Optimal Operation of Dynamic Processes

Miroslav Fikar

Institute of Information Engineering, Automation, and Mathematics
Faculty of Chemical and Food Technology
Slovak University of Technology, Bratislava, Slovakia
http://www.kirp.chtf.stuba.sk/~fikar

SSCHE 2010

Motivation

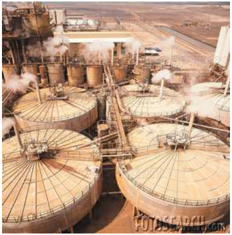
Optimisation is a natural choice:

- To reduce production costs and improve product quality
- To respect environmental regulations
- To meet safety requirements

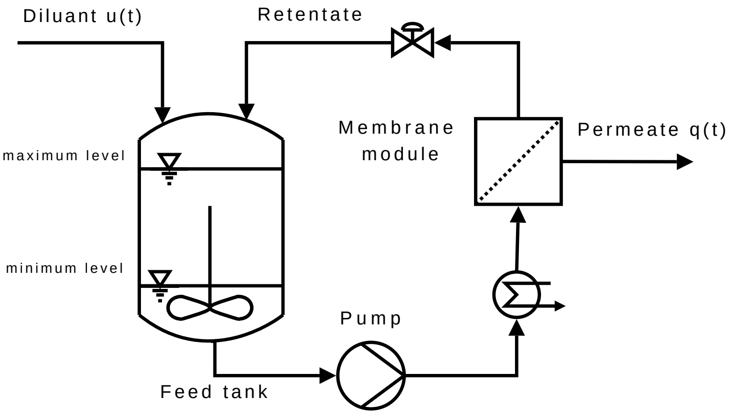
Inherently transient processes require dynamic optimisation (e.g., batch)

Questions:

- What is the best possible control trajectory?
- What are properties of optimal control?
- What is the setpoint trajectory for lower level control?



Motivation Example



Overview

1. Dynamic Optimisation of Processes
   - Problem Definition
   - Classification of Methods
2. Numerical Methods
   - Total Parametrisation
   - Control Vector Parametrisation
3. Process Control Examples
   - Diafiltration Process
   - Waste-water Treatment Plant
   - Depropanizer
4. Conclusions

Overview

Problem Definition

For a given process described by a set of ODEs

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t)), \quad \boldsymbol{x}(0) = \boldsymbol{x}_0$$

find such **control** $\boldsymbol{u}(t)$ that **minimises** cost function J_0

$$\min_{t_P, \boldsymbol{u}(t)} J_0 = G(t_P, \boldsymbol{x}(t_P), \boldsymbol{u}(t_P)) + \int_{t_0}^{t_P} F(\boldsymbol{x}(t), \boldsymbol{u}(t)) \mathrm{d}t$$

subject to equality and inequality **constraints**

$$J_j = 0, \quad j = \overline{1, k_e}$$
$$J_j \geq 0, \quad j = k_e + \overline{1, k_i}$$

with **bounds** on optimised variables

$$\Delta \boldsymbol{t}_i \in [\Delta \boldsymbol{t}_i^{\min}, \Delta \boldsymbol{t}_i^{\max}], \quad \boldsymbol{u}(t) \in [\boldsymbol{u}^{\min}, \boldsymbol{u}^{\max}]$$

Overview

Classification of Methods

- **Analytical Methods**
  - Dynamic programming (DP)
  - Pontryagin's maximum principle (PMP)
  - Calculus of variations (CV)

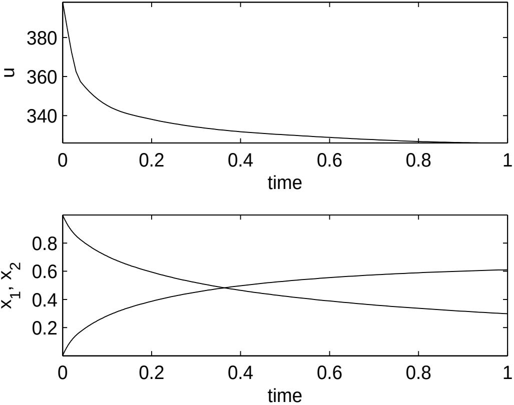
- **Numerical Methods**
  - Continuous states and controls:
    - Boundary condition iteration (BCI)
    - Control vector iteration (CVI)
  - Continuous states, approximated controls:
    - Sequential method – control vector parametrisation (CVP)
  - Approximated states and controls:
    - Simultaneous method (total parametrisation)

Classification of Methods

- **Analytical Methods**
  - Dynamic programming (DP)
  - Pontryagin's maximum principle (PMP)
  - Calculus of variations (CV)
- **Numerical Methods**
  - Continuous states and controls:
    - Boundary condition iteration (BCI)
    - Control vector iteration (CVI)
  - Continuous states, approximated controls:
    - Sequential method – control vector parametrisation (CVP)
  - Approximated states and controls:
    - Simultaneous method (total parametrisation)

