mI
x40 = x0(4); % mAPol
x50 = x0(5); % mBPol
x60 = x0(6); % mol pol
x70 = 0;     % additional state cost related

% vector preallocation
thetadot_u1=zeros(NPAR_u1,1);
thetadot_u2=zeros(NPAR_u2,1);
thetadot_u3=zeros(NPAR_u3,1);
thetadot_tf=zeros(1,1);
temp=zeros(N,1);

Clock1=clock;

nn=ni+1;
JJ=zeros(nn,1);
TJ=zeros(nn,1);
XDIF=zeros(nn,NS);

tau0=0;
X=x0;
Xo=X;
X1T=[x0,0];
T0=tau0;
To=tau0;
tm=0;


Td = Tem - 273.15 ;
pSol = 8.92e-01 - 1.3e-03*Td ;

V0=msol/pSol;
Vff=per*Vf;

TC=x70+ M*((x40+x50-mpol)^2)+wff3*theta0_tf*ni;
IC=x70;
ICmat=x70;

Grad_Jic=zeros(nn,1);
```

```
tt=tau0;

% data storage for plotting
vec_iter = zeros(nn+1,1);
viter=zeros(nn,1);
vdtheta_u1=zeros(nn,NPAR_u1);
vdtheta_u2=zeros(nn,NPAR_u2);
vdtheta_u3=zeros(nn,NPAR_u3);
vdtheta_tf = zeros(nn,1);
GradT_vec = zeros(nn,N);


% piece wise constant - not a function of tt
U10=theta0_u1(1,1);
U20=theta0_u2(1,1)*theta0_u1(1,1);
U30=theta0_u3(1,1)*theta0_u1(1,1);


if U10 <=0
U10 =0;
elseif U10 >=u1b(2)
U10 =u1b(2);
end

if U20 <=0
U20=0;
elseif U20>=u2b(2)
U20=u2b(2);
end

if U30 <=0
U30=0;
elseif U30 >=u3b(2)
U30=u3b(2);
end

U1=U10;
U2=U20;
U3=U30;

Th_u1=[theta0_u1'];
Th_u2=[theta0_u2'];
Th_u3=[theta0_u3'];
Th_tf=[theta0_tf'];

summ=0;
```

```
tic;

T=1;
options=odeset('RelTol',1.0e-6,'AbsTol',1.0e-8);

epsilons=[epsilon1 epsilon2 epsilon3]; % back-off

p1=vec_paramSF(wff1,wff2,wff3,u1b,u2b,u3b,x1b,x2b,x3b,mu1,epsilons,M_sp,F_sp,Te
m,msol,ni,mpol,per,Vf);
for ij=1:nn
vec_iter(ij+1,1)=ij;
viter(ij,1)=ij; % vector of iterations
cc1=cputime;
delta_tau=1*(1/ni);

if ij==1
y=[x10,x20,x30,x40,x50,x60,x70];
taus=tau0; %=0
tauf=tau0;
else
tauf= tau0 + 1/ni ;
[taus,y]=ode15s('ReactorModelSF',[tau0,tauf],[x10,x20,x30,x40,x50,x60,x70],options,N,
T,theta0_u1,theta0_u2,theta0_u3 ,theta0_tf,p1);
end

cc2=cputime-cc1;
summ=summ+cc2;
m=size(y,1);
xx=y(:,1:end-1);
Costm=y(:,end);
xobs=xx(m,:);
cost(ij,1)=Costm(end);

for irk=1:1
if (irk<4&irk>1)
thetam_u1=theta0_u1'+M1_u1(irk-1,:)*(delta_tau)/2;
thetam_u2=theta0_u2'+M1_u2(irk-1,:)*(delta_tau)/2;
thetam_u3=theta0_u3'+M1_u3(irk-1,:)*(delta_tau)/2;
elseif irk==1
thetam_u1=theta0_u1';
thetam_u2=theta0_u2';
thetam_u3=theta0_u3';
elseif irk==4
thetam_u1=theta0_u1'+M1_u1(irk-1,:)*(delta_tau);
thetam_u2=theta0_u2'+M1_u2(irk-1,:)*(delta_tau);
thetam_u3=theta0_u3'+M1_u3(irk-1,:)*(delta_tau);
```

```
end

 thetam_rowvec = [thetam_u1 , thetam_u2 , thetam_u3 ,theta0_tf,vec];
ytemp = odef(tauf,1,thetam_rowvec,[xobs Costm(m,1)]);

itemp=1;
for i=1:NS
for j=1:N+1
yt(i,j)=ytemp(1,itemp);
itemp=itemp+1;
end
end

x1t=yt(1,1);  % x1(t)
x2t=yt(2,1);  % x2(t)
x3t=yt(3,1);  % x3(t)
x4t=yt(4,1);  % x4(t)
x5t=yt(5,1);  % x5(t)
x6t=yt(6,1);  % x6(t)
x7t=yt(7,1);  % x7(t)


x1c=yt(1,2:end);
x2c=yt(2,2:end);
x3c=yt(3,2:end);
x4c=yt(4,2:end);
x5c=yt(5,2:end);
x6c=yt(6,2:end);
x7c=yt(7,2:end);
Jic=yt(end,1); % x7c
JJ(ij,1)=Jic;
TJ(ij,1)=tauf;

Td = Tem - 273.15 ;
pA = 9.01e-01 - 8.35e-04*Td;
pB = 9.19e-01 - 6.65e-04*Td;
pI = 8.85e-01 ;
pSol = 8.92e-01 - 1.3e-03*Td ;
pPol = 1.19 - 8.07e-04*Td;

V =  x1t/pA + x2t/pB  + x3t/pI + msol/pSol + x4t/pPol + x5t/pPol ;


termc = M*((x4t+x5t-mpol)^2) ;              % terminal equality constraint
Jip = Jic + termc + wff3*theta0_tf*ni;       % modified cost
Grad_termc =2*M*((x4t+x5t-mpol)*(x4c+x5c));
```

```
Grad_Jic = x7c;
gradtheta0_tf_dp=[zeros(1,N-1) , 1];
Grad_Jip = Grad_Jic' + Grad_termc' + wff3*ni*gradtheta0_tf_dp' ;

if norm(Grad_Jip)==0   % euclidean norm of dJip/dp
temp=zeros(N,1);       % temp = - factor*dJip/dp
else
temp(1,1)=-kk(1,1)*Grad_Jip(1,1);
temp(2,1)=-kk(2,2)*Grad_Jip(2,1);
temp(3,1)=-kk(3,3)*Grad_Jip(3,1);
temp(4,1)=-kk(4,4)*Grad_Jip(4,1);
end

low_u1=[1e-06*ones(1,NPAR_u1)];
low_u2=[1e-06*ones(1,NPAR_u2)];
low_u3=[1e-06*ones(1,NPAR_u3)];
low_tf=[1e-06];

high_u1=[0.5*ones(1,NPAR_u1)];
high_u2=[3*ones(1,NPAR_u2)];
high_u3=[0.5*ones(1,NPAR_u3)];
high_tf=[18];

delta=1e-05;
deltatf=1e-05;
delta_tau1=1*delta_tau;
delta_tau2=1*delta_tau;
delta_tau3=1*delta_tau;
delta_tau_tf=1*delta_tau;

% %% for U1---theta 1
for i=1:NPAR_u1
if theta0_u1(i,1) + delta_tau1*temp(i,1) >= low_u1(i)
if theta0_u1(i,1) + delta_tau1*temp(i,1) <= high_u1(i)
thetadot_u1(i,1) = temp(i,1);
end
end

if theta0_u1(i,1)+delta_tau1*temp(i,1) < low_u1(i)
thetadot_u1(i,1)=0
% if temp(i,1)<0
% thetadot_u1(i,1)=(1+(theta0_u1(i,1)-low_u1(i))/delta)*temp(i,1);
% else
% thetadot_u1(i,1)=temp(i,1);
% end
end
```

```
if theta0_u1(i,1)+delta_tau1*temp(i,1) > high_u1(i)
if temp(i,1)>0
% thetadot_u1(i,1)=0;
thetadot_u1(i,1)=(1+(high_u1(i)-theta0_u1(i,1))/delta)*temp(i,1);
else
mark_u1_4=1;
thetadot_u1(i,1)=temp(i,1);
end
end
end

%%%  for U2 --- theta 2
for i=1:NPAR_u2
if theta0_u2(i,1) + delta_tau2*temp(i+NPAR_u1,1) >= low_u2(i)
if theta0_u2(i,1) + delta_tau2*temp(i+NPAR_u1,1) <= high_u2(i)
thetadot_u2(i,1) = temp(i+NPAR_u1,1);
end
end

if theta0_u2(i,1)+delta_tau2*temp(i+NPAR_u1,1) < low_u2(i)
thetadot_u2(i,1)=0;
% if temp(i+NPAR_u1,1)<0
% thetadot_u2(i,1)=(1+(theta0_u2(i,1)-low_u2(i))/delta)*temp(i+NPAR_u1,1);
% else
% thetadot_u2(i,1)=temp(i+NPAR_u1,1);
% end
end

if theta0_u2(i,1)+delta_tau2*temp(i+NPAR_u1,1) > high_u2(i)
if temp(i+NPAR_u1,1)>0
%  thetadot_u2(i,1)=0;
thetadot_u2(i,1)=(1+(high_u2(i)-theta0_u2(i,1))/delta)*temp(i+NPAR_u1,1);
else
thetadot_u2(i,1)=temp(i+NPAR_u1,1);
end
end
end

% for U3 --- theta 3
for i=1:NPAR_u3
if theta0_u3(i,1) + delta_tau3*temp(i+NPAR_u1+NPAR_u2,1) >= low_u3(i)
if theta0_u3(i,1) + delta_tau3*temp(i+NPAR_u1+NPAR_u2,1) <= high_u3(i)
thetadot_u3(i,1) = temp(i+NPAR_u1+NPAR_u2,1);
end
end
```

```
if theta0_u3(i,1)+delta_tau3*temp(i+NPAR_u1+NPAR_u2,1) < low_u3(i)
thetadot_u3(i,1)=0;
% if temp(i+NPAR_u1+NPAR_u2,1)<0
% thetadot_u3(i,1)=(1+(theta0_u3(i,1)-
low_u3(i))/delta)*temp(i+NPAR_u1+NPAR_u2,1)
% else
% thetadot_u3(i,1)=temp(i+NPAR_u1+NPAR_u2,1);
% end
end

if theta0_u3(i,1)+delta_tau3*temp(i+NPAR_u1+NPAR_u2,1) > high_u3(i)
if temp(i+NPAR_u1+NPAR_u2,1)>0
%  thetadot_u3(i,1)=0;
thetadot_u3(i,1)=(1+(high_u3(i)-theta0_u3(i,1))/delta)*temp(i+NPAR_u1+NPAR_u2,1);
else
thetadot_u3(i,1)=temp(i+NPAR_u1+NPAR_u2,1);
end
end
end

