INVESTIGATION OF POLYMER GRADE BLENDING IN ZIEGLER-NATTA CATALYSED ETHYLENE POLYMERISATION SYSTEMS

by

Candidate: Mogamat Thaabit Nacerodien

NCRMOG002

SUBMITTED TO THE UNIVERSITY OF CAPE TOWN

In fulfilment of the requirements for the degree

MSc (Chemical Engineering)

Engineering and the Built Environment

UNIVERSITY OF CAPE TOWN

May 4th 2014

Supervisor: Dr Randhir Rawatlal

Department of Chemical Engineering

PLAGIARISM DECLARATION

I know the meaning of plagiarism and declare that all the work in the document, save for that which is properly acknowledged, is my own.

Signed:_____

ABSTRACT

Polyethylene is one of the most widely used polymers to date and it is an important commodity in a variety of fields. Most existing polyethylene plants operate on technology involving heterogeneous Ziegler-Natta catalysts. Plants often change operating conditions to produce different polymer grades; this allows them to cater to a larger polymer market. A side-effect of this practice is the unwanted formation of off-spec polymer during the grade transition periods.

Numerous studies have been conducted to address the issue of off-spec polymer formation. These studies involve applying optimal control theory to minimise the grade transition time or to minimise the amount of off-spec polymer generated during the transient period. This field of study is known as grade transition optimisation.

The current study aims to provide an alternative approach to addressing the problem of off-spec accumulation. It is proposed that stored off-spec polymer is blended with virgin polymer to provide a saleable and desirable product. The approach might be different, but the same techniques used in grade transition optimisation are applied.

Polyethylene produced using Ziegler-Natta catalysts have relatively linear chains, thus a chain length distribution coefficient is sufficient to characterise the polymer product. The number average chain length and polydispersity index are adequate representatives of this distribution for reporting the properties of a polyethylene grade. For the purpose of applying optimal control theory, a polyethylene production process model was developed to calculate these average properties using a kinetic scheme based on fundamental principles. This process model is able to predict the polymer properties under both steady-state and unsteady-state behaviour. A key feature of the model is its ability to solve the system with low computational expense due to the use of the segregation approach to link particle properties to the overall bulk phase. This is especially useful since optimisation algorithms used in optimal control theory are iterative by nature. The Differential Evolution Algorithm (DEA) was used to minimise the objective functions that were developed for the optimisation schemes due to its ability to evaluate objective functions in parallel.

A model of the blending aspect of the process was developed where it was derived that the polymer moments are additive on a mass basis. Pure grades were blended in a laboratory in various mass ratios and analysed using GPC to determine their molecular weight distribution curves. It was found that the model-predicted curves and the experimentally-determined curves were an excellent match, thus validating the model.

In the current study, three procedures for blending off-spec material under standard industry conditions are proposed. The first method involves the introduction of off-spec polymer on a continuous basis to the virgin polymer stream during steady-state operation. The applied

control action involves adjusting the feed rates such that the reactor produces a virgin polymer that blends with the introduced off-spec material such that it results in a product that is within spec. The method proved successful in the determination of the required feed rate adjustments for the introduction of off-spec material over a wide range of flowrates. It was found that high flowrates of incoming off-spec material required virgin polymer to be produced that was itself off-spec, even though its blend would be on-spec. A further concern is that a transient period is created when the feed rate adjustments are made. It was found that blending off-spec polymer at a rate of approximately 57% of the virgin polymer production rate, produced both a blend and intermediate product within spec. and increased the production rate of on-spec polymer to over 8 tph.

The second method avoids the issue of creating a transient period by determining the optimal off-spec material flow profile for blending with virgin polymer directly during a grade transition. It was found that depending on the off-spec material properties, it was possible to blend significant quantities of off-spec material whilst generating negligible amounts of new off-spec material. Ultimately, this reduced a standard grade tradition time from 22 hours to a few minutes long, during which 63 tons of off-spec polymer were consumed.

In the final method, the reactor feed rate profiles are manipulated alongside the off-spec material flow profile, to further improve the blending scheme. In this scheme, some off-spec material types that were previously not feasible for blending became worthwhile. Manipulating the reactor feed rates resulted in grade transition time reductions of up to 50% and produced intermediate polymer of almost 900 kg with properties suitable for blending with previously infeasible off-spec polymer types.

The illustrative cases presented in the current study have shown that quantitatively blending off-spec polymer is a venture worth investigating more deeply, due to its potential in reducing large quantities of stored off-spec polymer on site.

.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

$\dot{m}_O$	Stored off-specification polymer mass flowrate	$\left[kg.s^{-1}\right]$
$\dot{m}_R$	Virgin polymer mass flowrate	$\left[kg.s^{-1}\right]$
$\dot{m}_D$	Blended polymer mass flowrate	$\left[kg.s^{-1}\right]$
F_s	Component molar flowrate	$\left[mol.s^{-1}\right]$
N_s	Component holdup	$\left[mol.s^{-1}\right]$
C_s	Component concentration	$\left[mol.m^{-3}\right]$
r_s	Reaction rate	$\left[mol.m^{-3}.s^{-1}\right]$

Components - Subscript (s):

cat	Ziegler-Natta catalyst
TEA	Triethylaluminium (co-catalyst)
M_1	Ethylene $\left(C_2H_4\right)$
M_2	1-butene $\left(C_4H_8\right)$
H_2	Hydrogen
dil	Diluent $\left(C_9H_{18}\right)$

P_n	Concentration of sites with live polymer chains attached of length n	$\left[mol.mol_{Ti}^{-1}\right]$
D_n	Concentration of dead polymer around catalyst, n chains long	$\left[mol.mol_{Ti}^{-1}\right]$
λ_k	k^{th} Instantaneous polymer moment per particle	$\left[mol.mol_{Ti}^{-1}\right]$
μ_k	k^{th} Dead polymer moment per particle	$\left[mol.mol_{Ti}^{-1}\right]$

$\lambda_k^{\,b}$	k^{th} Instantaneous polymer moment in bulk phase	$\left[mol.m^{-3}\right]$
$\mu_k^{\,b}$	k^{th} Dead polymer moment in bulk phase	$\left[mol.m^{-3}\right]$
k_t	Termination rate constant	$\left[m^3.s^{-1}.mol^{-1}\right]$
k_{st}	Site transfer rate constant	$\left[m^3.s^{-1}.mol^{-1}\right]$
k_0	Polymerisation initiation rate constant	$\left[m^3.s^{-1}.mol^{-1}\right]$
k_p	Propagation rate constant	$\left[m^3.s^{-1}.mol^{-1}\right]$
P_a	Concentration of catalyst sites having initial oxidation state	$\left[mol.mol_{Ti}^{\,-1}\right]$
P^*	Concentration of active catalyst sites	$\left[mol.mol_{Ti}^{\,-1}\right]$
P_d	Concentration of inactive catalyst sites	$\left[mol.mol_{Ti}^{\,-1}\right]$
t	Time	$\left[s\right]$
θ	Age	$\left[s\right]$
τ	Mean reactor residence time	$\left[s\right]$
I	Internal residence time distribution	$\left[s^{-1}\right]$
E	External residence time distribution	$\left[s^{-1}\right]$
H_0	Volumetric holdup in reactor	$\left[m^3\right]$
n_{Ti}	Catalyst loading	$\left[mol_{Ti}.g_{cat}^{\,-1}\right]$
ρ	Particle density	$\left[kg.m^{-3}\right]$
ε	Particle voidage	
l	Particle diameter	$\left[m\right]$

F	Step size	
C_R	Crossover probability parameter	
η	Intrinsic viscosity	$\left[dl.g^{-1}\right]$
K	Mark-Houwink parameter 1	
α	Mark-Houwink parameter 2	
t_r	Retention time	$\left[\min\right]$
$S\left(t_r\right)$	Detector signal	
$w\left(t_r\right)$	Molecular weight fraction	

ACKNOWLEDGEMENTS

Several individuals and organisations have made a positive contribution towards the development of this project in various ways. Before we commence, I would like to take the opportunity to acknowledge these contributions in full.

Firstly, I would like to thank my supervisor, Dr Randhir Rawatlal, for persuading me to undertake this research with him. It is through his industrial ties that I was able to work in a field that I enjoy at the top university in Africa on a project that is relevant to industry.

I would like to express my gratitude towards PhD candidate and fellow student John McCoy who played the role of mentor during the early stages of my work. His assistance in developing process models has been invaluable.

Certain aspects of this project are of interest to SAFRIPOL and I would like to thank them for exposing me to the practical issues that polymer industries face, for assisting me with the experimental side of the research and also for funding the project. A special mention is owed to Dr Mike Gradwell for his assistance in that regard.

I was fortunate enough to receive additional funding from both the National Research Foundation and from the Chemical Engineering Department at the University of Cape Town. For that, I am ever grateful.

Thanks to the Department of Polymer Science at the University of Stellenbosch for their analytical laboratory services. It is through them that we were able to obtain GPC data.

Lastly, I would like to thank my friends and family for their continued support during my research as well as everybody who kept me in their prayers.

1 INTRODUCTION

Hydrocarbons have numerous forms, sizes and structures, yet they comprise just two elements, namely hydrogen and carbon. It is the manner in which they are arranged in relation to one another that results in the variety of compounds observed. Technology has reached a stage where many types of hydrocarbons have been synthesised and used in all aspects of life. Some of these possess a single double bond and have a general chemical formula of C_nH_{2n}. They are known as polyolefins and are synthesised by monomer units known as olefins. Polyolefins are manufactured on a large scale, of which the world produced approximately 70 million tons per annum of polyethylene alone in 2008. The throughput is expected to increase to almost 95 million tons by 2015. Polyethylene makes up roughly 25% of the total amount of polymers produced annually (Global Markets Direct, 2009). It is an important commodity from a competitive industry in which new uses are still being discovered due to, in part, continual interest and input from academia.

1.1 History of Ziegler-Natta catalysts

The prospect of producing polyolefins on an industrial scale was realised in the 1950's due to the research of Karl Ziegler (1964). He discovered that high molecular weight polyethylene could be catalytically prepared at moderate temperatures and pressures with mixtures of aluminium tri-alkyls and metal compounds of Groups IV (a) to VI (a) of the Periodic Table with atomic numbers 22 to 74. During this period, Giulio Natta discovered that certain types of Ziegler catalysts (Natta & Pino, 1957) produced crystalline macromolecules from linear olefins, a feat which had never before been accomplished (Natta *et al.*, 1955). These two discoveries opened up a field filled with potential applications that are still being researched today (Busico, 2013).

The first generation of Ziegler-Natta catalysts consisted of titanium (III) chloride $(TiCl_3)$ and triethylaluminium (TEA) in solution. The nature of the catalyst structure and phase has evolved significantly since then. Current industrial catalytic systems are heterogeneous such as Titanium (IV) chloride $(TiCl_4)$ crystals based on Magnesium Chloride $(MgCl_2)$ silica supports activated in the presence of TEA (Malpass, 2010). These systems are more active than the original catalysts and also control more easily the morphological polymer properties such as size, shape and porosity (Muñoz-Escalona *et al.*, 1984).

The basic chemistry behind all Ziegler-Natta catalysed polymerisation reactions is highlighted in Figure 1.1. It is well known that complexes form between TEA molecules and titanium ions. Some of these complexes facilitate chain growth and are referred to as active sites. A monomer may attach itself to the active site, initiating polymeric reactions.

Subsequent monomer additions at the site increase the length of the attached polymer chain. Termination reactions break the bond between the active site and the polymer chain, causing the dead chains to accumulate around the catalyst particle. Hydrogen is a common terminating agent used in polymerisation systems for controlling the product properties. However, monomer and alky-organometallic compounds also participate in chain termination reactions albeit to a lesser extent.

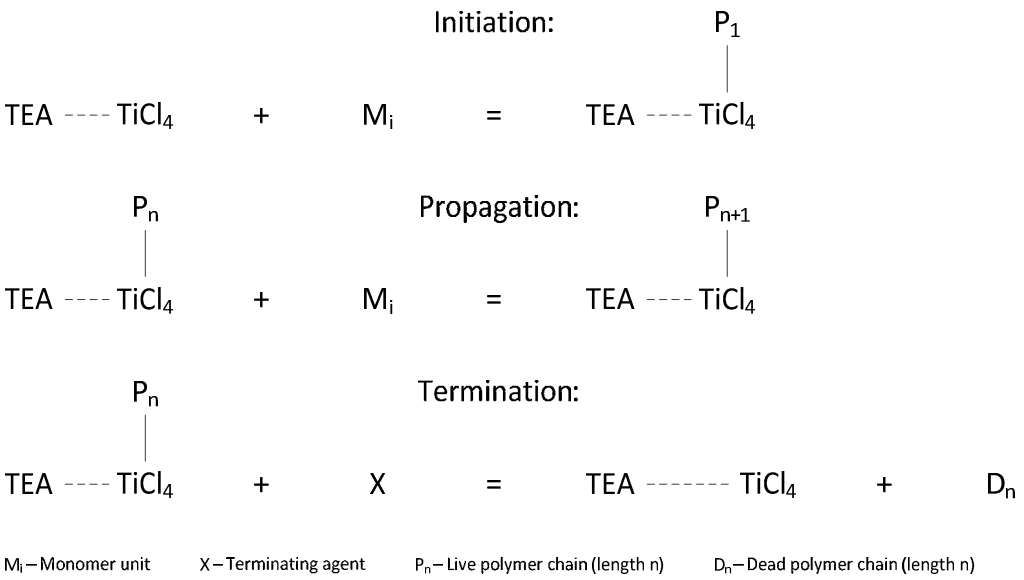

Figure 1.1 - Principle polymerisation reactions

Characterisation of the complex structures that constitute heterogeneous catalyst active site has proven to be a difficult task. Various detailed mechanisms for catalyst activity have been proposed and supported by molecular orbital calculations. The proposal by Cossee (1964) and Arlman (1964) has been widely accepted as the most accurate, yet there is still a large gap in knowledge regarding the details of the alkyl organometallic complex formation at the active site (Corradini *et al.*, 2004).

Henceforth, the supported metal salt is referred to as the catalyst and the organometallic compound is referred to as the co-catalyst. Regardless of the nature of the catalyst active site, polymer chains grow by monomer insertion. A monomer unit always attaches to the molecule previously added to the live chain, thus monomer arrangement is by order of insertion. A feature of Ziegler-Natta catalysts is the ability to produce stereo-regular polymers where the relative orientation of the alkyl groups is either sequentially the same (isotactic) or alternating (syndiotactic). The structure of the complex will always facilitate chain growth along the backbone, the main chain, of the catalyst, thus inhibiting propagation of monomer in orientations that will form long chain branches from the backbone.

Highly active homogeneous catalysts were synthesised in the 1970's that perform equally well as, if not better than heterogeneous catalysts. All polymerisation reactions take place in

the liquid phase. The homogeneous nature of these newer catalysts creates the potential for all the catalyst molecules to be converted to active site complexes, resulting in a higher rate of polymerisation than its predecessor. In contrast, the co-catalyst is restricted to the molecules it can physically gain access to within and around the support structure of a heterogeneous catalyst. An additional advantage of homogeneous catalysis is the ability to produce polymers with greater stereo-specificity. Furthermore, all the catalyst and co-catalyst molecules form the same complex type, a single active site type, resulting in polymers with much narrower distribution curves (Alt & Köppl, 2000). Overall, homogeneous catalysts offer increased productivity and finer control of the polymer properties than traditional heterogeneous catalysts.

Recent research on homogeneous catalysts has given some insight into the complicated activity mechanism of traditional Ziegler-Natta supported catalytic systems (Corradini *et al*., 2004). Despite this breakthrough, there is still a large deficit of knowledge on, and much debate over, the mechanism of active site formation (Cheng *et al*., 2013). Further developments in the field of homogeneous Ziegler-Natta catalysts could provide a better understanding of traditional heterogeneous catalysis for the improvement and more efficient use of current industrial processes that are based on heterogeneous catalyst technology.

1.2 Industrial overview of slurry-phase polymerisation

Subjecting ethylene to active Ziegler-Natta catalytic systems, consisting of the catalyst and the co-catalyst, results in the production of polyethylene with linear chains. Strong intra-molecular forces are observed in the bulk phase due to chain linearity, thus the product has both a high density and a high specific strength. This type of polyethylene is referred to as high density polyethylene (HDPE).

Commercial polymerisation of HDPE is performed either in the gas-phase or slurry-phase. There are significant advantages to gas-phase polymerisation over slurry-based processes (Jejelowo *et al*., 1991) and most new plants base their technology on the former mode of operation. However, the majority of existing industries operate with slurry-based reactors.

With slurry-phase polymerisation, a suitable diluent is used to precipitate polyethylene particles in the reactor at a given operating temperature (Malpass, 2010). The solid catalyst and liquid co-catalyst in a diluent are fed to a series of reactors whilst the gas-phase components, namely monomer, comonomer and terminating agent, are bubbled through the slurry, as illustrated in Figure 1.2, to create a multi-phase and multi-component system. Oxygen and water inhibit the formation of active sites, thus the catalyst is maintained in a nitrogen atmosphere to prevent catalyst poisoning. Reactors are commonly operated at temperatures between 80-150°C and pressures between 21-34 bar with residence times of approximately 2-3 hours (Farauto, 2007). The polymerisation process is exothermic, thus

cooling coils are placed around the reactor columns to maintain isothermal operation. Recycle gas is condensed and then re-introduced to the system, upon which it vaporises.

Within the reactor vessel, the activation of the catalyst by the co-catalyst allows monomer that is dissolved in the diluent to attach itself to an active site. At this point, chain propagation and chain termination occur as a set of series-parallel reactions. Monomer and comonomer units incorporate themselves into the chains at varying rates. These chains are not soluble in the liquid medium; therefore, solid particles are formed around the catalyst. Bonds between active sites and chains are continuously terminated mainly due to the presence of hydrogen in the vessel.

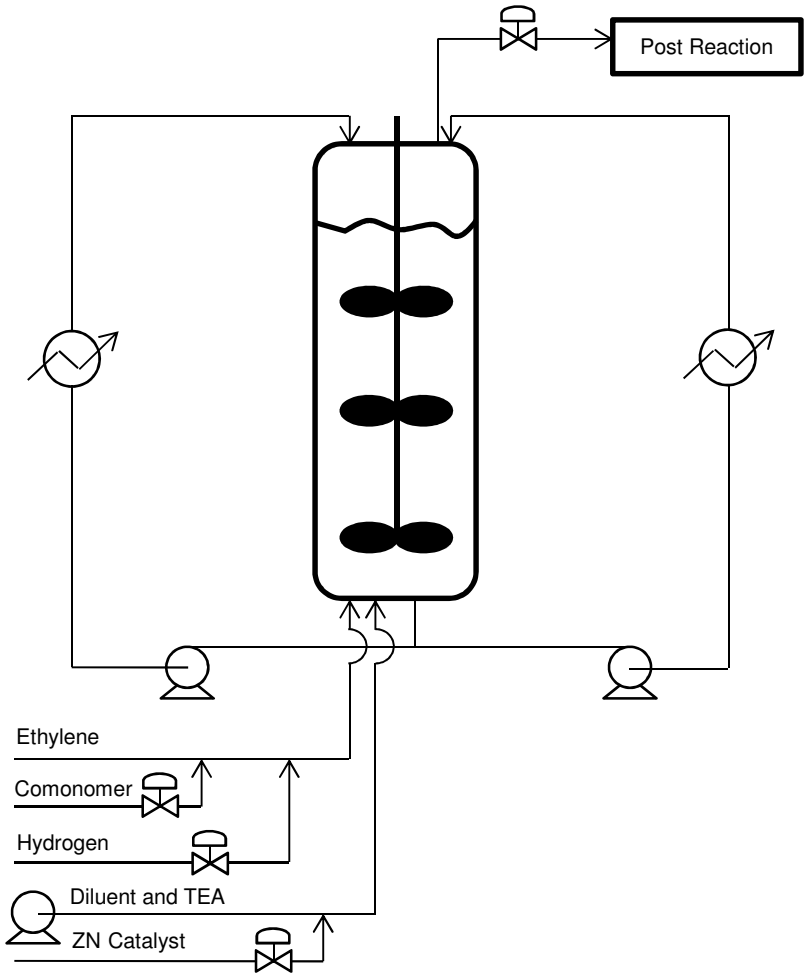

Figure 1.2 - Slurry-phase reactor setup for polyethylene with comonomer

Catalyst structures are often designed to influence the particles' size distribution, growth and shape in a specific manner (Cecchin *et al.*, 2000). The polymer inherits the shape of the particle as the bulk layers grow around it. Often, the rate of polymerisation is high enough to induce support structure fragmentation, which exposes additional potential active sites on the catalyst. This can be beneficial in contributing towards an increased productivity. On the other hand, catalyst fragmentation may lead to the production of undesirable fines, as the polymer layer that forms around the fragments are too small. As a preventative measure, the

catalyst is often treated with the co-catalyst and reactants in a solvent under mild conditions prior to being introduced to the reactor. This is a practice known as “pre-polymerisation” and is also performed to provide insulation around the catalyst particles, thus reducing the heat sensitivity in the initial stages (Pater *et al.*, 2002).

Polymer product is extracted from the continuous reactor train as a powder. This intermediate product is prepared in batches for post-processing operations such as injection moulding, blow moulding or extrusion. Figure 1.3 illustrates a typical layout of the post-reactor section of a plant. Part of the preparation includes mixing stabilisers and other additives with the powder to prevent further chain growth from occurring and to counter chain length degradation. To ensure homogeneity of the polymer product, the powder is mechanically mixed prior to post-processing in a blending stage.

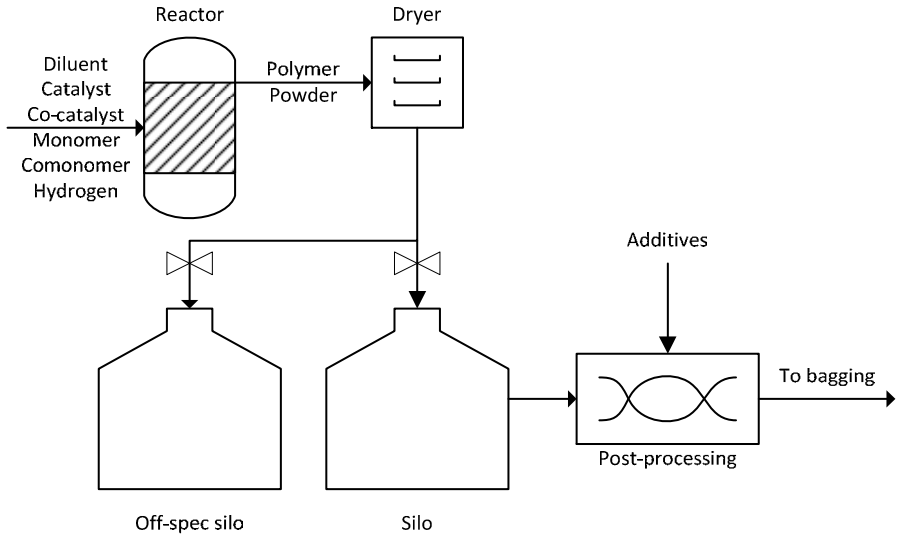

Figure 1.3 - Schematic of general industrial slurry-phase polymerisation system

1.3 High density polyethylene properties

For a specific industrial setup, reactant ratios, feed flowrates, as well as system temperature and pressure influence the nature of the polymer. For example, operating at a higher temperature may increase the reaction rates, leading to an increased productivity. However, due to the exothermic nature of the process, the temperature of the system could exceed the softening temperature of the precipitate. This is the point where a crystalline or semi-crystalline substance displays molten-like properties. In the case of slurry-phase polymerisation, it is undesirable for the precipitate to reach this state as it leads to particle agglomeration, which in turn causes particles to settle in the reactor (Dotson *et al.*, 1995).

The properties of a polymer product are characterised into physical, thermal, chemical, mechanical and rheological properties as shown in Table 1-1 (Kiparissides, 2006). Analysis of a polymer product will show that a particle comprises of a number of molecules with

varying chain lengths. Depending on the combination of feed rates and system conditions, polymer is produced with a specific chain length distribution (CLD) due to the effect of these inputs on the series-parallel nature of the initiation, propagation and termination reactions (Flory, 1953). An example of a typical CLD curve is shown in Figure 1.4. using parameters given by Holland and Anthony (1979). The CLD represents the probability of molecules with various lengths that exist in the product. The molecular weight distribution (MWD) curve is closely related to the CLD for homopolymers since the number of monomer units in a chain correlates to the chain's weight by the monomer molar mass. In the case of copolymerisation, chains of a particular length may have different weights depending on the comonomer composition distribution (CCD), since each monomer type has a different molar mass.

The MFI is an indirect measure of a polymer's molecular weight and it is based on the ease of flow according to a standard procedure. The MFI is defined as the mass of polymer that flows through a capillary of a specified diameter and length in 10 minutes under a prescribed temperature and pressure. A comparison of the MFI under various conditions indicates the spread of the MWD.

Table 1-1: Measure of polymer product quality

Molecular and morphological properties

- Average molecular weights and molecular weight distribution (MWD)
- Copolymer composition distribution (CCD)
- Sequence length distribution (SLD)
- Long-chain branching distribution (LCBD)
- Stereoregularity (tacticity)
- Particle size distribution (PSD)
- Particle porosity and surface area, etc.

End-use properties

- Physical and thermal properties (e.g., density, clarity, melting point), temperature stability, swellability, plasticiser uptake, etc.)
- Chemical (e.g., corrosion resistance, etc.)
- Mechanical properties (e.g., strength, toughness, stress crack resistance, abrasion resistance, impact resistance, etc.)
- Rheological properties (e.g., flow properties, shear viscosity and elasticity, melt index, extensional viscosity, extrudate swell, etc.)

HDPE is manufactured with specifications suitable for a wide range of applications. Generally, In terms of end-use properties, HDPE products are known for their excellent stiffness, high environmental stress crack resistance, high chemical resistance and low water absorption (Vasile & Pascue, 2005). HDPE is much harder and more rigid than the highly branched low density polyethylene (LDPE) formed by free-radical polymerisation. Ethylene is sometimes polymerised with longer chain olefins, such as 1-butene or propylene as a comonomer, in relatively small amounts to produce a copolymer with comparatively superior properties to LDPE. These improvements translate to higher environmental stress crack resistance and increased impact strength (Ram, 1997).

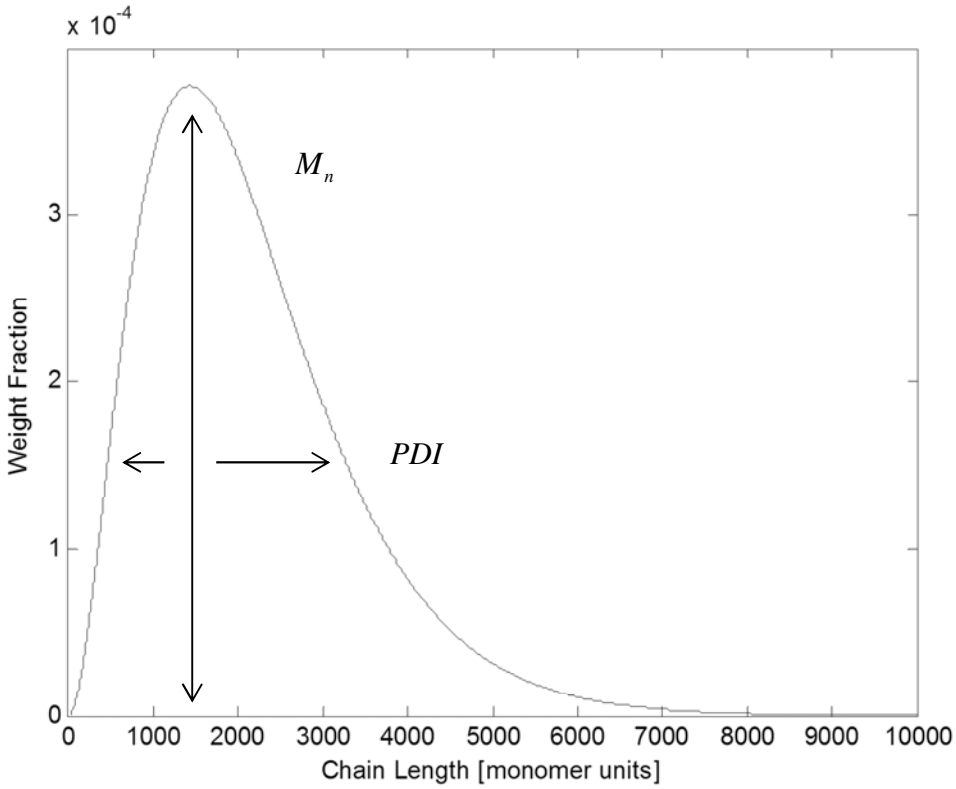

Figure 1.4 - Polymer product chain length distribution curve

Often, the orientation of a longer olefin comonomer in ethylene copolymerisation is such that short chain branches are formed along the polymer backbone. This is known as short chain branching and it disrupts the linearity of chains that would usually be produced under homopolymerisation. The outcome is a product that it less brittle than HDPE. Furthermore, the resulting increase in short chain branches (SCB) increases the intra-molecular forces such that the polymer density is lower than that of HDPE. Additionally, the high frequency of SCB allows the chains to glide alongside each other when subjected to a tensile force, rather than become entangled, as is the case with long chain branched LDPE. This product is referred to as linear low density polyethylene (LLDPE) and can offer similar properties to that of LDPE, but at a reduced film thickness (Vasile & Pascue, 2005).

More generally, polyethylene is categorised as a thermoplastic, which melts upon heating and solidifies upon cooling. Thus, in the molten state it is easily processed and machined into various shapes and sizes. The main applications of processed HDPE include blow-moulded containers, crates, drums, blown films and pipes. It is also ideally suited to food contact applications and has even found significant applications in the medical field (Nath *et al.*, 2009).

1.4 Polymer grades and off-spec generation

The applicability of a polymer product to suit a particular need is dependent on its end-use properties as requested by a client. It is common practice for plants to manufacture a range of polymer products to meet the requirements of many clients with different needs. These products are created by varying reactant ratios, feed rates and the system's temperature and pressure and are known as polymer grades. They are characterised by properties such as molecular weight, molecular weight distribution, melt flow index (MFI), copolymer composition and density (Osswald & Hernández-Ortiz, 2006).

Plant operation over a period of a month may involve as many as four changes in process conditions to satisfy the market demand for various grades. During the transition periods that arise by changing these conditions, large quantities of off-specification (off-spec) polymer are produced. The off-spec material is of little value. Some plants store the waste material for long periods of time and eventually sell it at largely discounted rates (Ram, 1997). Other plants extrude the off-spec material into products where the homogeneity of the properties is not important. One such example is that of plastic lumber products that can be made into outdoor furniture (Robeson, 2007). Regardless of the efforts to utilise it, the generation of off-spec material represents an inefficient use of raw material and an additional cost in terms of storage and disposal or sale at large discounts.

1.5 Grade transition optimisation

Existing plants continuously seek ways of optimising their processes by improving product quality or increasing productivity and operational efficiency. The grade transition procedure is a key area in which the process can be optimised. Either the transition time or the amount of off-spec polymer generated during the transition can be minimised by altering the dynamics of the polymerisation process. This is done by manipulating system inputs, such as feed rates, over time. Plant operators usually devise grade transition strategies based on years of experience. Dynamic optimisation can be used as a tool to find a sequence of events in order to better optimise the grade transition. A general review of the application of dynamic optimisation to optimal grade transitions of polymerisation processes is given by Congalidis and Richards (1998). An unsteady-state process model is an integral part of any dynamic optimisation problem. The model must be able to accurately relate system inputs such as

process conditions and feed rates to system outputs such as production rates and polymer properties at low computational expense in order for a solution to be applied in practice.

1.6 Polymerisation modelling

As previously mentioned, polymerisation models are useful for optimising plant processes such as grade transitions. The challenge in modelling polymer processes is to quantitatively relate abstract end-use properties to quantifiable molecular and morphological properties as highlighted in Table 1-1. These intermediate properties, shown in the table, can be predicted using fundamental principles or using empirical models. The latter type is generally easy to develop but is only accurate over a narrow range of system inputs. However, an optimal grade transition may require system inputs to be manipulated over a wide range. As such, a process model that is based on fundamental principles is much more suitable for dynamic optimisation problems since the region over which the model can be trusted is far greater than that of empirical models.

Polymerisation is inherently a complex process; therefore, it is complicated to develop a simulation from first principles. Ziegler-Natta technology has been implemented in industry since the 1950's. However, the understanding of the process is still very limited (Malpass, 2010). New types of catalysts have already been developed, such as metallocene catalysts (Alt & Köppl, 2000) and post-metallocene catalysts (Gibson & Spitzmesser, 2003) that allow for stricter control of the polymer properties. Both of these are homogenous single-site catalysts that differ depending with various structures and active metal centres.

Despite the promise these newer catalyst types hold, it is still important to research systems that use heterogeneous Ziegler-Natta catalysts due to the potential improvements that can still be implemented industrially (Boero *et al.*, 2001). In contrast with the newer catalyst types, the exact structure of the active sites on a support structure catalyst is unknown. Various attempts to model the performance of the catalyst have been made and met with some degree of success. These will be discussed further. The catalyst activity mechanism and the reaction mechanism at the active site form the basis of a Ziegler-Natta polymerisation, thus their characterisation is very important for accurately determining the polymer properties.

One of the most relevant properties for defining a polymer grade is the CLD curve. Polymerisation models often represent the CLD by average parameters, rather than having to compute a distribution of thousands of polymer chains. The number average chain length $\left(M_n\right)$ and the polydispersity index $\left(PDI\right)$ generally provide enough information about the distribution curve at a reduced computational cost. Figure 1.4 illustrates how these two parameters relate to the CLD. The kinetic mechanism used to generate the curve is from

Holland and Anthony (1979). The M_n represents the mean of the distribution, whereas PDI is an indication of the spread of the distribution curve about the mean.

For slurry-phase reactors, the reactants are introduced in gas, solid and liquid phases. However, polymerisation reactions take place in solution. It is thus necessary to accurately model the vapour-liquid equilibrium interface to determine the correct reactant concentrations in each phase. The Peng-Robinson equation of state has proven sufficient to describe the thermodynamics of this type of system (Sandler, 2006).

There have been numerous studies towards the development of universal steady-state polymerisation models. Significant progress was made in the development of models describing the polymerisation system response to unsteady-state behaviour (McKenna & Soares, 2001). More specifically, computationally efficient dynamic polymerisation models are being developed for the purpose of model predictive control (McCoy, Rawatlal, & Soares, 2012) with the ultimate goal being industrial and commercial applicability (Zheng *et al.*, 2011). The ability to accurately predict the end-use properties of a polymer will allow for finer quality control. In general, the polymerisation process is described by a system of highly non-linear algebraic or differential equations for steady-state and unsteady-state behaviour, respectively. Solving these systems usually comes at great computational expense. Additionally, the solution space could be multi-modal. Thus, depending on the method of solution, the risk of convergence to only a local minimum could be high. Greater opportunities for global optimisation of these types of processes are presenting themselves due to recent advances in solution techniques and computing.

1.7 Off-spec polymer blending

Online measurements of the reactor feed rates, exit liquid concentrations and system temperatures on a plant are becoming more widespread in practice, are easier to obtain and can be supplied on a continuous basis. In contrast, the polymer product can only be characterised at discrete intervals because characterisation of the powder needs to be performed in the laboratory.

With the field of polymerisation modelling approaching a stage where the polymer properties can be accurately predicted much faster than they can be determined via real-world analysis, it is becoming feasible to apply model predictive control over the process. Indeed, a large body of research is aimed towards using advanced control strategies for the reduction of grade transition times, using complex dynamic polymerisation models.

While grade transition optimisation serves to reduce the amount of off-spec polymer produced during grade changes, its generation is still unavoidable. As such, there will always be a mass of off-spec polymer on a plant that will have to be dealt with. Usually off-spec polymer is sold at discounted rates, but there are also ways to utilise it in more efficient

manners. For example, off-spec polymer can be blended with a polymer of the same type, having properties sufficiently achieved such that the properties of the blend are on-spec, thus rendering it saleable at the standard commercial rates. This has been done extensively for both butyl and ethylene-propylene products on an empirical basis (Mantia, 2002).

It could be the case that the same techniques applied in grade transition optimisation may be used to optimise the blending process of off-spec polymer with virgin polymer from the reactor to form an on-spec product. This could provide an additional solution to countering the formation of off-spec material, therefore it is deemed worth investigating.

1.8 Project goals

The generation of off-spec polymer during grade transitions has been identified as an unavoidable cost on any polymer plant. Measures can be taken to reduce the cost to the plant. These measures include off-spec blending and grade transition optimisation.

In the current study, the prospect of blending off-spec polymer quantitatively using techniques that are applied in grade transition optimisation studies is investigated. The results of the investigation are presented through case studies of a plant where ethylene and 1-butene are co-polymerised under Ziegler-Natta catalysis. An industrial-scale model of this system has been developed by McCoy (2012) and Rawatlal (2004) based on fundamental principles. A simplified version of this model is developed in this study and combined with a polymer blending model, developed for the first time in this study. Various optimisation algorithms will be developed and applied to the process model for steady-state and unsteady-state blending to determine optimal flowrates for blending. Based on the findings of the case studies, the practice of off-spec blending described in the current study will be deemed worthy or unworthy of further investigation.

2 LITERATURE REVIEW

Large quantities of the polymer generated during grade changes are off-spec due to process dynamics influencing the polymer properties. These grade changes are necessary to supply products to multiple clients with different needs and to meet various market demands. However, the generation of off-spec polymer as a result of inducing grade changes is seen as an expense in terms of a production time loss of on-spec polymer, replaced by a product of a much lower value. To offset this expense, efforts have been made to reduce the production of off-spec polymer and to utilise the off-spec material such that its value is as high as possible. In the current chapter, these efforts are reviewed in detail.

More specifically, the properties relevant to defining a polymer grade are discussed within the context of predicting those properties from process models. These polymerisation models are particularly useful for grade transition optimisation problems, which align with the goal of the first effort to reduce the production of off-spec polymer. A consolidation of the various methods that have been applied in optimal grade transitions is presented. Thereafter, the research regarding the uses of off-spec polymer as a means towards a valuable product is reviewed within the context of optimal control of the processes.

2.1 Modelling polymerisation systems

It has already been established in the previous chapter that Ziegler-Natta catalysed polyethylene products are highly linear, therefore phenomenon such as chain branching are not characteristic of the products. Thus, the chain length distribution (CLD) curve is most influential on the polymer properties. As a result, it can be said that the shape of the CLD curve defines a polymer grade and it is a function of the process conditions and reactor feed flowrates.

It is widely publicised in literature that there are three levels of detail to consider in modelling industrial polymerisation systems, namely micro-scale, meso-scale and macro-scale modelling (Ray ,1991). This section focuses on predicting the properties regarding the chain length distribution curve based on first principles at each of the three levels of detail. A comprehensive review was performed by McKenna and Soares (2001) that details the large amount of work in modelling olefin polymerisation on supported catalysts.

2.1.1 Micro-scale modelling

The first level of detail is on the micro-scale. At this level, an understanding of the molecular interactions is most important. Polymers formed under Ziegler-Natta catalysis generally exhibit broad molecular weight distributions. It is generally accepted that the cause of this observance is due to the existence of multiple active sites on the catalyst structure (Hutchinson *et al.*, 1992). Each active site type has its own set of elementary reactions with a

unique set of kinetic rates. Some studies have related the activity of a site to the oxidation state of the titanium molecule at that active centre, where it is believed that only certain oxidation states are active (Soga *et al*., 1982; Hutchinson *et al*., 1992; Han-Adebekun and Ray, 1997; Bahri-Laleh *et al*., 2011). An alternate theory was proposed by McCoy and Rawatlal (2011) where the observed broad molecular weight distribution is due to the presence of pseudo-sites. These pseudo-sites create a large range of polymer chains similar to the multi-sites theory, except it is the termination rates that differ from site to site instead of the propagation rates. This difference is due to the various terminating agents in the system.

Every polymerisation model requires a kinetics scheme to determine the reactions that take place at the active site and to determine the rates at which they occur. These rates are largely dependent on the concentration of the reactants near the active sites. Given a set of known elementary reaction steps, a number balance can be applied for chains of different lengths (McAuley *et al*., 1990). The result of these balances will yield a distribution of chain lengths due to the series-parallel nature of the reactions. In order to perform the balances for a specific system, the rate constants for the kinetic scheme must be determined. They can be determined via regression of data from experiments performed in a laboratory.

2.1.2 Meso-scale modelling

Developing the polymerisation model at the micro-scale level only yields the properties of live polymer chains, i.e. polymer chains still attached to the active site. In terms of meso-scale model development, transport phenomena, micro-mixing, polymer particle size distribution and particle morphology (Kiparissides, 1996) play an important role in determining the properties of dead polymer, i.e. polymer chains accumulated around a catalyst particle by chain termination. At this level, the number of polymer chains grow around a catalyst fragment, thus a growing particle is formed. A monomer unit has to diffuse through the catalyst structure in order to reach an active centre where chain growth can occur. Eventually, monomer units also have to diffuse through a layer of dead polymer chains surrounding the catalyst structure. Chain growth within the pores can occur rapidly, causing the structure to rupture into smaller fragments (Webb *et al*., 1989). This is desirable as it subjects the bulk phase to more active sites, promoting polymerisation (Webb *et al*., 1991). Overall, polymerisation is a highly exothermic process, thus the occurrence of hotspots can become an issue in the sense that exceeding the glass transition temperature will result in the agglomeration of polymer “melts”. Concentration gradients and temperature profiles need to be investigated to determine whether the impact will be significant. The well-developed multi-grain model has been used extensively to predict mass and heat transfer limitations based on the phenomena of fragmentation (Hutchinson *et al*., 1992). It has also been argued that the effects of heat and mass transfer are negligible and that the use of catalyst sub-sites and pseudo-sites models is sufficient to predict the polymer properties.

The major outcome of the model at this level of detail would be the chain length distribution on a particle basis.

2.1.3 Macro-scale modelling

Finally, the macro-scale level involves incorporating macro-mixing phenomena in the reactor, the overall mass and energy balances, particle population balances, the heat and mass transport phenomena as well as reactor dynamics into the models and determining control strategies (Kiparissides, 1996). Once the above-mentioned phenomena have been factored into the model, it should be able to determine the overall chain length distribution of the polymer in the particle bed or at the exit of the reactor.

2.1.4 Average polymer properties

There are many successful steady-state models that incorporate the concepts from all three levels of detail. Only a few simulations have been developed that predict the properties under unsteady-state conditions in the reactor. These models use either extended Kalman filters (Penlidis *et al*., 1985; Dimitratos *et al*., 1989) or the population balance model (Sayer *et al*., 1997; Prasetya *et al*., 1999; Yiannoulakis *et al*., 2001; Immanuel and Doyle III, 2003) to estimate particle properties. The latter approach is more fundamental, but a solution comes at great computational expense. Recently, progress has been made in the development of an unsteady-state polymerisation model for Ziegler-Natta catalysis using the segregation method (McCoy *et al*, 2012). It is a far more computationally efficient model that should be suitable for real-time control.

With population balance models, the number of chains of all lengths are accounted for at all times from the micro-scale level to the macro-scale level. In contrast, with the segregation approach models, the average properties of the CLD are already computed at the single particle level. Thus, only these average properties, namely M_n and PDI are accounted for in subsequent calculations.

These parameters quantitatively describe the mean and spread of the CLD. Further, the CLD greatly influences the properties of a polyethylene grade produced under Ziegler-Natta catalysis. Therefore, it follows that M_n and PDI are sufficient to characterise a grade for the purpose of distinguishing between on-spec and off-spec.

Based on the literature cited in the current section, it can be deduced that an unsteady-state process model for the Ziegler-Natta catalysed polymerisation of ethylene can be developed from first principles that can quantitatively predict the properties relevant to defining a grade. In addition, the computational expense associated with such a model can be low enough such that it is suitable for dynamic optimisation techniques which are often iterative by nature.

2.2 Dynamic optimisation methods

In general, a dynamic optimisation method is applied to a problem in which a specific outcome of a dynamic process is sought.

Biegler (1992) describes three methods for solving optimal control problems of dynamic processes.

1. Iterative methods based on variational calculus and Pontryagin's maximum/minimum principle.
2. Sequential non-linear programming (NLP).
3. Simultaneous dynamic optimisation (SDP).

A general overview of each method is given in the current section as a short review of the potential tools for the purpose of optimal control of a polymerisation plant.

2.2.1 Iterative methods

The calculus of variations is a branch of mathematics whereby the extreme value of a functional, an operation that maps a vector of variables to a real value, is determined. A typical example of a problem that can be solved using the calculus of variations would be to find the time profile of x that minimised the integral of a function of x over that time period subject to certain constraints. The extension of the calculus of variations to optimise physical processes is called optimal control theory. Process variables are divided into two classes, namely control variables vector $u(t)$ and state variables vector $x(t)$. Profiles for the control variables are sought that minimise a determined performance index or objective function. For further reading on the topic, the book by Kirk (2004) is recommended. More formally, the aim of optimal control is to find the control input vector $u(t)$ over $t \epsilon \left[t_0, t_f \right]$ that minimises the objective function $J\left(u\left(t\right)\right)$, typically of the form:

$$J(u) = h(x(t_f), t_f) + \int_{t_0}^{t_f} g(x(t), u(t))\, dt \tag{2.1}$$

$h(x(t_f), t_f)$ is the performance measurement at final time, t_f, whereas $g(x(t), u(t))$ is a time-varying measurement to be integrated over the entire time span.

The process is described by a set of differentials of the form shown in Equation (2.2):

$$\frac{dx}{dt} = a\left(x\left(t\right), u\left(t\right), t\right) \tag{2.2}$$

A convenient term that is used in solving the problem is called the Hamiltonian (H) and it is defined as:

$$H(x(t),u(t),p(t),t) \triangleq g(x(t),u(t),t) + p^T(t)\left[a(x(t),u(t),t)\right] \tag{2.3}$$

where $p(t)$ is the vector of co-state variables used to assist in finding the optimal solution.

The necessary conditions for optimality are thus defined as:

$$\frac{dx^*}{dt} = \frac{dH}{dp}(x^*(t),u^*(t),p^*(t),t) \tag{2.4}$$

$$\frac{dp^*}{dt} = -\frac{dH}{dx}(x^*(t),u^*(t),p^*(t),t) \tag{2.5}$$

$$0 = \frac{dH}{du}(x^*(t),u^*(t),p^*(t),t) \tag{2.6}$$

and

$$\begin{aligned}&\left[\frac{dh}{dx}(x^*(t_f),t_f) - p^*(t_f)\right]^T \delta x_f \\ &+\left[H(x^*(t_f),u^*(t_f),p^*(t_f),t_f) + \frac{dh}{dt}(x^*(t_f),t_f)\right]\delta t_f = 0\end{aligned} \tag{2.7}$$

All variables denoted with an asterisk are the optimal solutions.

Often there are physical constraints on the control variables, thus it is useful to have set bounds for them.

$$\underline{u}_i \le u_i(t) \le \overline{u}_i \quad i = 1,2,...m \tag{2.8}$$

$\underline{u}_i$ and $\overline{u}_i$ are the minimum and maximum bounds, respectively, of the control variable u_i. t_0 is the time at which the grade transition commences and t_f is generally a value chosen sufficiently large enough to allow the system to reach steady-state. The function $g(x(t),u(t),t)$ usually represents the deviation of the current properties from the desired properties and $h(x(t_f),t_f)$ is usually a process design specification to be met at steady-state.

If the control variable is bound, which is often the case in real-world systems, then Pontryagin's minimum principle (Pontryagin, 1962) is used to find the best possible solution.

It states that:

$$H(x^*(t), u^*(t), p^*(t), t) \leq H(x^*(t), u(t), p^*(t), t) \tag{2.9}$$

Thus, the problem is reformulated for the Hamiltonian to be minimised (Bryson & Ho, 1975).

2.2.1.1 Method of steepest descent

The method of steepest descent is a common iterative method that is applicable when the initial values of the state variables are all known as well as the time range over which the system is to be controlled. In this method, the control vector is discretised into a series of step functions over the known time range. The sets of differential equations describing the state and co-state variables are integrated so that the solutions can be used to evaluate the Hamiltonian and $\frac{dH}{du}$ terms. This process is repeated after adjusting the control vector profile until the Hamiltonian and its derivative with respect to the control vector profile are minimised to zero.

2.2.2 Control vector parameterisation

By expressing the control variable as a series of functions, polynomials, splines or similar functions, the dynamic optimisation problem becomes a NLP problem that can be solved using any standard optimisation algorithm (Biegler, 1992). This is known as control vector parameterisation (CVP) (Ray, 1981) and an overview of the optimisation procedure is shown in Figure 2.1. This approach has the main advantage of being able to reduce the problem to a set of parameters that approximate the time-varying control variable profile. A disadvantage of this method is in the difficulty of imposing constraints on the state variables.

Briefly, the procedure is as follows. Initial sets of parameters are created that define the control variable profiles. Thereafter the process model is integrated using the initial control variable profile, followed by evaluation of the objective function based on the outputs of the process model. The chosen optimisation algorithm then determines if the objective function value meets specified criteria. If not, the control variables profiles are adjusted by choosing new sets of parameters, prompting a continuation of the cycle.

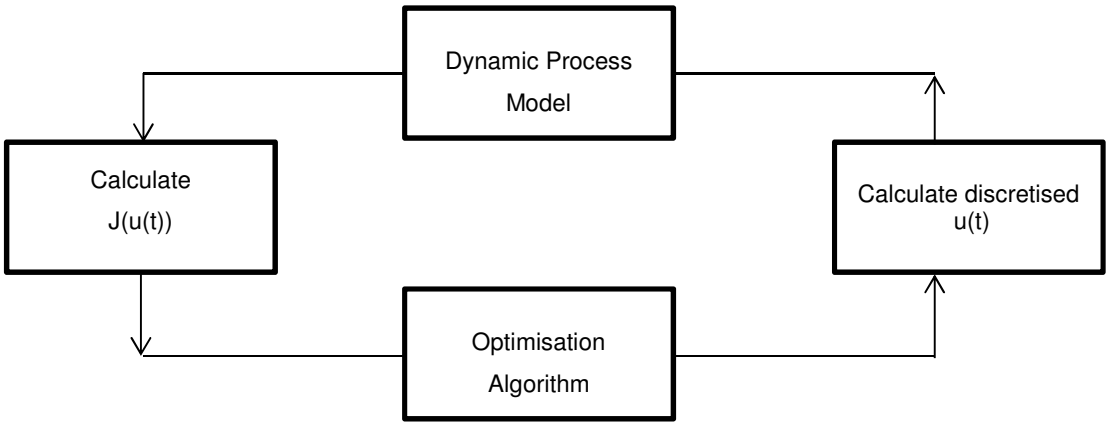

Figure 2.1 - Control vector parameterisation approach

Many different algorithms have been used in optimal grade transition studies and they have been applied to a variety of polymerisation systems. Further, numerous means of parameterising the control vector profiles have been presented. In most cases, the objective function has been posed as a square of the difference between a time-varying property and its desired state (weighted deviations). Equation (2.10) shows a general objective function that includes time-varying weighted deviations as well as final time weighted deviations:

$$J = w_1\left(x_1\left(t_f\right) - x_1{}^d\right)^2 + \int_{t_0}^{t_f} w_2\left(x_2\left(t\right) - x_2{}^d\right)^2 + w_3\left(x_3\left(t\right) - x_3{}^d\right)^2 dt \qquad (2.10)$$

Where weighting, w_i, corresponds to property, x_i, and its desired value, $x_i{}^d$.

2.2.2.1 Sequential quadratic programming

Sequential quadratic programming (SQP) is a very popular method for solving non-linear constrained optimisation problems that are not defined by a single algorithm, but rather by a conceptual procedure. Many of the algorithms that have evolved from this concept have been incorporated into commercial optimisation packages (Boggs & Tolle, 1995).

2.2.2.2 Genetic algorithms

A genetic algorithm is a search technique that mimics the process of natural evolution. Five basic components are summarised by Michalewicz (1996).

1. A genetic representation of the solution to the problem.
2. A way to create an initial population of solutions.
3. An evalution function rating solutions in terms of their fitness.
4. Genetic operators that alter the genetic composition of children during reproduction.
5. Values for the parameters of genetic algorithms.

The genetic algorithm maintains a fixed population of potential solutions with every generation. Each population member is evaluated to give some measure of its fitness in relation to the other members. Some of these members undergo transformations mimicking genetic behaviour, such as mutations and crossovers. A mutation operation creates a new individual by asserting some change to a single species, whereas a crossover operation creates a new individual by combining properties from two individuals. New individuals, termed offspring, obtain their properties from the parent population. The offspring members are also evaluated to measure their fitness such that the following generation is populated with the best individuals from both the parent and offspring pools. After several generations, the algorithm should converge towards the best individual that represents the global optimum (Gen & Cheng, 2000).

2.2.2.3 Differential evolution algorithms

When a system is highly non-linear and the objective function is non-differentiable, then direct search methods become preferable for solving an optimisation problem. This is often the case for problems involving polymerisation models.

Usually, a direct search method, such as the well-known Nelder-Mead simplex algorithm, is used to solve for input trajectories. This particular algorithm is most often found in numerical software packages. Generally, it can lead to much faster convergence than other methods. However, the algorithm often results in solutions at local extrema i.e. global convergence is not guaranteed (Lewis *et al*., 2000). A technique that guarantees global convergence is that of the differential evolution (DE) algorithm. In the case of minimising an objective function, a DE algorithm is a parallel direct search method; of which the advantages are listed (Storn & Price, 1997):

1. Ability to handle non-differential, non-linear and multi-modal cost functions.

2. Parallelisability to cope with computation intensive functions.

3. Ease of use, i.e. few control variables to steer the minimisation. These variables should also be robust and easy to choose.

4. Good convergence properties, i.e. consistent convergence to the global minimum in consecutive independent trials.

DE algorithms follow the same philosophy as GA in the sense that both mimic natural evolution. GA encodes parameters as bit-strings and modifies them with logical operators, whereas DE uses real floating-point numbers and performs standard arithmetic operations on them, which are less time consuming.

Similar to GA, a population of potential solutions is initiated using uniformly distributed random numbers. A crossover probability parameter, C_R, is compared to the random number

generated between 0 and 1. This is done for each variable in a population member. If the random number is less than or equal to the specified crossover probability parameter, then that decision variable in the vector undergoes mutation. Otherwise, the parent parameter is inherited. The classic DE approach defines mutation as the sum of a weighted vector difference between two randomly selected population members and a third randomly selected member. The weight factor, F ,is a control factor chosen to be less than 1 (Price *et al.*, 2005).

$$
\begin{aligned}
&v_{j,i,g} = \begin{cases} z_{j,r0,g} + F\left(z_{j,r1,g} - z_{j,r2,g}\right), \textit{if } \left(rand_j(0,1) \le C_R \textit{ or } j = j_{rand}\right), \\ z_{j,i,g} \end{cases} \\
&j = 0,1,...,D-1; \quad j_{rand} \in \{0,1,...,D-1\} \\
&i = 0,1,...N_P - 1 \\
&g = 0,1,...g_{\max} \\
&r_0, r_1, r_2 \in \{0,1,...,N_P - 1\}, \quad r_0 \ne r_1 \ne r_2 \ne i \\
&z_{i,g+1} = \begin{cases} v_{i,g} \textit{ if } f(v_{i,g}) \le f(z_{i,g}) \\ z_{i,g} \textit{ otherwise} \end{cases}
\end{aligned}
\tag{2.11}
$$

Equation (2.11) summarises the mutation where parent member $z_{i,g}$ is the j^{th} parameter of a D dimensional vector in a population of N_P members in the g^{th} generation.

Two vector populations of potential solutions are always maintained during the optimisation process, one for parent members and one for offspring members that are formed by mutation and/or crossover of parent members. The objective function is evaluated by the process model for each member in both populations in order to determine the performance of the off-spring against its parent. Thus, a time period exists between each competition state where only objective function evaluations are performed. By evaluating each member in parallel, either on multiple cores or over a network of workstations, the timeframe for the objective function evaluation state can be decreased. After a number of generations, the algorithm should converge to a solution of parameters that minimise the given objective function.

DE has shown favourable performance over GA due to the mutation operation using difference vectors. They play the role of search directions in DE as gradient vectors do in deterministic optimisation algorithms. The step function, F , need not be chosen arbitrarily, but generally falls within the range of 0 to 1. A low value is associated with robustness (tendency towards global minimum), whereas a high value is associated with faster convergence. This value can be optimised to find the best trade-off between robustness and convergence rate (Price *et al.*, 2005).

There are, however, problems associated with DE. Stagnation has been known to occur. It is also not uncommon for the solution to converge towards a local minimum if the parameters are not set up efficiently (Lampinen & Zelinka, 2000).

2.2.3 Simultaneous dynamic optimisation

If control variable profiles as well as state variable profiles are parameterised, then the dynamic optimisation problem is reformulated as a system of algebraic equations. The problem can then be solved by simultaneous NLP methods, where it is easy to impose constraints on both state variables, $x(t)$, and control variables, $u(t)$. It is of concern that parameterised state variables in this method could lead to inaccurate representation of the process model (Biegler, 1992).

2.2.4 Closing remarks

The methods outlined above may seem different in execution, but the overall philosophy remains the same - that optimal input trajectories can be solved using a dynamic process model to determine the extreme of a desired objective function. While the classic calculus of variations approach is ideal for calculating the true optimal input trajectories, a solution is only possible if the process model is linear and the set of equations describing the system are independent. Unsteady state polymerisations are generally complex systems and would thus be impossible to solve using this approach.

On the other hand, simultaneous dynamic optimisation (SDO) methods where both the input variables and the output variables are discretised can reduce the problem to a set of algebraic equations that are much easier to solve. However, these methods can lead to severe inaccuracies in the solutions of the unsteady-state model.

The CVP method seems to be the most suitable approach to solving dynamic optimisation problems in the polymerisation field, where only the input vectors are discretised in the same manner that control action is taken in practice. Evolutionary algorithms show the most promise for iteratively evaluating an objective function as they allow for global convergence of solutions.

2.3 Polymer grade transition problem

Dynamic optimisation is a widely researched field (Embirucu *et al*., 1996) and the techniques are often applied in the polymerisation field with the aim of reducing the amount of off-spec material produced during a grade transition or by reducing the grade transition time (McAuley and MacGregor, 1992; Lee *et al*., 1997, 1999; Chakravarthy, 1997; Takeda & Ray, 1999; Wang *et al*., 2000; Cervantes *et al*., 2002; Gisnas, 2003; Kasat *et al*., 2003; Wang & Yang, 2003; Chatzidoukas *et al*., 2003; BenAmor *et al*., 2004; Feather *et al*., 2004; Schlegel & Stockmann, 2005; Bonvin *et al*., 2005; Flores-Tlacuahuac, 2005; Babu & Angira, 2006;

Flores-Tlacuahuac *et al*., 2006; Lo & Ray, 2006; Tetiker & Artel, 2008; Flores-Tlacuahuac & Biegler, 2008; Al-haj Ali, 2010).

