Modular Optimization and Optimal Control of Polymerization Reactions

Michael Wulkow, CiT GmbH,
Oldenburger Str. 200, Rastede, 26180, Germany, m.wulkow[at]cit-wulkow.de

Modeling and simulation of polymerization reactions range from studies of elemental kinetic phenomena to real-time plant simulations. Where formerly monomer conversion and property averages had been studied, now product properties like full molecular weight distributions, copolymer composition, branching degree and network densities are under consideration.

Because of the numerical complexity of the underlying mathematical equations, even off-line optimization and optimal control of such properties is still challenging. An optimal control problem for polymerization might consist of several objectives and controls requiring the solution of high-dimensional population balances. For example, a typical task is to produce a certain shape of the molecular weight distribution in optimal time and under some constraints regarding feed strategy and maximal temperature. If various controls like initial mixture, time-dependent feed rates, reactor cooling function and variable process time are to be identified in one single problem set-up, the number of variables can easily increase to several dozens. Apart from the pure technical and mathematical aspects to solve such a problem, in practice one has to follow a typical course of action.

- Start with a well-tested, predictive model in a reliable simulation environment
- Define appropriate objective functions and their relative weighting. For example, if one wants to minimize the difference between a property A of the model and the requirement A*, an integral condition $\int_0^T \left(A(t) - A^* \right)^2$ will not suffice. Instead one should use a relative measure and a weighting $\frac{\omega}{T}\left(\int_0^T \left(\frac{A(t) - A^*}{A^*} \right)^2 \right)^{1/2}$. This term will describe the relative weighted deviation of state A per time.
- Decide about the controls (number, constraints, further parameters)
- Solve a nonlinear optimization problem
- Analyze the results (optimality, uniqueness, problems)
- Use the optimized model.

It can always happen, that at intermediate steps the model has to be reformulated, requirements change etc. As a consequence the optimized model drifts apart from the original model and a new optimization has to be performed. At the same time, quite new optimization tasks arise such that a fixed user-interface will be at its limits soon. Here the basic technical aspects mix with ergonomic and software issues. It is required to have a modularized approach, where the simulation part is separated from the optimization tasks. Therefore a modular framework has to be developed, where simulation, preparations and parameter identification are separated.

A simplified version of a classical optimal control problem often has the following structure:

$$\min_u F(u) = g(x_0, x_T) + \int_0^T h(x(t), u(t))dt$$
$$\frac{dx}{dt} = f(x(t), u(t)) \qquad (1)$$
$$b(x(t), u(t)) \le 0$$
$$r(x_0, x_T) = 0$$

Here the state variables are denoted by x(t) and the controls by u(t). For polymer systems describing a chain-length distribution, x(t) is a vector of formally infinite dimension, in practice including ten thousands or even more than one million single components. The control vector u(t) might consist of time-.dependent functions like feed rate per time and/or cooling temperature, but also of single parameters like initial mass of initiator. The right-hand side f(x(t),u(t)) summarizes all reactions and additional balances of a kinetic system. The setup of such systems is described in [1]. Among the state constraints there are conditions like a maximal temperature or heat production or an upper limit of feed rate. Moreover the process time T may be left open as another optimization variable in order to reach a “time optimal” control.

The functions g and h in the objective function formally describe the deviation of the actual states and controls from some required values. For polymer systems the evaluation of these expressions can be very difficult. If the shape of a molecular weight distribution is prescribed (usually at the end of the process), its deviation compared to a simulated distribution has to be defined in a reasonable way. Scaling and weighting play a particular role in this context.

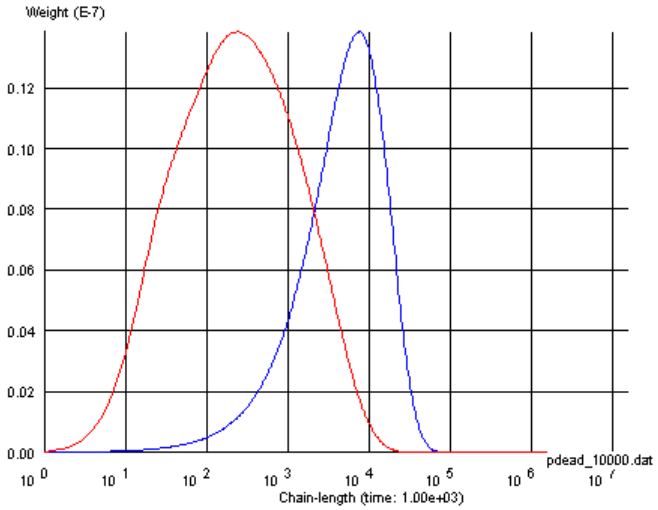

Figure 1: Normalized comparison of shapes of molecular weight distributions, distribution from GPC (red) and from actual simulation (blue).

