Overview of Online Process Optimization Approaches

Sigurd Skogestad, Dinesh Krishnamoorthy

Department of Chemical Engineering, NTNU

dinesh.krishnamoorthy@ntnu.no

16 February 2020, IIT Madras

NTNU
Norwegian University of Science and Technology


NTNU

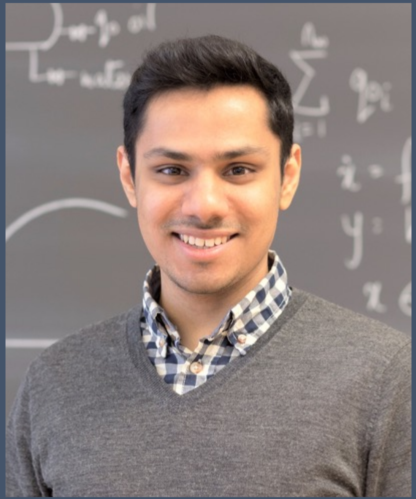

Trondheim
Sweden
Faroe Islands
Norway
Oslo
Stockholm
Edinburgh
Copenhagen
Denmark
United Kingdom
Isle of Man
Dublin
Manchester
Liverpool
Ireland
Hamburg
Amsterdam
Berlin
Netherlands
London
Brussels
Cologne
Germany

Presenters

Sigurd Skogestad

Professor NTNU

PhD Caltech (1987)



Dinesh Krishnamoorthy

Post-doc NTNU

PhD, NTNU (2019)

MS Imperial College (2012)

Introduction and Challenges

Online Process Optimization

Design vs. Operation

Design Optimization

Given certain user needs, what car should I buy

- horsepower?
- Engine size?
- Petrol or diesel engine?
- Miles per gallon?

Operational (Online) Optimization

Given a car, operate the car in an optimal way, on any given day!

- Speed ?
- Acceleration?
- Lane ?
- Distance to car?

Online Process Optimization

- Online operation
  - Adjust operational DOF:
  - Optimal operating conditions depending on actual disturbances
  - Minimize operational cost s.t constraints

$$J\ [\$/h] = p_F F + p_Q Q - p_p P$$

Continous Optimization (NLP) performed online

Operation ≈ minimize cost subject to constraints

Biegler, L.T., 2010. *Nonlinear programming: concepts, algorithms, and applications to chemical processes* (Vol. 10). Siam.

Main objectives during operation

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

2. Regulation: Stable operation

Focus of this workshop

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)

Objectives →

Present state →

Model of system →

CENTRALIZED OPTIMIZER

$$\min_{u} J(x,u,d)$$
$$\text{s.t. } \dot{x} = f(x,u,d),$$
$$h(x,u,d) = 0,$$
$$g(x,u,d) \leq 0.$$

(Physical) Degrees of freedom, u= valve positions

Approach:

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

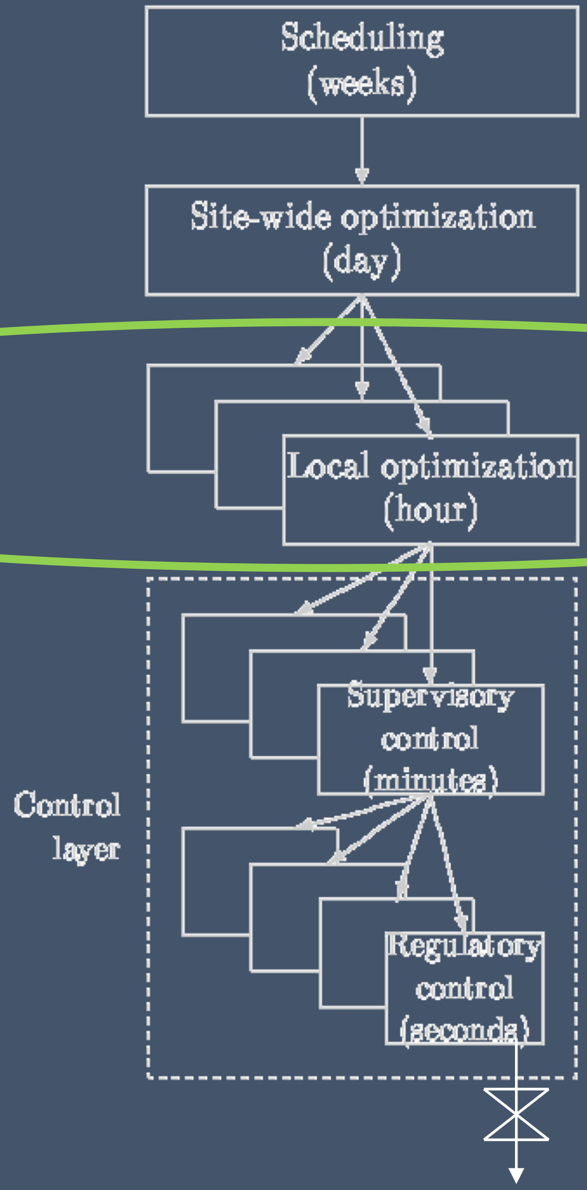
Process control:

- 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 based on timescale 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

Skogestad (2000)

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)

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

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

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

where

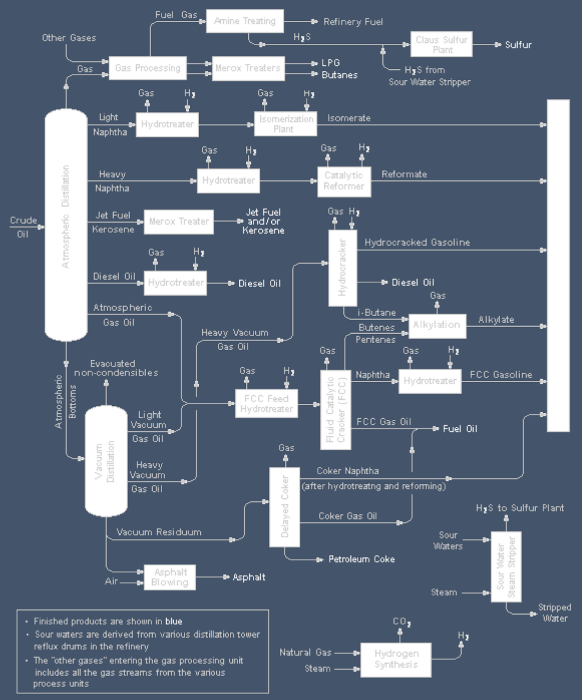
- $\sum p_F F$ = price of feed [$/kg] x feed flow rate [kg/s]
- $\sum p_Q Q$ = price of utility (energy) x energy usage
- $\sum p_P P$ = price (value) of product x product flow rate

Typical process constraints:

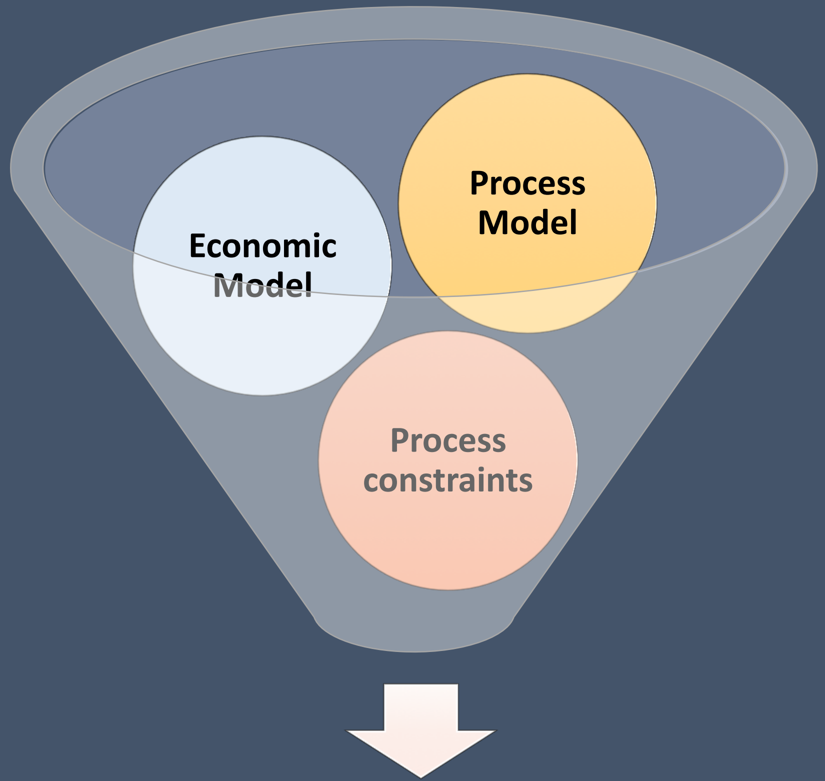
- Product quality (purity)
- Environment (amount and purity of waste products)
- Equipment (max. and min. flows, pressures)

Typical degrees of freedom (decision variables) (u)

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



Formulation of Real time optimization (RTO)



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

$$\min J(\mathbf{x}, \mathbf{u}, \mathbf{d})$$

s.t.

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

$$\mathbf{g}(\mathbf{x}, \mathbf{u}, \mathbf{d}) \leq 0$$

$$\mathbf{x} \in \mathcal{X}, \qquad \mathbf{u} \in \mathcal{U}$$

x : Internal variables

u: Decision variables

d: Parameter values / disturbances

Steady-state optimization

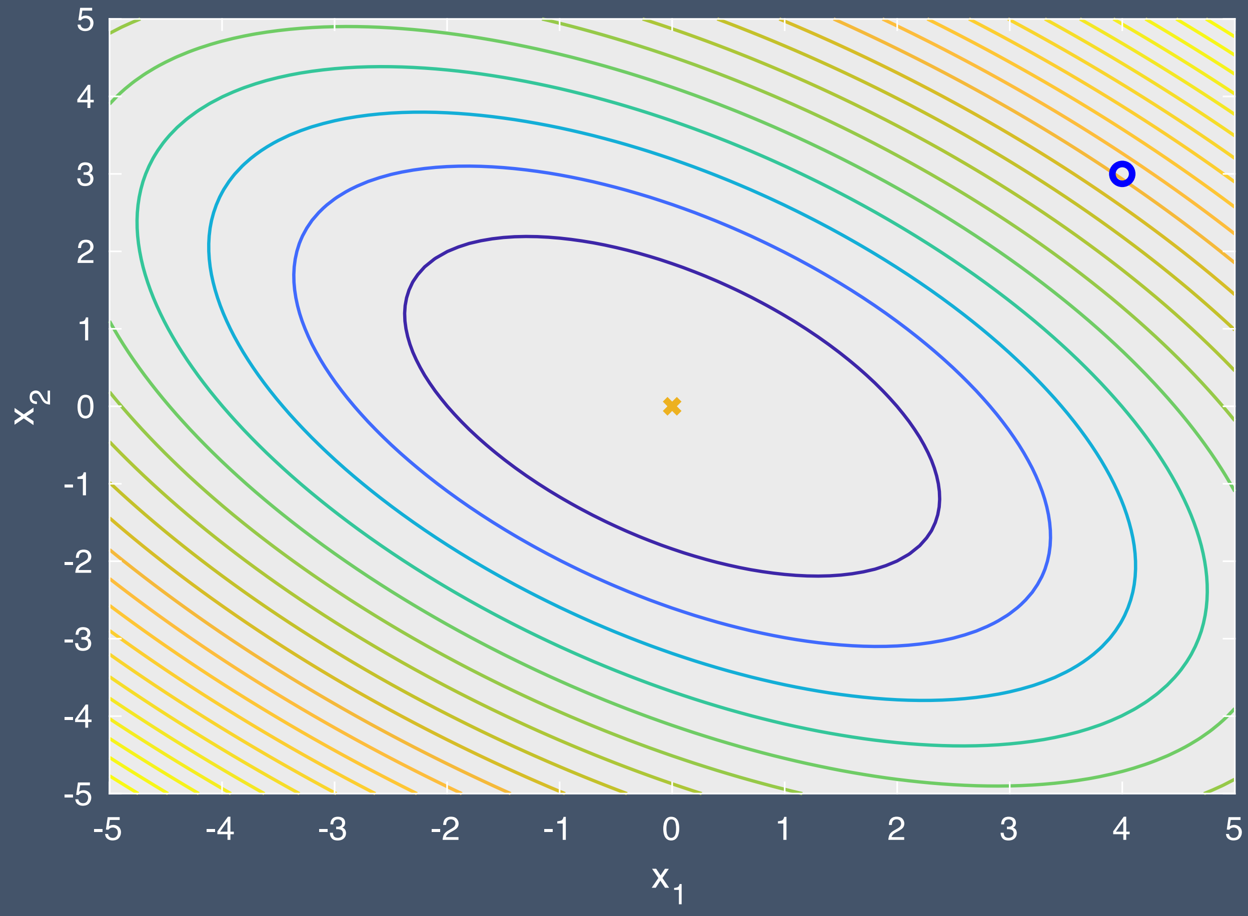
$$\min J(\mathbf{x}, \mathbf{u}, \mathbf{d})$$

s.t.

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

$$\mathbf{g}(\underbrace{\mathbf{x}, \mathbf{u}, \mathbf{d}}_{\mathbf{w}}) \leq 0$$

Find optimal operating point



Dynamic optimization

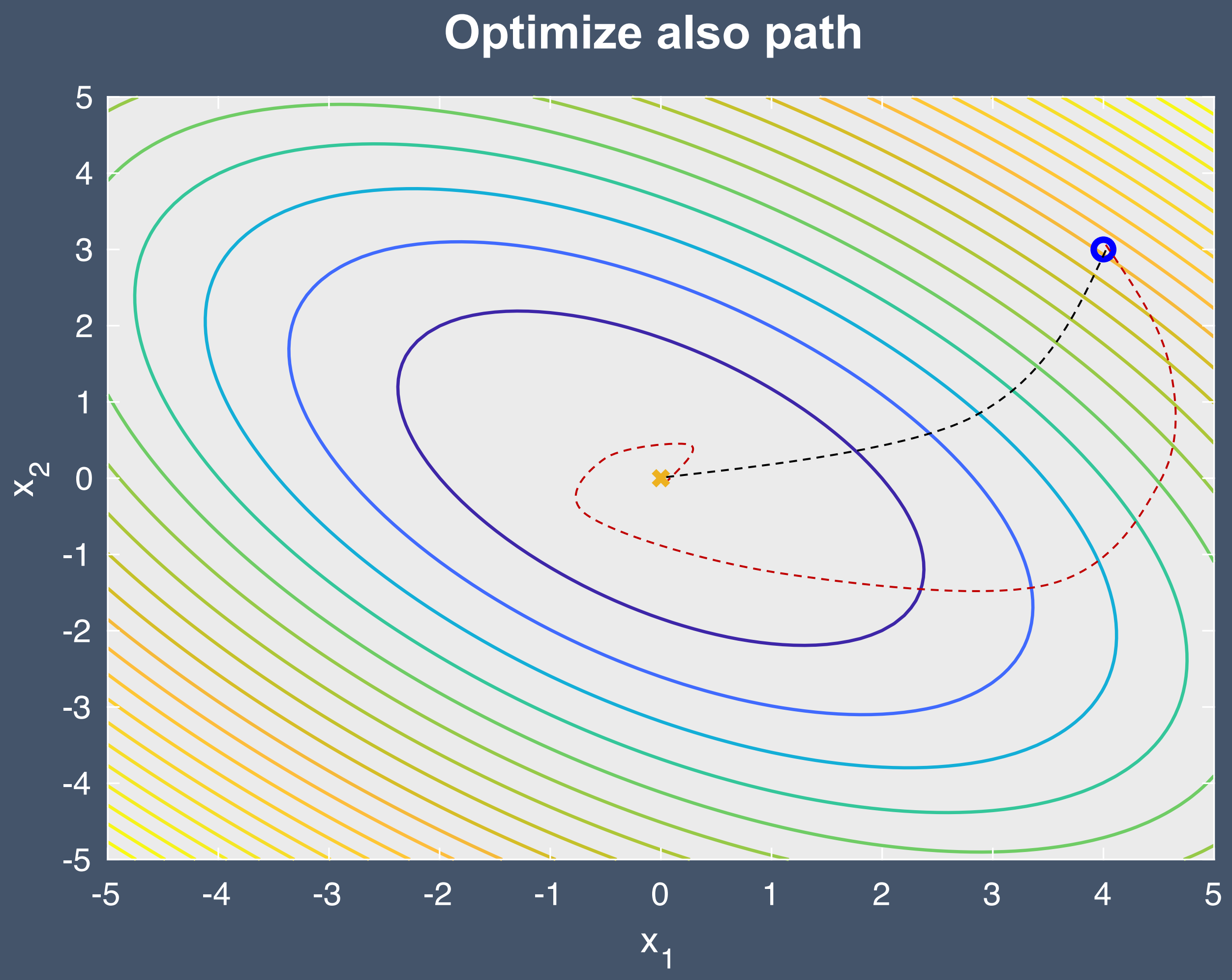
$$\min \sum_{t=1}^{T} J(\mathbf{x}(t), \mathbf{u}(t), \mathbf{d}(t))$$

s.t.

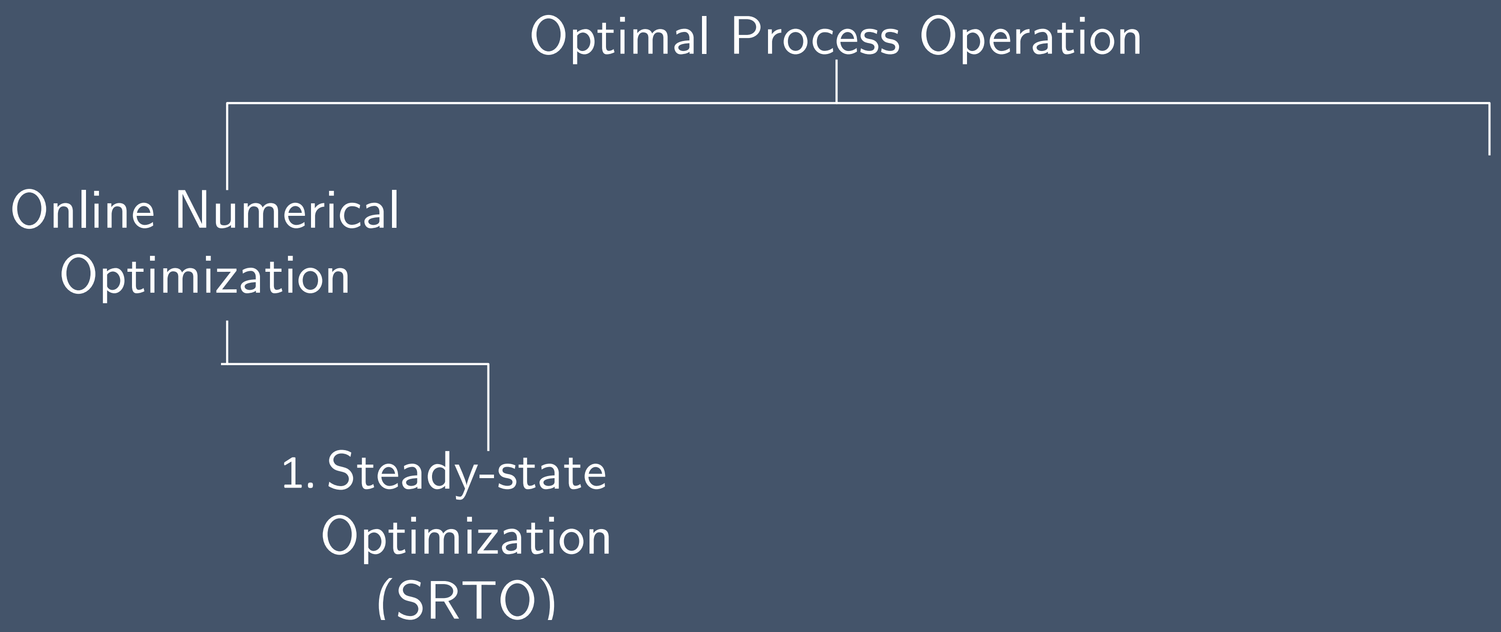
$$\mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), \mathbf{d}(t)) = \dot{\mathbf{x}}(t)$$

$$\mathbf{g}(\mathbf{x}(t), \mathbf{u}(t), \mathbf{d}(t)) \leq 0$$

$$\mathbf{x}(0) = \hat{\boldsymbol{x}}$$

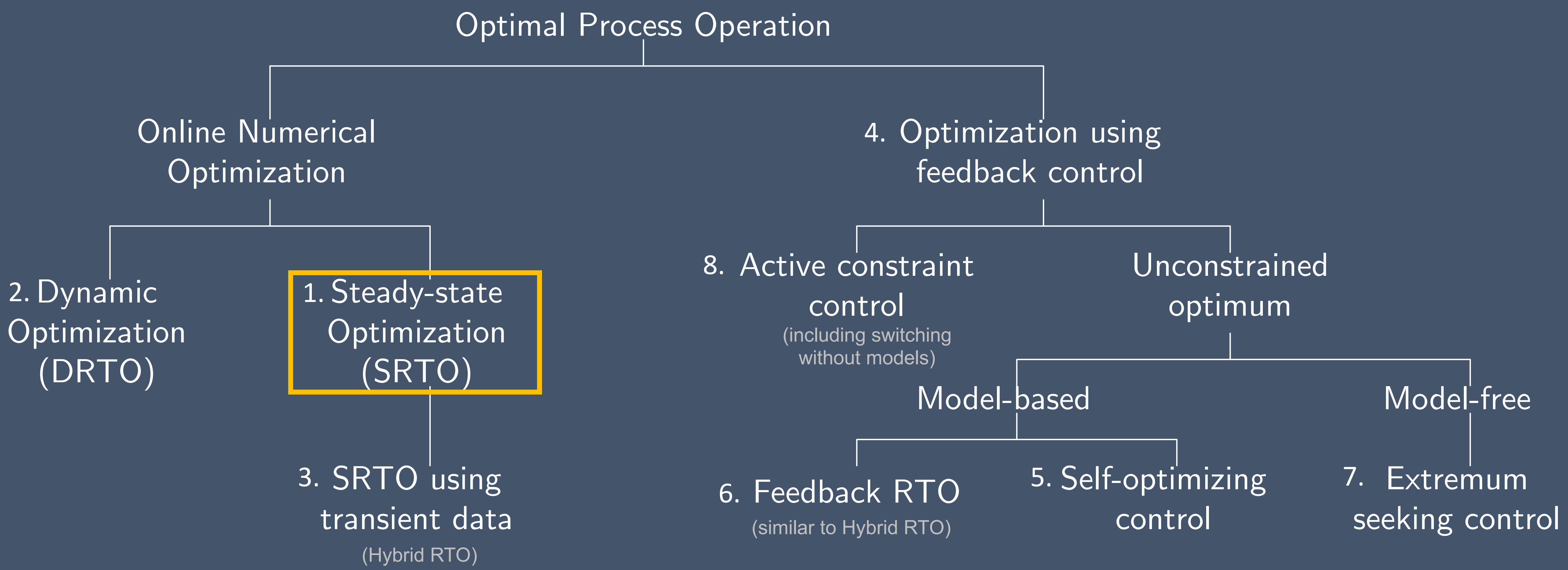


Workshop Roadmap – RTO Toolbox



Take home message: The different methods have their pros and cons. Choose the right tool for the problem at hand.

Workshop Roadmap

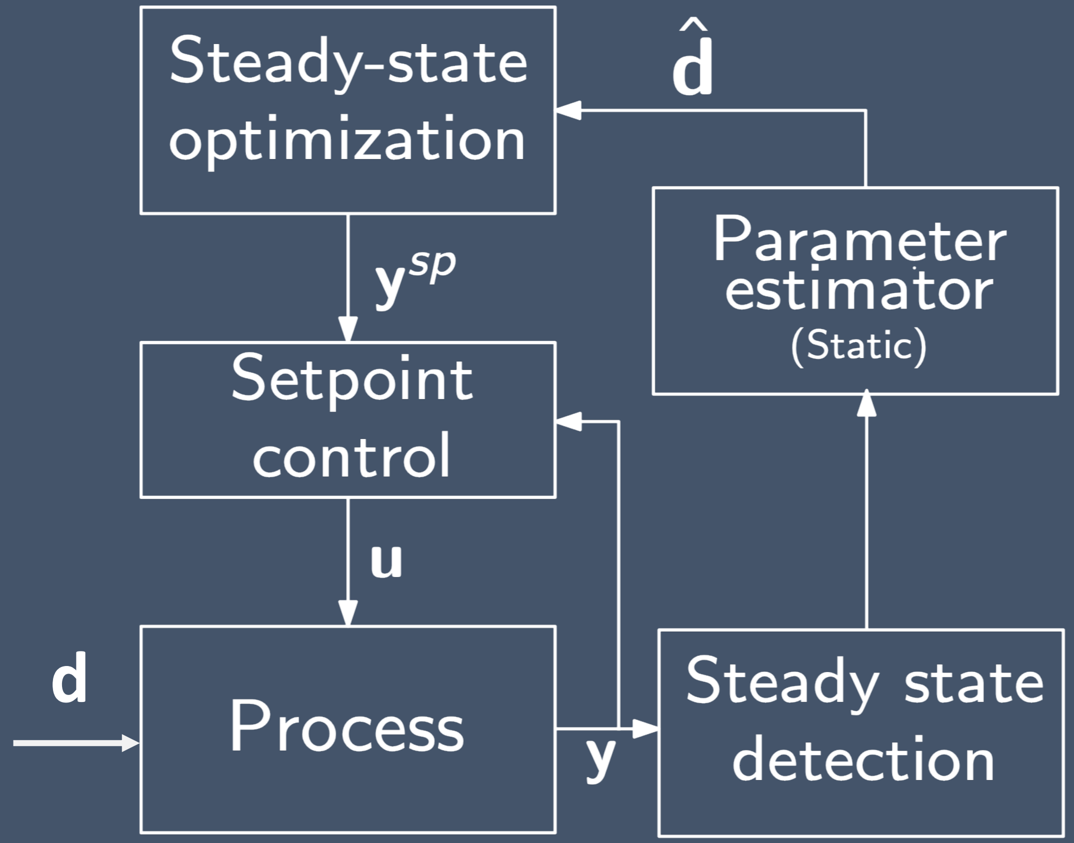


1. Conventional Steady-state Real time optimization (SRTO)

Commercial approach

Conventional (commercial) steady-state RTO

Conventional (commercial) steady-state RTO



Step 1: Steady-state Estimation

$$\hat{\mathbf{d}}_k = \arg\min_{\mathbf{d}_k} \|\mathbf{y}_{meas} - \mathbf{f}_{ss}(\mathbf{u}_k, \mathbf{d}_k)\|_2^2$$

Step 2: Steady-state Optimization

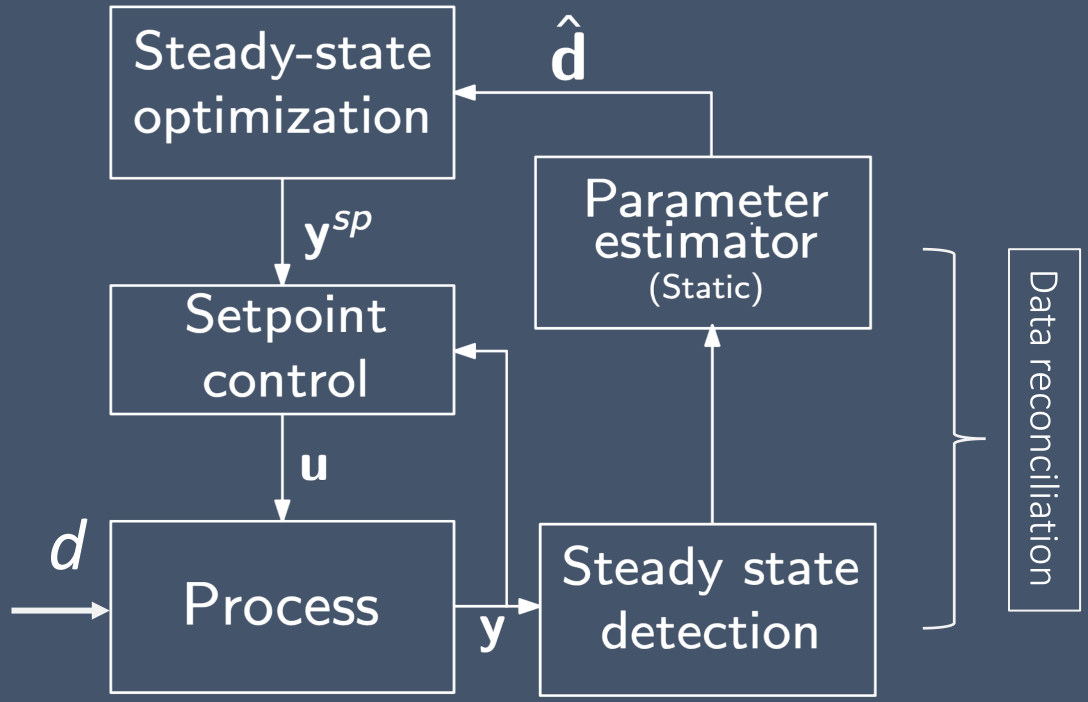
$$\mathbf{u}^*_{k+1} = \arg\min_{\mathbf{u}} J(\mathbf{y}, \mathbf{u})$$

$$\text{s.t. } \mathbf{y} = \mathbf{f}_{ss}(\mathbf{u}, \hat{\mathbf{d}}_k)$$

$$\mathbf{g}(\mathbf{y}, \mathbf{u}) \leq 0$$

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

Why is conventional RTO not commonly used?

Challenges (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 (steady-state wait time)
3. Not robust, including computational issues (computational cost)
4. Frequent grade changes make steady-state optimization less relevant
5. Dynamic limitations, including infeasibility due to (dynamic) constraint violation
6. Incorrect model structure

Krishnamoorthy, PhD Thesis (2019)
Krishnamoorthy, Foss, Skogestad, *Computers & Chemical Engineering* (2018)

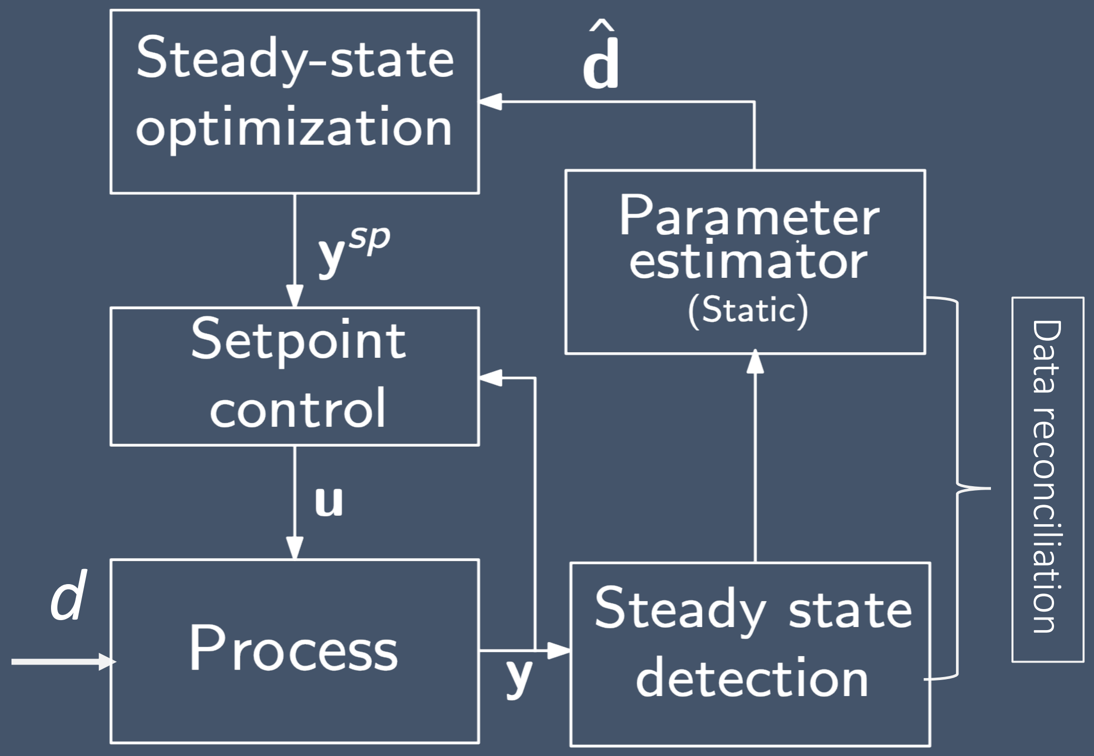
Challenge 1 - Costly offline model development and update

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

Possible Fix: Data-driven methods based on measuring Cost (J) (Extremum seeking control)
Recent interest – Machine learning and AI to develop surrogate models

Challenge 2 - Steady-state wait time

- Frequent disturbances (d)
- Long settling times
- Data reconciliation step is infrequent
- Wrong value of model parameter/disturbances
- Process operates sub-optimally for long periods of time



Fix: Use dynamic model in estimation step (Hybrid RTO)

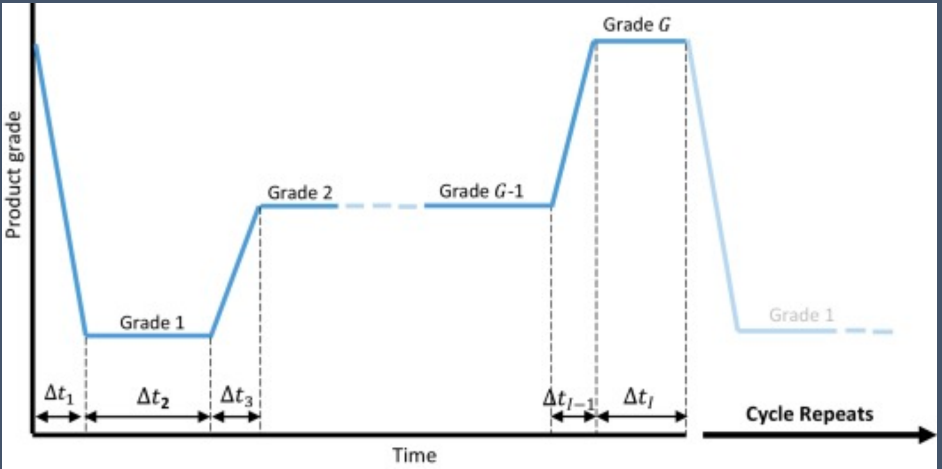
Challenge 3 - Computational issues

- Convergence issues and numerical failure
- CPU times
- scaling of variables
- Discontinouity and change in equations, e.g. new phase appears

Fix – Methods that do not need to solve numerical optimization problems online (Feedback RTO)

Challenge 4 - Frequent changes in product grades

- 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)

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)

Human factors

- Corporate culture
- Technology competence

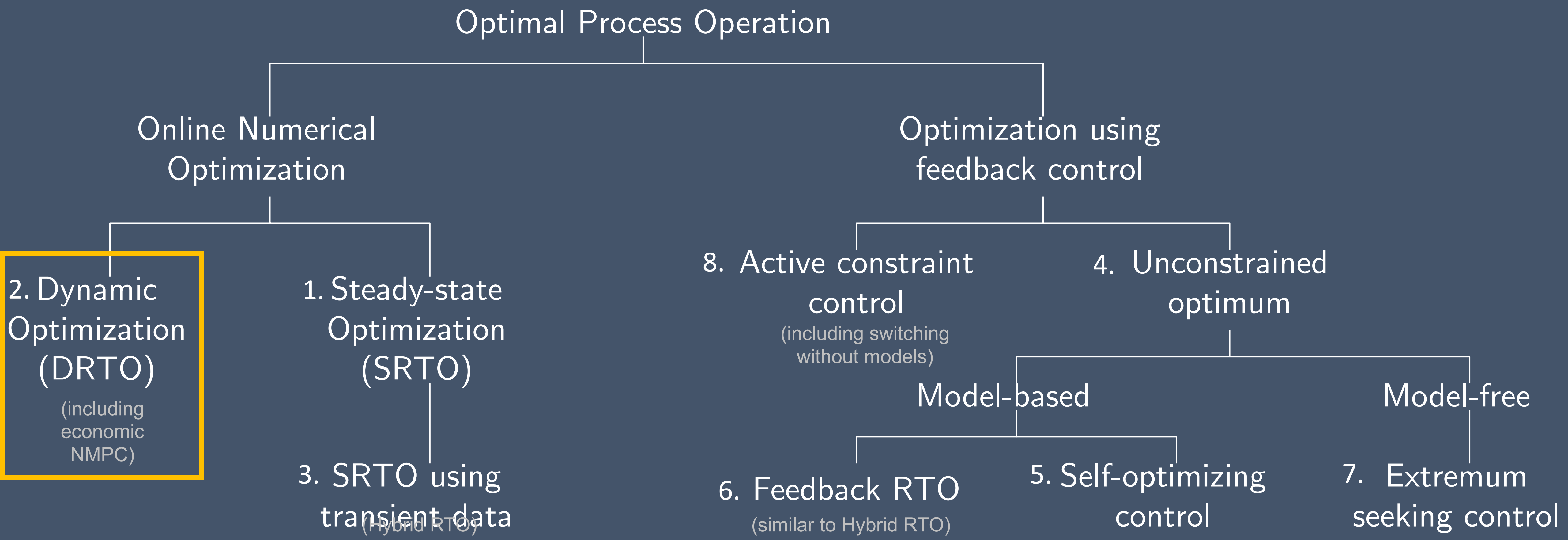
Fix – classical advanced control

Krishnamoorthy et al., (2019), Control Engineering Practice

2. Dynamic RTO ≈ Economic MPC

Academic approach

Workshop Roadmap



Dynamic optimization

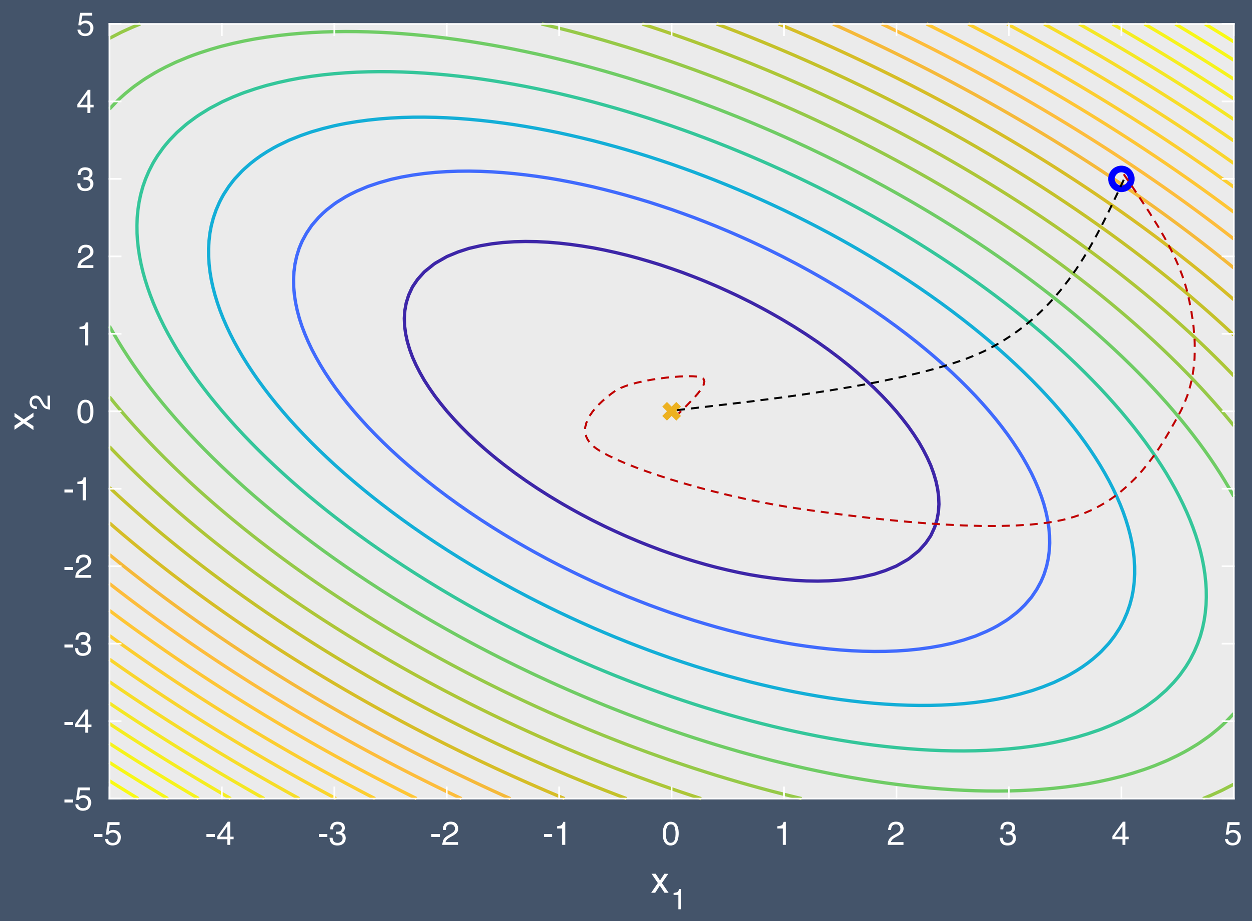
$$\min \sum_{t=1}^{T} J(\mathbf{x}(t), \mathbf{u}(t), \mathbf{d}(t))$$

s.t.

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

$$\mathbf{g}(\mathbf{x}(t), \mathbf{u}(t), \mathbf{d}(t)) \leq 0$$

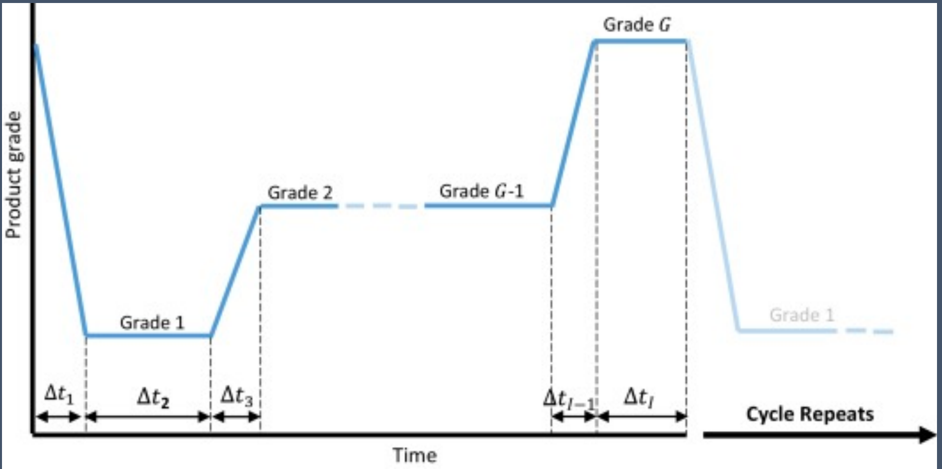
$$\mathbf{x}(0) = \widehat{\boldsymbol{x}}$$



DRTO / 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

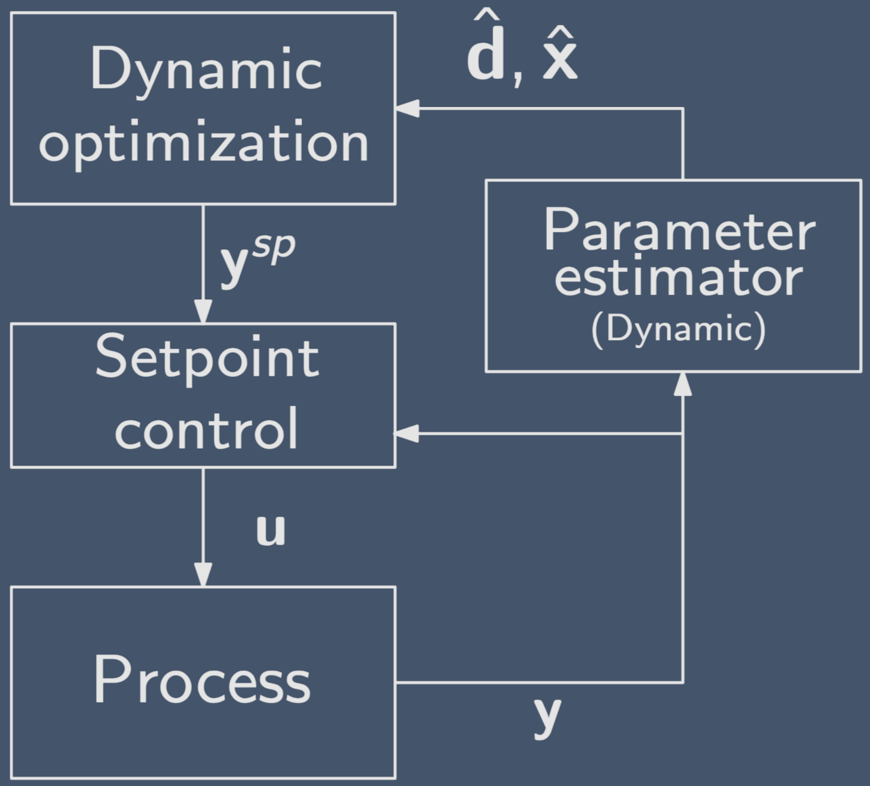


Source: Koller et al. (2017) *Comput& Chem Eng*, *106*, pp.147-159.