Overview

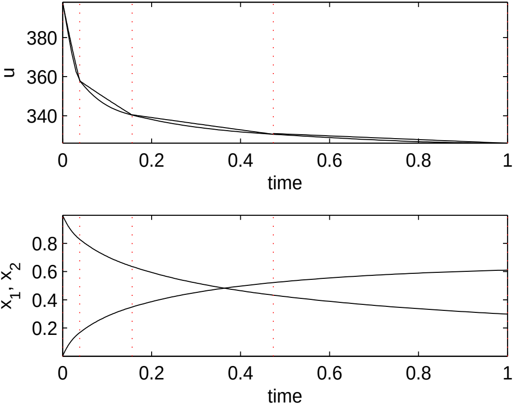
Total Parametrisation on Finite Elements



1. Time discretisation
   - choose Δt_i
2. Total parametrisation
   - $u(t) = \sum_{j=1}^{ncolu} u_{ij}\psi_j(t)$
   - $x(t) = \sum_{j=0}^{ncolx} x_{ij}\phi_j(t)$
3. Optimised variables

$$z = [\Delta t_1, \ldots, \Delta t_{ni}, \mathbf{u}_{11}, \ldots, \mathbf{u}_{ij}, \mathbf{x}_{10}, \ldots, \mathbf{x}_{ij}]^T$$

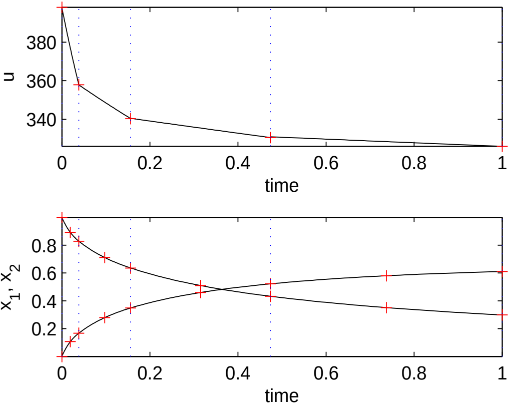
Total Parametrisation on Finite Elements



1. Time discretisation
   - choose Δt_i
2. Total parametrisation
   - $u(t) = \sum_{j=1}^{ncolu} u_{ij}\psi_j(t)$
   - $x(t) = \sum_{j=0}^{ncolx} x_{ij}\phi_j(t)$
3. Optimised variables

$$\boldsymbol{z} = [\Delta t_1, \dots, \Delta t_{ni}, \boldsymbol{u}_{11}, \dots, \boldsymbol{u}_{ij}, \boldsymbol{x}_{10}, \dots, \boldsymbol{x}_{ij}]^T$$

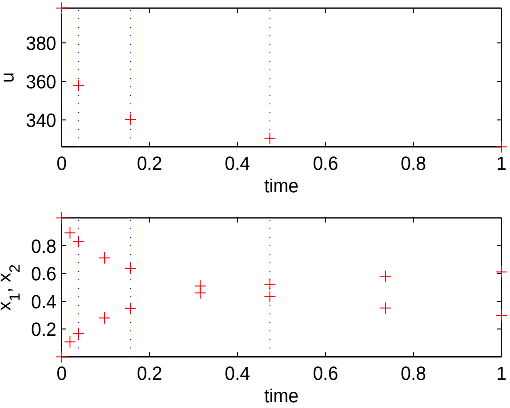
Total Parametrisation on Finite Elements



1. Time discretisation
   - choose Δt_i
2. Total parametrisation
   - $u(t) = \sum_{j=1}^{ncolu} u_{ij}\psi_j(t)$
   - $x(t) = \sum_{j=0}^{ncolx} x_{ij}\phi_j(t)$
3. Optimised variables

$$\boldsymbol{z} = [\Delta t_1, \ldots, \Delta t_{ni}, \boldsymbol{u}_{11}, \ldots, \boldsymbol{u}_{ij}, \boldsymbol{x}_{10}, \ldots, \boldsymbol{x}_{ij}]^T$$

Total Parametrisation on Finite Elements



1. Time discretisation
   - choose Δt_i
2. Total parametrisation
   - $u(t) = \sum_{j=1}^{ncolu} u_{ij}\psi_j(t)$
   - $x(t) = \sum_{j=0}^{ncolx} x_{ij}\phi_j(t)$
3. Optimised variables

$$
\begin{aligned}
\boldsymbol{z} \;=\; & [\Delta t_1, \ldots, \Delta t_{ni}, \\
& \boldsymbol{u}_{11}, \ldots, \boldsymbol{u}_{ij}, \\
& \boldsymbol{x}_{10}, \ldots, \boldsymbol{x}_{ij}]^T
\end{aligned}
$$

