8th International IFAC Symposium on
Dynamics and Control of Process Systems
Preprints Vol.3, June 6-8, 2007, Cancún, Mexico

IMPLEMENTATION OF OPTIMAL OPERATION USING OFF-LINE COMPUTATIONS

Sridharakumar Narasimhan [*], **Sigurd Skogestad** [*,1]

[*] *Dept of Chemical Engineering, Norwegian University of Science and Technology (NTNU), Trondheim, Norway*

Abstract: The computational effort involved in the solution of real-time optimization problems can be very demanding. Hence, simple but effective implementation of optimal policies are attractive. The main idea is to use off-line calculations and analysis to determine the structure and properties of the optimal solution. This will be used to determine alternate representations of the optimal solution that are more suitable for implementation.

Keywords: Optimality, self-optimizing control, off-line optimization, control structure design

1. INTRODUCTION

Optimal operation of a chemical process as well as many other systems can generally be formulated as a dynamic optimization problem. Let $x \in \mathbb{R}^{nx}$ be the state variables, $u \in \mathbb{R}^{nu}$ the set of available manipulations and $d \in \mathbb{R}^{nd}$ denote the set of disturbances affecting the plant. Let $J(x,u,d)$ be an economic objective that is to be minimized. The dynamic optimization problem that we seek to solve is:

$$\min_u J(x,u,d), \text{ s.t.} \begin{cases} \dot{x} = f(x,u,d), \\ h(x,u,d) = 0, \\ g(x,u,d) \leq 0 \end{cases} \quad (1)$$

There are two main paradigms when it comes to implementation of the optimal solution:

Paradigm 1 On-line optimizing control where measurements are primarily used to update the model. With the arrival of new measurements, the optimization problem is resolved for the inputs.

Paradigm 2 Pre-computed solutions based on off-line optimization. Typically, the measurements are used to (indirectly) update the inputs using feedback control schemes.

In Paradigm 1, the solution and resulting centralized implementation is very complex and is seldom used in practice. Originally, this arose out of necessity, because the computing power for on-line optimizing control (paradigm 1) was not available. Today, this is less of an issue, but it has become clear that paradigm 2 offers a number of additional potential advantages, including robustness, simplicity and reduced cost for modelling, implementation and maintenance.

To understand this better, it is useful to consider how the optimal operation of a complex continuous chemical process is implemented in practice. The main idea is to use time scale separation, where the centralized optimizing controller is broken down into several layers, for example, into regulatory control, supervisory control and optimization layer with the layers arranged in an increasing level of hierarchy (Refer Fig 1). Generally, the interaction between the layers is such that at the optimization layer, we try to determine optimal values of the setpoints or reference values by optimizing an economic cost function and the control layers are devoted to ensuring that the setpoints are maintained (Findeisen *et al.*, 1980). The efficient vertical decomposition between the optimization and control layers requires that the economic objective is determined largely by the slow time scale and that near-optimal operation on the faster time

[1] Corresponding author. Email: skoge@chemeng.ntnu.no

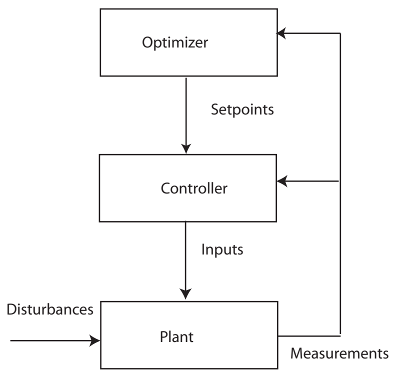

Fig. 1. Hierarchial decomposition

scale is possible by a constant set point policy in spite of disturbances and noise. This is the concept of self-optimizing control (Skogestad, 2004).

Building on our previous work on self-optimizing control, the objective of our present research is to broaden the scope and take a new look at the possibilities and advantages of Paradigm 2. The main idea is to use off-line analysis and computation to discover properties of the optimal solution that may be utilized for on-line implementation. Some specific results for Paradigm 2 may be:

R1 Determine the structure of the optimal solution. Typically, this involves identifying regions where different sets of constraints are active.
R2 Determine optimal values (or trajectories) for the unconstrained variables.
R3 Find analytical or pre-computed solutions suitable for on-line implementation.
R4 Find good self-optimizing controlled variables, c associated with the unconstrained degrees of freedom that satisfy the following (Skogestad, 2004):
- The optimal value c_{opt} is only weakly dependent on disturbances.
- Implementation errors in these variables does not result in a large loss or equivalently the optimum with respect to c is "flat".

