

## **Dynamic Optimisation of Batch Processes by Integrated Two-Time-Scale Scheme**

**Marián Podmajerský, Benoit Chachuat<sup>1</sup>, Miroslav Fikar**

*Institute of Information Engineering, Automation, and Mathematics, Faculty of Chemical and Food Technology, Slovak University of Technology, Radlinského 9, 812 37 Bratislava*

*<sup>1</sup>Department of Chemical Engineering, McMaster University, Canada*

*marian.podmajersky@stuba.sk*

### **Abstract**

The theory of neighbouring-extremal control has been developed over the last 4-5 decades to avoid the costly reoptimisation of dynamic systems, primarily in applications with fast non-linear dynamics. Perhaps the biggest drawback with this approach, when applied to chemical processes, is its poor performance in the presence of large parametric and structural model mismatch. On the other hand, model predictive control (MPC) and run-to-run optimisation are more resistant to model mismatch, but require time-consuming on-line reoptimisation that restricts their applications to slow dynamic systems. This paper proposes to combine both approaches in order to mitigate their deficiencies, thereby leading to an integrated two-time-scale scheme with enhanced performance and tractability for dynamic real-time optimization. This scheme is demonstrated by two batch reactor examples.

**Keywords:** dynamic optimisation, neighbouring extremals, optimal control, two-time-scale scheme

### **Introduction**

In dynamic processes, disturbances and process uncertainties usually give rise to a decrease in production quality along with operational constraint violations. Common sources of uncertainty include measurement noise, inaccurate kinetic rate parameters, feed impurities, and fouling or deposition. It has been known for many years that the application of optimal control can help mitigate the effect of uncertainty on process performance, especially in the presence of constraints (Kadam and Marquardt, 2007).

A number of methodologies for dealing with uncertainty and disturbances in batch process can be found in the literature. Robust optimal control (Terwiesch et al., 1994; Diehl et al., 2008) is a measurement- and reoptimisation-free approach that pre-computes control actions off-line for an expected range of uncertainty and can guarantee on-line feasibility, but is typically (very) conservative. At the other extreme, model predictive control (MPC) (Allgöwer and Zheng, 2000; Garcia et al. 1989) implements a reoptimisation strategy and uses measurements to update the current state of the model. This latter strategy suffers two important deficiencies: i) the presence of constraints may result in an infeasible solution; ii) the reoptimisations may not be tractable in real-time. Clearly, the time needed to reoptimise the system depends on both the problem complexity and the computing performance. Too large a computation time may lead to performance loss, or worse constraint violations, especially for systems with fast dynamics.

In the so-called explicit MPC approach (Bemporad et al., 2002; Kothare et al., 1996), multi-parametric programming is used to precompute off-line all possible control actions for a given range of the state variables. The control inputs are then adjusted by simply selecting the control law that corresponds to the actual state of the process, as given by the latest measurements. Although this method can accommodate fast sampling times, its foremost limitation comes from the off-line computational effort needed to determine the control actions. This currently limits the application of explicit MPC to problems having no more than a few state variables as well as linear dynamics.

Finally, adaptive methods, such as linear-quadratic-Gaussian (LQG) control (Zhou et al., 1995), adaptive control (Astrom and Wittenmark, 1989) and robust loop-shaping (McFarlane and Glover, 1989; Zhou et al., 1995), update the process model by using the most recent measurement data, and then reconfigure the controllers. However, this latter reconfiguration step can be time-consuming and therefore not compatible with applications to chemical processes with fast dynamics.

This paper presents a two-time-scale approach wherein a run-to-run adaptation strategy (Bonvin et al., 2006) is implemented at the slow time scale (outer loop) and is integrated with a (constrained) neighbouring-extremal (NE) controller (Bryson and Ho, 1975) that operates at the fast time scale (inner loop). More specifically, run-to-run adaptation of the terminal constraints (Marchetti et al., 2007) is considered for the outer loop. In its original form, this scheme proceeds by reoptimising the batch operation between each run and

adapting the terminal constraints based on the mismatch between their predicted and measured values; but no adaptation is made within a run. In order to reject disturbances within each run and at the same time promote feasibility and optimality, a NE controller is here considered as the inner loop. The theory of NE control, which has been developed over the last 4-5 decades to avoid the costly reoptimisation of (fast) dynamic systems, is indeed well-suited for batch process control. The integration between the outer- and inner-loops occurs naturally since the NE controllers are recalculated after each run based on the solution to the outer-loop optimisation problem. The resulting integrated two-time-scale optimisation scheme thus provides enhanced performance and tractability.

## Theoretical

### Problem formulation

Throughout the paper, the problem of dynamic optimisation with simple bounds (Problem 1) is considered in the form:

$$\min_{\mathbf{u}} J = \phi(\mathbf{x}(t_f)) + \int_0^{t_f} L(\mathbf{x}(t), \mathbf{u}(t)) dt \quad (1)$$

$$\text{s.t. } \dot{\mathbf{x}} = \mathbf{F}(\mathbf{x}(t), \mathbf{u}(t)), \quad 0 \leq t \leq t_f \quad (2)$$

$$\mathbf{x}(0) = \mathbf{x}_0 \quad (3)$$

$$\mathbf{u}_{\min} \leq \mathbf{u} \leq \mathbf{u}_{\max} \quad (4)$$

and the constrained dynamic optimisation problem (Problem 2) is then given by Problem 1 with additional terminal constraints:

$$\Psi(\mathbf{x}(t_f), t_f) \leq \Psi_{\text{ref}} \quad (5)$$

In (1)–(5),  $t \geq 0$  represents the time variable, with  $t_f$  the final time;  $\mathbf{u} \in \mathbb{R}^{n_u}$  the control vector;  $\mathbf{x} \in \mathbb{R}^{n_x}$  the state vector, with initial value  $\mathbf{x}_0$ ;  $J$ ,  $\phi$  and  $L$  the scalar cost, terminal cost, and integral cost, respectively; and  $\Psi$  the vector of  $n_\Psi$  terminal constraints. All the functions participating in (1)–(5) are assumed to be continuously differentiable with respect to all their arguments.

Let  $\mathbf{u}^*$  denote the optimal solution. It is assumed throughout the paper that  $\mathbf{u}^*$  is unique and that the Hamiltonian function (see below) is regular. These conditions typically lead to a

continuously differentiable extremal solution  $\mathbf{u}^*$ . Note that extremal solution  $\mathbf{u}^*$  is given around the single control arc.