%theta_tf (4th parameter)

if theta0_tf(1,1) + delta_tau_tf*temp(end,1) >= low_tf % =10
if theta0_tf(1,1) + delta_tau_tf*temp(end,1)<= high_tf %=20
thetadot_tf(1,1) = temp(end,1);
end
end

if theta0_tf(1,1)+delta_tau_tf*temp(end,1) < low_tf
thetadot_tf(1,1)=0;
%  if temp(end,1)<0
%  thetadot_tf(1,1)=(1+(theta0_tf(1,1)-low_tf)/deltatf)*temp(end,1);
%  else
%  thetadot_tf(1,1)=temp(end,1);
%  end
end

if theta0_tf(1,1)+delta_tau_tf*temp(end,1) > high_tf
if temp(end,1)>0
%  thetadot_tf(1,1)=0;
thetadot_tf(1,1)=(1+(high_tf-theta0_tf(1,1))/deltatf)*temp(end,1);
else
thetadot_tf(1,1)=temp(end,1);
end
end
```

```
M1_u1(irk,:)=thetadot_u1';
M1_u2(irk,:)=thetadot_u2';
M1_u3(irk,:)=thetadot_u3';

end;

% else,
% M1=zeros(1,N);
% end;
% theta1=theta0+deltaT*(M1(1,:)'/6+M1(2,:)'/3+M1(3,:)'/3+M1(4,:)'/6);

theta1_u1 = theta0_u1 + delta_tau1*M1_u1(1,:)';
theta1_u2 = theta0_u2 + delta_tau2*M1_u2(1,:)';
theta1_u3 = theta0_u3 + delta_tau3*M1_u3(1,:)';
theta1_tf = theta0_tf + delta_tau_tf*thetadot_tf';

vdtheta_u1(i,:) = delta_tau1*M1_u1(1,:);
vdtheta_u2(i,:) = delta_tau2*M1_u2(1,:);
vdtheta_u3(i,1) = delta_tau3*M1_u3(1,:);
vdtheta_tf(i,1) =  delta_tau_tf*thetadot_tf;

T0=[T0;taus];
To=[To;tauf];
tm=[tm;taus*theta1_tf*ni];
Th_u1=[Th_u1 ; theta1_u1'];
Th_u2=[Th_u2 ; theta1_u2'];
Th_u3=[Th_u3 ; theta1_u3'];
Th_tf=[Th_tf ; theta1_tf'];

sv=size(xx,1);
xx1=xx(:,1);
xx1v=zeros(sv,1);
xx2=xx(:,2);
xx2v=zeros(sv,1);
xx3=xx(:,3);
xx3v=zeros(sv,1);
xx4=xx(:,4);
xx4v=zeros(sv,1);
xx5=xx(:,5);
xx5v=zeros(sv,1);
xx6=xx(:,6);
xx6v=zeros(sv,1);

for w=1:sv
if xx1(w)<=x1b(1)
```

```
xx1v(w)=x1b(1);
elseif xx1(w)>=x1b(2)
xx1v(w)=x1b(2);
else
xx1v(w)=xx1(w);
end

if xx2(w)<=x2b(1)
xx2v(w)=x2b(1);
elseif xx2(w)>=x2b(2)
xx2v(w)=x2b(2);
else
xx2v(w)=xx2(w);
end

if xx3(w)<=x3b(1)
xx3v(w)=x3b(1);
elseif xx3(w)>=x3b(2)
xx3v(w)=x3b(2);
else
xx3v(w)=xx3(w);
end

if xx4(w)<=0
xx4v(w)=0;
else
xx4v(w)=xx4(w);
end

if xx5(w)<=0
xx5v(w)=0;
else
xx5v(w)=xx5(w);
end

if xx6(w)<=0
xx6v(w)=0;
else
xx6v(w)=xx6(w);
end
end
xx=[xx1v , xx2v , xx3v , xx4v , xx5v , xx6v];

X=[X;xx];
Xo=[Xo;xobs];
IC=[IC;Jic];  % x7 integral cost
```

```
TC=[TC;Jip];

theta0_u1 = theta1_u1;
theta0_u2 = theta1_u2;
theta0_u3 = theta1_u3;
theta0_tf = theta1_tf;

x10=y(m,1);
x20=y(m,2);
x30=y(m,3);
x40=y(m,4);
x50=y(m,5);
x60=y(m,6);
x70=y(m,7); %cost function
ICmat=[ICmat;x70];
tau0=tauf;

for ii=1:m
tto=taus(ii);
U1p=theta1_u1(1,1);
U2p=theta1_u2(1,1)*theta1_u1(1,1);
U3p=theta1_u3(1,1)*theta1_u1(1,1) ;

if U1p <=0
U1p =0;
elseif U1p >=u1b(2)
U1p =u1b(2);
end

if U2p <=0
U2p=0;
elseif U2p>=u2b(2)
U2p=u2b(2);
end

if U3p <=0
U3p=0;
elseif U3p >=u3b(2)
U3p=u3b(2);
end

U1=[U1;U1p];
U2=[U2;U2p];
U3=[U3;U3p];

end
```

```
%   Clock(ij)=cputime-Clock1;
%  [theta0_u1,theta0_u2,theta0_u3]


tf=Th_tf(end)*ni;

mreactor=x10+x20+x30+x40+x50+msol;
Vcontent= x10/pA + x20/pB  + x30/pI + msol/pSol + x40/pPol + x50/pPol  ;
Mw=(x40+x50)/sqrt(x60^2+eps);
Fp=(x40/Ma)/(sqrt(x40^2+eps)/Ma + sqrt(x50^2+eps)/Mb);

[mreactor,x10,x20,x30,x40+x50,V,Mw,Fp] ; % display

clear U1p
clear U2p
clear U3p

end % close MAIN LOOP

Clock=(toc-summ)/nn;
Vt=tm(2:end)-tm(1:end-1);
u1v=U1(1:end-1);
u2v=U2(1:end-1);
u3v=U3(1:end-1);
I=u3v'*Vt
BMA_fed=u1v'*Vt
STY_fed=u2v'*Vt

redx_init_fed=(1- I/9.70e-03)*100
difference_init_fed_GR=(9.70e-03-I)*1e+03
redx_batchtime=(21600-tf)/60
Cost=Jip(end)
Pol_produced =X(:,4)+X(:,5);
MPOL_optm=Pol_produced(end)

return
```

The subroutines: vec_paramSF, open_vec_paramSF, and ReactorModelSF have been omitted for space sake .The first two subroutines are intended to pass and receive constant values from one script to another; open_vec_paramSF is located inside ReactorModelSF subroutine. The latter is the process model used by the optimization algorithm. The same model; however, is also written in B.2 section in Fortran.

B.2 Fortran Optimization Routine

```
      SUBROUTINE odef(T,STEPS,PAR,Y0,YOUT)
      DOUBLE PRECISION ATOL,RWORK,RTOL,T,TOUT,Y,PAR,YOUT,STEPS,Y0
      EXTERNAL FEX, JEX, DFEX
      DIMENSION YOUT(1,35),Y0(7,1),Y(7,5),PAR(24),RTOL(7,5),ATOL(7,5)
      DIMENSION RWORK(358),IWORK(32),NEQ(2),IOPT(3)
      N = 7
      NPAR = 4
          NSTEP=STEPS
      NEQ(1) = N
      NEQ(2) = NPAR
      NSV = NPAR+1
      DO 10 I = 1,N
      DO 10 J = 1,NSV
 10     Y(I,J) = 0.0D0
      Y(1,1) = Y0(1,1)
          Y(2,1) = Y0(2,1)
          Y(3,1) = Y0(3,1)
          Y(4,1) = Y0(4,1)
          Y(5,1) = Y0(5,1)
          Y(6,1) = Y0(6,1)
          Y(7,1) = Y0(7,1)
      TOUT = T+ 0.001D0
      ITOL = 4
      ATOL(1,1) = 1.D-08
      ATOL(2,1) = 1.D-08
          ATOL(3,1) = 1.D-08
          ATOL(4,1) = 1.D-08
          ATOL(5,1) = 1.D-08
          ATOL(6,1) = 1.D-08
          ATOL(7,1) = 1.D-08

      DO 20 I = 1,N
      RTOL(I,1) = 1.D-06
      DO 15 J = 2,NSV
      RTOL(I,J) = 1.0D+00*RTOL(I,1)
15    ATOL(I,J) = 1.0D+00*ATOL(I,1)
20    CONTINUE
      ITASK = 1
      ISTATE = 1
      IOPT(1) = 0
      IOPT(2) = 1  ! 1=sensitivity analysis , 0 otherwise
      IOPT(3) = 1  ! 1=DFDP supplied by user, 0 otherwise
      LRW = 358
      LIW = 32
      MF = 21  ! MF=21 DFDY full jacobian supplied by user , 22 otherwise
          TOUT = 1
      DO 60 IOUT = 1,NSTEP
      CALL ODESSA(FEX,DFEX,NEQ,Y,PAR,T,TOUT,ITOL,RTOL,ATOL,ITASK, &
                         ISTATE,IOPT,RWORK,LRW,IWORK,LIW,JEX,MF)
          ITEMP=1
          DO 40 I = 1,N
      DO 30 J = 1,NSV
          YOUT(IOUT,ITEMP)=Y(I,J)
 30   ITEMP=ITEMP+1
 40   CONTINUE

