ON THE USE OF MODELS FOR DYNAMIC REAL-TIME OPTIMIZATION

Dominique Bonvin* and Bala Srinivasan**

* *Laboratoire d'Automatique, Ecole Polytechnique Fédérale de Lausanne CH-1015 Lausanne, Switzerland*
** *Department of Chemical Engineering, Ecole Polytechnique Montreal, Montreal, Canada*

Abstract

The operation of dynamic processes can be optimized using models that predict the system behavior well, in particular its optimality features. In practice, however, process models are often structurally inaccurate, and on-line adaptation is typically required for appropriate prediction and optimization. Furthermore, it is difficult to identify process model parameters on-line during optimization because of lack of persistent excitation. This paper addresses the modeling issue for the purpose of real-time optimization. It will be shown that the models used for real-time optimization need not be valid as a whole; instead, it suffices that they represent the optimality conditions well. Two types of models are considered, namely, the traditional "plant models" and the tailor-made "solution models". The features of each type, in particular their ability to be adapted using on-line measurements, are discussed and illustrated through a simple car example.

Keywords

Real-time optimization, Dynamic optimization, Plant-model mismatch, Modeling for optimization, Model adequacy, Plant model, Solution model.

Introduction

Optimization is important in science and engineering as a way of finding "optimal" situations, designs or operating conditions. Optimization is typically performed on the basis of a mathematical model of the object of attention. In engineering, we are concerned with the optimal operation of processes that either operate at steady state or undergo transient changes. The object of attention, or reality, is called the "plant", whereas the "model" is a set of algebraic, differential or differential-algebraic equations.

In practice, optimization is complicated by the presence of uncertainty in the form of plant-model mismatch and unknown disturbances. Without uncertainty, one could use the model at hand, optimize it numerically off-line and implement the optimal inputs in an open-loop fashion. However, because of uncertainty, additional information such as uncertainty description or plant measurements must be included. In the former case, robust optimization computes a set of inputs that guarantees feasibility either for all possible realizations or with a desired probability, however at the expense of a conservative solution (Srinivasan et al. (2003)). In the latter case, the inputs are updated in real-time based on measurements. This is the field of *real-time optimization*, which is labeled RTO for static optimization problems (Marlin & Hrymak (1997)) and DRTO for dynamic optimization problems (Biegler (2009)). This paper deals with DRTO, although some of the arguments also hold for RTO.

DRTO has two major implementation issues, namely, computational aspects and model quality.

- Regarding computational aspects, there has been considerable efforts in recent years to speed up the computations by considering convex optimization problems and algorithms that exploit the structure of the problem (Wang & Boyd (2010)). Furthermore, recent trends in DRTO have included attempts to move the heavy computations off-line, where time and computational power are more available, and limit the on-line operations to quick decisions or easy computations. For example, multi-parametric programming generates off-line a lookup table of control laws, which are then used on-line based on the estimated states of the plant (Bemporad et al. (2002)). NCO tracking uses a parameterized model of the solution determined off-line to design a multivariable feedback scheme that tracks the first-order necessary conditions of optimality (NCO), thereby pushing the system toward optimality (Srinivasan & Bonvin (2007)).

The issue of model quality raises an important question: Does good performance require a good model? This is not necessarily the case for control, since errors resulting from a poor model are offset by the action of feedback. In optimization, without feedback to make up for modeling errors, the model needs to represent the reality rather well, in particular the optimality conditions of the plant. The situation is slightly different in real-time optimization since the measurements used on-line represent some form of feedback. However, this feedback is limited and cannot possibly lead to a model that is valid as a whole. Hence, it is important to adapt there, where it matters most for the purpose of optimization. As an alternative to plant models, solution models can be developed, whereby the structure of the optimal solution is determined and parameterized in terms of input arcs and parameters. This paper considers the realistic case of structural plant-model mismatch, that is, when the plant is not in the model set. Hence, adaptation using measurement is necessary. Plant and solution models will be compared in their ability to process on-line measurements to lead to optimality.

