

ON-LINE OPTIMIZATION OF BATCH PROCESSES BY TRACKING STATES AND COSTATES

E. Visser, B. Srinivasan, S. Palanki*, D. Bonvin

*Institut d'automatique, École Polytechnique Fédérale de Lausanne,
CH-1015 Lausanne, Switzerland.*

**Florida State University, Tallahassee, Florida.*

Abstract. The terminal-cost optimization of a control-affine nonlinear system via Pontryagin's minimum principle leads to a non-smooth solution which can be characterized in a piecewise manner. To implement such an optimal trajectory despite disturbances, a *cascade optimization scheme* is proposed, where a given optimal set point is tracked. In this paper, the question of the choice of outputs to track is addressed. Optimality is achieved by a piecewise definition of outputs to track, i.e., the choice of appropriate outputs (input bounds, state constraints, and switching functions) in various sub-intervals. The main difficulty of the approach lies in the estimation of the costates, a solution for which is analyzed. These issues are illustrated on the optimization of a reversible batch reaction system and a biofilter used in wastewater treatment.

Keywords. On-line optimization, Uncertainty, Feedback control

1. INTRODUCTION

A wide variety of specialty chemicals are made in batch reactors. To cope with competition, it is important to operate them in an optimal manner. The optimal operating policy for a given batch process is usually calculated under the assumption of a perfect model. However, realistic applications are subject to uncertainty in initial conditions, model mismatch, and process disturbances, all of which affect the optimal solution. This provides the motivation for *on-line* calculation and implementation of the optimal operating policy.

Two different approaches to on-line optimization have been proposed in the literature:

Repeated optimization: In this approach, an optimizing control law is calculated that updates the optimal inputs by solving a finite horizon optimization problem at each time step. The solution is computed numerically (Eaton *et al.*, 1990) or analytically (Palanki *et al.*, 1993). Some of the drawbacks of this approach include high computational burden, especially in the presence of state and input constraints, and possible infeasibility of the obtained solution.

Cascade optimization: The optimal set-point trajectory that ensures optimal performance is computed. A 'low level' tracking controller ensures that the system does not stray very far away from the optimal trajectory (Srinivasan *et al.* 1997, Krothapally and Palanki, 1997). In addition, a 'high level' optimizer is invoked periodically to ensure optimality despite disturbances.

The cascade optimization approach combines the positive features of optimal control and feedback control. The basis of the cascade optimization scheme is tracking and hence the most important question is "What to track?". This will be the main issue addressed in this paper. Since there are typically more states than inputs, one cannot guarantee that all the state trajectories will be accurately tracked in the presence of disturbances.

It is shown in this paper that, in general, there does not exist one single combination of states that can be tracked during the entire time interval. This, hence, calls for a piecewise definition of outputs, i.e., different combinations of states/inputs for different time intervals.

The paper is organized as follows. Section 2 formulates the problem. Section 3 explains the cascade optimization framework in detail. The selection of outputs to track is treated in Section 4, and Section 5 discusses the estimation of costates. Section 6 gives two simulated examples, and Section 7 concludes the paper.

2. PROBLEM FORMULATION

The end-point optimization of a nonlinear, control-affine batch process can be mathematically formulated as:

$$\begin{aligned} \min_{u(t)} J &= \phi(x(t_f)) \\ \text{s.t.} \quad \dot{x} &= f(x) + \sum_{i=1}^m g_i(x)u_i, \quad x(0) = x_0, \\ S(x, u) &\leq 0, \quad T(x(t_f)) \leq 0, \end{aligned} \tag{1}$$

where u is the m -vector of manipulated inputs, x is the n -vector of states, t_f is the final time, ϕ is a smooth scalar function, f and g_i are smooth vector functions, $S(x, u)$ is a σ -dimensional vector of path constraints, and $T(x(t_f))$ is a τ -dimensional vector of terminal constraints. Also, the vector fields g_i are assumed to be of full rank for all x .