 60   TOUT = TOUT+ 0.001D0
     RETURN
     END

     SUBROUTINE FEX (NEQ, T, Y, PAR,YDOT)
     DOUBLE PRECISION T,Y,YDOT,PAR,U1,U2,U3 ,Td,mfed,Mmax
     DIMENSION Y(7), YDOT(7), PAR(24)
     DOUBLE PRECISION W1,W_1,W2,W_2,W3,W_3,W4,W5,W6,W7,W8,W9,W10
     DOUBLE PRECISION W12,W13,W14,W15,W16,W17,W18,W19,W20
     DOUBLE PRECISION W11,W_11,W21,W_21,W31,W_31,W41,W51,W61,W71
     DOUBLE PRECISION W81,W91,W101,W121,W131,W141,W151
     DOUBLE PRECISION W161,W171,W181,W191,W201
     DOUBLE PRECISION kd,kpaa,kpbb,kpab,kpba,ktrm_aa,ktrm_bb,ktrm_ab
     DOUBLE PRECISION ktrm_ba,Cs_a,Cs_b,ktrs_a,ktrs_b
     DOUBLE PRECISION ktermaa,ktermbb,m1,m2,s,ktc_ba,Rinit,Rtrm,Rtrs
     DOUBLE PRECISION Rtc,Ruo,Ltot,La,Lb,V,ktc_aa,ktc_bb,ktc_ab
     DOUBLE PRECISION pA,pB,pI,pSol,pPol,f2,f3,frA,frB,DE,SS,CC,Ktcop
     DOUBLE PRECISION Fcnum,Fcden,Fc,Mnum,Mden,Mn,Msp,Fsp,Tem,r1,r2,f
     DOUBLE PRECISION alpha,betha,gamma,Mi,Ma,Mb,Ms,frac
     DOUBLE PRECISION mu1,epsilon1,epsilon2,wff1,wff2,wff3,eff,mPOL
     DOUBLE PRECISION y1upsf,y2upsf,y3upsf,u1ins,u2ins,u3ins,Vf,Vff
     DOUBLE PRECISION x1p,x2p,x3p,x4p,x5p,x6p,x7p,Mr ,facc,prodc,msol
     INTEGER ni

     r1=1.0*0.42D0
     r2=1.0*0.61D0
     f=5.1548D-01
     alpha=0.65D0
     betha= 0.01D0
     gamma= 0.33D0
     Mi= 1.3210D-01
```

```
Ma= 1.42D-01;
Mb= 1.0415D-01
Ms= 1.0617D-01
eps= 2.2204D-016

Msp=PAR(5)
Fsp=PAR(6)
Tem=PAR(7)
msol=PAR(8)
wff1= PAR(9)
wff2= PAR(10)
y1upsf=PAR(11)
y2upsf=PAR(12)
y3upsf=PAR(13)
u1ins=PAR(14)
u2ins=PAR(15)
u3ins=PAR(16)
ni=PAR(17)
mPOL=PAR(18)
 frac=PAR(19)
 Vf=PAR(20)
 mu1=PAR(21)
 epsilon1=PAR(22)
 epsilon2=PAR(23)
 epsilon3=PAR(24)

U1=PAR(1);
U2=PAR(2)*PAR(1)
U3=PAR(3)*PAR(1)
IF ((U1-u1ins).GE.0) U1=u1ins
IF (U1.LE.0)  U1=0
IF ((U2-u2ins).GE.0) U2=u2ins
IF (U2.LE.0)  U2=0
IF ((U3-u3ins).GE.0) U3=u3ins
IF (U3.LE.0)  U3=0

x1p=(Y(1)**2+eps)**0.5; !mBMA
x2p=(Y(2)**2+eps)**0.5; !mSty
x3p=(Y(3)**2+eps)**0.5; !mI
x4p=(Y(4)**2+eps)**0.5; !mAPol
x5p=(Y(5)**2+eps)**0.5; !mBPol
x6p=(Y(6)**2+eps)**0.5; !mol pol
x7p=(Y(7)**2+eps)**0.5; !cost

 Mr=x1p+x2p+x3p+x4p+x5p+msol;
```

```
W1 = -x1p + y1upsf +epsilon1;
W11= -x1p + y1upsf;
W_1 = x1p + epsilon1;
W_11= x1p
IF((x1p-y1upsf).GE.0) W1=epsilon1
IF((x1p-y1upsf).GE.0) W11=0
IF(x1p.LE.0) W_1=epsilon1
IF(x1p.LE.0) W_11=0

W2 = -x2p + y2upsf +epsilon2;
W21= -x2p + y2upsf
W_2 = x2p + epsilon2;
W_21= x2p
IF((x2p-y2upsf).GE.0) W2=epsilon2
IF((x2p-y2upsf).GE.0) W21=0
IF(x2p.LE.0) W_2=epsilon2
IF(x2p.LE.0) W_21=0

W3 = -x3p + y3upsf +epsilon3;
W31= -x3p + y3upsf
W_3 = x3p + epsilon3;
W_31= x3p
IF((x3p-y3upsf).GE.0) W3=epsilon3
IF((x3p-y3upsf).GE.0) W31=0
IF(x3p.LE.0) W_3=epsilon3
IF(x3p.LE.0) W_31=0

Td = Tem-273.15D0

pA = 9.01D-01 - 8.35D-04*Td;
pB = 9.19D-01 - 6.65D-04*Td;
pI = 8.85D-01 ;
pSol = 8.92D-01 - 1.3D-03*Td ;
pPol = 1.19D0 - 8.07D-04*Td;

V=x1p/pA+x2p/pB+x3p/pI+msol/pSol+x4p/pPol+x5p/pPol

W6 = x4p + epsilon1;
W61= x4p
IF(x4p.LE.0) W6=epsilon1
IF(x4p.LE.0) W61=0

W7 = x5p + epsilon1;
W71= x5p
IF(x5p.LE.0) W7=epsilon1
IF(x5p.LE.0) W71=0
```

```
W8 = x6p + epsilon1;
W81= x6p
IF(x6p.LE.0) W8=epsilon1
IF(x6p.LE.0) W81=0

! U1
W13 = U1 + epsilon1;
W131= U1;
W14 = -U1 + u1ins +epsilon1;
W141= -U1 + u1ins
IF (U1.LE.0) W13=epsilon1
IF (U1.LE.0) W131=0
IF ((U1-u1ins).GE.0) W14=epsilon1
IF ((U1-u1ins).GE.0) W141=0

! U2
W15 = U2 + epsilon2
W151= U2;
W16 = -U2 + u2ins +epsilon2
W161= -U2 + u2ins
IF(U2.LE.0) W15=epsilon2
IF(U2.LE.0) W151=0
IF((U2-u2ins).GE.0) W16=epsilon2
IF((U2-u2ins).GE.0) W161=0


! U3
W17 = U3 + epsilon3;
W171= U3;
W18 = -U3 + u3ins +epsilon3;
W181= -U3 + u3ins;
IF (U3.LE.0) W17=epsilon3
IF (U3.LE.0) W171=0
IF ((U3-u3ins).GE.0) W18=epsilon3
IF ((U3-u3ins).GE.0) W181=0

!initiation coefficient < s^-1 >
kd = 6.78D+15*DEXP(-17714D0/Tem) ;

!propagation rates < l/(mol.s) >
kpaa = 3.802D+06*DEXP(-2754D0/Tem);
kpbb = 4.266D+07*DEXP(-3910D0/Tem);
kpab = kpaa/r1 ;
kpba = kpbb/r2 ;
```

```
 !termination coefficients < l/(mol.s) >
 ktermaa = 7.10D+07*DEXP(-830.0D0/Tem);
 ktermbb = 3.818D+09*DEXP(-958.0D0/Tem);

 !transfer to monomer < l/(mol.s) >
 ktrm_aa = 1.56D+02*DEXP(-2621.0D0/Tem) ;
 ktrm_bb = 2.31D+06*DEXP(-6377.0D0/Tem) ;
 ktrm_ba = ktrm_aa*kpbb/(kpaa*r2);
 ktrm_ab = ktrm_bb*kpaa/(kpbb*r1);

 !transfer to solvent < l/(mol.s) >
 Cs_a = 5.55D0*DEXP(-4590.0D0/Tem)
 Cs_b = 1D-04;
 ktrs_a = Cs_a*kpaa ;
 ktrs_b = Cs_b*kpbb;

 DE = Mr/(x1p/pA+x2p/pB+x3p/pI+msol/pSol+x4p/pPol+x5p/pPol)
 CC = DSQRT(2.0D0*f*kd/Mi);
 f2 = x1p*Mb/(x1p*Mb + x2p*Ma);
 f3 = x2p*Ma/(x1p*Mb + x2p*Ma);
 frA = kpba*Mb*x1p/( kpba*Mb*x1p + kpab*Ma*x2p);
 frB = kpab*Ma*x2p/( kpba*Mb*x1p + kpab*Ma*x2p);
 Ktcop=10**((f2*DLOG(ktermaa)+f3*DLOG(ktermbb))/DLOG(10.0D0));
 Ltot = CC*DSQRT(DE*x3p/(Ktcop*Mr))

 La = Ltot*frA
Lb = Ltot*frB
V = Mr/DE

ktc_aa = (1.0D0-alpha)*Ktcop
ktc_bb = (1.0D0-betha)*Ktcop
ktc_ab = (1.0D0-gamma)*Ktcop
ktc_ba = ktc_ab

Rinit = (CC**2)*DE*x3p/Mr
m1 = (ktrm_aa*frA+ktrm_ba*frB)/Ma
m2 = (ktrm_bb*frB+ktrm_ab*frA)/Mb
Rtrm = (m1*(x1p/Mr) + m2*(x2p/Mr))*Ltot*DE
s = (ktrs_a*frA + ktrs_b*frB)*msol/Ms
Rtrs = s*Ltot*DE/Mr
Rtc=ktc_aa*((La)**2)+ ktc_bb*((Lb)**2)+2*ktc_ab*La*Lb

Ruo = Rinit+ Rtrm + Rtrs - 0.5D0*Rtc

Fcnum=x4p/Ma;
Fcden=x4p/Ma + x5p/Mb;
```

```
Fc=Fcnum/(Fcden);

Mnum=x4p+x5p;
Mden=x6p;
Mn=Mnum/(Mden);


prodc=W1*W_1*W2*W_2*W3*W_3*W6*W7*W8*W13*W14*W15*W16*W17*
W18

YDOT(1) = PAR(4)*ni*(U1 - x1p*( kpaa*La + kpba*Lb ))
YDOT(2) = PAR(4)*ni*(U2 - x2p*( kpbb*Lb + kpab*La ))
YDOT(3) = PAR(4)*ni*(U3 - kd*x3p)
YDOT(4) = PAR(4)*ni*(x1p*( kpaa*La + kpba*Lb ))
YDOT(5) = PAR(4)*ni*(x2p*( kpbb*Lb + kpab*La ))
YDOT(6) = PAR(4)*ni*(V*Ruo)
YDOT(7) = PAR(4)*ni*(wff1*(((Mn/Msp)-1)**2) + wff2*(((Fc/Fsp)-1)**2)) &
                    - mu1*DLOG(prodc)
RETURN
END


