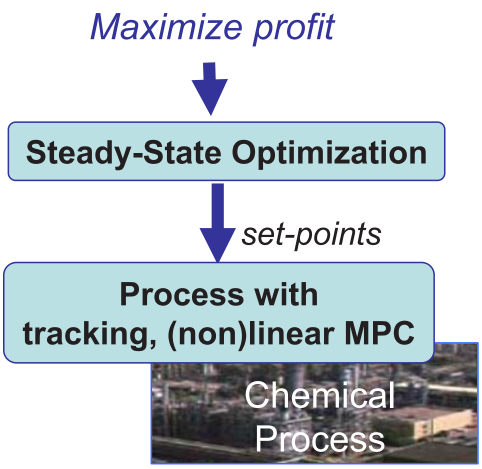
An efficient computational strategy for dynamic real-time optimization

Lynn Würth and Wolfgang Marquardt

Promatch Symposium REDUCIT
Frankfurt, 4 November 2008

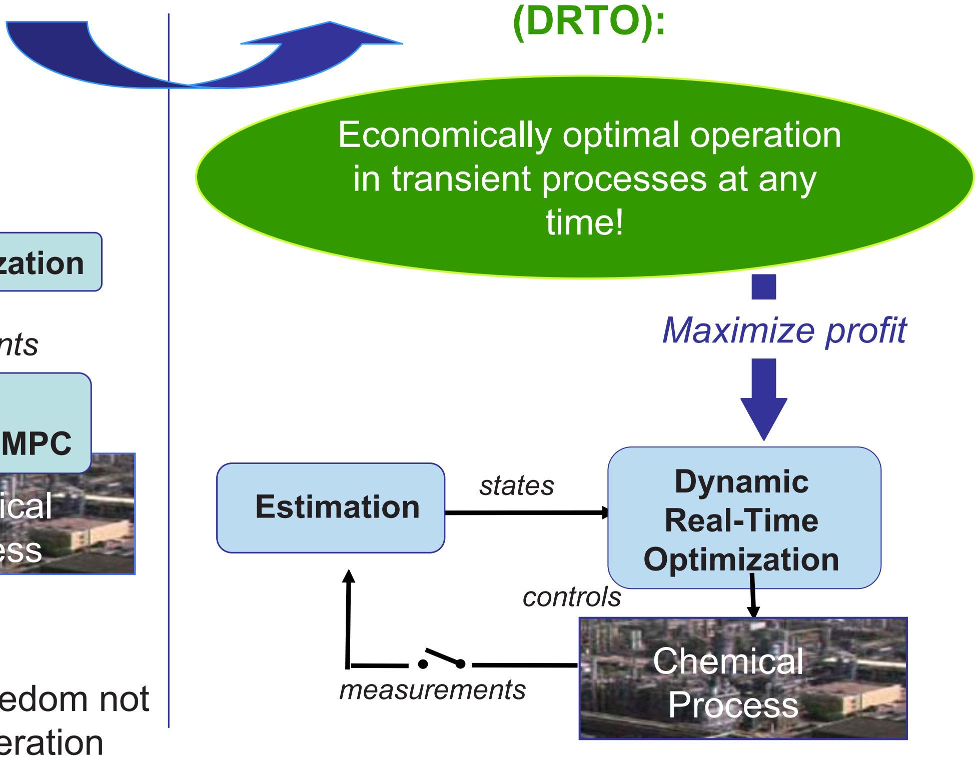
Dynamic Real-Time Optimization

Current set-up for RTO & MPC



- Inconsistent objectives
- Dynamic degrees of freedom not exploited for profitable operation

Dynamic Real-Time Optimization (DRTO):



Reduction of computational load

Obtain solutions of dynamic optimization problem in real-time but:

- Large-scale systems, complex models
- Different time-scales
- Severe nonlinearities e.g. reaction kinetics

Reduce computational time

Model Reduction

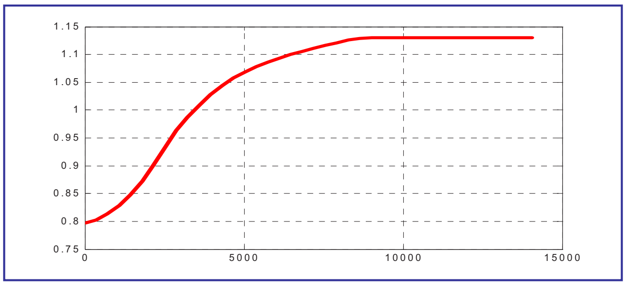
- Large-scale processes
- Trade-off with accuracy