The classical approach to solving Problem (1) is via the application of Pontryagin's minimum principle (Bryson & Ho, 1975), which is reviewed below. Problem (1) is equivalent to minimizing the Hamiltonian:

$$H(x, u, \lambda, \mu) = \lambda^T \left[f(x) + \sum_{i=1}^m g_i(x) u_i \right] + \mu^T S(x, u)$$

$$\dot{\lambda}^T = -\frac{\partial H}{\partial x}, \quad \lambda(t_f) = \frac{\partial \phi}{\partial x} \Big|_{t_f} + \nu^T \frac{\partial T}{\partial x} \Big|_{t_f} \quad (2)$$

where λ is the vector of costates, μ the Lagrange multipliers for the state constraints and input bounds, and ν the Lagrange multipliers for the terminal constraints. The necessary conditions for optimality are $H_u = 0$ or:

$$\frac{\partial H}{\partial u_i} = \lambda^T g_i(x) + \mu^T \frac{\partial S}{\partial u_i} = 0, \quad i = 1, \dots, m \quad (3)$$

The optimal control problem is **singular** when the inputs cannot be determined directly by the necessary conditions as is the case here. For such problems, the optimal solution has the following properties:

- The inputs are in general discontinuous, yet inputs are analytic between discontinuities.
- The solution between two discontinuities will be referred to as an *arc*. Three types of arcs are possible:
 - (1) inputs determined by active input bounds,
 - (2) inputs determined by active state constraints,
 - (3) singular arc, when inputs are not determined by any of the active constraints.
- Analytic expressions for these arcs can be obtained, though the sequence and switching times have to be computed numerically in most cases.

A piecewise analytic characterization of the optimal inputs helps both improve the computational efficiency and choose the implementation strategy.

Whether or not an arc is singular depends on the function $\psi_i = \lambda^T g_i(x)$, which is referred to in the literature as the **switching function**. This function vanishes over the singular time interval. Outside the singular interval, the manipulated input $u_i(t)$ is on a state constraint or an input bound.

Methods for calculating the singular arcs are available in the optimal control literature (Palanki *et al.*, 1993). The idea is as follows: Since the switching function is zero over a time interval, its derivatives with respect to time are also zero. Hence, a sequence of time differentiations

is performed until the inputs $u_i(t)$ appears explicitly. The resulting expression is then solved for $u_i(t)$ in terms of x and λ .

3. OPTIMIZATION FRAMEWORK

A *cascade optimization* structure is proposed to incorporate feedback into the optimization framework (Figure 1). The 'high level' optimizer solves the optimization problem and selects the appropriate outputs to track for specific time intervals. Thus, it provides (i) the feedforward inputs, u^* , (ii) the reference signal, y^* , and (iii) the switching strategy between various subsequent outputs. The optimizer constitutes the outer loop and is indicated by the thin lines in Figure 1. The reference signal of the corresponding output is then tracked with the help of the 'low level' feedback controller (inner loop - thick lines in Figure 1).

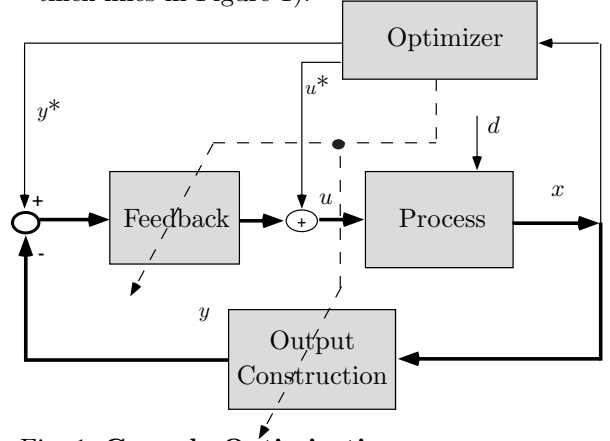


Fig. 1. **Cascade Optimization**

Due to the presence of the feedforward term u^* , the feedback is inactive in the absence of uncertainties (model mismatch, disturbances). However, in the presence of small uncertainties, the feedback ensures that the system does not stray far away from the optimal trajectory. To ensure optimality despite large uncertainties, the reference signal and switching strategy of the optimizer can be updated during the course of a run. In addition, if runs are repeated, the optimizer can adapt itself on a run-to-run basis.

Though the *cascade optimization* framework is quite general, most of the issues that follow will be restricted to terminal-cost optimization of control-affine nonlinear systems. The restriction is motivated by the fact that powerful geometric control concepts can be utilized for such systems.

The two goals of the optimizer are: (i) select appropriate outputs to track, and (ii) calculate the reference signals numerically. The next step is the design of the feedback part. When the solution lies on the feasibility boundaries, there is no maneuverability to implement the feedback. In particular, care should be taken to ensure that

the constraints are not violated. This calls for the introduction of conservatism.

