Dynamic Real-time Optimization

Jitendra Kadam, Wolfgang Marquardt, Martin Schlegel

Lehrstuhl für Prozesstechnik

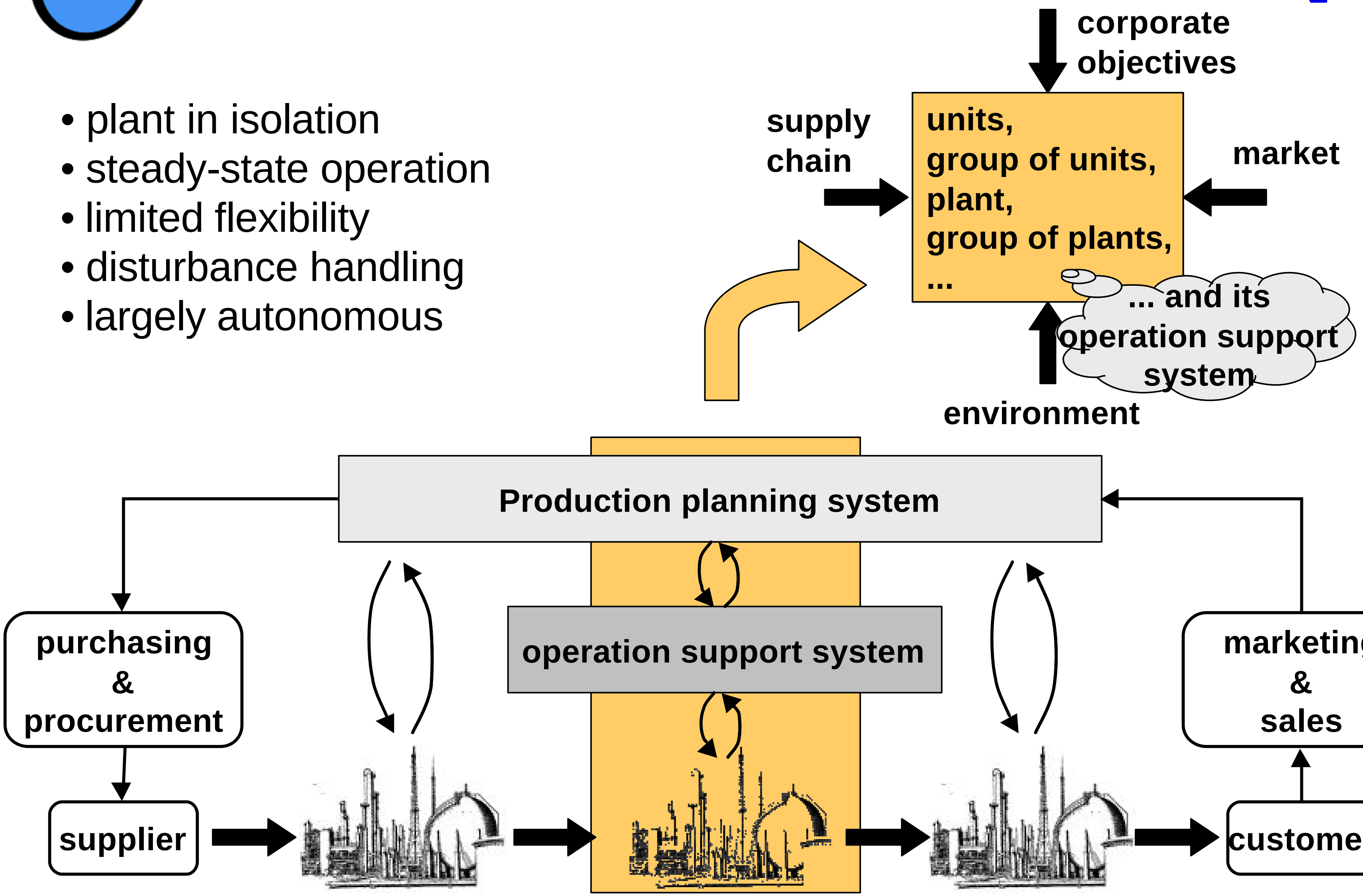
RWTH Aachen

INCOOP Workshop

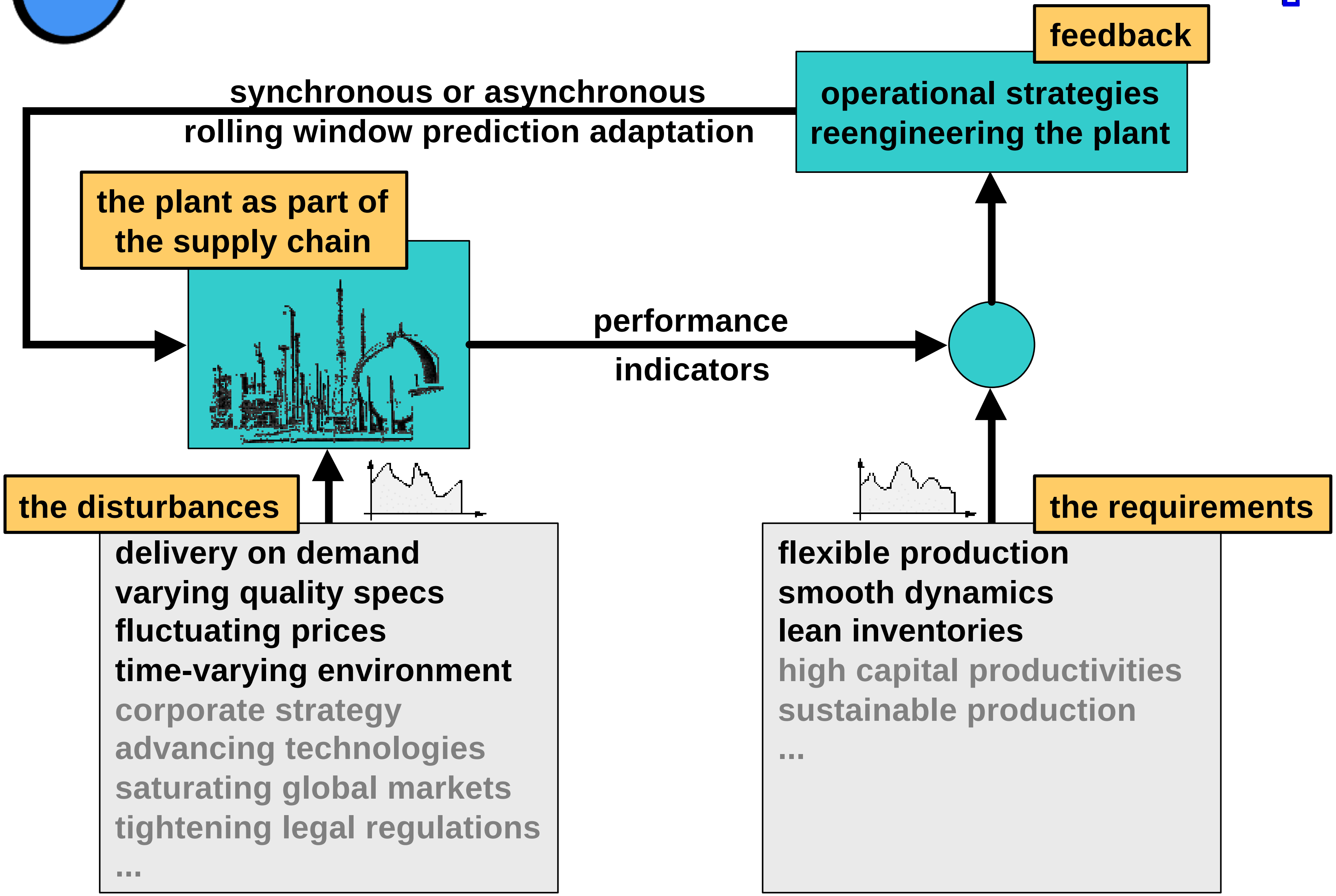
Düsseldorf, January 23 – 24, 2003

Operational strategies – the status

- plant in isolation
- steady-state operation
- limited flexibility
- disturbance handling
- largely autonomous



Manufacturing in the future



General operational objectives

economy

F

model uncertainties
disturbances

d

constraints
- equipment, safety, environment
- capacity, quality, reproducability

h

optimal operation of chemical processes

process model

$$0 = f(\dot{x}, x, u, p, d)$$

$$x(t_0) = x_0$$

$$y = g(x)$$

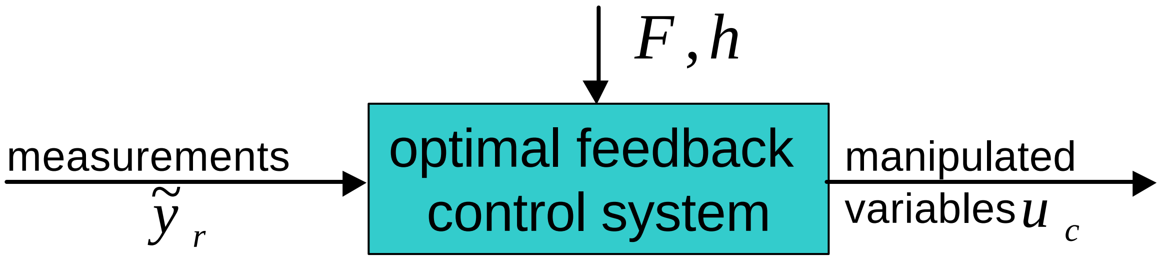
optimal profiles

manipulated variables

state variables

Why should they be constant over time?

Optimization-based control



2 coupled problems:

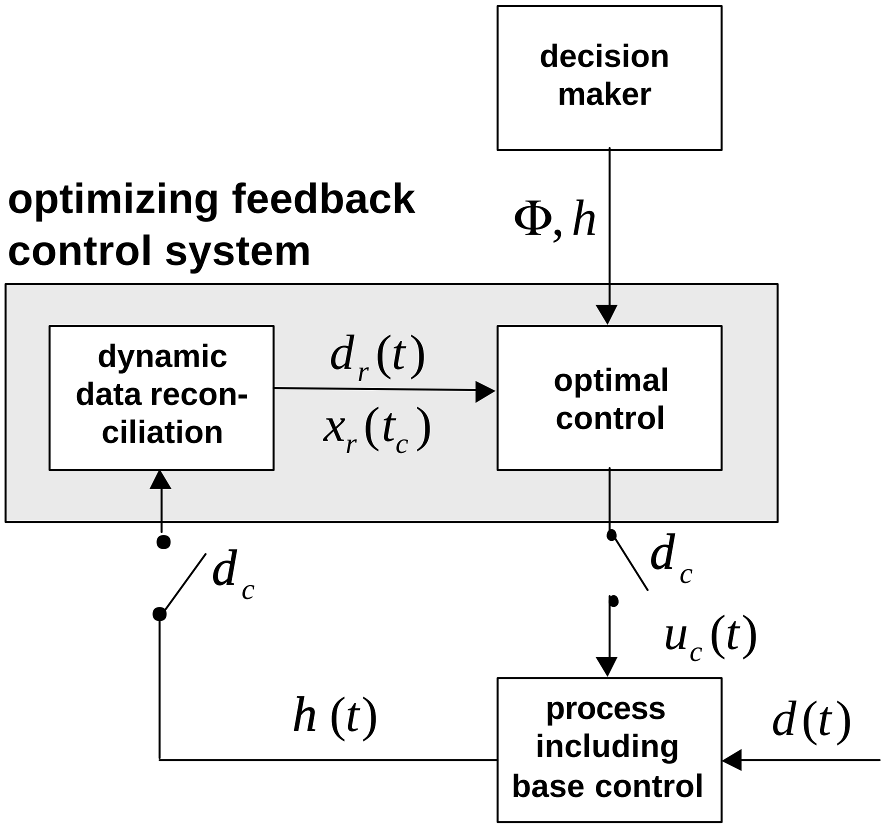
dynamic data reconciliation

$$
\begin{aligned}
&\min_{x_{r,0},\, d_r} \Phi_r(y_r, h, x_{r,0}, d_r, t_c, t_f) \\
\text{s.t.}\quad & 0 = f(\dot{x}_r, x_r, u_r, d_r) \\
& y_r = g(x_r) \\
& x_r(t_r) = x_{r,0} \\
& u_r = \mathrm{U}(u_c(\cdot)) \\
& 0 \geq h_r(x_r, d_r) \\
& t \in [t_r, t_c]
\end{aligned}
$$

optimal control

$$
\begin{aligned}
&\min_{u_c} \Phi_c(x_c, u_c, t_c, t_f) \\
\text{s.t.}\quad & 0 = f(\dot{x}_c, x_c, u_c, d_c) \\
& y_c = g(x_c) \\
& x_c(t_c) = x_r(t_c) \\
& d_c = \mathrm{D}(d_r(\cdot)) \\
& 0 \geq h_c(x_c, u_c) \\
& t \in [t_c, t_f]
\end{aligned}
$$