Efficient computational methods

- Rigorous small/medium-scale models
- Combine with reduced models

Neighboring-Extremal Updates

Objective:
Fast updates of optimal trajectory under uncertainty



Neighboring Extremal Updates: Fast solution of related problem in the neighborhood of a nominal optimal solution

Related methods:

Indirect methods for optimal control: Neighboring extremal feedback law (Bryson and Ho, 1969, Pesch, 1979)

Direct methods: Diehl et al., 2002: Real-time iterations
Kadam and Marquardt, 2004: Sensitivity-based updates
Zavala and Biegler, 2008: Advanced-step NMPC

Optimization problem definition

Optimization problem formulation on receding horizon:

$$\min_{\boldsymbol{u}^j(t)} \Phi(\boldsymbol{x}(t), \boldsymbol{u}_j(t), t_0, t_f)$$ *Economic objective function*

s.t.

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}^j(t), \boldsymbol{d}^j(t), \hat{\boldsymbol{\theta}}),$$
$$\boldsymbol{y}(t) = \boldsymbol{g}(\boldsymbol{x}(t), \boldsymbol{u}^j(t), \boldsymbol{d}^j(t), \hat{\boldsymbol{\theta}}),$$ *DAE system (process model)*

$$\boldsymbol{x}(t^j) = \hat{\boldsymbol{x}}^j,$$ *initial conditions*

$$\boldsymbol{0} \geq \boldsymbol{h}(\boldsymbol{x}(t), \boldsymbol{y}(t), \boldsymbol{u}^j(t)),$$ *path constraints*

$$\boldsymbol{0} \geq \boldsymbol{e}(\boldsymbol{x}(t_f^j)),$$ *endpoint constraints*

$$t \in [t^j, t_f^j],$$ *time horizon*

$$t^j := t^{j-1} + \Delta t$$ *horizon shift (index j)*

Reformulation of the infinite-dimensional problem

- Control vector parameterization (single-shooting)

$$u_i(t) \approx \sum_{k \in \Lambda_i} z_{i,k}\phi_{i,k}(t)$$

- Collect different uncertainties in vector p:
  - Model uncertainties $\hat{\boldsymbol{\theta}}$
  - Disturbances d(t) parametrized by $\boldsymbol{c}^j$
  - Estimated state variables $\hat{\boldsymbol{x}}^j$

$$\boldsymbol{p} = (\boldsymbol{c}^j, \hat{\boldsymbol{\theta}}, \hat{\boldsymbol{x}}^j) \in \mathbb{R}^{n_p}$$

- **Reformulation as nonlinear parametric program problem (NLP)**

$$\min_{\boldsymbol{z}, \boldsymbol{t_f}} \Phi(\boldsymbol{z}, \boldsymbol{p})$$
$$\text{s.t.} \quad \boldsymbol{0} \geq \boldsymbol{g}(\boldsymbol{z}, \boldsymbol{p})$$
$$\boldsymbol{0} \geq \boldsymbol{e}(\boldsymbol{x}(t_f))$$

DAE system is solved by underlying numerical integration

Sensitivity-based updates

Idea: Exploit sensitivity information of offline/previously solved optimization problem to generate a fast approximation of the optimal update *(Fiacco, 1983)*

Solve Quadratic Program online *(Kadam and Marquardt, 2004)*

$$\min_{\Delta z} \quad \frac{1}{2}\Delta z^T L_{zz}(p_0)\Delta z + \Delta p^T L_{pz}(p_0)\Delta z + f_z(p_0)\Delta z$$

