Overview and Classification of online process optimization approaches

- IFAC DYCOPS Pre-symposium workshop

Sigurd Skogestad
Johannes Jäschke
Dinesh Krishnamoorthy
Trondheim

Norway

United Kingdom

Ireland

Edinburgh

Manchester

Dublin

Liverpool

London
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)
<table>
<thead>
<tr>
<th>NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alasdair Jack Speakman</td>
</tr>
<tr>
<td>Alireza Olama</td>
</tr>
<tr>
<td>Aris Papasavvas</td>
</tr>
<tr>
<td>Bruno Morabito</td>
</tr>
<tr>
<td>Carlos Roberto Chaves</td>
</tr>
<tr>
<td>Christiam Segundo Morales Alvarado</td>
</tr>
<tr>
<td>Cristina Zotica</td>
</tr>
<tr>
<td>Diogo Filipe Mateus Rodrigues</td>
</tr>
<tr>
<td>Diogo Ortiz Machado</td>
</tr>
<tr>
<td>Gabriel Lapa Grandi</td>
</tr>
<tr>
<td>Hiago Antonio Sirino Dangui</td>
</tr>
<tr>
<td>JESUS DAVID HERNANDEZ ORTIZ</td>
</tr>
<tr>
<td>Joakim Rostrup Andersen</td>
</tr>
<tr>
<td>José Diogo Forte de Oliveira Luna</td>
</tr>
<tr>
<td>Jose Dolores Vergara Dietrich</td>
</tr>
<tr>
<td>José Eduardo Weber dos Santos</td>
</tr>
<tr>
<td>Lucian Silva</td>
</tr>
<tr>
<td>Mandar Thombre</td>
</tr>
<tr>
<td>Marta Zagorowska</td>
</tr>
<tr>
<td>Mathilde Hotvedt</td>
</tr>
<tr>
<td>Otávio Fonseca Ivo</td>
</tr>
<tr>
<td>Rafael Sartori</td>
</tr>
<tr>
<td>Reinaldo Enrique Hernandez Rivas</td>
</tr>
<tr>
<td>Rodrigo Juliani Correa de Godoy</td>
</tr>
<tr>
<td>Seyed Jamal Haddadi</td>
</tr>
</tbody>
</table>
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)
Real-time Process Optimization with Simple Control Structures, Economic MPC or Machine Learning

Guest Editors
Prof. Sigurd Skogestad, Dr. Dinesh Krishnamoorthy

Deadline
15 November 2019

Invitation to submit
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)

Objectives

Present state

Model of system

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

Operator/RTO

Focus of this workshop

min J (economics)

Setpoint control (+ look after other variables)

Stabilize + avoid drift

"Advanced classical control”/MPC

PID-control

u = valves
General objective process operation (RTO):
Minimize cost $J = \text{maximize profit} \ (-J) [\$/s]$
subject to constraints

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

where

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

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

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

\[ \min J(x, u, d) \]

s.t.

\[ \dot{x} = f(x, u, d) \]
\[ g(x, u, d) \leq 0 \]

\[ x \in X, \quad u \in U \]

Internal variables \quad Decision variables

\[ d: \text{Parameter values / disturbances} \]
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

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 conventional 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
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

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)
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

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
\[ \hat{\theta}_k = \arg \min_{\theta} \| y_{\text{meas},k} - h(x_k, u_k) \| \]
subject to
\[ x_k = f(x_{k-1}, u_{k-1}, \theta) \]

Step 2: Dynamic Optimization
\[ u_t^* = \arg \min_{u_t} \sum_{t=k}^{k+T} J(y_t, u_t) \]
subject to
\[ x_{t+1} = f(x_t, u_t, \hat{\theta}_k) \]
\[ y_t = h(x_t, u_t) \]
\[ g(y_t, u_t) \leq 0 \]
\[ x_k = \hat{x}_k \quad \forall t \in \{k, \ldots, k+T\} \]
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

\[ u^*_t = \arg \min_{u_t} \sum_{t=k}^{k+T} J(y_t, u_t) \]
\[ s.t. \ x_{t+1} = f(x_t, u_t, \hat{\theta}_k) \]
\[ y_t = h(x_t, u_t) \]
\[ g(y_t, u_t) \leq 0 \]
\[ x_k = \hat{x}_k \quad \forall t \in \{k, \ldots, k + T\} \]
Economic MPC \sim 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)

\[ J_{\text{DRTO}} = J_{\text{eco}} + J_{\text{trac}} + J_{\text{input}} \]

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

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
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

Steady-state wait time

• 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
  
  \[ R = \frac{\max(s_{d}^2, \tau_{SM})}{s^2} \]

How to address steady-state wait time?