SUBROUTINE DFEX (NEQ, T, Y, PAR,DFDP, JPAR)
DOUBLE PRECISION T,Y,PAR,DFDP,U1,U2,U3,prodc,mfed,Mmax,Td
DOUBLE PRECISION DU1_DP1,DU1_DP2,DU1_DP3,DU1_DP4,DU1_DP5
DOUBLE PRECISION DU1_DP6,DU2_DP1,DU2_DP2,DU2_DP3,DU2_DP4
DOUBLE PRECISION DU2_DP5,DU2_DP6,DU3_DP1,DU3_DP2,DU3_DP3
DOUBLE PRECISION DU3_DP4,DU3_DP5,DU3_DP6
DOUBLE PRECISION W1,W_1,W2,W_2,W3,W_3,W4,W5,W6,W7,W8,W9,W10
DOUBLE PRECISION W12,W13,W14,W15,W16,W17,W18,W19,W20
DOUBLE PRECISION W11,W_11,W21,W_21,W31,W_31,W41,W51,W61,W71
DOUBLE PRECISION W81,W91,W101,W121,W131,W141,W151
DOUBLE PRECISION W161,W171,W181,W191,W201
DIMENSION Y(7), PAR(24), DFDP(7)
DOUBLE PRECISION kd,kpaa,kpbb,kpab,kpba,ktrm_aa,ktrm_bb
DOUBLE PRECISION ktrm_ab,ktrm_ba,Cs_a,Cs_b,ktrs_a,ktrs_b
DOUBLE PRECISION ktermaa,ktermbb,m1,m2,s,frac,prodc_u
DOUBLE PRECISION pA,pB,pI,pSol,pPol,f2,f3,frA,frB,DE,SS,CC
DOUBLE PRECISION Ktcop,Ltot,La,Lb,V,ktc_aa,ktc_bb,ktc_ab
DOUBLE PRECISION ktc_ba,Rinit,Rtrm,Rtrs,Rtc,Ruo,cte_s
DOUBLE PRECISION Fcnum,Fcden,Fc,Mnum,Mden,Mn,Msp,Fsp,Tem
DOUBLE PRECISION r1,r2,f,alpha,betha,gamma,Mi,Ma,Mb,Ms,msol
DOUBLE PRECISION mu1,epsilon1,epsilon2,wff1,wff2,eff,mPOL
DOUBLE PRECISION y1upsf,y2upsf,y3upsf,u1ins,u2ins,u3ins,Vf,Vff
DOUBLE PRECISION x1p,x2p,x3p,x4p,x5p,x6p,x7p,Mr ,facc,eps
DOUBLE PRECISION dprodcu_dp1,dprodcu_dp2,dprodcu_dp3
```

```
INTEGER ni

r1=1.0*0.42D0
r2=1.0*0.61D0
f=5.1548D-01
alpha=0.65D0
betha= 0.01D0
gamma= 0.33D0
Mi= 1.3210D-01
Ma=  1.42D-01;
Mb=  1.0415D-01
Ms= 1.0617D-01
eps=  2.2204D-016
Fsp=PAR(6)
Tem=PAR(7)
 msol=PAR(8)
 wff1= PAR(9)
 wff2= PAR(10)
 y1upsf=PAR(11)
 y2upsf=PAR(12)
 y3upsf=PAR(13)
 u1ins=PAR(14)
 u2ins=PAR(15)
 u3ins=PAR(16)
 ni=PAR(17)
 mPOL=PAR(18)
 frac=PAR(19)
 Vf=PAR(20)
 mu1=PAR(21)
 epsilon1=PAR(22)
 epsilon2=PAR(23)
 epsilon3=PAR(24)

 U1=PAR(1);
 U2=PAR(2)*PAR(1);
 U3=PAR(3)*PAR(1);

IF ((U1-u1ins).GE.0) U1=u1ins
IF (U1.LE.0)  U1=0
IF ((U2-u2ins).GE.0) U2=u2ins
IF (U2.LE.0)  U2=0
IF ((U3-u3ins).GE.0) U3=u3ins
IF (U3.LE.0)  U3=0

x1p=(Y(1)**2+eps)**0.5;
x2p=(Y(2)**2+eps)**0.5;
```

```
x3p=(Y(3)**2+eps)**0.5;
x4p=(Y(4)**2+eps)**0.5;
x5p=(Y(5)**2+eps)**0.5;
x6p=(Y(6)**2+eps)**0.5;
x7p=(Y(7)**2+eps)**0.5;

Mr=x1p+x2p+x3p+x4p+x5p+msol;

W1 = -x1p + y1upsf +epsilon1;
W11= -x1p + y1upsf;
W_1 = x1p + epsilon1;
W_11= x1p
IF((x1p-y1upsf).GE.0) W1=epsilon1
IF((x1p-y1upsf).GE.0) W11=0
IF(x1p.LE.0) W_1=epsilon1
IF(x1p.LE.0) W_11=0

W2 = -x2p + y2upsf +epsilon2;
W21= -x2p + y2upsf
W_2 = x2p + epsilon2;
W_21= x2p
IF((x2p-y2upsf).GE.0) W2=epsilon2
IF((x2p-y2upsf).GE.0) W21=0
IF(x2p.LE.0) W_2=epsilon2
IF(x2p.LE.0) W_21=0

W3 = -x3p + y3upsf +epsilon3;
W31= -x3p + y3upsf
W_3 = x3p + epsilon3;
W_31= x3p
IF((x3p-y3upsf).GE.0) W3=epsilon3
IF((x3p-y3upsf).GE.0) W31=0
IF(x3p.LE.0) W_3=epsilon3
IF(x3p.LE.0) W_31=0

Td = Tem-273.15D0

pA = 9.01D-01 - 8.35D-04*Td;
pB = 9.19D-01 - 6.65D-04*Td;
pI = 8.85D-01 ;
pSol = 8.92D-01 - 1.3D-03*Td ;
pPol = 1.19D0 - 8.07D-04*Td;

V=x1p/pA+x2p/pB+x3p/pI+msol/pSol+x4p/pPol+x5p/pPol

W6 = x4p + epsilon1;
```

```
W61= x4p
IF(x4p.LE.0) W6=epsilon1
IF(x4p.LE.0) W61=0

W7 = x5p + epsilon1;
W71= x5p
IF(x5p.LE.0) W7=epsilon1
IF(x5p.LE.0) W71=0

W8 = x6p + epsilon1;
W81= x6p
IF(x6p.LE.0) W8=epsilon1
IF(x6p.LE.0) W81=0

! U1
W13 = U1 + epsilon1;
W131= U1;
W14 = -U1 + u1ins +epsilon1;
W141= -U1 + u1ins
IF (U1.LE.0) W13=epsilon1
IF (U1.LE.0) W131=0
IF ((U1-u1ins).GE.0) W14=epsilon1
IF ((U1-u1ins).GE.0) W141=0

! U2
W15 = U2 + epsilon2;
W151= U2;
W16 = -U2 + u2ins +epsilon2;
W161= -U2 + u2ins
IF(U2.LE.0) W15=epsilon2
IF(U2.LE.0) W151=0
IF((U2-u2ins).GE.0) W16=epsilon2
IF((U2-u2ins).GE.0) W161=0

! U3
W17 = U3 + epsilon3;
W171= U3;
W18 = -U3 + u3ins +epsilon3;
W181= -U3 + u3ins;
IF (U3.LE.0) W17=epsilon3
IF (U3.LE.0) W171=0
IF ((U3-u3ins).GE.0) W18=epsilon3
IF ((U3-u3ins).GE.0) W181=0

!initiation coefficient < s^-1 >
kd = 6.78D+15*DEXP(-17714.0D0/Tem) ;
```

```
!propagation rates < l/(mol.s) >
kpaa = 3.802D+06*DEXP(-2754D0/Tem);
kpbb = 4.266D+07*DEXP(-3910.0D0/Tem);
kpab = kpaa/r1 ;
kpba = kpbb/r2 ;

!termination coefficients < l/(mol.s) >
ktermaa = 7.10D+07*DEXP(-830.0D0/Tem);
ktermbb = 3.818D+09*DEXP(-958.0D0/Tem);


!transfer to monomer < l/(mol.s) >
ktrm_aa = 1.56D+02*DEXP(-2621.0D0/Tem) ;
ktrm_bb = 2.31D+06*DEXP(-6377.0D0/Tem) ;
ktrm_ba = ktrm_aa*kpbb/(kpaa*r2);
ktrm_ab = ktrm_bb*kpaa/(kpbb*r1);

!transfer to solvent < l/(mol.s) >
Cs_a = 5.55D0*DEXP(-4590.0D0/Tem) ;
Cs_b = 1D-04;
ktrs_a = Cs_a*kpaa ;
ktrs_b = Cs_b*kpbb;

!densities (Kg/l)
pA = 9.01D-01 - 8.35D-04*Td;
pB = 9.19D-01 - 6.65D-04*Td;
pI = 8.85D-01 ;
pSol = 8.92D-01 - 1.3D-03*Td ;
pPol = 1.19D0 - 8.07D-04*Td;

CC = DSQRT(2.0D0*f*kd/Mi);
DE = Mr/(x1p/pA +x2p/pB + x3p/pI+ msol/pSol+ x4p/pPol+ x5p/pPol);
f2 = x1p*Mb / (x1p*Mb + x2p*Ma);
f3 = x2p*Ma / (x1p*Mb + x2p*Ma);
frA = kpba*Mb*x1p/( kpba*Mb*x1p + kpab*Ma*x2p);
frB = kpab*Ma*x2p/( kpba*Mb*x1p + kpab*Ma*x2p);
Ktcop=10**((f2*DLOG(ktermaa)+f3*DLOG(ktermbb))/DLOG(10.0D0));
Ltot = CC*DSQRT(DE*x3p/(Ktcop*Mr))

La = Ltot*frA ;
Lb = Ltot*frB ;
V = Mr/DE ;

ktc_aa = (1.0D0-alpha)*Ktcop ;
ktc_bb = (1.0D0-betha)*Ktcop ;
```

```
   ktc_ab = (1.0D0-gamma)*Ktcop ;
   ktc_ba = ktc_ab ;

   Rinit = (CC**2)*DE*x3p/Mr ;
   m1 = (ktrm_aa*frA+ktrm_ba*frB)/Ma;
   m2 = (ktrm_bb*frB+ktrm_ab*frA)/Mb;
   Rtrm = (m1*(x1p/Mr) + m2*(x2p/Mr))*Ltot*DE ;
   s = (ktrs_a*frA + ktrs_b*frB)*msol/Ms;
   Rtrs = s*Ltot*DE/Mr;
   Rtc=ktc_aa*((La)**2)+ktc_bb*((Lb)**2)+2*ktc_ab*La*Lb;