$$\text{s. t.} \quad g(p_0) + g_z(p_0)\Delta z + g_p(p_0)\Delta p \geq 0$$

L: Lagrange function
g: constraints

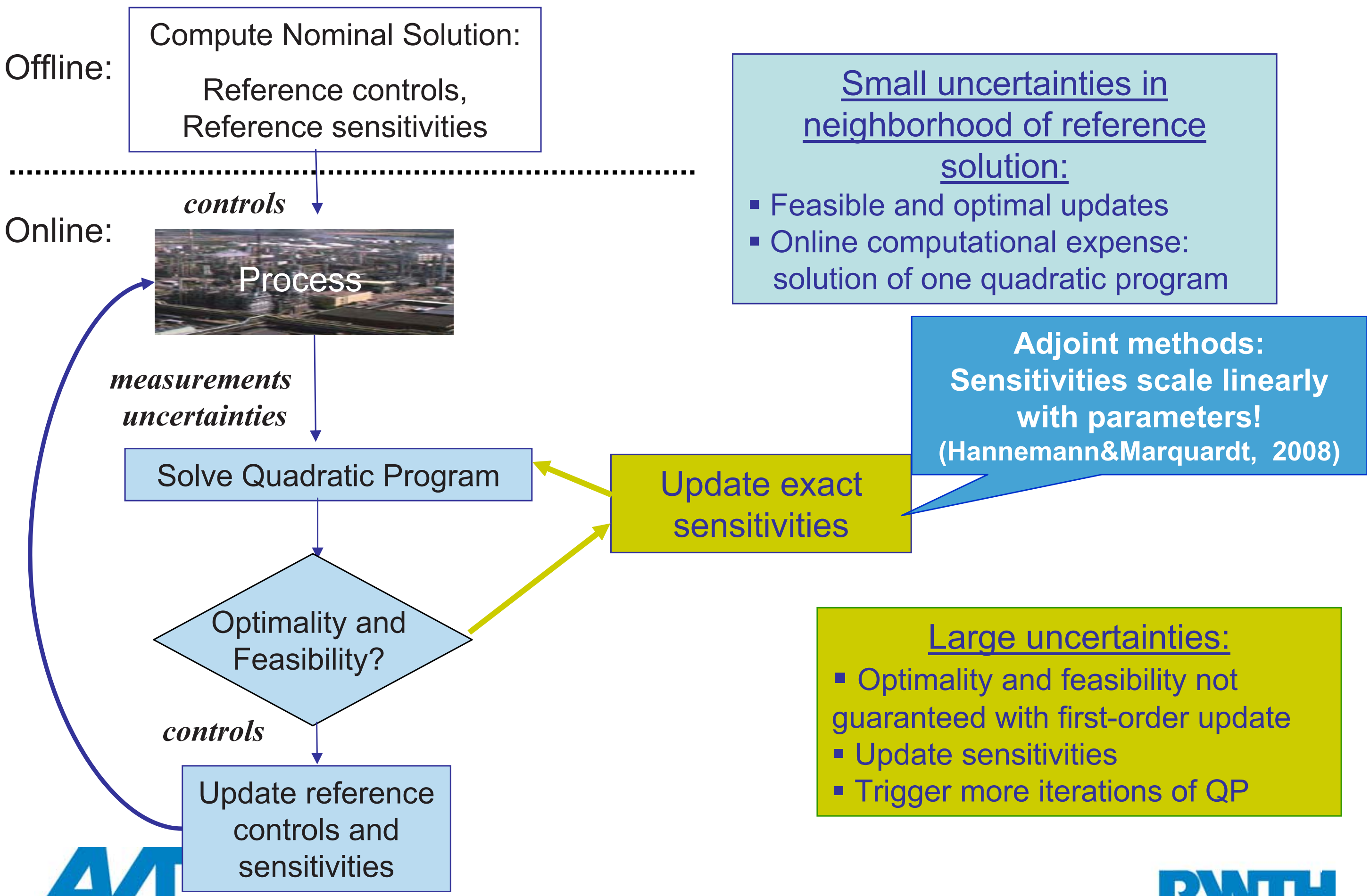
Parametric uncertainty: $\Delta p = p - p_0$

