Proceedings of the 9th International Symposium on
Dynamics and Control of Process Systems (DYCOPS 2010),
Leuven, Belgium, July 5-7, 2010
Mayuresh Kothare, Moses Tade, Alain Vande Wouwer, Ilse Smets (Eds.)

Modelica Based Grade Change Optimization for a Polyethylene Reactor

Per-Ola Larsson [*] **Niklas Andersson** [**] **Johan Åkesson** [*]
Staffan Haugwitz [***]

[*] *Department of Automatic Control, Lund University, Lund, Sweden*
(e-mail: Per-Ola.Larsson|Johan.Akesson@control.lth.se)
[**] *Department of Chemical Engineering, Lund University, Lund, Sweden*
(e-mail: Niklas.Andersson@chemeng.lth.se)
[***] *Borealis AB, Stenungsund, Sweden.*
(e-mail: Staffan.Haugwitz@borealisgroup.com)

Abstract: This paper presents a dynamic optimization procedure of grade changes of polyethylene production. The optimization is built upon a novel modular Modelica library containing e.g., non-linear DAE models for polyethylene reactors based on models currently used in non-linear MPC of industrial reactors at Borealis AB. Using Optimica, which extends the Modelica language with constructs for optimization problems, and JModelica.org, a novel framework to translate such optimization problems into NLP problems, grade transition optimization problems can be solved. The solution procedure and a transition example with optimal inputs and outputs are given in the paper showing promising results.

Keywords: optimization, Modelica, grade change, chemical industry, polymerization

1. INTRODUCTION

In the chemical industry today, as in many other industries, there exists a problem of increasing the capital productivity while the market competition has increased the last decades. A way to overcome this is by continuously developing and increasing the profitability of the manufacturing process. The market today is highly varying, both in raw material pricing but also in terms of demand of produced material. It is therefore important that manufacturers can respond rapidly to these changes and thereby increase their profit. In the polymer industry, this is the case where manufacturers are forced to switch between different polymer grades to suit market demand, that is, running product campaigns on the same plant. The campaign time can vary between a few days up to weeks. It is therefore of importance that the transitions are made in a way such that off-specification polymer, i.e., polymer that does not fulfill grade specification of the former grade nor the grade to be, is minimized. However, also pricing of raw material and time, i.e., economics, must be included in the transition cost, see e.g., van Brempt et al. (2004). Additionally, also the production rate can be used in a grade change, both for storage control and market adaptation. For instance, if time is not of essence, one may lower production rate at a transition and thus producing less off-specification material. All the above properties make a grade change an intricate optimization problem to pose and to solve.

Not only is a single transition important, but also planning of several grade changes must be comprised such that the order of the grade changes minimizes the total loss, see e.g. Prata et al. (2008). Thus, a tool for production planning at a higher level than automation and control is needed.

In this paper, a prototype for such a tool is presented. Using the Modelica language, which has received attention in industry during the last decade, a novel model library has been constructed. The reactor models, described by non-linear differential algebraic equations (DAE), are currently used for non-linear Model Predictive Control (MPC) of industrial polyethylene reactors at Borealis AB. Using the constructed library, an example of a grade transition for a polyethylene reactor model will be optimized. A key contribution of the paper is the formulation of the optimization problem in the high level language Optimica using the framework of JModelica.org, an open source project targeted towards dynamic optimization. The foundation of the optimization solution procedure is the constructed Modelica library, giving simulation, optimization, and verification models.

The paper is organized as follows. Section 2 describes the Borstar® process, key parameters and how grade transitions are performed in practice. Section 3 concerns the modeling languages and optimization tool used when solving the grade transition problem while Section 4 gives an overview of the mathematical reactor model. Further, Section 5 describes the Modelica library constructed and Section 6 shows the optimization formulation and solution of a grade transition problem. Finally, summary and future work are given in sections 7 and 8.

