Overview and Classification of online process optimization approaches

- IFAC DYCOPS Pre-symposium workshop

Sigurd Skogestad

Johannes Jäschke

Dinesh Krishnamoorthy

NTNU
Norwegian University of Science and Technology

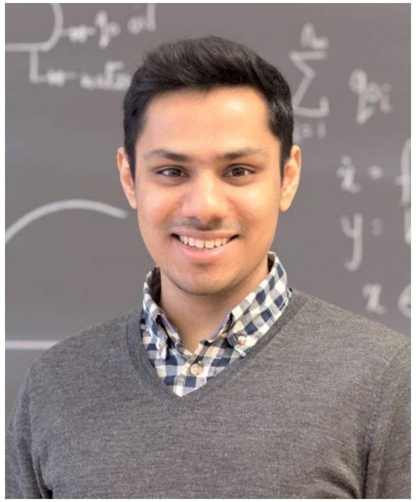

NTNU


Norwegian Sea
Trondheim
Sweden
Norway
Faroe Islands
Oslo
Stockholm
North Sea
Edinburgh
Copenhagen
Denmark
United Kingdom
Isle of Man
Hamburg
Dublin
Manchester
Ireland
Liverpool
Berlin
Poland
Amsterdam
Netherlands
London
Brussels
Cologne
Germany
Belgium
Frankfurt
Prague
Luxembourg
Czechia

Presenters

Sigurd Skogestad
Professor NTNU
PhD Caltech (1987)

Johannes Jäschke
Associate Prof. NTNU
PhD NTNU (2011)



Dinesh Krishnamoorthy
PhD student NTNU (2019)
MS Imperial College (2012)

NAME
Alasdair Jack Speakman
Alireza Olama
Aris Papasavvas
Bruno Morabito
Carlos Roberto Chaves
Christiam Segundo Morales Alvarado
Cristina Zotica
Diogo Filipe Mateus Rodrigues
Diogo Ortiz Machado
Gabriel Lapa Grandi
Hiago Antonio Sirino Dangui
JESUS DAVID HERNANDEZ ORTIZ
Joakim Rostrup Andersen
José Diogo Forte de Oliveira Luna
Jose Dolores Vergara Dietrich
José Eduardo Weber dos Santos
Lucian Silva
Mandar Thombre
Marta Zagorowska
Mathilde Hotvedt
Otávio Fonseca Ivo
Rafael Sartori
Reinaldo Enrique Hernandez Rivas
Rodrigo Juliani Correa de Godoy
Seyed Jamal Haddadi

NTNU

Agenda

13:30 – 13:50	Welcome and info
13:50 – 14:00	Introduction Real time optimization (Sigurd Skogestad)
14:00 – 14:10	1. Conventional approach: Steady-state optimization - and challenges (Sigurd Skogestad)
14:10 – 14:40	2. Academic approach: Economic MPC and Dynamic RTO (Johannes Jäschke)
14:40 – 15:00	3. Steady-state optimization with transient measurements – Hybrid RTO (Dinesh Krishnamoorthy)
15:00 – 15:20	4. Feedback-based RTO using model-based gradient estimation (Dinesh Krishnamoorthy)
15:20 – 15:40	5. Extremum seeking control (Dinesh Krishnamoorthy)
15:40 – 16:00	Coffee Break
16:00 – 16:20	6. Modifier Adaptation (Johannes Jäschke)
16:20 – 16:40	7. Self-optimizing Control (Sigurd Skogestad)
16:40 – 17:00	8. Classical Approach: Optimal operation using conventional advanced control (Sigurd Skogestad
17:00 – 17:30	The different approaches are complementary, not contradictory! (Johannes Jäschke)
17:30 – 18:00	Discussion and Concluding remarks (all)

NTNU
Norwegian University of Science and Technology

Introduction to Real time optimization

IFAC DYCOPS Pre-symposium workshop

Sigurd Skogestad

Main objectives control system

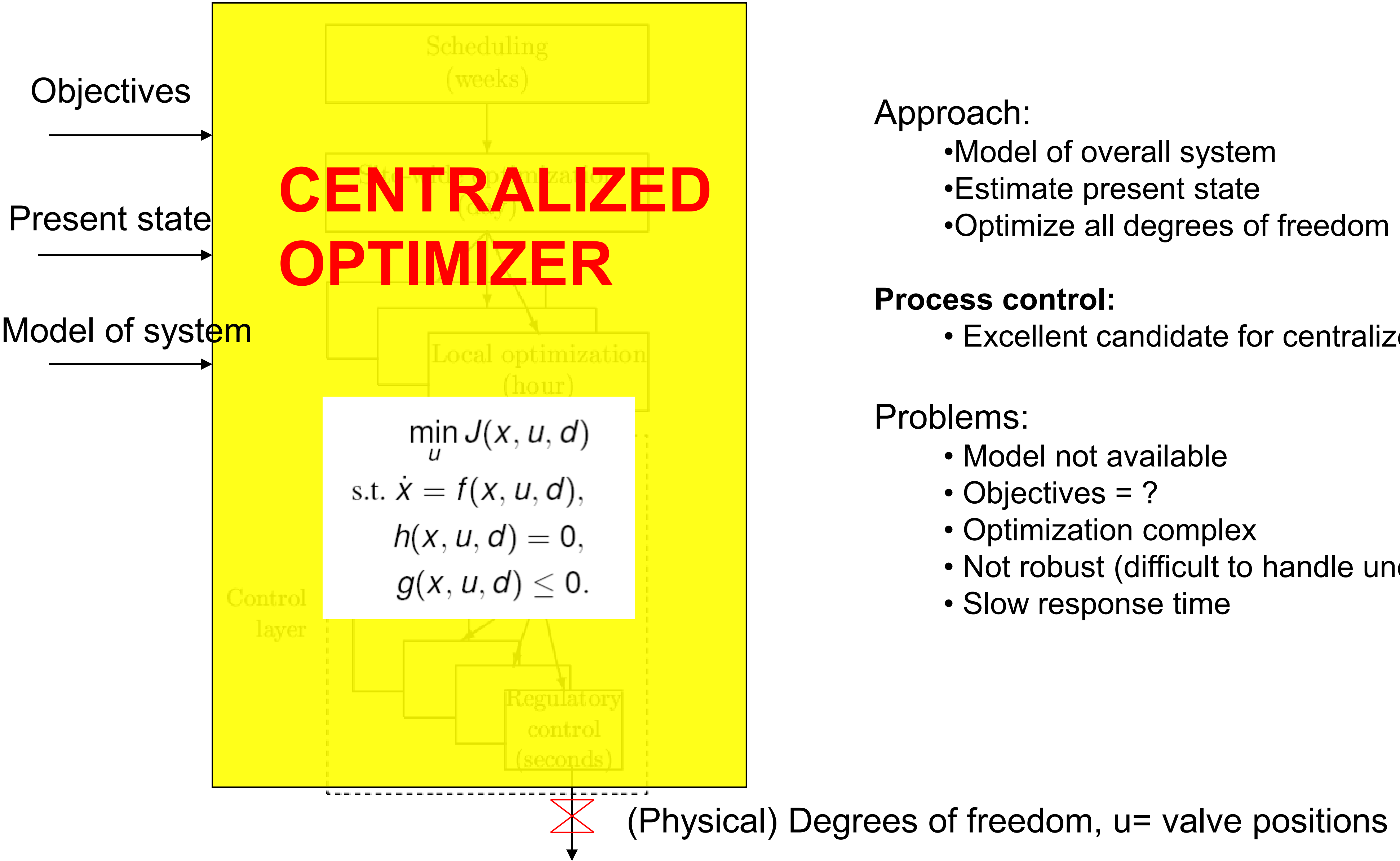
1. **Economics**: Implementation of acceptable (near-optimal) operation

2. **Regulation:** Stable operation

ARE THESE OBJECTIVES CONFLICTING?

- Usually NOT
  - Different time scales
    - Stabilization fast time scale
  - Stabilization doesn’t “use up” any degrees of freedom
    - Reference value (setpoint) available for layer above
    - But it “uses up” part of the time window (frequency range)

In theory: Centralized controller is always optimal (e.g., EMPC)



Approach:

- Model of overall system
- Estimate present state
- Optimize all degrees of freedom

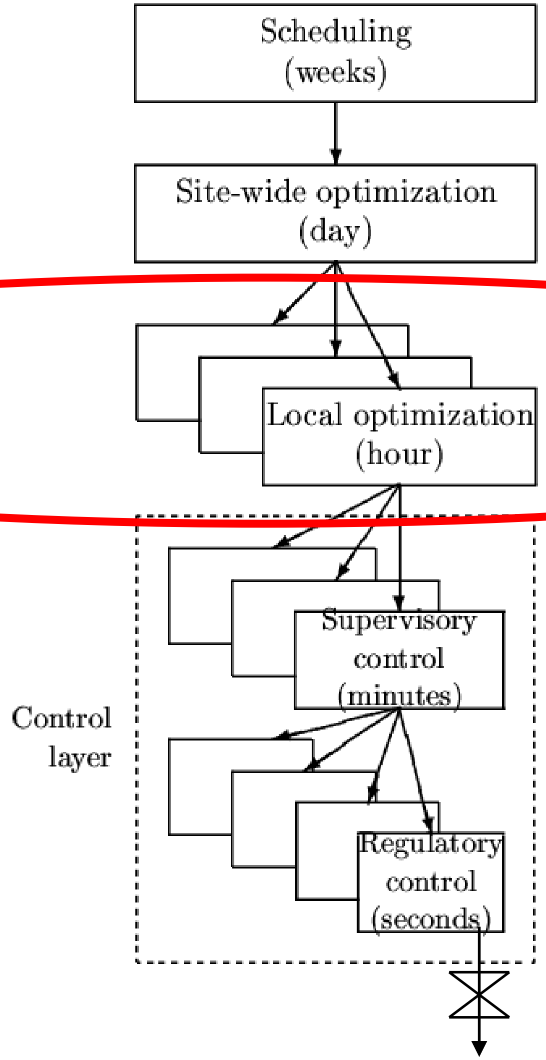
Process control:

- Excellent candidate for centralized control

Problems:

- Model not available
- Objectives = ?
- Optimization complex
- Not robust (difficult to handle uncertainty)
- Slow response time

In practice: Hierarchical decision system based on time scale separation



Manager

Process engineer

Focus of this workshop

min J (economics) — Operator/RTO

Setpoint control
(+ look after other variables) — "Advanced classical control"/MPC

Stabilize + avoid drift — PID-control

u = valves

NTNU

General objective process operation (RTO):
Minimize cost J = maximize profit (–J) [$/s]
subject to constraints

$$J = \sum p_F\, F + \sum p_Q\, Q - \sum p_p\, P$$

where

- $\sum p_F\, F$ = price of feed [$/kg] x feed flow rate [kg/s]
- $\sum p_Q\, Q$ = price of utility (energy) x energy usage
- $\sum p_P\, P$ = price (value) of product x product flow rate

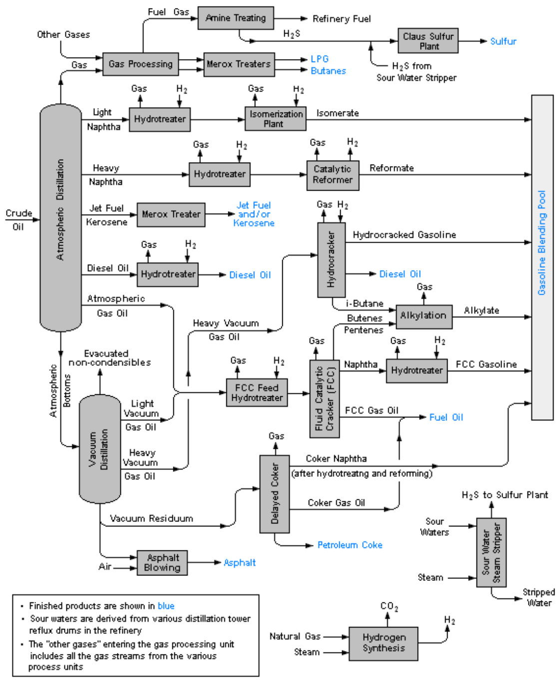
Note: No capital costs or costs for operators (assumed fixed for time scale of interest, a few hours)

Typical process constraints:

- Product quality (purity)
- Environment (amount and purity of waste products)
- Equipment (max. and min. flows, pressures)

Typical degrees of freedom (decision variables) (u)

- Flowrates: Feeds, splits (recycles), heating/cooling



Two main operation modes

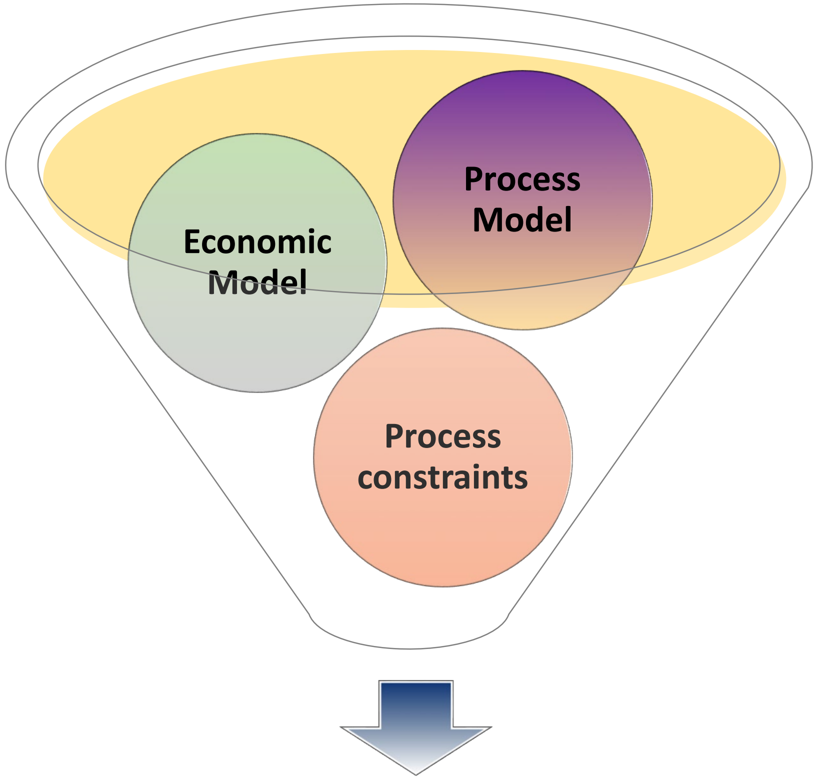
I. Sales limited by market: Given production (constraint)

- Optimal with high energy efficiency (good for environment)

II. High price product and high demand: Maximize production

- Lower energy efficiency
- Optimal to overpurify waste products to recover more (good for environment)

Formulation of Real time optimization



Optimal decision variables (u) / Optimal setpoints (y)

$$\min J(x, u, d)$$

s.t.

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

$$g(x, u, d) \leq 0$$

$x \in \mathcal{X}$, Internal variables

$u \in \mathcal{U}$ Decision variables

d: Parameter values / disturbances

NTNU
Norwegian University of Science and Technology

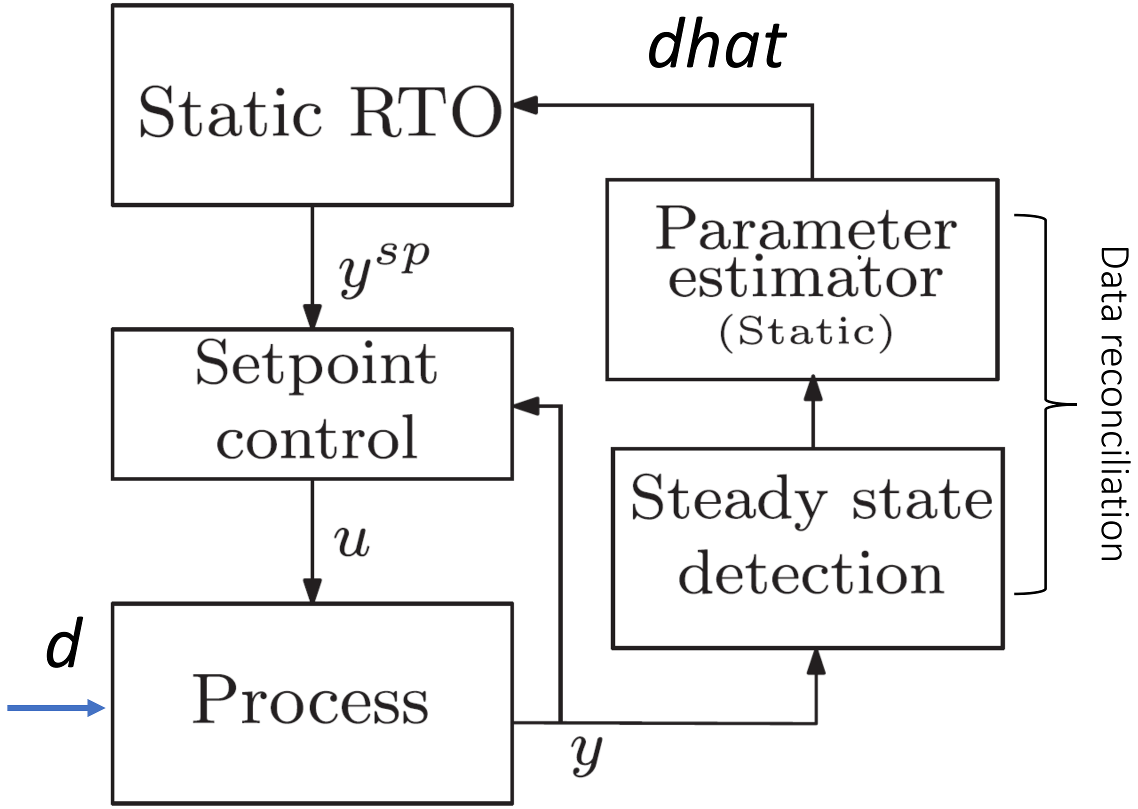
1. Conventional approach: Steady-state Real time optimization - and Challenges

IFAC DYCOPS Pre-symposium workshop

Sigurd Skogestad

Conventional (commercial) steady-state RTO

- Steady-state models
- Two-step approach
  1. "Data reconciliation":
     - Steady-state detection
     - Update estimate of d: model parameters, disturbances (feed), constraints
  2. Re-optimize to find new optimal steady state



D. E. Seborg, D. A. Mellichamp, T. F. Edgar, F. J. Doyle III, Process dynamics and control, John Wiley & Sons, 2010.

Conventional steady-state RTO

- Typically uses detailed process models with full thermo package
  - Hysys / Unisim (Honeywell)
  - Aspen
  - PROCESS
  - …..

- But traditional RTO less used in practice than one should expect
  - Ethylene plants (furnace)
  - Some refinery applications

Why is conventinal static RTO not commonly used?

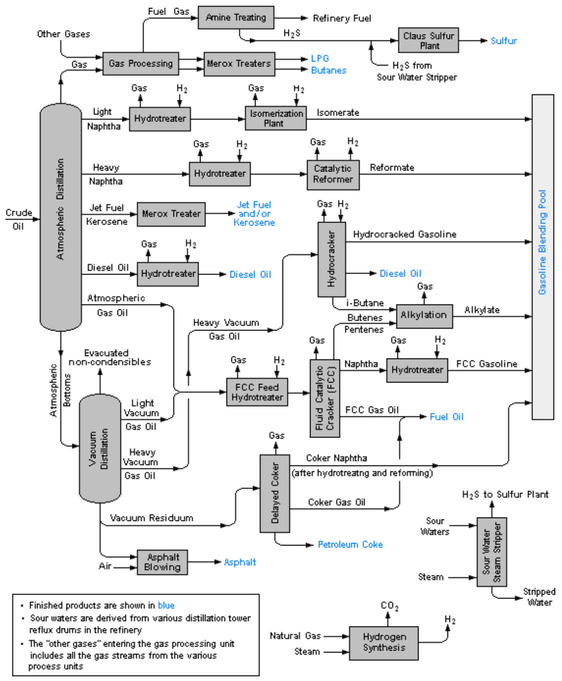
Problems (in expected order of importance):

1. Cost of developing and updating the model (costly offline model update)
2. Wrong value of model parameters and disturbances d (slow online model update)
3. Not robust, including computational issues
4. Frequent grade changes make steady-state optimization less relevant
5. Dynamic limitations, including infeasibility due to (dynamic) constraint violation
6. Incorrect model structure

Not problems, but Challenges ☺

Challenge 1 - Costly offline model development and update

- Lack of domain/expert knowledge
- Change in process configuration
- Model simplification



Possible Fix: Data-methods based on measuring Cost (J) (Extremum seeking control; see Approach 5)

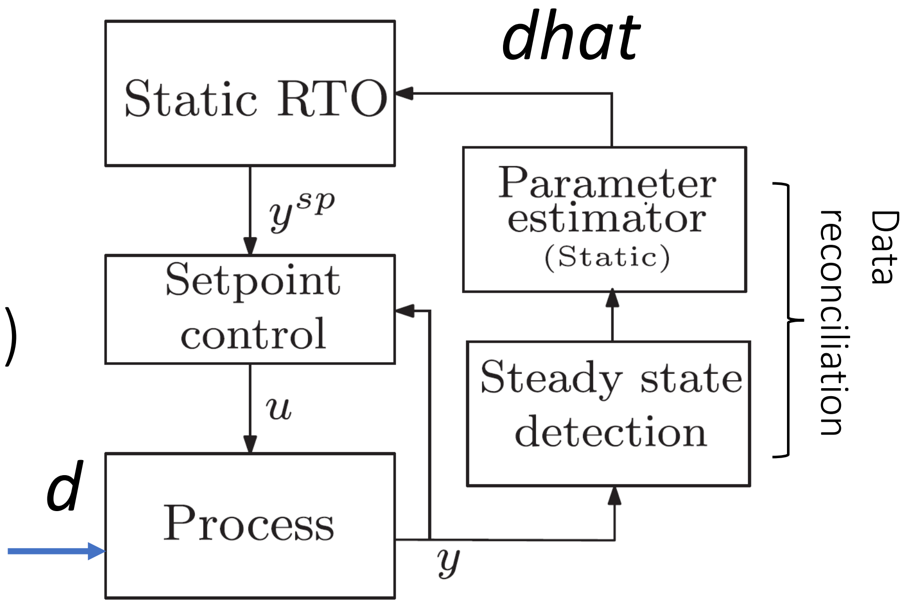
Recent interest – Machine learning and AI to develop models

NTNU

Challenge 2 - Steady-state wait time

- Frequent disturbances (d)
- Long settling times

- Data reconciliation step is infrequent
- Wrong value of model parameter/disturbances (dhat)
- Process operates sub-optimally for long periods of time



Fix: Use dynamic model in estimation step (Hybrid RTO; see Approach 3)

Challenge 3 - Computational issues

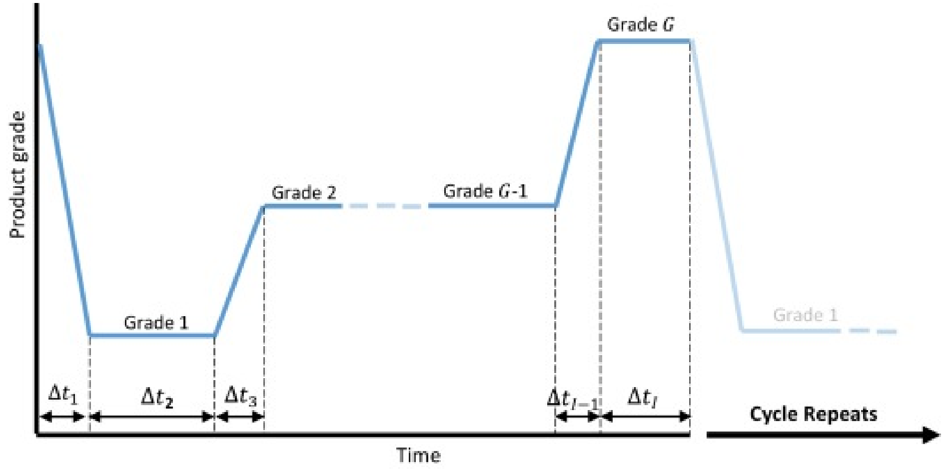
- Convergence issues and numerical failure
- CPU times
- scaling of variables
- Complementary constraints

Fix – Methods that do not need to solve numerical optimization problems online

(Novel Feedback RTO; see Approach 4)

Challenge 4 - Frequent grade changes

- Continuous process with frequent changes in feed, product specifications, market disturbances, slow dynamics/long settling time
- Continuous with frequent grade transitions
- Batch processes
- Cyclic operations



Source: Koller et al. (2017) *Comput& Chem Eng*, *106*, pp.147-159.

Fix (if relevant) – Dynamic optimization methods (DRTO or EMPC; see Approach 2)

Challenge 5 - Dynamic limitations

- Dynamic constraint violations
- Force variables to fixed set points, may not utilize all degrees of freedom
- A steady-state optimization layer and a control layer may lead to model inconsistency

Partial Fix – Use setpoint tracking control layer below RTO

Challenge 6 – Incorrect model structure

- E.g. missing one chemical reaction
- Cannot be fixed by parameter updates

Fix – Modifier adaptation based on measuring Cost (J)

NTNU

Norwegian University of Science and Technology

2. Academic approach : Dynamic RTO and Economic MPC

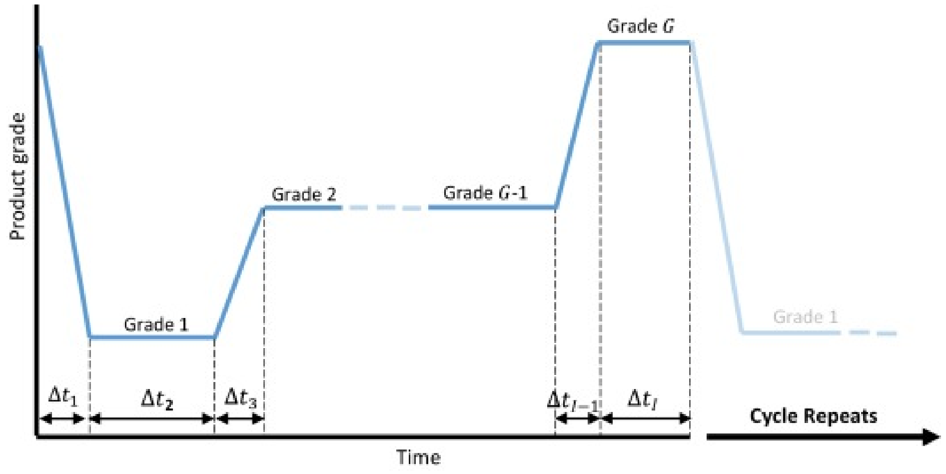
IFAC DYCOPS Pre-symposium workshop

Johannes Jäschke

DRTO and Economic MPC

Optimize not only steady state, but also transients

- Continuous process with frequent changes in feed, product specifications, market disturbances, slow dynamics/long settling time
- Continuous with frequent grade transitions
- Batch processes
- Energy storage
- Cyclic operations

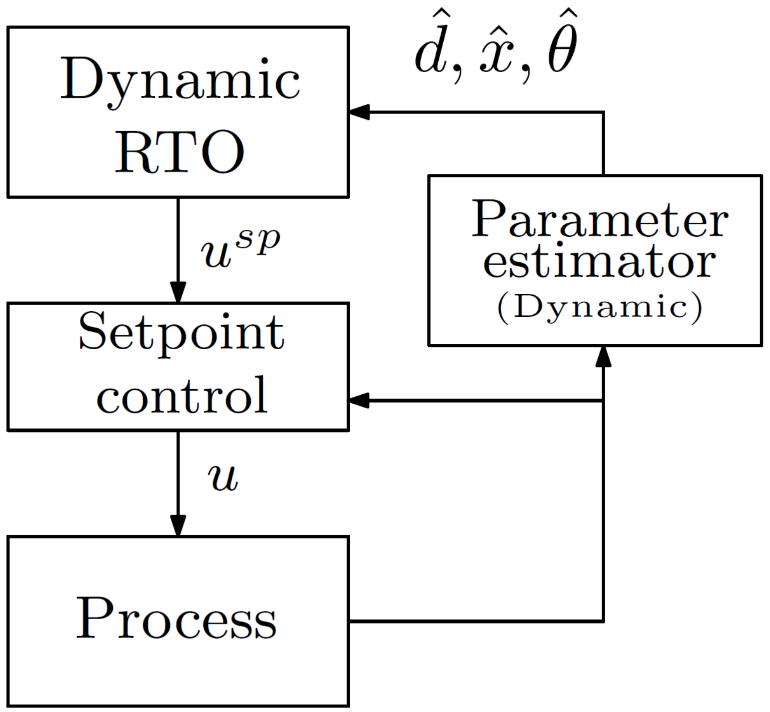


Source: Koller et al. (2017) *Comput& Chem Eng*, *106*, pp.147-159.

Directly address challenge 4 (frequent changes, non-negligible transient operation)

Dynamic RTO

- Uses dynamic models online
- Repeatedly solve Dynamic RTO problem for a given horizon
- Closely related to economic MPC



Main idea

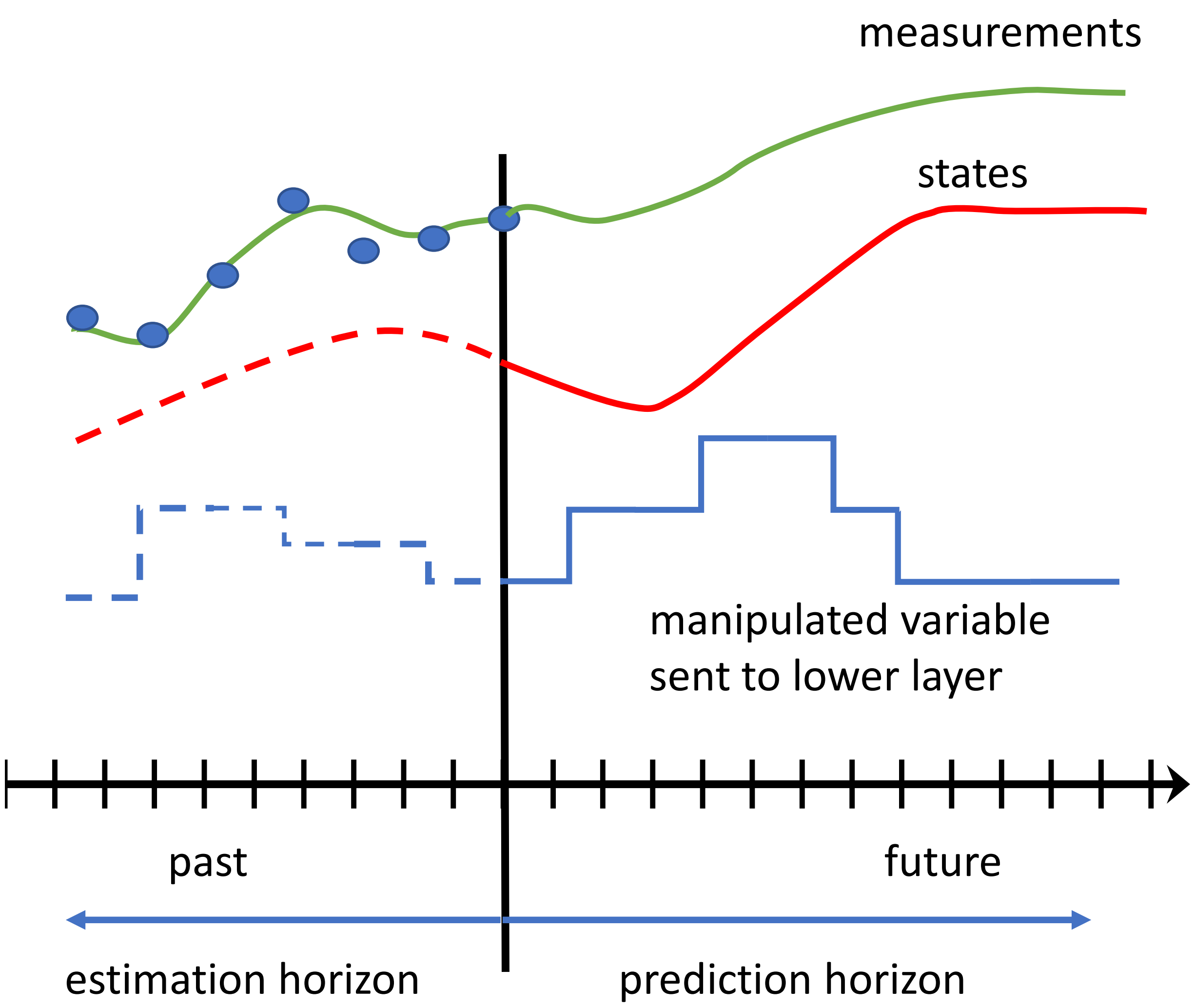
Repeatedly solve

Step 1: Dynamic Estimation

$$\hat{\theta}_k = \arg \underset{\theta}{min} \|y_{meas,k} - h(x_k, u_k)\|$$
$$s.t.\ \ x_k = f(x_{k-1}, u_{k-1}, \theta)$$

Step 2: Dynamic Optimization

$$u_t^* = \arg\min_{u_t} \sum_{t=k}^{k+T} J(y_t, u_t)$$
$$\begin{aligned} s.t.\ \ & x_{t+1} = f(x_t, u_t, \hat{\theta}_k) \\ & y_t = h(x_t, u_t) \\ & g(y_t, u_t) \le 0 \\ & x_k = \hat{x}_k \qquad\qquad \forall t \in \{k, \ldots, k+T\} \end{aligned}$$



Main idea

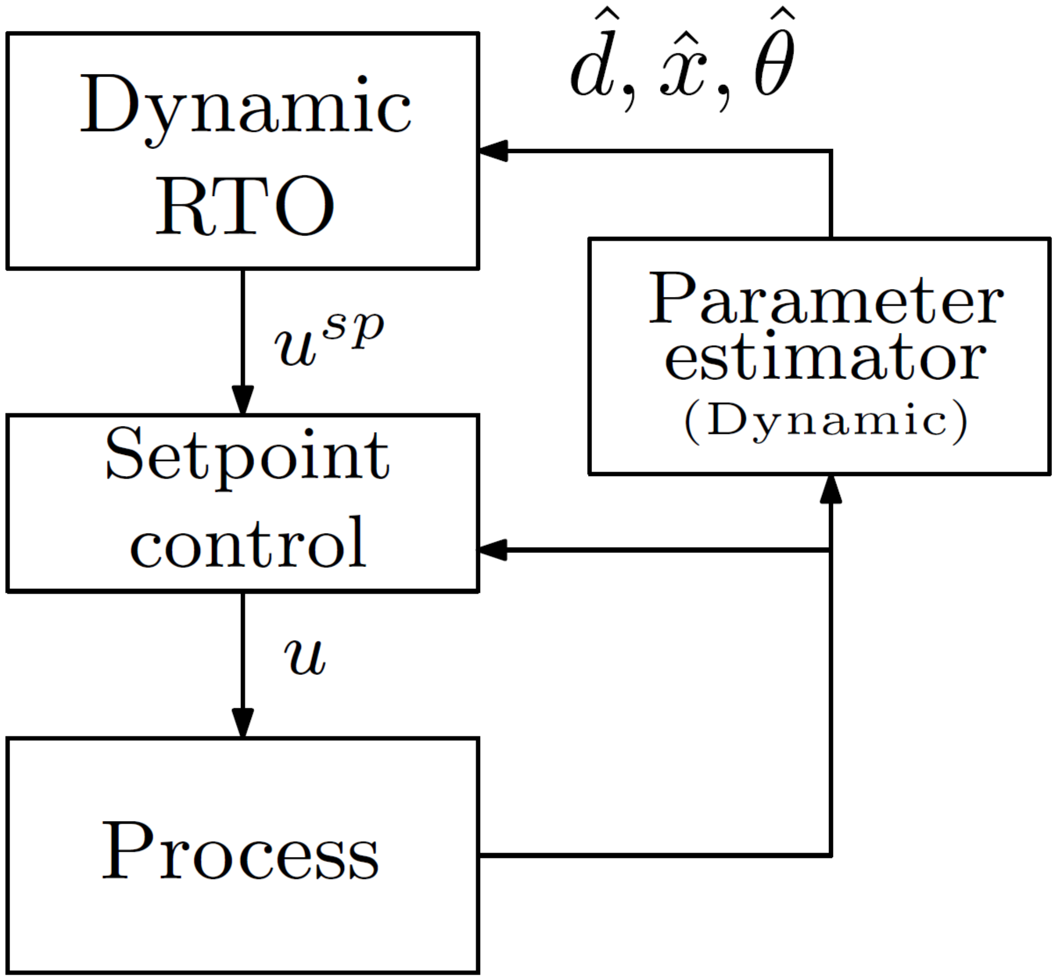
Repeatedly solve

Step 1: Dynamic Estimation

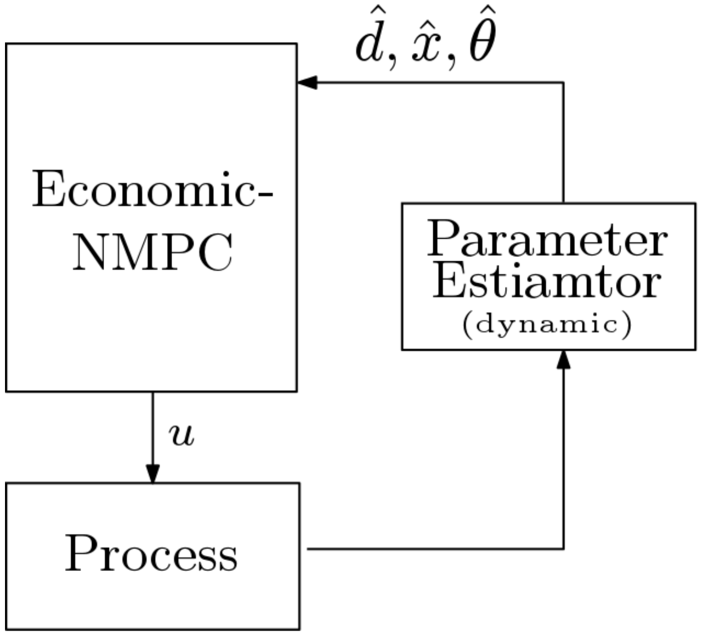
$$\hat{\theta}_k = \arg\min_{\theta} \|y_{meas,k} - h(x_k, u_k)\|$$
$$s.t.\ \ x_k = f(x_{k-1}, u_{k-1}, \theta)$$

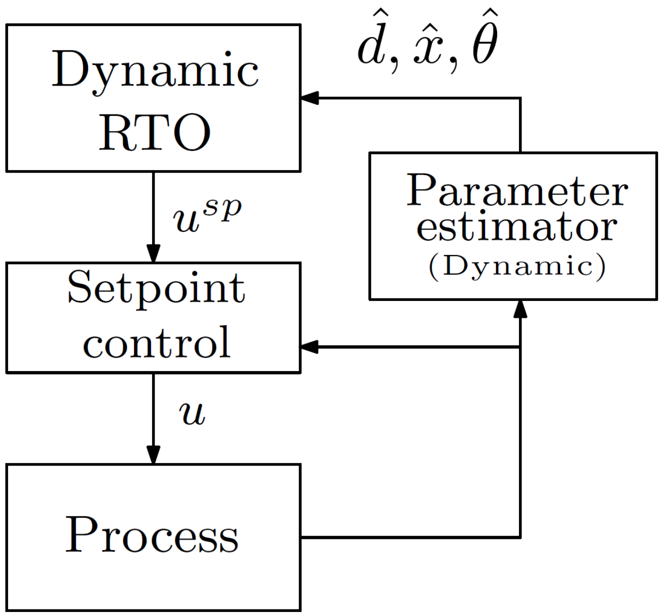
Step 2: Dynamic Optimization

$$u_t^* = \arg\min_{u_t} \sum_{t=k}^{k+T} J(y_t, u_t)$$
$$\begin{aligned} s.t.\ \ & x_{t+1} = f(x_t, u_t, \hat{\theta}_k) \\ & y_t = h(x_t, u_t) \\ & g(y_t, u_t) \leq 0 \\ & x_k = \hat{x}_k \qquad\qquad \forall t \in \{k, \ldots, k+T\} \end{aligned}$$



Economic MPC ~ Dynamic RTO





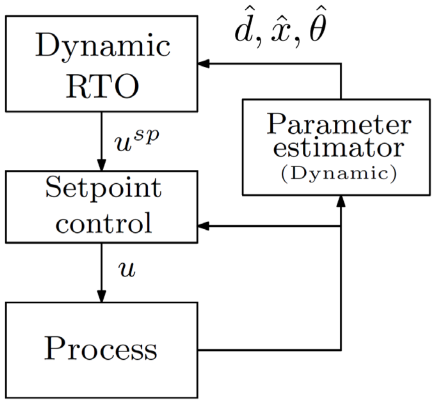
- Centralized "*All-in-one*" optimizer
- Higher sampling rates

- Hierarchical layers with time scale separation
- Lower sampling rates

Usually in Economic MPC a lower layer is also included, e.g. perfect level control, etc..