The main issue addressed in this paper is the first question raised concerning the optimizer, i.e., the choice of outputs to track. The other issues will be treated elsewhere.

4. SELECTION OF OUTPUTS TO TRACK

The ‘high level’ optimizer provides an open-loop solution which consists of input and state trajectories that minimize the objective function at the final time. In the absence of uncertainty, the tracking of *any* state will result in all the other states evolving on their optimal trajectories. However, in the presence of uncertainty, this is no longer true. In fact, since there are m inputs, only m states or combination of states (outputs) can be kept at desired values over time.

Secondly, since the disturbances take the system to a state different from that expected in the nominal condition, one is interested to know the optimal trajectory from that new operating point onwards. Is tracking still a viable option to achieve optimality? If so, what are the outputs or the combination of states that need to be tracked? The answer to this question is provided in the following theorem.

Theorem 1. Consider the end-point optimization problem (1) for which measurements or estimates of all the states and costates are available. Feedback optimality is achieved by:

- (1) Open-loop application of the input when an input bound is active
- (2) Ideal tracking of the state constraint when a state constraint is active
- (3) Ideal tracking of switching functions during a singular interval

along with appropriate switching between the various sub-intervals.

Proof: The problem of feedback optimality is equivalent to that of satisfying the necessary conditions, which in turn, is reformulated as the problem of tracking $H_u = 0$. Tracking $H_u = 0$ has different interpretations with respect to the three types of arcs and is discussed next.

In the case where an input is determined by its bound, let us assume, without loss of generality, that the bound corresponding to μ_1 is active, i.e., $H_{u_i} = \psi_i + \mu_1 = 0$. As long as $\psi_i < 0$, $H_{u_i} = 0$ implies that $\mu_1 = -\psi_i$. The optimal solution remains on the active input bound as long as μ_1 is non-zero.

If the input is determined by a state constraint, following the same argument, it is optimal to keep the constraint

active as long as the ψ_i does not change sign.

In the input bound case, it is straightforward to determine the input that keep the constraint active. On the contrary, an algorithmic approach is required to keep a state constraint active. Hence, optimality is achieved by choosing the output $y = S(x, u)$ with a zero set point.

For a singular arc, Condition (3) reduces to $H_{u_i} = \psi_i = 0$. Therefore, the choice of $y = \psi_i$ with a zero set point ensures optimality. \square

An important point to note is that there is no single combination of states that can be tracked during the entire optimization interval. This is due to the fact that tracking $H_u = 0$ means tracking different combinations of states in different types of intervals.

The result provided in Theorem 1 not only chooses the outputs to track but also performs the input-output pairing. An active state constraint or the switching function is differentiated with respect to time until an input appears. The first input that appears is then paired with the appropriate output.

The numerical computation of the optimal solution has two parts, i.e., (i) the switching instants, and (ii) the value of the inputs between the switching instants. The theorem presented above deals only with the second problem, while the first remains an area of active research.

Having selected the outputs to track, the first issue that needs to be addressed is whether or not the outputs can be tracked. Remark 1 below addresses this issue and concludes that, though the outputs are controllable, one may need margins to ensure internal stability. Remark 2 discusses circumventing the estimation of the costates which is the main bottleneck of the proposed approach. Remark 3 deals with the problem of switching between different arcs.

Remark 1: Controllability: The outputs chosen for tracking should be controllable. A recent result (Srinivasan *et al.* 1999) states that input-affine nonlinear systems *always* lose first-order differential controllability along the optimal solution. This implies that a mode exists in the optimal solution that cannot be controlled to its desired trajectory. However, it can be shown that the outputs chosen by Theorem 1 are controllable. This can be seen as follows: (i) During open-loop application, the issue of controllability does not arise. (ii) When on a state constraint ($y = S(x, u)$), or along a singular arc ($y = \psi_i$), the inputs can be obtained by repeated differentiations of y . Thus y is controllable because of this explicit relationship between u and y . Yet, the internal modes could be unstable, but that issue is not discussed here.