2. BORSTAR PE

2.1 Polyethylene

Polyethylene belongs to the family polyolefins and its properties vary much with the molecular weight. It is

[⋆] Sponsored by the Swedish Foundation of Strategic Research in the framework of Process Industry Centre at Lund University (PICLU).

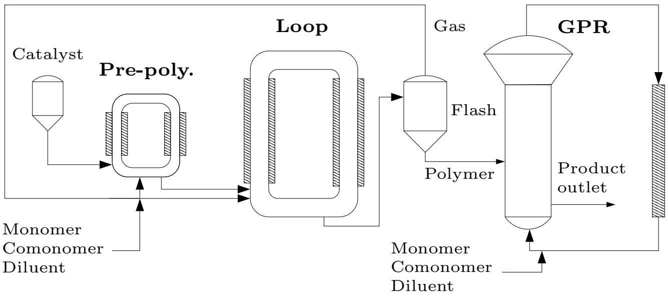

Fig. 1. Reactor chain of a Borstar® process: Pre-polymerization, Loop, and Gas phase reactor (GPR).

formed by the polymerization of the monomer ethylene and the termination of polymer chains is controlled by the ratio between hydrogen and ethylene. Comonomers are added to control, e.g., density. The reaction used in this paper is catalyzed using a Ziegler-Natta catalyst with the advantage that the polymerization can operate at low temperatures and pressures. By combining polymer with low molecular weight and high molecular weight, a bimodal molecular weight distribution can be achieved. This gives the polymer good resistance and mechanical properties and at the same time good processability.

2.2 The Borstar® Process

Bimodal polyethylene products are polymerized in cascaded reactors and the Borstar® process consists of at least three reactors, see Figure 1. The purpose of the pre-polymerization reactor is to let the first part of the polymerization on the surface of the catalyst particles to be slow since a fast reaction may damage the particles. The diluent is propane, which makes it possible to operate above the critical thermodynamic point. This gives very low solubility and the risk of fouling is decreased. The first peak of the bimodal molecular weight distribution is mainly formed in the loop reactor. In the subsequent gas phase reactor (GPR), i.e., a fluidized bed reactor, the second peak is formed with higher molecular weight, van Brempt et al. (2004).

2.3 Key parameters

There exists a couple of key parameters that define a grade specification. First, the split is the ratio of polymer formed in the fluidized bed to the total polymer formed and gives a measure of the distribution of the two bimodal peaks.

A common way to estimate the molecular weight is by the melt flow indices, defined as the weight of polymer extruded over a specified time interval and temperature through an extrusion die with standard equipment, see Fried (2003). When characterizing bimodal polyethylene, a second key parameter is the melt flow ratio (MFR), defined as the ratio between melt flow indexes of two different gravimetric weights.

A third important key parameter is the density that must be kept within a certain region to be on specification. Other parameters, which are harder to measure, are odour and taste.

2.4 Current Grade Change Practice

At current time, to go from one grade to another, operators follow a recipe derived from earlier experiences which may not be the optimal path. Input flows and control references are often changed by operators in a discrete manner, i.e., steps, and the performance of the grade changes may vary.

Production planning, i.e., planning of several grade changes ahead at higher production level, is often made in an *ad hoc* manner using process know-how and experience of former grade changes.

Hence, both at single transitions, but also at the production planning stage, a tool for grade transition optimization is needed.

3. MODELING LANGUAGES AND TOOLS

3.1 Modelica and Optimica

The modeling language used to express the mathematical model is Modelica which is a high-level language for encoding of complex physical systems, supporting object oriented concepts such as classes, components and inheritance. Also, text-book style declarative equations can be expressed as well as acausal component connections representing physical interfaces. This modeling paradigm has significant advantages over the block-based paradigm in the context of physical modeling. In particular, acausal modeling systems do not require the user to solve for the derivatives of a mathematical model. Instead, differential and algebraic equations may be mixed, which then typically results in a differential algebraic equation (DAE). Modelica also targets modeling of heterogeneous systems, where components from different physical domains need to be included in the same model. This situation is common in realistic applications where e.g., the reaction kinetics in a reactor, the thermal properties of the cooling system, and the electrical and mechanical properties of the mass transportation systems need to be investigated. In this respect, Modelica differs from other modeling systems such as gPROMS and VHDL-AML, see Process Systems Enterprise (2007) and IEEE (1997), which are targeting the chemical and electrical domains, respectively.

