Dynamic Real-time Optimization of Batch Membrane Processes using Pontryagin's Minimum Principle

R. Paulen, A. Sharma, M. Fikar

Faculty of Chemical and Food Technology

Slovak University of Technology in Bratislava

MOTIVATION

- Dynamic real-time optimization approaches are rarely robust
- Robust optimization approaches are rarely real-time
- Robust real-time optimization is rarely computationally efficient

Motivation

- Dynamic real-time optimization approaches are rarely robust
- Robust optimization approaches are rarely real-time
- Robust real-time optimization is rarely computationally efficient

- We study robust dynamic real-time optimization of a class of batch processes.
  - Pontryagin's minimum principle is used to tackle the problem of computational efficiency
  - Set-membership (guaranteed) parameter estimation and set-based propagation is used to guarantee robustness

We study the problem of a form

$$\min_{u(t)\in[u_L,u_U],t_f} \mathcal{J}(p) := \min_{u(t)\in[u_L,u_U],t_f} \int_0^{t_f} F_0(x(t),p) + F_u(x(t),p)u(t)\,\mathrm{d}t$$

$$\text{s.t. } \dot{x}(t) = f_0(x(t),p) + f_u(x(t),p)u(t), \quad x(0) = x_0, \quad x(t_f) = x_f$$

Dynamic real-time optimization

We study the problem of a form

$$\min_{u(t)\in[u_L,u_U],t_f} \mathcal{J}(p) := \min_{u(t)\in[u_L,u_U],t_f} \int_0^{t_f} F_0(x(t),p) + F_u(x(t),p)u(t)\,\mathrm{d}t$$

$$\text{s.t. } \dot{x}(t) = f_0(x(t),p) + f_u(x(t),p)u(t), \quad x(0) = x_0, \quad x(t_f) = x_f$$

Real-time implementation of the optimal control policy is tricky due to unknown parameters $p \in P$. Many ways to approach the problem:

- Off-line (nominal) optimization
- Robust (worst-case) off-line optimization
- On-line re-optimization based on available data
- This talk: computationally efficient combination of the above

We study the problem of a form

$$\min_{u(t)\in[u_L,u_U],t_f} \mathcal{J}(p) := \min_{u(t)\in[u_L,u_U],t_f} \int_0^{t_f} F_0(x(t),p) + F_u(x(t),p)u(t)\,\mathrm{d}t$$

$$\text{s.t. } \dot{x}(t) = f_0(x(t),p) + f_u(x(t),p)u(t), \quad x(0) = x_0, \quad x(t_f) = x_f$$

One can use Pontryagin's minimum principle in order to define the state-feedback policy parameterized with $\pi(p) := (t_1(p), t_2(p), t_f(p))^T$

$$u^*(t,\pi(p)) := \begin{cases} u_L, & t\in[0,t_1],\ S(x(t),p) > 0, \\ u_U, & t\in[0,t_1],\ S(x(t),p) < 0, \\ u_s(x(t),p), & t\in[t_1,t_2],\ S(x(t),p) = 0, \\ u_L, & t\in[t_2,t_f],\ S(x_f,p) < 0, \\ u_U, & t\in[t_2,t_f],\ S(x_f,p) > 0, \end{cases}$$

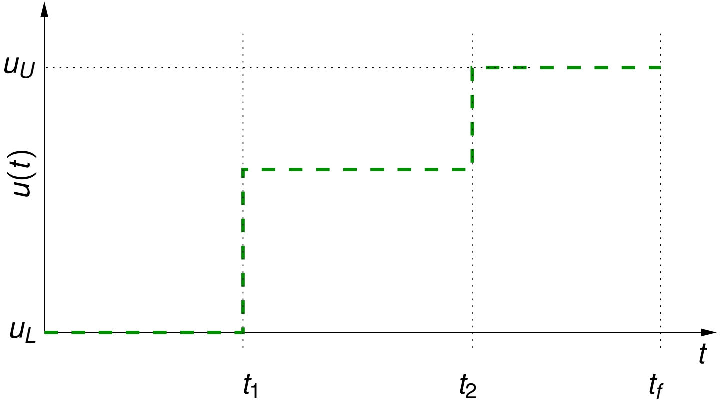
State-feedback policy parameterized with $\pi(p) := (t_1(p), t_2(p), t_f(p))^T$