The paper is organized as follows. The next section reviews strategies that use measurements on-line for the purpose of optimization. The following section discusses plant models and the features that a model tailored to optimization should possess; this will naturally lead to the topic of solution model. Finally, the various ideas developed in this paper are illustrated through a simple car example.

Adaptation Strategies for DRTO

In real-time optimization, the model is a vehicle to process the available measurements and move toward optimality. The inputs that are applied to the plant are computed using a (possibly) inaccurate model and a limited number of on-line measurements. The on-line measurements are used to update the plant model or the solution model as follows:

(1) *Plant model: use measurements to adapt the model parameters and estimate the current states, then repeat the optimization with the updated model.* The estimated states serve as initial conditions for the optimization performed repeatedly on-line (Eaton & Rawlings (1990)). This scheme is also known as the two-step approach of repeated parameter identification and performance optimization.

(2) *Solution model: use measurements to adapt the inputs.* This implies using feedback to update the plant inputs. The control scheme is set up using information regarding the structure of the optimal solution (in particular the active constraints), which can be determined off-line by optimization of the nominal plant model. This knowledge about the solution is collected in a "solution model" (Srinivasan & Bonvin (2007)).

Models for Optimization

This section starts with a discussion on plant models and develops the idea that the validity of a model depends on its intended use, then introduces the concept of solution model, and finally elaborates on the way each model can be adapted.

Plant Models

To construct a mathematical model, the modeler typically uses both prior information on and measurements from the plant. The modeler goes through several steps that include (i) abstraction from the reality to define the "system", (ii) simplification to arrive at a mathematical model of manageable complexity, (iii) parameter identification to fit the model to the plant, and (iv) model validation to ensure that the model will be useful for its intended goal. Since the later steps influence the early ones, this procedure is typically iterative.

The model validation step is very important. How can we ensure that the model will be adequate for solving the optimization problem at hand. Model identification and validation are often done by comparing model prediction with observed data, typically plant outputs. This is convenient because the outputs are, by definition, available. Furthermore, it is fully justified if the model's main purpose is to predict the outputs, for example in a simulation study. But is it still justified if the model is used for optimization? This question is addressed next.

It is well known that the two-step approach works well provided that (i) there is no structural plant-model mismatch, i.e. the plant lies in the model set, and (ii) the operating conditions yield sufficient excitation for all the uncertain model parameters to be estimated. Yet, these conditions are rarely met in practice. Regarding the latter condition, in particular, the situation is somewhat similar to that found in the area of system identification and control, where the two tasks of identification and control are typically conflicting (dual control problem, Aström & Wittenmark (1995)).

A way of improving the synergy between the identification and optimization steps is to reconcile the objective functions of these two problems. Consider the optimality conditions for a dynamic optimization problem:

$$\underbrace{0 = \frac{\partial H_p}{\partial u}}_{\text{objective}} \;=\; \underbrace{\frac{\partial H}{\partial u}}_{\text{optimization}} \;+\; \underbrace{\left(\frac{\partial H_p}{\partial u} - \frac{\partial H}{\partial u}\right)}_{\text{identification}}, \qquad (1)$$

where H denotes the Hamiltonian of the optimization problem and u the vector of input or decision variables. Optimality of the plant implies zeroing the gradient of the Hamiltonian, $\frac{\partial H_p}{\partial u}$. For its part, numerical optimization of the model enforces $\frac{\partial H}{\partial u} = 0$. Hence, if the identification step can ensure that $(\frac{\partial H_p}{\partial u} - \frac{\partial H}{\partial u})$ is negligible, the identified model will be suited for optimization. One possibility is to minimize a filtered version of the output prediction error, with the filter designed to make the output prediction error resemble $(\frac{\partial H_p}{\partial u} - \frac{\partial H}{\partial u})$. Along these lines, Srinivasan & Bonvin (2002) proposed to modify the objective function of the identification problem so as to include the cost function and the constraints of the optimization problem. The approach has been inspired by the work done in "identification for control" (see for example, Aström (1993)) and could therefore be labeled "modeling for optimization".