While Modelica offers strong support for modeling of physical systems, the language lacks important constructs needed when formulating dynamic optimization problems, notably cost functions, constraints, and a mechanism to select inputs and parameters to optimize. In order to strengthen the optimization capabilities of Modelica, the Optimica extension has been proposed, see Åkesson (2008). Optimica adds to Modelica a small number of constructs, enabling the user to conveniently specify dynamic optimization problems based on Modelica models.

In the context of dynamic optimization, the use of high-level description formats is particularly attractive, since the interfaces of algorithms for solution of such programs are typically written in C or FORTRAN. Implementing the optimization formulation for such an algorithm may require a significant effort. In addition, once finalized, the implementation is typically difficult to reuse with another algorithm.

3.2 JModelica.org

JModelica.org is a novel Modelica-based open source project targeted at dynamic optimization, see Åkesson et al. (2009). JModelica.org features compilers supporting code generation of Modelica/Optimica models to C, a C API for evaluating model equations and their derivatives and optimization algorithms. The compilers and the model C API has also been interfaced with Python, see for instance Python Software Foundation (2009), in order to enable scripting and custom application development. In order to support formulation of dynamic optimization of Modelica models, JModelica.org supports the Optimica extension.

The JModelica.org platform contains an implementation of a simultaneous optimization method based on orthogonal collocation on finite elements, Biegler et al. (2002). Using this method, state and input profiles are parametrized by Lagrange polynomials of order three and four, respectively, based on Radau points. This method corresponds to a fully implicit Runge-Kutta method, and accordingly it possesses well known and strong stability properties. By parametrizing the variable profiles by polynomials, the dynamic optimization problem is translated into a non-linear programming (NLP) problem which may be solved by a numerical NLP solver. This NLP is, however, very large. In order to efficiently find a solution to the NLP, derivative information as well as the sparsity patterns of the constraint Jacobians need to be provided to the solver. The simultaneous optimization algorithm has been interfaced with the large-scale NLP solver IPOPT, Wächter and Biegler (2006), which has been developed particularly to solved NLP problems arising in simultaneous dynamic optimization methods.

The choice of a simultaneous optimization algorithm fits well with the properties of the dynamic optimization problems treated in this paper and has been used also in e.g. Flores-Tlacuahuac et al. (2006). In particular, simultaneous methods handle highly non-linear systems well, and also, state and input inequality constraints are easily incorporated.

4. MATHEMATICAL PLANT MODEL

Modeling a series of reactors, as in Figure 1, is a task including theoretical and empirical challenges. A resulting model of such work at Borealis AB for the Borstar® process is today used on-site in a non-linear Model Predictive Control (MPC) software, OnSpot, see e.g. Saarinen and Andersen (2003). Due to confidentiality reasons, the full model can not be shown, however, the structure is presented.

Inputs used in the model are for instance flows of ethylene, hydrogen and propane, but also comonomer flows and catalyst flows together with its specific properties. Several outputs are available including substance masses, mass ratios, mass flows, concentrations, MFR, density, production rates and split factor.

Each reactor is modelled from material balances of the general form

$$\dot{m} = \sum_j q_j^{\text{in}} - \sum_k q_k^{\text{out}} + r, \tag{1}$$

where the inflows q_j^{in} can be either from previous reactor or a side feed, and outflows q_k^{out} that go to the subsequent reactor, a bleed, recycle, or product outlet. The model is based on a catalyst activity profile along the reactors and include polymerization and termination rates r from monomer, comonomers and hydrogen.