Industrial applications of Dynamic RTO

- Bartusiak (2007)
  - polyolefin polymerization processes
- Odloak+Petrobras
  - FCC unit
- Cybernetica
  - Polymerization reactors
- HVAC (Stanford) Rawlings



Challenges with Dynamic RTO

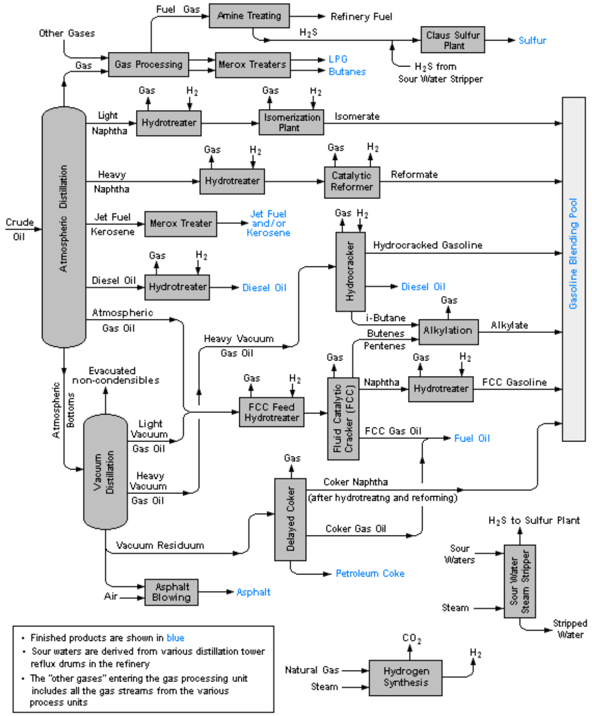
Main Challenge: Manage complexity

- Trade-off

Cost to make it work ⟷ and improved profit.

Complexity: The challenge with Dynamic RTO

- Obtaining and maintaining an accurate dynamic model
- Computational issues
- Robustness issues
- Implementation issues



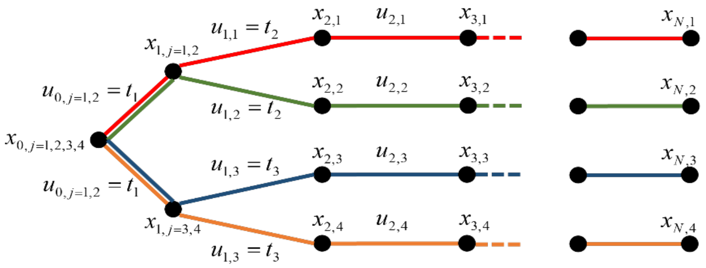
Obtaining and maintaining an accurate dynamic model

- Modelling efforts
- Requires plant testing over larger operation range
- Trade-off between learning model parameters and optimal operation

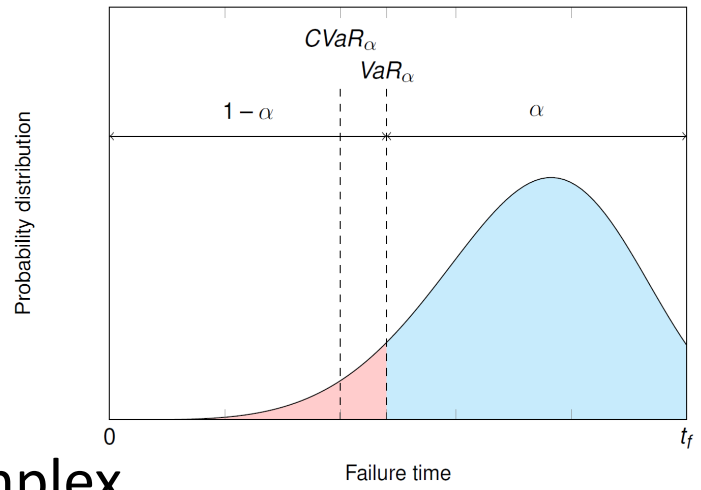
Computational issues

- Computational cost for solving the large NLP
  - NLP solvers (IPOPT (Biegler) Conopt, others…
  - Fast Sensitivity-based methods
    - Realtime iterations (Diehl et al 2002)
    - Advanced step NMPC (Zavala & Biegler 2009, Jäschke et al 2015)
  - Convergence issues
    - Thermodynamics model crashes, Flash computations
- Discrete and nonsmooth decisions
  - Lead to mixed integer optimization problems
  - Cannot be solved in real-time for large systems

Robustness issues



- Robustness issues
  - Model errors
  - Uncertain parameters (predictions)
  - Implications for stability
  - Tend to make computations significantly more complex



Implementation issues

- Tuning, regularization weights in cost function
  - Typical cost in practice (Bartusiak 2007)

$$J_{DRTO} = J_{eco} + J_{trac} + J_{input}$$

- Allowing for manual operations

- What to put into which layer?

- Measurement faults, reliable state and parameter estimation

Require many Ad-hoc problem-dependent solutions

Academia

- Stability
- Numerical issues
  - Computation speed
  - Decentralizing
- Uncertainty
  - Stochastic MPC
  - Robust MPC
  - Chance constraint
  - Dual MPC

Proofs mainly of concern for academia

Handle complexity in real-time

Typically add complexity

DRTO and EMPC has many potential benefits

- Reduced amount of off-spec product
- Changing operational strategy:
  - Agile operation switching productions
    - Demand-side management
    - Load-balancing services
- For some processes, optimal operation is not at a steady state (Angeli 2011)
- Promise: Truly optimal operation
  - But that is hard/impossible to deliver

NTNU
Norwegian University of Science and Technology

3. Steady-state optimization using Transient Measurements – Hybrid RTO

IFAC DYCOPS Pre-symposium workshop

Dinesh Krishnamoorthy

Krishnamoorthy, D., Foss, B. and Skogestad, S., 2018. Steady-state real-time optimization using transient measurements. *Computers & Chemical Engineering*, *115*, pp.34-45.

Why is traditional static RTO not commonly used?

1. Cost of developing and updating the model (costly offline model update)
2. Wrong value of model parameters and disturbances (slow online model update)
3. Not robust, including computational issues
4. Frequent grade changes make steady-state optimization less relevant
5. Dynamic limitations, including infeasibility due to (dynamic) constraint violation
6. Incorrect model structure

Traditional Steady-state RTO

Step 1: Static Estimation

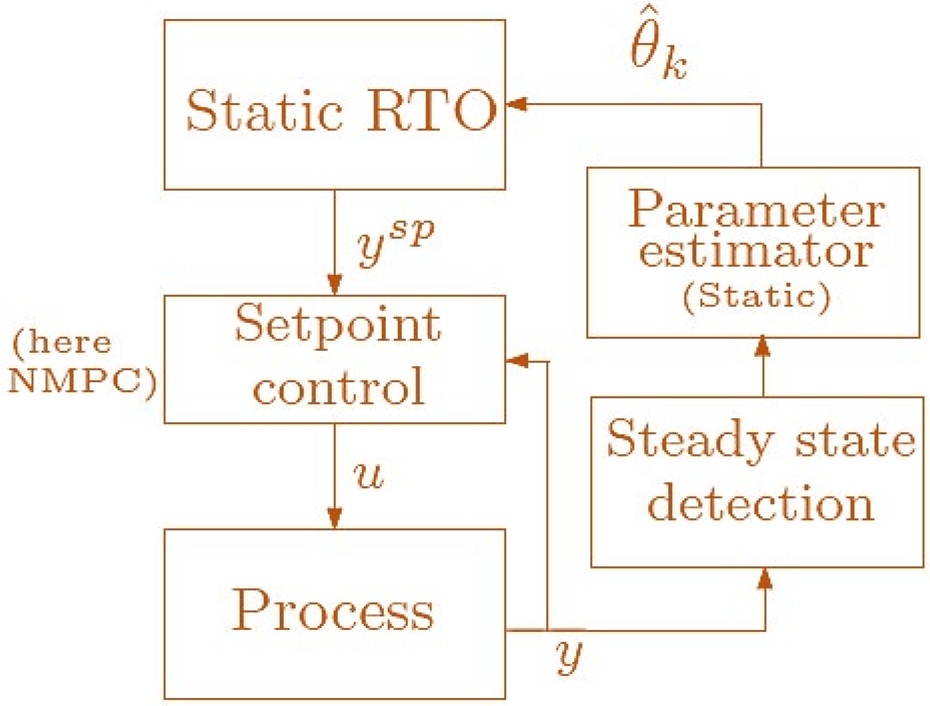
$$\hat{\theta}_k = \arg\min_{\theta} \|y_{meas} - f_{ss}(u_k, \theta)\|_2^2$$

Step 2: Static Optimization

$$u^*_{k+1} = \arg\min_{u} J(y, u)$$

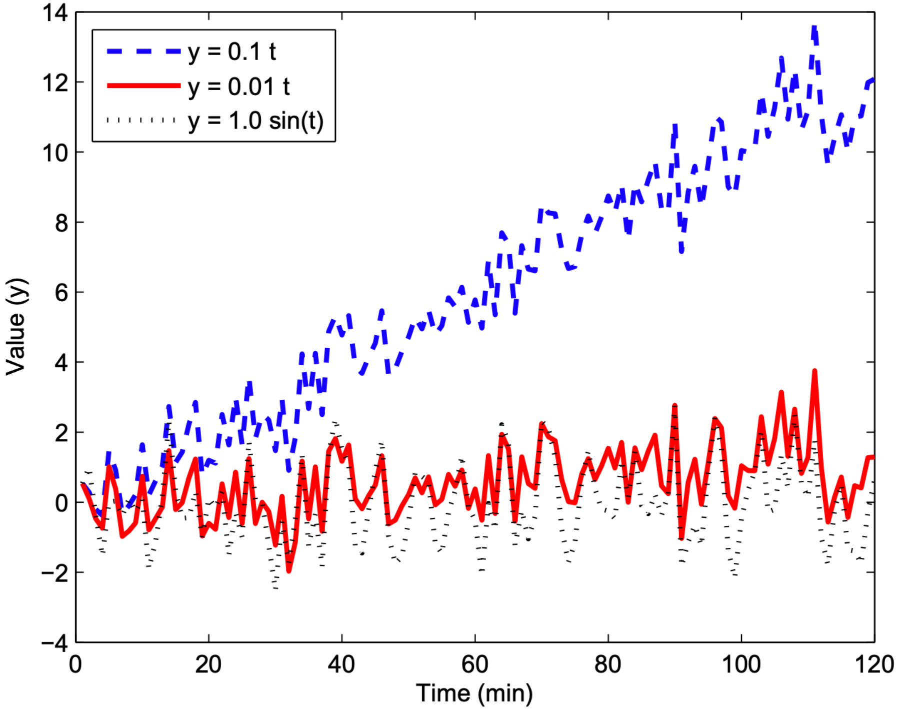
$$\text{s.t. } y = f_{ss}(u, \hat{\theta}_k)$$

$$g(y, u) \leq 0$$

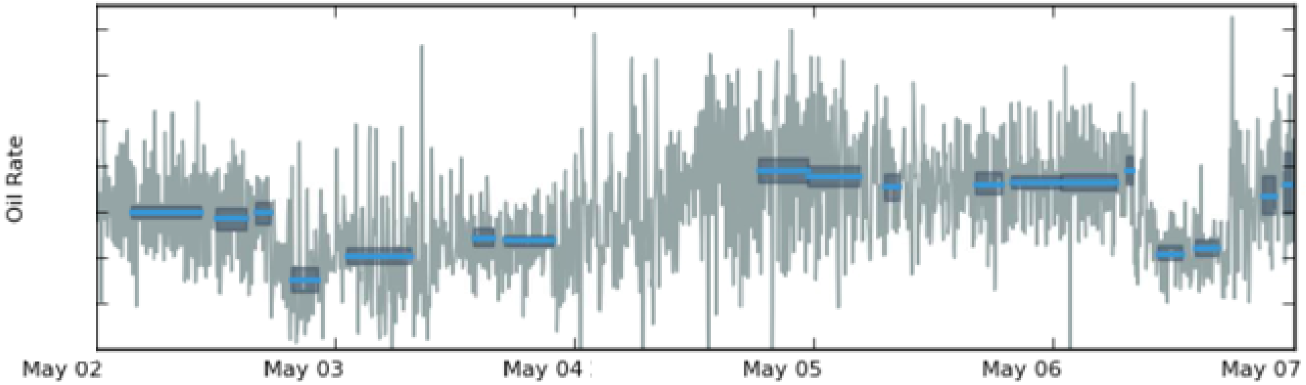


Steady-state wait time

- Transient measurements cannot be used
- Large chunks of data discarded
- Steady state detection issues
  - Erraneously accept transient data
  - Non-stationary drifts



Source: Kelly, J.D. and Hedengren, J.D., 2013. A steady-state detection (SSD) algorithm to detect non-stationary drifts in processes. *Journal of Process Control*, *23*(3), pp.326-331.



Steady-state wait time

- Based on statistical tests, e.g:

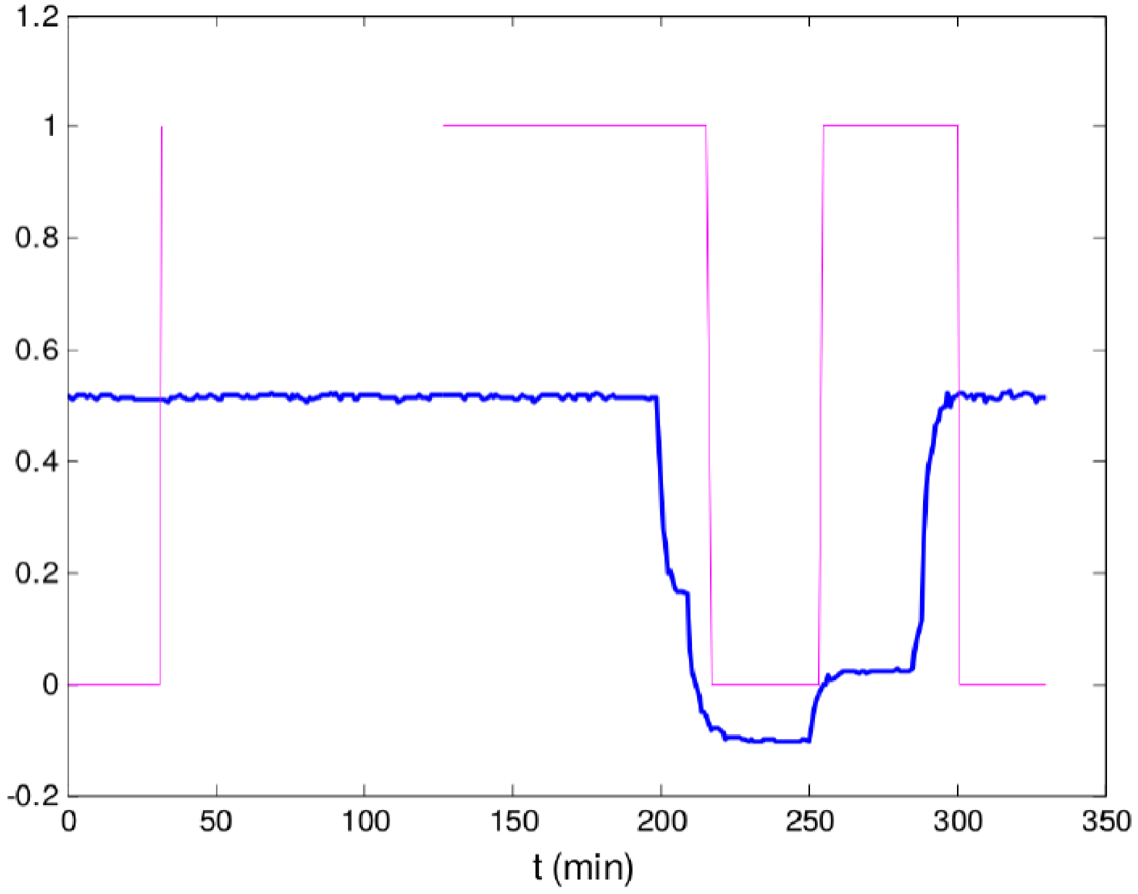
$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{X})^2$$

$$s_d^2 = \frac{1}{n-1}\sum_{i=2}^{n}(x_i - x_{i-1})^2$$

$$R = \frac{s_d^2}{s^2}$$

- In practice - some heuristics

$$R = \frac{\max(s_d^2, \tau_{SM})}{s^2}$$



Source: Câmara MM, Quelhas AD, Pinto JC. *Performance Evaluation of Real Industrial RTO Systems*. Processes. 2016, 4(4).

How to address steady-state wait time?

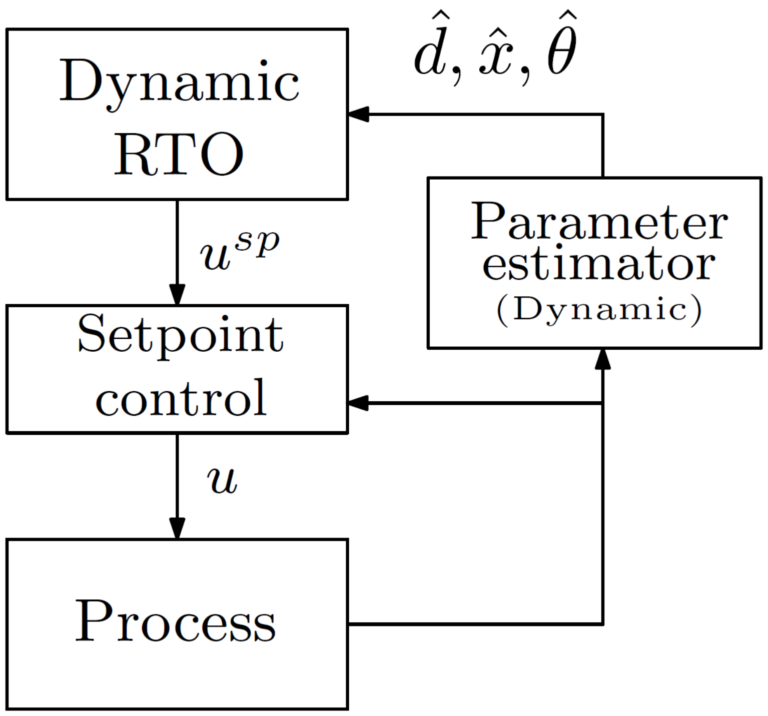
- <u>OBVIOUS: DYNAMIC RTO</u>

Step 1: Dynamic Estimation

$$\hat{\theta}_k = \arg\min_{\theta} \|y_{meas,k} - h(x_k, u_k)\| \quad (5)$$
$$s.t.\ \ x_k = f(x_{k-1}, u_{k-1}, \theta)$$

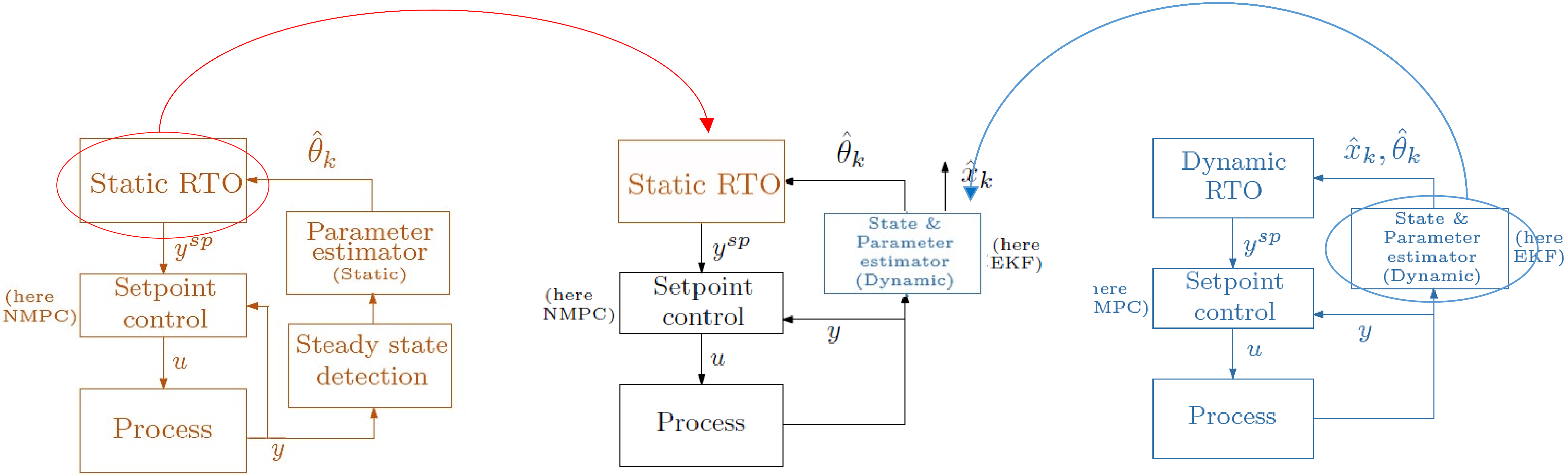
Step 2: Dynamic Optimization

$$u_t^* = \arg\min_{u_t} \sum_{t=k}^{k+T} J(y_t, u_t) \quad (6)$$
$$\begin{aligned} s.t.\ \ x_{t+1} &= f(x_t, u_t, \hat{\theta}_k) \\ y_t &= h(x_t, u_t) \\ g(y_t, u_t) &\le 0 \\ x_k &= \hat{x}_k \qquad \forall t \in \{k, \dots, k+T\} \end{aligned}$$



Dynamic RTO has problems – especially the optimization part

Hybrid RTO



Dynamic Estimation

+

Static Optimization

Krishnamoorthy, D., Foss, B. and Skogestad, S., 2018. Steady-State Real-time Optimization using Transient Measurements. Computers and Chemical Engineering, Vol 115, p.34-45.

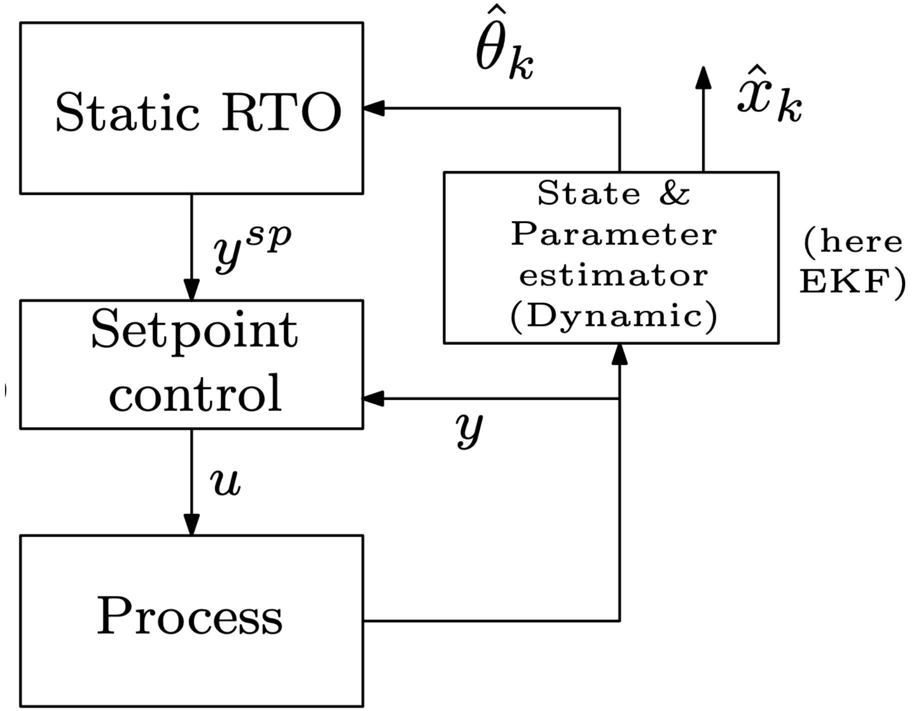
Hybrid RTO

Step 1: Dynamic Estimation

$$\hat{\theta}_k = \arg\min_{\theta} \|y_{meas,k} - h(x_k, u_k)\|$$
$$s.t.\ \ x_k = f(x_{k-1}, u_{k-1}, \theta)$$

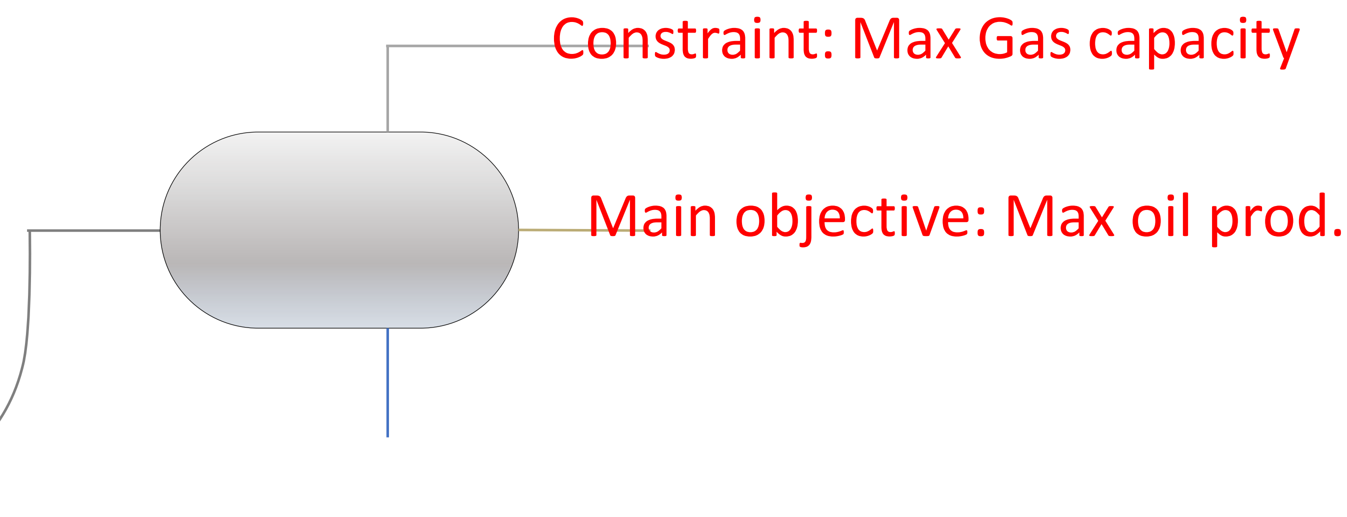
Step 2: Static Optimization

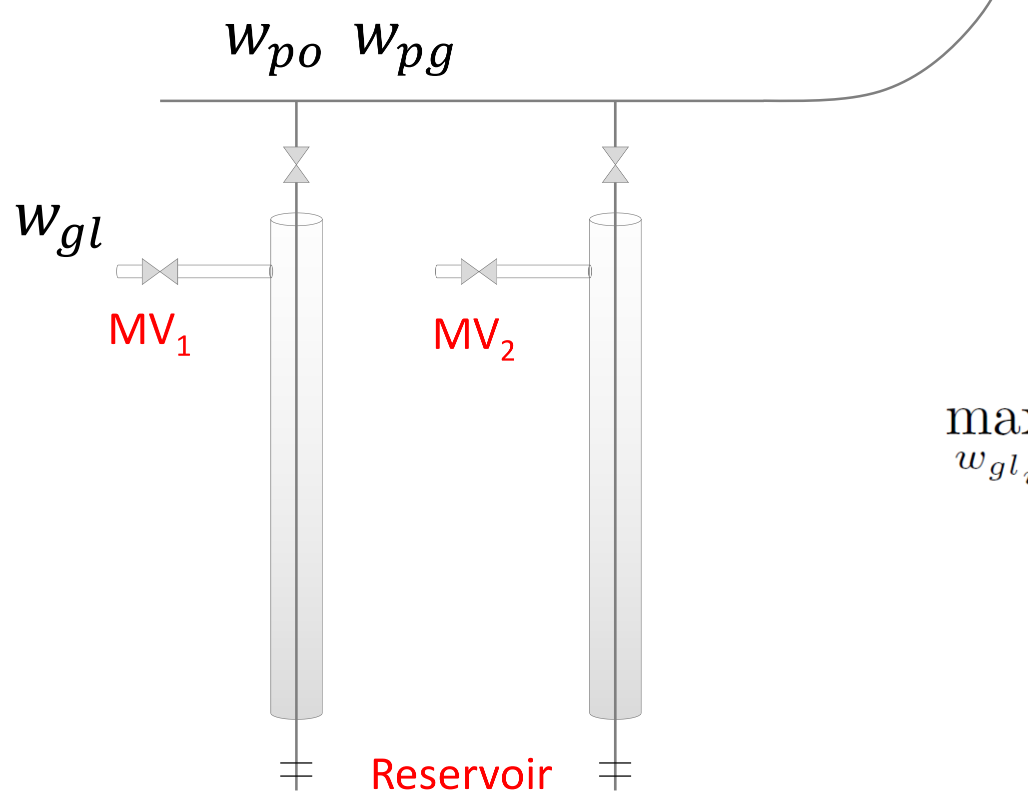
$$u^*_{k+1} = \arg\min_{u} J(y,u)$$
$$s.t.\ \ y = f_{ss}(u, \hat{\theta}_k)$$
$$g(y,u) \leq 0$$



Krishanmoorthy, Foss, Skogestad, Comput & Chem Eng (2018) – *Hybrid RTO*
Matias & Le Roux, J. Proc. Control (2018) – *ROPA*
Valluru & Patwardhan, Ind. Eng. Chem. Res (2019) – *Frequent RTO*

CASE STUDY: Gas lift





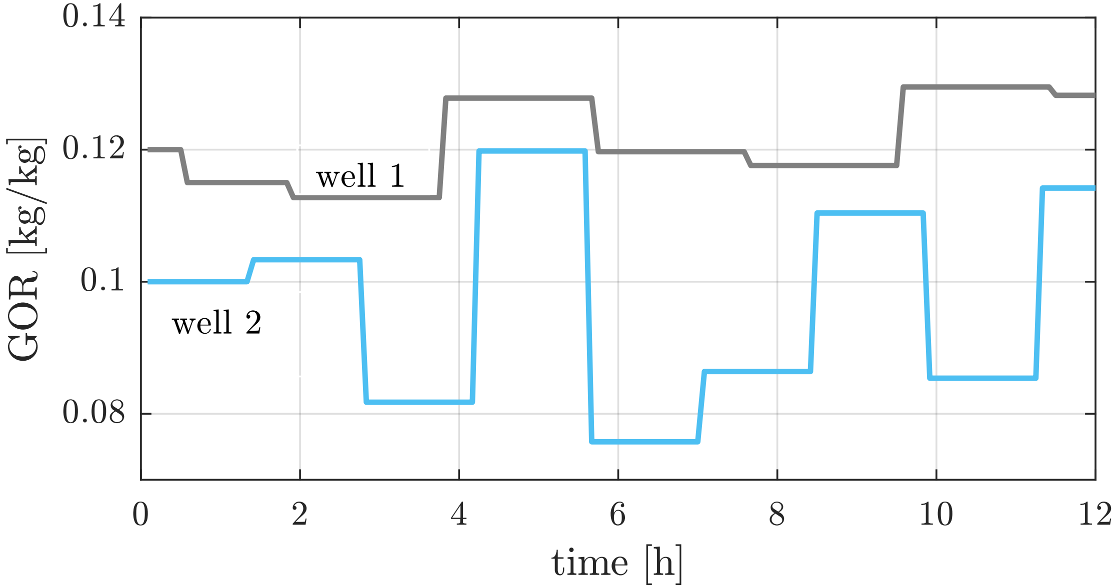
GOR = Gas/Oil ratio in feed (reservoir)

$$\max_{w_{gl_i}} \; J' = \left(\$_o \sum_{i \in \mathcal{N}} w_{po_i} - \$_{gl} \sum_{i \in \mathcal{N}} w_{gl_i} \right) - \$_{fl} \| w_{fl} \|$$

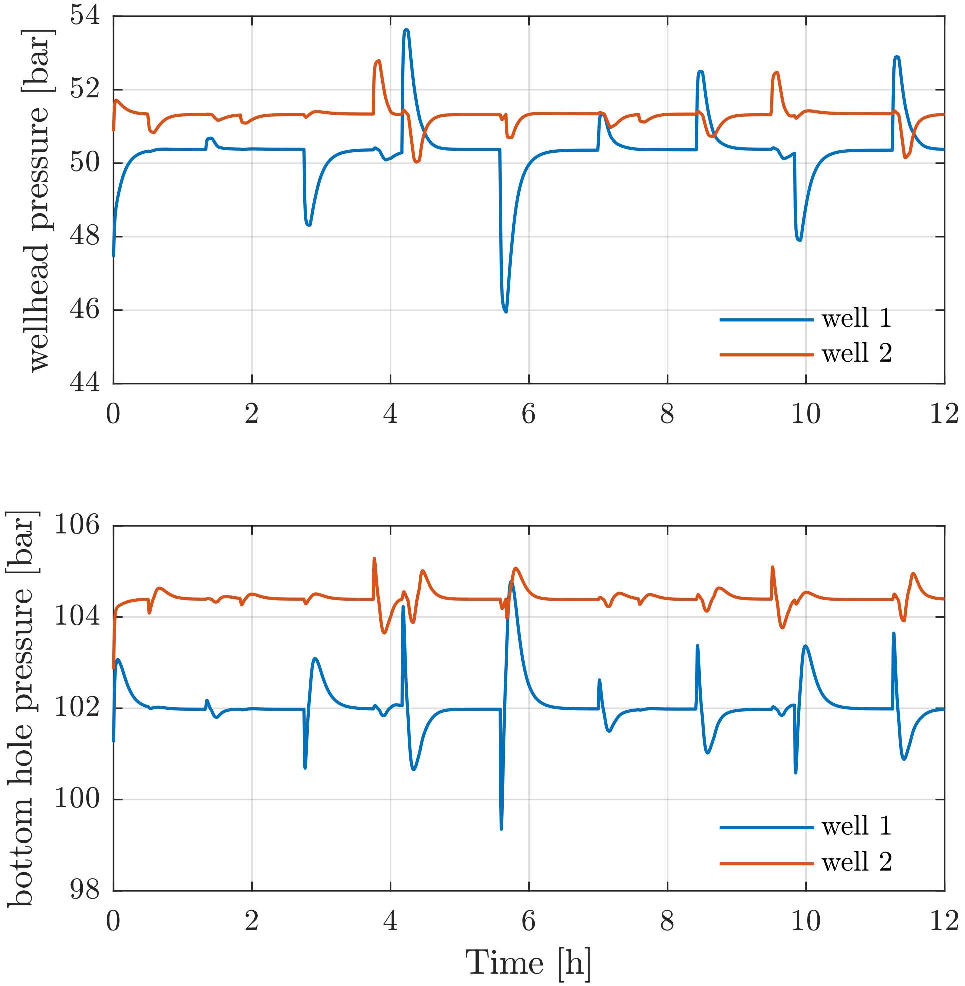
$$\text{s.t.} \sum_{i \in \mathcal{N}} w_{pg_i} \leq w_{g_{max}} + w_{fl}$$