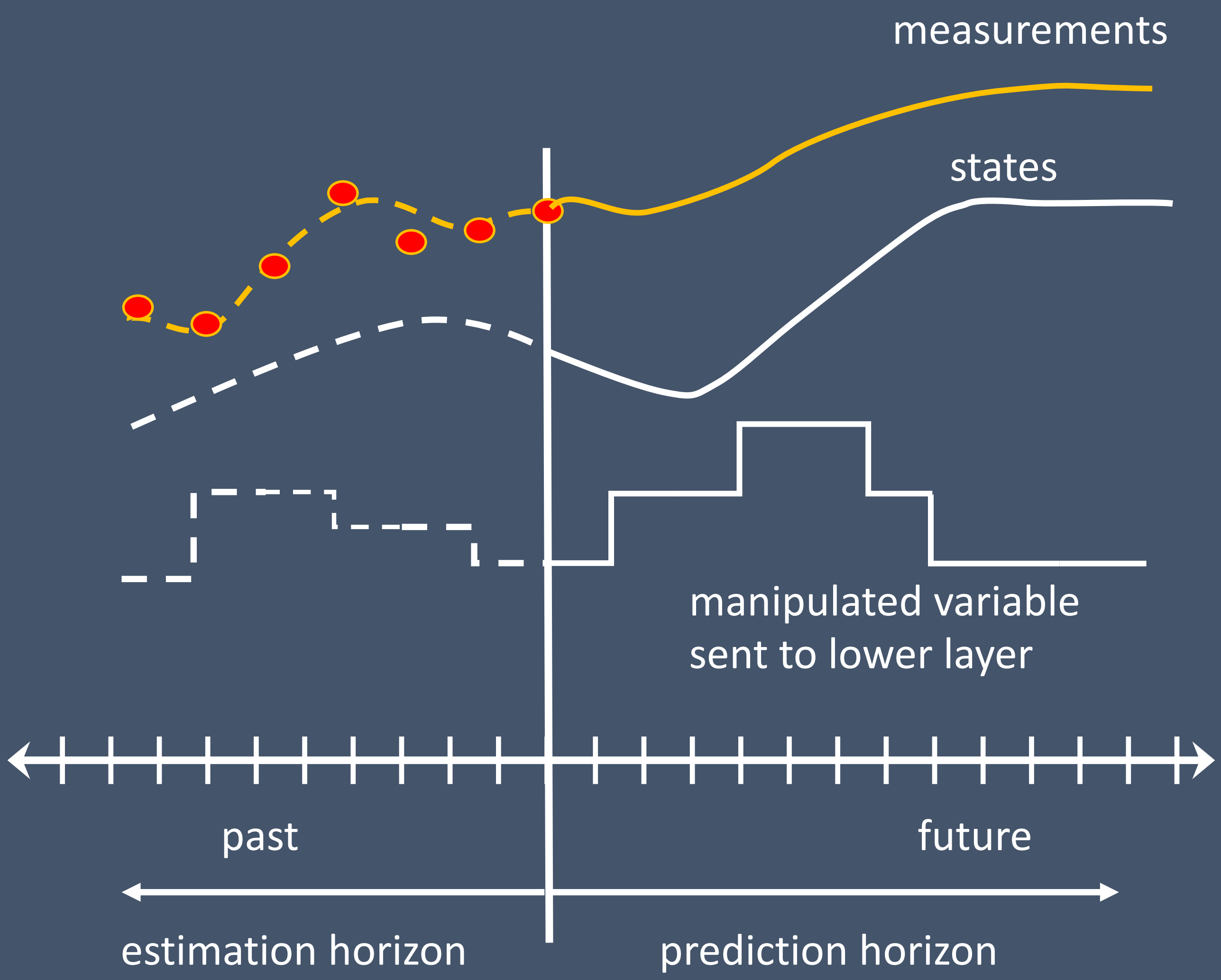
Directly address challenge 4 (frequent changes, non-negligible transient operation)

Dynamic RTO

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



Dynamic RTO/ Economic NMPC



Main idea

Step 1: Dynamic Estimation

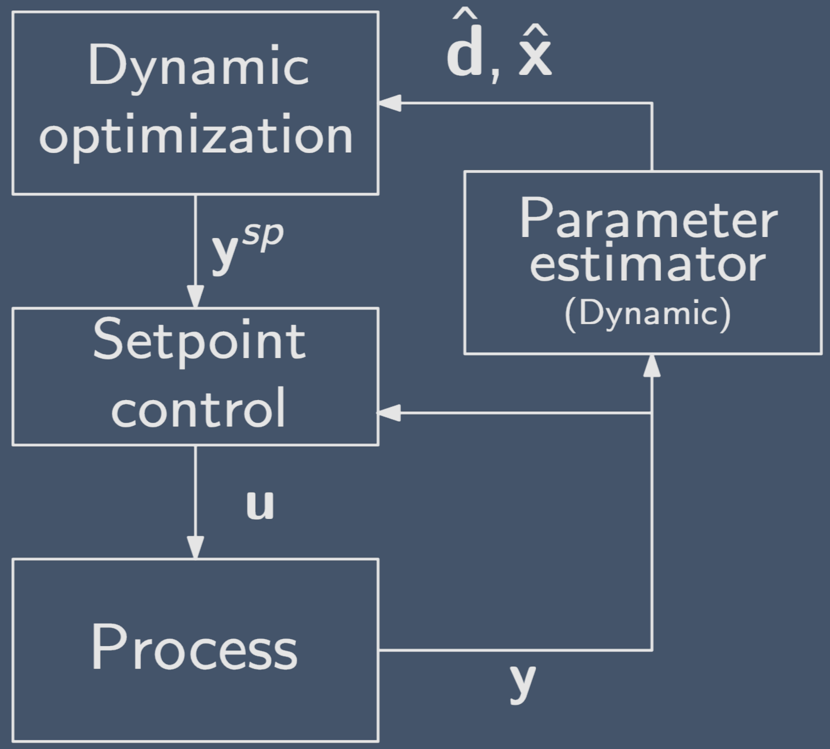
$$\hat{\mathbf{d}}_k = \arg\min_{\mathbf{d}_k} \|\mathbf{y}_{meas,k} - \mathbf{h}(\mathbf{x}_k, \mathbf{u}_k)\|$$

$$s.t.\ \ \mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \mathbf{d}_{k-1})$$

Step 2: Dynamic Optimization

$$\mathbf{u}_k^* = \arg\min_{\mathbf{u}_k} \sum_{k=0}^{N-1} J(\mathbf{y}_k, \mathbf{u}_k)$$

$$\begin{aligned} s.t.\ \ \mathbf{x}_{k+1} &= \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k, \hat{\mathbf{d}}_k) \\ \mathbf{y}_k &= \mathbf{h}(\mathbf{x}_k, \mathbf{u}_k) \\ \mathbf{g}(\mathbf{y}_k, \mathbf{u}_k) &\le 0 \\ \mathbf{x}_k &= \hat{\boldsymbol{x}}_t \qquad \forall k \in \{0, \ldots, N-1\} \end{aligned}$$



DRTO vs. economic NMPC vs. MPC

The main difference is the cost function

- DRTO – finds optimal y_{sp}

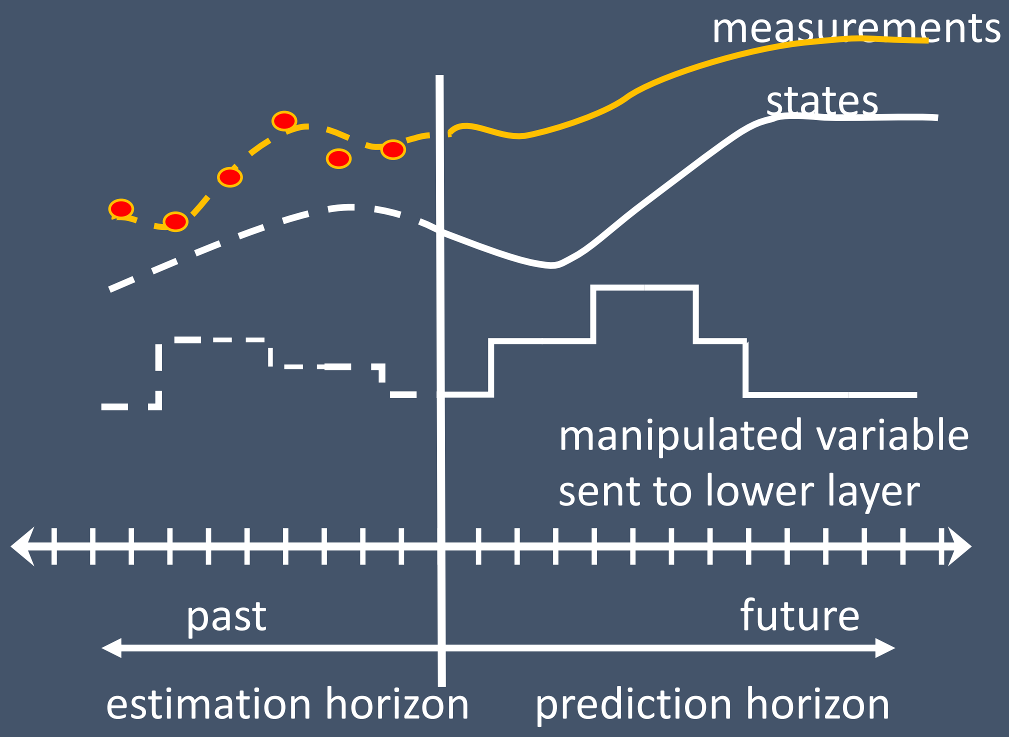
$$J_{econ} = p_F F + p_Q Q - p_p P$$

- ENMPC – finds optimal u

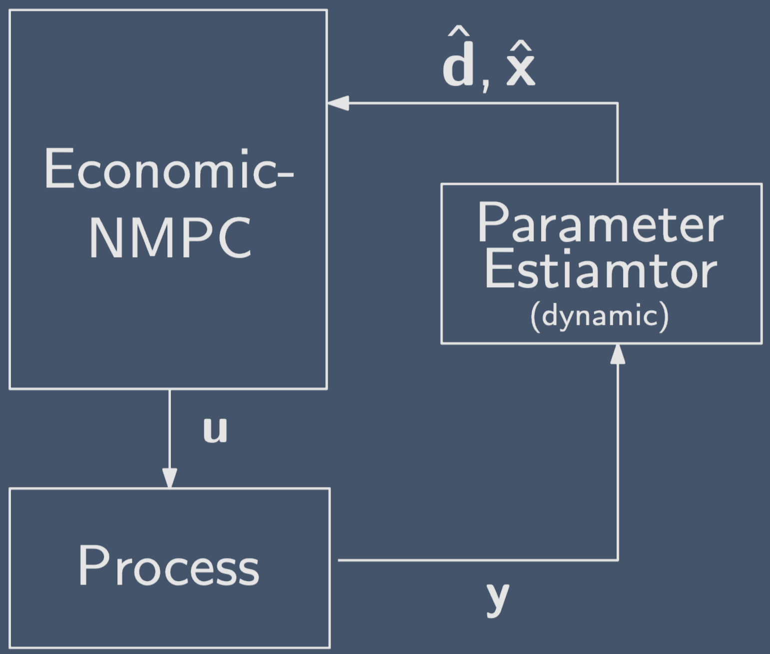
$$J = J_{econ} + w\,\Delta u^2$$

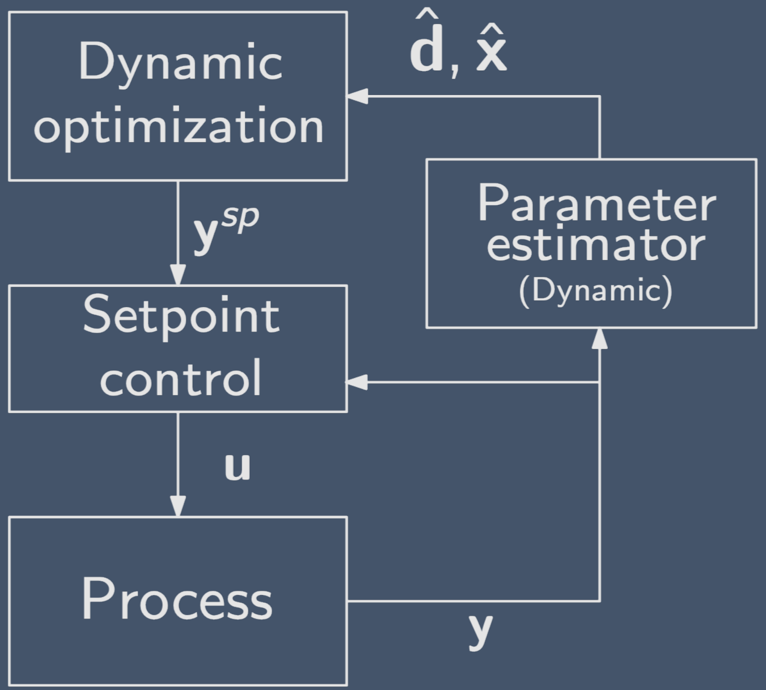
- MPC – finds optimal u

$$J_{sp} = \left(y - y_{sp}\right)^2 + w\,\Delta u^2$$



Economic MPC ~ Dynamic RTO



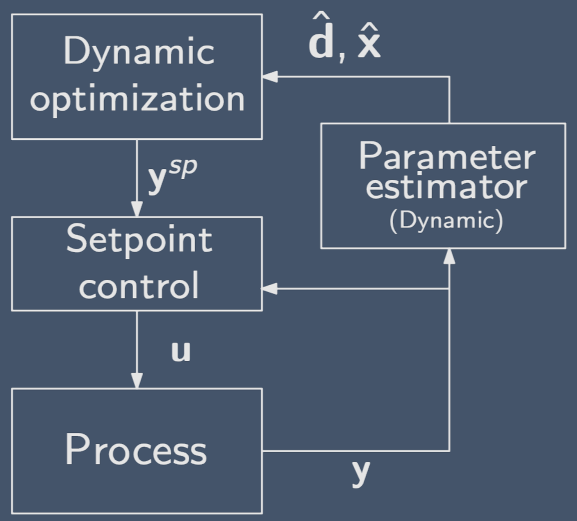


- Centralized "All-in-one" optimizer
- Higher sampling rates

- 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..

Dynamic RTO



Main advantages:

- Optimize dynamic path
- Avoid steady-state wait time

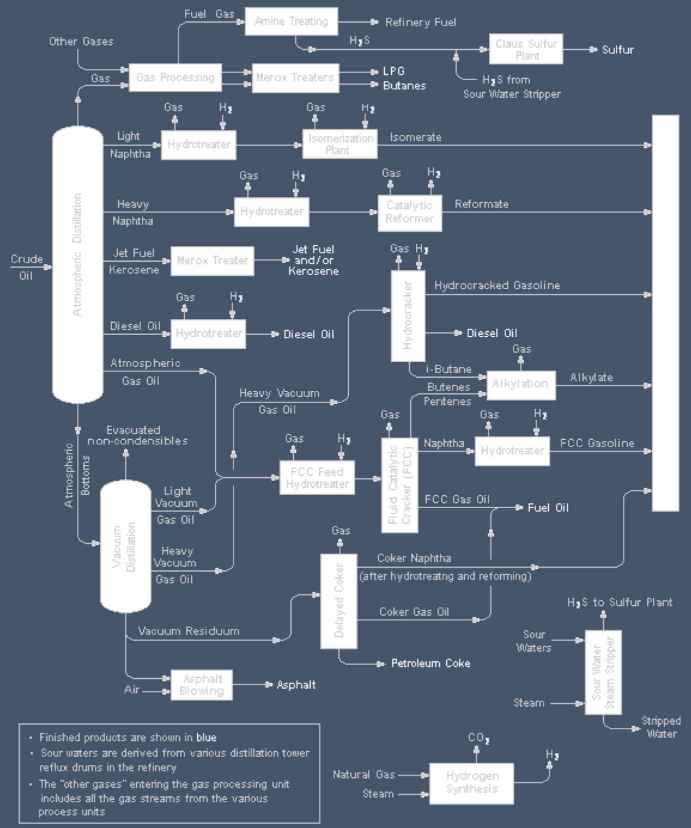
Main Challenge: Complexity, cost of modelling and implementation

- 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
- Uncertainty
- 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

Challenge 1 – worse than before

Computational issues (challenge 3)

- Computational cost for solving the large NLP
  - Computational robustness and convergence issues
- Discrete and nonsmooth decisions
  - Lead to mixed integer optimization problems
  - Cannot be solved in real-time for large systems

Challenge 3 – worse than before

Implementation issues

- Tuning, regularization weights in cost function
  - Typical cost in practice

$$J_{ENMPC} = J_{econ} + w_1 J_{sp} + w_2 J_{input}$$

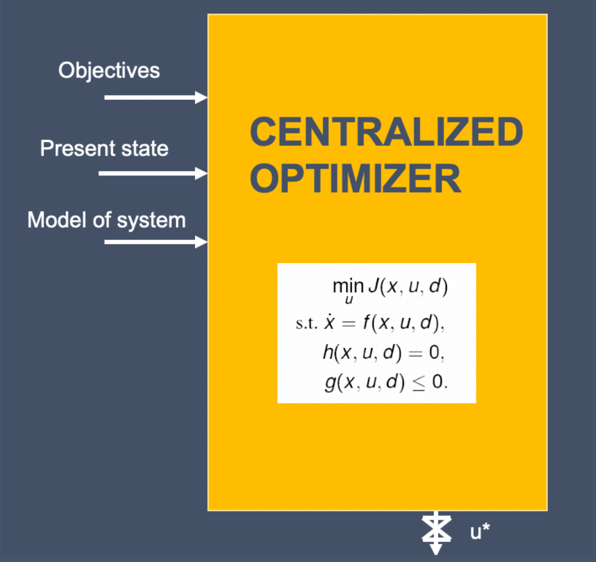
- Allowing for manual operations
- What to put into which layer?
- Measurement faults, reliable state and parameter estimation

Require many Ad-hoc problem-dependent solutions

DRTO and EMPC has many potential benefits

- Reduced amount of off-spec product
- Agile operational strategy:
  - Demand-side management
  - Load-balancing services
- For some processes, optimal operation is not at a steady state (Angeli 2011)

- Promise: Truly optimal operation
  - Process control excellent candidate for centralized control



Research focus in academia

- Stability of ENMPC
- Numerical issues
  - Computation speed
  - Decentralizing
- Uncertainty
  - Stochastic MPC
  - Robust MPC
  - Chance constraints
  - Dual and adaptive MPC

Proofs mainly of concern for academia

Handle computational complexity in real-time

Typically add complexity

But these are not addressing the main limitations of current industrial practice!

Main Challenge: Complexity, cost of modelling and implementation

Economic NMPC – skeleton in the closet

$$\min \sum_{t=1}^{T} J(\mathbf{x}(t), \mathbf{u}(t), \mathbf{d}(t))$$

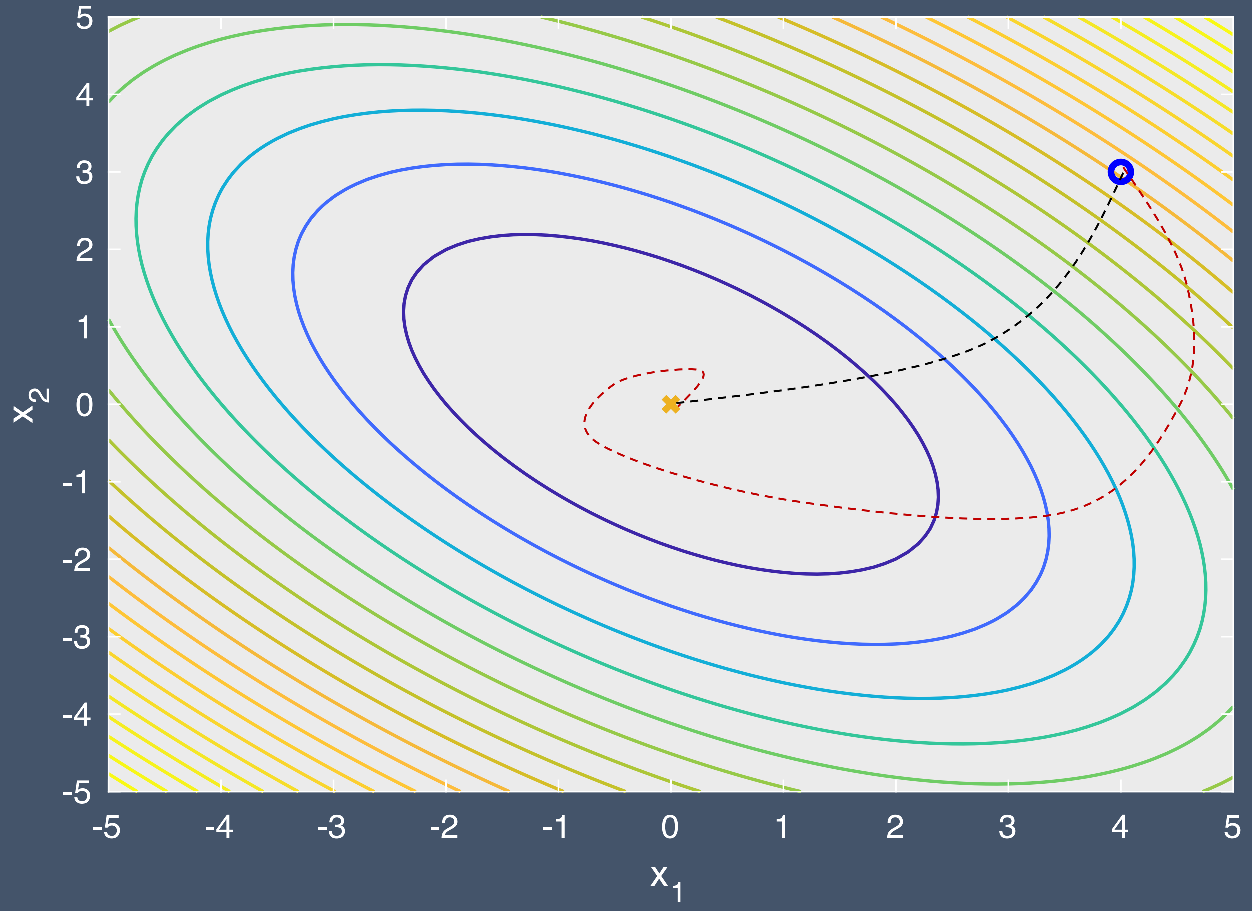
s.t.

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

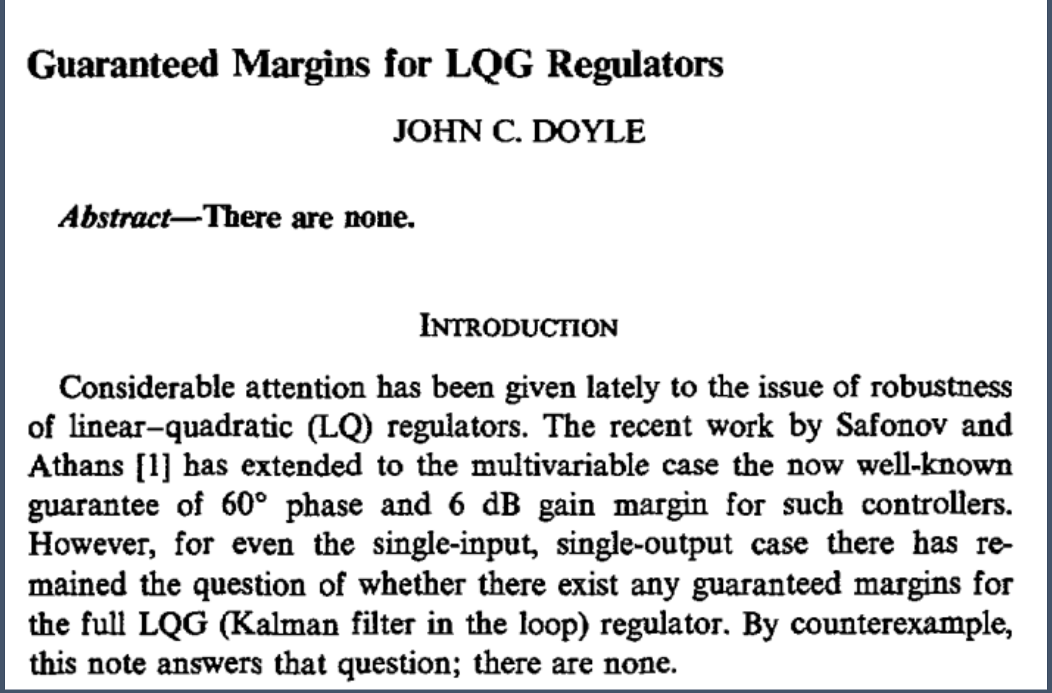
$$\mathbf{g}(\mathbf{x}(t), \mathbf{u}(t), \mathbf{d}(t)) \leq 0$$

$$\mathbf{x}(0) = \hat{\boldsymbol{x}}$$

- Need to know the current state of the system
- Often state measurements are not available
- Need to estimate states



NB! Estimator + Controller



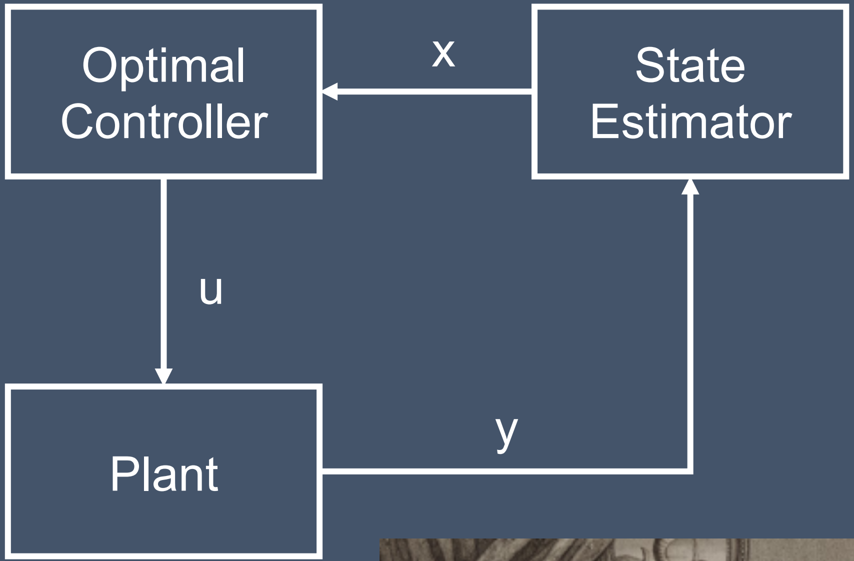
Guaranteed Margins for LQG Regulators

JOHN C. DOYLE

Abstract—There are none.

INTRODUCTION

Considerable attention has been given lately to the issue of robustness of linear–quadratic (LQ) regulators. The recent work by Safonov and Athans [1] has extended to the multivariable case the now well-known guarantee of 60° phase and 6 dB gain margin for such controllers. However, for even the single-input, single-output case there has remained the question of whether there exist any guaranteed margins for the full LQG (Kalman filter in the loop) regulator. By counterexample, this note answers that question; there are none.

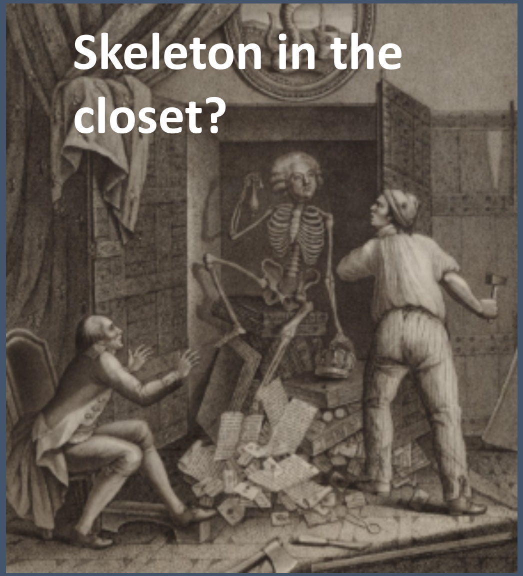


John Doyle (1985):

There are two ways a theorem can be wrong

(from an engineering point of view):

- ***Either it's simply wrong***
- ***Or the assumptions make no sense***



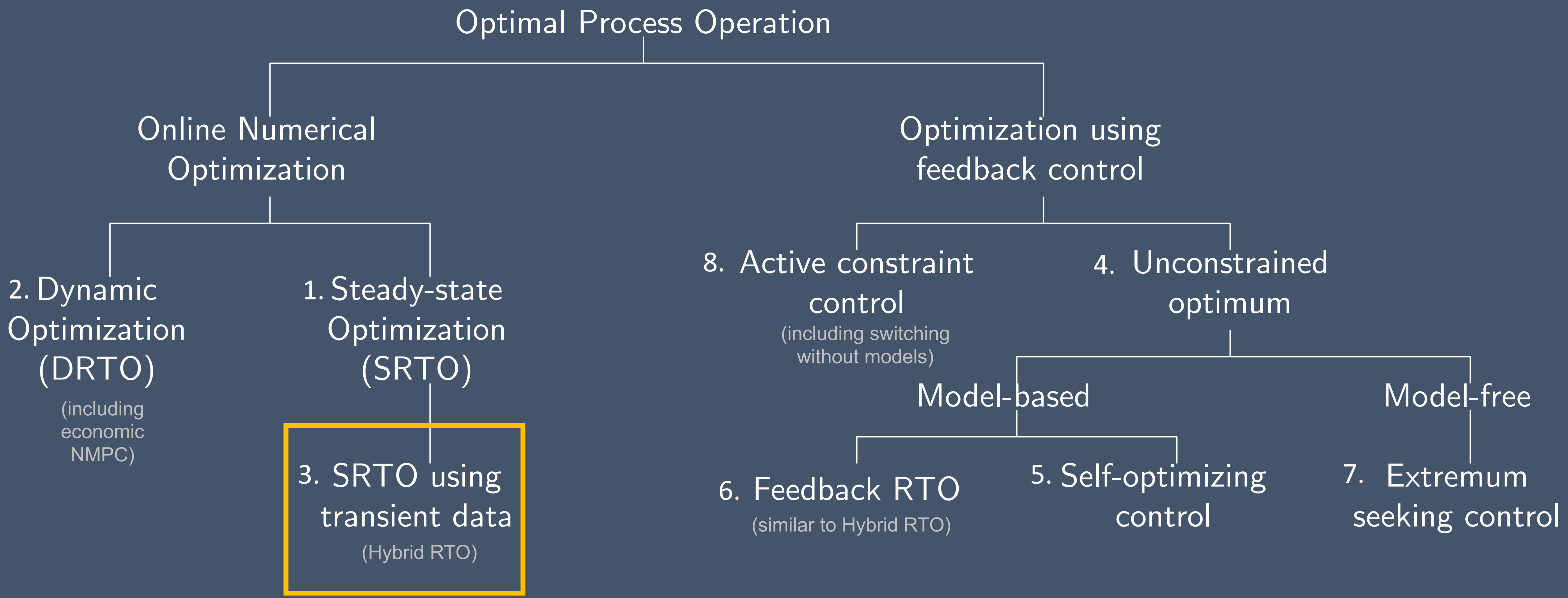
Doyle, J.C., 1978. Guaranteed margins for LQG regulators. *IEEE Transactions on automatic Control*, 23(4), pp.756-757.

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

No need for steady-state detection (SSD)

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

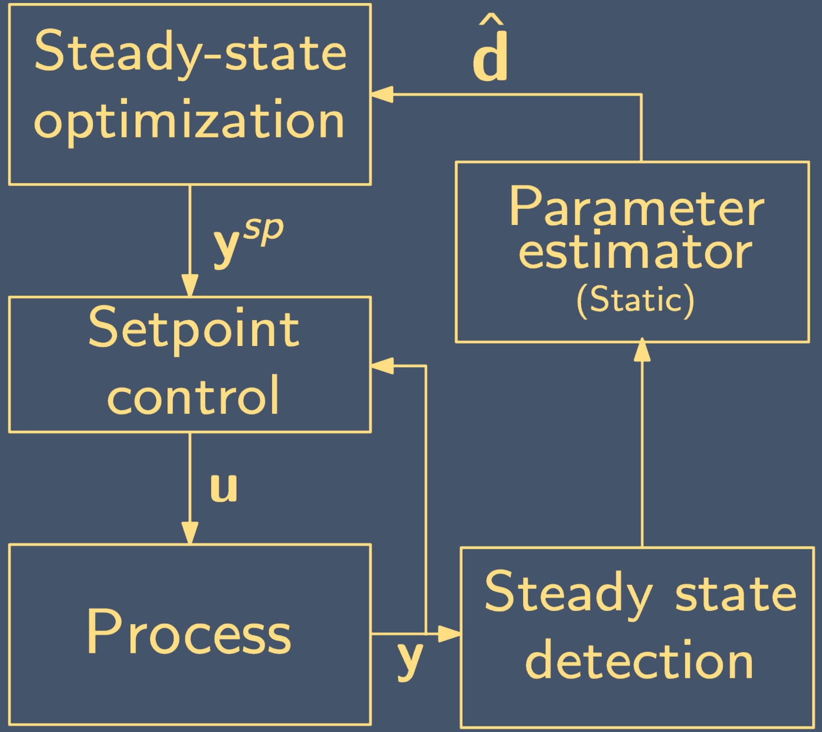
Workshop Roadmap



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: Steady-state Estimation

$$\hat{\mathbf{d}}_k = \arg\min_{\mathbf{d}_k} \|\mathbf{y}_{meas} - \mathbf{f}_{ss}(\mathbf{u}_k, \mathbf{d}_k)\|_2^2$$

Step 2: Steady-state Optimization

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

$$\text{s.t. } \mathbf{y} = \mathbf{f}_{ss}(\mathbf{u}, \hat{\mathbf{d}}_k)$$

$$\mathbf{g}(\mathbf{y}, \mathbf{u}) \leq 0$$

Steady-state detection

- Based on statistical tests,

  e.g:

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

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

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

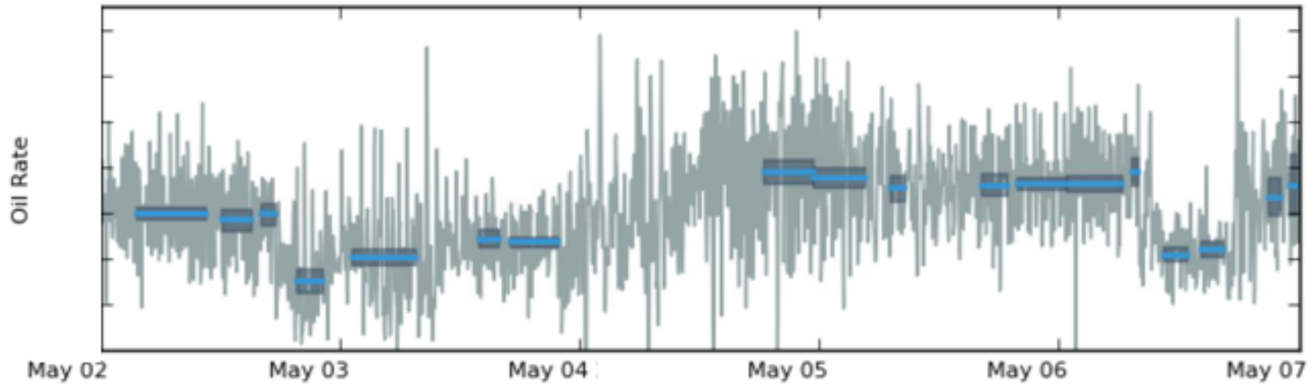
In practice - some heuristics

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

- Von Neumann, J. Distribution of the ratio of the mean square successive difference to the variance. Ann. Math. Stat. 1941, 12, 367–395
- Cao, S.; Rhinehart, R.R. An efficient method for on-line identification of steady state. J. Process Control 1995, 5, 363–374

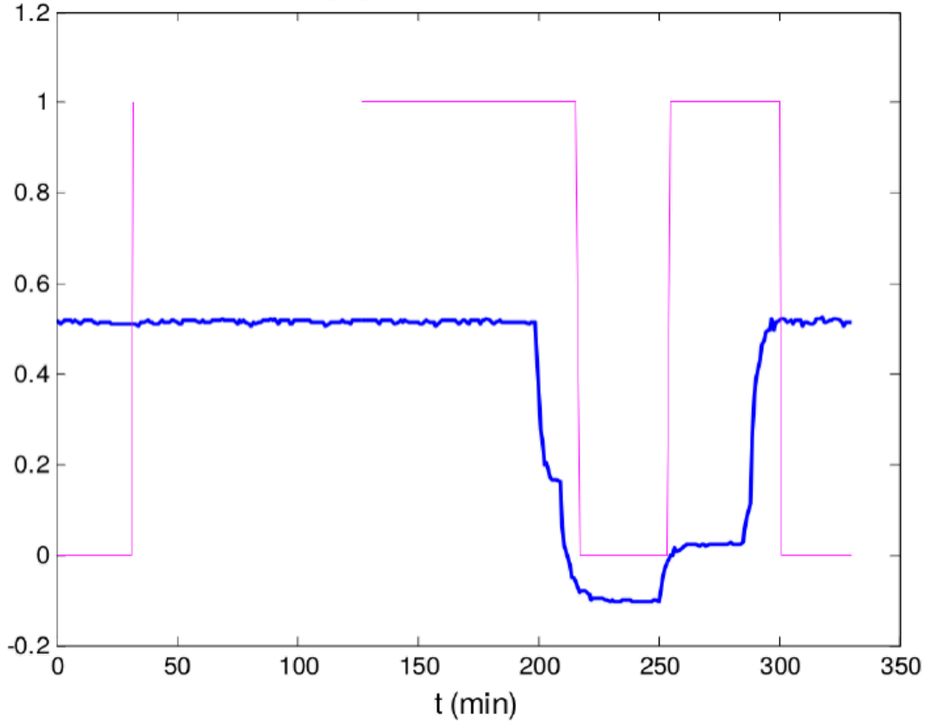
Steady-state wait time

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

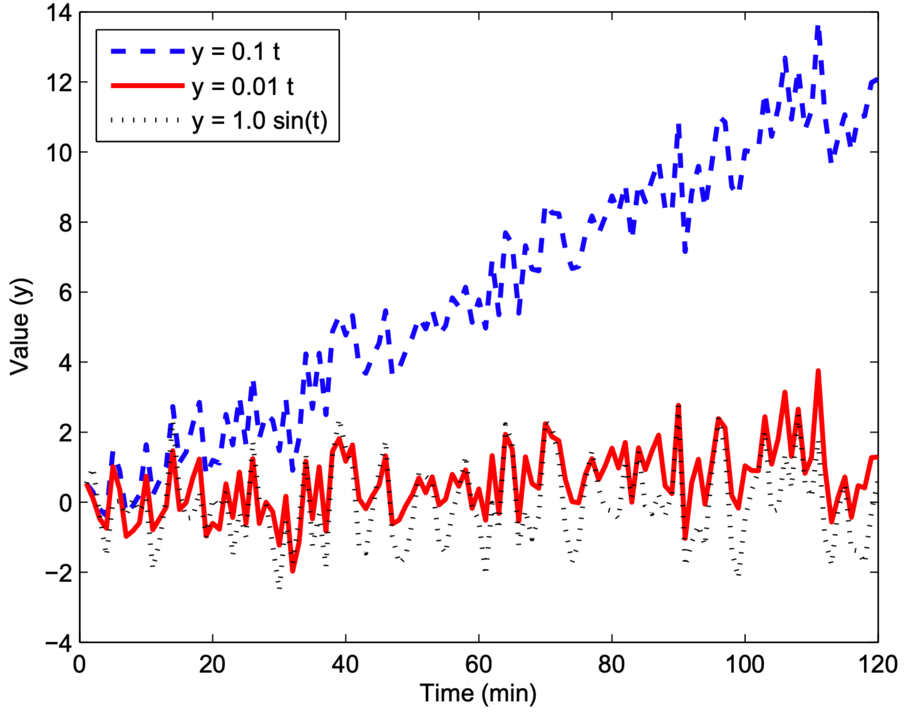


Large chunks of data discarded

Erroneous SSD



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



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.

How to address steady-state wait time?

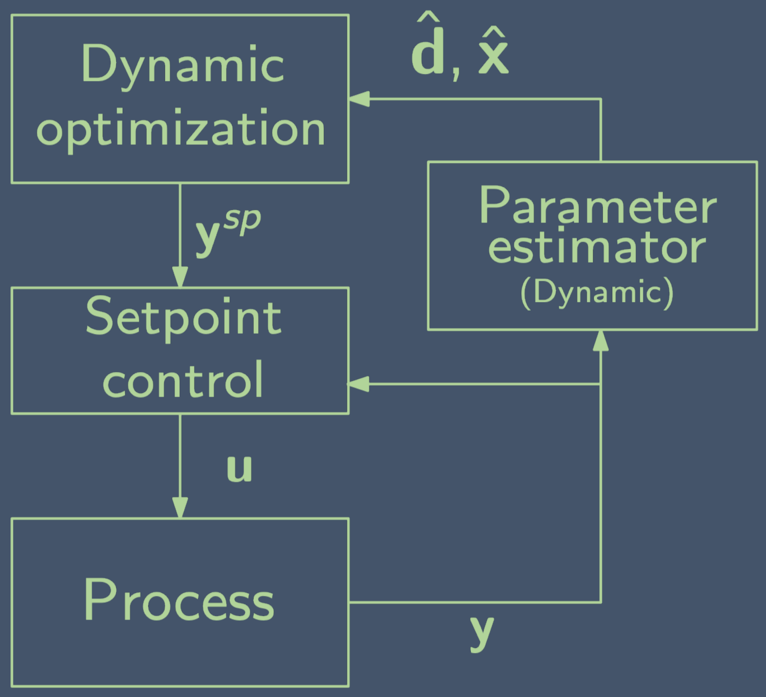
- **OBVIOUS: DYNAMIC RTO**

Step 1: Dynamic Estimation

$$\hat{\mathbf{d}}_k = \arg\min_{\mathbf{d}_k} \|\mathbf{y}_{meas,k} - \mathbf{h}(\mathbf{x}_k, \mathbf{u}_k)\|$$
$$s.t.\ \ \mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \mathbf{d}_{k-1})$$

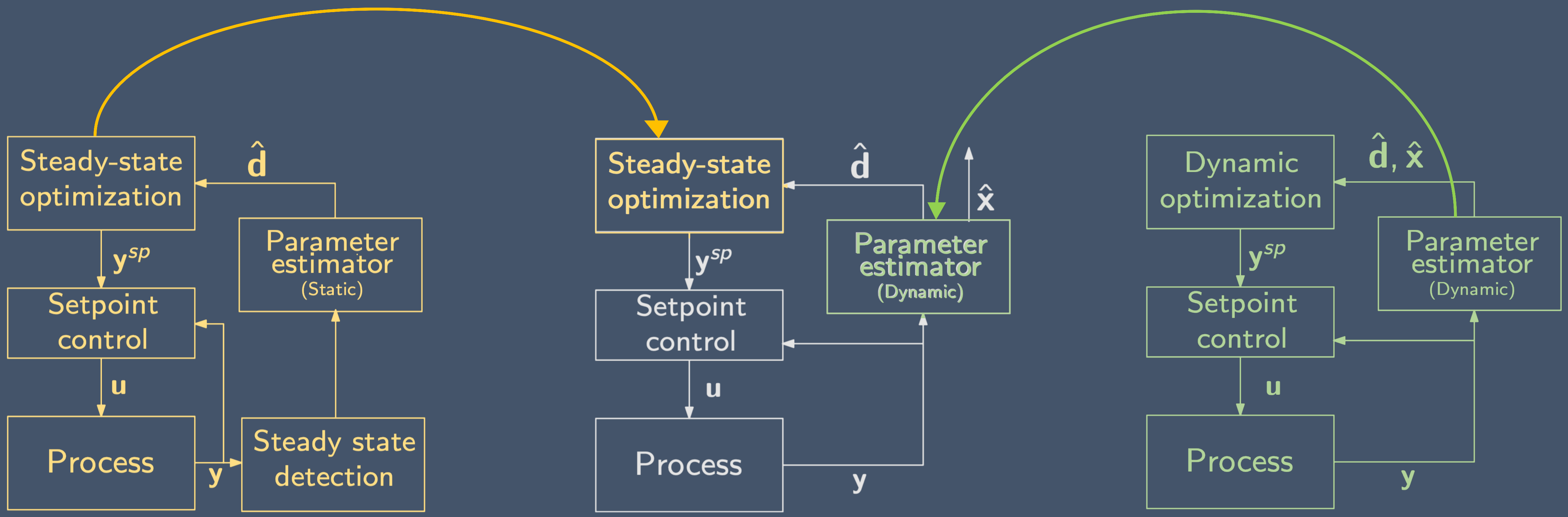
Step 2: Dynamic Optimization

$$\mathbf{u}_k^* = \arg\min_{\mathbf{u}_k} \sum_{k=0}^{N-1} J(\mathbf{y}_k, \mathbf{u}_k)$$
$$\begin{aligned} s.t.\ \ \mathbf{x}_{k+1} &= \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k, \hat{\mathbf{d}}_k) \\ \mathbf{y}_k &= \mathbf{h}(\mathbf{x}_k, \mathbf{u}_k) \\ \mathbf{g}(\mathbf{y}_k, \mathbf{u}_k) &\leq 0 \\ \mathbf{x}_k &= \hat{\boldsymbol{x}}_t \qquad \forall k \in \{0, \ldots, N-1\} \end{aligned}$$



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

$$\hat{\mathbf{d}}_k = \arg\min_{\mathbf{d}_k} \|\mathbf{y}_{meas,k} - \mathbf{h}(\mathbf{x}_k, \mathbf{u}_k)\|$$

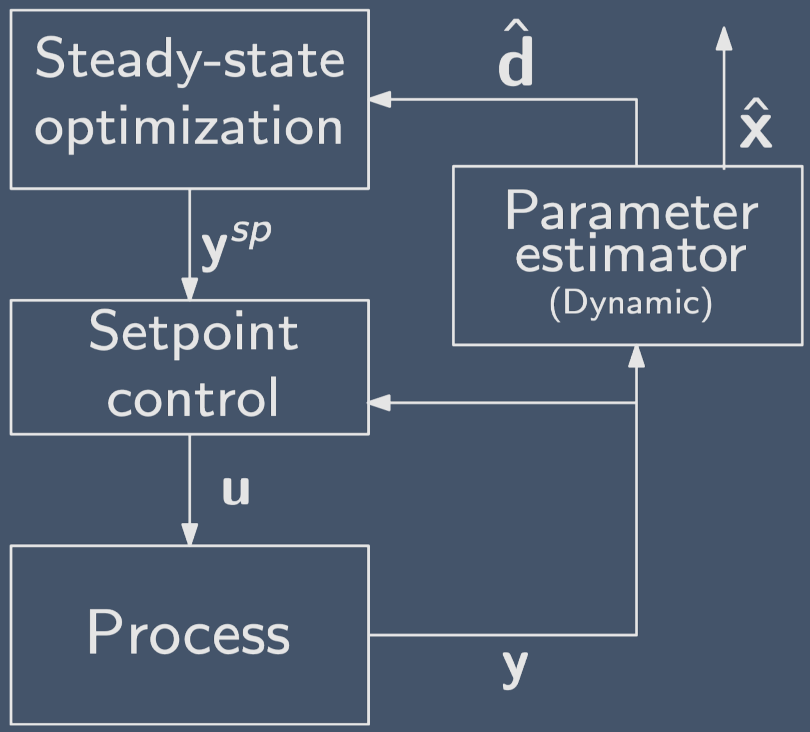
$$\text{s.t.} \quad \mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \mathbf{d}_{k-1})$$

Step 2: Steady-state Optimization

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

$$\text{s.t.} \ \mathbf{y} = \mathbf{f}_{ss}(\mathbf{u}, \hat{\mathbf{d}}_k)$$

$$\mathbf{g}(\mathbf{y}, \mathbf{u}) \leq 0$$



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***

Modelling effort

- Mass balance
- Energy balance
- …

Often leads to Differential Algebraic Equations (in chemical processes, typically Index-1 DAE)

$$\begin{aligned} \frac{dx}{dt} &= f(x, z, u, d) \\ z &= g(x, z, u, d) \end{aligned}$$

Corresponding steady-state model:

$$\begin{aligned} f(x, z, u, d) &= 0 \\ g(x, z, u, d) &= z \end{aligned}$$

Augmented Kalman filter for state and parameter estimation

- Uncertain paramteres **d** are added with a small artificial Gaussian noise term

$$\mathbf{d}_{k+1} = \mathbf{d}_k + \mathbf{w}_{\mathbf{d},k}$$

- The Augmented system is then given by :

$$\mathbf{x}'_{k+1} = \begin{bmatrix} \mathbf{x}_{k+1} \\ \mathbf{d}_{k+1} \end{bmatrix} = \begin{bmatrix} f(\mathbf{x}_k, \mathbf{u}_k, \mathbf{d}_k) + \mathbf{w}_k \\ \mathbf{d}_k + \mathbf{w}_{\mathbf{d},k} \end{bmatrix}$$

$$\mathbf{y}_{meas,k} = \begin{bmatrix} \mathbf{h}(\mathbf{x}_k, \mathbf{u}_k) & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}_k \\ \mathbf{d}_k \end{bmatrix} + \mathbf{v}_k$$

- Use e.g. EKF to estimate the augmented state and parameters

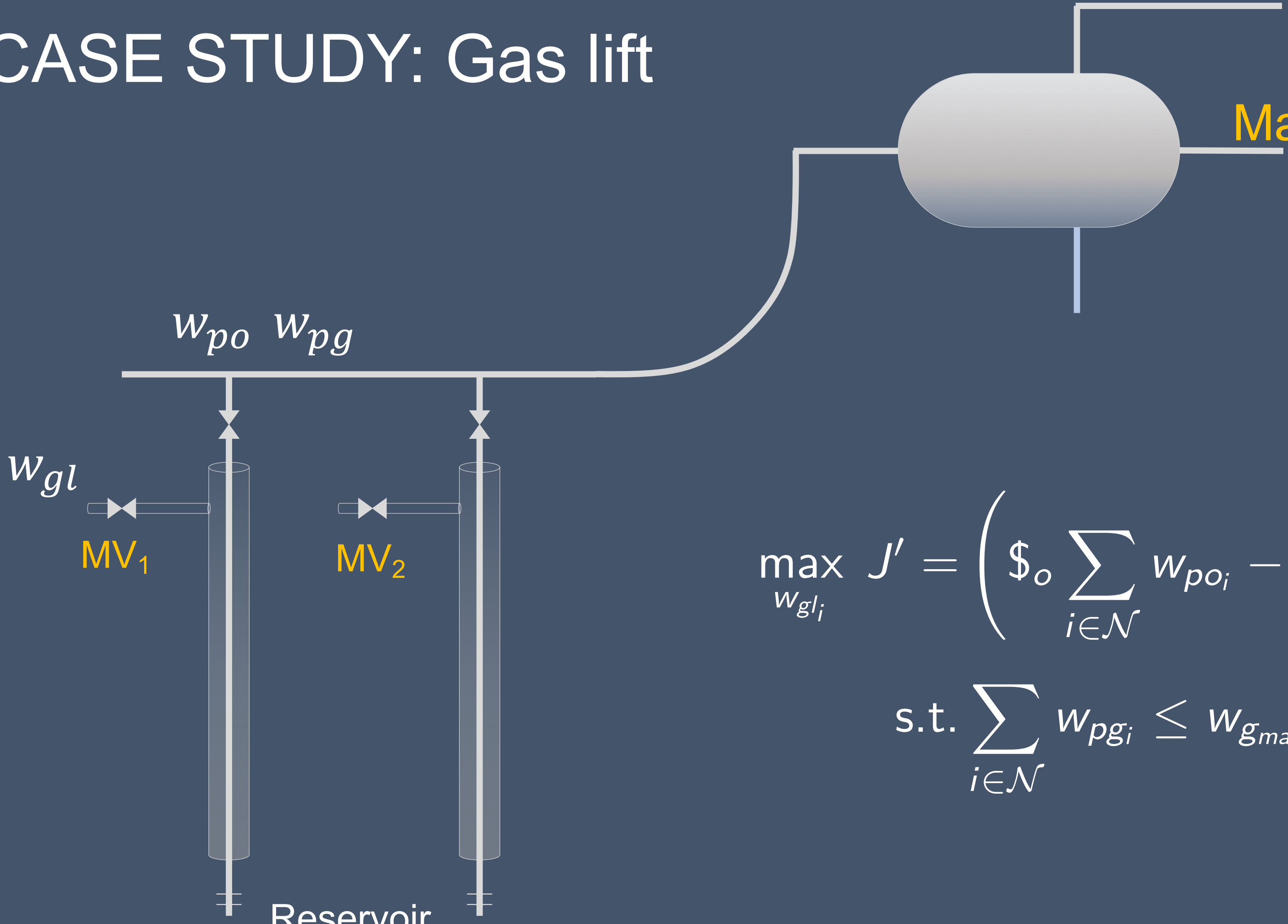
The maximum dimension of the disturbance **d** that one can choose, such that the augmented system remains detectable is equal to the number of measurements (i.e. $n_d \leq n_y$)

Simon, D., 2006. Optimal state estimation: Kalman, H infinity, and nonlinear approaches. John Wiley & Sons.