The reaction kinetics are modelled with extended Arrhenius expressions, as

$$\dot{r} = K_1(x)e^{K_2(x)}, \tag{2}$$

and is dependent on the state of the reactor, x, in the form of concentrations, pressures, and temperatures of the reactants. The kinetics functions K_1 and K_2 are non-linear in the state x.

The state of the Ziegler-Natta catalyst is of major impact on the reactor performance and must hence be well modelled. The catalyst properties, c, which is a part of the reactor state x, is dependent on reaction rates but also on the reactor state itself, i.e.,

$$\dot{c} = K_3(x, r), \tag{3}$$

where K_3 is a non-linear function.

Densities must be modelled and since it is possible to operate above the critical thermodynamic point, this is a major task. Amongst others, empirical non-linear expressions dependent on the reactor state x are used, i.e.,

$$\rho = K_4(x) \tag{4}$$

The model contains, apart from the equations in Eq. (1)–(4) additional algebraic equations. If the inputs and outputs of the model are denoted u and y, respectively, and the algebraic variables are denoted w, the model can be written in the general non-linear differential algebraic form

$$\begin{aligned} 0 &= F(\dot{x}, x, w, u) \\ y &= g(x, w, u), \end{aligned} \tag{5}$$

which will be used in the optimization problems.

To simplify modeling, some assumptions are made on the control system of the reactors. Firstly, the heat produced by the exothermic reaction can be handled by the cooling system. Secondly, the reactor pressures are controlled by outlet valves holding the pressures constant.

Moreover, it is also assumed that the polymer and the gases/fluids are well mixed and the temperatures are uniform in the reactors.

5. MODELICA MODEL LIBRARY

The novel Modelica library, which is the core of a grade transition tool, includes the following sub-packages,

Connectors define how the physical interface between one reactor or input source to another reactor or output sink is modelled. A connector holds both flow variables, such as gas/fluid or polymer flows, but also potential variables such as mass ratios of the flows and catalyst properties. There are two types of connectors in the library, one for pure gas/fluid flow and one for both polymer and gas/fluid flow combined.

Interfaces is a package that defines input and output structure for different reactor types by using the defined connectors. Note that the interface models do not contain any equations concerning reactor models, only the interfaces are defined.

ReactorModels contains the mathematical descriptions, i.e., non-linear DAEs, of e.g., the reactors in the reactor chain in Figure 1. The models are extended from the interface package, giving it a predefined input-output structure. This gives the possibility to have more than one model for a certain reactor, but at the same time have an identical interface. This is very useful when different model fidelities are desired, e.g., for optimization and verification of results.

Source/Sink withhold sources and a sink. One source for pure gas/fluid and one for gas/fluid and catalyst and a sink retrieving both gas/fluid and polymer. They use the defined connectors such that they can be interfaced with the reactors.

Templates package holds different templates of setups, i.e., definition of reactor setups to be used later in experiments and optimization. Note here that the template models only defines the structure of a setup, i.e., replaceable parts retrieved from the Interfaces package, not actual mathematical models.

Experiments withholds models that extend from the template models. Into the experiment models different reactor models can be inserted from the ReactorModels package. Note the modularity due to ease of exchanging reactor models, e.g., model fidelities, while maintaining the same over all structure.

Optimization withholds models extended from the Template package where different reactor models have been inserted, analogous to the Experiment package. These models are used by the JModelica.org framework when solving for stationary points of the reactor chain corresponding to a grade or when solving the dynamic optimization problem of finding the optimal trajectories of a grade transition problem.

Constants contains physical constants, both theoretical and empirical, and is used by the reactor models in the ReactorModels package.

6. OPTIMAL GRADE TRANSITION EXAMPLE

In this example we will consider a grade transition of the first reactor. The ReactorPackage described in the previous section is used to set up the models to be subjected to optimization while the construction of cost function, constraints and optimization options will be encoded using Optimica constructs. The model is then passed on to the JModelica.org framework obtaining the optimal grade transition trajectories.