- **OBVIOUS: DYNAMIC RTO**

**Step 1: Dynamic Estimation**

\[ \hat{\theta}_k = \arg\min_{\theta} \| y_{\text{meas},k} - h(x_k, u_k) \| \]  \hspace{1cm} (5)

**Step 2: Dynamic Optimization**

\[ u^*_t = \arg\min_{u_t} \sum_{t=k}^{k+T} J(y_t, u_t) \]  \hspace{1cm} (6)

\[ s.t. \quad x_{t+1} = f(x_t, u_t, \hat{\theta}_k) \]

\[ y_t = h(x_t, u_t) \]

\[ g(y_t, u_t) \leq 0 \]

\[ x_k = \hat{x}_k \quad \forall t \in \{k, \ldots, k+T\} \]

Dynamic RTO has problems – especially the optimization part
Hybrid RTO

Dynamic Estimation + Static Optimization

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. \( y = f_{ss}(u, \hat{\theta}_k) \)

\[ g(y, u) \leq 0 \]

CASE STUDY: Gas lift

Main objective: Max oil prod.

Constraint: Max Gas capacity

GOR = Gas/Oil ratio in feed (reservoir)

\[ \text{max } J' = \left( \sum_{i \in N} w_{po_i} - \sum_{i \in N} w_{gl_i} \right) - f_l ||w_{fl}|| \]

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

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)
Oil and gas rates

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

### Computation Time [s]

<table>
<thead>
<tr>
<th></th>
<th>SRTO</th>
<th>HRT0</th>
<th>DRTO</th>
</tr>
</thead>
<tbody>
<tr>
<td>avg.</td>
<td>0,0184</td>
<td>0,0199</td>
<td>0,9025</td>
</tr>
<tr>
<td>max.</td>
<td>0,0223</td>
<td>0,0282</td>
<td>3,3631</td>
</tr>
</tbody>
</table>

### Integrated Profit

<table>
<thead>
<tr>
<th></th>
<th>SRTO</th>
<th>HRT0</th>
<th>DRTO</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1,8256</td>
<td>2,7019</td>
<td>2,7509</td>
</tr>
</tbody>
</table>
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
4. Feedback RTO based on novel steady-state gradient estimation method

IFAC DYCOPS Pre-symposium workshop

Dinesh Krishnamoorthy

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

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

- 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

\[ J_u < 0 \]

\[ J_u > 0 \]

\[ J_u = 0 \]
Feedback RTO: Replace steady-state optimization by feedback control

Feedback RTO

- **Step 1** - Linearize the dynamic model
  \[
  \dot{x} = f(x, u, d) \quad \Rightarrow \quad \dot{x} = Ax + Bu
  \]
  \[
  J = g(x, u) \quad \Rightarrow \quad J = Cx + Du
  \]

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

- **Step 2** - At steady-state \( \dot{x} = 0 \)
  \[
  J = \left( -CA^{-1}B + D \right) u
  \]

Feedback RTO

\[ \dot{x} = f(x, u, d) \]
\[ J = g(x, u) \]

Gradient Estimation

\[ \hat{J}_u = -CA^{-1}B + D \]

Feedback Controller (e.g. PID)

Feedback RTO

\[ \dot{x} = f(x, u, d) \]
\[ y = h(x, u) \]

State and parameter estimation

\[ \dot{\hat{x}}, \hat{d} \]

CSTR case study

\[ A \Leftrightarrow B \]

\[ x = [C_A, C_B, T]^T \]
\[ u = T_i \]
\[ d = [C_{A_{in}}, C_{B_{in}}]^T \]

\[ J = -p_{cB} C_B + (p_{T_{in}} T_{in})^2 \]

Comparison of RTO approaches: MV

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

closer look

\[ t = 400 \text{ s}, \quad d1: \text{Increase } C_{Ain} \]
\[ t = 1400 \text{ s}, \quad d2: \text{Increase } C_{Bin} \]
Comparison of RTO approaches

**COMPUTATION TIME**

<table>
<thead>
<tr>
<th>Feedback</th>
<th>SRTO</th>
<th>HRT0</th>
<th>DRTO</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.004</td>
<td>0.007</td>
<td>0.01</td>
<td>0.14</td>
</tr>
</tbody>
</table>

**INTEGRATED LOSS**

<table>
<thead>
<tr>
<th>Feedback</th>
<th>SRTO</th>
<th>HRT0</th>
<th>DRTO</th>
</tr>
</thead>
<tbody>
<tr>
<td>248.07</td>
<td>342.38</td>
<td>257.97</td>
<td>247.68</td>
</tr>
</tbody>
</table>
Feedback RTO - Other case studies

• Evaporator process¹
• Gas lift wells²
• Ammonia reactor³

1. Krishnamoorthy, D., Jahanshahi, E. and Skogestad, S., Control of steady-state gradient for an Evaporator process, PSE Asia (Submitted 2019)
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

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

• 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

\[ \dot{x} = f(x, \alpha(x, \theta)) \]
\[ y = h(x) \]

\[ \hat{\theta} \]
\[ \frac{k}{s} \]
\[ \frac{\omega_l}{s + \omega_l} \]
\[ \frac{s}{s + \omega_h} \]

\[ a \sin \omega t \]