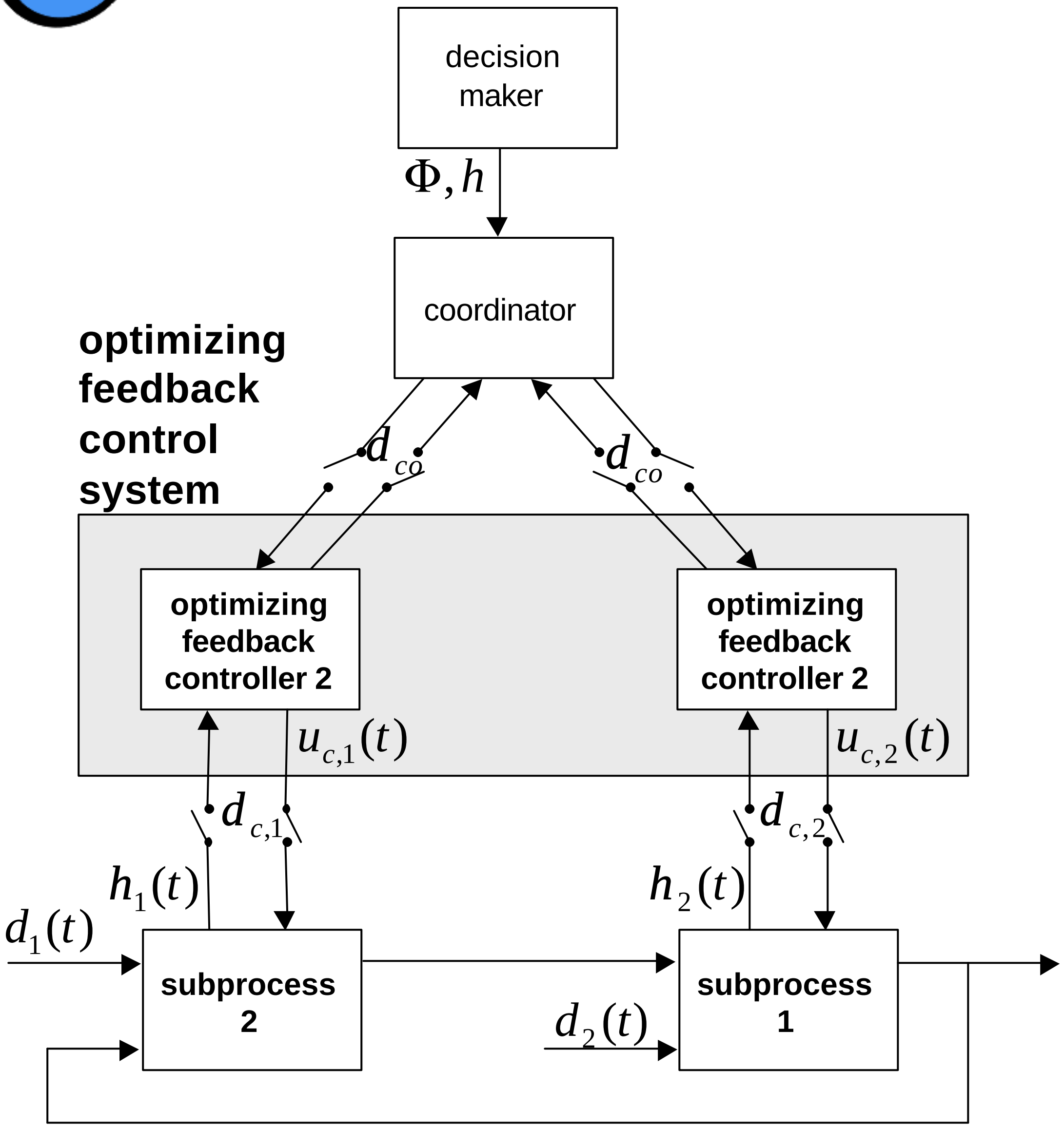
Direct solution approach



- solution of optimal control reconciliation problems at controller sampling frequency
- computationally demanding
- model complexity limited (INCOOP benchmarks $\Rightarrow$ large models!)
- lack of transparency, redundancy and reliability

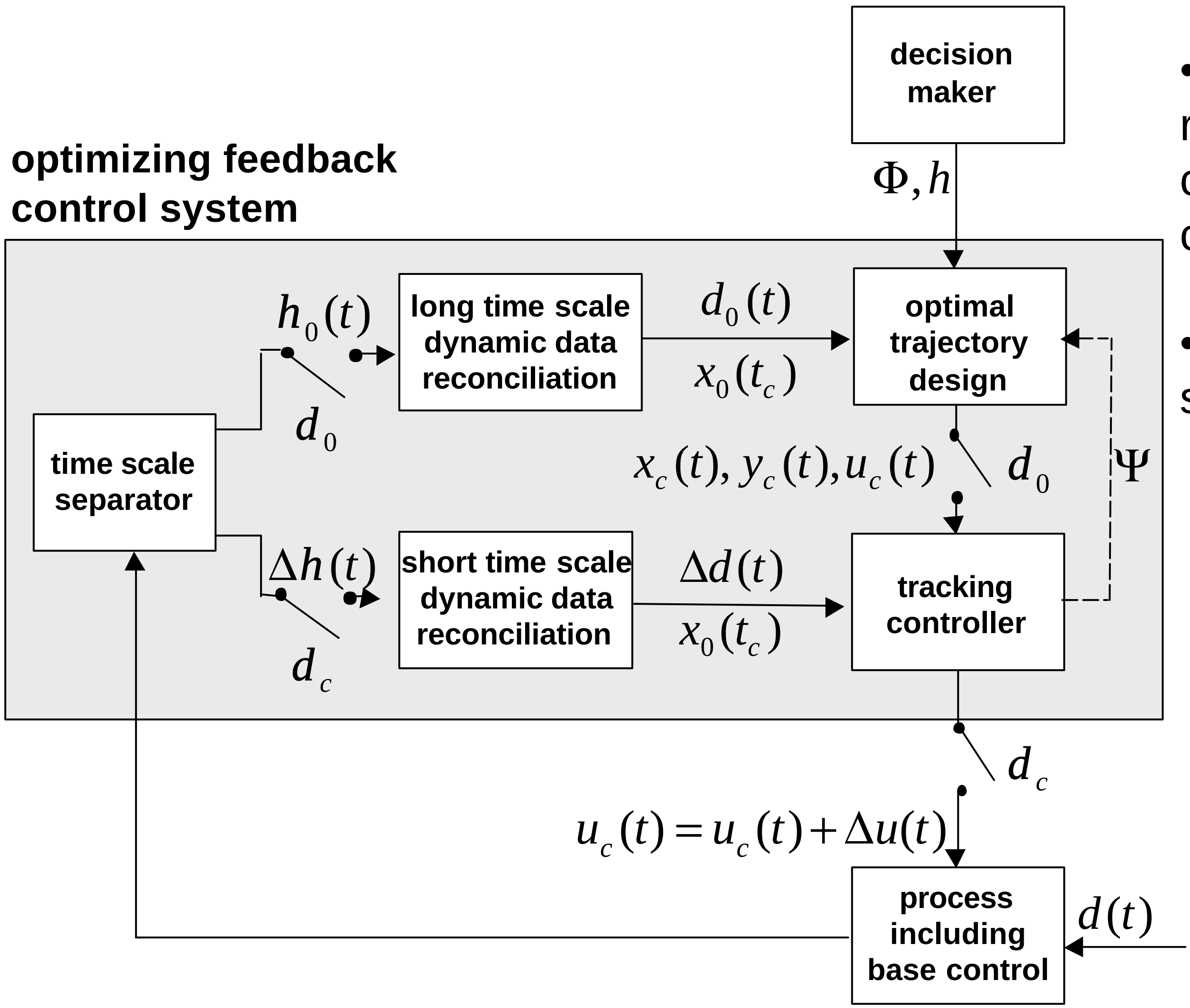
(Terwiesch et al., 1994; Helbig et al., 1998; Wisnewski & Doyle, 1996; Biegler & Sentoni, 2000)

Horizontal decomposition



- decentralization typically oriented at functional constituents of the plant
- coordination strategies enable approximation of ”true” optimum
- not adequately covered in optimization-based control and operations yet

(Mesarovic et al., 1970; Findeisen et al., 1980; Morari et al., 1980; Lu, 2000)



- generalizes steady-state real-time optimization and constrained predictive control

- requires (multiple) time-scale separation, e.g.

  $d(t) = d_0(t) + \Delta d(t)$

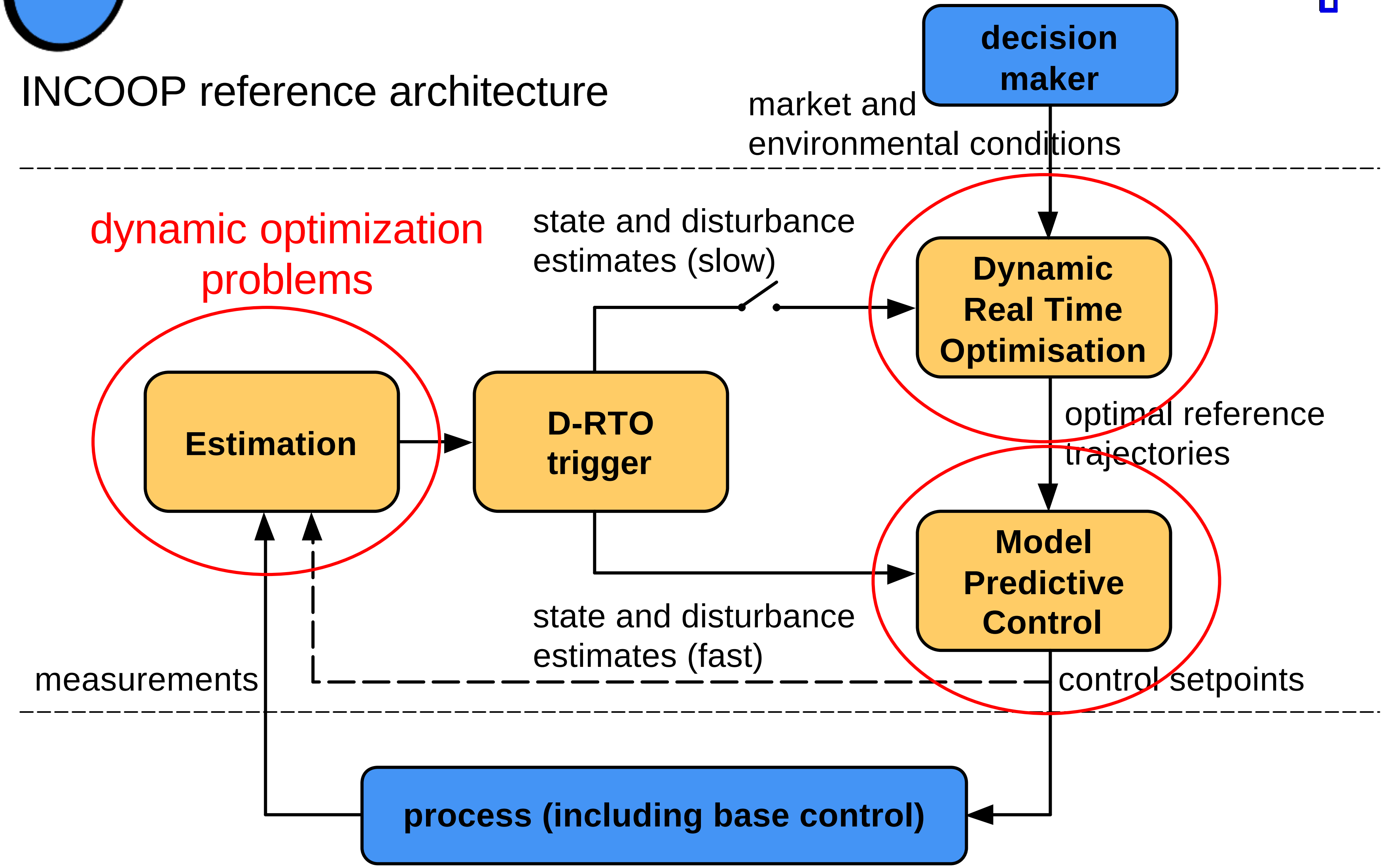
  with

  trend $d_0(t)$

  zero mean fluctuation $\Delta d(t)$

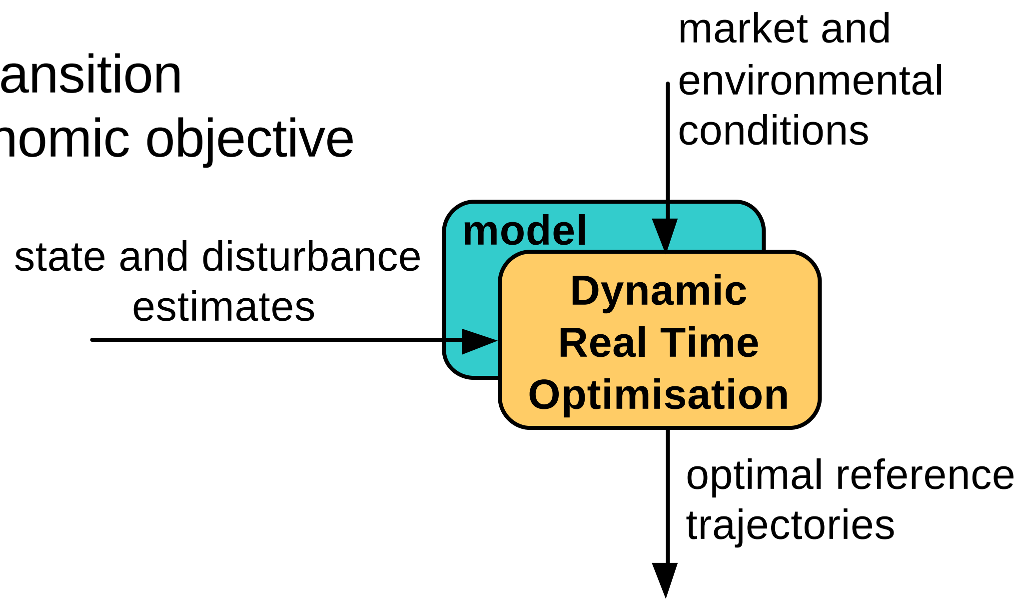
Vertical decomposition – INCOOP approach

INCOOP reference architecture



Focus: D-RTO block

Goal: determine optimal transition trajectories using an economic objective function



Challenges:

- Develop numerical **solution methods** which solve the problem robustly and sufficiently fast
- Develop techniques for **triggering a re-optimization** based on external conditions
- Implement **software framework** for enabling interaction with MPC and estimator

Mathematical problem formulation

$$\min_{u(t),p,t_f} \Phi(x(t_f))$$ objective function (e.g. cost)

s.t.