Main disturbance (d): GOR variation

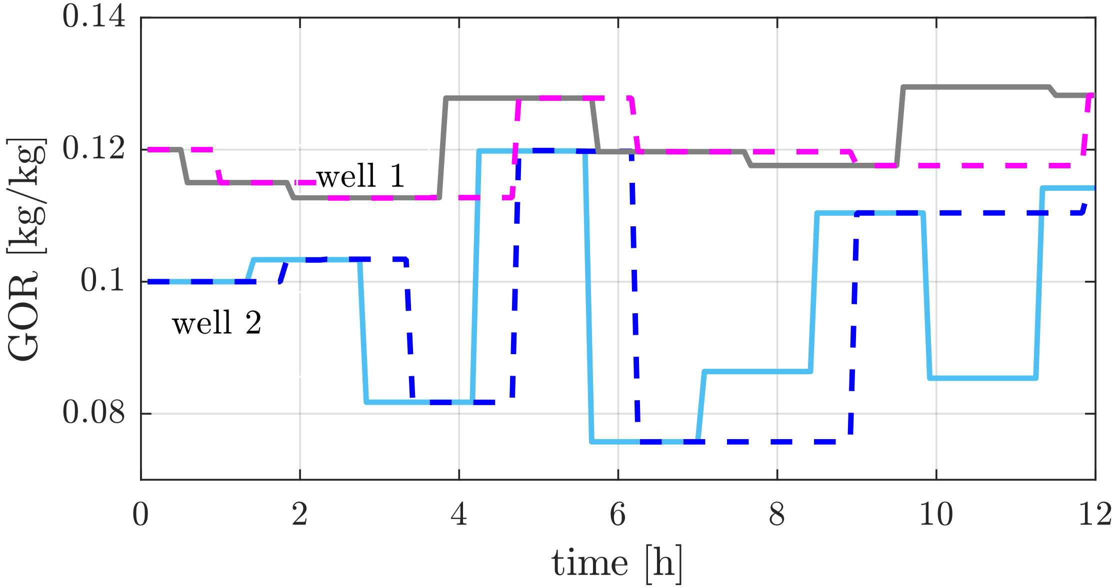
Disturbance: GOR variation



Typical measured data (pressures and flowrates)



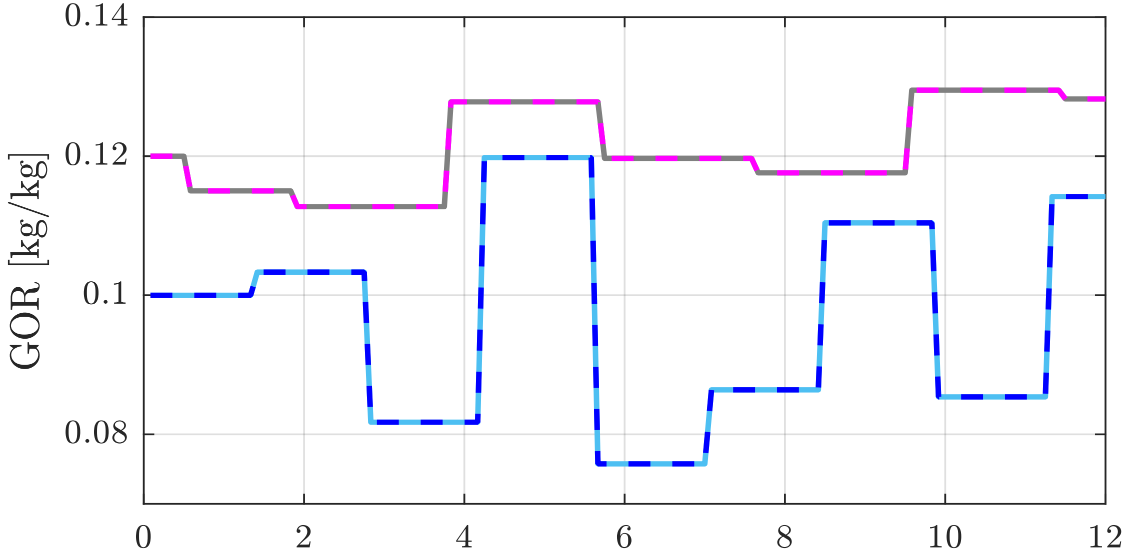
GOR estimation - using “data reconciliation” (traditional static RTO)



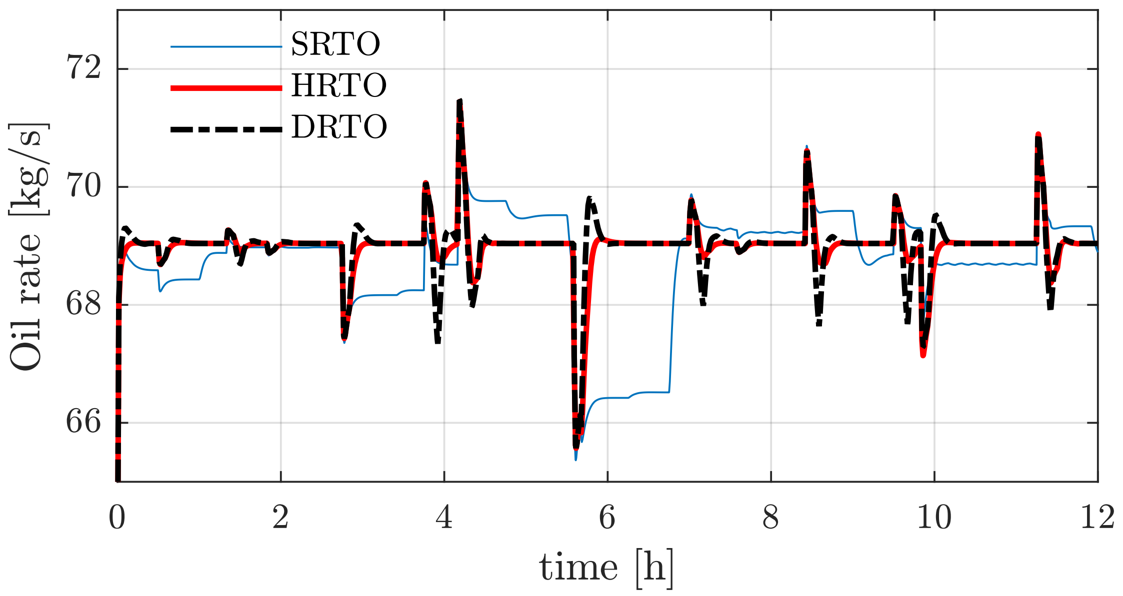
Problem: Steady-state wait time for data reconciliation

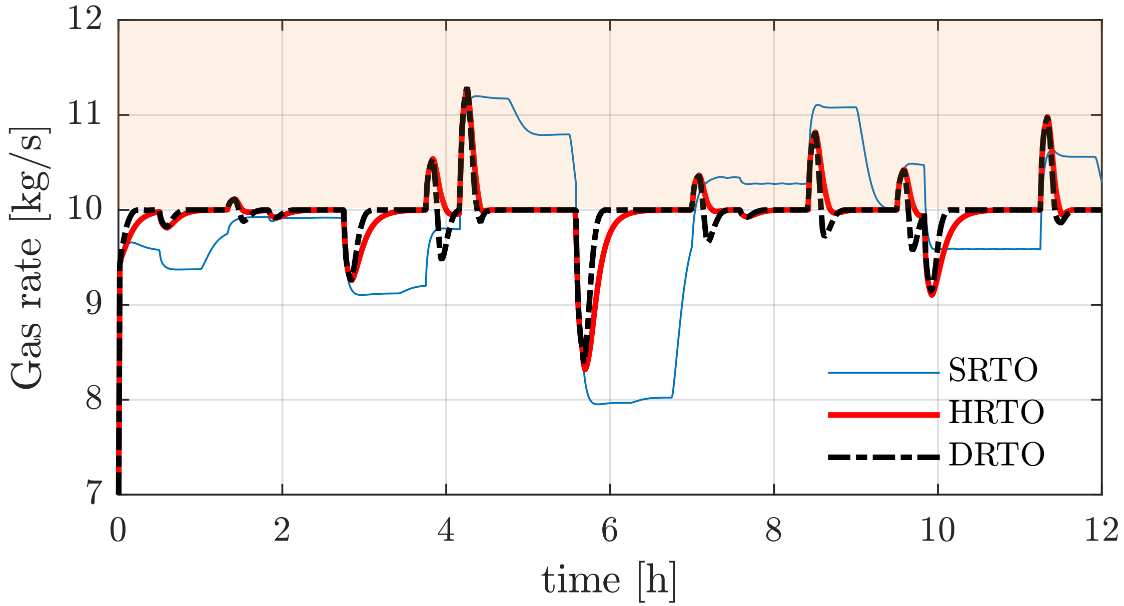
GOR estimation – using extended Kalman filter (DRTO & HRTO)

NTNU



Oil and gas rates

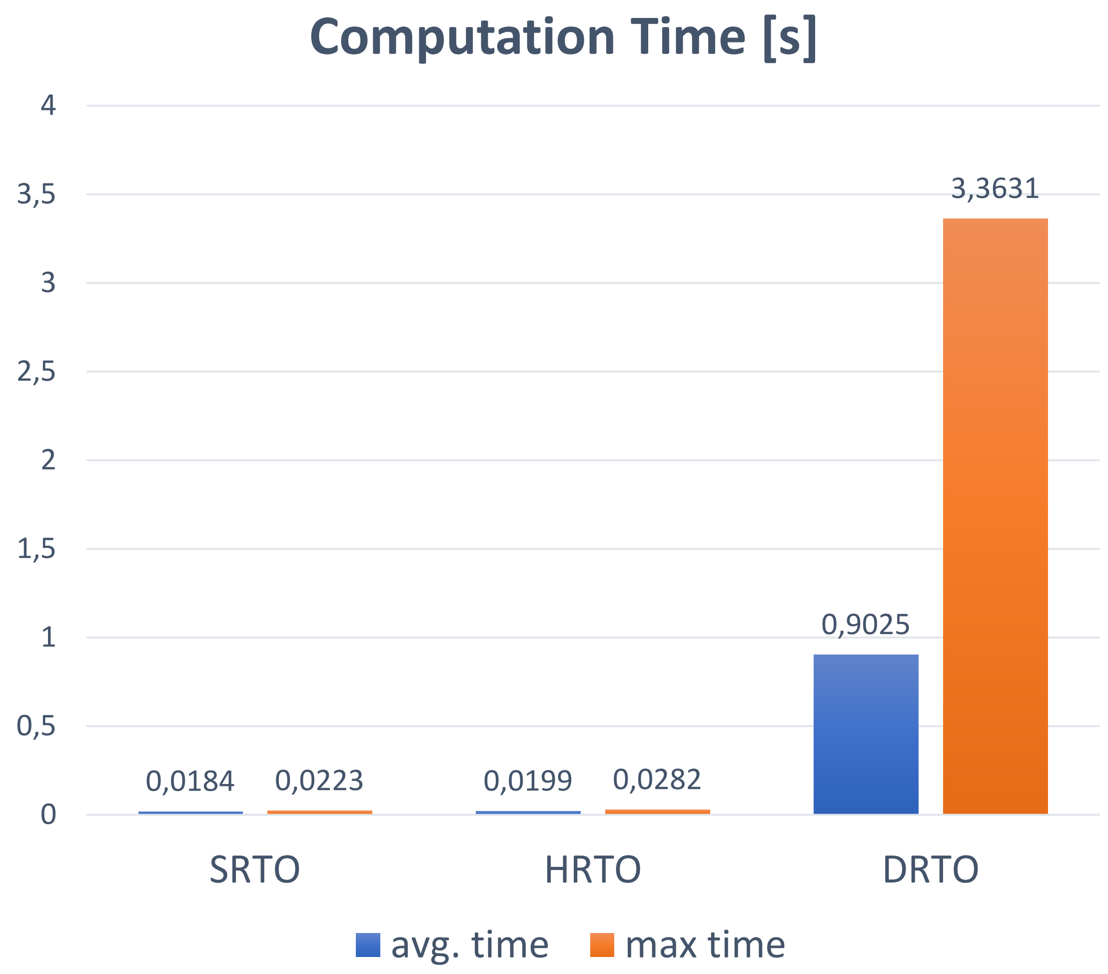


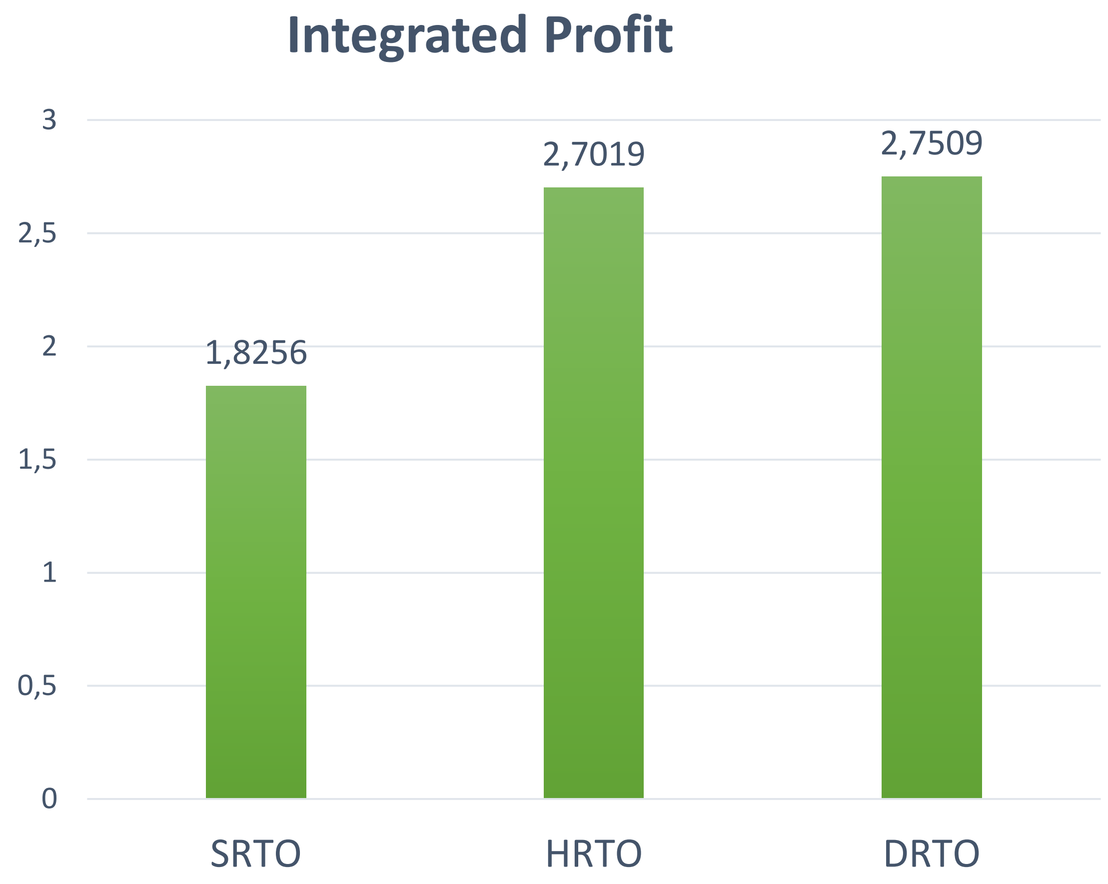


SRTO = traditional static RTO
HRTO = hybrid RTO
DRTO = dynamic RTO

NTNU

Results





Advantage of steady-state optimization (SRTO & HRTO)

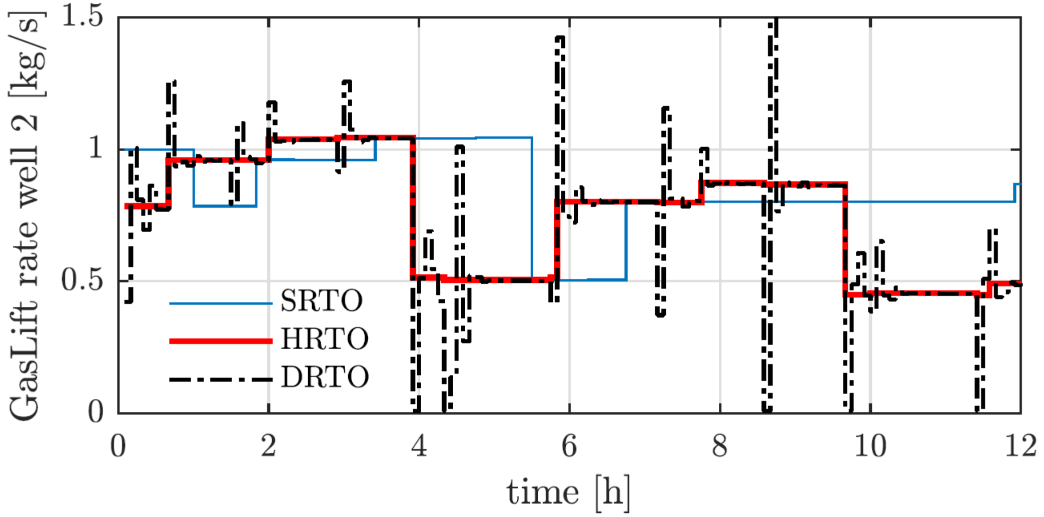
- Computation time & numerical robustness
- Avoids causality issue / index problems
- Allows optimization on decision variables other than the MVs
  - Simplifies the optimization
  - Slower time scale (choose slow varying variables as decision variables)

Why is traditional static RTO not commonly used?

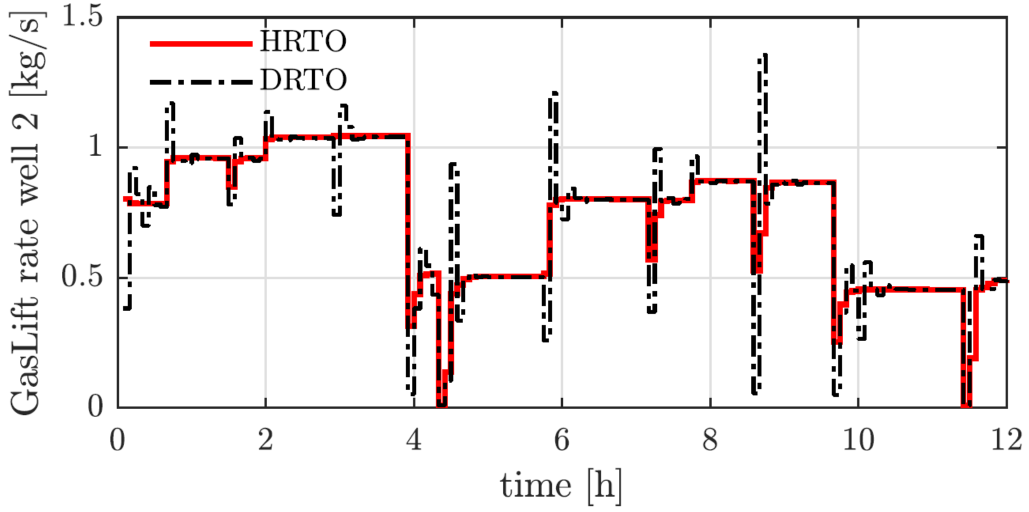
1. Cost of developing and updating the model structure (costly offline model update)
2. Wrong value of model parameters and disturbances (slow online model update)
3. Not robust, including computational issues
4. Frequent grade changes make steady-state optimization less relevant
5. Dynamic limitations, including infeasibility due to (dynamic) constraint violation
6. Incorrect model structure

Dynamic limitations – not a big issue

MV2: Setpoint provided to tracking controller

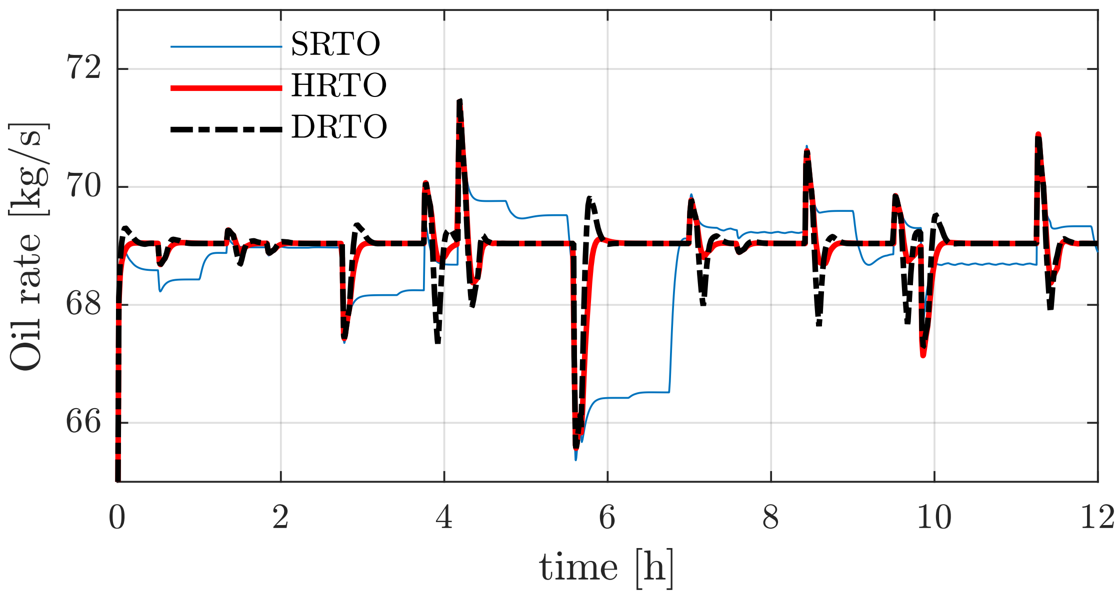


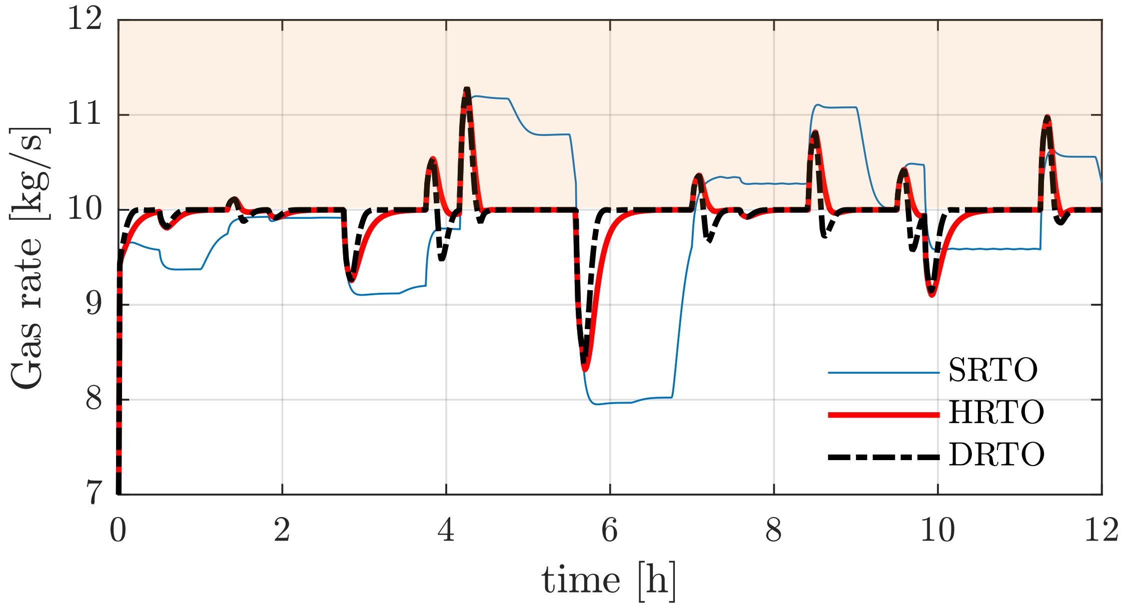
Actual MV move by setpoint tracking controller



SRTO = traditional static RTO
HRTO = hybrid RTO
DRTO = dynamic RTO

Oil and gas rates





SRTO = traditional static RTO
HRTO = hybrid RTO
DRTO = dynamic RTO

NTNU
Norwegian University of Science and Technology

4. Feedback RTO based on novel steady-state gradient estimation method

IFAC DYCOPS Pre-symposium workshop

Dinesh Krishnamoorthy

Krishnamoorthy, D., Jahanshahi, E. and Skogestad, S., 2019. Steady-state real-time optimization using transient measurements. *Industrial and Engineering Chemistry Research 115*, pp.34-45.

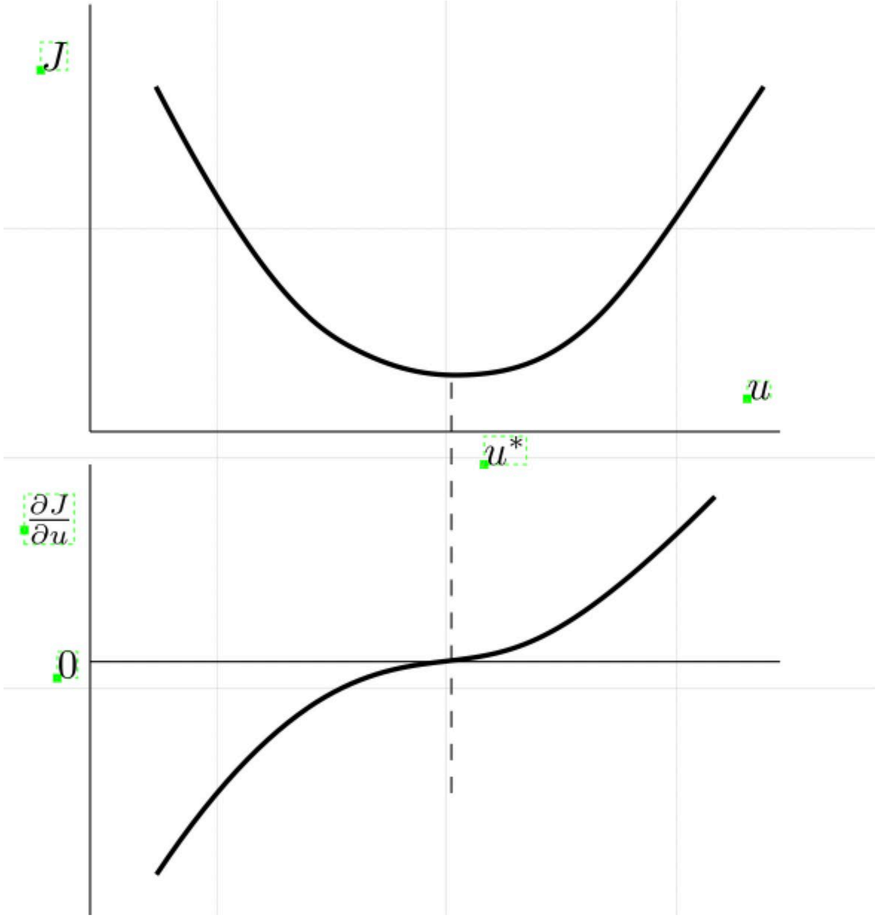
Why is traditional static RTO not commonly used?

1. Cost of developing and updating the model structure (costly offline model update)
2. Wrong value of model parameters and disturbances (slow online model update)
3. Not robust, including computational issues
4. Frequent grade changes make steady-state optimization less relevant
5. Dynamic limitations, including infeasibility due to (dynamic) constraint violation
6. Incorrect model structure

Necessary condition of optimality

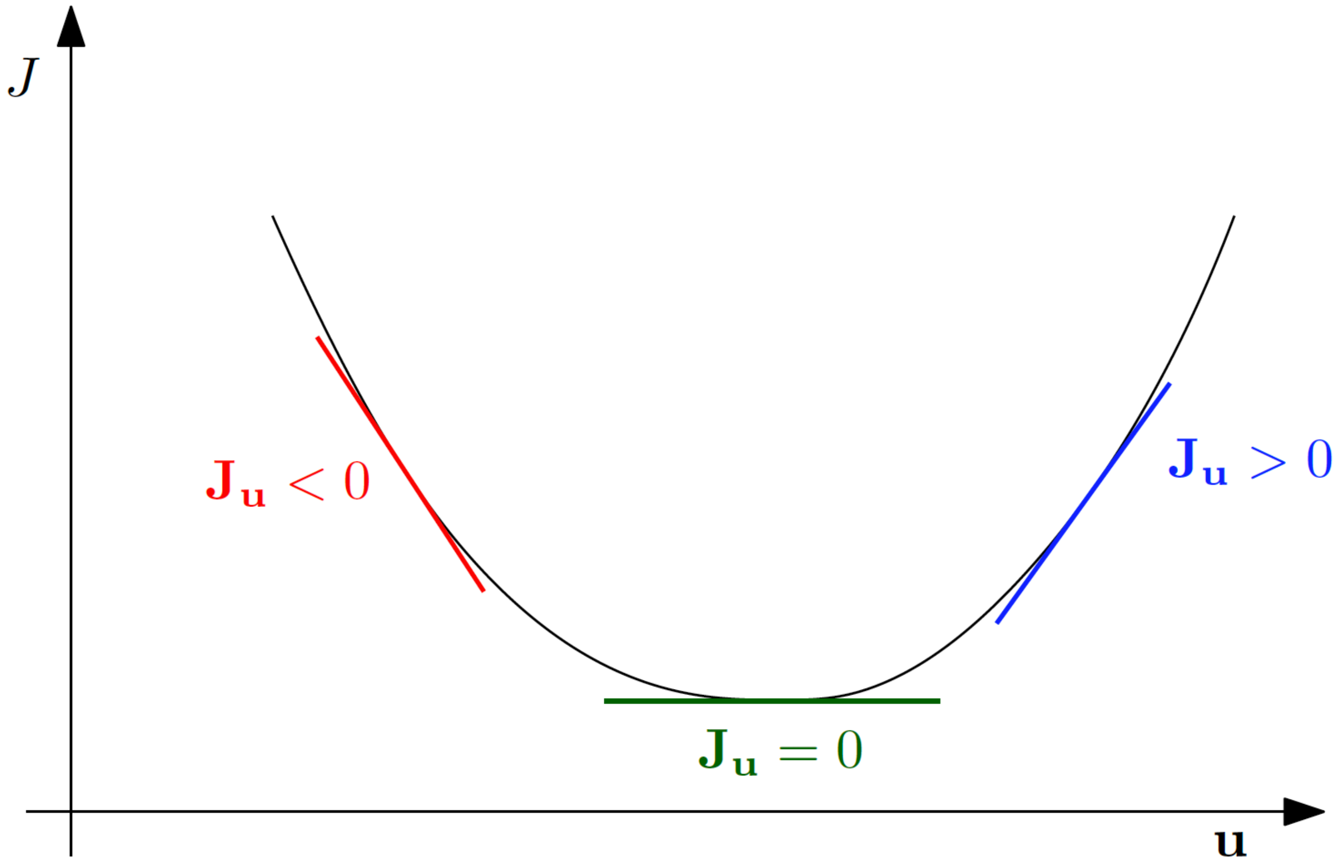
$$\frac{\partial J}{\partial \mathbf{u}}(\mathbf{u}^*, \mathbf{d}) = \mathbf{J_u}(\mathbf{u}^*, \mathbf{d}) = 0$$

- The ideal controlled variable is the gradient
- May use simple feedback controller to control the gradient to constant setpoint of zero.



Problem: We do not usually have gradients as measurements

Feedback RTO

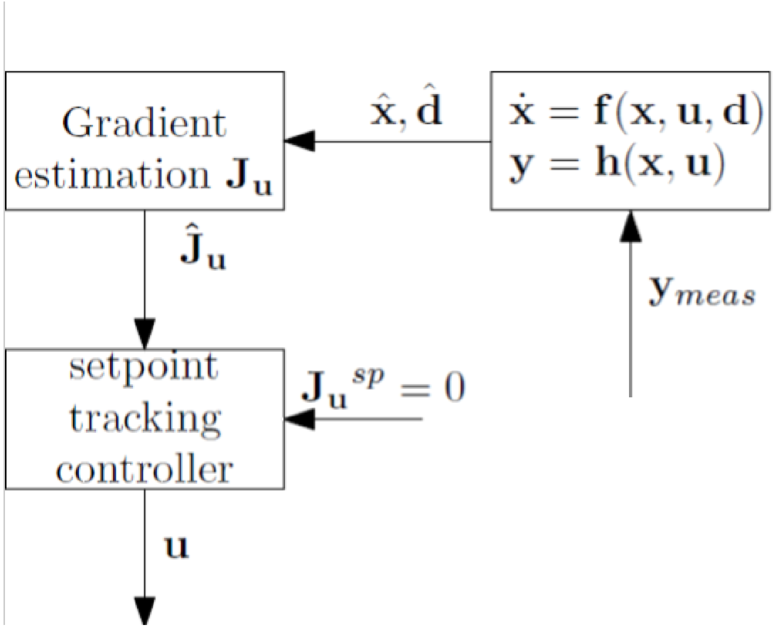


Krishnamoorthy, D., Jahanshahi, E. and Skogestad, S., 2018. A feedback real-time optimization strategy, Ind. Eng. Res. Chem

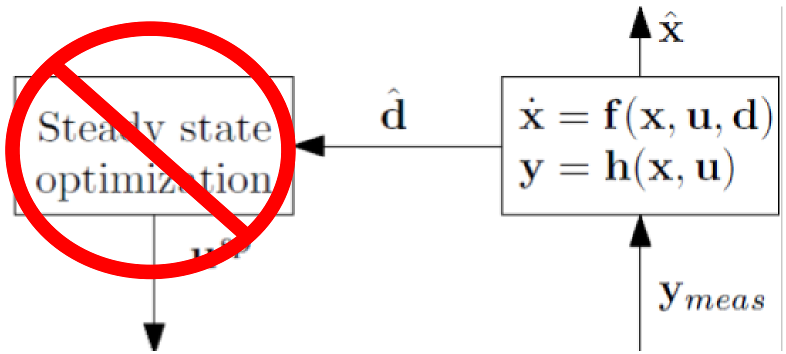
Feedback RTO: Replace steady-state optimization by feedback control

NTNU

Feedback RTO



Hybrid RTO



Krishnamoorthy, D., Jahanshahi, E. and Skogestad, S., 2018. A feedback real-time optimization strategy, Ind. Eng. Res. Chem (submitted).

Feedback RTO

- Step 1 - Linearize the dynamic model

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x}, \mathbf{u}, \mathbf{d}) \\ J &= \mathbf{g}(\mathbf{x}, \mathbf{u}) \end{aligned} \quad \Rightarrow \quad \begin{aligned} \dot{\mathbf{x}} &= A\mathbf{x} + B\mathbf{u} \\ J &= C\mathbf{x} + D\mathbf{u} \end{aligned}$$

$$A = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}} \quad B = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{u}}\right|_{\mathbf{x}=\hat{\mathbf{x}}}$$

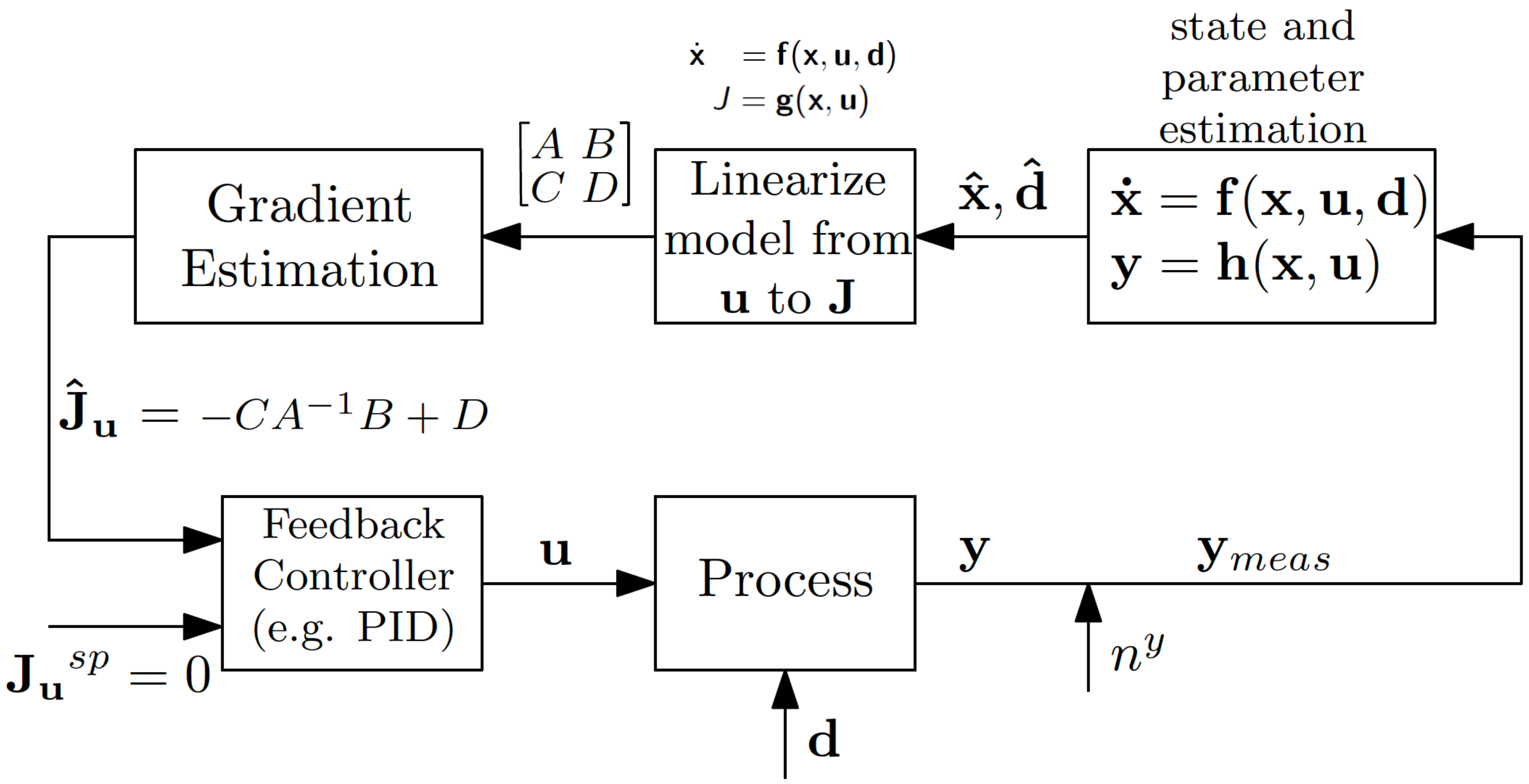
$$C = \left.\frac{\partial \mathbf{g}}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}} \quad D = \left.\frac{\partial \mathbf{g}}{\partial \mathbf{u}}\right|_{\mathbf{x}=\hat{\mathbf{x}}}$$

- Step 2 - At steady-state $\dot{x} = 0$

$$J = \underbrace{\left(-CA^{-1}B + D\right)}_{\hat{\mathbf{J}}_{\mathbf{u}}} \mathbf{u}$$

Krishnamoorthy, D., Jahanshahi, E. and Skogestad, S., 2018. A feedback real-time optimization strategy, Ind. Eng. Res. Chem (submitted).