6.1 Grade Transition Definition

The grade transition considered will include three main objectives,

(1) Increase production rate by 20%.
(2) Increase hydrogen-ethylene ratio by 20%.
(3) Keep amount of polymer in reactor within bounds during grade transition.

It is known that a change in the ethylene-hydrogen ratio is well correlated with the melt-flow index, one of the key parameters. The constraint on amount of polymer in the reactor is due to, besides volume constraint of reactor, also avoidance of plugging the reactor.

The outputs used in the optimization problem are a subset of the model output y. Specifically, introduce the following notation for the controlled outputs,

$$
\begin{aligned}
y_r &= \text{production rate of polymer in the reactor}\\
y_{he} &= \text{ratio of hydrogen and ethylene.}\\
y_s &= \text{amount of polymer in the reactor}
\end{aligned}
$$

Table 1. Normalized grade definitions/recipe.

	Grade 1	Grade 2
y_s	1	1
y_r	1	1.2
y_{he}	1	1.2

The two grades, Grade 1 and Grade 2, can now be defined using Grade 1 as a normalizer, see Table 1.

As the optimization problem is posed on the Prepolymerization reactor, there is no polymer inflow to the reactor. Besides the catalyst inflow that is considered constant, three inflows are available for control, i.e.,

$$
\begin{aligned}
u_p &= \text{propane input flow}\\
u_e &= \text{ethylene input flow}\\
u_h &= \text{hydrogen input flow.}
\end{aligned}
$$

In the dynamic optimization problem, these three inputs are the decision variables and constraint on their derivatives will be set due to limitations on e.g., pumps, external control loops and safety precautions.

6.2 Finding Stationary Points

Prior to and after the grade transition, the system is in steady state representing on-specification production and correct production rate. Hence, at start and final time, $t = t_0$ and $t = t_f$, the system fulfills the static equations

$$
\begin{aligned}
0 &= F(0, x^\circ, w^\circ, u^\circ)\\
y^\circ &= g(x^\circ, w^\circ, u^\circ),
\end{aligned}
\tag{6}
$$

where y and u are the considered inputs and outputs,

$$
u = [u_p \; u_e \; u_h]^T, \quad y = [y_s \; y_r \; y_{he}]^T,
$$

and x represents the reactor internal states. Note that the two stationary points will give different solutions (x°, w°, u°, y°) to the non-linear algebraic equation above, where superscript $^\circ$ indicates constant value.

The two different grades concerned in the transition are only defined by the output y as specified in the recipe in Table 1. The first step in the grade change solution procedure is to find the two stationary points corresponding to the two grades in terms of reactor states x and input flows u. This is achieved using a constructed model in the optimization package where the system inflows are free variables. A grade is defined by setting the outputs as fixed and all state derivatives to 0 in the model while other variables such as e.g., outflows and mass ratios are free and are given reasonable initial values for the optimization problem. The constructed model is utilized by the JModelica.org framework, which compiles the model and casts the system in Eq. (6) into an optimization problem. The resulting NLP problem contains approximately around 200 variables and is solved rapidly.

By initializing the model with different grades, the stationary points corresponding to Table 1 are calculated. Note that not only are the inflows corresponding to the two grades given as a result of the optimization problem, but also the full state vector of the reactor is given and can be used as references in the dynamic optimization problem later on. However, in this paper only the output y and input u will be used.

6.3 Dynamic Optimization of Grade Transition

In the dynamic optimization problem, the optimal transition trajectories between the two grades are found. In the previous section, start and end points were calculated. Hence, there is access to both initial values and reference values of all variables of the system.