R5 Determine a switching policy between different regions.

Results R3, R4, R5 and partially R1 are related to the implementation of the optimal solution. The main idea is to be able to implement optimal operation using simple feedback loops with a minimal need for on-line optimization. Result R2 and partially R1 are related to conventional "open loop" optimization.

Result R3 includes the conventional feedback control paradigm, which may be viewed as a subclass of paradigm 2. The idea is to use off-line calculations and analysis to obtain a pre-computed controller or solution, which may be, for example, a PID controller or a state feedback controller corresponding to the "optimal control" solution. The use of a pre-computed fixed controller, may be contrasted with conventional Model Predictive Control (MPC) which is more in spirit with paradigm 1. Here, the model is obtained off-line, but the estimation, model update and optimization is done on-line.

In summary, the focus of this contribution is to derive alternate, but effective implementation of the optimal policies by primarily exploiting the known properties of the optimal solution. We adopt the position that in principle, off-line calculations can be performed easily and on-line computations need to be minimized to achieve efficient on-line implementation. These ideas are further motivated in the following sections with examples. The examples are only illustrative and not exhaustive.

This paper deals with both steady-state optimization ($\dot{x} = 0$) in (1) and dynamic optimization problems. In the latter case, the addition of time as a variable adds to the complexity, but otherwise the main idea of using off-line calculations to obtain pre-computed solutions suitable for implementation remains the same.

2. INTRODUCING FEEDBACK TO THE OPTIMIZATION

Consider the optimization problem (1). As far as implementation is concerned, a naive possibility is to solve the optimization problem off-line at a nominal value of $d_0(t)$ for the disturbance and implement the resulting policy, $u_{opt}(t)$. This feedforward strategy, which is essentially an open-loop version of paradigm 1, is clearly not robust with respect to changing disturbances d or model uncertainty.

It is clear that some update (or feedback) needs to be introduced, and the alternative ways of implementation are closely related to how this is done. One approach is to use measurements to estimate disturbances and update the model, and then resolve the optimization problem on-line. This is the general (feedback) version of paradigm 1.

Another possibility is to pre-compute the solution for various disturbances, and use some interpolation and gridding strategy based on on-line measurements. This very general idea may work in certain cases.

A third general approach is to explore the mathematical conditions corresponding to optimal solution. For example, the steady state version of the optimization problem, i.e., $\dot{x} = 0$ in (1), is a typical nonlinear program (NLP) and the necessary conditions for optimality are given by the KKT conditions. One implementation policy is to control and enforce the KKT conditions directly. In principle, this leads to a simple feedback policy. However, it is very unlikely that a direct measurement of the KKT conditions (for e.g., the Lagrangian) is available. Another possibility is to obtain an explicit expression for the KKT condition (Cao, 2004). However, this expression generally involves unmeasured disturbances and states, so it would need to be combined with an on-line estimation procedure. In the dynamic case, the corresponding first-order optimality conditions are given by the Pontryagin Maximum Principle (Welz *et al.*, 2006). Again, the problems associated with measuring or estimating the necessary conditions are similar to those for the KKT conditions.

A fourth approach, which is studied in the main part of this paper, is to directly use the known structure of the solution of specific problem classes. In particular, in this paper, we focus on the control of active constraints and policies for identifying the correct region, and on the use and selection of "self-optimizing" unconstrained variables.

Example 1. Marathon runner. As an example of selecting self-optimizing controlled variables, consider the optimal policy employed by a marathon runner. There is one degree of freedom, which could be viewed as the power. Clearly, operating at maximum power is not optimal. The optimal policy is therefore unconstrained, and the issue is whether we can find a self-optimizing variable to keep constant. One feedback policy is to select speed as a controlled variable, and operate at constant speed. This policy may be acceptable on a flat track, but is not optimal on hilly terrain. Also, setting the constant (optimal) setpoint for the speed may be difficult. Another option is to select the heart rate or lactate level (sensed as pain) as a controlled variable. This works also on hilly terrain, and it seems that the operation is not very sensitive to the setpoint. This variable is easy to measure, and selecting a setpoint seems rather easy (Skogestad, 2004).

3. FINDING PRE-COMPUTED SOLUTIONS SUITABLE FOR ON-LINE OPTIMIZATION

In some cases, an explicit solution to the problem can be computed off-line, which can be implemented effectively, as in the following examples.