$$u^*(t,\pi(p)) := \begin{cases} u_L, & t \in [0, t_1],\ S(x(t), p) > 0, \\ u_U, & t \in [0, t_1],\ S(x(t), p) < 0, \\ u_s(x(t), p), & t \in [t_1, t_2],\ S(x(t), p) = 0, \\ u_L, & t \in [t_2, t_f],\ S(x_f, p) < 0, \\ u_U, & t \in [t_2, t_f],\ S(x_f, p) > 0, \end{cases}$$

$$x_f = x(t_f) := x(t_2) + \int_{t_2}^{t_f} f_0(x(t), p) + f_u(x(t), p) u^*(t)\, \mathrm{d}t.$$

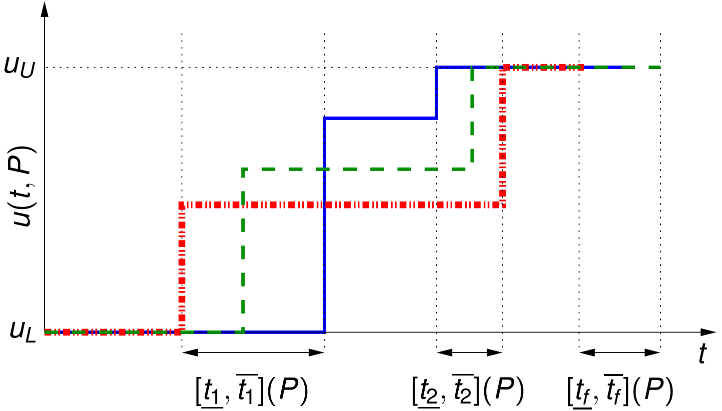
Switching function $S(x(t), p)$ and singular control trajectory $u_s(x(t), p)$:
a) found explicitly in certain cases, b) determined numerically or
c) approximated by low-order polynomial function.

Using set propagation (e.g. Taylor models) one can project uncertainty in the model parameters $p \in P$ into “uncertain” control policy



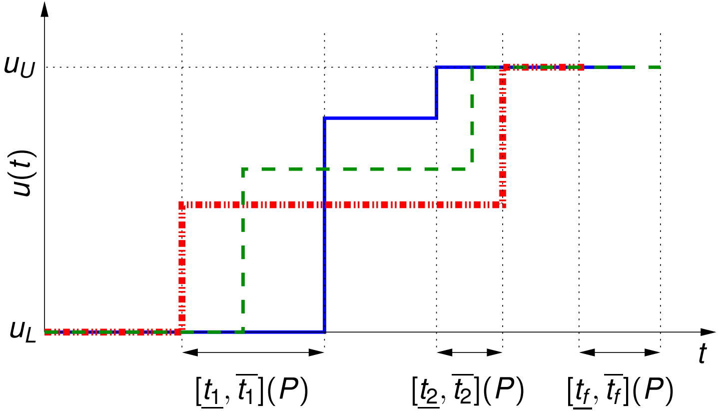
Paulen, Villanueva, Chachuat. *IMA J. of Mathematical Control and Information*, 2016.

Using set propagation (e.g. Taylor models) one can project uncertainty in the model parameters $p \in P$ into "uncertain" control policy



Paulen, Villanueva, Chachuat. *IMA J. of Mathematical Control and Information*, 2016.

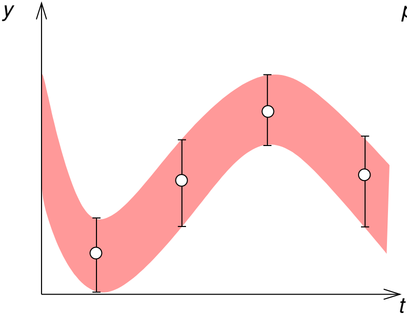
1. No need to re-optimize until switching time $\underline{t}_i$ is reached. Just take measurements.