Properties of Total Parametrisation

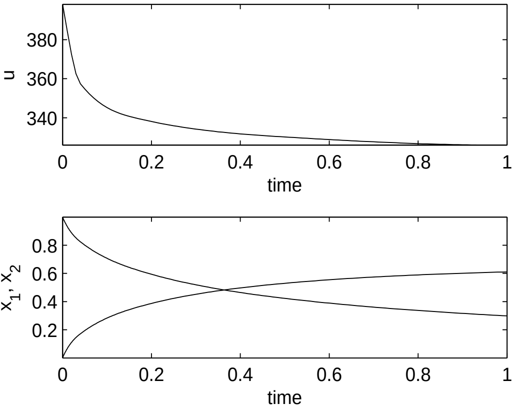
- + No need to integrate process differential equations

$$\dot{x} = f(x,u) \rightarrow \sum x_{ij}\dot{\phi}_j(t) = f(x_{ij},u_{ij})$$

- + Suitable for problems with state path constraints, hybrid systems
- – Large-scale NLP formulations
- – Tailor-made SQP solver is preferable that uses inherent structure of NLP formulation
- – Infeasible type of method

Overview

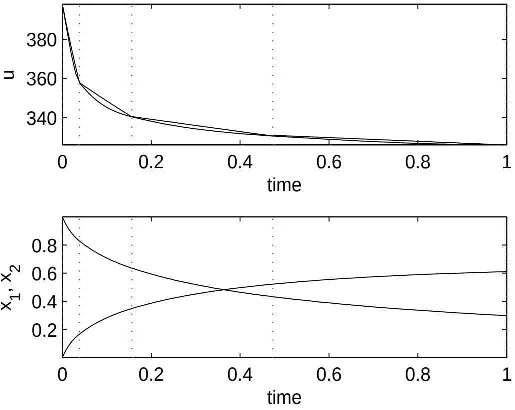
Control Vector Parametrisation – CVP



1. Time discretisation
   - choose Δt_i
2. Parametrisation of control
   - $u(t) = \sum_{j=1}^{ncolu} u_{ij}\psi_j(t)$
   - $\dot{x}(t) = f(x(t), u(t))$
3. Optimised variables

$$\boldsymbol{z} = [\Delta t_1, \ldots, \Delta t_{ni}, \boldsymbol{u}_{11}, \ldots, \boldsymbol{u}_{ij}]$$

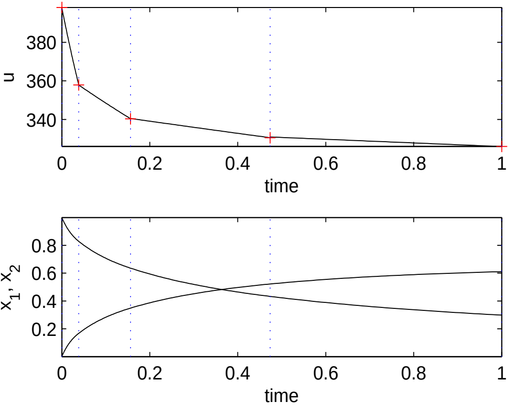
Control Vector Parametrisation – CVP



1. Time discretisation
   - choose Δt_i
2. Parametrisation of control
   - $u(t) = \sum_{j=1}^{ncolu} u_{ij}\psi_j(t)$
   - $\dot{x}(t) = f(x(t), u(t))$
3. Optimised variables

$$\boldsymbol{z} = [\Delta t_1, \ldots, \Delta t_{ni}, \boldsymbol{u}_{11}, \ldots, \boldsymbol{u}_{ij}]$$

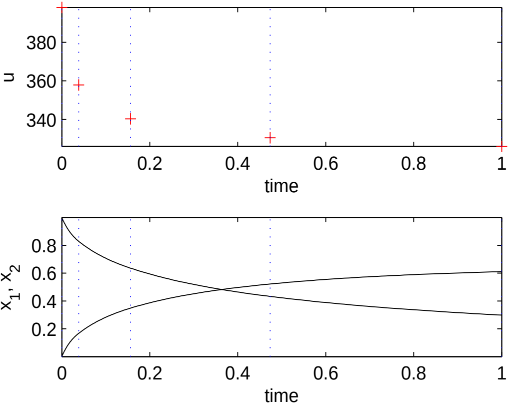
Control Vector Parametrisation – CVP



1. Time discretisation
   - choose Δt_i
2. Parametrisation of control
   - $u(t) = \sum_{j=1}^{ncolu} u_{ij}\psi_j(t)$
   - $\dot{x}(t) = f(x(t), u(t))$
3. Optimised variables

$$z = [\Delta t_1, \ldots, \Delta t_{ni}, u_{11}, \ldots, u_{ij}]$$

Control Vector Parametrisation – CVP



1. Time discretisation
   - choose Δt_i
2. Parametrisation of control
   - $u(t) = \sum_{j=1}^{ncolu} u_{ij}\psi_j(t)$
   - $\dot{x}(t) = f(x(t), u(t))$
3. Optimised variables

$$\boldsymbol{z} = [\Delta t_1, \ldots, \Delta t_{ni}, \boldsymbol{u}_{11}, \ldots, \boldsymbol{u}_{ij}]$$

Properties of Control Vector Parametrisation

- \+ Smaller size of NLP formulations – suitable for problems with larger number of differential equations
- \+ Method of gradient calculations depends on problem formulation
- \+ Feasible type of method – suitable for model predictive control
- – Much time spent in integration of process differential equations
- – More difficult to solve for problems with state path constraints, hybrid systems