Generally, the optimal time varying flow profiles of the feed components to the reactor are determined such that either the amount of off-spec produced during a grade transition is minimised or the transition time itself is minimised. For this purpose, it is required that the polymer be defined by relevant properties so as to quantitatively distinguish between the various grades and that of off-spec material. In addition, a process model is required that relates the dynamic feed rates to these polymer properties. As illustrated in the Section 2.1, the process is highly complicated and is described by a set of dependent differential equations. Despite these complexities, dynamic optimisation techniques that incorporate process models of this nature have been successfully implemented.

A short review of the studies in which the techniques described in the previous section have been implemented is presented below.

2.3.1 Calculus of variations

Early applications of the calculus of variations in optimal control of polymerisation systems were mainly applied in the start-up of batch and semi-batch processes. Tsoukas *et al*. (1982) successfully optimised the temperature profile for semi-batch copolymerisation reactors to produce narrower CCD and MWD curves. The authors noted that previous dynamic optimisation studies used a single objective function posed as the weighted sum of multiple objectives. Often, control action that is taken to minimise one objective affects another objective negatively. Thus, a method was proposed to optimise each objective individually in order to quantify the trade-off between each objective.

The iterative method described in Section 2.2.1.1 was also used by Ponnuswamy *et al* (1987) for temperature and initiator feed rate profiles in batch polymerisation of methyl methacrylate (MMA) in order to minimise the time taken to reach steady-state operation. In this system, the final time is a variable and the objective function is defined as the sum of the integral of the square of the deviations of monomer concentration and 0^{th} moment from their respective desired values over the time range. The system of equations describing the process are only slightly more complicated than that used in the study by Tsoukas *et al*. (1982) , yet a considerable amount of computational effort is required in solving this problem. In general, solving optimal control problems by obtaining the optimal input trajectory using the classical calculus of variations approach is very difficult for highly non-linear systems. Convergence towards a true solution is not guaranteed as there is a strong dependence on the initial guesses for the input parameters.

2.3.2 Control vector parameterisation

The following sections discuss the grade transition optimisation studies for the polymerisation processes using the CVP approach and have been categorised according to the optimisation algorithm that has been implemented.

2.3.2.1 Sequential quadratic programming

McAuley and MacGregor (1992) applied CVP to a polyethylene copolymerisation system by parameterising hydrogen and 1-butene feed rates, reactor temperature set point, gas bleed flow, catalyst feed rate and bed level set point. These control variables are parameterised as a series of ramps. The objective function is in the form of the standard weighted deviation function for the MFI and density of polyethylene formed under gas phase polymerisation. MINOS 5.1, a commercial NLP package was used to determine the optimal control trajectory based on the developed process model and the defined objective function. The authors noted that the feasibility of online implementation is dependent on the ability of the optimisation procedure to adapt to process disturbances and model mismatch. Thus, the system is coupled with a feedback control loop to account for model mismatch and process disturbances.

A study by Takeda and Ray (1999) demonstrated how to minimise the grade transition time for multistage polyolefin reactors with bimodal products. A similar study was conducted by Wang *et al*. (2000) for slurry-phase polyethylene reactors using CVP where SQP was implemented. By coupling the SQP algorithm in MATLAB with a dynamic polyolefin process simulator in POLYRED, optimal grade transition trajectories were successfully established. The use of SQP allowed for state variable constraints to be easily incorporated. Both studies parameterised the control variable profiles as series of step functions where the amplitudes at each time interval were optimised.

On defining an objective function, Takeda and Ray (1999) modified the standard weighted sum of deviation functions slightly by incorporating the concept of a product specification band where the target value is the average of the limits of the band. This suggests that a polymer is on-spec if its property is between the upper and lower limit. However, if the property falls outside of the product band, then the target value is set as the nearest limit of the product band. Deviation functions for trajectories outside the specification band are assigned higher weightings than those for trajectories within the band.

Gisnas *et al*. (2003) approached the formulation of the objective function differently. The first form of the objective function proposed by the authors was simply the minimum time at which all the product properties fall within their respective specification bands and remain therein. The second proposal was developed to explicitly minimise the amount of off-spec generated by integrating the polymer production rate from the initial time to the time the properties enter the product band as described in the first proposal. Both of these forms

were successfully tested for the grade transition optimisation of polyethylene reactors under Ziegler-Natta catalysis. Posing the objective functions in these forms is a more realistic representation of the classification of polymer grades industrially.

Wang and Yang (2003) further extended the work of McAuley and MacGregor (1992) to a two-level optimisation problem. The lower level involves optimising the grade transition time from one grade to another, whereas the higher level involves optimising the grade change order to maximise total production over a specific time period. The authors incorporated MFI in the objective function in order to restrain sharp adjustments of control variables that could lead to runaway temperatures. In this study, the SQP algorithm was used in MATLAB to minimise the objective function.

Like McAuley and MacGregor, Al-haj Ali (2010) supports the implementation of a control loop coupled with dynamic optimisation. The author also argues that to make real-time model-predictive control (MPC) feasible, a detailed process model needs to be simplified for faster evaluation. It is suggested that fitting an empirical model to the more robust model could produce this simplification. However, this task could prove troublesome owing to the nonlinearity of typical polymerisation processes. An additional problem with simplified models is that its accuracy may only be applicable over a short input range. In this study, the author was able to simplify the model with reasonable accuracy over a short range using semi-empirical equations. Dynamic optimisation techniques were implemented over the simplified model whereby the control variables of the polypropylene plant, namely gas flow, catalyst flow and inlet temperature, are approximated by CVP using series of step functions. Here the time intervals are not fixed, thus both the amplitudes and switching times are decision variables. Optimal control variables are calculated offline through dynamic optimisation and implemented as set-points in a non-linear generic model control (GMC) based controller for feedback control.

The method of steepest descent, explained by Kirk (2004b), is usually applied to variational calculus problems, but Paulen *et al*. (2010) applied it to a system where the control vectors were parameterised. This is a so-called hybrid approach, where the optimisation is based on the more fundamental approach, using discrete techniques. The authors found that this approach converged on local minima. This is most likely due to the use of a constant step size for the adjustment of the control variables and is a common problem with gradient-based optimisers.

2.3.2.2 Genetic algorithms

Banu *et al*. (2008) compared the performance of the iterative methods using the calculus of variations approach to that of direct methods represented by the genetic algorithm. While the variational approach is more rigorous and deterministic from a mathematical point of view, convergence towards a global minimum is poor for non-linear systems. Using genetic

algorithms grants more security in terms of global convergence albeit at the expense of computational effort. Chakravarty *et al*. (1997) and Lee *et al*. (1997) were the first to apply genetic algorithms to grade transition optimisation for MMA polymer and MMA-Vinyl Acetate (VA) copolymer, respectively.

Chakravarty, *et al*. differed in the approach to CVP by using curve-fitted piecewise splines to discretise the control variables. The profile is much smoother, thus providing a closer approximation to the true optimal solution.

Lee, *et al*. incorporated heuristics into GA in order to reduce computation time by reducing the control variable range at each time interval. The heuristic reduces the search space by calculating a new range of parameters based on the parameter value at the previous time interval. The following equations describe the heuristic:

$$\underline{u}_{i,j}{}^{\alpha} = u_i^{ss} - \frac{(\overline{u}_i - \underline{u}_i)}{2}\alpha_i^{\,j} \tag{2.12}$$

$$\overline{u}_{i,j}{}^{\alpha} = u_i^{ss} + \frac{(\overline{u}_i - \underline{u}_i)}{2}\alpha_i^{\,j} \tag{2.13}$$

$$\underline{u}_{i,j}{}^{\alpha} \geq \underline{u}_i \text{ and } \overline{u}_{i,j}{}^{\alpha} \leq \overline{u}_i \tag{2.14}$$

$$0 < \alpha_i < 1 \tag{2.15}$$

Subscript i identifies the control variable and subscript j denotes the time interval. The overbar and underscore represent the upper and lower bounds, respectively. The step factor α_i is raised to the power j and is always a fraction. Thus, the factor approaches zero as time proceeds, narrowing the bounds around the steady-state value, which is denoted by superscript ss. This is known as the method of converging constraints.

2.3.2.3 Differential evolution algorithms

Lee *et al*. (1999) used a modified differential evolution (MDE) algorithm to optimise the process start-up operation and grade transitions for MMA-VA copolymerisation plants. MDE differs from DE in two regards. First, it improves the search performance by performing local searches. The base step function, F, is initially selected as low. During the competing stage of the algorithm, a local search is initiated when the current generation has a lower objective function value than that of the previous generation. At this point, F is increased by a factor and applied to the same vector difference as the current generation. . Objective function value reduction ends the local search, sets the new point as the current generation, resets the step size to the base value and continues the normal DE algorithm. Second, it

incorporates the method of converging range constraints as introduced in the previous section for GA. This allows for overall faster speeds and better performance in terms of objective function values compared to classic DE, but without a loss of robustness. Each modification was shown to improve DE independently and the combination of the two modifications resulted in a significant improvement in convergence rate over classic DE.

In an attempt to reduce memory space usage, thereby decreasing the number of function evaluations, Babu and Angira (2006) formulated DE in such a way that only one population is needed, instead of two. The modification proved successful, allowing population members to compete with its mutations in a single vector, thus reducing memory usage.

Most of the above-mentioned studies discretise the grid at fixed intervals. Convergence speed can be increased by iteratively updating and refining the grid intervals. Further, alternative functions are proposed to parameterise control variables, including splines and wavelets to increase the accuracy of the solutions (Schlegel et al., 2005).

2.3.3 Simultaneous dynamic optimisation

The simultaneous dynamic optimisation approach was used by Cervantes *et al*. (2002) for the optimisation of a low density polyethylene plant. State and control variables were parameterised by the method of orthogonal collocation on finite elements. The subsequent NLP is solved using an interior point algorithm, the particulars of which are explained in a study by Cervantes *et al*. (2000). In order to increase the accuracy of the solution, a mesh refinement strategy is also implemented. Results showed a reduction in transition time of up to 44%. The performance of the solution was governed by the number of finite elements and the extent of mesh refinement. However, increasing these aspects can severely affect the computational efficiency.

BenAmor *et al*. (2004) promoted the industrial usage of model-predictive control for polymerisation models based on fundamental principles. A real-time optimisation package called Rigorous Online Modelling and equation-based optimisation (ROMeo) was implemented to solve the orthogonal collocation points that are used to discretise the state and control variables profiles.

Additional examples of grade transition times being minimised using simultaneous NLP methods can be seen in studies by Flores-Tlacuahuac (2005) and Flores-Tlacuahuac *et al*. (2006) for high-impact polystyrene (HIPS) systems. In these studies, the state and control variables profiles were approximated using Radua collocation on finite elements and the collocation points were solved by an interior point algorithm.

2.3.4 Dynamic optimisation discussion

In all of the cases reviewed above, save for the earliest ones, the polymerisation process models were too complicated to be posed as a dynamic optimisation problem that is solved

using classic calculus of variations approach. As such, it is impossible to determine the true continuous optimal input trajectories that would minimise grade transition times or off-spec polymer production. This has proven to be true regardless of whether the process model is empirical or developed from first principles. At best, the input trajectories can be parameterised so as to approximate the true optimal profile. The manner in which the trajectories are parameterised has a profound effect on the performance of the optimisation procedure. Usually, a greater number of parameters produce a more accurate approximation of the true optimal profile. However, the more parameters there are that need to be optimised, the higher the computational cost becomes. Thus, there exists a trade-off between solution accuracy and performance.

There is also freedom of choice with regard to the definition of the objective function. In most cases, there have been multiple objectives i.e. numerous properties deviations to minimise that have been lumped into a single summation regardless of whether there are contradictory consequences of control action on the individual property deviations. Only Tsoukas *et al*. (1982) performed a Pareto analysis to determine a more practical and realistic alternative to weighting objectives.

In addition to the parameterisation method and the objective function definition, there is also a large amount of algorithms that are applicable to dynamic optimisation problems. A significant amount of these can be disregarded due to their tendency to converge on local minima or their requirement that state variables also be parameterised. The importance of a polymerisation model based on fundamental principles for the purpose of model-predictive control has already been stressed. It would be contrary to this philosophy to perform optimisation over the model where the output search space is also discretised. At present, the algorithms based on evolutionary principles seem most suited to a polymerisation optimisation problem for their ability to converge on global solutions and for their computational performance.

2.4 Off-spec polymer recycling

Despite all the work that has been done to reduce the amount of off-spec polymer produced during grade transitions, particularly in the field of dynamic optimisation discussed in the previous section, the generation of off-spec polymer is unavoidable. Polymer production typically requires far greater rigour in quality control and product reproducibility than in other industries. A polymeric material labelled as off-spec is perceived as being of inferior quality, thus it has a low selling price. However, the off-spec polymer produced during grade transitions exhibit properties of an intermediate nature relative to both intended grade properties and that which is produced prior to initiation the grade transition. To avoid having to sell it at a low price, this off-spec polymer is off-spec polymer is sometimes blended with on-spec polymer such that its properties remain within spec.

2.4.1 Off-spec blending

Claims of off-spec polymer being recycled either to obtain specific properties or to upgrade a product to within specification have been made by Heaton (1994) and Akkapeddi (2003). These practices are purely empirical, similar to the way grade transitions are handles traditionally (Utracki, 2003), where the flowrate of off-spec material added to the virgin polymer is decided by plant operators based on experience.

It must be noted that off-spec material can be captured and stored at one of two areas in a plant. Off-spec material captured after it has undergone some form of post-processing (extrusion, injection moulding, etc.) may exhibit property degradation during those procedures, while off-spec material captured prior to post-processing exhibits properties that can be inferred from a process model.

2.4.2 Published blending studies

There appears to be a lack of quantitative studies regarding the blending of the same polymer type having different properties. The majority of studies published regard the blending of different polymer types with one another. Comprehensive overviews on this topic are available (Robeson, 2007; Utracki, 2003; Akkapeddi, 2003; Isayev, 2010). The aim of blending in these studies is to produce a polymer with properties superior to either of its constituents (synergistic properties).

A study conducted by Bhateja and Andrews (1983) on the blending of ultra-high molecular weight polyethylene (UHMWPE) with normal molecular weight polyethylene, such as HDPE, showed that the properties of the blend were of an intermediate nature when compared with that of its respective constituents. The aim of the study was to seek useful ratios on a mass basis by which to blend the two polymer types in order to determine if the high strength properties of UHMWPE could be obtained in conjunction with the flow characteristics of HDPE for post-processing ease. These ratios are known as blending ratios. The study was unsuccessful in fulfilling its aim, but it was noted that no synergistic effects were observed when blending i.e. the properties of the blends were not greater than the sum of their individual components. No attempt was made to predict the properties of the blends based on different mass ratios of the pure grades.

Studies by Abbas *et al*. (1978), Throne (1987) and Bernado *et al*. (1996) have explored the concept of a recycle loop around the post-reactor processing area. Within these studies, mathematical expressions were derived for predicting the properties of a mixture of processed polymer with virgin powder from the reactor. The effects of reprocessing were investigated for properties such as MFI, M_w, M_n and PDI. The equations for predicting the properties of the mixture were proposed as either linear (Equation (2.16)) or logarithmic (Equation (2.17)), depending on which model matched experimental data more

accurately. No physical or mathematical basis was given as to why the mixture's properties were described by either of the proposed functions.

$$x_3 = Ax_1 + Bx_2 \tag{2.16}$$

$$\log(x_3) = C\log(x_1) + D\log(x_2) \tag{2.17}$$

Depending on whether off-spec polymer is stored before or after extrusion and granulation, the material has either never been processed or processed once. Thus, the effect of reprocessing on the properties is negligible. While these studies do not involve off-spec blending, they are the only documented attempts to mathematically predict the properties of a mixture based on the individual properties of each stream.

Another example in literature of off-spec polymer being recycled on a plant is that of re-granulating scrap materials that remain in extruders after a batch has been processed. The main issue with regard to re-granulation is that the product may undergo degradation during the procedure. Mixing of the polymer with fresh (virgin) polymer can minimise the effect of degradation. A number of studies regarding this form of recycling have been performed over the years as highlighted in the review by Al-Salem, (2009).

Literature suggests that off-spec blending is practised in industry, yet the technical aspects regarding the procedure are not well-documented. The reluctance to pursue this idea may have been based on the lack of confidence in the tracking of the properties of a polymer product formed under unsteady-state conditions. In recent years, however, the field of polymerisation modelling has grown to the point where the prediction of polymer properties under dynamic behaviour has been realised. Furthermore, numerous studies in the field of grade transition minimisation have proven that optimising plant operation through application of control theory has significantly reduced the transition times. It is suggested that similar techniques can be of benefit to the process of off-spec blending.

2.5 Summary

Numerous studies have been published pertaining to modelling polymerisation processes for both steady-state and unsteady-state operation. A long term goal of this academic field is to accurately predict polymer properties fast enough for control action to be implemented online (model-predictive control). The majority of recent published work is geared towards developing polymerisation models with quick runtimes as well as implementing these models in fast converging global search algorithms for optimal grade transition times.

In formulating a solution to an optimal grade transition, a number of techniques can be implemented. Conceptual approaches have been categorised into three different areas, namely iterative methods, CVP and SDO. Various algorithms have been developed in each

of these categories. The form of the objective function that is to be minimised by the chosen algorithm is of great importance. In the case of CVP and SDO, the manner in which the state and/or control variable profiles are discretised creates even more combinations for the setup of an efficient solution.

A discussion of the previous attempts to recycle off-spec polymer has brought to light the fact that off-spec blending is practised in industry, but is not well-documented. Furthermore, there is a lack of quantitative studies regarding blending grades of the same polymer type and predicting the properties of a blend.

An off-spec blending model should predict the same properties relevant to defining a polymer grade as its respective process model. Control of the flowrates of polymer into a blending tank could be posed as a dynamic optimisation problem where the same techniques employed in grade transition optimisation can be applied.

It is suggested that a study for quantitatively predicting the properties of blended grades can open the way to dynamic optimisation of off-spec blending by incorporating a rigorous polymerisation model with an effective global optimisation algorithm.

3 THESIS OBJECTIVES

A large variance in polymer properties during production is possible due to the sensitivity of the product types and catalyst performance to the presence of additives and basic operating conditions like temperature, pressure and reactant (monomer and hydrogen) concentration. Often the operating conditions are intentionally altered to produce different grades of polymer that are suitable for a variety of applications. Large quantities of off-spec polymer are generated during the transition periods that arise from these grade changes.

Due to the recent developments in the field of polymerisation modelling, as shown in the previous chapter, these grade changes can be accurately simulated with low computational expense. An important component of these process models is its ability to predict the polymer properties under dynamic behaviour. These process models can be particularly useful for the purpose of model-predictive control. Already, numerous studies have been conducted on optimisation of the grade transition either by minimising the grade transition time or the mass of off-spec polymer produced during this period. In all of these studies, a process model was required to perform the optimisation; many of the successful studies included process models that were developed from first principles.

Regardless of the attempts to reduce grade transition times, the formation of off-spec polymer is still physically impossible to avoid. It was noted that off-spec polymer is sometimes blended with pure grade products in industry to deal with the off-spec material that is generated. However, this practice is not well-documented and there are no quantitative studies in this regard. As such, an opportunity is present to investigate the prospect of off-spec polymer blending with pure grade. Given that the properties relevant to defining a grade are the same properties that can be used to distinguish between a polymer that is within specification or off-spec, that polymerisation process models can be developed that predict these properties and that numerous studies on the optimisation of grade transitions have been conducted where these types of process models were used; it would seem practical to pursue the idea of off-spec blending by posing it as an optimisation problem in a similar manner to the grade transition problem.

The primary goal of this study is to develop a control strategy for blending the maximum amount of stock-piled off-spec material with fresh polymer formed under either steady-state or unsteady-state operation where the resultant blends must meet the specifications of a desired grade.

Prospective control strategies will be applied to a Ziegler-Natta catalysed polymerisation system with ethylene monomer and 1-butene comonomer. This process yields linear chains with no long-chain branching (LCB) and a relatively small amount of short-chain branching (SCB), brought about by varying comonomer orientations along the polymer backbone

(Malpass, 2010). Industries often characterise their product grades according to MFI values and particle density. The study will focus on properties similar to the former property, which is related to the MWD of a polymer. It it is difficult to predict the MFI from a process model, thus it will be represented by average properties of the MWD, namely M_n and PDI.

As mentioned, the properties of a polymer are influenced by the process conditions and the feed stream reactant composition. In order to simulate the system for control purposes, a process model will be developed based on fundamental principles that accurately relate control variables, such as the feed rates, to the relevant properties that define a polymer grade. An additional model will be developed that quantitatively predicts the properties of a polymer blend relevant to defining a grade. Dynamic optimisation techniques that are usually applied to optimal grade transition problems will be used in scenarios that incorporate both process and blending models. These techniques will be used to demonstrate whether an off-spec source with relatively low chain lengths can be used to create a polymer with higher chain lengths and vice versa. The spread of the chain length distribution about the mean value will also be taken into account.

Given the project objectives, a hypothesis is proposed: the flowrate from an off-spec polymer source and/or reactor feed flowrates can be manipulated by a model-predictive controller in order to reduce the overall amount of off-spec material on a plant by consumption of the existing off-spec polymer source and/or by reduction of off-spec material generated during a grade transition.

The following key questions need to be answered in order to achieve the desired objectives.

- When blending polymers with different properties, such as on-spec and off-spec polymer, what mathematical relationship can be used to deduce the properties of the blended polymer from the properties of its constituents?
- Ziegler-Natta catalysed polymerisation products generally exhibit large polydispersity indices, which is theorised to be due to catalyst site heterogeneity. A blend of two polymers is most likely to exhibit a PDI value higher than that of its constituents. In light of these two statements, is it possible to produce a blend with a relatively narrow variance to satisfy the specifications of a desired grade whilst maintaining the required M_n?
- Polymerisation systems are usually described by non-linear reactor models and are computationally expensive to solve. A long-term goal of process modelling is the prospect of applying model-predictive control to real-world systems. Will the computation time for unsteady-state blending be fast enough to realise this goal?

4 POLYMER BLENDING MODEL DEVELOPMENT

A key feature of Ziegler-Natta catalysed ethylene polymerisation is the production of relatively linear chains. Thus, the main physical characteristic describing a product formed under this type of catalytic reaction is its Chain Length Distribution (CLD). As such, it has been deemed appropriate to base the investigation of off-spec blending on models that predict the properties relating to a polymer's CLD.

It is expected that the CLD of a blend would differ from that of its individual components in such a way that it can be predicted from first principles. Thus, a model is presented in the current chapter where the blend properties are calculated as functions of each constituent's CLD. These properties include the M_n, M_w and PDI of a blend, since these parameters represent the CLD in a manner that is easier to report. Thereafter, an analysis of experimental data is conducted to verify the validity of the proposed model.

This chapter aims to look solely at the influence of varying the mass ratio of two polymer samples with prescribed polymer properties on the properties of its blend. The model of this process will be developed on a fundamental basis with the aim of linking it with a polymerisation reactor model at a later stage.

4.1 Governing equations

A CLD can be described numerically by calculating the moments of the curve. These moments are statistical parameters from which polymer properties can be determined, such as M_n, M_w and PDI.

The moments of a polymer product are based on the number of monomer units in a chain, n, and the concentration of both dead chains, D_n, and live chains, P_n, that consists of n monomer units.

$$\mu_k = \sum_{i=1}^{n} n^k \left(D_n + P_n \right) \tag{4.1}$$

The concentration of dead chains is far greater than the concentration of live chains, thus it can be assumed that the latter concentration is negligible. Therefore, definition of bulk polymer moments can be simplified to Equation .

$$\mu_k = \sum_{i=1}^{n} n^k D_n \tag{4.2}$$

If polymer grade blending is understood as pooling the data of two distribution curves, then a method of calculating the moments of the blend product will exist. For instance, the overall mass balance around a blend of mass m_A of grade A and mass m_B of grade B is:

$$m_A + m_B = m_f \tag{4.3}$$

To compare polymers, the distribution curves must first be normalised. The polymer chain length distribution is normalised if the following equation is satisfied:

$$\int_{-\infty}^{\infty} D_n \, dn = 1 \tag{4.4}$$

Using normalised distribution curves, a mass balance for polymer chains having length n can be performed as described by Equation (4.5).

$$m_A D_n{}^A + m_B D_n{}^B = m_f D_n{}^f \tag{4.5}$$

The probability of a polymer chain having length n after blending two grades is, thus, the weighted sum of the individual grades on a mass basis, where w is defined as the weight fraction:

$$D_n{}^f = w_A D_n{}^A + w_B D_n{}^B \tag{4.6}$$

This balance can be extended for the sum of all polymer chains having length 1 to having infinite length:

$$\sum n^k D_n{}^f = \sum (w_A n^k D_n{}^A + w_B n^k D_n{}^B) \tag{4.7}$$

$$\sum n^k D_n{}^f = \sum (w_A n^k D_n{}^A) + \sum (w_B n^k D_n{}^B) \tag{4.8}$$

$$\sum n^k D_n{}^f = w_A \sum n^k D_n{}^A + w_B \sum n^k D_n{}^B \tag{4.9}$$

Each sum in Equation (4.9) conforms to the definition of moments stated in Equation (4.1). Thus, equation (4.10) indicates that the moments of a polymer blend can be predicted from its respected constituents as a linear mean.

$$\mu_k{}^f = w_A \mu_k{}^A + w_B \mu_k{}^B \tag{4.10}$$

The zero, first and second moments are often quoted in polymerisation. Physically, the zero moment is the total concentration of polymer chains, whereas the first moment is the total

concentration of monomer units. The second moment gives emphasis to the larger polymers. The nature of the blend's distribution can then be represented by M_n, M_w and PDI using the calculated zero, first and second moments, as shown in the following three equations:

$$M_n{}^f = \frac{\mu_1{}^f}{\mu_0{}^f} \tag{4.11}$$

$$M_w{}^f = \frac{\mu_2{}^f}{\mu_1{}^f} \tag{4.12}$$

$$PDI^f = \frac{\mu_0{}^f \mu_2{}^f}{\left(\mu_1{}^f\right)^2} \tag{4.13}$$

4.2 GPC analysis

The lack of published blending models and studies was noted in the literature review. Those studies that were published have empirical models, thus had no physical justification for their development. Although the previous section implies that modelling blending is simple, the lack of published studies gives cause for concern that this is not the case. It is for this reason that experiments have been conducted to determine whether the proposed blending model gives reasonable results.

In the current section, real data is reported to compare polyethylene blends with its constituents from gel permeation chromatography (GPC) measurements. GPC is a technique that is widely used in laboratories for polymer analysis where the components of a mixture are separated according to size. The process involves elution of the sample through a column of packed beads where the retention time of a polymer particle in the pores of the beads is related to its size. The outcome of the analysis is the molecular weight distribution, which can be related to the chain length distribution, from which M_n, M_w and PDI can be inferred.

In the present study, three industrial polymer grades, labelled B, C and D, were made available for analysis. Two out of the three grades were blended in different combinations and ratios to form blends named CB, CD and BD. Furthermore, the grades were mixed with mass ratios of 1:3, 1:1 and 3:1 for each blend type. The naming convention for the blends consists of a pair of letter-number combinations, where the letters represent the pure grade and the number that follows defines the mass part in relation to the other grade. For instance, blend C3B1 is made of 3 parts grade C and 1 part grade B by mass. Thereafter, samples of the pure grades and the blends were prepared for GPC analysis. Some of the pure grade samples and blends were analysed in triplicate to ensure the repeatability of the

results. The aim is to use the experimental data to validate the blending model developed in Section 4.1.

GPC analysis takes place in solution, of which there are many popular solvents. In this case, the solvent was trichlorobenzene (TCB) stabilised with 0.0125% butylated hydroxytoluene (BHT). The column was operated at 150°C. The direct output measurement of a polymer sample is a time-varying electronic signal in mV that produces a characteristic curve in terms of detector response and retention time. The value of the detector response at a particular time is indicative of the frequency of polymer chains having a particular hydrodynamic volume. The hydrodynamic volume of a polymer type is, in turn, related to the molecular weight of that particle.

4.2.1 Calibration of the GPC data

To obtain the MWD, a calibration curve must be produced according to a set of standards in order to determine the molecular weight of a particle from its retention time. Only a calibration curve for polystyrene has been produced for the current system. As such, the use of this curve would produce an inaccurate representation of the MWD of the polyethylene samples.

Under the same set of conditions in the same solvent, the retention time of all polymer types having a specific hydrodynamic volume will be the same (Boni *et al.*, 1968). Thus a plot of hydrodynamic volume vs. retention time will be universal and is known as a Mark-Houwink plot. The following procedure was used to determine the calibration curve for the polyethylene samples used in the current study.

The hydrodynamic volume is defined as the product of intrinsic viscosity (η) and molecular weight (MW).Furthermore; the intrinsic viscosity can be presented as a function of the molecular weight by the empirical Mark-Houwink equation, originally proposed by Mark (1938) and subsequently verified by Houwink (1940).

$$\eta = K\left(MW\right)^{\alpha} \tag{4.14}$$

The calibration curve for polystyrene was given in the form of a cubic equation. Using this calibration curve, the molecular weights, intrinsic viscosity and hydrodynamic volumes can be calculated in terms of the retention times of the polymer particles.

Since the universal calibration curve shows that the relationship between hydrodynamic volume and retention time is the same for all polymers, the product of molecular weight and intrinsic viscosity for polystyrene and polyethylene, indicated by subscripts 1 and 2 respectively, can be equated.

The equation can be rearranged to solve for the molecular weight of polyethylene in terms of the molecular weight of polystyrene.

$$MW_2 = \frac{K_1}{K_2}\left(MW_1\right)^{\frac{\alpha_1+1}{\alpha_2+1}} \tag{4.15}$$

The Mark-Houwink parameters of each component in TCB at 150 °C according to IUPAC are presented in the following table (Cervenka, 1973).

Table 4-1: Mark-Houwink parameters for polystyrene and polyethylene

	K	α
Polystyrene	12.1	0.707
Polyethylene	40.6	0.725

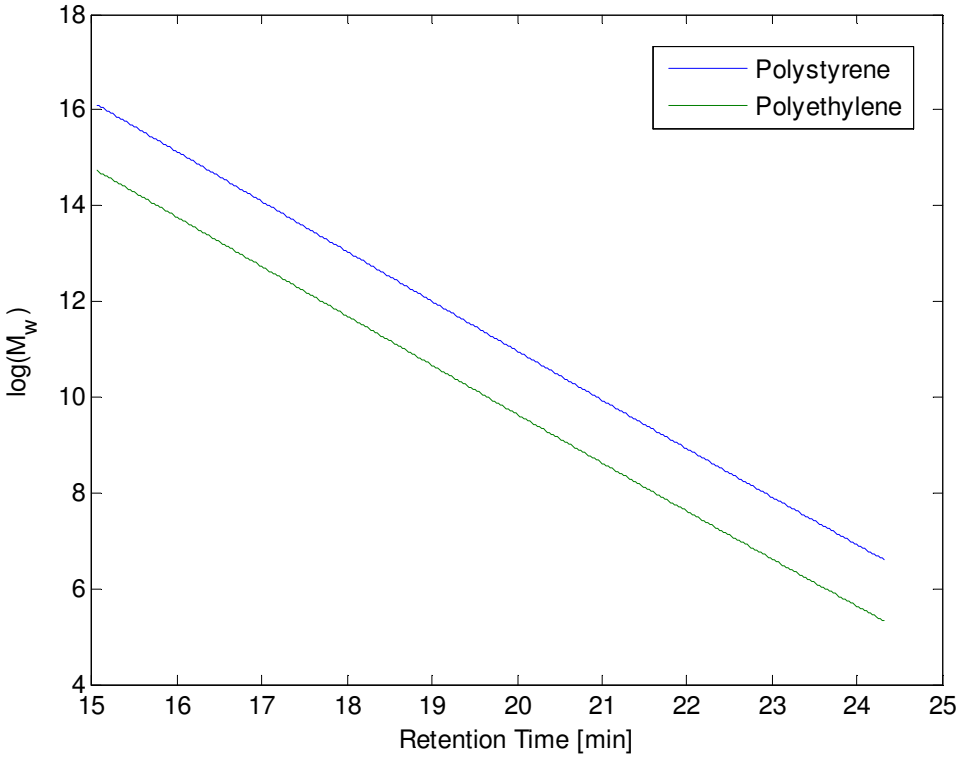

Figure 4.1 - Calibration curves of polystyrene and polyethylene

A comparison of the molecular weights vs. retention time is given in Figure 4.1. From this plot, a cubic equation was fitted to the polyethylene data to calculate the molecular weight of the particles at any given retention time $\left(t_r\right)$. The equation is of the form:

$$\log\left(MW\right) = P\left(t_r\right) = At_r^{\,3} + Bt_r^{\,2} + Ct_r + D \tag{4.16}$$

Table 4-2: Coefficients of cubic equation to predict molecular weight from retention time

Coefficient	Polystyrene	Polyethylene
A	13.09	12.43
B	-0.323	-0.319
C	-0.008	-0.008
D	1.5×10^{4}	2.0×10^{4}

4.2.2 Data smoothing

Close inspection of the raw data have shown that the curves are clipped at the high chain length tail-ends. This gives the impression that the distribution curve ends abruptly. In order to avoid skewing the results, the missing data have been approximated by fitting smooth curves to the complete data set. This was achieved using local regression (LOESS) in MATLAB. In this technique, a 2nd degree polynomial is fitted at each point in the data set using a subset of the data and weighted linear least squares The further a point is in the subset from the point at which the polynomial is to be fitted, the lower its weighting is (Cleveland & Loader, 1996). If any data points were estimated as negative values, thereafter they were set to zero. A portion of a MWD curve is shown in Figure 4.2 to illustrate the outcome of LOESS. All subsequent MWD curves presented have been smoothed in this manner.

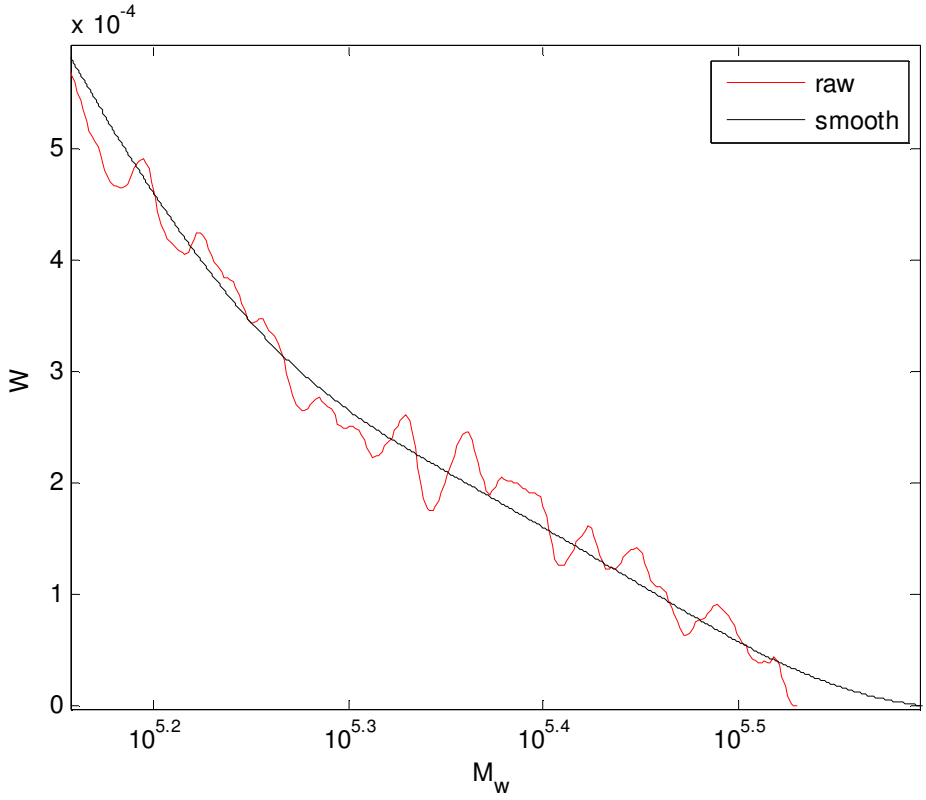

Figure 4.2 – Tail-end comparison between raw MWD and smoothed MWD at high MWD

4.2.3 Generating a normalised MWD

In order to compare MWD's amongst the various polymer samples, the measurements must be normalised to remove the dependence on polymer solvent-solvent concentrations as follows:

$$w(MW) = \frac{dw(t_r)}{d\log(MW)} \tag{4.17}$$

where dw is the weight fraction of polymer eluted between t_r and $t_r + dt_r$. It is represented by the differential of the normalised signal height:

$$dw = \frac{S(t_r)\,dt_r}{\int_0^\infty S(t_r)\,dt_r} \tag{4.18}$$

Since the calibration curve is represented by a cubic equation, the analytical solution to its first derivate, $\frac{d\log(MW)}{dt_r}$ can be easily obtained. This is enough information to compute the normalised weight distribution, which reduces to the normalised detector response divided by the slope of the calibration curve:

$$w(MW) = \frac{S(t_r)}{\left(\int_0^\infty S(t_r)\,dt_r\right)\left(3At_r^{\,2} + 2Bt_r + C\right)} \tag{1.19}$$

4.2.4 Pure grades

The MWD curves of the three pure grades are compared in Figure 4.3. Each curve resembles that of a normal distribution curve and each has a clearly distinguishable peaks. The following deductions have been made. Grade D has more chains with higher chain lengths than both Grades C and B, since its peak is situated to the right of the other peaks. Grade B has the narrowest distribution, followed by Grade D, whilst Grade C is spread the widest. Thus, Grade C should have the highest PDI value. These observations are confirmed by calculating the average properties of each grade as shown in the following table:

Table 4-3: Average properties of pure grades

	M_n	PDI
B	5.53 x10^4	7.61
C	5.73 x10^4	7.57
D	6.13 x10^4	5.81

4.2.5 Repeatability

Despite the fact that the pure grade curves are clearly distinguishable from one another, their average properties are quite similar. In order to prove that the differences between the pure grade curves are statistically significant, they have been compared to the differences between the measurements of the repeat blend samples.

The MWD curves of the triplicate measurements of blends CD and C3B1 are presented in Figure 4.4 and Figure 4.5, respectively. The curves of the repeat measurements are nearly indistinguishable from one another. As such, they appear to be in excellent agreement with one another.

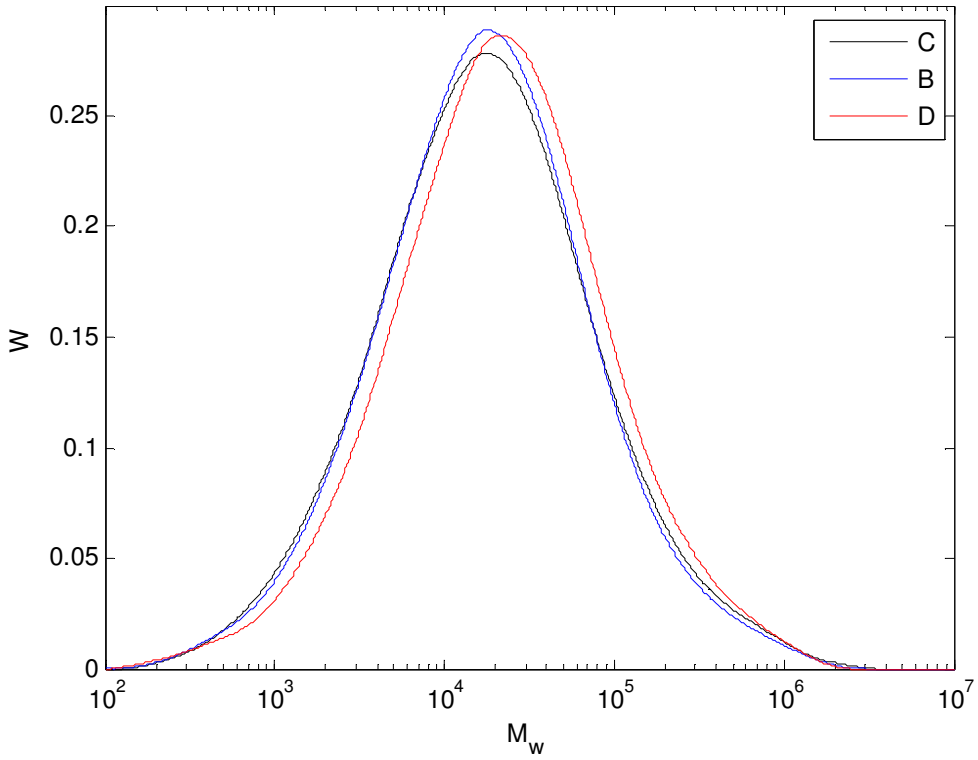

Figure 4.3 – Molecular weight distributions of pure grades

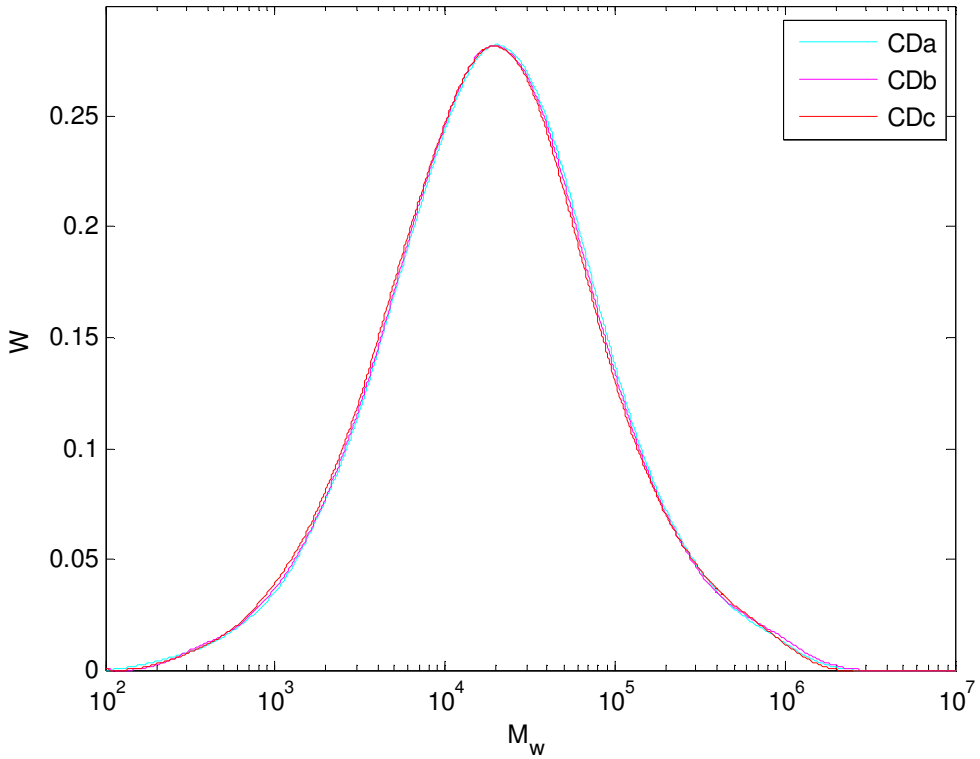

Figure 4.4 – Molecular weight distributions of repeat samples for blend CD

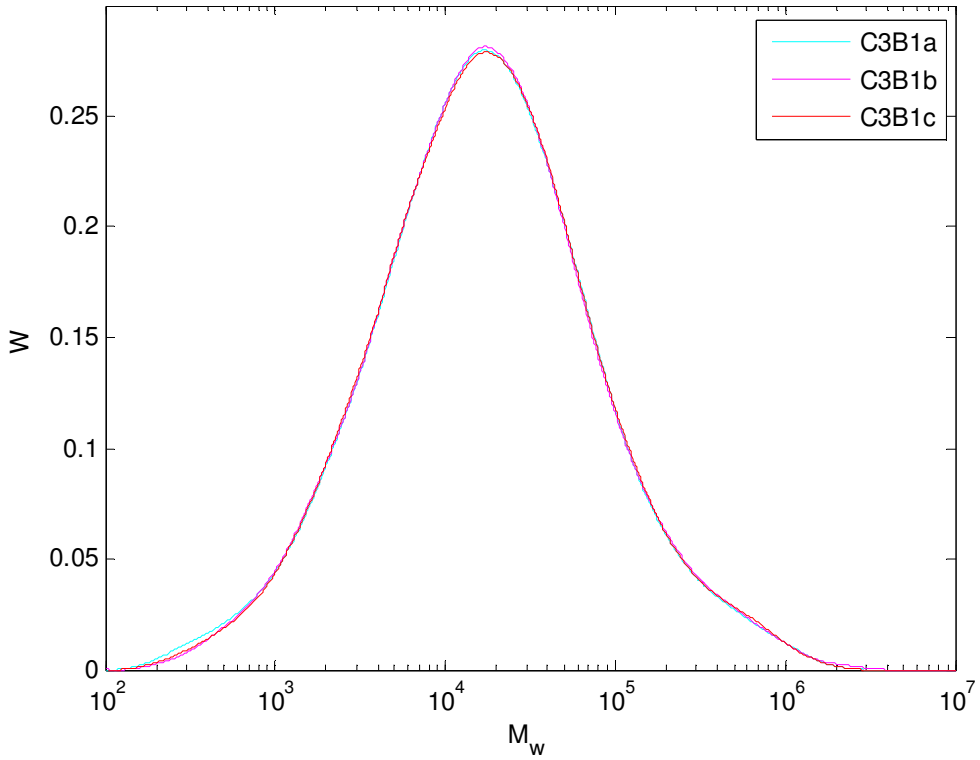

Figure 4.5 – Molecular weight distributions of repeat samples for blend C3B1

By inspection, the differences between the curves of the pure grades shown in Figure 4.3 appear to be more significant than the differences between the triplicate measurement curves shown in Figure 4.4 and Figure 4.5. Quantitative proof of this significance is evident

in Table 4-4 where the relative standard deviations (RSD) of the 1st moments of the triplicate blends are shown to be lower than the relative standard deviation of the pure grades.

Table 4-4: Comparison of the relative standard deviations of repeat samples

	$\overline{\mu}_1$	% RSD
pure grades	4.96 x10^6	7.47
blend CD	5.07 x10^6	2.78
blend C3B1	4.82 x10^6	1.85

The triplicate measurements have RSD values that are lower than 5%, thus they can be deemed precise. Conversely, the curves of Grades C, D and B, which have been pooled into one data set as if they were measurements of repeated samples, have a RSD value that is greater than 5%. From the table and figures provided above, it can be deduced that there is good repeatability of the experiment due to the low RSD values of the repeat measurements. Therefore, the differences between the curves of the pure grades are due to their distinct properties and not due to measurement error.

4.2.6 Blends results

The following three plots contain the MWD curves of the pure grades and the curves of the blends that were created in the various blend ratios from those pure grades. They are presented in Figure 4.6, Figure 4.7 and Figure 4.8 for blends of CB, BD and CD, respectively.

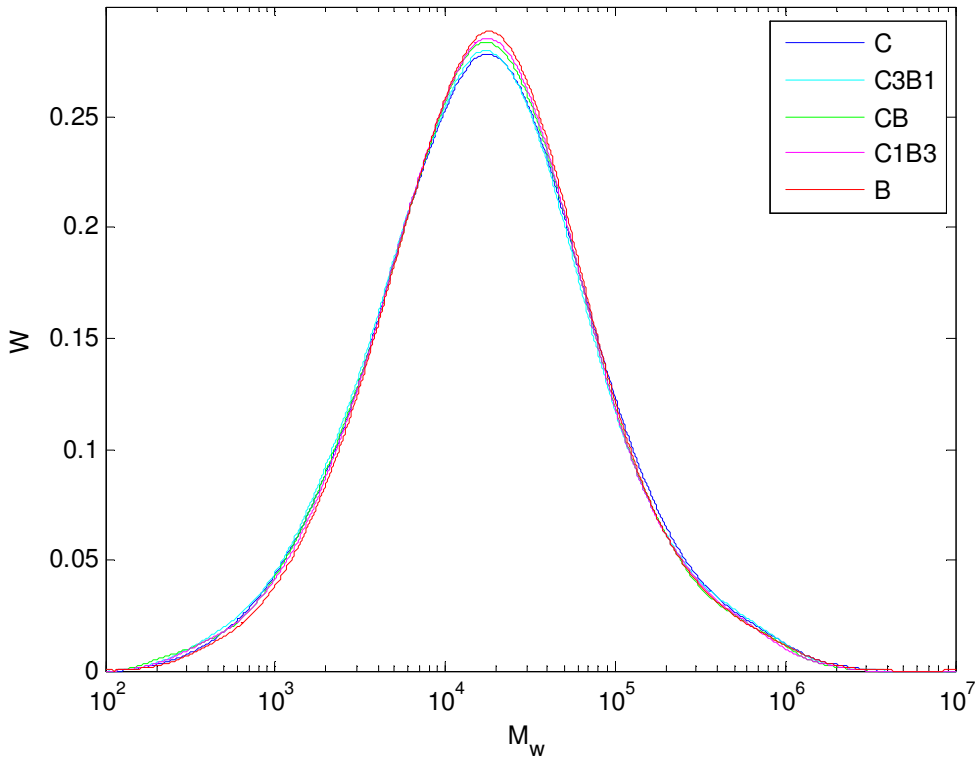

Figure 4.6 – Molecular weight distributions of Grades C and B blends

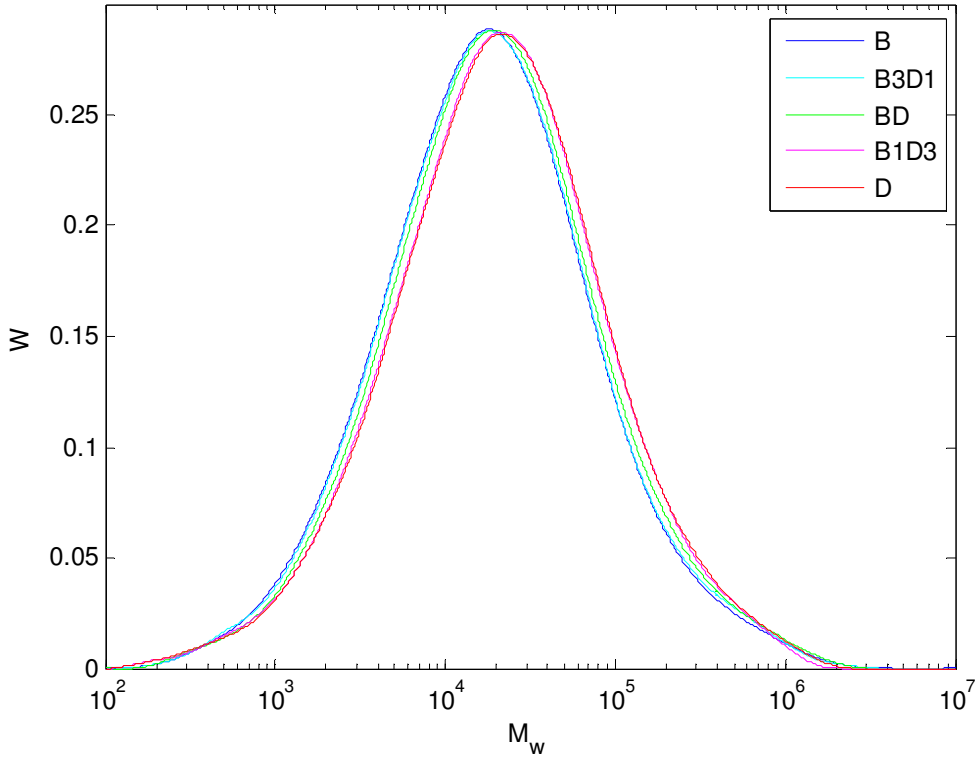

Figure 4.7 - Molecular weight distributions of Grades B and D blends

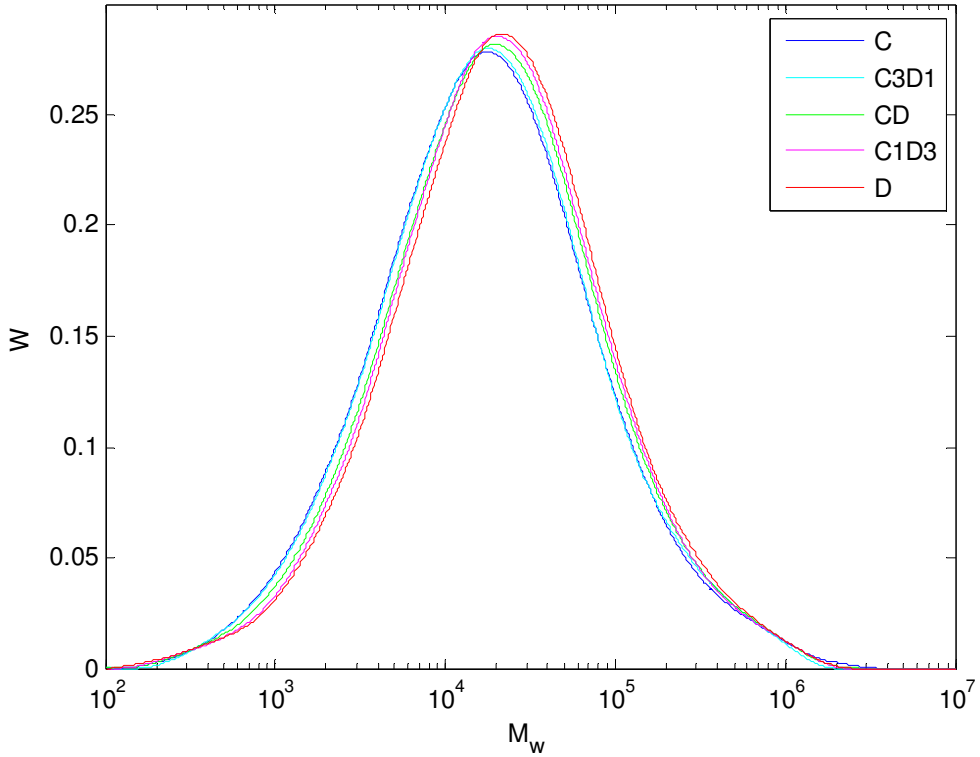

Figure 4.8 - Molecular weight distributions of Grades C and D blends

The purpose of the figures is to illustrate how the molecular weight of a blend is distributed in relation to that of its constituents. It is evident that a blend has a MWD that is intermediary of its original constituents, since the curve of a blend appears to be shifted proportionately towards the curve of the dominant grade according to the mass ratio in which it is blended.

4.2.7 Model validation

Equation (4.6) states that the linear mean of the concentration of chains having length n in each sample is the concentration of chains in the blend. Assuming that the molecular weight of a chain is related to the chain length, the molecular weight distribution of the blend can be predicted using the distributions of the pure grades. The figures shown in the current section provide a visual representation of the excellent agreement between the model-predicted MWD curves and the GPC measurements of the blends.

An R^2 value is provided in each figure as a quantitative representation of the agreement between the curves. This value is a measure of the goodness of fit for non-linear curves that resemble normal distributions (Cameron & Windmeijer, 1997), such as the MWD curves presented. The R^2 values are calculated using the formula shown in Equation (4.20).

$$R^2 = 1 - \frac{\left\| W^e - W^m \right\|^2}{\left\| W^e - \bar{W}^e \right\|^2} \tag{4.20}$$

Most of the curves are almost indistinguishable from one another, with the exceptions being the curves of B1D3, C3D1 and C3B1 that are slightly shifted apart. The R^2 values for all the data sets are very close to one, indicating very good fits.

Slight mismatches between the models and data could be due to the following types of handling errors. There is an inherent error in any measurement device; the error in GPC measurement is no exception. The mass measurements of the pure grade samples for blending could have been inaccurate and the pure grades might not have been sufficiently mixed so as to achieve homogeneity of the blend properties.

The polymer properties and their respective model predictions can be quantitatively compared by calculating the moments of each using Equation (4.1). The results are summarised in Figure 4.9 where the 1st moments of each blend type in the various blend ratios are compared. Where available, the standard deviations of the triplicate measurements are included.

Generally, the plots in Figure 4.9 are in agreement with the comparisons of the distribution curves in Appendix B. The model predictions follow the same trend as the experimentally determined moments when varying the blend ratio and their respective values are reasonably close.

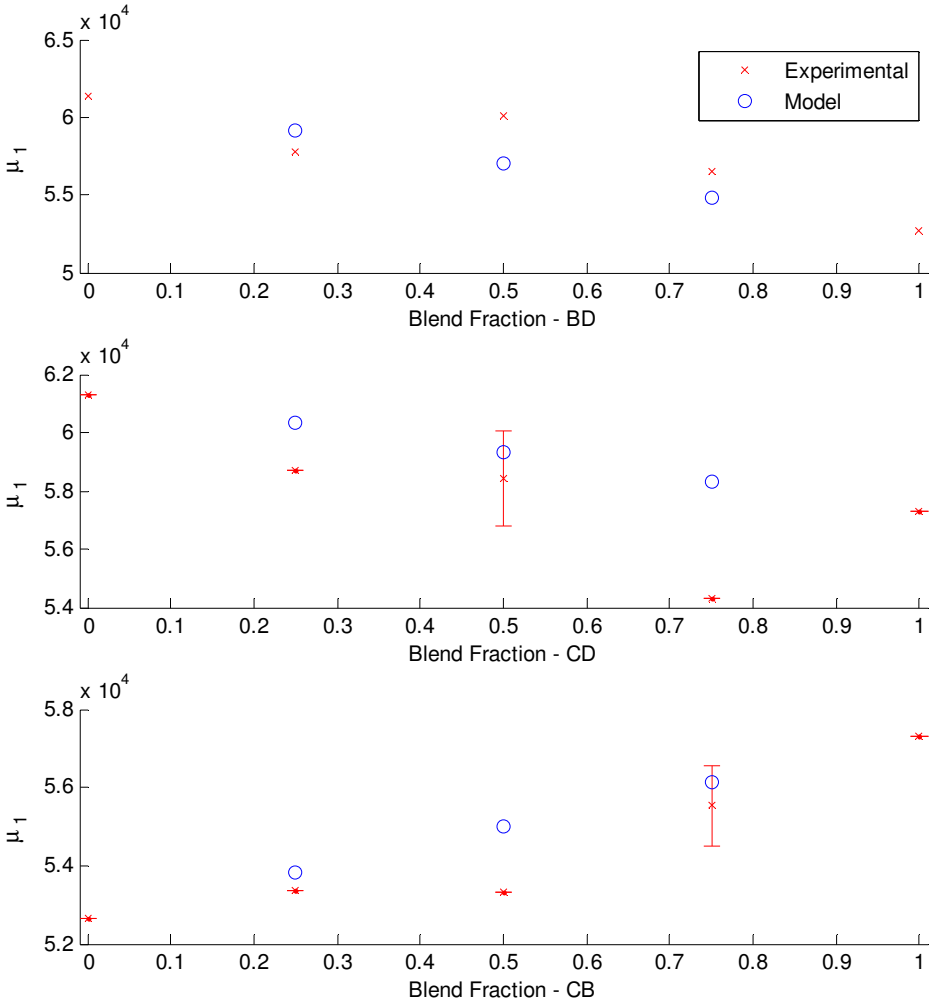

Figure 4.9 - Comparison of experimental and model-predicted 1st moments of polyethylene grades blended in various mass fractions

Small differences in the high molecular weight region of the distribution curve could account for the large error encountered in determining the moments. Essentially, this means that the average properties of an MWD can be skewed by small fluctuations in this high molecular weight region. Calculation of the average properties will propagate the errors of each moment involved, thus creating more uncertainty in the results. Therefore, one could argue for model-prediction of polymer properties to avoid such a bias as opposed to a reliance on analytical instrumentation. There is also a high operating cost associated with GPC analysis, thus the use of a model to predict blend properties may be more feasible for property determination. The wealth of plant data available should provide accurate information on the pure grades' MWDs, which the model can then use.