Remark 2: Relative Degree: Tracking $y = \psi_i$ requires a good estimate of the switching functions which is often difficult to obtain due to incorrect initialization or integration errors. In such a case, tracking a controllable state with the highest possible relative degree is a good option. The relative degree indicates how many integrations an input has to go through to affect an output. In other words, it is a measure of how ‘indirectly’ an input affects an output. In conventional regulatory controller design, the outputs are fixed and the designer has the choice to pick the inputs. In this situation, one tends to pick the input that have the *lowest* relative degrees. In contrast, in the design of tracking controllers for end-point optimization, the inputs are fixed (by the optimizer) and the designer can select the outputs. Since the objective is to keep as many state trajectories as possible on track, one would clearly select the outputs that result in the *highest* possible relative degrees. Note that the relative degree is for a specific input-output pair. To summarize, for each singular input, one needs to identify and track the output which has the highest relative degree with respect to that input.

Remark 3: Switching between arcs: Theorem 1 assumes “appropriate” switching between various types of arcs. This mostly has to be done numerically and hence calls for a periodic reoptimization. However, in some cases, it is possible to identify the conditions under which the switching between arcs takes place (illustrated in the examples below - Section 6). In such a case, it is easier to implement an on-line optimization without any explicit reoptimization.

5. ESTIMATION OF COSTATES

The major problem in the proposed approach is the estimation of the costates for tracking the switching function. For the purpose of this section, it is assumed that all the states are available. If full state measurement is not available, an appropriate state estimator needs to be set up.

The best scenario is when analytical expressions for the costates in terms of the present states are available, as in the ‘sweep method’ ($\lambda = \mathcal{P}x$) used in the linear quadratic regulator.

The simplest case is to start with the initial condition $\lambda(0)$ given by the optimizer and integrate it forward through the costate system (2), i.e., $\dot{\lambda}(t) = -A^T(x)\hat{\lambda}(t)$, where $A(x) = \left[\frac{\partial f}{\partial x} + \sum_{i=1}^m \frac{\partial g_i}{\partial x} u_i \right]$. The first question is whether to use the nominal states x^* or the actual states x for the computation of $A(x)$. To take the uncertainties into account one needs to use the actual state, but having done so, the λ trajectory deviates from its nominal value, so that there is no guarantee that the final

condition on λ is met. Note that it is more important to satisfy the final condition on λ than the initial condition. Hence, a closed-loop observer is proposed to force the costates at the final time to their desired values.

To explain the idea suggested here, consider the case where all inputs are singular during the entire interval with no active terminal constraint. An estimate $\hat{\lambda}(t)$ of $\lambda(t)$ is obtained with the observer

$$\dot{\hat{\lambda}}(t) = -A^T(x)\hat{\lambda}(t) - K(\hat{\lambda}(t_f|t) - \bar{\lambda}(t_f|t)) \quad (4)$$

The states and costates at terminal time are predicted using a state transition matrix, \mathcal{T}_{t,t_f} , to avoid repeating explicit integration of state and costate dynamic equations until final time at each time step. Note that the costate transition matrix is the inverse of the state transition matrix. From the terminal states $\hat{x}(t_f|t)$, the costates required for the sake of optimality, $\bar{\lambda}(t_f|t)$, are also calculated.

$$\mathcal{T}_{t,t_f}(x) = e^{\int_t^{t_f} A(x(\tau)) d\tau} \quad (5)$$

$$\hat{x}(t_f|t) = x^*(t_f) + \mathcal{T}_{t,t_f}(x(t) - x(t)^*) \quad (6)$$

$$\hat{\lambda}(t_f|t) = \mathcal{T}_{t,t_f}^{-1}(x) \hat{\lambda}(t) \quad (7)$$

$$\bar{\lambda}(t_f|t) = \frac{\partial \phi}{\partial x} \Big|_{x(t_f)=\hat{x}(t_f|t)} \quad (8)$$

Equation 4 shows that estimation of the costates is done by comparing the costates predicted by the evolution and the costates required for optimality at terminal time. The matrix K is used to control the speed of correction, with $K = 0_{n \times n}$ corresponding to open-loop prediction.

The next complexity lies in the determination of $\mathcal{T}_{t,t_f}(x)$. As a first approximation, the nominal trajectory can be used to calculate the state transition matrix, i.e., $\mathcal{T}_{t,t_f}(x) \approx \mathcal{T}_{t,t_f}(x^*)$. In some cases, this is insufficient, and a first-order Taylor series expansion can be used $\mathcal{T}_{t,t_f}(x) \approx \mathcal{T}_{t,t_f}(x^*) + (x - x^*) \frac{\partial \mathcal{T}}{\partial x}$.