Solution Models

A solution model is a parameterized model of the optimal inputs that can be used to implement optimizing control. The development of a solution model involves three steps: (i) the characterization of the optimal solution in terms of the types and sequence of arcs, (ii) the selection of the manipulated and controlled variables for the control problem, and (iii) model validation to ensure that the model will be useful for its intended goal. If the resulting errors are too large, it is necessary to rethink the structure of the solution model and repeat the procedure.

The optimal solution is typically characterized via optimization of the available (possibly inaccurate) plant model. A robustness analysis should then be performed to assess the validity of the solution structure with respect to parametric variations and disturbances. The optimal inputs are parameterized in terms of the various arcs and switching times that are affected by uncertainty. The solution model formally expresses the NCO (path constraints, path sensitivities, terminal constraints and terminal sensitivities) that need to be enforced for optimality. The various NCO elements can be implemented with various degrees of ease or difficulty: a path constraint can often be enforced on-line via constraint control; a path sensitivity is more difficult to implement as it involves the adjoint variables, which are not available on-line without the use of a plant model; the terminal constraints and sensitivities call for prediction, which requires some model, or else they can be met iteratively over several runs. To ease implementation, it is often possible to approximate the optimal inputs with simpler profiles. This represents the strength of the approach, as the approximations introduced at the solution level can be assessed in terms of optimality loss.

The manipulated variables (MV) of the controlled problem are the handles available to reach optimality. The controlled variables (CV) are the NCO for the selected input parameterization. By definition of the optimality conditions, there are as many optimality conditions as there are degrees of freedom, thus resulting in a square control system. The pairing of MV and CV can be done in a centralized (multivariable control) or decentralized way (multi-loop control). Note that there are different ways of implementing a given solution model, for example using alternative MV via a change of variables, using different pairings of MV and CV, or using a plant model for prediction, each way defining a different solution model.

Adaptation of Plant and Solution Models

The plant model consists of dynamic equations that include the uncertain plant parameters θ:

$$\dot{x} = f(x, u, \theta), \qquad x(0) = x_0. \tag{2}$$

The uncertain parameters can be identified with the optimization objective in mind as suggested in (1):

$$\min_{\theta} \| \frac{\partial H_p}{\partial u} - \frac{\partial H(x, u, \theta)}{\partial u} \|. \tag{3}$$

In solution models, the inputs are expressed in terms of the input arcs $\eta(t)$ and parameters π:

$$u(t) = \mathcal{U}(\eta(t), \pi). \tag{4}$$

The input elements are adjusted using measurements to drive the plant to optimality:

$$\frac{\partial H_p}{\partial \eta(t)} \to 0, \qquad \frac{\partial H_p}{\partial \pi} \to 0. \tag{5}$$

Two issues affect the validity of the approaches:

(1) Are the quantities to be estimated good handles to drive the plant to optimality?
(2) How easy is it to identify these quantities, that is, how much excitation is required for identification?

With plant models, both issues might be problematic. Although the uncertain parameters can be adapted for the model to predict the plant outputs rather well, the resulting model might not be able to push the plant to the optimum, an indication that the model does not adequately represent the optimality conditions of the *plant* (Srinivasan & Bonvin (2002)). Furthermore, it is well known that parameter identification is a difficult task in the presence of structural plant-model mismatch, that is, when the plant does not belong to the model set (Ljung (1999)). In this case, parameter identification requires appropriate experimental design (Montgomery (2005)) and persistency of excitation (Walter & Pronzato (1997)).

With solution models, the situation is clearly different. The uncertain input elements define the optimal solution and are therefore directly linked to the NCO. In essence, the optimization problem has been reformulated in such a way that all uncertain elements are intimately connected to the NCO. Hence, there is no risk of wasting excitation to identify quantities that are not relevant to the optimization. On the other hand, since the parameters are obtained from measurement/estimation of NCO elements, the difficulty in this approach lies in estimating these NCO elements, namely, the active plant constraints and the reduced plant gradients.