Example 2. Linear Quadratic Regulator (Optimal control). It is well known that the optimal solution to an infinite time dynamic optimization problem with a quadratic performance objective and linear dynamic model can be expressed as a time invariant feedback law: $u = -Kx$ (Kalman, 1963; Bryson, 1999). The optimal matrix K is obtained off-line by solving Riccatti equations. This alternate feedback representation $u = -Kx$ can be used for implementation and turns out to be much more robust to disturbances than the original open-loop solution. Interestingly, it seems that Kalman and others who first obtained this result, were not looking for a feedback controller but rather for the optimal (open-loop) solution. The realization that the optimal solution could be implemented in a feedback fashion was a serendipitous result.

Example 3. Explicit MPC. Recently, this idea has been extended to control of linear constrained systems (Pistikopoulos *et al.*, 2002). The authors show that the control law is a continuous piece-wise affine function of the states for the finite horizon problem (model predictive control) and the infinite time horizon problem (constrained linear quadratic regulation). The optimal solution is explicitly calculated off-line. The critical regions corresponding to each individual control law are polyhedral in shape. During implementation, given a measurement or estimate of the current states, the region of the partition to which the current state belongs is determined and the corresponding optimal controller is implemented.

Example 4. Elevator dispatch: A different example is that of elevator dispatch during uppeak traffic (Pepyne and Cassandras, 1997), where the authors show that the structure of the optimal dispatching policy that minimizes the average or discounted waiting time is a threshold based policy.

The determination of the constant gain K (in the optimal control problem) or the thresholds (in the elevator despatch problem described above) or the piece-wise controller (in explicit MPC) is non-trivial. However, these essentially involve off-line calculations and are examples of Result R3 mentioned in the introduction.

4. STRUCTURE OF THE OPTIMAL SOLUTION: ACTIVE CONSTRAINTS

In this section, we discuss several examples where the optimal solution can be characterized by a set of active constraints.

Example 5. Sprinter. Consider the "optimal operation" of a 100 m runner. It is clear that the optimal solution (and implementation) corresponds to running as fast as possible, i.e., the maximum power constraint is active.

Example 6. Linear Program. Consider a steady-state optimization problem. The simplest case is when the model equations, constraints and objective are linear (Linear Program). It is well known that the optimal solution (if it exists) is at a vertex of the simplex and therefore the optimal policy is to use all the inputs to satisfy the active constraints,

In the above examples, the optimal solution is such that all the degrees of freedom (inputs) u are used to satisfy active constraints.

Example 7. Network throughput maximization. Another example of control with constraints is the problem of maximizing throughput in a network. Under certain conditions, optimal operation of the plant is equivalent to maximizing throughput. The solution to the problem of determining maximum flow in a

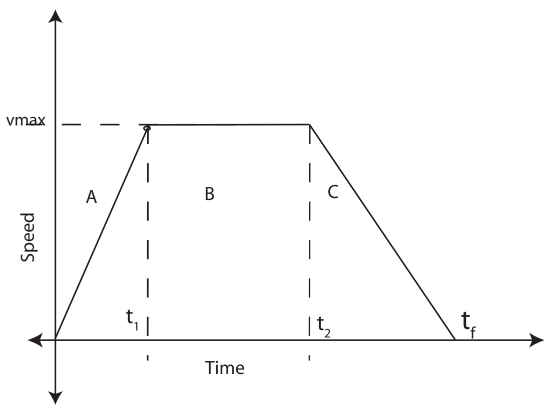

Fig. 2. Optimal speed profile

flow network has the property that maximal amount of a flow is equal to the capacity of a minimal cut (Nemhauser and Wolsey, 1999). Hence, the maximum flow is limited by a bottleneck and optimal operation can be achieved by focussing on the bottleneck. This results in the following implementation policy: identify the bottleneck and maintain maximum flow at the bottleneck (Aske *et al.*, 2006). The authors have described how this can be achieved by a control hierarchy using a coordinator MPC at the top and several local MPCs at the lower levels.

Example 8. Minimum time driving. Consider driving from the origin (at rest) to final destination (with final speed 0) in minimum amount of time while respecting the maximum speed limit. This is a dynamic optimization problem and the optimal speed profile would be as shown in Figure 2. The available manipulations are typically the acceleration, braking and times t_1 and t_2. As in the static optimization problem, it is possible to determine regions corresponding to active constraints: Regions A, B and C correspond to maximum acceleration, maximum speed and maximum deceleration (braking) respectively. The times t_1 and t_2 will be used to switch between regions.