As mentioned in Section 6.1, constraints on the input flows must be set. Constraints on the outputs are also set, since it is desired to reach Grade 2 in a pre-specified way. In this example, it is desired that the production rate y_r does not have an overshoot, while the amount of polymer in the reactor y_s does not vary more than $\pm 5\%$ compared to initial value during the transition.

A quadratic cost function is constructed that includes deviations from Grade 2 specifications expressed in outputs and inputs solved for in the previous section. It also includes the inflow derivatives giving an option to control the smoothness of the input signals.

Introducing the reference vectors

$$y^{\text{ref}} = \begin{bmatrix} y_s^{\text{ref}} & y_r^{\text{ref}} & y_{he}^{\text{ref}} \end{bmatrix}^T, \quad u^{\text{ref}} = \begin{bmatrix} u_p^{\text{ref}} & u_e^{\text{ref}} & u_h^{\text{ref}} \end{bmatrix}^T,$$

comprising the stationary solution of Grade 2, and the diagonal weighting matrices $Q_{\Delta y}$, $Q_{\Delta u}$ and $Q_{\dot{u}}$, the optimal control problem representing the transition can be formulated as

$$\min_{u_p, u_e, u_h} \int_{t_0}^{t_f} \begin{bmatrix} \Delta y \\ \Delta u \\ \dot{u} \end{bmatrix}^T \begin{bmatrix} Q_{\Delta y} & 0 & 0 \\ 0 & Q_{\Delta u} & 0 \\ 0 & 0 & Q_{\dot{u}} \end{bmatrix} \begin{bmatrix} \Delta y \\ \Delta u \\ \dot{u} \end{bmatrix} dt \qquad (7)$$

$$\begin{aligned} \text{subj. to} \quad & 0 = F(\dot{x}, x, w, u) \\ & y = g(x, w, u) \\ & 0 \leq h(x, w, u) \\ & y_{\min} \leq y \leq y_{\max} \\ & \dot{u}_{\min} \leq \dot{u} \leq \dot{u}_{\max}, \end{aligned}$$

where the deviation vectors are defined as

$$\Delta y = y - y^{\text{ref}}, \quad \Delta u = u - u^{\text{ref}},$$

and the initial values at time $t = t_0$ are defined by the steady state solution for Grade 1 in the previous section.

An optimization model of the problem utilizing Optimica constructs and the ReactorPackage can easily be encoded, see Listing 1. The algebraic contraint, i.e., $0 \leq h(x, w, u)$, is encoded in the reactor model while the constraints on outputs and inflow derivatives are encoded in the optimization model below. For details on the Optimica syntax, see Åkesson (2008).

```
optimization ReacOpt(objective=cost(finalTime),
                     startTime=0.0,finalTime=1)
   extends OptimizationParameters;
   extends ReactorPackage.Optimization.OptModel;
   equation
        der(cost) = q_yr*(yr - yr_ref)^2
                  + q_ys*(ys - ys_ref)^2
                  + q_yhe*(yhe - yhe_ref)^2
                  + q_up*(up - up_ref)^2
                  + q_ue*(ue - ue_ref)^2
                  + q_uh*(uh - uh_ref)^2
                  + q_up_dot*(up_dot)^2
                  + q_ue_dot*(ue_dot)^2
                  + q_uh_dot*(uh_dot)^2;
   constraint
               yr  <= yr_ref;
               ys  <= ys_ref*1.05;
     ys_ref*0.95  <= ys;
           up_dot  <= u_p_dot_lim;
          -up_dot  <= u_p_dot_lim;
           ue_dot  <= ue_dot_lim;
          -ue_dot  <= ue_dot_lim;
           uh_dot  <= uh_dot_lim;
          -uh_dot  <= uh_dot_lim;
end ReacOpt;
```

Listing 1. Optimization model ReacOpt defining grade change optimization problem using Optimica constructs.