Illustrative Example

The use of solution models for real-time optimization will be illustrated on the simple car example that is presented next:

- *System:* Movement of a car from one point to another.

- *Uncertainty:* Slope of the road: ± 5%.
- *Objective:* Minimize final time.
- *Manipulated input:* Accelerating/braking force.
- *Path constraints:* Input bounds; speed limit.
- *Terminal constraints:* Zero velocity at final time; cover at least the prescribed distance.

Problem Formulation

Variables and parameters

x: position, v: velocity, u: accelerating/braking force, s: slope of the road, f: friction coefficient, g: gravitational constant, m: mass of the car.

Model equations

$$\dot{x} = v\,, \qquad x(0) = 0\,, \tag{6}$$

$$\dot{v} = \frac{u - fv^2}{m} - s(x)g\,, \qquad v(0) = 0\,. \tag{7}$$

m	1300	kg
f	0.5	$\frac{Ns^2}{m^2}$
u_{min}	-8000	N
u_{max}	3600	N
v_{max}	40	$\frac{\text{m}}{\text{s}}$
x_{des}	1000	m

Table 1. Model parameters and operating bounds

The nominal model assumes zero slope, i.e. $s(x) = 0$, while in reality the unknown elevation profile, which is the integral of the slope profile, is as shown in Figure 1.

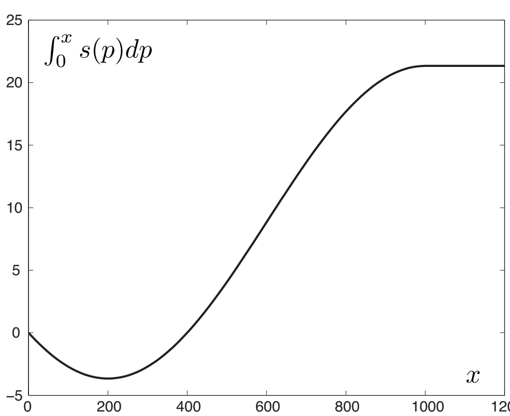

Figure 1. Elevation profile

Optimization problem

The optimization problem can be formulated mathematically as follows:

$$\min_{u(t),\, t_f} \quad J = t_f \tag{8}$$

$$\begin{aligned} s.t.\ & \text{dynamic system } (6)-(7) \\ & u_{min} \leq u(t) \leq u_{max} \\ & v(t) \leq v_{max} \\ & v(t_f) = 0 \\ & x(t_f) \geq x_{des}\,. \end{aligned}$$

If the elevation profile were known, the minimal time would be $J^* = 34.68$ s.

Characterization of the Optimal Solution

Figure 2 shows that the nominal optimal solution consists of three arcs, with the successive inputs u_{max}, u_{path} and u_{min}:

- The first arc u_{max} corresponds to maximum acceleration in order to reach v_{max} as quickly as possible. The duration of this arc, t_1, depends on the slope, which is uncertain. However, t_1 can be determined implicitly upon reaching v_{max}, i.e. $v(t_1) = v_{max}$.
- The second arc keeps the velocity at v_{max}, for which the corresponding input value u_{path} can be determined as $u_{path} = fv_{max}^2 + s(x)mg$. The value u_{path} is also a function of the uncertain slope.
- The third arc corresponds to full braking in order to achieve $v(t_f) = 0$. The switching time t_2 between the second and third arcs is chosen so that $x(t_f) = x_{des}$, i.e. the desired distance will be exactly covered when the velocity goes to zero. The final time is determined upon reaching zero velocity, i.e. $v(t_f) = 0$.