Feedback RTO



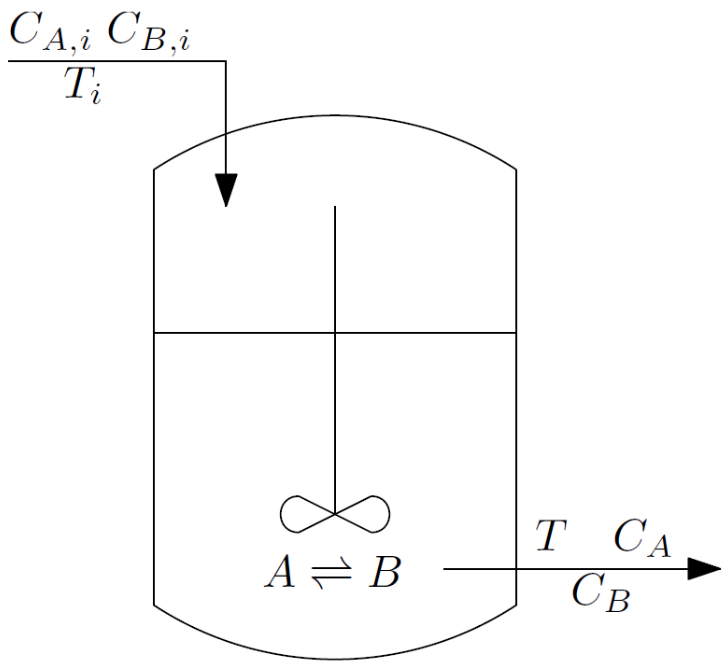
Krishnamoorthy, D., Jahanshahi, E. and Skogestad, S., 2018. A feedback real-time optimization strategy, Ind. Eng. Res. Chem (submitted).

CSTR case study

$$A \rightleftharpoons B$$

$$\mathbf{x} = [C_A \quad C_B \quad T]^\mathsf{T}$$
$$\mathbf{u} = T_i$$
$$\mathbf{d} = [C_{A_{in}} \quad C_{B_{in}}]^\mathsf{T}$$

$$J = -p_{c_B} C_B + (p_{T_{in}} T_{in})^2$$

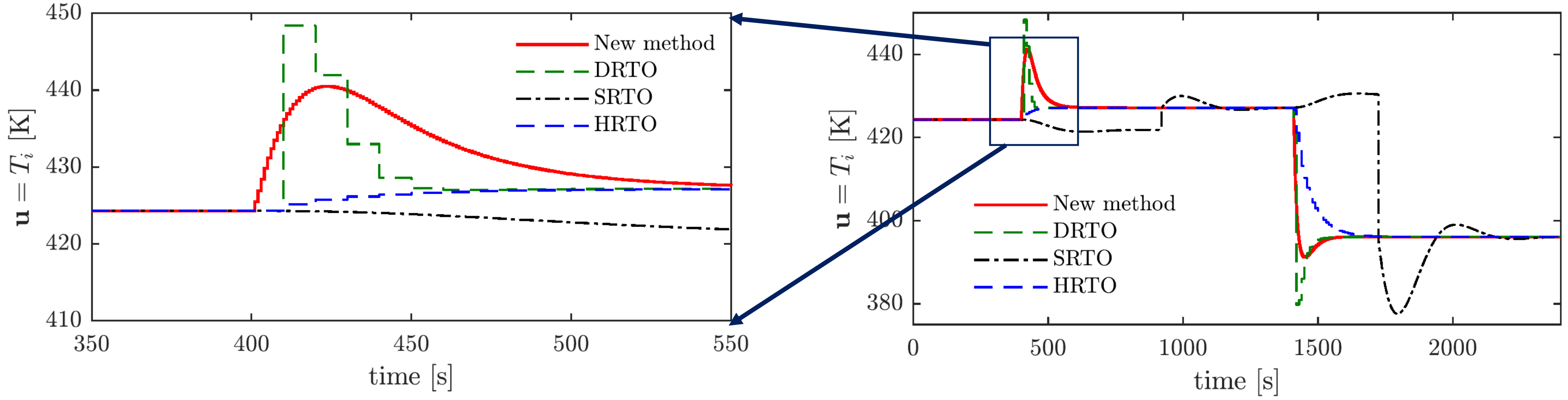


- Economou, C. G.; Morari, M.; Palsson, B. O. Internal model control: Extension to nonlinear system. Industrial & Engineering Chemistry Process Design and Development 1986, 25, 403–411.
- Ye, L.; Cao, Y.; Li, Y.; Song, Z. Approximating Necessary Conditions of Optimality as Controlled Variables. Industrial & Engineering Chemistry Research 2013, 52, 798–808.

Comparison of RTO approaches: MV

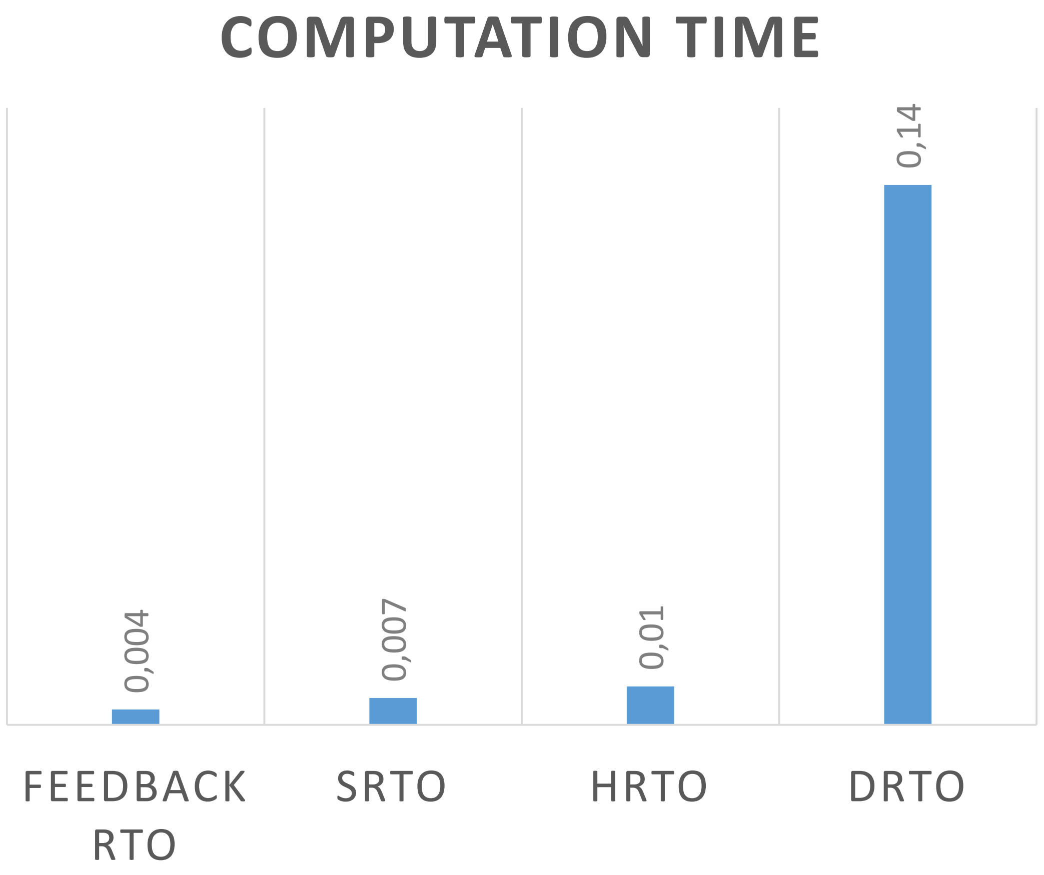
SRTO = traditional static RTO
HRTO = hybrid RTO
DRTO = dynamic RTO
New method = Feedback RTO

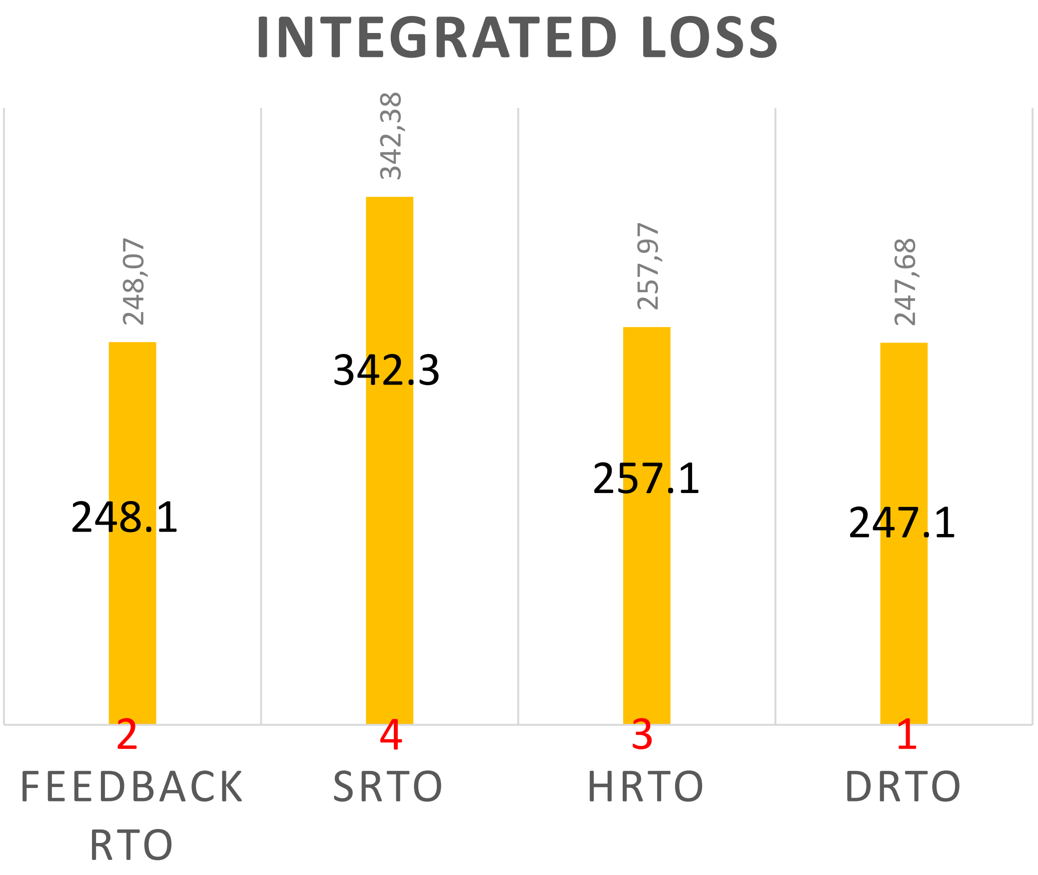
closer look



t = 400 s, d1: Increase C_{Ain}
t = 1400 s, d2: Increase C_{Bin}

Comparison of RTO approaches





Feedback RTO - Other case studies

- Evaporator process[1]
- Gas lift wells[2]
- Ammonia reactor[3]

1. Krishnamoorthy, D., Jahanshahi, E. and Skogestad, S., Control of steady-state gradient for an Evaporator process, PSE Asia (Submitted 2019)
2. Krishnamoorthy, D., Jahanshahi, E. and Skogestad, S., 2018. Gas-lift Optimization by Controlling Marginal Gas-Oil Ratio using Transient Measurements (in-Press), IFAC OOGP, Esbjerg, Denmark
3. Bonnowitz, H., Straus, J., Krishnamoorthy, D., and Skogestad, S., 2018. Control of the Steady-State Gradient of an Ammonia Reactor usingTransient Measurements, Computer aided chemical engineering, Vol.43, p.1111-1116 (ESCAPE 28)

NTNU
Norwegian University of Science and Technology

5. Extremum Seeking Control

IFAC DYCOPS Pre-symposium workshop

Dinesh Krishnamoorthy

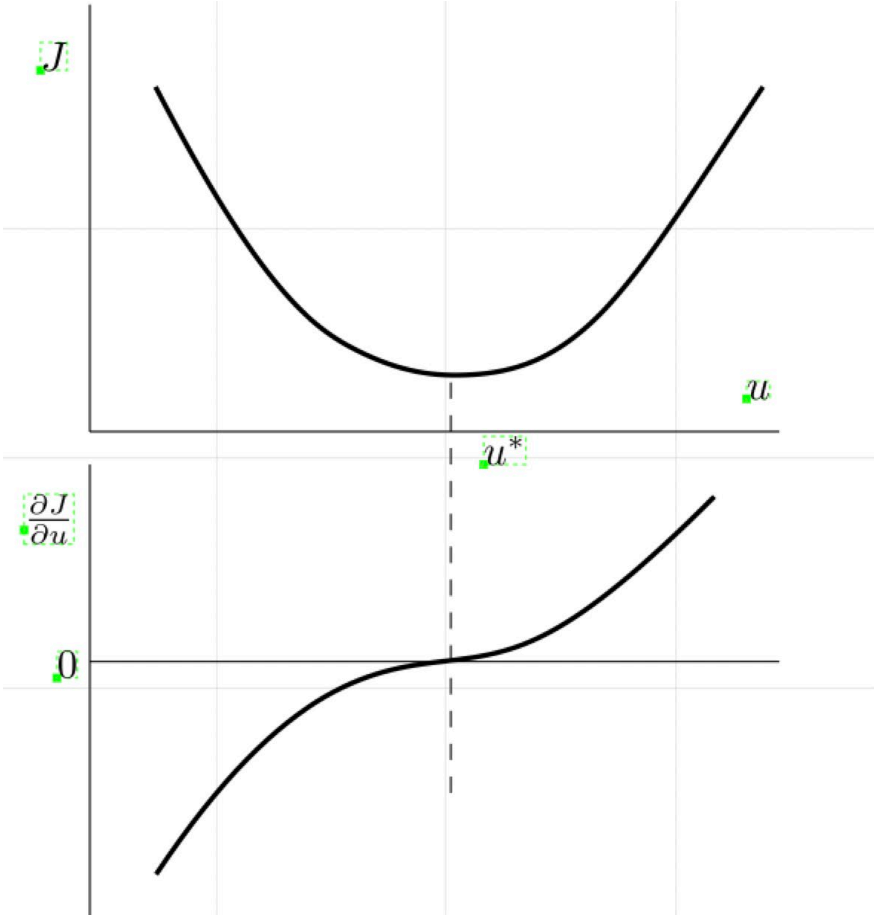
Why is traditional static RTO not commonly used?

1. Cost of developing and updating the model (costly offline model update)
2. Wrong value of model parameters and disturbances (slow online model update)
3. Not robust, including computational issues
4. Frequent grade changes make steady-state optimization less relevant
5. Dynamic limitations, including infeasibility due to (dynamic) constraint violation
6. Incorrect model structure

Necessary condition of optimality

$$\frac{\partial J}{\partial \mathbf{u}}(\mathbf{u}^*, \mathbf{d}) = \mathbf{J_u}(\mathbf{u}^*, \mathbf{d}) = 0$$

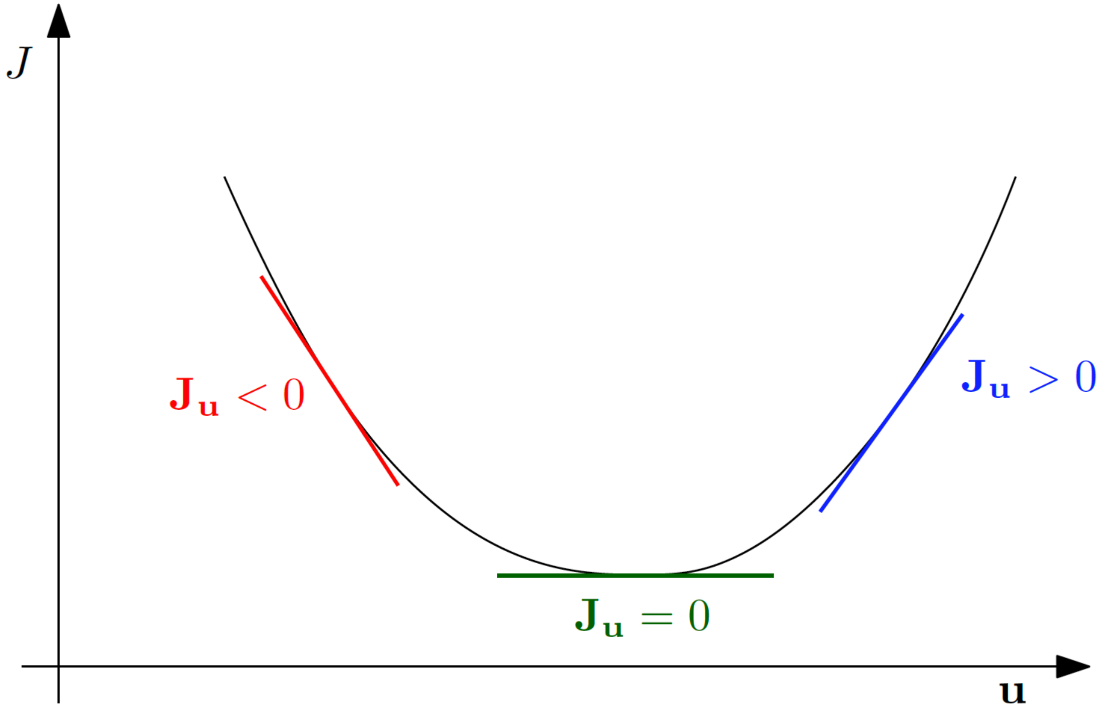
- The ideal controlled variable is the gradient
- May use simple feedback controller to control the gradient to constant setpoint of zero.



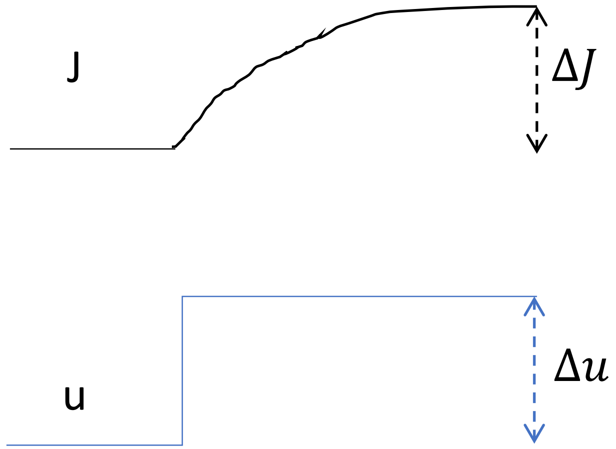
Problem: We do not usually have gradients as measurements

Data-driven method

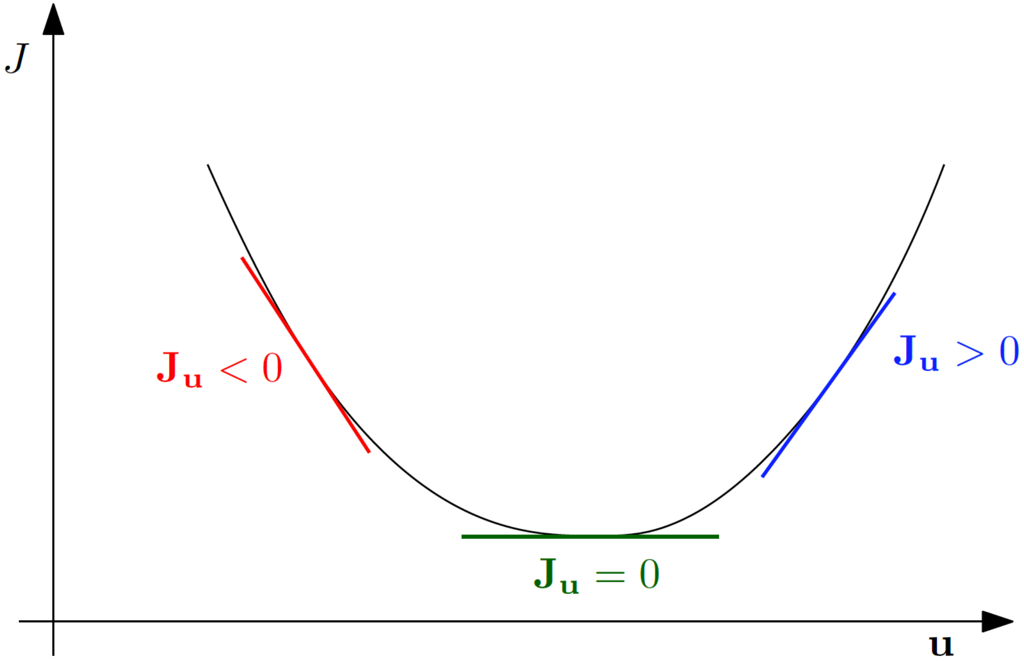
- We do not use a model to estimate the gradient
- Estimate gradient Experimentally
  - NB! Need Cost measurement
- Similar approaches
  - Extremum seeking
  - NCO tracking
  - Hill climbing control
  - Experimental optimization
  - ....
- Difference is in the way gradient is estimated



Steady-state gradient

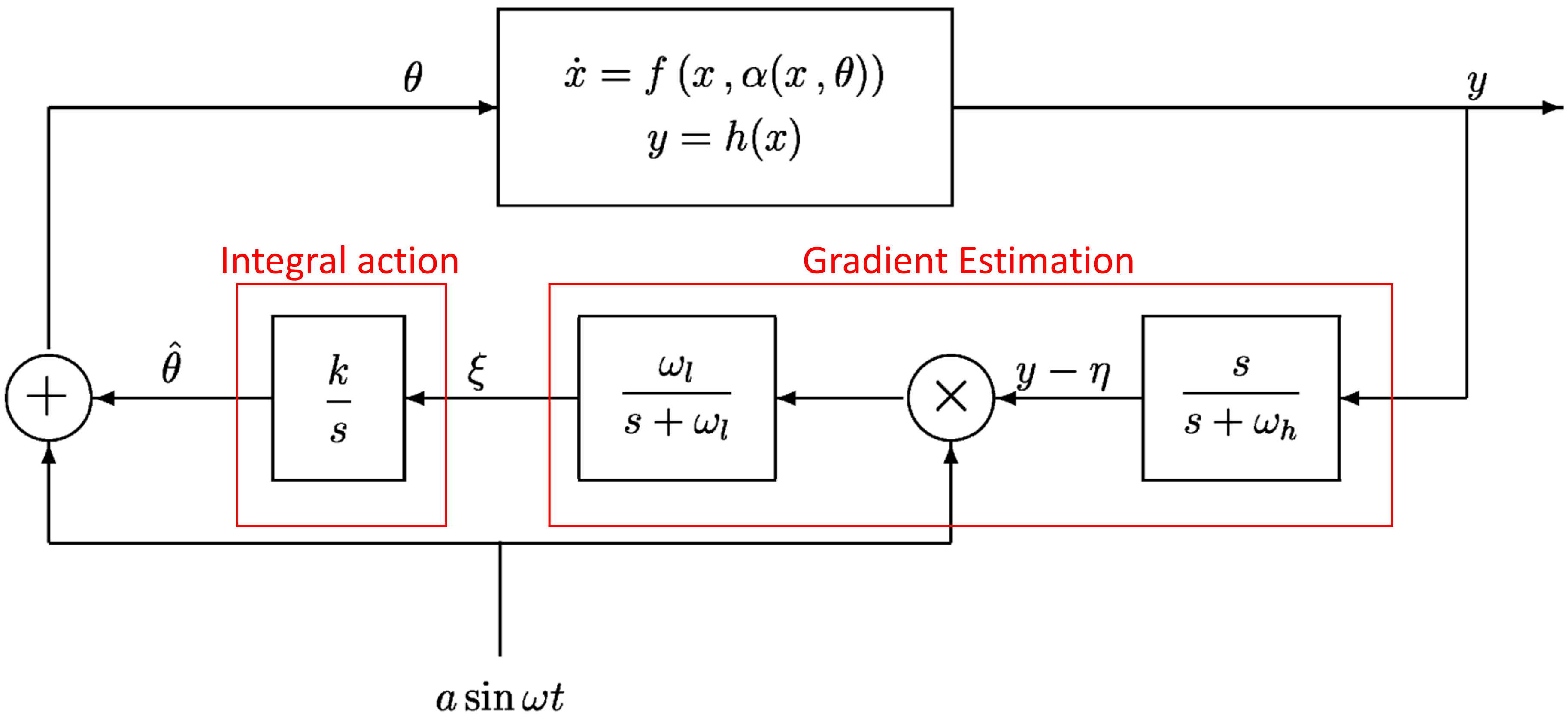


$$J_u = \frac{\Delta J}{\Delta u}$$



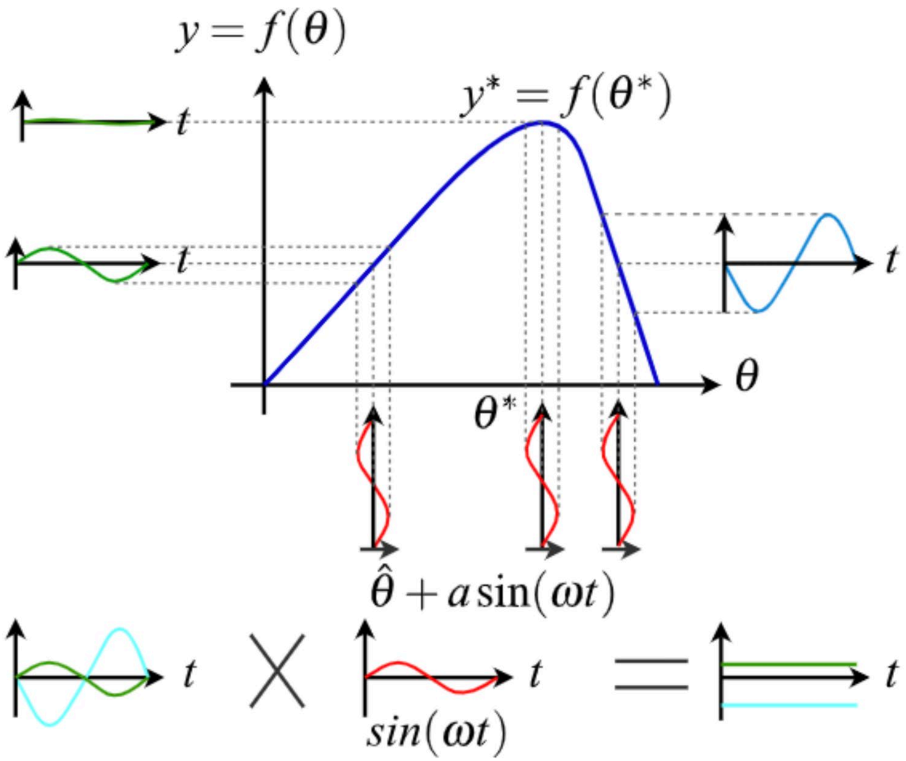
Francois & Bonvin (2007)
Jäschke & Skogestad(2011)

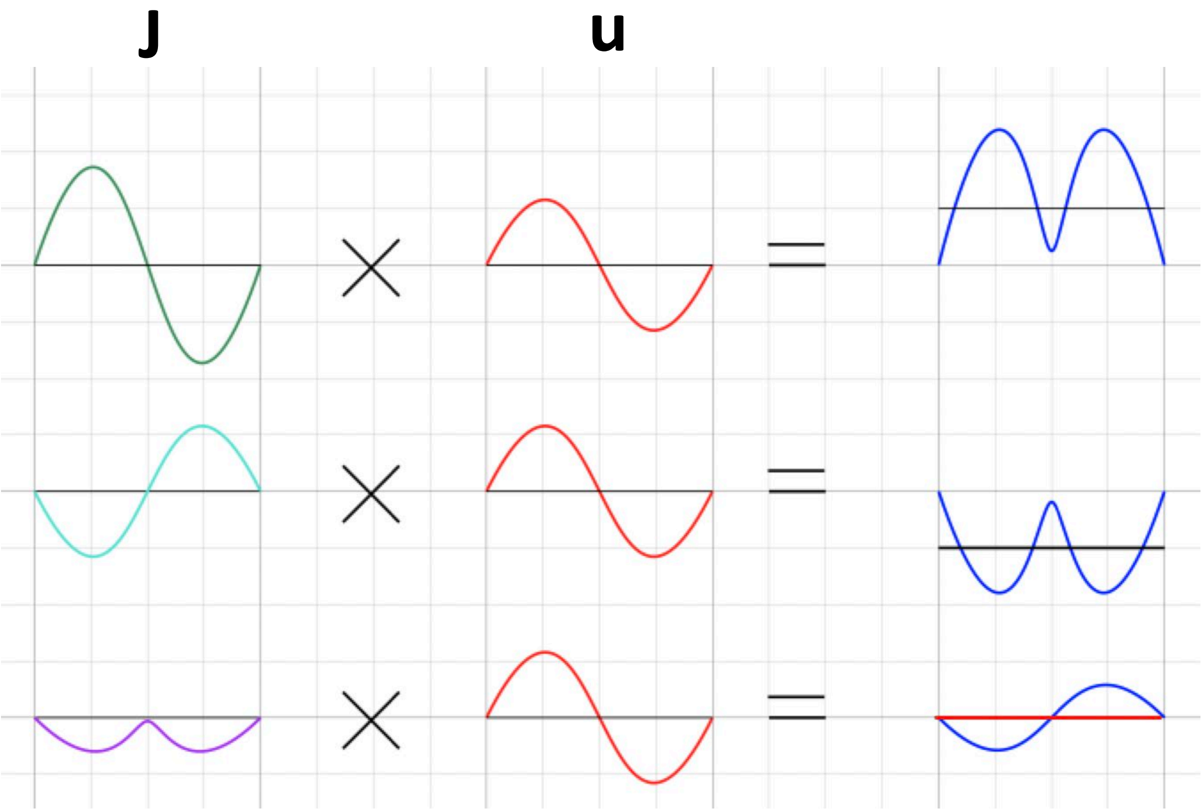
Classical Extremum seeking control



Draper & Li (1951)
Krstic & Wang (2000)

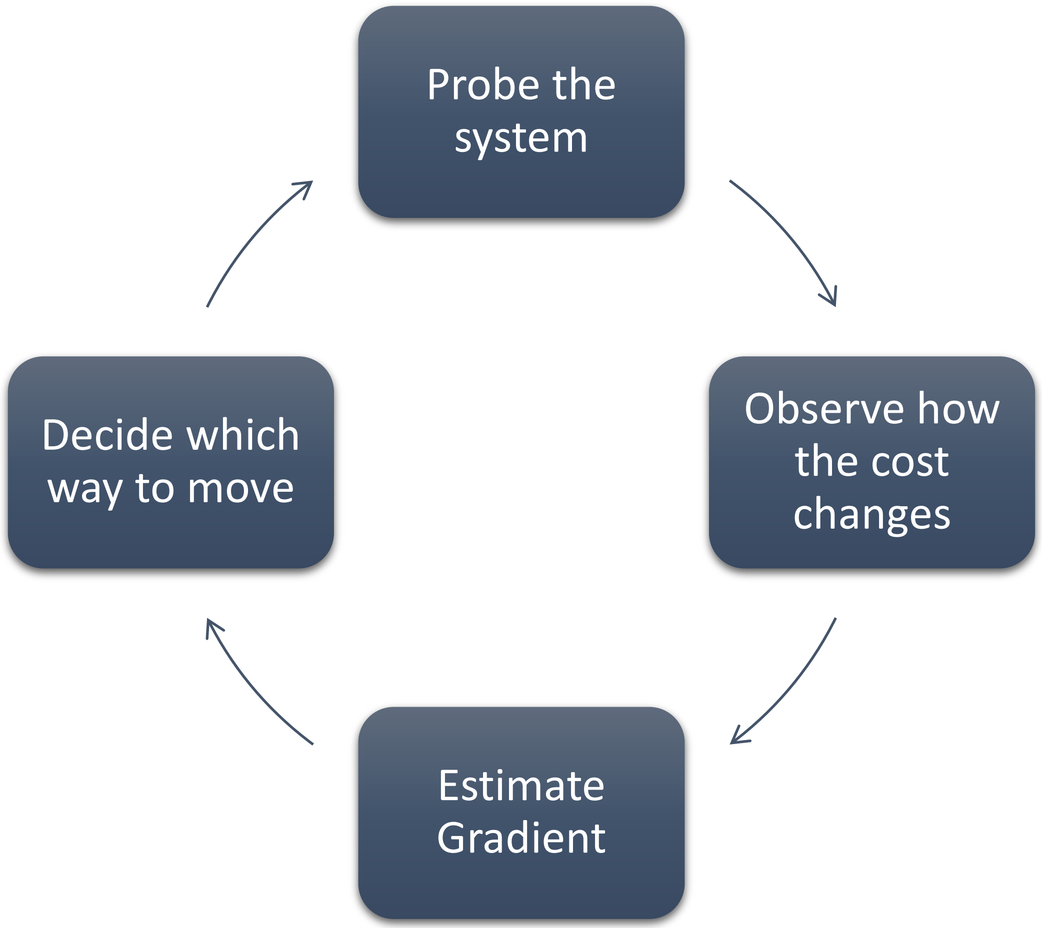
Sinusoidal perturbation

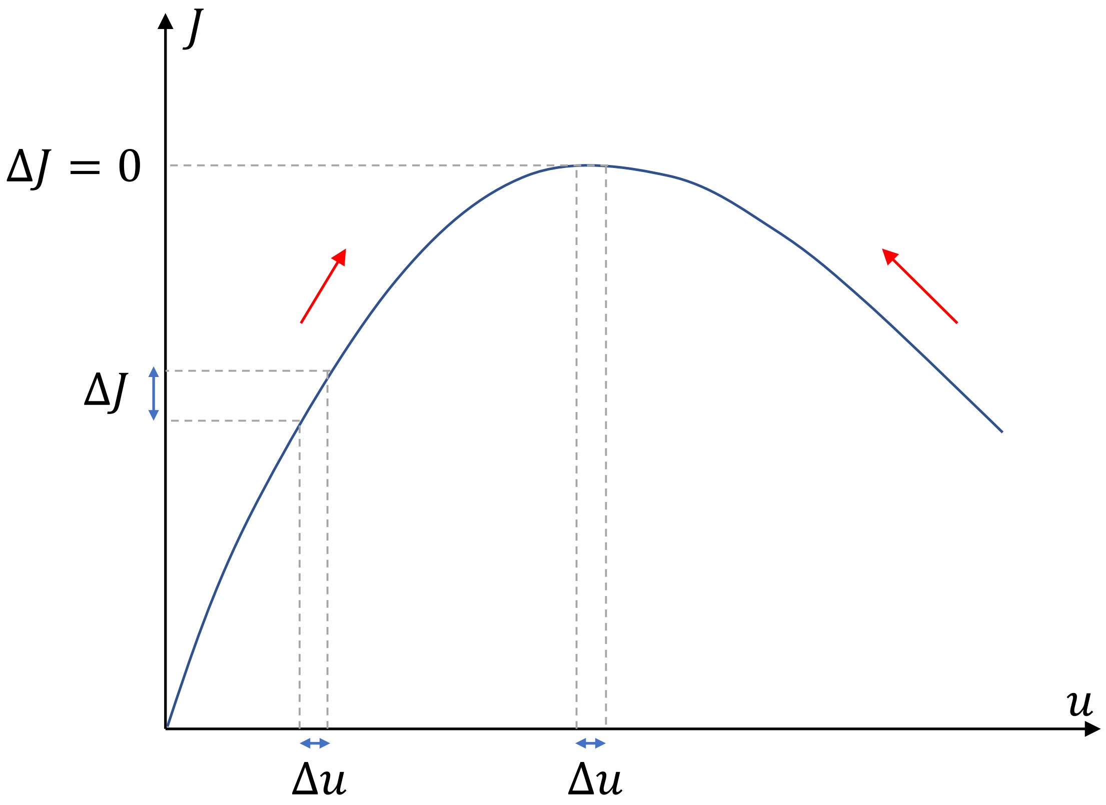




Special case of Fast Fourier Transform (FFT) - single frequency case

Extremum Seeking Control



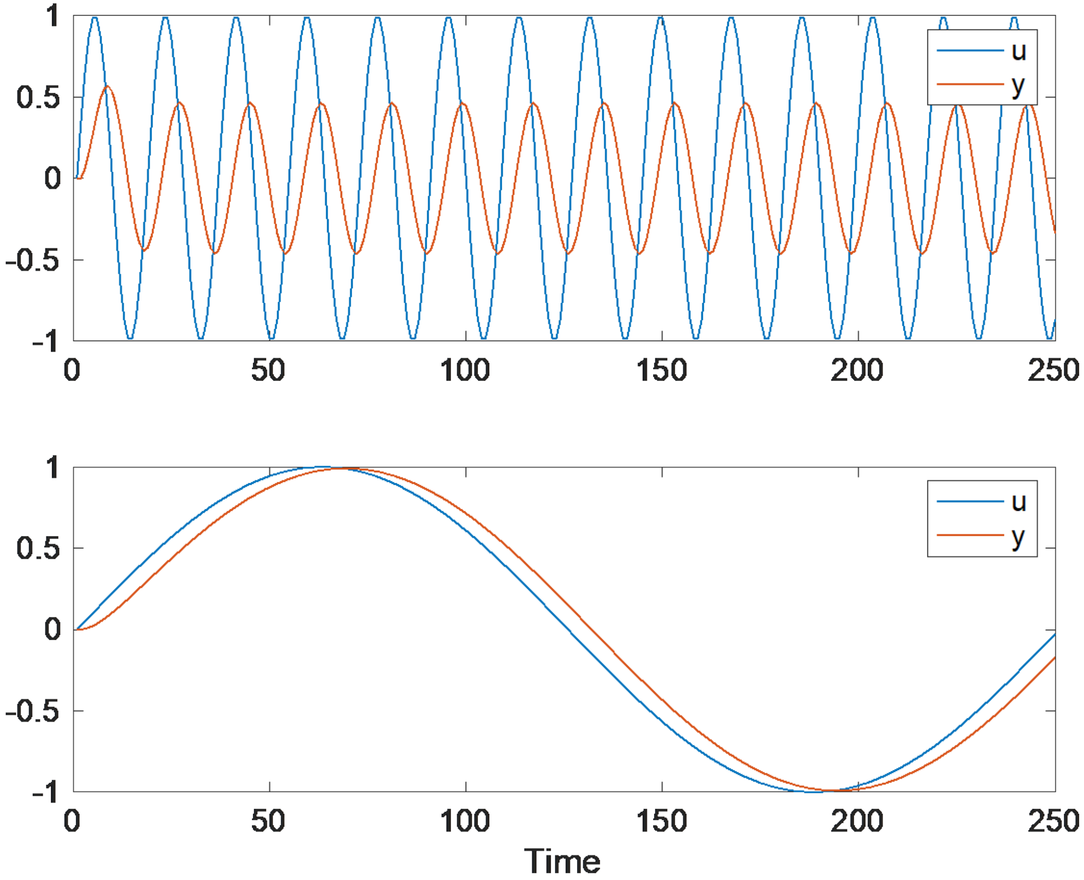


Classical Extremum Seeking Control

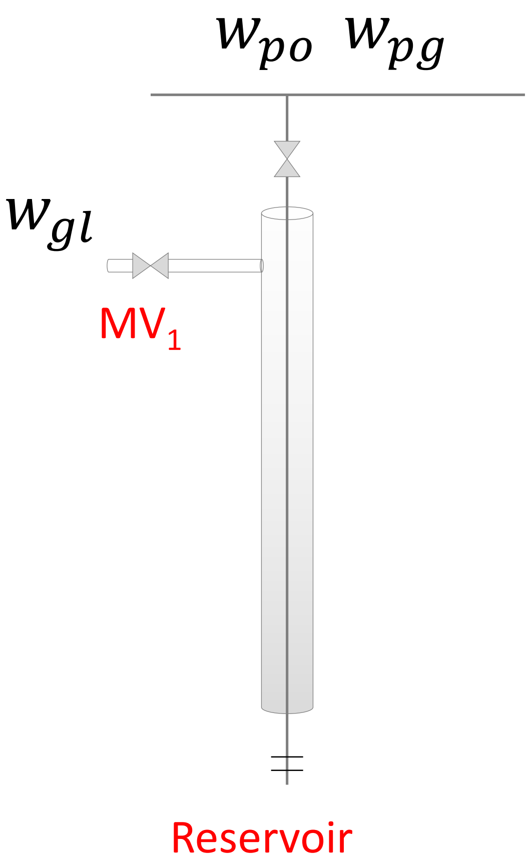
NTNU
Kunnskap for en bedre verden

- Needs time scale separation to approximate plant as static map
- Prohibitively slow convergence for systems with slow dynamics
- Typically 100 times slower than the system dynamics !
- Can we remove the static map assumption?