### Necessary conditions of optimality

Following Bryson and Ho (1975), the Hamiltonian function  $H$  is defined as follows:

$$H(\mathbf{x}, \mathbf{u}, \boldsymbol{\lambda}) = L(\mathbf{x}, \mathbf{u}) + \mathbf{F}(\mathbf{x}, \mathbf{u})^T \boldsymbol{\lambda}, \quad (6)$$

where  $\boldsymbol{\lambda} \in \mathbb{R}^{n_x}$  denotes the so-called adjoint (or costate) vector, which satisfy

$$\dot{\boldsymbol{\lambda}} = -\mathbf{H}_{\mathbf{x}} = -\mathbf{F}_{\mathbf{x}}^T \boldsymbol{\lambda} - \mathbf{L}_{\mathbf{x}}, \quad 0 \leq t \leq t_f, \quad (7)$$

where the terminal state for the adjoint vector is given as

$$\boldsymbol{\lambda}(t_f) = [\boldsymbol{\phi}_{\mathbf{x}}]_{t=t_f} \quad (8)$$

for Problem 1 and for Problem 2 is given as

$$\boldsymbol{\lambda}(t_f) = [\boldsymbol{\phi}_{\mathbf{x}} + \mathbf{v}^T \boldsymbol{\psi}_{\mathbf{x}}]_{t=t_f} \quad (9)$$

Lagrange multipliers for the terminal constraints are expressed as  $\mathbf{v} \in \mathbb{R}^{n_{\psi}}$ . (The subscript such as  $_{\psi}$  for a given variable denotes partial derivatives of that variable with respect to  $\psi$ .)

Provided that the optimal control problems: i) Problem 1 expressed by (1)–(4), and ii) Problem 2 expressed by (1)–(5) are not abnormal, the first-order necessary conditions for optimality (NCO) read:

$$\mathbf{H}_{\mathbf{u}} = \mathbf{L}_{\mathbf{u}} + \mathbf{F}_{\mathbf{u}}^T \boldsymbol{\lambda} = \mathbf{0}, \quad \mathbf{H}_{\mathbf{uu}} > 0 \quad (10)$$

along with the additional conditions for Problem 2:

$$\mathbf{0} = \nu_k \boldsymbol{\psi}_k, \quad \nu_k \geq 0, \quad \text{for each } k = 1, \dots, n_{\psi}. \quad (11)$$

This latter determines the set of active constraints at the optimum, which is denoted by the vector  $\bar{\boldsymbol{\psi}}$  of dimension  $n_{\bar{\psi}}$ .

### Neighbouring-extremal control

Consider a small variation  $\delta \mathbf{x}_0$  in the initial states and a small variation  $\delta \bar{\boldsymbol{\psi}}$  (for Problem 2 only) in active terminal constraints,

$$\mathbf{x}(0) = \mathbf{x}_0 + \delta \mathbf{x}_0, \quad (12)$$

$$\bar{\boldsymbol{\psi}}(\mathbf{x}(t_f), t_f) = \delta \bar{\boldsymbol{\psi}}. \quad (\text{Problem 2 only}) \quad (13)$$

The corresponding variations in optimal control vector  $\delta\mathbf{u}(t)$ , state vector  $\delta\mathbf{x}(t)$ , adjoint vector  $\delta\boldsymbol{\lambda}(t)$  and Lagrange multiplier vector  $\delta\bar{\mathbf{v}}$  (for the active terminal constraints  $\bar{\boldsymbol{\psi}}$ ) can be calculated from the linearisation of the first-order NCO represented by (10)–(11) around the extremal path (Bryson and Ho, 1975):

$$\delta\dot{\mathbf{x}} = \mathbf{F}_x^* \delta\mathbf{x} + \mathbf{F}_u^* \delta\mathbf{u} \quad (14)$$

$$\delta\dot{\boldsymbol{\lambda}} = -\mathbf{H}_{xx}^* \delta\mathbf{x} - \mathbf{F}_x^{*T} \delta\boldsymbol{\lambda} - \mathbf{H}_{xu}^* \delta\mathbf{u} \quad (15)$$

$$\mathbf{0} = \mathbf{H}_{ux}^* \delta\mathbf{x} + \mathbf{F}_u^{*T} \delta\boldsymbol{\lambda} + \mathbf{H}_{uu}^* \delta\mathbf{u} \quad (16)$$

$$\delta\mathbf{x}(0) = \delta\mathbf{x}_0 \quad (17)$$

for Problem 1 and Problem 2. An additional condition for Problem 1 is:

$$\delta\boldsymbol{\lambda}(t_f) = \left[ \boldsymbol{\phi}_{xx}^* \delta\mathbf{x} \right]_{t=t_f} \quad (18)$$

and additional conditions for Problem 2 are then denoted as:

$$\delta\boldsymbol{\lambda}(t_f) = \left[ \left( \boldsymbol{\phi}_{xx}^* + \bar{\mathbf{v}}^{*T} \bar{\boldsymbol{\psi}}_{xx}^* \right) \delta\mathbf{x} + \bar{\boldsymbol{\psi}}_x^{*T} \delta\bar{\mathbf{v}} \right]_{t=t_f} \quad (19)$$

$$\delta\bar{\boldsymbol{\psi}} = \left[ \bar{\boldsymbol{\psi}}_x^* \delta\mathbf{x} \right]_{t=t_f}, \quad (20)$$

A superscript  $*$  indicates that the corresponding quantity is evaluated along the extremal path  $\mathbf{u}^*(t)$ ,  $0 \leq t \leq t_f$ , and corresponding states, adjoints and Lagrange multipliers.

It can be shown that the optimal control variation  $\delta\mathbf{u}(t)$  is also obtained as the solution to the so-called accessory minimum problem (Pesh, 1990):

- for Problem 1 is denoted as

$$\min_{\delta\mathbf{u}} \delta^2 J = \frac{1}{2} \left[ \delta\mathbf{x}(t_f)^T \boldsymbol{\phi}_{xx}^* \delta\mathbf{x}(t_f) \right] + \frac{1}{2} \int_0^{t_f} \begin{pmatrix} \delta\mathbf{x} \\ \delta\mathbf{u} \end{pmatrix}^T \begin{pmatrix} \mathbf{H}_{xx}^* & \mathbf{H}_{xu}^* \\ \mathbf{H}_{ux}^* & \mathbf{H}_{uu}^* \end{pmatrix} \begin{pmatrix} \delta\mathbf{x} \\ \delta\mathbf{u} \end{pmatrix} dt \quad (21)$$

$$\text{s.t.} \quad \delta\dot{\mathbf{x}} = \mathbf{F}_x^* \delta\mathbf{x} + \mathbf{F}_u^* \delta\mathbf{u} \quad (22)$$

$$\delta\mathbf{x}(0) = \delta\mathbf{x}_0 \quad (23)$$

- and for Problem 2 is given as

$$\min_{\delta\mathbf{u}} \delta^2 J = \frac{1}{2} \left[ \delta\mathbf{x}(t_f)^T \left( \boldsymbol{\phi}_{xx}^* + \bar{\mathbf{v}}^{*T} \bar{\boldsymbol{\psi}}_{xx}^* \right) \delta\mathbf{x}(t_f) \right] + \frac{1}{2} \int_0^{t_f} \begin{pmatrix} \delta\mathbf{x} \\ \delta\mathbf{u} \end{pmatrix}^T \begin{pmatrix} \mathbf{H}_{xx}^* & \mathbf{H}_{xu}^* \\ \mathbf{H}_{ux}^* & \mathbf{H}_{uu}^* \end{pmatrix} \begin{pmatrix} \delta\mathbf{x} \\ \delta\mathbf{u} \end{pmatrix} dt \quad (24)$$

$$\text{s.t.} \quad \delta\dot{\mathbf{x}} = \mathbf{F}_x^* \delta\mathbf{x} + \mathbf{F}_u^* \delta\mathbf{u} \quad (25)$$

$$\delta\mathbf{x}(0) = \delta\mathbf{x}_0 \quad (26)$$

$$\delta\bar{\Psi} = \left[ \bar{\Psi}_x^* \delta\mathbf{x} \right]_{t=t_f}, \quad (27)$$

Note that the adjoint vector function  $\delta\lambda$  and Lagrange multiplier vector  $\delta\bar{\Psi}$  associated to this linear-quadratic (LQ) problem are identical to those obtained from the solution of the linear two-point boundary-value problem (TPBVP), note (14)–(17) with (18) for Problem 1 and (14)–(17) with (19)–(20) for Problem 2.