When singular arcs are concatenated with other types of arcs, then prediction of the costates at final time becomes more involved. In such a case, one can use the costates at the end of the singular interval instead of final time. The issue gets even more involved in the presence of state constraints, since the costates can be discontinuous when entering or leaving the constraints.

6. SIMULATED EXAMPLES

Two examples are presented below: The first one illustrates the concept while the second one shows that this scheme can indeed bring about considerable gains in performance.

6.1 Example 1: Reversible reaction

To illustrate the concepts presented in this paper, optimization of the exothermic reversible batch reaction system $2A \rightleftharpoons B$ is considered. The rate of the forward reaction is second order, i.e., $r_f = k_f C_A^2$, while the reverse reaction is first order, $r_r = k_r C_B$. From material and energy balances, the system equations are given by:

$$\begin{aligned} \dot{C}_A &= -2 \dot{C}_B = -2 (r_f - r_r) \\ \dot{T} &= \frac{-\Delta H}{\rho C_p} (r_f - r_r) - \frac{k_w A_r (T - u)}{V_r \rho C_p} \\ k_f &= k_{f0} e^{\left(\frac{-E_f}{R T}\right)}, \quad k_r = k_{r0} e^{\left(\frac{-E_r}{R T}\right)} \end{aligned} \quad (9)$$

where C_A and C_B are the concentrations of the species A and B , T the temperature, ΔH the heat of reaction, ρ and C_p the density and heat capacity of the reaction medium, A_r and V_r the reactor heated wall surface and volume and k_w the heat transfer coefficient of the reactor wall. The input u represents the temperature of the reactor jacket cooling fluid. The rate constants k_f and k_r are given by Arrhenius expressions, where E_f and E_r are the corresponding activation energies and R the gas constant.

It is desired to adjust the jacket temperature to maximize the production of C_B subject to the safety constraint $T \leq T_{max}$ and the operational constraint $u_{min} \leq u \leq u_{max}$. The optimization problem then reads:

$$\begin{aligned} \min_{u(t)} J &= -C_B(t_f) \\ \text{s.t. (9), } T &\leq T_{max}, \quad u_{min} \leq u \leq u_{max} \end{aligned} \quad (10)$$

where $x = [C_A \ T]^T$, and f and g are defined appropriately. Numerical values are given in Table 1.

The optimal input profile consists of three parts: i) the input is determined by the upper bound, ii) the input is determined by state constraint, and iii) a singular arc.

Initially, the jacket temperature is on the upper bound to bring the temperature quickly to T_{max} . The switching between input bound and state constraint occurs when $T = T_{max}$. Once the state constraint is reached, the solution stays on that constraint. The solution eventually switches to the singular region when the reaction rates obey, $E_f r_f = E_r r_r$ (Srinivasan *et al.* 1999). This condition reflects the optimal unconstrained dynamic equilibrium between the forward and the reverse reactions.

To compare the performances of the open-loop nominal solution and the proposed cascade optimization scheme under uncertainty, a change in the initial conditions was considered: $C_A(0) = 9 \frac{\text{kmol}}{\text{m}^3}$, $T(0) = 15^\circ\text{C}$.

We assume that full state information is available. For the cascade scheme, the input is first applied open-loop until the temperature reaches T_{max} . Then, PI-controllers are used to track (i) the temperature when $T = T_{max}$, and (ii) the switching function when $E_f r_f = E_r r_r$. The costates were estimated using (4), with the state transition matrices being obtained from the nominal trajectory.

A simulation run is illustrated in Figures 2-5. Note that the cascade optimization scheme adjusts the switching times between the different optimal arcs. It can be seen from Figure 2 and 3 that the switching to state constraint tracking is delayed and the singular arc is also started later compared to open-loop implementation. The switching function does not converge to zero in the open-loop implementation, thus resulting in a loss of optimality (Figure 4). Figure 5 show the switching function which is correctly forced to zero with tracking. Although the benefit of on-line optimization is very low here ($1.728 \frac{\text{kmol}}{\text{m}^3}$ of B versus 1.724 in the open-loop case), it illustrates how on-line optimization can be achieved by the proposed cascade optimization scheme, without reoptimization.