   Ruo = Rinit+ Rtrm + Rtrs - 0.5D0*Rtc ;
   Fcnum=x4p/Ma;
   Fcden=x4p/Ma + x5p/Mb;
   Fc=Fcnum/(Fcden);

   Mnum=x4p+x5p;
   Mden=x6p;
   Mn=Mnum/(Mden);

   prodc_u=W13*W14*W15*W16*W17*W18

   dprodcu_dp1=(W13*W14*W15)*(PAR(3)*W16*(W18-W17)-PAR(2)*W17*W18)
                    + (W16*W17*W18)*(W13*(PAR(2)*W14-W15)+W14*W15)
   dprodcu_dp2=(PAR(1)/(W15*W16))*(W16-W15)
   dprodcu_dp3=(PAR(1)/(W17*W18))*(W18-W17)

   GO TO (1,2,3,4),JPAR
1  DU1_DP1 = 1
   DU2_DP1  = PAR(2)
   DU3_DP1  = PAR(3)
   DFDP(1) = PAR(4)*ni*DU1_DP1
   DFDP(2) = PAR(4)*ni*DU2_DP1
   DFDP(3) = PAR(4)*ni*DU3_DP1
   !DFDP(4) = 0
   !DFDP(5) = 0
   !DFDP(6) = 0
   DFDP(7) = -mu1*dprodcu_dp1/prodc_u
   RETURN
2  DU1_DP2 = 0
   DU2_DP2  = PAR(1)
   DU3_DP2  = 0
   !DFDP(1) = PAR(4)*ni*DU1_DP2
   DFDP(2) = PAR(4)*ni*DU2_DP2
   !DFDP(3) = PAR(4)*ni*DU3_DP2
   !DFDP(4) = 0
```

```
      !DFDP(5) = 0
      !DFDP(6) = 0
      DFDP(7) = -mu1*dprodcu_dp2
      RETURN
3     DU1_DP3 = 0
      DU2_DP3 = 0
      DU3_DP3  = PAR(1)
      !DFDP(1) = PAR(4)*ni*DU1_DP3
      !DFDP(2) = PAR(4)*ni*DU2_DP3
      DFDP(3) = PAR(4)*ni*DU3_DP3
      !DFDP(4) = 0
      !DFDP(5) = 0
      !DFDP(6) = 0
      DFDP(7) = -mu1*dprodcu_dp3
      RETURN
4     DU1_DP4 = 0
      DU2_DP4 = 0
      DU3_DP4 = 0
     DFDP(1) = ni*(U1 - x1p*( kpaa*La + kpba*Lb ))
     DFDP(2) = ni*(U2 - x2p*( kpbb*Lb + kpab*La ))
     DFDP(3) = ni*(U3 - kd*x3p)
     DFDP(4) = ni*(x1p*( kpaa*La + kpba*Lb ))
     DFDP(5) = ni*(x2p*( kpbb*Lb + kpab*La ))
     DFDP(6) = ni*(V*Ruo)
     DFDP(7) = ni*(wff1*(((Mn/Msp)-1)**2) + wff2*(((Fc/Fsp)-1)**2))
     RETURN
     END


     SUBROUTINE JEX (NEQ, T, Y, PAR, ML, MU, PD, NRPD)
     DOUBLE PRECISION T,Y,PAR,U1,U2,U3,prodc,ES ,PD ,prodc_s
     DOUBLE PRECISION W1,W_1,W2,W_2,W3,W_3,W4,W5,W6,W7,W8,W9,W10
     DOUBLE PRECISION W12,W13,W14,W15,W16,W17,W18,W19,W20
     DOUBLE PRECISION W11,W_11,W21,W_21,W31,W_31,W41,W51,W61,W71
     DOUBLE PRECISION W81,W91,W101,W121,W131,W141,W151,W161
     DOUBLE PRECISION W171,W181,W191,W201,fac,Td,mfed,Mmax
     DIMENSION Y(7),PD(NRPD,7), PAR(24)
     DOUBLE PRECISION kd,kpaa,kpbb,kpab,kpba,ktrm_aa,ktrm_bb
     DOUBLE PRECISION ktrm_ab,ktrm_ba,Cs_a,Cs_b,ktrs_a,ktrs_b
     DOUBLE PRECISION ktermaa,ktermbb,frac,prodc_u
     DOUBLE PRECISION pA,pB,pI,pSol,pPol,f2,f3,frA,frB,DE,SS
     DOUBLE PRECISION CC,Ktcop,Ltot,La,Lb,V,ktc_aa,ktc_bb,ktc_ab
     DOUBLE PRECISION ktc_ba,Rinit,Rtrm,Rtrs,Rtc,Ruo
     DOUBLE PRECISION Fcnum,Fcden,Fc,Mnum,Mden,Mn,Msp,Fsp,Tem
     DOUBLE PRECISION r1,r2,f,alpha,betha,gamma,Mi,Ma,Mb,Ms,msol
     DOUBLE PRECISION mu1,epsilon1,epsilon2,wff1,wff2,eff
DOUBLE PRECISION y1upsf,y2upsf,y3upsf,u1ins,u2ins,u3ins,Vf
DOUBLE PRECISION x1p,x2p,x3p,x4p,x5p,x6p,x7p,Mr,eps,Vff
DOUBLE PRECISION dDE_dy1,dDE_dy2,dDE_dy3,dDE_dy4,dDE_dy5
DOUBLE PRECISION dDE_dy6,dDE_dy7,df2dy2,df3dy2,df2dy3,df3dy3
DOUBLE PRECISION dKtcop_dy1,dKtcop_dy2,dKtcop_dy3,dKtcop_dy4
DOUBLE PRECISION dKtcop_dy5,dKtcop_dy6,dKtcop_dy7
DOUBLE PRECISION dLtot_dy1,dLtot_dy2 ,dLtot_dy3 ,dLtot_dy4
DOUBLE PRECISION dLtot_dy5 ,dLtot_dy6 ,dLtot_dy7
DOUBLE PRECISION dfrA_dy1,dfrB_dy1,dfrA_dy2,dfrB_dy2,dfrA_dy3
DOUBLE PRECISION dfrB_dy3,dfrA_dy4,dfrB_dy4
DOUBLE PRECISION dfrA_dy5,dfrB_dy5,dfrA_dy6,dfrB_dy6,dfrA_dy7
DOUBLE PRECISION dfrB_dy7,dLa_dy1,dLb_dy1,dLa_dy2
DOUBLE PRECISION dLb_dy2,dLa_dy3,dLb_dy3,dLa_dy4,dLb_dy4
DOUBLE PRECISION dLa_dy5,dLb_dy5,dLa_dy6,dLb_dy6
DOUBLE PRECISION dLa_dy7,dLb_dy7
DOUBLE PRECISION dF1_dy1,dF1_dy2,dF1_dy3,dF1_dy4,dF1_dy5
DOUBLE PRECISION dF1_dy6,dF1_dy7,dF2_dy1,dF2_dy2
DOUBLE PRECISION dF2_dy3,dF2_dy4,dF2_dy5,dF2_dy6,dF2_dy7
DOUBLE PRECISION dF3_dy1,dF3_dy2,dF3_dy3,dF3_dy4,dF3_dy5
DOUBLE PRECISION dF3_dy6,dF3_dy7,dF4_dy1,dF4_dy2
DOUBLE PRECISION dF4_dy3,dF4_dy4,dF4_dy5,dF4_dy6,dF4_dy7
DOUBLE PRECISION dF5_dy1,dF5_dy2,dF5_dy3,dF5_dy4,dF5_dy5
DOUBLE PRECISION dF5_dy6,dF5_dy7,dF6_dy1,dF6_dy2
DOUBLE PRECISION dF6_dy3,dF6_dy4,dF6_dy5,dF6_dy6,dF6_dy7
DOUBLE PRECISION dF7_dy1,dF7_dy2,dF7_dy3,dF7_dy4,dF7_dy5
DOUBLE PRECISION dF7_dy6,dF7_dy7
DOUBLE PRECISION m1,dm1_dy1 ,dm1_dy2 ,dm1_dy3 ,dm1_dy4
DOUBLE PRECISION dm1_dy5,dm1_dy6,dm1_dy7,m2,dm2_dy1,dm2_dy2
DOUBLE PRECISION dm2_dy3,dm2_dy4 ,dm2_dy5 ,dm2_dy6 ,dm2_dy7
DOUBLE PRECISION s ,ds_dy1 ,ds_dy2 ,ds_dy3 ,ds_dy4
DOUBLE PRECISION ds_dy5 ,ds_dy6 ,ds_dy7
DOUBLE PRECISION dRinit_dy1,dRinit_dy2,dRinit_dy3,dRinit_dy4
DOUBLE PRECISION dRinit_dy5 ,dRinit_dy6 ,dRinit_dy7
DOUBLE PRECISION dRtrm_dy1 ,dRtrm_dy2 ,dRtrm_dy3 ,dRtrm_dy4
DOUBLE PRECISION dRtrm_dy5 ,dRtrm_dy6 ,dRtrm_dy7
DOUBLE PRECISION dRtrs_dy1 ,dRtrs_dy2 ,dRtrs_dy3 ,dRtrs_dy4
DOUBLE PRECISION dRtrs_dy5 ,dRtrs_dy6 ,dRtrs_dy7
DOUBLE PRECISION dRtc_dy1,dRtc_dy2 ,dRtc_dy3 ,dRtc_dy4
DOUBLE PRECISION dRtc_dy5 ,dRtc_dy6 ,dRtc_dy7
DOUBLE PRECISION dRuo_dy1 ,dRuo_dy2 ,dRuo_dy3 ,dRuo_dy4
DOUBLE PRECISION dRuo_dy5 ,dRuo_dy6 ,dRuo_dy7
DOUBLE PRECISION dV_dy1 ,dV_dy2 ,dV_dy3 ,dV_dy4
DOUBLE PRECISION dV_dy5 ,dV_dy6 , dV_dy7
DOUBLE PRECISION dFc_dy1,dFc_dy2,dFc_dy3,dFc_dy4,dFc_dy5
DOUBLE PRECISION dFc_dy6 ,dFc_dy7 ,dFc_dy8
DOUBLE PRECISION dMn_dy1,dMn_dy2,dMn_dy3,dMn_dy4,dMn_dy5
DOUBLE PRECISION dMn_dy6 ,dMn_dy7 ,dMn_dy8,dprodc_u_dy4
DOUBLE PRECISION dprodc_s_dy1,dprodc_s_dy2,dprodc_s_dy3
DOUBLE PRECISION dprodc_s_dy5 ,dprodc_s_dy6 ,dprodc_s_dy7
DOUBLE PRECISION dES_dy2,dES_dy3,Z1,Z2,Z3,facc,mPOL,cte_u
INTEGER ni
r1=1.0*0.42D0
r2=1.0*0.61D0
f=5.1548D-01
alpha=0.65D0
betha= 0.01D0
gamma= 0.33D0
Mi= 1.3210D-01
Ma=  1.42D-01;
Mb=  1.0415D-01
Ms= 1.0617D-01
eps=  2.2204D-016