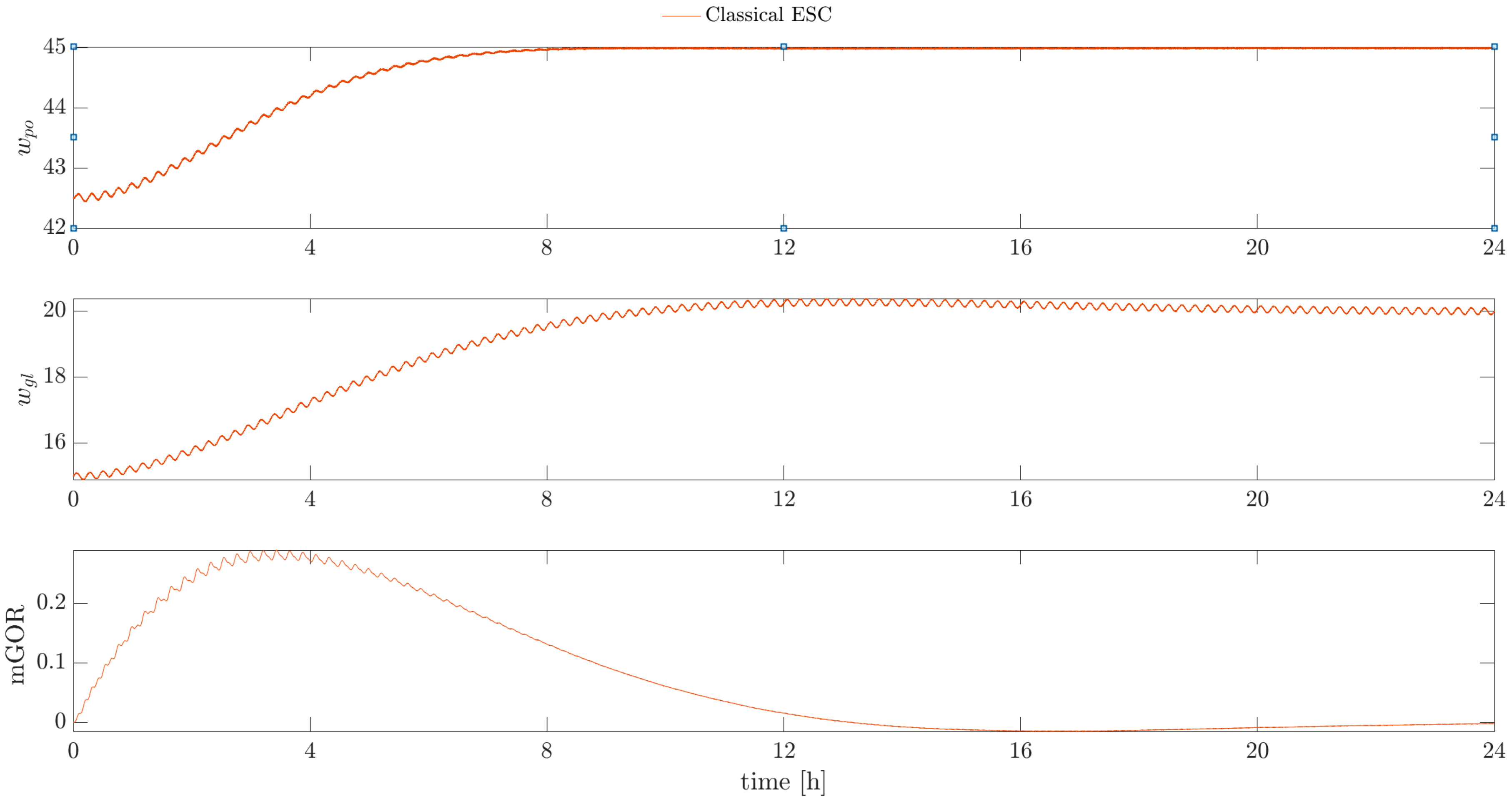
Come to my talk at....



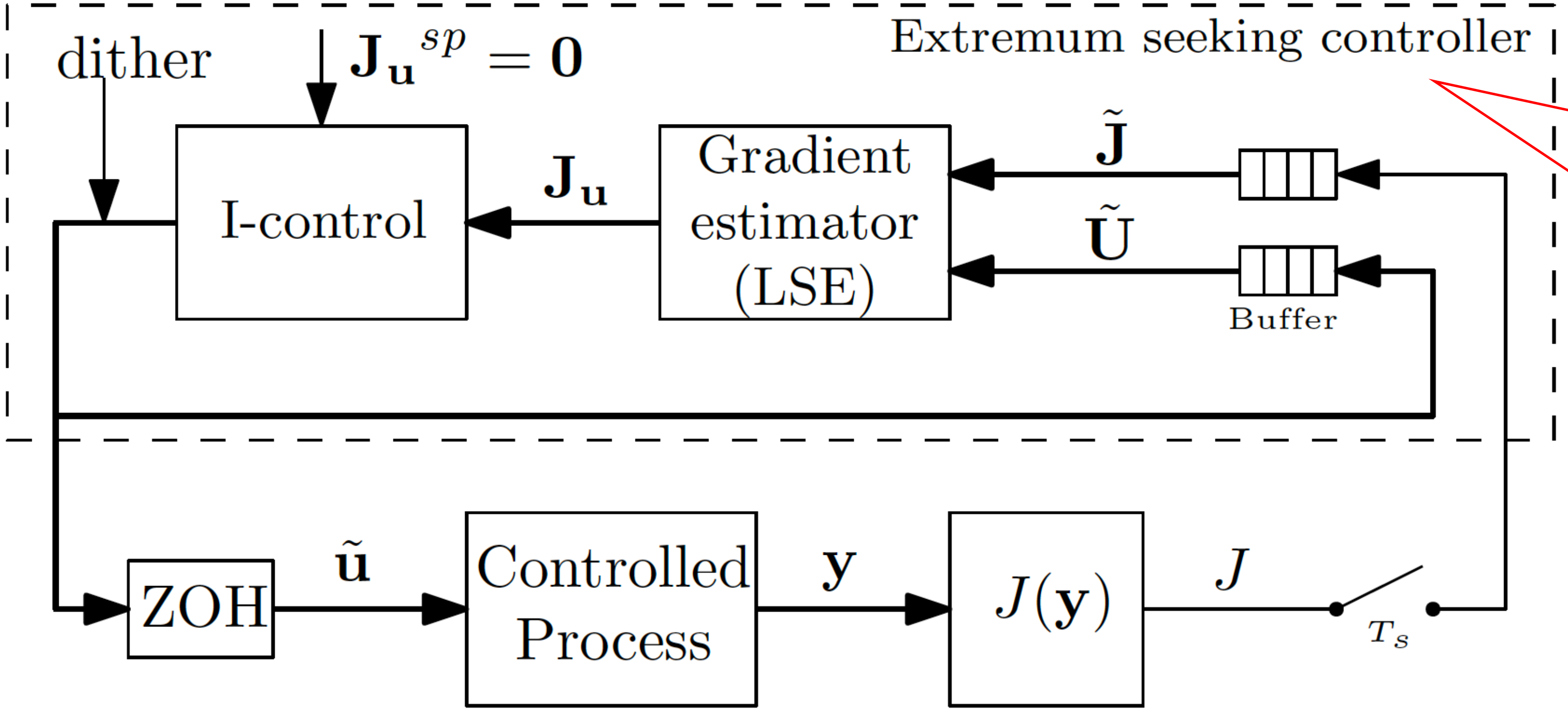
CASE STUDY: Gas lift well



GOR = Gas/Oil ratio in feed



Least square Extremum seeking control



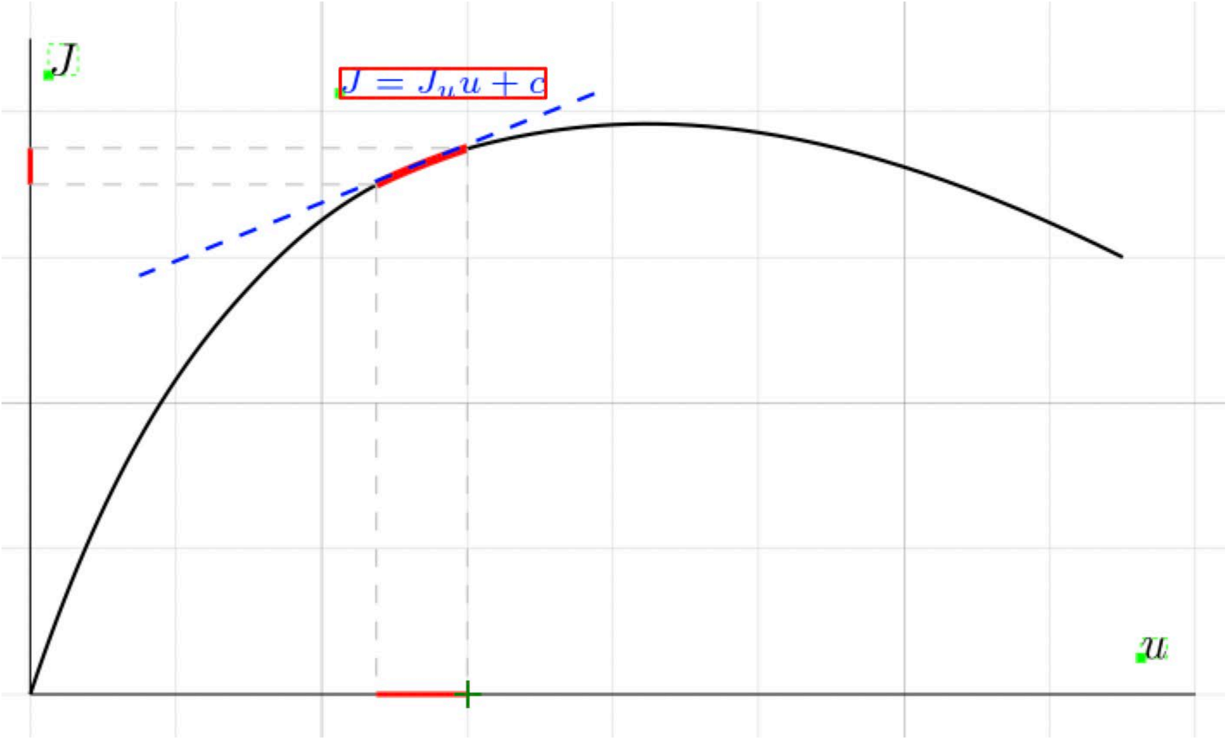
Fit a linear model

$$J = \mathbf{J}_{\tilde{\mathbf{u}}}{}^{\mathsf{T}} \tilde{\mathbf{u}} + m$$

Using least squares fit

Hunnekens et al. (2011)

Least square Extremum seeking control



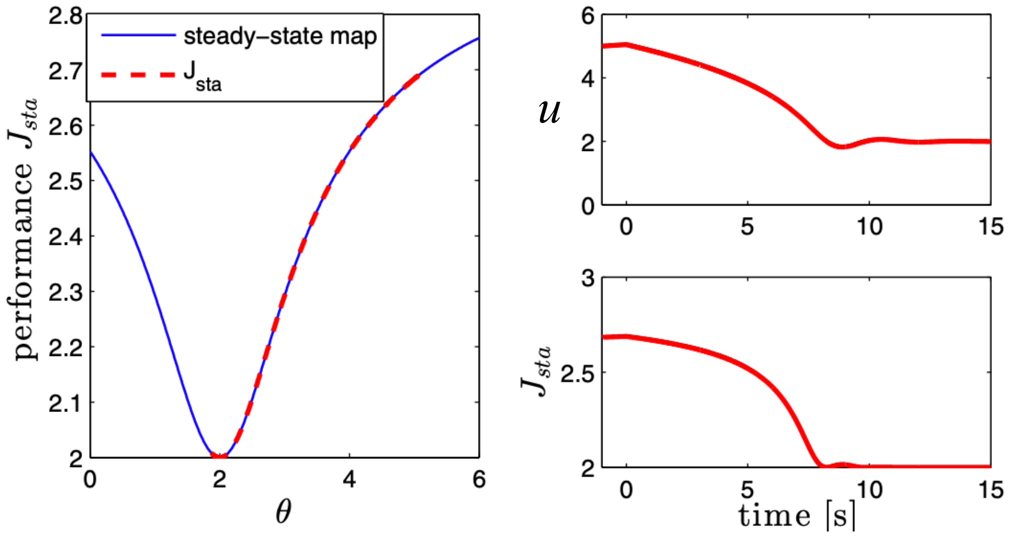


Image source: Hunnekens et al. (2011)

Other gradient estimation schemes

- Multiple units (Srinivasan et al. 2007)
- Recursive least squares estimation (Chioua, 2016)
- Phasor based extremum seeking control (Trolleberg & Jacobsen, 2012)

... and some other model-based schemes

- Neibhouring extremals (Gros et al. 2009)
- Parameter estimation (Adetola & Guay, 2007)

Issues with Extremum seeking

- Need Cost measurement
  - Often cost function is a sum of several terms

$$J = \sum p_F F + \sum p_Q Q - \sum p_p P$$

  - All terms must be measured
  - Estimation of cost requires model (dependency on model – no longer model free)
- Time scale separation
  - Process dynamics affects gradient estimation
  - Prohibitively slow convergence to the optimum
- Constant probing of the system
- Unknown and abrupt disturbances affects gradient estimation

ESC more suited for single units, but not for entire chemical plants

NTNU
Norwegian University of Science and Technology

6. Modifier Adaptation

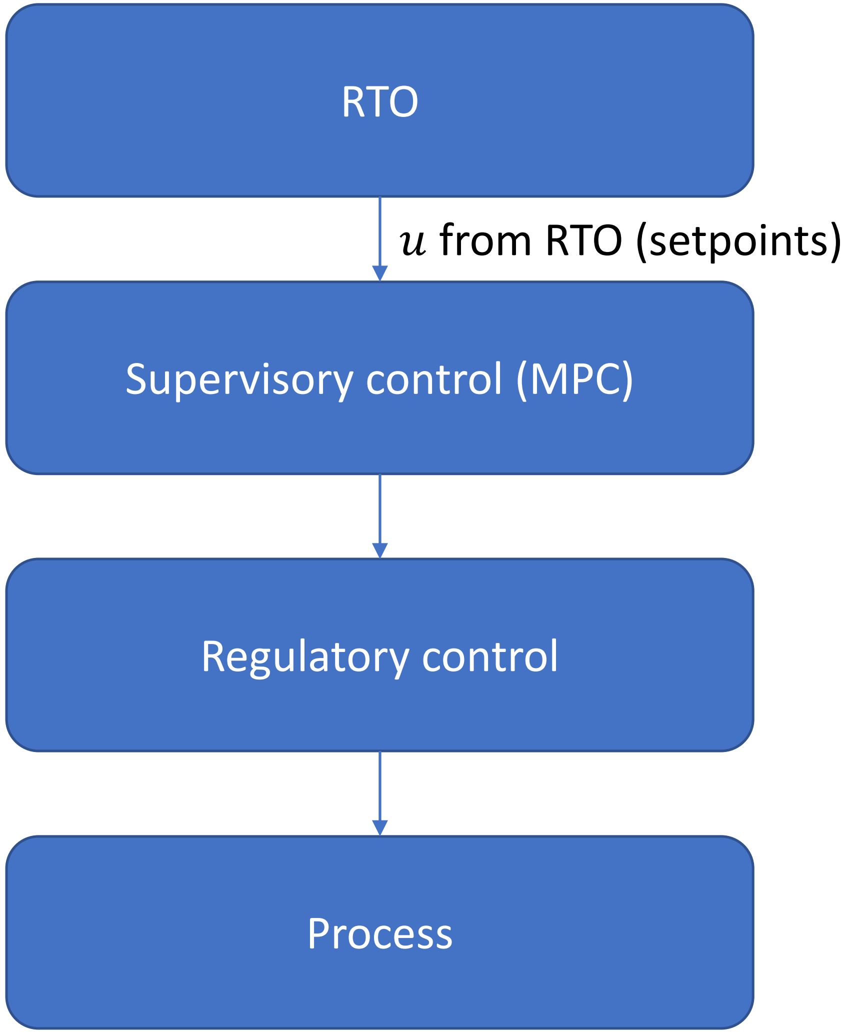
IFAC DYCOPS Pre-symposium workshop

Johannes Jäschke

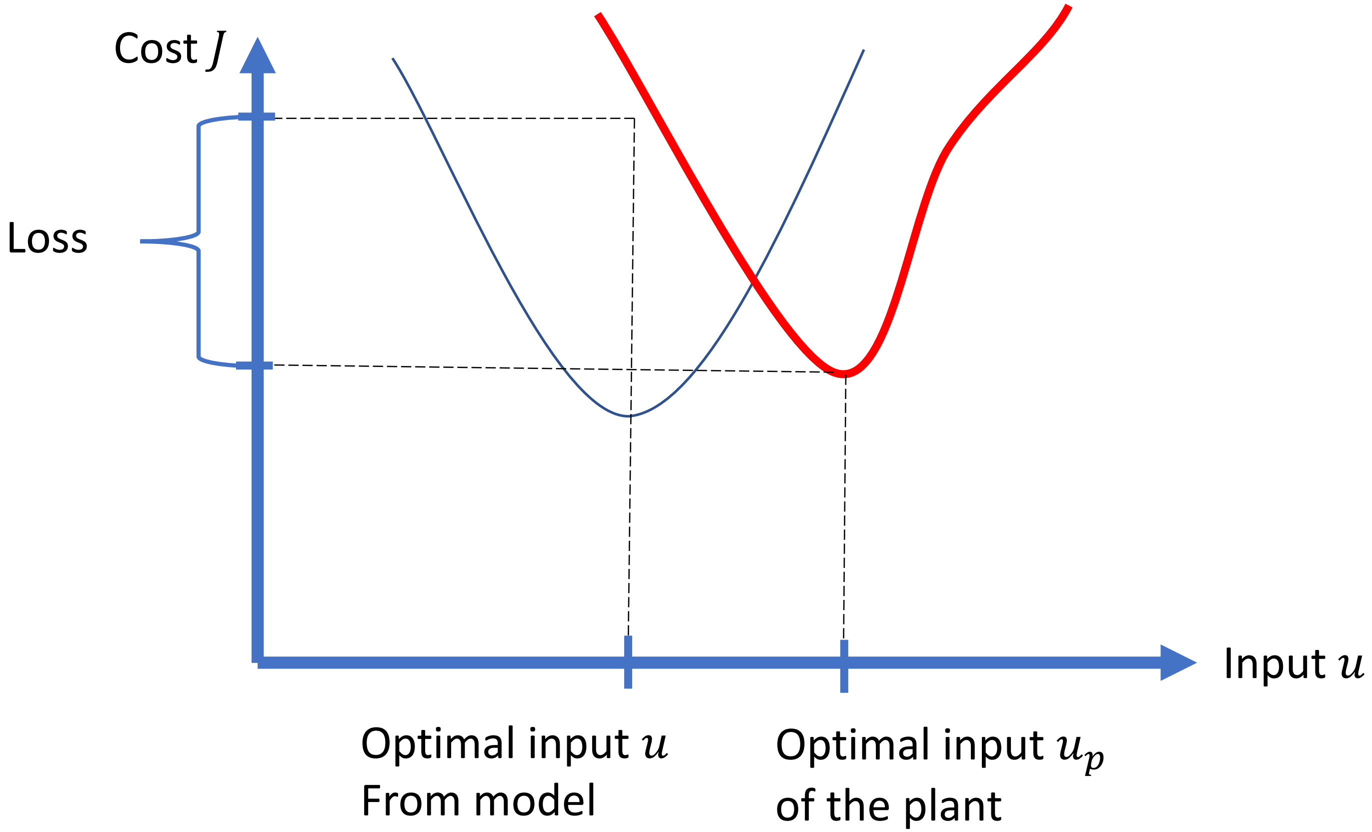
Why is traditional static RTO not commonly used?

1. Cost of developing and updating the model (costly offline model update)
2. Wrong value of model parameters and disturbances (slow online model update)
3. Not robust, including computational issues
4. Frequent grade changes make steady-state optimization less relevant
5. Dynamic limitations, including infeasibility due to (dynamic) constraint violation
6. Wrong model structure

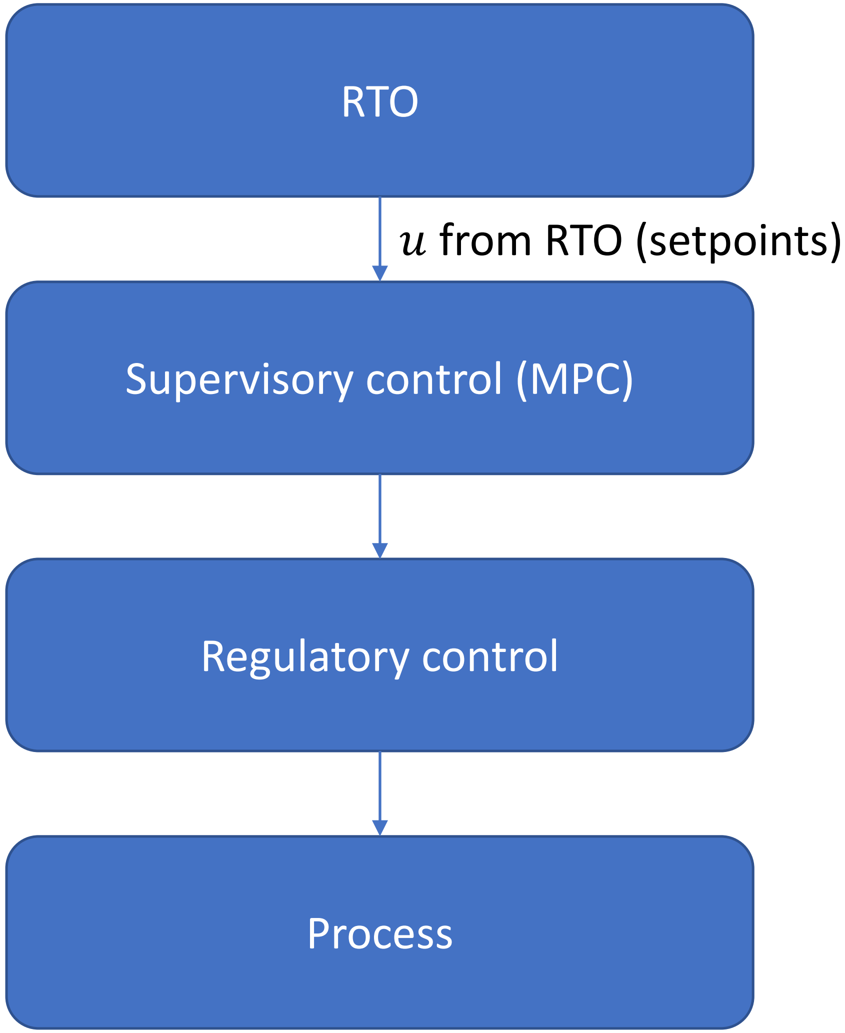
Modifier adaptation addresses the problem of plant model mismatch



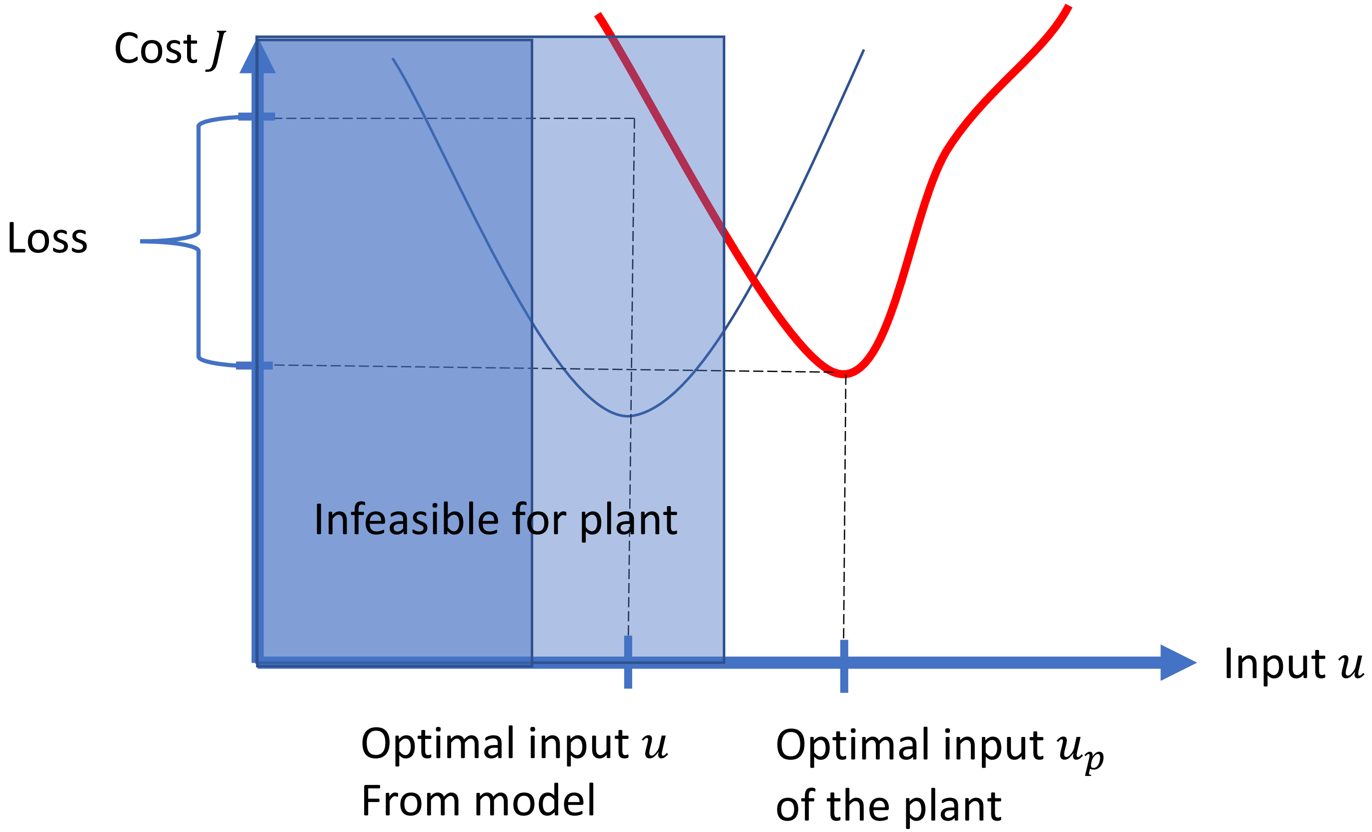
- Mismatch between model and plant leads to performance loss



Modifier adaptation addresses the problem of plant model mismatch



- Mismatch between model and plant leads to constraint violations (infeasibility)



Possible solutions

1. Find a better model
   - Better parameters
   - Better structure (that matches the plant better)
2. Modify optimization problem directly (Marchetti et al, 2009, Gao et al 2016)
   - Use plant measurements
   - No need to have an exact model

Marchetti et al. 2009, *Ind. Eng. Chem. Res.*, 48 (13), pp 6022–6033
Gao et al.2016 Comp. & Chem. Eng. 91, pp 318–328

How should modify the optimization problem

- Plant Optimization problem

$$\min_u J_p$$

s.t.

$$g_p(u) \leq 0$$

- Model optimization problem

$$\min_u J$$

s.t.

$$g(u) \leq 0$$

- Key idea
  - Add modifiers to make "optimality conditions" of the plant and optimality conditions of the model match
- Iteratively repeat the optimization at sample times k.

How should modify the optimization problem

- Plant Optimization problem

$$\min_u J_p$$

s.t.

$$g_p(u) \leq 0$$

- Modified model optimization problem

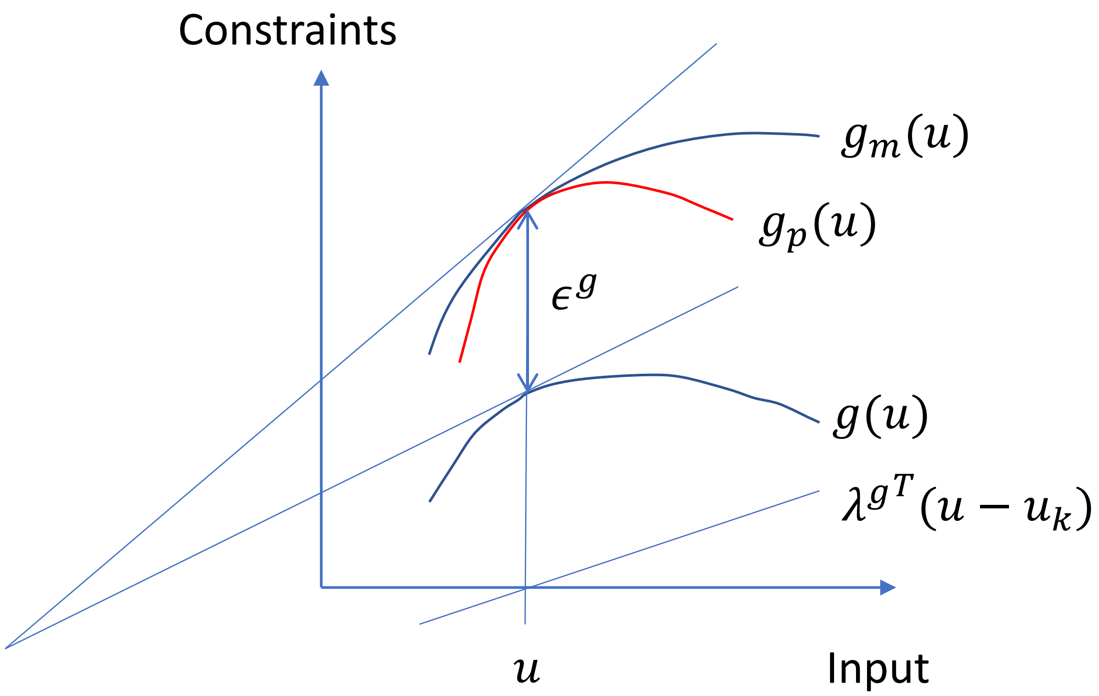
$$\min_u J_m = J(u) + \epsilon_k^J + \lambda_k^J (u - u_k)$$

s.t.

$$g_m = g(u) + \epsilon_k^g + \lambda_k^g (u - u_k) \leq 0$$

- Key idea
  - Add modifiers to make "optimality conditions" of the plant and optimality conditions of the model match
- Iteratively repeat the optimization at sample times k.