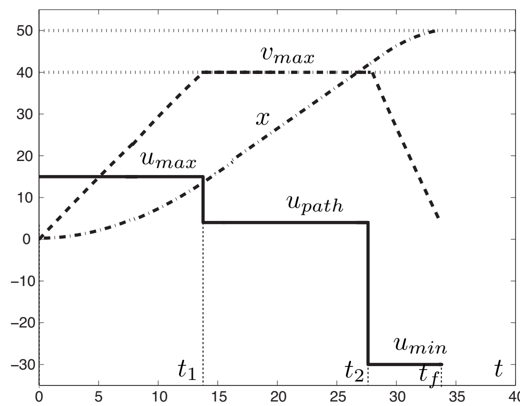

Figure 2. Input u, velocity v and position x profiles for the nominal optimal solution ($s = 0$)

Remarks

(1) This car example does not involve sensitivities, i.e. the optimal solution is entirely determined by active constraints. The reader is referred to Srinivasan & Bonvin (2007) for cases where sensitivities are involved.

(2) The types and sequence of arcs (u_{max} followed by u_{path} and then u_{min}) hold for any car regardless of its weight and acceleration or braking power. They even hold generically for a bicycle, for which the second arc vanishes (t_1 does not exist). This generic aspect of the optimal solution provides much robustness to the solution-model approach.

Selection of Manipulated and Controlled Variables

Input parameterization is straightforward in this problem, with the input elements affected by uncertainty chosen as handles, here $\eta(t) = u_{path}(t)$ and $\pi = [t_1\ t_2\ t_f]^T$. We present next three solution models that correspond to different ways of meeting the NCO.

Solution model A

The pairing of MV (t_1, $u_{path}(t)$, t_2 and t_f) and CV ($v(t_1) = v_{max}$, $v(t) = v_{max}$, $x(t_f) = x_{des}$, $v(t_f) = 0$) follows directly from the characterization of the optimal solution and is given in Table 2. As mentioned above, t_1 and t_f are determined implicitly upon reaching the velocities v_{max} and 0, respectively. Since there is a prediction involved in the pairing $t_2 \mapsto x(t_f) = x_{des}$, meeting this constraint will either require a predictive model or be implemented over several runs. In the absence of a predictive model, NCO tracking will use on-line control to enforce $v(t) = v_{max}$ in the second arc and run-to-run control to adapt t_2 so as to enforce $x(t_f) = x_{des}$ over several runs.

	Path objectives	Terminal objectives
Constraints	$t_1:\ v(t_1) = v_{max}$ $u_{path}(t):\ v(t) = v_{max}$	$t_2 \mapsto x(t_f) = x_{des}$ $t_f:\ v(t_f) = 0$
Sensitivities	–	–

Table 2. Pairing of MV and CV in Solution model A

Solution model B

The error resulting from not meeting the terminal constraint $x(t_f) = x_{des}$ in the initial runs can be reduced by re-parameterization of the problem. The switching time t_2 is determined upon reaching the position x_{brake}, which becomes a decision variable.[1] On-line control is used to implement $v(t) = v_{max}$ in the second arc and run-to-run control to adapt x_{brake} so as to enforce $x(t_f) = x_{des}$. The pairing of MV and CV is given in Table 3.

	Path objectives	Terminal objectives
Constraints	$t_1:\ v(t_1) = v_{max}$ $u_{path}(t):\ v(t) = v_{max}$	$x_{brake} \mapsto x(t_f) = x_{des}$ $t_f:\ v(t_f) = 0$
Sensitivities	–	–

Table 3. Pairing of MV and CV in Solution model B

Solution model C

To avoid having to optimize over several runs, one needs to be able to predict the final position $x(t_f)$ during the run in order to initiate the breaking action. This can be done using the plant model on-line as discussed next.