6.2 Example 2: Biofilter

The process is an aerobic biofilter that is operated as a co-current ascending column, that is, wastewater and process air are pumped in the bottom and leave the column at the top (Benthack, 1997). The following reduced model approximates the reactor operation reasonably well:

$$\begin{aligned} \dot{s}_1 &= -k_q Q (s_1 - S_{in}) - k_r s_1 m_b \\ \dot{s}_2 &= -k_q Q (s_2 - s_1) - k_r s_2 m_b \\ \dot{s}_3 &= -k_q Q (s_3 - s_2) - k_r s_3 m_b \\ \dot{s}_4 &= -k_q Q (s_4 - s_3) - k_r s_4 m_b \\ \dot{m}_b &= k_b (s_1 + s_2 + s_3 + s_4) m_b \end{aligned} \quad (11)$$

where s_1 to s_4 represent the substrate concentration at 4 different locations along the height of the reactor, s_4 being the substrate outlet concentration. S_{in} is the concentration of the inlet wastewater. m_b is the overall biomass concentration. The system parameters and initial conditions are listed in Table 1. System (11) is affine in the input $Q(\frac{m^3}{h})$.

The objective is to maximize the amount of biomass in the reactor at the end of one cycle ($t_f = 40$ h) by adjusting the input flowrate $Q(t)$

$$\begin{aligned} \min_{Q(t)} J &= -m_b(t_f) \\ \text{s.t. (11), } 0 &\leq Q(t) \leq 10 \quad s_4 \leq 0.1 \end{aligned} \quad (12)$$

It can be shown using heuristic arguments that the optimal input profile keeps s_4 on its upper bound (Benthack, 1997). As the reactor has to be filled before the state constraint can be met, the constraint tracking phase is preceded by a filling phase realized in minimum time to bring s_4 to its upper bound without overshoot.

The actual process parameter k_b is now considered to be 0.0782. The profiles for state s_4 and input Q resulting from open-loop and closed-loop operation are shown in Figures 6 and 7. Whereas s_4 drifts away from its optimal value when the process is run open-loop, the cascade optimization scheme corrects the optimal input by tracking the state constraint with a suitably tuned PI-controller. During the filling phase, no feedback is possible. The performance in the closed-loop case is $5.55 \frac{kg \text{ DW}}{m^3}$ compared to 4.89 in the open-loop case, i.e. a significant 13.5% improvement.

7. CONCLUSIONS

Some implementation issues in the terminal-cost optimization problem of nonlinear affine-in-input systems in the presence of uncertainty were addressed. The proposed methodology was based on tracking appropriate outputs, with the key idea being the use of measurements (full/partial state) to compensate for the uncertainty. The choice of outputs for the sake of optimality was addressed in this paper.

An approximation of the costate vector was used to compute the switching functions. Future research is necessary to obtain more accurate approximations of the costates. In the examples presented, it was possible to accurately determine the switching instants. However, such a situation is not always possible, and it is necessary to analyze how such cases can be handled.

8. REFERENCES

- Eaton, J.W., Rawlings, J.B., *Feedback Control of Nonlinear Processes using On-Line Optimization Techniques*, Comp.chem.Eng., 14, 469-479, 1990
- Palanki, S., Kravaris, C., Wang, H.Y., *Synthesis of State Feedback Laws for End-Point Optimization in Batch Processes*, Chem.Eng.Science, Vol 48, 1, 135-152, 1993
- Benthack, C., *Feedback-based Optimization of a Class of Constrained Nonlinear Systems: Application to a Biofilter*, Ph.D thesis, EPFL, Lausanne 1997
- Srinivasan, B., Visser, E., Bonvin, D., *Optimization-based Control with Imposed Feedback Structures*, Proceedings ADCHEM'97, 635-640, Banff 1997
- Krothapally, M., Palanki, S., *A Neural Network Strategy for Batch Process Optimization*, Comp.chem.Eng., 21, 463-468, 1997
- Bryson, A.E., Ho, Y.-C., *Applied Optimal Control, Optimization, Estimation and Control*, Revised Printing, Hemisphere Publishing Corporation, USA, 1975

Srinivasan, B., Visser, E., Bonvin, D., *On the Controllability of Optimal Trajectories*, Internal Report, Institut d'Automatique, EPFL, Switzerland, 1999