A general approach to optimal control problems related to differential equations is to apply the Maximum principle to the Hamilton function of the system leading to a boundary value problem and implicit expressions for the controls. The related analytical preparations and distinction of cases are more or less prohibitive in the context if infinite-dimensional population balances. Therefore we pursue the so-called direct approach, where all controls are parameterized using certain basis functions (e.g. piecewise constant/linear functions, splines, polynomials etc.). By that the optimal control problem can be transferred into a large scale parameter optimization problem with constraints on the parameters, the resulting control functions and the state variables. The direct approach has the advantage that it can be applied to a basic differential equation system without much preparation. Furthermore constrains on controls and states can be set directly. A disadvantage might be the fact, that dependent on the chosen parameterization, large numbers of optimization variables can arise and that the obtained results are usually not unique and not optimal in the strict sense of the Maximum principle. Fortunately and in contrast to classical parameter identification of kinetic rate constants, for control purposes parameters and controls have not to be unique - but (sub-) optimal „only“. This requires special numerical algorithms

efficient enough to ensure fast convergence, but capable of dealing with underestimated least-square problems.

Therefore we sketch some brief details of a new algorithm, originally developed for parameter estimation problems, which turned out to be particularly helpful for parameterized control problems too. The basic idea is to solve the optimization problem by use of a Gauss-Newton method and perform an advanced analysis of the related Jacobian matrix (containing the derivatives of all objectives with respect to all parameters) aiming at the number of so-called “essential degrees of freedom”, i.e. the number of parameters which can be uniquely identified in the present model in view of the present information (i.e. data for classical parameter estimation or constraints and requirements for optimal control problems).

If the number of essential parameters is not maximal, there will be correlations or total insensitivities among the parameters. Technically we define the essential directions of the Jacobian matrix as those directions which belong (under singular value decomposition) to *large* singular values, hoping that large singular values hold more characteristic of the matrix than small ones and that neglecting small singular values will not imply loss of too much information. It has to be mentioned, that the obtained number of essential directions in parameter space (leading to number of essential parameters) depends on a threshold, which cannot be given as a fixed theoretical value, but has to be chosen as parameter of the algorithm itself. However, this threshold could be identified to be in a small range for many tested examples (as a rule of thumb: the ratio of singular values of the remaining lower-dimensional sub problem should be smaller than 100).

The program package Predici [1] has been used for more than 12 years for the modeling of polymerization kinetics. The core feature of Predici and its underlying Galerkin h-p-method - the computation of full chain-length distribution - has been extended to additional distributed properties such as mentioned above. The question is how to make use of this complexity when it comes to optimization without change of model structure and definition (since models are permanently changed and improved, an optimization procedure should leave the model itself unchanged). Whereas for classical optimization problems special algorithms are available - mostly based on first and second derivatives of a differentiable objective function - in the context of full chain-length distributions a computation of higher order derivatives is prohibitive. Therefore in Predici the basic parameter estimation tool has been extended by additional functions to allow for optimal control problems.

Principally all input for a control problem (1) can be done in any typical Predici version. For example, inequality conditions of the form $b(x_i) \le c_i$ can be reformulated in terms of a slack variable L_i to $b(x_i) + L_i^2 = c_i$. The new parameter L_i can be added to the list of model parameters and included in the estimation process. Control functions can also be formulated by use of commands of Predici’s script language (polygon, polynomial). However, after first successful tests with the general approach, it turned out, that for the daily work the permanent redefinition of parameters and script function would be error prone and time consuming.

Instead a preprocessor has been implemented and equipped with a full user interface, called Priamoz. The idea is to take the unchanged model set-up, define optimization based requirements in the user-interface and automatically generate all functions, parameters and weightings by an automatic procedure leading to an augmented system. The subsequent parameter estimation – applying the essential directions approach – leads to results which can be checked and confirmed and are finally transferred back to a basic model without traces of the intermediate solution process.