Polymer moments are statistical parameters that describe aspects of a distribution curve. It has been shown that the entire distribution curve can be predicted with excellent accuracy. Based on the results of these distribution fits, it can be concluded that the polymer properties of a blend can be predicted from the polymer properties of its constituents.

4.3 Summary

A method is presented here for predicting the blended bulk moments of two polymer grades, depending on their masses in batch mode. The molecular weight distributions of the blends are determined using GPC. A comparison of the predicted MWD curves with the experimentally determined MWD of a blend showed the curves to be in excellent agreement, thus confirming that the moments are additive. This relationship will be used to simulate blending on a plant.

5 MODEL DEVELOPMENT

The first step towards developing a control strategy is to define the system over which said controller is to be imposed. This system must be represented by appropriate process models. In the current project, an MPC will be developed for the blending of off-spec material from a storage silo with virgin polymer particles from a reactor. Thus, the system that falls within the control bounds encompasses the reactor and blending sections of a plant. Ethylene and 1-butene are co-polymerised by a third generation Ziegler-Natta catalyst in the reactor vessel under slurry-phase operation. The resulting polymer precipitate is then blended with off-spec material such that the blend is of a desired grade.

In the current chapter, the fundamental development of the reactor model is presented. This includes calculation for the average polymer properties that define a grade. The process model simulates polymer growth around a catalyst particle, based on the concentration of active sites on the catalyst surface and the concentration of reactant components in the liquid phase. The hydrodynamics and vapour-liquid equilibrium of the system are taken into account in order to determine these concentrations. Further, the distribution of polymer particles in the bulk phase, having various chain length distributions are represented by average properties, namely M_n and PDI because these two properties are believed to sufficiently characterise a polymer grade.

5.1 Polymer growth model

5.1.1 Catalyst activity

It is experimentally observed that Ziegler-Natta catalysts produce polymers with high PDI values, often in the range of 4 to 40. However, Flory (1953) has demonstrated that the maximum PDI value for a single site is 2. In an attempt to explain this contradiction, two schools of thought have emerged as discussed by Galvan and Tirrell (1986).

One theory postulates that the high PDI values are due to mass transfer limitations through a growing polymer particle. It is believed that a monomer concentration gradient is formed through the layer, resulting in differing propagation rates that ultimately lead to a wider CLD. The Multi-grain model (Nagel *et al*., 1980) is one of the most rigorously developed models to account for diffusion through a polymer layer. According to this model, the catalyst should be viewed as an agglomeration of particles around which each micro-particle experiences polymer growth. Thus, the model accounts for the presence of mass transfer limitations that causes a variance in the rate of polymerisation. Although it can be shown that this approach will higher PDI values, it still under-predicted experimental results.

The other theory postulates that the catalyst houses numerous active site types where polymerisation occurs at different rates. The basic form of a third generation Ziegler-Natta catalyst, such as the one used in the current system, is that of a titanium salt supported on a crystallite structure. In this case, $TiCl_4$ is supported on a structure of $MgCl_2$. In its natural state, the titanium molecules have a valence number of 4+. Numerous experimental studies point towards a link between catalyst activity and titanium oxidation state (Doi *et al*., 1982; Chien *et al*., 1989; Kim *et al*., 1990; Chien & Nozaki, 1991; Bhaduri *et al*., 2006). This relationship was most rigorously explored by Soga *et al*. (1982). It has been observed that all olefins are polymerised at Ti^{3+} metal centres on the catalyst, whereas ethylene is polymerised at both Ti^{3+} and Ti^{2+} metal centres (Han-Adebekun & Ray, 1997). Most experimentally determined activity profiles are characterised by a sharp increase in active site concentration followed by a slow decay. Initially, this decay was thought to be caused by the increasing mass transfer resistance due to the growing polymer around the catalyst. However, reduction of active titanium sites to lower oxidation states that do not facilitate polymerisation could also explain this decay.

Based on the activity profiles in literature, it is reasonable to assume that the following mechanism adequately describes the site transfer reactions. Reduction from the oxidation state of 4+ to 3+ is irreversible, whereas the transfer from the oxidation state of 3+ to 2+ is reversible:

$$a^{4+} \rightarrow a^{3+} \tag{5.1}$$

$$a^{3+} \rightleftharpoons a^{2+} \tag{5.2}$$

Note that the concentration of potential sites on a single catalyst particle is denoted by a^{4+}. In this mechanism, only sites with an oxidation state of 3+ are deemed active polymerisation centres for ethylene and 1-butene. The further reduced state of 2+ is deemd inactive towards these monomers; but can be oxidised to become the active state of 3+.

According to this model, it is generally possible for each reactant in the system to both oxidise and reduce the titanium sites in phenomena known as site transfer reactions. The contribution made by the co-catalyst, namely TEA, would largely overshadow that of the other components (Hansjörg Sinn & Walter Kaminsky, 1980). However, the influence of the other components on the overall site transfer rate may still be significant, thus their contribution is not ignored.

With reference to the kinetic scheme shown in Table 5-1, the polymeric reactions that take place at active sites are termed initiation, propagation and termination. An active site can be either vacant or occupied at any particular time. Monomer insertion at a vacant site is known

as chain initiation, which changes the vacant site to being in an occupied state. Subsequent monomer addition is known as chain propagation, which occurs at a different rate to chain initiation. In the case of co-polymerisation, the initiation and propagation rates differ for each monomer type. It is assumed that the rate of propagation for a particular monomer is the same regardless of the chain length. This is known as the long chain assumption, which has been known to yield reasonable results (Galvan & Tirrell, 1986; Matyjaszewski & Davis, 2003). Occupied sites are reverted into vacant sites each time a component in the system severs the bond between a chain and the site which it occupies. Reactions of this type are classified as chain termination reactions.

Simulation studies have shown that the cause of the observed broad molecular weight distribution in Ziegler-Natta catalysed polymers is due to the contribution of a number of particles having different MWD's. These distributions are distinct from one another owing to the presence of multiple sites on the catalyst with differing propagation rates (Soares & Hamielec, 1995). In order to obtain the expected high PDI values, the catalyst needs to have at least four distinct active sites. This is a widely accepted theory that gives results consistent with that of the experimental observations, yet there is little physical justification for the presence of such a number of active sites on a catalyst.

Table 5-1: Proposed kinetic framework

	Reaction	Rate Constant
A. Catalytic Reactions		
Transformation		
Spontaneous	$P_*^q \rightarrow P_0^r + D_n^q$	$k_{st,sp}^{q,r}$
Aluminium Alkyl (A)	$P_*^q + \alpha_A A \rightarrow P_0^r + D_n^q$	$k_{st,A}^{q,r}$
Electron Donor (E)	$P_*^q + \alpha_E E \rightarrow P_0^r + D_n^q$	$k_{st,E}^{q,r}$
Hydrogen (H_2)	$P_*^q + \alpha_{H_2} H_2 \rightarrow P_0^r + D_n^q$	$k_{st,H_2}^{q,r}$
Monomer (M_i)	$P_*^q + \alpha_{M_i} M_i \rightarrow P_0^r + D_n^q$	$k_{st,M_i}^{q,r}$
B. Polymeric Reactions		
Propagation		
Initiation	$P_0^q + M_i \rightarrow P_{1,i}^q$	$k_{0,i}^q$
Propagation	$P_{n,i}^q + M_j \rightarrow P_{n+1,j}^q$	$k_{p,ij}^q$
Termination		
Spontaneous	$P_{n,i}^q \rightarrow P_0^q + D_n^q$	$k_{t,sp}^q$
Aluminium Alkyl (A)	$P_{n,i}^q + A \rightarrow P_0^q + D_n^q$	$k_{t,A}^q$
Electron Donor (E)	$P_{n,i}^q + E \rightarrow P_0^q + D_n^q$	$k_{t,E}^q$
Hydrogen (H_2)	$P_{n,i}^q + H_2 \rightarrow P_0^q + D_n^q$	k_{t,H_2}^q
Monomer (M_i)	$P_{n,i}^q + M_i \rightarrow P_0^q + D_n^q$	k_{t,M_i}^q

An alternate model (McCoy & Rawatlal, 2011) that involves the presence of so-called of pseudo-can also be used to predict the broad MWD observed in Ziegler-Natta catalysed polymers, whilst maintaining the physically justifiable presence of one active site type that corresponds to the theory that relates active site to oxidation state. Instead of physical active centres where propagation rates differ, each terminating agent is seen as being attracted to terminate polymer growing at a specific site type, resulting in a MWD unique to that site type.

In summary, oxidation states of titanium molecules are constantly changing. The kinetic behaviour of the set of polymeric reactions that can occur at an active site differs changes with the oxidation number of that site. Furthermore, each reactant type in the bulk phase contributes to the polymeric reactions and site transfer reactions. The kinetic scheme proposed by (Rawatlal, 2004) and shown in Table 5-1 is devised such that it is flexible enough to be applied to most systems. In the present study, it applies to the system with one active site and multiple pseudo-sites, having different termination rates.

Using the nomenclature from the kinetic scheme, the evolution of the concentrations of sites on a particular catalyst particle as it ages in the reactor is described by Equations 5.3 – 5.5:

$$\frac{da^{4+}}{d\theta} = -\beta_{st}{}^{4,3} a^{4+} \tag{5.3}$$

$$\frac{da^{3+}}{d\theta} = \beta_{st}{}^{4,3} a^{4+} - \beta_{st}{}^{3,2} a^{3+} + \beta_{st}{}^{2,3} a^{2+} \tag{5.4}$$

$$\frac{da^{2+}}{d\theta} = \beta_{st}{}^{3,2} a^{3+} - \beta_{st}{}^{2,3} a^{2+} \tag{5.5}$$

$\beta_{st}{}^{q,r}$ is a lumped parameter used for the sake of simplicity that describes the contribution towards site transformations by all components present in the system:

$$\beta_{st}{}^{q,r} = k_{st,sp}{}^{q,r} + k_{st,A}{}^{q,r} [A]^{\alpha_A} + \sum_i k_{st,M_i}{}^{q,r} [M_i]^{\alpha_{M_i}} + k_{st,H_2}{}^{q,r} [H_2]^{\alpha_{H_2}} \tag{5.6}$$

The concentrations of each site type are normalised by the total concentration of sites, such that the concentrations are expressed as fractions. Initially, all the sites consist of Ti^{4+} atoms.

$$a^{4+}(0) = 1 \quad a^{3+}(0) = 0 \quad a^{2+}(0) = 0 \tag{5.7}$$

The analytical solution of the system of equations describing site transfers using these initial values is presented below as a function of age within the reactor:

$$a^{4+}(\theta) = \exp(-\beta_{st}^{4,3}\theta) \tag{5.8}$$

$$a^{3+}(\theta) = \alpha + \eta \exp(-\beta_{st}^{4,3}\theta) + \delta \exp[-(\beta_{st}^{3,2} + \beta_{st}^{2,3})\theta] \tag{5.9}$$

where

$$\alpha = -\frac{\beta_{st}^{4,3}\beta_{st}^{2,3} - (\beta_{st}^{2,3})^2 - \beta_{st}^{3,2}\beta_{st}^{2,3}}{(\beta_{st}^{3,2} + \beta_{st}^{3,2})(\beta_{st}^{3,2} - \beta_{st}^{4,3} + \beta_{st}^{2,3})}$$

$$\eta = -\frac{(\beta_{st}^{2,3})^3 - \beta_{st}^{4,3}\beta_{st}^{2,3} - \beta_{st}^{3,2}\beta_{st}^{4,3} + \beta_{st}^{3,2}\beta_{st}^{2,3}}{(\beta_{st}^{3,2} + \beta_{st}^{2,3})(\beta_{st}^{3,2} - \beta_{st}^{4,3} + \beta_{st}^{2,3})}$$

$$\delta = \frac{-\beta_{st}^{3,2}\beta_{st}^{4,3}}{(\beta_{st}^{3,2} + \beta_{st}^{2,3})(\beta_{st}^{3,2} - \beta_{st}^{4,3} + \beta_{st}^{2,3})}$$

and

$$a^{2+}(\theta) = 1 - a^{4+}(\theta) - a^{3+}(\theta) \tag{5.10}$$

Under typical industrial conditions, the concentrations of Ti^{4+} and Ti^{2+} sites are expected to undergo rapid exponential decay and growth, respectively, as the particle ages in the reactor.

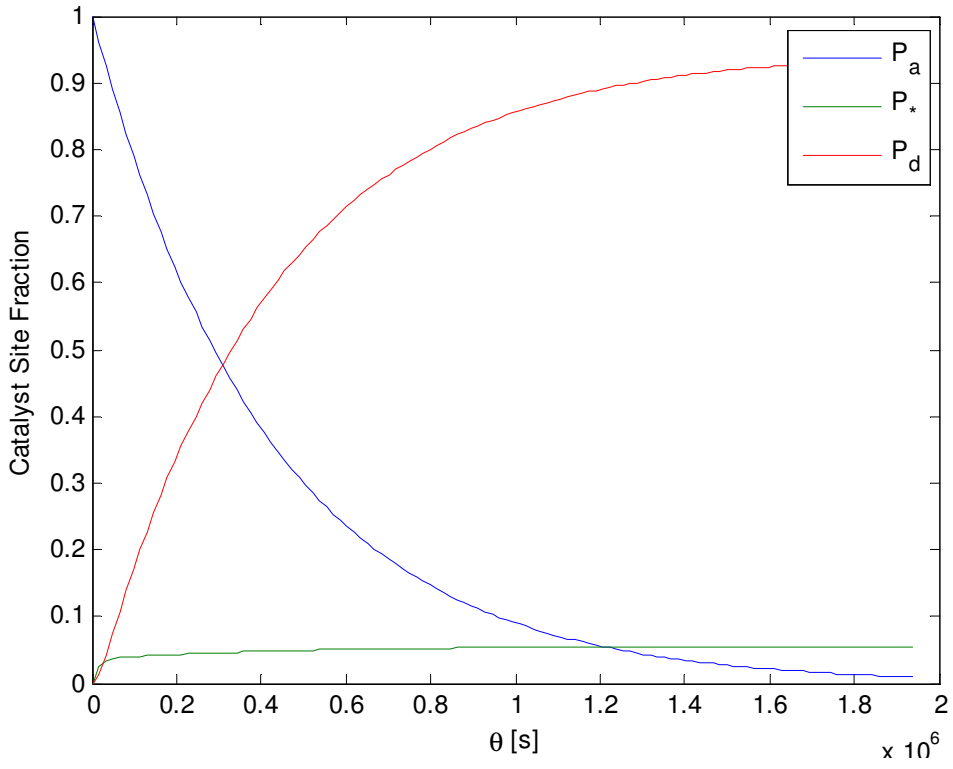

Figure 5.1 – Titanium oxidation state dynamics, single particle, typical industrial conditions

5.1.2 Live polymer chain growth

Having developed the model of catalyst activity, it is now possible to simulate the polymeric reactions at active sites in accord with the kinetic scheme previously presented. The concentrations of chains attached to active sites having a particular length, n, is affected by a number of reactions at any particular time. Active sites that are reduced to the lower oxidation state cause any present live chains to detach and accumulate around the catalyst particle, thus decreasing the concentration of live chains. The population of live chains with length n can increase via propagation if a monomer unit is incorporated on a polymer chain having length n-1. Conversely, the population can decrease if a monomer unit is incorporated on a chain with length n. Terminating agents break the bond between active site and live chain, thus reducing the concentration of live chains. The balances for sites with live chains having length n are separated according to whether monomer or comonomer was last incorporated. This is to account for the different propagation rates associated with each monomer type. The balances are further grouped according to the pseudo-site at which polymerisation is occurring. The balances for sites with attached chains of length 1 and n, ending in monomer i, are thus grouped according to terminating agent m as shown in Equations (5.11) and (5.12), respectively. The active site concentrations, vacant or occupied, are calculated with units of $\left[mol.mol_{Ti}^{-1} \right]$.

$$\frac{dP_{1,i}^{\ m}}{dt} = \beta_{st}^{\ 3,2} P_{1,i}^{\ m} + \sum_i k_{0,i} P_0^{\ m} M_i - \sum_j k_{p,ij} P_{1,i}^{\ m} M_j - k_{t,i}^{\ m} P_{1,i}^{\ m} C_m \qquad (5.11)$$

$$\frac{dP_{n,i}^{\ m}}{dt} = -\beta_{st}^{\ 3,2} P_{n,i}^{\ m} + \sum_j k_{p,ji} P_{n-1,j}^{\ m} M_i - \sum_j k_{p,ij} P_{n,i}^{\ m} M_j - k_{t,i}^{\ m} P_{n,i}^{\ m} C_m \qquad (5.12)$$

The live polymer moments of chains ending in monomer i on pseudo-site m are defined by Equation (5.13).

$$\lambda_{k,i}^{\ m} = \sum n^k P_{n,i}^{\ m} \qquad (5.13)$$

The zero moment is defined as the concentration of all chains from length 1 to chains having infinite length. By applying the definition of live moments to Equations (5.11) and (5.13), results in the rate of change for the zero moment of live chains where monomer i was last incorporated and was terminated by agent m as shown in Equation (5.14).

$$\frac{d\lambda_{0,i}^{\ m}}{dt} = -\beta_{st}^{\ 3,2} \lambda_{0,i}^{\ m} + k_{0,i} P_0^{\ m} M_i + \sum_j k_{p,ji} \lambda_{0,j}^{\ m} M_i - \sum_j k_{p,ij} \lambda_{0,i}^{\ m} M_j - k_{t,i}^{\ m} \lambda_{0,i}^{\ m} C_m \qquad (5.14)$$

In order to present Equation (5.14) in terms of measurable quantities, the variable representing the concentration of vacant sites, P_0^m, must be eliminated, since this value is physically difficult to determine. However, it can be replaced by the difference between the concentration of active sites $\left(P_*^m\right)$ and the concentration of all sites with chains attached:

$$P_0^m = P_*^m - \sum_j \sum_n P_{n,j}^m = P_*^m - \sum_j \lambda_{0,j}^m \tag{5.15}$$

Substituting Equation (5.15) into Equation (5.14) then yields the rate of change of the live 0th moment in terms of measurable variables:

$$\frac{d\lambda_{0,i}^m}{dt} = -\beta_{st}^{3,2}\lambda_{0,i}^m + k_{0,i}M_i(P_*^m - \sum_j \lambda_{0,j}^m) + \sum_j k_{p,ji}\lambda_{0,j}^m M_i - \sum_j k_{p,ij}\lambda_{0,i}^m M_j - k_{t,i}^m \lambda_{0,i}^m C_m \tag{5.16}$$

It is normal to assume that the polymerisation mechanism occurs rapidly enough to render the rate of change of site concentration as negligible. Thus, the quasi-steady state hypothesis (QSSH) can be applied. For a system with two monomer types, Equation (5.16) can be rearranged for the 0th moment of each monomer type as follows:

$$\lambda_{0,1}^m = \xi_1^m P_*^m \tag{5.17}$$

$$\lambda_{0,2}^m = \xi_2^m P_*^m \tag{5.18}$$

where

$$\xi_1^m = \frac{c - \frac{bf}{e}}{a - \frac{bd}{e}}$$

$$\xi_2^m = \frac{f - \frac{cd}{a}}{e - \frac{bd}{a}}$$

and

$$a = \beta_{st}^{3,2} + k_{0,1}M_1 + k_{p,12}M_2 + k_{t,1}^m C_m$$

$$b = k_{0,1}M_1 - k_{p,21}^q M_1$$

$$c = k_{0,1}M_1$$

$$d = k_{0,2}M_2 - k_{p,12}M_2$$

$$e = \beta_{st}^{3,2} + k_{0,2}M_2 + k_{p,21}M_1 + k_{t,2}^m C_m$$

$$f = k_{0,2}M_2$$

Physically, $\xi_i{}^m$ represents the fraction of catalyst sites occupied by chains ending in monomer i for pseudo-site m. Henceforth it will be referred to as the comonomer content and is defined by Equation (5.19).

$$\xi_i{}^m = \frac{\lambda_{0,i}{}^m}{P_*{}^m} \tag{5.19}$$

Equation (5.15) shows that the concentration of all active sites is determined from the sum of vacant sites and sites occupied by chains. If the concentration of vacant sites is taken to be negligible in comparison to that of the sites with polymers chains attached, then the comonomer content can be approximated by the ratio between the concentration of all sites occupied by chains with monomer i last inserted and the total concentration of sites occupied by chains:

$$\xi_i{}^m = \frac{\lambda_{0,i}{}^m}{\lambda_0{}^m} \tag{5.20}$$

It is assumed that the comonomer content is distributed evenly over the range of chain lengths; an implication of the long chain assumption. Thus, Equation (5.21) applies:

$$\xi_i{}^m = \frac{P_{n,i}{}^m}{P_n{}^m} \tag{5.21}$$

Substituting Equation (5.21) into Equations (5.11) and (5.12); applying the QSSH and rearranging the resulting equations, yields the set of equations required to calculate all the live chain length concentrations:

$$P_1^m = \frac{\sum_i k_{0,i} M_i P_0^m}{\sum_i (\xi_i^m (\beta_{st}{}^{3,2} + k_{t,i}{}^m C_m) + \sum_j k_{p,ij} \xi_i^m M_j)} \tag{5.22}$$

$$P_n{}^m = \frac{\sum_i \sum_j k_{p,ij} \xi_j{}^m M_i P_{n-1}{}^m}{\sum_i (\xi_i^m (\beta_{st}{}^{3,2} + k_{t,i}{}^m C_m) + \sum_j k_{p,ij} \xi_i^m M_j)} \tag{5.23}$$

From the previous equation, it is clear that the concentration of polymer chains having a specific chain length is dependent on the concentration of polymer chains having shorter chain lengths.

By recursion, P_n^m can be reformulated and represented by the following equation:

$$P_n^m = \varphi^m (\gamma^m)^n \tag{5.24}$$

where

$$\varphi^q = \left(\frac{\sum_i k_{0,i} M_i}{\sum_i (\xi_i^m (\beta_{st}^{3,2} + k_{t,i}^m C_m) + \sum_j k_{p,ij} \xi_i^m M_j)} \right) \frac{1}{\gamma^m} \tag{5.25}$$

and

$$\gamma^m = \frac{\sum_i \sum_j k_{p,ij} \xi_j^m M_i}{\sum_i (\xi_i^m (\beta_{st}^{3,2} + k_{t,i}^m C_m) + \sum_j k_{p,ij} \xi_i^m M_j)} \tag{5.26}$$

The term γ^m represents the ratio of the propagation rate to all reaction rates including that of propagation. Relative to all other reaction rates, propagation is the fastest by far, thus the parameter γ^m is always close to; but less than 1. The total concentration of live chains, i.e. the 0th moment is obtained by summing the concentration of sites for all chain lengths, which becomes an infinite convergence series. Thus, Equation (5.27) represented the formula for determining the total concentration of active sites with live chains attached without having to compute the entire distribution of concentrations:

$$\lambda_0^m = \sum_n P_n^m = \sum_n \varphi^m \left(\gamma^m\right)^n = \varphi^m \frac{\gamma^m}{1-\gamma^m} \tag{5.27}$$

Therefore, the fraction of live polymer chains attached to site m having chain length n is given as:

$$\nu_n^m = \frac{P_n^m}{\lambda_0^m} = \frac{\varphi^m (\gamma^m)^n}{\varphi^m \frac{\gamma^m}{1-\gamma^m}} = (\gamma^m)^{n-1} \left(1-\gamma^m\right) \tag{5.28}$$

The concentration of live polymer chains having a specific chain length can then be reformulated into Equation (5.29), where evaluation of the function for all n will yield the chain length distribution.

$$P_n^m = \nu_n^m \lambda_0^m = (\gamma^m)^{n-1} \left(1-\gamma^m\right) P_*^m \sum_i \xi_i^m \tag{5.29}$$

Using this formulation of a live polymer chain, the definition of moments given in Equation (5.13) can be simplified, since the summation becomes a converging geometric series for each moment. The simplifications are shown in Equations (5.30)-(5.32).

$$\lambda_{0,i}{}^{m} = \xi_i{}^{m} P_*{}^{m} \tag{5.30}$$

$$\lambda_{1,i}{}^{m} = \xi_i{}^{m} P_*{}^{m} \frac{1}{1-\gamma^m} \tag{5.31}$$

$$\lambda_{2,i}{}^{m} = \xi_i{}^{m} P_*{}^{m} \frac{1+\gamma^m}{(1-\gamma^m)^2} \tag{5.32}$$

In order to calculate the overall live polymer moments, the fraction that each pseudo-site contributes towards the overall concentration of sites needs to be determined. Equation (5.33) shows that this composition is the ratio of termination rate for terminating agent m to the sum of termination rates.

$$f^m = \frac{\sum_i k_{t,i}{}^{m} \xi_i{}^{m} C_m}{\sum_m \sum_i k_{t,i}{}^{m} \xi_i{}^{m} C_m} \tag{5.33}$$

By incorporating this parameter into the definition of the moments, the overall polymer moments can be found:

$$\lambda_0 = P_* \sum_m f^m \sum_i \xi_i{}^{m} \tag{5.34}$$

$$\lambda_1 = P_* \sum_m f^m \frac{1}{1-\gamma^m} \sum_i \xi_i{}^{m} \tag{5.35}$$

$$\lambda_2 = P_* \sum_m f^m \frac{1+\gamma^m}{(1-\gamma^m)^2} \sum_i \xi_i{}^{m} \tag{5.36}$$

Thus far, the growth of polymer chains on a single catalyst particle has been modelled for a single physical active site based on the oxidation state of titanium molecules on the supported catalyst and for multiple pseudo-sites based on terminating agents. The model is able to compute the CLD as well as its statistical moments. In the previous chapter these moments were related to the average properties relevant to defining a grade.

The various polymeric reactions that occur at the active site are dependent on the concentration of the reactants in the bulk phase. In addition, the distribution of dead polymer

chains accumulating around the particle differs over time as reactants are consumed. Further model development is required to determine the concentration of these reactants and dead chains and will be dealt with at a later stage.

5.1.3 Mass and heat transfer resistances

Meso-scale modelling involves the simulation of polymerisation behaviour on a particle level. Initially, the catalyst particle is introduced to the system with its entire surface exposed to the bulk phase. However, as polymerisation proceeds, an interlinked layer of live and dead chains form around the catalyst surface, the thickness of which increases as time progresses. Theoretically, this layer represents a form of mass and heat transfer resistance. It is expected that the polymer growth rate will vary significantly across a particle with significant mass and heat transfer limitations. However, researchers (Böhm, 1978; de Carvalho *et al*., 1990) have found that the influence of the mass and heat transfer rates on the polymer properties are negligible. The particle model can therefore be considered a lumped parameter model for which mass and heat transfer phenomena need not be considered.

5.1.4 Hydrodynamic considerations

Ziegler-Natta catalysed polymerisation involves the interaction between two phases, namely a bulk continuum containing the monomer and the population of solid catalyst particles. The flow patterns that evolve in each phase are governed by the reactor design and type. In the case of polymerisation, the flow pattern governs the length of time that a catalyst particle is subjected to reaction conditions. The longer the particle spends in the system, the more dead polymer chains accumulate around the particle. In turn, this affects the amount of reagents that are consumed throughout this period, which influences the nature of the polymer CLD.

Normally, the population balance model (PBM) is used to account for the particles flowing through the vessel and reacting with surrounding reagents. The use of this method requires a considerable amount of computational effort because it accounts for the behaviour of all the particles in the reactor at a particular time. A model that uses the segregation approach will yield the same result, but at a much lower computational expense.

By definition, catalyst particles enter the reactor system at age $\theta = 0$ and, due to the flow pattern, exit the reactor at different times. To approximate the mixing pattern in the vessel, a residence time distribution (RTD) is assumed based on the reactor type. Describing the system as perfectly mixed i.e. a continuously stirred tank reactor (CSTR) is a good starting point for modelling the flow pattern of slurry-phases polymerisation.

The internal RTD is given by Equation (5.37) (Fogler, 2006).

$$I(\theta) = \frac{1}{\tau}\exp(-\frac{\theta}{\tau}) \tag{5.37}$$

For a perfectly-mixed vessel, the internal RTD is the same as the external RTD as shown in Equation (5.38).

$$E(\theta) = \frac{1}{\tau}\exp(-\frac{\theta}{\tau}) \tag{5.38}$$

In essence, the RTD describes the probability of a particle having a specific age that is present either in the reactor or at the exit from the reactor. Figure 5.2 illustrates that this probability is highest as it enters the reactor and decays exponentially as the particle ages.

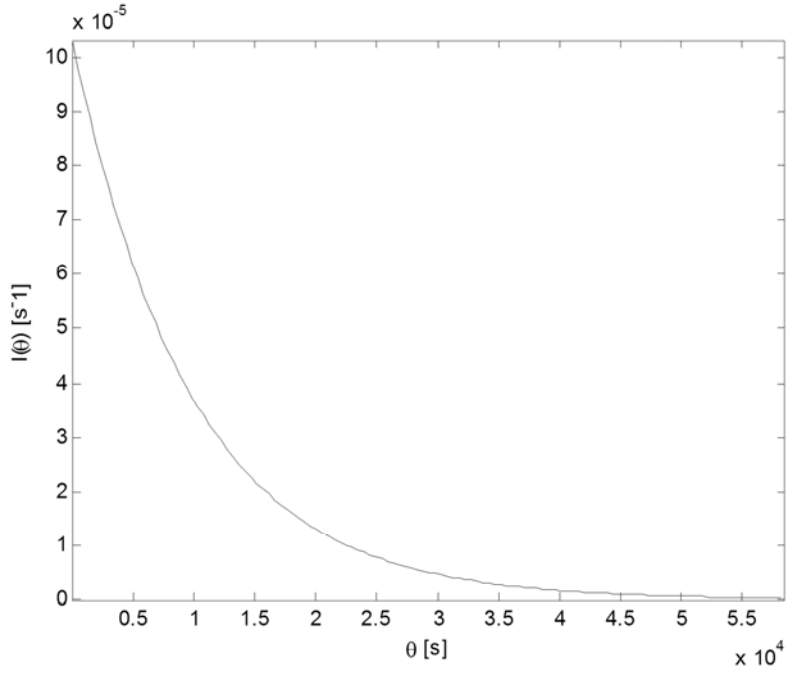

Figure 5.2 – Residence time distribution for a perfectly mixed vessel

Taking into account the overall interaction between the catalyst particles and the bulk phase due to hydrodynamics falls under the scope of macro-scale modelling. Suppose a particle has an arbitrary age-varying property $c(\theta)$. The mean value of this property over the entire age range is calculated by integrating with respect to age the product of the age-based property and the internal RTD from zero to infinity.

$$\overline{c} = \int_{\theta=0}^{\infty} I(\theta)c(\theta)\, d\theta \tag{5.39}$$

Since the contents in the exit steam of a perfectly-mixed vessel are the same as the contents within the vessel, Equation (5.39) is also applicable to define an average property in the exit stream. The segregation approach will be used at a later stage to link the micro-scale aspects of the model, i.e. polymer growth around a single catalyst particle, to the average to macro-scale aspects of the model, including hydrodynamics and vapour-liquid equilibrium.

5.1.5 Particle size distribution

The particle size of the fresh catalyst in the feed is seldom uniform. Polymer growth around the catalyst particle can increase the diameter of the particle more than tenfold and results in a final particle size distribution (PSD) different to that of the fresh feed PSD. Some post-processing units are sensitive to particle size, for example some are unable to handle fine particles. Thus, it is important to determine the PSD of the polymer product.

It is assumed that the feed PSD curve follows a normal distribution pattern and that the final particle size is dependent on the initial particle diameter and its age within the reactor. The mean particle size of the product $(\bar{l})$ will be determined by applying the segregation approach to the particle growth model (McCoy & Rawatlal, 2011). Assuming particle fragmentation and agglomeration do not occur, each particle undergoes growth based on the rate of polymerisation; its age within the reactor and the initial mass of catalyst. It is observed that the distribution curve shifts and skews towards the higher particle sizes as the catalyst ages.

The concentration of active centres on a single catalyst particle in terms of moles per particle volume is calculated using the initial particle mass, the titanium content, the catalyst density, the particle voidage and the fraction of active sites on the catalyst as shown in Equation (5.40).

$$P_*(t) = n_{Ti} \frac{m_{P,0}}{\rho_{cat}\varepsilon} a^{3+}(t) \quad [mol.m^{-3}] \tag{5.40}$$

The major constituents of a polymer chain are its monomer and comonomer units, thus the rate of polymerisation can be approximated as the rate of consumption of the two monomer types via propagation:

$$r_P = \sum_{i=1} \sum_{j=1} k_{p,i,j} P_*(t) C_{M_j} \tag{5.41}$$

Converting this polymerisation rate from a mole basis to a mass basis, yields the rate of change of polymer mass (Equation (5.42)), since all of the consumed reactants contribute towards the polymer mass.

$$\frac{dm_P}{dt} = r_P = \sum_{i=1}\sum_{j=1} k_{p,i,j} P_* (t) C_{M_j} MM_{M_j} \tag{5.42}$$

Substituting Equation (5.40) into the rate of polymerisation yields:

$$\frac{dm_P}{dt} = \eta_{Ti} \frac{m_{P,0}}{\rho\varepsilon} a(t) \sum_{i=1}\sum_{j=1} k_{p,i,j} C_{M_j} MM_{M_j} \tag{5.43}$$

For the sake of simplicity, the growth parameter G is defined in Equation (5.44).

$$G = \frac{\eta_{Ti}}{\rho\varepsilon} \sum_{i=1}\sum_{j=1} k_{p,i,j} C_{M_j} MM_{M_j} \tag{5.44}$$

Equation (5.43) can then be rewritten as:

$$\frac{dm_P}{dt} = m_{P,0} G a^{3+}(t) \tag{5.45}$$

Rearranging and integrating the left and right sides of Equation (5.45) with respect to time will yield the particle mass as a function of time.

$$m_P - m_{P,0} = m_{P,0} G \int_0^\theta a^{3+}(t)\, dt \tag{5.46}$$

It is necessary to integrate the function for active site fraction, which is given by Equation (5.4) for an initial value of 0. Equation (5.47) is the analytical solution to the integral.

$$\chi(\theta) = \int_0^\theta a^{3+}(t)\, dt = \alpha\theta + \gamma\left(\frac{1}{\beta_{st}^{4,3}}\left[e^{-\beta_{st}^{4,3}\theta} - 1\right]\right) + \delta\left(\frac{1}{\beta_{st}^{3,2} + \beta_{st}^{2,3}}\left[e^{-\left(\beta_{st}^{3,2} + \beta_{st}^{2,3}\right)\theta} - 1\right]\right) \tag{5.47}$$

The particle mass as a function of age simplifies to Equation (5.48).

$$m_P = m_{P,0}\left(\chi(\theta) G + 1\right) \tag{5.48}$$

Polymer particles are generally spherical, thus the mass of a particle can be related to the diameter of a particle as shown in Equation (5.49). This relationship can be used to

determine the catalyst size as a function of its initial diameter and age by substituting Equation (5.49) into Equation (5.48) for the initial mass and the general particle mass.

$$m_P = \frac{\pi}{6} l^3 \rho_{cat} \varepsilon \tag{5.49}$$

$$l(\theta, l_0) = l_0 \sqrt[3]{\chi(\theta) G + 1} \tag{5.50}$$

A surface plot of this function is presented in Figure 4.3. The surface can be viewed as a series of trajectories describing the growth of particles with different initial sizes as they age in the reactor. It is evident that feed particles with larger diameters grow at faster rates than smaller catalyst particles.

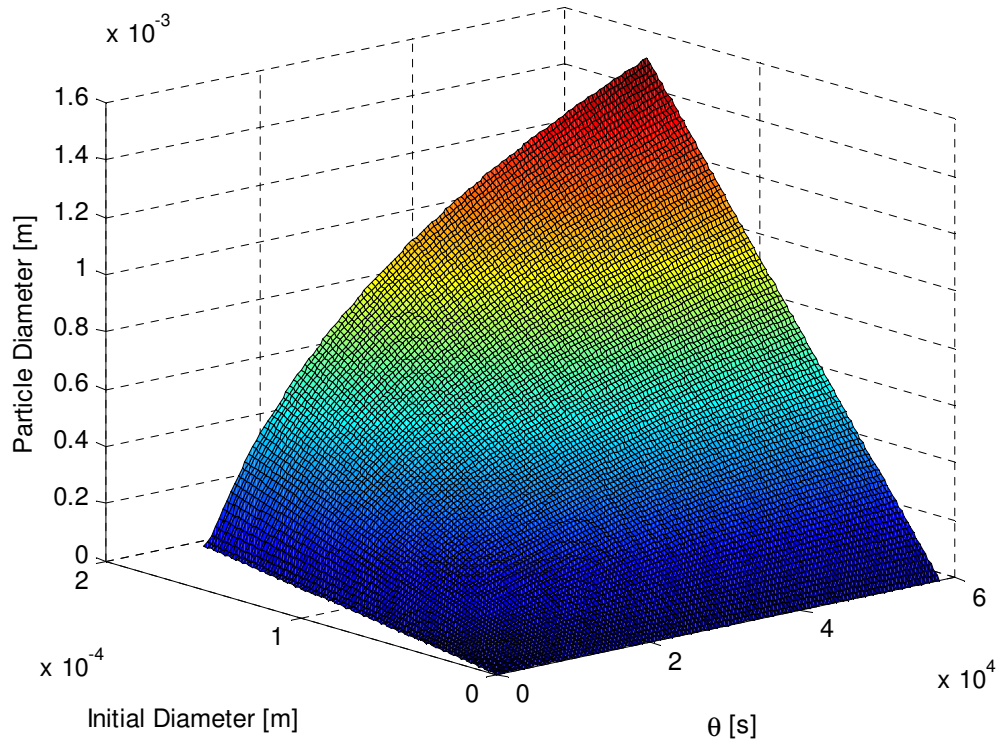

Figure 5.3 – Effect of age and initial diameter on particle size

Having determined an analytical solution to the particle size, it is now possible to calculate the mean diameter based on the segregation approach as shown in Equation (5.51). One can view it as calculating the mean diameter of particles with different initial sizes at each age and then using the RTD to produce the average over time.

$$\overline{l} = \int_0^\infty \int_0^\infty l(\theta, l_0) f_{m,0}(l_0) I(\theta) dl_0 d\theta \tag{5.51}$$

5.1.6 Bulk polymer moments

Thus far, a set of equations has been developed that determines the instantaneous polymer moments for a catalyst particle. However, in order to define the property specifications of a polymer product, it is necessary to determine the polymer moments of the layer of dead polymer chains that accumulate around a catalyst particle in addition to the instantaneous polymer moments. The distribution of dead chains around a particle differs depending on its age within the reactor. The overall contribution of each distribution is required to calculate the average polymer properties.

Dead polymer chains are created either by live chain termination or by active site transfer reactions. Thus, the rate at which dead polymer chains accumulate around a catalyst particle is given by Equation (5.52).

$$\frac{d\mu_k}{d\theta} = \left(\beta_{st}^{*,d} + \sum_m k_t^m C_m \right) \lambda_k \tag{5.52}$$

The dead polymer moment is a function of age and it is obtained by integrating the accumulation rate from 0 to θ:

$$\mu_k(\theta) = \left(\beta_{st}^{*,d} + \sum_m k_t^m C_m \right) \lambda_k \theta \tag{5.53}$$

The live chain polymer moments are time-independent due to the assumption that the QSSH applies. However, the concentration of dead polymer chains around a catalyst particle is related to the amount of time it spends in the reactor. As previously mentioned, an RTD curve is used to approximate the mixing behaviour of the solid catalyst particles in the vessel due to the flow pattern. The properties of the product within the reactor can be sufficiently described by averaging the properties over all particles. This is done using the segregation approach by integrating over all age ranges the product of the particle at a specific age and the corresponding probability of that particle having said age. Thus, the bulk polymer moments are calculated using Equation (5.54).

$$\bar{\mu}_k = \int_0^\infty I(\theta) \mu_k(\theta) \, d\theta \tag{5.54}$$

5.2 Reactor modelling

The equations that define the rest of the model are dependent on the reactor type and mode of operation. The current system is modelled as a perfectly mixed vessel for both steady-state and unsteady-state operation.

Thus far, the polymer moments have been calculated on a basis of moles per gram catalyst. In developing the material balances, the contribution of all particles to the bed of polymer must be accounted for. It is therefore necessary to change the original basis to account for these macro-scale modelling aspects. The conversion from active site concentration on a particle to active site concentration in the particle bed is shown in the following equation.

$$P_*^b = P_* \frac{\eta_{Ti} \dot{m}_{cat}}{H_0} \quad \left[mol.m^{-3} \right] \tag{5.55}$$

In a similar manner, the bulk polymer moments are converted to this new basis of moles per cubic metre in the reactor. These bulk moments are used to calculate the reaction rates for the material balances and also to calculate M_n and PDI that define the polymer product spec.

5.2.1 Vapour-liquid equilibrium

The liquid phase serves as the bulk continuum in which monomer is consumed for polymerisation. However, some of the reactants enter the vessel in the gaseous phase. Thus, it is necessary to estimate the vapour-liquid equilibrium to obtain the correct reactant concentrations for the polymeric reactions shown in the kinetic scheme. For this purpose, the Peng-Robinson equation of state is used. It correlates temperature and pressure to gas and liquid densities and volume splits, allowing the concentrations in each phase to be determined.

The concentrations in the respective phases are determined via an isothermal flash calculation for a multi-component mixture. The method is based on the algorithms and heuristics given in Sandler (2006). The parameters for vapour pressure, critical pressure and critical temperature are obtained from Perry and Green (2007). Binary interaction parameters are obtained from both of the above-mentioned sources.

5.2.2 Steady-state material balances

For steady-state operation in a CSTR, the material balance of each reactant is described by an algebraic equation. The summation of flow of reactant into and out of the system together with the consumption or generation of reactant, due to reaction, must be zero.

The set of material balances governing the system are presented below:

$$0 = F_{TEA} - \frac{N_{TEA}}{\tau} - r_{TEA} M_{cat} \tag{5.56}$$

$$0 = F_{H_2} - \frac{N_{H_2}}{\tau} - r_{H_2} M_{cat} \tag{5.57}$$

$$0 = F_{M_1} - \frac{N_{M_1}}{\tau} - r_{M_1} M_{cat} \tag{5.58}$$

$$0 = F_{M_2} - \frac{N_{M_2}}{\tau} - r_{M_2} M_{cat} \tag{5.59}$$

$$0 = \frac{N_P}{\tau} - r_p M_{cat} \tag{5.60}$$

The consumption and generation terms in each material balance are given in the following equations:

$$r_{TEA} = C_{TEA} \sum_i k_{t,i,TEA} \lambda_{0,i}{}^b \tag{5.61}$$

$$r_{H_2} = C_{H_2} \sum_i k_{t,i,H_2} \lambda_{0,i}{}^b \tag{5.62}$$

$$r_{M_1} = C_{M_1} \sum_i \left(k_{t,i,M_1} + k_{p,i,M_1} \right) \lambda_{0,i}{}^b \tag{5.63}$$

$$r_{M_2} = C_{M_2} \sum_i \left(k_{t,i,M_2} + k_{p,i,M_2} \right) \lambda_{0,i}{}^b \tag{5.64}$$

$$r_P = r_{TEA} + r_{H_2} + r_{M_1} + r_{M_2} \tag{5.65}$$

Components TEA and H_2 are thought to be consumed mainly by chain termination, whereas monomer and comonomer are thought to be consumed via both chain termination and propagation. The generation of polymer is seen as the summation of the consumption of all reactant types in the system, thus the summation of their respective rates determines the rate of polymerisation.

The mass balances governing the system form a set of algebraic equations that are solved in MATLAB using the *fsolve* function. This function is able to solve a system of nonlinear equations containing several variables.

5.2.3 Unsteady-state material balances

Reactor models that can predict dynamic behaviour are useful tools for industries in terms of process optimisation. On a particle level, the model remains unchanged from that described in Section 5.1. The reaction mechanism still applies, but the concentrations of the components in the bulk phase surrounding the particles are time dependent. Furthermore, the holdup and RTD are also time-varying. The following unsteady-state mass balances now govern the system.

$$\frac{dN_{TEA}}{dt} = F_{TEA,0} - r_{TEA}M_{cat} - \frac{N_{TEA}}{\tau} \tag{5.66}$$

$$\frac{dN_{H_2}}{dt} = F_{H_2,0} - \frac{N_{H_2}}{\tau} - r_{H_2}M_{cat} \tag{5.67}$$

$$\frac{dN_{M_1}}{dt} = F_{M_1,0} - r_{M_1}M_{cat} - \frac{N_{M_1}}{\tau} \tag{5.68}$$

$$\frac{dN_{M_2}}{dt} = F_{M_2} - r_{M_2}M_{cat} - \frac{N_{M_2}}{\tau} \tag{5.69}$$

$$\frac{dN_P}{dt} = N_{P0} - r_pM_{cat} - \frac{N_P}{\tau} \tag{5.70}$$

$$\frac{d\mu_k{}^b}{dt} = \left(\beta_{st}{}^{*,d} + \sum \beta_t\right)\lambda_k{}^b - \frac{\mu_k{}^b}{\tau} \tag{5.71}$$

The balance for the polymer moments in the particle bed inside the reactor is given by Equation (5.71). It is a modification of the dead chain accumulation term given in Equation (5.52) to account for the flow of polymer out of the reactor. The live polymer moments of a particle are still converted to a basis that represents the bulk phase as shown in Equation (5.55).

With unsteady-state operation, the pseudo-steady state hypothesis still applies for the polymerisation reactions on a particle surface and the segregation approach is still used to account for the flow of particles through the system, since the development of unsteady-state RTD (Rawatlal & Starzak, 2003) has made it possible to extend these principles to dynamic systems. For the time being, the change in holdup and the total flowrate is assumed to be

minimal, thus the change in RTD over time is negligible. Therefore, a constant RTD profile is used over all time instances for this model.

Since the mass balances are now described by a set of differential equations as opposed to the set of algebraic equations for steady-state operations, the *ode45* function in MATLAB is used for its ability to solve non-stiff differential equations. This function makes use of the 5th order Runge-Katta method.

5.2.4 Continuous blending

In Section 4.1, equations for predicting the properties of a blend of two grades in batches were proposed. These equations are hereby modified to consider the blending of two polymer streams, namely the off-spec polymer stream (subscript O) from the silos and the virgin polymer stream (subscript R) from the reactor as can be seen in Figure 5.4. This mode of operation negates the need for intermediate storage tanks prior to a blending tank on a continuous polymerisation plant.

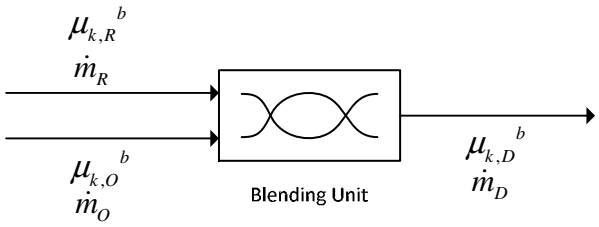

Figure 5.4 - Continuous blending concept

The governing equations for blending two grades in a perfectly mixed CSTR under dynamic conditions are presented below:

$$\frac{dM_D}{dt} = \dot{m}_R + \dot{m}_O - \frac{M_D}{\tau} \tag{5.72}$$

$$\frac{d\left(M_D \cdot \mu_k{}^{D}\right)}{dt} = \dot{m}_R \mu_{k,R}{}^{b} + \dot{m}_O \mu_{k,O}{}^{b} - \frac{M_D}{\tau}\mu_k{}^{D} \tag{5.73}$$

A quasi-steady-state approach is used to determine the mean residence time based on the holdup and volumetric flowrate in the vessel. The holdup is assumed to be constant, but the volumetric flowrate is determined by the time-varying mass flowrate.

$$\nu_0 = \frac{\left(\dot{m}_R + \dot{m}_O\right)}{\rho_{cat}} \tag{5.74}$$

$$\tau = \frac{H_0}{\nu_0} \tag{5.75}$$

Expansion of the dot product in, substitution of the mass balance; Equation (5.72); into and rearrangement of Equation (5.73) gives an expression for the rate of change of blended moments; Equation (5.78).

$$M_D \frac{d\mu_{k,D}{}^b}{dt} + \mu_{k,D}{}^b \frac{dM_D}{dt} = \dot{m}_R \mu_{k,R}{}^b + \dot{m}_O \mu_{k,O}{}^b - \frac{M_D}{\tau} \mu_{k,D}{}^b \tag{5.76}$$

$$M_D \frac{d\mu_{k,D}{}^b}{dt} + \mu_{k,D}{}^b \left(\dot{m}_R + \dot{m}_O - \frac{M_D}{\tau} \right) = \dot{m}_R \mu_{k,R}{}^b + \dot{m}_O \mu_{k.O}{}^b - \frac{M_D}{\tau} \mu_{k,D}{}^b \tag{5.77}$$

$$\frac{d\mu_{k,D}{}^b}{dt} = \frac{\dot{m}_R}{M_D} \left(\mu_{k,R}{}^b - \mu_{k,D}{}^b \right) + \frac{\dot{m}_O}{M_D} \left(\mu_{k,O}{}^b - \mu_{k,D}{}^b \right) \tag{5.78}$$

Under steady-state conditions, the accumulation term falls away and Equation (5.78) reduces to the blending rule proposed in Equation (4.10).

5.3 Summary

In the current chapter, the particulars involving the simulation of polymerisation on an industrial scale for both steady-state and transient conditions are presented. A model is developed that is able to predict the bulk moments and PSD of the precipitating polymer in a slurry-phase reactor from the catalyst; co-catalyst and reactant feed rates, temperature and pressure. Development of the model is based on fundamental principles that take into account catalyst activity, reaction kinetics, vapour-liquid equilibrium and hydrodynamics. The segregation approach provides an important link between discrete polymer particles and the bulk fluid phase for solving the overall mass balances within a quicker timeframe than population balance approaches. Finally, the blending model that was previously developed in Chapter 4 has been modified to predict the bulk polymer moments during continuous flow operation from its original batch mode form.

6 CONTROL STRATEGIES FOR OFF-SPEC BLENDING

Thus far, the optimisation of off-spec blending has only been proposed in a general sense. In the objectives chapter it states that off-spec blending involves the mixing of off-spec polymer with a powder from the reactor to obtain a product within spec.

In the current chapter, three procedures for blending off-spec polymer on a plant are proposed in more detail; its optimisation strategies are explained and the objective functions and optimisation algorithms specific to each procedure are presented.

A robust process model, such as the polymerisation model developed in Chapter 5, consists of several coupled non-linear ODE's. As a result, it is difficult; if not impossible; to directly determine optimal control variable trajectories that fulfils a specified objective. As such, this dynamic optimisation problem will be solved using numerical techniques rather than by determining an analytical solution. Numerical techniques are beneficial in the sense that they are relatively easy to apply to any system, thus the control strategies proposed in the current chapter are general enough to be applied to any polymerisation system. More specifically, time-varying profiles of the control variables are posed as parameters to be optimised in an NLP problem. A host of optimisation algorithms are available to determine the best parameters for minimising a given objective function subject to a process model and system constraints. Some of those applied to polymerisation systems were discussed in Section 2.3.

In all proposed cases, it is assumed that the source of off-spec powder has been homogenised prior to blending by adequate mixing such that analysis of a sample taken from anywhere in the storage tank will have the same properties.

6.1 Steady-state off-spec blending

The first proposal for off-spec blending is presented in the current section. It is suggested that two streams be mixed under steady-state operation such that a continuous stream of the resultant blend is within a desired spec. One stream flows from the off-spec storage tank and the other stream flows from the reactor, as illustrated in Figure 6.1.

6.1.1 Scheme description

The scheme is analogous to a closed-loop control system where the set points are the M_n and PDI values of the polymer in the blended stream, denoted by subscript D, and are set to that of the desired grade's properties. The reactor exit stream, denoted by subscript R, has polymer properties equivalent to the blended stream when no blending is performed i.e. under normal operating conditions. The reactor feed rates and conditions are maintained at a

specific combination associated with producing a particular grade. The off-spec stream, denoted by subscript O, is viewed as a disturbance variable, which affects the output of the blending unit.

Continuing the analogy of the close-loop control system, off-spec polymer that is introduced into the system creates a deviation of the average polymer properties in the blended stream from their respective set points. At this stage, control action is to be taken that will vary the feed rates of monomer, comonomer, co-catalyst, hydrogen and diluent to adjust the product properties in the reactor exit stream. The resultant blend of the off-spec polymer and this adjusted reactor product should return the average properties in the blended stream to their respective set point values.

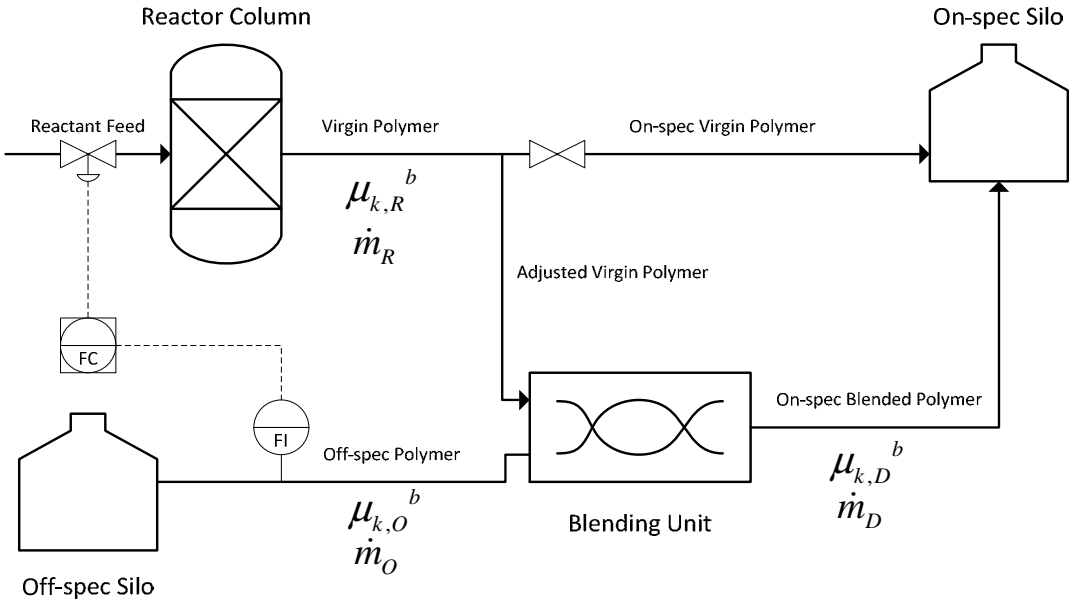

Figure 6.1 – Process flow diagram of continuous steady-state off-spec blending

6.1.2 Objective function

The objective function given in Equation (6.1) is posed such that its minimum value is zero. In addition, the objective function is designed to allow only the optimal solution to the blending scheme to have this minimum value of zero.

$$J = \left(1 - \frac{M_n^{\ D}}{M_n^{\ *}}\right)^2 + \left(1 - \frac{PDI^D}{PDI^*}\right)^2 \tag{6.1}$$

The desired average properties, denoted by $M_n^{\ *}$ and PDI^*, are chosen to define the polymer grade and it is desired that the actual average properties of the polymer flowing out

of the blending tank, M_n^D and PDI^D, match these values. If the values do match, then the objective function, shown in Equation (6.1), is minimised to zero.

The objective function is presented as a sum of weighted deviations where the weighting parameters are determined *a priori*, as discussed in Section 2.2.2 of the literature review. In the above equation, the ratio between a property and its desired value is presented as an alternative to using weighting parameters. Thus, each error is brought to the same order of magnitude such that there is no bias towards any particular property.

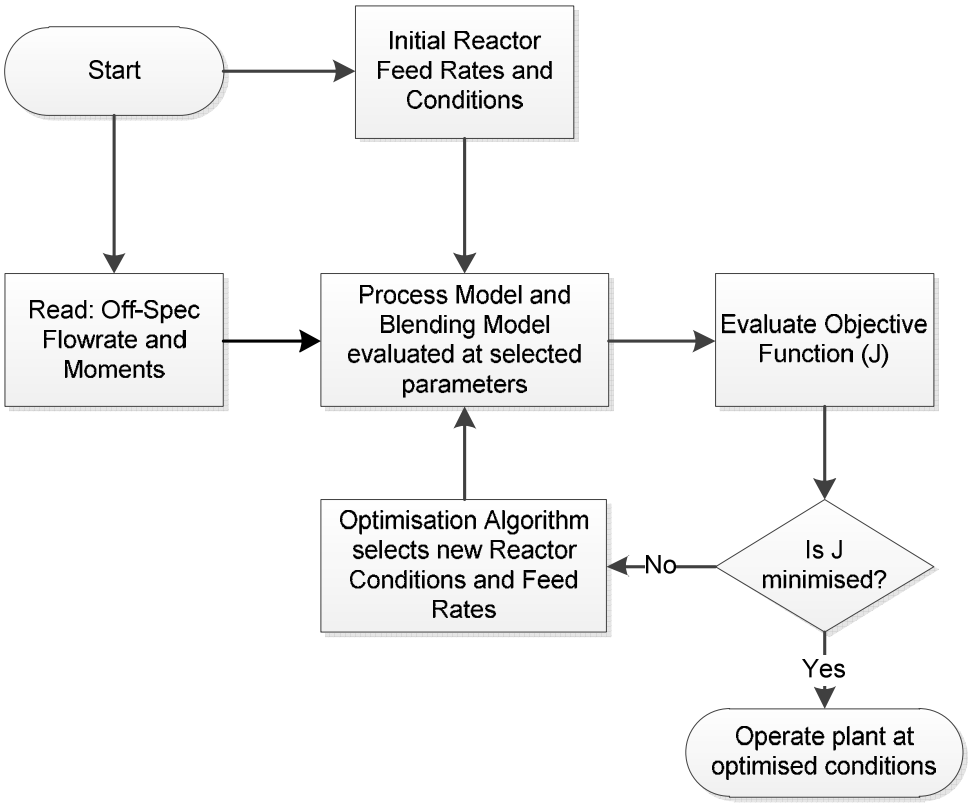

Figure 6.2 – Continuous steady-state off-spec blending algorithm

6.1.3 Method of optimisation – direct search method

For this problem, the *fminsearch* function in MATLAB is used to determine the optimal parameters by minimising the objective function given in Equation (6.1). It applies the well-known Nelder-Mead optimisation algorithm (Nelder & Mead, 1965). Due to the nature of this algorithm, solutions often converge on local minima where the optimal parameters are near the initial guess in the search space. This is acceptable in this case since it is desired that the adjusted feed rates have not differed greatly from that of the nominal steady-state values. The aim is to avoid solutions that could cause system instability.