Overview

Features:

- Steps:
  - Problem transformation of **dynamic** optimisation to **static** optimisation (**NLP** – nonlinear programming).
  - NLP is solvable using a suitable **gradient method** and **SQP** algorithm.
- ODE integration:
  - **forward** – process to be optimised
  - **backward** – adjoint process
- Gradient method

$$\frac{\partial J}{\partial t_i} = 0, \quad \frac{\partial J}{\partial \boldsymbol{u}_i} = 0$$

Adjoint Variables

Define adjoint system of ODEs:

$$\dot{\boldsymbol{\lambda}} = -\frac{\partial H}{\partial \boldsymbol{x}}, \quad \boldsymbol{\lambda}(t_P) = \left.\frac{\partial G}{\partial \boldsymbol{x}}\right|_{t_P}$$

with Hamiltonian

$$H(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\lambda}) = F(\boldsymbol{x}, \boldsymbol{u}) + \boldsymbol{\lambda}^T \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})$$

Variation of J_j w.r.t. **optimised** variables $(0 \leq j \leq k)$

$$\delta J = \sum_{i=1}^{P-1} \left[H(t_i^-) - H(t_i^+) + \frac{\partial G}{\partial t_i} \right] \delta t_i + \left[H(t_P^-) + \frac{\partial G}{\partial t_P} \right] \delta t_P + \sum_{i=1}^{P} \left[\frac{\partial G}{\partial \boldsymbol{u}_i^T} + \int_{t_{i-1}^+}^{t_i^-} \frac{\partial H}{\partial \boldsymbol{u}_i^T} \mathrm{dt} \right] \delta \boldsymbol{u}_i$$

Bracketed expressions are **gradients** w.r.t. **optimised variables** for **NLP** $(t_i, \boldsymbol{u}_i)$.

Adjoint Variables

Define adjoint system of ODEs:

$$\dot{\boldsymbol{\lambda}} = -\frac{\partial H}{\partial \boldsymbol{x}}, \quad \boldsymbol{\lambda}(t_P) = \left.\frac{\partial G}{\partial \boldsymbol{x}}\right|_{t_P}$$

with Hamiltonian

$$H(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\lambda}) = F(\boldsymbol{x}, \boldsymbol{u}) + \boldsymbol{\lambda}^T \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})$$

Variation of J_j w.r.t. **optimised** variables ($0 \leq j \leq k$)

$$\delta J = \sum_{i=1}^{P-1} \left[H(t_i^-) - H(t_i^+) + \frac{\partial G}{\partial t_i} \right] \delta t_i + \left[H(t_P^-) + \frac{\partial G}{\partial t_P} \right] \delta t_P + \sum_{i=1}^{P} \left[\frac{\partial G}{\partial \boldsymbol{u}_i^T} + \int_{t_{i-1}^+}^{t_i^-} \frac{\partial H}{\partial \boldsymbol{u}_i^T} \mathrm{d}t \right] \delta \boldsymbol{u}_i$$

Bracketed expressions are **gradients** w.r.t. **optimised variables** for **NLP** $(t_i, \boldsymbol{u}_i)$.

Adjoint Variables

Define adjoint system of ODEs:

$$\dot{\boldsymbol{\lambda}} = -\frac{\partial H}{\partial \boldsymbol{x}}, \quad \boldsymbol{\lambda}(t_P) = \left.\frac{\partial G}{\partial \boldsymbol{x}}\right|_{t_P}$$