Modifiers to match plant derivatives



- Plant and modified model function gradients are equal

$$\frac{\partial J_p}{\partial u} = \frac{\partial J_m}{\partial u}$$

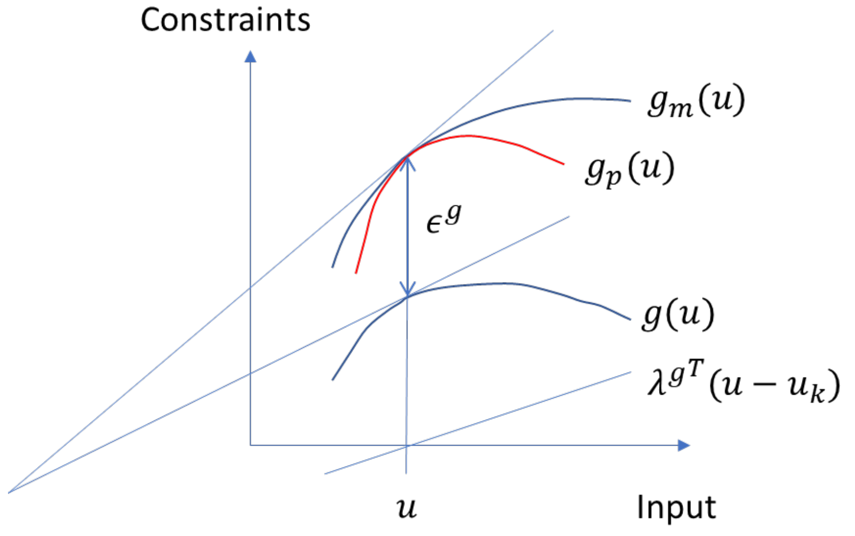
And

$$\frac{\partial g_p}{\partial u} = \frac{\partial g_m}{\partial u}$$

$$g_p = g_m$$

Cost and constraint gradients are modified to match $\Rightarrow$ Plant and model optimum coincide

How to compute modifiers



- Zero order modifiers

$$\epsilon^J = J_p(u) - J(u)$$
$$\epsilon^g = g_p(u) - g(u)$$

First order modifiers

$$\lambda^{J^T} = \frac{\partial J_p(u)}{\partial u} - \frac{\partial J(u)}{\partial u}$$

Gradients from real plant

$$\lambda^{g_i^T} = \frac{\partial g_{p,i}(u)}{\partial u} - \frac{\partial g_i(u)}{\partial u}$$

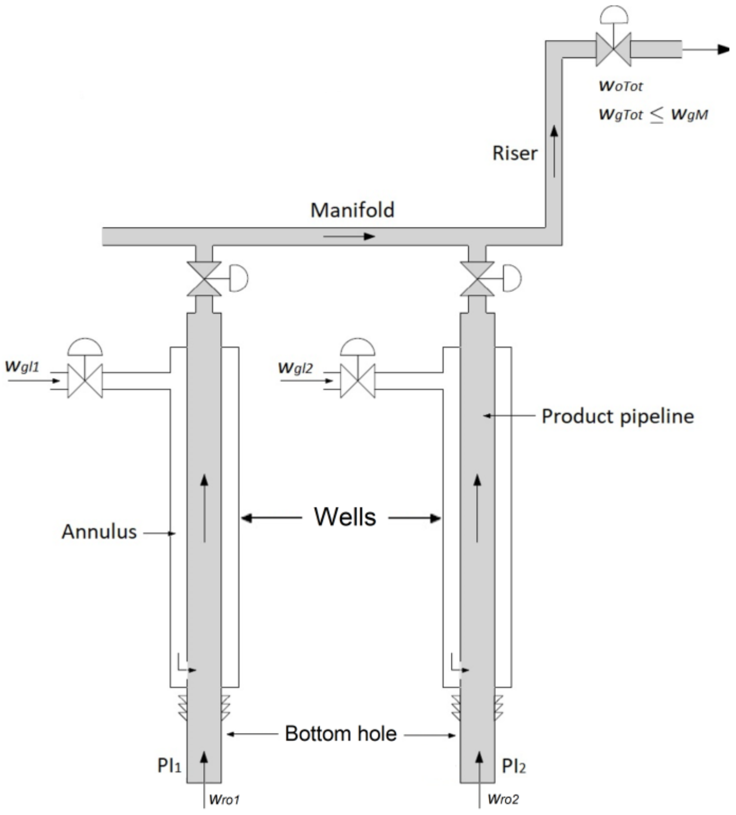
Challenges

- Finding gradients of the plant

- Requires excitation
  - Finite differences (Marchetti et al. 2009),
    - Experiments and past points
  - Broydens method
  - Gradients from fitted surface (Gao et al 2016, Matias, J. 2019)
  - Dynamic model identification (using transient data)
  - Parallel units (Srinivasan 2007)

Case Study

- Gas lifted oil well

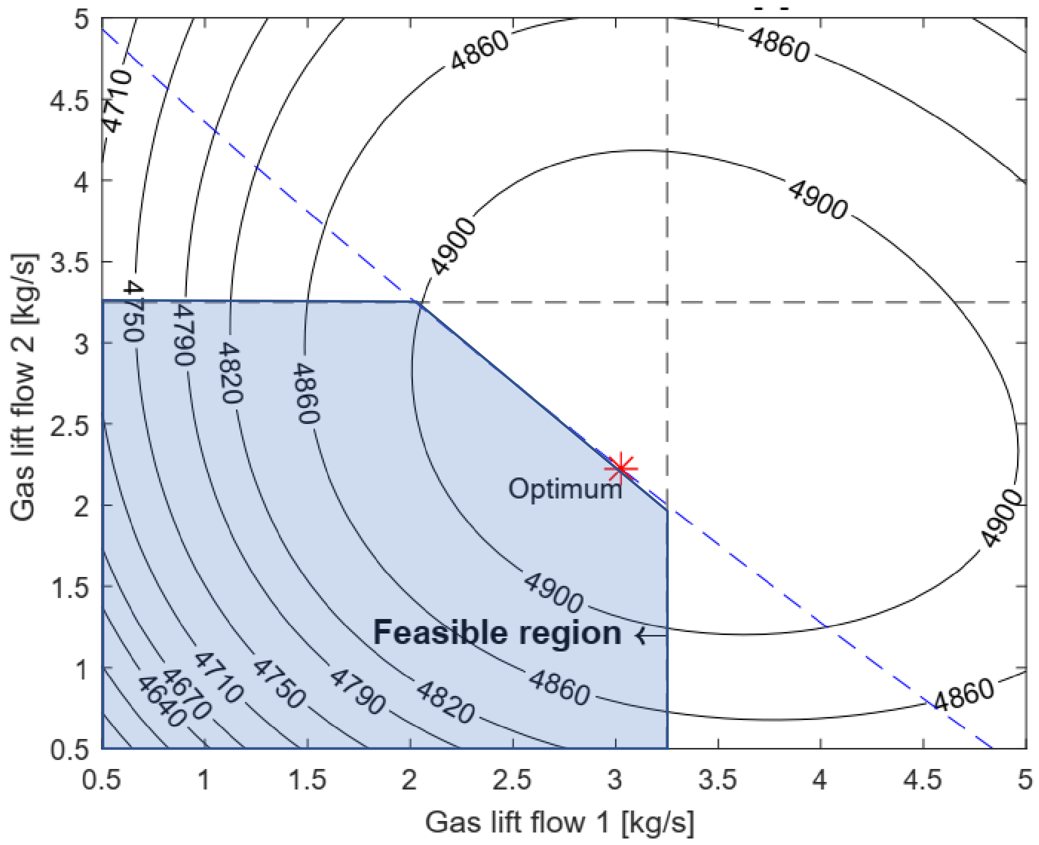


- 2Degrees of freedom
- $u = \left[w_{gl1} \;\; w_{gl2}\right]^T$
- Constraints:
  - Max gas lift for each well
  - Max total gas handling capacity
- Objective

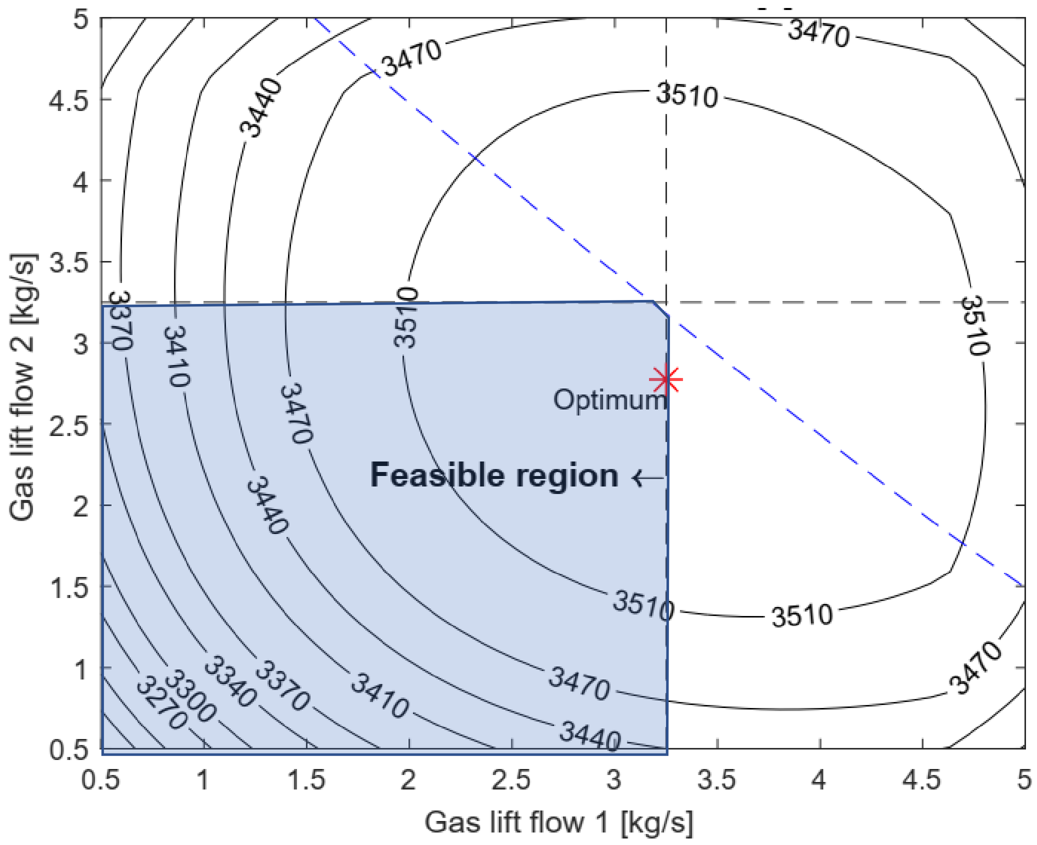
$$J = w_o^{tot} - 0.5(w_{g1}^2 + w_{g2}^2)$$

Case Study – Plant-model mismatch

Plant

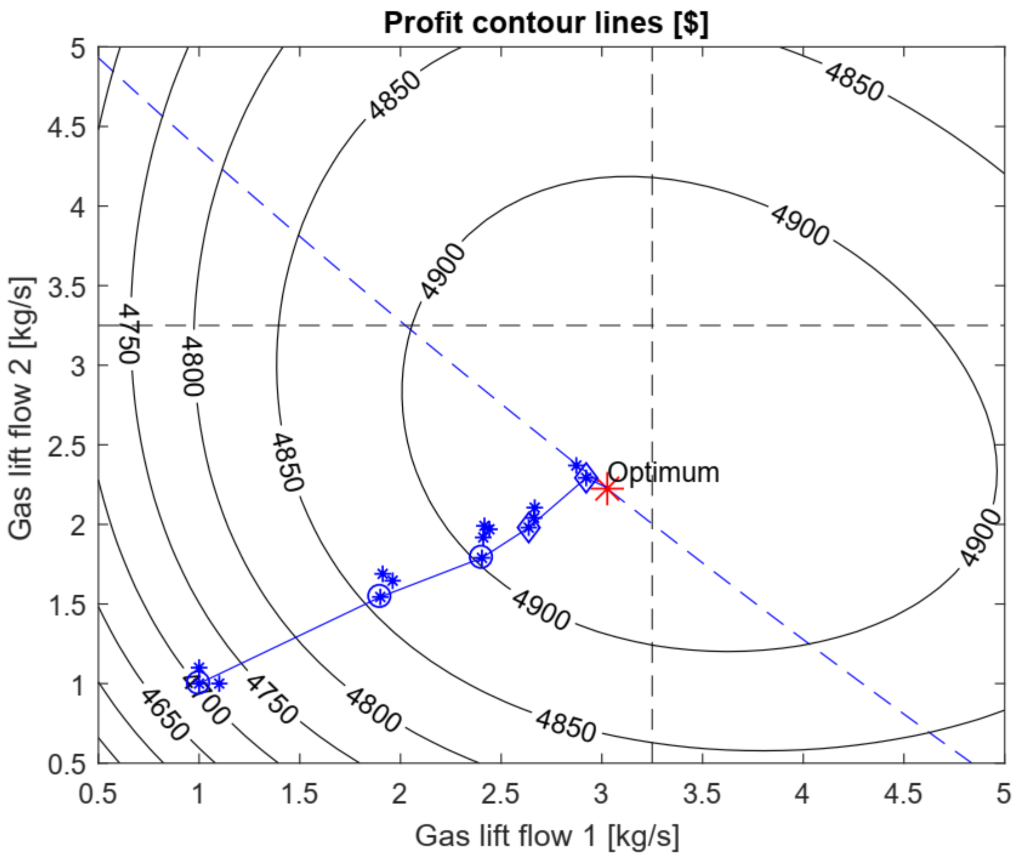


Model



- Blue dashed line: max gas constraint
- Black dashed line: max individual gas flow rate

Iterative RTO using Modifier adaptation



- Circle: and Diamond:

Points from: MA-RTO iterations

- Stars: Probing point for estimating gradients

Alternative: Output modifier adaptation

- Instead of adjusting cost and constraints, adjust output model

$$y_m(u) = y(u) + \epsilon^y + \lambda^{y^T}(u - u_k)$$

- Modified RTO problem

$$\boldsymbol{u_{k+1}} = \arg\min J_m \ := J(u, y(u) + \epsilon^y + \lambda_k^{y^T}(u - u_k))$$

s.t.

$$g(u, y(u) + \epsilon^y + \lambda_k^{y^T}(u - u_k) \leq 0$$

Marchetti et al 2009

Cost and constraint gradients are modified to match $\longrightarrow$ Plant and model optimum coincide

Conclusion

- A effective way to handle plant-model mismatch
  - Combines properties from model-based and data-based optimization
- Optimization problem is updated using plant gradient estimates
  - Same gradient estimation problems as ESC
- Iteratively converges to an optimum.
  - Relatively slow, but we start at a better point (from the model).
  - Can (should) be combined with other approaches
  - Better than doing nothing, and living with the mismatch
- Other refinements
  - Decentralized schemes (Schneider et al. 2018)
  - Second order modifiers (Faulwasser, Bonvin 2014)

NTNU
Norwegian University of
Science and Technology

7. Self-optimizing control

“Move the optimization into the control layer”

IFAC DYCOPS Pre-symposium workshop

Sigurd Skogestad

Do we really need real-time optimization?

- Often not!
- We often know or can guess the **active constraints**
  - Example: Assume it’s optimal with max. reactor temperature
  - No need to have a comples dynamic model with energy balance to find the optimal cooling
  - Just use a PI-controller
    - CV = reactor temperature
    - MV = cooling

Systematic procedure for economic process control

Start "top-down" with economics (steady state):

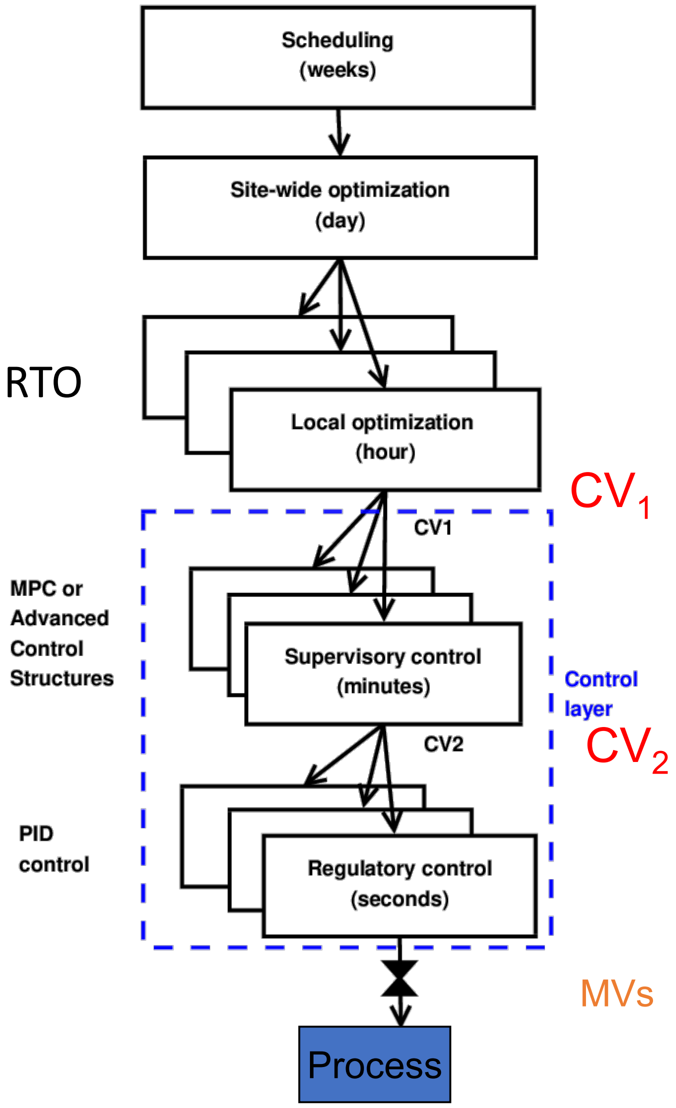
- Step 1: Define operational objectives (J) and constraints
- Step 2: Optimize steady-state operation
- Step 3: Decide what to control (CVs)
  - Step 3A: Identify active constraints = primary CV1.
  - Step 3B: Remaining unconstrained DOFs: Self-optimizing CV1 (find H)
- Step 4: Where do we set the throughput? TPM location

Then bottom-up (dynamics):

- Step 5: Regulatory control
  - Control variables to stop "drift" (sensitive temperatures, pressures, ...)

Finally: Make link between "top-down" and "bottom up"

- Step 6: "Advanced/supervisory control"
  - Control economic CVs: Active constraints and self-optimizing variables
  - Look after variables in regulatory layer below (e.g., avoid saturation)
- **Step 7: Real-time optimization** (Do we need it?)



S. Skogestad, ``Control structure design for complete chemical plants'', *Computers and Chemical Engineering*, **28** (1-2), 219-234 (2004).

Step 1. Define optimal operation (economics)

Usually steady state

NTNU

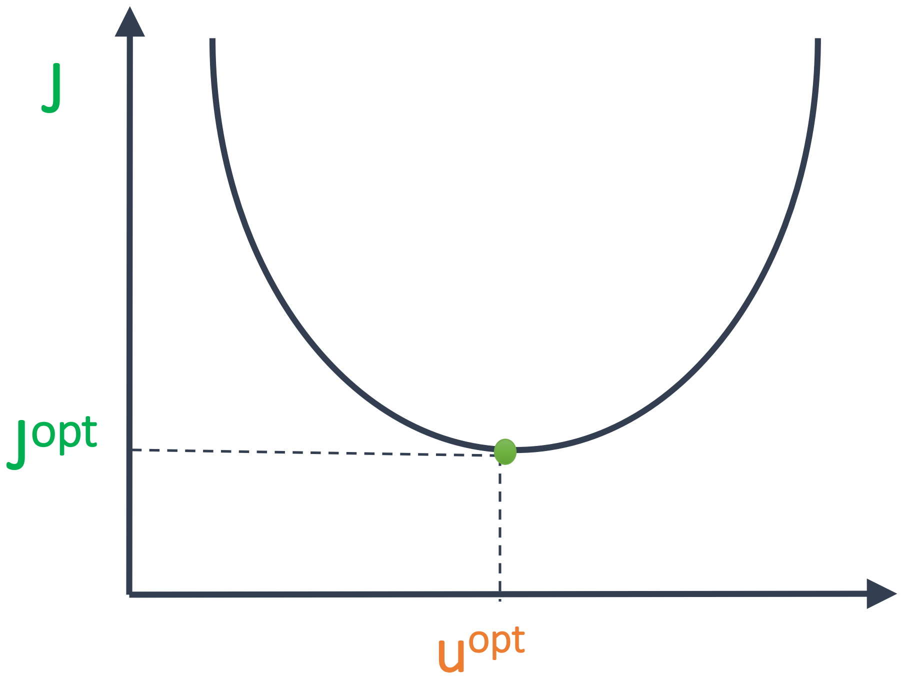
Minimize cost $J = J(u,x,d)$

subject to:

Model equations: $f(u,x,d) = 0$

Operational constraints: $g(u,x,d) < 0$

- u = degrees of freedom
- x = states (internal variables)
- d = disturbances



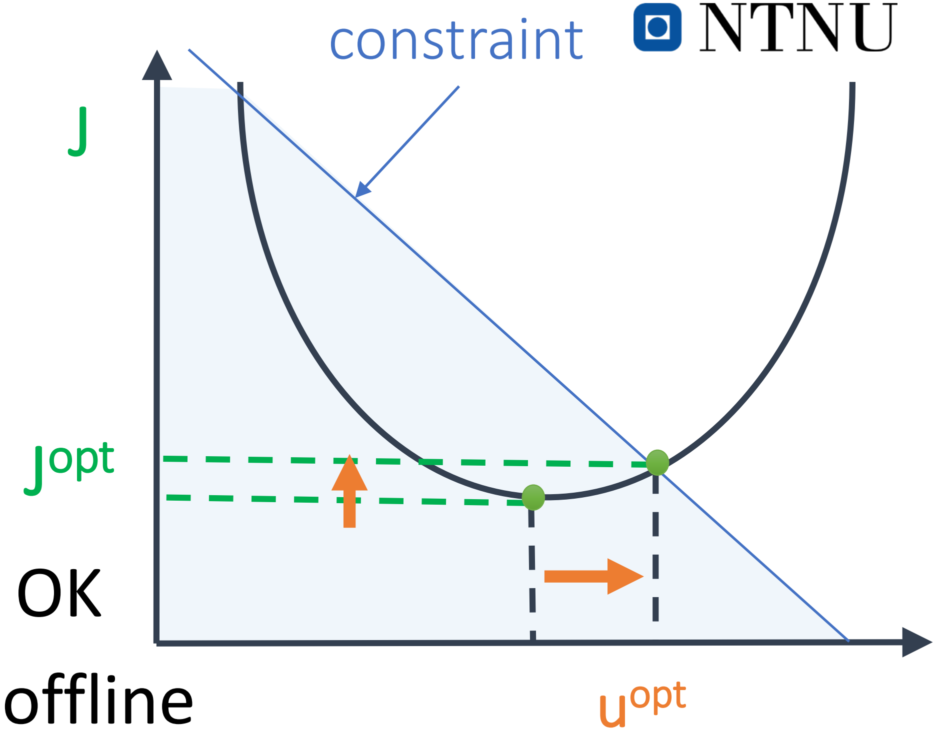
Typical cost function in process control:

J = cost feed + cost energy – value of products

Step 2. Optimize

NTNU

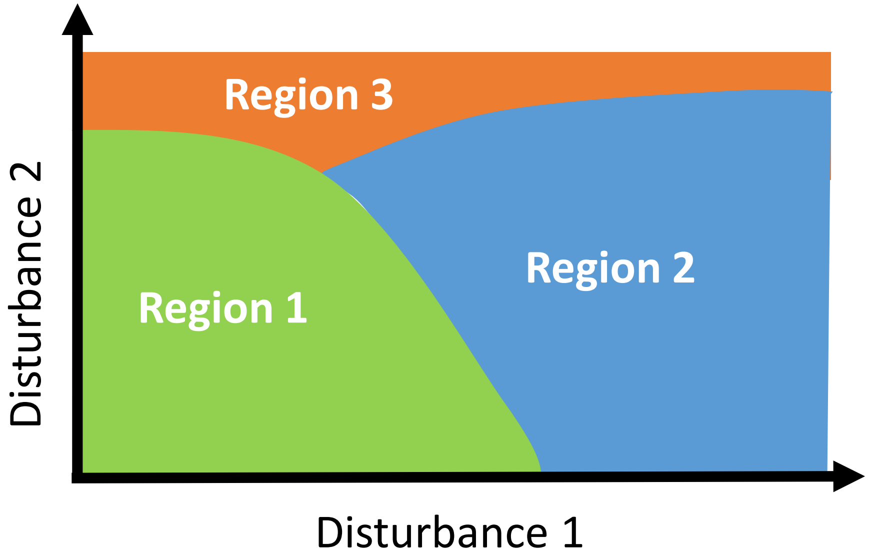
(a) Identify degrees of freedom
(b) Optimize for expected disturbances



- Need good model, usually steady-state is OK
- Optimization is time consuming! But it is offline
- Main goal: Identify **ACTIVE CONSTRAINTS**
- A good engineer can often guess the active constraints

Active constraints

- **Active constraints:**
  - variables that should optimally be kept at their limiting value.
- **Active constraint region:**
  - region in the disturbance space defined by which constraints are active within it.



Optimal operation:
Need to switch between regions
using control system

Step 3. Implementation of optimal operation

- Have found the optimal way of operation. How should it be implemented?
- **What to control ?** (CV_1).
  1. Active constraints
  2. Self-optimizing variables (for unconstrained degrees of freedom)

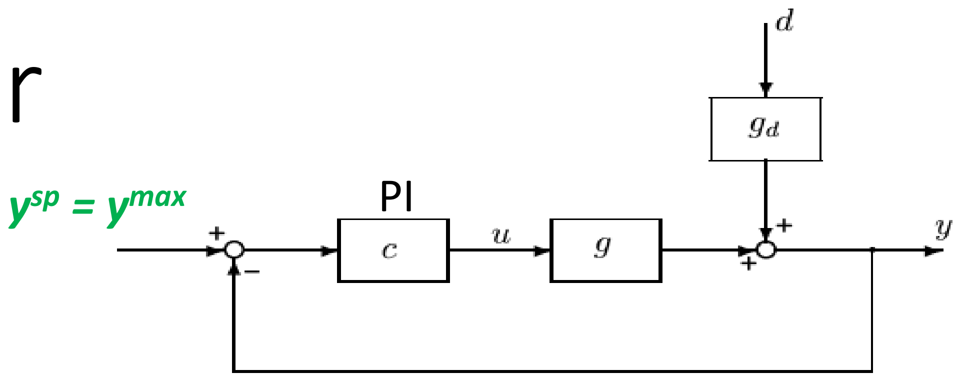
Always try first: Move optimization into control layer

Optimization with PI-controller

max y

s.t. $y \leq y^{max}$

$u \leq u^{max}$



Example: Drive as fast as possible to airport (u=power, y=speed, y^{max} = 120 km/h)

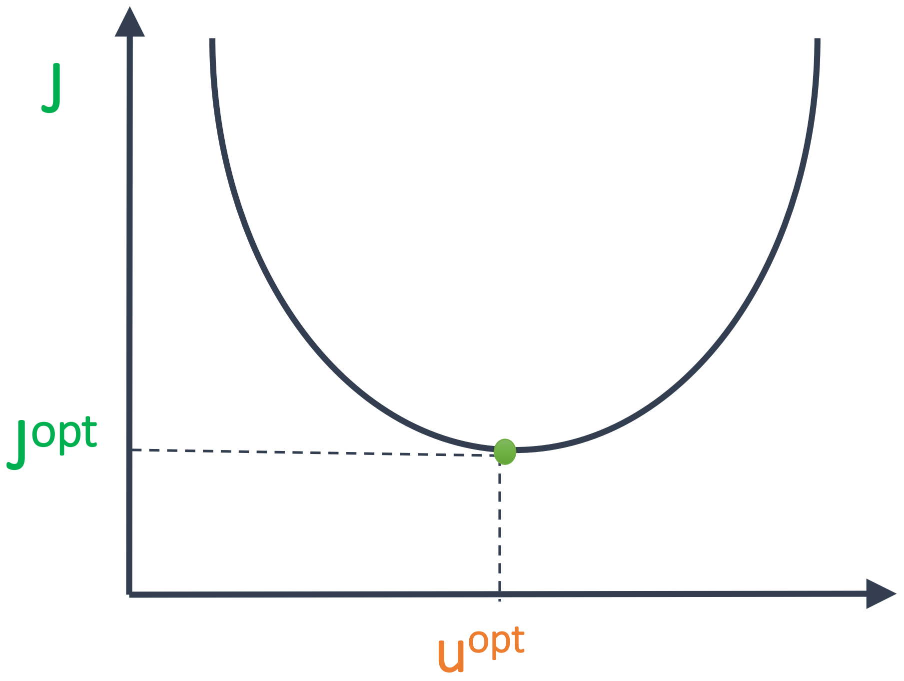
- **Optimal solution has two active constraint regions:**
  1. $y = y^{max}$ → **speed limit**
  2. $u = u^{max}$ → **max power**
- Note: Positive gain from MV (u) to CV (y)
- **Solved with PI-controller**
  - $y^{sp} = y^{max}$
  - Anti-windup: I-action is off when $u=u^{max}$

s.t. = subject to

y = CV = controlled variable

The less obvious case: Unconstrained optimum

- u: unconstrained MV
- What to control? y=CV=?



Example: Optimal operation of runner

- Cost to be minimized, J=T
- One degree of freedom (u=power)
- What should we control?

NTNU

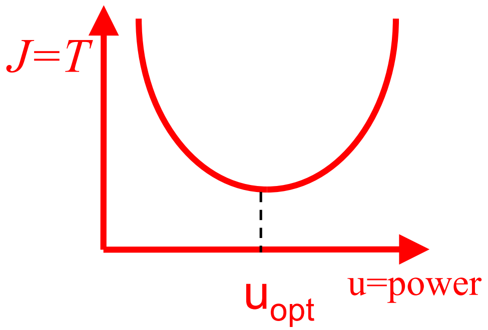
1. Optimal operation of Sprinter

- 100m. J=T
- Active constraint control:
  - Maximum speed (”no thinking required”)
  - CV = power (at max)

NTNU

2. Optimal operation of Marathon runner

- 40 km. J=T
- What should we control? CV=?
- Unconstrained optimum

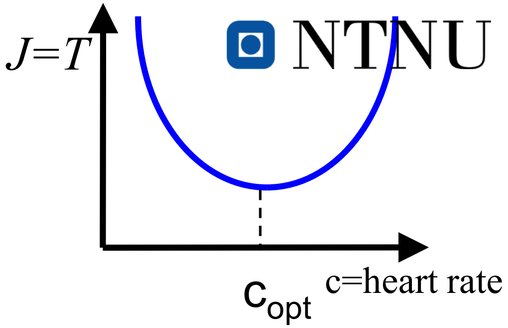


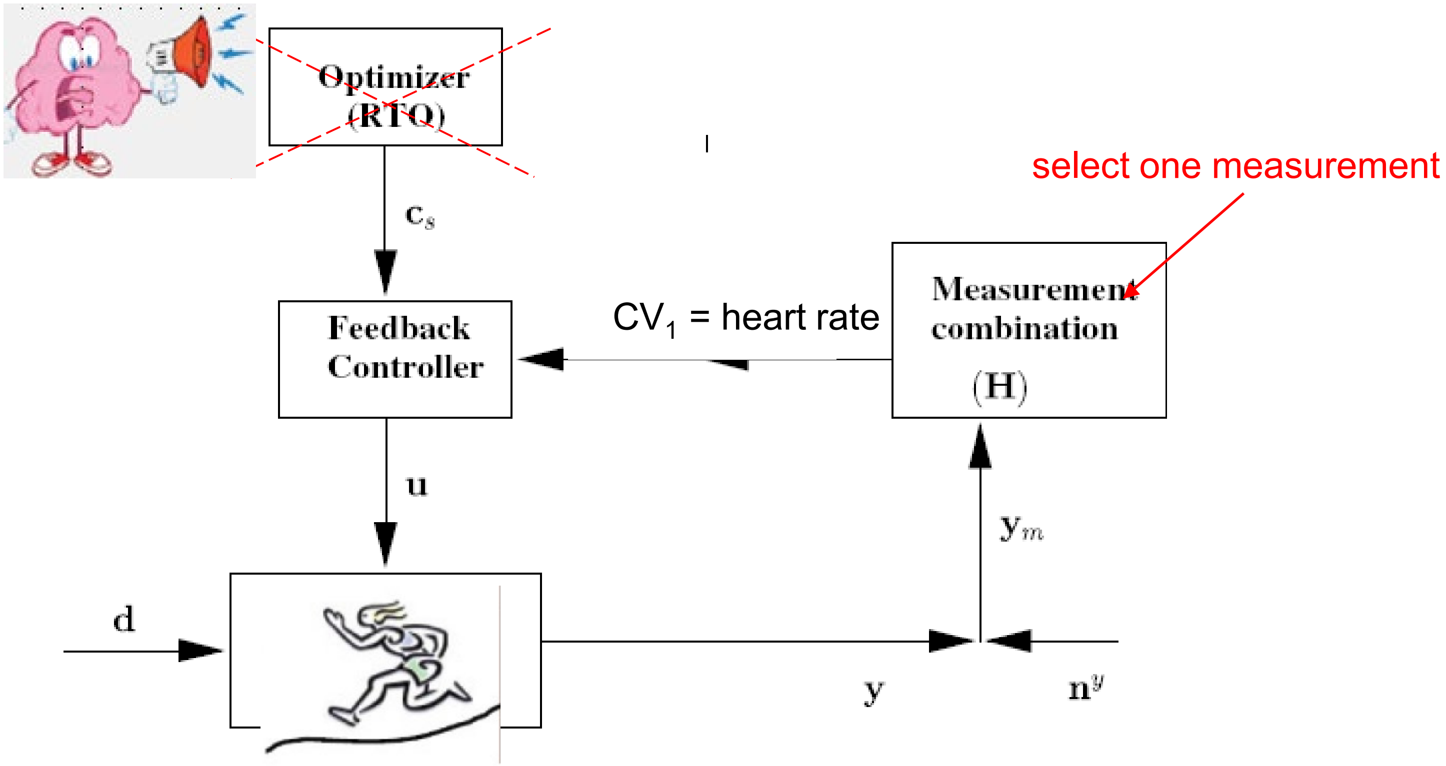
NTNU

Marathon runner (40 km)

- Any self-optimizing variable (to control at constant setpoint)?
  - c_1 = distance to leader of race
  - c_2 = speed
  - **c_3 = heart rate**
  - c_4 = level of lactate in muscles

Conclusion Marathon runner

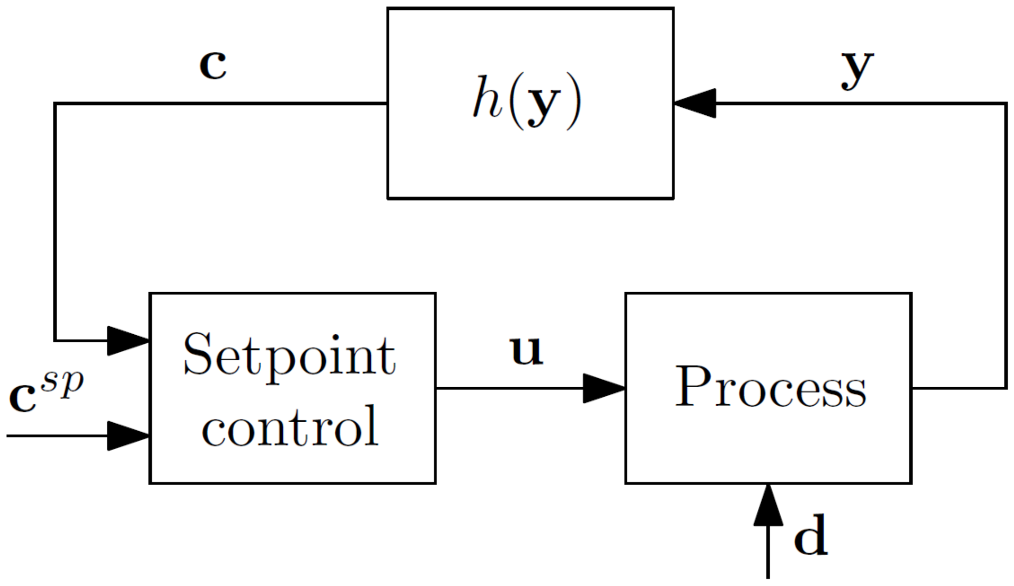


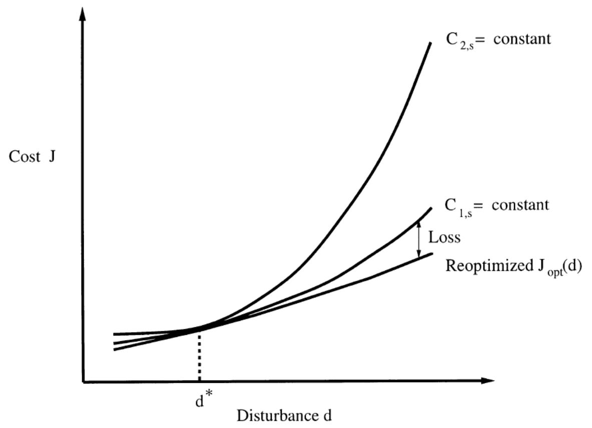


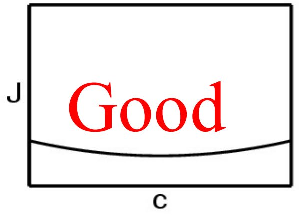
- CV = heart rate is good “self-optimizing” variable
- Simple and robust implementation
- Disturbances are indirectly handled by keeping a constant heart rate
- <u>May</u> have infrequent adjustment of setpoint (c_s)

Self-optimizing control

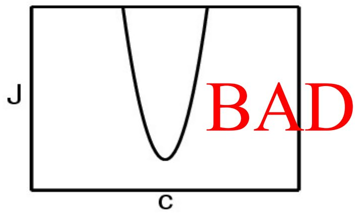
Self-optimizing control is when we can achieve an acceptable loss with constant setpoint values for the controlled variables







(b) Flat optimum: Implementation easy



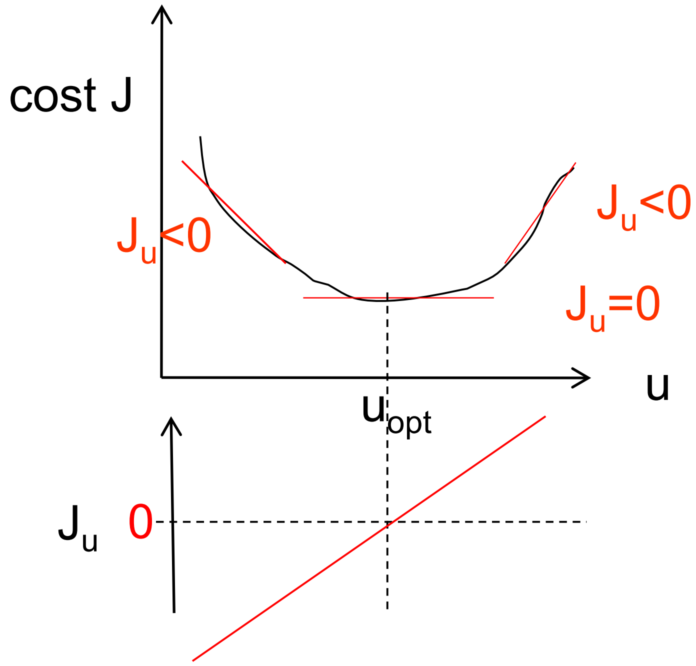
(c) Sharp optimum: Sensitive to implementation erros

NTNU

The ideal "self-optimizing" variable is the gradient, J_u

$$c = \partial J/\partial u = J_u$$

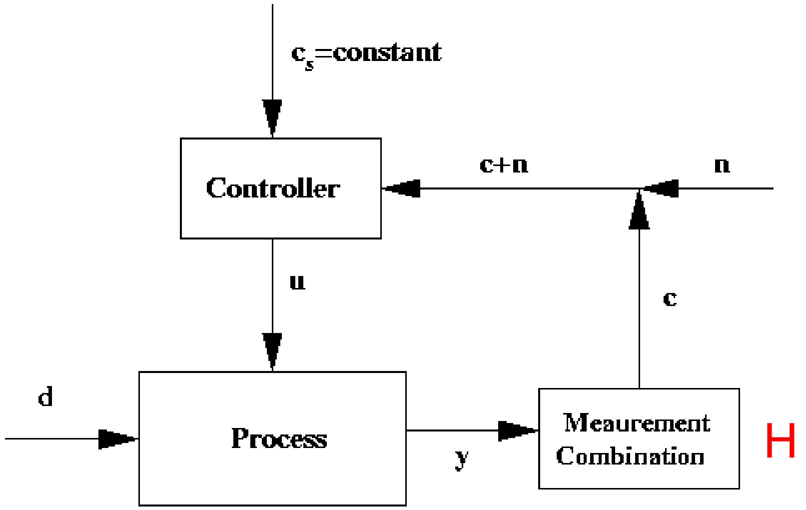
- Keep gradient at zero for all disturbances ($c = J_u = 0$)



Problem: Usually no measurement of gradient

Ideal: $c = J_u$

In practise, use available measurements: c = H y. Task: Select H!



- Single measurements:

$$\mathbf{c} = \mathbf{H}\mathbf{y} \qquad \mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

- Combinations of measurements:

$$\mathbf{c} = \mathbf{H}\mathbf{y} \qquad \mathbf{H} = \begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{14} \\ h_{21} & h_{22} & h_{23} & h_{24} \end{bmatrix}$$

NTNU

Combinations of measurements, c= Hy

Nullspace method for H (Alstad):

HF=0 where $F = dy_{opt}/dd$

- Proof. Appendix B in: Jäschke and Skogestad, "NCO tracking and self-optimizing control in the context of real-time optimization", *Journal of Process Control*, 1407-1416 (2011)

NTNU

Example. Nullspace Method for Marathon runner

u = power, d = slope [degrees]

y_1 = hr [beat/min], y_2 = v [m/s]

c = Hy, H = $[h_1 \;\; h_2]]$

F = dy_{opt}/dd = [0.25 -0.2]'

HF = 0 -> $h_1 f_1 + h_2 f_2 = 0.25\, h_1 - 0.2\, h_2 = 0$

Choose $h_1 = 1$ -> $h_2 = 0.25/0.2 = 1.25$

Conclusion: **c = hr + 1.25 v**

Control **c = constant** -> hr increases when v decreases (OK uphill!)