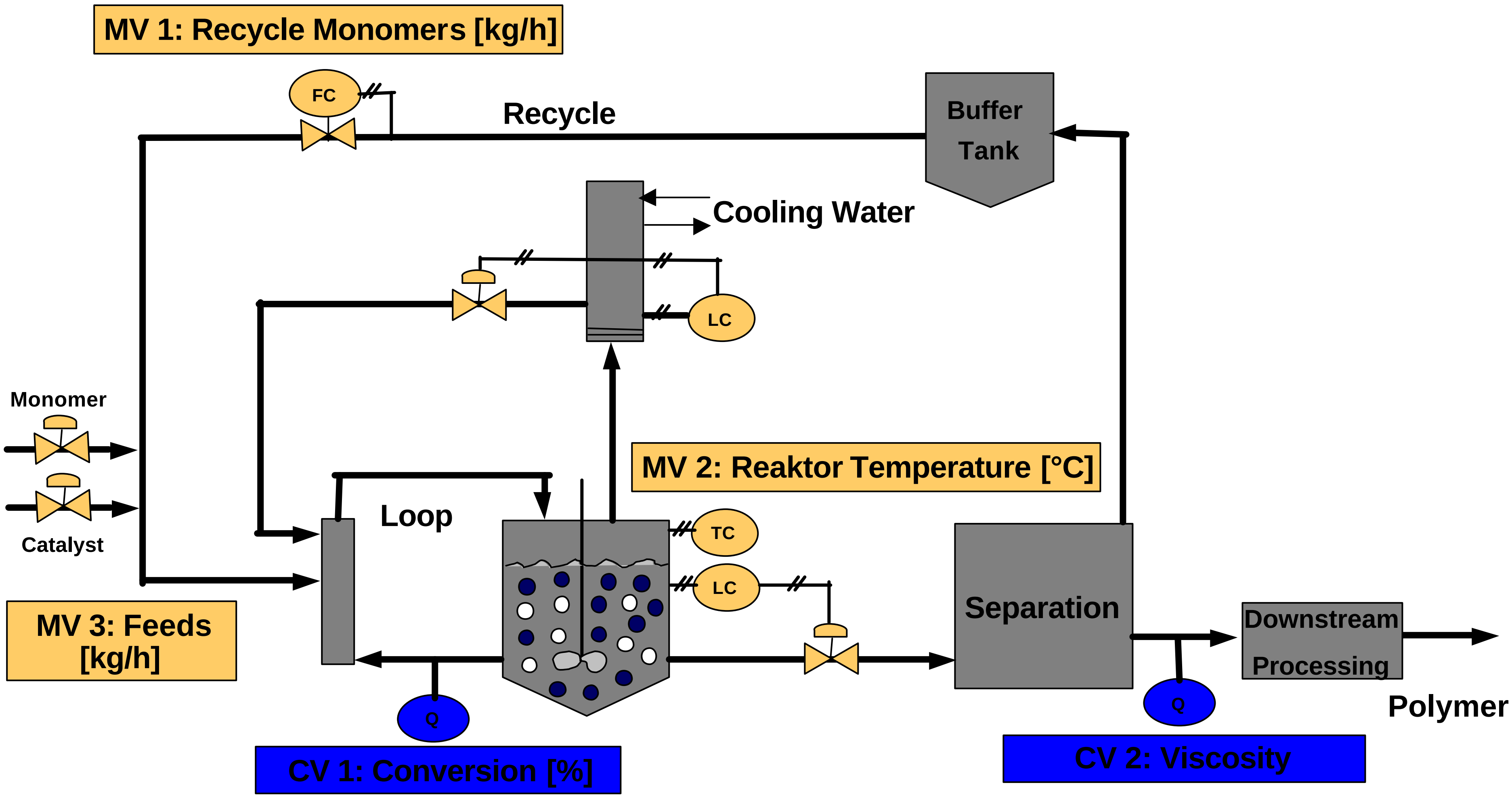
$$\left.\begin{aligned} M\,\dot{x} &= F(x,u,p,t), \quad t \in [t_0,t_f], \\ 0 &= x(t_0) - x_0, \end{aligned}\right\}$$ DAE system (process model)

$$0 \geq P(x,u,p,t), \quad t \in [t_0,t_f],$$ path constraints (e.g. temp. bound)

$$0 \geq E(x(t_f))$$ endpoint constraints (e.g. prod. spec.)

Degrees of freedom:	$u(t)$	time-variant control variables
	p	time-invariant parameters
	t_f	final time

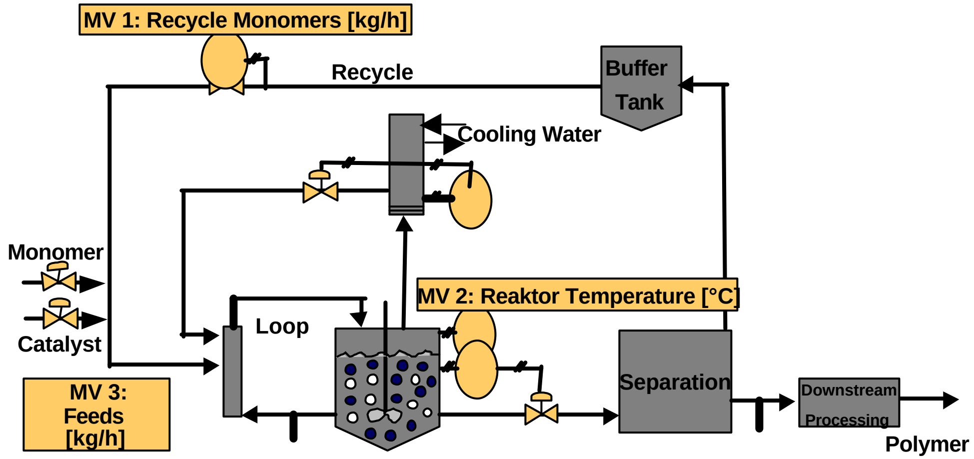
Example: Bayer Benchmark Process (I)

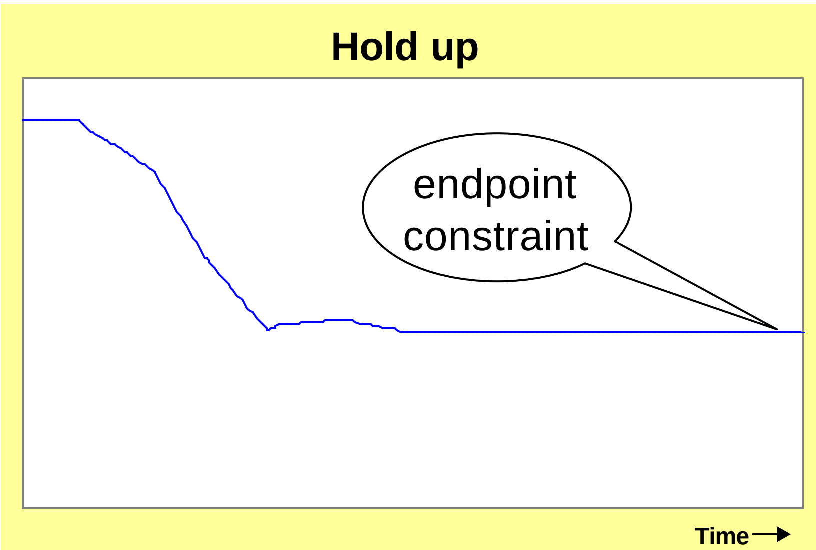


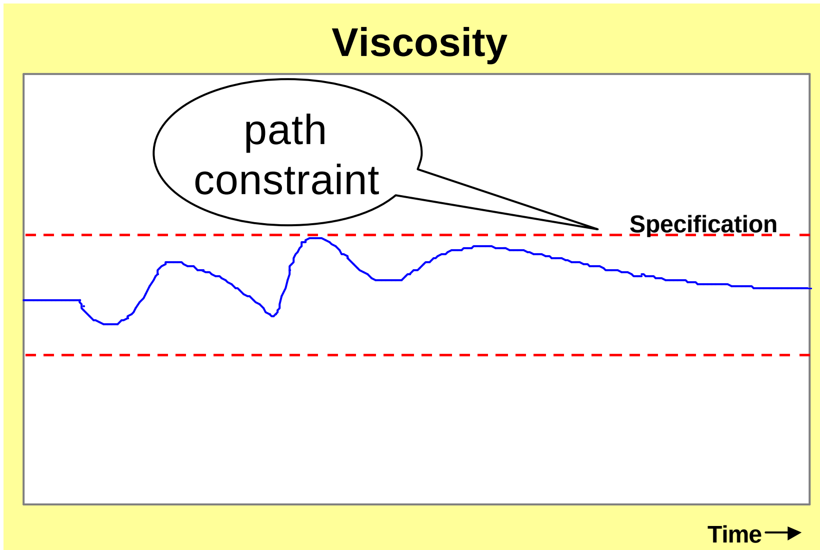
(From: Dünnebier & Klatt: Industrial challenges and requirements for optimization of polymerisation processes)

Polymerization process

- minimize time for load change
- three degrees of freedom
- path constraints on specifications







(From: Dünnebier & Klatt: Industrial challenges and requirements for optimization of polymerisation processes)

Solution approaches

<u>Indirect solution methods</u>
Necessary optimality conditions lead to multipoint boundary value problems:

- Highly accurate solutions with shooting techniques.
- Solution requires detailed a-priori knowledge of the optimal solution structure and appropriate estimates for adjoint variables.

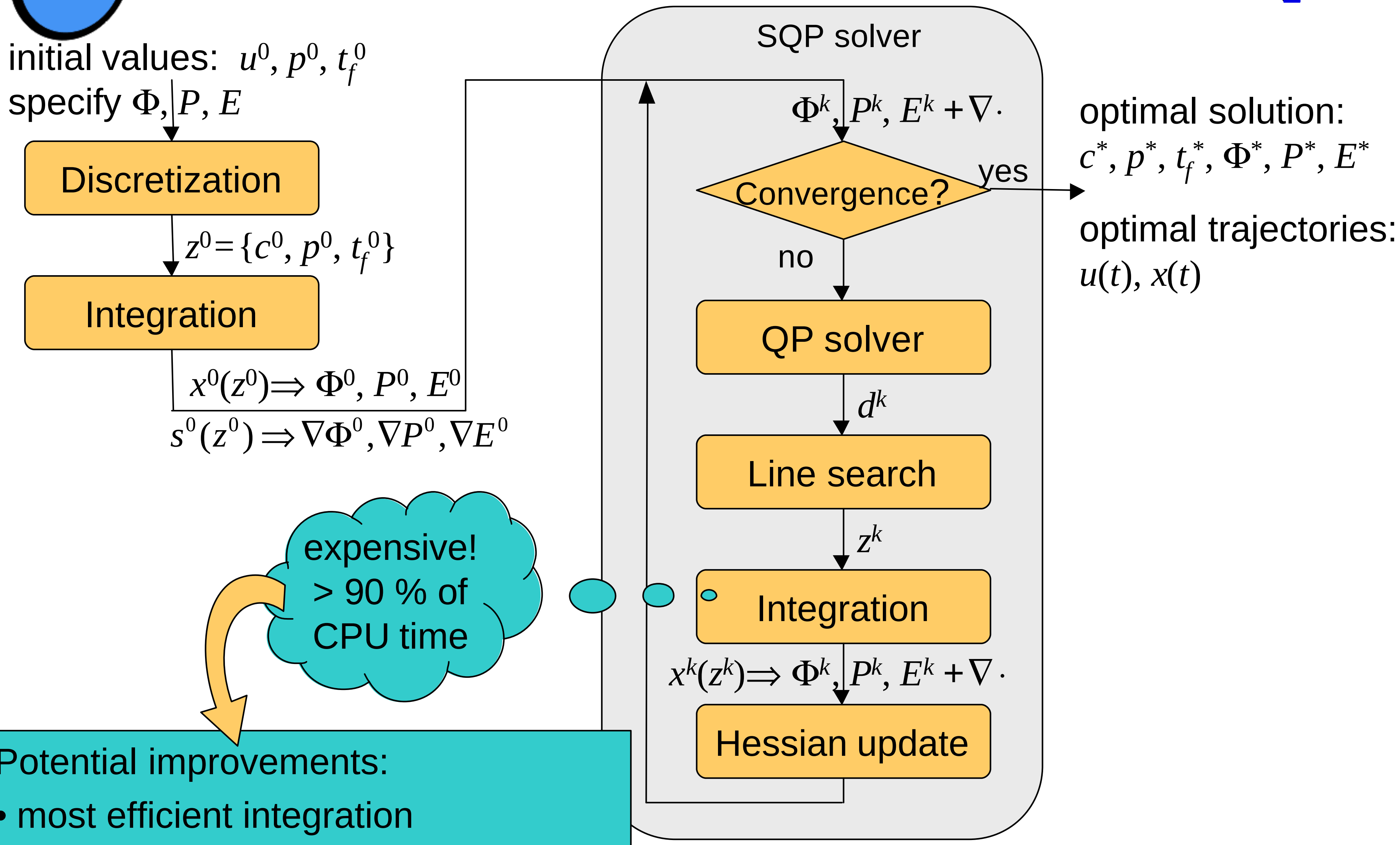
<u>Direct solution methods</u>
Conversion of dynamic optimization problem into nonlinear programming problem (NLP) by discretization…