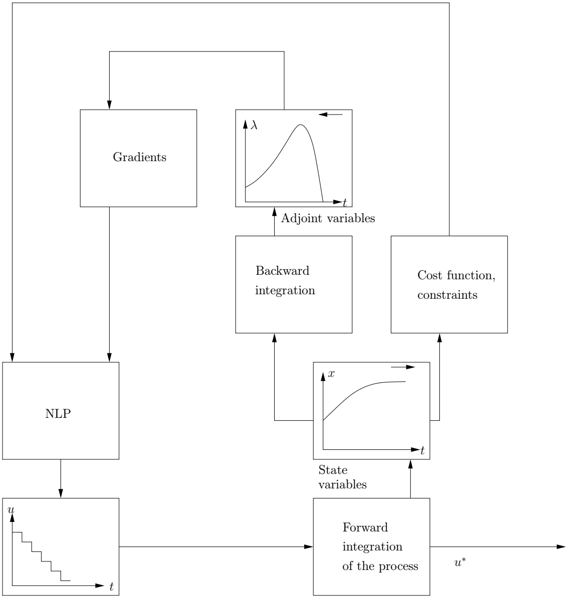
with Hamiltonian

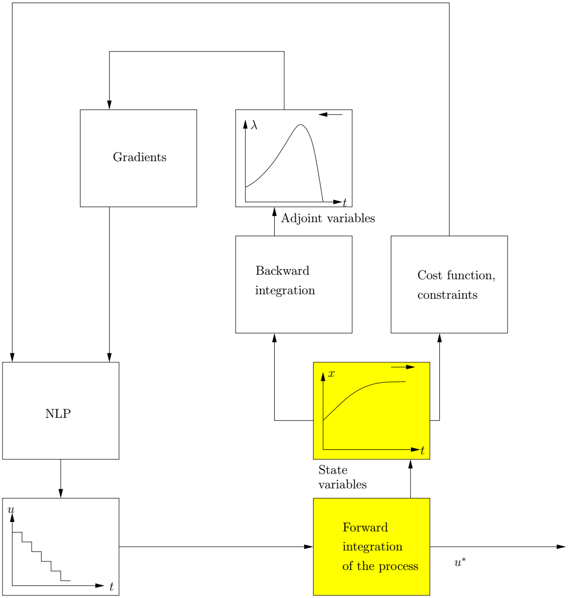
$$H(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\lambda}) = F(\boldsymbol{x}, \boldsymbol{u}) + \boldsymbol{\lambda}^T \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})$$

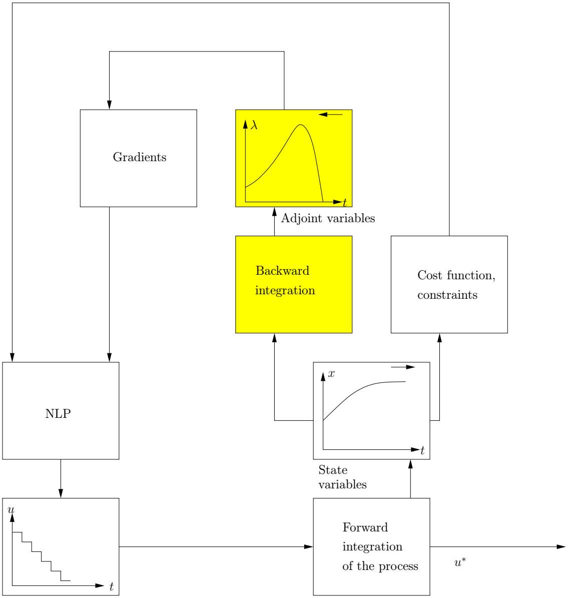
Variation of J_j w.r.t. **optimised** variables ($0 \leq j \leq k$)

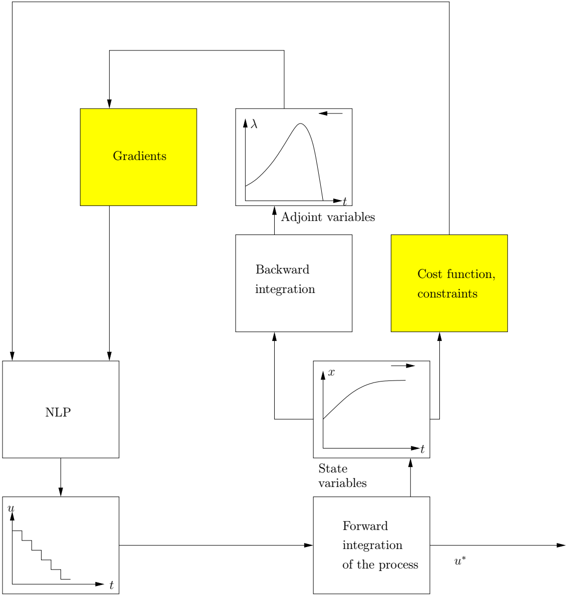
$$\delta J = \sum_{i=1}^{P-1} \left[H(t_i^-) - H(t_i^+) + \frac{\partial G}{\partial t_i} \right] \delta t_i + \left[H(t_P^-) + \frac{\partial G}{\partial t_P} \right] \delta t_P + \sum_{i=1}^{P} \left[\frac{\partial G}{\partial \boldsymbol{u}_i^T} + \int_{t_{i-1}^+}^{t_i^-} \frac{\partial H}{\partial \boldsymbol{u}_i^T} \mathrm{dt} \right] \delta \boldsymbol{u}_i$$

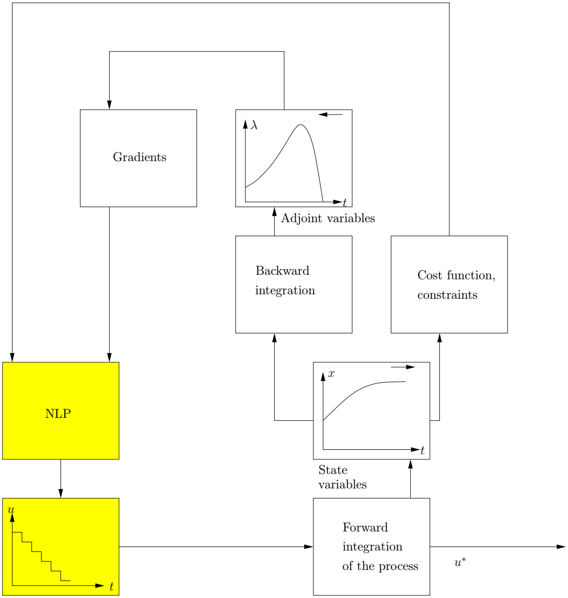
Bracketed expressions are **gradients** w.r.t. **optimised variables** for **NLP** ($t_i, \boldsymbol{u}_i$).