6.4 Optimal Grade Transition Trajectories

Using the JModelica.org framework and the optimization model in the previous section resulted in an NLP problem to be solved by IPOPT. The original optimization problem has 14 differentiated variables and 163 algebraic variables. The resulting NLP problem contains about 20.000–40.000 variables depending on collocation point selection. Using an Intel® Core™2 Duo CPU@3.00GHz, a solution is obtained in approximately 5-30 minutes depending on number of variables and initial values.

Figures 2–4 show the resulting optimal trajectories for the grade transition problem. Note that the trajectories have been scaled, either such that initial value is 1 or by constraint and the total transition time is equal to 1. Both inflows u and controlled outputs y tend to the desired reference values specified by the static optimization problem while all constraints are fulfilled.

Since the production rate y_r is to be increased, the reaction rate of ethylene is increase by changing the ethylene feed, see Figure 2. To be able to keep the amount of polymer in the reactor near a constant value, the inflow of diluent is also increased. This, together with pressure control loops, increases the outflow of both polymer and gases/fluids of the reactor. However, ethylene and diluent feeds are not increased at maximum speed, see Figure 3, since the objective of correct hydrogen-ethylene ratio y_{he} is also taken into consideration. There is a constraint on how much the hydrogen inflow can be increased which is active during the beginning of the transition. This constraint is the main limiter of the grade transition time considering the chosen optimization criteria.

From Figure 4, it is seen that the constraint of no overshoot in the production rate y_r is fulfilled. Also, the mass of polymer in the reactor during the transition is almost constant, which is due to the high availability of diluent.

7. SUMMARY

This paper has presented a novel modular Modelica library built upon existing non-linear DAE models currently used at Borealis AB in an industrial non-linear MPC controller for the Borstar® process. The key contribution of the paper is the use of Optimica constructs to pose grade change optimization problems and the JModelica.org framework for compilation and generation of C code utilized by an

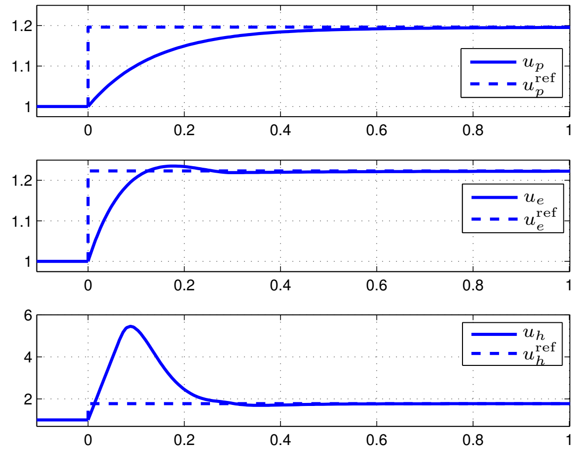

Fig. 2. Scaled optimal input flows at grade transition – propane u_p, ethylene u_e and hydrogen u_h.

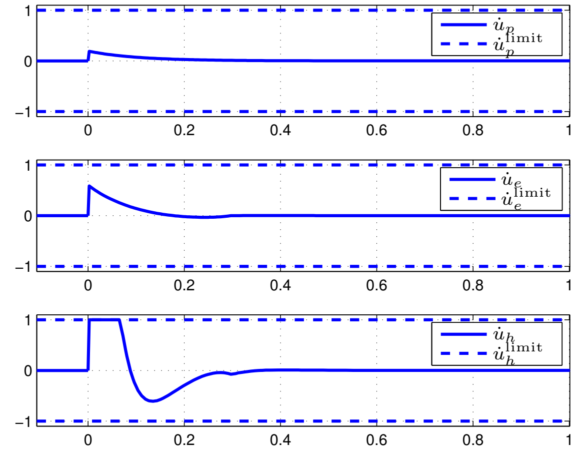

Fig. 3. Scaled time derivatives of optimal input flows at grade transition – propane $\dot{u}_p$, ethylene $\dot{u}_e$ and hydrogen $\dot{u}_h$.