It is possible that the set of active constraints and the corresponding optimal policy changes with disturbances or with time (as in Example 8). In fact, it may not even be feasible to continue to operate with the current strategy. The regions where a certain set of constraints remain active at optimality are called critical regions. For certain classes of optimization problems, these regions are polyhedral in shape and can be determined by treating the disturbances as parameters and employing parametric programming (Gal, 1979; Pistikopoulos *et al.*, 2002). In more general dynamic optimization problems, a solution model can be obtained by numerical optimization of a nominal model (Welz *et al.*, 2006) and the structure detected using an automated method (Schlegel and Marquardt, 2006).

In summary, it is common that most of the degrees of freedom are associated with satisfying active constraints. In more general problems, the optimum can be unconstrained. Further unconstrained degrees of freedom can be used to control the self-optimizing variables as discussed previously or to switch between regions which will be discussed subsequently.

5. SWITCHING POLICY BETWEEN DIFFERENT REGIONS

To avoid on-line optimization, one may use off-line calculations to 1) determine the set of regions with different active constraints and 2) device rules for identifying when to switch between regions. The first issue of determining the set of different regions was discussed previously. The second issue is the switching policy which is discussed in this section. One systematic approach for detecting events and switching is automata theory which has been used for modelling, control and diagnosis of discrete event systems (Phillips, 2001). However, this may be complicated and so simpler schemes may be preferred. In most problems, switching may usually be effected by either:
(Case A) observing when a new constraint is reached,
(Case B) observing when the self-optimizing unconstrained variable reaches its setpoint.
(Case C) In the dynamic case, we also have switching caused by future constraints.

5.1 *Static optimization*

For a linear program only case A can occur, whereas case B can occur for a quadratic program. This does not necessarily mean that it is easy to determine when to switch because the regions can be very complex. In the following example (Ex. 9), we consider a class of problems where 2 manipulations, called primary and secondary are combined together. These pairings are chosen so that when the primary manipulation saturates, the secondary manipulation is used. More detailed information on implementing this switching policy using split range control can be found in (Lersbamrungsuk *et al.*, 2006; Glemmestad, 1997).

Example 9. Optimal operation of heat exchanger networks. The general mathematical model of a Heat Exchanger Network (HEN) is non-linear and hence, optimization of the same would involve solving a NLP. However, under certain conditions, the problem of optimal operation of a HEN can be reformulated as a Linear Program (Aguilera and Marchetti, 1998; Lersbamrungsuk *et al.*, 2006) with obvious advantages. The following HEN (Fig. 3) is a modified example from Aguilera and Marchetti (1998).

The network consists of 3 process-process exchangers, 3 utilities and 4 streams. The outlet temperatures of all streams are to be controlled and maintained at target values. Inlet temperatures of all streams are assumed