With measurement noise

NTNU

Exact local method for H

$$\min_H \left\| \underbrace{J_{uu}^{1/2}(HG^y)^{-1}}_{\substack{\text{“Minimize” in Maximum gain rule} \\ \text{(maximize } S_1 \, G \, J_{uu}^{-1/2} \text{ , } G=HG^y \text{)}}} \underbrace{\overbrace{H[FW_d}^{\text{“=0” in nullspace method (no noise)}} \quad W_{n^y}]}_{\text{“Scaling” } S_1} \right\|_2$$

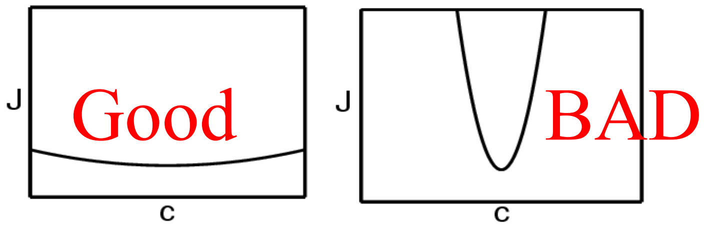
Analytical solution:

$H = G^{yT}(YY^T)^{-1}$ where $Y = [FW_d \quad W_{n^y}]$

What variable c=Hy should we control? (self-optimizing variables)

$$\min_H \left\| J_{uu}^{1/2} (\underbrace{HG^y}_{2})^{-1} \underbrace{H[F}_{1} W_d \quad W_{n^y}] \right\|_2$$

1. **The *optimal value* of c should be *insensitive* to disturbances**
   - Small HF = dc_{opt}/dd
2. **The *value* of c should be *sensitive* to the inputs ("maximum gain rule")**
   - Large G = HG^y = dc/du
   - Equivalent: Want flat optimum



(b) Flat optimum: Implementation easy

(c) Sharp optimum: Sensitive to implementation erros

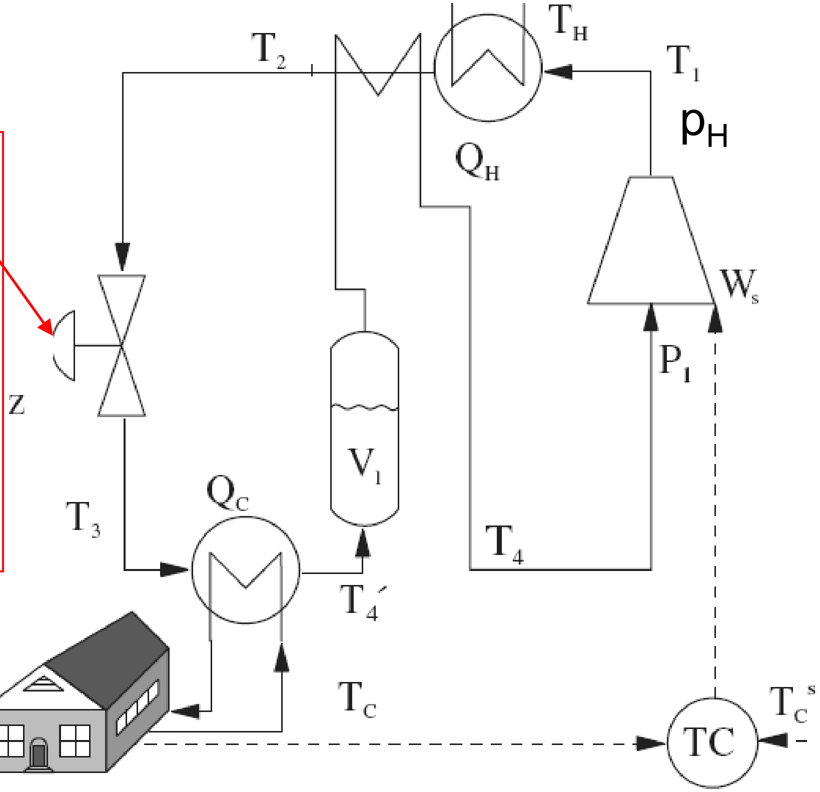
Note: Must also find optimal setpoint for $c=CV_1$

Example: CO2 refrigeration cycle

$J = W_s$ (work supplied)
DOF = u (valve opening, z)
Main disturbances:

$d_1 = T_H$
$d_2 = T_{Cs}$ *(setpoint)*
$d_3 = UA_{loss}$

What should we control?



CO2 refrigeration cycle

Step 1. One (remaining) degree of freedom (u=z)

Step 2. Objective function. J = W_s (compressor work)

Step 3. Optimize operation for disturbances (d_1=T_C, d_2=T_H, d_3=UA)

- Optimum always unconstrained

Step 4. Implementation of optimal operation

- No good single measurements (all give large losses):
  - p_h, T_h, z, …
- Nullspace method: Need to combine $n_u+n_d=1+3=4$ measurements to have zero disturbance loss
- Simpler: Try combining two measurements. Exact local method:
  - $c = h_1\, p_h + h_2\, T_h = p_h + k\, T_h$;   k = -8.53 bar/K
- Nonlinear evaluation of loss: OK!

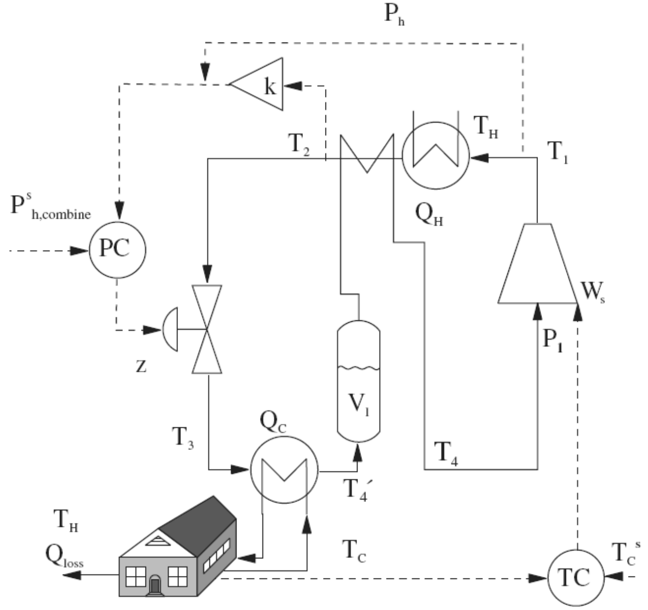
CO2 cycle: Maximum gain rule

NTNU

Linear "maximum gain" analysis of controlled variables for CO_2 case

Variable (y)	Nom.	$G = \frac{\Delta y}{\Delta u}$	$\lvert\Delta y_{\mathrm{opt}}(d_i)\rvert$ d_1 (T_H)	d_2 (T_C)	d_3 (UA_{loss})	$\lvert\Delta y_{\mathrm{opt}}\rvert$	n	Span y $\lvert\Delta y_{\mathrm{opt}}\rvert + n$	$\lvert G'\rvert = \frac{\lvert G\rvert}{\mathrm{span} y}$
P_h/T_2' (bar °C^{-1})	0.32	−0.291	0.140	−0.047	0.093	0.174	0.0033	0.177	0.25
P_h (bar)	97.61	−78.85	48.3	−15.5	31.0	59.4	1.0	60.4	1.31
T_2' (°C)	35.5	36.7	16.27	−2.93	7.64	18.21	1	19.2	1.91
$T_2' - T_H$ (°C)	3.62	24	4.10	−1.92	5.00	6.75	1.5	8.25	2.91
z	0.34	1	0.15	−0.04	0.18	0.24	0.05	0.29	3.45
V_1 (m^3)	0.07	0.03	−0.02	0.005	−0.03	0.006	0.001	0.007	4.77
T_2 (°C)	25.5	60.14	8.37	0.90	3.18	9.00	1	10.0	6.02
$P_{h,\mathrm{combine}}$ (bar)	97.61	−592.0	−23.1	−23.1	3.91	33.0	9.53	42.5	13.9
m_{gco} (kg)	4.83	−11.18	0.151	−0.136	0.119	0.235	0.44	0.675	16.55

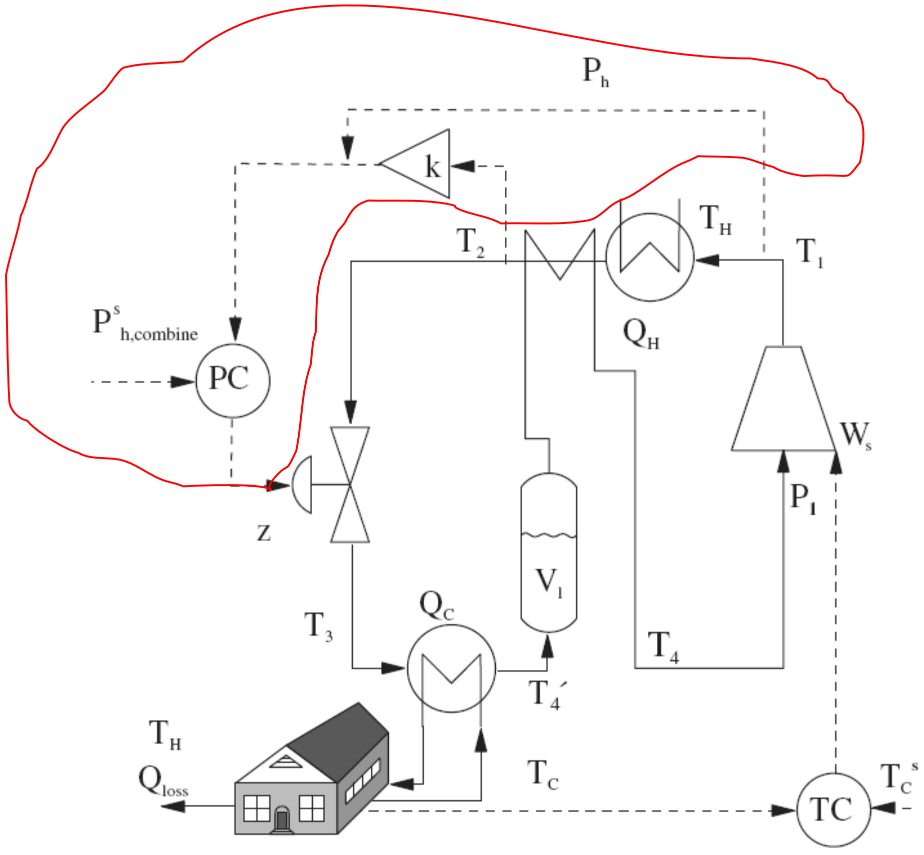
Nullspace method: $c = p_{h,\mathrm{combine}} = h_1 p_h + h_2 T_2 = p_h + k T_2; k = -8.53$ bar/K



J.B. Jensen, S. Skogestad / Computers and Chemical Engineering 31 (2007) 1590–1601

Refrigeration cycle: Proposed control structure

CV=Measurement combination



CV1= Room temperature
CV2= “temperature-corrected high CO2 pressure”

Summary Step 3. What should we control (CV_1)?

Selection of primary controlled variables $c = CV_1$

1. **Control active constraints!**
2. **Unconstrained variables: Control self-optimizing variables!**

- Self-optimizing control is an old idea (Morari *et al.*, 1980):

"We want to find a function c of the process variables which when held constant, leads automatically to the optimal adjustments of the manipulated variables, and with it, the optimal operating conditions."

Self-optimizing control (SOC)

- Local approximation: $c = Hy$.
- Need detailed steady-state model to find optimal H
  - Nullspace method
  - Exact local method
- Must reoptimize for each expected disturbance
- But calculations are offline

Challenges SOC:

- Nonlinearity
- Need new SOC variables for each active constraint region
  - Similar to multiparametric optimization and lookup tables

NTNU
Norwegian University of Science and Technology

8. Classical Approach: Optimal operation using conventional advanced control

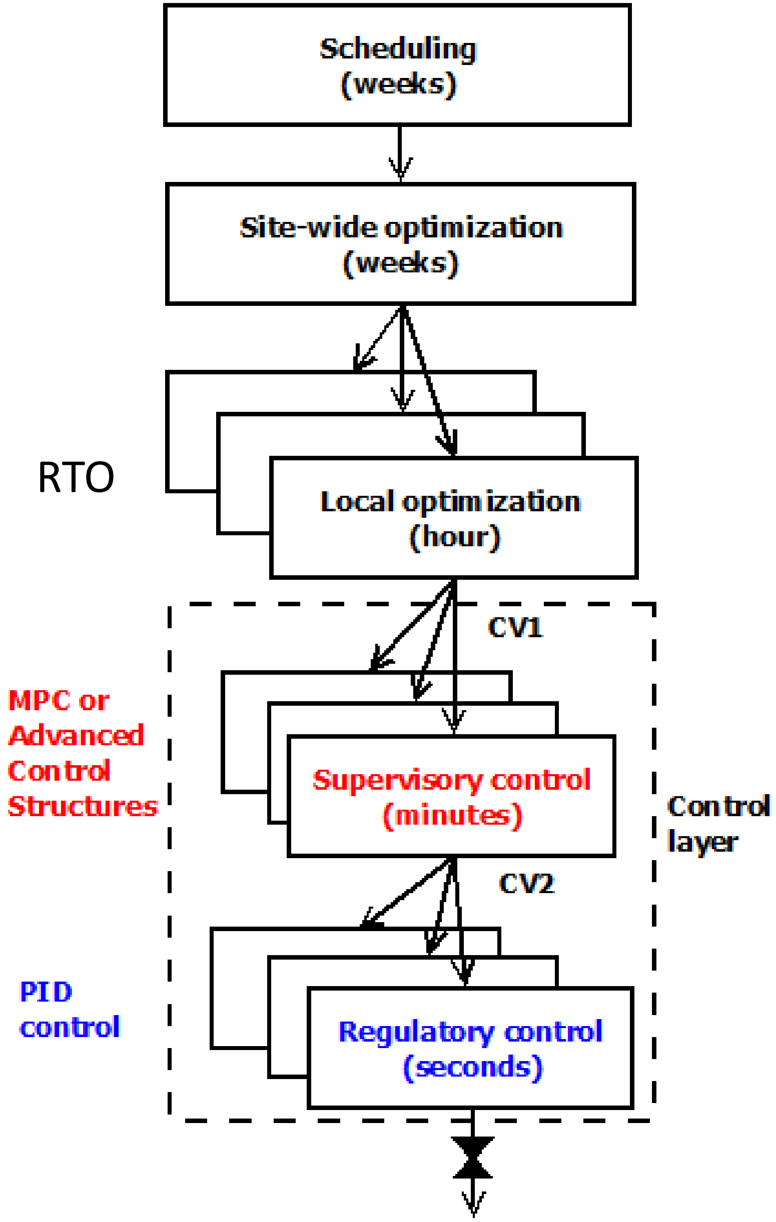
IFAC DYCOPS Pre-symposium workshop

Sigurd Skogestad

Supervisory control layer

Alternative implementations:

- Model predictive control (MPC)
- Classical advanced control structures (PID, selectors, etc.)



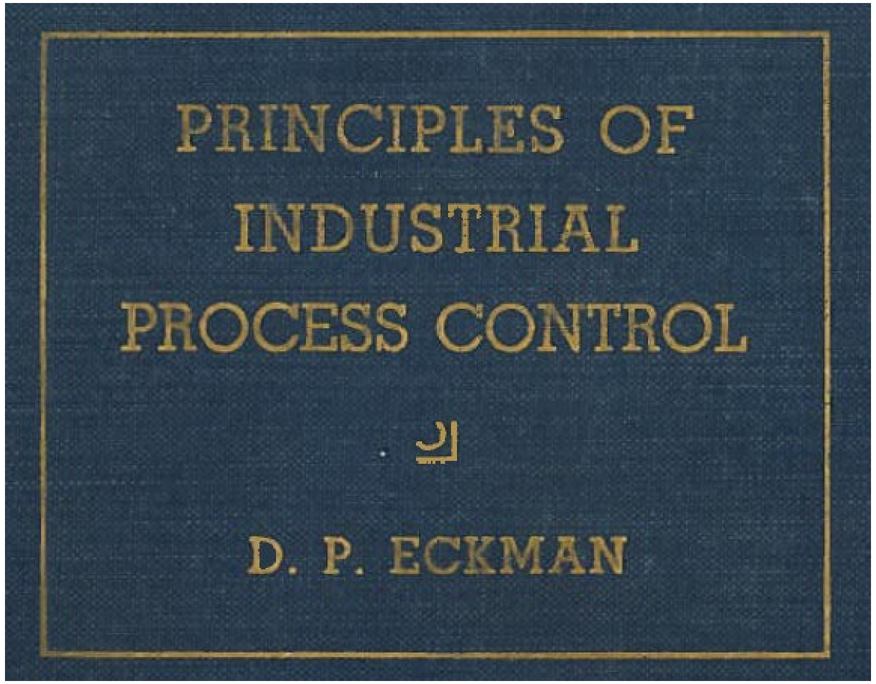
Classical "Advanced control" structures

1. Cascade control (measure and control internal variable)
2. Feedforward control (measure disturbance, d)
   - Including ratio control
3. Change in CV: Selectors (max,min)
4. Extra MV dynamically: Valve position control (=Input resetting =midranging)
5. Extra MV steady state: Split range control (+2 alternatives)
6. Multivariable control (MIMO)
   - Single-loop control (decentralized)
   - Decoupling
   - MPC (model predictive control)

Extensively used in practice, but almost no academic work

CV = controlled variable (y)
MV = manipulated variable (u)

Split range control: Donald Eckman (1945)



The temperature of plating tanks is controlled by means of dual control agents. The temperature of the circulating water is controlled by admitting steam when the temperature is low, or cold water when it is high. Figure 10–12 illustrates a system where pneumatic proportional control and diaphragm valves with split ranges are used. The steam valve is closed at 8.5 lb per sq in. pressure from the controller, and fully open at 14.5 lb per sq in. pressure. The cold water valve is closed at 8 lb per sq in. air pressure and fully open at 2 lb per sq in. air pressure.

If more accurate valve settings are required, pneumatic valve positioners will accomplish the same function. The zero, action, and range adjustments of valve positioners are set so that both the steam and cold water valves are closed at 8 lb per sq in. controller output pressure. The advantages gained with valve positioners

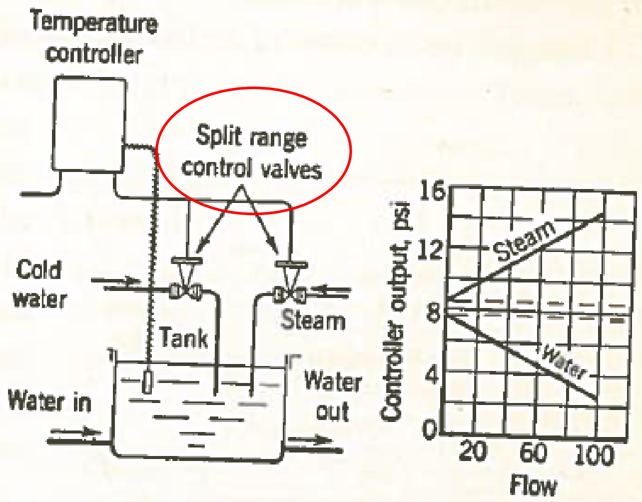


Fig. 10–12. Dual-Agent Control System for Adjusting Heating and Cooling of Bath.

Switching between active constraints

1. Output to Output (CV - CV) switching (SIMO)
   - Selector

2. Input to output (CV – MV) switching
   - Do nothing if we follow the pairing rule: «Pair MV that saturates with CV that can be given up»

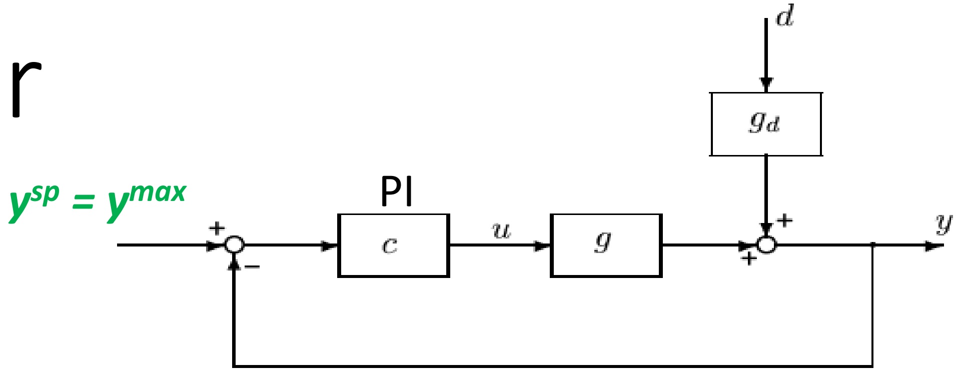
3. Input to input (MV – MV) switching (MISO)
   - **Split range control**
   - OR: Controllers with different setpoint value
   - OR: Valve position control (= midranging control)

Optimization with PI-controller

max y

s.t. $y \leq y^{max}$

$u \leq u^{max}$



Example: Drive as fast as possible to airport (u=power, y=speed, y^{max} = 120 km/h)

- **Optimal solution has two active constraint regions:**
  1. $y = y^{max}$ → **speed limit**
  2. $u = u^{max}$ → **max power**
- Note: Positive gain from MV (u) to CV (y)
- **Solved with PI-controller**
  - $y^{sp} = y^{max}$
  - Anti-windup: I-action is off when $u=u^{max}$

s.t. = subject to

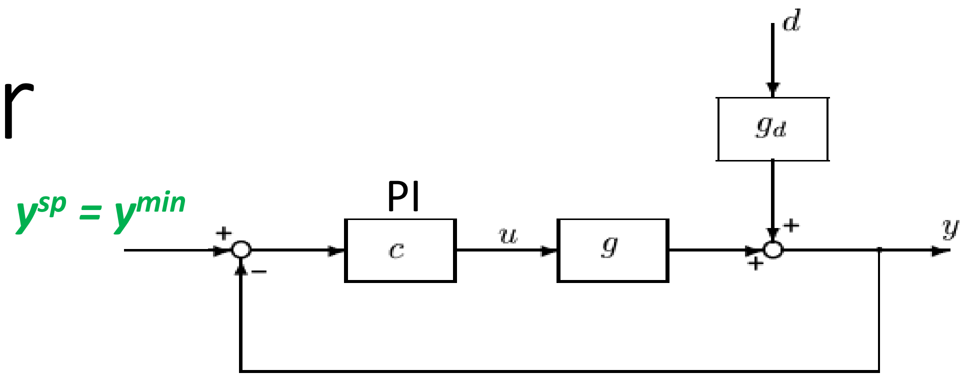
y = CV = controlled variable

Optimization with PI-controller

min u

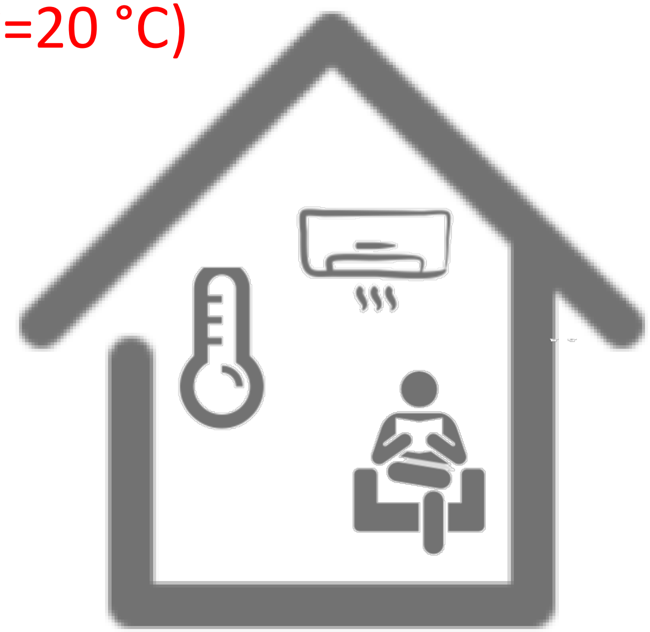
s.t. $y \geq y^{min}$

$u \geq u^{min}$



Example: Minimize heating cost (u=heating, y=temperature, y^{min}=20 °C)

- **Optimal solution has two active constraint regions:**
  1. ***$y = y^{min}$* → minimum temperature**
  2. ***$u = u^{min}$* → heating off**
- Note: Positive gain from MV (u) to CV (y)
- **Solved with PI-controller**
  - $y^{sp} = y^{min}$
  - Anti-windup: I-action is off when $u=u^{min}$



s.t. = subject to

y = CV = controlled variable

NTNU

Optimization with PI-controller

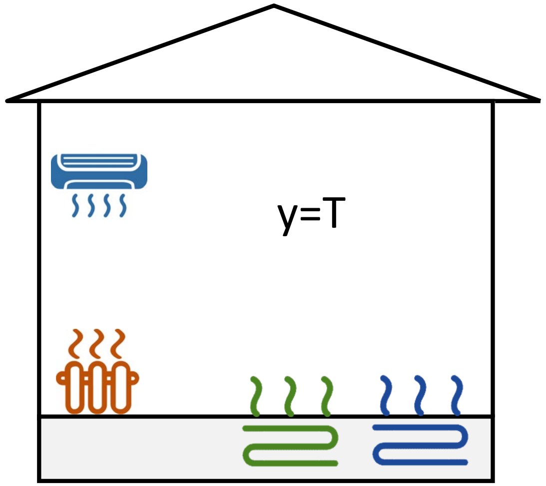
The two examples:

- Optimal operation: Switch between CV constraint and MV saturation
- A simple PI-controller was possible because we followed the pairing rule:

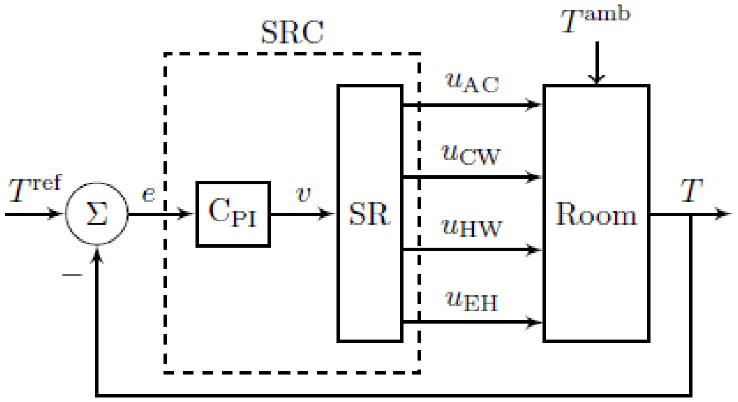
  «Pair MV that saturates with CV that can be given up»

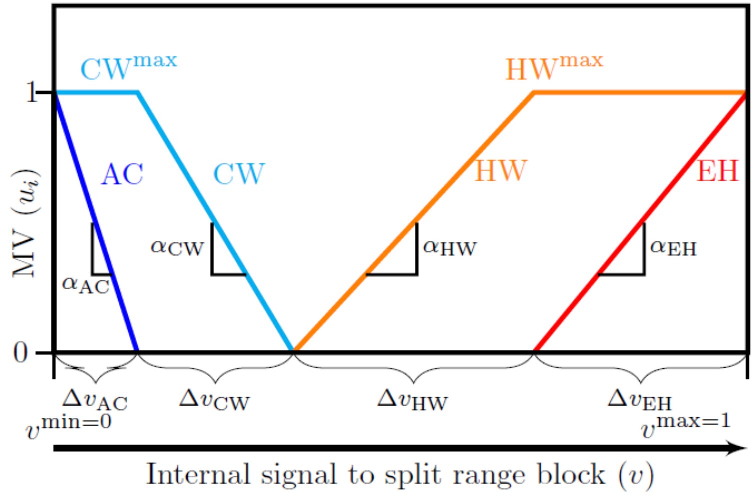
Split-range control (SRC): One CV (y). Two or more MVs (u1,u2)

Example: Room heating with 4 MVs



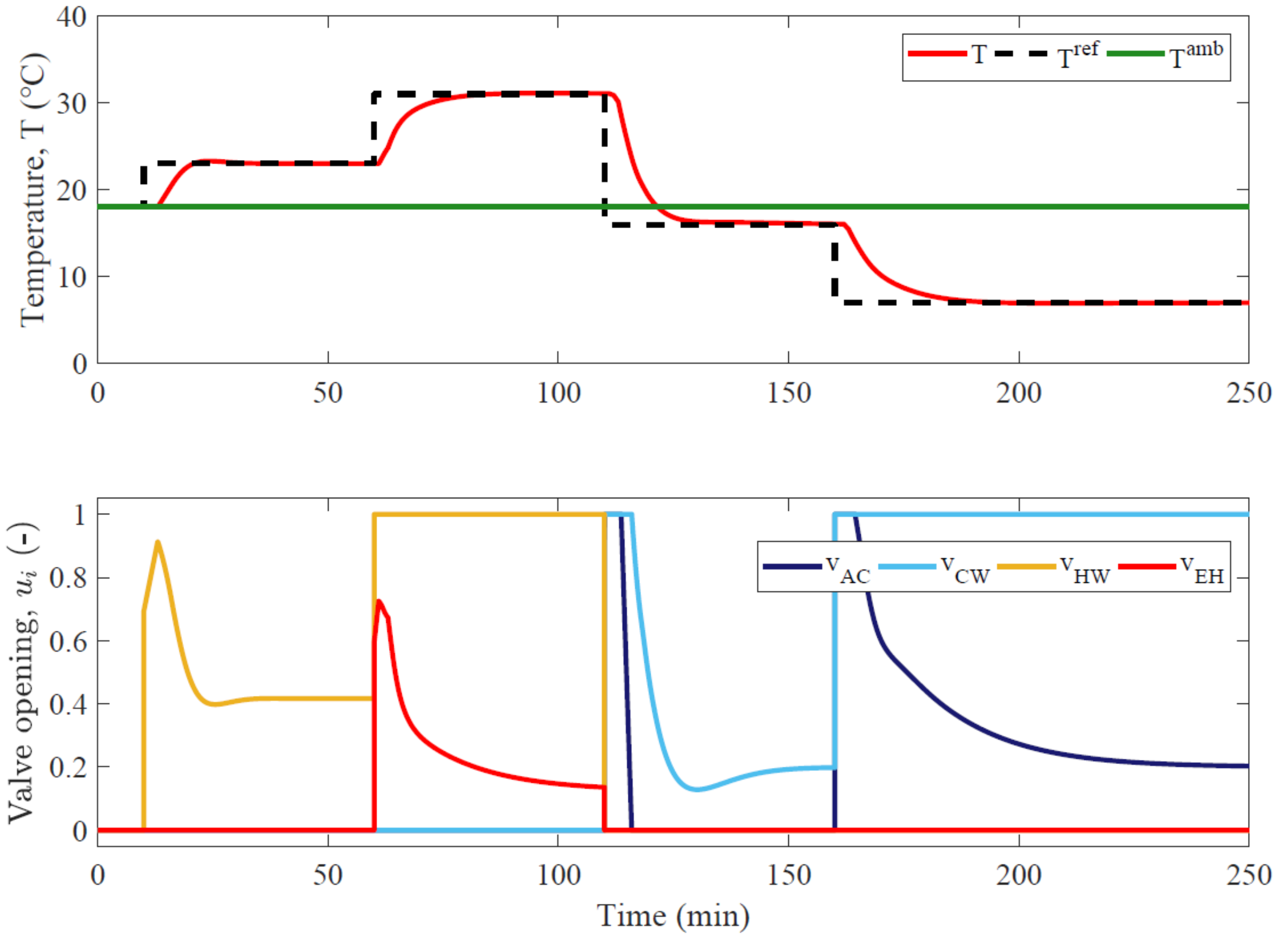
MVs:
1. AC (expensive cooling)
2. CW (cooling water; cheap)
3. HW (hot water, quite cheap)
4. Electric heat, EH (expensive)





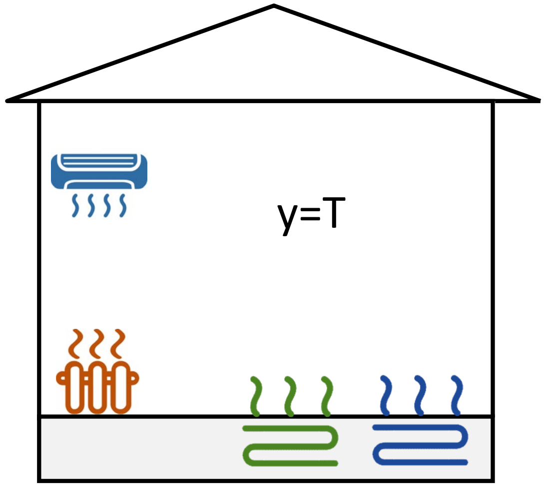
MV-MV switching

Simulation PI-control: Setpoint changes temperature



MV-MV switching

Example: Room heating with 4 MVs

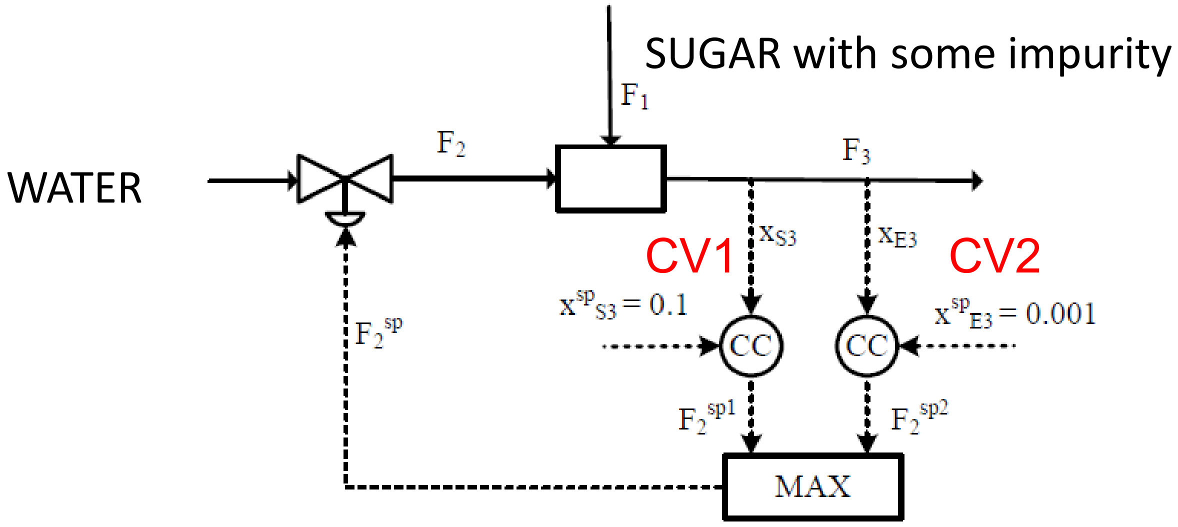


MVs:
1. AC (expensive cooling)
2. CW (cooling water; cheap)
3. HW (hot water, quite cheap)
4. Electric heat, EH (expensive)

Three Alternatives:

1. **Split range control (SP=22C)**
2. **Controllers with different setpoint values (SP=24C, 23C, 22C, 21C)**
3. **Valve position control (= midranging control)** (Use always HW for SP=22C)

Blending process with max selector



MV = Water feed (F_2)

CV1 = Sugar concentration (Should be at SP=0.1 whenever feasible|)
CV2 = Impurity concentarion (Max. 0.001)

Disturbances: Variation in sugar feed (F_1) and concentration of impurity in sugar

CV-CV switching

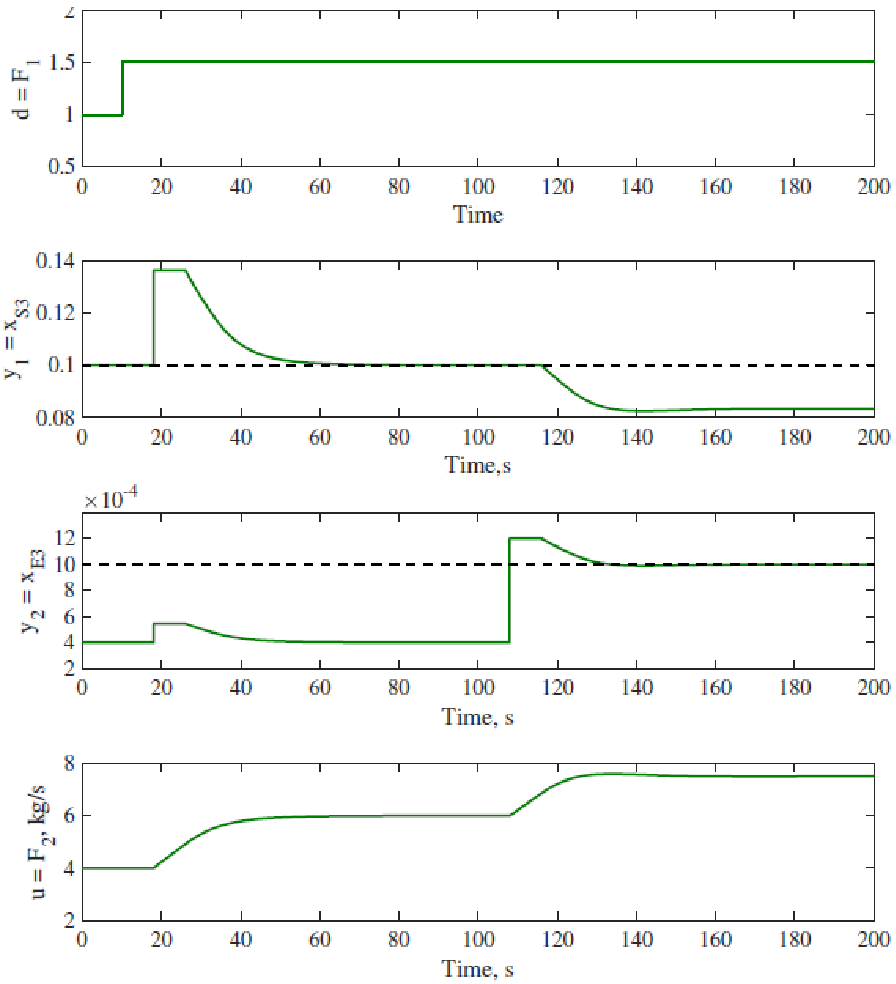


Figure 10: Simulation results for a step disturbance $F_1 = 1.5\,kg/s$ at $t = 10\,s$ and $x_{E1} = 0.006$ at $t = 100\,s$ (extreme case). The black dotted lines show the concentration specification for x_{S3} and x_{E3} respectively. In the normal case, the controller is controlling $y_1 = x_{S3}$ at $y_{1s} = 0.1$, while in the extreme case, the controller is controlling $y_2 = x_{E3}$ at $y_{2s} = 0.001$.