- …of state and control variables. (simultaneous meth[...] e. collocation, mul[...]g)

used in INCOOP

- …of control variables only. (sequential method, i.e. single shooting)
- Succesfully applied with large-scale process models

Solution algorithm



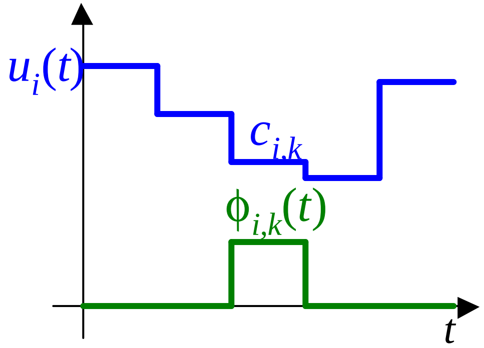
Potential improvements:

- most efficient integration
- as few NLP steps as possible

Sequential approach → single shooting

Control vector parameterization

$$u_i(t) \approx \sum_{k \in \Lambda_i} c_{i,k} f_{i,k}(t)$$



$f_{i,k}(t)$ parameterization functions

$c_{i,k}$ parameters

⇒ Reformulation as nonlinear programming problem (NLP)

$$\min_{c,p,t_f} \Phi(x(c,p,t_f))$$

s.t.

$$0 \geq P(x,c,p,t_i), \quad \forall t_i \in \mathrm{T},$$

$$0 \geq E(x(t_f))$$

DAE system solved by underlying numerical integration

- DAE system solved by underlying numerical integration
- Gradients for NLP solver typically obtained by integration of sensitivity systems

⇒ Numerical integration computationally most expensive (> 90 % of CPU time)

⇒ Computational effort strongly depends on size and complexity of process model

Algorithmic improvements – sequential approach

Sensitivity integration is expensive

State integration is expensive

Improve efficiency of sensitivity integration

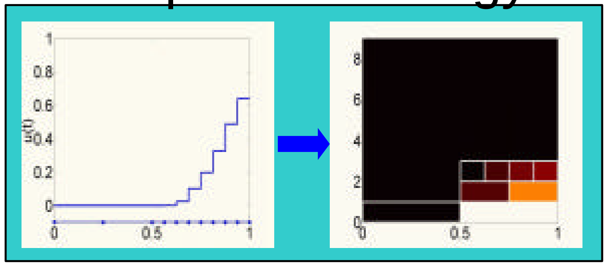
Reduce number of sensitivity parameters

Reduce model complexity

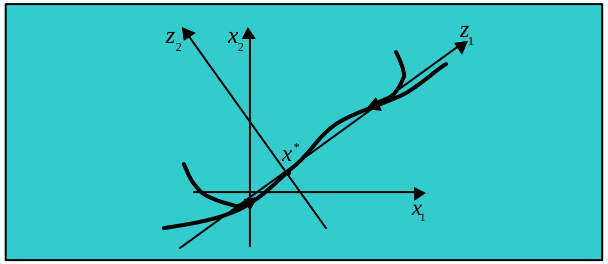
New solver for sensitivity integration

$$X_{k+1} = X_k - \begin{bmatrix} A - \frac{M}{h_j} & & & \\ A_1 & A - \frac{M}{h_j} & & \\ \vdots & & \ddots & \\ A_{n_z} & & & A - \frac{M}{h_j} \end{bmatrix}^{-1} f(X_k, z)$$

Control grid adaptation strategy



Methods for model reduction



Dynamic optimization problem:

$$\min_{z} \Phi(x,z,t)$$

$$\text{s.t.} \quad M\,\dot{x} = f(t,x,z)$$

$$0 = x(t_0) - x_0$$

$$0 \geq P(x,z,t)$$

$$0 \geq E(x(t_f))$$

$$\xrightarrow[z=\{c,\,p,\,t_f\}]{\frac{\partial}{\partial z}}$$

$$\boxed{M\,\dot{s} = \left(\frac{\partial f}{\partial x}\right) s_i + \frac{\partial f}{\partial z_i} \qquad i = 1,\ldots,n_z}$$

Typical solution approaches based on BDF-type integrators

- Caracotsios & Stewart (1985), Maly & Petzold (1996), Feehery et al. (1997)

New idea: Use one-step extrapolation method

- Based on LIMEX algorithm (Deuflhard et al. (1983,87))
- Extension for sensitivity computation: Schlegel et al. (2002)

Combined state and sensitivity system

$$\left.\begin{aligned} M\,\dot{x} &= F(x,z,t) \\ M\,\dot{s} &= \underbrace{\left(\frac{\partial F}{\partial x}\right)}_{=:A} s_i + \frac{\partial F}{\partial z_i} \qquad i = 1,\ldots,n_z \end{aligned}\right\} \; M\,\dot{X} = \mathrm{f}(X,z,t)$$

with $X = [x, \quad s_1, \quad \ldots \quad s_{n_z}]^T$

Efficient solution of the combined system

- M is identical in both systems.
- A is already required for state integration.

Linear-implicit
Euler discretization

Reuse LU decomposition

$$X_{k+1} = X_k - \begin{bmatrix} A - \dfrac{M}{h_j} & & \\ A_1 & A - \dfrac{M}{h_j} & \\ \vdots & & \ddots \\ A_{n_z} & & A - \dfrac{M}{h_j} \end{bmatrix}^{-1} \mathrm{f}(X_k, z)$$

neglect off-diagonal elements

Extrapolation algorithm for simultaneous state and sensitivity integration

Compute $A_0 = \frac{\partial}{\partial y}(f(y_0, p))$

for j=1,...,j_{max} while convergence criterion not satisfied

$$h_j = H / j$$

$$LU = A_0 - \frac{B}{h_j}$$

for k=0,...,j-1

$$y_{k+1} = y_k - (LU)^{-1} \; f(y_k, p)$$

$$s_{i,k+1} = s_{i,k} - (LU)^{-1}\left(A(y_k)s_{i,k} + \frac{\partial f}{\partial p_i}(y_k) \right), \quad i = 1,\ldots,n_z$$

Reuse LU decomposition

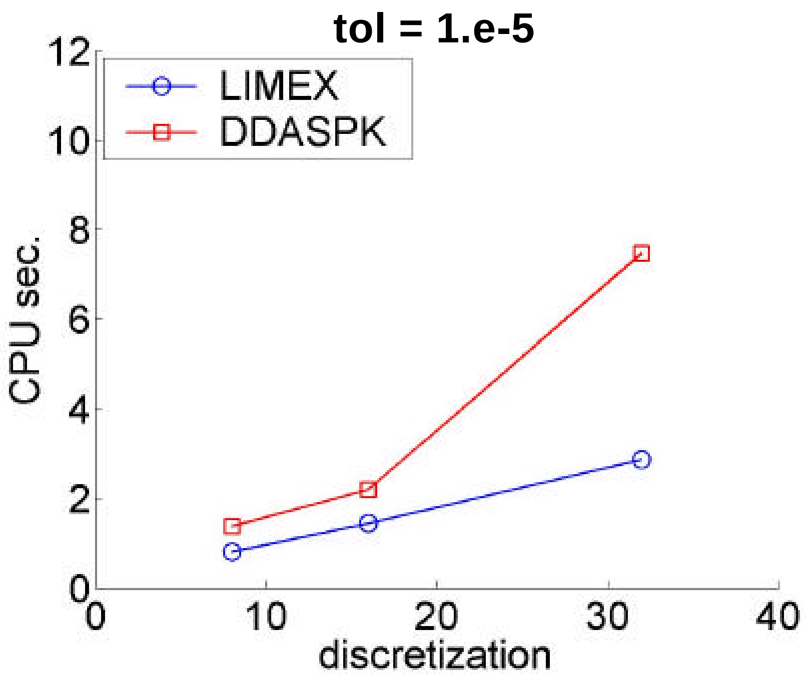
$$T_{j,1} = Y_j$$

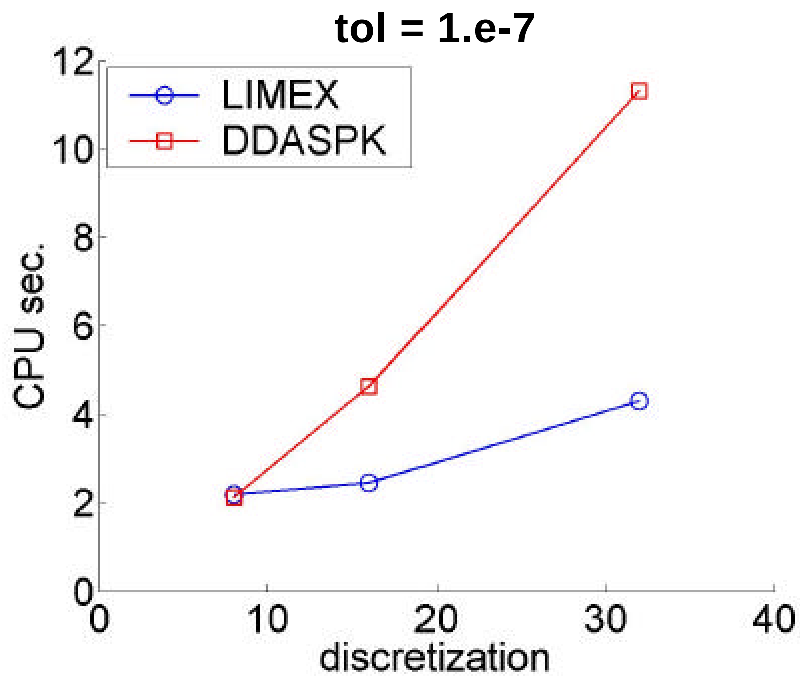
if j>1 compute $T_{j,j}$ and check convergence

$$X_{new} = X_{j,j}$$

(here simplified for M = const.)

Small example problem, solved for two different tolerances





⇒ One-step extrapolation faster than multistep BDF with increasing level of discretization

⇒ Used as standard for optimization of INCOOP benchmark problems

Algorithmic improvements – sequential approach

Sensitivity integration is expensive

State integration is expensive

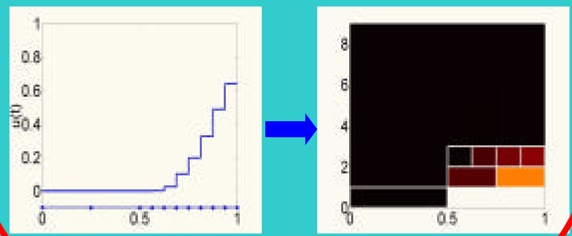
Improve efficiency of sensitivity integration

New solver for sensitivity integration

$$X_{k+1} = X_k - \begin{bmatrix} A - \frac{M}{h_j} & & & \\ A_1 & A - \frac{M}{h_j} & & \\ \vdots & & \ddots & \\ A_{n_z} & & & A - \frac{M}{h_j} \end{bmatrix}^{-1} f(X_k, z)$$