The procedure for off-spec blending from the perspective of the controller is given in Figure 6.2. Successful implementation requires the choice of an adequate objective function as well as a suitable NLP solver.

6.2 Off-spec blending during grade transition

The common approach to off-spec polymer production minimisation is to reduce the transient period during grade transitions. An alternative approach is proposed in the current section whereby off-spec material is blended with fresh material from the reactor during a grade transition in order to produce an on-spec polymer blend. The benefits of this approach are twofold: Apart from reducing the amount of new off-spec material that is produced, the process also facilitates the consumption of stored off-spec polymer in the process.

6.2.1 Scheme description

While the previous scheme was analogous to a closed-loop feedback control system, the current proposal resembles that of an offline control system. Here, a process model predicts the trajectories of the polymer properties that are relevant to determining its grade specifications during the transition. Figure 6.3 illustrates the concept.

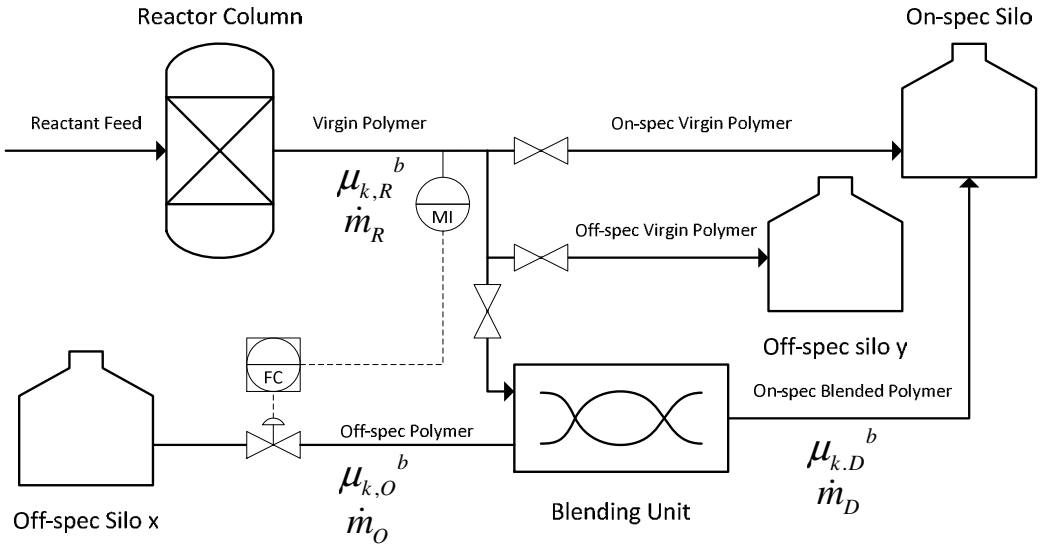

Figure 6.3 – Process flow diagram of concept of blending during grade transitions

The controller predicts the optimal mass flowrate trajectory of off-spec polymer from storage, denoted by stream O, to blend with fresh off-spec material from the reactor, stream R, in order to create as much product within spec, denoted by stream D, over the transition period.

6.2.2 Objective function

Under unsteady-state operation, an effective performance index is not only formulated to account for the property deviation at the final time, but is also formulated to consider the deviations over the entire transient period. Traditionally, it has been formulated in the form of an integral of the property deviation function over the transient time period as shown in Equation (6.1):

$$J = \int_{t_0}^{t_f} \left(1 - \frac{M_n{}^D}{M_n{}^*} \right)^2 + \left(1 - \frac{PDI^D}{PDI^*} \right)^2 dt \tag{6.2}$$

However, an objective function based on property deviations, such as that shown in Equation (6.2), cannot effectively indicate the performance of blending under dynamic conditions.

Objective functions of this form will promote blending at any stage during the transition as long as the blended properties are closer to their respective desired properties than normal operation, whether the resultant blend is on-spec or not. An alternative formulation is required in order to prevent this unnecessary blending.

In the previous scheme, a polymer grade was defined by a single value for each relevant property. In this scheme, the specification of a polymer is relaxed to include a range of properties. If at least one polymer property falls outside its respective specification band, it is considered off-spec. Boolean logic is used to keep track of whether a polymer is off-spec or not, subject to the following rules:

$\dot{m}_{off}(t) = \dot{m}_D$ when $M_n{}^{*,u} < M_n{}^D(t) < M_n{}^{*,l}$ and $PDI^{*,u} < PDI^D(t) < PDI^{*,l}$

$\dot{m}_{off}(t) = 0$ when $M_n{}^{*,u} > M_n{}^D(t) > M_n{}^{*,l}$ and/or $PDI^{*,u} > PDI^D(t) > PDI^{*,l}$

$$J = \int_{t_0}^{t_f} \dot{m}_{off} dt \tag{6.3}$$

Evaluating the objective function for the case where no blending is performed will yield the mass of off-spec material produced during a traditional grade transition. A flow profile of off-spec polymer that promotes the generation of a blended product within spec will reduce the overall mass of off-spec polymer that is produced. Conversely, a flow profile for off-spec polymer where the resultant blend remains off-spec at any particular time will remain off-spec, thus it will increase the value of the objective function by an amount proportional to the

mass that is added to the system. Therefore, the objective function presented above aims to facilitate effective off-spec blending and inhibit unnecessary blending.

6.2.3 Method of optimisation – control vector parameterisation

The type of problem outlined in Section 6.2.1 falls under the category of dynamic optimisation where a time-varying parameter, which is the mass flowrate of stream O in this case, is to be optimised. The popular techniques used to solve these problem types were covered in Section 2.3. This particular scheme has been formulated as an NLP by applying CVP to the off-spec flowrate, detailed in Section 2.2.2.

One of the easiest approaches to discretising the time-varying profile is to create a series of step functions. This approach is used in the current study for its simplistic purposes, since the main aim of the study is proof of concept. The number of parameters will be decided beforehand, where each value of the parameter represents the height of a step function over a specified time interval. The step functions form a series over equal time intervals covering the specified time range. Thus, the flow profile of the component will switch between constant values over equal time intervals. Better approximations by more complicated functions, such as splines, that produce more accurate representations of the true optimal profile can be used at a later stage.

The polymer property trajectories of the reactor product are obtained by evaluating the process model for a grade change from initial time, t_0 , to a final time, t_f , chosen sufficiently large enough for steady-state to be reached. A point exists (t_s) within this time range that signifies the time at which on-spec polymer is produced. Therefore, the blending profile is discretised over the period t_0 to t_s. The off-spec flowrate profile is approximated by a step function at each interval over this range where the amplitudes of each step function are the input parameters to be optimised. Figure 6.4 summarises the algorithm for finding the optimal solution.

The use of an algorithm that is prone to converge on local minima was convenient for the previous scheme because the general region of the optimal solution was known. In this case, no knowledge of the position of the optimal solution in the search space exists. The search space may be large and multimodal. Thus, an optimisation algorithm that is known for converging on global solutions is required. The differential evolution (DE) algorithm (Storn & Price, 1997) was chosen to perform this task due to its ability to easily handle non-linear, complex process models with multiple optima; as well as its ability to implement parallel processing for faster runtimes. MATLAB source code has been developed by Buehren (2008) for the DE algorithm where modifications to the original algorithm were made to take

advantage of its ability to evaluate objective functions in parallel. The MATLAB code was is attached in Appendix D.

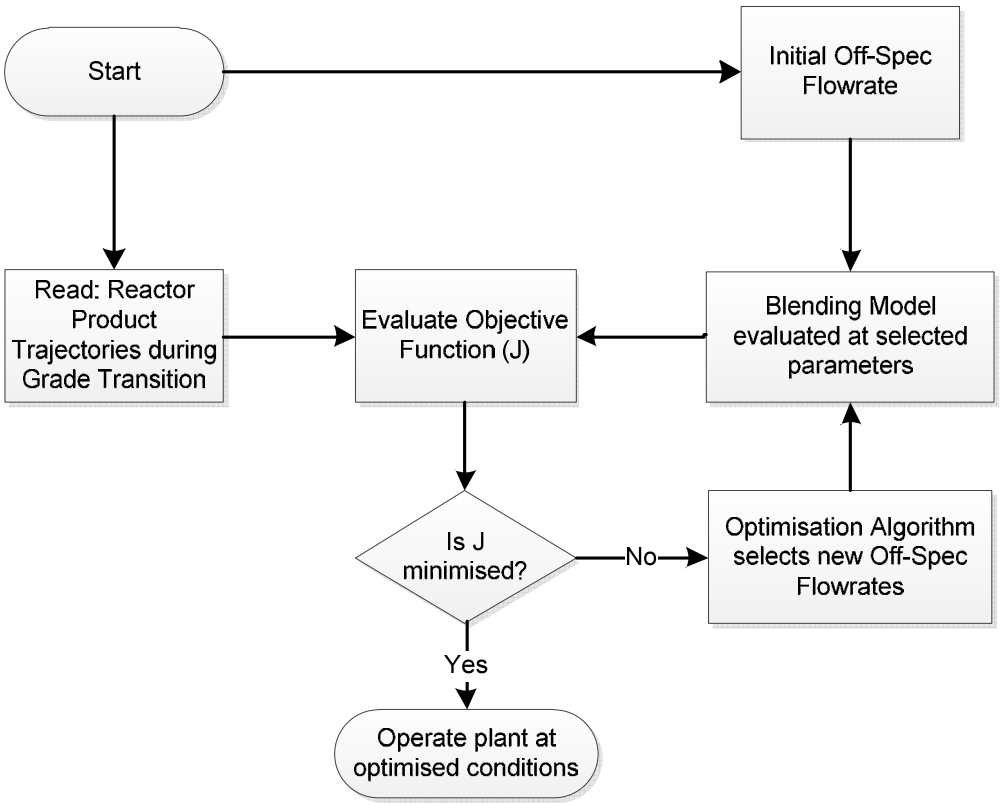

Figure 6.4 – Algorithm for off-spec blending during grade transitions

6.3 Off-spec blending during optimal grade transition

A large body of academic work has been focused on optimising the grade transition process by manipulating input trajectories, as shown in the literature review, specifically in Section 2.3. Grade transition optimisation is done either by explicitly minimising the amount of off-spec polymer formed or by minimising the time taken to reach steady-state for the new grade. The consequence of which, is the increase in overall production of fresh polymer.

The main aim of grade transition optimisation is to determine the appropriate profile for selected feed component flowrates that will bring the system to the required steady-state in the least amount of time and/or producing the least amount of off-spec polymer during the transition period. Both grade transition optimisation and dynamic off-spec blending share the same goal of, ultimately, minimising off-spec polymer generated. Additionally, both fields share the same technical approach to minimising this goal. It is for these reasons that, in the final scheme presented, it is desired to determine both the optimal reactor trajectories for grade transitions and the optimal off-spec flowrate trajectory for blending.

6.3.1 Scheme description

With reference to Figure 6.5, optimal time-varying profiles for reactant feed rates and the off-spec flowrate in stream O are determined offline by the controller in order to generate the least amount of off-spec powder during a grade transition. Manipulation of these variables is based on the property trajectories of stream D, at the exit of the blending unit.

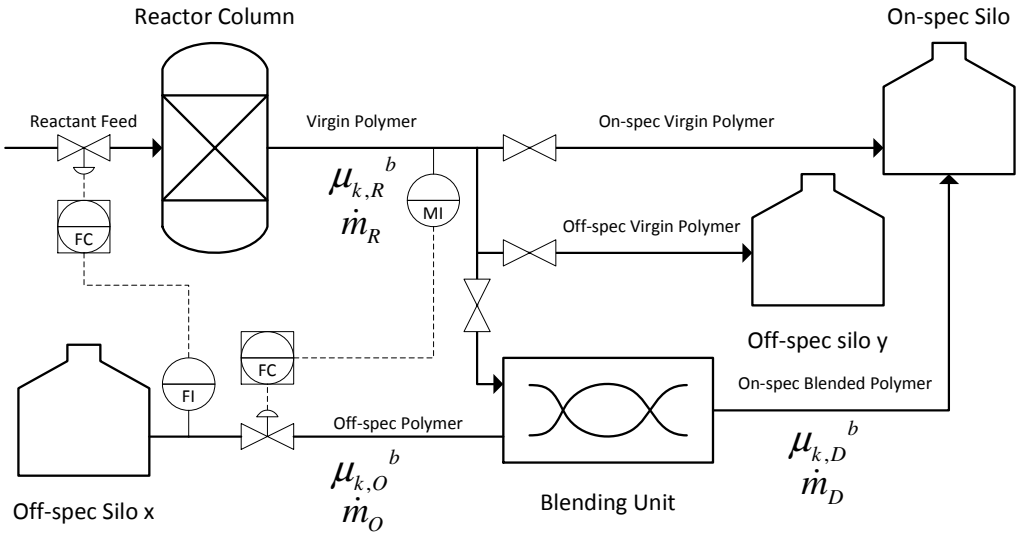

Figure 6.5 – Concept of simultaneous optimal grade transition and dynamic off-spec blending

6.3.2 Objective functions

Since the goal of both grade transition optimisation and off-spec blending is to reduce the mass of off-spec material generated, the objective function defined in the previous scheme is used for both optimisation stages in the current scheme.

6.3.3 Method of optimisation – control vector parameterisation

Using a similar approach to that in Section 6.2, the scheme is formulated as an NLP by applying CVP. Not only is the off-spec flowrate in stream O discretised, but so are feed rate profiles such as hydrogen, monomer, comonomer, co-catalyst and diluent. All of the true optimal profiles of these flowrates are approximated by series of step functions with equal time intervals, the amplitudes of which are the subject of the optimisation algorithm.

The set of differential equations describing the process model are integrated over a time range chosen sufficiently large enough so as to allow the system to reach steady-state $\left[t_0, t_f\right]$. The feed rate profiles are discretised over this entire time range, whereas the off-spec flowrate profile is discretised over the time range spanning the initial time until the time the reactor produces its first on-spec product for the new grade $\left(t_s\right)$. This time range is

chosen for the latter profile to prevent the algorithm from converging on a solution where off-spec material is blended with product that is already within specification.

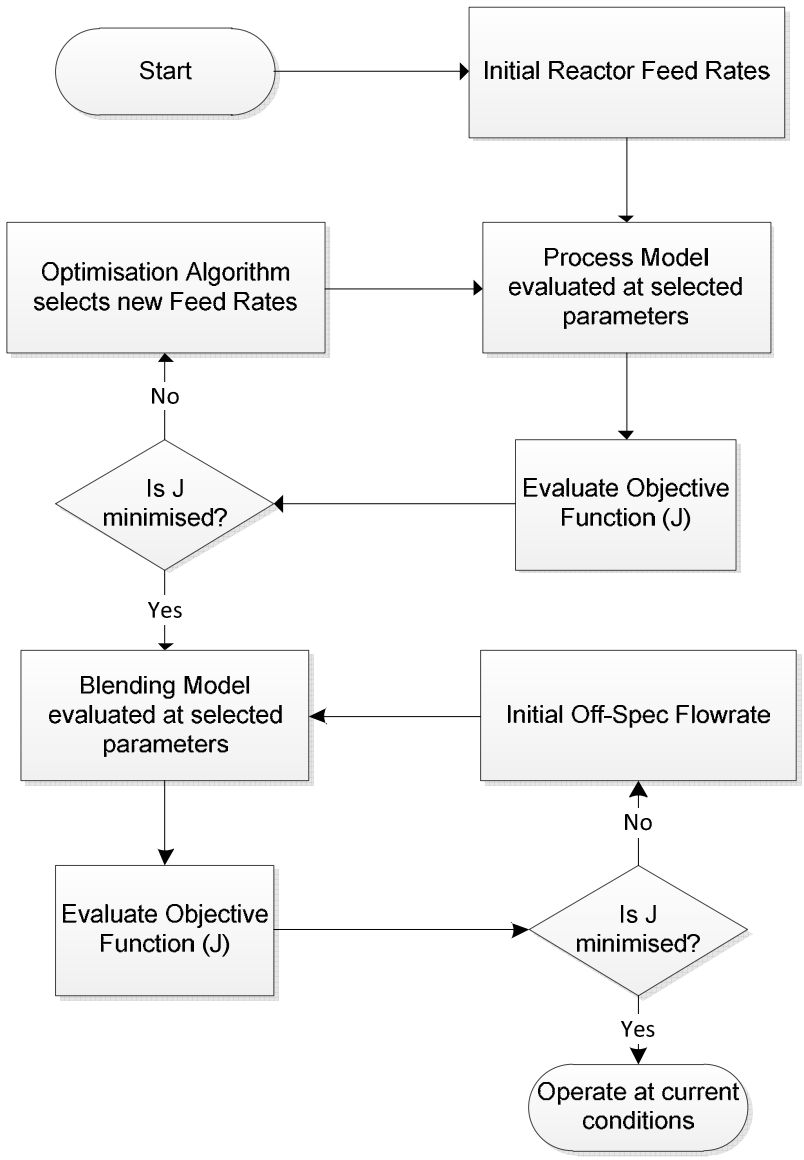

Figure 6.6 – Sequential grade transition and off-spec blending optimisation algorithm

The optimisation procedure has a two-stage design. The first stage involves optimising the grade transition by manipulating the feed rate profiles. These optimal profiles will be used as inputs to the second stage where blending is optimised by determining the optimal off-spec flowrate. Comparing the algorithm for this case, as shown in Figure 6.6, to the second stage of the algorithm for dynamic blending, shown in Figure 6.5, it can be seen that the approaches are, effectively, the same. As in the previous scheme, the differential evolution

algorithm is used to solve the optimisation problem. It is the dimensionality of the problem that increases due to the comparatively large number of parameters to be optimised in this case. Thus, one can expect a greater amount of evolutionary generations to pass than in the previous scheme before convergence is achieved.

6.4 Summary

Three schemes that involve the blending of off-spec polymer from storage with fresh powder from the reactor were presented in detail. The first scheme proposed that the reactor conditions and feed rates be adjusted to accommodate a continuous steady stream of off-spec polymer, such that the resultant blend is on-spec. The second scheme is designed as a controller that determines the optimal off-spec flowrate profile for blending during a grade transition such that a portion of off-spec material produced during a grade transition is blended to create a polymer within spec. The third scheme is an extension of the second scheme in that the reactor feed rate profiles are also optimised to minimise the transient period over which said off-spec blending is to occur.

In each case, the objective function was carefully formulated in order to best portray the performance of the system in relation to its goal. Further, a suitable algorithm was chosen for the schemes based on whether local or global optimisation was necessary. The well-known Nelder-Mead algorithm was used for the first case, where convergence around a local minimum in the region of the initial state was actually preferred. For the latter two schemes, the relatively new differential evolution algorithm was implemented with parallel processing due to its ability to converge towards global extremes relatively faster than other global optimisers.

7 RESULTS AND DISCUSSION

The primary goal of this study requires control strategies- that optimise off-spec blending to be blended. Chapter 6 presented and described three methods whereby optimal off-spec blending can be performed. Each method is applied and its effectiveness analysed in the current chapter by way of polyethylene production case studies.

7.1 Model parameters

Table 7-1 lists the model parameters, reactant properties and system conditions used in the process model to simulate the Ziegler-Natta catalysed polymerisation of ethylene with 1-butene in the slurry-phase. The reaction rate constants were determined by McCoy (2012) in accordance with the mechanism proposed by Rawatlal (2004) as shown in Table 5-1.

Table 7-1: Model Parameters and Properties for LLDPE polymerisation

Rate Constants		*sp*	*TEA*	M_1	M_2	H_2	*dil*
k_0	$[m^3.s^{-1}.mol^{-1}]$	-	-	90	1	-	-
$k_p^{M_1}$	$[m^3.s^{-1}.mol^{-1}]$	-	-	4.40	0.14	-	-
$k_p^{M_2}$	$[m^3.s^{-1}.mol^{-1}]$	-	-	68.9	0	-	-
$k_{st}^{a,*}$	$[m^3.s^{-1}.mol^{-1}]$	0	$5.17\text{x}10^{-2}$	0	0	0	-
$k_{st}^{*,d}$	$[m^3.s^{-1}.mol^{-1}]$	0	0	0	0	$1.64\text{x}10^{-2}$	-
$k_{st}^{d,*}$	$[m^3.s^{-1}.mol^{-1}]$	0	$0.12\text{x}10^{-2}$	0	0	0	-
$a_{st}^{a,*}$	-	1	1	1	1	0.5	-
$a_{st}^{*,d}$	-	1	1	1	1	0.5	-
$a_{st}^{d,*}$	-	1	1	1	1	0.5	-
$k_t^{M_1}$	$[m^3.s^{-1}.mol^{-1}]$	$1.58\text{x}10^{-4}$	$9.01\text{x}10^{-4}$	$1.35\text{x}10^{-4}$	$9.03\text{x}10^{-4}$	$4.07\text{x}10^{-3}$	-
$k_t^{M_2}$	$[m^3.s^{-1}.mol^{-1}]$	$1.58\text{x}10^{-4}$	$9.01\text{x}10^{-4}$	$1.35\text{x}10^{-4}$	$9.03\text{x}10^{-4}$	$4.07\text{x}10^{-3}$	-
Reactant Properties		*sp*	*TEA*	M_1	M_2	H_2	*dil*
P_{sat}	$[Pa]$	-	-	$5.06\text{x}10^{6}$	$1.13\ \text{x}10^{6}$	$1.29\ \text{x}10^{6}$	$1.34\ \text{x}10^{4}$
ρ	$[kg.m^{-3}]$	-	-	570	630	100	718
MM	$[g.mol^{-1}]$	-	228	28	56	2	128
Catalyst Particle Properties							
$\bar{l}_0$	$[m]$	$50\text{x}10^{-6}$					
σ	$[m]$	$7\text{x}10^{-4}$					
η_{Ti}	$[mol_{Ti}.kg_{cat}^{-1}]$	$2.716\text{x}10^{-4}$					

The saturation pressures required for vapour-liquid equilibrium calculations are taken from the ASPEN Properties databases (AspenTech, 2011).

7.2 Polymer grade transitions

The model parameters listed in the previous section are used to simulate a polymerisation plant that produces two theoretical grades, labelled Grade A and Grade B. These grades are characterised by the average properties of a chain length distribution curve, i.e. M_n and PDI. Table 7-2 lists the set of feed rates which produce each grade under the reactor conditions that are specified in Table 7-1. The exact properties of the polymer that result from evaluating the process model under steady-state conditions at the given set of feed rates are also displayed in the table.

Table 7-2: Reactant feed rates required for different polymer grades

	$\left[kmol.hr^{-1}\right]$					$\left[g.s^{-1}\right]$		
	$\dot{F}_{TEA,0}$	$\dot{F}_{M_1,0}$	$\dot{F}_{M_2,0}$	$\dot{F}_{H_2,0}$	$\dot{F}_{dil,0}$	$\dot{m}_{cat,0}$	M_n^*	PDI^*
Grade A	0.129	51.57	0.16	1.89	36.1	0.52	$5.20\text{x}10^4$	5.34
Grade B	0.127	48.36	0.36	1.71	33.85	0.36	$5.67\text{x}10^4$	6.58

A polymer is of a certain grade if its properties fall within a specific range. The upper and lower limits of this range for each grade are shown in Table 7-3. Clearly, the properties of the polymers produced under the conditions shown in Table 7-2 fall within the respective ranges.

Table 7-3: Specifications for polymer grades

	$M_n^{*,l}$	$M_n^{*,u}$	$PDI^{*,l}$	$PDI^{*,u}$
Grade A	$5.10\text{x}10^4$	$5.30\text{x}10^4$	5.20	5.50
Grade B	$5.60\text{x}10^4$	$5.80\text{x}10^4$	6.45	6.75

The sensitivity of the system towards variations in the flowrates of the reactants was tested to observe the non-linearity of the process. Each component was varied individually and the effect on the polymer properties was recorded. The results of this analysis are presented in Appendix C.

7.2.1 Standard grade transition

Traditional grade transitions are performed by directly switching the feed rates and reactor conditions from one set to another. The component concentrations in the bulk phase continually change during this transition time, affecting the various polymeric reactions. As a result, the polymer accumulated around the particles has properties that are influenced by a continuously varying set of conditions. The influence of the system dynamics on the polymer properties over time can be tracked by trajectory curves.

The M_n and PDI trajectories as well as the polymer production rates for traditional grade transitions from Grade A to Grade B and from Grade B to Grade A are illustrated in Figure 7.1 and Figure 7.2, respectively. The limits of these manipulated variables, in accordance with the values in Table 7-3, are displayed in the figures to indicate the time taken for the properties to be within the correct ranges. The source code for all the process models as well as the optimisation schemes can be found in Appendix E.

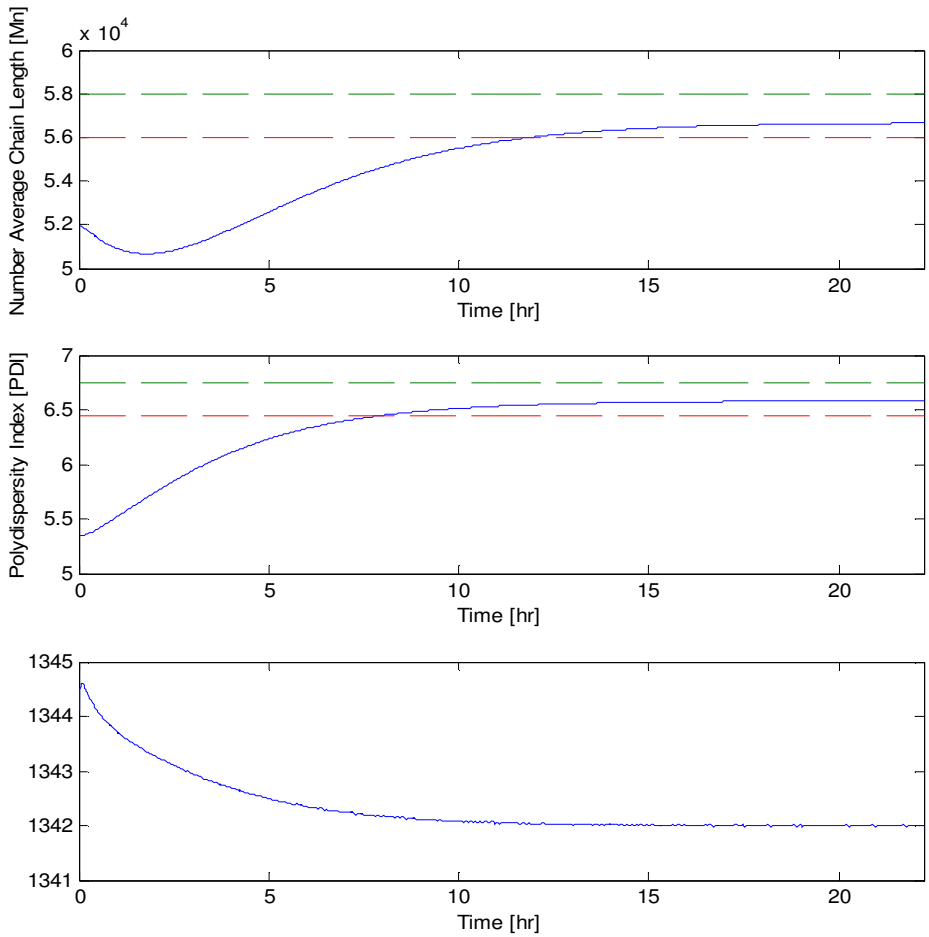

Figure 7.1 – Standard grade transition from Grade A to Grade B

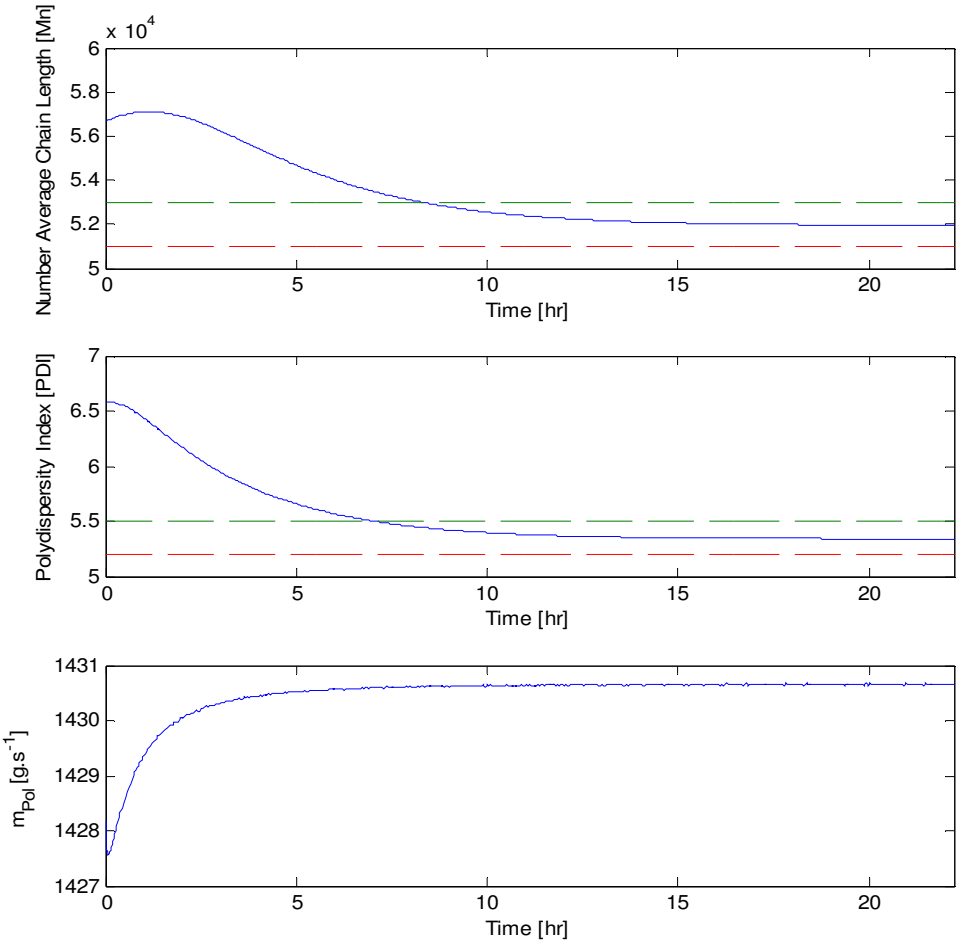

Figure 7.2 - Standard grade transition from Grade B to Grade A

From the figures it can be seen that these grade transitions last approximately 19 hours, of which on-spec polymer is produced after 8-12 hours. The amounts of off-spec material generated during these transient periods are shown in Table 7-4. These values serve as the benchmark for improvement by the optimal control strategies.

Table 7-4: Grade transition benchmark

	$t_{off}\,[hrs]$	$M_{off}\,[tons]$
Grade A to Grade B	12	57
Grade B to Grade A	8	43

7.2.2 Off-spec sources

The schemes proposed in Chapter 6 all involve blending off-spec polymer from stored silos. The following table lists the properties of four sources of off-spec polymer that will be used for blending with a reactor product to test these various control schemes in the subsequent sections. A comparison of these off-spec properties to that of the properties of the grades, as per Table 7-3, shows that off-spec sources have properties where either one or both M_n and PDI values are higher than or lower than either of the grades of interest.

Table 7-5: Off-spec polymer properties

	M_n	PDI
Source 1	5.0×10^4	5
Source 2	6.0×10^4	7
Source 3	5.0×10^4	7
Source 4	6.0×10^4	5

7.3 Steady-state off-spec blending

The results in the current section are due to application of the scheme proposed in Section 6.1, where the steady-state reactor feed rates necessary to produce a polymer blend within spec after the continuous constant flow addition of off-spec material to the system are predicted. The performance of this controller is tested subject to the addition of different quantities and different types of off-spec material.

The properties of the off-spec material; the properties of the virgin polymer as a result of the controller response to the addition of off-spec material; and the properties of the resulting blend of the virgin polymer and off-spec material are presented in Figure 7.3 at four steady-state flowrates of off-spec source 1 during the production of Grade A polyethylene. These incoming off-spec flowrates varied from roughly 1/10th to double that of the reactor polymer production rate.

The properties of the blend in all cases meet that of the target value (the property value of the virgin polymer that is produced under normal operating conditions, represented by the yellow dashed line) and are between the upper and lower limits of the specification band (represented by the green dashed lines).

The sets of reactant feed rates determined by the controller in each case are listed in Table 7-6 and can be compared to the set of feed rates under normal operating conditions.

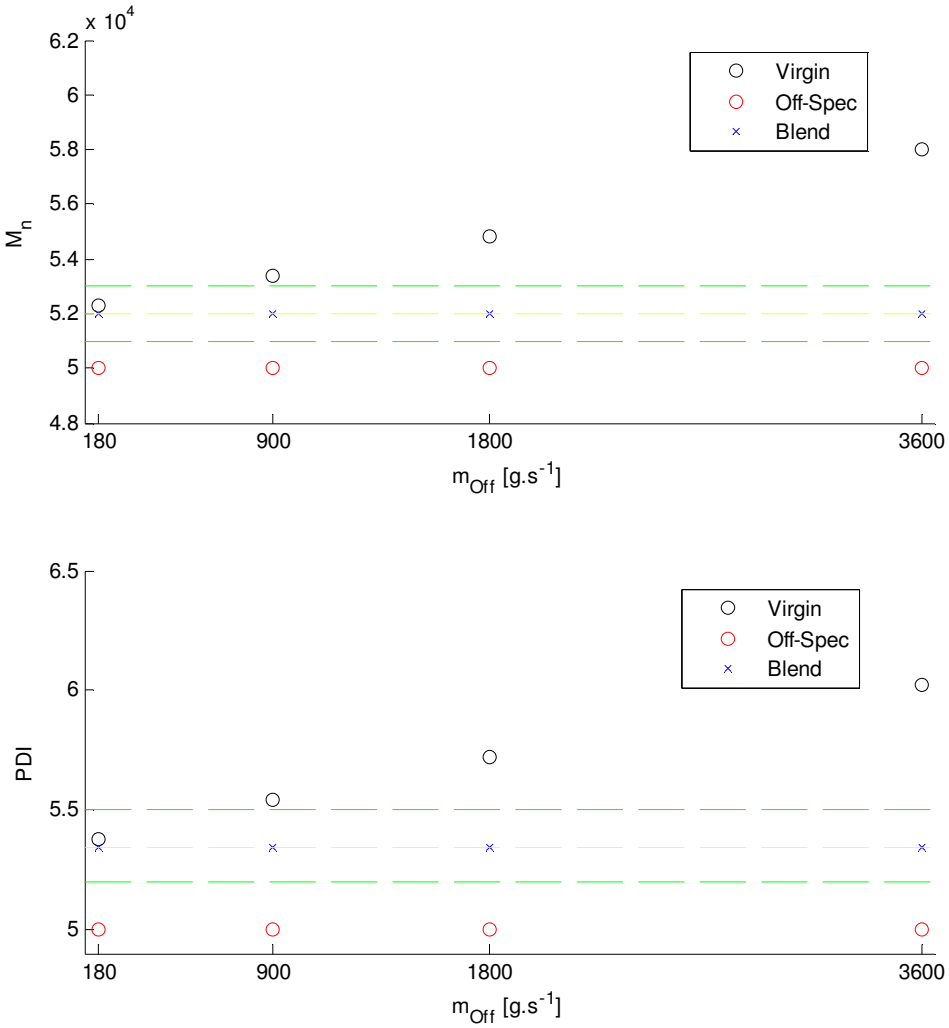

Figure 7.3 - Comparison of the outcome of control action on the virgin product and blend properties for various flowrates of off-spec source 1

Table 7-6: Feed rate adjustments to counter introduction of off-spec source 1

$\left[g.s^{-1}\right]$			$\left[kmol.hr^{-1}\right]$			$\left[g.s^{-1}\right]$
$\dot{m}_{Off}$	$\dot{F}_{TEA,0}$	$\dot{F}_{M_1,0}$	$\dot{F}_{M_2,0}$	$\dot{F}_{H_2,0}$	$\dot{F}_{dil,0}$	$\dot{m}_{cat,0}$
0	0.129	51.57	0.16	1.89	36.1	0.52
180	0.113	50.21	0.16	1.29	35.8	0.53
900	0.115	50.21	0.18	1.29	35.8	0.50
1800	0.117	50.21	0.19	1.29	35.8	0.47
3600	0.120	50.21	0.22	1.29	35.8	0.42

The following table and figure present in a similar manner as the previous data set the results of the response of the theoretical controller to off-spec source 2 being fed to the

system producing Grade B. This source of off-spec polymer has high M_n and PDI values relative to Grade B spec.

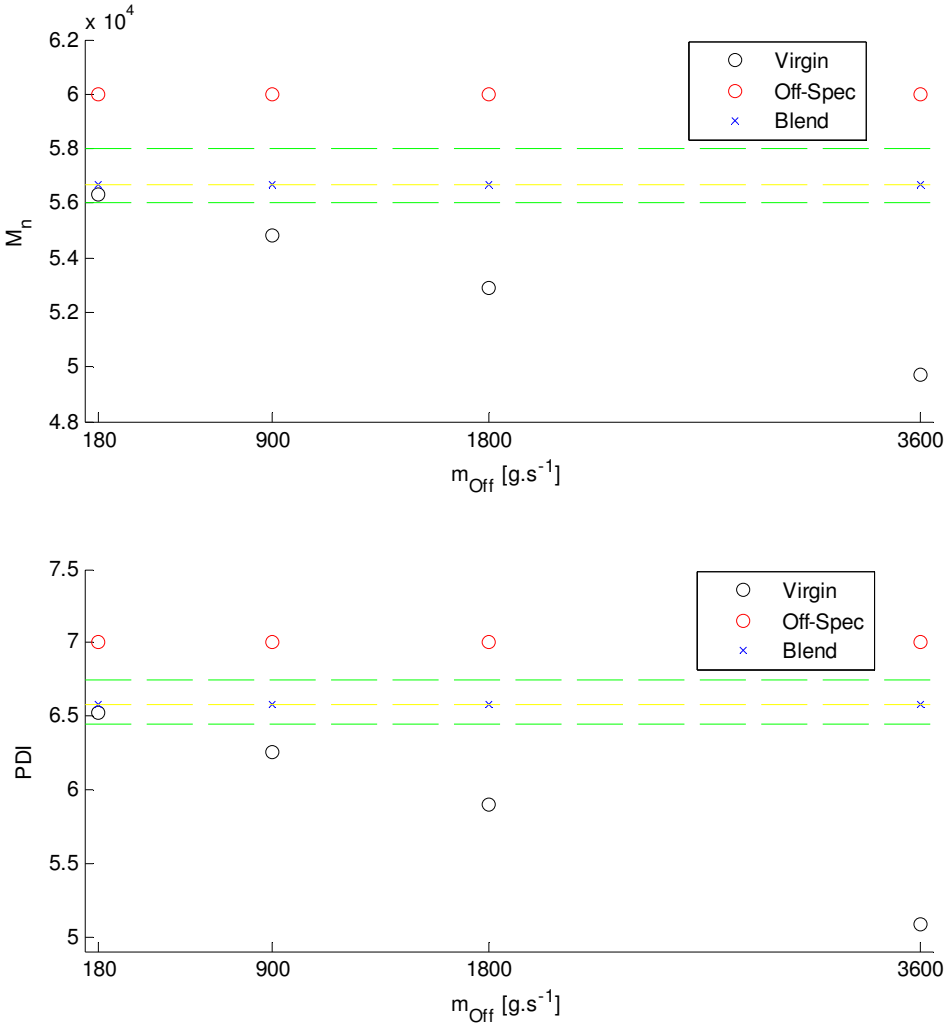

Figure 7.4 - Comparison of the outcome of control action on the virgin product and blend properties for various flowrates of off-spec source 2

Table 7-7: Feed rate adjustments to counter introduction of off-spec source 2

$\left[g.s^{-1}\right]$	$\left[kmol.hr^{-1}\right]$					$\left[g.s^{-1}\right]$
$\dot{m}_{Off}$	$\dot{F}_{TEA,0}$	$\dot{F}_{M_1,0}$	$\dot{F}_{M_2,0}$	$\dot{F}_{H_2,0}$	$\dot{F}_{dil,0}$	$\dot{m}_{cat,0}$
0	0.127	48.36	0.36	1.71	33.9	0.36
180	0.123	50.13	0.36	1.26	35.5	0.40
900	0.121	50.10	0.30	1.24	35.4	0.43
1800	0.120	50.20	0.24	1.28	35.7	0.47
3600	0.120	50.20	0.24	1.28	35.7	0.47

The results of the two cases presented above confirm that the required feed rate adjustments for blending can be calculated to accommodate a wide range of off-spec flowrates. Therefore, an off-spec polymer is allowed to be used for blending regardless of whether it has a high or a low average chain length and/or a narrow or wide CLD curve.

The higher the flowrate of off-spec material is, the greater the deviations of the properties of the intermediate polymer are from that of the original grade properties. The virgin polymer that is required to be produced when a blend ratio of 2:1 by mass of off-spec to virgin polymer is chosen has properties that resemble more closely that of the other grade than its own target value. In most cases, the properties of the intermediate polymer are even further off-spec than that of the off-spec source itself. Fortunately, there seems to exist a relationship between the properties of the virgin polymer and the off-spec flowrate, thus the maximum flowrate can be calculated where the resulting intermediate polymer properties required are within the limits of the desired specification band.

It would be ideal to invert the blending model thereby directly calculating the maximum off-spec flowrate. However, 3 known variables are required to determine the 3 polymer moments. Unfortunately, there are only 2 known variables available, namely M_n and PDI. As such, its maximum flowrate was determined empirically by fitting a second degree polynomial to the data over the given interval. Data regression yielded excellent fits resulting in R^2 values close to, if not, 1. The plots for these fits can be found in Appendix B along with the equations of each polynomial. From these equations the maximum allowable flowrate was calculated for both M_n and PDI in each case. The minimum of the two values is the flowrate that allows both properties to fall within spec. This flowrate is listed in Table 7-8 as the optimal flowrate in each case along with the set of feed rates required to produce the intermediate polymer that will accommodate the off-spec polymer at this flowrate, as well as the properties of this intermediate polymer.

Table 7-8 - Optimal flowrates such that virgin properties remain within spec

	$\left[g.s^{-1}\right]$	$\left[kmol.hr^{-1}\right]$					$\left[g.s^{-1}\right]$		
Case	$\dot{m}_{Off}$	$\dot{F}_{TEA,0}$	$\dot{F}_{M_1,0}$	$\dot{F}_{M_2,0}$	$\dot{F}_{M_2,0}$	$\dot{F}_{dil,0}$	$\dot{m}_{cat,0}$	M_n	PDI
1	648	0.113	51.48	0.17	1.85	35.8	0.53	$5.5\text{x}10^4$	5.48
2	323	0.121	51.37	0.35	1.81	35.4	0.42	$5.6\text{x}10^4$	6.47

Recall, the Nelder-Mead simplex method was used to solve these steady-state adjustment problems. It is well known that this method is prone to convergence at local extremes. In this case, this outcome was preferred since it results in the adjusted feed flowrates being close to their initial values. It is not desirable to adjust the flowrates to such an extent that the entire system becomes unstable. Furthermore, any feed rate adjustment will result in a transient period. While the initial properties are on-spec and the properties of the polymer produced at the end of the transition period are suitable for blending with off-spec polymer at a particular flowrate to produce a blend that is on-spec, the intermediate polymer properties may not be on-spec or may not be suitable for blending. Remember, the property is only off-spec of is falls outside the upper or lower limits of the specification band. It is for this reason that the feed rate adjustments previously calculated were simulated in the unsteady-state process model to test the system response and see if the virgin polymer or the blended polymer trajectory falls outside the specification band at any period during the transition. The adjustment results for blending with off-spec source 1at various flowrates are presented in Figure 7.5 to Figure 7.8.

In Figure 7.5, the transition period is illustrated as a result of feed rate adjustments to accommodate blending with off-spec polymer having a flowrate that is 10% that of the virgin polymer production rate. It is clear from the figure that the property trajectories of both the virgin polymer and the blend remain within the specification bands throughout the transition period. In contrast, when blending is performed at higher off-spec flowrates, the required feed rate adjustments result in transition periods where the trajectory of the virgin polymer properties temporarily stray outside the specification bands. In addition, the property trajectories of the blended polymers are al initially outside the specification band but progress towards their respective target values.

Referring to Figure 6.1, the plant could be set up in such a way that the reactor product stream and the blend stream can be diverted either to the off-spec silo or the on-spec silo at any time. If this were the case and if it were not important whether the properties of the virgin polymer for blending was within spec or not, then the flow could be diverted to the correct silos accordingly, such that the flow of polymer to the on-spec silo is not interrupted during the transition.

In Figure 7.8 it is shown that if a flowrate of off-spec polymer, used for blending, is too high then it may bring about an interval during the transition period where neither intermediate polymer nor blend are within the specification band. Thus, the optimal flowrate is the maximum allowable flowrate of an off-spec material with which an intermediate polymer, the properties of which are determined by the optimisation algorithm, is blended such that at least the intermediate polymer properties or the blend properties are within the specification band during the transition period.

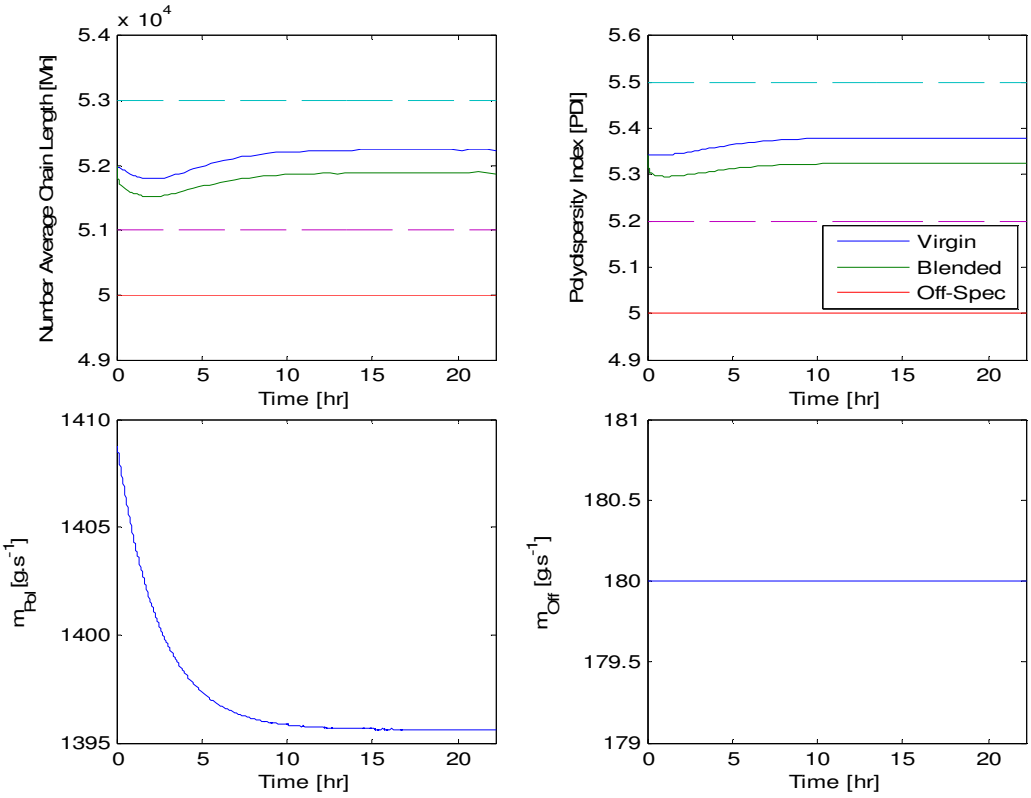

Figure 7.5 - Transient period created by feed adjustment for blending case 1

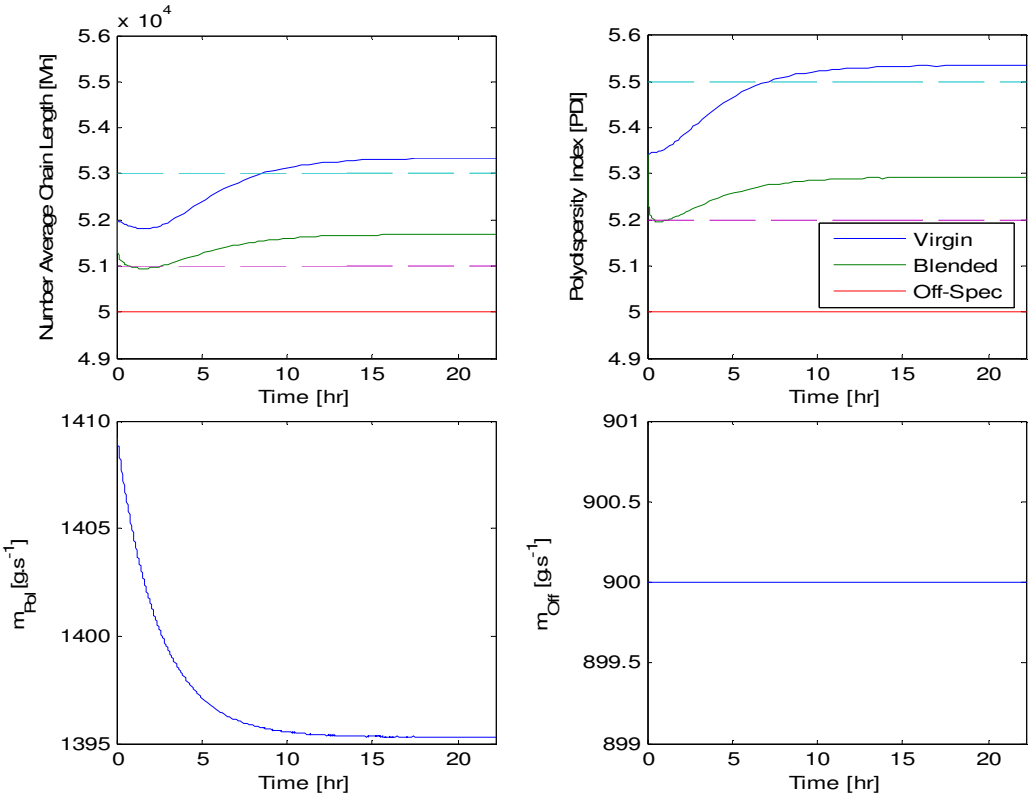

Figure 7.6 - Transient period created by feed adjustment for blending case 2

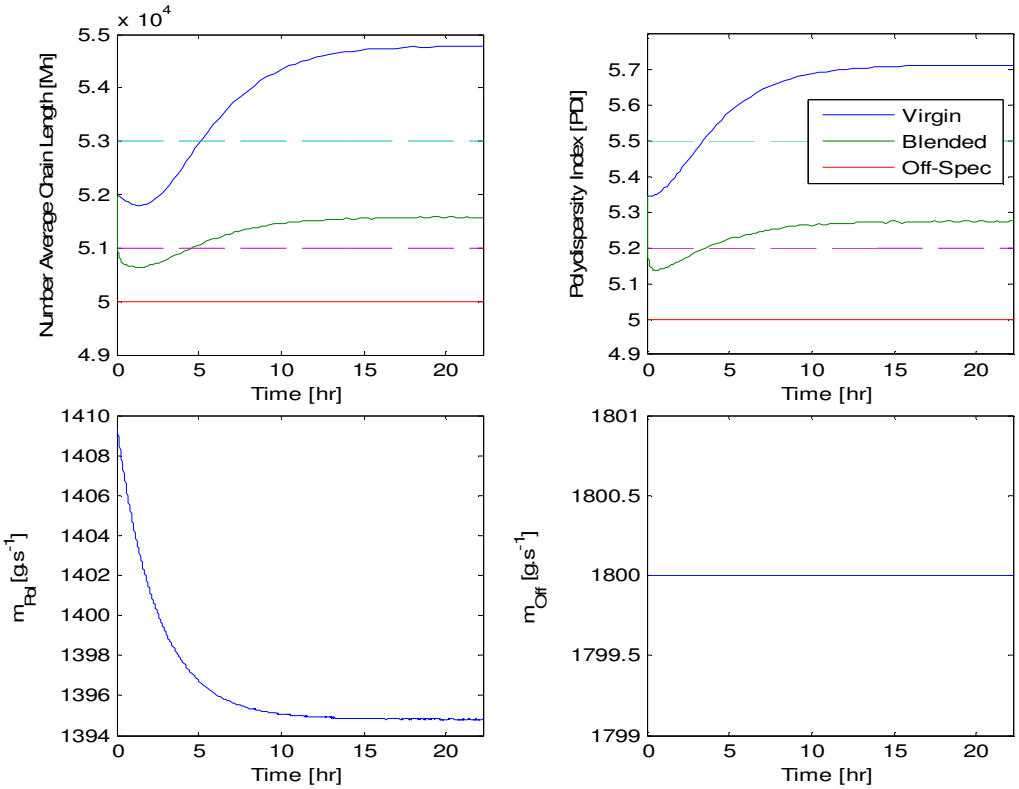

Figure 7.7 - Transient period created by feed adjustment for blending case 3

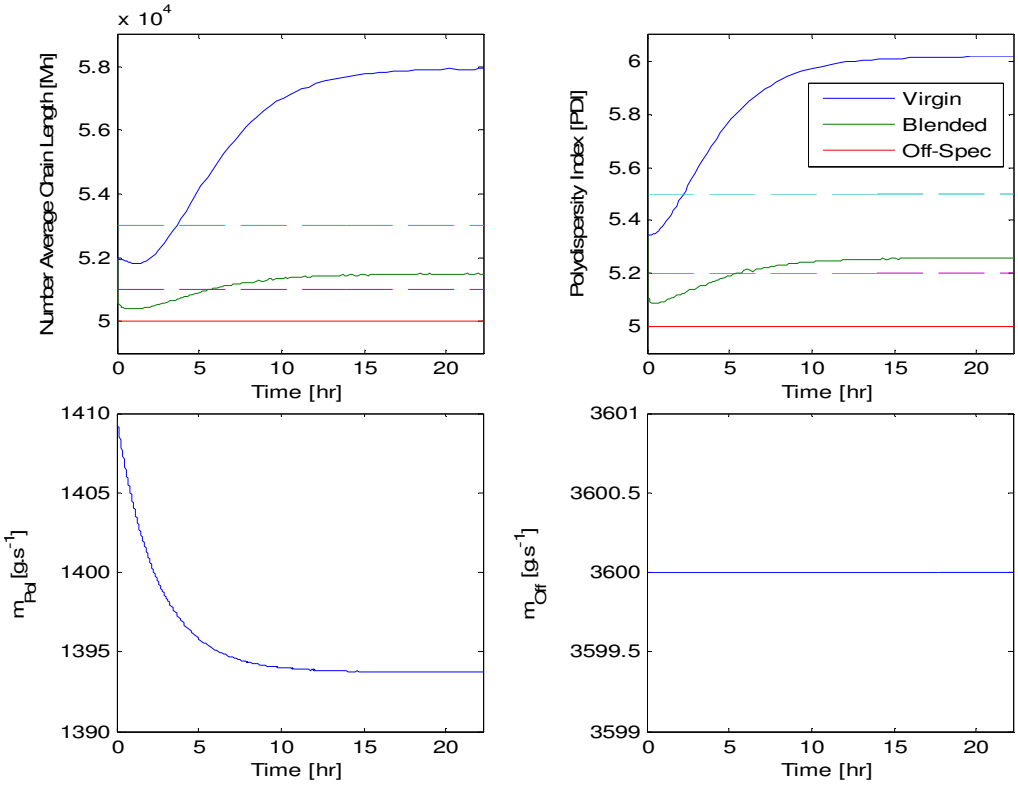

Figure 7.8 - Transient period created by feed adjustment for blending case 4

The control strategy has been devised such that it only predicts and takes into account the properties of the blend and the intermediate polymer at steady-state. It is limited by its inability to determine the influence of the feed rate adjustments on the process dynamics, thus it is unable to automatically determine the optimal flowrates that were previously calculated. It is recommended that the optimisation algorithm be modified so as to determine the optimal off-spec flowrate for blending such that the steady-state properties of the intermediate polymer are within the specification band. Successful implementation will prevent the generation of any off-spec polymer during the transient period that is induced by the feed rates adjustment.

7.4 Off-spec blending during grade transition

The current section displays the results of the control strategy described in Section 6.2 that has been applied during a standard grade transition from Grade B to Grade A for each of the off-spec sources listed in Section 7.2.2. In this strategy, the optimal unsteady-state flowrate trajectory of off-spec polymer is calculated by applying control vector parameterisation to the dynamic optimisation problem to solve it as an NLP.

A simple step function approach was used to approximate the off-spec flow profile as a proof of concept. The control vector (off-spec polymer flowrate) was discretised into 5 equally spaced intervals over the time range in which the reactor product is off-spec, roughly 12 hours. This means that control action is set to occur every 2.5 hours.

The following table lists the parameters specific to setting up the algorithm. All optimisation was performed on a 1.7 GHz quad-core processor with hyper-threading. Thus the DE algorithm could take advantage of multiple cores to evaluate objective functions in parallel. The system allowed for 8 instances (1 Master and 7 slaves) of the MATLAB code to be run simultaneously. This resulted in a runtime of less than 7 minutes to find an optimal trajectory for over 22 hours of production time.