Msp=PAR(5)
Fsp=PAR(6)
Tem=PAR(7)
msol=PAR(8)
wff1= PAR(9)
wff2= PAR(10)
y1upsf=PAR(11)
y2upsf=PAR(12)
y3upsf=PAR(13)
u1ins=PAR(14)
u2ins=PAR(15)
u3ins=PAR(16)
ni=PAR(17)
mPOL=PAR(18)
frac=PAR(19)
Vf=PAR(20)
mu1=PAR(21)
epsilon1=PAR(22)
epsilon2=PAR(23)
epsilon3=PAR(24)

U1=PAR(1);
U2=PAR(2)*PAR(1);
U3=PAR(3)*PAR(1);

IF ((U1-u1ins).GE.0) U1=u1ins
IF (U1.LE.0)  U1=0
IF ((U2-u2ins).GE.0) U2=u2ins
IF (U2.LE.0)  U2=0
IF ((U3-u3ins).GE.0) U3=u3ins
IF (U3.LE.0)  U3=0

x1p=(Y(1)**2+eps)**0.5;
x2p=(Y(2)**2+eps)**0.5;
x3p=(Y(3)**2+eps)**0.5;
x4p=(Y(4)**2+eps)**0.5;
x5p=(Y(5)**2+eps)**0.5;
x6p=(Y(6)**2+eps)**0.5;
x7p=(Y(7)**2+eps)**0.5;

Mr=x1p+x2p+x3p+x4p+x5p+msol;

W1 = -x1p + y1upsf +epsilon1;
W11= -x1p + y1upsf;
W_1 = x1p + epsilon1;
W_11= x1p
IF((x1p-y1upsf).GE.0) W1=epsilon1
IF((x1p-y1upsf).GE.0) W11=0
IF(x1p.LE.0) W_1=epsilon1
IF(x1p.LE.0) W_11=0

W2 = -x2p + y2upsf +epsilon2;
W21= -x2p + y2upsf
W_2 = x2p + epsilon2;
W_21= x2p
IF((x2p-y2upsf).GE.0) W2=epsilon2
IF((x2p-y2upsf).GE.0) W21=0
IF(x2p.LE.0) W_2=epsilon2
IF(x2p.LE.0) W_21=0

W3 = -x3p + y3upsf +epsilon3;
W31= -x3p + y3upsf
W_3 = x3p + epsilon3;
W_31= x3p
IF((x3p-y3upsf).GE.0) W3=epsilon3
IF((x3p-y3upsf).GE.0) W31=0
IF(x3p.LE.0) W_3=epsilon3
IF(x3p.LE.0) W_31=0

Td = Tem-273.15D0

pA = 9.01D-01 - 8.35D-04*Td;
pB = 9.19D-01 - 6.65D-04*Td;
pI = 8.85D-01 ;
pSol = 8.92D-01 - 1.3D-03*Td
```

```
pPol = 1.19D0 - 8.07D-04*Td

V=x1p/pA+x2p/pB+x3p/pI+msol/pSol+x4p/pPol+x5p/pPol

W6 = x4p + epsilon1;
W61= x4p
IF(x4p.LE.0) W6=epsilon1
IF(x4p.LE.0) W61=0

W7 = x5p + epsilon1;
W71= x5p
IF(x5p.LE.0) W7=epsilon1
IF(x5p.LE.0) W71=0

W8 = x6p + epsilon1;
W81= x6p
IF(x6p.LE.0) W8=epsilon1
IF(x6p.LE.0) W81=0

! U1
W13 = U1 + epsilon1;
W131= U1;
W14 = -U1 + u1ins +epsilon1;
W141= -U1 + u1ins
IF (U1.LE.0) W13=epsilon1
IF (U1.LE.0) W131=0
IF ((U1-u1ins).GE.0) W14=epsilon1
IF ((U1-u1ins).GE.0) W141=0

! U2
W15 = U2 + epsilon2;
W151= U2;
W16 = -U2 + u2ins +epsilon2;
W161= -U2 + u2ins
IF(U2.LE.0) W15=epsilon2
IF(U2.LE.0) W151=0
IF((U2-u2ins).GE.0) W16=epsilon2
IF((U2-u2ins).GE.0) W161=0



! U3
W17 = U3 + epsilon3
W171= U3;
W18 = -U3 + u3ins +epsilon3
W181= -U3 + u3ins;
```

```
IF (U3.LE.0) W17=epsilon3
IF (U3.LE.0) W171=0
IF ((U3-u3ins).GE.0) W18=epsilon3
IF ((U3-u3ins).GE.0) W181=0

!initiation coefficient < s^-1 >
kd = 6.78D+15*DEXP(-17714.0D0/Tem) ;

!propagation rates < l/(mol.s) >
kpaa = 3.802D+06*DEXP(-2754.2D0/Tem);
kpbb = 4.266D+07*DEXP(-3910.0D0/Tem);
kpab = kpaa/r1 ;
kpba = kpbb/r2 ;

!termination coefficients < l/(mol.s) >
ktermaa = 7.10D+07*DEXP(-830.0D0/Tem);
ktermbb = 3.818D+09*DEXP(-958.0D0/Tem);

!transfer to monomer < l/(mol.s) >
ktrm_aa = 1.56D+02*DEXP(-2621.0D0/Tem) ;
ktrm_bb = 2.31D+06*DEXP(-6377.0D0/Tem) ;
ktrm_ba = ktrm_aa*kpbb/(kpaa*r2); ! ktrm_aa*kpba/kpaa;
ktrm_ab = ktrm_bb*kpaa/(kpbb*r1); ! ktrm_bb*kpab/kpbb;

!transfer to solvent < l/(mol.s) >
Cs_a = 5.55D0*DEXP(-4590.0D0/Tem) ;
Cs_b = 1D-04
ktrs_a = Cs_a*kpaa ;
ktrs_b = Cs_b*kpbb ;

!densities (Kg/l)
pA = 9.01D-01 - 8.35D-04*Td;
pB = 9.19D-01 - 6.65D-04*Td;
pI = 8.85D-01 ;
pSol = 8.92D-01 - 1.3D-03*Td ;
pPol = 1.19D0 - 8.07D-04*Td;

CC = DSQRT(2.0D0*f*kd/Mi);
DE = Mr/(x1p/pA+x2p/pB+x3p/pI+msol/pSol+x4p/pPol+x5p/pPol)

dV_dy1=1.0D0/pA
dV_dy2=1.0D0/pB
dV_dy3=1.0D0/pI
dV_dy4=1.0D0/pSol
dV_dy5=1.0D0/pSol
```

```
dMr_dy1=1.0D0
dMr_dy2=1.0D0
dMr_dy3=1.0D0
dMr_dy4=1.0D0
dMr_dy5=1.0D0

dDE_dy1= -1.0D0*Mr*dV_dy1/(V**2) + dMr_dy1/V
dDE_dy2= -1.0D0*Mr*dV_dy2/(V**2) + dMr_dy2/V
dDE_dy3= -1.0D0*Mr*dV_dy3/(V**2) + dMr_dy3/V
dDE_dy4= -1.0D0*Mr*dV_dy4/(V**2) + dMr_dy4/V
dDE_dy5= -1.0D0*Mr*dV_dy5/(V**2) + dMr_dy5/V
!dDE_dy6= 0.0D0
!dDE_dy7= 0.0D0

f2 = x1p*Mb/(x1p*Mb + x2p*Ma);
f3 = x2p*Ma/(x1p*Mb + x2p*Ma);
frA = kpba*Mb*x1p/( kpba*Mb*x1p + kpab*Ma*x2p);
frB = kpab*Ma*x2p/( kpba*Mb*x1p + kpab*Ma*x2p);

Ktcop = 10**((f2*DLOG(ktermaa) + f3*DLOG(ktermbb))/DLOG(10.0D0));
Ltot = CC*DSQRT(DE*x3p/(Ktcop*Mr))

Fcnum=x4p/Ma;
Fcden=x4p/Ma + x5p/Mb;
Fc=Fcnum/(Fcden);!Fden+eps
Mnum=x4p+x5p;
Mden=x6p;
Mn=Mnum/(Mden);!Mden+eps

df2dy1 = Mb/( x1p*Mb + x2p*Ma) -  x1p*(Mb*Mb)/((x1p*Mb + x2p*Ma)**2);
df3dy1 = -1.0D0*df2dy1
df2dy2 = -1.0D0*x1p*Mb*Ma/((x1p*Mb + x2p*Ma)**2);
df3dy2 = -1.0D0*df2dy2

ES=(f2*DLOG(ktermaa)+f3*DLOG(ktermbb))/DLOG(10.0D0)
dES_dy1=(df2dy1*DLOG(Ktermaa)+ df3dy1*DLOG(Ktermbb))/DLOG(10.0D0)
dES_dy2=(df2dy2*DLOG(Ktermaa)+ df3dy2*DLOG(Ktermbb))/DLOG(10.0D0)
dKtcop_dy1=(10**ES)*DLOG(10.0D0)*dES_dy1
dKtcop_dy2=(10**ES)*DLOG(10.0D0)*dES_dy2

dfrA_dy1=kpba*Mb/(kpba*Mb*x1p+ kpab*Ma*x2p) &
        -((kpba*Mb)**2)*x1p/((kpba*Mb*x1p+ kpab*Ma*x2p)**2)
dfrB_dy1=-1.0D0*dfrA_dy1
dfrA_dy2=-1.0D0*kpba*Mb*x1p*kpab*Ma/((kpba*Mb*x1p+kpab*Ma*x2p)**2) ;
dfrB_dy2=-1.0D0*dfrA_dy2
```