CASE STUDY: Gas lift

Constraint: Max Gas capacity

Main objective: Max oil prod.



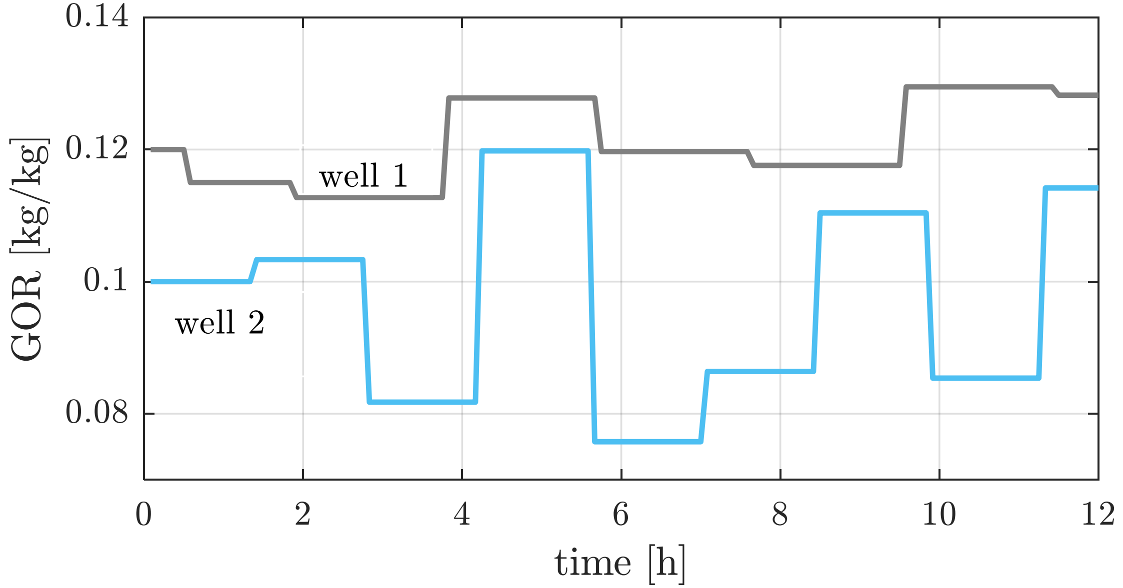
$$\max_{w_{gl_i}} \; J' = \left(\$_o \sum_{i \in \mathcal{N}} w_{po_i} - \$_{gl} \sum_{i \in \mathcal{N}} w_{gl_i} \right) - \$_{fl} \| w_{fl} \|$$

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

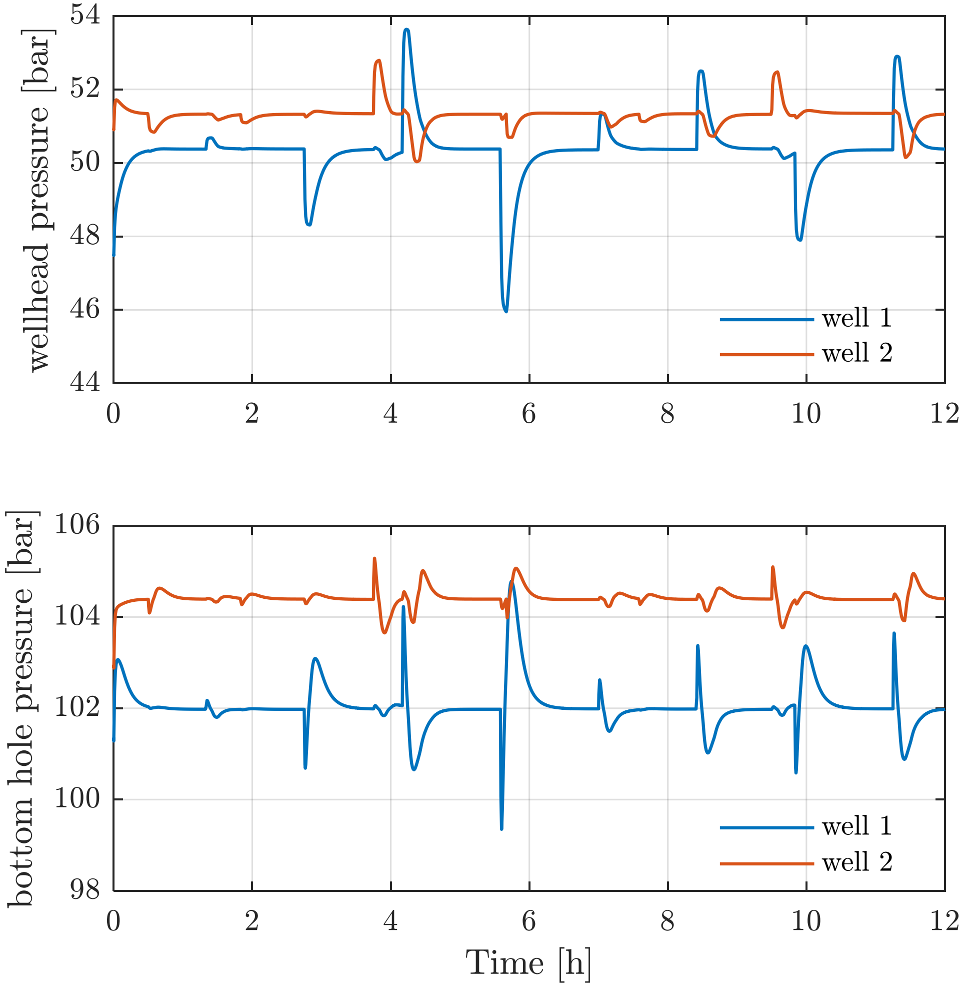
Main disturbance (d): GOR variation

GOR = Gas/Oil ratio in feed (reservoir)

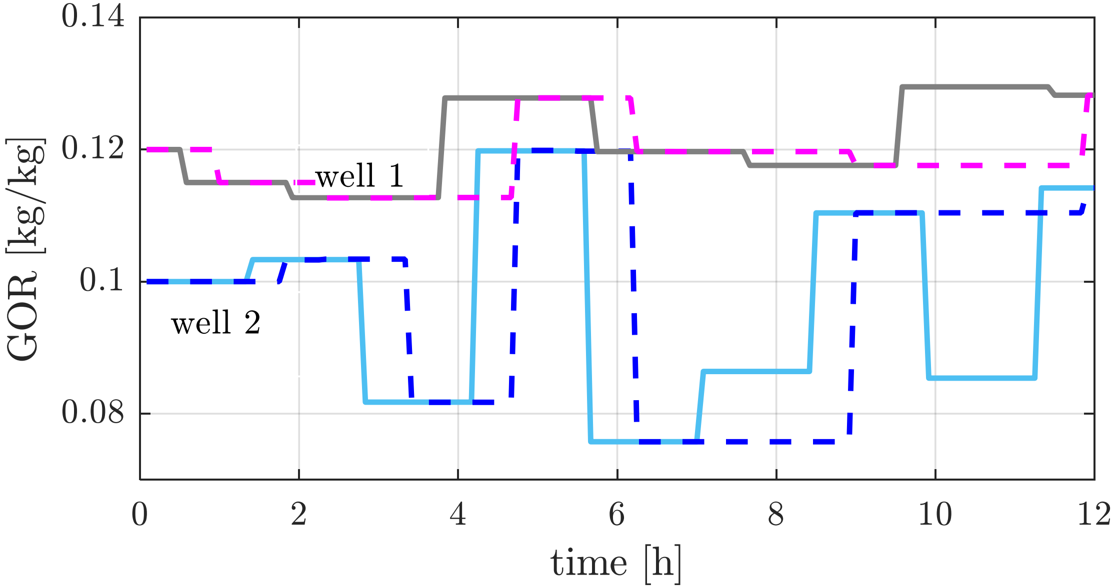
Disturbance: GOR variation



Typical measured data (pressures and flowrates)

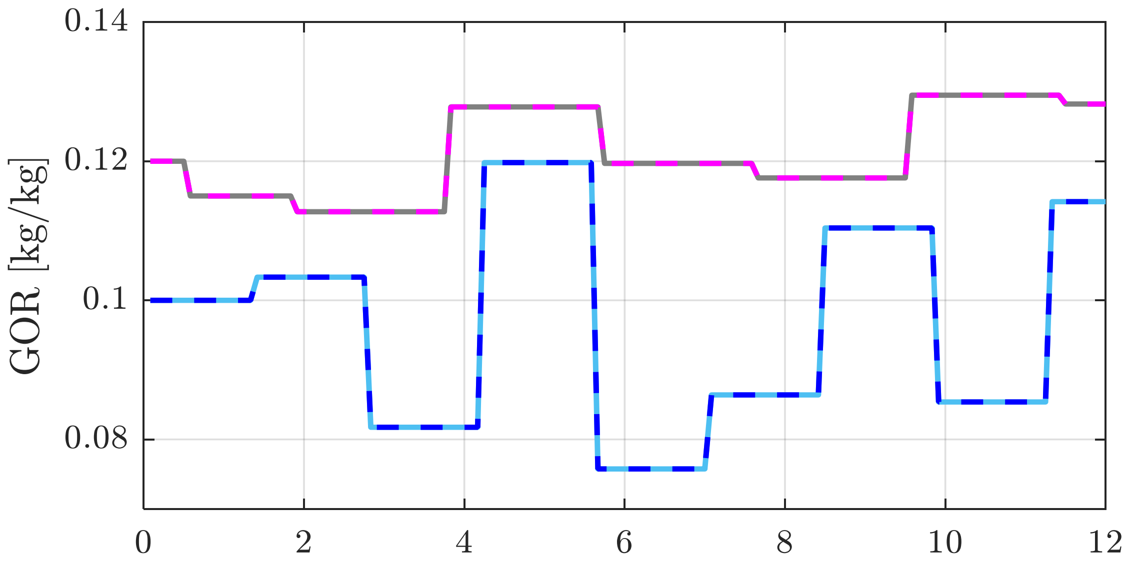


GOR estimation - using “data reconciliation” (traditional static RTO)

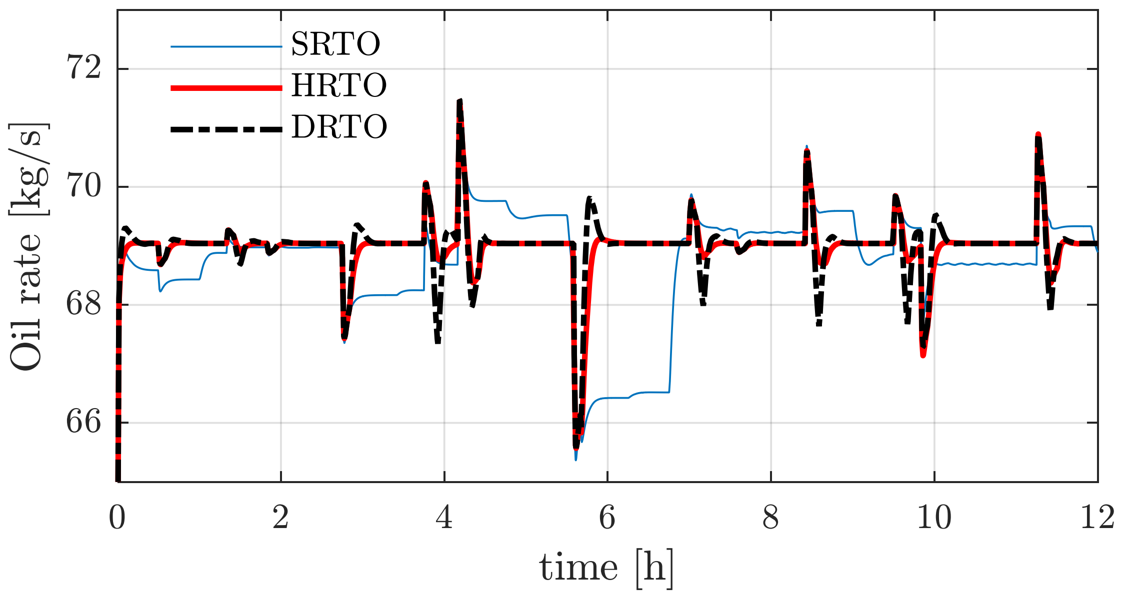


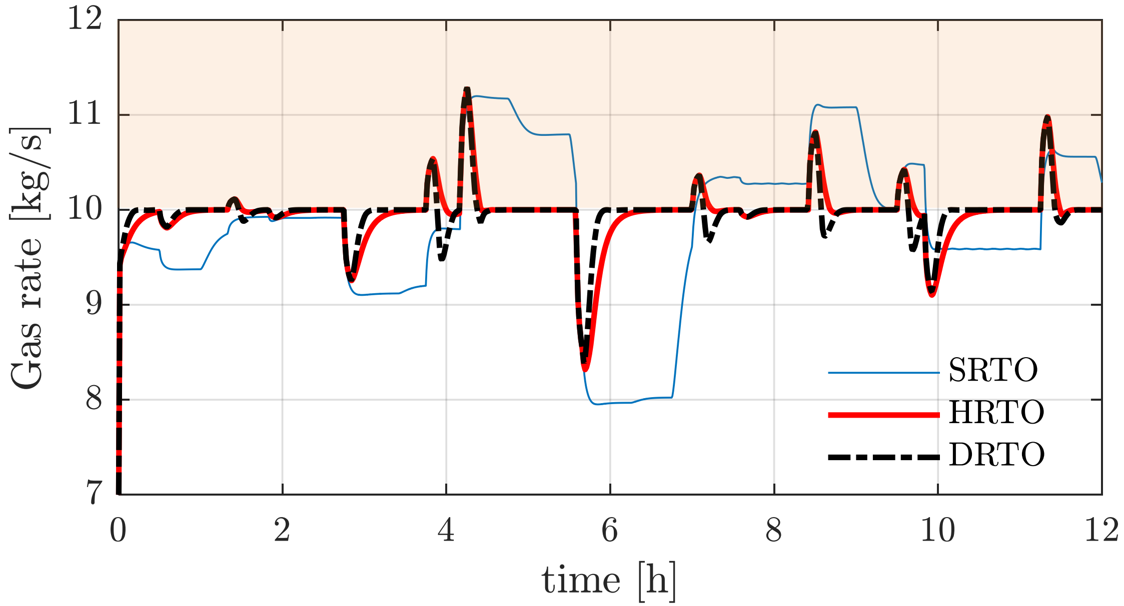
Problem: Steady-state wait time for data reconciliation

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



Oil and gas rates

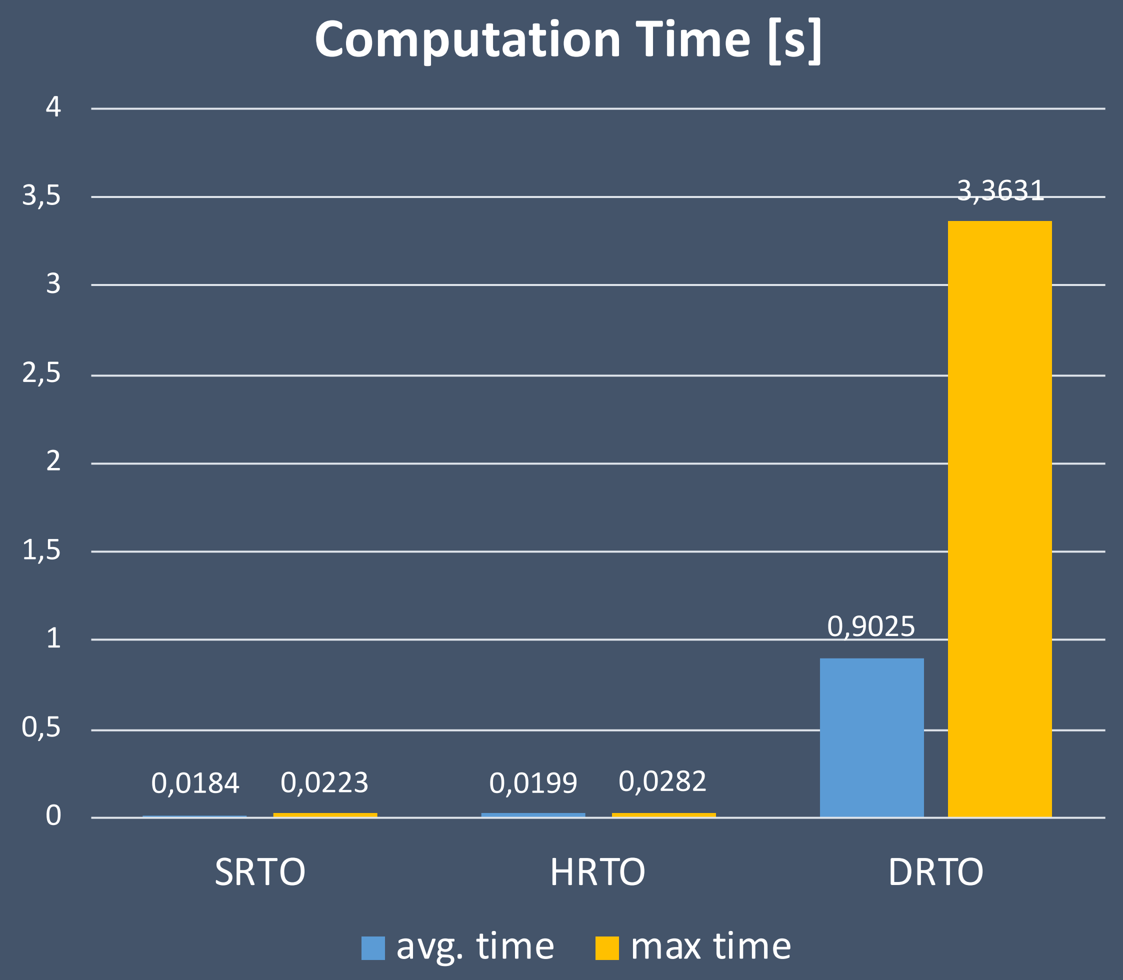




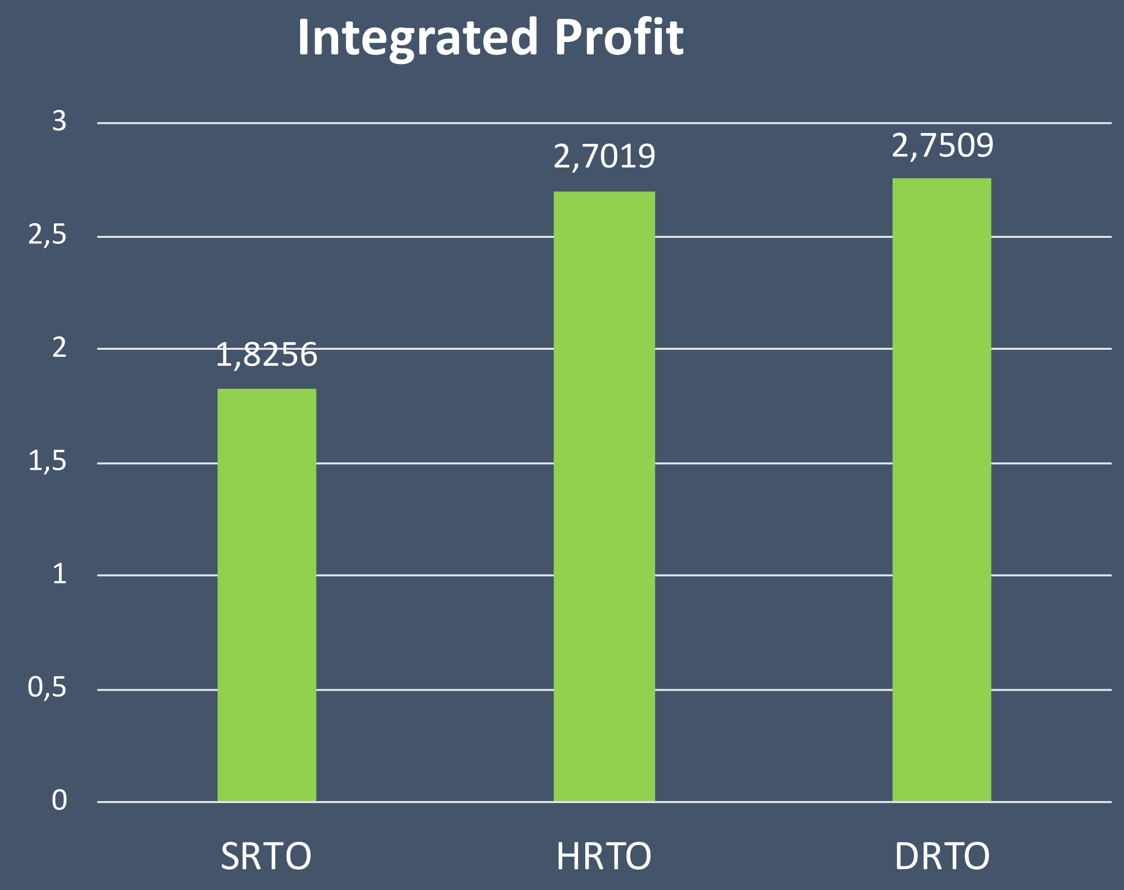
SRTO = traditional static RTO
HRTO = hybrid RTO
DRTO = dynamic RTO

For MATLAB code, contact: dineshk@ntnu.no

Results



Computation Time [s]
4
3,5
3
2,5
2
1,5
1
0,5
0
0,0184
0,0223
0,0199
0,0282
0,9025
3,3631
SRTO
HRTO
DRTO
avg. time
max time



Integrated Profit
3
2,5
2
1,5
1
0,5
0
1,8256
2,7019
2,7509
SRTO
HRTO
DRTO

Advantage of steady-state optimization (SRTO & HRTO)

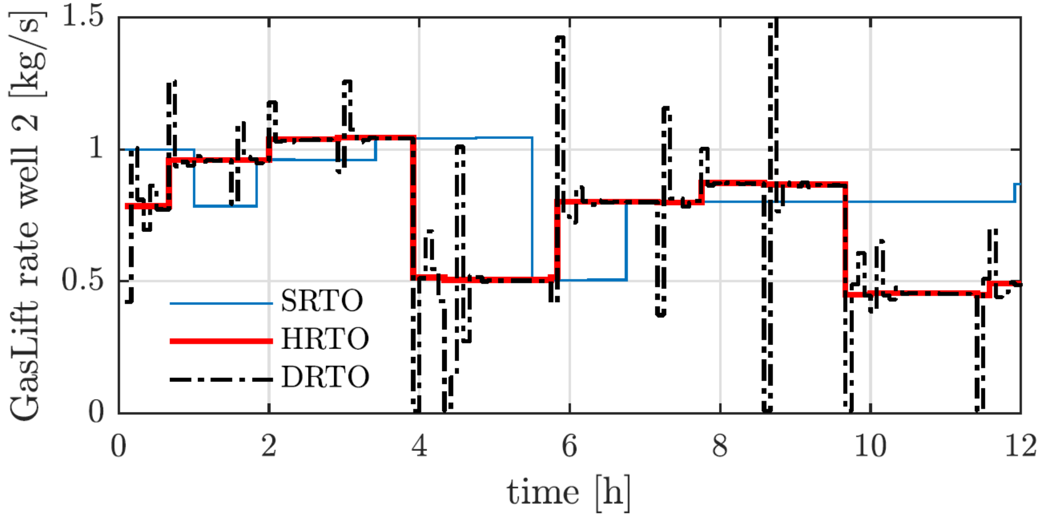
- Computation time & numerical robustness
- Avoids causality issue / index problems
- Allows optimization on decision variables other than the MVs
  - Simplifies the optimization
  - Slower time scale (choose slow varying variables as decision variables)

Why is traditional static RTO not commonly used?

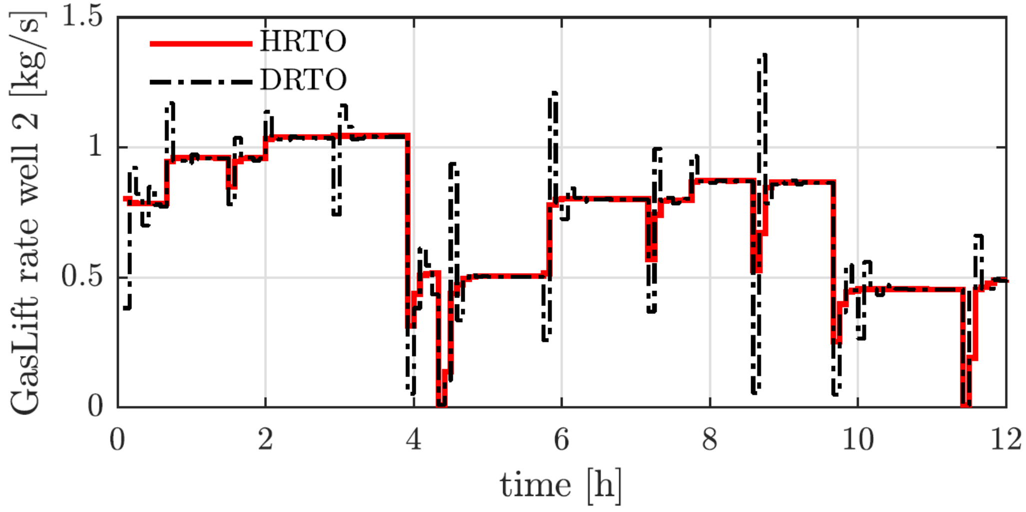
1. Cost of developing and updating the model structure (costly offline model update)
2. Wrong value of model parameters and disturbances (slow online model update)
3. Not robust, including computational issues
4. Frequent grade changes make steady-state optimization less relevant
5. Dynamic limitations,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

Dynamic limitations – not a big issue

MV2: Setpoint provided to tracking controller

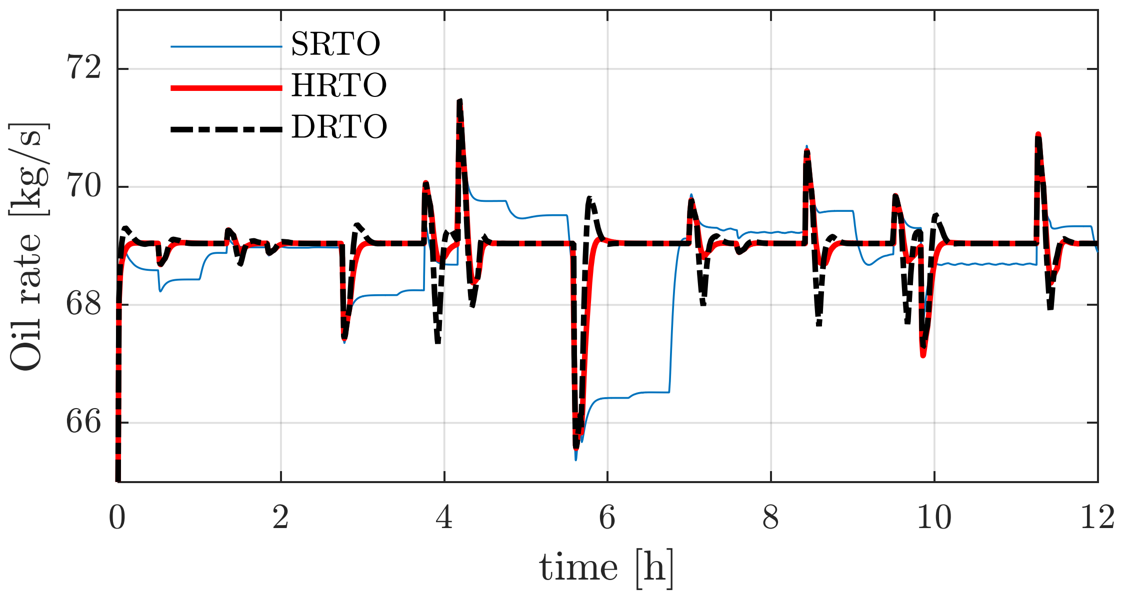


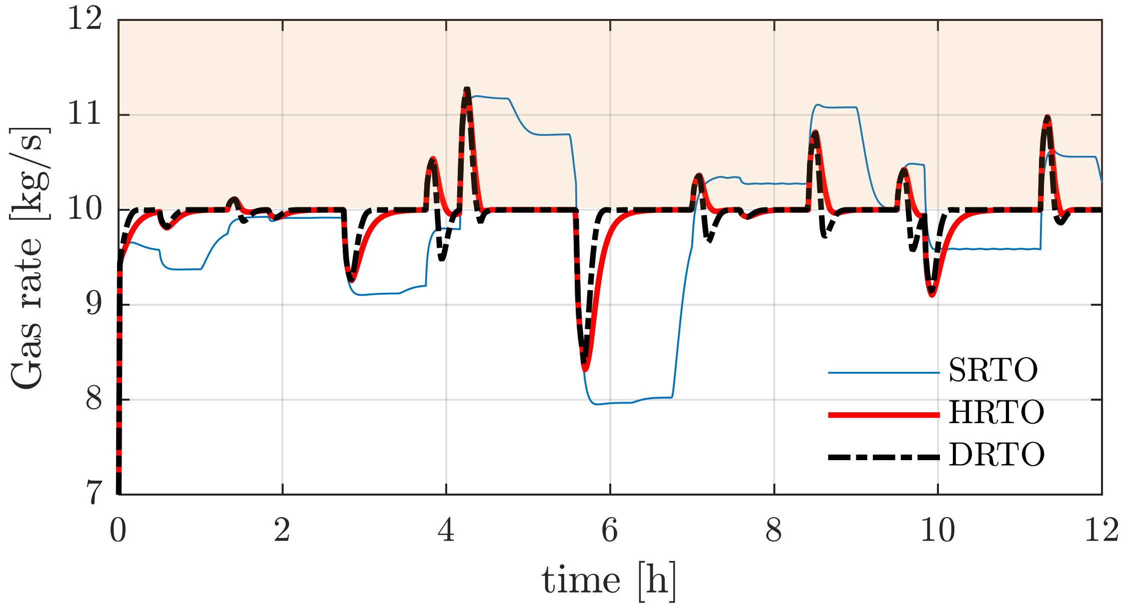
Actual MV move by setpoint tracking NMPC



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

Oil and gas rates





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

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

4. Optimal operation using Feedback Control

Translate economic objectives into control objectives

Course Roadmap

- Online process optimization
  - Numerical Optimization
    - Dynamic Optimization (DRTO)
    - Steady-state Optimization (SRTO)
      - SRTO using transient data (Hybrid RTO)
  - **Optimization using feedback control** (including switching between active constraint regions using classical advanced control elements)
    - **Active constraint control** (no models needed)
    - Unconstrained optimum
      - Model-based
        - Feedback RTO (similar to Hybrid RTO)
        - Self-optimizing control
      - Model-free
        - Extremum seeking control

Feedback optimizing control

- Translate economic objectives → control objectives
- Dates back to the 1980’s (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.”*

- Gained popularity since 2000 (Skogestad, 2000)

Benefit : Avoid solving numerical optimization problems online

- M. Morari, Y. Arkun, and G. Stephanopoulos. Studies in the synthesis of control structures for chemical processes: Part I: Formulation of the problem. process decomposition and the classification of the control tasks. analysis of the optimizing control structures. *AIChE Journal*, 26(2):220–232, 1980.
- S. Skogestad ``Plantwide control: the search for the self-optimizing control structure", J. Proc. Control, 10, 487-507 (2000).

Do we always need to solve an optimization problem?

- We often know or can guess the **optimal solution**
    - Example: Drive from A → B ***in shortest time***

- CV = speed → speed limit 50km/h
- MV = gas pedal

Can translate economic objectives into control objectives

Do we always need to solve an optimization problem?

- We often know or can guess the **optimal solution**
  - Example: Drive from A → B ***in fuel economical way***

    - CV = ??
    - MV = gas pedal

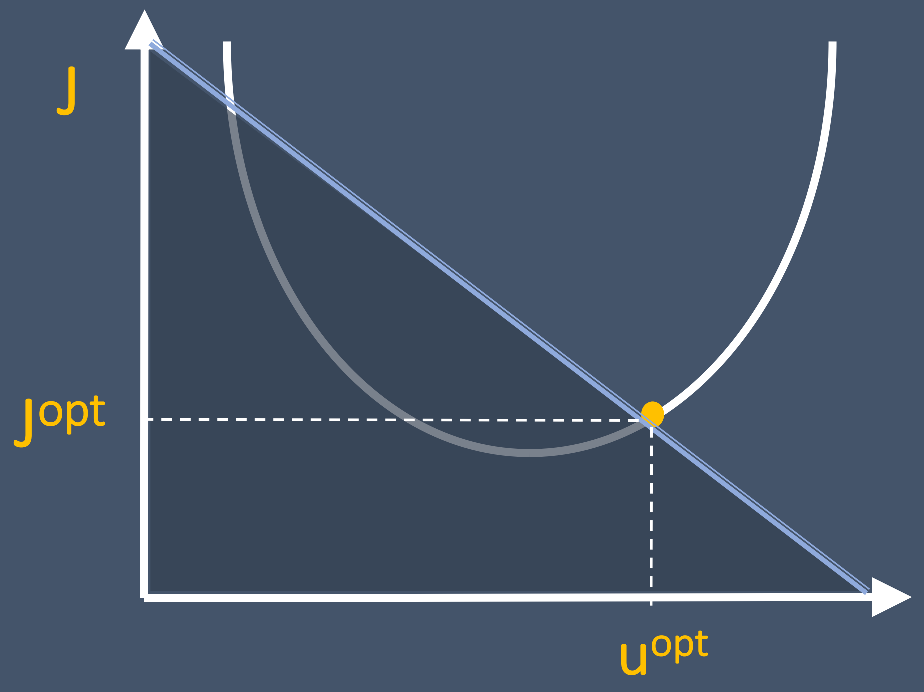
Do we always need to solve an optimization problem?

- We often know or can guess the **optimal solution**
  - Example: Drive from A → B ***in fuel economical way***

- CV = self-optimizing variable (less obvious)
- MV = gas pedal

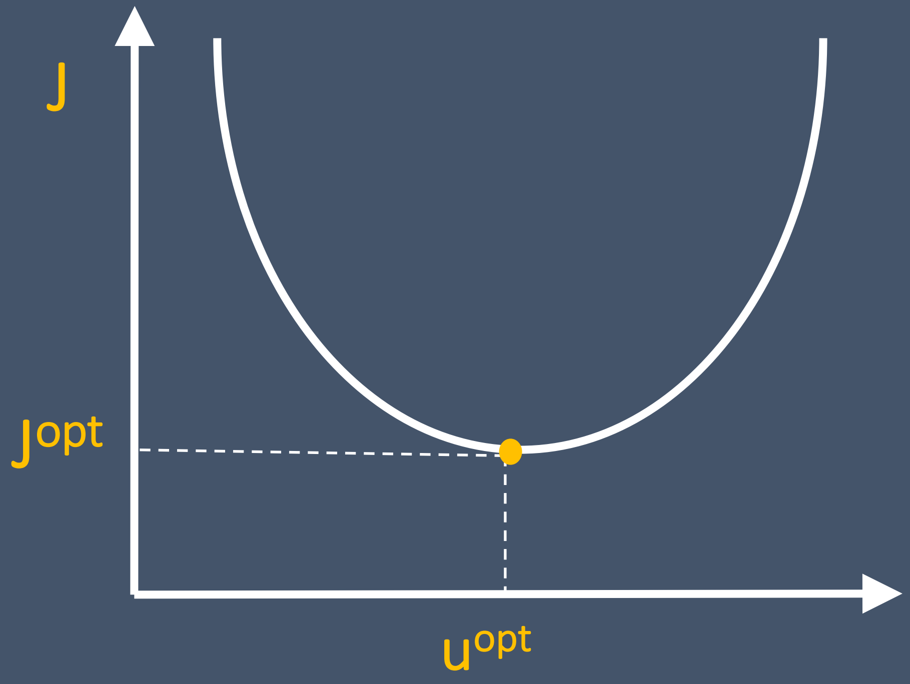
Easy case: Constrained optimum

- u: MV
- What to control?
  - Control the active constraint at its limit



The less obvious case: Unconstrained optimum

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



What to control ?

Control (in this order) :

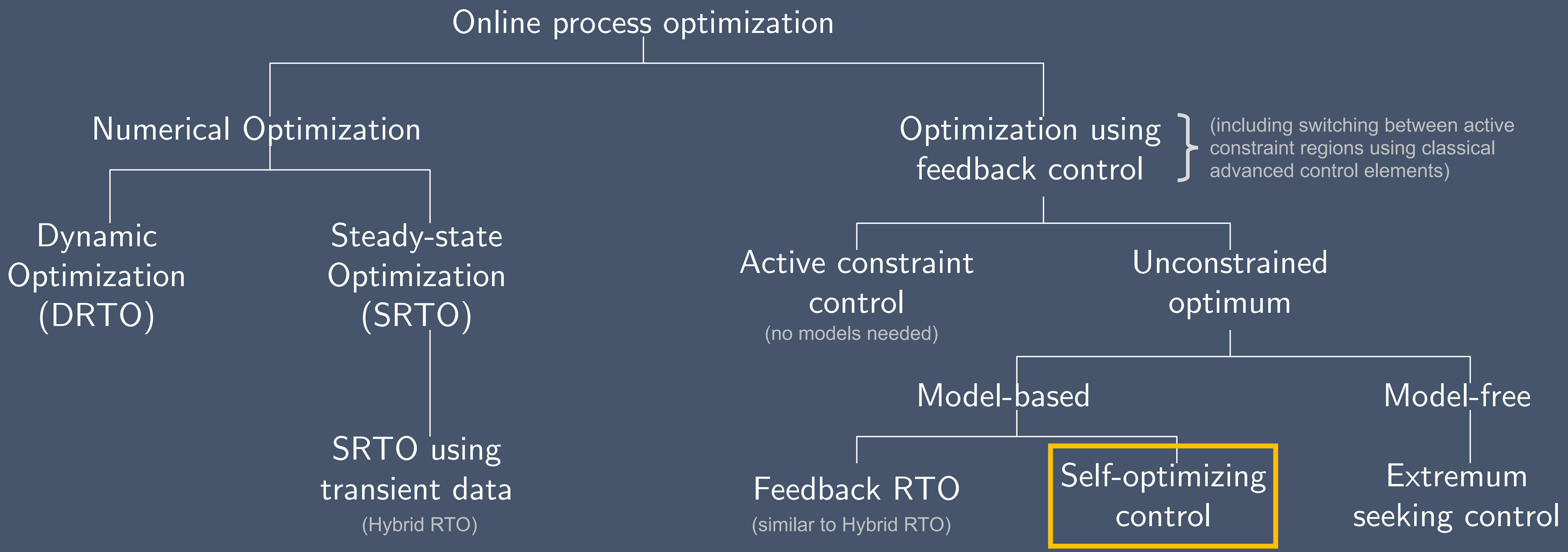
1. Active constraints
2. Self-optimizing variables (for remaining unconstrained MVs)

But what are these?

5. Self-optimizing Control

S. Skogestad ``Plantwide control: the search for the self-optimizing control structure'', J. Proc. Control, 10, 487-507 (2000).

Course Roadmap



Example: Optimal operation of runner

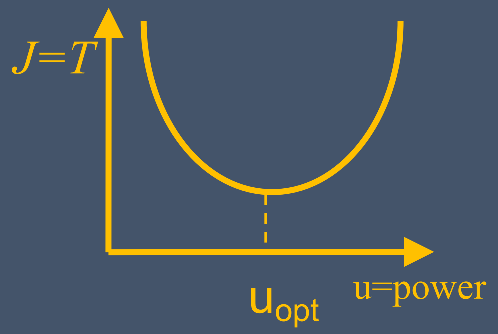
- Cost to be minimized, J=T
- One degree of freedom (u=power)
- What should we control?

1. Optimal operation of Sprinter

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

2. Optimal operation of Marathon runner

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



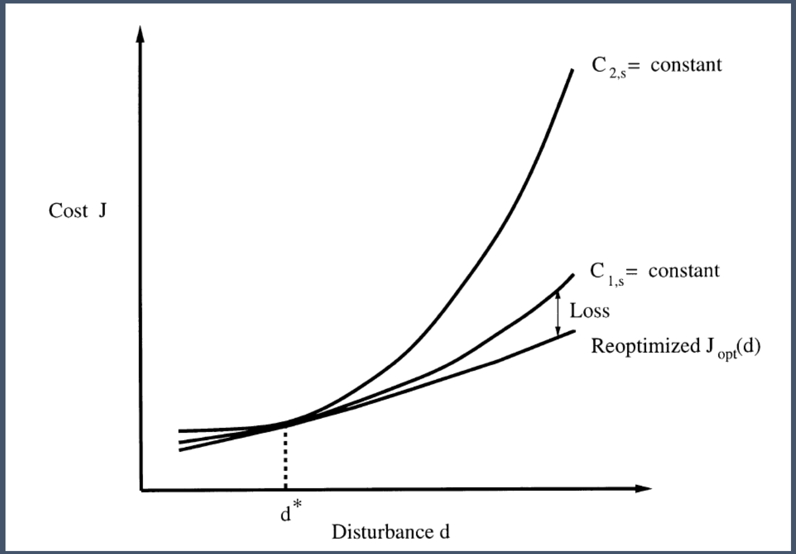
Marathon runner (40 km)

- 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

Self-optimizing control

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

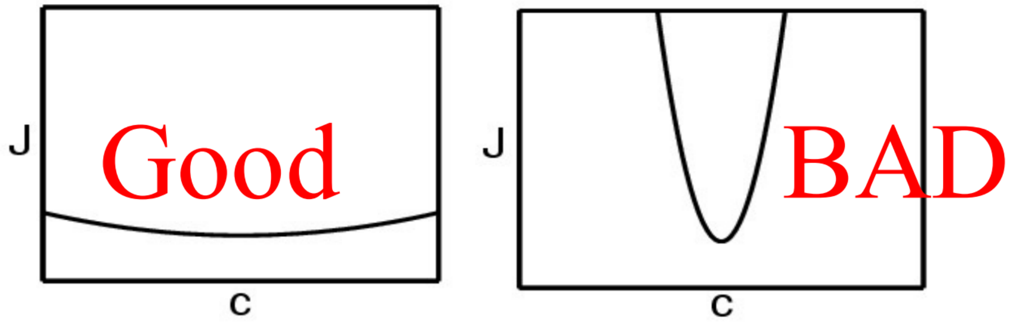
(Skogestad, 2000)



S. Skogestad ``Plantwide control: the search for the self-optimizing control structure'', J. Proc. Control, 10, 487-507 (2000).