Table 7-9: DEA parameters of off-spec blending during grade transition

Parameter	Value
Population size (NP)	60
Crossover Probability (Cr)	0.7
Mutant Scaling Factor (F)	0.5
Maximum Iterations	50
Maximum Time	$1x10^6$ seconds
Algorithm Strategy	1 (Default)

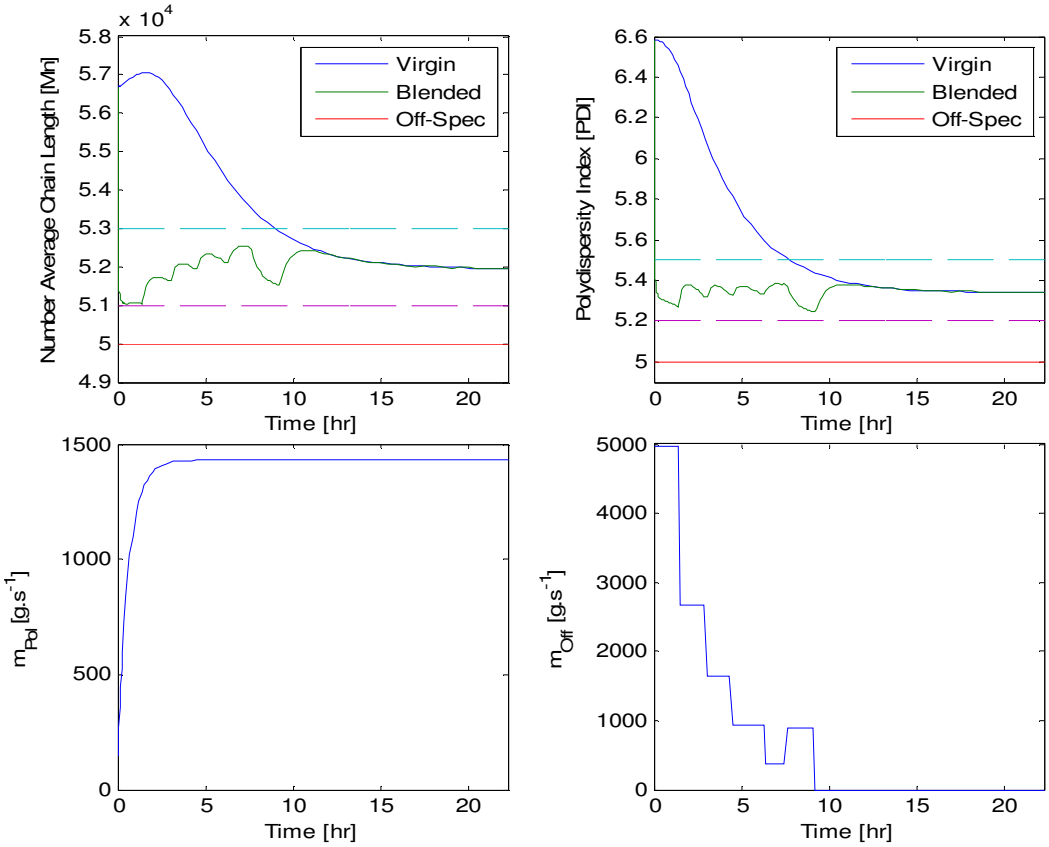

Figure 7.9 - Blend profile for grade change from B to A with off-spec source 1

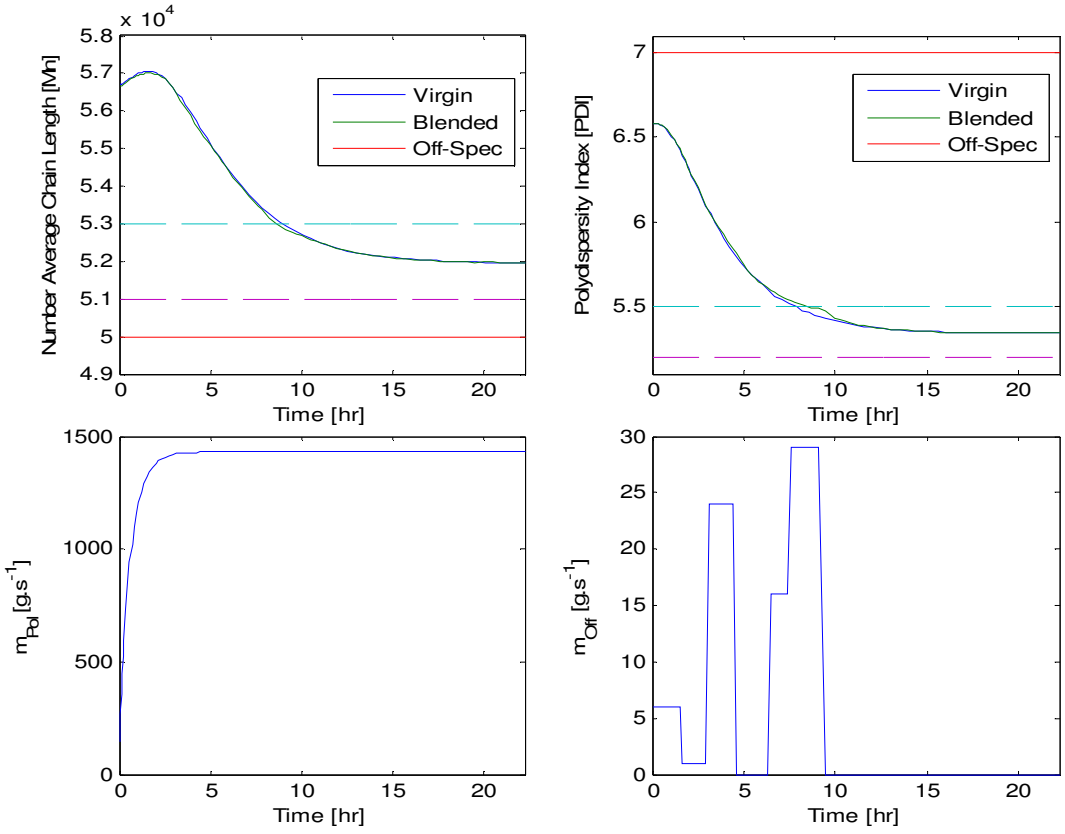

Figure 7.10 - Blend profile for grade change from B to A with off-spec source 3

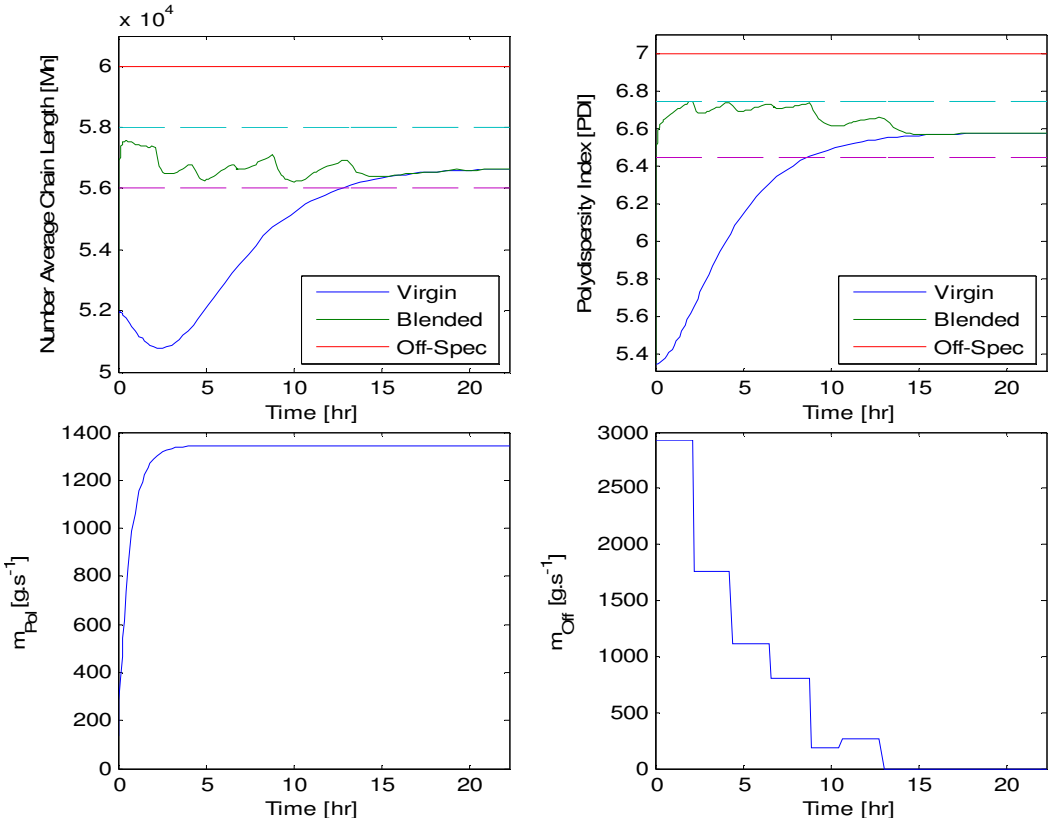

Figure 7.11 - Blend profile for grade change from A to B with off-spec source 2

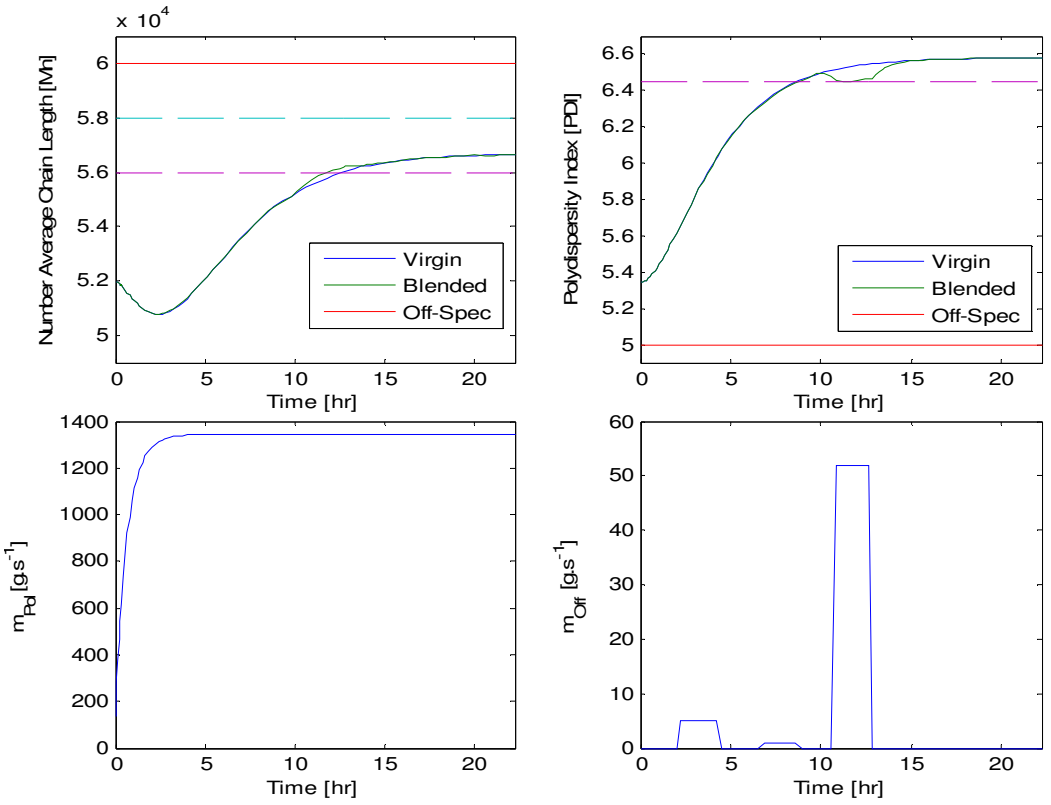

Figure 7.12 - Blend profile for grade change from A to B with off-spec source 4

For the grade change from B to A, only two sources of off-spec material are viable for blending, namely source 1 and source 3. The flow profiles for these sources are presented in Figure 7.9 and Figure 7.10, respectively. Similarly, for the reverse grade change, only sources 2 and 4 are viable, the flow profiles of which are presented in Figure 7.11 and Figure 7.12, respectively. Each flow profile is accompanied with a comparison of the property trajectories for M_n and PDI between a standard grade transition and that of a transition with blending. In addition, the polymer production rate is given as an indication of the blend ratio of virgin polymer to off-spec material.

The controller is able to predict a flow profile for source 1 off-spec polymer such that the resultant blend has properties that are always within their respective specification bands. The same is true for the profile determined for off-spec source 2, during the reverse grade change. These are highly desirable outcomes since no off-spec material is generated during the transient period, yet the maximum amount of stored off-spec polymer is being consumed to create a more saleable product.

The reason for the success of these two cases is due to the fact that the target values lay between the off-spec properties and the virgin polymer properties. With source 3 and 4, only one of the target values was between the off-spec and virgin polymer properties. Thus, blending was only feasible for a small duration of time, rendering it not worthwhile to blend these sources during the respective grade transitions they were tested under, as expected. For this reason, the choice of off-spec material should always be made prior to operation of the algorithm.

In Figure 7.10 it can be seen that the shortening of the average chain length due to blending is accompanied by the undesirable widening of the distribution curve. Since the trajectory of the PDI property is within spec before the trajectory of the M_n property, there is some leeway in which the PDI value can be increased within the confines of the specification band, whilst the M_n of the blend becomes within spec. Thus, a small amount of off-spec polymer can be consumed to create a blend within spec. A similar situation can be seen in Figure 7.11 for the reverse grade change except off-spec polymer source 4 has a high average chain length and a relatively narrow distribution curve.

It should be noted that there are large fluctuations in the production rate of the on-spec blend throughout the procedure. It is generally desirable to have a stable polymer production rate. However, blending is taking place during a transient period where previously no on-spec polymer would have been produced; therefore these fluctuations can be overlooked. Furthermore, these fluctuations are a result of the relatively infrequent step changes

characterising the control action. The use of smoother functions to approximate the true optimal flowrates should result in more stable production rates.

7.5 Off-spec blending during optimal grade transition

The previous set of results show that, for blending, some types of off-spec polymer are useful; others can only be blended in small amounts and others are not at all viable. The results shown in the current section are from application of the scheme outlined in Section 6.3. This scheme is very similar to the scheme presented in Section 6.2, except that the algorithm is devised to find optimal flowrate profiles for off-spec polymer as well as some reactants. It is believed that the grade transition trajectories can be manipulated in a manner that would make the blending of the latter off-spec types viable.

In the current scheme, ethylene and hydrogen profiles are controlled, so as to manipulate the trajectories of the virgin polymer properties in such a way that it allows for off-spec polymer source 4 to be blended during a grade transition from B to A and off-spec polymer source 3 to be blended during a grade transition from A to B. Both reactant control vectors are discretised into 6 equally spaced intervals, where the value of the final interval for each reactant is fixed as the steady-state flowrate for that reactant. Thus, 5 parameters are optimised for each control vector.

The parameters chosen to optimise the grade transitions with the DE algorithm are listed in Table 7-10. Roughly 1 hour was needed for the algorithm to find a solution for each grade change. Figure 7.13 shows the ethylene and hydrogen flow profiles that optimise the properties trajectories for a transition from production of Grade B to Grade A and Figure 7.14 shows optimal flow profiles for the reverse grade change. For comparative purposes, the trajectories of the virgin polymer properties during a traditional grade transition are included in the figures.

Table 7-10: DEA parameters of off-spec blending during grade transition

Parameter	Value
Population size (NP)	100
Crossover Probability (Cr)	0.8
Mutant Scaling Factor (F)	0.7
Maximum Iterations	10
Maximum Time	$1x10^6$ seconds
Algorithm Strategy	1 (Default)

It is observed that the feed rate manipulations result in somewhat oscillatory behaviour of the M_n trajectory such that the property becomes on-spec much quicker than it would during a standard grade transition. It is also observed that the PDI property trajectory has a low sensitivity to feed rate manipulations. These two observances apply to transitions from Grade A to Grade B and vice versa.

Of course, the polymer is only considered on-spec if both properties fall within their respective specification bands. With the standard grade changes, the polymer was limited by the slower M_n trajectories. By manipulating the feed rates, the M_n curves had improved to the point that PDI had become the limiting property. Unfortunately, due to the low sensitivity of PDI to the feed rate adjustments; coupled with the relative closeness of the times at which the curves of M_n and PDI reach their steady-state values during a standard grade change; the overall transition time improvement was small.

For the current polymerisation system, optimal control of grade transitions does not greatly reduce the grade transition times or the amount of off-spec material generated. In addition, certain types of off-spec polymer have narrow windows in which blending during some grade transitions is allowed. However, the combination of both off-spec blending and grade transition optimisation could provide superior improvements over each separate practice. The parameters for running the DE algorithm to find the optimal off-spec polymer flowrate during these controlled grade changes are shown in Table 7-11.

Where previously, source 3 and source 4 were not viable for blending during the transitions from Grade B to Grade A and Grade A to Grade B, respectively, they are suitable for blending during the corresponding optimal grade transitions as shown in Figures 7.16 and 7.15, respectively. In contrast, source 3 and source 4 were previously viable for blending during the reverse grade changes, yet are not able to produce feasible blends during the respective optimised grade transitions.

The compatibility of an off-spec polymer for blending during a particular grade transition depends on the properties of both off-spec polymer properties in relation to the approach of the virgin polymer properties' trajectories towards their respective target values. For instance, if both virgin polymer properties are lower than their target values and both properties of the off-spec polymer used for blending are higher than the target values, then blending will be feasible.

The situations where only small amounts of off-spec material could be blended during a grade transition came about when one of the off-spec polymer properties would result in the blend's property trajectory moving away from its target value.

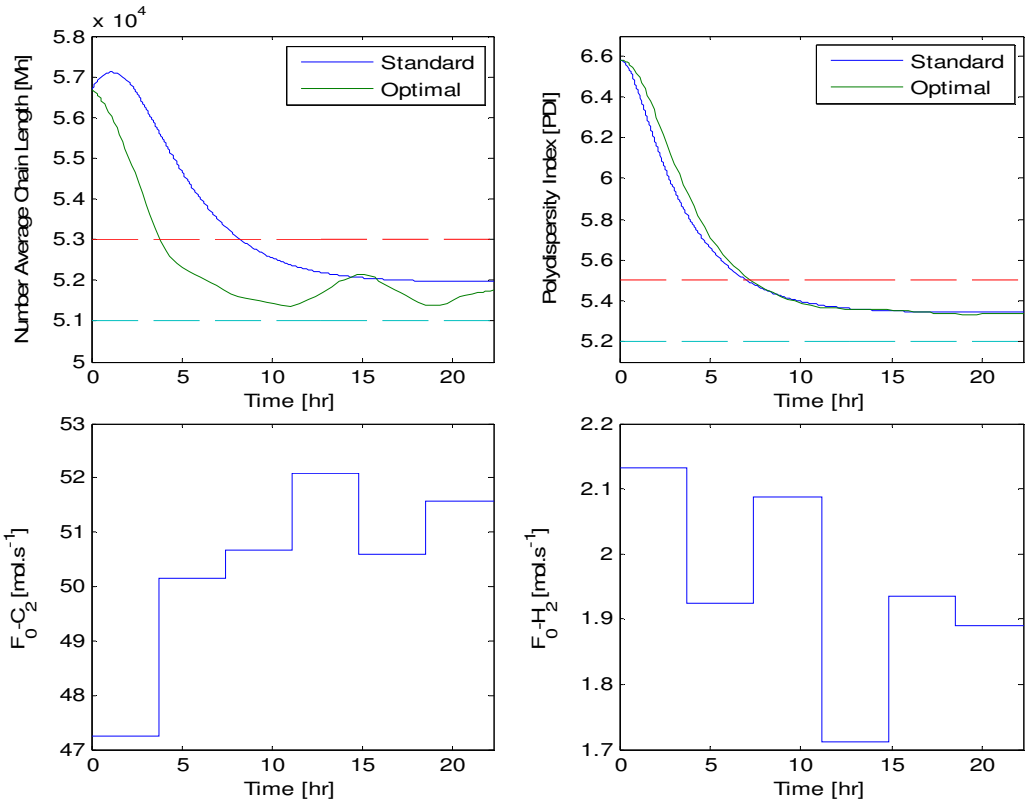

Figure 7.13 – Optimally determined trajectories for grade change from B to A

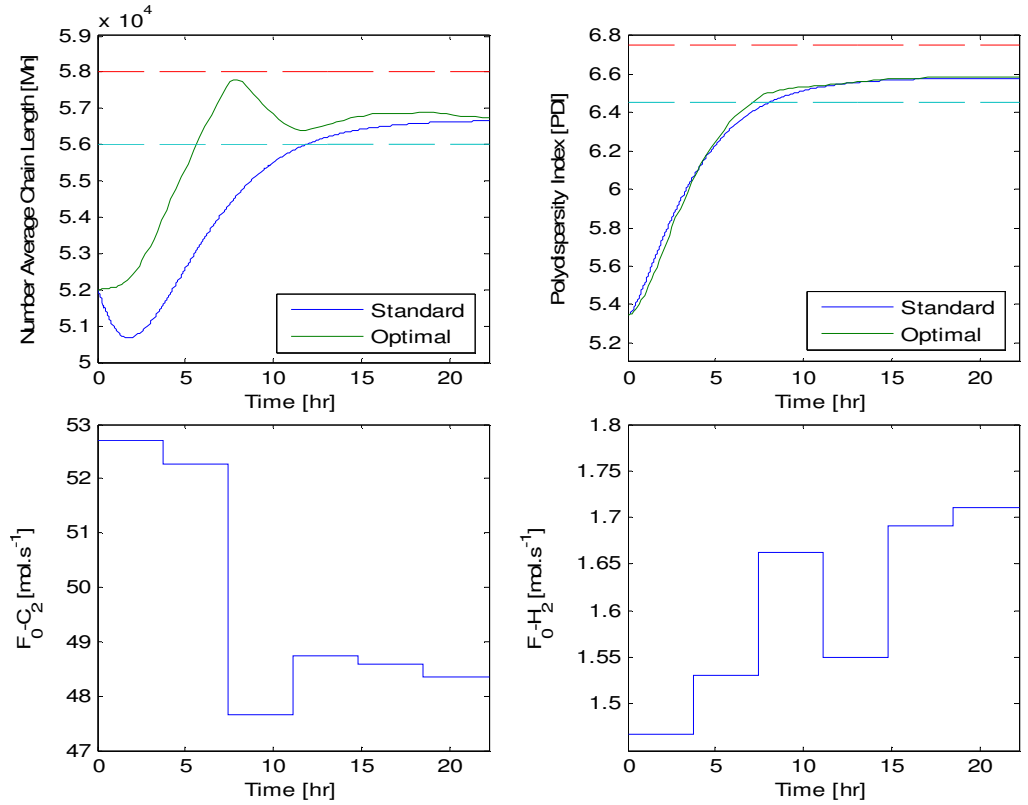

Figure 7.14 – Optimally determined trajectories for grade change from A to B

Table 7-11: DEA parameters of off-spec blending during grade transition

Parameter	Value
Population size (NP)	60
Crossover Probability (Cr)	0.7
Mutant Scaling Factor (F)	0.5
Maximum Iterations	1000
Maximum Time	1x10^6 seconds
Algorithm Strategy	1 (Default)

The blend profiles are presented below in a similar manner to the results displayed in the previous section. It is important to note that off-spec source 3 and off-spec source 4 are very different to one another. Relative to both grades, off-spec source 3 has low M_n and high PDI values, whereas off-spec source 4 has high M_n and low PDI values. Thus, both off-spec polymers will blend with either grade such that the blend properties will deviate from the fresh polymer properties in opposite directions.

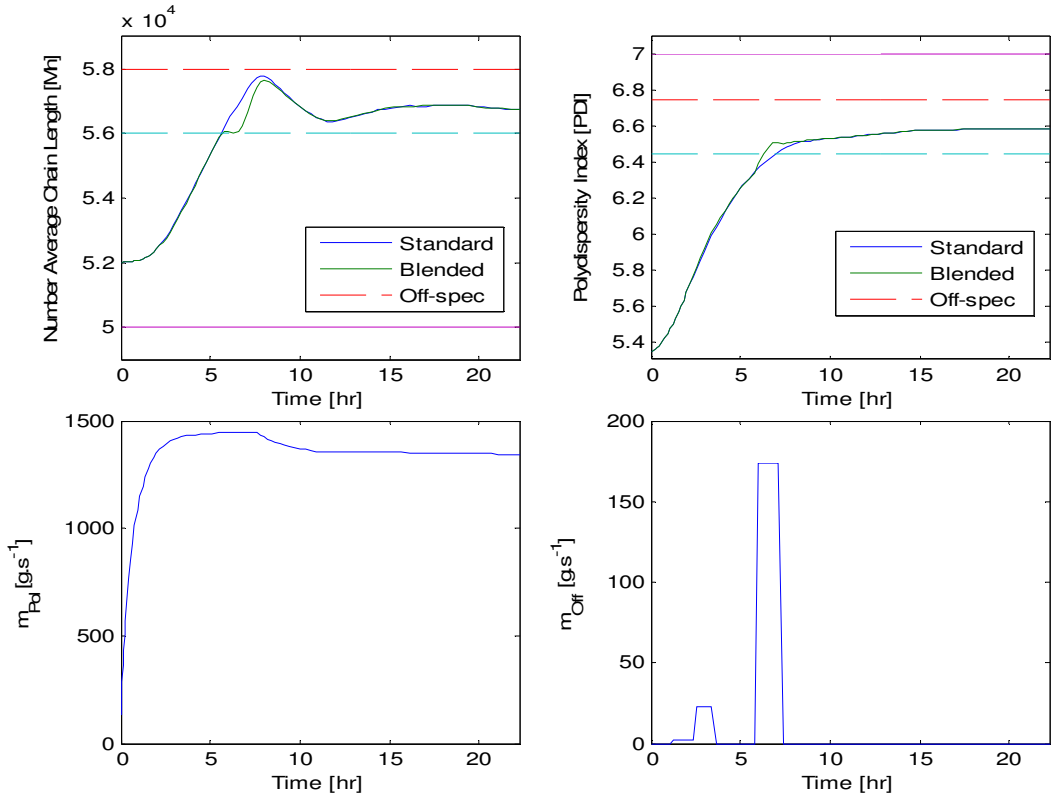

Figure 7.15 - Blend profile for optimal grade change from A to B with off-spec source 3

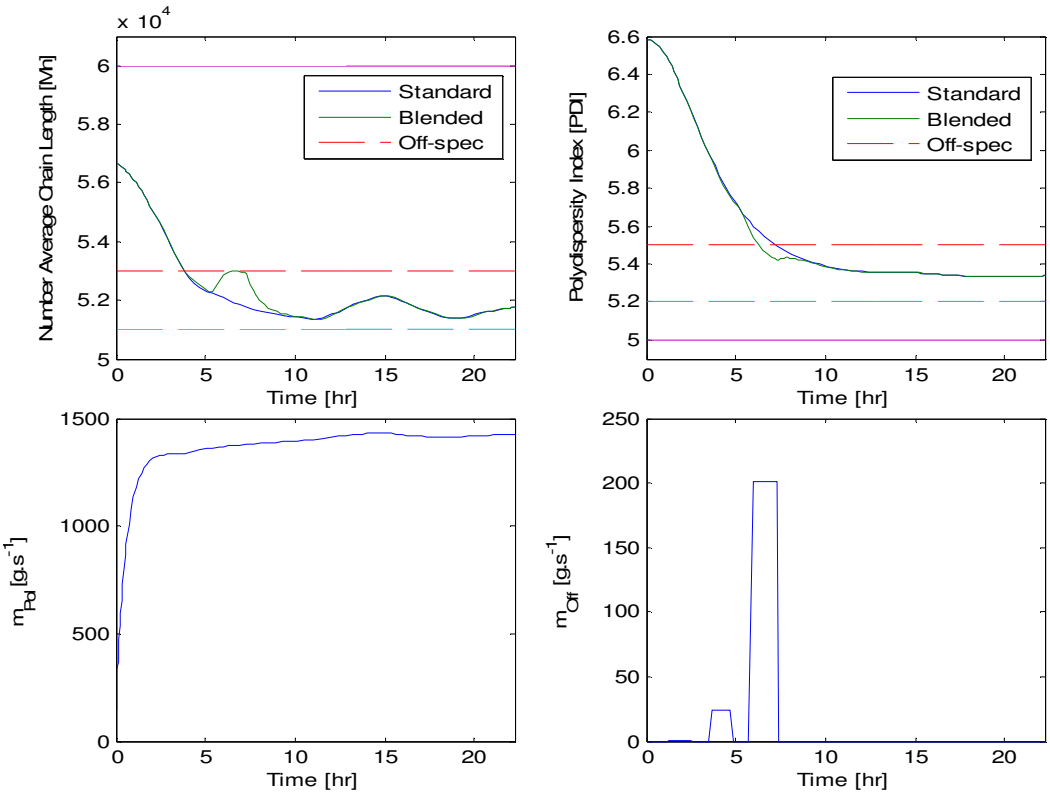

Figure 7.16 - Blend profile for optimal grade change from B to A with off-spec source 4

During a standard grade transition from B to A, off-spec source 3 was a feasible blend constituent because it had an M_n value that was appropriate for blending with the virgin polymer, which was limiting the grade transition. With the optimised grade transition, PDI is limiting and a blend would need to lower this value to result in the transient polymer properties becoming within spec sooner. Off-spec source 3 is not viable for this type of blend as it has higher PDI value. However, for these purposes, off-spec source 4 is more suitable. Its lower PDI value does allow for a blend where the property is within spec, whilst increasing the M_n in such a way that it does not veer off-spec.

It is for similar reasons that off-spec source 4 is viable for blending during a traditional grade transition from A to B and not for the profile of this transition determined through optimal control. Thus, it can be concluded that the feasibility of a blend is dependent on the properties of off-spec polymer in relation to the properties of the both current and desired grades as well as the limiting property of the virgin polymer.

Summary tables are presented below to clearly compare the extent to which each off-spec source is able to reduce the transition time and amount of off-spec material generated; and increase the amount of off-spec source material consumed for both grade transitions.

Table 7-12: Off-spec blending summary for grade transition of B to A

Standard GT		$m_{Off}\,(tons)$			Desired		Off-spec	
Case	$t_{off}\,(hr)$	generated	consumed	net	M_{n} (x1e4)	PDI	M_{n} (x1e4)	PDI
BA	8.87	43.6	-	43.6	5.2	5.34	-	-
BA1	0.05	0.30	62.0	-61.7	5.2	5.34	5	5
BA2	-	-	-	-	5.2	5.34	6	7
BA3	8.24	41.0	0.09	40.9	5.2	5.34	5	7
BA4	-	-	-	-	5.2	5.34	6	5
Optimal GT								
BA	6.88	32.0	-	32.0	5.2	5.34	-	-
BA1	0.08	0.51	40.1	-39.6	5.2	5.34	5	5
BA2	-	-	-	-	5.2	5.34	6	7
BA3	-	-	-	-	5.2	5.34	5	7
BA4	5.97	27.5	0.89	26.6	5.2	5.34	6	5

Table 7-13: Off-spec blending summary for grade transition of A to B

Standard GT		$m_{Off}\,(tons)$			Desired		Off-spec	
Case	$t_{off}\,(hr)$	generated	consumed	net	M_{n} (x1e4)	PDI	M_{n} (x1e4)	PDI
AB	12.5	58.3	-	58.3	5.67	6.58	-	-
AB1	-	-	-	-	5.67	6.58	5	5
AB2	0.07	0.28	54.6	-54.3	5.67	6.58	6	7
AB3	-	-	-	-	5.67	6.58	5	7
AB4	11.4	53.2	0.10	53.2	5.67	6.58	6	5
Optimal GT								
AB	6.78	32.6	-	32.6	5.67	6.58	-	-
AB1	-	-	-	-	5.67	6.58	5	5
AB2	0.05	0.20	41.9	-41.7	5.67	6.58	6	7
AB3	6.01	28.7	0.67	28.0	5.67	6.58	5	7
AB4	-	-	-	-	5.67	6.58	6	5

7.6 Closing remarks

The approaches used in the current chapter are by no means exhaustive in terms of optimally reducing the amount of off-spec material on a plant, yet the potential to greatly improve the savings on a plant are already evident. As previously mentioned, the optimal flow profiles were set up as series of step functions for proof of concept. The control vectors should be set up in such a way that they can better approximate true optimal profiles in order

to truly minimise the amount of off-spec generated whilst maximising the amount of existing off-spec polymer consumed.

The amount of the latter type of off-spec material depends entirely on its properties. These illustrative case studies have shown that tons of off-spec material can be blended with negligible new off-spec material generated during a grade transition.

8 CONCLUSIONS AND RECOMMENDATIONS

The main aim of this study was to address the issue of the accumulation of off-spec polymer generated on a plant by way of blending it with virgin polymer to produce a desirable and saleable product. Currently, the most promising solution for the minimisation of off-spec polymer created during grade transitions is that of optimally controlling feed rates and process conditions such that the transition time is reduced. The work done in this project applies similar mathematical techniques to off-spec blending.

A method has been proposed whereby the polymer product of a reactor is blended with off-spec material in a blending unit for the purpose of creating a blend with the same average properties of a grade that is normally produced by the plant.

A blending model was developed to predict the polymer properties of the blend from its constituents. It was proven that the blend of polymers with different properties could be determined by a mathematical relation, which indicated that the mass moments of individual polymers are additive on a mass basis. Four polymer grades were blended with one another in various blend ratios. The properties of the blends and the pure grades were determined using GPC analysis to validate the proposed mathematical relationship. The comparison of the model predictions with this experimental data showed that they were in agreement. Thus, the relationship forms the underlying basis for the simulation of blending where a reactor product is blended with off-spec material from storage to create a saleable polymer.

From the bulk moments, the average properties describing a polymer's CLD can be determined, namely M_n and PDI. These two properties were chosen to fully describe the grade of the product. Thus, a rigorous process model was developed based on fundamental principles to predict these properties and the rate of production under various process conditions and reactant feed rates.

Using these two models, a control strategy was developed to demonstrate that properties of the reactor product could be adjusted to accommodate the introduction of a continuous stream of off-spec polymer to the system such that the resultant blend of the two polymers remained within specification. Combinations of feed rates were found for off-spec polymer with properties that are extreme in relation to the desired grade's properties.

Grade A precipitate and off-spec source 1 are able to be comfortably blended with virgin to off-spec ratios of up to 2.15:1, whereas Grade A and off-spec source 2 can be blend with ratios of up to 4.38:1. After these points, the controller adjusts the feed rates such that the properties of the intermediate precipitate for blending are outside of their respective specification bands. These threshold values were calculated post-optimisation, thus it is

recommended that the model be extended to automatically determine the maximum flowrate at which the off-spec polymer can be continuously fed to the system. The resultant fresh polymer must have properties that remain within their respective bands. Further, constraints should be determined for the reactant flowrates, such that the controller does not predict a combination of feed rates that are physically impossible or dangerous to operate or could lead to a significantly reduced production rate. For instance, a reactor might be designed to operate at a certain pressure. Significantly altering the catalyst flowrate would affect the production rate, which would, in turn, change the concentration of reactants in the gas phase. This could occur to such an extent that the design pressure of the reactor is exceeded.

Two more approaches were proposed whereby blending was performed during grade transitions in order to instantly blend off-spec polymer as it is generated with a stored off-spec polymer in predetermined blend ratios via model-predictive control. In order to do so, the controller was developed to determine the optimal trajectory for the off-spec polymer flowrate, in the same manner that feed rate profiles are determined for grade transition optimisation. The first approach has blending performed during a traditional grade transition, whereas the second approach has it performed during an optimised grade transition where feed rate profiles are also predetermined.

It was found that depending on the properties of the polymer relative to the specifications of the grades and depending on which property is limiting, certain off-spec polymer types are highly, only slightly, or not at all suitable for blending during a grade transition. Not only does manipulating feed rate profiles reduce the transition time and mass of off-spec polymer generated, but it can also change which polymer property is limiting, thus allowing other off-spec polymer types to be slightly suitable for blending.

For the transition from grade B to grade A where 44 tons of off-spec is generated , 62 tons of off-spec source 1 was able to form an on-spec blend throughout most of the transient period, whereas off-spec source 3 could only reduce the transient time by 7%. In contrast, adjusting the monomer and hydrogen flow profiles reduced the transition time by 22%. The best result is obtained by adjusting these feed rate profiles and blending with off-spec source 4 to reduce the transition to 33% of the original time.

Similarly, for the transition from Grade A to Grade B, 58 tons of off-spec from source 2 can be blended such that the amount of off-spec generated weighs less than a ton. Varying the profiles of the feed rates can reduce the transition time by over 45%, generating off-spec material that amounts to 32 tons. Approximately 42 tons of off-spec material from source 2 can be blended with most of this amount to produce an on-spec polymer.

In these approaches, it is suggested that other discretisation strategies be investigated so as to better approximate the true optimal profiles. A starting point would be to increase the number of time intervals; or to replace the step functions with ramp functions. This could result in smoother trajectories such that the properties are more uniform within the specification bands throughout blending. In addition, the interval over which blending is feasible can increase by better approximating the control flow profiles, thus resulting in a further reduction in transition time and the amount of off-spec material.

For all of the approaches presented, the chain length distribution, represented by M_n and PDI, was decided to be sufficient to characterise a grade. With this limited definition of a polymer grade, it can be stated that the primary objectives of the study have been realised. Control strategies have been developed that effectively manipulate the flowrate of an off-spec polymer source and/or the flowrates of the reactor feed in order to reduce the overall amount of off-spec material on a plant by consumption of the existing off-spec polymer source and/or by reduction of off-spec material generated during a grade transition.

Based on these results, it can be confirmed that the flowrates of reactor feed streams and/or the flowrate from an off-spec polymer source can be manipulated by a model-predictive controller in order to reduce the overall amount of off-spec material on a plant by consumption of the existing off-spec polymer source and/or reduction of off-spec material generated during a grade transition.

Further investigation is required that extends the definition of a polymer grade to include the short-chain branching distribution and the comonomer content distribution. This could pave the way to quantitatively predict the density of a polymer based on fundamental principles. Such an investigation would include the further development of the process model to predict the SCB distribution and CCD of the precipitate. The blending model will also need to be updated to incorporate these two properties. Only then can the dynamic optimisations covered in this study be revisited in order to determine the feasibility of off-spec blending.

Realistically, the stored off-spec polymer is not homogenised by mixing, as it was assumed in the current study. Thus, the properties of an off-spec polymer stream will vary with time. Obtaining these properties online is an obstacle that needs to be overcome in order to realistically control a system that takes into account non-homogeneity of the off-spec polymer properties.

Off-spec blending has been investigated for two properties that describe only the chain length distribution. Some simplifications were made in simulating the process, since it is only the principle that is being investigated. Standard optimisation tools were used to optimise a fictitious plant for steady-state blending and for blending during grade transitions. The latter

method was compared to and combined with optimal grade transition practices. In all of the methods, there were cases where off-spec blending proved highly beneficial in creating saleable products. It is for this reason that it is believed that the approach of control through blending is justified and should be pursued further.

REFERENCES

Abass, K.B., Knutsson, A.B. & Berglund, S.H., 1978. New thermoplastics from old. *Chemtech*. 8:502–508.

Akkapeddi, M.K., 2003. Commercial Polymer Blends. In Utracki, L. A., ed. *Polymer Blends Handbook, Volume 2*. Springer. 1023–1117.

Al-haj Ali, M., 2010. Optimal grade transition control for liquid-propylene polymerization reactor. *Asian Journal of Control*. 12(3):413–425.

Al-Salem, S.M., Lettieri, P. & Baeyens, J., 2009. Recycling and recovery routes of plastic solid waste (PSW): a review. *Waste management*. 29(10):2625–43.

Alt, H.G. & Köppl, A., 2000. Effect of the nature of metallocene complexes of Group IV metals on their performance in catalytic ethylene and propylene polymerization. *Chemical Reviews*. 100(4):1205–1222.

Arlman, E., 1964. Ziegler-Natta catalysis II. Surface structure of layer-lattice transition metal chlorides. *Journal of Catalysis*. 3(1):89–98.

AspenTech, 2011. Aspen Properties. [Computer Software]

Babu, B.V. & Angira, R., 2006. Modified differential evolution (MDE) for optimization of non-linear chemical processes. *Computers & Chemical Engineering*. 30(6-7):989–1002.

Bahri-Laleh, N., Correa, A., Mehdipour-Ataei, S., Arabi, H., Haghighi, M.N., Zohuri, G. & Cavallo, L., 2011. Moving up and down the titanium oxidation state in Ziegler–Natta catalysis. *Macromolecules*. 44(4):778–783.

Banu, I., Bozga, G., Nagy, I. & Puaux, J.-P., 2008. A Comparison of Variational and Genetic Algorithm Performances in the Optimization of a Polymerization Process. *Chemical Engineering & Technology*. 31(10):1516–1525.

BenAmor, S., Doyle, F.J. & McFarlane, R., 2004. Polymer grade transition control using advanced real-time optimization software. *Journal of Process Control*. 14(4):349–364.

Bernardo, C.A., Cunha, A.M. & Oliveira, M.J., 1996. The recycling of thermoplastics: Prediction of the properties of mixtures of virgin and reprocessed polyolefins. *Polymer Engineering & Science*. 36(4):511–519.

Bhaduri, S., Mukhopadhyay, S. & Kulkarni, S. a., 2006. Role of titanium oxidation states in polymerization activity of Ziegler–Natta catalyst: A density functional study. *Journal of Organometallic Chemistry*. 691(12):2810–2820.

Bhateja, S.K. & Andrews, E.H., 1983. Thermal, mechanical, and rheological behavior of blends of ultrahigh and normal-molecular-weight linear polyethylenes. *Polymer Engineering & Science*. 23(16):888–894.

Biegler, L.T., 1992. Optimization strategies for complex process models. In Anderson, J. J. L. & Bischof, K. B., eds. *Advances in Chemical Engineering, Volume 18*. New York: Academic Press, Inc. 197–256.

Boero, M., Parrinello, M., Weiss, H. & Hüffer, S., 2001. A First Principles Exploration of a Variety of Active Surfaces and Catalytic Sites in Ziegler–Natta Heterogeneous Catalysis. *The Journal of Physical Chemistry A*. 105(21):5096–5105.

Boggs, P.T. & Tolle, J.W., 1995. Sequential Quadratic Programming. *Acta Numerica*. 4:1–51.

Böhm, L.L., 1978. Ethylene polymerization process with a highly active Ziegler-Natta catalyst: 1. Kinetics. *Polymer*. 19(5):553–561.

Boni, K.A., Sliemers, F.A. & Stickney, P.B., 1968. Development of gel permeation chromatography for polymer characterization. II. Universal calibration. *Journal of Polymer Science Part A-2: Polymer Physics*. 6(9):1579–1591.

Bonvin, D., Bodizs, L. & Srinivasan, B., 2005. Optimal Grade Transition for Polyethylene Reactors via NCO Tracking. *Chemical Engineering Research and Design*. 83(6):692–697.

Bryson, A.E. & Ho, Y.C., 1975. *Applied Optimal Control: Optimization, Estimation, and Control*, Taylor & Francis.

Buehren, M., 2008. Differential Evolution [MATLAB Package]. *MATLAB Central*. Available: http://www.mathworks.com/matlabcentral/fileexchange/18593-differential-evolution

Busico, V., 2013. Ziegler-Natta catalysis: Forever young. *MRS Bulletin*. 38(03):224–228.

Cameron, A.C. & Windmeijer, F.A.G., 1997. An R-squared measure of goodness of fit for some common nonlinear regression models. *Journal of Econometrics*. 77(2):329–342.

De Carvalho, A.B.M., Gloor, P.E. & Hamielec, A.E., 1990. A kinetic model for heterogeneous Ziegler-Natta (co)polymerization. Part 2: stereochemical sequence length distributions. *Polymer*. 31(7):1294–1311.

Cecchin, G., Morini, G. & Piemontesi, F., 2000. Ziegler-Natta Catalysts. In *Kirk-Othmer Encyclopedia of Chemical Technology*. John Wiley & Sons, Inc.

Cervantes, A.M., Tonelli, S., Brandolin, A., Bandoni, J.A. & Biegler, L.T., 2002. Large-scale dynamic optimization for grade transitions in a low density polyethylene plant. *Computers & Chemical Engineering*. 26(2):227–237.

Cervantes, A.M., Wächter, A., Tütüncü, R.H. & Biegler, L.T., 2000. A reduced space interior point strategy for optimization of differential algebraic systems. *Computers & Chemical Engineering*. 24(1):39–51.

Cervenka, A., 1973. The MARK-HOUWINK equation for linear polyethylene in 1,2,4-trichlorobenzene at 135°C. *Die Makromolekulare Chemie*. 170(1):239–241.

Chakravarthy, S.S.S., Saraf, D.N. & Gupta, S.K., 1997. Use of genetic algorithms in the optimization of free radical polymerizations exhibiting the trommsdorff effect. *Journal of Applied Polymer Science*. 63(4):529–548.

Chatzidoukas, C., Perkins, J.D., Pistikopoulos, E.N. & Kiparissides, C., 2003. Optimal grade transition and selection of closed-loop controllers in a gas-phase olefin polymerization fluidized bed reactor. *Chemical Engineering Science*. 58(16):3643–3658.

Cheng, R., Luo, J., Liu, Z., Sun, J., Huang, W., Zhang, M., Yi, J. & Liu, B., 2013. Adsorption of TiCl4 and electron donor on defective MgCl2 surfaces and propylene polymerization over Ziegler-Natta catalyst: A DFT study. *Chinese Journal of Polymer Science*. 31(4):591–600.

Chien, J.C.W. & Nozaki, T., 1991. High activity magnesium chloride supported catalysts for olefin polymerization. XXIX. Molecular basis of hydrogen activation of magnesium chloride supported Ziegler–Natta catalysts. *Journal of Polymer Science Part A: Polymer Chemistry*. 29(4):505–514.

Chien, J.C.W., Weber, S. & Hu, Y., 1989. Magnesium chloride supported catalysts for olefin polymerization. XIX. Titanium oxidation states, catalyst deactivation, and active site structure. *Journal of Polymer Science Part A: Polymer Chemistry*. 27(5):1499–1514.

Cleveland, W. & Loader, C., 1996. Smoothing by Local Regression: Principles and Methods. In Härdle, W. & Schimek, M., eds. *Statistical Theory and Computational Aspects of Smoothing SE - 2*. Contributions to Statistics. Physica-Verlag HD. 10–49.

Congalidis, J.P. & Richards, J.R., 1998. Process Control of Polymerization Reactors: An Industrial Perspective. *Polymer Reaction Engineering*. 6(2):71–111.

Corradini, P., Guerra, G. & Cavallo, L., 2004. Do new century catalysts unravel the mechanism of stereocontrol of old Ziegler-Natta catalysts? *Accounts of chemical research*. 37(4):231–241.

Cossee, P., 1964. Ziegler-Natta catalysis I. Mechanism of polymerization of α-olefins with Ziegler-Natta catalysts. *Journal of Catalysis*. 3(1):80–88.

Dimitratos, J., Georgakis, C., El-Aasser, M.S. & Klein, A., 1989. Dynamic modeling and state estimation for an emulsion copolymerization reactor. *Computers & Chemical Engineering*. 13(1-2):21–33.

Doi, Y., Murata, M., Yano, K. & Keii, T., 1982. Gas-phase polymerization of propene with the supported Ziegler catalyst: TiCl4/MgCl2/C6H5COOC2H5/Al(C2H5)3. *Industrial & Engineering Chemistry Product Research and Development*. 21(4):580–585.

Dotson, N.A., Galvan, R. & Laurence, R.L., 1995. *Polymerization Process Modeling*, Wiley.

Embirucu, M., Lima, E.L. & Pinto, J.C., 1996. A survey of advanced control of polymerization reactors. *Polymer Engineering & Science*. 36(4):433–447.

Farauto, R.J., 2007. Industrial Catalysis: A Practical Guide. In Kent, J. A., ed. *Kent and Riegel's Handbook of Industrial Chemistry and Biotechnology*. Springer Science and Business Media. 302.

Feather, D., Harrell, D., Lieberman, R. & Doyle, F.J., 2004. Hybrid approach to polymer grade transition control. *AIChE Journal*. 50(10):2502–2513.

Flores-Tlacuahuac, A. & Biegler, L.T., 2008. Integrated control and process design during optimal polymer grade transition operations. *Computers & Chemical Engineering*. 32(11):2823–2837.

Flores-Tlacuahuac, A., Biegler, L.T. & Saldívar-Guerra, E., 2005. Dynamic Optimization of HIPS Open-Loop Unstable Polymerization Reactors. *Industrial & Engineering Chemistry Research*. 44(8):2659–2674.

Flores-Tlacuahuac, A., Biegler, L.T. & Saldívar-Guerra, E., 2006. Optimal Grade Transitions in the High-Impact Polystyrene Polymerization Process. *Industrial & Engineering Chemistry Research*. 45(18):6175–6189.

Flory, P.J., 1953. *Principles of Polymer Chemistry*, Cornell University Press.

Fogler, H.S., 2006. *Elements of Chemical Reaction Engineering* 4th ed., Prentice Hall.

Galvan, R. & Tirrell, M., 1986. Molecular weight distribution predictions for heterogeneous Ziegler-Natta polymerization using a two-site model. *Chemical Engineering Science*. 41(9):2385–2393.

Gen, M. & Cheng, R., 2000. *Genetic Algorithms and Engineering Optimization*, Wiley.

Gibson, V.C. & Spitzmesser, S.K., 2003. Advances in non-metallocene olefin polymerization catalysis. *Chemical reviews*. 103(1):283–315.

Gisnas, A., 2003. Optimal grade transitions for polyethylene reactors. *Computer Aided Chemical Engineering*. 15:463–468.

Global Markets Direct, 2009. *The Future of the Polyethylene Industry to 2015: Investment Opportunities, Analysis and Forecasts of All Active and Planned Polyethylene Plants*, Ireland.

Han-Adebekun, G.C. & Ray, W.H., 1997. Polymerization of olefins through heterogeneous catalysis. XVII. Experimental study and model interpretation of some aspects of olefin polymerization over a TiCl4/MgCl2 catalyst. *Journal of Applied Polymer Science*. 65(6):1037–1052.

Hansjörg Sinn & Walter Kaminsky, 1980. Ziegler-Natta Catalysis. In Gordon, A., ed. *Advances in Organometallic Chemistry, Volume 18*. New York: Academic Press, Inc. 99–149.

Heaton, C.A., 1994. *The Chemical Industry* 2nd ed., Glascow: Blackie Academic & Professional.

Holland, C.D. & Anthony, R.G., 1979. *Fundamentals of Chemical Reaction Engineering*, Prentice-Hall.

Houwink, R., 1940. Zusammenhang zwischen viscosimetrisch und osmotisch bestimmten Polymerisationsgraden bei Hochpolymeren. *Journal für praktische Chemie*. 157(1):15–18.

Hutchinson, R.A., Chen, C.M. & Ray, W.H., 1992. Polymerization of olefins through heterogeneous catalysis X: Modeling of particle growth and morphology. *Journal of Applied Polymer Science*. 44(8):1389–1414.

Immanuel, C.D. & Doyle III, F.J., 2003. Computationally efficient solution of population balance models incorporating nucleation, growth and coagulation: application to emulsion polymerization. *Chemical Engineering Science*. 58(16):3681–3698.

Isayev, A.I. ed., 2010. *Encyclopedia of Polymer Blends: Volume 1: Fundamentals*, Weinheim: John Wiley & Sons.

Jejelowo, M., Lynch, D. & Wanke, S., 1991. Comparison of ethylene polymerization in gas-phase and slurry reactors. *Macromolecules*. 24(8):1755–1761.

Kasat, R.B., Ray, A.K. & Gupta, S.K., 2003. Applications of genetic algorithm in polymer science and engineering. *Materials and Manufacturing Processes*. 18(3):523–532.

Kim, I., Kim, J.H. & Woo, S.I., 1990. Kinetic study of ethylene polymerization by highly active silica supported TiCL4/MgCl2 catalysts. *Journal of Applied Polymer Science*. 39(4):837–854.

Kiparissides, C., 2006. Challenges in particulate polymerization reactor modeling and optimization: A population balance perspective. *Journal of Process Control*. 16(3):205–224.

Kiparissides, C., 1996. Polymerization reactor modeling: a review of recent developments and future directions. *Chemical Engineering Science*. 51(10):1637–1659.

Kirk, D.E., 2004a. Numerical Determination of Optimal Trajectories. In *Optimal Control Theory: An Introduction*.

Kirk, D.E., 2004b. The Calculus of Variations. In *Optimal Control Theory: An Introduction*.

Lampinen, J. & Zelinka, I., 2000. On stagnation of the differential evolution algorithm. In *Proceedings of MENDEL 2000, 6th International Mendel Conference on Soft Computing*. 76–83.

Lee, M.H., Han, C. & Chang, K.S., 1999. Dynamic optimization of a continuous polymer reactor using a modified differential evolution algorithm. *Industrial & Engineering Chemistry Research*. 38(12):4825–4831.

Lee, M.H., Han, C. & Chang, K.S., 1997. Hierarchical time-optimal control of a continuous copolymerization reactor during start-up or grade change operation using genetic algorithms. *Computers & Chemical Engineering*. 21(Supplement):S1037–S1042.

Lewis, R.M., Torczon, V. & Trosset, M.W., 2000. Direct search methods: then and now. *Journal of Computational and Applied Mathematics*. 124(1-2):191–207.

Lo, D.P. & Ray, W.H., 2006. Dynamic Modeling of Polyethylene Grade Transitions in Fluidized Bed Reactors Employing Nickel–Diimine Catalysts. *Industrial & Engineering Chemistry Research*. 45(3):993–1008.

Malpass, D., 2010. *Introduction to Industrial Polyethylene: Properties, Catalysts, and Processes*, Salem: Scrivener Publishing LLC.

Mantia, F. La, 2002. *Handbook of Plastics Recycling*, iSmithers Rapra Publishing.

Mark, H., 1938. *Der feste Körper* Hirzel, S., ed., Leipzig.

Matyjaszewski, K. & Davis, T.P., 2003. *Handbook of Radical Polymerization*, John Wiley & Sons.

McAuley, K.B. & MacGregor, J.F., 1992. Optimal grade transitions in a gas phase polyethylene reactor. *AIChE Journal*. 38(10):1564–1576.

McAuley, K.B., MacGregor, J.F. & Hamielec, A.E., 1990. A kinetic model for industrial gas-phase ethylene copolymerization. *AIChE Journal*. 36(6):837–850.

McCoy, J.T. & Rawatlal, R., 2012. A more efficient simulator of particle size distribution in slurry phase polyolefin systems. *Computers & Chemical Engineering*. 36:68–78.

McCoy, J.T. & Rawatlal, R., Analysis of experimental activity profiles for ZN catalysts.

McCoy, J.T. & Rawatlal, R., 2011. Another look at site heterogeneity in Ziegler-Natta catalysts for polyolefin production. In *11th Annual UNESCO/IUPAC Workshop and Conference*.

McCoy, J.T., Rawatlal, R. & Soares, J.P.B., 2012. *Development of a computationally efficient model for the control of the Ziegler-Natta catalysed industrial production of high density polyethylene*. University of Cape Town.

McKenna, T.F. & Soares, J.B.P., 2001. Single particle modelling for olefin polymerization on supported catalysts: A review and proposals for future developments. *Chemical Engineering Science*. 56(13):3931–3949.

Michalewicz, Z., 1996. *Genetic Algorithms + Data Structures = Evolution Programs*, Springer.

Muñoz-Escalona, A., Hernandez, J.G. & Gallardo, J.A., 1984. Catalytic activity and control of the nascent morphology of polyethylenes obtained with first and second generation of Ziegler–Natta catalysts. *Journal of Applied Polymer Science*. 29(4):1187–1202.

Nagel, E., Kirillov, V. & Ray, W., 1980. Prediction of molecular weight distributions for high-density polyolefins. *Industrial & Engineering Chemistry Product Research and Development*. 19(3):372–379.

Nath, S., Bodhak, S. & Basu, B., 2009. HDPE-Al2O3-HAp composites for biomedical applications: processing and characterizations. *Journal of biomedical materials research. Part B, Applied biomaterials*. 88(1):1–11.

Natta, G. & Pino, P., 1957. A crystallizable organometallic complex containing titanium and aluminum. *Journal of the American Chemical Society*. 1004(1):2975–2976.

Natta, G., Pino, P., Corradini, P., Danusso, F., Mantica, E., Mazzanti, G. & Moraglio, G., 1955. Crystalline high polymers of α-olefins. *Journal of the American Chemical Society*. 77(6):1708–1710.

Nelder, J.A. & Mead, R., 1965. A Simplex Method for Function Minimization. *The Computer Journal*. 7(4):308–313.

Osswald, T.A. & Hernández-Ortiz, J.P., 2006. *Polymer Processing: Modeling and Simulation* 1st ed. Verlag, C. H., ed., Munich.

Pater, J.T.M., Weickert, G. & Swaaij, W.P.M. Van, 2003. Polymerization of liquid propylene with a fourth-generation Ziegler-Natta catalyst: Influence of temperature, hydrogen, monomer concentration, and prepolymerization method on powder morphology. *Journal of Applied Polymer Science*. 87(9):1421–1435.

Paulen, R., Fikar, M. & Latifi, M.A., 2010. Dynamic optimization of a polymerization reactor. In *Control & Automation (MED)*. 733–738.

Penlidis, A., MacGregor, J.F. & Hamielec, A.E., 1985. Dynamic modeling of emulsion polymerization reactors. *AIChE Journal*. 31(6):881–889.

Perry, R.H. & Green, D.W., 2007. *Perry's Chemical Engineers' Handbook*, McGraw-Hill Professional Publishing.

Ponnuswamy, S.R., Shah, S.L. & Kiparissides, C.A., 1987. Computer optimal control of batch polymerization reactors. *Industrial & Engineering Chemistry Research*. 26(11):2229–2236.

Pontriagin, L.S., 1962. *The mathematical theory of optimal processes*, John Wiley.

Prasetya, A., Liu, L., Litster, J., Watanabe, F., Mitsutani, K. & Ko, G.H., 1999. Dynamic model development for residence time distribution control in high-impact polypropylene copolymer process. *Chemical Engineering Science*. 54(15-16):3263–3271.

Price, K., Storn, R.M. & Lampinen, J.A., 2005. *Differential Evolution: A Practical Approach to Global Optimization*, Springer.

Ram, A., 1997. *Fundamentals of Polymer Engineering*, Springer.

Rawatlal, R., 2004. *Unsteady-state Modelling of Ziegler-Natta Catalyzed Olefin Polymerization Reactor Systems*. University of Kwa-Zulu Natal.

Rawatlal, R. & Starzak, M., 2003. Unsteady-state residence-time distribution in perfectly mixed vessels. *AIChE Journal*. 49(2):471–484.

Ray, W.H., 1981. *Advanced Process Control*, McGraw-Hill.

Ray, W.H., 1991. Modelling of addition polymerization processes - Free radical, ionic, group transfer, and ziegler-natta kinetics. *The Canadian Journal of Chemical Engineering*. 69(3):626–629.

Robeson, L.M., 2007. *Polymer Blends: A Comprehensive Review*, Hanser Verlag.

Sandler, S.I., 2006. *Chemical, Biochemical, and Engineering Thermodynamics*, John Wiley & Sons.

Sayer, C., Lima, E.L. & Pinto, J.C., 1997. Dynamic modeling of SBR emulsion polymerization reactors refrigerated by thermosyphons. *Chemical Engineering Science*. 52(3):341–356.

Schlegel, M., Stockmann, K., Binder, T. & Marquardt, W., 2005. Dynamic optimization using adaptive control vector parameterization. *Computers & Chemical Engineering*. 29(8):1731–1751.

Soares, J.B.P. & Hamielec, A.E., 1995. Deconvolution of chain-length distributions of linear polymers made by multiple-site-type catalysts. *Polymer*. 36(11):2257–2263.

Soga, K., Chen, S.-I. & Ohnishi, R., 1982. Correlation between the oxidation states of titanium and the polymerization activities for higher alpha-olefins and diene compounds. *Polymer Bulletin*. 8(9-10):473–478.

Storn, R. & Price, K., 1997. Differential evolution-a simple and efficient heurstic for global optimization over continuous spaces. *Journal of global optimization*. 11(4):341–359.

Takeda, M. & Ray, W.H., 1999. Optimal-grade transition strategies for multistage polyolefin reactors. *AIChE Journal*. 45(8):1776–1793.

Tetiker, M.D., Artel, A., Teymour, F. & Cinar, A., 2008. Control of grade transitions in distributed chemical reactor networks—An agent-based approach. *Computers & Chemical Engineering*. 32(9):1984–1994.

Throne, J.L., 1987. Effect of recycle on properties and profits: Algorithms. *Advances in Polymer Technology*. 7(4):347–360.