The characterization of the optimal solution has led to a rather parsimonious parameterization of the input compared to the infinite dimension of $u(t)$. For example, based on the solution model given in Table 2, t_1, $u_{path}(t)$ and t_f can be determined by their corresponding NCO during simulation (integration) of the dynamic model. The switching time t_2 remains a decision variable since its determination requires prediction of the final position, which cannot be done "on-line" during a single simulation. It follows that the optimization problem (8) can be rewritten as:

$$\min_{t_2} J = t_f \tag{9}$$

$$s.t.\ \text{dynamic system } (6)-(7)$$

$$u(t) = \begin{cases} u_{max} & \text{for} \quad 0 \le t < t_1 \\ f v_{max}^2 + s(x(t))mg & \text{for} \quad t_1 \le t < t_2 \\ u_{min} & \text{for} \quad t_2 \le t < t_f \end{cases}$$

$$t_1:\ v(t_1) = v_{max}$$

$$t_f:\ v(t_f) = 0.$$

Problem (9) is simpler to solve than the original problem (8) for at least two reasons: (i) the number of degrees of freedom has been reduced from ∞ to 1, and (ii) the discontinuities at the switching instants can be handled much more easily and without oscillations (Schlegel & Marquardt (2006)).

For implementation, the current state information is used as initial conditions for re-optimization using the plant model. Note that the model parameters are not updated. At each re-optimization instant, the optimal value of t_2 is computed. Breaking is then implemented when the running time equals the value of t_2 computed last.[2] The pairing of MV and CV is given in Table 4, where $\hat{x}(t_f)$ is the final distance predicted by the model.

	Path objectives	Terminal objectives
Constraints	$t_1:\ v(t_1) = v_{max}$ $u_{path}(t):\ v(t) = v_{max}$	$t_2 \mapsto \hat{x}(t_f) = x_{des}$ $t_f:\ v(t_f) = 0$
Sensitivities	–	–

Table 4. Pairing of MV and CV in Solution model C.

NCO Tracking

The performance of NCO tracking using Solution model A is discussed next. The performance of Solution models B and C is indistinguishably similar and therefore is not shown here. It is assumed that the plant (the car with varying unknown slope) will have the same types and sequence of arcs, but different values of t_1, $u_{path}(t)$, t_2 and t_f.[3] The solution after 3 runs is shown in Figure 3. Numerical results are given in Table 5. Optimality in this problem is guaranteed by satisfaction of the maximum-velocity and final-position constraints.

Run number	Maximum velocity (m/s)	Final position (m)	Final time (s)
1	40.0	1020	35.19
2	40.0	1002	34.72
3	40.0	1000	34.68

Table 5. Constraint and cost values with NCO tracking based on Solution model A for various numbers of runs

[1] We choose here to advance with the position x rather than with the time t. This is possible because x in monotonic with respect to t. The position x is more representative of the state of the car than the time t as it has "felt" all past disturbances.

[2] Or, similarly, the running distance equals the value of x_{brake} computed last.

[3] This can be verified numerically off-line by perturbing the nominal model and computing the corresponding optimal inputs.

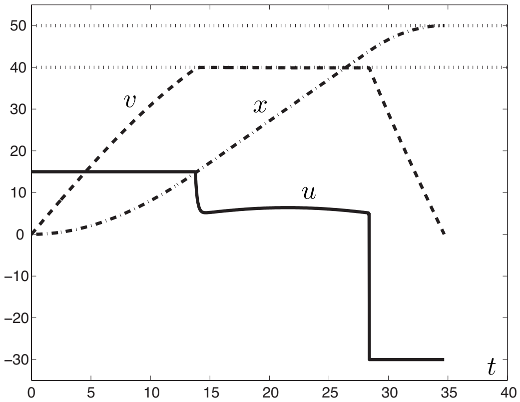

Figure 3. Input u, velocity v and position x profiles after the 3rd run with NCO tracking based on Solution model A. Note that Solution model C generates nearly the same input from a single run.

Conclusions

This paper has addressed the quality of models used for dynamic real-time optimization. According to the quote "All models are wrong but some are useful" (Box (1979)), the model is not viewed as the "truth", but rather as a tool that must be tailored to the optimization scheme. Modeling is really about making educated approximations to arrive at a model of acceptable complexity that is *adequate for optimization* in the presence of uncertainty. When dealing with a real plant, it is important to find a good way of introducing approximations. Is it at the plant-model level, before going through the optimization machinery? Or is it at the implementation level, when the user can see the implications of selected approximations?