From the assumption that the Hamiltonian function is regular,  $\mathbf{H}_{uu}^*$  is non-singular along  $0 \leq t \leq t_f$ , and (16) can be solved for  $\delta\mathbf{u}(t)$  in terms of  $\delta\mathbf{x}(t)$  and  $\delta\lambda(t)$ :

$$\delta\mathbf{u}(t) = -(\mathbf{H}_{uu}^*)^{-1} \left[ \mathbf{F}_u^{*T} \delta\lambda(t) + \mathbf{H}_{ux}^* \delta\mathbf{x}(t) \right]. \quad (28)$$

Substituting (28) into (14)–(15) gives:

$$\delta\dot{\mathbf{x}}(t) = \mathbf{a}(t)\delta\mathbf{x}(t) - \mathbf{\beta}(t)\delta\lambda(t) \quad (29)$$

$$\delta\dot{\lambda}(t) = -\gamma(t)\delta\mathbf{x}(t) - \mathbf{a}(t)^T \delta\lambda(t), \quad (30)$$

where

$$\mathbf{a}(t) := \mathbf{F}_x^* - \mathbf{F}_u^* (\mathbf{H}_{uu}^*)^{-1} \mathbf{H}_{ux}^* \quad (31)$$

$$\mathbf{\beta}(t) := \mathbf{F}_u^* (\mathbf{H}_{uu}^*)^{-1} \mathbf{F}_u^{*T} \quad (32)$$

$$\gamma(t) := \mathbf{H}_{xx}^* - \mathbf{H}_{xu}^* (\mathbf{H}_{uu}^*)^{-1} \mathbf{H}_{ux}^*. \quad (33)$$

with the boundary conditions given by (17) and (18) for Problem 1, and by (17), (19), and (20) for Problem 2.

### Numerical Computation of Neighbouring Feedback Control

The linear TPBVP given by (29)–(30) can be used to calculate the neighbouring-extremal control correction  $\delta\mathbf{u}(t)$ ,  $0 \leq t \leq t_f$ , in either one of two situations:

- I. The variations  $\delta\mathbf{x}_0$  or possibly  $\delta\bar{\Psi}$  are available continuously in time, in which case the backward sweep method (Bryson and Ho, 1975) can be used to derive an explicit feedback control law.
- II. The initial state  $\delta\mathbf{x}_0$  or in case of Problem 2 also (active) terminal constraint variations  $\delta\bar{\Psi}$  are available at discrete time instants, in which case the discrete feedback control can be obtained by directly re-solving the TPBVP. This can be done via a shooting method as described in Shooting Method section, below.

## Shooting Method

According to Pesh (1989) and Pesh (1990), the linear TPBVP given by (29)–(30) can be rewritten in the form:

$$\begin{pmatrix} \delta\dot{\mathbf{x}}(t) \\ \delta\dot{\boldsymbol{\lambda}}(t) \end{pmatrix} = \underbrace{\begin{pmatrix} \boldsymbol{\alpha}(t) & -\boldsymbol{\beta}(t) \\ -\boldsymbol{\gamma}(t) & -\boldsymbol{\alpha}^T(t) \end{pmatrix}}_{=:\boldsymbol{\Lambda}(t)} \begin{pmatrix} \delta\mathbf{x}(t) \\ \delta\boldsymbol{\lambda}(t) \end{pmatrix}, \quad (34)$$

with the boundary conditions for Problem 1 given as

$$\begin{pmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \delta\mathbf{x}(0) \\ \delta\boldsymbol{\lambda}(0) \end{pmatrix} + \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ -[\boldsymbol{\phi}_{\mathbf{x}\mathbf{x}}^*]_{t_f} & \mathbf{I} \end{pmatrix} \begin{pmatrix} \delta\mathbf{x}(t_f) \\ \delta\boldsymbol{\lambda}(t_f) \end{pmatrix} = \begin{pmatrix} \delta\mathbf{x}_0 \\ \mathbf{0} \end{pmatrix} \quad (35)$$

or alternatively, the boundary conditions for Problem 2 read

$$\begin{pmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \delta\mathbf{x}(0) \\ \delta\boldsymbol{\lambda}(0) \end{pmatrix} + \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ -[\boldsymbol{\phi}_{\mathbf{x}\mathbf{x}}^* + \bar{\mathbf{v}}^{*T} \bar{\boldsymbol{\Psi}}_{\mathbf{x}\mathbf{x}}^*]_{t_f} & \mathbf{I} \end{pmatrix} \begin{pmatrix} \delta\mathbf{x}(t_f) \\ \delta\boldsymbol{\lambda}(t_f) \end{pmatrix} = \begin{pmatrix} \delta\mathbf{x}_0 \\ [\bar{\boldsymbol{\Psi}}_{\mathbf{x}}^{*T}]_{t_f} \delta\bar{\mathbf{v}} \end{pmatrix} \quad (36)$$

Substantially, the Euler-Lagrange equations (34) can be solved, in principle, by forward or backward finding of transition matrix. Then, the approach proceeds by guessing the missing initial (or terminal) conditions in (35), or alternatively in (36), and consequently by adjusting them in such a way that the corresponding conditions are satisfied. These can be determined by inverting partitions of the transition matrix (Pesh, 1989) at the initial (or terminal) time. The solution is provided directly by integration of the coupled equations, iteration is not required. Having solved the missing boundary conditions, an open-loop optimal solution with the demanded control histories is clearly a solution of initial-value problem described by Euler-Lagrange equations (29)–(30) with these estimated boundary conditions.

Given the guess  $\delta\boldsymbol{\lambda}(0) = \delta\boldsymbol{\lambda}_0$  for the adjoint variations at initial time  $t=0$ , the (unique) solution to the linear Euler-Lagrange equations (34) is of the form:

$$\begin{pmatrix} \delta\mathbf{x}(t; \delta\boldsymbol{\lambda}_0) \\ \delta\boldsymbol{\lambda}(t; \delta\boldsymbol{\lambda}_0) \end{pmatrix} = \underbrace{\begin{pmatrix} \boldsymbol{\Upsilon}_1(t; 0) & \boldsymbol{\Upsilon}_2(t; 0) \\ \boldsymbol{\Upsilon}_3(t; 0) & \boldsymbol{\Upsilon}_4(t; 0) \end{pmatrix}}_{=:\boldsymbol{\Upsilon}(t; 0)} \begin{pmatrix} \delta\mathbf{x}_0 \\ \delta\boldsymbol{\lambda}_0 \end{pmatrix}, \quad (37)$$

where the transition matrix  $\boldsymbol{\Upsilon}(t; 0)$  is obtained as the solution to the initial value problem

$$\frac{\partial}{\partial t} \boldsymbol{\Upsilon}(t; 0) = \boldsymbol{\Lambda}(t) \boldsymbol{\Upsilon}(t; 0), \quad 0 \leq t \leq t_f; \quad \boldsymbol{\Upsilon}(0; 0) = \mathbf{I}. \quad (38)$$