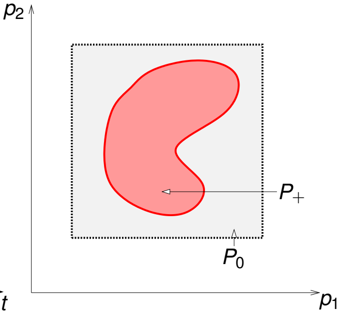


Paulen, Villanueva, Chachuat. *IMA J. of Mathematical Control and Information*, 2016.

Exploiting known control uncertainty

1. No need to re-optimize until switching time $\underline{t}_i$ is reached. Just take measurements.
2. Update parameter uncertainty; project it into control uncertainty.





We use set-membership (guaranteed) estimation ($|y - c^T p| \leq \sigma$)

1. No need to re-optimize until switching time $\underline{t}_i$ is reached. Just take measurements.
2. Update parameter uncertainty; project it into control uncertainty.

$$\hat{p}_+ := \hat{p} + \frac{\beta\Sigma c}{\sigma^2 + \beta c^T \Sigma c}(y - c^T\hat{p}),$$

$$\Sigma_+ := \left(1 + \beta - \frac{\beta(y - c^T\hat{p})^2}{\sigma^2 + \beta c^T\Sigma c}\right)\left(\Sigma - \frac{\beta}{\sigma^2 + \beta c^T \Sigma c}\Sigma c c^T \Sigma\right),$$

$$P_+ := \left[\hat{p}_+ - \mathrm{diag}\left(\Sigma_+^{\frac{1}{2}}\right), \hat{p}_+ + \mathrm{diag}\left(\Sigma_+^{\frac{1}{2}}\right)\right].$$

We use set-membership (guaranteed) estimation ($|y - c^T p| \leq \sigma$)

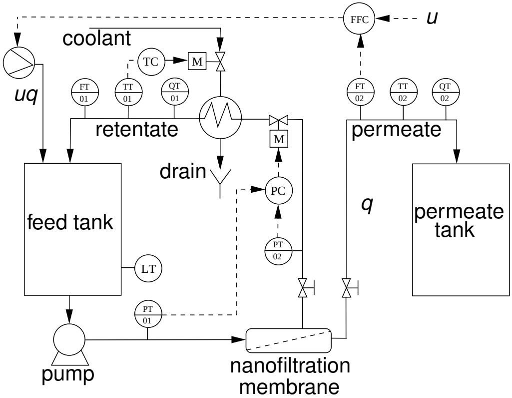
E. Fogel, Y. Huang, Automatica, 1982.

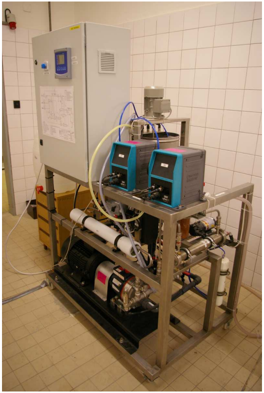
1. No need to re-optimize until switching time $\underline{t}_i$ is reached. Just take measurements.
2. Update parameter uncertainty; project it into control uncertainty.
3. Re-optimize only when necessary. E.g. minimize variance of the objective under the worst-case realization of $p \in P$

$$\min_{\substack{\tilde{u}_S(t,p)\in[u_L,u_U]\\ t_i\in T_i,\ \forall i\in\{1,2,f\}}} \max_{p\in P} \left\|\mathcal{J}(p) - \min_{p^*} \mathcal{J}(p^*)\right\|_2^2$$

Re-optimize when necessary $\rightarrow$ optimal mid-course correction

LABORATORY MEMBRANE PLANT

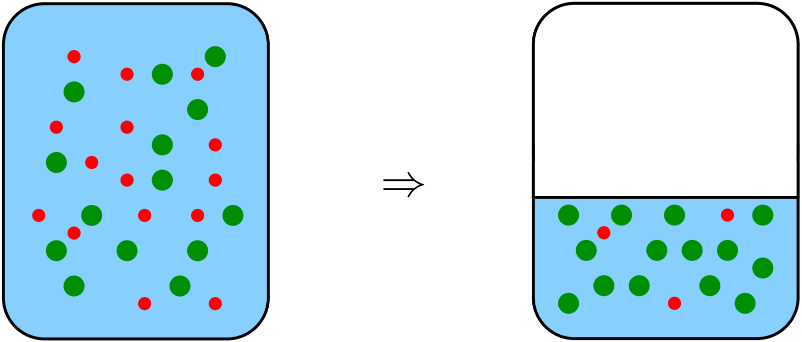




The plant is highly automated; all process variables (concentrations and flowrates) measured.