This paper has also presented the concept of solution model and its use for tracking the NCO in the context of real-time optimization. A solution model is obtained by dissecting computed optimal input profiles and relating their elements to different parts of the NCO. One strength of NCO tracking is the possibility of combining off-line tasks (numerical optimization based on the nominal plant model to determine the active set) and on-line activities (optimizing control that adjusts the inputs on the basis of measurements). Another nice feature is the possibility, if necessary, of introducing approximations in the various profiles to ease implementation. This is particularly effective in dealing with sensitivity-seeking arcs, which are often difficult to compute but, at the same time, contribute only negligibly to the cost. A feature that distinguishes plant and solution models regards the complexity of the model: indeed, the complexity of solution models depends on the number of inputs, and not on the number of states or the nonlinearity of the plant. Hence, NCO tracking tends to work well for problems with only a few numbers of arcs (and thus also only a few inputs), and this regardless of the order of the system. Finally, one can view NCO tracking as a data-driven scheme for RTO and compare it to fully data-driven schemes such as response surface methods (Georgakis (2009). Instead of building the model that will predict the plant performance from scratch, NCO tracking starts with a robust parameterized model of the solution and adjusts the few input elements that are intimately linked to plant optimality.

Acknowledgment

The authors would like to acknowledge insightful discussions with Benoît Chachuat and Grégory François.

References

Aström, K. J. (1993). Matching criteria for control and identification. *in* European Control Conference. Groningen, Netherlands. 248–251.

Aström, K. J. & Wittenmark, B. (1995). *Adaptive Control*. 2nd edn. Addison-Wesley, Reading, MA.

Bemporad, A., Morari, M., Dua, V. & Pistikopoulos, E. N. (2002). The explicit linear quadratic regulator for constrained systems. *Automatica* 38, 3–20.

Biegler, L. T. (2009). Technology advances for dynamic real-time optimization. *in* Computer Aided Chemical Engineering. Vol. 27. 1–6.

Box, G. E. P. (1979). Robustness in the strategy of scientific model building. *in* R. L. Launer & G. N. Wilkinson, eds, Workshop on Robustness in Statistics.

Eaton, J. W. & Rawlings, J. B. (1990). Feedback control of nonlinear processes using on-line optimization techniques. *Comp. Chem. Eng.* 14, 469–479.

Georgakis, C. (2009). A model-free methodology for the optimization of batch processes: Design of dynamic experiments. *in* ADCHEM 2009. Leuven, Belgium.

Ljung, L. (1999). *System Identification: Theory for the User*. 2nd edn. PTR Prentice Hall, Upper Saddle River, N.J.

Marlin, T. E. & Hrymak, A. N. (1997). Real-time operations optimization of continuous processes. *in* AIChE Symposium Series - CPC-V. Vol. 93. 156–164.

Montgomery, D. C. (2005). *Design and Analysis of Experiments*. 6th edition edn. John Wiley & Sons, New York.

Schlegel, M. & Marquardt, W. (2006). Detection and exploitation of the control switching structure in the solution of dynamic optimization problems. *J. Process Contr.* 16, 275–290.

Srinivasan, B. & Bonvin, D. (2002). Interplay between identification and optimization in run-to-run optimization schemes. *in* American Control Conference. Anchorage, AK. 2174–2179.

Srinivasan, B. & Bonvin, D. (2007). Real-time optimization of batch processes via tracking the necessary conditions of optimality. *Ind. Eng. Chem. Res.* 46(2), 492–504.

Srinivasan, B., Bonvin, D., Visser, E. & Palanki, S. (2003). Dynamic optimization of batch processes: II. Role of measurements in handling uncertainty. *Comp. Chem. Eng.* 27, 27–44.

Walter, E. & Pronzato, L. (1997). *Identification of Parametric Models from Experimental Data*. Springer-Verlag, Berlin.

Wang, Y. & Boyd, S. (2010). Fast model predictive control using online optimization. *IEEE Trans. Contr. Sys. Technol.* 18(2), 267–278.