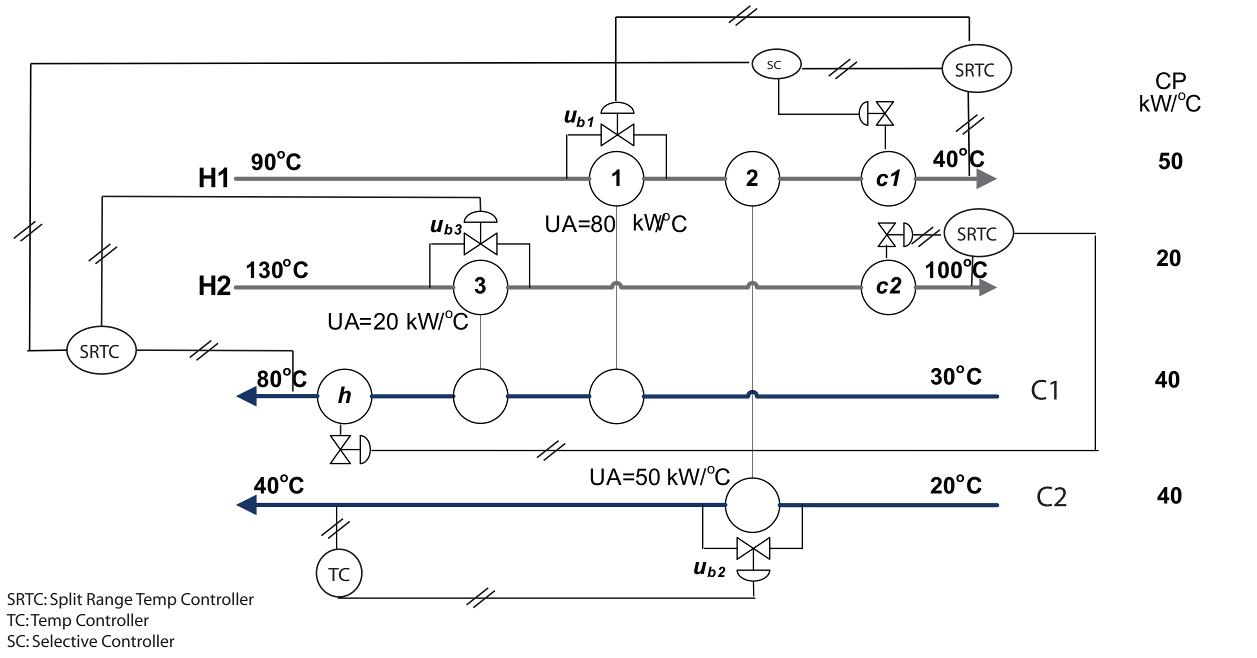

Fig. 3. Control structure for optimal operation of HEN(Lersbamrungsuk *et al.*, 2007)

to be unmeasured disturbances. Nominal inlet and outlet temperatures (targets) and other relevant data are indicated in the figure. The utility costs are 0.05 $/kWh$ for Q_h, 0.02 $\$/kWh$ for Q_{c1} and 0.01 $\$/kWh$ for Q_{c2} respectively. The objective function to be minimized is the overall utility consumption. Following is a table listing which manipulations are saturated in different regions of the disturbance space, where A and I denotes that the corresponding manipulation has saturated and not saturated respectively (Lersbamrungsuk *et al.*, 2007). This can be determined by numerical optimization for different values of the disturbance or parametric programming.

Table 1. List of saturated manipulations

Set of active constraints	Q_{c1}	Q_{c2}	Q_h	u_{b1}	u_{b2}	u_{b3}
1	A	I	A	I	I	I
2	A	A	I	I	I	I
3	I	A	I	A	I	I
4	I	I	A	A	I	I
5	I	I	A	I	I	A

Since the optimal operation problem is a LP, the optimal solution is at a vertex and available degrees of freedom are used to control active constraints. u_{b2} does not saturate and so does not appear in a split range and is used to control T_{c2out}. Q_{c2} and Q_h are combined in a split range pair to control T_{h2out}, u_{b3} is used to control T_{c1out}, u_{b1} is used to control T_{h1out} and Q_{c1} is used when either u_{b1} or u_{b3} are saturated. This results in the control structure as shown in Fig. 3 (Lersbamrungsuk *et al.*, 2007). In a general problem, the manipulations that need to be combined in a split range structure can be determined by solving an ILP (Lersbamrungsuk *et al.*, 2007). It must be noted that it may not be possible to implement this switching strategy for all classes of linear programs. Dynamic control performance would be important if there were hard constraints on the outputs. In this example, this would imply hard constraints on the target temperatures. Usually, these are not hard constraints and one is only interested in ensuring that the target temperatures are attained in the "average" which is possible using integral action in the controllers.

5.2 *Dynamic optimization*

Unlike in static optimization problems, switching between regions can occur based on a future constraint (Case C mentioned in the introduction to Section 5). This has to be predicted based on an estimate of the current state and the disturbances and prediction of disturbances. However, it may be possible to use feedback solutions, for example based on self-optimizing control. We explore this further by reverting to some of the examples discussed previously.

Example 3, Explicit MPC, contd. In explicit MPC, the problem of switching is addressed by using the estimated or measured value of the current state (Pistikopoulos *et al.*, 2002). Each region of the state space partition corresponds to a set of constraints that are active. However, some of these could be future constraints and so, there is no direct feedback connected from the actual constraint, but only from the measurement (or estimate) of the current state and the local feedback controller.

Example 8, Minimum time driving, contd. It is clear that the switch from regime A (acceleration) to regime B (maximum speed) occurs when the maximum speed is reached (case A). However, moving from regime B to regime C (deceleration) is not trivial because it is associated with a future constraint (case C). One solution is to use an off-line policy, i.e., start decelerating at time t_2, which is however not robust and may also result in infeasibility. Another solution is to use feedback, based on the measurement of distance to the finish. More generally, the idea of self-optimizing control can be used to determine the most suitable measurement or measurement combination c, so that the switching occurs when c reaches a given value.