Substituting (37) into (35) leads to the following linear system in the variables  $\delta\boldsymbol{\lambda}(0)$  (Problem 1):

$$\left( \begin{bmatrix} \Phi_{xx}^* \end{bmatrix}_{t_f} \Upsilon_2(t_f;0) - \Upsilon_4(t_f;0) \right) \delta\lambda(0) = - \left( \begin{bmatrix} \Phi_{xx}^* + \bar{\mathbf{v}}^{*T} \bar{\Psi}_{xx}^* \end{bmatrix}_{t_f} \Upsilon_1(t_f;0) - \Upsilon_3(t_f;0) \right) \delta\mathbf{x}_0. \quad (39)$$

Substituting (37) into (36) and (27) leads to the following linear system in the variables  $\delta\lambda(0), \delta\bar{\mathbf{v}}$  (Problem 2):

$$\begin{aligned} & \left( \begin{array}{cc} \begin{bmatrix} \Phi_{xx}^* + \bar{\mathbf{v}}^{*T} \bar{\Psi}_{xx}^* \end{bmatrix}_{t_f} \Upsilon_2(t_f;0) - \Upsilon_4(t_f;0) & \begin{bmatrix} \bar{\Psi}_x^{*T} \end{bmatrix}_{t_f} \\ \begin{bmatrix} \bar{\Psi}_x^* \end{bmatrix}_{t_f} \Upsilon_2(t_f;0) & \mathbf{0} \end{array} \right) \begin{pmatrix} \delta\lambda(0) \\ \delta\bar{\mathbf{v}} \end{pmatrix} \\ & = \begin{pmatrix} \mathbf{0} \\ \mathbf{I} \end{pmatrix} \delta\bar{\Psi} - \left( \begin{array}{c} \begin{bmatrix} \Phi_{xx}^* + \bar{\mathbf{v}}^{*T} \bar{\Psi}_{xx}^* \end{bmatrix}_{t_f} \Upsilon_1(t_f;0) - \Upsilon_3(t_f;0) \\ \begin{bmatrix} \bar{\Psi}_x^* \end{bmatrix}_{t_f} \Upsilon_1(t_f;0) \end{array} \right) \delta\mathbf{x}_0. \end{aligned} \quad (40)$$

In case of Problem 1, for given initial state variations  $\delta\mathbf{x}_0$ , the solution to the linear system (39) provides the corresponding initial adjoint variations  $\delta\lambda(0)$ . In case of Problem 2, the wanted initial adjoint and Lagrange multiplier variations  $\delta\lambda(0)$  and  $\delta\bar{\mathbf{v}}$  are the solution of linear system (40), for given initial state and active terminal constraint variations  $\delta\mathbf{x}_0$  and  $\delta\bar{\Psi}$ . Finally, the NE control variation can be calculated from (28) as:

$$\delta\mathbf{u}(t) = -(\mathbf{H}_{uu}^*)^{-1} \begin{pmatrix} \mathbf{H}_{ux}^* & \mathbf{F}_u^{*T} \end{pmatrix} \Upsilon(t;0) \begin{pmatrix} \delta\mathbf{x}_0 \\ \delta\lambda(0) \end{pmatrix}. \quad (41)$$

### Run-to-run Constraint Adaptation

The principle behind run-to-run optimization is similar to MPC. But instead of adapting the initial conditions and moving the control horizon as is done in MPC, the adaptation is performed on the optimization model (e.g., model parameters or constraint biases) before re-running the optimizer. More specifically, run-to-run constraint adaptation (Marchetti et al., 2007) more specifically, adapts terminal constraints (5) in the optimisation model after each run as

$$\delta\boldsymbol{\psi}(\mathbf{x}(t_f), t_f) \leq \delta\boldsymbol{\Psi} \quad (42)$$

where  $\delta\boldsymbol{\psi}$  stands for the terminal constraint bias. Such a bias can be directly updated as the difference between the available terminal constraint measurements,  $\boldsymbol{\psi}^{\text{meas}}$ , at the end of each run and the predicted constraint values. This simple strategy may however lead to excessive correction when operating far away from the optimum, and it may also render the adaptation

scheme very sensitive to measurement noise. A better strategy consists of filtering the bias, e.g., with a first-order exponential filter:

$$\delta\psi_{k+1} = [\mathbf{I} - \mathbf{W}] \delta\psi_k + \mathbf{W} [\psi_k^{\text{meas}} - \psi(\mathbf{x}_k^*(t_f), t_f)], \quad (43)$$

with  $k$  the run index, and  $\mathbf{W}$  a gain matrix—typically, a diagonal matrix.

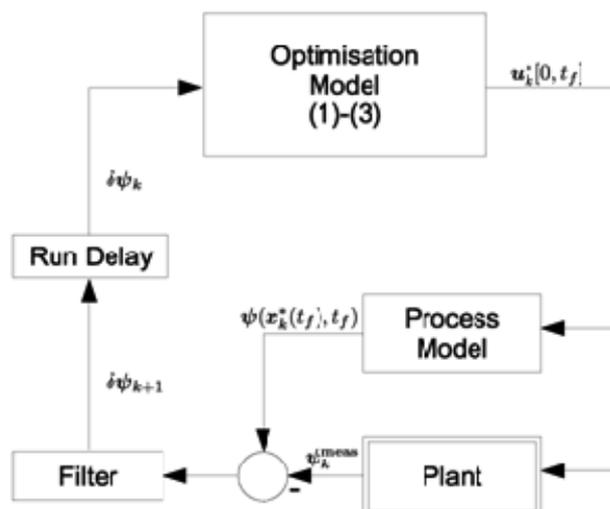


Fig. 1. Run-to-run constraint adaptation scheme.

The run-to-run constraint-adaptation scheme is shown in Fig. 1. The constrained dynamic optimisation problem uses the available process model. It is solved between each run, using any numerical procedure, such as the sequential (Edgar et al., 1988; Guntern et al., 1998; Ray, 1981) or the simultaneous (Hertzberg, 1997; Biegler, 1984) approach of dynamic optimisation. The optimal control trajectory  $\mathbf{u}_k^*(t), 0 \leq t \leq t_f$ , is computed and applied to the plant during the  $k$ th run. The predicted optimal response is denoted by  $\mathbf{x}_k^*(t)$ . The discrepancy between the measured terminal constraint values  $\psi_k^{\text{meas}}$  and the optimizer predictions  $\psi(\mathbf{x}_k^*(t_f), t_f)$  is then used to adjust the constraint bias as described earlier, before re-running the optimizer for the next run.