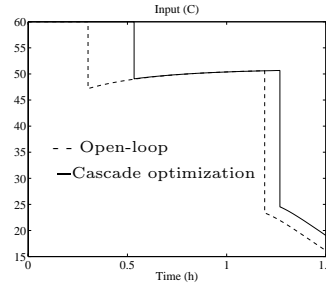


Fig 2: Optimal input

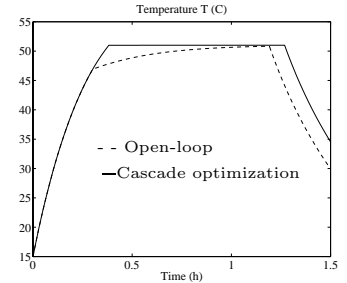


Fig 3: Temperature

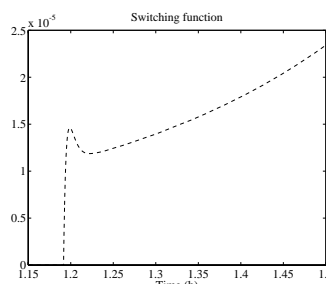


Fig 4: Open-loop switch. funct.

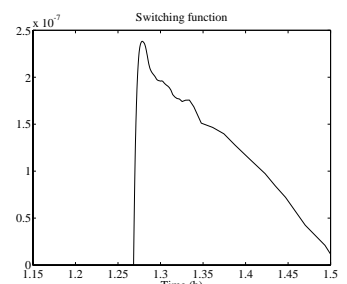


Fig 5: Closed-loop switch. funct.

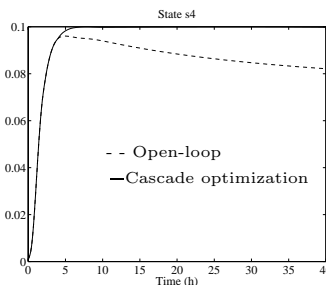


Fig 6: Tracked state

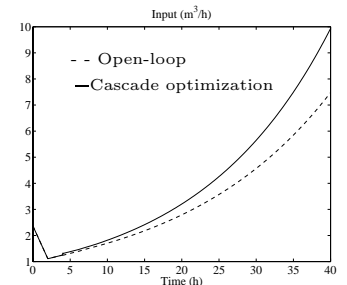


Fig 7: Optimal input

	Ex. 1		Ex. 2	
k_{f0}	10^2	$\frac{m^3}{kmol \cdot h}$	k_r	0.5851 $\frac{m^3}{kg \text{ DW} \cdot h}$
k_{r0}	5×10^3	$\frac{1}{h}$	k_b	0.0682 $\frac{m^3}{kg \text{ COD} \cdot h}$
E_f	2×10^4	$\frac{kJ}{kmol \cdot K}$	k_q	0.7874 $\frac{m^3}{m^3}$
E_r	2.3×10^4	$\frac{kJ}{kmol \cdot K}$	S_{in}	0.4 $\frac{kg \text{ COD}}{m^3}$
$-\Delta H$	2×10^4	$\frac{kJ}{kmol}$	$s_1(0)$	0.02 $\frac{kg \text{ COD}}{m^3}$
ρ	10^3	$\frac{kg}{m^3}$	$s_2(0)$	0.01 $\frac{kg \text{ COD}}{m^3}$
C_p	4.18	$\frac{kJ}{kg \cdot K}$	$s_3(0)$	0.005 $\frac{kg \text{ COD}}{m^3}$
A_r	5	m^2	$s_4(0)$	0.001 $\frac{kg \text{ COD}}{m^3}$
V_r	1	m^3	$m_b(0)$	0.6 $\frac{kg \text{ DW}}{m^3}$
k_w	3×10^3	$\frac{kJ}{h \cdot m^2 \cdot K}$	t_f	40 $\frac{m^3}{h}$
T_{max}	51	$^{\circ}C$		
u_{min}	10	$^{\circ}C$		
u_{max}	60	$^{\circ}C$		
t_f	1.5	h		
$C_A(0)$	10	$\frac{kmol}{m^3}$		
$C_B(0)$	0	$\frac{kmol}{m^3}$		
$T(0)$	25	$^{\circ}C$		

Table 1. Parameters and initial conditions

DW =
Dry
Weight

COD =
Chemical
Oxygen
Demand