6. DISCUSSION

The approach of implementing optimal operation by tracking the necessary conditions of optimality (NCO) given by the Pontryagin Minimum Principle proposed by Srinivasan and Bonvin (2007) is similar in spirit to Paradigm 2. They refer to their scheme and the conventional method of on-line optimizing control (Paradigm 1) as implicit and explicit methods respectively. However, this nomenclature can be misleading as the pre-computed explicit solution to MPC is referred to as explicit MPC in literature and naturally would be classified as an example of Paradigm 2.

7. CONCLUSIONS

We discussed with examples an alternate to the conventional paradigm of on-line optimizing control, viz., using off-line computations and feedback solutions for efficient on-line implementation. It may not be known *a priori* if an alternate implementation of the optimal solution following Paradigm 2 is possible for the given problem and hence it is natural to analyze classes of systems. The alternate paradigm was motivated with several examples and demonstrated using a particular case of linear programs. Current and future research is focussed on wider classes of systems, including but not limited to quadratic and general convex programs and batch processes such as distillation. These ideas can be extended to more general problems using local linear, quadratic or convex approximations of the cost function.

Acknowledgment

The authors would like to thank V. Lersbamrungsuk for assistance in generating the results for the HEN and Håkon Dahl-Olsen and Henrik Manum for valuable discussions.

8. REFERENCES

Aguilera, N. and J.L. Marchetti (1998). Optimizing and controlling the operation of heat-exchanger networks. *AIChE Journal* **44**(5), 1090–1104.

Aske, E., S. Strand and S. Skogestad (2006). Coordinator MPC with a focus on maximizing throughput. In: *16th European Symposium on Computer Aided Process Engineering and 9th International Symposium on Process Systems Engineering*. pp. 1203–1208.

Bryson, A.E. (1999). *Dynamic optimization*. Addison Wesley.

Cao, Y. (2004). Constrained self-optimizing control via differentiation. In: *Proceedings of the 7th International Symposium on Advanced Control of Chemical Processes (ADCHEM). Hong Kong*. pp. 63–70.

Findeisen, W., F.N. Bailey, M. Bryds, M. Malinowski, P. Tatjewski and A. Wozniak (1980). *Control and Coordination in Hierarchical Systems*. John Wiley and sons.

Gal, T. (1979). *Postoptimal analyses, parametric programming and related topics*. McGraw-Hill.

Glemmestad, B. (1997). Optimal operation of integrated processes: Studies on Heat Recovery Systems. PhD thesis. Norwegian University of Science and Technology.

Kalman, R.E. (1963). When is a linear control system optimal? In: *Joint Automatic Control Conference, Minneapolis, USA*.

Lersbamrungsuk, V., S. Narasimhan, S. Skogestad and T. Srinophakun (2007). Control structure design for optimal operation of heat exchanger networks. *In preparation*.

Lersbamrungsuk, V., S. Skogestad and T. Srinophakun (2006). A simple strategy for optimal operation of heat exchanger networks. In: *International Conference on Modeling in Chemical and Biological Engineering Sciences, Bangkok, Thailand*.

Nemhauser, G.L. and L.A. Wolsey (1999). *Integer and combinatorial optimization*. John Wiley and Sons.

Pepyne, D.L and C.G. Cassandras (1997). Optimal dispatching control for elevator systems during uppeak traffic. *IEEE Transactions on Control Systems Technology* **5**, 629–643.

Phillips, P. (2001). Modelling, control and fault detection of discretely observed systems. PhD thesis. TU Eindhoven.

Pistikopoulos, E.N., D. Dua, N.A. Bozinis, A. Bemporad and M. Morari (2002). On-line optimization via off-line parametric optimization tools. *Computers and Chemical Engineering* **26**, 175–185.

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

Skogestad, S. (2004). Near-optimal operation by self-optimizing control: From process control to marathon running and business systems. *Computers and Chemical Engineering* **29**, 127–137.

Srinivasan, B. and D. Bonvin (2007). Real-time optimization of batch processes by tracking the necessary conditions of optimality. *I&EC Research* **46**, 492–504.

Welz, C., A. Srinivasan, B. Marchetti, D. Bonvin and N.L. Ricker (2006). Evaluation of input parameterization for batch process optimization. *AIChE Journal* **52**, 3155–3163.