### Two-times-scale optimization scheme

The Run-to-run constraint adaptation was shown to be a promising technology in (Marchetti et al., 2007). This approach provides a natural framework for handling changes in active constraints in dynamic process systems and it is quite robust towards model mismatch and process disturbances. Moreover, its implementation is simple. Inherent limitation of this

scheme, however, are that (i) it does not perform any control corrections during the runs, and (ii) it typically leads to suboptimal performance. On the other hand, neighbouring-extremal control as described above is able to correct small deviations around the nominal extremal path in order to deliver similar performance as with re-optimisation. Since no costly on-line re-optimisation is performed, this approach is especially suited for processes with fast dynamics. However, the performance of NE control typically decreases dramatically in the presence of large model mismatch and process disturbances, and it requires a full-state measurement. This leads to sub-optimality and, worse, infeasibility when constraints are present or limited measurements are available. Our proposal is to combine the advantages of these two approaches: Run-to-run constraint adaptation is applied at a slow time scale (outer loop) to handle large model mismatch and changes in active constraints, based on run-end measurements only; and NE control is applied at a fast time scale (inner loop), and uses measurement information available within each run, in order to enhance convergence speed and mitigate sub-optimality. The resulting integrated two-time-scale optimization scheme is depicted in Fig. 2.

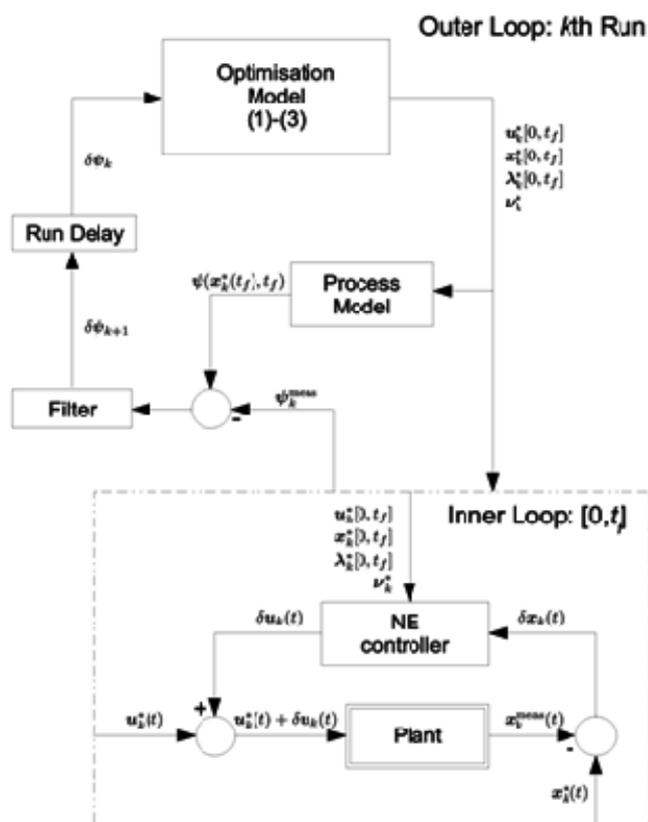


Fig. 2. Two-times-scale optimisation scheme employing neighbouring controller in the inner loop and run-to-run constraint adaptation in the outer loop.

The implementation procedure is as follows:

**Initialisation:**

0. Initialise the constraint bias  $\delta\psi = \mathbf{0}$ , select a gain matrix  $\mathbf{W}$  and set the run index to  $k = 1$

**Outer Loop:**

1. Determine  $\mathbf{u}_k^*$  by solving the dynamic optimisation problem (1)–(5), then obtain the corresponding states  $\mathbf{x}_k^*$  and adjoints  $\lambda_k^*$ , together with the Lagrange multipliers  $\bar{\mathbf{v}}_k^*$  that satisfy the NCO. If  $\bar{\mathbf{v}}_k^* \neq 0$ , the terminal constraints are active and they satisfy the NCO expressed by (6), (7), (9), (10), and (11). If  $\bar{\mathbf{v}}_k^* = 0$ , the terminal constraints are inactive and they satisfy the NCO expressed by (6), (7), and (8).
2. Design a NE controller around the extremal path  $\mathbf{u}_k^*$ , either by using the backward sweep approach (continuous measurements, or by applying the transition matrix method (discrete measurements). Note that, if  $\bar{\mathbf{v}}_k^* = 0$ , the NE controller design is given by Problem 1, if  $\bar{\mathbf{v}}_k^* \neq 0$  the NE controller design denotes Problem 2.

**Inner Loop:**

3. Implement the NE controller during the  $k$ th run in order to calculate the corrections  $\delta\mathbf{u}_k(t)$  to  $\mathbf{u}_k^*(t)$  based on the available (continuous or discrete) process measurements.
4. Update the constraint bias  $\delta\psi_{k+1}$  as the filtered difference between the measured values of the terminal constraints and their predicted counterparts.
5. Increment the run index  $k \leftarrow k + 1$ , and return to Step 1.

The performance of this integrated scheme are illustrated with a case study in the subsequent section.

## Experimental

A batch reactor example taken from Crescitelli and Nicoletti, 1973 is considered to illustrate the proposed integrated two-times-scale approach. The following series reaction takes place in the reactor:



that is initially loaded by reactant  $\mathbf{R}$  and by a small amount of product  $\mathbf{Q}$ . The goal is to maximise the production of  $\mathbf{P}$ , whereas  $\mathbf{Q}$  is an undesired by-product. The optimisation objective is then given as a maximisation of product  $\mathbf{P}$  concentration at the final time,  $c_p(t_f)$ , while keeping the residual concentration of reactant  $\mathbf{P}$  at the final time,  $c_p(t_f)$ , below the maximum threshold  $c_R^{\min}$ . The manipulated variable is the reactor temperature  $T_R(t)$ . No bound constraints are imposed on the reactor temperature. Kinetic rate constants for the reactions are given by Arrhenius expressions:

$$k_i(t) = k_{i,0} \exp\left(-\frac{\frac{E_i}{4.2}}{\frac{R}{4.2}T_R(t)}\right), \quad i = 1, 2. \quad (45)$$

Overall, the optimisation problem reads:

$$\max_{\mathbf{T}} J = c_p(t_f) \quad (46)$$

$$\text{s.t.} \quad \dot{c}_R = -k_1(t)c_R(t); \quad c_R(0) = \beta_R \quad (47)$$

$$\dot{c}_P = k_1(t)c_R(t) - k_2(t)c_P(t); \quad c_P(0) = \beta_P \quad (48)$$

$$c_R(t_f) \leq c_R^{\min}, \quad (49)$$

where  $c_R$ ,  $c_Q$  and  $c_P$  are the concentrations [mol/L] of the species  $\mathbf{R}$ ,  $\mathbf{Q}$ , and  $\mathbf{P}$ , respectively,  $k_1$  and  $k_2$  the kinetic coefficients [1/min],  $T_R$  the reactor temperature [K], and  $\beta_R$ ,  $\beta_P$  the initial states [mol/L]. The model parameters and initial conditions are given in Table 1, below.

Table 1. Model parameters, and initial conditions

$k_{1,0} = 0.535e11$	[1/min]
$k_{2,0} = 0.461e18$	[1/min]
$E_1 = 75.4$	[kJ/mol]
$E_2 = 125.6$	[kJ/mol]
$R = 8.4$	[J/mol/K]
$t_f = 8$	[min]
$\beta_R = 0.53$	[mol/L]
$\beta_P = 0.43$	[mol/L]
$c_R^{\min} = 0.1$	[mol/L]

## Design of control methods

We have implemented the controller with constraint adaptation described in the theoretical section. As mentioned before, the neighbouring-extremal controller can be implemented in two ways: continuously or in discrete time intervals. In this paper, NE controller is always implemented in discrete time intervals. It is assumed that this implementation, can deliver more accurate and more stable performance.