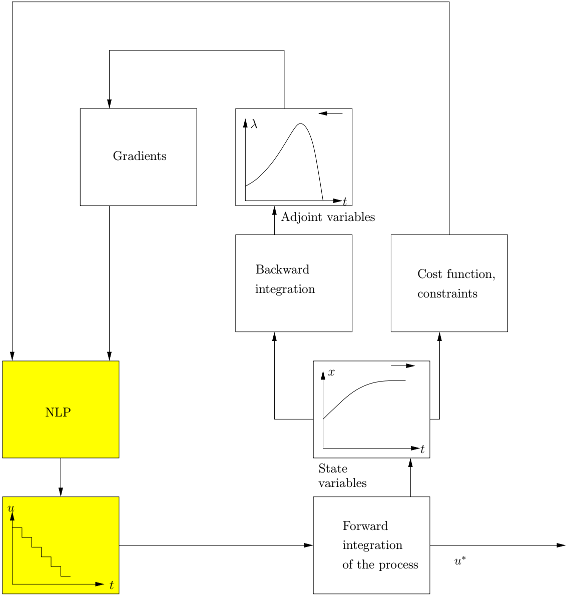

Gradients
λ
t
Adjoint variables
Backward integration
Cost function, constraints
NLP
x
t
State variables
u
t
Forward integration of the process
u^*


Gradients
λ
t
Adjoint variables
Backward integration
Cost function, constraints
NLP
x
t
State variables
u
t
Forward integration of the process
u^*


Gradients
λ
t
Adjoint variables
Backward integration
Cost function, constraints
NLP
x
t
State variables
u
t
Forward integration of the process
u^*

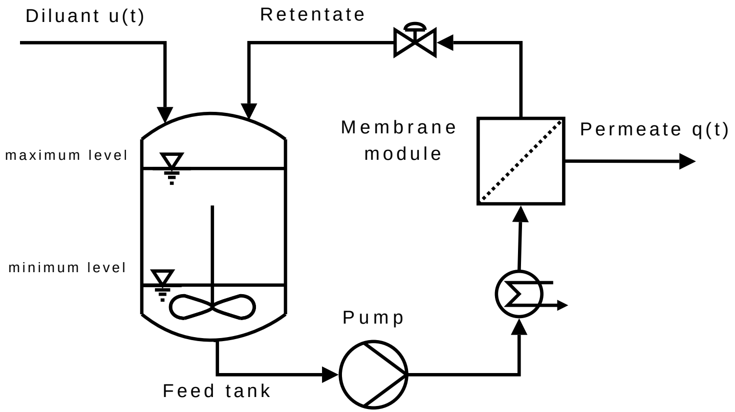

Gradients
λ
t
Adjoint variables
Backward integration
Cost function, constraints
NLP
x
t
State variables
u
t
Forward integration of the process
u*


Gradients
λ
t
Adjoint variables
Backward integration
Cost function, constraints
NLP
x
t
State variables
u
t
Forward integration of the process
u*


Gradients
λ
t
Adjoint variables
Backward integration
Cost function, constraints
NLP
x
t
State variables
u
t
Forward integration of the process
u^*

Overview

Process


Diluant u(t)
Retentate
Membrane module
Permeate q(t)
maximum level
minimum level
Pump
Feed tank

Process Model

Three differential equations:

$$\frac{\mathrm{d}V(t)}{\mathrm{d}t} = u(t) - q(t), \; V(0) = V^0$$

$$V(t)\frac{\mathrm{d}c_i(t)}{\mathrm{d}t} = c_i(t)\left[q(t)\mathcal{R}_i(t) - u(t)\right], \; c_i(0) = c_i^0, \quad i = 1, 2$$

Experimental data:

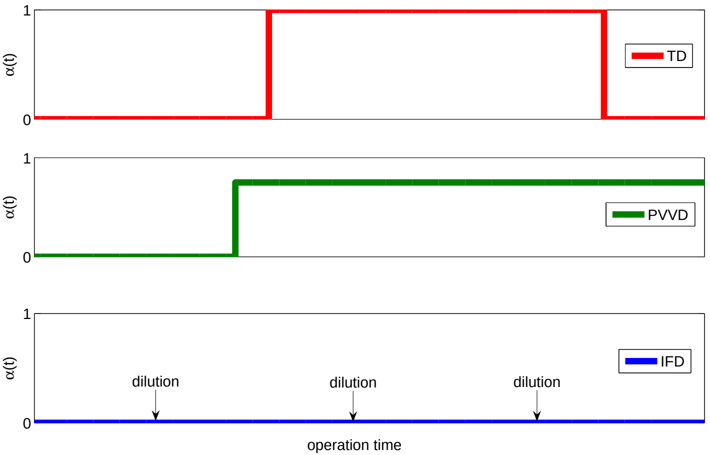
$$q = S_1(c_2)\mathrm{e}^{S_2(c_2)c_1}$$

$$\mathcal{R}_1 = (z_1c_2 + z_2)c_1 + (z_3c_2 + z_4)$$

$$\mathcal{R}_2 = W_1(c_2)\mathrm{e}^{W_2(c_2)c_1}$$

Control variable can be $u(t)$ or $\alpha(t) = u(t)/q(t)$

Traditional Operation



Optimisation problem

Cost function:

$$\min_{u(t)} J = c_2(t_f), \quad t_f = 6$$

Constraints:

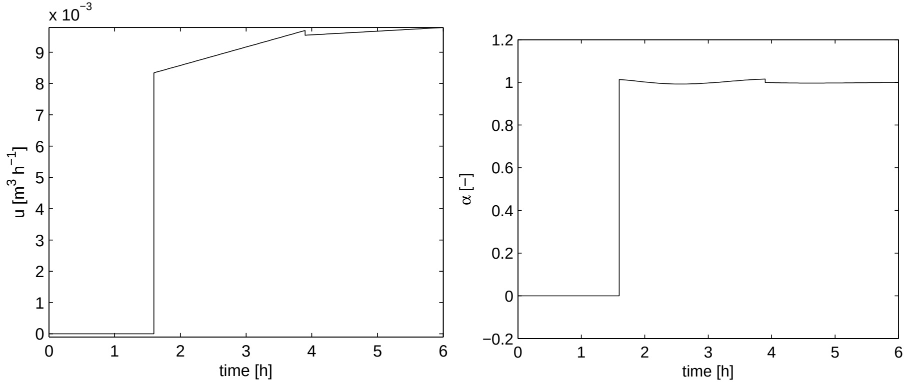
$$V^0 = 0.03\,\text{m}^3, \qquad V^f = 0.01\,\text{m}^3, \qquad V(t) \in [V^0, V^f]$$

$$c_{1,f}^0 = 150\,\text{mol/m}^3, \qquad c_{2,f}^0 = 300\,\text{mol/m}^3$$

$$u(t) \in [0, 1]\,\text{m}^3/\text{h}$$

Results

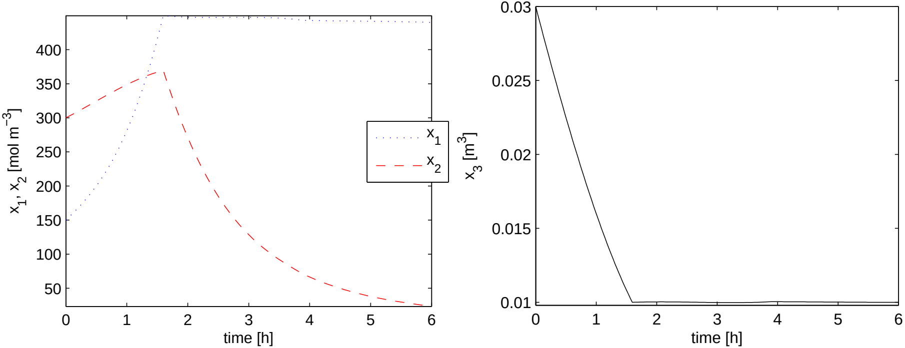
TD mode is optimal for this process and cost:



Control $(u(t), \alpha(t))$

Results

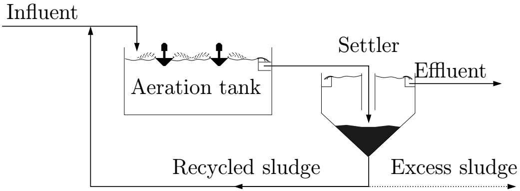
TD mode is optimal for this process and cost:



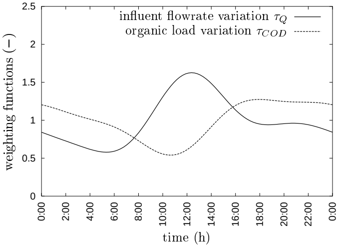
States ($c_{1,2}(t)$, $V(t)$), $c_2(6) = 23.38\,\text{mol/m}^3$

Overview

Process



Influent
Settler
Effluent
Aeration tank
Recycled sludge
Excess sludge



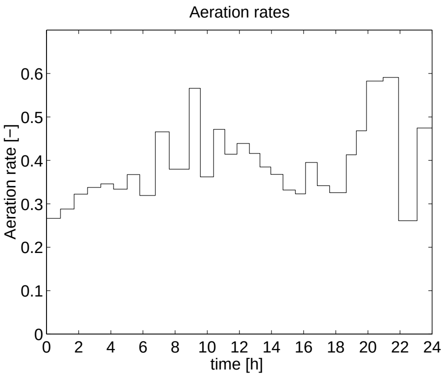
influent flowrate variation τ_Q
organic load variation τ_{COD}
weighting functions (–)
time (h)
0:00
2:00
4:00
6:00
8:00
10:00
12:00
14:00
16:00
18:00
20:00
22:00
0:00
0
0.5
1
1.5
2
2.5