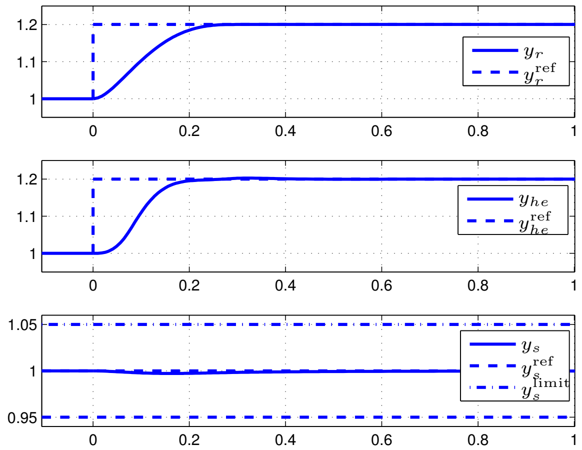

Fig. 4. Scaled optimal outputs at grade transition – mass of polymer in reactor y_s, reaction rate y_r and hydrogen-ethylene ratio y_{he}.

NLP solver. The grade transition solution procedure has been shown and examples of optimal input and output trajectories with constraints have been calculated The example has shown e.g., where one of the limitations during a grade transition can be found.

8. FUTURE WORK

Due to the modularity of the Modelica library the optimization can easily be extended to include more than one reactor, which is work in progress and shows promising results. In the example shown, only three input flows have been used as optimization variables. However, in a reactor chain, each reactor has input flows but also controlled output flows such as bleed flows, increasing the number of optimization variables and degrees of freedom. Current work in progress is also concentrating on using more outputs than is shown in the example. This includes key parameters such as split factor, MFR and densities.

REFERENCES

Åkesson, J. (2008). Optimica—An Extension of Modelica Supporting Dynamic Optimization. In *In 6th International Modelica Conference 2008*. Modelica Association.

Åkesson, J., Bergdahl, T., Gäfvert, M., and Tummescheit, H. (2009). Modeling and Optimization with Modelica and Optimica Using the JModelica.org Open Source Platform. In *Proceedings of the 7th International Modelica Conference 2009*. Modelica Association.

Biegler, L., Cervantes, A., and Wächter, A. (2002). Advances in simultaneous strategies for dynamic optimization. *Chemical Engineering Science*, 57, 575–593.

Flores-Tlacuahuac, A., Biegler, L.T., and Saldivar-Guerra, E. (2006). Optimal grade transitions in the high-impact polystyrene polymerization process. *Industrial & engineering chemistry research.*, 45(18), 6175–6189.

Fried, J. (2003). *Polymer Science and Technology*. Prentice Hall, Englewood Cliffs, N.J.

IEEE (1997). Standard VHDL Analog and Mixed-Signal Extensions. Technical report, IEEE.

Meyers, R.A. (2005). *Handbook of Petrochemicals Production Processes*. McGraw-Hill Handbooks, 2 Penn Plaza, NY.

Prata, A., Oldenburg, J., and Kroll, A. (2008). Integrated scheduling and dynamic optimization of grade transitions for a continuous polymerization reactor. *Computers & Chemical Engineering*, 32(3), 463–477.

Process Systems Enterprise (2007). gPROMS Home Page. `http://www.psenterprise.com/gproms/index.html`.

Python Software Foundation (2009). Python Programming Language – Official Website. `http://www.python.org/`.

Saarinen, M. and Andersen, K.S. (2003). Applying Model Predictive Control in a Borstar Pilot Plant Polymerization Process. Prepared for presentation at the session New Trends in Pilot Plants I, AIChE 2003 Spring National Meeting, New Orleans LA, March 31st.

van Brempt, W., van Overschee, P., Backx, T., and Moen, O. (2004). Plantwide economical dynamic optimization: Application on a Borealis Borstar process model. In *ADCHEM2003*. Hong Kong.

Wächter, A. and Biegler, L.T. (2006). On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1), 25–58.