What is a good self-optimizing variable?

- High gain
- Insensitive to disturbances
- Easily measurable (Available for feedback)
- e.g. ratio control



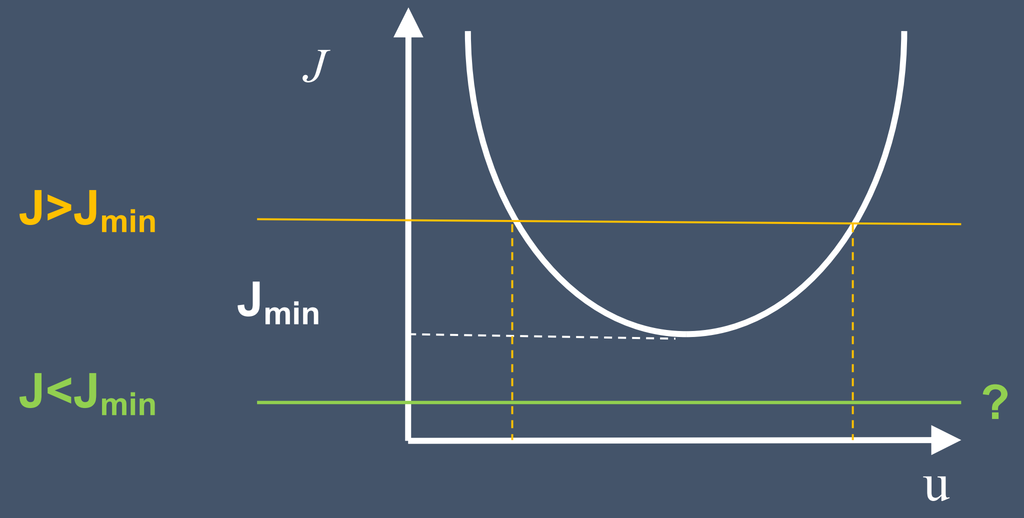
(b) Flat optimum: Implementation easy

(c) Sharp optimum: Sensitive to implementation erros

Is there any systematic procedure?

1. Sensitive variables: “Max. gain rule” (Gain= Minimum singular value)
2. “Brute force” loss evaluation
3. Gradient, $J_u=0$
4. Optimal linear combination of measurements, $c = Hy$

Unconstrained optimum: NEVER try to control directly the cost

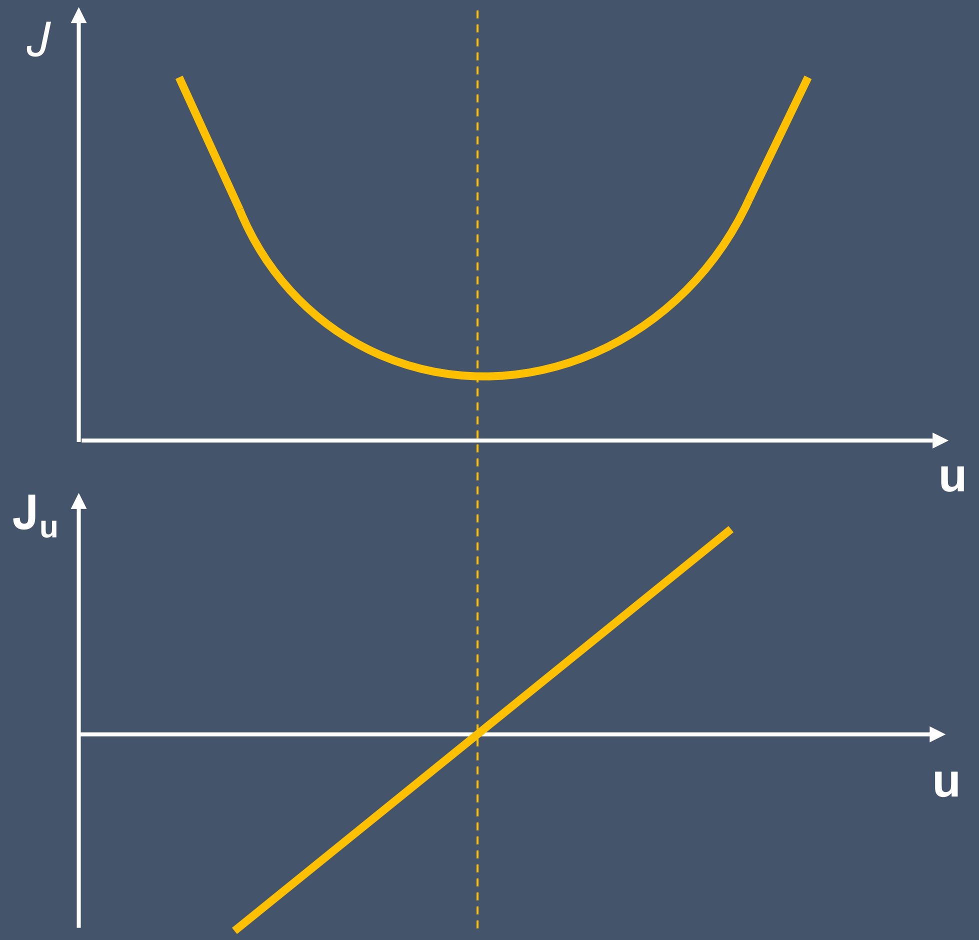


- Assume we want to minimize J (e.g., J = V = energy) - and we make the stupid choice os selecting CV = V = J
  - **Then setting J < Jmin: Gives infeasible operation (cannot meet constraints)**
  - **and setting J > Jmin: Forces us to be nonoptimal (two steady states: may require strange operation)**
- Control gain sign changes !! (cannot control with a single PID controller)

Ideal self-optimizing variable : Cost Gradient

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

- 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

In practise: use available measurements: **c = Hy**

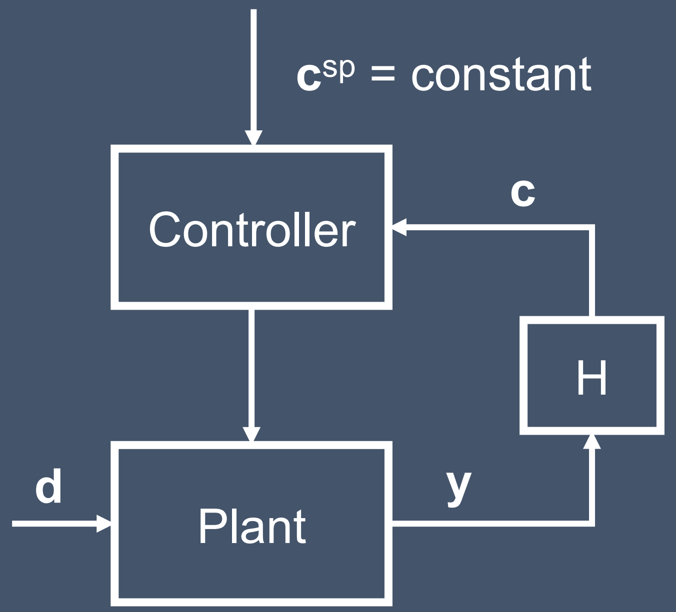
Task: Select optimal selection matrix H

- Single measurements (e.g. select y_2 and y_4)

$$H = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

- Linear combination of measurements

$$H = \begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{14} & h_{15} \\ h_{21} & h_{22} & h_{23} & h_{24} & h_{25} \end{bmatrix}$$



Nullspace method

- Linear combinations of measurements, **c = Hy**
- Given sufficient measurements $n_y \geq n_u + n_d$, and no measurement noise, select the optimal selection matrix H, such that

- **HF = 0**

- H - Optimal selection matrix
- F – Optimal sensitivity matrix $F = \frac{\partial y_{opt}}{\partial d}$

V. Alstad and S. Skogestad, ``Null Space Method for Selecting Optimal Measurement Combinations as Controlled Variables'', Ind.Eng.Chem.Res, 46 (3), 846-853 (2007).

Nullspace method

Controlling **c = Hy** to a constant setpoint, yields locally zero loss from optimal operation

Proof:

Measurements: $y = Gy\, u + Gyd\, d$

Assume $c = Hy$,

$\partial y_{opt} = F\, \partial d$

$\partial c_{opt} = HF\, \partial d$

To make ∂c_{opt} =0 for any ∂d, we must have $HF = 0$

V. Alstad and S. Skogestad, ``Null Space Method for Selecting Optimal Measurement Combinations as Controlled Variables'', Ind.Eng.Chem.Res, 46 (3), 846-853 (2007).

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]

$F = \frac{\partial y_{opt}}{\partial d}$ = [0.25 -0.2]^T

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!)

V. Alstad and S. Skogestad, ``Null Space Method for Selecting Optimal Measurement Combinations as Controlled Variables'', Ind.Eng.Chem.Res, 46 (3), 846-853 (2007).

Exact local method for H

"=0" in nullspace method (no noise)

$$\min_{\mathbf{H}} \left\| \underbrace{\mathbf{J_{uu}}^{1/2}(\mathbf{HG^y})^{-1}}_{\text{"Minimize" in Maximum gain rule}} \underbrace{\overbrace{\mathbf{H}[\mathbf{F}}\mathbf{W_d} \quad \mathbf{W_{n^y}}]}_{\text{"Scaling" } S_1} \right\|_2$$

"Minimize" in Maximum gain rule
(maximize S_1 G $J_{uu}^{-1/2}$, $G=HG^y$)

"Scaling" S_1

Analytical Solution:

$$\mathbf{H}^{\mathsf{T}} = (\mathbf{YY}^{\mathsf{T}})^{-1}\mathbf{G^y}$$

where, $\mathbf{Y} = [\mathbf{FW_d} \quad \mathbf{W_{n^y}}]$

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

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

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

Note: Must also choose setpoint for self-optimizing variables **c**

- Many cases: Fix at nominal optimum
- Some cases: Slowly update by optimizing layer

CSTR case study

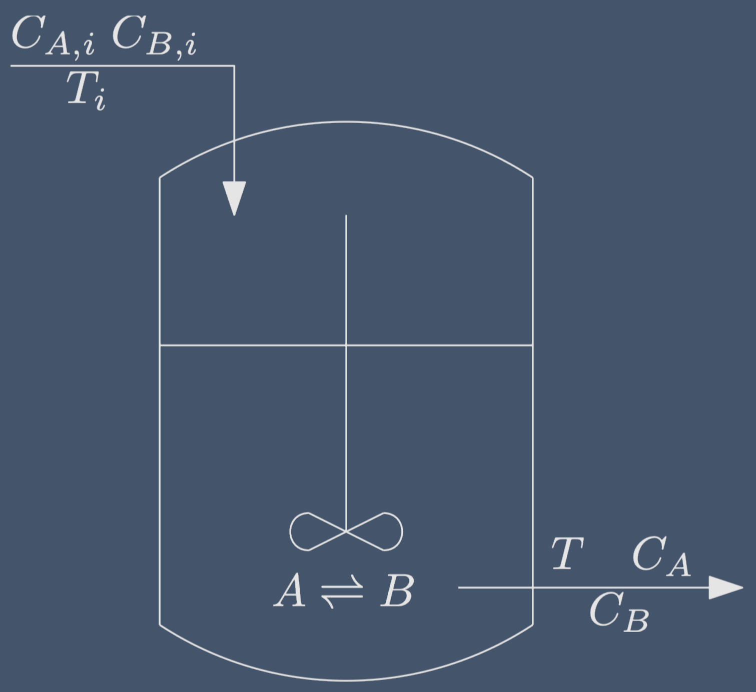
$$A \rightleftharpoons B$$

$$\mathbf{x} = [C_A \quad C_B \quad T]^\mathsf{T}$$

$$\mathbf{u} = T_i$$

$$\mathbf{d} = [C_{A_{in}} \quad C_{B_{in}}]^\mathsf{T}$$

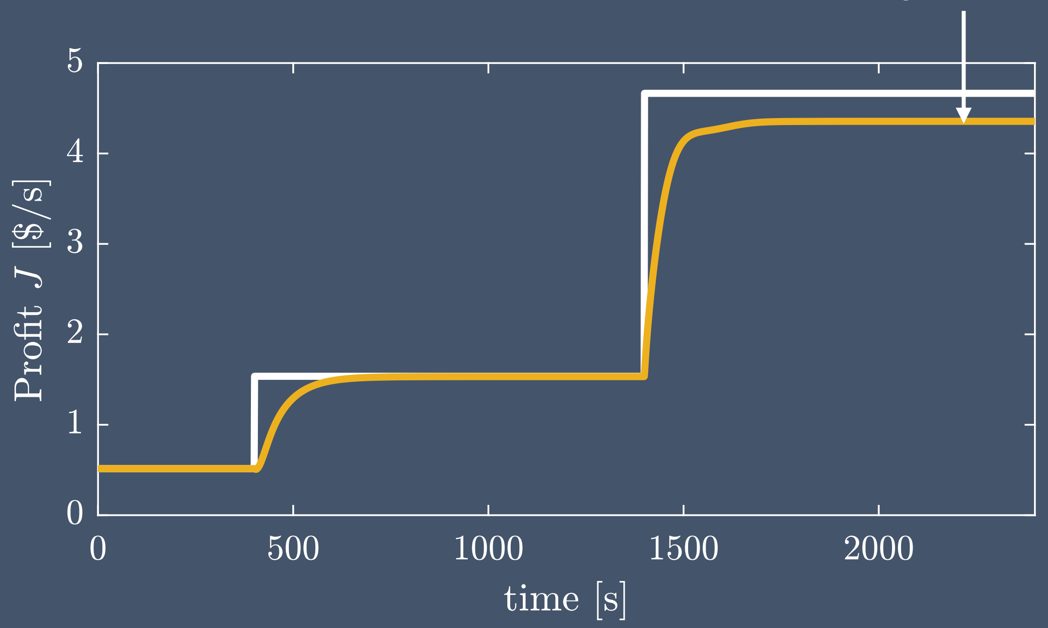
$$J = -p_{c_B} C_B + (p_{T_{in}} T_{in})^2$$



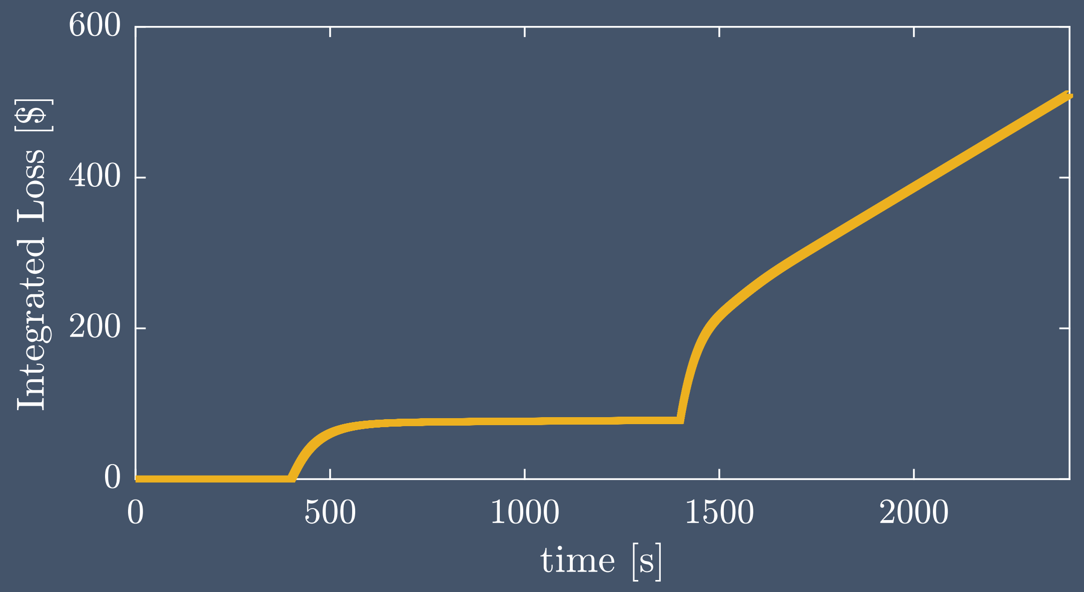
Economou, C. G.; Morari, M.; Palsson, B. O. Internal model control: Extension to nonlinear system. Industrial & Engineering Chemistry Process Design and Development 1986, 25, 403–411.

Simulation results c = Hy

Acceptable loss



Not so acceptable in the long run?



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

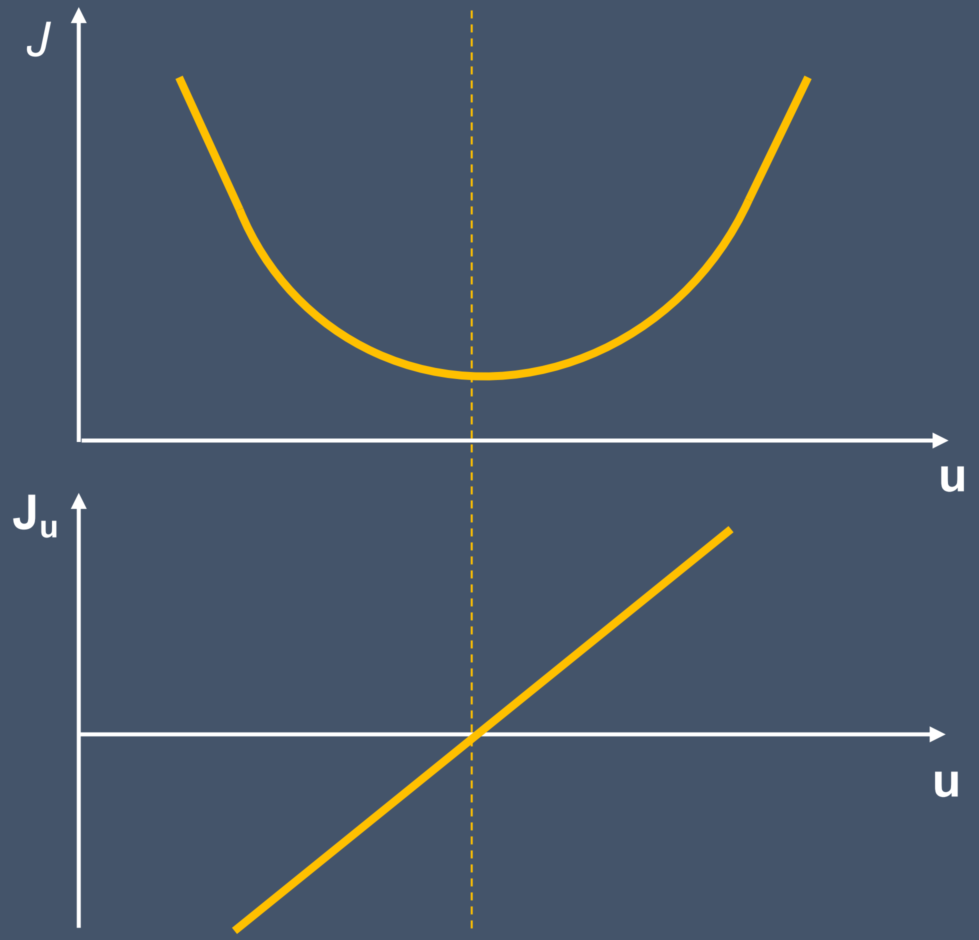
Challenges SOC:

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

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.



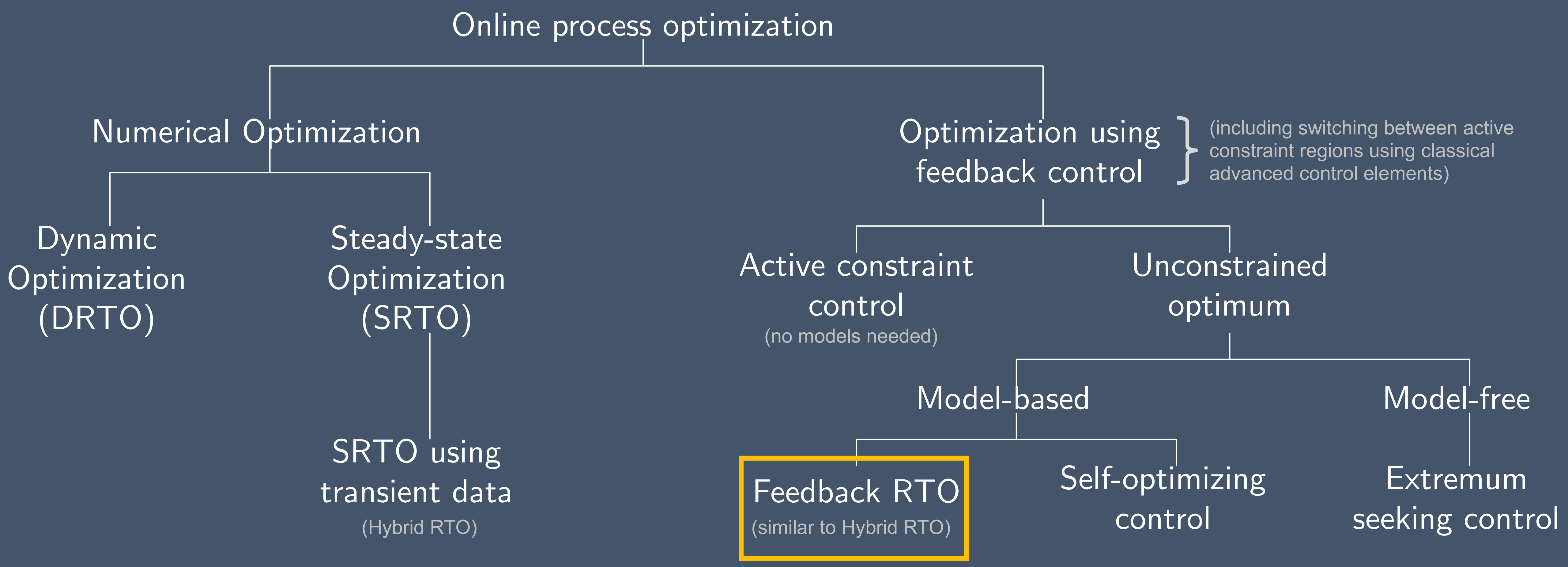
Problem: We do not usually have gradients as measurements; But we can estimate !

6. Feedback RTO

Model-based gradient estimation

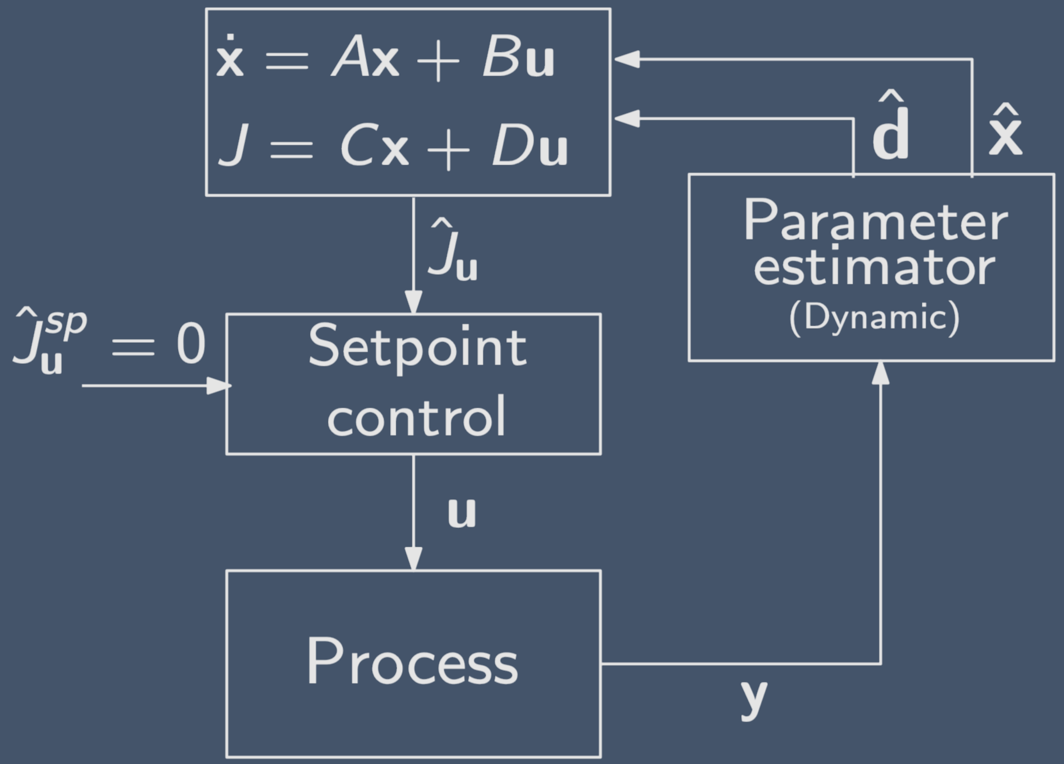
Krishnamoorthy, D., Jahanshahi, E. and Skogestad, S., 2019. Steady-state real-time optimization using transient measurements. *Industrial and Engineering Chemistry Research 115*, pp.34-45.

Course Roadmap

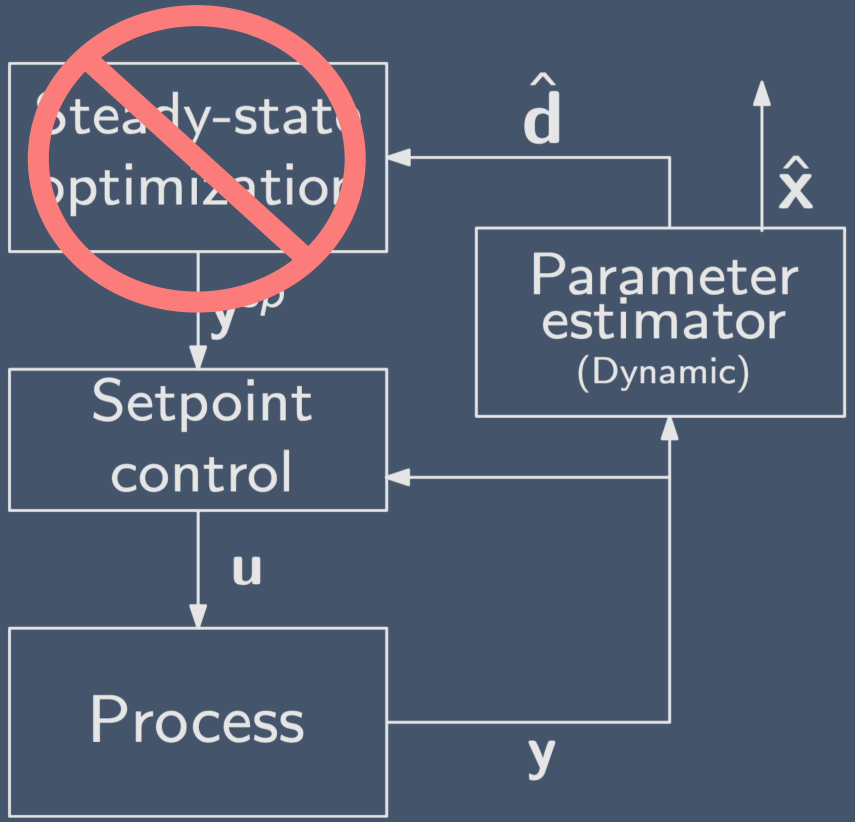


Feedback RTO: Replace steady-state optimization by feedback control

Feedback RTO



Hybrid RTO



D. Krishnamoorthy, E. Jahanshahi, and S. Skogestad. A feedback real time optimization strategy using a novel steady-state gradient estimate and transient measurements. Industrial and Engineering Chemistry Research, 58: 207–216, 2019

Feedback RTO

- Step 1 – Linearize the dynamic model around the current operating point

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, \mathbf{d}) \qquad\qquad \dot{x} = Ax + Bu$$
$$\Rightarrow$$
$$J = \mathbf{g}(\mathbf{x}, \mathbf{u}, \mathbf{d}) \qquad\qquad J = Cx + Du$$

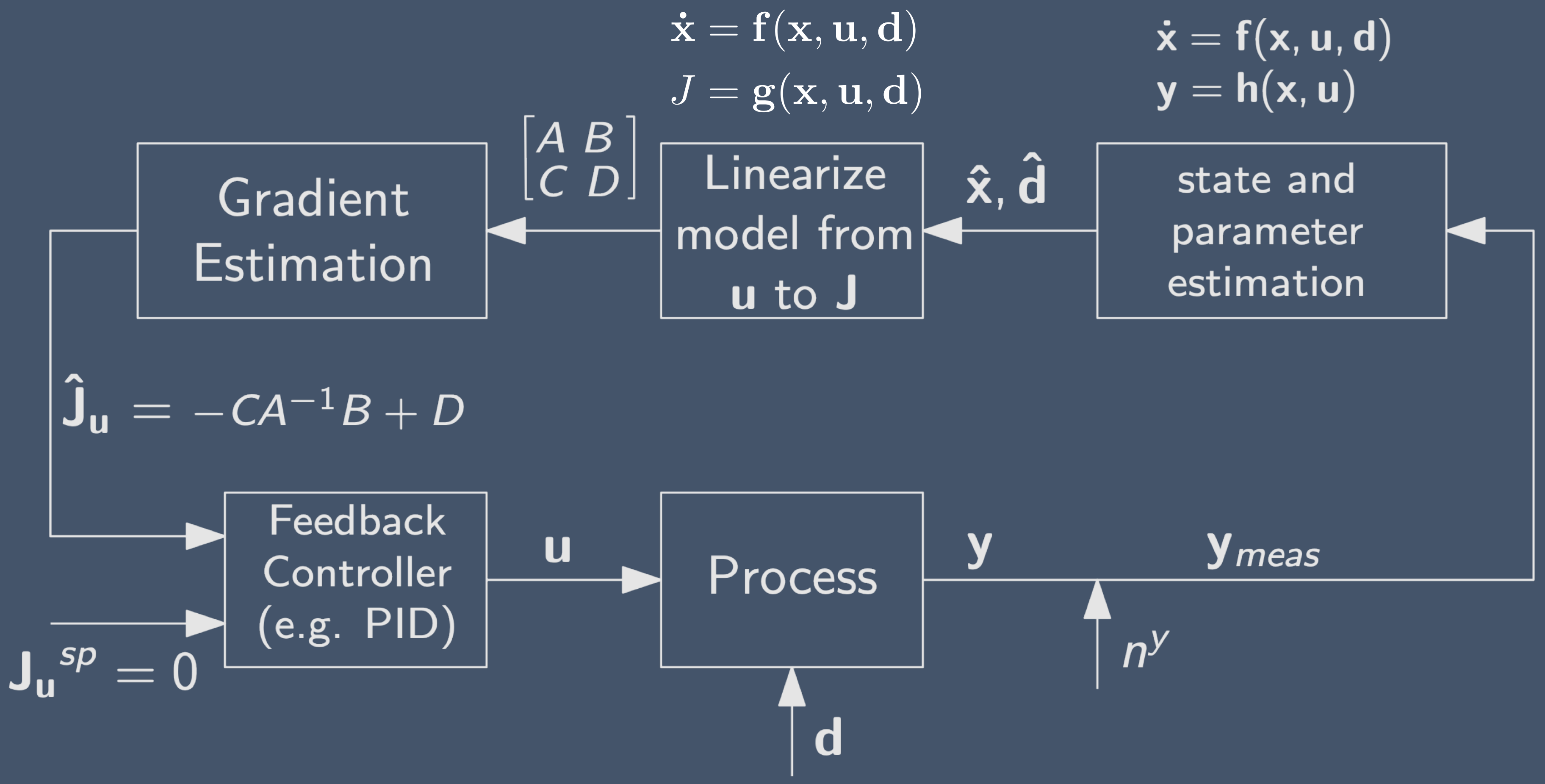
- Step 2 – At steady-state $\dot{x} = 0$

$$J = \underbrace{(-CA^{-1}B + D)}_{J_u}u$$

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

D. Krishnamoorthy, E. Jahanshahi, and S. Skogestad. A feedback real time optimization strategy using a novel steady-state gradient estimate and transient measurements. Industrial and Engineering Chemistry Research, 58: 207–216, 2019

Feedback RTO



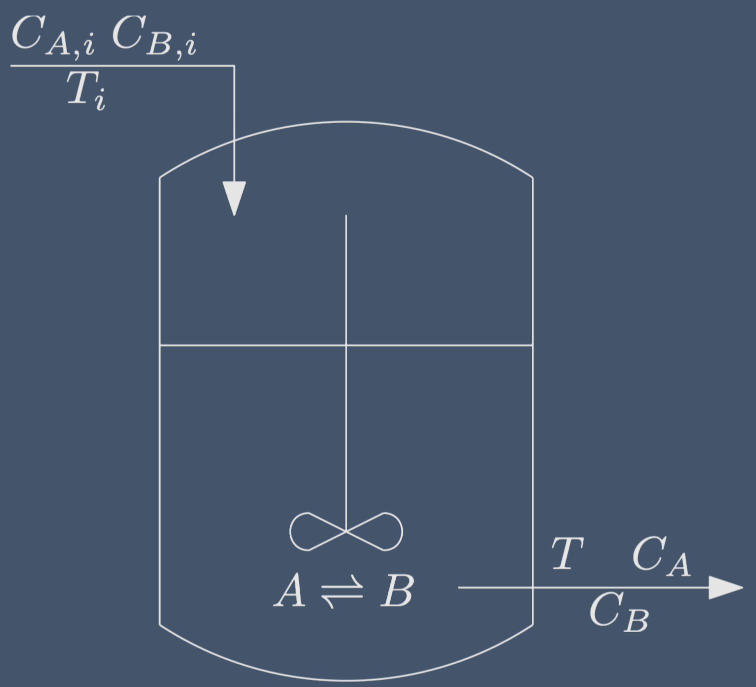
D. Krishnamoorthy, E. Jahanshahi, and S. Skogestad. A feedback real time optimization strategy using a novel steady-state gradient estimate and transient measurements. Industrial and Engineering Chemistry Research, 58: 207–216, 2019

CSTR case study

$$A \rightleftharpoons B$$

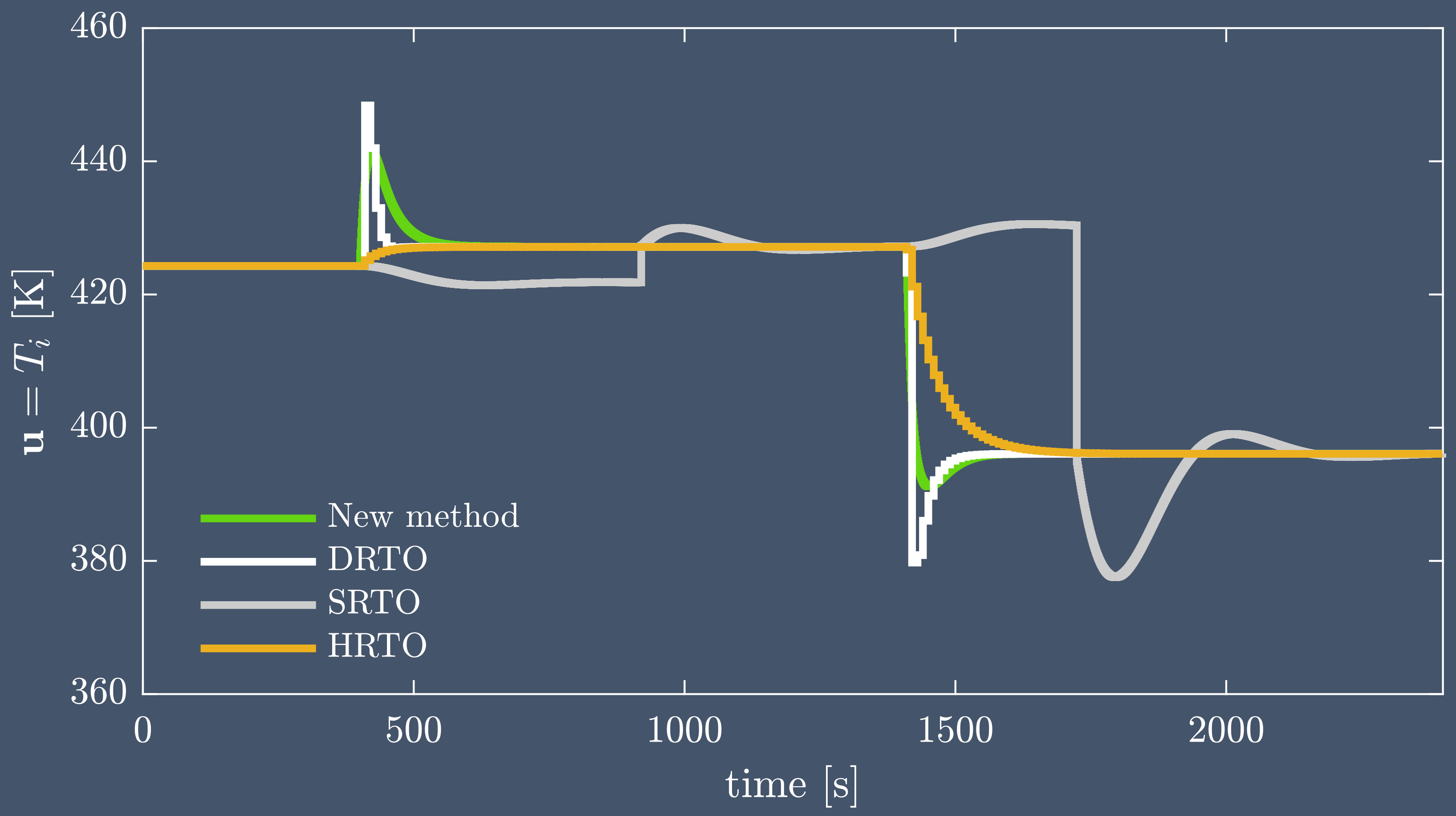
$$\mathbf{x} = [C_A \quad C_B \quad T]^\mathsf{T}$$
$$\mathbf{u} = T_i$$
$$\mathbf{d} = [C_{A_{in}} \quad C_{B_{in}}]^\mathsf{T}$$

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



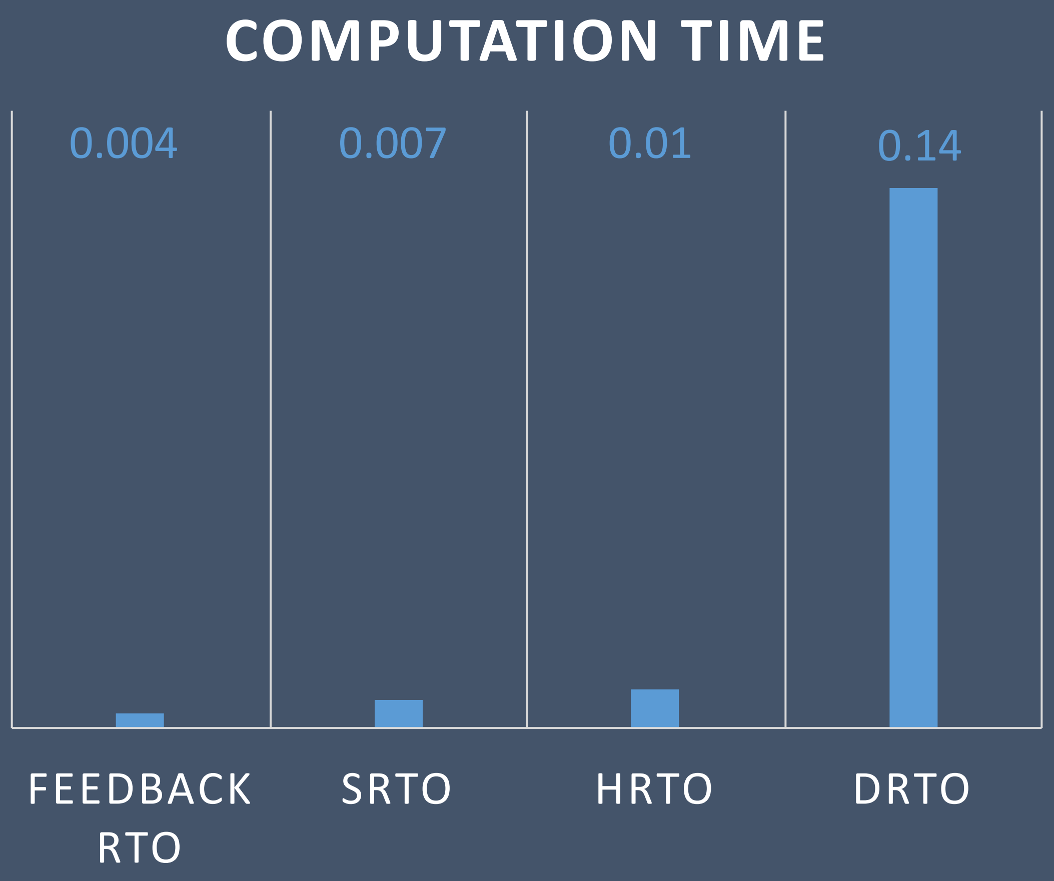
Economou, C. G.; Morari, M.; Palsson, B. O. Internal model control: Extension to nonlinear system. Industrial & Engineering Chemistry Process Design and Development 1986, 25, 403–411.