Tsoukas, A., Tirrell, M. & Stephanopoulos, G., 1982. Multiobjective dynamic optimization of semibatch copolymerization reactors. *Chemical Engineering Science*. 37(12):1785–1795.

Utracki, L.A., 2003. Role of Polymer Blends' Technology in Polymer Recycling. In Zaccaria, V. K., ed. *Polymer Blends Handbook, Volume 2*. 1119.

Vasile, C. & Pascue, M., 2005. *Practical Guide to Polyethylene*, Shrewsbury: Rapra Technology Limited.

Wang, J. & Yang, Y., 2003. Study on optimal strategy of grade transition in industrial fluidized bed gas-phase polyethylene production process. *Chinese Journal of Chemical Engineering*. 11(1):1–8.

Wang, Y., Seki, H., Ohyama, S., Akamatsu, K., Ogawa, M. & Ohshima, M., 2000. Optimal grade transition control for polymerization reactors. *Computers & Chemical Engineering*. 24(2-7):1555–1561.

Webb, S.W., Conner, W.C. & Laurence, R.L., 1989. Monomer transport influences in the nascent polymerization of ethylene by silica-supported chromium oxide catalyst. *Macromolecules*. 22(7):2885–2894.

Webb, S.W., Weist, E.L., Chiovetta, M.G., Laurence, R.L. & Conner, W.C., 1991. Morphological influences in the gas phase polymerization of ethylene by silica supported chromium oxide catalysts. *The Canadian Journal of Chemical Engineering*. 69(3):665–681.

Yiannoulakis, H., Yiagopoulos, A. & Kiparissides, C., 2001. Recent developments in the particle size distribution modeling of fluidized-bed olefin polymerization reactors. *Chemical Engineering Science*. 56(3):917–925.

Zheng, Z.-W., Shi, D.-P., Su, P.-L., Luo, Z.-H. & Li, X.-J., 2011. Steady-State and Dynamic Modeling of the Basell Multireactor Olefin Polymerization Process. *Industrial & Engineering Chemistry Research*. 50(1):322–331.

Ziegler, K., 1964. Folgen und Werdegang einer Erfindung Nobel-Vortrag am 12. Dezember 1963. *Angewandte Chemie*. 76(13):545–553.

A. GPC data validation

In the following section, the calculations are given for processing GPC analysis in preparation for validating the model predictions.

A portion of the raw data from GPC measurements of blend B1D3 are shown below.

<Administrative_Information>		
Decimal Separator	.	
Analyst	HT-SEC	
Analysis Type	Single Channel	
Analysed As	Unknown	
Software Version	Cirrus 3.3	
</Administrative_Information>		
<Sample_Information>		
Sample Type	Unknown	
Sample Name	Sample 1	
K * 10e5 Used		14.1
Alpha Used		0.7
Conc (mg/ml)		0.9
Injection Volume (ul)		200
</Sample_Information>		
<Sample_GLP_Information>		
Prep Procedure		
History		
LIMS ID		
Origin		
Urgency		0
</Sample_GLP_Information>		
<Collection_Information>		
Date Collected		2012/10/08 12:32
Run Length (mins)		40
Sampling Interval (Hz)		5
Collected By	HT-SEC	
Autosampler Position		1
Vial Number		1
Bar Code		
</Collection_Information>		
<Analysis_Conditions>		
Eluent	TCB stabilised with 0.0125% BHT	
Eluent RI		1.571
Set Flow Rate (ml/min)		1
Set Temperature		150
Column Set Name	3 x PLGel Olexis + 1 x PLGel Olexis Guard	
Column Set Length (mm)		950
Detector Name	RI	
</Analysis_Conditions>		
<Processing_Method_Information>		
Method Name		Jan-12
Created On		2012/01/06 08:18
Method Created By	HT-SEC	
Last Modified		2012/01/12 15:50
Last Calibrated		2012/10/04 10:49
Method Comments		
</Processing_Method_Information>		
<Flowrate_Information>		
FRM Name	BHT	
FRM Peak Number		2
RT (mins)		25.61
FRCF		1.01679
</Flowrate_Information>		
<Processed_Peak_Information>		
No of Processed Peaks		2

Peak No	Name	Start RT (mins)	Max RT (mins)	End RT (mins)	Height (mV)	% Height	Area (mV.secs)	% Area
1		15.0667	19.88	24.3167	19.7362	84.3077	3961.97	96.4856
2		24.9833	25.61	26.37	3.67354	15.6923	144.312	3.51443

</Processed_Peak_Information>

<Peak_Detection_Information>
No of Detected Peaks 2

Peak No	Type	St Detect Code	End Detect Code	Is St Mod	Is Max Mod	Is End Mod
1	0	66	86	No	No	No
2	0	1	1	No	No	No

</Peak_Detection_Information>

<Baseline_Information>
No of Baselines 2

No	Start RT (mins)	End RT (mins)	Start Height	End Height	Is St Mod	Is End Mod
1	14.9867	24.5133	-6.32294	-4.58221	No	No
2	24.9833	26.37	-4.27963	-2.88345	No	No

</Baseline_Information>

<Calibration_Used>

Calibration Type	Narrow Standard	
Created On		2012/10/04 10:46
Last Modified		2012/10/04 10:49
Version Number		10
Calibration Created By	HT-SEC	
High Limit RT (mins)		15.5033
Low Limit RT (mins)		24.5583
High Limit MW		6294491
Low Limit MW		588
K Calibration		14.1
Alpha Calibration		0.7
Calibration Flow Marker Name	BHT	
Calibration Flow Marker RT (mins)		26.04
Calibration Comments		

</Calibration_Used>

<Calibration_Curve_Details>

Number Of Regions			1					
Region No	Polynomial Order	Coeff a	Coeff bx	Coeff cx^2	Coeff dx^3	Coeff ex^4	Coeff fx^5	
1	3	13.0961	-0.322665	-0.00779011	0.000155017	0	0	

</Calibration_Curve_Details>

<Calibration_Curve_Statistics>

Residual Sum Of Squares	0.00558343
Corrected Sum Of Squares	17.7989
Coefficient Of Determination	0.999686
Standard Y Error Estimate	0.0334169
Linear Correlation Coefficient	-0.99981

</Calibration_Curve_Statistics>

<MW_Averages>

No of GPC Peaks			2				
Peak No	Mp	Mn	Mw	Mz	Mz+1	Mv	PD
1		46894	16690	141534	1018024	2722441	101243	8.48017
2		0	0	0	0	0	0	0

</MW_Averages>

<Slice_Table>

	A	B	C	D	E	F	G	H	I	J
1	</MW_Averages>									
2	<Slice_Table>									
3	Peak No	1								
4	RT (mins)	Response (mV)	Norm Ht	Cum Ht	MW	logMW	dw_dlogM	W(M)	Conc	Outside Calib
5	15.2833	0.0949304	3.24E-06	3.24E-06	6332309	6.80156	0.00211544	1.45E-10	0	Yes
6	15.2867	0.0852528	2.91E-06	6.14E-06	6310139	6.80004	0.00189977	1.31E-10	0	Yes
7	15.29	0.0722183	2.46E-06	8.60E-06	6287937	6.79851	0.00160929	1.11E-10	0	No
8	15.2933	0.0617779	2.11E-06	1.07E-05	6265813	6.79698	0.00137663	9.54E-11	0	No
9	15.2967	0.0539314	1.84E-06	1.25E-05	6243767	6.79545	0.00120177	8.36E-11	0	No
10	15.3	0.0509677	1.74E-06	1.43E-05	6221798	6.79392	0.00113572	7.93E-11	0	No
11	15.3033	0.0515135	1.76E-06	1.60E-05	6200014	6.79239	0.00114787	8.04E-11	0	No
12	15.3067	0.0523645	1.78E-06	1.78E-05	6178199	6.79086	0.00116683	8.20E-11	0	No
13	15.31	0.0542837	1.85E-06	1.97E-05	6156460	6.78933	0.00120958	8.53E-11	0	No
14	15.3133	0.0587968	2.00E-06	2.17E-05	6134798	6.7878	0.00131013	9.27E-11	0	No
15	15.3167	0.0636151	2.17E-06	2.38E-05	6113211	6.78627	0.00141748	1.01E-10	0	No
16	15.32	0.0716377	2.44E-06	2.63E-05	6091806	6.78475	0.00159623	1.14E-10	0	No
17	15.3233	0.0817966	2.79E-06	2.91E-05	6070371	6.78322	0.00182258	1.30E-10	0	No
18	15.3267	0.0887511	3.02E-06	3.21E-05	6049011	6.78168	0.00197752	1.42E-10	0	No
19	15.33	0.0950953	3.24E-06	3.53E-05	6027725	6.78015	0.00211886	1.53E-10	0	No
20	15.3333	0.102813	3.50E-06	3.88E-05	6006515	6.77862	0.0022908	1.66E-10	0	No
21	15.3367	0.109462	3.73E-06	4.26E-05	5985379	6.77709	0.00243893	1.77E-10	0	No
22	15.34	0.115959	3.95E-06	4.65E-05	5964421	6.77557	0.00258367	1.88E-10	0	No
23	15.3433	0.122761	4.18E-06	5.07E-05	5943432	6.77404	0.0027352	2.00E-10	0	No
24	15.3467	0.124527	4.24E-06	5.49E-05	5922518	6.77251	0.00277453	2.03E-10	0	No
25	15.35	0.123395	4.20E-06	5.92E-05	5901676	6.77098	0.00274927	2.02E-10	0	No
26	15.3533	0.124246	4.23E-06	6.34E-05	5880908	6.76944	0.00276821	2.04E-10	0	No
27	15.3567	0.129064	4.40E-06	6.78E-05	5860315	6.76792	0.00287554	2.13E-10	0	No
28	15.36	0.133272	4.54E-06	7.23E-05	5839692	6.76639	0.00296927	2.21E-10	0	No
29	15.3633	0.134428	4.58E-06	7.69E-05	5819142	6.76486	0.002995	2.24E-10	0	No
30	15.3667	0.125056	4.26E-06	8.12E-05	5798664	6.76333	0.00278617	2.09E-10	0	No
31	15.37	0.114005	3.89E-06	8.51E-05	5778257	6.7618	0.00253994	1.91E-10	0	No
32	15.3733	0.107837	3.67E-06	8.87E-05	5758023	6.76027	0.0024025	1.81E-10	0	No
33	15.3767	0.109146	3.72E-06	9.24E-05	5737759	6.75874	0.00243164	1.84E-10	0	No
34	15.38	0.11488	3.91E-06	9.64E-05	5717567	6.75721	0.00255937	1.94E-10	0	No
35	15.3833	0.120613	4.11E-06	0.000100474	5697445	6.75568	0.00268709	2.05E-10	0	No
36	15.3867	0.127415	4.34E-06	0.000104816	5677394	6.75415	0.0028386	2.17E-10	0	No
37	15.39	0.129334	4.41E-06	0.000109224	5657512	6.75263	0.00288133	2.21E-10	0	No
38	15.3933	0.130185	4.44E-06	0.00011366	5637601	6.75109	0.00290027	2.23E-10	0	No
39	15.3967	0.132562	4.52E-06	0.000118177	5617760	6.74956	0.0029532	2.28E-10	0	No
40	15.4	0.138144	4.71E-06	0.000122885	5597989	6.74803	0.00307751	2.39E-10	0	No
41	15.4033	0.142657	4.86E-06	0.000127746	5578287	6.7465	0.00317803	2.47E-10	0	No
42	15.4067	0.148391	5.06E-06	0.000132803	5558751	6.74498	0.00330574	2.58E-10	0	No
43	15.41	0.155955	5.31E-06	0.000138118	5539187	6.74345	0.00347423	2.72E-10	0	No
44	15.4133	0.166725	5.68E-06	0.000143799	5519692	6.74191	0.00371411	2.92E-10	0	No
45	15.4167	0.178409	6.08E-06	0.000149879	5500265	6.74038	0.00397438	3.14E-10	0	No
46	15.42	0.186432	6.35E-06	0.000156232	5480907	6.73885	0.00415307	3.29E-10	0	No
47	15.4233	0.19415	6.62E-06	0.000162848	5461711	6.73733	0.00432495	3.44E-10	0	No
48	15.4267	0.201409	6.86E-06	0.000169712	5442488	6.7358	0.00448664	3.58E-10	0	No

The measurements result in columns that are several thousand cells long, thus all the data have not been included. In addition, the length of the columns makes processing the data in Microsoft Excel cumbersome. In order to simplify the analysis procedure, the important columns were imported into MATLAB using the following code.

```
%=====================================================================
% Thaabit Nacerodien
% MSc Chemical Engineering
% University of Cape Town
% Filename: RawDataCapture.m
% 2013
% Read GPC data from excel - signal height and retention time
%=====================================================================

Ba(:,1) = xlsread('SEC GPC results.xlsx','Ba','A129:A2904');
Ba(:,2) = xlsread('SEC GPC results.xlsx','Ba','B129:B2904');
Ba = sortrows(Ba);
Bb(:,1) = xlsread('SEC GPC results.xlsx','Bb','A129:A3002');
Bb(:,2) = xlsread('SEC GPC results.xlsx','Bb','B129:B3002');
Bb = sortrows(Bb);
Bc(:,1) = xlsread('SEC GPC results.xlsx','Bc','A129:A3004');
Bc(:,2) = xlsread('SEC GPC results.xlsx','Bc','B129:B3004');
Bc = sortrows(Bc);
C(:,1) = xlsread('SEC GPC results.xlsx','C','A129:A3061');
C(:,2) = xlsread('SEC GPC results.xlsx','C','B129:B3061');
C = sortrows(C);
D(:,1) = xlsread('SEC GPC results.xlsx','D','A129:A3032');
D(:,2) = xlsread('SEC GPC results.xlsx','D','B129:B3032');
D = sortrows(D);
C1B3(:,1) = xlsread('SEC GPC results.xlsx','C1B3','A129:A3033');
C1B3(:,2) = xlsread('SEC GPC results.xlsx','C1B3','B129:B3033');
C1B3 = sortrows(C1B3);
CB(:,1) = xlsread('SEC GPC results.xlsx','CB','A129:A3055');
CB(:,2) = xlsread('SEC GPC results.xlsx','CB','B129:B3055');
CB = sortrows(CB);
C3B1a(:,1) = xlsread('SEC GPC results.xlsx','C3B1a','A129:A3041');
C3B1a(:,2) = xlsread('SEC GPC results.xlsx','C3B1a','B129:B3041');
C3B1a = sortrows(C3B1a);
C3B1b(:,1) = xlsread('SEC GPC results.xlsx','C3B1b','A129:A3066');
C3B1b(:,2) = xlsread('SEC GPC results.xlsx','C3B1b','B129:B3066');
C3B1b = sortrows(C3B1b);
C3B1c(:,1) = xlsread('SEC GPC results.xlsx','C3B1c','A129:A2989');
C3B1c(:,2) = xlsread('SEC GPC results.xlsx','C3B1c','B129:B2989');
C3B1c = sortrows(C3B1c);
C1D3(:,1) = xlsread('SEC GPC results.xlsx','C1D3','A129:A2939');
C1D3(:,2) = xlsread('SEC GPC results.xlsx','C1D3','B129:B2939');
C1D3 = sortrows(C1D3);
CDa(:,1) = xlsread('SEC GPC results.xlsx','CDa','A129:A3022');
CDa(:,2) = xlsread('SEC GPC results.xlsx','CDa','B129:B3022');
CDa = sortrows(CDa);
CDb(:,1) = xlsread('SEC GPC results.xlsx','CDb','A129:A2972');
CDb(:,2) = xlsread('SEC GPC results.xlsx','CDb','B129:B2972');
CDb = sortrows(CDb);
CDc(:,1) = xlsread('SEC GPC results.xlsx','CDc','A129:A2907');
CDc(:,2) = xlsread('SEC GPC results.xlsx','CDc','B129:B2907');
CDc = sortrows(CDc);
C3D1(:,1) = xlsread('SEC GPC results.xlsx','C3D1','A129:A2857');
```

```
C3D1(:,2) = xlsread('SEC GPC results.xlsx','C3D1','B129:B2857');
C3D1 = sortrows(C3D1);
B1D3(:,1) = xlsread('SEC GPC results.xlsx','B1D3','A129:A2969');
B1D3(:,2) = xlsread('SEC GPC results.xlsx','B1D3','B129:B2969');
B1D3 = sortrows(B1D3);
BD(:,1) = xlsread('SEC GPC results.xlsx','BD','A129:A2995');
BD(:,2) = xlsread('SEC GPC results.xlsx','BD','B129:B2995');
BD = sortrows(BD);
B3D1(:,1) = xlsread('SEC GPC results.xlsx','B3D1','A129:A3062');
B3D1(:,2) = xlsread('SEC GPC results.xlsx','B3D1','B129:B3062');
B3D1 = sortrows(B3D1);

save('GPCRawData','Ba','Bb','Bc','C','D','C1B3','CB','C3B1a','C3B1b'
,'C3B1c','C1D3','CDa','CDb','CDc','C3D1','B1D3','BD','B3D1');
```

The following MATLAB file computes the smoothed and normalised molecular weight distribution curves from the raw data.

```
%=========================================================================
% Thaabit Nacerodien
% MSc Chemical Engineering
% University of Cape Town
% 2013
% Filename: MWDRecalibrated.m
% Relates retention time to molecular weight using the polystyrene
% calibration curve.
% Thereafter, the molecular weight is recalibrated for polyethylene
% using Mark-Houwink parameters.
% Finally, the signal height is normalised to produce the MWD curves
% and smoothed.
%=========================================================================

load('GPCRawData','Ba','Bb','Bc','C','D','C1B3','CB','C3B1a','C3B1b'
,'C3B1c','C1D3','CDa','CDb','CDc','C3D1','B1D3','BD','B3D1');

MWx = logspace(2,7,1000)';

RT.Ba = Ba(:,1);
RT.Bb = Bb(:,1);
RT.Bc = Bc(:,1);
RT.C = C(:,1);
RT.D = D(:,1);
RT.C1D3 = C1D3(:,1);
RT.CDa = CDa(:,1);
RT.CDb = CDb(:,1);
RT.CDc = CDc(:,1);
RT.C3D1 = C3D1(:,1);
RT.C1B3 = C1B3(:,1);
RT.CB = CB(:,1);
RT.C3B1a = C3B1a(:,1);
RT.C3B1b = C3B1b(:,1);
RT.C3B1c = C3B1c(:,1);
RT.B1D3 = B1D3(:,1);
RT.BD = BD(:,1);
RT.B3D1 = B3D1(:,1);
```

```
S.Ba = Ba(:,2);
S.Bb = Bb(:,2);
S.Bc = Bc(:,2);
S.C = C(:,2);
S.D = D(:,2);
S.C1D3 = C1D3(:,2);
S.CDa = CDa(:,2);
S.CDb = CDb(:,2);
S.CDc = CDc(:,2);
S.C3D1 = C3D1(:,2);
S.C1B3 = C1B3(:,2);
S.CB = CB(:,2);
S.C3B1a = C3B1a(:,2);
S.C3B1b = C3B1b(:,2);
S.C3B1c = C3B1c(:,2);
S.B1D3 = B1D3(:,2);
S.BD = BD(:,2);
S.B3D1 = B3D1(:,2);

MWs.Ba = 10.^CalibrateGPC(Ba(:,1));
MWs.Bb = 10.^CalibrateGPC(Bb(:,1));
MWs.Bc = 10.^CalibrateGPC(Bc(:,1));
MWs.C = 10.^CalibrateGPC(C(:,1));
MWs.D = 10.^CalibrateGPC(D(:,1));
MWs.C1D3 = 10.^CalibrateGPC(C1D3(:,1));
MWs.CDa = 10.^CalibrateGPC(CDa(:,1));
MWs.CDb = 10.^CalibrateGPC(CDb(:,1));
MWs.CDc = 10.^CalibrateGPC(CDc(:,1));
MWs.C3D1 = 10.^CalibrateGPC(C3D1(:,1));
MWs.C1B3 = 10.^CalibrateGPC(C1B3(:,1));
MWs.CB = 10.^CalibrateGPC(CB(:,1));
MWs.C3B1a = 10.^CalibrateGPC(C3B1a(:,1));
MWs.C3B1b = 10.^CalibrateGPC(C3B1b(:,1));
MWs.C3B1c = 10.^CalibrateGPC(C3B1c(:,1));
MWs.B1D3 = 10.^CalibrateGPC(B1D3(:,1));
MWs.BD = 10.^CalibrateGPC(BD(:,1));
MWs.B3D1 = 10.^CalibrateGPC(B3D1(:,1));

MW.Ba = RecalibrateGPC(MWs.Ba);
MW.Bb = RecalibrateGPC(MWs.Bb);
MW.Bc = RecalibrateGPC(MWs.Bc);
MW.C = RecalibrateGPC(MWs.C);
MW.D = RecalibrateGPC(MWs.D);
MW.C1D3 = RecalibrateGPC(MWs.C1D3);
MW.CDa = RecalibrateGPC(MWs.CDa);
MW.CDb = RecalibrateGPC(MWs.CDb);
MW.CDc = RecalibrateGPC(MWs.CDc);
MW.C3D1 = RecalibrateGPC(MWs.C3D1);
MW.C1B3 = RecalibrateGPC(MWs.C1B3);
MW.CB = RecalibrateGPC(MWs.CB);
MW.C3B1a = RecalibrateGPC(MWs.C3B1a);
MW.C3B1b = RecalibrateGPC(MWs.C3B1b);
MW.C3B1c = RecalibrateGPC(MWs.C3B1c);
MW.B1D3 = RecalibrateGPC(MWs.B1D3);
MW.BD = RecalibrateGPC(MWs.BD);
MW.B3D1 = RecalibrateGPC(MWs.B3D1);

P = polyfit(RT.Ba,log(MW.Ba),3);
```

```
sigma.Ba = dMW_dRT(P,RT.Ba);
sigma.Bb = dMW_dRT(P,RT.Bb);
sigma.Bc = dMW_dRT(P,RT.Bc);
sigma.C = dMW_dRT(P,RT.C);
sigma.D = dMW_dRT(P,RT.D);
sigma.B1D3 = dMW_dRT(P,RT.B1D3);
sigma.BD = dMW_dRT(P,RT.BD);
sigma.B3D1 = dMW_dRT(P,RT.B3D1);
sigma.C1D3 = dMW_dRT(P,RT.C1D3);
sigma.CDa = dMW_dRT(P,RT.CDa);
sigma.CDb = dMW_dRT(P,RT.CDb);
sigma.CDc = dMW_dRT(P,RT.CDc);
sigma.C3D1 = dMW_dRT(P,RT.C3D1);
sigma.C1B3 = dMW_dRT(P,RT.C1B3);
sigma.CB = dMW_dRT(P,RT.CB);
sigma.C3B1a = dMW_dRT(P,RT.C3B1a);
sigma.C3B1b = dMW_dRT(P,RT.C3B1b);
sigma.C3B1c = dMW_dRT(P,RT.C3B1c);

W.Ba = abs(S.Ba./(nint(RT.Ba,S.Ba,1)*sigma.Ba));
W.Bb = abs(S.Bb./(nint(RT.Bb,S.Bb,1)*sigma.Bb));
W.Bc = abs(S.Bc./(nint(RT.Bc,S.Bc,1)*sigma.Bc));
W.B1D3 = abs(S.B1D3./(nint(RT.B1D3,S.B1D3,1)*sigma.B1D3));
W.BD = abs(S.BD./(nint(RT.BD,S.BD,1)*sigma.BD));
W.B3D1 = abs(S.B3D1./(nint(RT.B3D1,S.B3D1,1)*sigma.B3D1));
W.C = abs(S.C./(nint(RT.C,S.C,1)*sigma.C));
W.D = abs(S.D./(nint(RT.D,S.D,1)*sigma.D));
W.C1D3 = abs(S.C1D3./(nint(RT.C1D3,S.C1D3,1)*sigma.C1D3));
W.CDa = abs(S.CDa./(nint(RT.CDa,S.CDa,1)*sigma.CDa));
W.CDb = abs(S.CDb./(nint(RT.CDb,S.CDb,1)*sigma.CDb));
W.CDc = abs(S.CDc./(nint(RT.CDc,S.CDc,1)*sigma.CDc));
W.C3D1 = abs(S.C3D1./(nint(RT.C3D1,S.C3D1,1)*sigma.C3D1));
W.C1B3 = abs(S.C1B3./(nint(RT.C1B3,S.C1B3,1)*sigma.C1B3));
W.CB = abs(S.CB./(nint(RT.CB,S.CB,1)*sigma.CB));
W.C3B1a = abs(S.C3B1a./(nint(RT.C3B1a,S.C3B1a,1)*sigma.C3B1a));
W.C3B1b = abs(S.C3B1b./(nint(RT.C3B1b,S.C3B1b,1)*sigma.C3B1b));
W.C3B1c = abs(S.C3B1c./(nint(RT.C3B1c,S.C3B1c,1)*sigma.C3B1c));

SW.Ba = Process(MW.Ba,W.Ba,MWx);
SW.Bb = Process(MW.Bb,W.Bb,MWx);
SW.Bc = Process(MW.Bc,W.Bc,MWx);
SW.C = Process(MW.C,W.C,MWx);
SW.D = Process(MW.D,W.D,MWx);
SW.C1B3 = Process(MW.C1B3,W.C1B3,MWx);
SW.CB = Process(MW.CB,W.CB,MWx);
SW.C3B1a = Process(MW.C3B1a,W.C3B1a,MWx);
SW.C3B1b = Process(MW.C3B1b,W.C3B1b,MWx);
SW.C3B1c = Process(MW.C3B1c,W.C3B1c,MWx);
SW.B1D3 = Process(MW.B1D3,W.B1D3,MWx);
SW.BD = Process(MW.BD,W.BD,MWx);
SW.B3D1 = Process(MW.B3D1,W.B3D1,MWx);
SW.C1D3 = Process(MW.C1D3,W.C1D3,MWx);
SW.CDa = Process(MW.CDa,W.CDa,MWx);
SW.CDb = Process(MW.CDb,W.CDb,MWx);
SW.CDc = Process(MW.CDc,W.CDc,MWx);
SW.C3D1 = Process(MW.C3D1,W.C3D1,MWx);

save('GPCRecalibrated','MW','W','SW','MWx');
```

The function file to convert the retention time to polystyrene molecular weight using a calibration curve is given below.

```
%=====================================================================
% Thaabit Nacerodien
% MSc Chemical Engineering
% University of Cape Town
% 2013
% Filename: CalibrateGPC.m
% Convert retention time to molecular weight of polystyrene
%=====================================================================

function logMW = CalibrateGPC(RT)

x = RT;
a = 13.0961;
b = -0.322665;
c = -0.00779011;
d = 0.000155017;

% a = 13.1088;
% b = -0.3302;
% c = -0.0079;
% d = 1.6088e-4;

logMW = a + b*x + c*x.^2 + d*x.^3;
```

The function file to recalibrate the molecular weight for polyethylene is given below.

```
%=====================================================================
% Thaabit Nacerodien
% MSc Chemical Engineering
% University of Cape Town
% 2013
% Filename: RecalibrateGPC.m
% Molecular weight correction for polyehtylene using Mark-Houwink
% parameters
%=====================================================================

function MW = RecalibrateGPC(MWs)
Ks = 12.1e-5;
as = 0.707;

Ke = 40.6e-5;
ae = 0.725;
%
% etas = Ks*MWs.^(as);
% HV = logMWs + log(etas);

MW = (Ks/Ke)*MWs.^((as+1)/(ae+1));
```

The following function file is used to compute the derivative of the polyethylene calibration curve for use in normalising the molecular weight distribution curves.

```
%=====================================================================
% Thaabit Nacerodien
% MSc Chemical Engineering
% University of Cape Town
% 2013
% Filename: dMW_dRT.m
% Description: compute the derivative of the calibration curve with
respect
% to retention time
%=====================================================================

function sigma = dMW_dRT(P,RT)

sigma = 3*P(1)*RT.^2 + 2*P(2)*RT + P(3);
```

The following function file is used for smoothing the data.

```
%=====================================================================
% Thaabit Nacerodien
% MSc Chemical Engineering
% University of Cape Town
% Filename: Process.m
% 2013
% Smooth data using LOESS and zeroing of negative values
%=====================================================================

function out = Process(x,y,X)

out = interp1(x,y,X);
out(isnan(out)) = 0;
out = smooth(out,0.1,'loess');
out(out < 0) = 0;
if(out(end)>0)
   for i = 1:length(X)
    while(out(end+1-i)~= 0)
        out(end+1-i) = 0;
        i = i + 1;
    end
    break;
end
end
```

The following code produces average curves based on repeat measurements. The curves of the blends are predicted using the curves of the pure grades and are compared against their experimentally determined counterparts in figures. The moments of all the curves are also determined such that the average properties can be displayed in a table for comparison.

```
%=========================================================================
% Thaabit Nacerodien
% MSc Chemical Engineering
% University of Cape Town
% 2013
% Filename: SmoothValidation.m
% Compare experimental curves to model predicted curves
%=========================================================================

clc; clear; close all;

load('GPCRecalibrated','MW','W','SW','MWx');

%=========================================================================
% Averaging repeat measurements
%=========================================================================
%B
M_Bas = nint(log(MWx),SW.Ba,1);
M_Bbs = nint(log(MWx),SW.Bb,1);
M_Bcs = nint(log(MWx),SW.Bc,1);

M_Bs = M_Bas + M_Bbs + M_Bcs;
w_Bs = [M_Bas M_Bbs M_Bcs]/M_Bs;

SW.B = sum(repmat(w_Bs,length(MWx),1).*[SW.Ba SW.Bb SW.Bc],2);
m_Bs = nint(log(MWx),SW.B,1);

%C
m_C = nint(log(MWx),SW.C,1);

%C3B1
M_as = nint(log(MWx),SW.C3B1a,1);
M_bs = nint(log(MWx),SW.C3B1b,1);
M_cs = nint(log(MWx),SW.C3B1c,1);

M_C3B1s = M_as + M_bs + M_cs;
w_C3B1s = [M_as M_bs M_cs]/M_C3B1s;

SW.C3B1 = sum(repmat(w_C3B1s,length(MWx),1).*[SW.C3B1a SW.C3B1b
SW.C3B1c],2);

%D
m_Ds = nint(log(MWx),SW.D,1);

%CD
M_CDas = nint(log(MWx),SW.CDa,1);
M_CDbs = nint(log(MWx),SW.CDb,1);
M_CDcs = nint(log(MWx),SW.CDc,1);

M_CDs = M_CDas + M_CDbs + M_CDcs;
w_CDs = [M_CDas M_CDbs M_CDcs]/M_CDs;

SW.CD = sum(repmat(w_CDs,length(MWx),1).*[SW.CDa SW.CDb SW.CDc],2);
```

```
%=========================================================================
%Calculate Average Properties of smoothed curves
%=========================================================================
e.B = Moments(MWx,SW.B);
e.Ba = Moments(MWx,SW.Ba);
e.Bb = Moments(MWx,SW.Bb);
e.Bc = Moments(MWx,SW.Bc);
e.C = Moments(MWx,SW.C);
e.D = Moments(MWx,SW.D);

e.B1D3 = Moments(MWx,SW.B1D3);
e.BD = Moments(MWx,SW.BD);
e.B3D1 = Moments(MWx,SW.B3D1);

e.C1D3 = Moments(MWx,SW.C1D3);
e.CD = Moments(MWx,SW.CD);
e.CDa = Moments(MWx,SW.CDa);
e.CDb = Moments(MWx,SW.CDb);
e.CDc = Moments(MWx,SW.CDc);
e.C3D1 = Moments(MWx,SW.C3D1);

e.C1B3 = Moments(MWx,SW.C1B3);
e.CB = Moments(MWx,SW.CB);
e.C3B1 = Moments(MWx,SW.C3B1);
e.C3B1a = Moments(MWx,SW.C3B1a);
e.C3B1b = Moments(MWx,SW.C3B1b);
e.C3B1c = Moments(MWx,SW.C3B1c);

%=========================================================================
%Calculate average properties based on blending rule
%=========================================================================
m.B1D3 = Blend(e.B,e.D,0.25);
m.BD = Blend(e.B,e.D,0.5);
m.B3D1 = Blend(e.B,e.D,0.75);

m.C1D3 = Blend(e.C,e.D,0.25);
m.CD = Blend(e.C,e.D,0.5);
m.C3D1 = Blend(e.C,e.D,0.75);

m.C1B3 = Blend(e.C,e.B,0.25);
m.CB = Blend(e.C,e.B,0.5);
m.C3B1 = Blend(e.C,e.B,0.75);

%=========================================================================
%Computing Distributions based on blending rule
%=========================================================================

w_1 = [0.25 0.75];
w_2 = [0.5 0.5];
w_3 = [0.75 0.25];

m.B1D3.W = sum(repmat(w_1,length(MWx),1).*[SW.B SW.D],2);
m.B1D3.r2 = rSquare(SW.B1D3,m.B1D3.W);
m.BD.W = sum(repmat(w_2,length(MWx),1).*[SW.B SW.D],2);
m.BD.r2 = rSquare(SW.BD,m.BD.W);
m.B3D1.W = sum(repmat(w_3,length(MWx),1).*[SW.B SW.D],2);
m.B3D1.r2 = rSquare(SW.B3D1,m.B3D1.W);
```

```
m.C1D3.W = sum(repmat(w_1,length(MWx),1).*[SW.C SW.D],2);
m.C1D3.r2 = rSquare(SW.C1D3,m.C1D3.W);
m.CD.W = sum(repmat(w_2,length(MWx),1).*[SW.C SW.D],2);
m.CD.r2 = rSquare(SW.CD,m.CD.W);
m.C3D1.W = sum(repmat(w_3,length(MWx),1).*[SW.C SW.D],2);
m.C3D1.r2 = rSquare(SW.C3D1,m.C3D1.W);

m.C1B3.W = sum(repmat(w_1,length(MWx),1).*[SW.C SW.B],2);
m.C1B3.r2 = rSquare(SW.C1B3,m.C1B3.W);
m.CB.W = sum(repmat(w_2,length(MWx),1).*[SW.C SW.B],2);
m.CB.r2 = rSquare(SW.CB,m.CB.W);
m.C3B1.W = sum(repmat(w_3,length(MWx),1).*[SW.C SW.B],2);
m.C3B1.r2 = rSquare(SW.C3B1,m.C3B1.W);

%=========================================================================
% %Blended Property Tables
%=========================================================================
fprintf('Blend \t\tMn_m \t\tMn_e \t\tPDI_m \tPDI_e \tMw_m
\t\tMw_e\n');
fprintf('B \t\t\t%2.2s \t%2.2s \t%3.2f \t%3.2f \t%2.2s
\t%2.2s\n',e.B.Mn,e.B.Mn,e.B.PDI,e.B.PDI,e.B.Mw,e.B.Mw);
fprintf('Ba \t\t\t%2.2s \t%2.2s \t%3.2f \t%3.2f \t%2.2s
\t%2.2s\n',e.B.Mn,e.Ba.Mn,e.B.PDI,e.Ba.PDI,e.B.Mw,e.Ba.Mw);
fprintf('Bb \t\t\t%2.2s \t%2.2s \t%3.2f \t%3.2f \t%2.2s
\t%2.2s\n',e.B.Mn,e.Bb.Mn,e.B.PDI,e.Bb.PDI,e.B.Mw,e.Bb.Mw);
fprintf('Bc \t\t\t%2.2s \t%2.2s \t%3.2f \t%3.2f \t%2.2s
\t%2.2s\n',e.B.Mn,e.Bc.Mn,e.B.PDI,e.Bc.PDI,e.B.Mw,e.Bc.Mw);

fprintf('B3D1 \t\t%2.2s \t%2.2s \t%3.2f \t%3.2f \t%2.2s
\t%2.2s\n',m.B3D1.Mn,e.B3D1.Mn,m.B3D1.PDI,e.B3D1.PDI,m.B3D1.Mw,e.B3D
1.Mw);
fprintf('BD \t\t\t%2.2s \t%2.2s \t%3.2f \t%3.2f \t%2.2s
\t%2.2s\n',m.BD.Mn,e.BD.Mn,m.BD.PDI,e.BD.PDI,m.BD.Mw,e.BD.Mw);
fprintf('B1D3 \t\t%2.2s \t%2.2s \t%3.2f \t%3.2f \t%2.2s
\t%2.2s\n',m.B1D3.Mn,e.B1D3.Mn,m.B1D3.PDI,e.B1D3.PDI,m.B1D3.Mw,e.B1D
3.Mw);
fprintf('D \t\t\t%2.2s \t%2.2s \t%3.2f \t%3.2f \t%2.2s
\t%2.2s\n',e.D.Mn,e.D.Mn,e.D.PDI,e.D.PDI,e.D.Mw,e.D.Mw);

fprintf('Blend \t\tMn_m \t\tMn_e \t\tPDI_m \tPDI_e \tMw_m
\t\tMw_e\n');
fprintf('C \t\t\t%2.2s \t%2.2s \t%3.2f \t%3.2f \t%2.2s
\t%2.2s\n',e.C.Mn,e.C.Mn,e.C.PDI,e.C.PDI,e.C.Mw,e.C.Mw);
fprintf('C3D1 \t\t%2.2s \t%2.2s \t%3.2f \t%3.2f \t%2.2s
\t%2.2s\n',m.C3D1.Mn,e.C3D1.Mn,m.C3D1.PDI,e.C3D1.PDI,m.C3D1.Mw,e.C3D
1.Mw);
fprintf('CD \t\t\t%2.2s \t%2.2s \t%3.2f \t%3.2f \t%2.2s
\t%2.2s\n',m.CD.Mn,e.CD.Mn,m.CD.PDI,e.CD.PDI,m.CD.Mw,e.CD.Mw);
fprintf('C1D3 \t\t%2.2s \t%2.2s \t%3.2f \t%3.2f \t%2.2s
\t%2.2s\n',m.C1D3.Mn,e.C1D3.Mn,m.C1D3.PDI,e.C1D3.PDI,m.C1D3.Mw,e.C1D
3.Mw);
fprintf('D \t\t\t%2.2s \t%2.2s \t%3.2f \t%3.2f \t%2.2s
\t%2.2s\n',e.D.Mn,e.D.Mn,e.D.PDI,e.D.PDI,e.D.Mw,e.D.Mw);

fprintf('Blend \t\tMn_m \t\tMn_e \t\tPDI_m \tPDI_e \tMw_m
\t\tMw_e\n');
```

```
fprintf('C \t\t\t%2.2s \t%2.2s \t%3.2f \t%3.2f \t%2.2s
\t%2.2s\n',e.C.Mn,e.C.Mn,e.C.PDI,e.C.PDI,e.C.Mw,e.C.Mw);
fprintf('C3B1 \t\t%2.2s \t%2.2s \t%3.2f \t%3.2f \t%2.2s
\t%2.2s\n',m.C3B1.Mn,e.C3B1.Mn,m.C3B1.PDI,e.C3B1.PDI,m.C3B1.Mw,e.C3B
1.Mw);
fprintf('CB \t\t\t%2.2s \t%2.2s \t%3.2f \t%3.2f \t%2.2s
\t%2.2s\n',m.CB.Mn,e.CB.Mn,m.CB.PDI,e.CB.PDI,m.CB.Mw,e.CB.Mw);
fprintf('C1B3 \t\t%2.2s \t%2.2s \t%3.2f \t%3.2f \t%2.2s
\t%2.2s\n',m.C1B3.Mn,e.C1B3.Mn,m.C1B3.PDI,e.C1B3.PDI,m.C1B3.Mw,e.C1B
3.Mw);
fprintf('B \t\t\t%2.2s \t%2.2s \t%3.2f \t%3.2f \t%2.2s
\t%2.2s\n',e.B.Mn,e.B.Mn,e.B.PDI,e.B.PDI,e.B.Mw,e.B.Mw);

%=========================================================================
%Precision Test
%=========================================================================
CD.X.Mn = mean([e.CDa.Mn e.CDb.Mn e.CDc.Mn]);
CD.X.Mw = mean([e.CDa.Mw e.CDb.Mw e.CDc.Mw]);
CD.X.PDI = mean([e.CDa.PDI e.CDb.PDI e.CDc.PDI]);
CD.X.mu0 = mean([e.CDa.mu0 e.CDb.mu0 e.CDc.mu0]);
CD.X.mu1 = mean([e.CDa.mu1 e.CDb.mu1 e.CDc.mu1]);
CD.X.mu2 = mean([e.CDa.mu2 e.CDb.mu2 e.CDc.mu2]);
CD.sd.Mn = std([e.CDa.Mn e.CDb.Mn e.CDc.Mn]);
CD.sd.Mw = std([e.CDa.Mw e.CDb.Mw e.CDc.Mw]);
CD.sd.PDI = std([e.CDa.PDI e.CDb.PDI e.CDc.PDI]);
CD.sd.mu0 = std([e.CDa.mu0 e.CDb.mu0 e.CDc.mu0]);
CD.sd.mu1 = std([e.CDa.mu1 e.CDb.mu1 e.CDc.mu1]);
CD.sd.mu2 = std([e.CDa.mu2 e.CDb.mu2 e.CDc.mu2]);
CD.rsd.Mn = 100*CD.sd.Mn/CD.X.Mn;
CD.rsd.Mw = 100*CD.sd.Mw/CD.X.Mw;
CD.rsd.PDI = 100*CD.sd.PDI/CD.X.PDI;
CD.rsd.mu0 = 100*CD.sd.mu0/CD.X.mu0;
CD.rsd.mu1 = 100*CD.sd.mu1/CD.X.mu1;
CD.rsd.mu2 = 100*CD.sd.mu2/CD.X.mu2;

B.X.Mn = mean([e.Ba.Mn e.Bb.Mn e.Bc.Mn]);
B.X.Mw = mean([e.Ba.Mw e.Bb.Mw e.Bc.Mw]);
B.X.PDI = mean([e.Ba.PDI e.Bb.PDI e.Bc.PDI]);
B.X.mu0 = mean([e.Ba.mu0 e.Bb.mu0 e.Bc.mu0]);
B.X.mu1 = mean([e.Ba.mu1 e.Bb.mu1 e.Bc.mu1]);
B.X.mu2 = mean([e.Ba.mu2 e.Bb.mu2 e.Bc.mu2]);
B.sd.Mn = std([e.Ba.Mn e.Bb.Mn e.Bc.Mn]);
B.sd.Mw = std([e.Ba.Mw e.Bb.Mw e.Bc.Mw]);
B.sd.PDI = std([e.Ba.PDI e.Bb.PDI e.Bc.PDI]);
B.sd.mu0 = std([e.Ba.mu0 e.Bb.mu0 e.Bc.mu0]);
B.sd.mu1 = std([e.Ba.mu1 e.Bb.mu1 e.Bc.mu1]);
B.sd.mu2 = std([e.Ba.mu2 e.Bb.mu2 e.Bc.mu2]);
B.rsd.Mn = 100*B.sd.Mn/B.X.Mn;
B.rsd.Mw = 100*B.sd.Mw/B.X.Mw;
B.rsd.PDI = 100*B.sd.PDI/B.X.PDI;
B.rsd.mu0 = 100*B.sd.mu0/B.X.mu0;
B.rsd.mu1 = 100*B.sd.mu1/B.X.mu1;
B.rsd.mu2 = 100*B.sd.mu2/B.X.mu2;

C3B1.X.Mn = mean([e.C3B1a.Mn e.C3B1b.Mn e.C3B1c.Mn]);
C3B1.X.Mw = mean([e.C3B1a.Mw e.C3B1b.Mw e.C3B1c.Mw]);
C3B1.X.PDI = mean([e.C3B1a.PDI e.C3B1b.PDI e.C3B1c.PDI]);
C3B1.X.mu0 = mean([e.C3B1a.mu0 e.C3B1b.mu0 e.C3B1c.mu0]);
```

```
C3B1.X.mu1 = mean([e.C3B1a.mu1 e.C3B1b.mu1 e.C3B1c.mu1]);
C3B1.X.mu2 = mean([e.C3B1a.mu2 e.C3B1b.mu2 e.C3B1c.mu2]);
C3B1.sd.Mn = std([e.C3B1a.Mn e.C3B1b.Mn e.C3B1c.Mn]);
C3B1.sd.Mw = std([e.C3B1a.Mw e.C3B1b.Mw e.C3B1c.Mw]);
C3B1.sd.PDI = std([e.C3B1a.PDI e.C3B1b.PDI e.C3B1c.PDI]);
C3B1.sd.mu0 = std([e.C3B1a.mu0 e.C3B1b.mu0 e.C3B1c.mu0]);
C3B1.sd.mu1 = std([e.C3B1a.mu1 e.C3B1b.mu1 e.C3B1c.mu1]);
C3B1.sd.mu2 = std([e.C3B1a.mu2 e.C3B1b.mu2 e.C3B1c.mu2]);
C3B1.rsd.Mn = 100*C3B1.sd.Mn/C3B1.X.Mn;
C3B1.rsd.Mw = 100*C3B1.sd.Mw/C3B1.X.Mw;
C3B1.rsd.PDI = 100*C3B1.sd.PDI/C3B1.X.PDI;
C3B1.rsd.mu0 = 100*C3B1.sd.mu0/C3B1.X.mu0;
C3B1.rsd.mu1 = 100*C3B1.sd.mu1/C3B1.X.mu1;
C3B1.rsd.mu2 = 100*C3B1.sd.mu2/C3B1.X.mu2;

gCD.X.Mn = mean([e.C.Mn e.D.Mn]);
gCD.X.Mw = mean([e.C.Mw e.D.Mw]);
gCD.X.PDI = mean([e.C.PDI e.D.PDI]);
gCD.X.mu0 = mean([e.C.mu0 e.D.mu0]);
gCD.X.mu1 = mean([e.C.mu1 e.D.mu1]);
gCD.X.mu2 = mean([e.C.mu2 e.D.mu2]);
gCD.sd.Mn = std([e.C.Mn e.D.Mn]);
gCD.sd.Mw = std([e.C.Mw e.D.Mw]);
gCD.sd.PDI = std([e.C.PDI e.D.PDI]);
gCD.sd.mu0 = std([e.C.mu0 e.D.mu0]);
gCD.sd.mu1 = std([e.C.mu1 e.D.mu1]);
gCD.sd.mu2 = std([e.C.mu2 e.D.mu2]);
gCD.rsd.Mn = 100*gCD.sd.Mn/gCD.X.Mn;
gCD.rsd.Mw = 100*gCD.sd.Mw/gCD.X.Mw;
gCD.rsd.PDI = 100*gCD.sd.PDI/gCD.X.PDI;
gCD.rsd.mu0 = 100*gCD.sd.mu0/gCD.X.mu0;
gCD.rsd.mu1 = 100*gCD.sd.mu1/gCD.X.mu1;
gCD.rsd.mu2 = 100*gCD.sd.mu2/gCD.X.mu2;

gCB.X.Mn = mean([e.C.Mn e.B.Mn]);
gCB.X.Mw = mean([e.C.Mw e.B.Mw]);
gCB.X.PDI = mean([e.C.PDI e.B.PDI]);
gCB.X.mu0 = mean([e.C.mu0 e.B.mu0]);
gCB.X.mu1 = mean([e.C.mu1 e.B.mu1]);
gCB.X.mu2 = mean([e.C.mu2 e.B.mu2]);
gCB.sd.Mn = std([e.C.Mn e.B.Mn]);
gCB.sd.Mw = std([e.C.Mw e.B.Mw]);
gCB.sd.PDI = std([e.C.PDI e.B.PDI]);
gCB.sd.mu0 = std([e.C.mu0 e.B.mu0]);
gCB.sd.mu1 = std([e.C.mu1 e.B.mu1]);
gCB.sd.mu2 = std([e.C.mu2 e.B.mu2]);
gCB.rsd.Mn = 100*gCB.sd.Mn/gCB.X.Mn;
gCB.rsd.Mw = 100*gCB.sd.Mw/gCB.X.Mw;
gCB.rsd.PDI = 100*gCB.sd.PDI/gCB.X.PDI;
gCB.rsd.mu0 = 100*gCB.sd.mu0/gCB.X.mu0;
gCB.rsd.mu1 = 100*gCB.sd.mu1/gCB.X.mu1;
gCB.rsd.mu2 = 100*gCB.sd.mu2/gCB.X.mu2;

gCBD.X.Mn = mean([e.C.Mn e.B.Mn e.D.Mn]);
gCBD.X.Mw = mean([e.C.Mw e.B.Mw e.D.Mw]);
gCBD.X.PDI = mean([e.C.PDI e.B.PDI e.D.PDI]);
gCBD.X.mu0 = mean([e.C.mu0 e.B.mu0 e.D.mu0]);
gCBD.X.mu1 = mean([e.C.mu1 e.B.mu1 e.D.mu1]);
```

```
gCBD.X.mu2 = mean([e.C.mu2 e.B.mu2 e.D.mu2]);
gCBD.sd.Mn = std([e.C.Mn e.B.Mn e.D.Mn]);
gCBD.sd.Mw = std([e.C.Mw e.B.Mw e.D.Mw]);
gCBD.sd.PDI = std([e.C.PDI e.B.PDI e.D.PDI]);
gCBD.sd.mu0 = std([e.C.mu0 e.B.mu0 e.D.mu0]);
gCBD.sd.mu1 = std([e.C.mu1 e.B.mu1 e.D.mu1]);
gCBD.sd.mu2 = std([e.C.mu2 e.B.mu2 e.D.mu2]);
gCBD.rsd.Mn = 100*gCBD.sd.Mn/gCBD.X.Mn;
gCBD.rsd.Mw = 100*gCBD.sd.Mw/gCBD.X.Mw;
gCBD.rsd.PDI = 100*gCBD.sd.PDI/gCBD.X.PDI;
gCBD.rsd.mu0 = 100*gCBD.sd.mu0/gCBD.X.mu0;
gCBD.rsd.mu1 = 100*gCBD.sd.mu1/gCBD.X.mu1;
gCBD.rsd.mu2 = 100*gCBD.sd.mu2/gCBD.X.mu2;

%=========================================================================
%Average Properties Comparative Plots
%=========================================================================

figure(1)
w_e = [1 0.75 0.5 0.25 0];
w_m = [0.75 0.5 0.25];
subplot(3,1,1)
hold on
plot(w_e,[e.B.mu1 e.B3D1.mu1 e.BD.mu1 e.B1D3.mu1 e.D.mu1],'xr');
plot(w_m,[m.B3D1.mu1 m.BD.mu1 m.B1D3.mu1],'o');
xlabel('Blend Fraction - BD');
ylabel('\mu_{ 1}');
legend('Experimental','Model');
xlim([-0.01 1.01]);
hold off

w_e = [1 0.75 0.5 0.25 0];
w_m = [0.75 0.5 0.25];
subplot(3,1,2)
hold on
errorbar(w_e,[e.C.mu1 e.C3D1.mu1 e.CD.mu1 e.C1D3.mu1 e.D.mu1],[0 0
CD.sd.mu1 0 0],[0 0 CD.sd.mu1 0 0],'xr');
plot(w_m,[m.C3D1.mu1 m.CD.mu1 m.C1D3.mu1],'o');
xlabel('Blend Fraction - CD');
ylabel('\mu_{ 1}');
xlim([-0.01 1.01]);
hold off

w_e = [1 0.75 0.5 0.25 0];
w_m = [0.75 0.5 0.25];

subplot(3,1,3)
hold on
errorbar(w_e,[e.C.mu1 e.C3B1.mu1 e.CB.mu1 e.C1B3.mu1 e.B.mu1],[0
C3B1.sd.mu1 0 0 0],[0 C3B1.sd.mu1 0 0 0],'xr');
plot(w_m,[m.C3B1.mu1 m.CB.mu1 m.C1B3.mu1],'o');
xlabel('Blend Fraction - CB');
ylabel('\mu_{ 1}');
xlim([-0.01 1.01]);
hold off
```

```
%=========================================================================
%Model vs Experimental Distribution Curves
%=========================================================================
figure(4)
semilogx(MWx,SW.B1D3,MWx,m.B1D3.W,'--');
xlabel('M_w');
ylabel('W');
legend('e','m');
title('B1D3');
text(2*10^2,0.25,sprintf('R^2: %3.7f',m.B1D3.r2));
ylim([0 0.3]);

figure(5)
semilogx(MWx,SW.BD,MWx,m.BD.W,'--');
xlabel('M_w');
ylabel('W');
legend('e','m');
title('BD');
text(2*10^2,0.25,sprintf('R^2: %3.7f',m.BD.r2));
ylim([0 0.3]);

figure(6)
semilogx(MWx,SW.B3D1,MWx,m.B3D1.W,'--');
xlabel('M_w');
ylabel('W');
legend('e','m');
title('B3D1');
text(2*10^2,0.25,sprintf('R^2: %3.7f',m.B3D1.r2));
ylim([0 0.3]);

figure(7)
semilogx(MWx,SW.C1D3,MWx,m.C1D3.W,'--');
xlabel('M_w');
ylabel('W');
legend('e','m');
title('C1D3');
text(2*10^2,0.25,sprintf('R^2: %3.7f',m.C1D3.r2));
ylim([0 0.3]);

figure(8)
semilogx(MWx,SW.CD,MWx,m.CD.W,'--');
xlabel('M_w');
ylabel('W');
legend('e','m');
title('CD');
text(2*10^2,0.25,sprintf('R^2: %3.7f',m.CD.r2));
ylim([0 0.3]);

figure(9)
semilogx(MWx,SW.C3D1,MWx,m.C3D1.W,'--');
xlabel('M_w');
ylabel('W');
legend('e','m');
title('C3D1');
text(2*10^2,0.25,sprintf('R^2: %3.7f',m.C3D1.r2));
ylim([0 0.3]);
```

```
figure(10)
semilogx(MWx,SW.C1B3,MWx,m.C1B3.W,'--');
xlabel('M_w');
ylabel('W');
legend('e','m');
title('C1B3');
text(2*10^2,0.25,sprintf('R^2: %3.7f',m.C1B3.r2));
ylim([0 0.3]);

figure(11)
semilogx(MWx,SW.CB,MWx,m.CB.W,'--');
xlabel('M_w');
ylabel('W');
legend('e','m');
title('CB');
text(2*10^2,0.25,sprintf('R^2: %3.7f',m.CB.r2));
ylim([0 0.3]);

figure(12)
semilogx(MWx,SW.C3B1,MWx,m.C3B1.W,'--');
xlabel('M_w');
ylabel('W');
legend('e','m');
title('C3B1');
text(2*10^2,0.25,sprintf('R^2: %3.7f',m.C3B1.r2));
ylim([0 0.3]);

%=========================================================================
%Comparison of repeat curves
%=========================================================================
figure(13)
semilogx(MWx,SW.Ba,'c',MWx,SW.Bb,'m',MWx,SW.Bc,'r',MWx,SW.B,'k',MWx,
SW.C);
xlabel('M_w');
ylabel('W');
legend('Ba','Bb','Bc','mean');
% ylim([0 0.3]);

figure(14)
semilogx(MWx,SW.CDa,'c',MWx,SW.CDb,'m',MWx,SW.CDc,'r');%,MWx,SW.CD,'
k');
xlabel('M_w');
ylabel('W');
legend('CDa','CDb','CDc')%,'mean');
% ylim([0 0.3]);

figure(15)
semilogx(MWx,SW.C3B1a,'c',MWx,SW.C3B1b,'m',MWx,SW.C3B1c,'r');%,MWx,S
W.C3B1,'k');
xlabel('M_w');
ylabel('W');
legend('C3B1a','C3B1b','C3B1c');%,'mean');
% ylim([0 0.3]);
```

```
%=========================================================================
%Comparison of blended curves
%=========================================================================
figure(16)
semilogx(MWx,SW.C,'b',MWx,SW.C3B1,'c',MWx,SW.CB,'g',MWx,SW.C1B3,'m',
MWx,SW.B,'r');
xlabel('M_w');
ylabel('W');
legend('C','C3B1','CB','C1B3','B');
% ylim([0 0.3]);

figure(17)
semilogx(MWx,SW.C,'b',MWx,SW.C3D1,'c',MWx,SW.CD,'g',MWx,SW.C1D3,'m',
MWx,SW.D,'r');
xlabel('M_w');
ylabel('W');
legend('C','C3D1','CD','C1D3','D');
% ylim([0 0.3]);

figure(18)
semilogx(MWx,SW.B,'b',MWx,SW.B3D1,'c',MWx,SW.BD,'g',MWx,SW.B1D3,'m',
MWx,SW.D,'r');
xlabel('M_w');
ylabel('W');
legend('B','B3D1','BD','B1D3','D');
% ylim([0 0.3]);

%=========================================================================
%Comparison of pure grades
%=========================================================================
figure(19)
semilogx(MWx,SW.C,'k',MWx,SW.B,'b',MWx,SW.D,'r');
xlabel('M_w');
ylabel('W');
legend('C','B','D');
% ylim([0 0.3]);

figure(20)
semilogx(MWx,SW.C,'k',MWx,SW.D,'r');
xlabel('M_w');
ylabel('W');
legend('C','D');
ylim([0 0.3]);

figure(21)
semilogx(MWx,SW.CDa,MWx,m.CD.W,'--');
xlabel('M_w');
ylabel('W');
legend('e','m');
title('CDa');
m.CDa.r2 = rSquare(SW.CDa,m.CD.W);
text(2*10^2,0.25,sprintf('R^2: %3.7f',m.CDa.r2));
ylim([0 0.3]);

figure(22)
semilogx(MWx,SW.CDb,MWx,m.CD.W,'--');
xlabel('M_w');
ylabel('W');
```

```
legend('e','m');
title('CDb');
m.CDb.r2 = rSquare(SW.CDb,m.CD.W);
text(2*10^2,0.25,sprintf('R^2: %3.7f',m.CDb.r2));
ylim([0 0.3]);

figure(23)
semilogx(MWx,SW.CDc,MWx,m.CD.W,'--');
xlabel('M_w');
ylabel('W');
legend('e','m');
title('CDc');
m.CDc.r2 = rSquare(SW.CDc,m.CD.W);
text(2*10^2,0.25,sprintf('R^2: %3.7f',m.CDc.r2));
ylim([0 0.3]);
```

The function file to numerically integrate the curves was developed is as follows:

```
% Copyright R Rawatlal, School of Chemical Engineering, University
of
% Natal, 2003
%
% Integration of y with respect to x, where y varies with respect
% to x along the n-th dimension. This function returns two arrays;
the
% first is the integral where x has been 'integrated out' - the n-th
% dimension has been eliminated. On the other hand, the second array
is the
% cumulative integral.
% Uses a rather primitve Euler/Riemann-type integration. Can improve
this
% easily.
% Syntax: [Iy, cIy] = nint(x,y,n)

function [Iy, cIy] = nint(x,y,n)

xs = size(x);
if xs(2)>1
    x = x';
end;

sy = size(y);

if length(x)~=sy(n)
    error(['Array y dimension ',num2str(n),' is not of the same
length as x.']);
end;
N = length(sy);

y2 = permute(y,[n 1:(n-1) (n+1):N]); % Put the n-th dim 1st
ss = prod(sy)/sy(n);
y3 = reshape(y2,[sy(n) ss]);
avy = 0.5*(y3(1:end-1,:) + y3(2:end,:));
dx = diff(x);
dxM = repmat(dx,[1 ss]);
cIy1 = [zeros(1,ss);cumsum(avy.*dxM,1)];
```

```
cIy2 = reshape(cIy1,[sy(n) sy(1:n-1) sy(n+1:N)]);

for i = 1:N
    a{i} = 1:sy(i);
end;

cIy = squeeze(permute(cIy2,[2:n 1 n+1:N]));
Iy = squeeze(cIy2(end,a{1:n-1},a{n+1:N}));
if size(Iy,1) == 1
    Iy = Iy';
end;
```