Sensitivities are computed at the nominal parameter p0

→ Fast first-order prediction of optimal controls

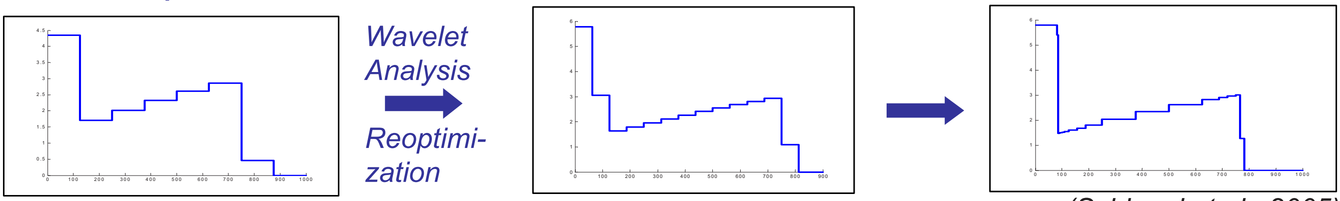
& update of active constraints

Algorithm for online trajectory updates



Grid refinement strategy

- Offline adaptation

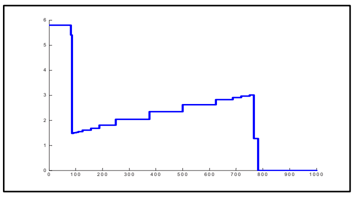


(Schlegel et al., 2005)

→ Reduction of degrees of freedom while maintaining solution accuracy!

- Online adaptation under uncertainties:

Adapted profile



Similar profile

- Small number of degrees of freedom
- **Fast computation**

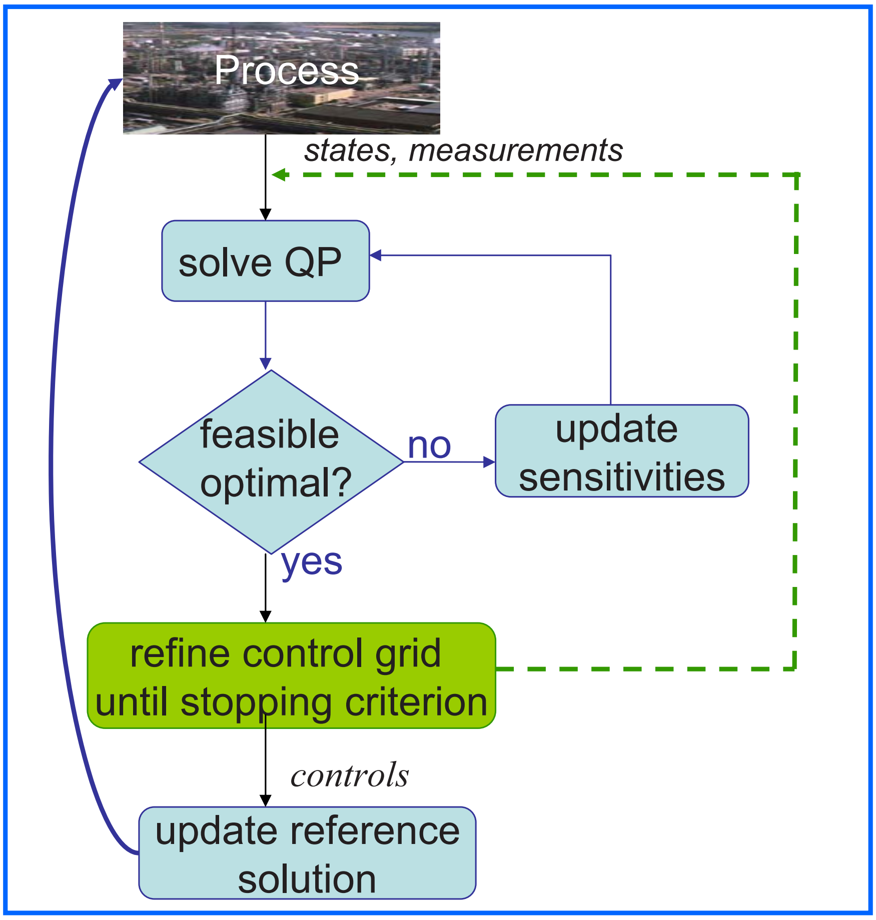
Change in profile

- Sensivity-based update on previous grid not sufficient
- **More iterations** to detect new profile

Online adaptation of control grid

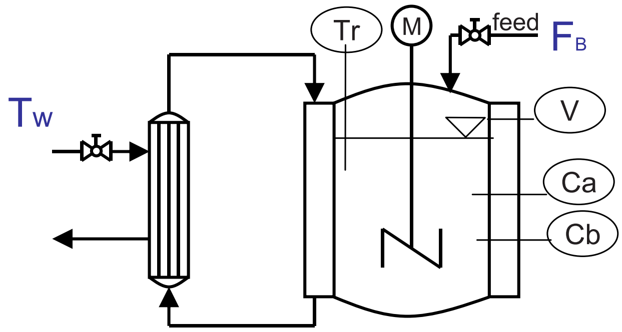
Implementation of grid refinement in the online algorithm:

Refinement of solution if time is available



Example 1: Real-time optimization of semi-batch reactor

Williams-Otto Benchmark Example



$$A+B \xrightarrow{k_1} C$$
$$C+B \xrightarrow{k_2} P+E$$
$$P+C \xrightarrow{k_3} G$$

Operational Objective: Maximize Product P

Optimization problem formulation

$$\max_{F_{B_{in}}(t), T_w(t)} \quad Yield(t_f)$$

s.t. *process model*

$$0 \le F_{B_{in}}(t) \le 5.784 \frac{\text{kg}}{\text{sec}},$$
$$20 \le T_w(t) \le 100 \text{ ℃},$$
$$60 \le T_r(t) \le 90 \text{ ℃},$$
$$0 \le V(t_f) \le 5 \text{ m}^3.$$

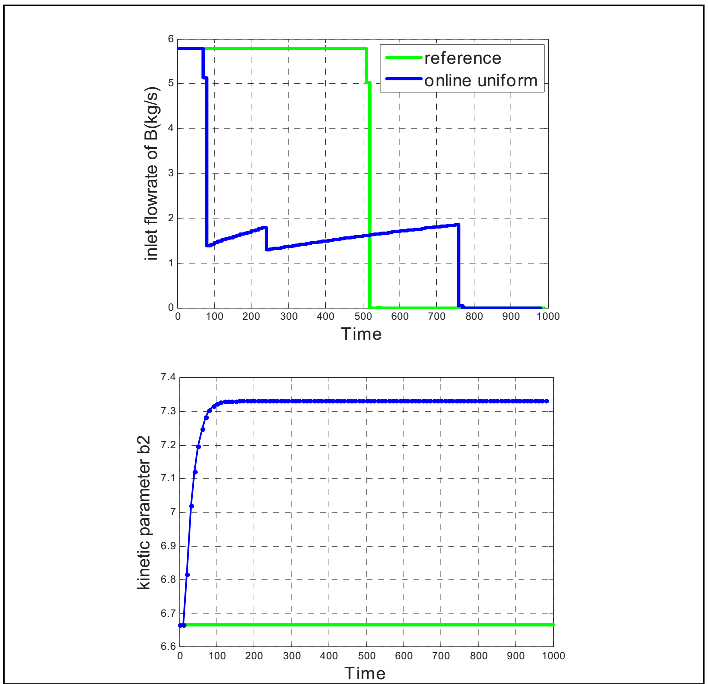
Uncertainties

$$k_i = a_i exp(\frac{-b_i}{T_r + 273.15}), i = 1, ..., 3$$
$$b_{1,0} = 6.6667 sec^{-1}$$
$$T_{in} = 35 \text{ ℃}$$
$$p := [T_{in}, b_1, \hat{\boldsymbol{x}}]$$

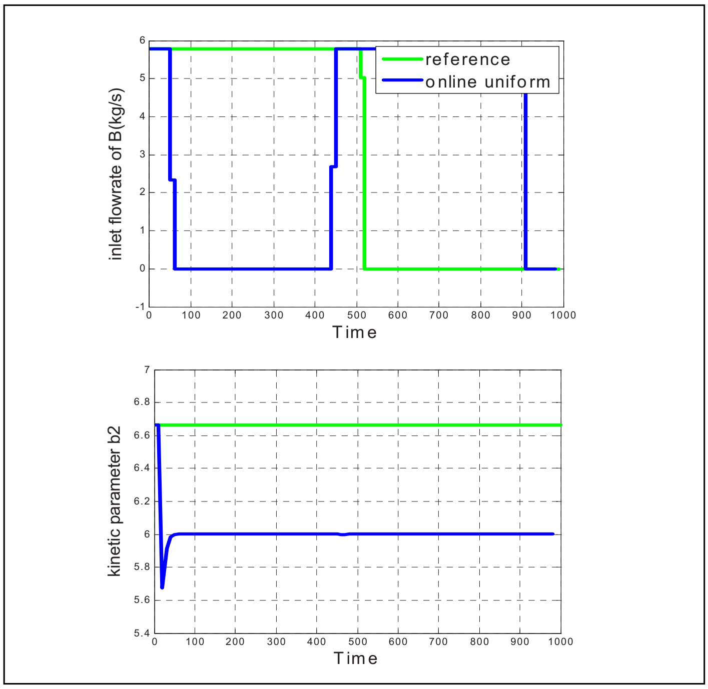
State estimation with EKF, sampling time of 10 s.