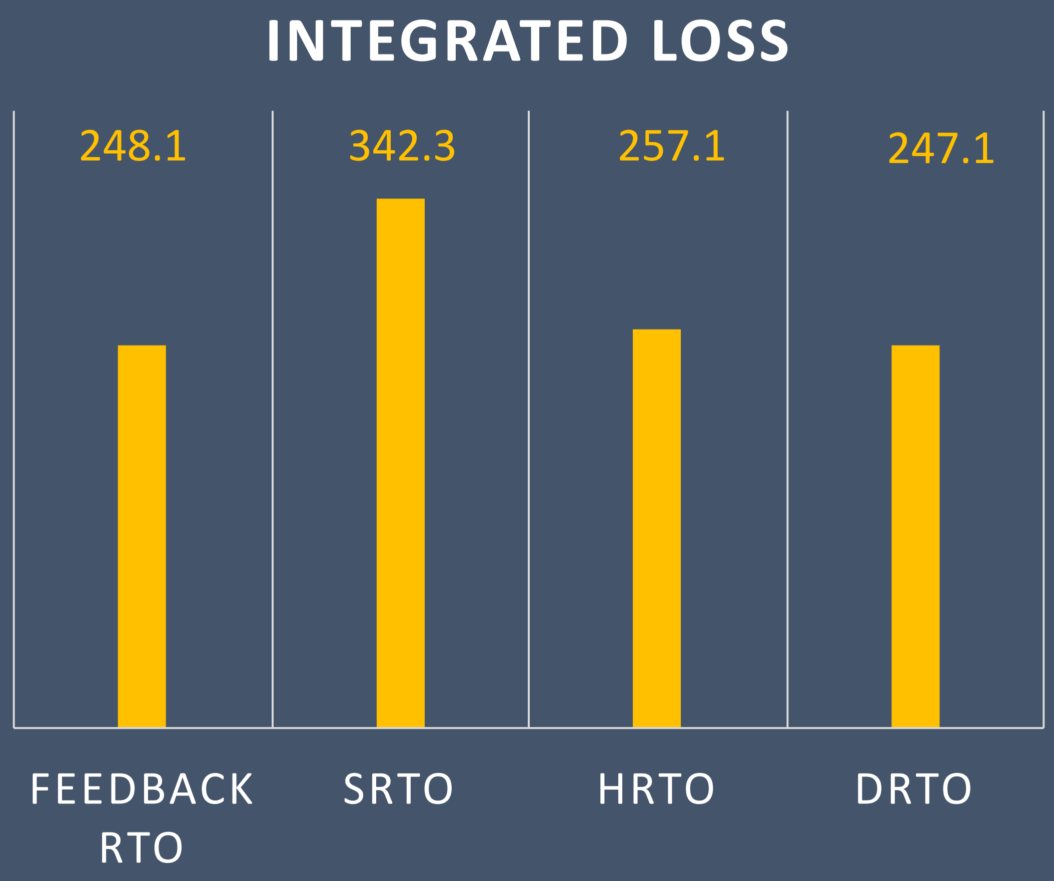
Comparison of RTO approaches: MV



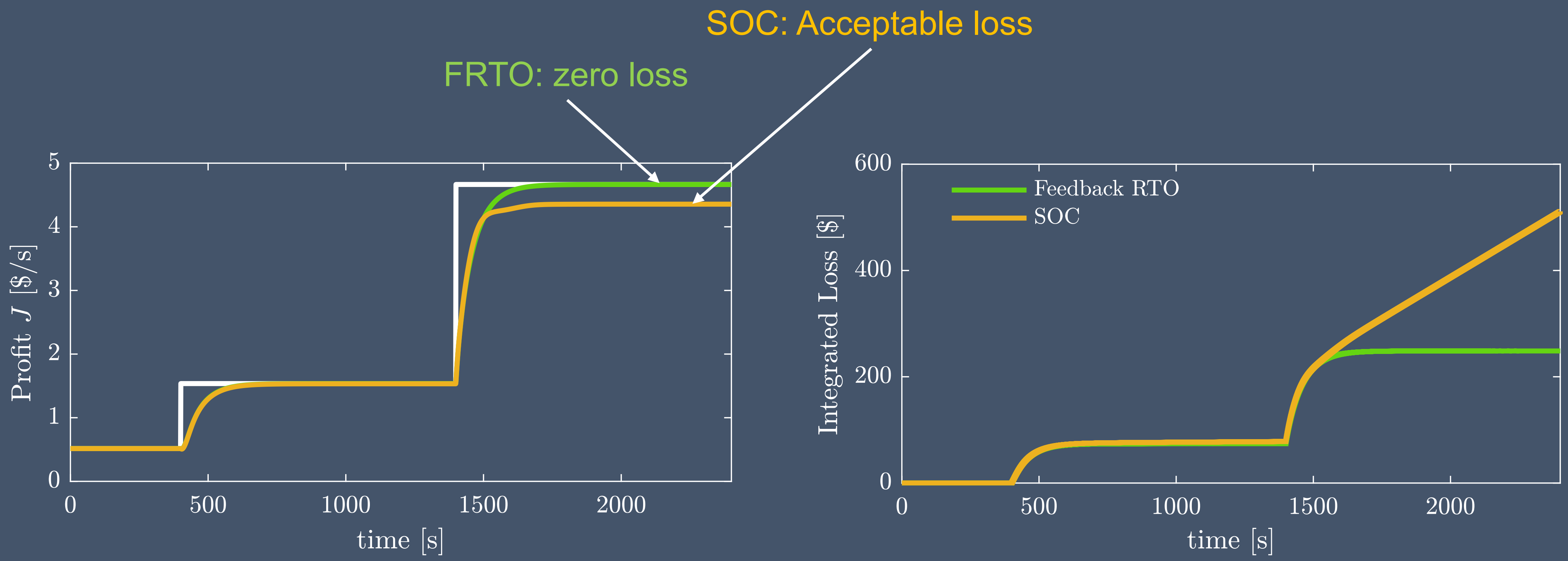
t = 400 s, d1: Increase C_{Ain}
t = 1400 s, d2: Increase C_{Bin}

Comparison of RTO approaches



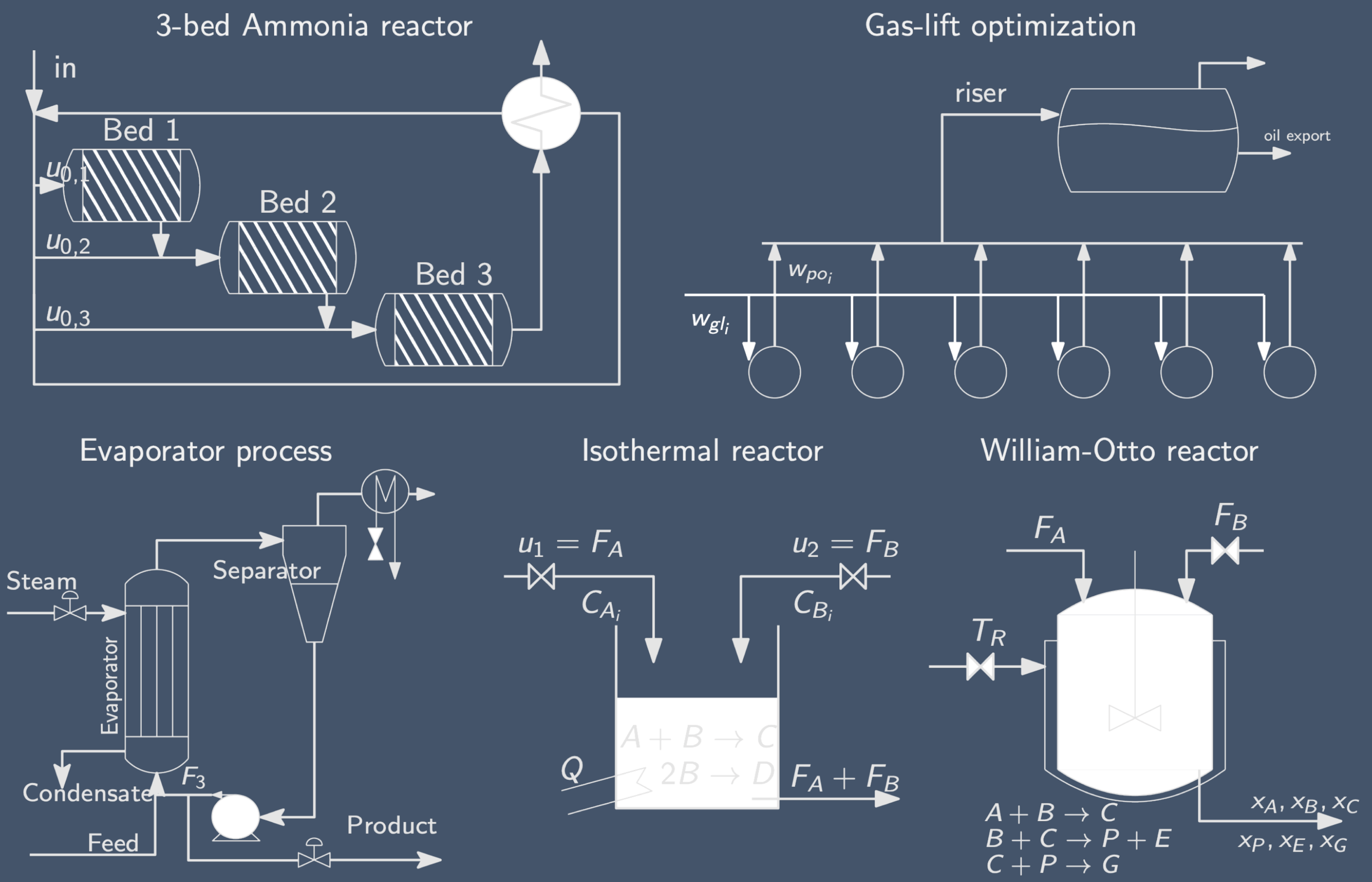


Comparison with self-optimizing control



For MATLAB code, contact: dineshk@ntnu.no

Other application examples



- Bonnowitz et al,. CACE (2018)
- Krishnamoorthy et al., IFAC OOGP (2018)
- Krishnamoorthy et al., (2019) PSE Asia
- Krishnamoorthy & Skogestad, I&ECR (2019)
- Krishnamoorthy & Skogestad, CACE (2020)

4. Generalized framework for optimal CV selection

Linear Gradient Combination

Consider the Optimization problem

$$\min_{\mathbf{u}} \quad J(\mathbf{u},\mathbf{d})$$

s.t.

$$\mathbf{g}(\mathbf{u},\mathbf{d}) \leq 0$$

$$\mathbf{u} \in \mathbb{R}^{n_u} \qquad \mathbf{g} \in \mathbb{R}^{n_g}$$

no .of active constraints : $n_a \leq n_g$

$$\mathbf{g}_{\mathbb{A}} \subseteq \mathbf{g}$$

- Lagrangian function:

$$\min_{\mathbf{u}} \mathcal{L}(\mathbf{u},\mathbf{d}) = \boxed{J(\mathbf{u},\mathbf{d}) + \lambda^T \mathbf{g}(\mathbf{u},\mathbf{d})}$$

- KKT conditions / Necessary conditions of optimality:

$$\nabla_{\mathbf{u}}\mathcal{L}(\mathbf{u},\mathbf{d}) = \nabla_{\mathbf{u}} J(\mathbf{u},\mathbf{d}) + \lambda^{\mathrm{T}} \nabla_{\mathbf{u}} \mathbf{g}(\mathbf{u},\mathbf{d}) = 0$$

$\mathbf{g}(\mathbf{u},\mathbf{d}) \leq 0$ (Primal feasibility)

$\lambda^{\mathrm{T}} \mathbf{g}(\mathbf{u},\mathbf{d}) = 0$ (complementary slackness)

$\lambda \geq 0$ (Dual feasibility)

Complementary slackness $\Rightarrow$

$$\mathbf{g}_{\mathbb{A}}(\mathbf{u},\mathbf{d}) = 0 \rightarrow \lambda_{\mathbb{A}} > 0$$
$$\mathbf{g}_{\mathbb{I}}(\mathbf{u},\mathbf{d}) < 0 \rightarrow \lambda_{\mathbb{I}} = 0$$

$$\mathcal{L}(\mathbf{u},\mathbf{d}) = J(\mathbf{u},\mathbf{d}) + \begin{bmatrix}\lambda_{\mathbb{A}} & \lambda_{\mathbb{I}}\end{bmatrix}^{\mathrm{T}} \begin{bmatrix}\mathbf{g}_{\mathbb{A}}(\mathbf{u},\mathbf{d}) \\ \mathbf{g}_{\mathbb{I}}(\mathbf{u},\mathbf{d})\end{bmatrix} \Rightarrow \boxed{J(\mathbf{u},\mathbf{d}) + \lambda_{\mathbb{A}}^{\mathrm{T}} \mathbf{g}_{\mathbb{A}}(\mathbf{u},\mathbf{d})}$$

Linear gradient combination as self-optimizing CV

- Necessary condition of optimality is satisfied when:

$$\nabla_{\mathbf{u}} J(\mathbf{u}, \mathbf{d}) + \lambda_{\mathbb{A}}^{\mathsf{T}} \nabla_{\mathbf{u}} \mathbf{g}_{\mathbb{A}}(\mathbf{u}, \mathbf{d}) = 0$$

$$\Rightarrow \nabla_{\mathbf{u}} J(\mathbf{u}, \mathbf{d}) = -\lambda_{\mathbb{A}}^{T} \nabla_{\mathbf{u}} \mathbf{g}_{\mathbb{A}}(\mathbf{u}, \mathbf{d})$$

- Pre-multiply by N, such that **N** is in the nullspace of the constraint gradient

$$\mathbf{N}^{\mathsf{T}} \nabla_{\mathbf{u}} J(\mathbf{u}, \mathbf{d}) = -\underbrace{\mathbf{N}^{\mathsf{T}} \nabla_{\mathbf{u}} \mathbf{g}_{\mathbb{A}}(\mathbf{u}, \mathbf{d})^{\mathsf{T}}}_{=0} \lambda$$

- Self-optimizing variable: Linear gradient combination

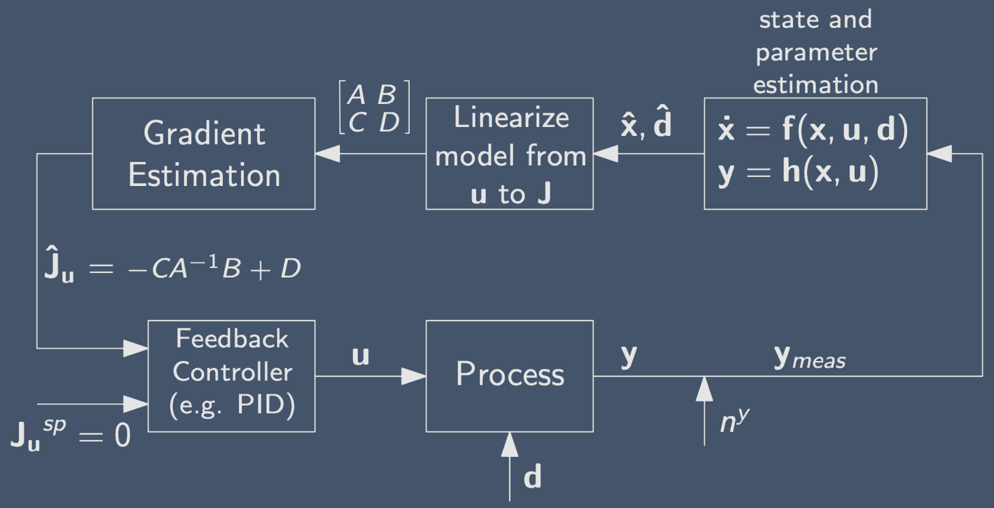
$$\boxed{\mathbf{c} = \mathbf{N}^{\mathsf{T}} \nabla_{\mathbf{u}} J(\mathbf{u}, \mathbf{d}) = 0}$$

Krishnamoorthy, D. and Skogestad, S. (2020), Linear Combination of Gradients as Optimal Controlled Variables, Computer Aided Chemical Engineering

What to control ?

Control (in this order) :

1. Active constraints $\mathbf{g}_{\mathbb{A}}(\mathbf{u}, \mathbf{d}) = 0$
2. Linear gradient combination $\mathbf{c} = \mathbf{N}^{\mathsf{T}} \nabla_{\mathbf{u}} J(\mathbf{u}, \mathbf{d}) = 0$



Different operating scenarios

$n_a = n_u$

- Case 1: fully constrained - active constraint control

Krishnamoorthy, D., and Skogestad, S., 2019. Online process optimization with changes in active constraint sets using simple feedback control structures, *Ind. Eng. Chem. Res.* Vol. 58 (30), pp. 13555-13567

Different operating scenarios

$n_a = n_u$ • Case 1: fully constrained - active constraint control

$n_a = 0$ • Case 2: fully unconstrained - control cost gradient to zero (i.e. **N** = ***I***)

$0 < n_a < n_u$ • Case 3: partially constrained – active constraint control + linear gradient combination $\mathbf{c} = \mathbf{N}^{\mathsf{T}} \nabla_{\mathbf{u}} J(\mathbf{u}, \mathbf{d}) = 0$

$n_a > n_u$ • Case 4: over constrained – active constraint control (give up less important constraints)

Krishnamoorthy, D., and Skogestad, S., 2019. Online process optimization with changes in active constraint sets using simple feedback control structures, *Ind. Eng. Chem. Res.* Vol. 58 (30), pp. 13555-13567

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**

Recap: What to control ?

Control (in this order) :

1. Active constraints (no need for models)
2. Self-optimizing variables (for remaining unconstrained MVs)
   a. Linear measurement combination (model used offline)
   b. Linear gradient combination (model used online to estimate gradient)

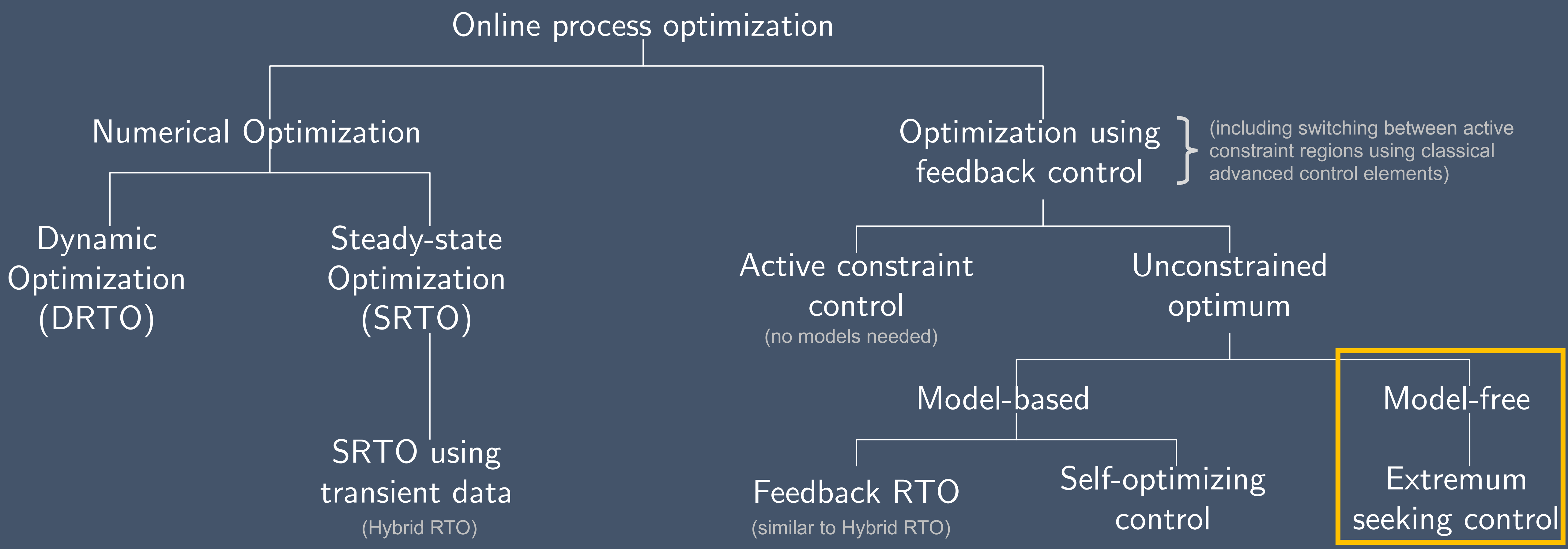
Can we estimate the gradient model-free?

Model-free approaches for unconstrained optimum

7. Extremum Seeking Control

Data driven optimization approach

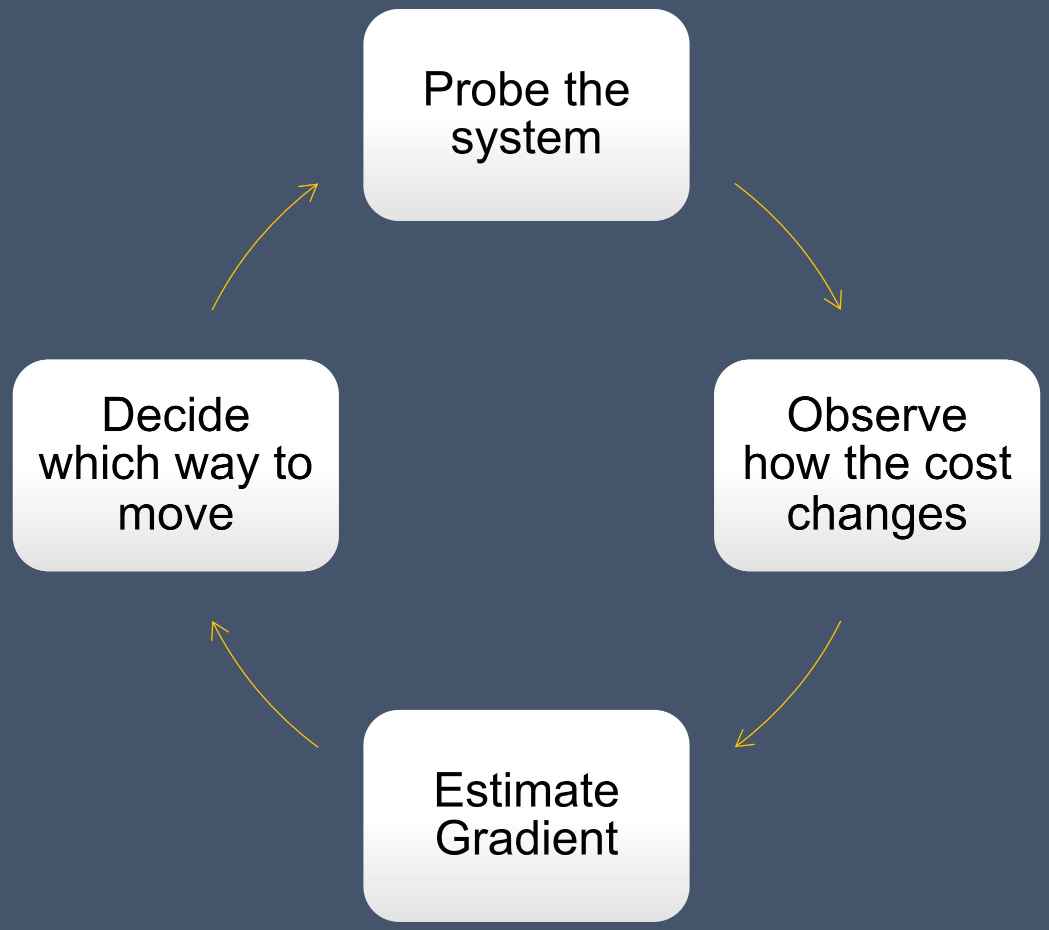
Course Roadmap

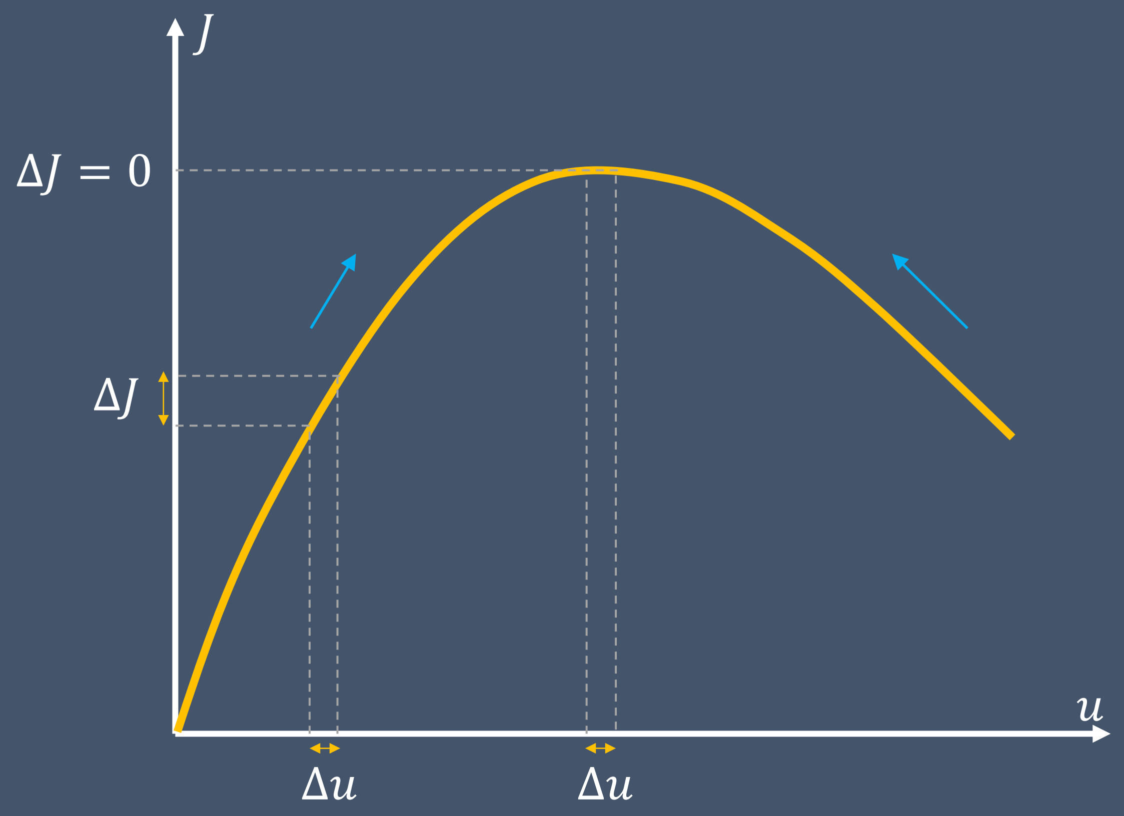


Data-driven gradient estimation

- We do not use a model to estimate the steady-state 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

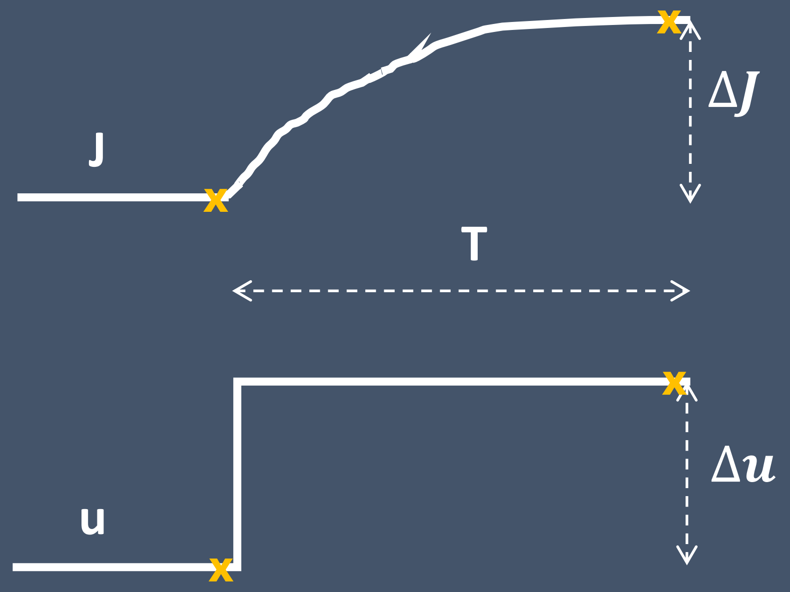
Extremum Seeking Control – Main idea





- Assume the plant to behave like a static map

Steady-state gradient: Finite difference approach



- Choose T, such that the system reaches steady-state within T
- Perturb the input

$$\mathbf{u}(t) = \begin{cases} \mathbf{u}_k & 2kT \leq t < (2k+1)T \\ \mathbf{u}_k + \Delta\mathbf{u} & (2k+1)T \leq t < (2k+2)T \end{cases}$$

- Estimate the cost gradient

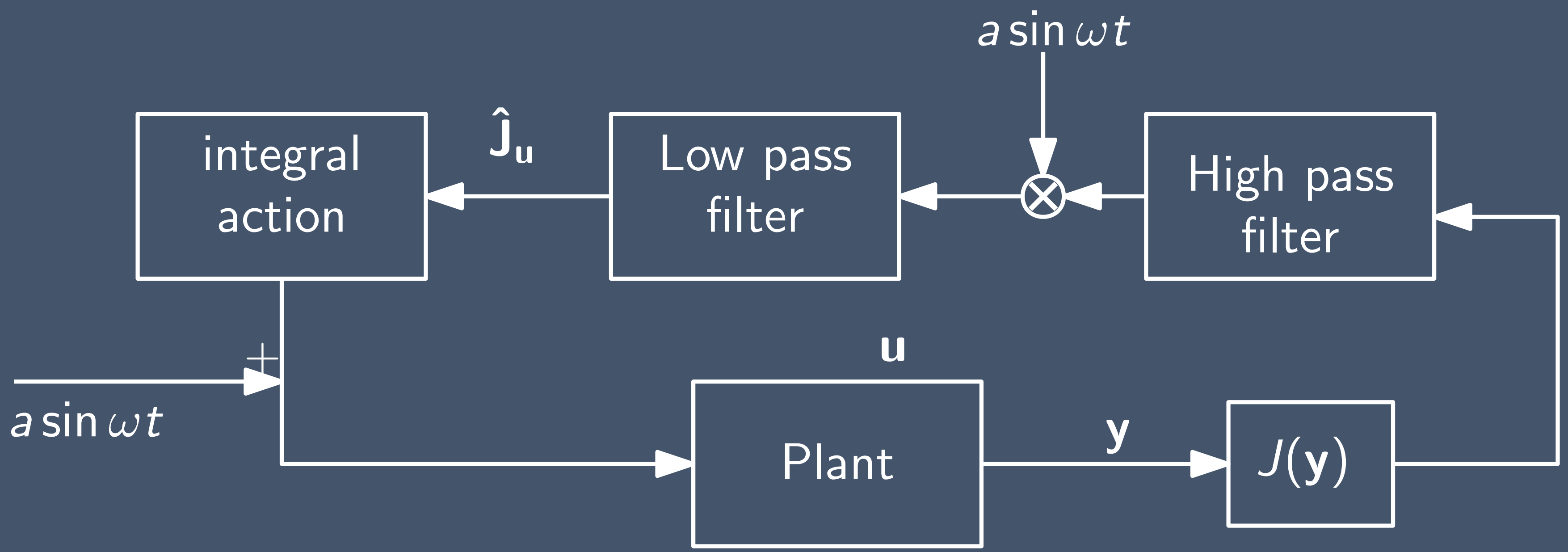
$$\mathbf{J_u} = \frac{\Delta J}{\Delta \mathbf{u}} = \frac{J((2k+2)T) - J((2k+1)T)}{\Delta \mathbf{u}}$$

- Update input (gradient descent)

$$\mathbf{u} = \mathbf{u} + \mathbf{K_I} \cdot \mathbf{J_u}$$

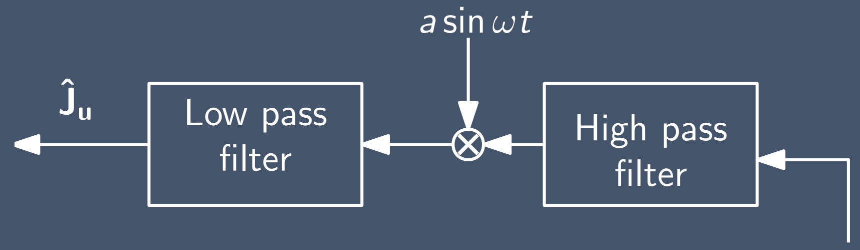
- NCO-tracking uses Finite difference approach

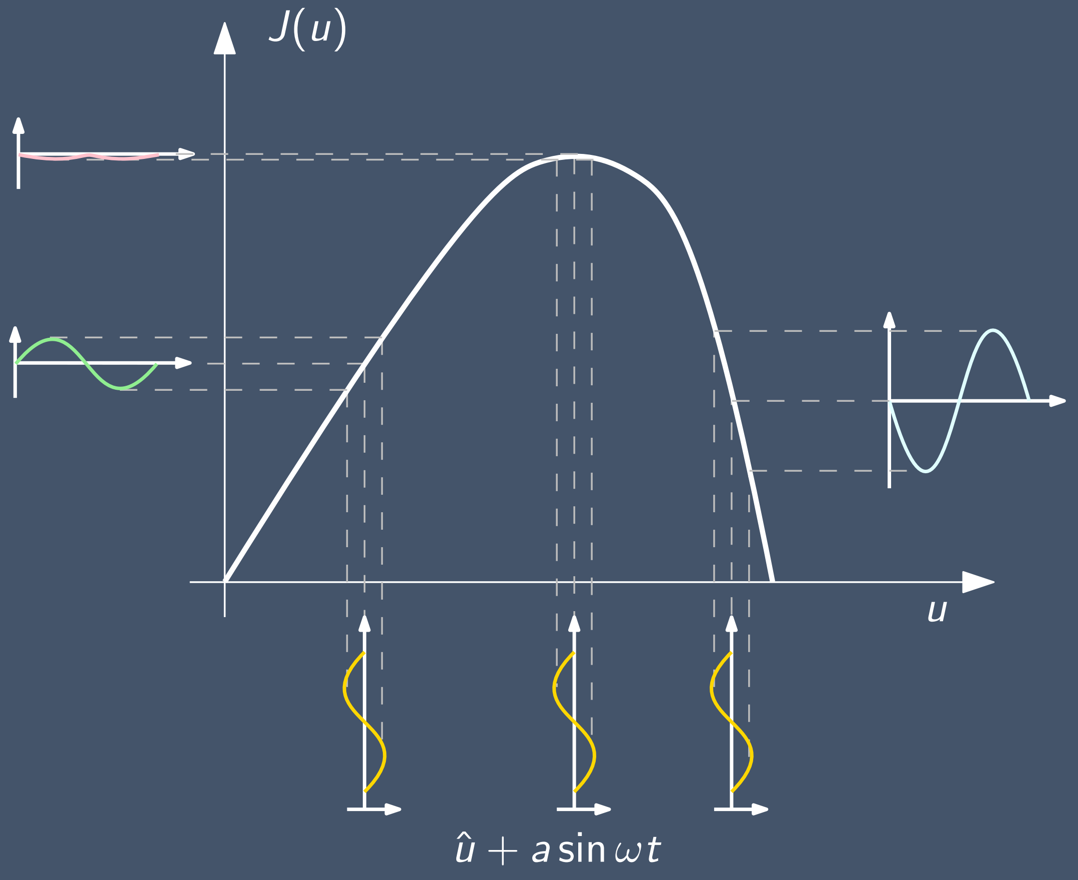
Classical Extremum seeking control



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

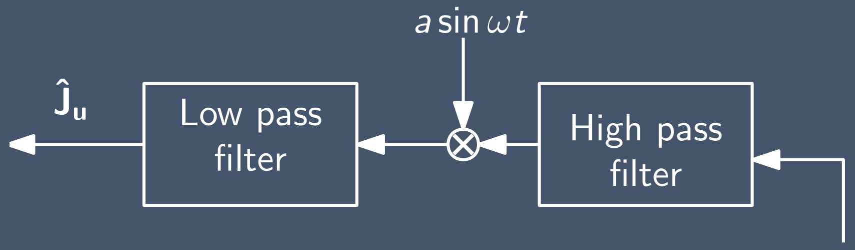
Sinusoidal perturbation

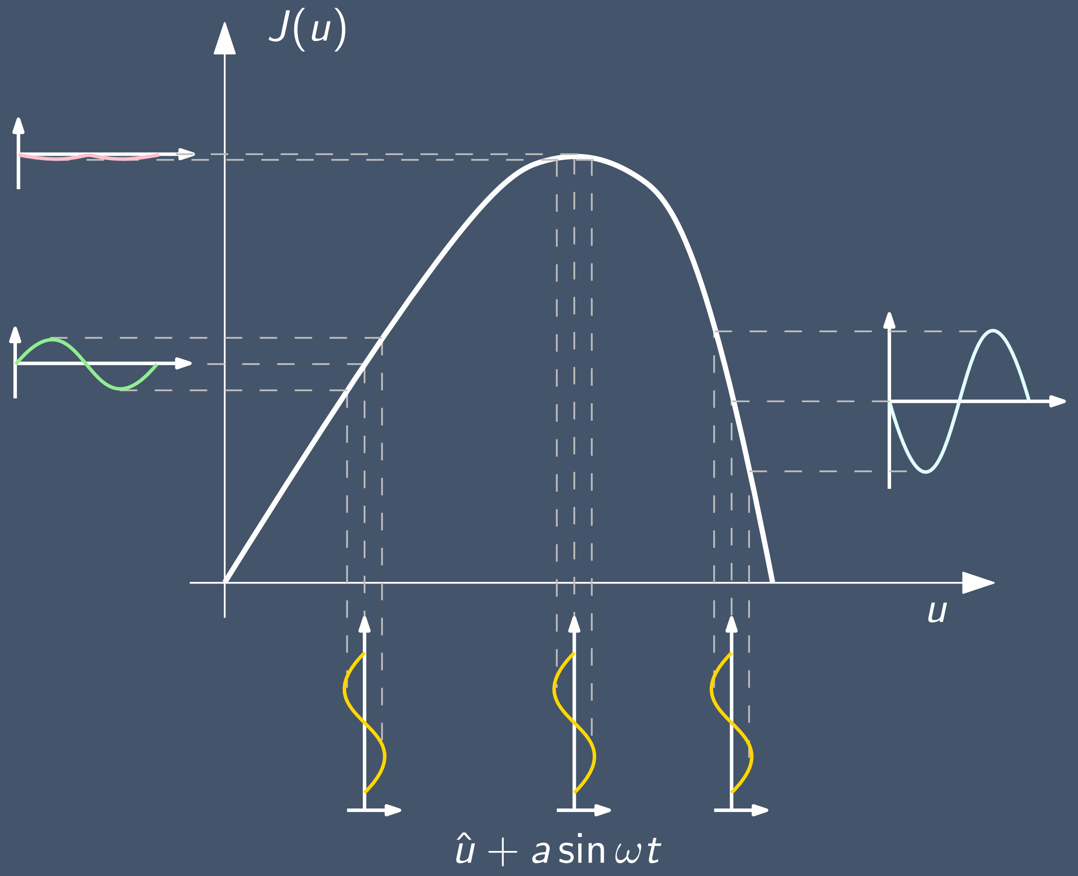


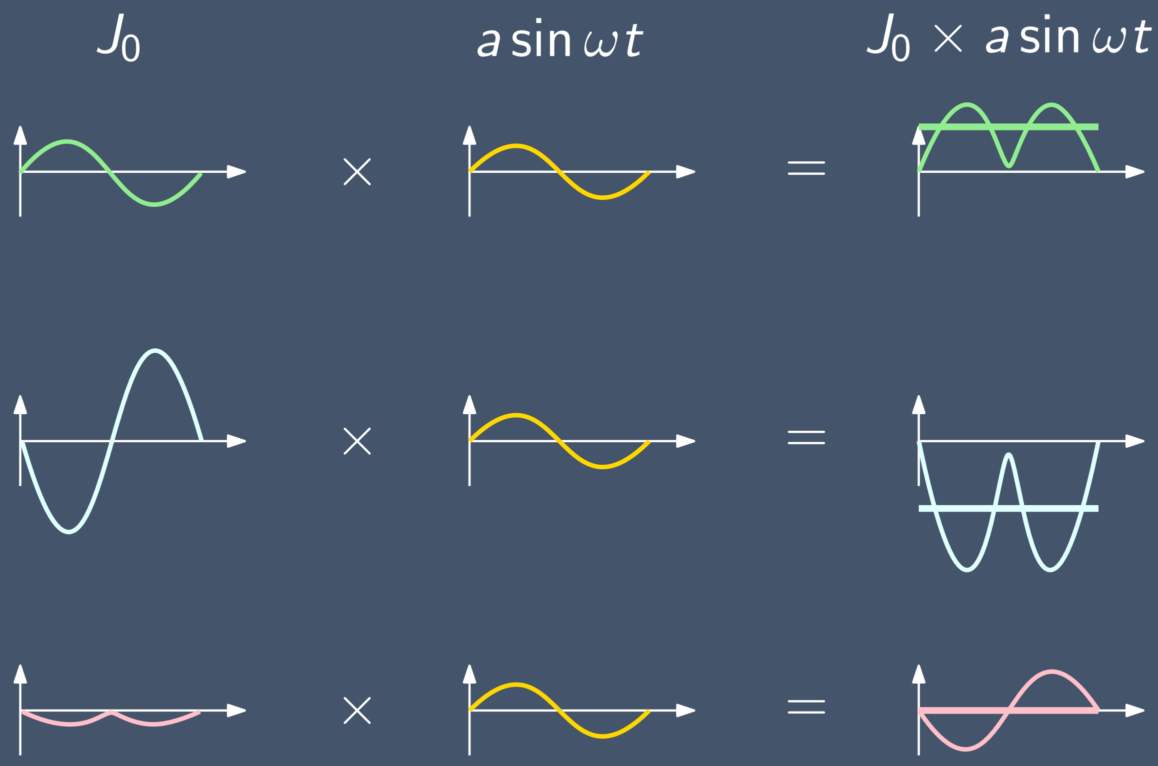


Special case of Fast Fourier Transform (FFT) - single frequency case

Sinusoidal perturbation



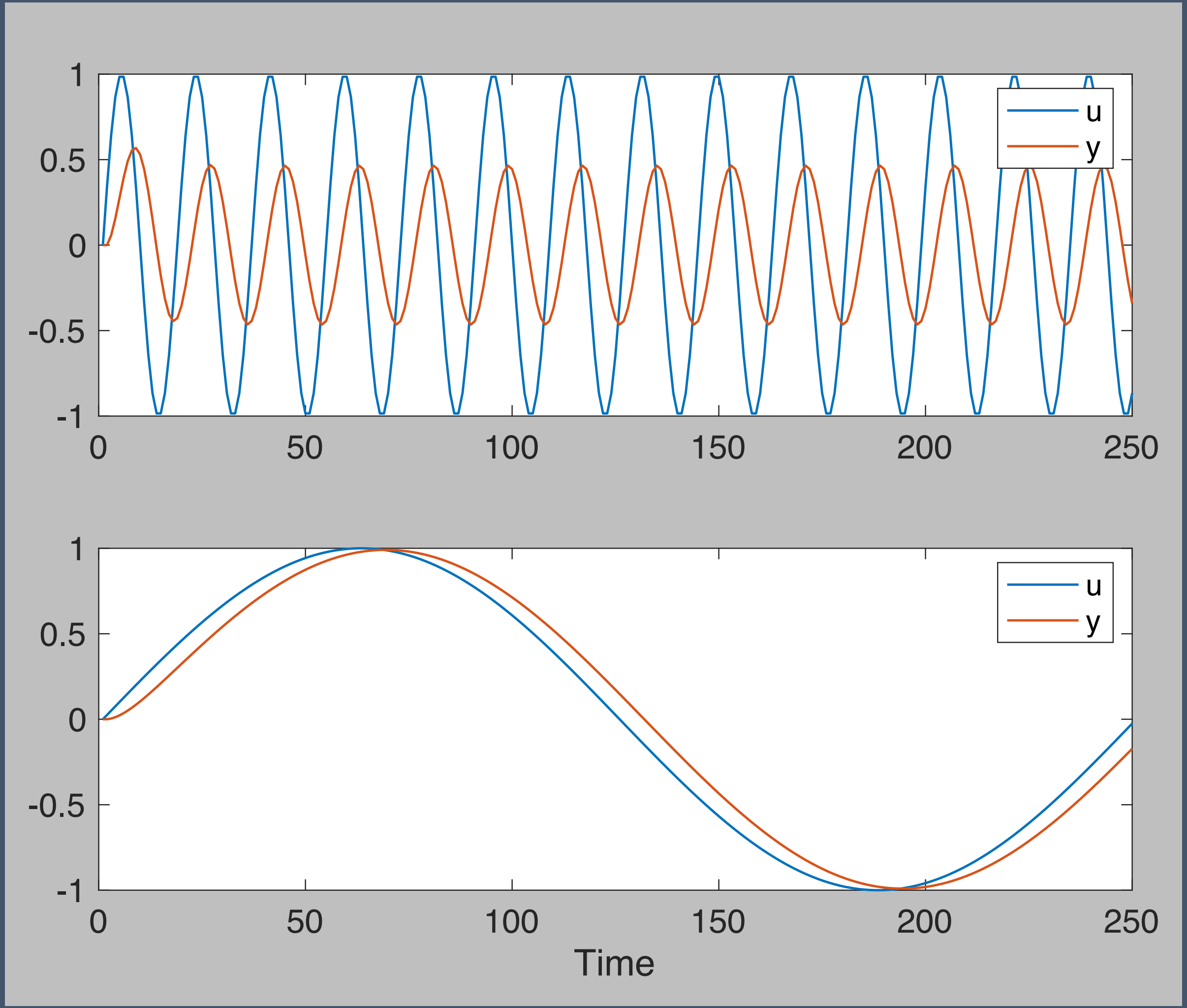




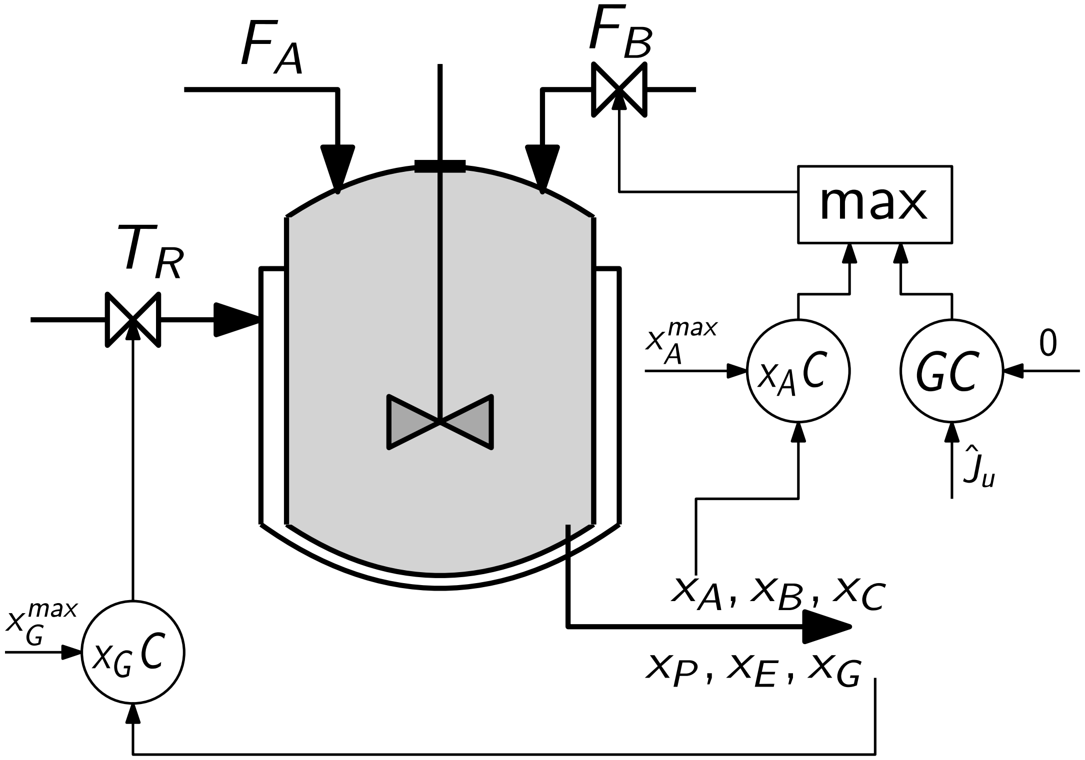
Special case of Fast Fourier Transform (FFT) - single frequency case

Classical Extremum Seeking Control

- Needs time scale separation to approximate plant as static map
- Typically 100 times slower than the system dynamics !
- Usually not a problem for electro-mechanical systems
  - System dynamics in ms → converges within a seconds
- Prohibitively slow convergence for chemical processes with slow dynamics
  - System dynamics in min → takes days to converge!