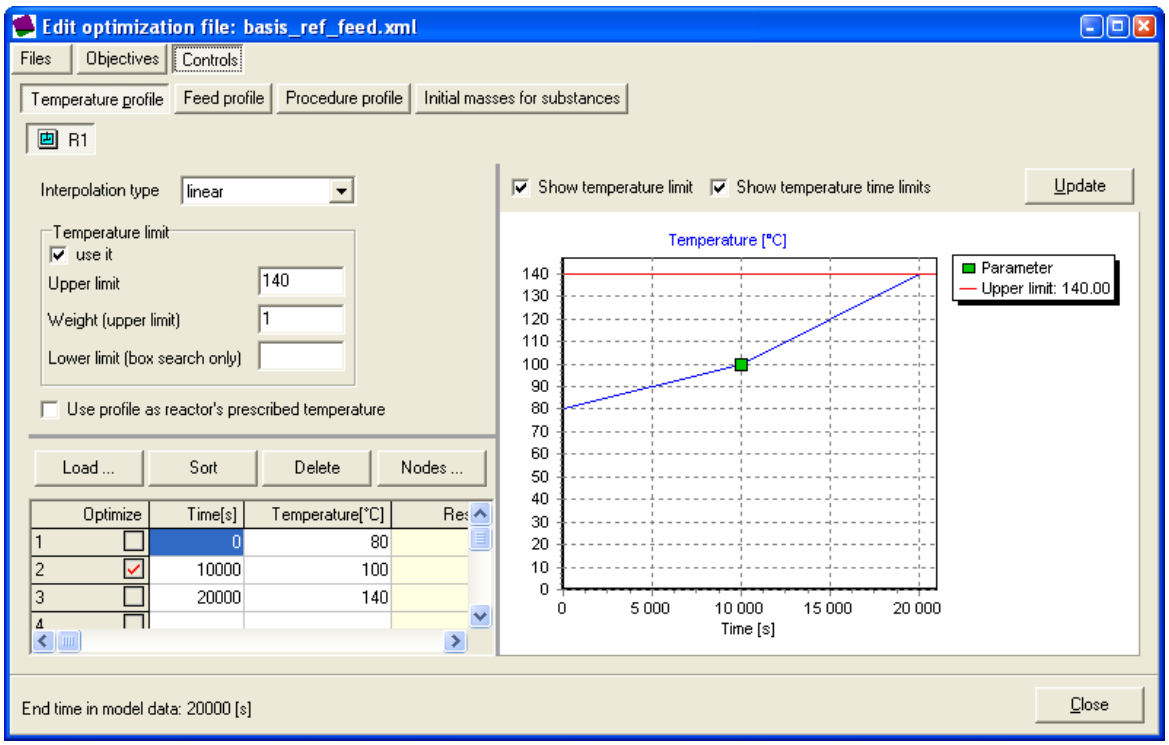

Figure 2: Selection and initialization of a temperature profile as control function in Priamoz. The input is transformed to Predici model input.

Regarding the objectives all model outputs can be selected. Furthermore also a time-optimal control can be added. This requires a variable end time of the process which in turn leads to transformation of the controls to be identified. In many theoretical papers thus a transformation [0,T] → [0,1] is applied to the differential equation system. Such a transformation, based on an optimization parameter T_{end}, has also been introduced in Predici. Instead of using the unit interval, however, a reasonable process time T_p is used to keep all other parts of the dynamic system nearly unchanged and to allow direct suggestive graphical output.

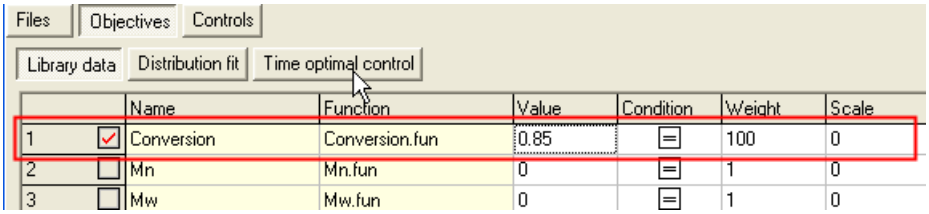

Figure 3: Selection of objectives in Priamoz. The reported functions for conversion and polymer mean values are part of a Predici model and defined by the user.

This approach might look like an aspect of pure software architecture, but actually provides an efficient and comfortable way to extent the current status of the used tools to new problems without programming. It can be used for many kinds of objectives, even in connection to direct rheology computations (for mostly linear polymer chains) which are available employing the full information of the molecular weight distribution. The whole framework could also be a way to use a common interface for the generation of model files of different simulation tools.

References:

[1] M. Wulkow, *Computer Aided Modeling of Polymer Reaction Engineering - The Status of Predici, 1 – Simulation*, Macromol. React. Eng. 2008, 2 (Issue 6), in print; published online: 04.09.2008, DOI: 10.1002/mren.200800024