```
Z1=0.5D0*CC*((Ktcop*Mr/(DE*x3p))**0.5)
Z2=1.0D0/((Ktcop*Mr)**2)
Z3=1.0D0/(Ktcop*Mr)

dLtot_dy1 = Z1*x3p*(-1.0D0*DE*Z2*(dKtcop_dy1*Mr+Ktcop*dMr_dy1)+ &
                              Z3*dDE_dy1)
dLtot_dy2 = Z1*x3p*(-1.0D0*DE*Z2*(dKtcop_dy2*Mr+Ktcop*dMr_dy2)+ &
                Z3*dDE_dy2)

 dLtot_dy3 = Z1*(1/Ktcop)*(-1.0D0*DE*x3p*dMr_dy3/((Mr)**2) + &
                     (1/Mr)*(x3p*dDE_dy3+DE))

 dLtot_dy4 = Z1*(x3p/Ktcop)*(-1.0D0*DE*dMr_dy4/((Mr)**2) + &
                      (1/Mr)*dDE_dy4)

 dLtot_dy5 = Z1*(x3p/Ktcop)*(-1.0D0*DE*dMr_dy5/((Mr)**2) + &
                      (1/Mr)*dDE_dy5)
 !dLtot_dy6 =0
 !dLtot_dy7 = 0

 dFc_dy4 = (1.0D0/Ma)/( x4p/Ma + x5p/Mb ) - (x4p/(Ma**2))/(( x4p/Ma +
              x5p/Mb)**2);
 dFc_dy5 = (-1.0D0*x4p/(Ma*Mb))/((x4p/Ma + x5p/Mb )**2)

 dMn_dy4 = 1.0D0/(x6p);
 dMn_dy5 = 1.0D0/(x6p);
 dMn_dy6 = -1.0D0*(x4p+x5p)/((x6p)**2);

  La = Ltot*frA ;
  Lb = Ltot*frB ;
  ktc_aa = (1.0D0-alpha)*Ktcop
  ktc_bb = (1.0D0-betha)*Ktcop
  ktc_ab = (1.0D0-gamma)*Ktcop
  ktc_ba = ktc_ab

  Rinit = (CC**2)*DE*x3p/Mr ;
  m1 = (ktrm_aa*frA+ktrm_ba*frB)/Ma;
  m2 = (ktrm_bb*frB+ktrm_ab*frA)/Mb;
  Rtrm = (m1*x1p/Mr + m2*x2p/Mr)*Ltot*DE ;
  s = (ktrs_a*frA + ktrs_b*frB)*msol/Ms;
  Rtrs = s*Ltot*DE/Mr;
  Rtc = ktc_aa*(La**2) + ktc_bb*(Lb**2) +2*ktc_ab*La*Lb

  Ruo = Rinit+ Rtrm + Rtrs - 0.5D0*Rtc ;

  dLa_dy1 = dLtot_dy1*frA + dfrA_dy1*Ltot
 dLb_dy1 = dLtot_dy1*frB + dfrB_dy1*Ltot

 dLa_dy2 = dLtot_dy2*frA + dfrA_dy2*Ltot
 dLb_dy2 = dLtot_dy2*frB + dfrB_dy2*Ltot

 dLa_dy3 = dLtot_dy3*frA
 dLb_dy3 = dLtot_dy3*frB

 dLa_dy4 = dLtot_dy4*frA
 dLb_dy4 = dLtot_dy4*frB

 dLa_dy5 = dLtot_dy5*frA
 dLb_dy5 = dLtot_dy5*frB

 !dLa_dy6 = 0
 !dLb_dy6 = 0

!dLa_dy7 = 0
!dLb_dy7 = 0

dF1_dy1=-(kpaa*La + kpba*Lb)-x1p*(kpaa*dLa_dy1+kpba*dLb_dy1)
dF1_dy2 = -x1p*(kpaa*dLa_dy2 + kpba*dLb_dy2)
dF1_dy3 = -x1p*(kpaa*dLa_dy3 + kpba*dLb_dy3)
dF1_dy4 = -x1p*(kpaa*dLa_dy4 + kpba*dLb_dy4)
dF1_dy5 = -x1p*(kpaa*dLa_dy5 + kpba*dLb_dy5)
!dF1_dy6 = 0
!dF1_dy7 = 0.0D0

dF2_dy1 = -x2p*(kpbb*dLb_dy1 + kpab*dLa_dy1);
dF2_dy2=-(kpbb*Lb + kpab*La)-x2p*(kpbb*dLb_dy2+kpab*dLa_dy2);
dF2_dy3 = -x2p*(kpbb*dLb_dy3 + kpab*dLa_dy3);
dF2_dy4 = -x2p*(kpbb*dLb_dy4 + kpab*dLa_dy4);
dF2_dy5 = -x2p*(kpbb*dLb_dy5 + kpab*dLa_dy5);
!dF2_dy6 = 0
!dF2_dy7 = 0.0D0

!dF3_dy1 = 0.0D0 ;
!dF3_dy2 = 0.0D0 ;
dF3_dy3 =-kd;
!dF3_dy4 = 0.0D0 ;
!dF3_dy5 = 0.0D0 ;
!dF3_dy6 = 0.0D0 ;
!dF3_dy7 = 0.0D0 ;

dF4_dy1 = (kpaa*La + kpba*Lb) + x1p*(kpaa*dLa_dy1 + kpba*dLb_dy1);
dF4_dy2 = x1p*(kpaa*dLa_dy2 + kpba*dLb_dy2) ;
```

```
dF4_dy3 = x1p*(kpaa*dLa_dy3 + kpba*dLb_dy3) ;
dF4_dy4 = x1p*(kpaa*dLa_dy4 + kpba*dLb_dy4) ;
dF4_dy5 = x1p*(kpaa*dLa_dy5 + kpba*dLb_dy5) ;
!dF4_dy6 = 0
!dF4_dy7 = 0.0D0 ;

dF5_dy1 = x2p*(kpbb*dLb_dy1 + kpab*dLa_dy1);
dF5_dy2 = (kpbb*Lb + kpab*La) + x2p*(kpbb*dLb_dy2 + kpab*dLa_dy2);
dF5_dy3 = x2p*(kpbb*dLb_dy3 + kpab*dLa_dy3);
dF5_dy4 = x2p*(kpbb*dLb_dy4 + kpab*dLa_dy4);
dF5_dy5 = x2p*(kpbb*dLb_dy5 + kpab*dLa_dy5);
!dF5_dy6 = 0
!dF5_dy7 = 0.0D0

!m1 = (ktrm_aa*frA+ktrm_ba*frB )/Ma;

dm1_dy1 = (ktrm_aa*dfrA_dy1 + ktrm_ba*dfrB_dy1)/Ma;
dm1_dy2 = (ktrm_aa*dfrA_dy2 + ktrm_ba*dfrB_dy2)/Ma;
!dm1_dy3 = 0.0D0;
!dm1_dy4 = 0.0D0;
!dm1_dy5 = 0.0D0;
!dm1_dy6 = 0.0D0;

!m2 = (ktrm_bb*frB+ktrm_ab*frA)/Mb;

dm2_dy1 = (ktrm_bb*dfrB_dy1 + ktrm_ab*dfrA_dy1)/Mb;
dm2_dy2 = (ktrm_bb*dfrB_dy2 + ktrm_ab*dfrA_dy2)/Mb;
!dm2_dy3 = 0.0D0;
!dm2_dy4 = 0.0D0;
!dm2_dy5 = 0.0D0;
!dm2_dy6 = 0.0D0;

!s = (ktrs_a*frA + ktrs_b*frB)*msol/Ms;

ds_dy1 = (ktrs_a*dfrA_dy1 + ktrs_b*dfrB_dy1)*msol/Ms;
ds_dy2 = (ktrs_a*dfrA_dy2 + ktrs_b*dfrB_dy2)*msol/Ms;
!ds_dy3 = 0.0D0;
!ds_dy4 = 0.0D0;
!ds_dy5 = 0.0D0;
!ds_dy6 = 0.0D0;

dRinit_dy1=(CC**2)*x3p*(-1.0D0*DE*dMr_dy1/((Mr)**2)+dDE_dy1*(1/Mr))
dRinit_dy2=(CC**2)*x3p*(-1.0D0*DE*dMr_dy2/((Mr)**2)+dDE_dy2*(1/Mr))
dRinit_dy3=(CC**2)*(x3p*(-1.0D0*DE*dMr_dy3/((Mr)**2)+dDE_dy3*(1/Mr)) &
                 + DE*(1/Mr))
```

```
dRinit_dy4=(CC**2)*x3p*(-1.0D0*DE*dMr_dy4/((Mr)**2)+dDE_dy4*(1/Mr))
dRinit_dy5=(CC**2)*x3p*(-1.0D0*DE*dMr_dy5/((Mr)**2)+dDE_dy5*(1/Mr))
!dRinit_dy6 = 0.0D0
!dRinit_dy7 = 0.0D0

dRtrm_dy1 = (m1+x1p*dm1_dy1 +x2p*dm2_dy1)*(1/Mr)*Ltot*DE  &
    + (m1*x1p+m2*x2p)*(Ltot*(dDE_dy1*(1/Mr) &
    - DE*dMr_dy1/((Mr)**2))+dLtot_dy1*DE*(1/Mr))

dRtrm_dy2 = (m2+x1p*dm1_dy2 +x2p*dm2_dy2)*(1/Mr)*Ltot*DE  &
    + (m1*x1p+m2*x2p)*(Ltot*(dDE_dy2*(1/Mr) &
    - DE*dMr_dy2/((Mr)**2))+dLtot_dy2*DE*(1/Mr))

 dRtrm_dy3 = (m1*x1p+m2*x2p)*(Ltot*(dDE_dy3*(1/Mr)-
 DE*dMr_dy3/((Mr)**2)) & +dLtot_dy3*DE*(1/Mr))

 dRtrm_dy4 = (m1*x1p+m2*x2p)*(Ltot*(dDE_dy4*(1/Mr) -
 DE*dMr_dy4/((Mr)**2)) & +dLtot_dy4*DE*(1/Mr))

 dRtrm_dy5 = (m1*x1p+m2*x2p)*(Ltot*(dDE_dy5*(1/Mr)-
 DE*dMr_dy5/((Mr)**2)) & +dLtot_dy5*DE*(1/Mr))
!dRtrm_dy6 = 0
!dRtrm_dy7 = 0.0D0