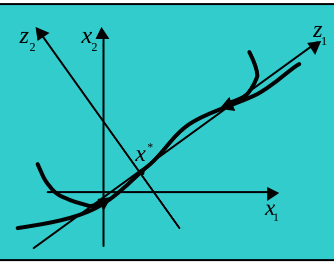
Reduce number of sensitivity parameters

Control grid adaptation strategy



Reduce model complexity

Methods for model reduction



Multiscale representation

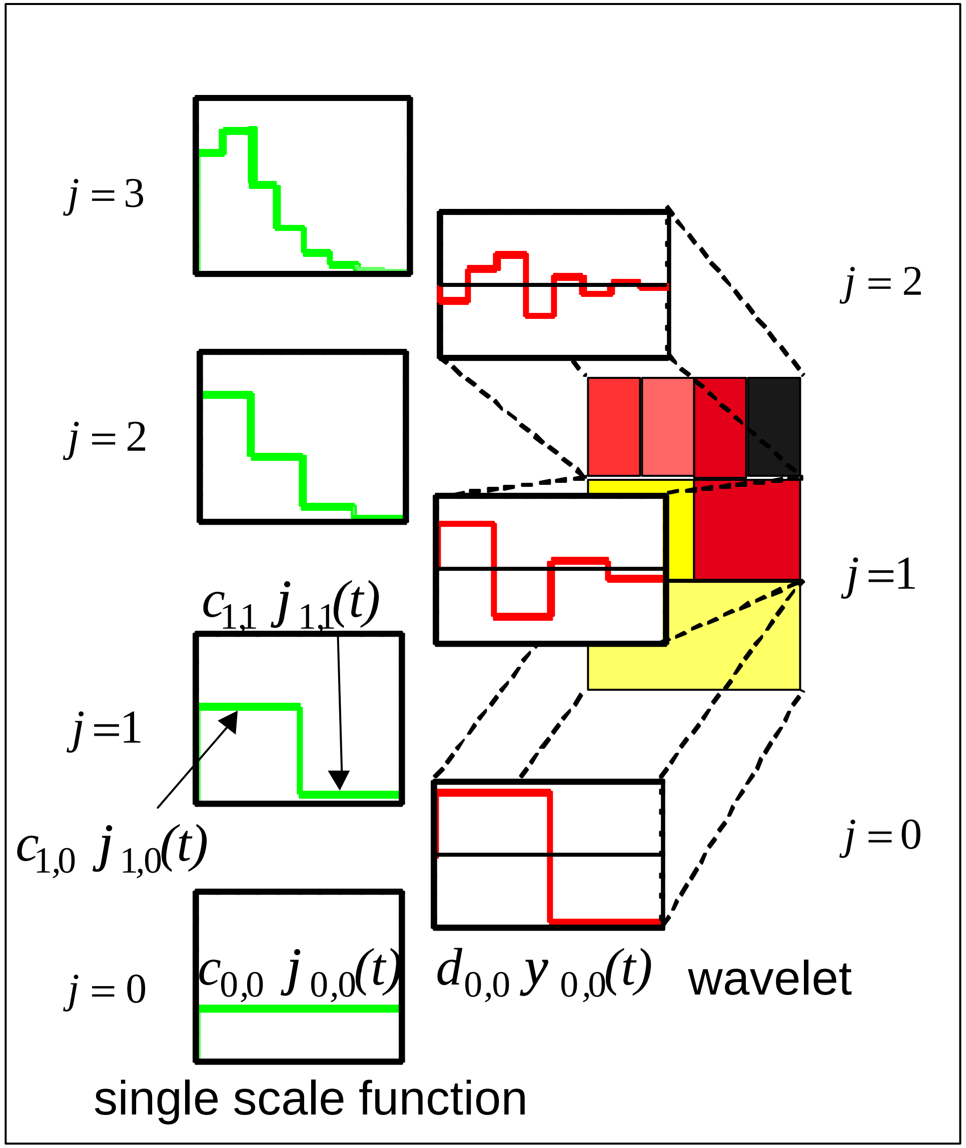
Different **representations** of the **same function …**

… for problem discretization:

$$u = \sum_{(j,k)\in\Lambda_j} c_{j,k}\,\varphi_{j,k}(t)$$

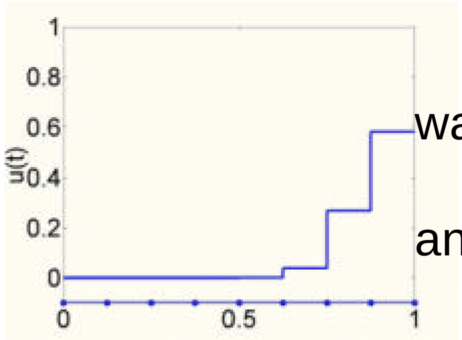
… for grid point elimination analysis:

$$u = c_{0,0}\,\varphi_{0,0}(t) + \sum_{(j,k)\in\Lambda_\psi} d_{j,k}\,\psi_{j,k}(t)$$



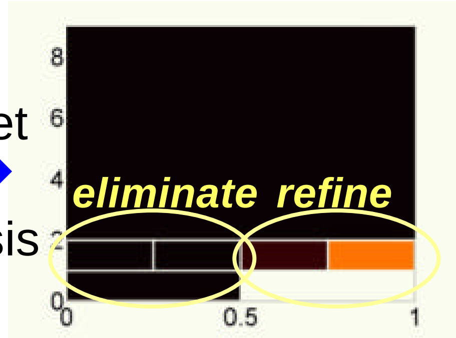
Mesh analysis

- Concepts from signal analysis
- Grid point elimination
- Grid point insertion

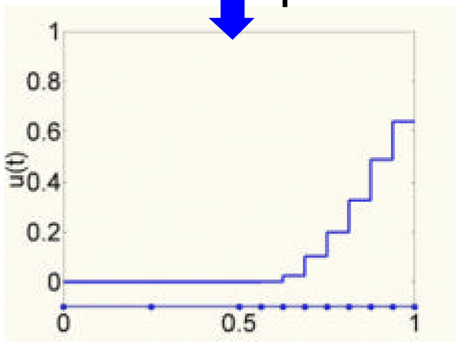


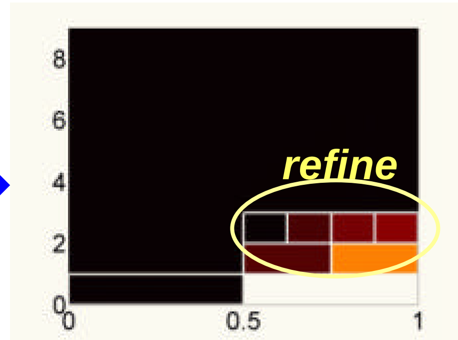
coarse initial mesh

wavelet analysis

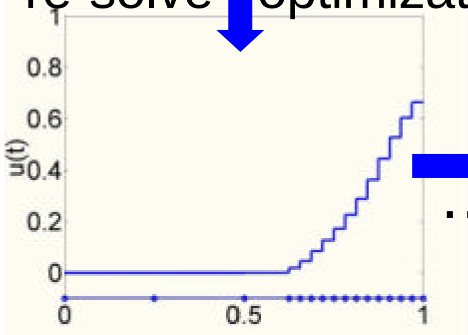


re-solve optimization





re-solve optimization

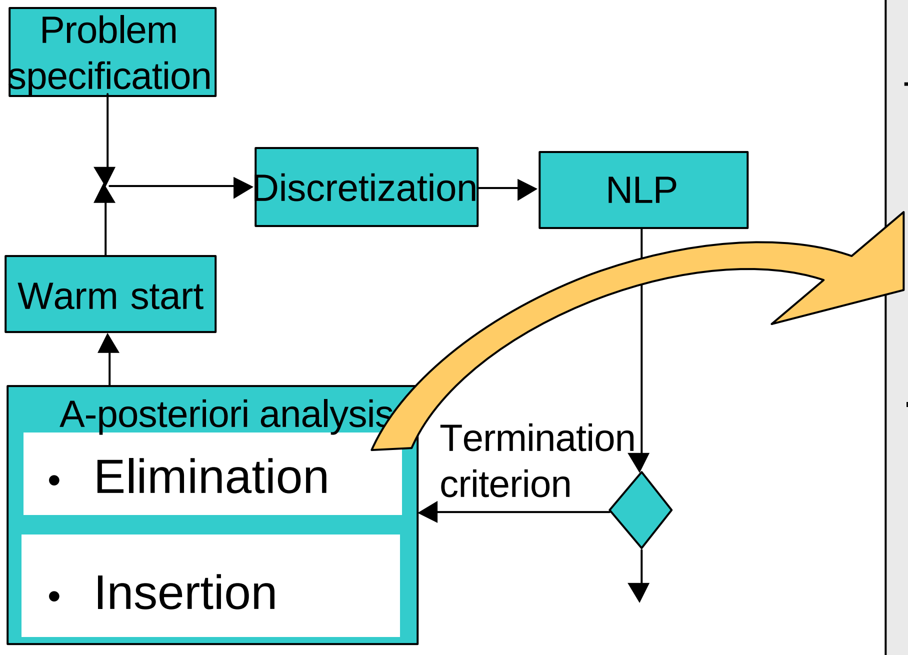


... until stopping criterion met.

Repetitive procedure

- Re-optimize problem on refined mesh
- Profile from previous solution as initial guess
- Decouple optimization and adaptation

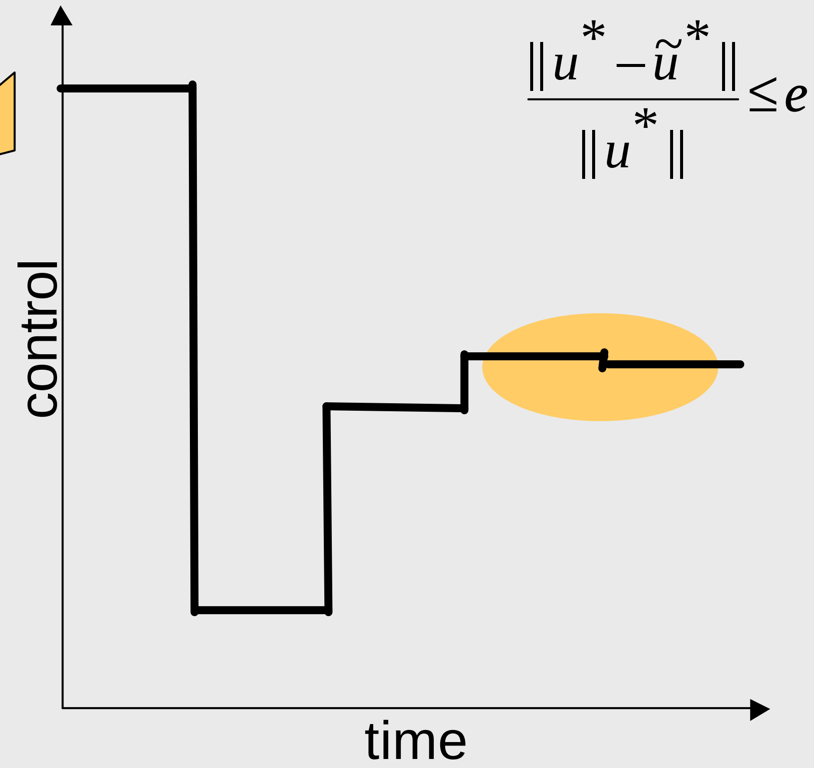
Elimination of parameterization functions



Analysis of wavelet coefficients of control variables

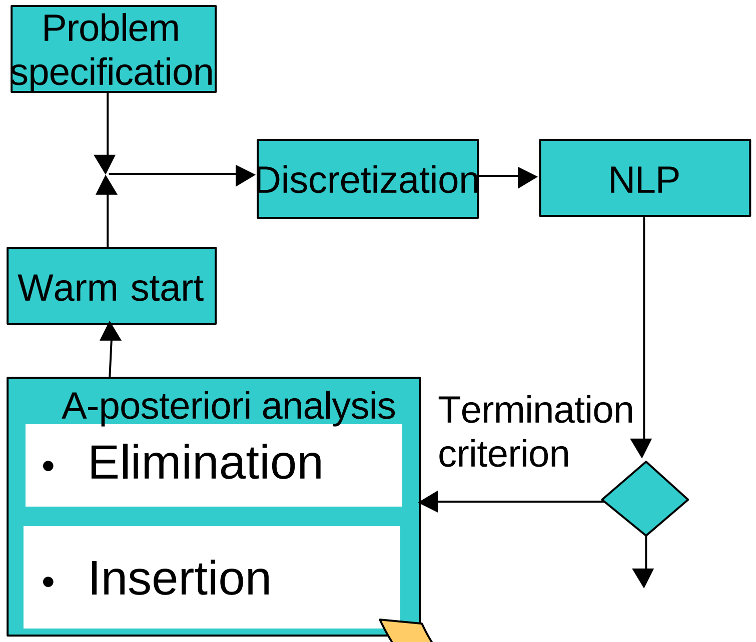
Minimize number of parameterization functions in u^* such that:

$$\frac{\|u^* - \tilde{u}^*\|}{\|u^*\|} \leq e$$



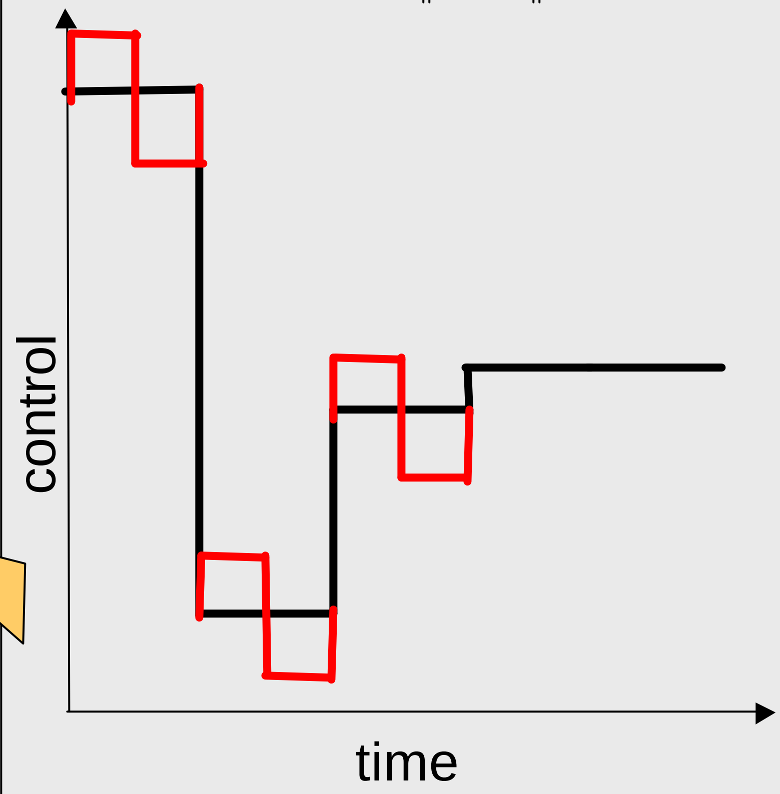
Approximation: Norm equivalence $\|u\|_{L_2} \sim \|d\|_{l_2}$ ➡ Discarding small $d_{j,k}$ causes only small changes in approximate representation