Real-time optimization results

Initial condition for kinetic parameter +10%

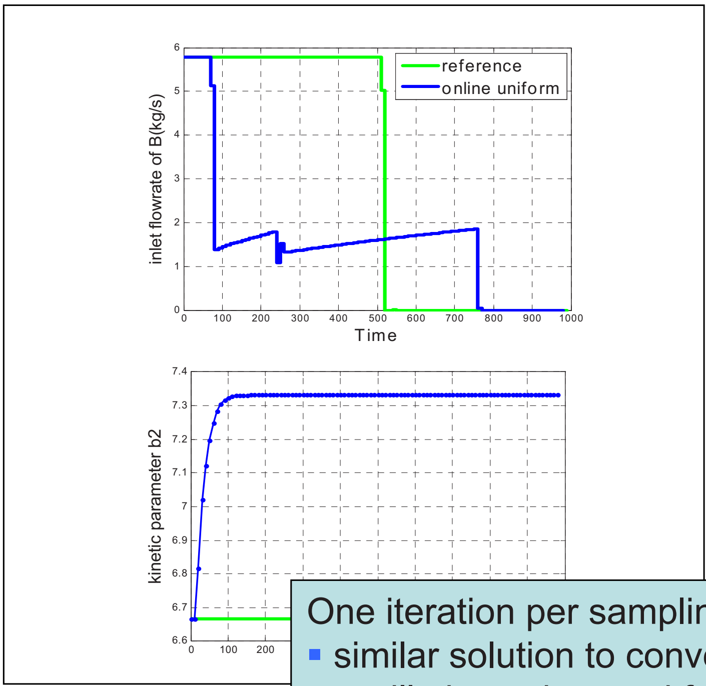


Initial condition for kinetic parameter -10%,

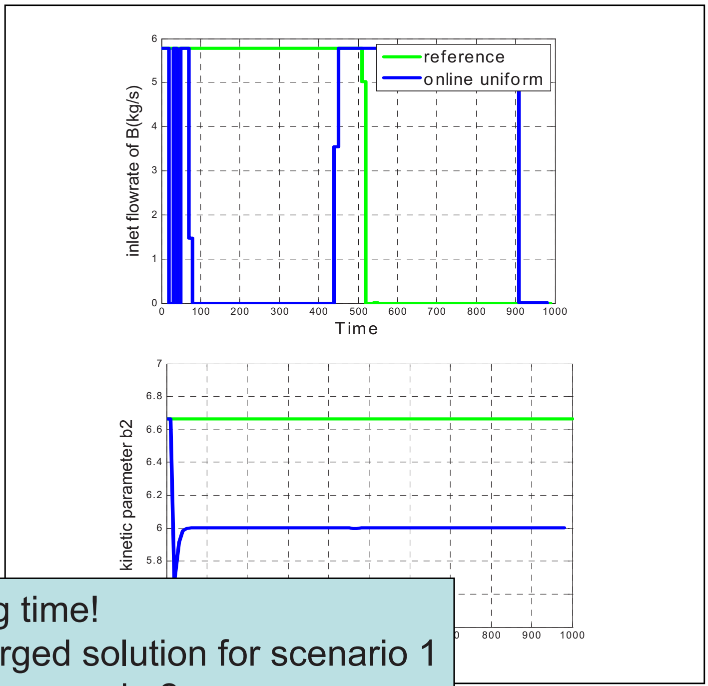


Comparison with online iterations

Initial condition for kinetic parameter +10%



Initial condition for kinetic parameter -10%,



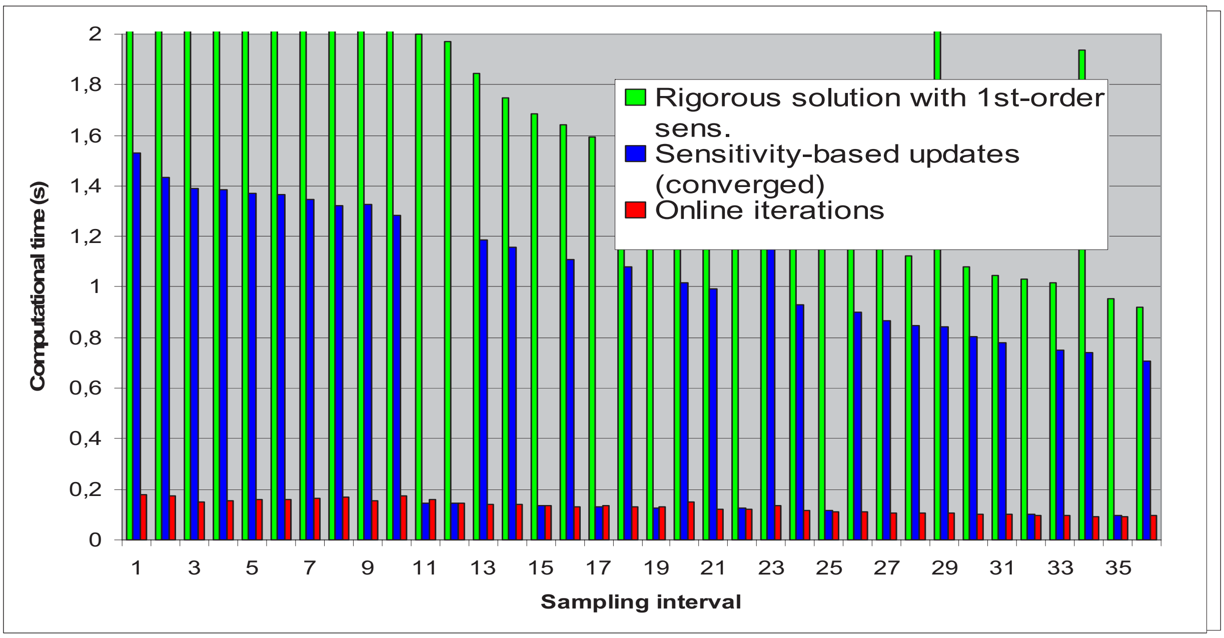