Conclusion: Systematic procedure to avoid RTO-layer and even MPC-layer

Start "top-down" with economics (steady state):

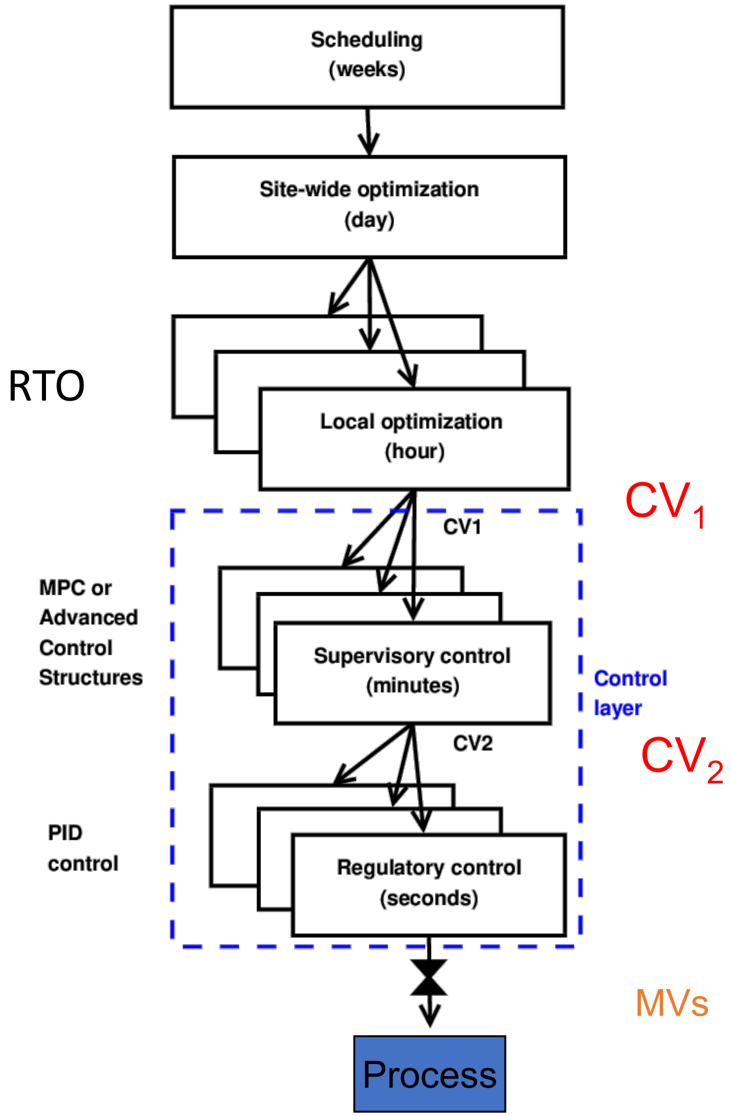
- Step 1: Define operational objectives and constraints
- Step 2: Optimize steady-state operation
- Step 3: Decide what to control (CVs)
  - Step 3A: Identify active constraints = primary CV1.
  - Step 3B: Remaining unconstrained DOFs: Self-optimizing CV1 (find H)
- Step 4: Where do we set the throughput? TPM location

Then bottom-up (dynamics):

- Step 5: Regulatory control
  - Control variables to stop "drift" (sensitive temperatures, pressures, ...)

Finally: Make link between "top-down" and "bottom up"

- Step 6: "Advanced/supervisory control"
  - Control economic CVs: Active constraints and self-optimizing variables
  - Look after variables in regulatory layer below (e.g., avoid saturation)



S. Skogestad, ``Control structure design for complete chemical plants'', *Computers and Chemical Engineering*, **28** (1-2), 219-234 (2004).

NTNU
Norwegian University of
Science and Technology

Hierarchical Combination of different approaches

IFAC DYCOPS Pre-symposium workshop

Johannes Jäschke
Dinesh Krishnamoorthy

Why not combine different approaches to give improved performance?!

Examples:

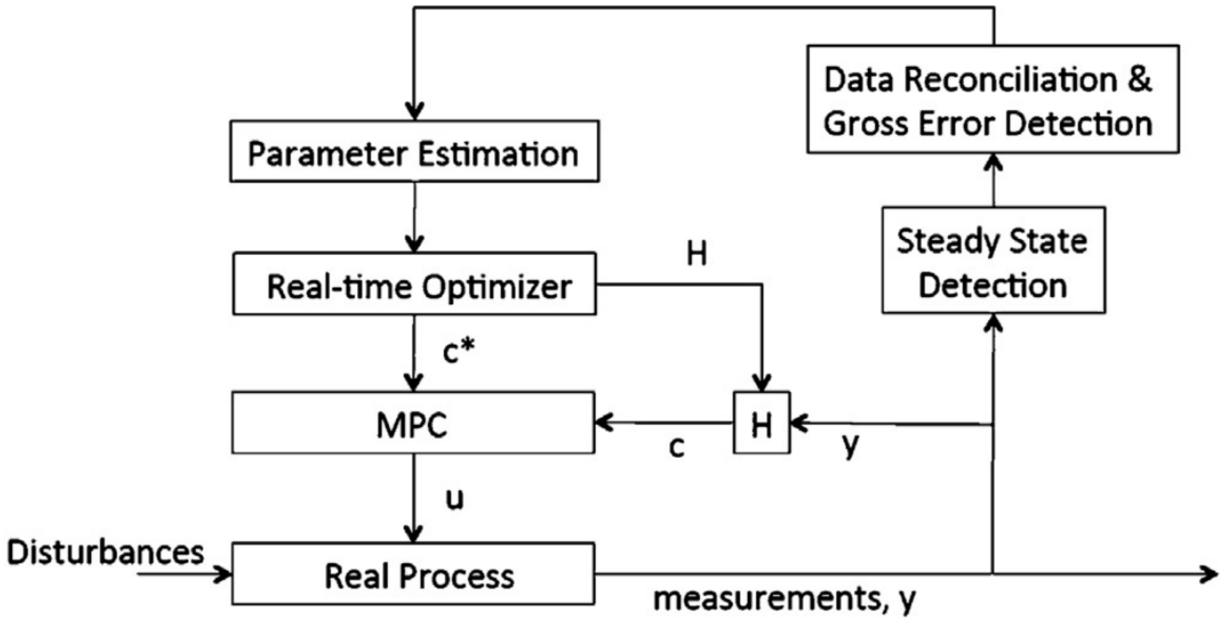
- Combining model and data-based approaches
- Combining online and offline methods

Some benefits

- Faster rejection of known disturbances
- Capability of handling unmodeled disturbances

Standart RTO + MPC + self-optimizing control

Idea: take the best from all worlds

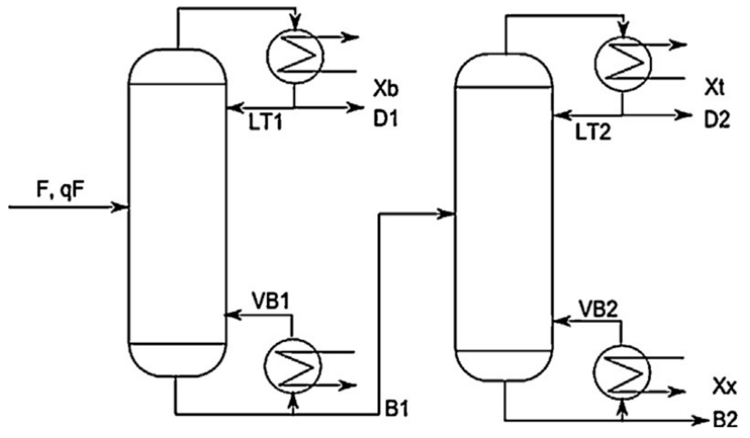


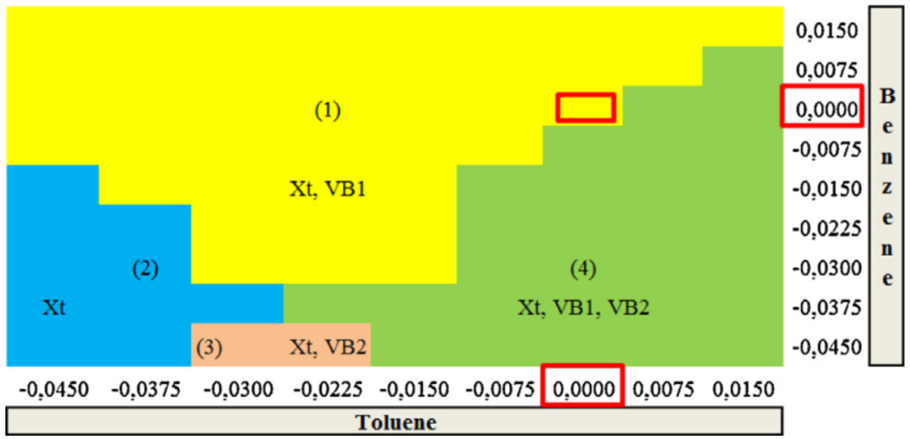
Graciano et al. 2015, Journal of Process Control 34, 35–48

- Self-optimizing Control
  - Fast correction for known and modelled disturbances
- MPC:
  - Predicting responses, and good constraint handling
- Standard RTO:
  - Handling nonlinearity and large disturbances optimally

Distillation Case Study

- Separation of 3 components





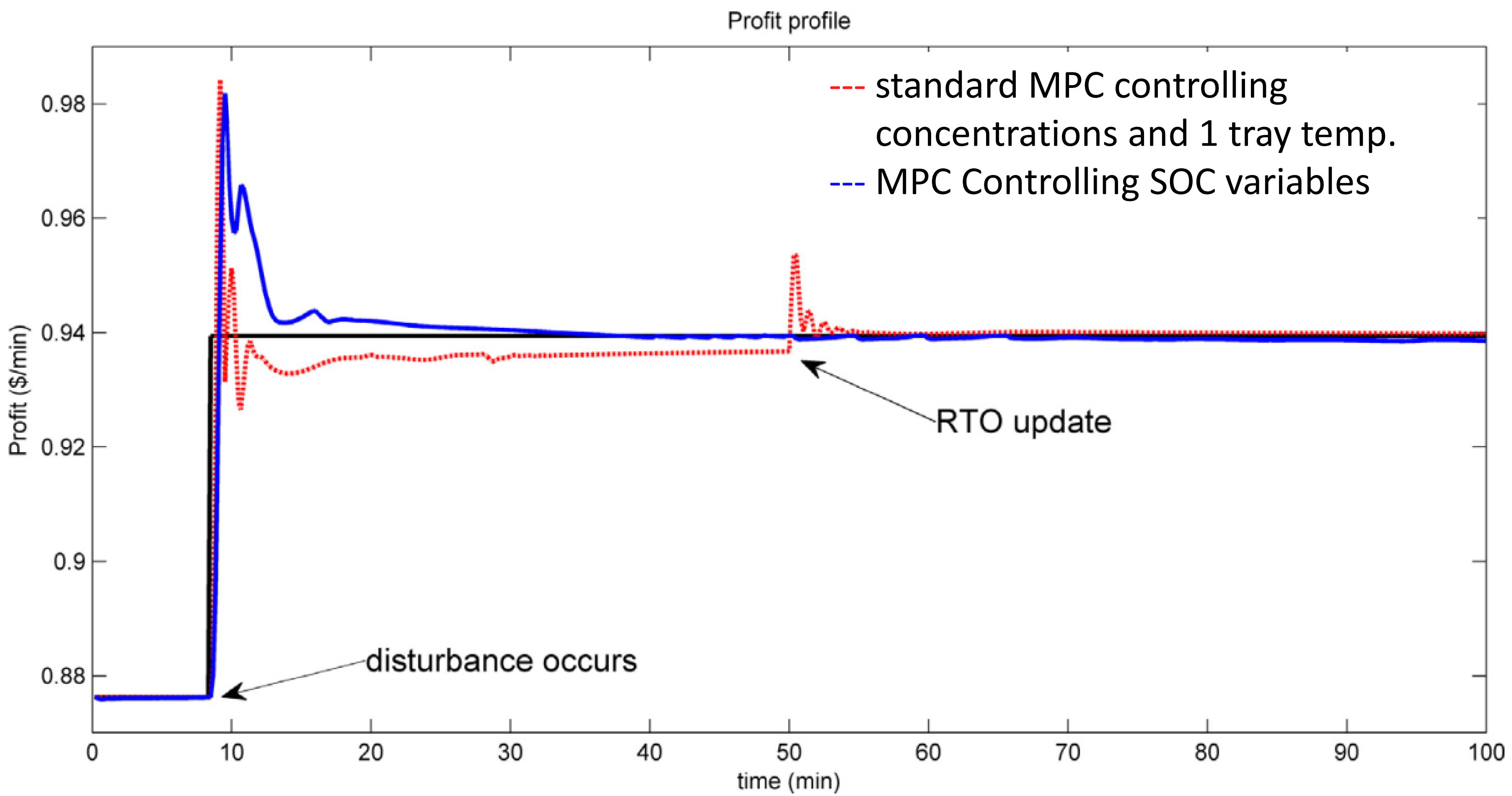
RTO Problem

$$\min_{u} \quad Cost^{opt} = p_F F + p_V(VB1 + VB2) - p_B D1 - p_T D2 - p_X B2$$

$$Xb \geq 0.95$$
$$Xt \geq 0.95$$
$$Xx \geq 0.95$$
$$VB1 \leq 4.080\,[\mathrm{kmol}/min]$$
$$VB2 \leq 2.405\,[\mathrm{kmol}/min]$$

- MPC
- Setpoint tracking
  - Standard (concentrations+temperature)
  - Self-optimizing variable combinations
- Constraint handling
  - Enforce constraints as they become active

Profit of combined approach



- Disturbances are rejected also inbetween RTO updates.

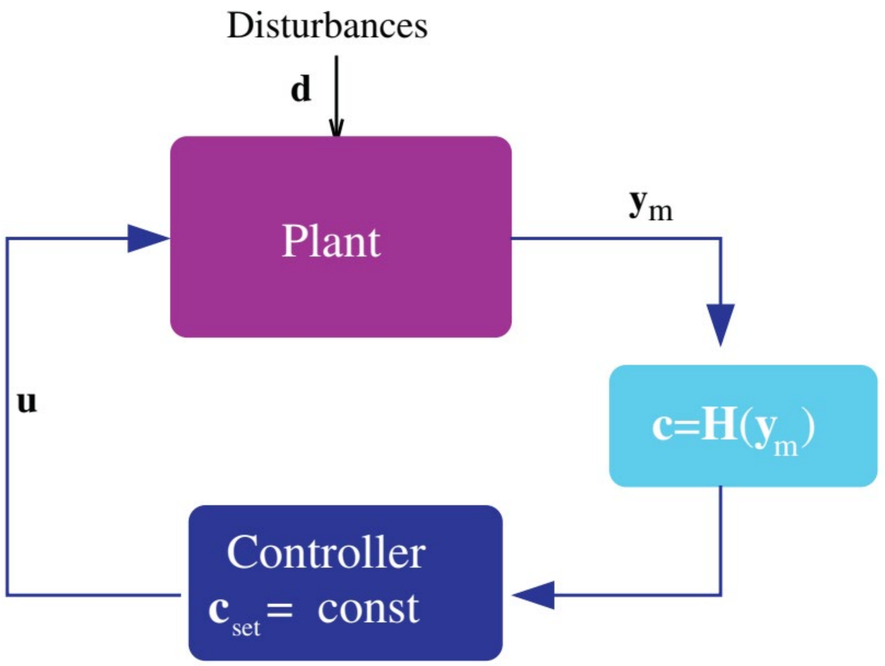
NCO tracking + self-optimizing control

- Idea: take the best from both worlds
  - Self-optimizing Control: Fast correction for known and modelled disturbances
  - NCO tracking: use Plant gradient estimates to handle unmodeled disturbances

Self-optimizing Control and NCO tracking

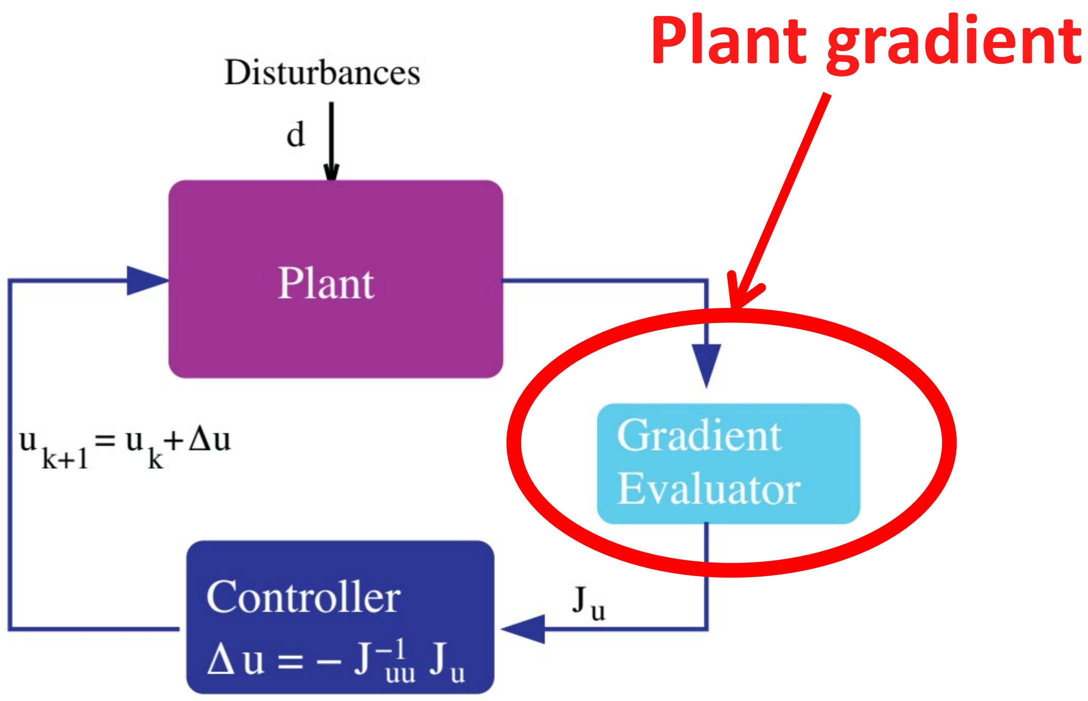
Self-optimizing control

- Find a good output combination
- Control to zero with favourite controller



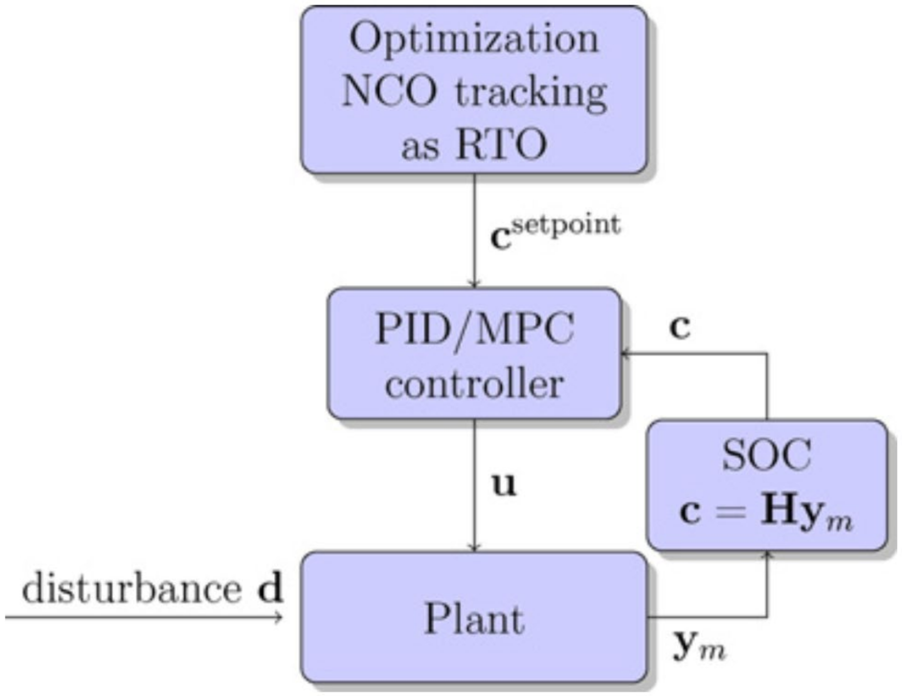
NCO tracking idea

- Measure gradient
- Adjust input to make it zero



NCO: Necessary conditions of Optimality

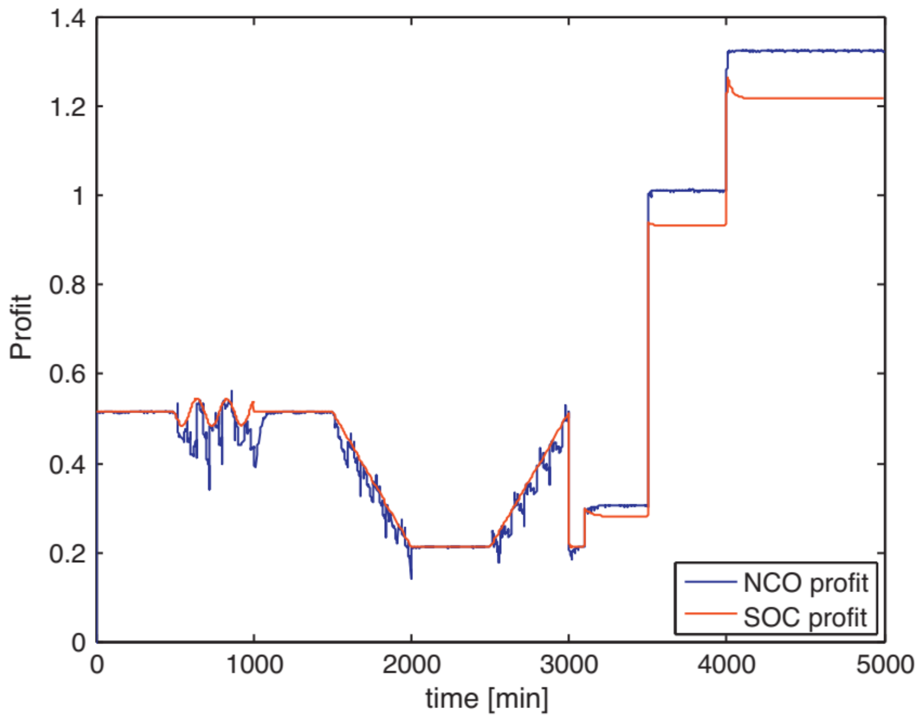
Combination of self-optimizing control and NCO tracking



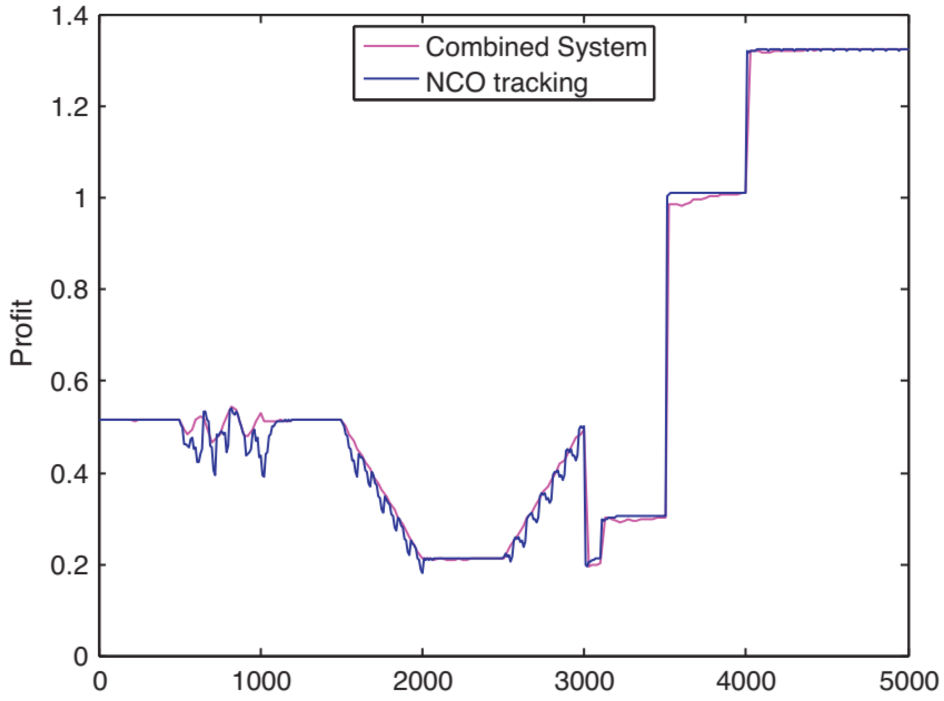
Jäschke & Skogestad (2011)

- Fast time scale (lower layer):
  - reject known (modelled) disturbances using self-optimizing control
- Slow time scale (upper layer)
  - reject unknown (unmodeled) disturbances in NCO tracking

- Self-optimizing and NCO tracking (sampling time 10 min)

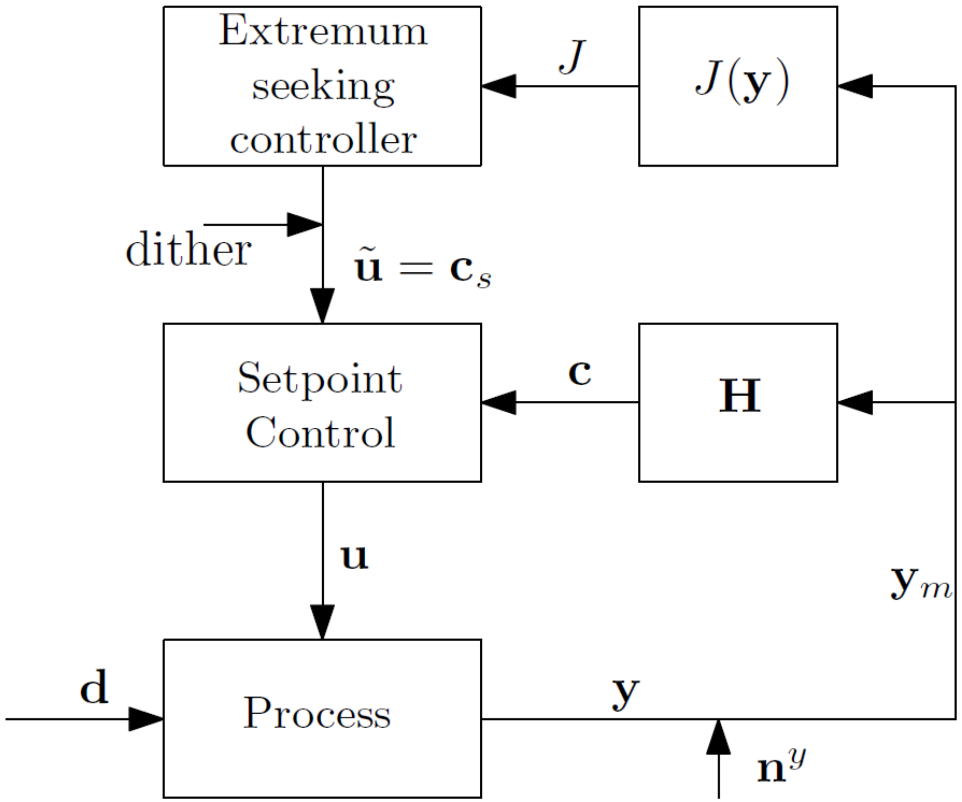


- Combined Self-optimizing and NCO tracking (sampling time 25 min)



Similar approach: ESC/RTO + Self-optimizing control

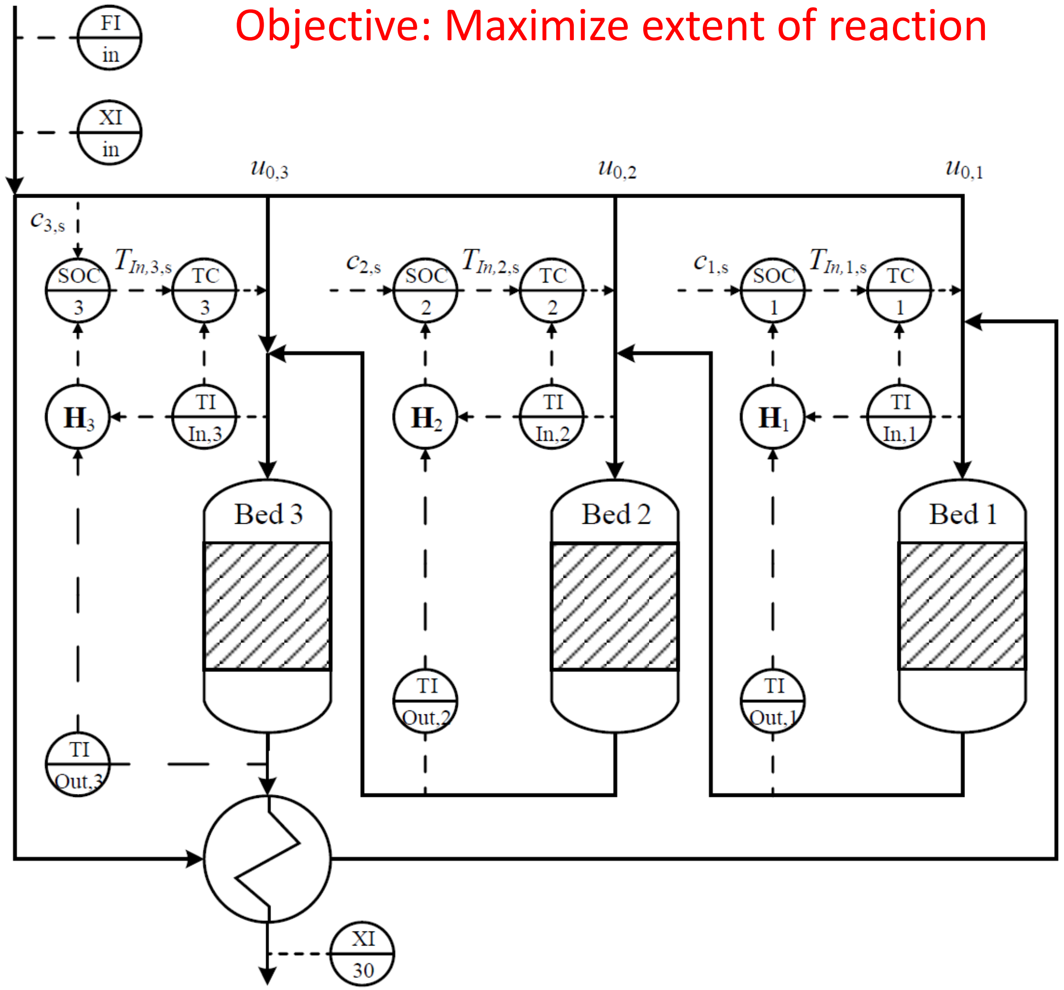
- Self-optimization control is always complementary
- Can combine with
  - Extremum-seeking control
  - Traditional Static RTO

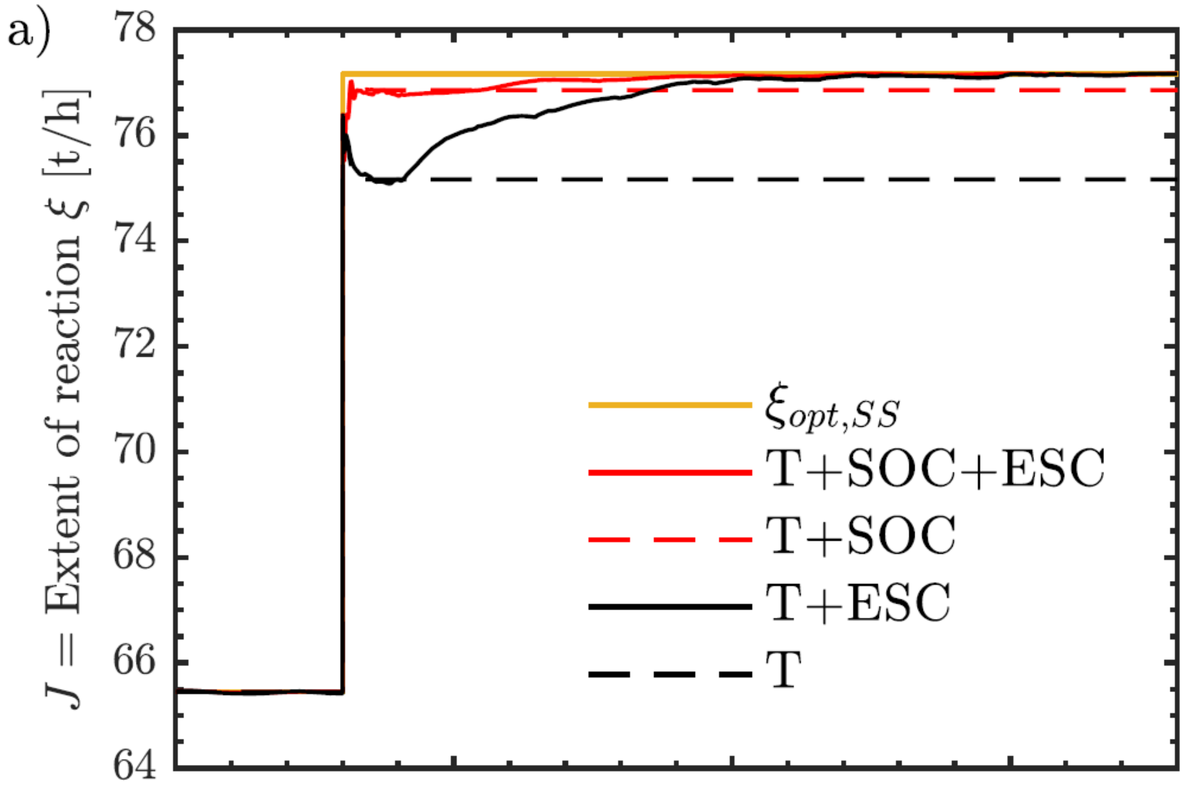


Straus. J, Krishnamoorthy, D., and Skogestad, S., 2018. On combining extremum seeking control and self-optimizing control. J. Proc. Control (In-Press).

Case study: Ammonia Reactor

Objective: Maximize extent of reaction





Response to disturbance in inlet mass flow rate

CONCLUSION: Why is traditional static RTO not commonly used? Some alternatives

1. Cost of developing and updating the model (costly offline model update)

   → Fix: estimate plant gradients directly, like extremum-seeking - Machine learning (new)

2. Wrong value of model parameters and disturbances (slow online model update)

   → Fix: DRTO, HRTO, self-optimizing control (fastest)

3. Not robust, including computational issues

   → Fix: Feedback RTO, self-optimizing control

4. Frequent grade changes make steady-state optimization less relevant

   → Fix: Dynamic RTO (DRTO) or EMPC

5. Dynamic limitations, including infeasibility due to (dynamic) constraint violation

   → Fix: DRTO, EMPC (also HRTO ok!)

6. Incorrect model Structure

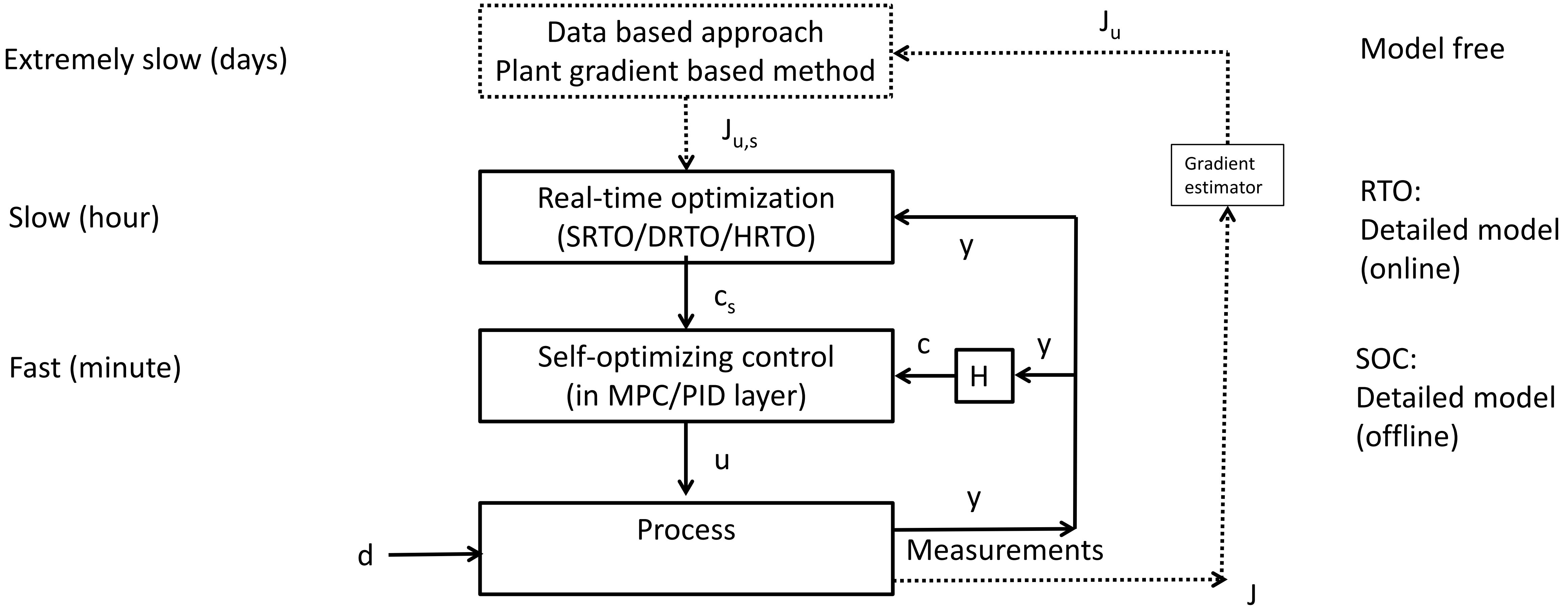
   → Fix: Modifier adaptation

Proposal : Combine RTO with other approaches

- ESC / modifier adaptation layer: make RTO approach the real optimum .

- SOC layer: make optimization faster, reduce wait time for model update and online optimization

Conclusion



- **Thank you !**

- Next slide

  Red box = bad,
  green box = good,

  No box = neutral

	self-optimizing control[1]	extremum seeking control[2]	new proposed method (Feedback RTO)	Static RTO	Hybrid RTO	economic MPC/ Dynamic RTO[3]
Cost Measured	No	$\boxed{\text{Yes}}$	No	No	No	No