Williams-Otto Reactor



$$A + B \rightarrow C \qquad k_1 = 1.6599 \times 10^6 e^{-6666.7/T_r}$$

$$B + C \rightarrow P + E \qquad k_2 = 7.2177 \times 10^8 e^{-8333.3/T_r}$$

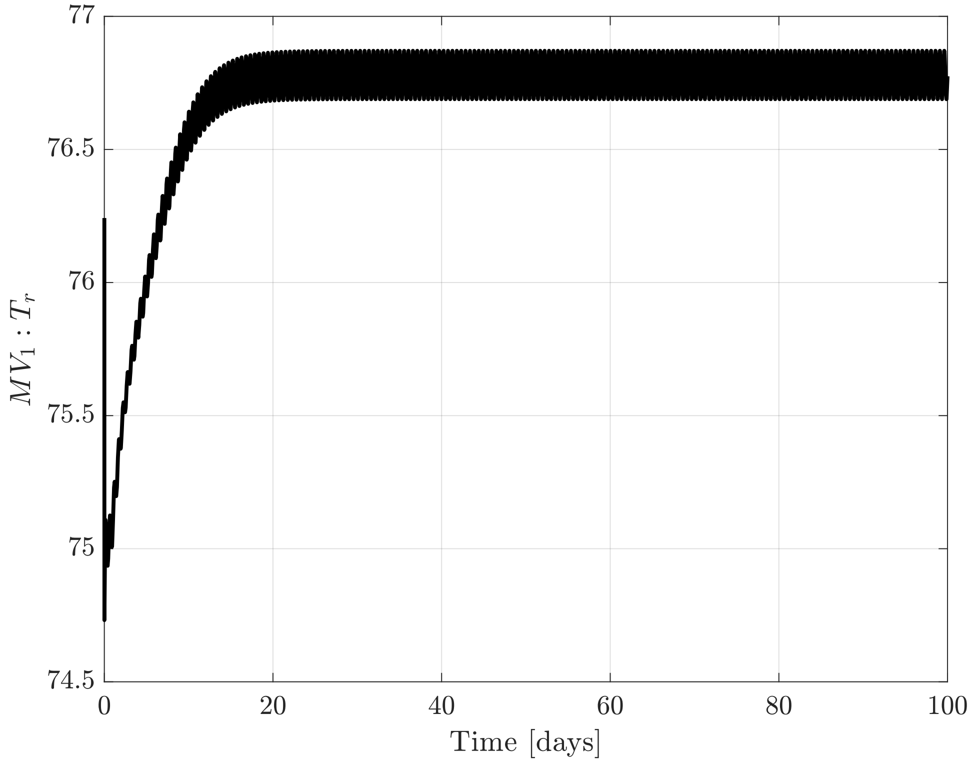
$$C + P \rightarrow G \qquad k_3 = 2.6745 \times 10^{12} e^{-11111/T_r}$$

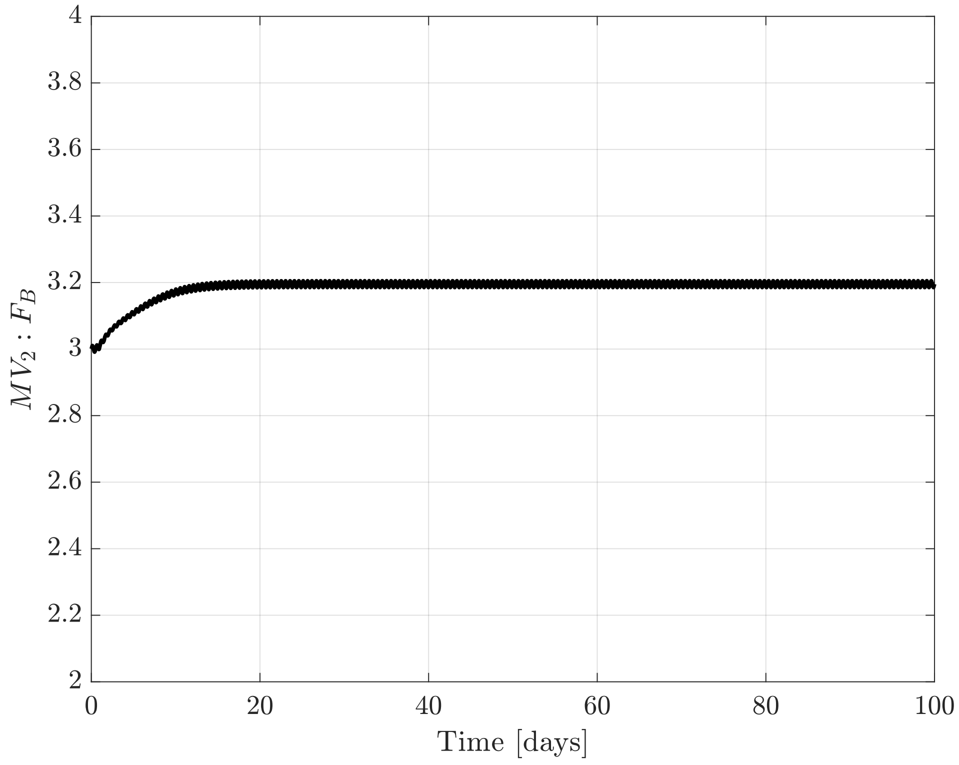
$$\min_{T_r, F_B} \; -1043.38 x_P (F_A + F_B) - 20.92 x_E (F_A + F_B) + 79.23 F_A + 118.34 F_B$$

$$\text{s.t.} \quad x_G \leq 0.08, \qquad x_A \leq 0.12$$

x_G always active

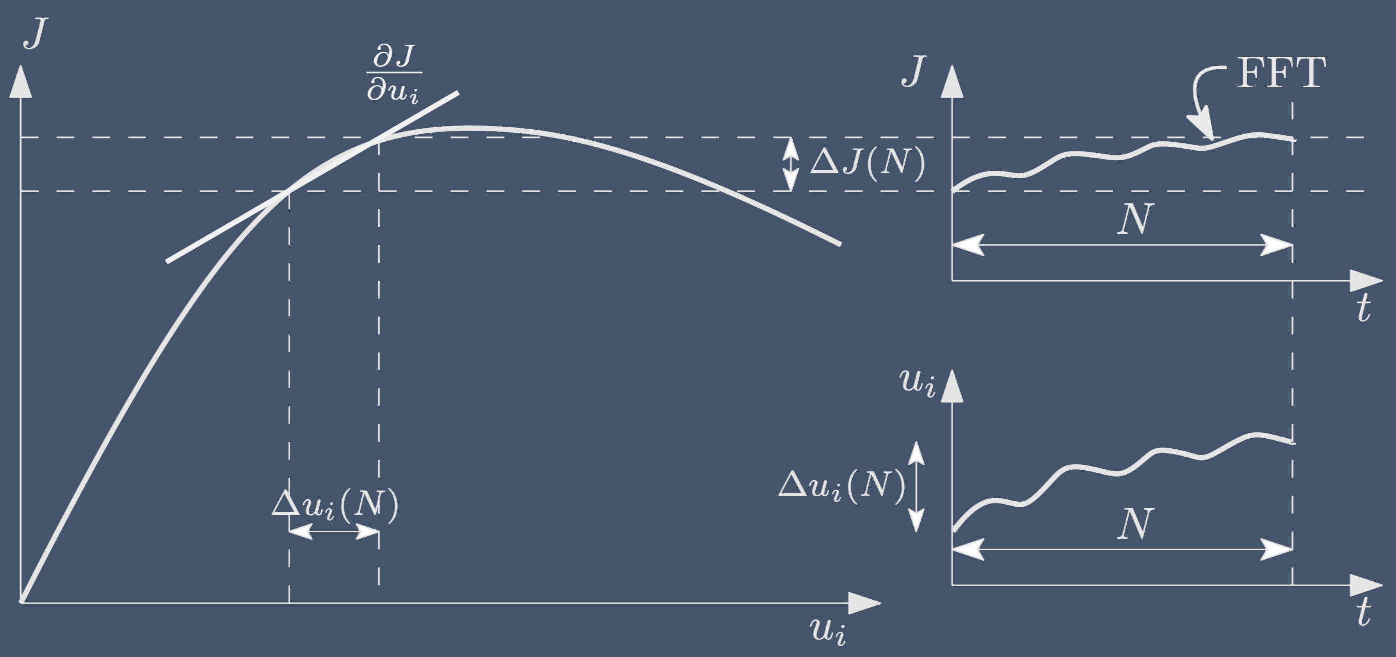
Classical Extremum seeking control



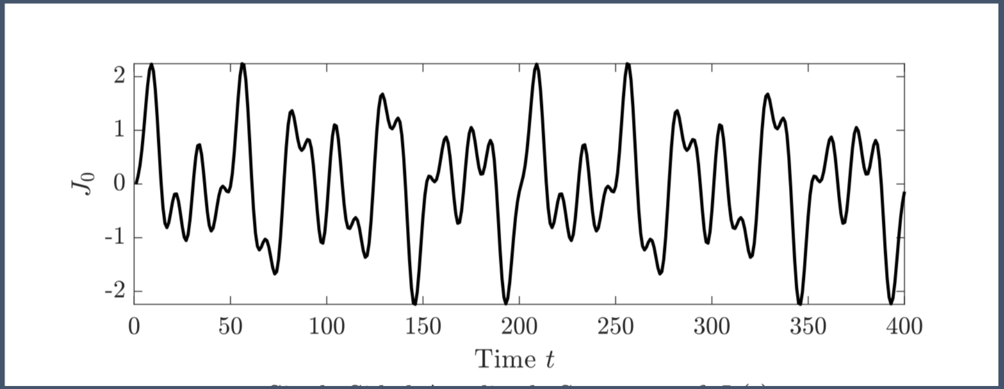


For MATLAB code, contact: dineshk@ntnu.no

Fast Fourier Transform (FFT)

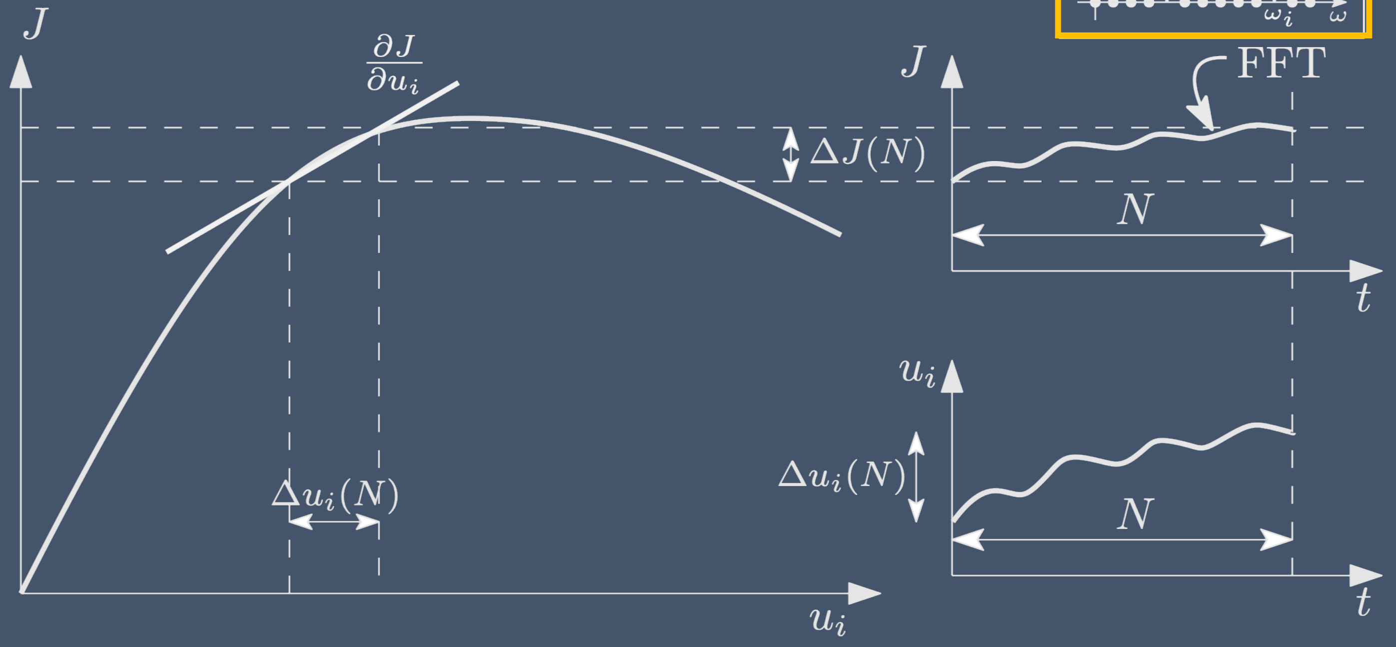


Multivariable Gradient estimation

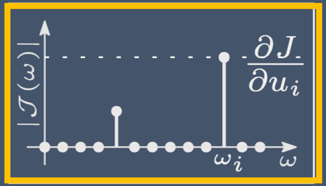


Krishnamoorthy & Skogestad (2020)

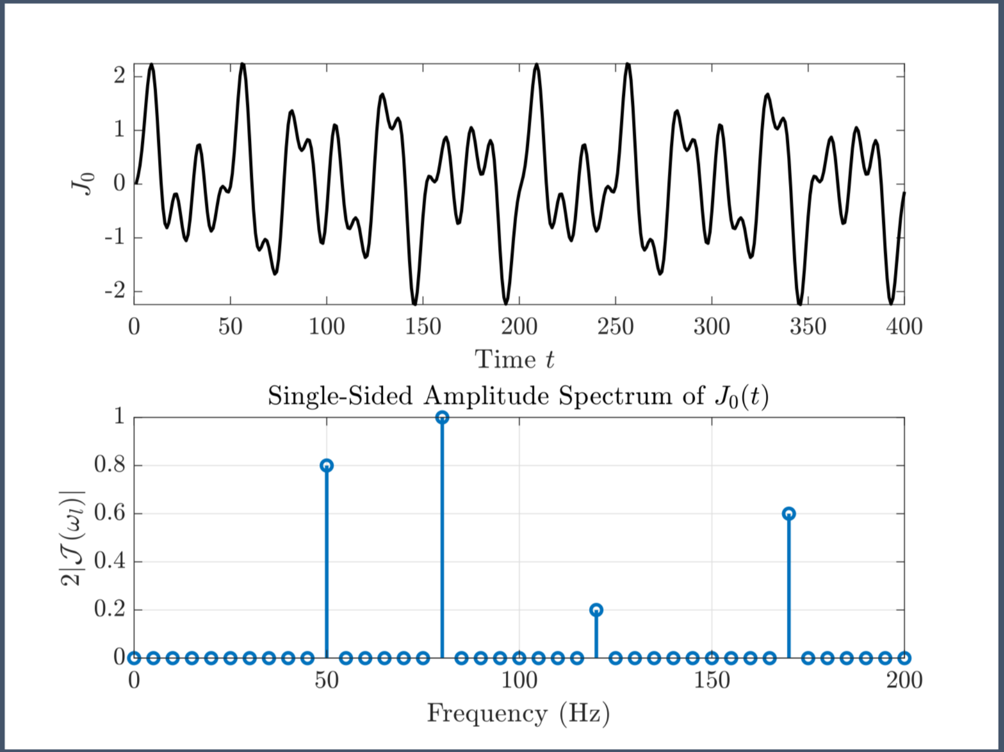
Fast Fourier Transform (FFT)



Amplitude spectrum



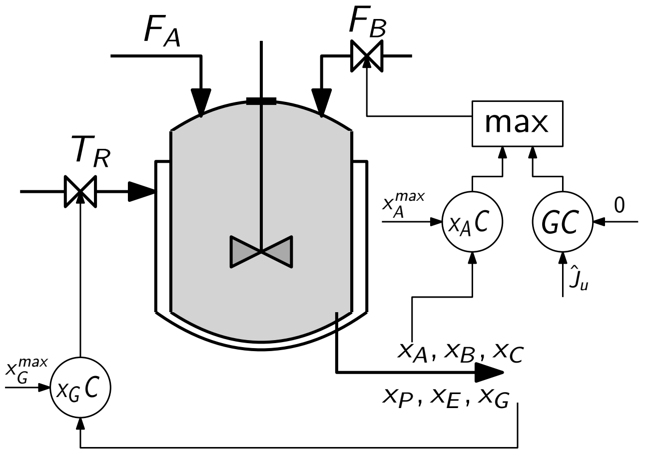
Multivariable Gradient estimation

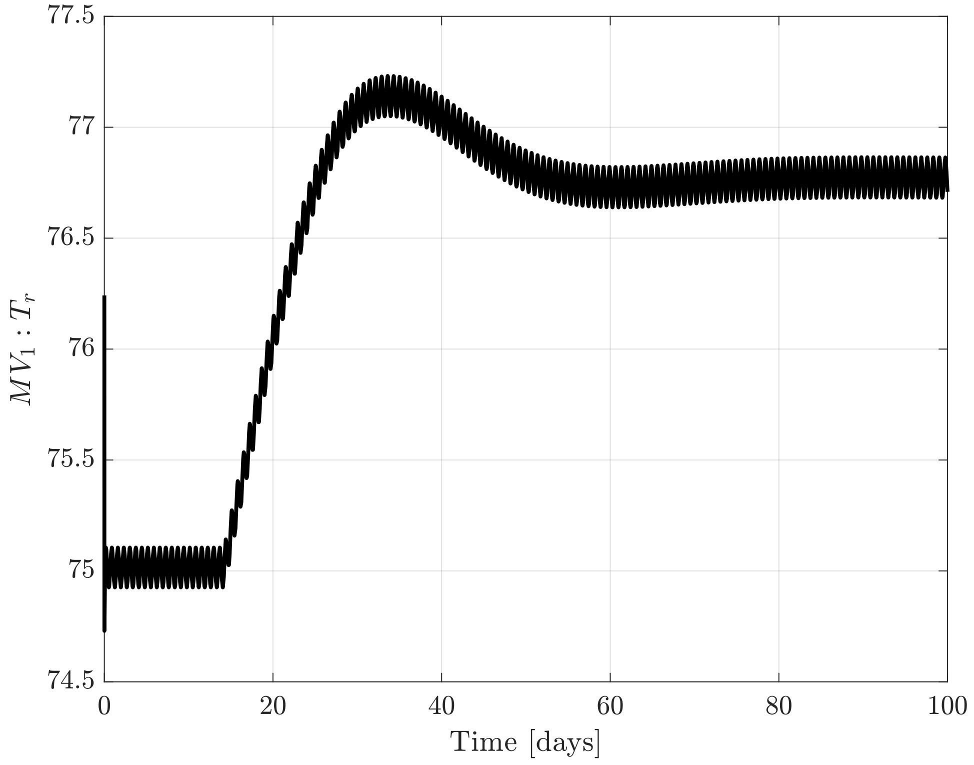


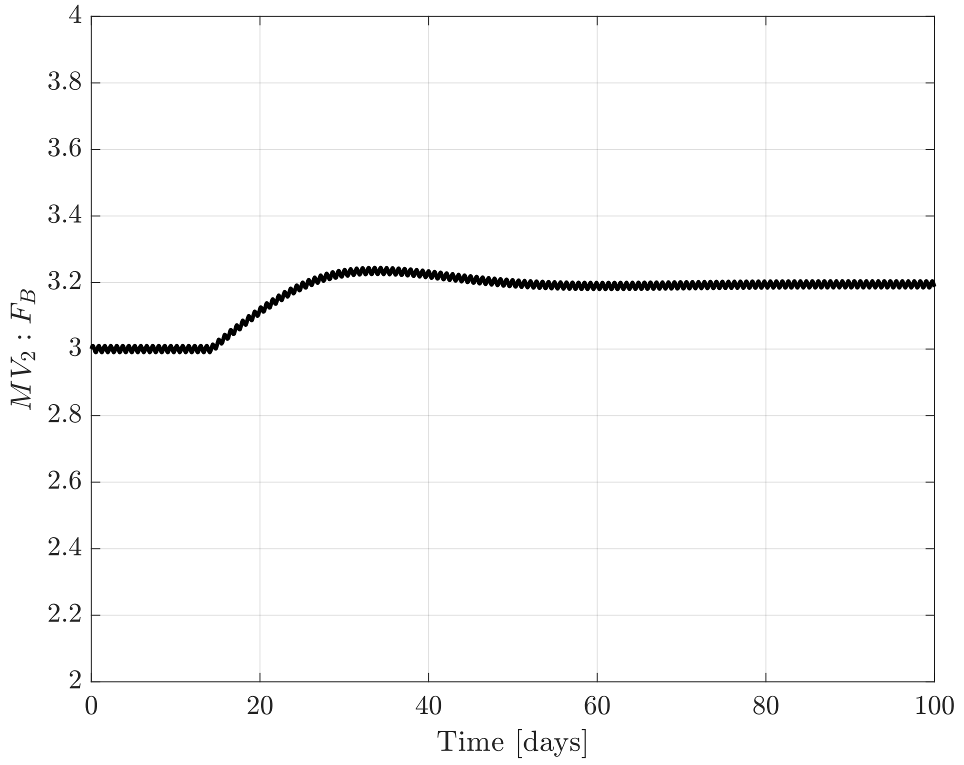
$$J_u = \frac{2}{a_i} |\mathcal{J}(\omega_i)| \operatorname{sgn}\left[\phi_J(\omega_i) \cdot \phi_{u_i}(\omega_i)\right], \ \forall i = 1, \ldots, n$$

Krishnamoorthy & Skogestad (2020)

Williams-Otto reactor example

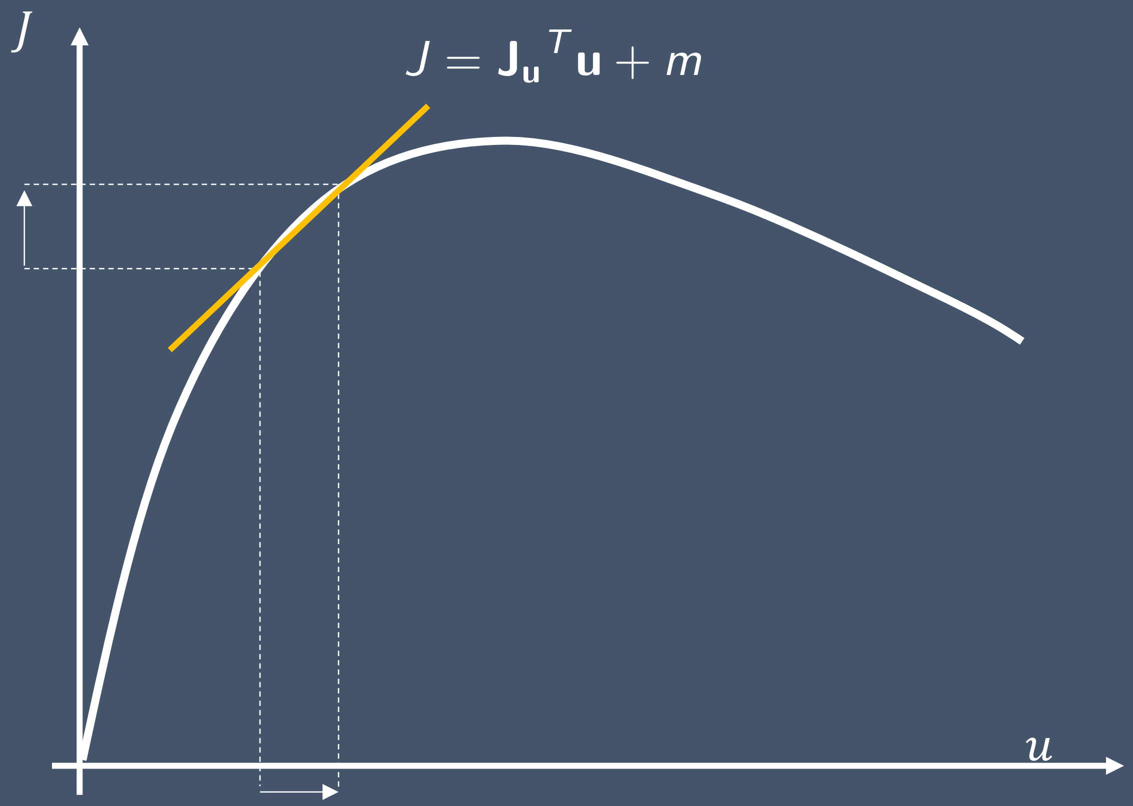






For MATLAB code, contact: dineshk@ntnu.no

Least square Extremum seeking control



- Over a moving window of past N data samples, fit

$$J = \mathbf{J_u}^T \mathbf{u} + m$$

- Using linear least squares estimation

$$\hat{\theta} = \arg\min_{\theta} \|\mathbf{Y} - \Phi^T \theta\|_2^2$$

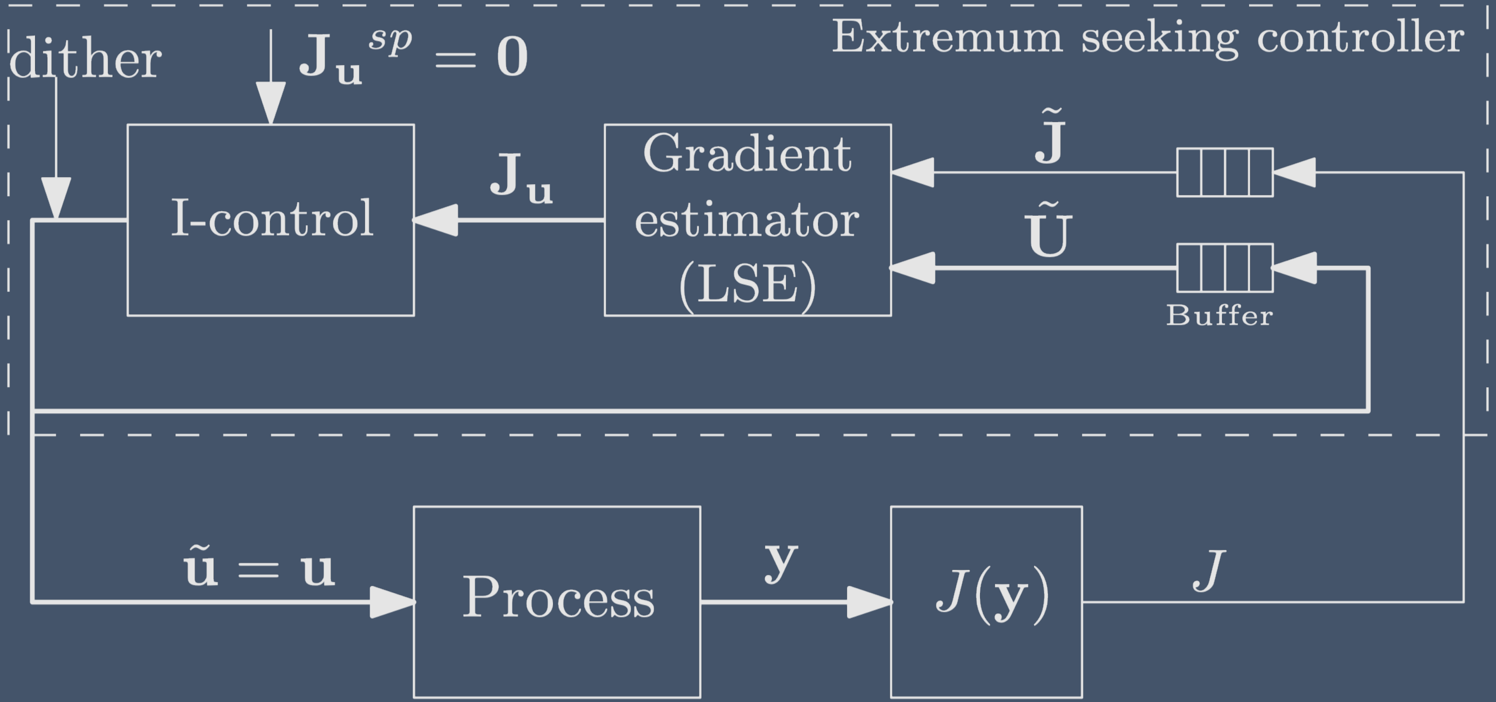
$$\mathbf{Y} = [J_k, J_{k-1}, \ldots, J_{k-N+1}]^T$$

$$\mathbf{U} = [\mathbf{u}_k, \ldots, \mathbf{u}_{k-N+1}]^T$$

$$\theta = [\mathbf{J_u}^T, m]^T, \; \Phi = [\mathbf{U}, 1]^T$$

Hunnekens et al. (2011)

Least square Extremum seeking control



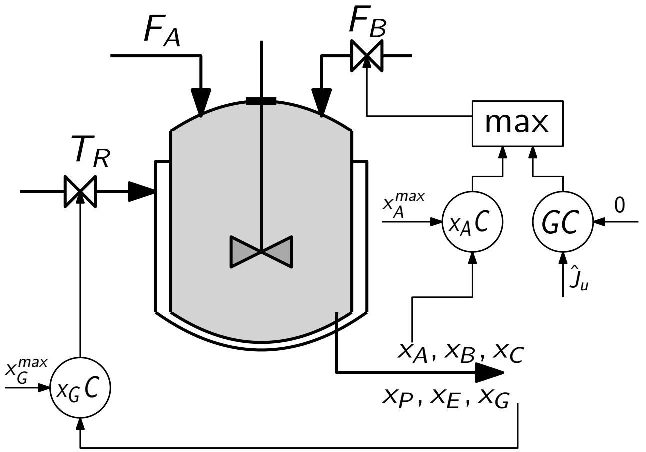
Fit a linear model

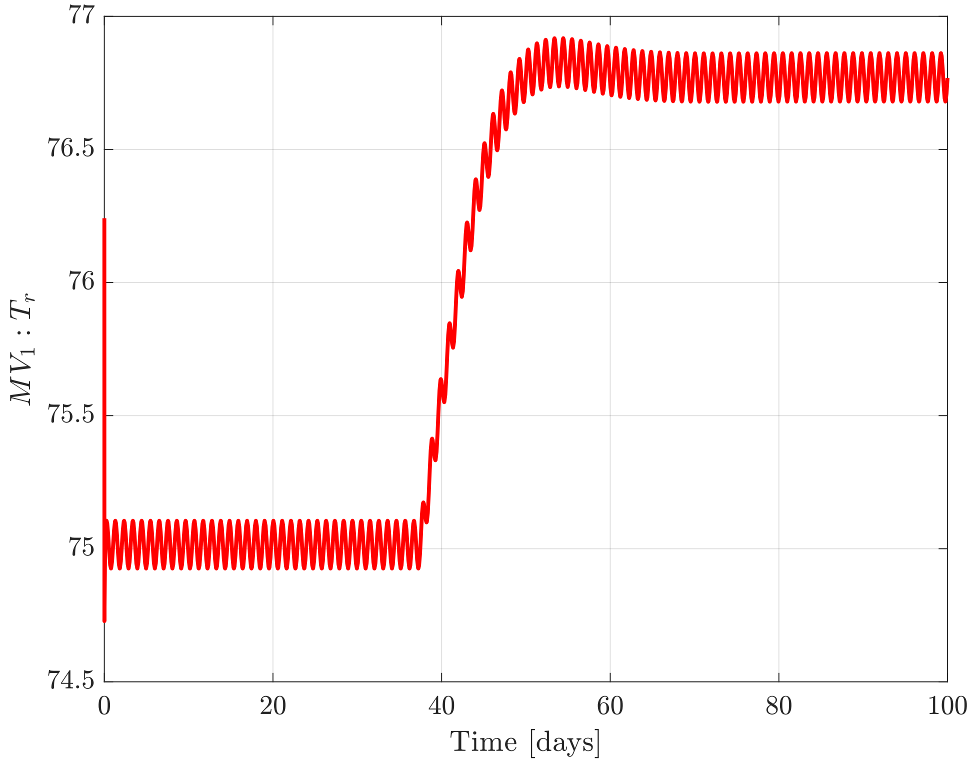
$$J = \mathbf{J_u}^T \mathbf{u} + m$$

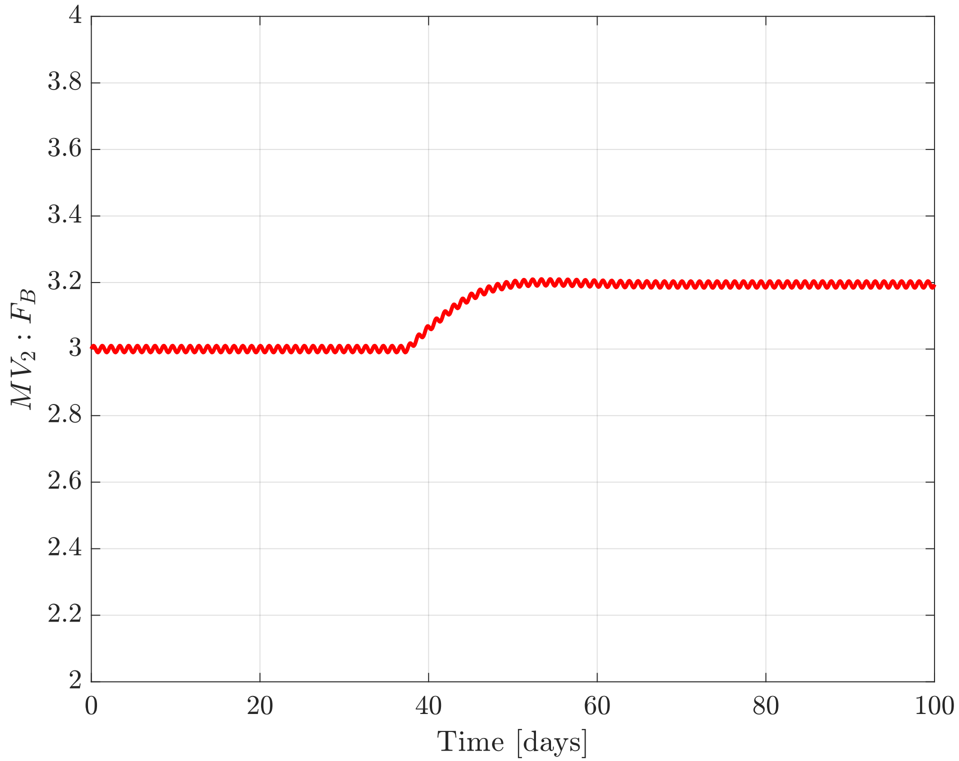
Using least squares estimation

Hunnekens et al. (2011)

Williams-Otto reactor example







For MATLAB code, contact: dineshk@ntnu.no

Issues with Extremum seeking

- **Need Cost measurement**
  - Often cost function is a sum of several terms
  - All terms must be measured
  - Estimation of cost requires model (dependency on model – no longer model free)

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

- **Slow convergence**
  - Dynamic plant assumed to be a static map
  - Three timescale separation: dynamics – perturbation – convergence
- **Constant probing** of the system may be undesirable
- Unknown and abrupt disturbances affects gradient estimation

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

Extremum seeking using transient measurements

- Repeatedly Identify local linear “dynamic” black-box model.
  - e.g. ARX, ARMAX,…

$$\left.\begin{aligned} \dot{x} &= Ax + Bu \\ J &= Cx + Du \end{aligned}\right\} \text{Black-box linear dynamic model}$$

- Estimate gradient from black-box linear dynamic model

$$J = (\underbrace{-CA^{-1}B + D}_{J_u})u$$

Bamberger & Isermann, Automatica (1978)
Garcia & Morari, AIChE J. (1981)

Recap: Gradient Estimation methods

Model-free

- Finite Difference
- Dither demodulation (Draper & Li, 1951)
- Linear least squares (Hunnekens et al., 2014)
- Dynamic system identification (Bamberger & Isermann, 1978)
- Fast Fourier Transform (Krishnamoorthy & Skogestad)
- Multiple units (Srinivasan, 2007)
- Gausian process regression (Ferriera et al.,2018/ Matias & Jäschke, 2019)
- Fitted surfaces (Gao & Engell, 2005)
- Kalman Filter (Gelbert et al., 2012)

Model-based

- Analytical gradient from updated model
- Nullspace method (Alstad & Skogetad, 2007)
- Feedback RTO (Krishnamoorthy et al., 2019)
- Neighbouring extremals (Gros et al., 2009)

Recap: What to control ?

Control (in this order) :

1. Active constraints (no need for models)
2. Self-optimizing variables (for remaining unconstrained MVs)
   a. Linear measurement combination (model used offline)
   b. Linear gradient combination (model-based or model-free*)

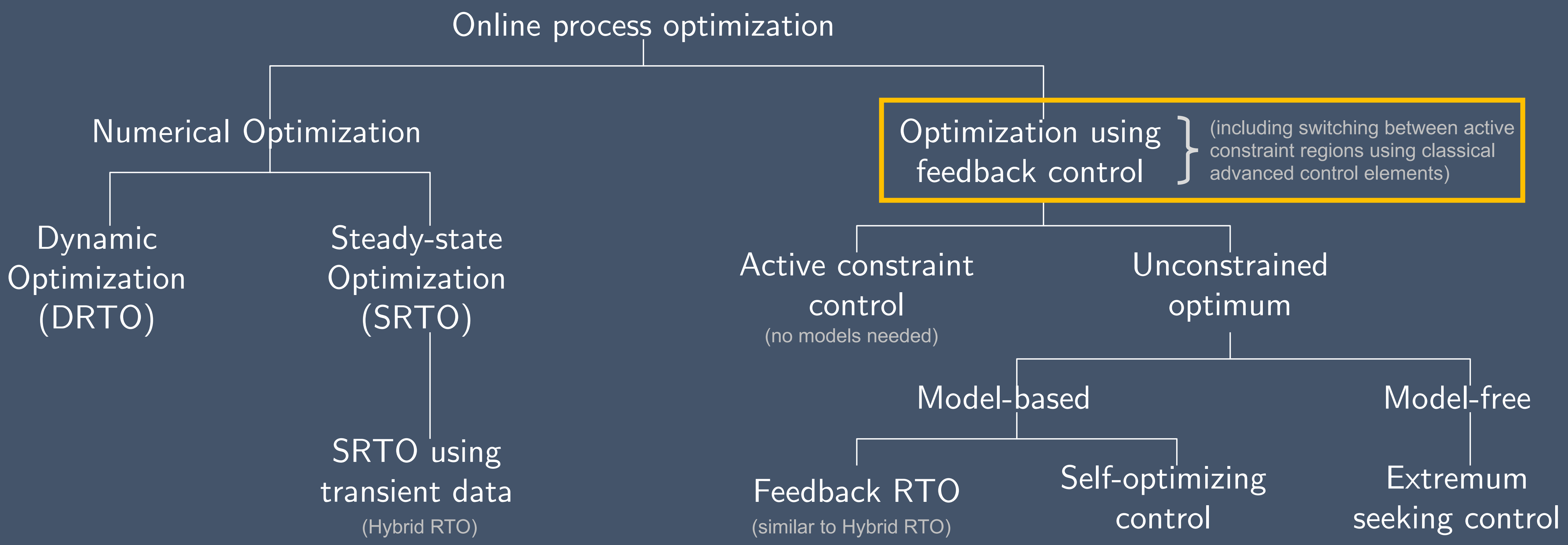
*model-free – if cost and constraint measured!

Model-free approaches for constrained optimum

8. Optimal operation using classical advanced control elements

Switching between active constraint regions

Course Roadmap



Do we always need a model to optimize?

- We often know or can guess the **active constraints**
  - Example: Drive from A → B in shortest time

- CV = speed → speed limit 50km/h
- MV = gas pedal

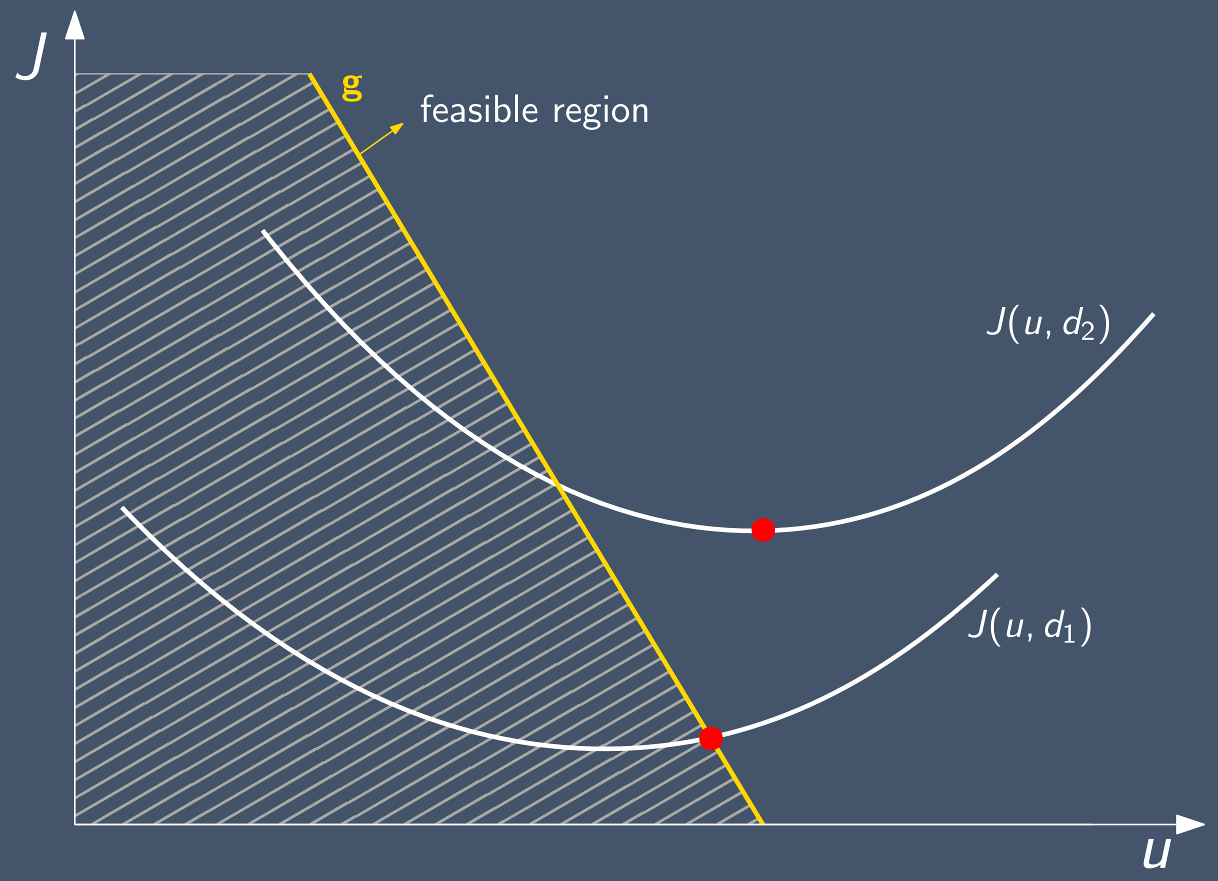
Do we always need a model to optimize?

- We often know or can guess the **active constraints**
  - Example: Drive from A → B in shortest time

    - MV = gas pedal → max

Optimal operation requires changing between active constraint regions !

Active constraint regions



How many active constraint regions?

$$\min_{\mathbf{u}} \quad J(\mathbf{u}, \mathbf{d})$$

s.t.

$$\mathbf{g}(\mathbf{u}, \mathbf{d}) \leq 0$$

$$\mathbf{u} \in \mathbb{R}^{n_u} \qquad \mathbf{g} \in \mathbb{R}^{n_g}$$

- Maximum: 2^{n_g}
- But there are usually fewer in practice
  - Some constraint combinations are infeasible, e.g. $n_a > n_u$
  - Some constraints are always active, e.g. most valuable product, material balance in optimal flow split
  - Some constraint combinations are never active, e.g. max and min constraint on the same variable

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)

Changing active contraints

Procedure for maintaining optimal operation when changing between active constraint regions

- **Step 1:** Define all the constraints
- **Step 2:** Identify relevant active constraint combinations and switches
- **Step 3:** Propose a control structure for the nominal operating point.
- **Step 4:** Propose switching schemes (see next slide)
- **Step 5:** Design controllers for all cases (active constraint combinations)

Switching between active constraints

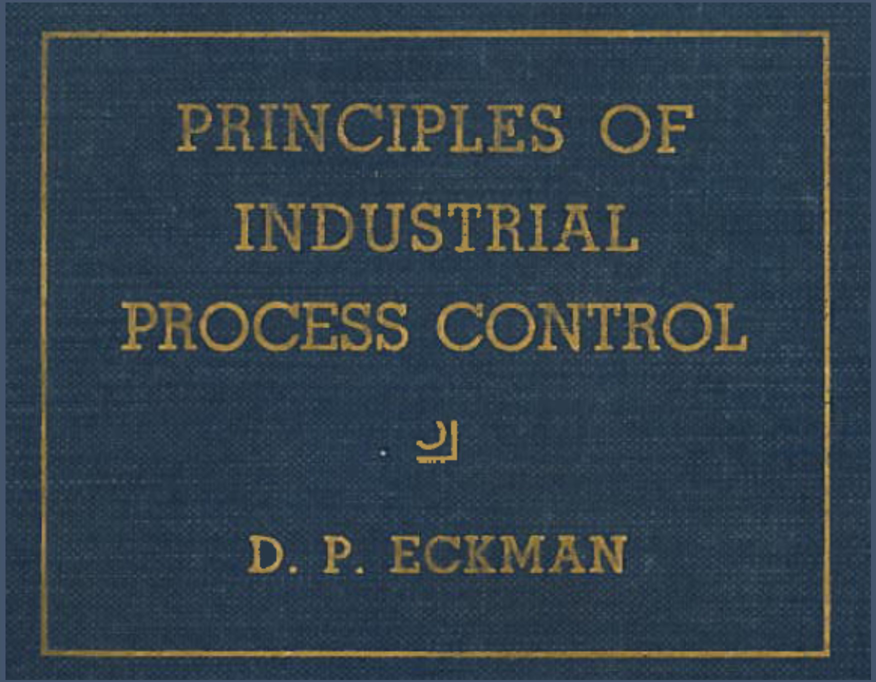
1. Output to Output (CV - CV) switching (SIMO)
   - Selector

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)

- Krishnamoorthy, D., and Skogestad, S., 2019. Online process optimization with changes in active constraint sets using simple feedback control structures, *Ind. Eng. Chem. Res.* Vol. 58 (30), pp. 13555-13567
- A. Reyes-Lúa, C. Zotica, and S. Skogestad. Optimal operation with changing active constraint regions using classical advanced control. IFAC ADCHEM, Shenyang, China, 2018.

Split range control: Donald P. 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

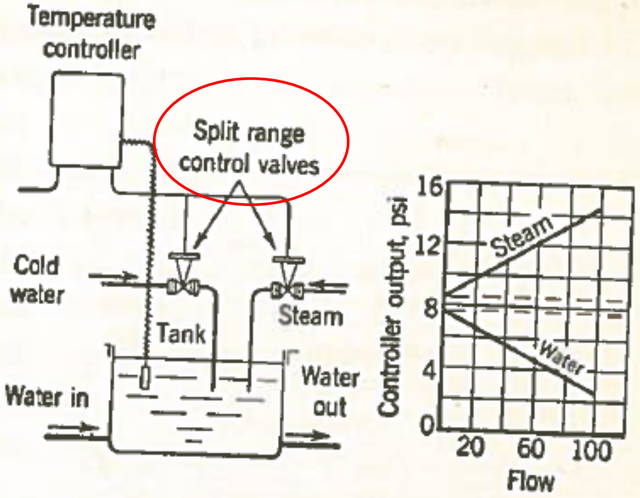
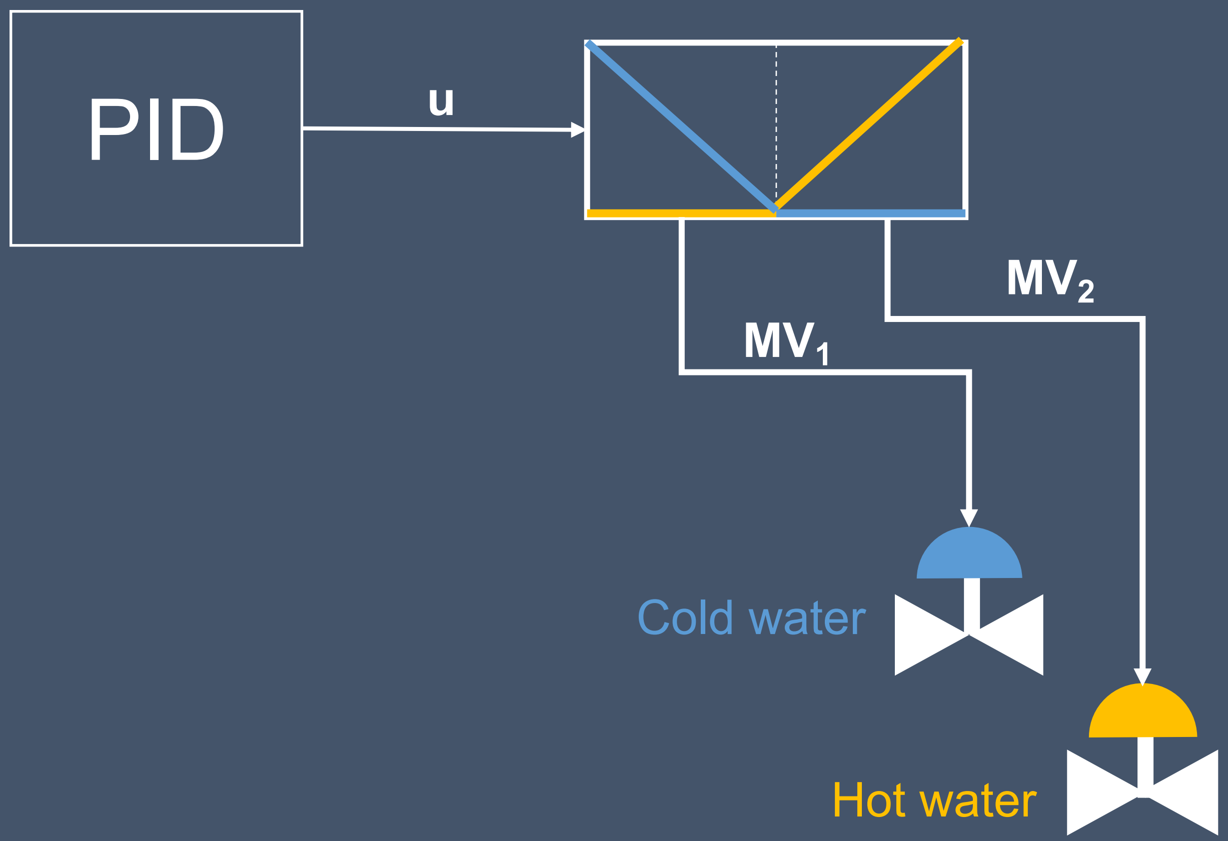


FIG. 10–12. Dual-Agent Control System for Adjusting Heating and Cooling of Bath.