One iteration per sampling time!

- similar solution to converged solution for scenario 1
- oscillations observed for scenario 2

Comparison of computational performance

- Compared to rigorous solution with first order sensitivities
- Sensitvity-based updates with second-order exact sensitivities

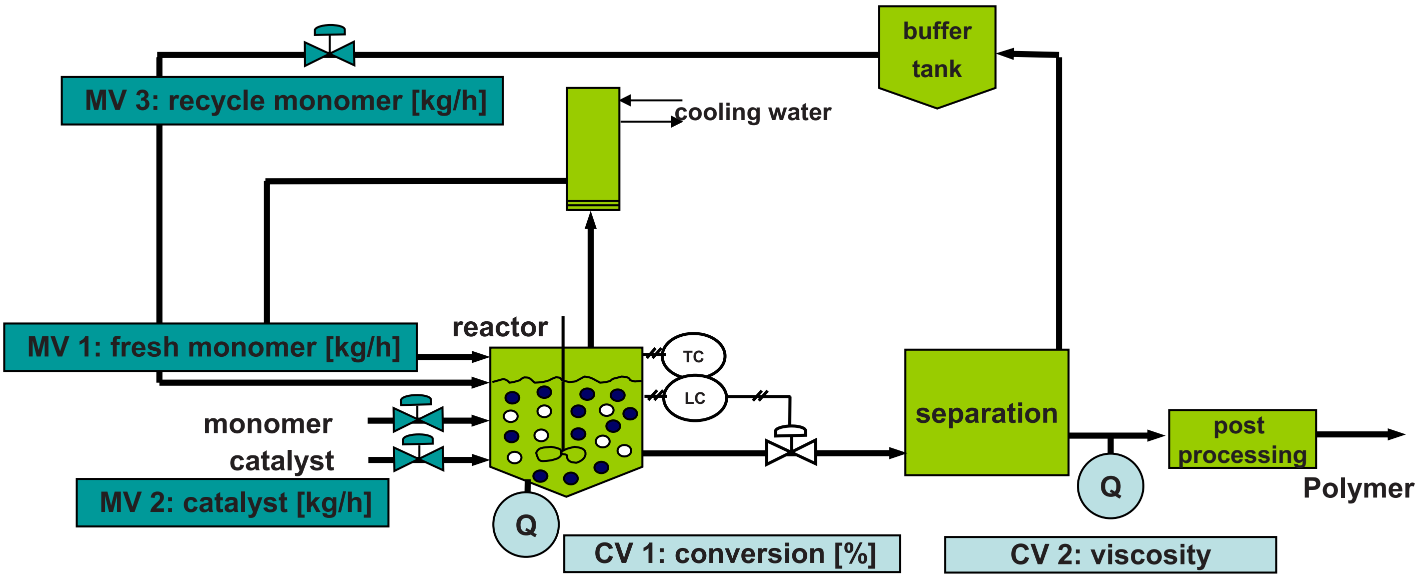
  ➡ fast convergence
- Online iterations: computational effort very low