 dRtrs_dy1=s*(Ltot*(dDE_dy1*(1/Mr)-
 DE*dMr_dy1/((Mr)**2))+(DE/Mr)*dLtot_dy1) &+ Ltot*DE*ds_dy1*(1/Mr)

 dRtrs_dy2=s*(Ltot*(dDE_dy2*(1/Mr)-
 DE*dMr_dy2/((Mr)**2))+(DE/Mr)*dLtot_dy2) &+Ltot*DE*ds_dy2*(1/Mr)

 dRtrs_dy3=s*(Ltot*(dDE_dy3*(1/Mr)-
 DE*dMr_dy3/((Mr)**2))+(DE/Mr)*dLtot_dy3 )
 dRtrs_dy4=s*(Ltot*(dDE_dy4*(1/Mr)-
 DE*dMr_dy4/((Mr)**2))+(DE/Mr)*dLtot_dy4 )
 dRtrs_dy5=s*(Ltot*(dDE_dy5*(1/Mr)-
 DE*dMr_dy5/((Mr)**2))+(DE/Mr)*dLtot_dy5 )
!dRtrs_dy6=0.0D0
!dRtrs_dy7=0.0D0

dRtc_dy1 = 2.0D0*(ktc_aa*La*dLa_dy1 + ktc_bb*Lb*dLb_dy1 &
                + ktc_ab*(dLa_dy1*Lb+La*dLb_dy1));

dRtc_dy2 = 2.0D0*(ktc_aa*La*dLa_dy2 + ktc_bb*Lb*dLb_dy2 &
                + ktc_ab*(dLa_dy2*Lb+La*dLb_dy2));

dRtc_dy3 = 2.0D0*(ktc_aa*La*dLa_dy3 + ktc_bb*Lb*dLb_dy3 &
             + ktc_ab*(dLa_dy3*Lb+La*dLb_dy3));

dRtc_dy4 = 2.0D0*(ktc_aa*La*dLa_dy4 + ktc_bb*Lb*dLb_dy4 &
             + ktc_ab*(dLa_dy4*Lb+La*dLb_dy4));

dRtc_dy5 = 2.0D0*(ktc_aa*La*dLa_dy5 + ktc_bb*Lb*dLb_dy5 &
             + ktc_ab*(dLa_dy5*Lb+La*dLb_dy5));
!dRtc_dy6 = 0
!dRtc_dy7 = 0.0D0

dRuo_dy1 = dRinit_dy1 + dRtrm_dy1 + dRtrs_dy1 -0.5D0*dRtc_dy1;
dRuo_dy2 = dRinit_dy2 + dRtrm_dy2 + dRtrs_dy2 -0.5D0*dRtc_dy2;
dRuo_dy3 = dRinit_dy3 + dRtrm_dy3 + dRtrs_dy3 -0.5D0*dRtc_dy3;
dRuo_dy4 = dRinit_dy4 + dRtrm_dy4 + dRtrs_dy4 -0.5D0*dRtc_dy4;
dRuo_dy5 = dRinit_dy5 + dRtrm_dy5 + dRtrs_dy5 -0.5D0*dRtc_dy5;
!dRuo_dy6 = 0
!dRuo_dy7 = 0.0D0

dF6_dy1 = dV_dy1*Ruo + V*dRuo_dy1 ;
dF6_dy2 = dV_dy2*Ruo + V*dRuo_dy2 ;
dF6_dy3 = dV_dy3*Ruo + V*dRuo_dy3 ;
dF6_dy4 = dV_dy4*Ruo + V*dRuo_dy4 ;
dF6_dy5 = dV_dy5*Ruo + V*dRuo_dy5 ;
!dF6_dy6 = 0
!dF6_dy7 = 0.0D0

!prodc_s=W1*W_1*W2*W_2*W3*W_3*W6*W7*W8


dprodc_s_dy1=(1.0D0/(W1*W_1))*(W1-W_1)
dprodc_s_dy2=(1.0D0/(W2*W_2))*(W2-W_2)
dprodc_s_dy3=(1.0D0/(W3*W_3))*(W3-W_3)
dprodc_s_dy4=1.0D0/W6
dprodc_s_dy5=1.0D0/W7
dprodc_s_dy6=1.0D0/W8
!dprodc_s_dy7 = 0.0D0

dF7_dy1 = -mu1*dprodc_s_dy1 ;
dF7_dy2 = -mu1*dprodc_s_dy2 ;
dF7_dy3 = -mu1*dprodc_s_dy3 ;

dF7_dy4=PAR(4)*ni*(2.0D0*wff2*((Fc/Fsp)-1.0D0)*dFc_dy4/Fsp &
         + 2.0D0*wff1*((Mn/Msp)-1.0D0)*dMn_dy4/Msp) - mu1*dprodc_s_dy4;

dF7_dy5=PAR(4)*ni*(2.0D0*wff2*((Fc/Fsp) - 1.0D0)*dFc_dy5/Fsp &
         + 2.0D0*wff1*((Mn/Msp)-1.0D0)*dMn_dy5/Msp)  - mu1*dprodc_s_dy5;
```

```
dF7_dy6=PAR(4)*ni*(2.0D0*wff1*((Mn/Msp)-1.0D0)*dMn_dy6/Msp) &
          - mu1*dprodc_s_dy6
!dF7_dy7 = 0.0D0;


PD(1,1) = PAR(4)*ni*dF1_dy1
PD(1,2) = PAR(4)*ni*dF1_dy2
PD(1,3) = PAR(4)*ni*dF1_dy3
PD(1,4) = PAR(4)*ni*dF1_dy4
PD(1,5) = PAR(4)*ni*dF1_dy5
!PD(1,6) = PAR(4)*ni*dF1_dy6
!PD(1,7) = PAR(4)*ni*dF1_dy7

PD(2,1) = PAR(4)*ni*dF2_dy1
PD(2,2) = PAR(4)*ni*dF2_dy2
PD(2,3) = PAR(4)*ni*dF2_dy3
PD(2,4) = PAR(4)*ni*dF2_dy4
PD(2,5) = PAR(4)*ni*dF2_dy5
!PD(2,6) = PAR(4)*ni*dF2_dy6
!PD(2,7) = PAR(4)*ni*dF2_dy7

!PD(3,1) = PAR(4)*ni*dF3_dy1
!PD(3,2) = PAR(4)*ni*dF3_dy2
PD(3,3) = PAR(4)*ni*dF3_dy3
!PD(3,4) = PAR(4)*ni*dF3_dy4
!PD(3,5) = PAR(4)*ni*dF3_dy5
!PD(3,6) = PAR(4)*ni*dF3_dy6
!PD(3,7) = PAR(4)*ni*dF3_dy7

PD(4,1) = PAR(4)*ni*dF4_dy1
PD(4,2) = PAR(4)*ni*dF4_dy2
PD(4,3) = PAR(4)*ni*dF4_dy3
PD(4,4) = PAR(4)*ni*dF4_dy4
PD(4,5) = PAR(4)*ni*dF4_dy5
!PD(4,6) = PAR(4)*ni*dF4_dy6
!PD(4,7) = PAR(4)*ni*dF4_dy7

PD(5,1) = PAR(4)*ni*dF5_dy1
PD(5,2) = PAR(4)*ni*dF5_dy2
PD(5,3) = PAR(4)*ni*dF5_dy3
PD(5,4) = PAR(4)*ni*dF5_dy4
PD(5,5) = PAR(4)*ni*dF5_dy5
!PD(5,6) = PAR(4)*ni*dF5_dy6
!PD(5,7) = PAR(4)*ni*dF5_dy7
```

```
PD(6,1) = PAR(4)*ni*dF6_dy1
PD(6,2) = PAR(4)*ni*dF6_dy2
PD(6,3) = PAR(4)*ni*dF6_dy3
PD(6,4) = PAR(4)*ni*dF6_dy4
PD(6,5) = PAR(4)*ni*dF6_dy5
!PD(6,6) = PAR(4)*ni*dF6_dy6
!PD(6,7) = PAR(4)*ni*dF6_dy7

PD(7,1) = dF7_dy1
PD(7,2) = dF7_dy2
PD(7,3) = dF7_dy3
PD(7,4) = dF7_dy4
PD(7,5) = dF7_dy5
PD(7,6) = dF7_dy6
!PD(7,7) = dF7_dy7
RETURN
END


! The gateway routine
subroutine mexFunction(nlhs, plhs, nrhs, prhs)

! (pointer) Replace integer by integer*8 on the DEC Alpha
! 64-bit platform

integer plhs(*), prhs(*)
integer mxCreateDoubleMatrix
integer T_pr, steps_pr, p_pr, y0_pr, y_pr

integer nlhs, nrhs
integer m, n, size
integer mxGetM, mxGetN, mxIsNumeric
real*8  T, steps, p(24,1), y0(7,1), y(1,35)

! Check for proper number of arguments.
!  if (nrhs .ne. 2) then
!  call mexErrMsgTxt('Two inputs required.')
!  elseif (nlhs .ne. 1) then
!  call mexErrMsgTxt('One output required.')
!  endif

! Check to see both inputs are numeric.
! if (mxIsNumeric(prhs(1)) .ne. 1) then
! call mexErrMsgTxt('Input #1 is not a numeric.')
! elseif (mxIsNumeric(prhs(2)) .ne. 1) then
! call mexErrMsgTxt('Input #2 is not a numeric array.')
! endif

! Check that input #1 is a scalar.
! m = mxGetM(prhs(1))
! n = mxGetN(prhs(1))
! if(n .ne. 1 .or. m .ne. 1) then
! call mexErrMsgTxt('Input #1 is not a scalar.')
! endif

! Get the size of the input matrix.
! m = mxGetM(prhs(1))
! n = mxGetM(prhs(2))+1
m = 1
n = 35
size = m*n

! Create matrix for the return argument.
plhs(1) = mxCreateDoubleMatrix(m, n, 0)
T_pr = mxGetPr(prhs(1))
steps_pr= mxGetPr(prhs(2))
p_pr = mxGetPr(prhs(3))
y0_pr = mxGetPr(prhs(4))
y_pr = mxGetPr(plhs(1))

! Load the data into Fortran arrays.
call mxCopyPtrToReal8(T_pr, T, 1)
call mxCopyPtrToReal8(steps_pr, steps, 1)
call mxCopyPtrToReal8(p_pr, p, 24)
call mxCopyPtrToReal8(y0_pr, y0, 7)

! Call the computational subroutine.
 call odef(T, steps, p, y0 ,y)

!Load the output into a MATLAB array.
call mxCopyReal8ToPtr(y, y_pr, size)

return
end
```