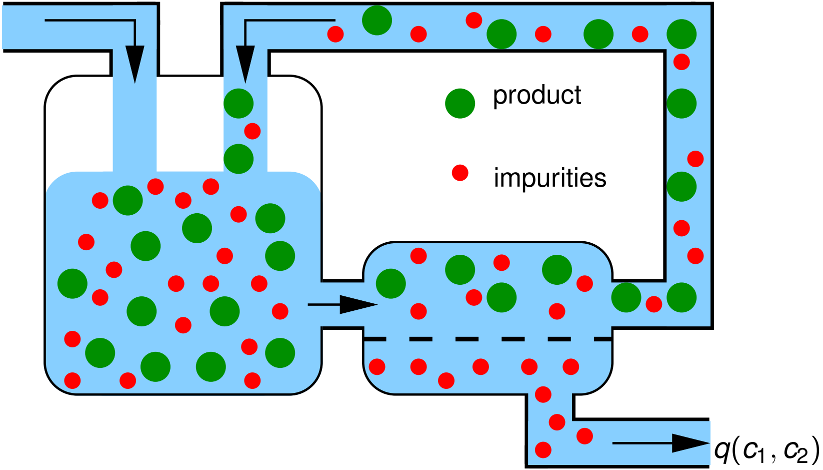
Experiments show significant batch-to-batch variations.

Initial state | Final state



$x_0 = [c_{1,0}\,; c_{2,0}]$ | $x_\mathrm{f} = [c_{1,\mathrm{f}}\,; c_{2,\mathrm{f}}]$

product $c_1 \uparrow$ | impurities $c_2 \downarrow$ | water



– key membrane characteristic to model: permeate flow $q(c_1, c_2)$

Generalized limiting flux model:

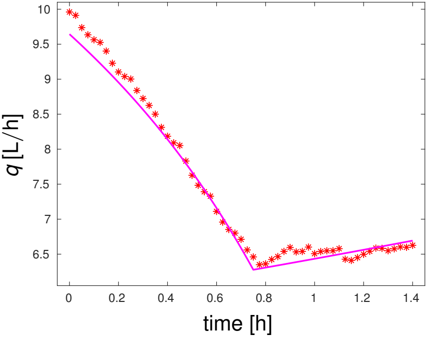
$$q = p_1 \ln \left(\frac{p_2}{c_1 c_2^{p_3}} \right)$$

$p_1 \in P_1 := [1.86, 3.91]\,\text{L/h}$

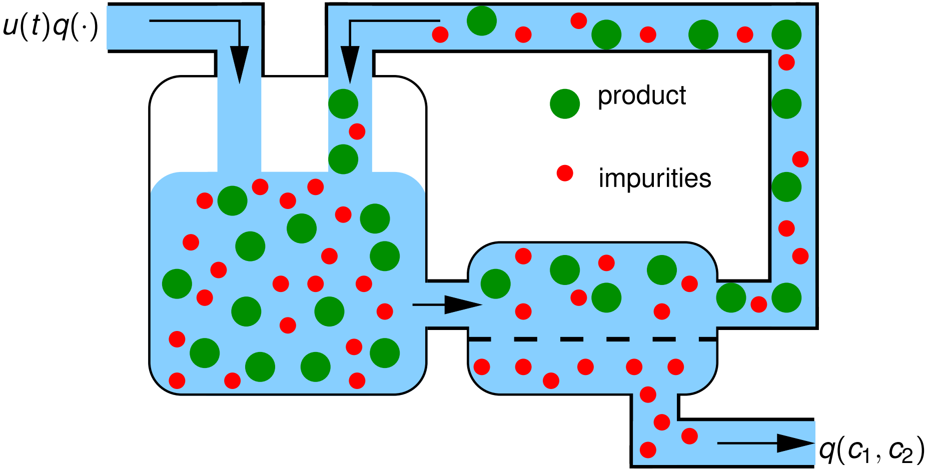
$p_2 \in P_2 := [658.93, 4117.15]\,\text{g/L}$