Split-range control



MV-MV switching

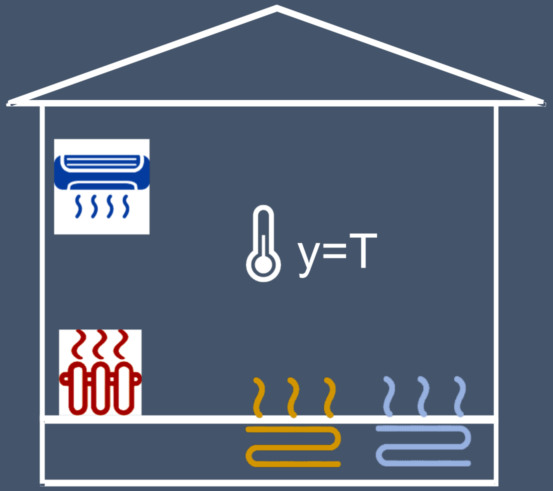
- When MV_1 saturates
- MV_2 takes over

Example: Temperature control, using hot and cold water!

Split-range control (SRC)

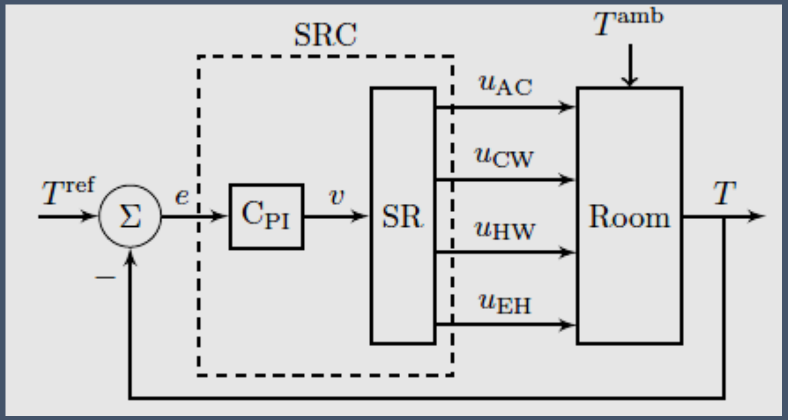
One CV (y). Two or more MVs (u1,u2)

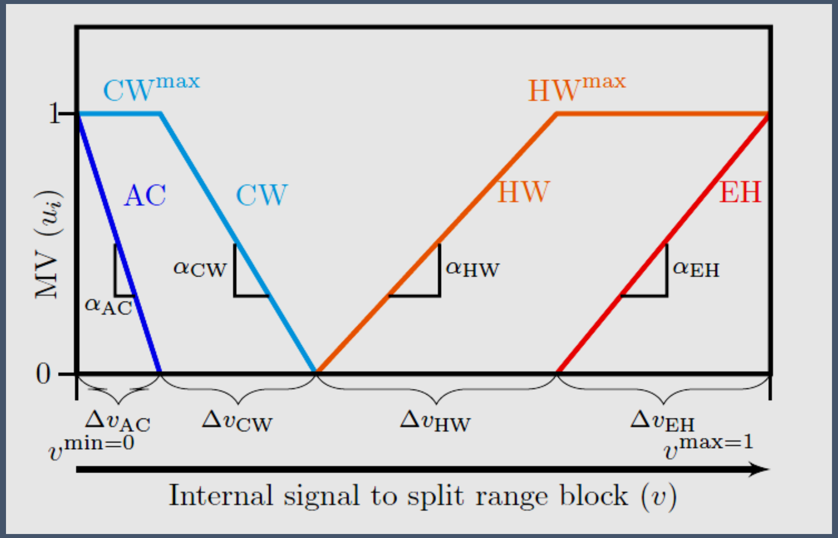
Example: Room heating with 4 MVs



MVs:

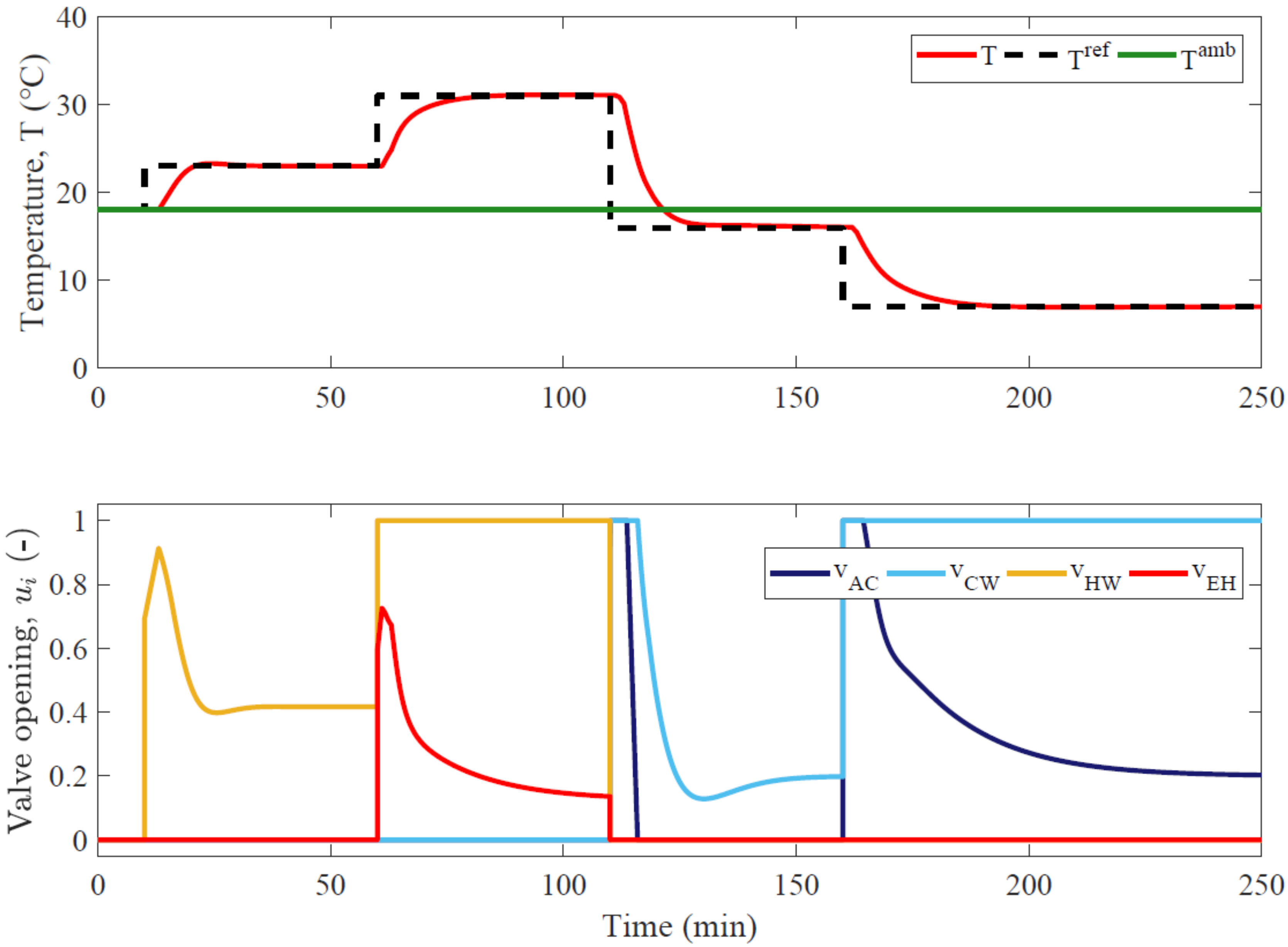
1. AC (expensive cooling)
2. CW (cooling water; cheap)
3. HW (hot water, quite cheap)
4. Electric heat, EH (expensive)



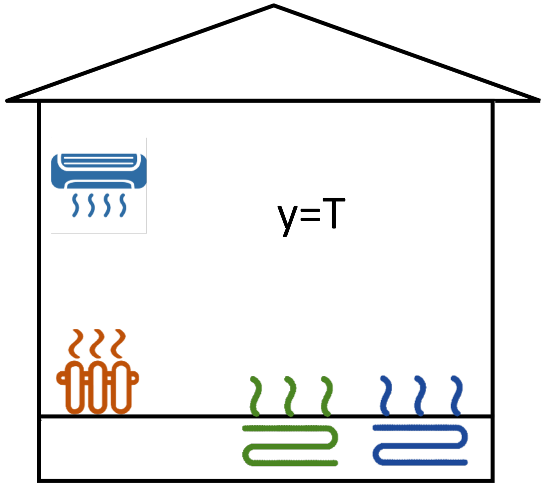


Reyes Lua & Skogestad, Processes (2020)

Simulation PI-control: Setpoint changes temperature



Example: Room heating with 4 MVs



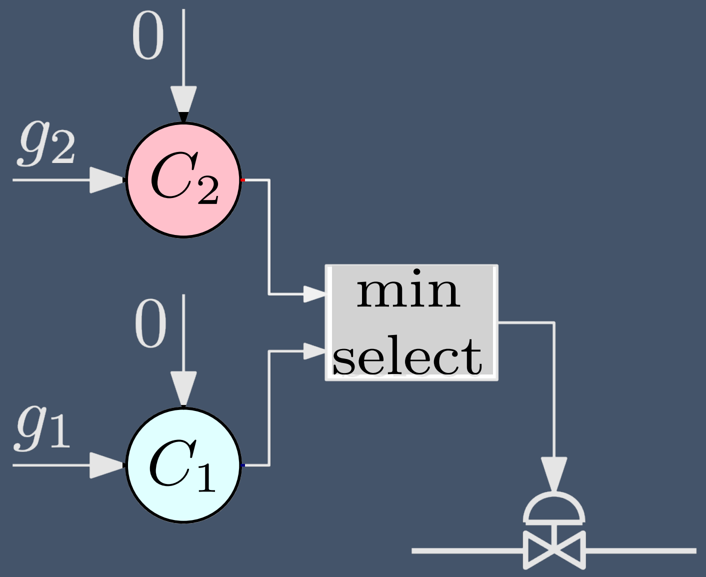
MVs:
1. AC (expensive cooling)
2. CW (cooling water; cheap)
3. HW (hot water, quite cheap)
4. Electric heat, EH (expensive)

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)

CV-CV switching: Selector block

- Used when one input is used to switch between controlling several outputs.
- Each output has a separate controller and the selectors chooses which output to use.

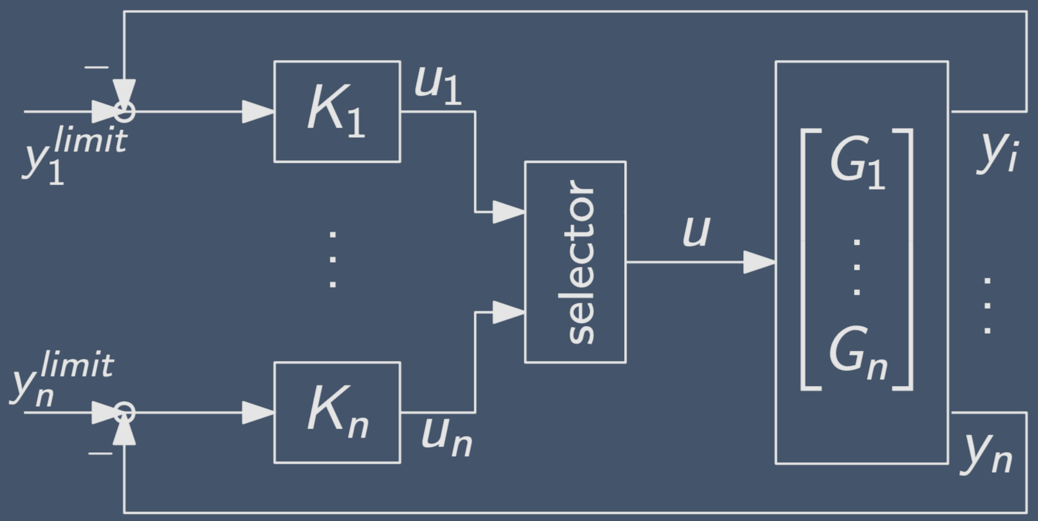


CV-CV switching: Selector block

For each MV

- at most 1 CV y_0 with setpoint control that may be given up
- n number of CV constraints y_i that may be optimally active

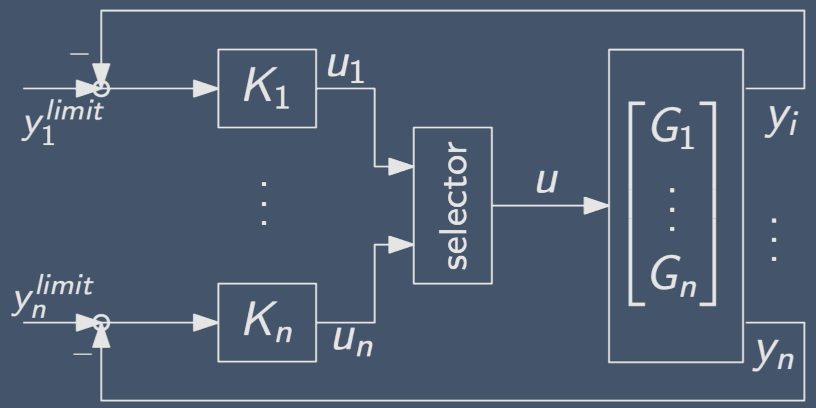
$$u = \min_{i\in[0,n]}(u_i) \quad \text{or} \quad u = \max_{i\in[0.n]}(u_i)$$



Min or Max selector?

Introduce logic variable y_i^{lim} for CV constraints

$$y_i^{lim} = \begin{cases} 1, & \text{for max-constraint} \\ -1, & \text{for min-constraint} \end{cases}$$



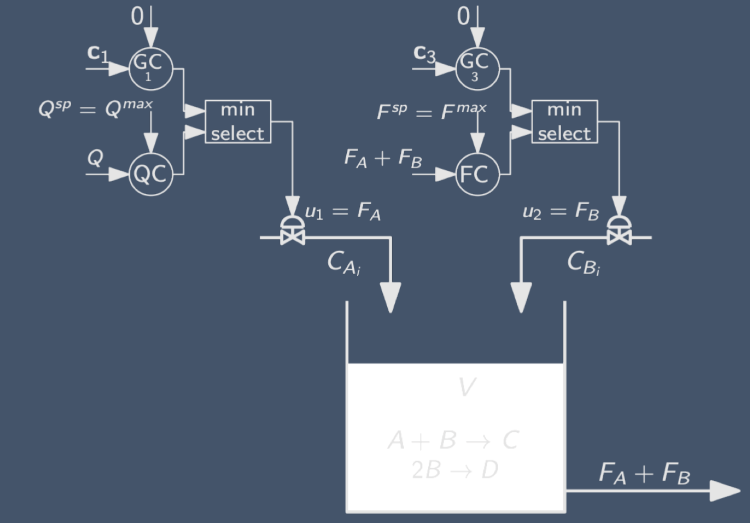
CV-CV switching is feasible only if

$$sgn(G_i)sgn(y_i^{lim}) = sgn(G_j)sgn(y_j^{lim}) \quad \forall i, j \in \{1, \dots, n\}$$

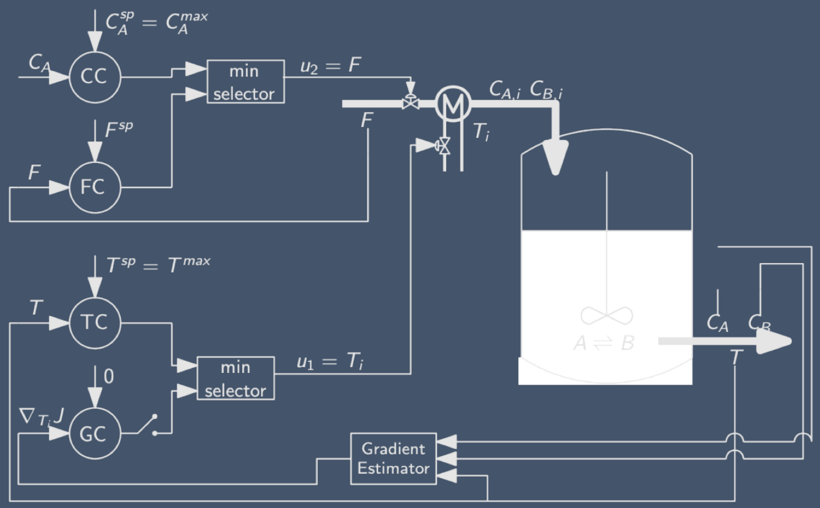
and,

$$sgn(G_i)sgn(y_i^{lim}) = 1 \Rightarrow \text{ min selector}$$
$$sgn(G_i)sgn(y_i^{lim}) = -1 \Rightarrow \text{ max selector}$$

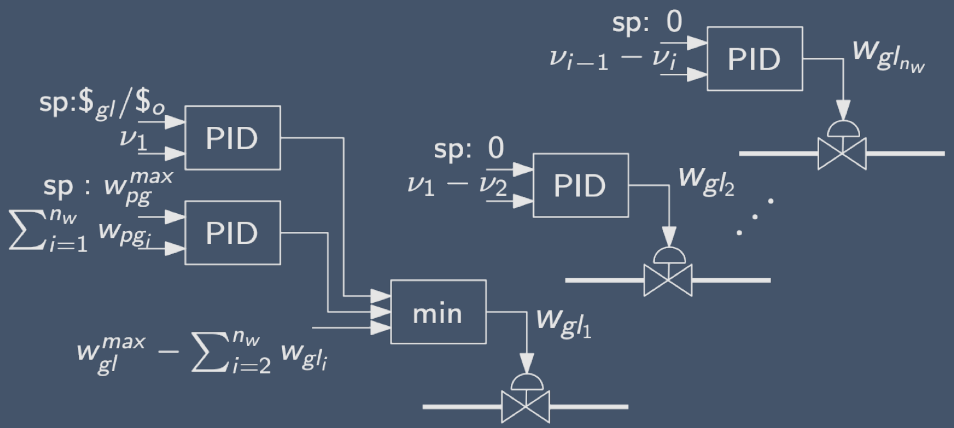
Isothermal CSTR



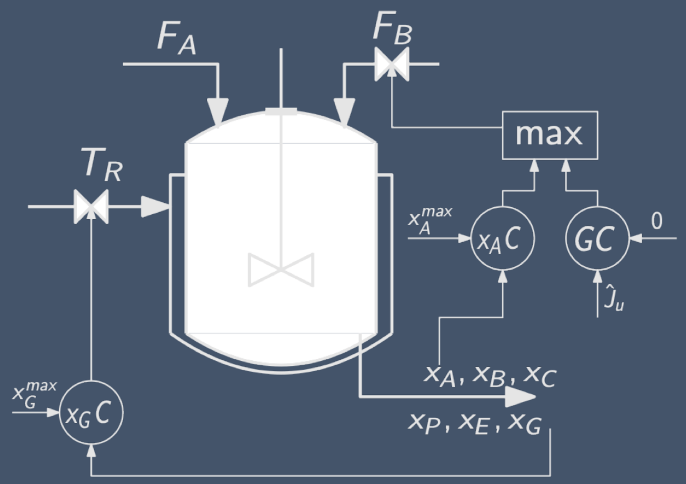
Exothermic Reactor



Gas-lift Optimization



William-Otto reactor



- Krishnamoorthy & Skogestad, I&ECR (2019)
- Krishnamoorthy et al., Control Engineering practice (2019)
- Krishnamoorthy & Skogestad, CACE (2020)

Case example: Exothermic Reactor

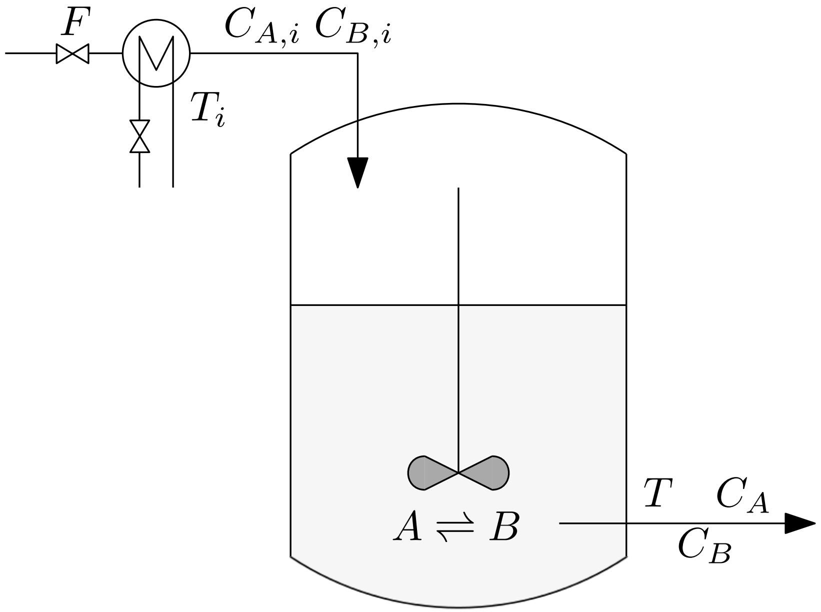
$$\min_{T_i, F} \quad -F - 2.009 C_B + (1.657 \times 10^{-3} T_i)^2$$

s.t.

$$\mathbf{g}_1 : F/F^{max} - 1 \leq 0$$

$$\mathbf{g}_2 : T/T^{max} - 1 \leq 0$$

$$\mathbf{g}_3 : C_A/C_A^{max} - 1 \leq 0$$



$$MV_1 := T_i$$

$$MV_2 := F$$

Active Constraints

$$\min_{T_i, F} \quad -F - 2.009C_B + (1.657 \times 10^{-3} T_i)^2$$

s.t.

$$\mathbf{g}_1 : F/F^{max} - 1 \leq 0$$
$$\mathbf{g}_2 : T/T^{max} - 1 \leq 0$$
$$\mathbf{g}_3 : C_A/C_A^{max} - 1 \leq 0$$

- Potential active constraint regions ($2^3 = 8$)
  1. Fully unconstrained (never)
  2. Only g_1 active (R-I)
  3. Only g_2 active (never)
  4. Only g_3 active (never)
  5. g_1 and g_2 active (R-II)
  6. g_2 and g_3 active (R-III)
  7. g_1 and g_3 active (unlikely)
  8. g_1, g_2 and g_3 active (infeasible)

Active Constraints

$$\min_{T_i, F} \quad -F - 2.009C_B + (1.657 \times 10^{-3} T_i)^2$$

s.t.

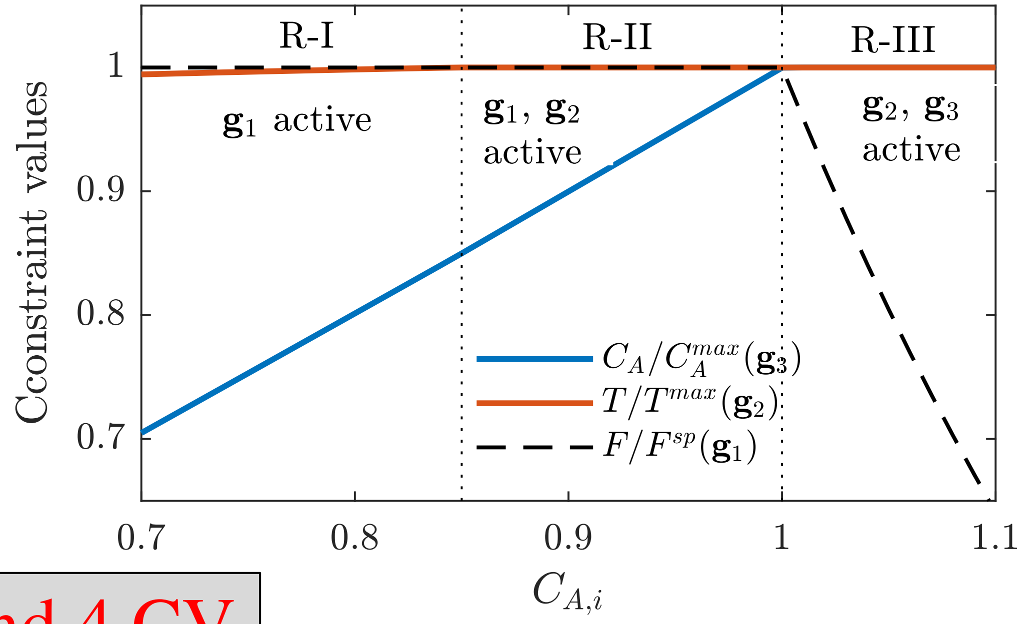
$$\mathbf{g}_1 : F/F^{max} - 1 \leq 0$$

$$\mathbf{g}_2 : T/T^{max} - 1 \leq 0$$

$$\mathbf{g}_3 : C_A/C_A^{max} - 1 \leq 0$$

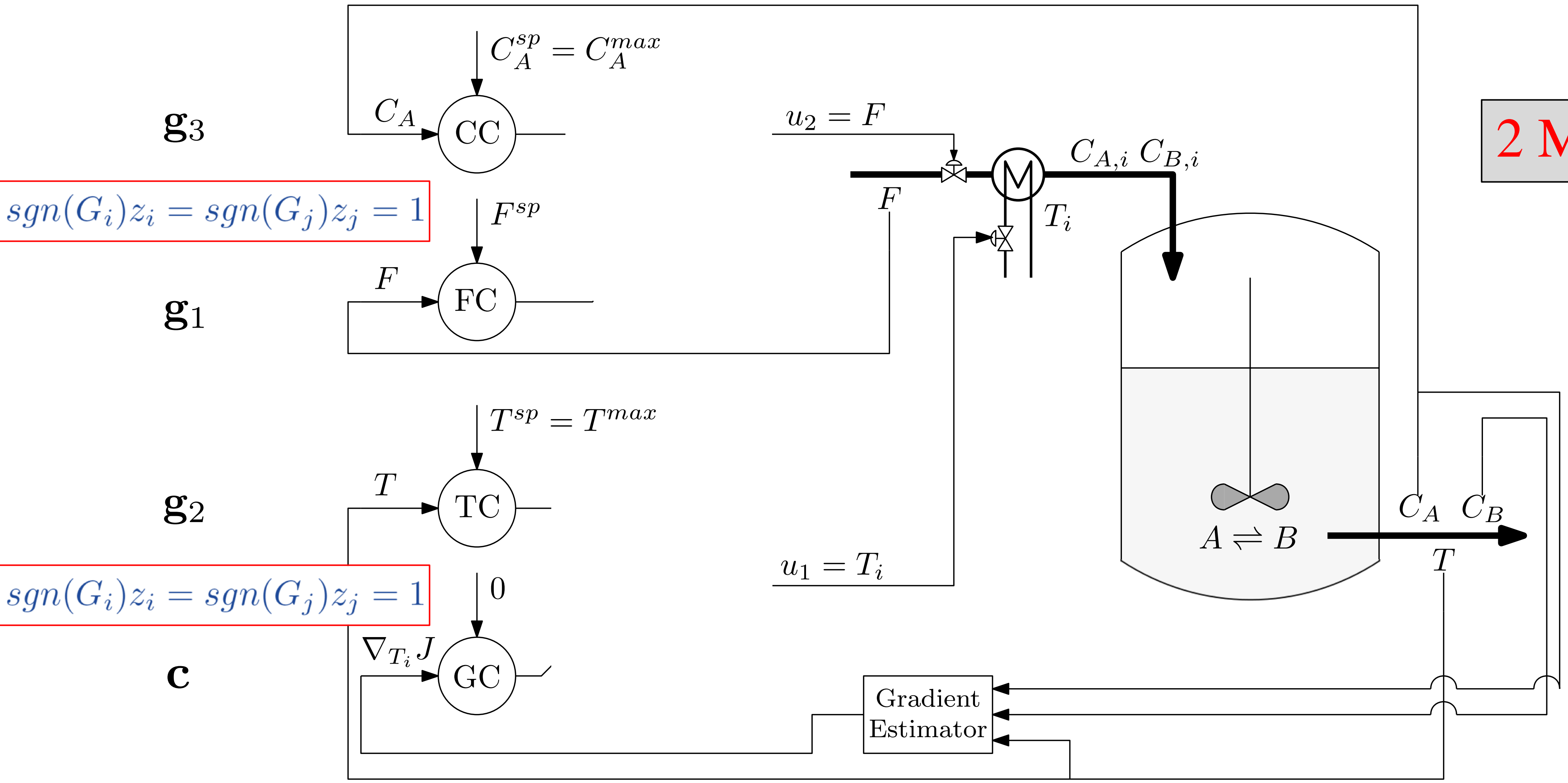
- Potential active constraint regions ($2^3 = 8$)

  1. ~~Fully unconstrained (never)~~
  2. Only g_1 active (R-I)
  3. ~~Only g_2 active (never)~~
  4. ~~Only g_3 active (never)~~
  5. g_1 and g_2 active (R-II)
  6. g_2 and g_3 active (R-III)
  7. ~~g_1 and g_3 active (unlikely)~~
  8. ~~g_1, g_2 and g_3 active (infeasible)~~



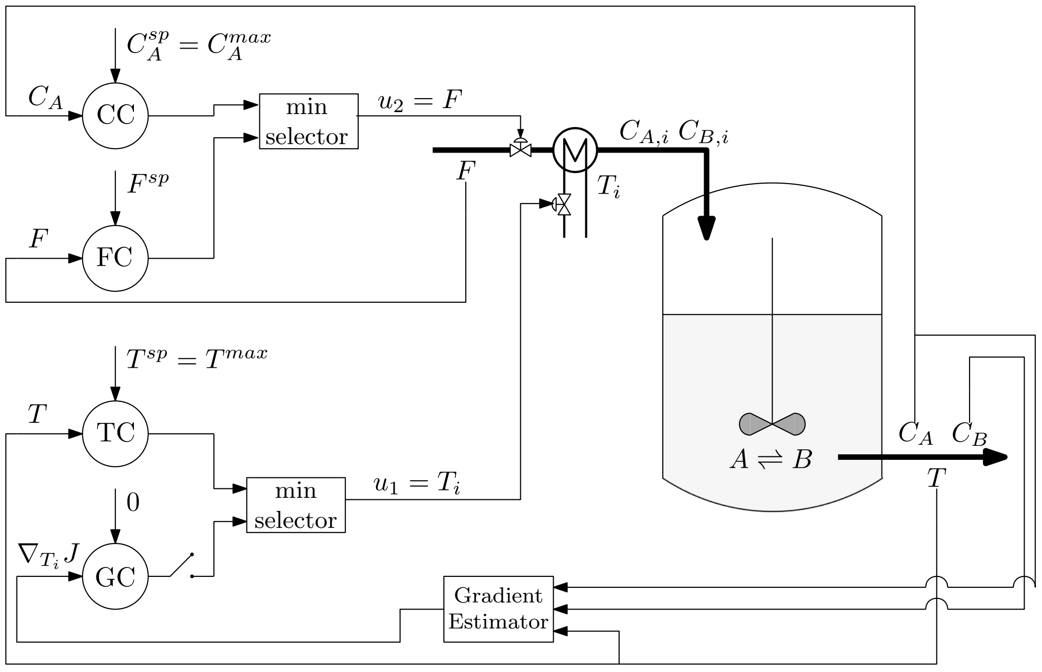
2 MV and 4 CV

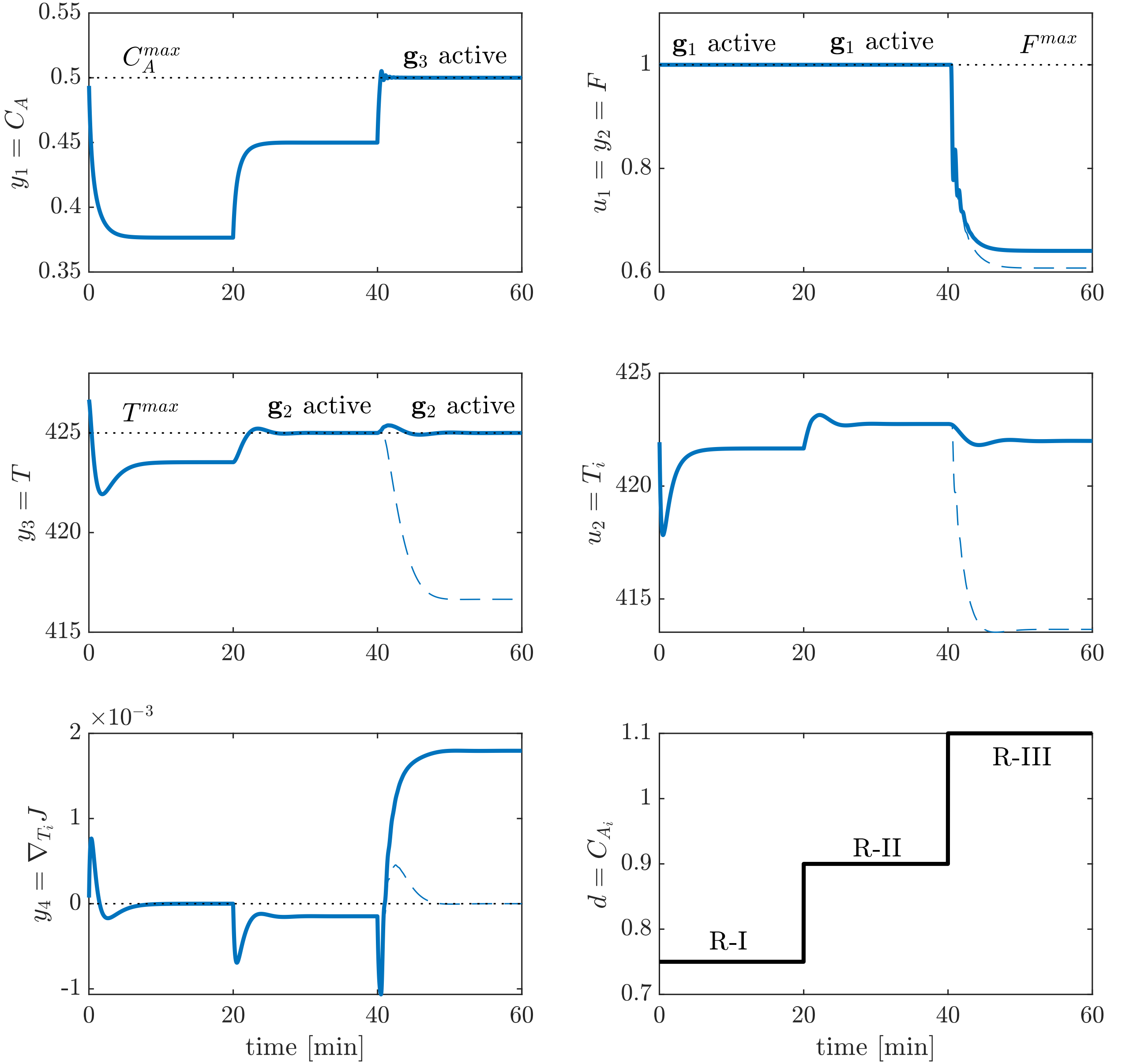
Proposed control structure design



2 MV and 4 CV

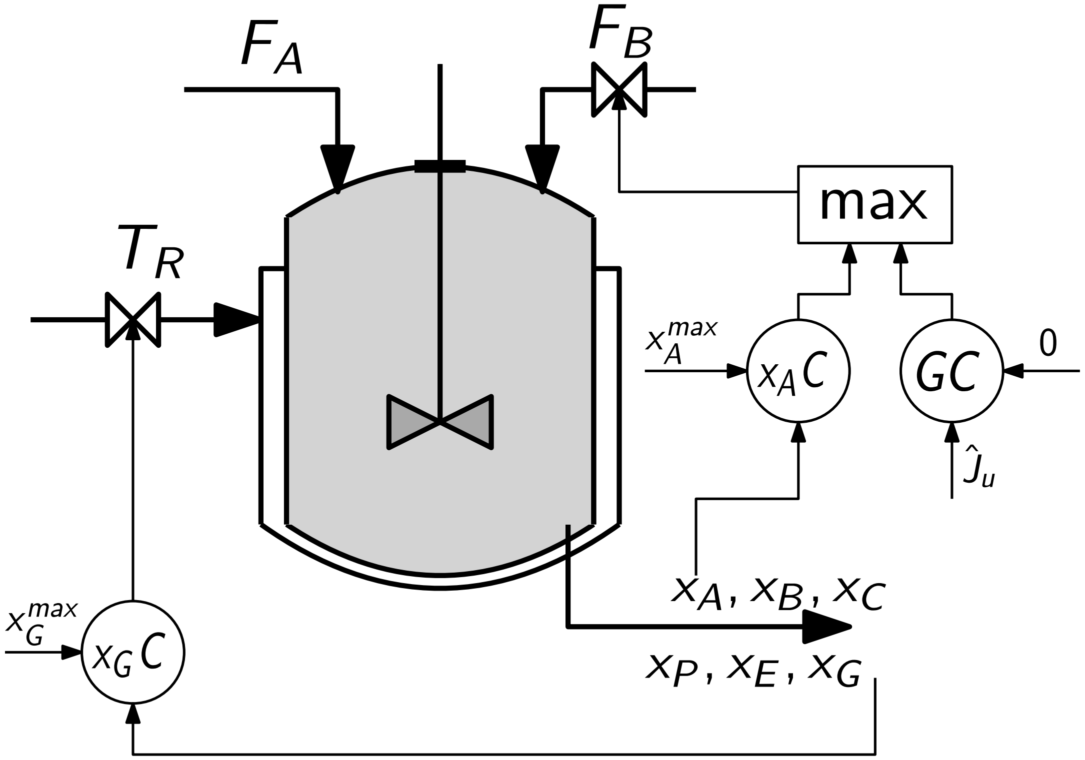
Simulation results





For MATLAB code, contact: dineshk@ntnu.no

Williams-Otto Reactor



$$A + B \rightarrow C \qquad k_1 = 1.6599 \times 10^6 e^{-6666.7/T_r}$$

$$B + C \rightarrow P + E \qquad k_2 = 7.2177 \times 10^8 e^{-8333.3/T_r}$$

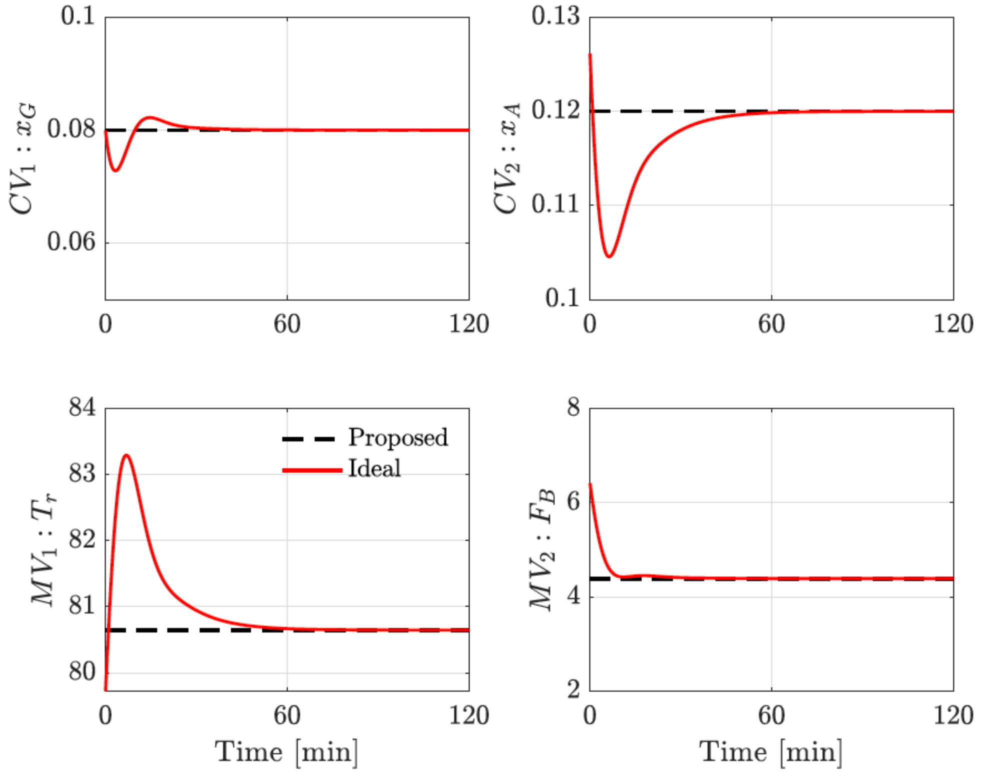
$$C + P \rightarrow G \qquad k_3 = 2.6745 \times 10^{12} e^{-11111/T_r}$$