The moments of an experimentally determined curve are determined using the following function file.

```
%=====================================================================
% Thaabit Nacerodien
% MSc Chemical Engineering
% University of Cape Town
% 2013
% Filename: Moments.m
% Computes the zeroth, first and second moments of a distribution
% curve.
% Determines the Mn, Mw and PDI of the curve from the moments
%=====================================================================
function prop = Moments(m,M)

n = m;
Pn = M;

prop.mu0 = nint(n,Pn./n,2);
prop.mu1 = nint(n,Pn,2);
prop.mu2 = nint(n,Pn.*n,2);

prop.Mn = prop.mu1/prop.mu0;
prop.Mw = prop.mu2/prop.mu1;
prop.PDI = prop.Mw/prop.Mn;
```

The moments of a blend are predicted from the pure grade moments using the function file given below.

```
%=====================================================================
% Thaabit Nacerodien
% MSc Chemical Engineering
% University of Cape Town
% 2013
% Filename: Blend.m
% Function file to determine the moments of a blend of two grades.
% Thereafter, the corresponding Mn, Mw and PDI values of the blend
% are
% determined
%=====================================================================

function out = Blend(X,Y,w)

out.mu0 = w*X.mu0 + (1-w)*Y.mu0;
out.mu1 = w*X.mu1 + (1-w)*Y.mu1;
out.mu2 = w*X.mu2 + (1-w)*Y.mu2;

out.Mn = out.mu1/out.mu0;
out.Mw = out.mu2/out.mu1;
out.PDI = out.Mw/out.Mn;
```

The goodness of fit of the predicted curves are determined using the following code.

```
%=====================================================================
% Thaabit Nacerodien
% MSc Chemical Engineering
% University of Cape Town
% 2013
% Filename: rSquare.m
% Calculate the Coeffient of Determination of the predicted curve
%=====================================================================

function out = rSquare(W,Wm)

sst = norm(W - mean(W))^2;
sse = norm(W - Wm)^2;
out = 1 -sse/sst;
```

B. Blending model comparison with experimental data

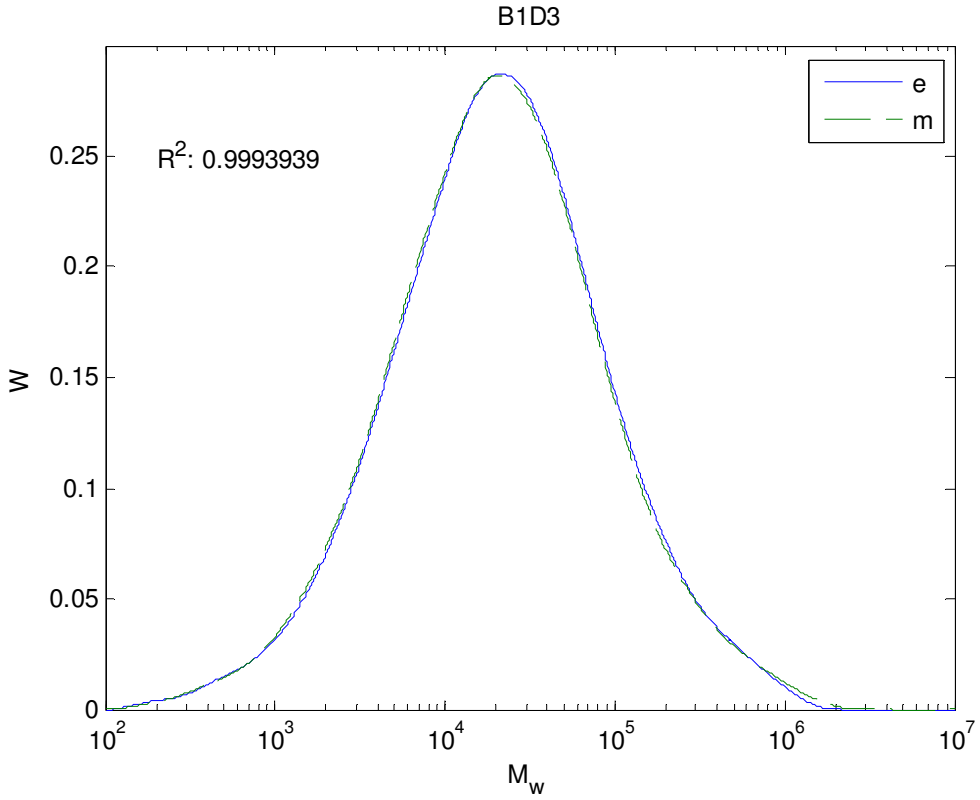

Figure B.1 – Comparison of predicted and experimental MWD for blend B1D3

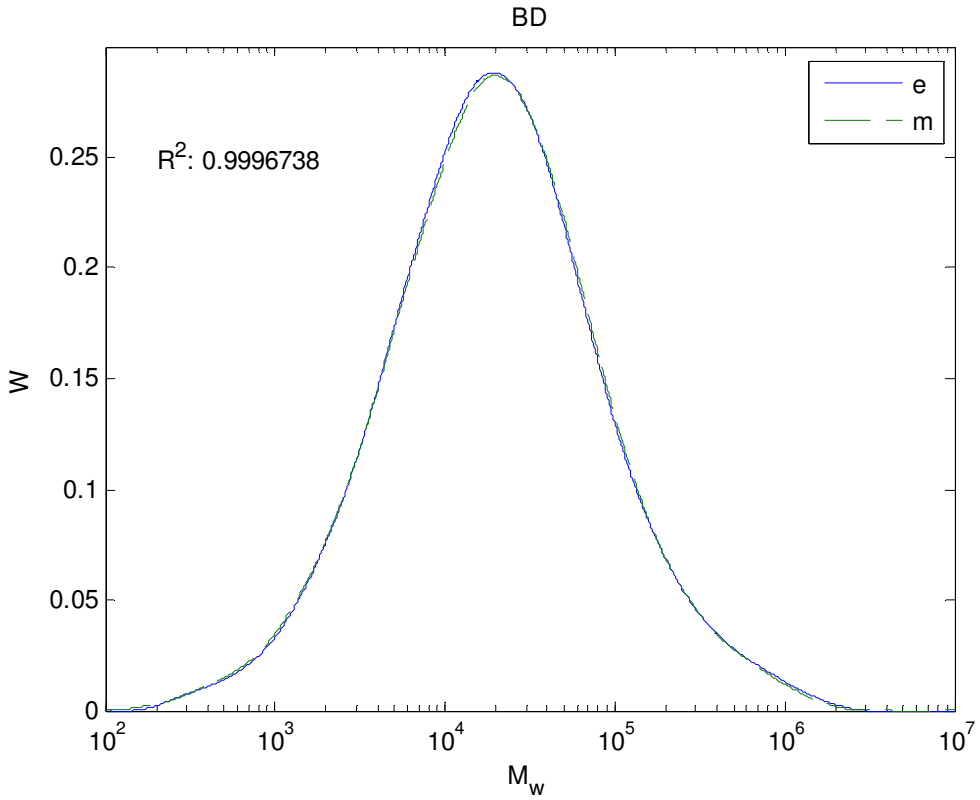

Figure B.2 – Comparison of predicted and experimental MWD for blend BD

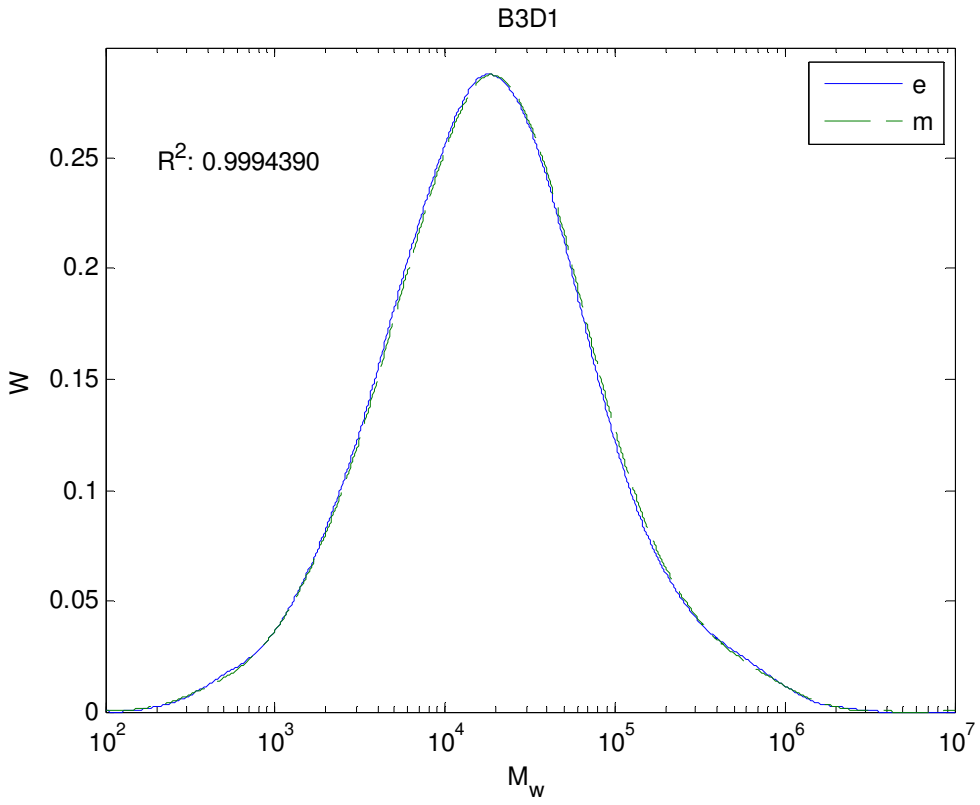

Figure B.3 – Comparison of predicted and experimental MWD for blend B3D1

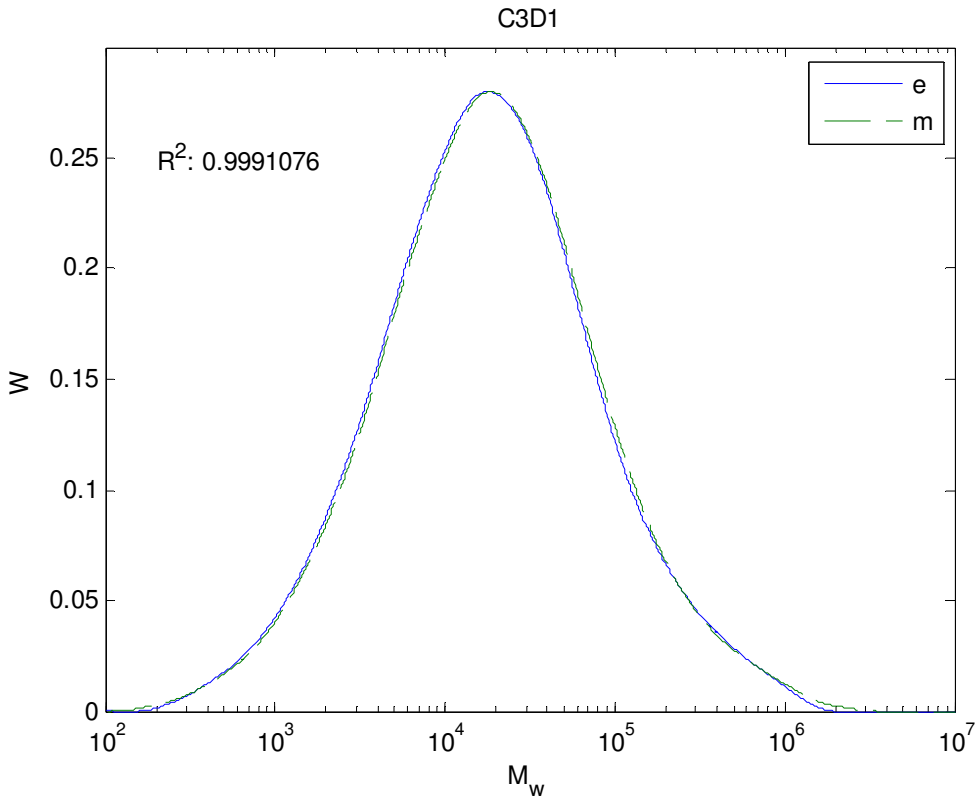

Figure B.4 - Comparison of predicted and experimental MWD for blend C3D1

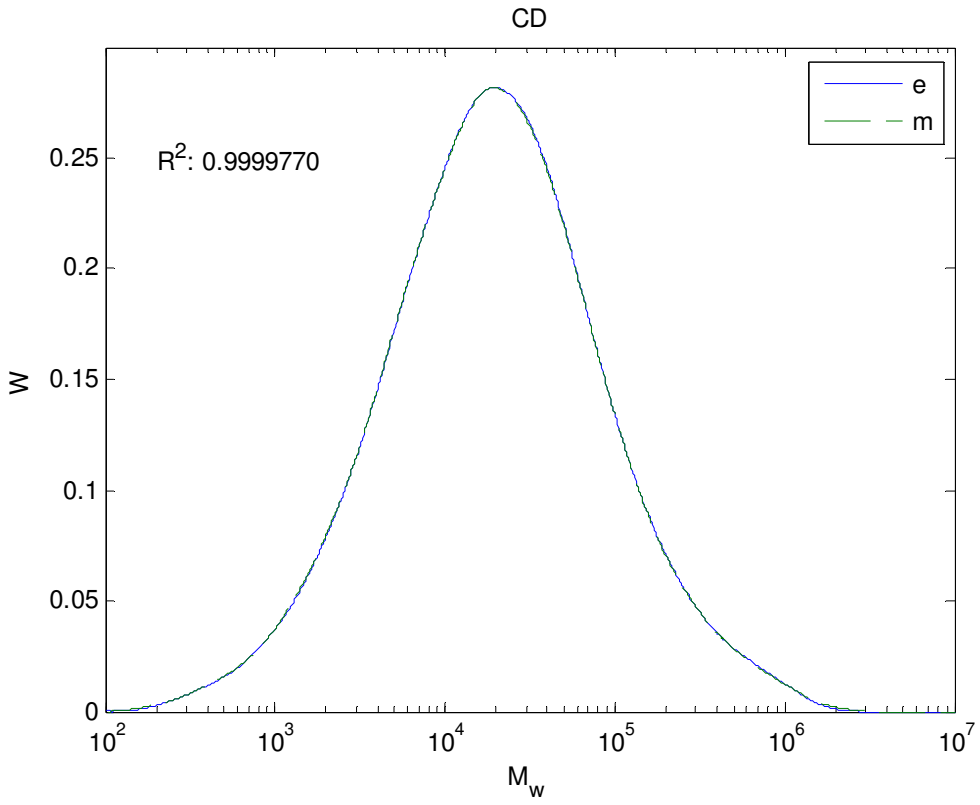

Figure B.5 – Comparison of predicted and experimental MWD for blend CD

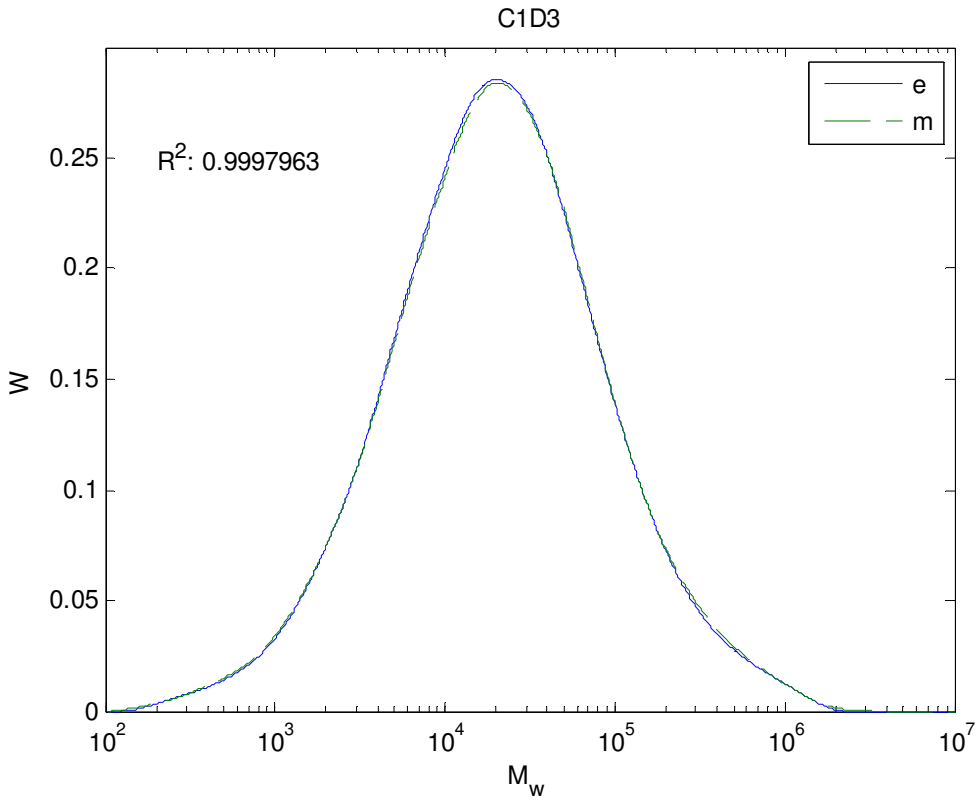

Figure B.6 – Comparison of predicted and experimental MWD for blend C1D3

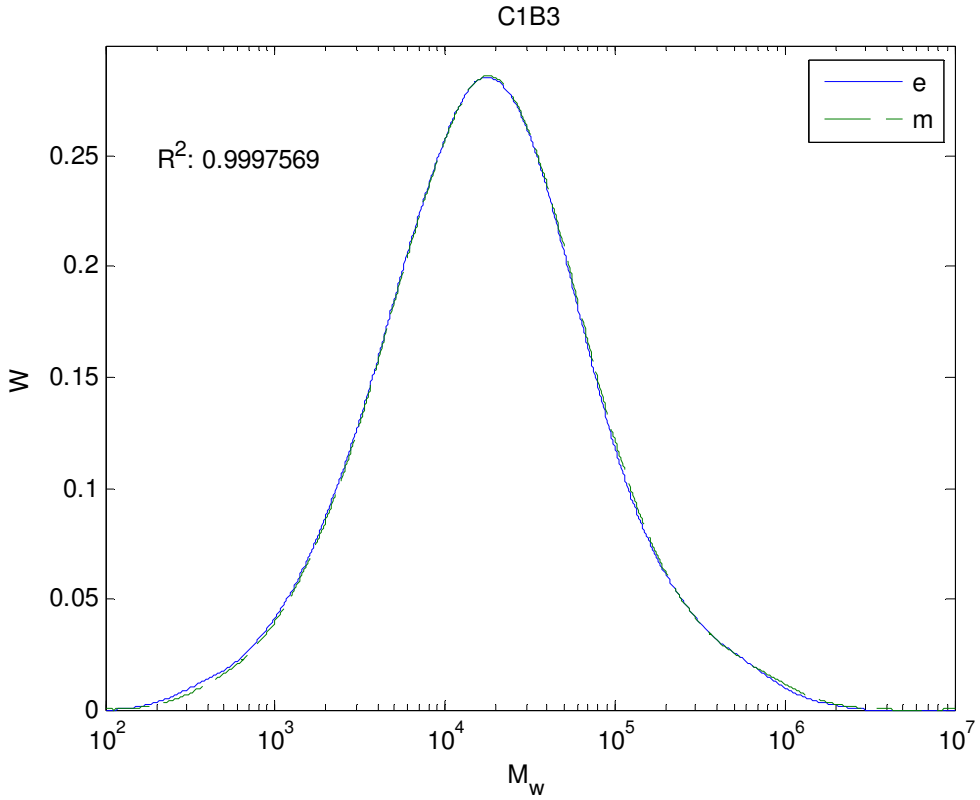

Figure B.7 - Comparison of predicted and experimental MWD for blend C1B3

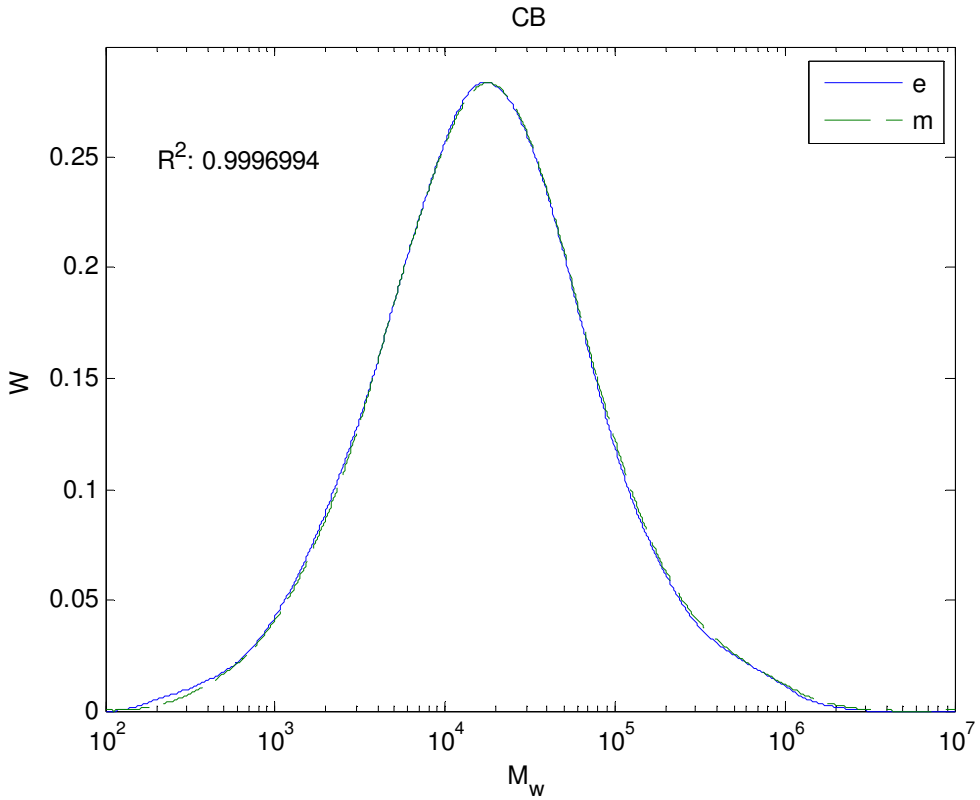

Figure B.8 - Comparison of predicted and experimental MWD for blend CB

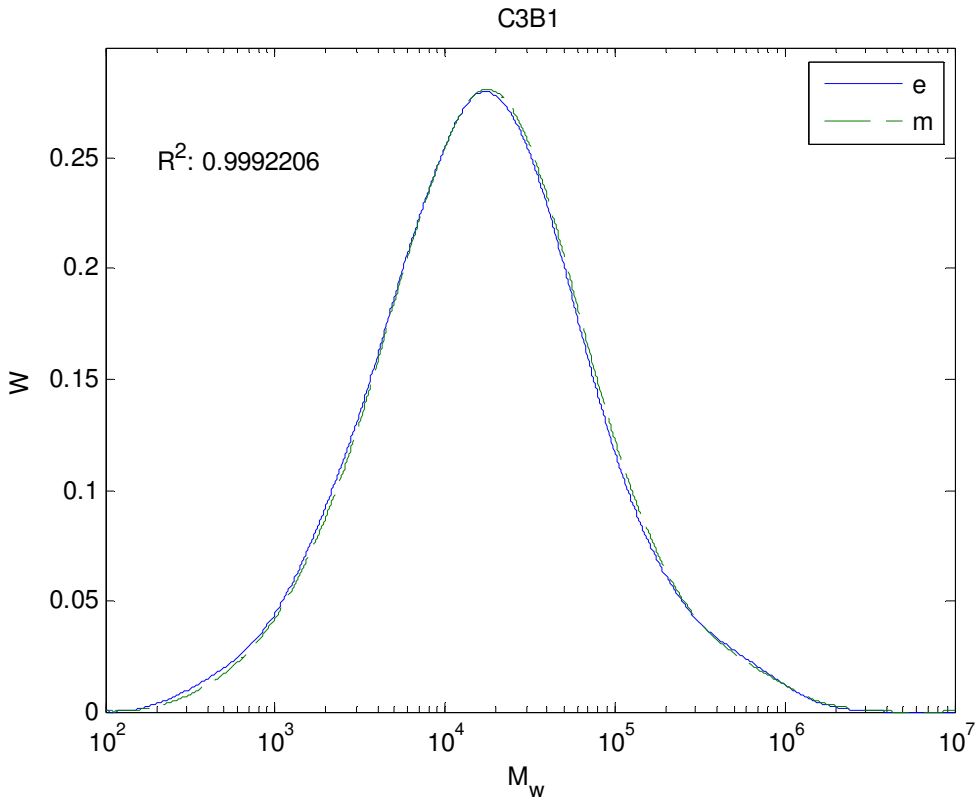

Figure B.9 - Comparison of predicted and experimental MWD for blend C3B1

C. Process model sensitivity analysis

In the following section, results are presented showing the influence of each component on the precipitate properties, namely Mn and PDI. The process model has been run several times to obtain these results. Each time, one of the feed rates of the components has been varied by a particular amount relative to a base case value.

The influence of co-catalyst, monomer, commoner, hydrogen, diluent and catalyst were all varied individually, relative to the base case. Each component is varied over a range that encompasses a 10% deviation from the base case value in both positive and negative directions.

The figures showing each component's influence are shown below. The purpose of including them is to illustrate the non-linearity of the process model when relating the feed rates to the polymer properties.

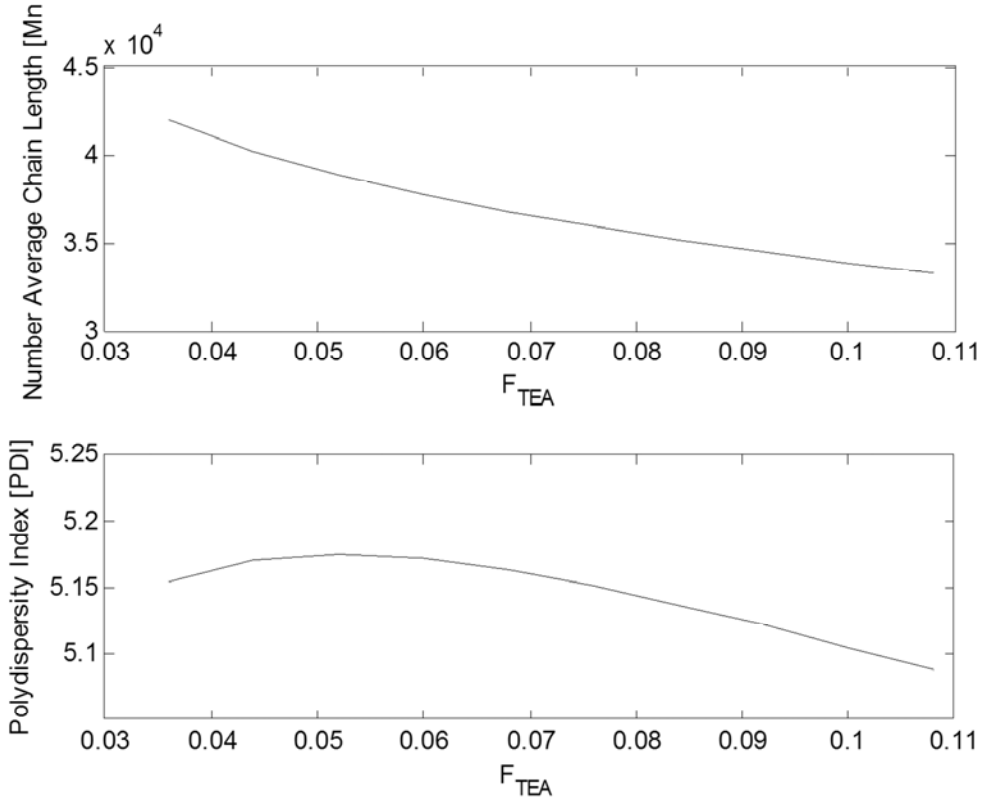

Figure C.1 – Influence of TEA flowrate on polymer properties

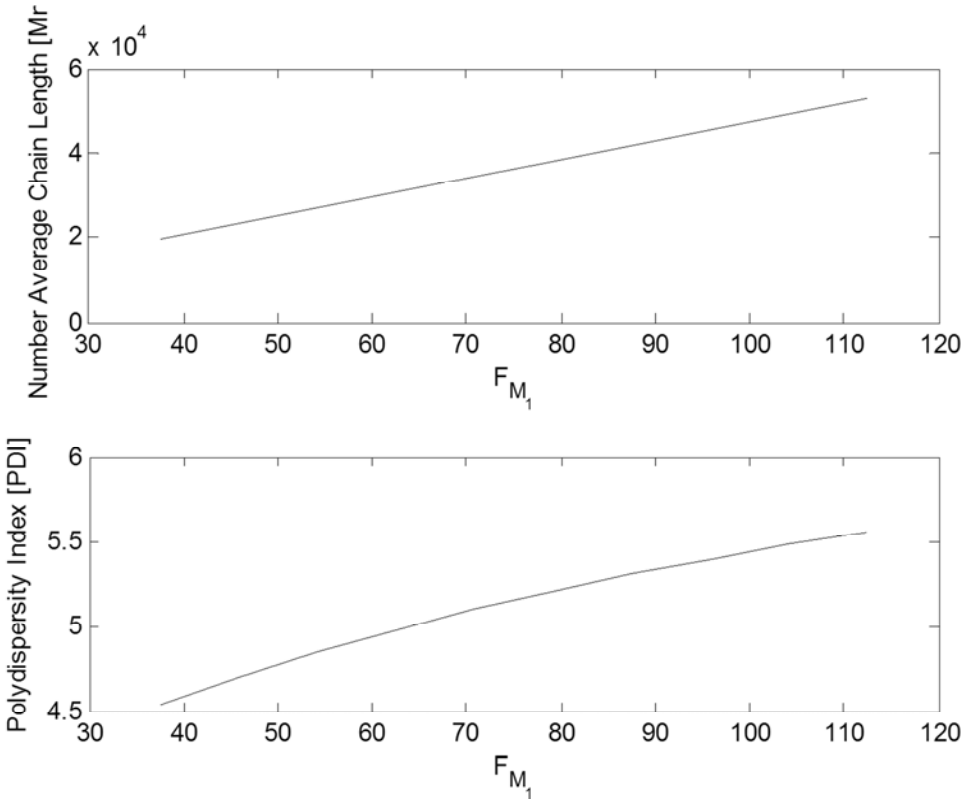

Figure C.2 - Influence of ethylene flowrate on polymer properties

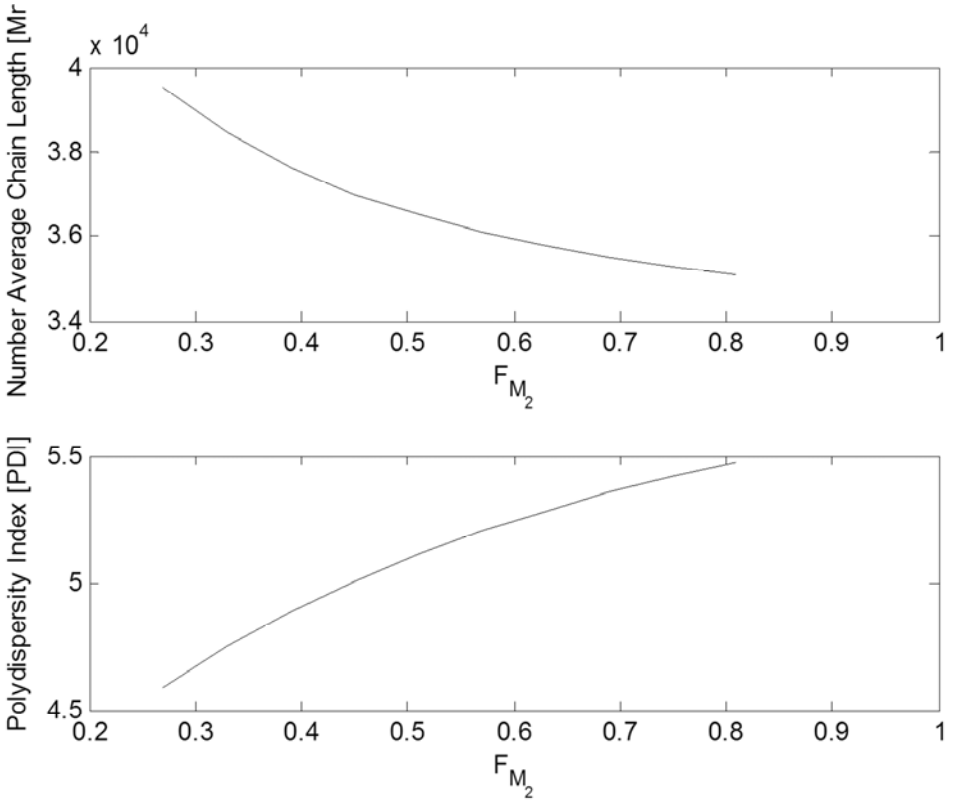

Figure C.3 - Influence of 1-butene flowrate on polymer properties

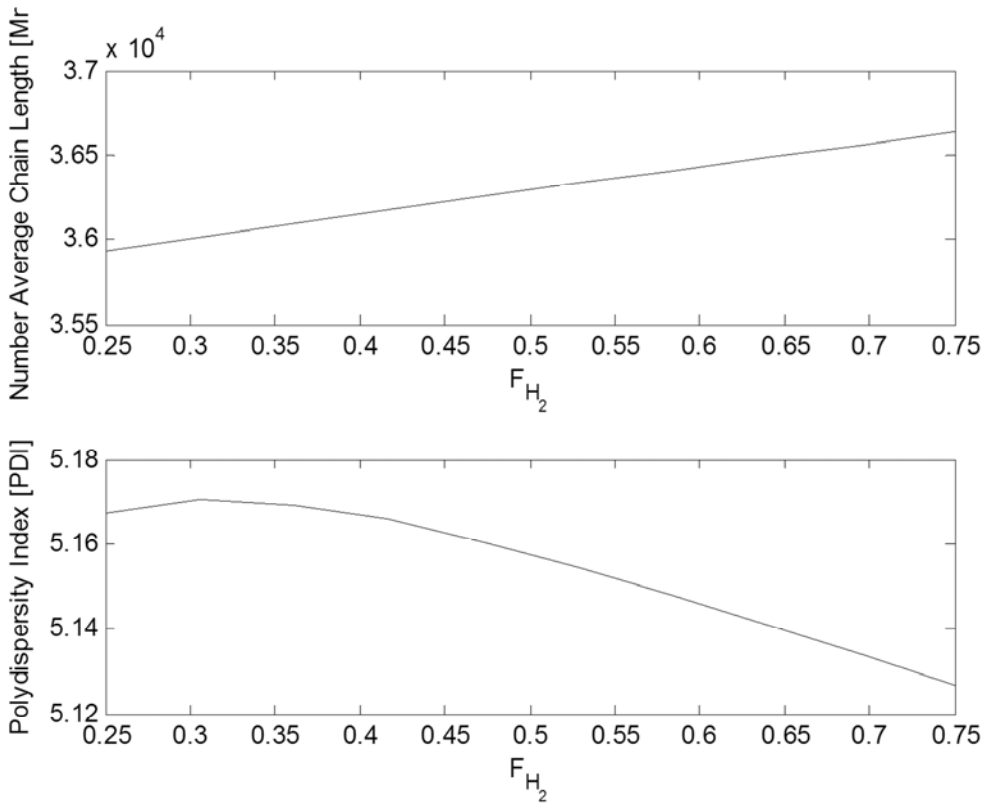

Figure C.4 - Influence of hydrogen flowrate on polymer properties

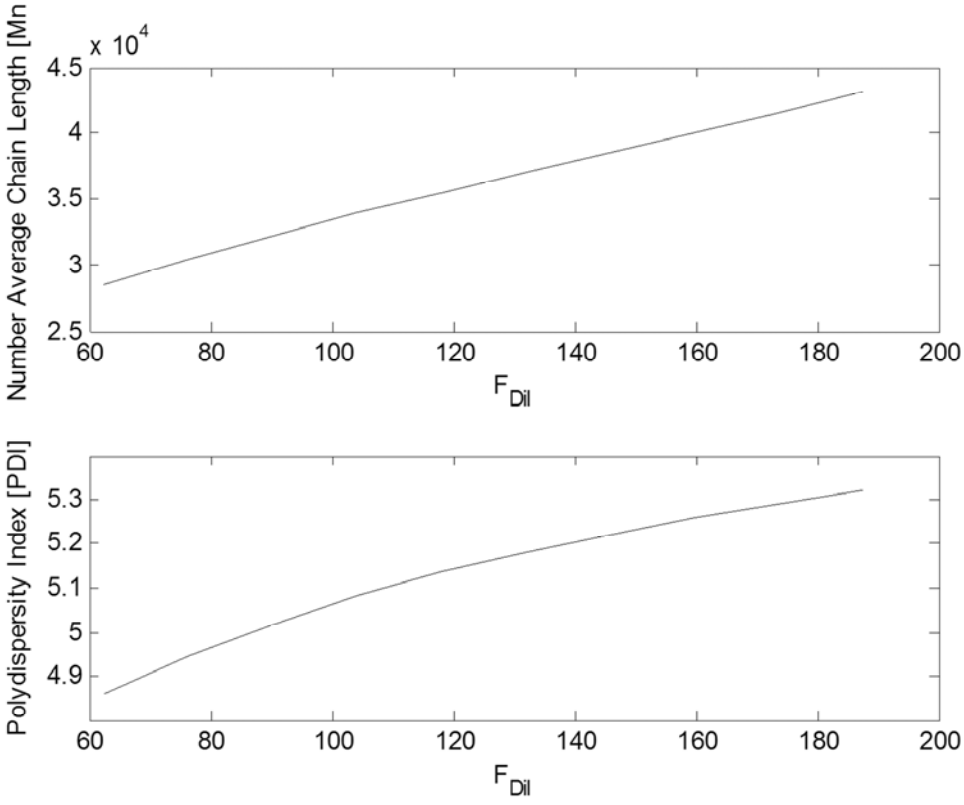

Figure C.5 - Influence of diluent flowrate on polymer properties

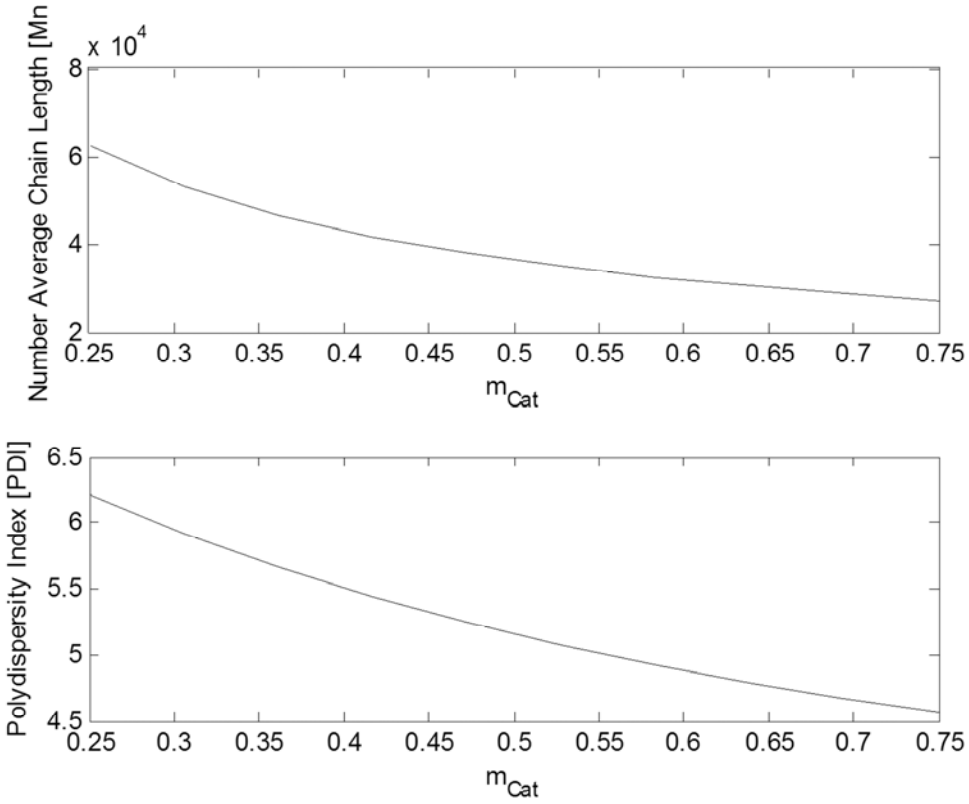

Figure C.6 - Influence of catalyst flowrate on polymer properties

D. Curve fitting for optimal steady-state blending

The following procedure applies to the blending scheme outlined in Section 6.1, in order to calculate the maximum flowrate at which an off-spec polymer source can be blended such that the intermediate polymer properties remain within their respective specification bands.

The flowrates of the off-spec polymer are plotted against the intermediate polymer properties that are required to blend with the off-spec source at the respective flowrate in Microsoft Excel. It was found that quadratic functions best approximated the trend over the data ranges. These equations determine the maximum off-spec flowrate, by substituting the upper or lower bound of the property specification band into the y-value and solving for x, depending on whether the property is decreasing or increasing, respectively, with increasing off-spec flowrate. The lowest positive x-value from either property is the maximum flowrate.

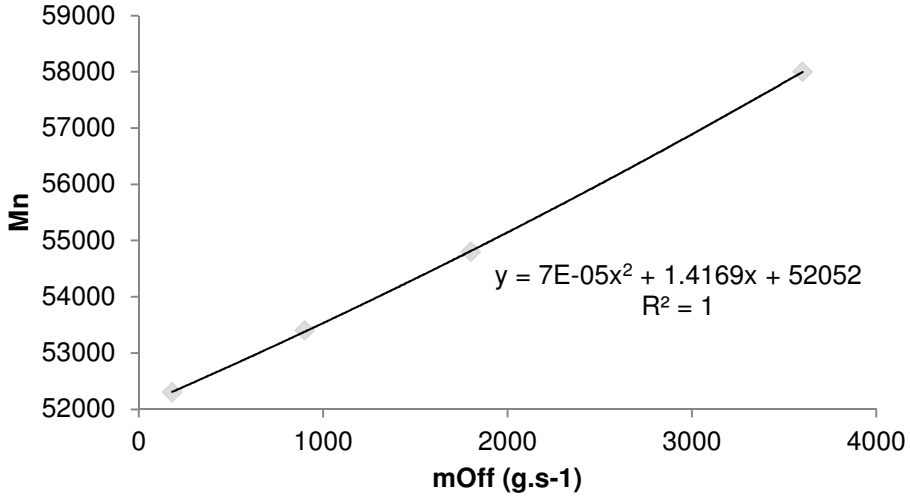

Figure D.1 - Curve fitting of Mn for off-spec source 1

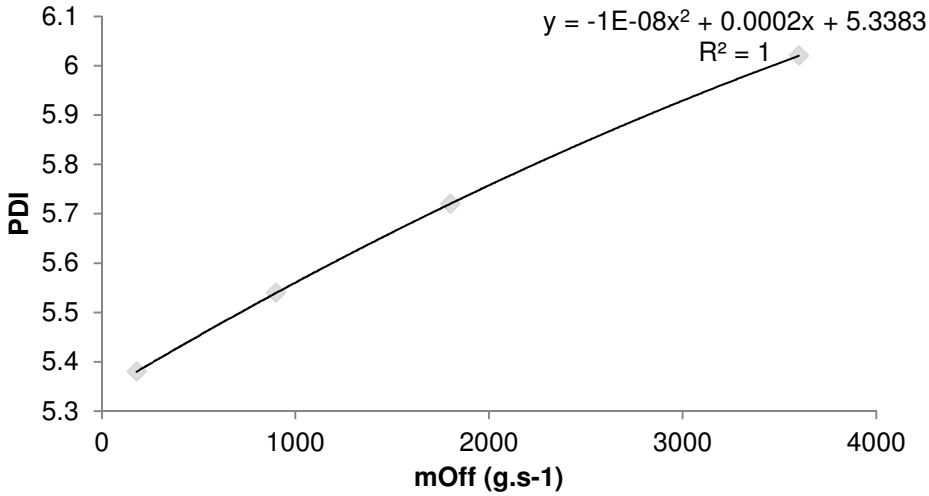

Figure D.2 - Curve fitting of PDI for off-spec source 1

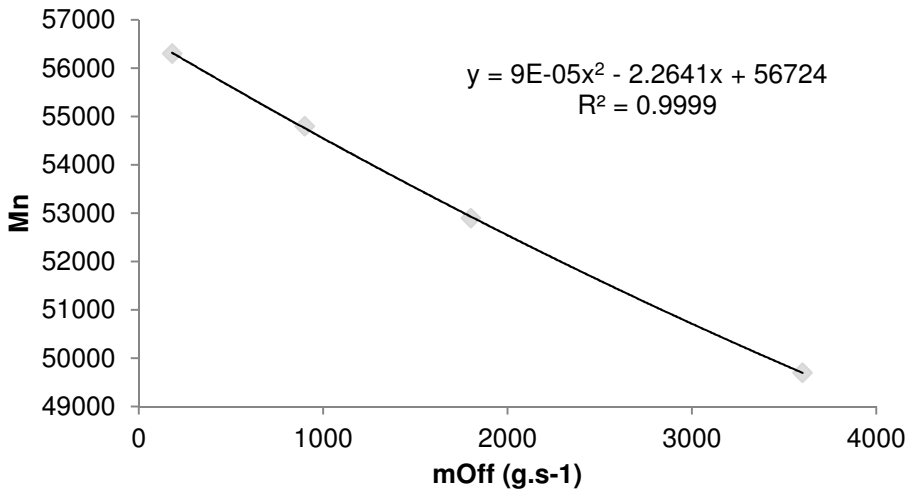

Figure D.3 - Curve fitting of Mn for off-spec source 2

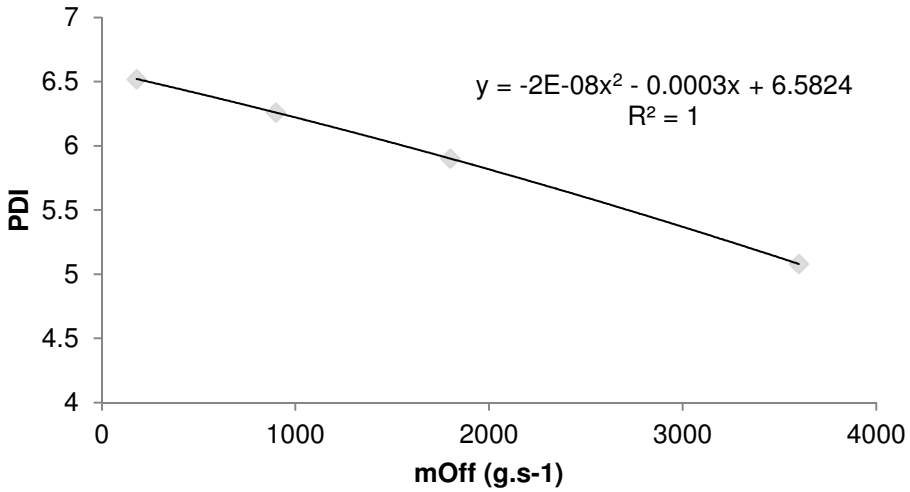

Figure D.4 - Curve fitting of PDI for off-spec source 2

E. Source code

i. Steady-state process model

The .m file presented below is used to determine the polymer properties under steady-state operation.

```
%=========================================================================
%Thaabit Nacerodien
%MSc Chemical Engineering
%University of Cape Town
%2013
%Filename: SSReactorModel.m
%Description: Main file for the simulation of polymerisation and
%off-spec blending. Calculates Mn, PDI and PSD of polymer produced
%in reactor based on feed rates and initial PSD. Also calculates Mn
%and PDI of this polymer blended with off-spec polymer.
%=========================================================================

cd(fileparts(mfilename('fullpath')));
parentpath = cd(cd('..'));
addpath(genpath(parentpath));

load('kinetics_constants','MM','rho','Conditions','l0m','sig');

%=========================================================================
%Feed Rates (Set F0i to the desired grade)
%=========================================================================
A = [0.129 51.57 0.16 1.89 36.1 0.52];
B = [0.127 48.36 0.36 1.71 33.85 0.36];

F0i = A;

Feed.F0 = F0i(1:5);
Feed.Cat0 = F0i(6);                                          %[g.s-1]

%Feed Conversions
F0m = Feed.F0.*MM;                                           %[g.s-1]
F0v = F0m(2:end)./rho/1000;                                  %[m3.s-1]

%=========================================================================
%Residence Time Data
%=========================================================================
Conditions.H0 = 80;
H0 = Conditions.H0;
RTD.tau = H0/sum(F0v);                                         %[s]
RTD.age = linspace(0,10*RTD.tau,150);                          %[s]
RTD.I = 1/RTD.tau*exp(-RTD.age/RTD.tau);                     %[s-1]
```

```
%=========================================================================
%Run
%=========================================================================
%Solve for exit concentrations in CSTR
%Initial Guess: [TEA C2H4 C4H8 H2 C9H18 Cat]
options = optimset('Display','off');
y = fsolve(@(x) CSTRPolySS(x,RTD,Feed,Conditions),1e-2*[Feed.F0
Feed.Cat0]*RTD.tau,options);
F = y(1:5)/RTD.tau;                                          %[mol.s-1]
Cat = y(6)/RTD.tau;                                            %[g.s-1]

%=========================================================================
%Parameter Extraction
%=========================================================================
prop = Moments(y,RTD,Conditions);                       %[mol/mol-cat]
mu = prop.mu;
prod = prop.Flow;
Vl = prop.Vl;
Bst = prop.Bst;
alpha = prop.alpha;
delta = prop.delta;
gamma = prop.gamma;

%=========================================================================
% Particle Size Distribution
%=========================================================================
Initial particle diameter range
l0 = linspace(0,l0m*3,100);                                        %[m]
%Initial PSD
f0m = normDist(l0,l0m,sig);
param.VlF = Vl/RTD.tau;
param.F = F;
param.Bst = Bst;
param.alpha = alpha;
param.gamma = gamma;
param.delta = delta;
Particle size: function of age and initial size
lm = meanPD(l0,RTD.age,param);                                     %[m]
%Mean final particle size
l =
nint(RTD.age,nint(l0,lm.*repmat(f0m',1,length(RTD.I)).*repmat(RTD.I,
length(l0),1),1),1);

%=========================================================================
%Ouptuts and Solutions
%=========================================================================
% Determine Polymer Production rate [g/s]
Min = sum(F0m) + Feed.Cat0;
Mout = sum(F.*MM) + Cat + prod;
MBerror = (Mout - Min)/Min;

% Determine Mn and PDI
prop = ChainProp(mu);
Mn = prop.Mn;
PDI = prop.PDI;
mu1 = mu(2);
```

```
fprintf('Mn: %3.2s \n',Mn);
fprintf('PDI: %3.2f \n',PDI);
fprintf('mu1: %3.2s \n',mu1);
fprintf('Polymer: %3.2f [tph] \n',prod/1000*3600/1000);
fprintf('Initial Particle Diameter: %3.2e [m] \n',l0m);
fprintf('Final Particle Diameter: %3.2e [m] \n',l);

% figure(1)
% plot(RTD.age,Act.f(:,1),RTD.age,Act.f(:,2),RTD.age,Act.f(:,3))
% xlabel('\theta [s]');
% ylabel('Catalyst Site Fraction')
% legend('P_a','P_*','P_d');

% figure(2)
% plot(RTD.age,RTD.I)
% xlabel('\theta [s]');
% ylabel('I(\theta) [s^-1]')

% surf(RTD.age,l0,lm)
% xlabel('\theta [s]');
% ylabel('Initial Diameter [m]');
% zlabel('Particle Diameter [m]');

%=========================================================================
%Off-Spec Polymer Properties (Set O to the desired Off-spec source)
%=========================================================================

%1
O1.mu1 = 5.5e5;
O1.Mn = 5e4;
O1.PDI = 5;

%2
O2.mu1 = 5.5e5;
O2.Mn = 6e4;
O2.PDI = 7;

%3
O3.mu1 = 5.5e5;
O3.Mn = 3e4;
O3.PDI = 6;

% 4
O4.mu1 = 5.5e5;
O4.Mn = 6e4;
O4.PDI = 5;

O = O3;
O.mu0 = O.mu1/O.Mn;
O.mu2 = O.PDI*O.mu1^2/O.mu0;
mu_O = [O.mu0 O.mu1 O.mu2];
m_O = 363;                                                    %[g/s]
```

```
%=========================================================================
%Polymer Blending
%=========================================================================
m_F = prod + m_O;

mu_F = mu_O*m_O/m_F + mu*prod/m_F;
prop_F = ChainProp(mu_F);
Mn_F = prop_F.Mn;
PDI_F = prop_F.PDI;

fprintf('Mn: %3.2s \n',Mn_F);
fprintf('PDI: %3.2f \n',PDI_F);
fprintf('mu1: %3.2s \n',mu_F(2));
fprintf('Polymer: %3.2f [tph] \n',m_F/1000*3600/1000);
```

The overall mass balances are solved in the following function file.

```
%=========================================================================
%Thaabit Nacerodien
%MSc Chemical Engineering
%University of Cape Town
%2013
%Filename: CSTRPolySS.m
%Description: Function file that returns the set of algebraic
%equations to be solved. These equations represent the CSTR mass
%balances as functions of the exiting concentrations.
%=========================================================================

function f = CSTRPolySS(y,RTD,Feed,Conditions)

load('kinetics_constants','NTi','H0');
T = Conditions.T;                                                  %[K]
P = Conditions.P;                                                %[bar]

N = y(1:5);                                                      %[mol]
Cat = y(end);                                                      %[g]

% perform isothermal flash
[xflash,yflash,L,ZL,ZV] = IsoFlash(N(2:end),T,P);

%Phase amounts [mol]
NVap = (1-L)*sum(N);
NLiq = L*sum(N);

% Phase Volumes [m^3]:
Vg = ZV*NVap*8.314*T/P/1e5;
Vl = ZL*NLiq*8.314*T/P/1e5;

C = [N(1) NLiq*xflash(1:3)]/Vl;            %[TEA C2 C4 H2] [mol.m-3]

Bst = StParameter(C);                                            %[s-1]

Act = ActiveSite(Bst,RTD.age);
```

```
%Concentration of active sites
parameters.p = Act.f(:,2);                                  %[mol/mol-Ti]
parameters.Bst = Bst;                                              %[s-1]

IM = repmat(RTD.I,5,1);                                            %[s-1]
Poly = PolyRates(C,parameters);
rates = IM'.*Poly.rates*NTi*Cat/H0;                       %[mol.m-3.s-1]
ri = nint(RTD.age,rates,1);                               %[mol.m-3.s-1]
xi = Poly.xi;
prod = ri(end)*H0*(28*mean(xi(:,1)) +
      56*mean(xi(:,2)));                                          %[g.s-1]

%System of equations to be solved
f(1:5) = Feed.F0' - N'/RTD.tau - [ri(1:end-1)*H0;0];
f(6) = Feed.Cat0 - Cat/RTD.tau;
end
```