$p_3 \in P_3 := [-0.11, 0.17]$

Data fit with operation $u = \{0, 1\}$

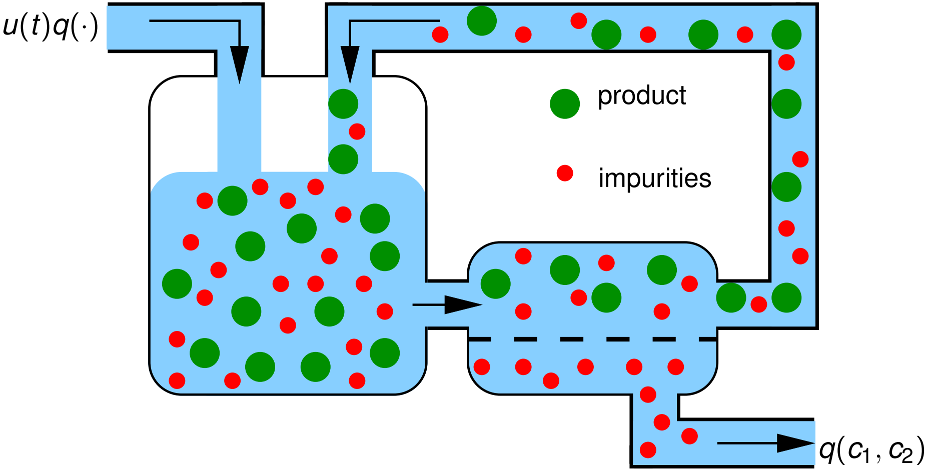


Fitting of parameters to the experimental data done using set-membership estimation with error bound $\sigma = 0.5\,\text{L/h}$.



– concentrations dynamically adjusted by the water inflow $u(t)q(\cdot)$

OPTIMIZATION GOAL



– find $u(t)$ to minimize final (batch) time

Model:

$$\frac{\mathrm{d}c_1(t)}{\mathrm{d}t} = \frac{c_1^2(t)}{c_{1,0}V_0} p_1 \ln\left(\frac{p_2}{c_1 c_2^{p_3}}\right)(1 - u(t))$$

$$\frac{\mathrm{d}c_2(t)}{\mathrm{d}t} = -\frac{c_1(t)c_2(t)}{c_{1,0}V_0} p_1 \ln\left(\frac{p_2}{c_1 c_2^{p_3}}\right) u(t)$$

Optimality conditions:

$$u_s(x(t), p) := 1/(1 + p_3),$$

$$S(x(t), p) := p_1 (\ln(p_2) - \ln(c_1) - p_3 \ln(c_2) - p_3 - 1) = 0.$$

$$u^*(t, \pi(p)) := \begin{cases} u_L, & t \in [0, t_1],\ S(c_1(t), c_2(t), p) > 0, \\ u_s(c_1(t), c_2(t), p), & t \in [t_1, t_2],\ S(c_1(t), c_2(t), p) = 0, \\ u_U, & t \in [t_2, t_f],\ S(c_{1,f}, c_{2,f}, p) > 0, \end{cases}$$

RESULTS

- true optimum: $t_f = 2.86\,\mathrm{h}$
- robust operation, no adaptation: $t_f = 4.23\,\mathrm{h}$
- proposed approach: $t_f = 2.87\,\mathrm{h}$

- The proposed approach requires no re-optimization! Only one control policy adaptation performed!

Conclusions

- Dynamic real-time optimization combined with ideas from robust optimization thanks to the use of guaranteed parameter estimation and set-based propagation
- Computational tractability achieved with the use of Pontryagin's minimum principle and parameterization of the optimal control policy
- Application to minimum batch time problem of membrane processes. Extensions to economic (multi-objective) operation.

- Acknowledgement: H2020 MSCA-IF project (No. 790017) GuEst