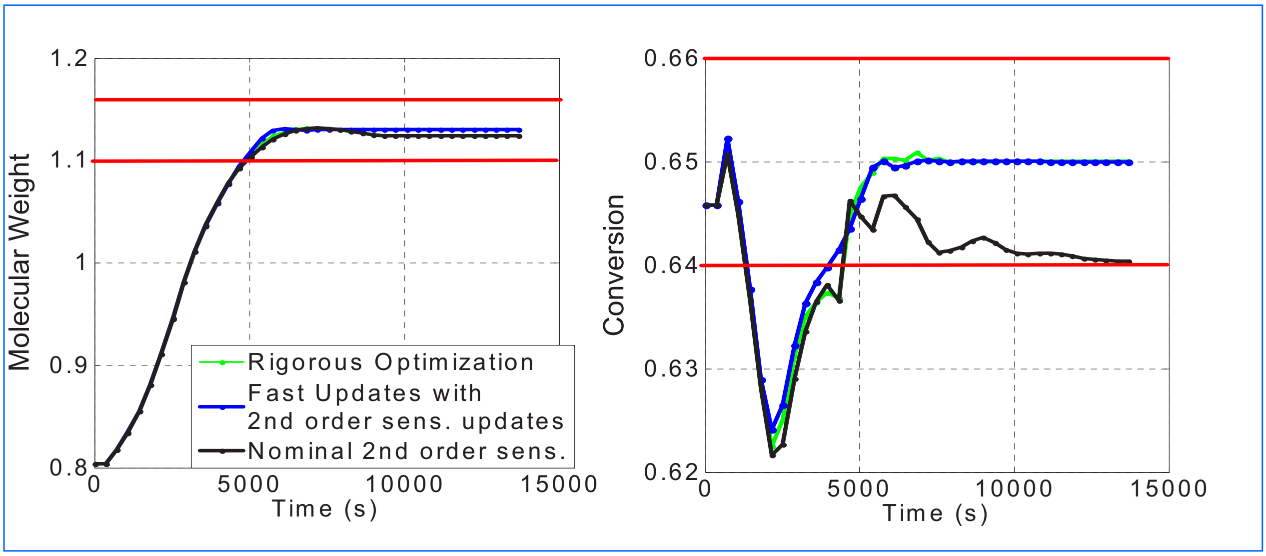
Example 2: Copolymerization Reactor

- complex reaction mechanism
- large-scale model (~ 2000 equations)
- 3 control variables, 6 path constraints

process operations tasks:
- optimal load change
- optimal grade change



Results with Fast Updates



Fast updates with online update of 2nd order sensitivities:

=> Close to rigorous optimal solution

Conclusions

- Method based on neighboring extremal updates was presented
  - Fast convergence
  - Solution accuracy is adjusted by a trigger

- Control performance:
  - Trade-off between computational delay and solution accuracy
  - Adjust number of iterations

- Adaptation:
  - Grid refinement allows reduction of degrees of freedom
  - Automatic structure detection => minimum number of parameters

Acknowledgments

Financial support from Marie-Curie Research Training Network PROMATCH funded by the European Commission (MRTN-CT-2004-512441)

and the German Research Foundation DFG (MA 1188/28-1) is gratefully acknowledged