The flash calculation to determine the vapour-liquid equilibrium was developed by McCoy using the algorithms given in Sandler (2006).

```
%=====================================================================
%IsoFlash makes use of the Peng-Robinson EoS to predict the
%isothermal flash of a multicomponent mixture. In this case,
%IsoFlash is set up for a flash of ethylene (C2), 1-butene (C4),
%hydrogen (H2), n-nonane (C9), n-hexane (nC6), propylene (C3),
%hexene (C6) and octene (C8). The call of IsoFlash is:
%[x,y,L,ZL,ZV] = IsoFlash([C2 C4 H2 C9 nC6 C3 C6 C8],T,P);
%Temperature T is in Kelvin and Pressure P is in bar (abs)

%The output is the mole fractions of liquid and gas phases (x,y),
%the fraction of the feed which is in the liquid phase (L) and the
%liquid and vapour compressibility factors (ZL,ZV)

%The IsoFlash function is based on algorithms presented in the
%textbook Chemical and Engineering Thermodynamics (3rd edition) by
%SI Sandler published by Wiley. Website for the fourth edition:
%http://he-cda.wiley.com/WileyCDA/HigherEdTitle/productCd-
%0471661740.html

% Filename: IsoFlash.m
% John McCoy, 2010
% Department of Chemical Engineering
% University of Cape Town
% South Africa
% john.themba.mccoy@gmail.com

%=====================================================================
function [xi,yi,L,ZL,ZV] = IsoFlash(x0,T,P)
global Lg

% Defining constants
R = 8.314e-5;                                          %[bar.m3/mol.K]
```

```
% components: ethylene butene hydrogen nonane hexane propylene
hexene octene
% critical properties from Perry
Tc = [282.34
    419.95
    33.19
    594.6
    507.6
    365.57
    504.03
    566.65];                                                    %[K]

Pc = [5.036
    4.023
    1.297
    2.29
    2.969
    4.63
    3.14
    2.57]*10;                                                  %[bar]

w = [0.086
    0.190
    -0.215
    0.446
    0.304
    0.137
    0.280
    0.377];

% constants for Pvap from Perry
Pvc = [74.242 68.49 12.69 109.35 104.65 57.263
      85.3 97.57;                                              %C1
      -2707.2 -4350.2 -94.896 -9030.4 -6995.5 -3382.4
      -6171.7 -7836;                                           %C2
      -9.8462 -7.4124 1.1125 -12.882 -12.702 -5.7077
      -9.702 -11.272;                                          %C3
      2.25E-02 1.05E-05 3.29E-04 7.85E-06 1.2381e-5 1.0431e-5
      8.9604e-6 7.7267e-6;                                     %C4
      1 2 2 2 2 2 2 2];                                        %C5

% Pvap = exp(c1 + c2/T + c3ln(T) + c4*T^c5) T in K, P in Pa

% binary interaction parameters
kij = [0     0.092  7.40E-03 0.11  0.04  0    0 0
    0.092    0      0        0.007 0     4e-4 0 0
    7.40E-03 0      0        0.1   -0.03 -0.1037 0 0
    0.11     0.007  0.1      0     0     0 0 0
    0.04     0      -0.03    0     0     0 0 0
    0        4e-4   -0.1037  0     0     0 0 0
    0        0      0        0     0     0 0 0
    0        0      0        0     0     0 0 0];
% mostly from Sandler, some from Aspen
```

```
%=========================================================================
% Number of components, and remove components which are 0 in x0
comp = 0;
for i = 1:length(x0)
    if x0(i)~=0
        comp = comp + 1;
        xhold(1,comp) = x0(i);
        Tchold(comp) = Tc(i);
        Pchold(comp) = Pc(i);
        whold(comp) = w(i);
        Pvchold(:,comp) = Pvc(:,i);
        comp2 = 0;
        for j = 1:length(x0)
            if x0(j)~=0
                comp2 = comp2 + 1;
                kijhold(comp,comp2) = kij(i,j);
            end
        end
    end
end
comp;
x0input = x0;
x0 = xhold;
Tc = Tchold';
Pc = Pchold';
w = whold';
Pvc = Pvchold;
kij = kijhold;

%=========================================================================
% Calculation of constant parameters

k = 0.37464 + 1.54226*w - 0.26992*w.^2;
alpha = (1 + k.*(1-(T./Tc).^(0.5))).^2;

a = 0.45724*R^2.*Tc.^2./Pc.*alpha;
b = 0.0778*R.*Tc./Pc;

aij = zeros(size(kij));

for i = 1:comp
    aij(i,i) = a(i);
end

for i = 1:comp
    for j = i:comp
        aij(i,j) = (1-kij(i,j))*(a(i)*a(j))^0.5;
        aij(j,i) = aij(i,j);
    end
end
```

```
%=========================================================================
% Begin calculations
x0 = x0/sum(x0); % normalise feed fractions

Kig = zeros(size(x0));
% initial guesses
for i = 1:comp
    Kig(i) = myPvap(Pvc(:,i),T)/P; % assume IG for guess
end

% Lg = 0.5;

count = 0;
conv = false;

while not(conv)&&(count<500)
    Lg = fminbnd(@(Lg) f_L(Lg,x0,Kig),0,1);
%     Lg = fzero(@(Lg) f_L(Lg,x0,Kig),Lg)

    x = x0./(Lg + Kig*(1-Lg));
    y = Kig.*x;

%=========================================================================
    % Calculation of vapour fugacity coefficient
    % a and b for mixture:
    mySuma = 0;
    mySumb = 0;
    for i = 1:comp
        mySumb = mySumb + b(i)*y(i);
        for j = 1:comp
            mySuma = mySuma + y(i)*y(j)*aij(i,j);
        end
    end
    aV = mySuma;
    bV = mySumb;

    % Solve cubic EOS for ZV:
    AV = aV*P/(R*T)^2;
    BV = bV*P/R/T;

    ZZ = myZroots(AV,BV);
    ZV = ZZ(3); % Vapour compressibility

    % Find fugacity coefficient for each component
    AVij = aij*P/(R*T)^2;
    BVi = b*P/R/T;
    Ahold = zeros(size(AVij));
    for i = 1:comp
        Ahold(i,:) = AVij(i,:).*y;
    end
    Asum = sum(Ahold,2);

    lnPhi = BVi/BV*(ZV-1)-log(ZV-BV)-AV/sqrt(8)/BV*(2*Asum/AV -
BVi/BV)*log((ZV + BV*(1+sqrt(2)))/(ZV + BV*(1-sqrt(2))));
    PhiV = exp(lnPhi');
```

```
%=========================================================================
    % Calculation of liquid fugacity coefficient
    % a and b for liquid mixture:
    mySuma = 0;
    mySumb = 0;
    for i = 1:comp
        mySumb = mySumb + b(i)*x(i);
        for j = 1:comp
            mySuma = mySuma + x(i)*x(j)*aij(i,j);
        end
    end
    aL = mySuma;
    bL = mySumb;

    % Solve cubic EOS for ZV:
    AL = aL*P/(R*T)^2;
    BL = bL*P/R/T;

    ZZ = myZroots(AL,BL);
    ZL = ZZ(1); % Liquid compressibility

    % Find fugacity coefficient for each component
    ALij = aij*P/(R*T)^2;
    BLi = b*P/R/T;
    ALhold = zeros(size(ALij));
    for i = 1:comp
        ALhold(i,:) = ALij(i,:).*x;
    end
    ALsum = sum(ALhold,2);

    lnPhi = BLi/BL*(ZL-1)-log(ZL-BL)-AL/sqrt(8)/BL*(2*ALsum/AL -
BLi/BL)*log((ZL + BL*(1+sqrt(2)))/(ZL + BL*(1-sqrt(2))));
    PhiL = exp(lnPhi');
```

```
%=====================================================================
    % Calculation of fugacities

    fV = PhiV.*y*P;
    fL = PhiL.*x*P;

    obj = norm2(fV - fL);

    if (obj < 1e-9)
        conv = true;
    else
        Kig = Kig.*ratio(fL,fV,0);
    end
    count = count + 1;
end

L = Lg;

myZero = 1;
for i = 1:length(x0input)
    if x0input(i)~=0
        xi(i) = x(myZero);
        yi(i) = y(myZero);
        myZero = myZero+1;
    else
        xi(i) = 0;
        yi(i) = 0;
    end
end

function [f,fvec] = f_L(L,x0,Ki)
f = (sum((1-Ki).*x0./(L + Ki*(1-L))))^2;
fvec = ((1-Ki).*x0./(L + Ki*(1-L)));

function Pvap = myPvap(C,T)
Pvap = exp(C(1) + C(2)/T + C(3)*log(T) + C(4)*T^C(5))/100000;
% Pvap is in bar, T in K

function out = myZroots(A,B)
p = [1 (-1 + B) (A - 3*B^2 - 2*B) (-A*B + B^2 + B^3)];

ZZ = roots(p);
```

```
% ================================
% vThis section of code from Sandler (www.wiley.com/college/sandler)
for i = 1:3
    if imag(ZZ(i)) ~= 0
        ZZ(i) = 0;
    end
end
%get rid of the imag root
%**************************
ZZ = sort(ZZ);

if abs(ZZ(1)) < 1e-8
    ZZ(1) = ZZ(3);
end
if abs(ZZ(3)) < 1e-8
    ZZ(3) = ZZ(1);
end
% ^This section of code from Sandler (www.wiley.com/college/sandler)
%================================

out = ZZ;

function norm = norm2(X)
n = length(X);
add = 0;
for i = 1:n
    add = add + X(i)^2; %add square of each element
end
norm = sqrt(add); %square root of sum of squares


function ydivx = ratio(num,denom,subsVal)
% Copyright R Rawatlal, School of Chemical Engineering, University
% of Natal, 2003
% Determine ratio of numerator num to denominator denom given that
% there might be zeros in the denominator. In the positions where
% there are indeed zeros in the denominator, subtitute the value
% subsVal in the result of the ratio.
% Syntax: ydivx = ratio(num,denom,subsVal)

ZMD = (denom == 0);
ydivx = squeeze((1 - ZMD).*num./(denom + ZMD) + subsVal.*ZMD);
```

```
%=========================================================================
%Thaabit Nacerodien
%MSc Chemical Engineering
%University of Cape Town
%2013
%Filename: StParameter.m
%Description: Function file determines the Lumped Site Transfer Rate
%=========================================================================

function Bst = StParameter(Conc)
%Thaabit Nacerodien
%function determines the Lumped Site Transfer Rate
load('kinetics_constants','kst','ast');

C = repmat([1 Conc],3,1);
Bst = sum(kst.*C.^ast,2);

end

%=========================================================================
%Thaabit Nacerodien
%MSc Chemical Engineering
%University of Cape Town
%2013
%Filename: ActiveSite.m
%Description: Function file that returns the fraction of each
%catalyst site also returns the value of each term for active site
%fraction
%=========================================================================

function out = ActiveSite(Bst,t)
global alpha gamma delta

%Ti4+
f1 = exp(-Bst(1)*t);

%Ti3+
alpha = Bst(3)/(Bst(2) + Bst(3));
gamma = (Bst(3) - Bst(1))/(Bst(1) - Bst(2) - Bst(3));
delta = (Bst(1)*Bst(2))/(Bst(1)*Bst(2) + Bst(1)*Bst(3) -
2*Bst(3)*Bst(2) - Bst(2)^2 - Bst(3)^2);
f2 = alpha + gamma*exp(-Bst(1)*t) + delta*exp(-(Bst(2)+Bst(3))*t);

%Ti2+
f3 = 1 - f1 - f2;

out.f = [f1' f2' f3'];
out.alpha = alpha;
out.gamma = gamma;
out.delta = delta;
end
```

```
%=====================================================================
%Thaabit Nacerodien
%MSc Chemical Engineering
%University of Cape Town
%2013
%Filename: PolyRates.m
%Description: Function file that returns the fraction of each
%catalyst site
%also returns the value of each term for active site fraction
%=====================================================================

function out = PolyRates(C,parameters)
load('kinetics_constants','kt','kp','k0');
Cm = [1 C];                                 %[mol.m-3]
p = parameters.p;                           %[mol/mol-Ti]
Bst = parameters.Bst;                       %[mol/mol-Ti]

%Parameters to determine xi
a = Bst(2) + k0(1)*C(2) + kp(1,2)*C(3) + kt(1,:).*Cm;
b = k0(1)*C(2) - kp(2,1)*C(2);
c = k0(1)*C(2);
d = k0(2)*C(3) - kp(1,2)*C(3);
e = Bst(2) + k0(2)*C(3) + kp(2,1)*C(2) + kt(2,:).*Cm;
f = k0(2)*C(3);

%Monomer Composition
xi(:,1) = (c-b*f./e)./(a-b*d./e);
xi(:,2) = (f-c*d./a)./(e-b*d./a);

%Pseudo-site fraction
fm = xi*(Cm*kt')'/sum(xi*(Cm*kt')');

%Zeroth moment for each monomer [mol/mol-Ti]
lam01 = p*fm'*xi(:,1);
lam02 = p*fm'*xi(:,2);

%Find component consumption rates [mol/mol-Ti.s-1]
rTEA = C(1)*(lam01*kt(1,2) + lam02*kt(2,2));
rC2H4 = C(2)*(lam01*(kp(1,1)+kt(1,3)) + lam02*(kp(2,1)+kt(2,3)));
rC4H8 = C(3)*(lam01*(kp(1,2)+kt(1,4)) + lam02*(kp(2,2)+kt(2,4)));
rH2 = C(4)*(lam01*kt(1,5) + lam02*kt(2,5));
rP = rC2H4 + rC4H8 + rH2;

M = [C(2) C(3)];                            %[mol.m-3] [C2H4 C4H8]

myNum = 0;
myDen = 0;
for i = 1:2
    mySum = 0;
    for j = 1:2
        myNum = myNum + kp(j,i)*M(i).*xi(:,j);
        mySum = mySum + kp(i,j)*M(j);
    end
    myDen = myDen + (mySum + kt(i,:).*Cm + Bst(2)).*xi(:,i)';
end
```

```
%Chain length Distribution Parameter (CLDP)
gamma = myNum./myDen';

%Determine the instantaneous moments [mol/mol-Me]
lam0 = p*sum(fm.*sum(xi,2));
lam1 = p*sum(fm.*sum(xi,2).*1./(1-gamma));
lam2 = p*sum(fm.*sum(xi,2).*(1 + gamma)./(1-gamma).^2);

out.rates = [rTEA rC2H4 rC4H8 rH2 rP];
out.xi = xi;
out.fm = fm;
out.lam01 = lam01;
out.lam02 = lam02;
out.lam0 = lam0;
out.lam1 = lam1;
out.lam2 = lam2;

end

%=====================================================================
%Thaabit Nacerodien
%MSc Chemical Engineering
%University of Cape Town
%2013
%Filename: Moments.m
%Description: Function file that returns the bulk polymer moments
%based on the exit concentrations in the reactor
%=====================================================================

function out = Moments(y,RTD,Conditions)
load('kinetics_constants','kt','kp','NTi','H0');

F = y(1:5)/RTD.tau;                                    %[mol.s-1]
Cat = y(6)/RTD.tau;                                    %[g.s-1]

[xflash,yflash,L,ZL,ZV] =
IsoFlash(F(2:end),Conditions.T,Conditions.P);
%Phase Flows [mol/s]
FVap = (1-L)*sum(F);
FLiq = L*sum(F);
% Phase Volumes [m^3.s-1]:
Vg = ZV*FVap*8.314*Conditions.T/Conditions.P/1e5;
Vl = ZL*FLiq*8.314*Conditions.T/Conditions.P/1e5;

C = [F(1) FLiq*xflash(1:3)]/Vl;        %[TEA C2H4 C4H8 H2] [mol.m-3]
Cm = [1 C];
M = [C(2) C(3)];                              %[C2H4 C4H8] [mol.m-3]
Bst = StParameter(C);
Bt = TermParameter(C);                                 %[s-1]

Act = ActiveSite(Bst,RTD.age);
parameters.p = Act.f(:,2);                      %[mol/mol-Ti]
parameters.Bst = Bst;                                  %[s-1]

IM = repmat(RTD.I,5,1);                                %[s-1]
Poly = PolyRates(C,parameters);
rates = IM'.*Poly.rates*NTi*Cat/H0*RTD.tau;          %[mol.m-3.s-1]
```

```
ri = nint(RTD.age,rates,1);                                %[mol.m-3.s-1]
xi = Poly.xi;
fm = Poly.fm;

myNum = 0;
myDen = 0;
for i = 1:2
    mySum = 0;
    for j = 1:2
        myNum = myNum + kp(j,i)*M(i).*xi(:,j);
        mySum = mySum + kp(i,j)*M(j);
    end
    myDen = myDen + (mySum + kt(i,:).*Cm + Bst(2)).*xi(:,i)';
end

%Chain length Distribution Parameter (CLDP)
gamma = myNum./myDen';

%Determine the instantaneous moments [mol/mol-Ti]
lam(:,1) = parameters.p*sum(fm.*sum(xi,2));
lam(:,2) = parameters.p*sum(fm.*sum(xi,2).*1./(1-gamma));
lam(:,3) = parameters.p*sum(fm.*sum(xi,2).*(1 + gamma)./(1-
gamma).^2);

%Rate of formation of bulk moments [mol/mol-Ti.s-1]
dmudt = (Bst(2) + sum(Bt))*lam;

%Determine bulk moments using segregation model
for i = 1:3
    mu(i) = nint(RTD.age,RTD.I'.*dmudt(:,i).*RTD.age',1);
                                                           %[mol/mol-Ti]
end

prod = ri(end)*H0*(28*mean(xi(:,1))
      + 56*mean(xi(:,2)));                                 %[g.s-1]

out.mu = mu*NTi*Cat*RTD.tau/H0;
out.Flow= prod;
out.Vl = Vl;
out.Bst = Bst;
out.alpha = Act.alpha;
out.delta = Act.delta;
out.gamma = Act.gamma;
```

```
%=========================================================================
%Thaabit Nacerodien
%MSc Chemical Engineering
%University of Cape Town
%2013
%Filename: TermParameter.m
%Description: Function file determines the Lumped Termination Rate
%=========================================================================

function Bt = TermParameter(Conc)

load('kinetics_constants','kt')
C = [1 Conc];
Bt = (C*kt')';

end

%=========================================================================
%Thaabit Nacerodien
%MSc Chemical Engineering
%University of Cape Town
%2013
%Filename: normDist.m
%Description: Function file returns normal distribution. Accepts
%range, mean and variance as inputs
%=========================================================================

function out = normDist(l0,l0m,sig)

out = 1/sqrt(2*pi()*sig^2).*exp(-(l0-l0m).^2./(2*sig^2));

%=========================================================================
%Thaabit Nacerodien
%MSc Chemical Engineering
%University of Cape Town
%2013
%Filename: meanPD.m
%Description: Function file returns the mean particle size of
%polymer precipitate
%=========================================================================
function out = meanPD(l0,age,param)

load('kinetics_constants','kp','MM','NTi');

VlF = param.VlF;
F = param.F;
Bst = param.Bst;
alpha = param.alpha;
gamma = param.gamma;
delta = param.delta;

M = F(2:3)/VlF;
G = NTi*(sum(kp(1,:).*M(1)*MM(2)) + sum(kp(2,:).*M(2)*MM(3)))/60;
intAct = alpha*age - gamma/Bst(1)*(exp(-Bst(1)*age) - 1) -
delta/(Bst(2) + Bst(3))*(exp(-Bst(2)*age-Bst(3)*age)-1);
```

```
for i = 1:length(age)
l(:,i) = l0'*(G*intAct(i) + 1).^(1/3);
end

out = l;

%=========================================================================
%Thaabit Nacerodien
%MSc Chemical Engineering
%University of Cape Town
%2013
%Filename: ChainProp.m
%Description: Function file to predict the Number Average Chain
%Length (Mn) and
%Polydispersity Index (PDI) of a polymer product based on the
%moments
%=========================================================================

function prop = ChainProp(mu)
prop.Mn = mu(:,2)./mu(:,1);
prop.PDI = mu(:,1).*mu(:,3)./mu(:,2).^2;
end
```

ii. Steady-state blending optimisation

```
%=========================================================================
%Thaabit Nacerodien
%MSc Chemical Engineering
%University of Cape Town
%2013
%Filename: SSFeedPredict.m
%Description: Main file to determine feed rate adjustments to
%produce intermediary polymer for blending with off-spec material as
%described in Section 5.1
%=========================================================================

cd(fileparts(mfilename('fullpath')));
parentpath = cd(cd('..'));
addpath(genpath(parentpath));

%=========================================================================
%Desired grade properties (Set F0i to desired grade)
%=========================================================================
A.Mn = 5.2e4;
A.PDI = 5.34;

B.Mn = 5.67e4;
B.PDI = 6.58;

F0i = B;

%=========================================================================
%Off-Spec polymer properties (Set O to the desired Off-spec source)
%=========================================================================
%1
O1.mu1 = 5.5e5;
O1.Mn = 5e4;
O1.PDI = 5;

%2
O2.mu1 = 5.5e5;
O2.Mn = 6e4;
O2.PDI = 7;

%3
O3.mu1 = 5.5e5;
O3.Mn = 3e4;
O3.PDI = 6;

%4
O4.mu1 = 5.5e5;
O4.Mn = 6e4;
O4.PDI = 5;

O = O3;
O.Flow = 900;
```

```
%=========================================================================
%Feed Rates
%=========================================================================
options=optimset('Display','iter','TolFun',1e-5,'TolX',1);
guess = [0.12 50 0.2 1.2 35 0.4];
F0i = fminsearch(@(x) LSIFlow(x,O,F),guess,options);


%=========================================================================
%Thaabit Nacerodien
%MSc Chemical Engineering
%University of Cape Town
%2013
%Filename: LSIFlow.m
%Description: Function to determine feed rate adjustments to produce
%intermediary polymer for blending with off-spec material as
%described in Section 5.1
%Objective Function is evaluated subject to selected feed rates
%=========================================================================
function I = LSIFlow(x,O,D)
load('kinetics_constants');

%=========================================================================
%Feed Rates
%=========================================================================
Feed.F0 = x(1:end-1);
Feed.Cat0 = x(end);
F0m = Feed.F0.*MM;                                              %[g.s-1]
F0v = F0m(2:end)./rho/1e3;                                      %[m3.s-1]

%=========================================================================
%Residence Time Data
%=========================================================================
H0 = 80;                                                          %[m3]
Conditions.H0 = H0;
RTD.tau = H0/sum(F0v);                                             %[s]
RTD.age = linspace(0,10*RTD.tau,150);                              %[s]
RTD.I = 1/RTD.tau*exp(-RTD.age/RTD.tau);
%=========================================================================
%Run
%=========================================================================
%Solve for exit concentrations in CSTR
options = optimset('Display','off');
y = fsolve(@(x) CSTRPolySS(x,RTD,Feed,Conditions),1e-
2*x*RTD.tau,options);                                          %[mol.s-1]
R = Moments(y,RTD,Conditions);
propR = ChainProp(R.mu);
R.Mn = propR.Mn;
R.PDI = propR.PDI;
F.mu = Blend(R,O);
propF = ChainProp(F.mu);
F.Mn = propF.Mn;
F.PDI = propF.PDI;

I = (1 - F.Mn/D.Mn)^2 + (1 - F.PDI/D.PDI)^2;
```

```
%=========================================================================
%Thaabit Nacerodien
%MSc Chemical Engineering
%University of Cape Town
%2013
%Filename: Blend.m
%Description: Function file returns the moments of a blend
%=========================================================================
function muF = Blend(R,O)

O.mu(1) = O.mu1/O.Mn;
O.mu(2) = O.mu1;
O.mu(3) = O.PDI*O.mu(2)^2/O.mu(1);
F.Flow = R.Flow + O.Flow;

muF = (R.Flow/F.Flow)*R.mu + (O.Flow/F.Flow)*O.mu;
```

iii. Unsteady-state polymerisation

In order to simulate a grade change, the steady-state values of all the components under production of the first grade are required as initial values. Thus, the unsteady-state polymerisation simulation is run once from start-up to steady-state for each grade and the final values saved. The code for Grade A is shown below. The values for Grade B are found in a similar manner.

```
%=========================================================================
%Thaabit Nacerodien
%MSc Chemical Engineering
%University of Cape Town
%2013
%Filename: InitialiseA.m
%Description: Start-up simulation of grade A
%=========================================================================

cd(fileparts(mfilename('fullpath')));
parentpath = cd(cd('..'));
addpath(genpath(parentpath));

load('kinetics_constants','MM','rho','Conditions','k0','kp','kst','k
t','ast','NTi','Psat');
tic
%=========================================================================
%Time Scale
%=========================================================================
t0 = 0;
tf = 8e4;

%=========================================================================
%Feed Rates
%=========================================================================
%A
f0.TEA = 0.129;
f0.C2 = 51.57;
f0.C4 = 0.16;
f0.H2 = 1.89;
f0.Dil = 36.1;
f0.Cat = 0.52;                                              %[g.s-1]

F0 = [f0.TEA f0.C2 f0.C4 f0.H2 f0.Dil];                   %[mol.s-1]
F0m = F0.*MM;                                               %[g.s-1]
F0v = F0m(2:end)./rho/1e3;                                 %[m3.s-1]
f0.Pol = 0;                                               %[mol.s-1]
tv = [];
%=========================================================================
%Residence Time Data
%=========================================================================
Conditions.H0 = 80;
H0 = Conditions.H0;
RTD.tau = H0/sum(F0v);                                          %[s]
RTD.age = linspace(0,RTD.tau*10,150);                           %[s]
```

```
RTD.I = 1/RTD.tau*exp(-RTD.age/RTD.tau);                    %[s-1]
constants = {k0,kp,kst,kt,ast,rho,MM,NTi,H0,Psat,tv};
%=====================================================================
%Run
%=====================================================================
[tA,yA] = ode45(@ReactorSys,[t0 tf],1e-2*[[f0.Cat F0(1:4) f0.Pol
F0(5)]*RTD.tau 1e-10 1e-10 1e-10],[],RTD,f0,Conditions,constants);

A = yA(end,:);

save('gradeA','A');
toc
```

A grade change is simulated using the following code.

```
%=====================================================================
%Thaabit Nacerodien
%MSc Chemical Engineering
%University of Cape Town
%2013
%Filename: FWDModel.m
%Description: Dynamic Polymerisation Reactor Model
%=====================================================================

cd(fileparts(mfilename('fullpath')));
parentpath = cd(cd('..'));
addpath(genpath(parentpath));

load('kinetics_constants','MM','rho','Conditions','k0','kp','kst','k
t','ast','rho','MM','NTi','Psat');

%=====================================================================
%Time Scale
%=====================================================================
t0 = 0;
tf = 8e4;

%=====================================================================
%Grade change AB
%=====================================================================
load('gradeA'); %Initial values

%Feed Rates for Grade B
f0.TEA = 0.127;
f0.C2 = 48.36;
f0.C4 = 0.36;
f0.H2 = 1.71;
f0.Dil = 33.85;
f0.Cat = 0.36;

%Upper and Lower Limits for Grade B
Mn_u = 5.8e4;
Mn_l = 5.6e4;

PDI_u = 6.75;
PDI_l = 6.45;
```

```
initial = A;


%=========================================================================
%Grade change BA
%=========================================================================
% load('gradeB'); %Initial values
%
% %Feed Rates for Grade A
% f0.TEA = 0.129;
% f0.C2 = 51.57;
% f0.C4 = 0.16;
% f0.H2 = 1.89;
% f0.Dil = 36.1;
% f0.Cat = 0.52;
%
% %Upper and Lower Limits for Grade A
% Mn_u = 5.3e4;
% Mn_l = 5.1e4;
%
% PDI_u = 5.5;
% PDI_l = 5.2;
%
% initial = B;

F0 = [f0.TEA f0.C2 f0.C4 f0.H2 f0.Dil];
F0m = F0.*MM;                                                    %[g.s-1]
F0v = F0m(2:end)./rho/1e3;                                       %[m3.s-1]
f0.Pol = 0;                                                      %[mol.s-1]
%=========================================================================
%New Residence Time Data
%=========================================================================
Conditions.H0 = 80;
H0 = Conditions.H0;
RTD.tau = H0/sum(F0v);                                             %[s]
RTD.age = linspace(0,RTD.tau*10,150);                              %[s]
RTD.I = 1/RTD.tau*exp(-RTD.age/RTD.tau);                         %[s-1]

%=========================================================================
%Second Run
%=========================================================================
tv = [];
constants = {k0,kp,kst,kt,ast,rho,MM,NTi,H0,Psat,tv};
[tB,yB] = ode45(@ReactorSys,[t0
tf],initial,[],RTD,f0,Conditions,constants);
```

```
%=========================================================================
%Time Dependent Solutions
%=========================================================================
mCatB = yB(:,1)/RTD.tau;                                          %[g.s-1]
FTEAB = yB(:,2)/RTD.tau;                                         %[mol.s-1]
FC2B = yB(:,3)/RTD.tau;                                          %[mol.s-1]
FC4B = yB(:,4)/RTD.tau;                                          %[mol.s-1]
FH2B = yB(:,5)/RTD.tau;                                          %[mol.s-1]
FPolB = yB(:,6)/RTD.tau;                                         %[mol.s-1]
FdilB = yB(:,7)/RTD.tau;                                         %[mol.s-1]
mu0B = yB(:,8);                                                  %[mol.m-3]
mu1B = yB(:,9);                                                  %[mol.m-3]
mu2B = yB(:,10);                                                 %[mol.m-3]

prop = ChainProp([mu0B mu1B mu2B]);
Mn = prop.Mn;
PDI = prop.PDI;

%=========================================================================
%Mass Balance Check
%=========================================================================
Min = sum(F0m) + f0.Cat;                                          %[g.s-1]
mB = FPolB*MM(2);
Mout = FTEAB*MM(1) + FC2B*MM(2) + mB + FC4B*MM(3) + FH2B*MM(4) +
FdilB*MM(5) + mCatB;                                              %[g.s-1]

%=========================================================================
%Graphical Outputs
%=========================================================================
figure(1)
subplot(3,1,1)
plot(tB,Mn,tB,repmat(Mn_u,1,length(tB)),'--
',tB,repmat(Mn_l,1,length(tB)),'--')
xlabel('Time [s]');
ylabel('Number Average Chain Length [Mn]');
ylim([5e4 6e4]);
subplot(3,1,2)
plot(tB,PDI,tB,repmat(PDI_u,1,length(tB)),'--
',tB,repmat(PDI_l,1,length(tB)),'--')
xlabel('Time [s]');
ylabel('Polydispersity Index [PDI]');
subplot(3,1,3)
plot(tB,mB);
xlabel('Time [s]');
ylabel('m_{Pol} [g.s^{-1}]');
```

The set of differential equations that are central to the unsteady-state model are defined in the following function file.

```
%=========================================================================
%Thaabit Nacerodien
%MSc Chemical Engineering
%University of Cape Town
%2013
%Filename: ReactorSys.m
%Description: Function file returns the solution to the set of
%differentials describing the material balances for polymerisation
%in a CSTR.
%=========================================================================
function dydt = ReactorSys(t,y,RTD,f0,Conditions,constants)

%=========================================================================
%Function File for Component Material and Bulk Moments Balances
%=========================================================================

%Feed Properties
kst = constants{3};
kt = constants{4};
ast = constants{5};
NTi = constants{8};
H0 = constants{9};
Psat = constants{10};
tv = constants{11};

MCat = y(1);                                                    %[g]
FTEA = y(2);                                                  %[mol]
FC2 = y(3);                                                   %[mol]
FC4 = y(4);                                                   %[mol]
FH2 = y(5);                                                   %[mol]
FPol = y(6);                                                  %[mol]
FDil = y(7);                                                  %[mol]
mu0 = y(8);                                                   %[mol]
mu1 = y(9);                                                   %[mol]
mu2 = y(10);                                                  %[mol]

F = [FC2 FC4 FH2 FDil];                                       %[mol]

% perform isothermal flash
[xflash,yflash,L,ZL,ZV] = IsoFlash(F,Conditions.T,Conditions.P);

%Phase amounts [mol]
FVap = (1-L)*sum(F);
FLiq = L*sum(F);

% Phase Volumes [m^3]:
Vl = ZL*FLiq*8.314*Conditions.T/Conditions.P/1e5;

C = [FTEA FLiq*xflash(1:3)]/Vl;                             %[mol.m-3]
%Lumped Parameters
Bt = ([1 C]*kt');                                             %[s-1]
Bst = sum(kst.*repmat([1 C],3,1).^ast,2);                     %[s-1]
```

```
%Catalyst Active Site
Pa0 = 1;                                                    %[mol/mol-Ti]
[Pa P Pd] = CatActivity(t,RTD.age,Bst',Pa0);                %[mol/mol-Ti]
P_s = nint(RTD.age,RTD.I.*P,2);
parameters.p = P_s;                                         %[mol/mol-Ti]
parameters.Bst = Bst;                                              %[s-1]
Poly = PolyRates(C,parameters);
ri = Poly.rates*NTi*MCat/H0;                              %[mol.m-3.s-1]

lamda = [Poly.lam0 Poly.lam1 Poly.lam2];
lam0 = lamda(1)*NTi*MCat/H0;
lam1 = lamda(2)*NTi*MCat/H0;
lam2 = lamda(3)*NTi*MCat/H0;

if length(f0.H2) > 1
    f0.H2(end+1) = f0.H2(end);
    f0.H2 = f0.H2(t>=tv);
    f0.H2 = f0.H2(end);
elseif length(f0.H2) == 1
    f0.H2 = f0.H2;
end

if length(f0.C2) > 1
    f0.C2(end+1) = f0.C2(end);
    f0.C2 = f0.C2(t>=tv);
    f0.C2 = f0.C2(end);
elseif length(f0.C2) == 1
    f0.C2 = f0.C2;
end

if length(f0.Cat) > 1
    f0.Cat(end+1) = f0.Cat(end);
    f0.Cat = f0.Cat(t>=tv);
    f0.Cat = f0.Cat(end);
elseif length(f0.Cat) == 1
    f0.Cat = f0.Cat;
end

rTEA = ri(1);
rC2 = ri(2);
rC4 = ri(3);
rH2 = ri(4);
rPol = ri(5);

f(1) = f0.Cat - MCat/RTD.tau;                                  %[mol.s-1]
f(2) = f0.TEA - rTEA*H0 - FTEA/RTD.tau;                        %[mol.s-1]
f(3) = f0.C2 - rC2*H0 - FC2/RTD.tau;                           %[mol.s-1]
f(4) = f0.C4 - rC4*H0 - FC4/RTD.tau;                           %[mol.s-1]
f(5) = f0.H2 - rH2*H0 - FH2/RTD.tau;                           %[mol.s-1]
f(6) = f0.Pol + rPol*H0 - FPol/RTD.tau;                        %[mol.s-1]
f(7) = f0.Dil - FDil/RTD.tau;                                  %[mol.s-1]
f(8) = (Bt(2)+sum(Bst))*lam0-mu0/RTD.tau;                 %[mol.m-3.s-1]
f(9) = (Bt(2)+sum(Bst))*lam1-mu1/RTD.tau;                 %[mol.m-3.s-1]
f(10) = (Bt(2)+sum(Bst))*lam2-mu2/RTD.tau;                %[mol.m-3.s-1]

dydt = f';
```

```
%=========================================================================
%Thaabit Nacerodien
%MSc Chemical Engineering
%University of Cape Town
%2013
%Filename: BlendSys.m
%Description: Function file returns the solution to the set of
%differentials for blending two polymers in a CSTR.
%=========================================================================
function dydt = BlendSys(t,y,initial)

MD = y(1);
muD1 = y(2);
muD2 = y(3);
muD3 = y(4);

mO = initial{1};
mR = initial{4};

muO = initial{2};
muO1 = muO(:,1);
muO2 = muO(:,2);
muO3 = muO(:,3);

muR = initial{5};
muR1 = muR(:,1);
muR2 = muR(:,2);
muR3 = muR(:,3);

tO = initial{3};
tR = initial{6};
M0 = initial{7};

if length(mO) > 1
    mO = mO(t>=tO);
    mO = mO(end);
elseif length(mO) == 1
    mO = mO;
end

if length(mR) > 1
    mR = mR(t>=tR);
    mR = mR(end);
elseif length(mR) == 1
    mR = mR;
end

if length(muO1) > 1
    muO1 = muO1(t>=tO);
    muO1 = muO1(end);
elseif length(muO1) == 1
    muO1 = muO1;
end
```

```
if length(muO2) > 1
    muO2 = muO2(t>=tO);
    muO2 = muO2(end);
elseif length(muO2) == 1
    muO2 = muO2;
end

if length(muO3) > 1
    muO3 = muO3(t>=tO);
    muO3 = muO3(end);
elseif length(muO3) == 1
    muO3 = muO3;
end

if length(muR1) > 1
    muR1 = muR1(t>=tR);
    muR1 = muR1(end);
elseif length(muR1) == 1
    muR1 = muR1;
end

if length(muR2) > 1
    muR2 = muR2(t>=tR);
    muR2 = muR2(end);
elseif length(muR2) == 1
    muR2 = muR2;
end

if length(muR3) > 1
    muR3 = muR3(t>=tR);
    muR3 = muR3(end);
elseif length(muR3) == 1
    muR3 = muR3;
end

tau = M0/(mO + mR);

f(1) = mO + mR - MD/tau;
f(2) = mO/MD*(muO1 - muD1) + mR/MD*(muR1 - muD1);
f(3) = mO/MD*(muO2 - muD2) + mR/MD*(muR2 - muD2);
f(4) = mO/MD*(muO3 - muD3) + mR/MD*(muR3 - muD3);

dydt = f';
```

iv. Unsteady-state blending optimisation

```
%=========================================================================
%Thaabit Nacerodien
%MSc Chemical Engineering
%University of Cape Town
%2013
%Filename: predictOffSpecFeedRate.m
%Description: Main file for dynamic optimisation using Differential
%Evolution Algorithm.
%Determines optimal off-spec flowrate for blending during a grade
%transition.
%=========================================================================

cd(fileparts(mfilename('fullpath')));
parentpath = cd(cd('..'));
addpath(genpath(parentpath));

load('kinetics_constants','MM','rho','Conditions','k0','kp','kst','k
t','ast','NTi','Psat');
load('gradeB');

%=========================================================================
%Time Scale
%=========================================================================
t0 = 0;
tf = 8e4;

%=========================================================================
%Feed Rates
%=========================================================================
f0 = {};

%%=========================================================================
% %Grade change AB
%%=========================================================================
% load('gradeA'); %Initial values
%
% %Feed Rates for Grade B
% f0.TEA = 0.127;
% f0.C2 = 48.36;
% f0.C4 = 0.36;
% f0.H2 = 1.71;
% f0.Dil = 33.85;
% f0.Cat = 0.36;
%
% %Upper and Lower Limits for Grade B
% Mn_u = 5.8e4;
% Mn_l = 5.6e4;
%
% PDI_u = 6.75;
% PDI_l = 6.45;
%
%initial = A;
```

```
%=========================================================================
%Grade change BA
%=========================================================================
load('gradeB'); %Initial values

%Feed Rates for Grade A
f0.TEA = 0.129;
f0.C2 = 51.57;
f0.C4 = 0.16;
f0.H2 = 1.89;
f0.Dil = 36.1;
f0.Cat = 0.52;

%Upper and Lower Limits for Grade A
Mn_u = 5.3e4;
Mn_l = 5.1e4;

PDI_u = 5.5;
PDI_l = 5.2;

initial = B;

limits = [PDI_l PDI_u;Mn_l Mn_u];

F0 = [f0.TEA f0.C2 f0.C4 f0.H2 f0.Dil];
F0m = F0.*MM;                                                    %[g.s-1]
F0v = F0m(2:end)./rho/1e3;                                       %[m3.s-1]
f0.Pol = 0;                                                      %[mol.s-1]
tv = [];

%=========================================================================
%New Residence Time Data
%=========================================================================
Conditions.H0 = 80;
H0 = Conditions.H0;
RTD.tau = H0/sum(F0v);                                             %[s]
RTD.age = linspace(0,RTD.tau*10,150);                              %[s]
RTD.I = 1/RTD.tau*exp(-RTD.age/RTD.tau);                         %[s-1]

%=========================================================================
%Run
%=========================================================================
constants = {k0,kp,kst,kt,ast,rho,MM,NTi,H0,Psat,tv};
 [t,y] = ode45(@ReactorSys,[t0
tf],initial,[],RTD,f0,Conditions,constants);
```

```
%=========================================================================
%Time Dependent Solutions
%=========================================================================
mCat = y(:,1)/RTD.tau;                                          %[g.s-1]
FTEA = y(:,2)/RTD.tau;                                         %[mol.s-1]
FC2 = y(:,3)/RTD.tau;                                          %[mol.s-1]
FC4 = y(:,4)/RTD.tau;                                          %[mol.s-1]
FH2 = y(:,5)/RTD.tau;                                          %[mol.s-1]
FPol = y(:,6)/RTD.tau;                                         %[mol.s-1]
Fdil = y(:,7)/RTD.tau;                                         %[mol.s-1]
mu0 = y(:,8);                                                  %[mol.s-1]
mu1 = y(:,9);                                                  %[mol.s-1]
mu2 = y(:,10);                                                 %[mol.s-1]

%=========================================================================
%Off-Spec Polymer Properties (Set O to the desired Off-spec source)
%=========================================================================

%1
O1.mu1 = 5.5e5;
O1.Mn = 5e4;
O1.PDI = 5;

%2
O2.mu1 = 5.5e5;
O2.Mn = 6e4;
O2.PDI = 7;

%3
O3.mu1 = 5.5e5;
O3.Mn = 3e4;
O3.PDI = 6;

% 4
O4.mu1 = 5.5e5;
O4.Mn = 6e4;
O4.PDI = 5;

O = O3;
O.mu0 = O.mu1/O.Mn;
O.mu2 = O.PDI*O.mu1^2/O.mu0;


%=========================================================================
%Differential Evolution Algorithm
%=========================================================================
optimInfo.title = 'Blending Conditions Optimisation';
objFctHandle = @ObjectiveOffSpecMin;
paramDefCell = {'',   [0 500; 0 500; 0 500; 0 500; 0 500; 0 500;],
[1;1;1;1;1;1],[40;20;90;40;30;40]};
objFctParams = [];
```

```
S.mu = [mu0 mu1 mu2];
S.muo = [O.mu0 O.mu1 O.mu2];
S.mP = FPol*MM(2);
S.tau = RTD.tau;
S.t0 = t0;
S.tf = tf;
S.t = t;
S.limits = limits;

objFctSettings = {S};

DEParams = getdefaultparams;
DEParams.NP = 60;
DEParams.VTR = 0;
DEParams.CR = 0.7;
DEParams.F = 0.5;
DEParams.feedSlaveProc = 1;
DEParams.maxiter       = 100;
DEParams.maxtime       = 1e6;                                    %[s]
DEParams.maxclock      = [];
DEParams.refreshiter   = 1;
DEParams.refreshtime   = 10;                                     %[s]
DEParams.refreshtime2  = 20;                                     %[s]
DEParams.refreshtime3  = 40;                                     %[s]
DEParams.strategy = 1;

emailParams = [];

[bestmem, bestval, bestFctParams] = ...
    differentialevolution(DEParams, paramDefCell, objFctHandle, ...
    objFctSettings, objFctParams, emailParams, optimInfo);

disp(' ');
disp('Best parameter set returned by function
differentialevolution:');
disp(bestFctParams);
```

```
%=========================================================================
%Thaabit Nacerodien
%MSc Chemical Engineering
%University of Cape Town
%2013
%Filename: ObjectiveOffSpecMin.m
%Description: Function file to evaluate objective function for
%dynamic optimisation method described in Section 5.2.
%=========================================================================

function val = ObjectiveOffSpecMin(S,CV)

t0 = S.t0;
tf = S.tf;
t = S.t;
muo = S.muo;
mu = S.mu;
mP = S.mP;

%=========================================================================
%Upper and Lower Limits for Grade
%=========================================================================

PDI_u = S.limits(1,2);
PDI_l = S.limits(1,1);

Mn_u = S.limits(2,2);
Mn_l = S.limits(2,1);

M0 = 0.25*mP(end)*S.tau;
tO = t;
tR = t;
muR = mu;
muO = muo;
mR = mP;

%=========================================================================
%Blending unit dynamics without blending occuring
%=========================================================================

mOs = 0;
initial = {mOs;muO;tO;mR;muR;tR;M0};
[ts,ys] = ode45(@BlendSys,[t0 tf],[M0*0.1 muR(1,:)],[],initial);
mus0 = ys(:,2);
mus1 = ys(:,3);
mus2 = ys(:,4);
s = ChainProp([mus0 mus1 mus2]);
Mns = s.Mn;
PDIs = s.PDI;
```

```
%=========================================================================
%Scaling the off-spec flow profile
%=========================================================================

mO = zeros(length(ts),1);
MnOn = (Mns <= Mn_u & Mns >= Mn_l);
PDIOn = (PDIs <= PDI_u & PDIs >= PDI_l);

On = (MnOn&PDIOn);
Off = length(On(On<1));

mOff = [CV;CV(end)];
tOff = linspace(t0,ts(Off),length(mOff));

for i = 1:Off
    temp = mOff(ts(i)>=tOff);
    mO(i,1) = temp(end);
end


%=========================================================================
%Blending unit dynamics with blending
%=========================================================================

initial = {mO;muO;ts;mR;muR;tR;M0};
[tBl,yBl] = ode45(@BlendSys,[t0 tf],[M0*0.1 muR(1,:)],[],initial);

for i = 1:length(tBl)
    temp1 = mO(tBl(i)>=ts);
    temp2 = mR(tBl(i)>=tR);
    mOv(i,1) = temp1(end);
    mRv(i,1) = temp2(end);
end

tau = M0./(mOv + mRv);
mBl = yBl(:,1)./tau;
muD0 = yBl(:,2);
muD1 = yBl(:,3);
muD2 = yBl(:,4);


D = ChainProp([muD0 muD1 muD2]);
MnBl = D.Mn;
PDIBl = D.PDI;

%=========================================================================
%On-spec and Off-spec
%=========================================================================
Mn_On = (MnBl <= Mn_u & MnBl >= Mn_l);
PDI_On = (PDIBl <= PDI_u & PDIBl >= PDI_l);

OnSpec = (Mn_On&PDI_On);
OffSpec = logical(1-OnSpec);

mOffSpec = mBl.*OffSpec;
val = nint(tBl,mOffSpec,1);
```

v. Unsteady-state blending during optimal grade transition

```
%=========================================================================
%Thaabit Nacerodien
%MSc Chemical Engineering
%University of Cape Town
%2013
%Filename: predictFeedRates.m
%Description: Main file for dynamic optimisation using DEA
%Control vectors for Hydrogen, Monomer Ethylene and Off-spec Polymer
%Flows
%=========================================================================
cd(fileparts(mfilename('fullpath')));
parentpath = cd(cd('..'));
addpath(genpath(parentpath));

load('kinetics_constants','MM','rho','Conditions','k0','kp','kst','k
t','ast','NTi','Psat');

%%=========================================================================
% %Grade change AB
%%=========================================================================
% load('gradeA'); %Initial values
%
% %Feed Rates for Grade B
% f0.TEA = 0.127;
% f0.C2 = 48.36;
% f0.C4 = 0.36;
% f0.H2 = 1.71;
% f0.Dil = 33.85;
% f0.Cat = 0.36;
%
% %Upper and Lower Limits for Grade B
% Mn_u = 5.8e4;
% Mn_l = 5.6e4;
%
% PDI_u = 6.75;
% PDI_l = 6.45;

% %Absolute Desired properties for Grade B
% MnD = 5.67e4;
% PDID = 6.58;
%
%initial = A;

%=========================================================================
%Grade change BA
%=========================================================================
load('gradeB'); %Initial values

%Feed Rates for Grade A
f0.TEA = 0.129;
f0.C2 = 51.57;
f0.C4 = 0.16;
f0.H2 = 1.89;
f0.Dil = 36.1;
f0.Cat = 0.52;
```

```
%Upper and Lower Limits for Grade A
Mn_u = 5.3e4;
Mn_l = 5.1e4;

PDI_u = 5.5;
PDI_l = 5.2;

%Absolute Desired properties for Grade B
MnD = 5.2e4;
PDID = 5.34;

initial = B;

limits = [PDI_l PDI_u;Mn_l Mn_u];

%=========================================================================
%Time Scale
%=========================================================================
t0 = 0;
tf = 8e4;

Conditions.H0 = 80;
H0 = Conditions.H0;
tv = [];
constants = {k0,kp,kst,kt,ast,rho,MM,NTi,H0,Psat,tv};

%=========================================================================
%Differential Evolution Algorithm
%=========================================================================
C2.alpha = 0.9;
C2.u = 54.78;
C2.l = 45.15;

H2.alpha = 0.9;
H2.u = 2.07;
H2.l = 1.53;

for i = 1:5
    C2.au(i,1) = fi.C2 + C2.alpha^i*(C2.u - C2.l)/2;
    C2.al(i,1) = fi.C2 - C2.alpha^i*(C2.u - C2.l)/2;
    H2.au(i,1) = fi.H2 + H2.alpha^i*(H2.u - H2.l)/2;
    H2.al(i,1) = fi.H2 - H2.alpha^i*(H2.u - H2.l)/2;
end

C2.n = (C2.au - C2.al)/100;
H2.n = (H2.au - H2.al)/100;

optimInfo.title = 'Grade Transition Optimisation';
objFctHandle = @ObjectiveFeed;
paramDefCell = {'C2',        [C2.al C2.au], C2.n,
[fi.C2;fi.C2;fi.C2;fi.C2;fi.C2];
                'H2',        [H2.al H2.au], H2.n,
[fi.H2;fi.H2;fi.H2;fi.H2;fi.H2]};
objFctParams = [];
```

```
S.initial = initial;
S.t0 = t0;
S.tf = tf;
S.limits = limits;
S.f0 = f0;
S.constants = constants;
S.conditions = Conditions;
S.fi = fi;
S.MnD = MnD;
D.PDID = PDID;

objFctSettings = {S};

DEParams = getdefaultparams;
DEParams.NP = 100;
DEParams.VTR = 0;
DEParams.CR = 0.8;
DEParams.F = 0.7;
DEParams.feedSlaveProc = 1;
DEParams.maxiter       = 100;
DEParams.maxtime       = 1e6;                                    %[s]
DEParams.maxclock      = [];
DEParams.refreshiter   = 1;
DEParams.refreshtime   = 10;                                     %[s]
DEParams.refreshtime2  = 20;                                     %[s]
DEParams.refreshtime3  = 40;                                     %[s]

emailParams = [];

[bestmem, bestval, bestFctParams] = ...
    differentialevolution(DEParams, paramDefCell, objFctHandle, ...
    objFctSettings, objFctParams, emailParams, optimInfo);

disp(' ');
disp('Best parameter set returned by function
differentialevolution:');
disp(bestFctParams);
```

```
%=========================================================================
%Thaabit Nacerodien
%MSc Chemical Engineering
%University of Cape Town
%2013
%Filename: ObjectiveFeed.m
%Description: Function file to evaluate objective function for
%dynamic optimisation of grade transition described in Section 5.3.
%=========================================================================

function val = ObjectiveFeed(S,CV)

rho = S.constants{6};
MM = S.constants{7};
H0 = S.constants{9};

t0 = S.t0;
tf = S.tf;

fi = S.fi;
%=========================================================================
%New Feed Rates
%=========================================================================
f0 = S.f0;
f0.C2 = CV.C2;
f0.C2(end+1) = fi.C2;
f0.H2 = CV.H2;
f0.H2(end+1) = fi.H2;

F0 = [f0.TEA f0.C2(end) f0.C4 f0.H2(end) f0.Dil];
F0m = F0.*MM;                                                    %[g.s-1]
F0v = F0m(2:end)./rho/1e3;                                      %[m3.s-1]
f0.Pol = 0;                                                    %[mol.s-1]
tv = linspace(t0,tf,length(f0.C2)+1);
S.constants{11} = tv;

%=========================================================================
%New Residence Time Data
%=========================================================================
RTD.tau = H0/sum(F0v);                                             %[s]
RTD.age = linspace(0,RTD.tau*10,150);                              %[s]
RTD.I = 1/RTD.tau*exp(-RTD.age/RTD.tau);                         %[s-1]

%=========================================================================
%Run
%=========================================================================
[t,y] = ode45(@ReactorSys,[t0
tf],S.initial,[],RTD,f0,S.conditions,S.constants);
```

```
%=========================================================================
%Time Dependent Solutions
%=========================================================================
FPol = y(:,6)/RTD.tau;                                          %[mol.s-1]
mu0 = y(:,8);                                                   %[mol.s-1]
mu1 = y(:,9);                                                   %[mol.s-1]
mu2 = y(:,10);                                                  %[mol.s-1]

mu = [mu0 mu1 mu2];
mP = FPol*MM(2);

%=========================================================================
%Upper and Lower Limits for Grade
%=========================================================================

PDI_u = S.limits(1,2);
PDI_l = S.limits(1,1);

Mn_u = S.limits(2,2);
Mn_l = S.limits(2,1);

prop = ChainProp([mu(:,1) mu(:,2) mu(:,3)]);
Mn = prop.Mn;
PDI = prop.PDI;

%=========================================================================
%On-spec and Off-spec
%=========================================================================
Mn_On = (Mn <= Mn_u & Mn >= Mn_l);
PDI_On = (PDI <= PDI_u & PDI >= PDI_l);

OnSpec = (Mn_On&PDI_On);
OffSpec = logical(1-OnSpec);

val = nint(t,(1-Mn/S.MnD).^2 + (1-PDI/S.PDID).^2,1);


%=========================================================================
%Thaabit Nacerodien
%MSc Chemical Engineering
%University of Cape Town
%2013
%Filename: PredictOffSpec.m
%Description: Main file for dynamic optimisation of off-spec
%blending during grade transition optimised in PredictFeed.m
%=========================================================================

cd(fileparts(mfilename('fullpath')));
parentpath = cd(cd('..'));
addpath(genpath(parentpath));

load('kinetics_constants','MM','rho','Conditions','k0','kp','kst','k
t','ast','NTi','Psat');
```

```
%=========================================================================
%Time Scale
%=========================================================================
t0 = 0;
tf = 8e4;

%=========================================================================
%Grade change AB
%=========================================================================
load('gradeA'); %Initial values
load('ABFeed'); %Optimised Profile

%Feed Rates for Grade B
f0.TEA = 0.127;
fi.C2 = 48.36;
f0.C4 = 0.36;
fi.H2 = 1.71;
f0.Dil = 33.85;
f0.Cat = 0.36;

%Upper and Lower Limits for Grade B
Mn_u = 5.8e4;
Mn_l = 5.6e4;

PDI_u = 6.75;
PDI_l = 6.45;

initial = A;

%%=========================================================================
% %Grade change BA
%%=========================================================================
% load('gradeB'); %Initial values
% load('BAFeed'); %Optimised Profile
%
% %Feed Rates for Grade A
% f0.TEA = 0.129;
% fi.C2 = 51.57;
% f0.C4 = 0.16;
% fi.H2 = 1.89;
% f0.Dil = 36.1;
% f0.Cat = 0.52;
%
% %Upper and Lower Limits for Grade A
% Mn_u = 5.3e4;
% Mn_l = 5.1e4;
%
% PDI_u = 5.5;
% PDI_l = 5.2;

% initial = B;

limits = [PDI_l PDI_u;Mn_l Mn_u];
```

```
f0.C2 = [C2;fi.C2];
f0.H2 = [H2;fi.H2];
F0 = [f0.TEA fi.C2 f0.C4 fi.H2 f0.Dil];
F0m = F0.*MM;                                                    %[g.s-1]
F0v = F0m(2:end)./rho/1e3;                                       %[m3.s-1]
f0.Pol = 0;                                                      %[mol.s-1]
tv = linspace(t0,tf,length(f0.C2)+1);

%======================================================================
%New Residence Time Data
%======================================================================
Conditions.H0 = 80;
H0 = Conditions.H0;
RTD.tau = H0/sum(F0v);                                             %[s]
RTD.age = linspace(0,RTD.tau*10,150);                              %[s]
RTD.I = 1/RTD.tau*exp(-RTD.age/RTD.tau);                         %[s-1]

%======================================================================
%Run
%======================================================================
constants = {k0,kp,kst,kt,ast,rho,MM,NTi,H0,Psat,tv};
[t,y] = ode45(@ReactorSys,[t0
tf],initial,[],RTD,f0,Conditions,constants);

%======================================================================
%Time Dependent Solutions
%======================================================================
mCat = y(:,1)/RTD.tau;                                           %[g.s-1]
FTEA = y(:,2)/RTD.tau;                                         %[mol.s-1]
FC2 = y(:,3)/RTD.tau;                                          %[mol.s-1]
FC4 = y(:,4)/RTD.tau;                                          %[mol.s-1]
FH2 = y(:,5)/RTD.tau;                                          %[mol.s-1]
FPol = y(:,6)/RTD.tau;                                         %[mol.s-1]
Fdil = y(:,7)/RTD.tau;                                         %[mol.s-1]
mu0 = y(:,8);                                                  %[mol.s-1]
mu1 = y(:,9);                                                  %[mol.s-1]
mu2 = y(:,10);                                                 %[mol.s-1]
```

```
%=========================================================================
%Off-Spec Polymer Properties (Set O to the desired Off-spec source)
%=========================================================================

%1
O1.mu1 = 5.5e5;
O1.Mn = 5e4;
O1.PDI = 5;

%2
O2.mu1 = 5.5e5;
O2.Mn = 6e4;
O2.PDI = 7;

%3
O3.mu1 = 5.5e5;
O3.Mn = 3e4;
O3.PDI = 6;

% 4
O4.mu1 = 5.5e5;
O4.Mn = 6e4;
O4.PDI = 5;

 O = O3;
O.mu0 = O.mu1/O.Mn;
O.mu2 = O.PDI*O.mu1^2/O.mu0;

%=========================================================================
%Differential Evolution Algorithm
%=========================================================================
optimInfo.title = 'Blending Conditions Optimisation';
objFctHandle = @ObjectiveOffSpec;
paramDefCell = {'',   [0 500; 0 500; 0 500; 0 500; 0 500; 0 500],
[1;1;1;1;1;1],[0;0.1;1.8;0;1.2;0.7]};
objFctParams = [];

S.mu = [mu0 mu1 mu2];
S.muo = [O.mu0 O.mu1 O.mu2];
S.mP = FPol*MM(2);
S.t0 = t0;
S.tf = tf;
S.t = t;
S.limits = limits;
S.tau = RTD.tau;

objFctSettings = {S};
```

```
DEParams = getdefaultparams;
DEParams.NP = 60;
DEParams.VTR = 0;
DEParams.CR = 0.7;
DEParams.F = 0.5;
DEParams.feedSlaveProc = 1;
DEParams.maxiter       = 1000;
DEParams.maxtime       = 1e6;                                    %[s]
DEParams.maxclock      = [];
DEParams.refreshiter   = 1;
DEParams.refreshtime   = 10;                                     %[s]
DEParams.refreshtime2  = 20;                                     %[s]
DEParams.refreshtime3  = 40;                                     %[s]
DEParams.strategy = 1;

emailParams = [];

[bestmem, bestval, bestFctParams] = ...
    differentialevolution(DEParams, paramDefCell, objFctHandle, ...
    objFctSettings, objFctParams, emailParams, optimInfo);

disp(' ');
disp('Best parameter set returned by function
differentialevolution:');
disp(bestFctParams);

%=========================================================================
%Thaabit Nacerodien
%MSc Chemical Engineering
%University of Cape Town
%2013
%Filename: ObjectiveOffSpec.m
%Description: Function file to evaluate objective function for
%dynamic optimisation of off-spec blending described in Section 5.3.
%=========================================================================

function val = ObjectiveOffSpec(S,CV)

t0 = S.t0;
tf = S.tf;
t = S.t;
muo = S.muo;
mu = S.mu;
mP = S.mP;
RTD.tau = S.tau;
```

```
%=========================================================================
%Upper and Lower Limits for Grade
%=========================================================================

PDI_u = S.limits(1,2);
PDI_l = S.limits(1,1);

Mn_u = S.limits(2,2);
Mn_l = S.limits(2,1);

M0 = 0.25*mP(end)*RTD.tau;
tO = t;
tR = t;
muR = mu;
muO = muo;
mR = mP;

%=========================================================================
%Blending unit dynamics without blending occuring
%=========================================================================

mOs = 0;
initial = {mOs;muO;tO;mR;muR;tR;M0};
[ts,ys] = ode45(@BlendSys,[t0 tf],[M0*0.1 muR(1,:)],[],initial);
mus0 = ys(:,2);
mus1 = ys(:,3);
mus2 = ys(:,4);
s = ChainProp([mus0 mus1 mus2]);
Mns = s.Mn;
PDIs = s.PDI;

%=========================================================================
%Scaling the off-spec flow profile
%=========================================================================

mO = zeros(length(ts),1);
MnOn = (Mns <= Mn_u & Mns >= Mn_l);
PDIOn = (PDIs <= PDI_u & PDIs >= PDI_l);

On = (MnOn&PDIOn);
Off = length(On(On<1));

mOff = CV;
mOff(end+1) = CV(end);
tz = ts(Off);

n = length(mOff);
tv = linspace(t0,tz,n);

for i = 1:Off
   temp = mOff(ts(i)>=tv);
   mO(i,1) = temp(end);
end
```

```
%=========================================================================
%Blending unit dynamics with blending
%=========================================================================

initial = {mO;muO;ts;mR;muR;tR;M0};
[tBl,yBl] = ode45(@BlendSys,[t0 tf],[M0*0.1 muR(1,:)],[],initial);

for i = 1:length(tBl)
    temp1 = mO(tBl(i)>=ts);
    temp2 = mR(tBl(i)>=tR);
    mOv(i,1) = temp1(end);
    mRv(i,1) = temp2(end);
end

tau = M0./(mOv + mRv);
mBl = yBl(:,1)./tau;
muD0 = yBl(:,2);
muD1 = yBl(:,3);
muD2 = yBl(:,4);

D = ChainProp([muD0 muD1 muD2]);
MnBl = D.Mn;
PDIBl = D.PDI;

%=========================================================================
%On-spec and Off-spec
%=========================================================================
Mn_On = (MnBl <= Mn_u & MnBl >= Mn_l);
PDI_On = (PDIBl <= PDI_u & PDIBl >= PDI_l);

OnSpec = (Mn_On&PDI_On);
OffSpec = logical(1-OnSpec);

val = nint(tBl,mBl.*OffSpec,1);
```

The source code for the differential evolution algorithm used can be downloaded at the following link:

http://www.mathworks.com/matlabcentral/fileexchange/18593-differential-evolution