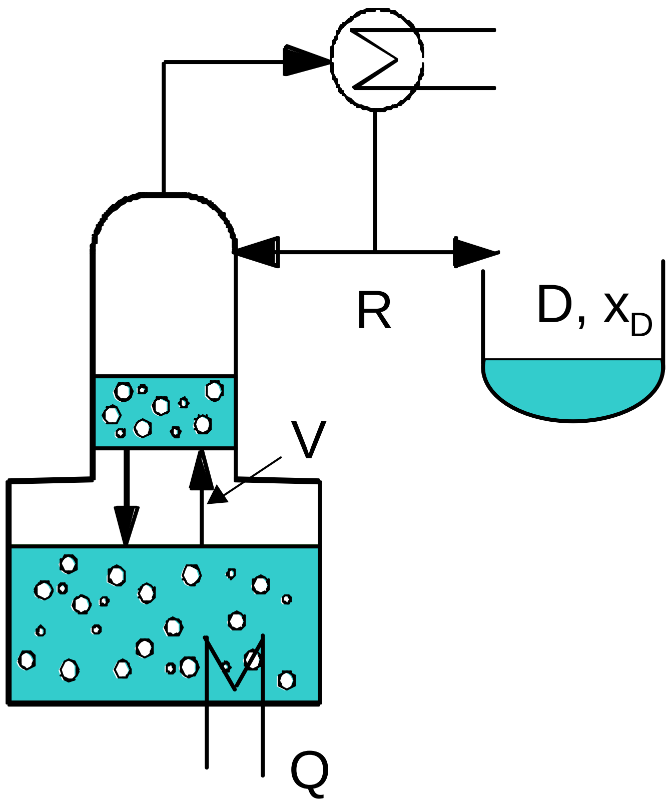
Insertion of parameterization functions



Analysis of wavelet coefficients of control variables

Introduction of new parameterization functions where $\|d_{j,k}\| \geq h$





Objective:

Minimize energy demand with given:

- Fixed batch time
- Amount of distillate $D \geq 6.0$ kmol
- Product purity $x_D \geq 0.46$

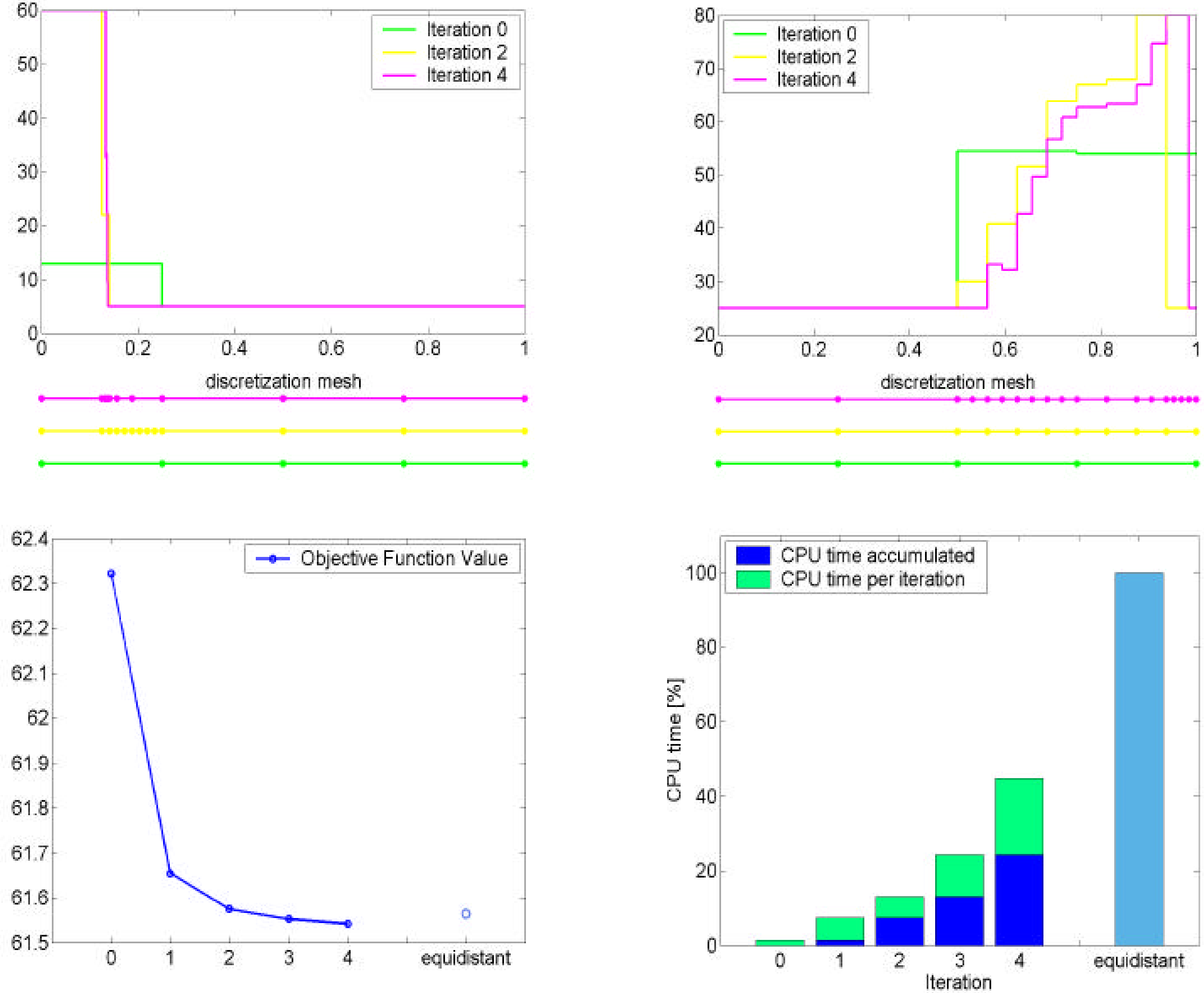
Controls:

- Reflux ratio R(t)
- Vapor rate V(t)

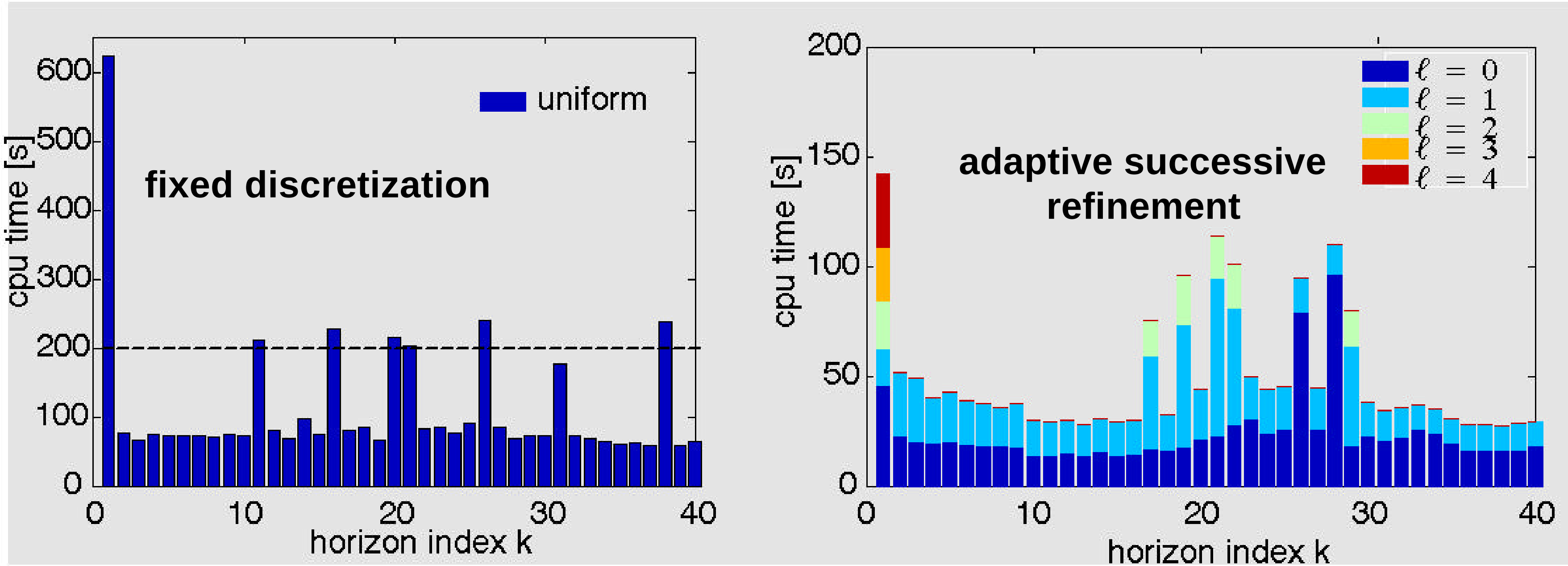
Dynamic model:

- 10 theoretical trays
- gPROMS model contains 418 DAEs
  (63 differential equations)

Results: Batch reactive distillation



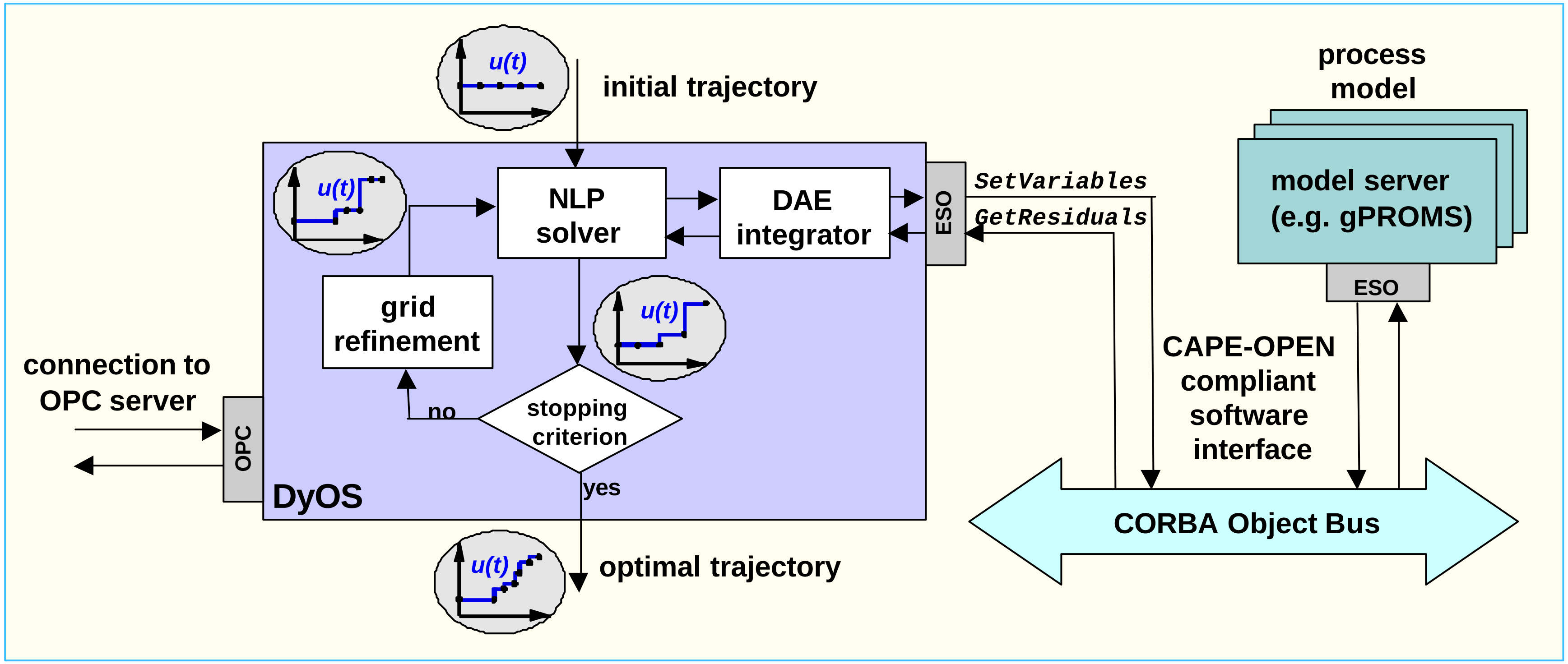
Application in direct approach setting



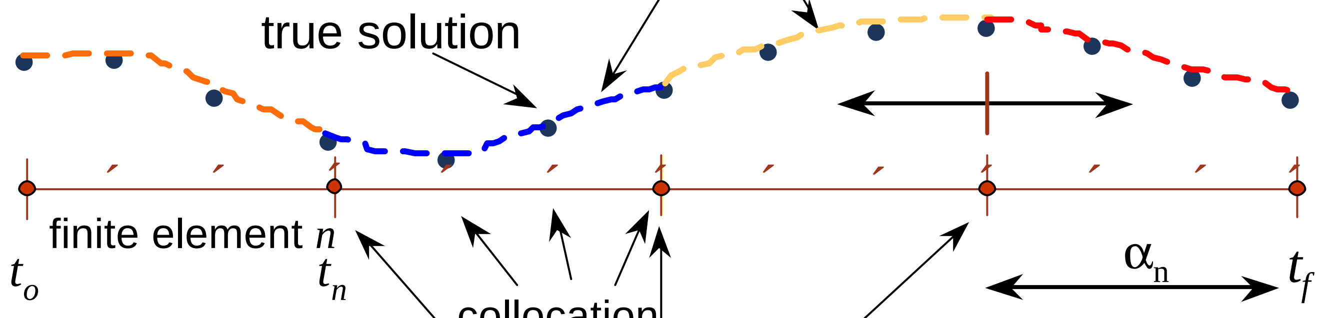
Adaptive approach (Binder et al., 2000):