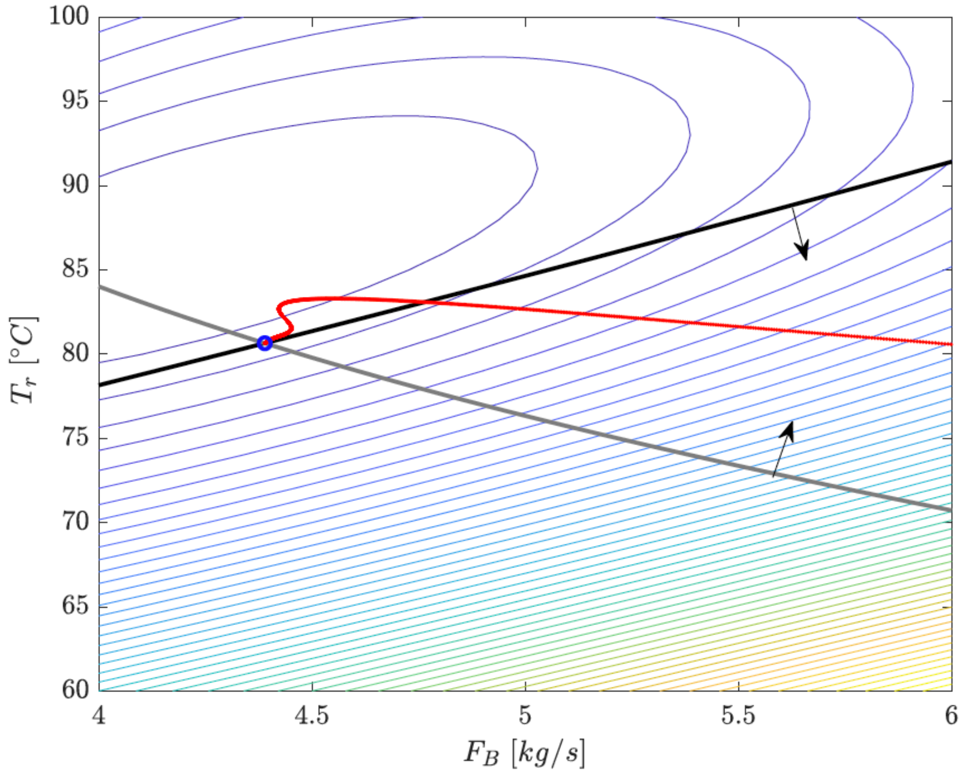
$$\min_{T_r, F_B} \; -1043.38 x_P (F_A + F_B) - 20.92 x_E (F_A + F_B) + 79.23 F_A + 118.34 F_B$$

$$\text{s.t.} \quad x_G \leq 0.08, \qquad x_A \leq 0.12$$

x_G always active

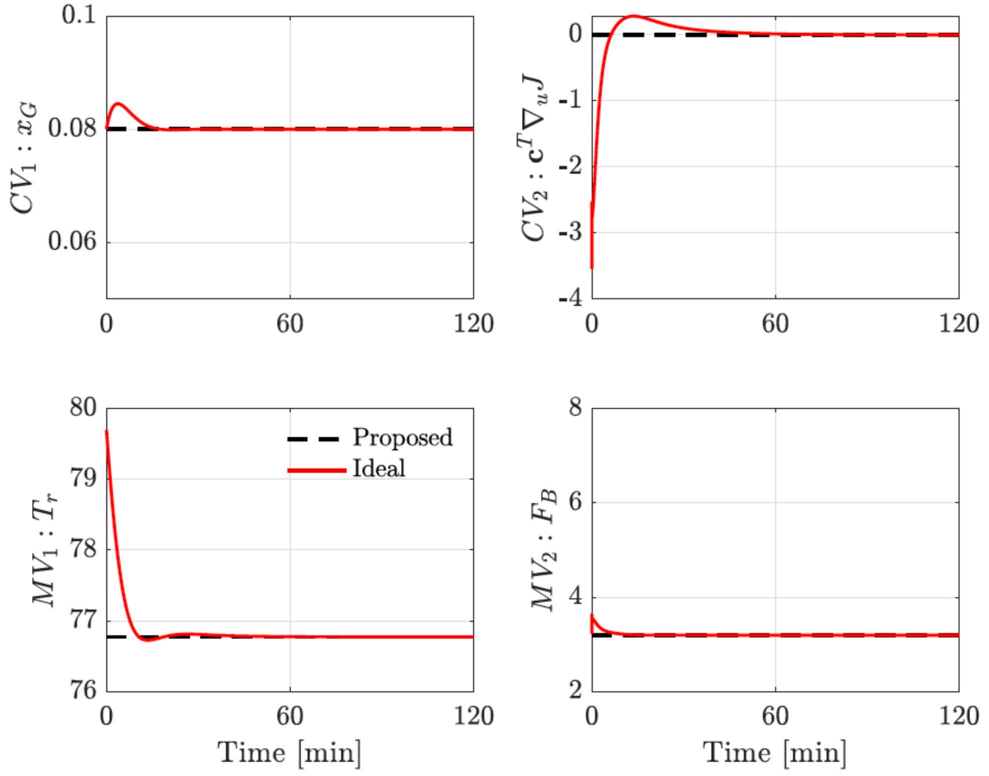
F_A = 1.8275 kg/s

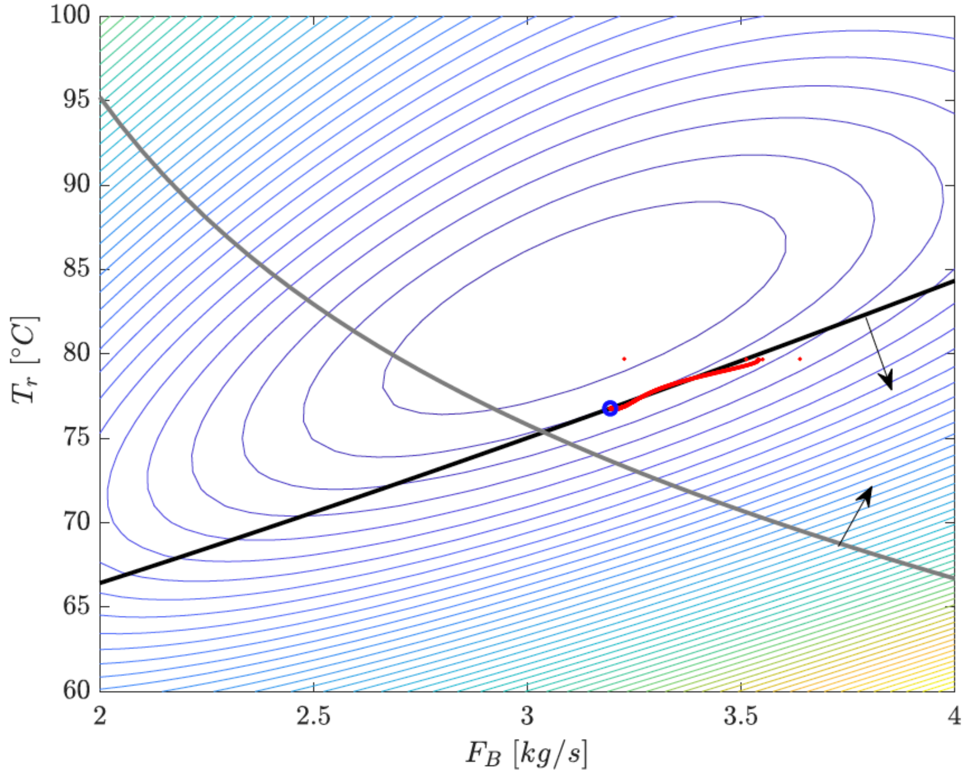




For MATLAB code, contact: dineshk@ntnu.no

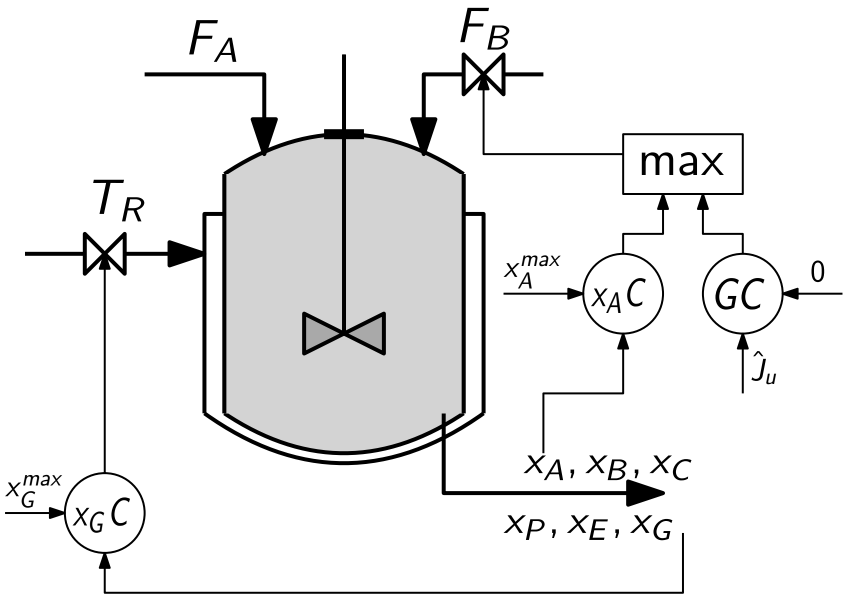
F_A = 1.3 kg/s

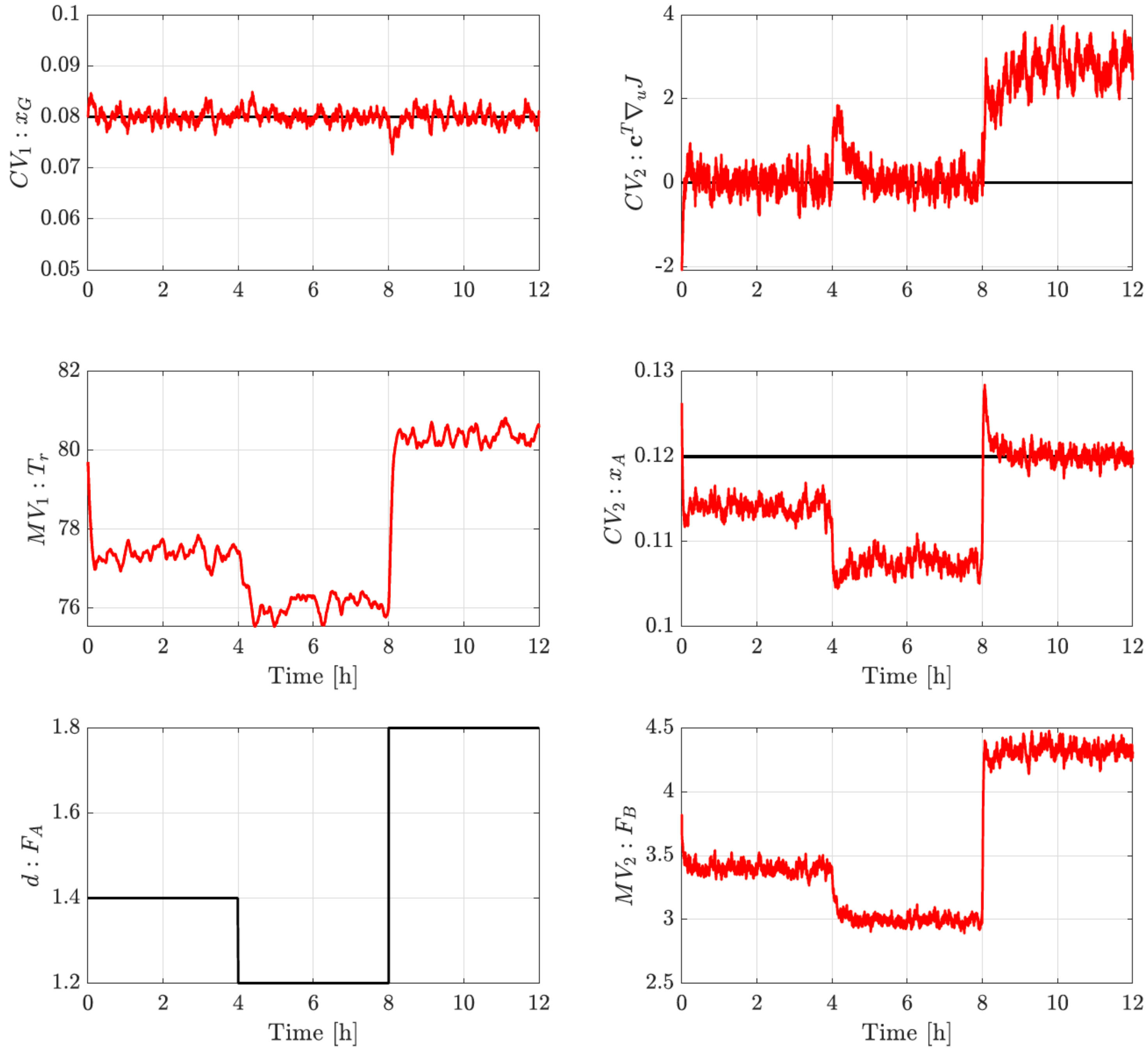




For MATLAB code, contact: dineshk@ntnu.no

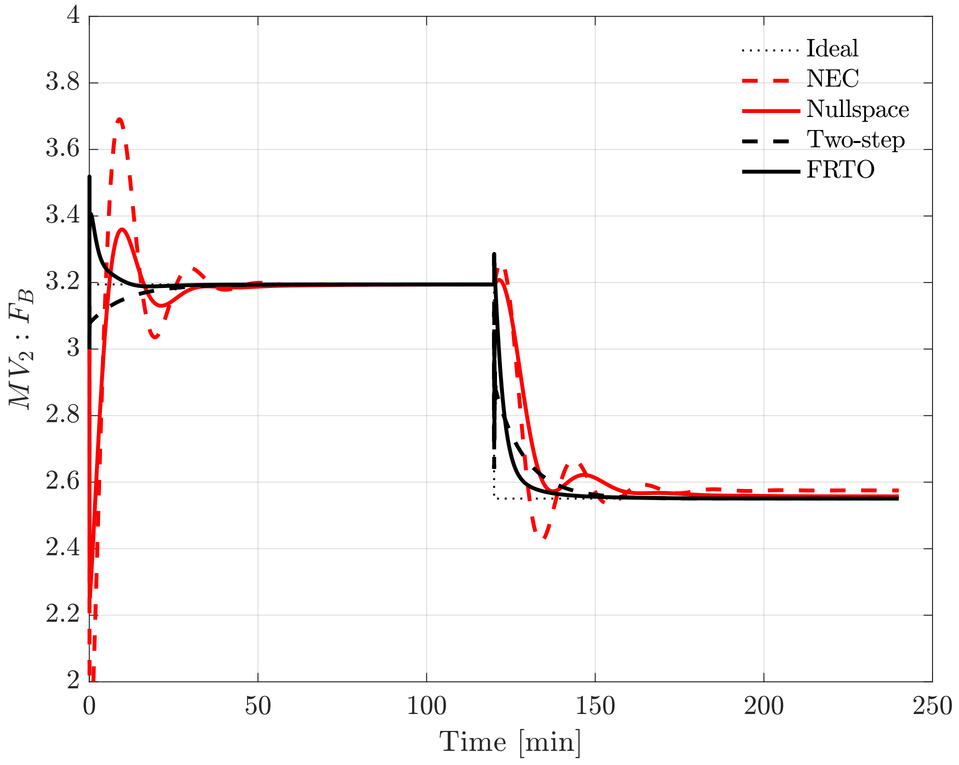
Results

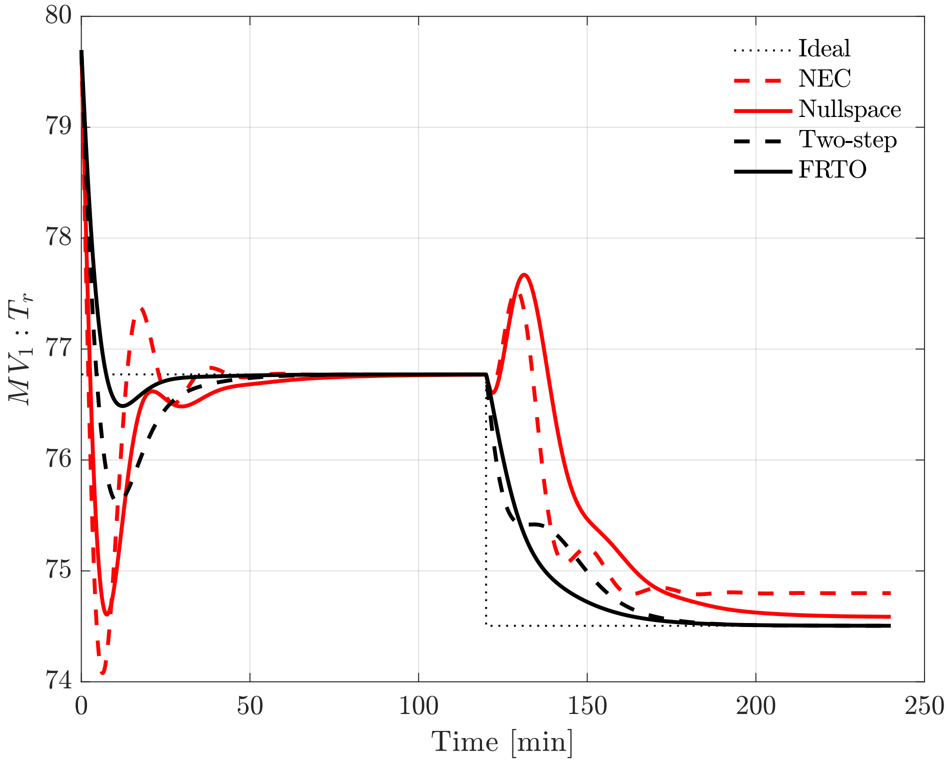




For MATLAB code, contact: dineshk@ntnu.no

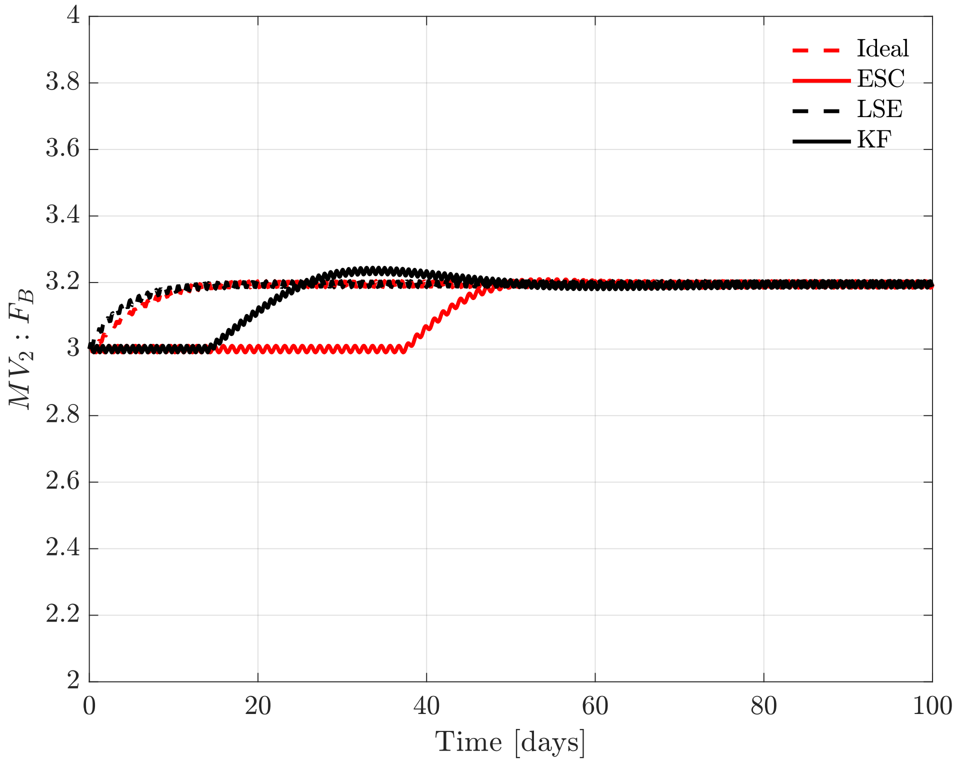
Comparison of model-based approaches

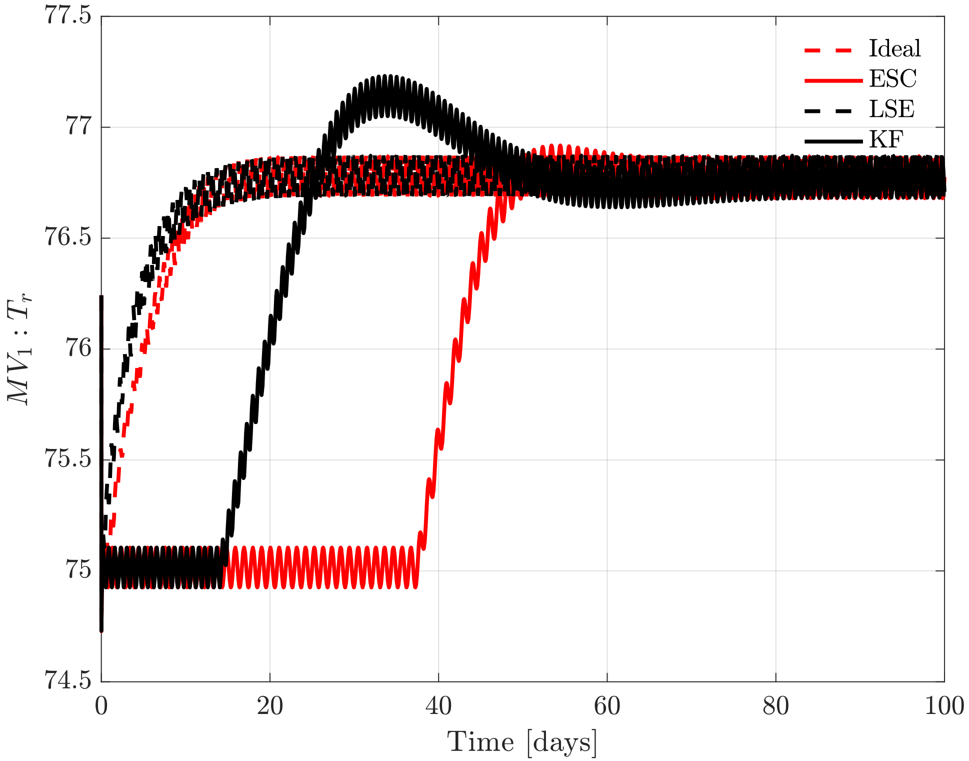




For MATLAB code, contact: dineshk@ntnu.no

Comparison of model-free approaches





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, especially for unconstrained optimum

Can partially handle using Augmented Kalman filter

- Unmodelled effects are added as bias term Δ

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k, \mathbf{d}_k) + \Delta_k$$

- The Augmented system is then given by :

$$\mathbf{x}'_{k+1} = \begin{bmatrix} \mathbf{x}_{k+1} \\ \Delta_{k+1} \end{bmatrix} = \begin{bmatrix} f(\mathbf{x}_k, \mathbf{u}_k, \mathbf{d}_k) + \Delta_k + \mathbf{w}_k \\ \Delta_k + \mathbf{w}_{\mathbf{d},k} \end{bmatrix}$$

$$\mathbf{y}_{meas,k} = \begin{bmatrix} \mathbf{h}(\mathbf{x}_k, \mathbf{u}_k) & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}_k \\ \Delta_k \end{bmatrix} + \mathbf{v}_k$$

- Use e.g. EKF to estimate the augmented state and the bias

Warning ! This is still not fully effective to handle structural mismatch

Simon, D., 2006. Optimal state estimation: Kalman, H infinity, and nonlinear approaches. John Wiley & Sons.

9. Handling error in model structure

for cases where cost is directly measured

Estimation does not update model structure

- Updating parameters to match model predictions is not sufficient if the model structure is wrong
- Good prediction model ≠ good optimization model
- A simple fix to get consistency between model and true optimum cost function is the method of "modifier adaptation".
- An even simpler fix is update the bias on the gradient to a non-zero value in a cascade manner.

P.D. Roberts (1979)

How should we modify the optimization problem

Key idea Add modifiers to make "optimality conditions" of the plant and optimality conditions of the model match

- 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.

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

- Iteratively repeat the optimization at sample times k.

Marchetti et al. (2009)

How to compute modifiers

- Zero order modifiers (bias)

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

- First order modifiers (simplest approach to get consistency between the model and the plant gradients)

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

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

Gradients from real plant

Use any model-free gradient estimation method
NB! Cost and Constraint must be measured

Marchetti et al. (2009)

Recap: Gradient Estimation methods

Model-free

- Finite Difference
- Dither demodulation (Draper & Li, 1951)
- Linear least squares (Hunnekens et al., 2014)
- Dynamic system identification (Bamberger & Isermann, 1978)
- Fast Fourier Transform (Krishnamoorthy & Skogestad)
- Multiple units (Srinivasan, 2007)
- Gausian process regression (Ferriera et al.,2018/ Matias & Jäschke, 2019)
- Fitted surfaces (Gao & Engell, 2005)
- Kalman Filter (Gelbert et al., 2012)

Model-based

- Analytical gradient from updated model
- Nullspace method (Alstad & Skogetad, 2007)
- Feedback RTO (Krishnamoorthy et al., 2019)
- Neibhouring extremals (Gros et al., 2009)

Recap: Issues with model-free gradient estimation

- **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)
- **Slow convergence**
  - Dynamic plant assumed to be a static map
  - Three timescale separation: dynamics – perturbation – convergence
- **Constant probing** of the system may be undesirable
- Unknown and abrupt disturbances affects gradient estimation

10. Hierarchical combination of different approaches

Some methods are complementary, not contradictory !

Why not combine different approaches to give improved performance?!

Examples:

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

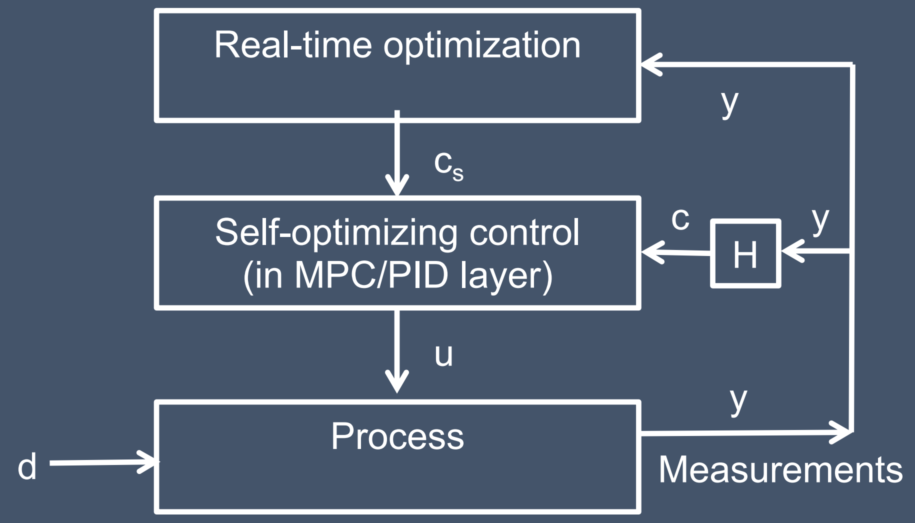
Some benefits

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

Self-optimizing control

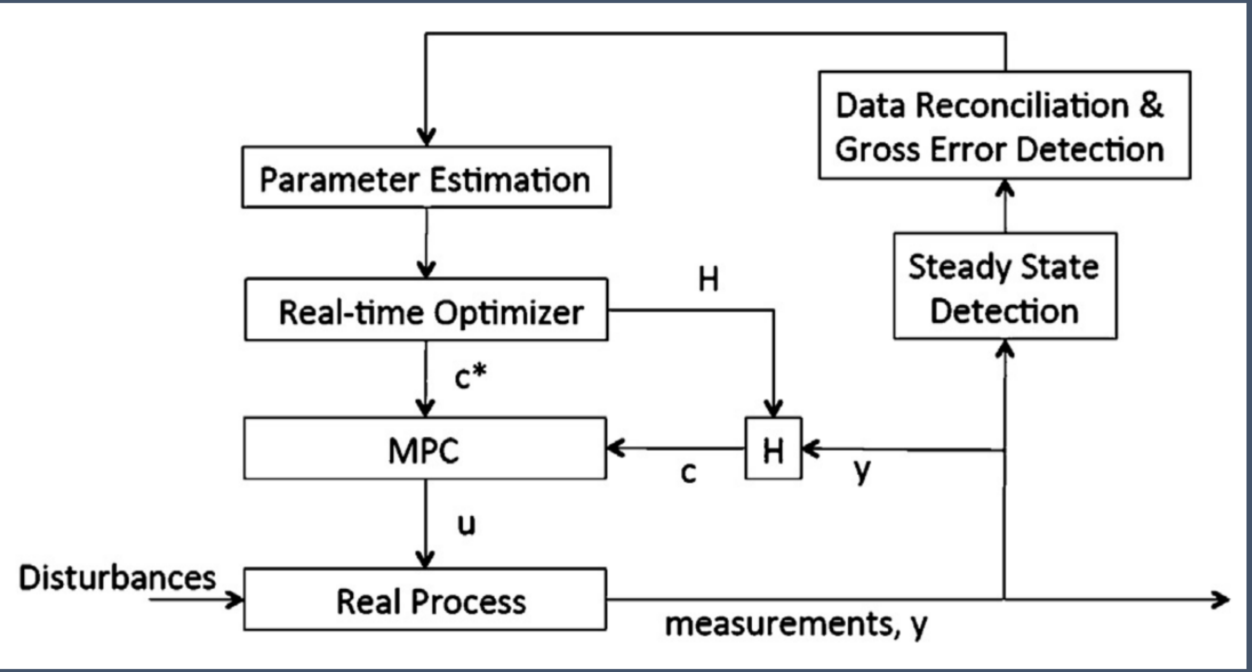
Always use self-optimizing control in supervisory/regulatory layer

- Fastest disturbance rejection
- Keeps the process in the near-optimal region
- Less effort needed from upper layers



Standard RTO + MPC + self-optimizing control

Idea: take the best from all worlds



- 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

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

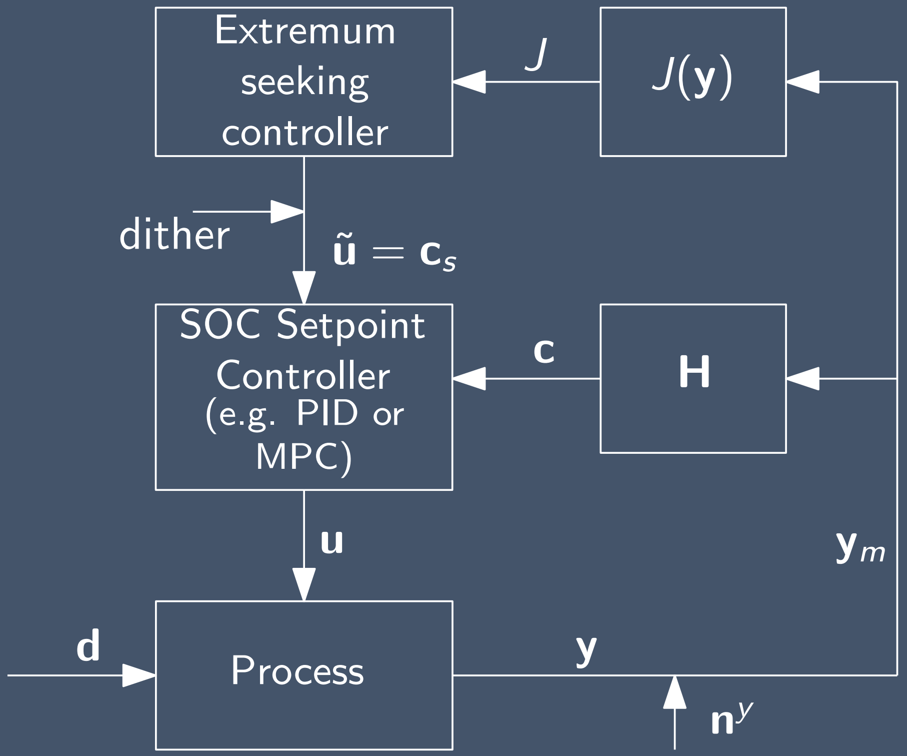
NCO tracking + self-optimizing control

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

Jäschke & Skogestad (2011)

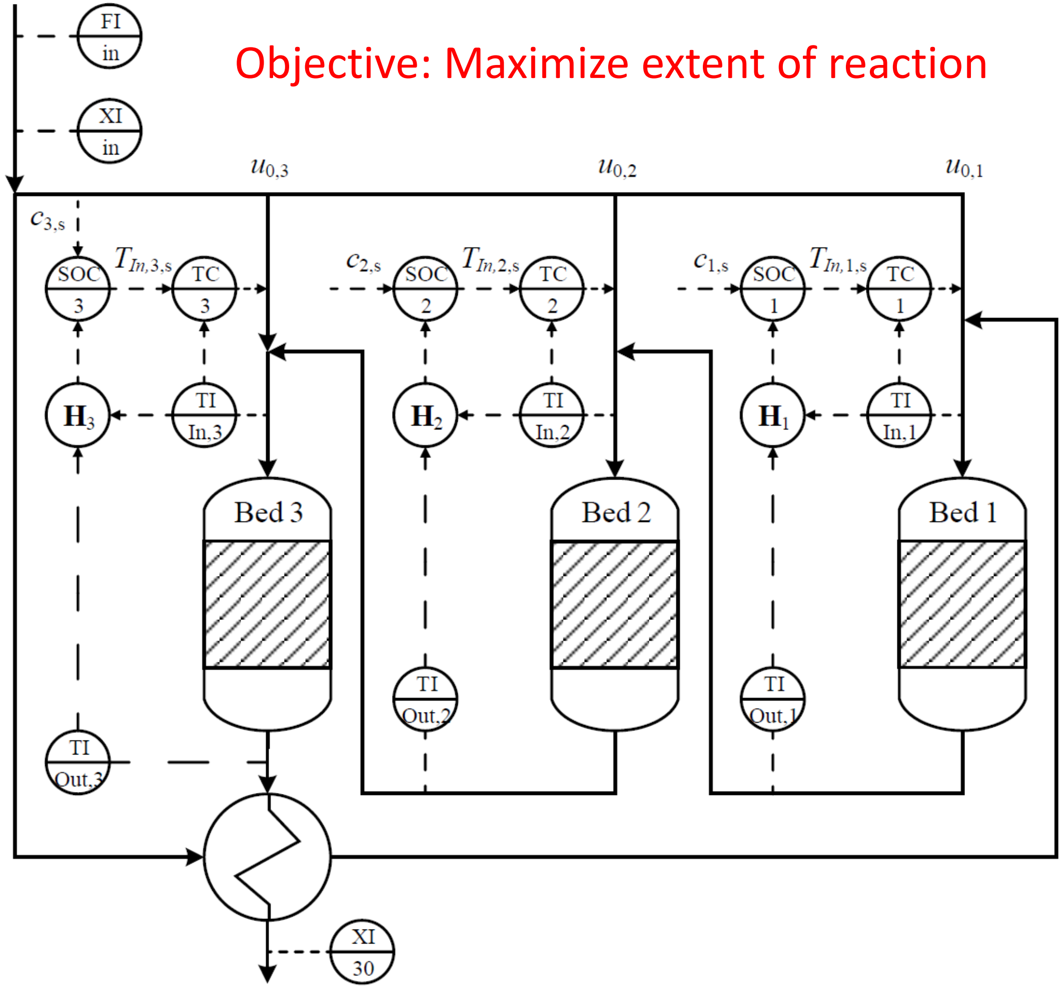
Similar approach: ESC/RTO + Self-optimizing control

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

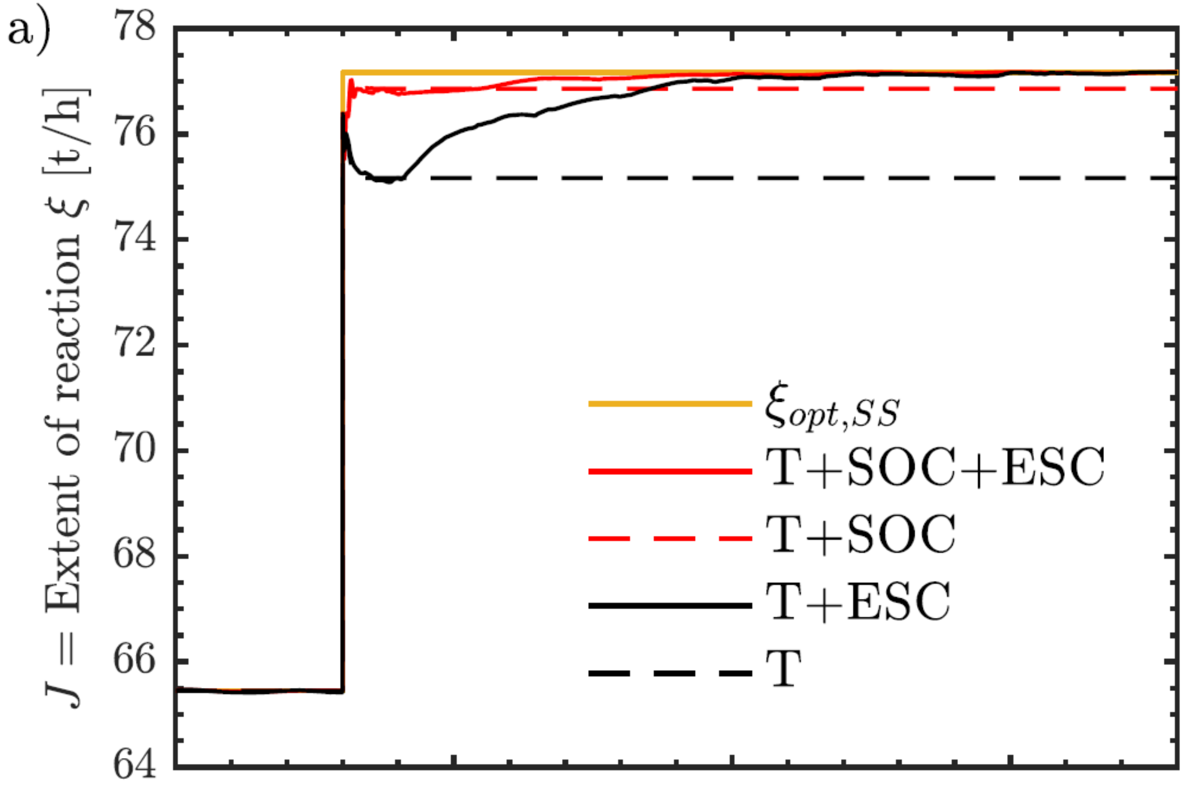


Straus, J., Krishnamoorthy, D. and Skogestad, S., 2019. On combining self-optimizing control and extremum-seeking control–Applied to an ammonia reactor case study. *Journal of Process Control*, *78*, pp.78-87.

Case study: Ammonia Reactor



Objective: Maximize extent of reaction



Response to disturbance in inlet mass flow rate

Straus, J., Krishnamoorthy, D. and Skogestad, S., 2019. On combining self-optimizing control and extremum-seeking control–Applied to an ammonia reactor case study. *Journal of Process Control*, *78*, pp.78-87.

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

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

   → **Fix: classical advanced control with constraint switching, for unconstrained DOF use extremum-seeking,**

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!)**

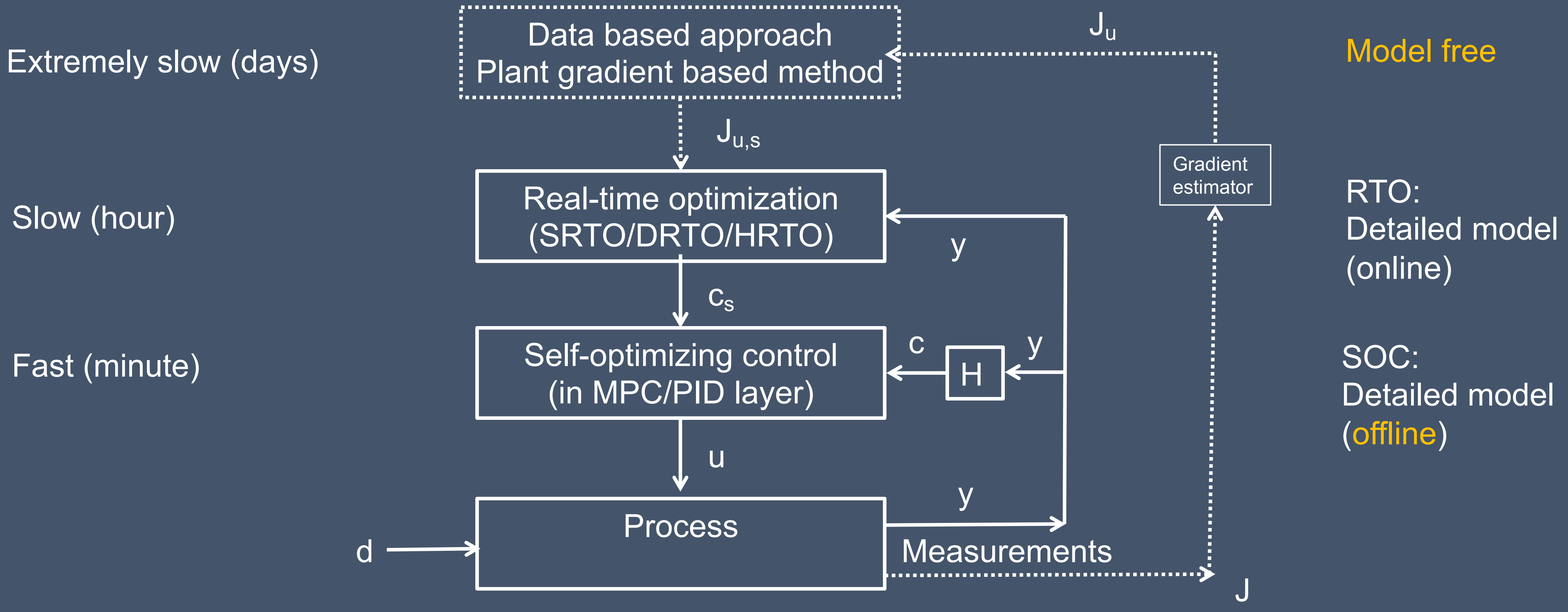
6. Incorrect model Structure

   → **Fix: Gradient bias correction / Modifier adaptation**

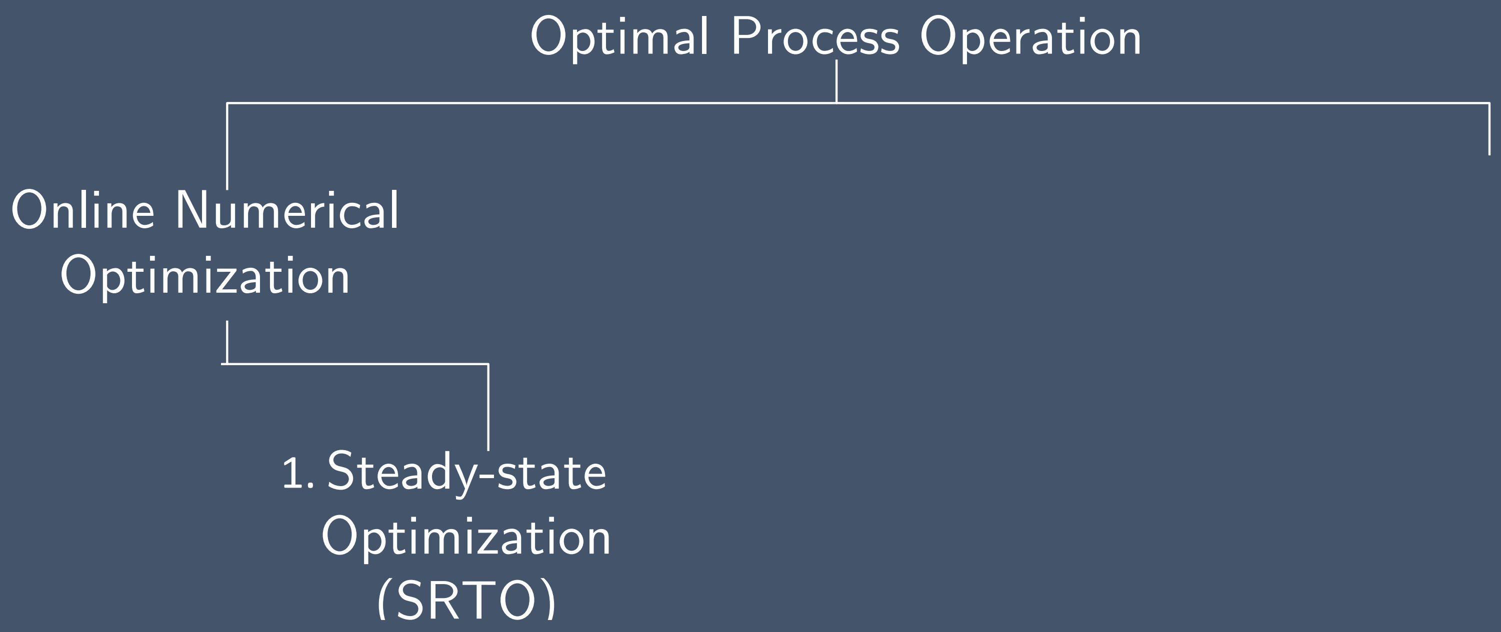
Proposal : Combine RTO with other approaches

- Model-free layer: make RTO approach the real optimum .
- SOC layer: make optimization faster, reduce wait time for model update and online optimization

Conclusion

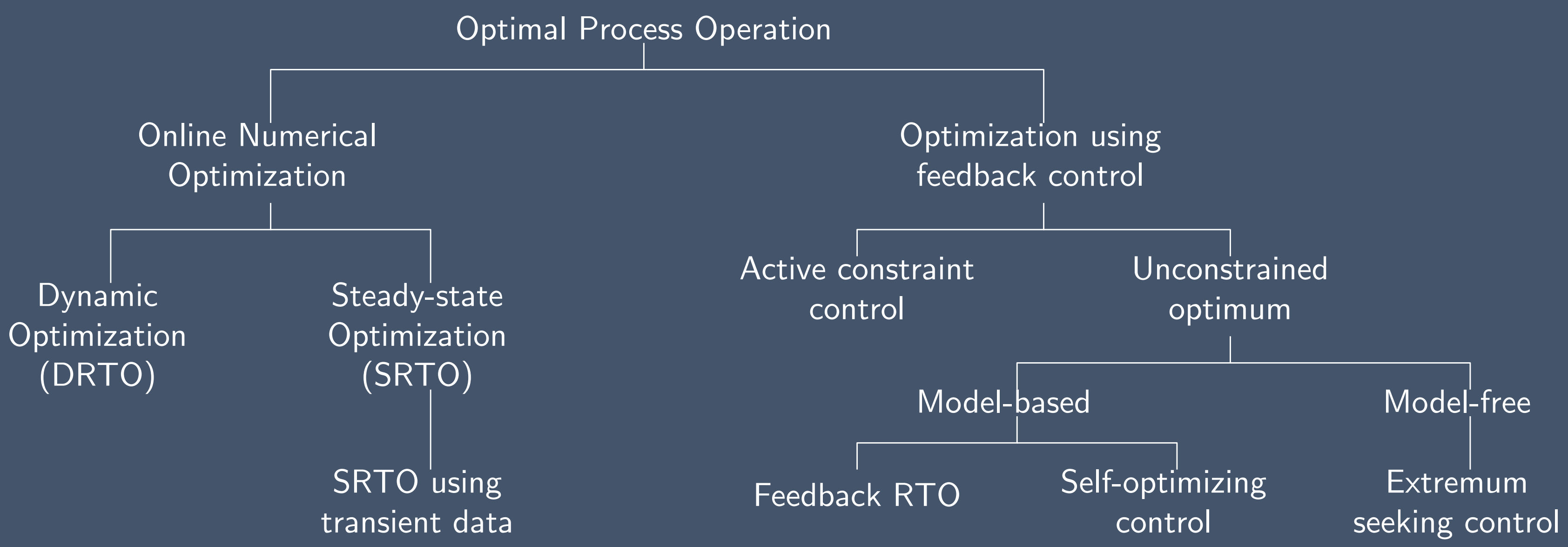

Extremely slow (days)
Slow (hour)
Fast (minute)
Data based approach
Plant gradient based method
J u
J u,s
Real-time optimization
(SRTO/DRTO/HRTO)
y
c s
Self-optimizing control
(in MPC/PID layer)
c
H
y
u
Process
y
d
Measurements
J
Gradient estimator
Model free
RTO:
Detailed model
(online)
SOC:
Detailed model
(offline)

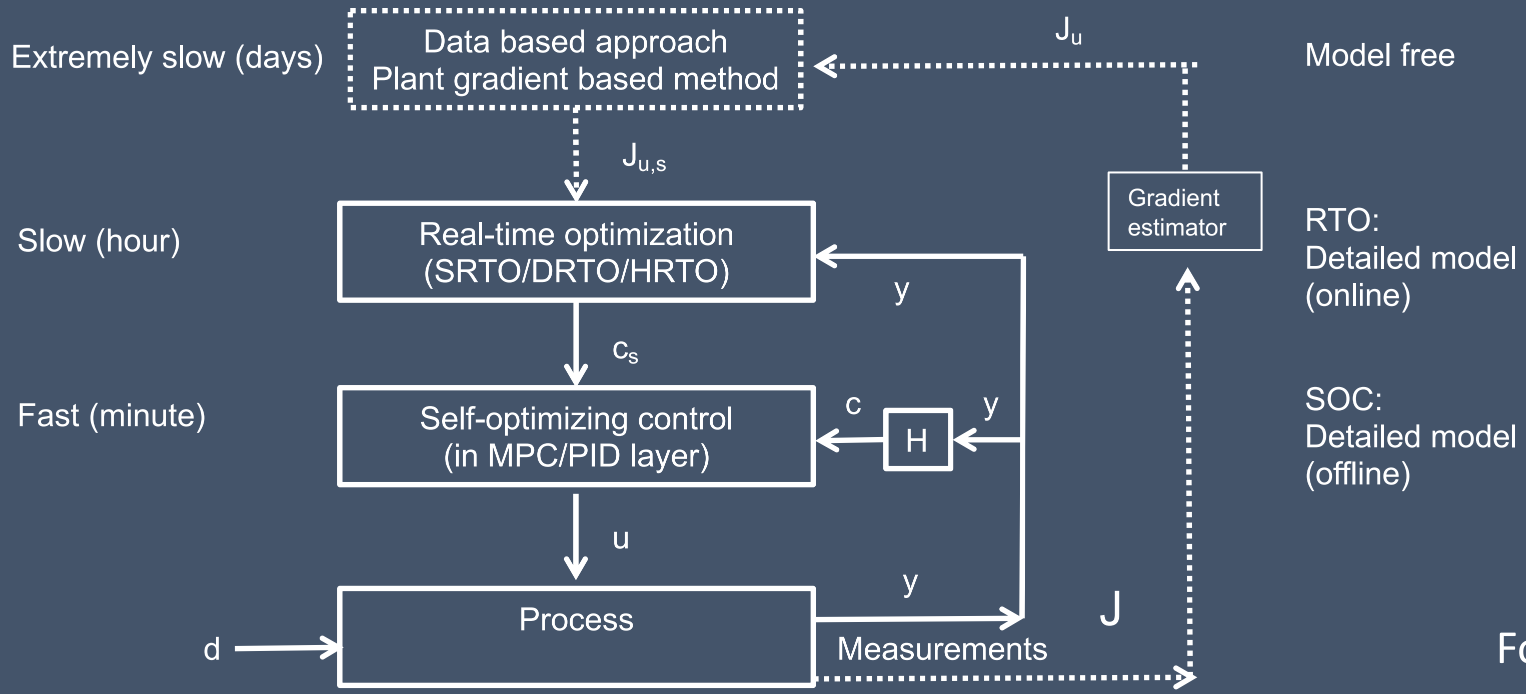
Workshop Roadmap – RTO Toolbox



Take home message: The different methods have their pros and cons. Choose the right tool for the problem at hand.

Summary





Thank you !

For MATLAB codes contact: dineshk@ntnu.no