## Results and Discussion

### Open-loop optimal control

Solving the optimisation problem introduced by (46)–(49), with well-known sequential or simultaneous method, the piece-wise control profile (Fig. 3) shows the presence of one interior arc. Also, note that the inequality constraint is active at the end-point. Along this interior arc, at the start-points and at the end-points, the necessary conditions given by (6)–(11) must hold.

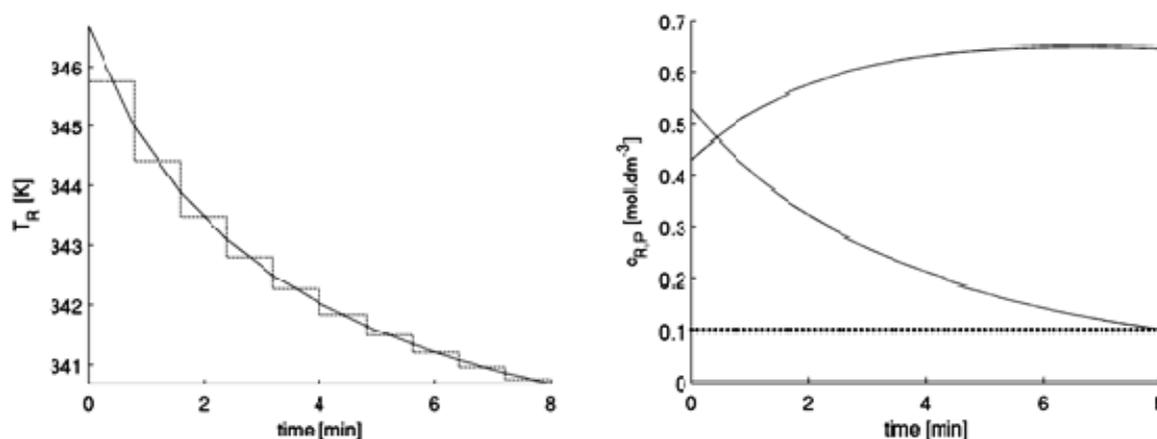


Fig. 3. Left: Optimal control solution – piece-wise control vs. smooth control; Right: Optimal state trajectories - piece-wise control vs. smooth control.

Hence,  $H_u$  depends on control variable, the reactor temperature  $T_R$  which is considered as control variable, the problem is non-singular. Thus, the control action is explicitly determined by necessary condition (10). The smooth optimal control trajectory for optimization problem (46)–(49) is then provided as a solution of the differential-algebraic problem:

$$X_i(t) = \exp\left(\frac{\frac{E_i}{4.2}}{\frac{R}{4.2}T_R(t)}\right) \quad (50)$$

$$\dot{c}_R = -\frac{k_{1,0}c_R(t)}{X_1(t)} \quad (51)$$

$$\dot{c}_P = \frac{k_{1,0}c_R(t)}{X_1(t)} - \frac{k_{2,0}c_P(t)}{X_2(t)} \quad (52)$$

$$\frac{1}{\frac{R}{4.2}(T_R(t))^2} \left( \frac{\lambda_P \frac{E_1}{4.2} k_{1,0} c_R(t)}{X_1(t)} - \frac{\lambda_P \frac{E_2}{4.2} k_{2,0} c_P(t)}{X_2(t)} - \frac{\lambda_R \frac{E_1}{4.2} k_{1,0} c_R(t)}{X_1(t)} \right) = 0 \quad (53)$$

with boundary conditions:

$$\begin{pmatrix} c_R \\ c_P \\ \lambda_R \\ \lambda_P \end{pmatrix} (t_0) = \begin{pmatrix} \beta_R \\ \beta_P \\ \lambda_{R,0} \\ \lambda_{P,0} \end{pmatrix}; \quad \begin{pmatrix} c_R \\ c_P \\ \lambda_R \\ \lambda_P \end{pmatrix} (t_f) = \begin{pmatrix} c_{R,f} \\ c_{P,f} \\ \nu \\ -1 \end{pmatrix} \quad (54)$$

The optimal control profile is estimated by guessing unknown vector  $z = [\nu \quad \lambda_{R,0} \quad \lambda_{P,0}]$ .

The vector  $z$  is iteratively updated until boundary conditions (54) are satisfied. Subsequently, the performance index evaluated at  $z^*$ ,  $J = 0.6477$  [mol/L], must match the objective value obtained by sequential or simultaneous method,  $J = 0.6475$  [mol/L].

### Closed-loop optimal control

The integrated two-time-scale scheme is benchmarked on simulated reality in presence of uncertainty. In order to simulate the reality, the nominal model is perturbed by variations in the initial values, in the reaction constants, or/and by added measurement noise. While the controller designs are calculated using nominal mathematical model, the simulations are performed for measured outputs of simulated reality, hence the discrepancy between given states and measured outputs is considered as another form of uncertainty.

Two cases are considered with following variations of reaction constants:  $\delta k_{1,0} = -5\%$ ,  $\delta k_{2,0} = +10\%$ ,  $\delta E_1 = +1\%$ , and  $\delta E_2 = -0.5\%$ . In each case, performance of proposed integrated scheme is compared to the results obtained by pure neighbouring-extremal control and by constraint adaptation approach. A benchmark performed with negatively perturbed

initial conditions by 40% and without measurement noise is considered as Case 1. Same conditions are considered for Case 2 with measurement noise, in addition. There are no discontinuous perturbations considered during the batch time, in both case studies. These results are reported in Fig. 4, Case 1, and in Fig. 5, for Case 2.

### Performance in the Presence of Uncertainty

In order to illustrate the benefits of the proposed integrated scheme, two case studies are investigated. The performance of the control methods is tested on system with modelled uncertainty. Within each case study, two adaptation strategies are compared along the proposed integrated two-time-scale control scheme in order to reject these perturbations (represented by uncertainty). Note that for the sake of comparison, the run-to-run adaptation is initialized with a constraint bias of  $\delta\psi = 0$  and considers a filter gain of  $\mathbf{W} = 1$  for Case 1 and  $\mathbf{W} = 0.2$  for Case 2. This filter parameters were chosen so as to achieve the reference as fast as possible while the oscillations during the adaptation process should be regarded. The exact values were found from set of simulations performed with varying gains.

For the Case 1, Fig. 4 displays converged response and control solutions and the evolution of terminal constraints for benchmarked control schemes, after 10 runs. Observe that in Case 1, the performance of the pure neighbouring controller is not sufficient, the single terminal constraint is not met and the performance index is the lowest during all batches. In contrast, pure constraint adaptation is able to fully recover optimality loss. This control and response solutions match the optimal solution for perturbed system. The same can be observed for proposed integrated two-times-scale scheme. Terminal constraint meets the reference value. The value of performance index slightly varies from reoptimised solution, but on the other hand, the optimality loss is almost recovered compared to the pure NE control scheme. Similar results are depicted for control profile that are closer to those from reoptimised solution in comparison to the pure NE control scheme. Hence, the NE controller is approximated and operates around the nominal solution, the optimality loss is expected and it highly depends on size of the variations from the origin. It is shown that the performance of NE controller is successfully improved by its integration into constraint adaptation scheme. Especially regard that the proposed solution almost satisfies the terminal constraint from the first to the fourth batch and it meets constraint, in proceeding batches. In opposite, the constraint adaptation approach needs at least 6 runs to meet the terminal constraint, in Case 1.