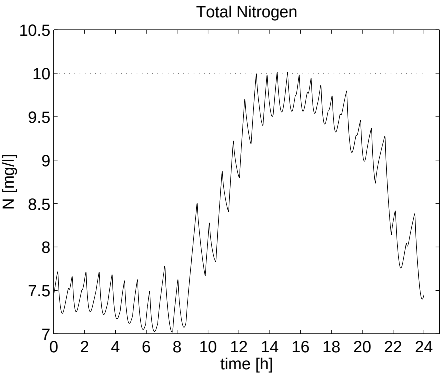
Optimisation problem

- Minimisation of aeration (energy consumption)
- Periodic steady state
- Environmental constraints, EU directives (TN <10)
- Model: ASM1

Results

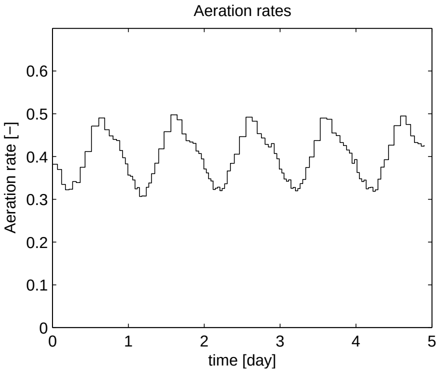
Average aeration below 40%

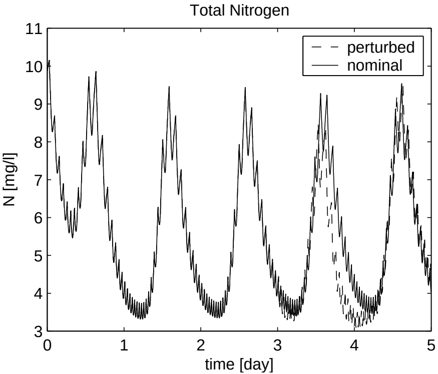




Feedback Rules

1. Start aeration when S_{NO} decreases sufficiently close to zero,
2. Stop aeration when S_O reaches a certain value.





Overview

Process

Mixture: $C_2 - C_5$, 40+2 trays
Local control: PI for levels in reboiler, condenser, PI for pressure on the head
Optimised variables: Reflux R, reboiler duty Q^{rebo}.

$$\begin{aligned} R &\in (6, 20) \text{ t/h} \\ Q^{rebo} &\in (6, 22) \text{ th} \end{aligned}$$

Outputs: Impurities in distillate and bottom flows X_t, X_b.

Model: Tray *j*

- material balances
- enthalpic balances
- accumulation terms
- thermodynamic equilibrium (SRK model)
- hydrodynamics (Gallund and Holland)

20 DAEs per tray
Together: 850 DAEs

Model: Tray j

- material balances
- enthalpic balances
- accumulation terms
- thermodynamic equilibrium (SRK model)
- hydrodynamics (Gallund and Holland)

20 DAEs per tray

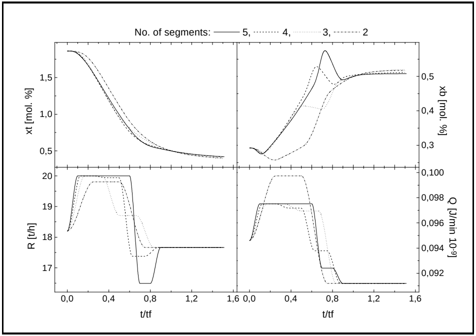
Together: 850 DAEs

Model: Tray j

- material balances
- enthalpic balances
- accumulation terms
- thermodynamic equilibrium (SRK model)
- hydrodynamics (Gallund and Holland)

20 DAEs per tray
Together: 850 DAEs

Optimisation Results



Number of times	t_f [min]	Time [h]	Num. of variables
2	43.3	0.44	5
3	34.7	0.68	7
4	32.4	0.83	9
5	29.0	0.56	11

Conclusions

- Transformation of the original dynamic problems to NLP
- Numerically efficient approaches: orthogonal collocation, control vector parametrisation
- Numerical calculation + analysis = optimal operation
- Understanding of process behaviour is essential

Packages:

Dynopt (MATLAB) – total parametrisation

DYNO (F77) – control vector parametrisation

M. Fikar.
Dynamická optimalizácia procesov.
STU Bratislava, Slovenská e-akadémia, n. o., 2007.

M. Fikar, B. Chachuat, and M. A. Latifi.
Optimal operation of alternating activated sludge processes.
Control Engineering Practice, 13(7):853–861, 2005.

M. Fikar, Z. Kovács, and P. Czermak.
Dynamic optimization of batch diafiltration processes.
Journal of Membrane Science, 355(1-2):168–174, 2010.

M. Fikar, M. A. Latifi, and Y. Creff.
Optimal changeover profiles for an industrial depropanizer.
Chem. Eng. Sci., 54(13-14):2715–2720, 1999.

Acknowledgements

This work would not be possible without work of my colleagues:

- Dr. B. Chachuat (Nancy)
- Ing. M. Čižniar (Bratislava)
- Ing. T. Hirmajer, PhD. (Bratislava)
- Dr. Z. Kovács (Giesen)
- Prof. M. A. Latifi (Nancy)
- Ing. R. Paulen (Bratislava)