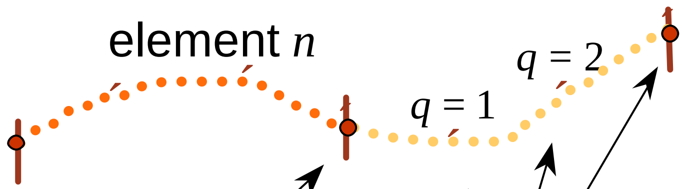
- numerical lower for the adaptive refinement approach
- intermediate solutions are available
  - back-up values in real-time environment
  - direct employment on the process at early time

Dynamic optimization software DyOS (LPT)



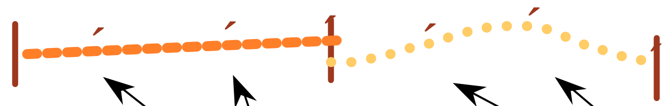
Collocation on finite elements





$$z(t) = z_{n-1} + (t - t_{n-1}) \sum_{q=1}^{k} g^q(t) \frac{dz^q}{dt}_n$$

differential variables
continuous



$$y(t) = \sum_{q=1}^{k} g^q(t) y_n^q \quad u(t) = \sum_{q=1}^{k} g^q(t) u_n^q$$

algebraic and control variables
discontinuous

Simultaneous approach → collocation (II)

Conversion into NLP problem yields

$$\min\ y\left(z_i, y_{i,q}, u_{i,q}, p, t_f\right)$$

$$\text{s.t.}\ \left(\frac{dz}{dt}\right)_{i,j} = F\left(z_{i\text{-}1}, \frac{dz}{dt}_{i,j}, z_i, y_{i,j}, u_{i,j}, p\right)$$

$$0 = G\left(z_{i\text{-}1}, \frac{dz}{dt}_{i,j}, z_i, y_{i,j}, u_{i,j}, p\right)$$

$$z_i = f\left(\frac{dz}{dt}_{i-1,j}, z_{i-1}\right)_i$$

$$z_0^o = z(0)$$

$$z_i^l \le z_i \le z_i^u$$

$$y_{i,j}^l \le y_{i,j} \le y_{i,j}^u$$

$$u_{i,j}^l \le u_{i,j} \le u_{i,j}^u$$

$$p^l \le p \le p^u$$

large-scale NLP problem

$$\min_{x \in R^n} f(x)$$

$$\text{s.t}\quad c(x) = 0$$

$$x^L \le x \le x^u$$

Requires specially tailored solution techniques:
- advanced interior-point solver
- filter-line search techniques

(implemented as IPOPT, Biegler et al., 2001)

Barrier function formulation

original formulation

$$\min_{x \in R^n} f(x)$$
$$\text{s.t} \quad c(x) = 0$$
$$x \geq 0$$

can be generalized for

$$a \leq x \leq b$$

barrier approach

$$\min_{x \in R^n} j_m(x) = f(x) - m \sum_{i=1}^{n} \ln s_i$$
$$\text{s.t} \quad c(x) = 0$$
$$s - x = 0$$

$$\Rightarrow \text{as } m \to 0, \quad x^*(m) \to x^*$$

Solution of the barrier problem (I)

$\Rightarrow$ Newton Directions (KKT System)

$$
\begin{aligned}
\nabla f(x) + A(x)\,l - v &= 0 \\
SVe - me &= 0 \\
c(x) &= 0 \\
s - x &= 0
\end{aligned}
$$

$\Rightarrow$ solve primal-dual version

$$
\begin{bmatrix} H & 0 & A & -I \\ 0 & S^{-1}V & 0 & I \\ A^T & 0 & 0 & 0 \\ -I & I & 0 & 0 \end{bmatrix}
\begin{bmatrix} d_x \\ d_s \\ d_l \\ d_v \end{bmatrix}
= -\begin{bmatrix} \nabla f + A\,l - v \\ v - m\,S^{-1}e \\ c \\ 0 \end{bmatrix}
$$

Solution of the barrier problem (II)

⇒ Range Space Step

$$A^T d_x + c = 0$$

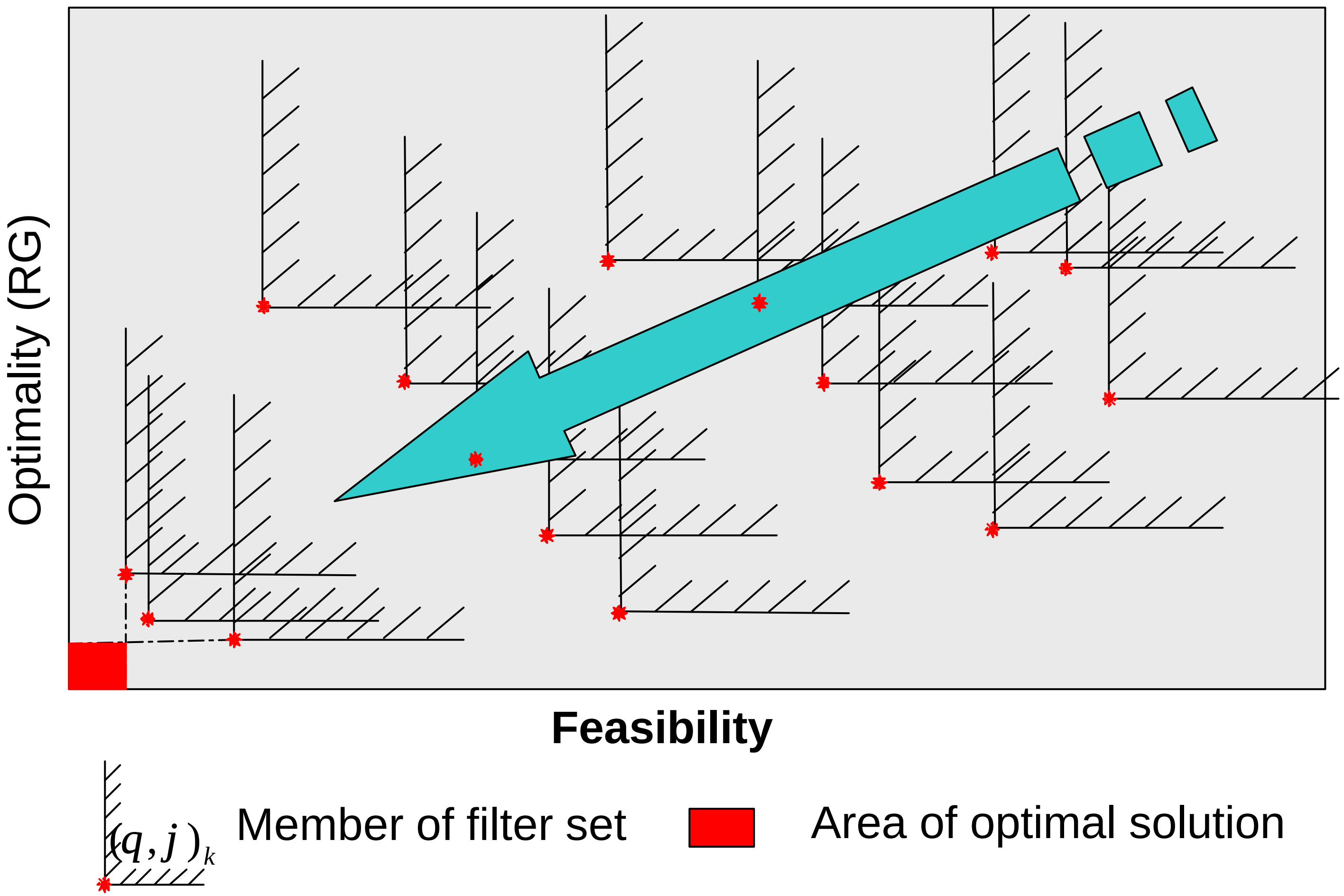
$$\Rightarrow d_R = -C^{-1} c$$

⇒ Null Space Step (reduced QP)

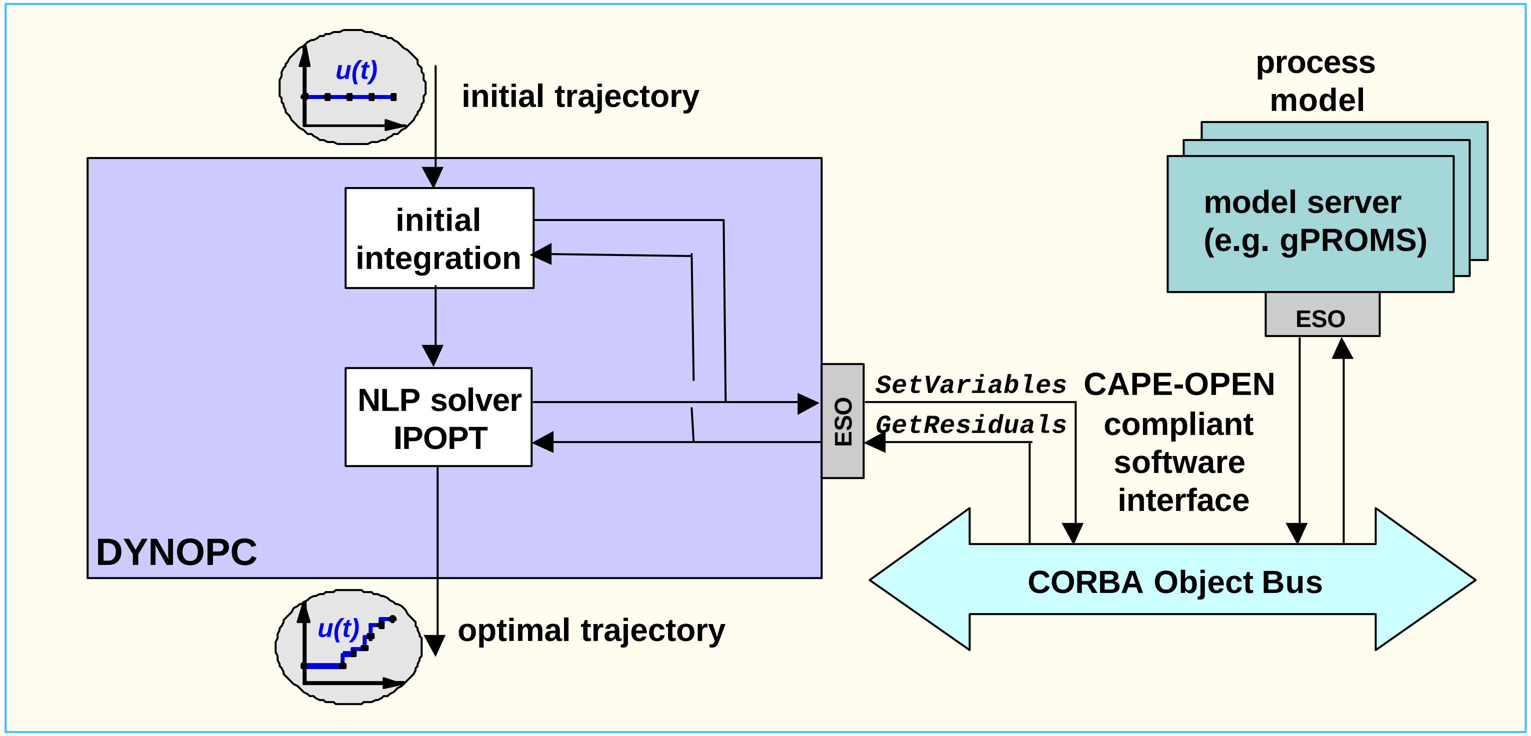
$$\min_{d_Q} \left(Q^T \nabla j_m + Q^T (H+\Sigma) R d_R\right)^T d_Q + \tfrac{1}{2} d_Q^T Q^T (H+\Sigma) Q d_Q$$

$$d_Q = -\left[\underbrace{Q^T (H+\Sigma) Q}_{\textbf{reduced Hessian}}\right]^{-1} \left(Q^T \nabla j_m + \underbrace{Q^T (H+\Sigma) R d_R}_{\textbf{cross term}}\right)$$

Illustration of filter concept



Dynamic optimization software DYNOPC/IPOPT (CMU)



Comparison of approaches

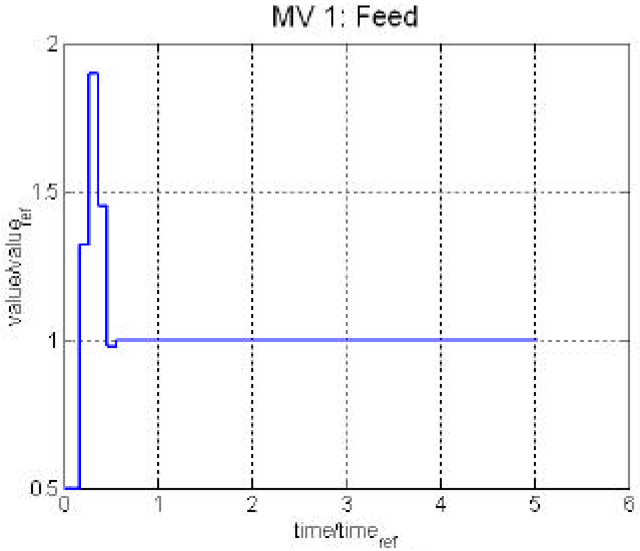
	Sequential approach	Simultaneous approach
size of NLP	small	large
DAE model fulfilled in each step?	yes	no
initial guess required for...	controls	states and controls