On the one side, the performance of converged solution of pure constraint adaptation seems better than the performance of the proposed solution, but on the other side, the first-mentioned method converges slower while during these batches, constraints are not met. In this manner, the proposed integrated two-time-scale scheme is more robust than the pure constraint adaptation. This observation is also illustrated by the next case study.

Fig. 5 depicts the results for similar benchmark as previous, except the presence of output measurements. See, in Case 2, a weaker performance of pure NE controller is presented. The controller is unable to fully recover the optimality loss as well as the terminal constraints are not met during 20 batches. As is obvious from Fig. 5, the pure constraint adaptation approach is able to recover optimality loss while operating around the terminal constraint reference.

Also, in Case 2, this approach needs several batches to intercept the neighbourhood of the reference. The integrated scheme shows similar results. In addition, this approach is able to recover most of the optimality loss during the first batches. Note that the terminal constraints in presence of noisy measurements are not exactly satisfied. They are held in close proximity from the desired reference. The filter parameter  $\mathbf{W}$  is chosen such as to eliminate oscillations but they can not be completely eliminated in the presence of measurement noise. Still, it can be observed that the integrated scheme oscillates a little bit less than pure constraint adaptation.

## Conclusion

In this paper, an integrated two-times-scale scheme has been proposed and investigated in order to improve the performance and tractability of dynamic real-time optimisation, with application to batch processes. The combination of two approaches, namely run-to-run constraint adaptation and neighbouring-extremal control, allows to complement the benefits of each other, while mitigating some of their deficiencies. Because of fast dynamics in chemical applications, the NE controller is able to adapt the control profile at a high frequency. On the other hand, run-to-run adaptation allows to deal with large model mismatch and handles changes in the set of active constraints after each run. This integrated scheme has been demonstrated through the case study of a batch reactor under ideal conditions with no measurement noise and under simulated reality with the presence of measurement noise. As

part of the future work, an extension of the current scheme to singular control problems is currently under investigation, as well as the ability to handle problems with path constraints.

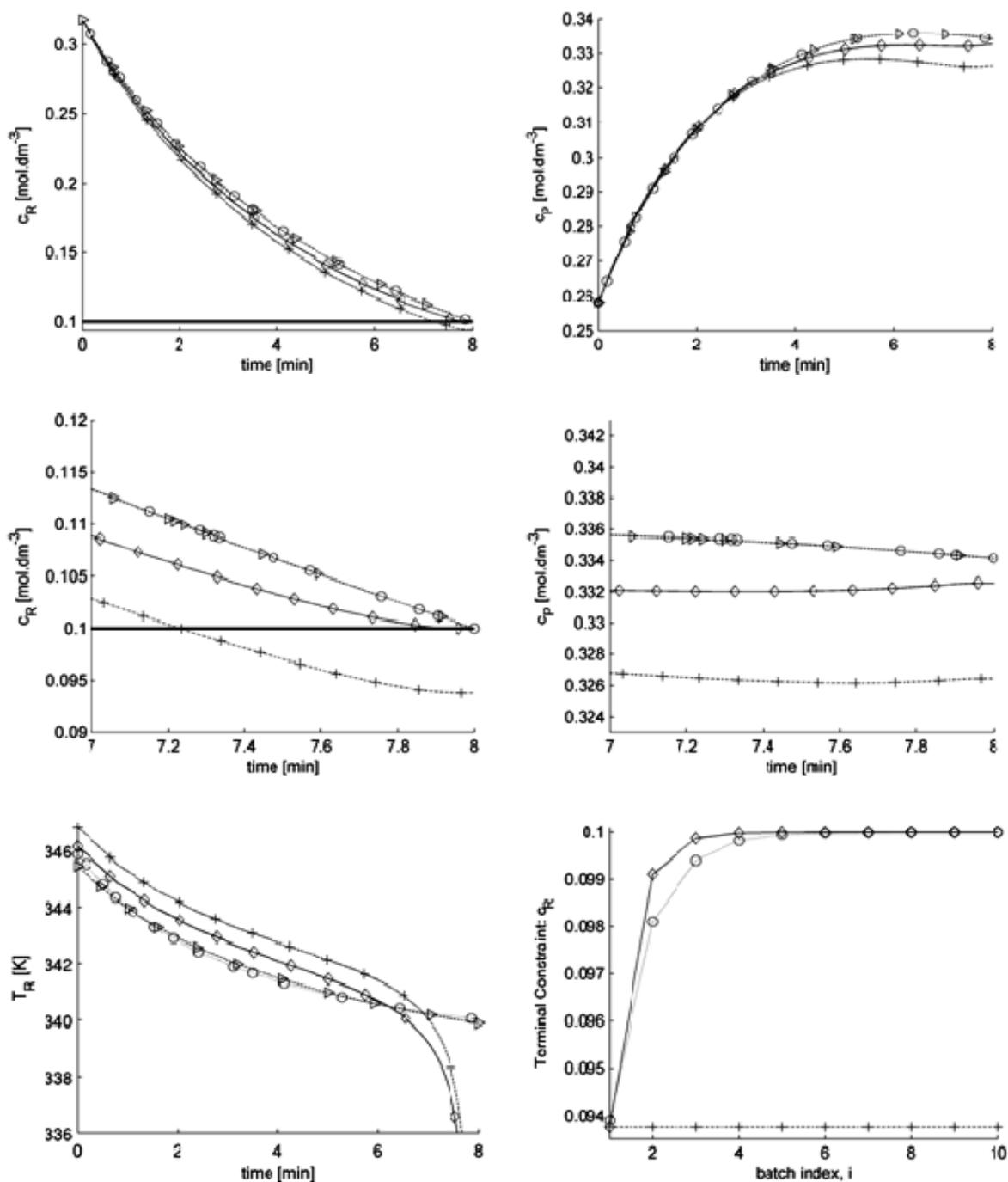


Fig. 4. Performance of Case 1, after 10 runs. Dotted lines with circles: optimal solution for perturbed system; dashed lines with crosses: neighbouring-extremal control; dash-dotted lines with triangles: constraint adaptation control; solid lines with diamonds: integrated two-time-scale scheme control. Top left: Converged solutions for  $c_R$ ; top right: Converged solutions for  $c_P$ ; middle left: converged terminal constraints (zoomed); middle right: converged performance indices (zoomed); bottom left: converged control profiles; bottom right: evolution of terminal constraints.

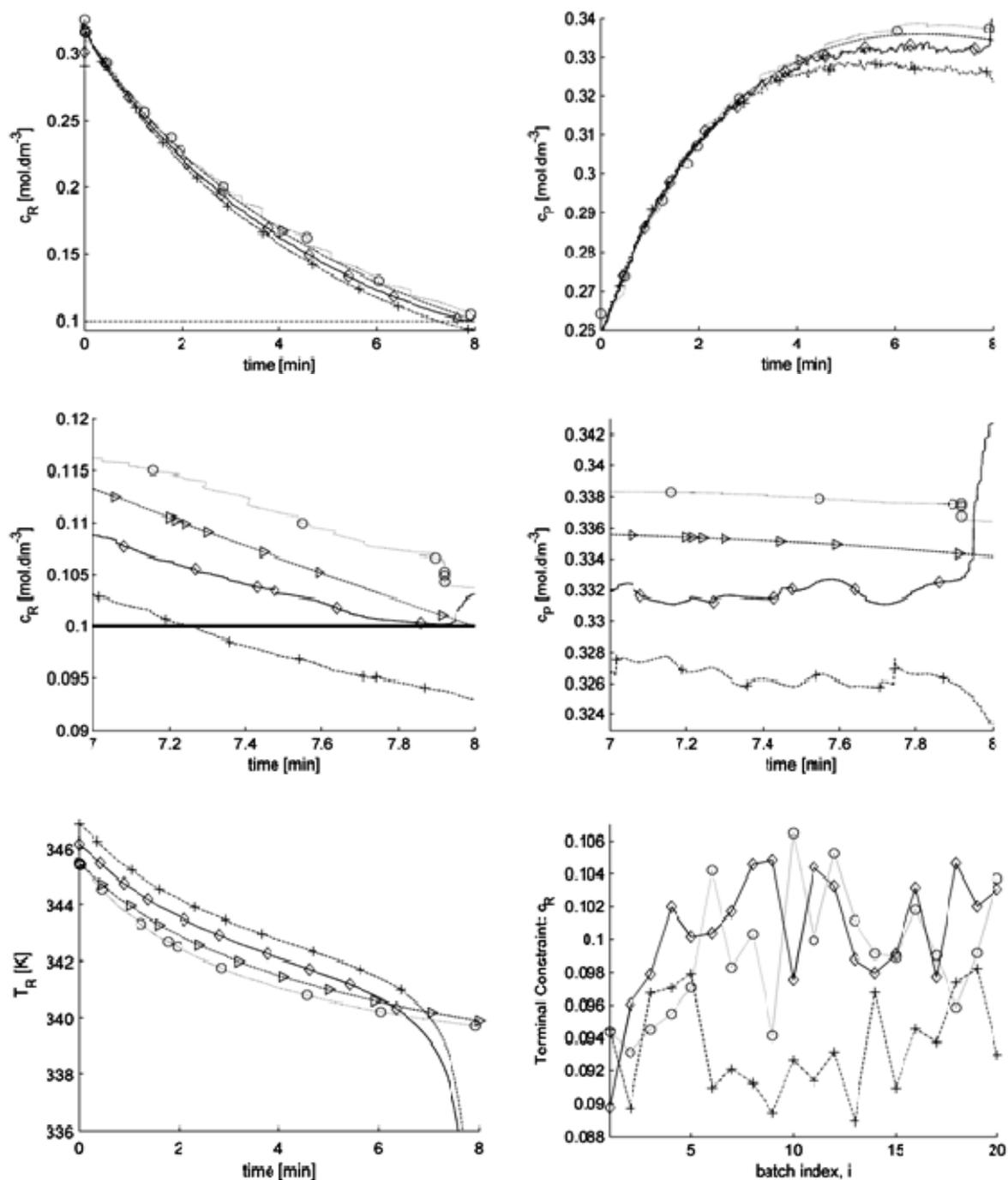


Fig. 5. Performance of Case 2, after 20 runs. Dotted lines with circles: optimal solution for perturbed system; dashed lines with crosses: neighbouring-extremal control; dash-dotted lines with triangles: constraint adaptation control; solid lines with diamonds: integrated two-time-scale scheme control. Top left: Converged solutions for  $c_R$ ; top right: Converged solutions for  $c_P$ ; middle left: converged terminal constraints (zoomed); middle right: converged performance indices (zoomed); bottom left: converged control profiles; bottom right: evolution of terminal constraints.

### *Acknowledgement*

*The authors gratefully acknowledge the contribution of the Scientific Grant Agency of the Slovak Republic under the grants 1/0071/09, 1/0537/10 and the Slovak Research and Development Agency under the project APVV-0029-07.*

### **References**

- Allgöwer, F., Zheng, A. (2000) *Nonlinear Model Predictive Control*, Birkhäuser Verlag
- Äström, K. Wittenmark, B. (1989) *Adaptive Control*, Addison-Wesley, Massachusetts
- Bemporad, A., Morari, M., Dua, V., Pistikopoulos, E. N. (2002) *Automatica* 38(1): 3–20
- Biegler, L. (1984) *Solution of Dynamic Optimization Problems by Successive Quadratic Programming and Orthogonal Collocation*. *Comp. Chem. Eng.* 8, 3/4, 243–248
- Bonvin, D.; Srinivasan, B.; Hunkeler, D. (2006) *IEEE Cont. Systems Magazine* 26(6): 34–45
- Bryson, A. E., Ho, Y.-C. (1975) *Applied Optimal Control – Optimization, Estimation and Control*, Hemisphere publishing corporation
- Crescitelli, S., Nicoletti, B. (1973) *Chem.Eng.Science* 28: 463–471
- Diehl, M., Gerhard, J., Marquardt, W., Mönnigmann, M. (2008) *Computers & Chem. Eng.* 32:1279–1292
- Edgar, T. F., Himmelblau, D. M. (1988) *Optimization of Chemical Processes*. McGraw-Hill, New York
- Garcia, C. E., Prett, D. M., Morari, M. (1989) *Automatica* 25(3): 335–348
- Guntern, C., Keller, A., Hungerbühler, K. (1998) *Ind.Eng.Chem. Res.*, 37: 4017–4022
- Hertzberg, T., Asbjornsen, O. (1977) *Parameter Estimation in Nonlinear Differential Equations: Computer Applications in the Analysis of Data and Plants*. Science Press, Princeton
- Kadam, J. V., Marquardt, W. (2007) *Lecture Notes in Control and Information Sciences* 358: 419–434
- Kothare, M., Balakrishnan, V., Morari, M. (1996) *Amer. Control Conference*, Baltimore, 440–444
- Marchetti, A., Chachuat, B., Bonvin, D. (2007) *European Control Conference 2007*, Kos, Greece
- McFarlane, D. C., Glover, K. (1989) *Robust Controller Design Using Normalized Co-prime Plant Descriptions (LNCIS)*. Springer, New York.
- Pesch, H. J. (1989) *Opt.Cont.Appl.&Methods* 10: 147–171
- Pesh, H. J (1990) *Conference on Decision and Control*, Honolulu, HI, 952–953
- Ray, W. (1981) *Advanced Process Control*. McGraw Hill, New York
- Terwiesch, P., Agarwal, M., Rippin, D. W. T. (1994) *J.Proc. Control* 4: 238–258
- Zhou, K., Doyle, J. C., Glover, K. (1995) *Robust and Optimal Control*. Prentice Hall, Englewood Cliffs, New Jersey