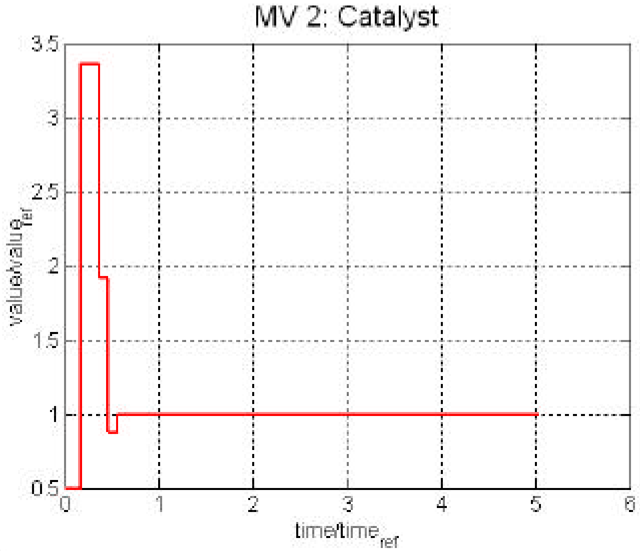
Experience from solving INCOOP benchmark problems

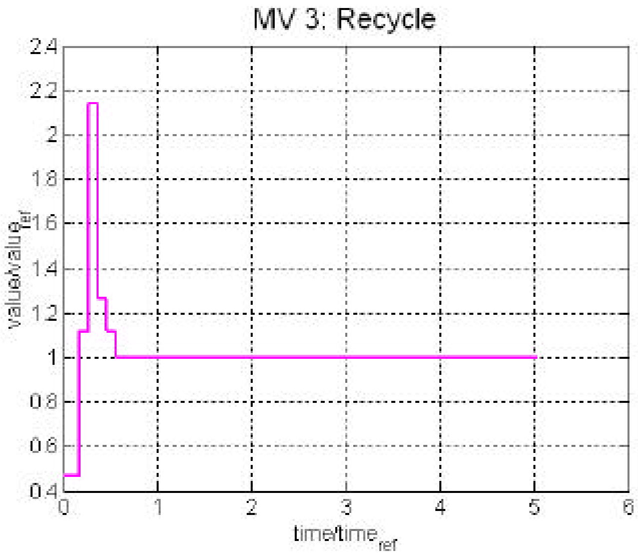
- sequential approach more robust and capable of handling bigger problems
- simultaneous approach can be faster with good initial guess, but more sensitive to initial guess
- accuracy problems with simultaneous approach for stiff problems

  (error controlled integration vs. fixed-grid collocation)

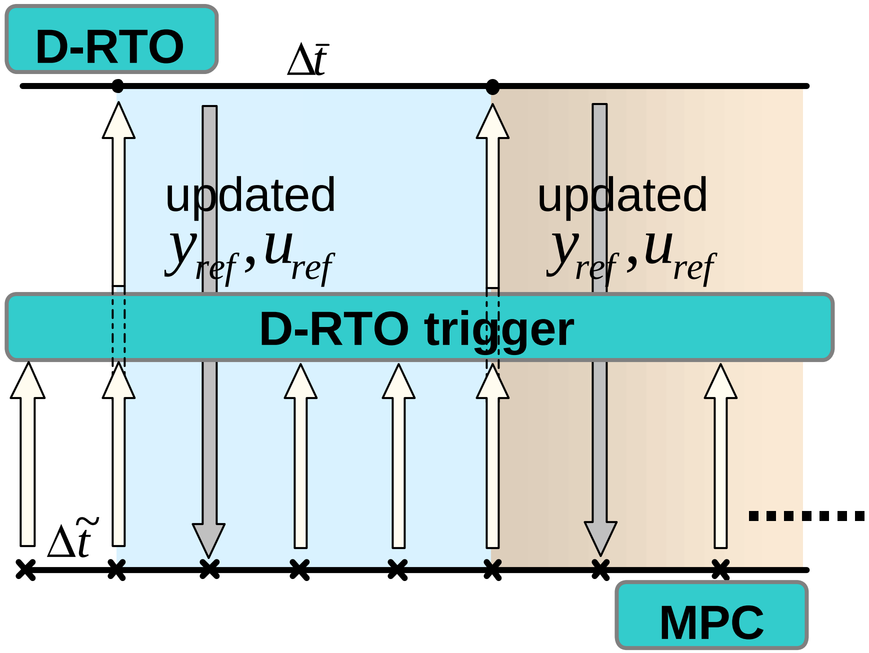
Results for Bayer Benchmark Problem







Interplay between D-RTO and MPC



- Soft constraints can be moved from MPC to D-RTO
- Longer time horizon for D-RTO to ensure feasibility
- D-RTO trigger for a possible re-optimization
- Delta-mode MPC computes updates to the control profiles for tracking the process in the strict operation envelope: rejects fast frequency process disturbances

Lagrange function sensitivities w.r.t. all estimated disturbances

$$\textit{compute} \quad S_j = dL_j \big/ d\hat{d}_j \Big|_{\tilde{t}_{0j}} ;$$

$$L_j = \overline{\Phi}(u_i^{ref}, \hat{d}_j) + m_i^T h(u_i^{ref}, \hat{d}_j)$$

- One sensitivity integration of process model at each sampling time $\bar{t}_{0i}$ using previous D-RTO results (and active constraint set) at $\tilde{t}_{0j}$ is required
- Compute change in sensitivities ($\Delta S_j = S_j - S_i$) and Lagrange function ($\Delta L_j = L_j - L_i$) can be then calculated

D-RTO trigger (II)

Optimal solution sensitivities w.r.t. all estimated disturbances

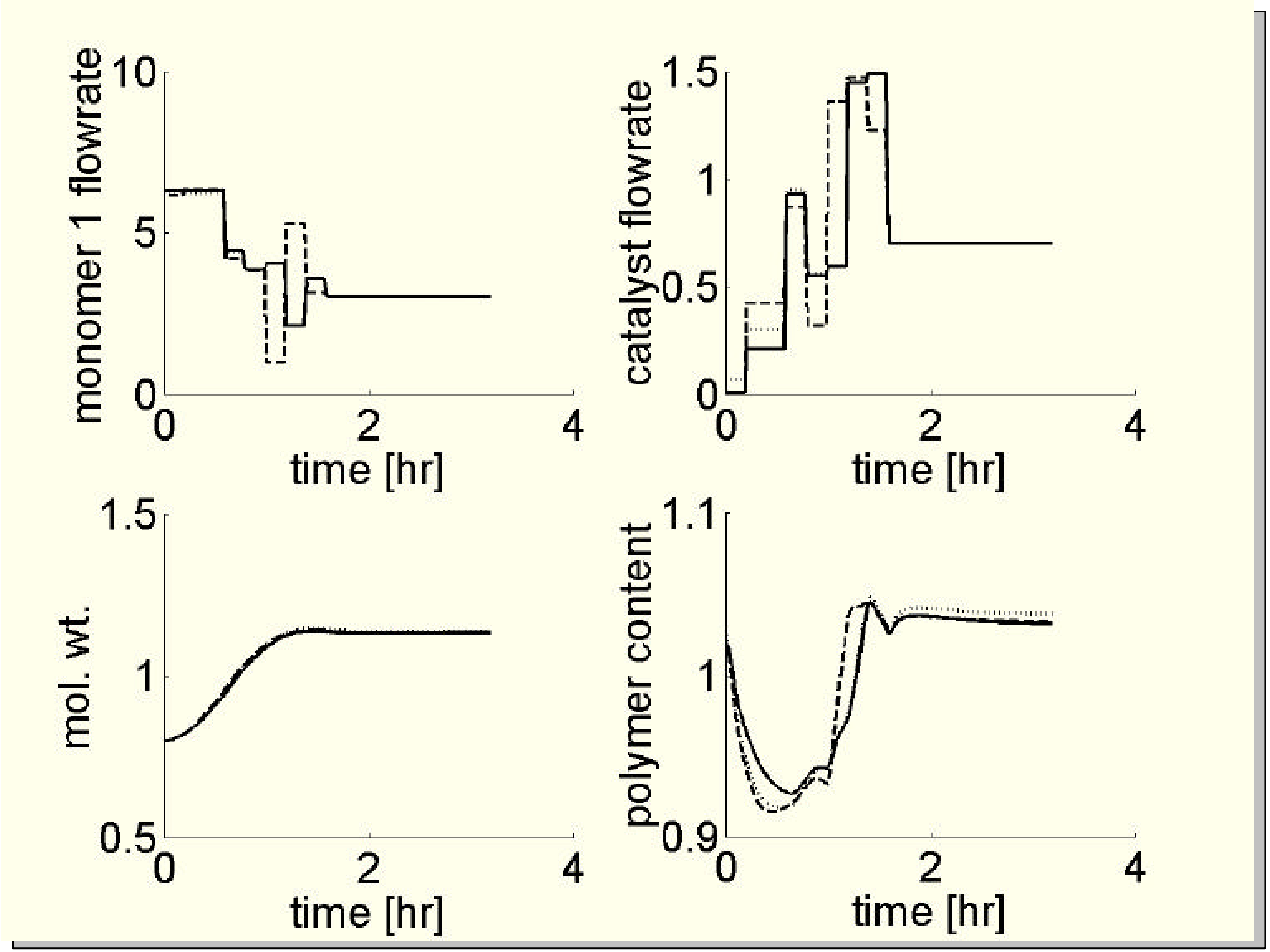
compute $U_j = du_j^{ref} \big/ d\hat{d}_j \Big|_{\tilde{t}_{0\,j}}$

and changed active constraint set

- Solution to QP problem:
  - using second order information (Hessian of Lagrange function)
    $\Rightarrow$ *optimal sensitivities*
  - using first order information
    $\Rightarrow$ *feasible only sensitivities*
- updates as $u_j^{ref} = u_i^{ref} + U_i^T(\hat{d}_j - \bar{d}_i)$

- If ΔS_j and ΔL_j are larger than a threshold value S_{th} and changed active constraint set is predicted, a re-optimization should be done
- Else linear updates based on optimal solution sensitivities U_j are sufficient

results with re-optimization and feasible updates

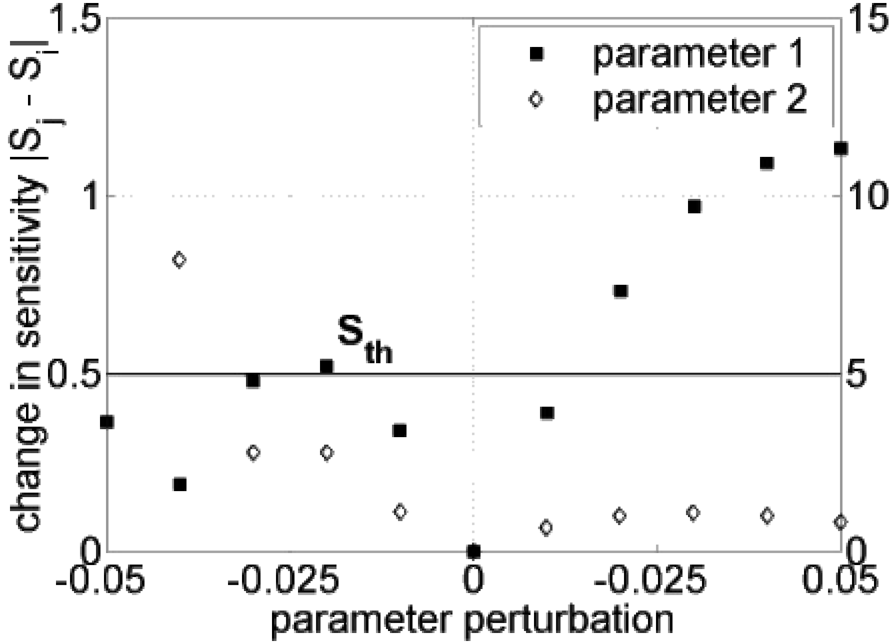


re-optimization results (2)

- Reaction parameters randomly perturbed between their bounds
- Re-optimization done only when necessary

  ⇒ steer to desired grade

- Feasible updates only possible up to +4% change in parameter 1



- D-RTO problem needs to be solved only necessary
- the hybrid integrated D-RTO and control with embedded sensitivity analysis is well suited for large-scale industrial process operation

Off-line dynamic optimization:

Today already many numerical and software techniques available for efficient and convenient solution of such problems

but...

dynamic optimization still not state-of-the-art (especially not in industry):

- Though pure solution time for solving one mathematical problem only in the order of *hours,*
- overall engineering time to solve the real application problem in the order of *weeks* or *months.*
- Problems: Modeling issues, problem formulation, convergence problems, ...

It is still not “pushing the button”.

Future perspectives

Experience from INCOOP: for large-scale process models application of dynamic optimization in real-time still time-critical

Dynamic real-time optimization

- further enhance sequential approach dynamic optimization
- more elaborate adaptation strategies
- interaction NLP solver / integrator
- adapt integration accuracy
- incorporate second order information

Integration of control and optimization

- further exploit re-optimization features
- apply adaptation strategies in real-time context
- gain speed by feasible-first optimizations
- interlink MPC and D-RTO by shifting the prediction to the D-RTO level