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

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

Probe the system

Observe how the cost changes

Estimate Gradient

Decide which way to move

\[ \Delta J = 0 \]

\[ \Delta u \]

\[ \Delta J \]

\[ u \]
Classical Extremum Seeking Control

- 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

$w_{pg}$

$w_{gl}$

$w_{po}$

MV$_1$

Reservoir

GOR = Gas/Oil ratio in feed

Classical ESC

GOR vs time [h]
Least square Extremum seeking control

Fit a linear model

\[ J = J_u^T \tilde{u} + m \]

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
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

Diagram:
- RTO
  - $u$ from RTO (setpoints)
- Supervisory control (MPC)
- Regulatory control
- Process

Input $u$ vs. Cost $J$

Optimal input $u$ from model
Optimal input $u_p$ of the plant
Modifier adaptation addresses the problem of plant model mismatch

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

RTO

Supervisory control (MPC)

Regulatory control

Process
Possible solutions

1. Find a better model
   - Better parameters
   - Better structure (that matches the plant better)

   - Use plant measurements
   - No need to have an exact model

How should modify the optimization problem

• Plant Optimization problem

\[
\min_u J_p \\
\text{s.t.} \\
g_p(u) \leq 0
\]

• Model optimization problem

\[
\min_u J \\
\text{s.t.} \\
g(u) \leq 0
\]

• 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**
  \[
  \min_{u} J_p \quad \text{s.t.} \quad g_p(u) \leq 0
  \]

- **Modified model optimization problem**
  \[
  \min_{u} J_m = J(u) + \epsilon^I_k + \lambda^I_k (u - u_k) \quad \text{s.t.} \quad g_m = g(u) + \epsilon^g_k + \lambda^g_k (u - u_k) \leq 0
  \]

- **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  
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

First order modifiers:

\[ \lambda_j^T = \frac{\partial J_p(u)}{\partial u} - \frac{\partial J(u)}{\partial u} \]

\[ \lambda_g^{iT} = \frac{\partial g_{p,i}(u)}{\partial u} - \frac{\partial g_i(u)}{\partial u} \]

Gradients from real plant
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

- 2 Degrees of freedom
  $$u = \begin{bmatrix} w_{gl1} & w_{gl2} \end{bmatrix}^T$$

- 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

- 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

\[ u_{k+1} = \arg \min J_m := J(u, y(u) + \epsilon^y + \lambda^y_k T (u - u_k)) \]

s.t.

\[ g(u, y(u) + \epsilon^y + \lambda^y_k T (u - u_k)) \leq 0 \]

Marchetti et al 2009

Cost and constraint gradients are modified to match 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)
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 complete 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?)

Step 1. Define optimal operation (economics)
Usually steady state

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:

\[ J = \text{cost feed} + \text{cost energy} - \text{value of products} \]
Step 2. Optimize

(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<sub>1</sub>).
  1. Active constraints
  2. Self-optimizing variables (for unconstrained degrees of freedom)

Always try first: Move optimization into control layer
Optimization with PI-controller

\[
\begin{align*}
\text{max } & \ y \\
\text{s.t. } & \ y \leq y_{\text{max}} \\
& \ u \leq u_{\text{max}}
\end{align*}
\]

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

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

s.t. = subject to

\(y = \text{CV} = \text{controlled variable}\)
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 ($u = \text{power}$)
- 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
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
- May have infrequent adjustment of setpoint ($c_s$)
Self-optimizing control is when we can achieve an **acceptable loss** with **constant setpoint** values for the controlled variables.
The ideal “self-optimizing” variable is the gradient, $J_u$

\[ c = \frac{\partial J}{\partial u} = J_u \]

- 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:

\[
\begin{align*}
    c &= H y \\
    H &= \begin{bmatrix}
        1 & 0 & 0 & 0 \\
        0 & 1 & 0 & 0
    \end{bmatrix}
\end{align*}
\]

- Combinations of measurements:

\[
\begin{align*}
    c &= H y \\
    H &= \begin{bmatrix}
        h_{11} & h_{12} & h_{13} & h_{14} \\
        h_{21} & h_{22} & h_{23} & h_{24}
    \end{bmatrix}
\end{align*}
\]
Combinations of measurements, \( c = H_y \)

**Nullspace method** for \( H \) (Alstad):

\[ \text{HF}=0 \text{ where } F=d\text{y}_{\text{opt}}/d\text{d} \]

Example. Nullspace Method for Marathon runner

\[ u = \text{power}, \; d = \text{slope [degrees]} \]
\[ y_1 = \text{hr [beat/min]}, \; y_2 = v \; [\text{m/s}] \]
\[ c = H y, \; H = [h_1 \; h_2] \]

\[ F = \frac{\text{dy}_{\text{opt}}}{\text{dd}} = [0.25 \; -0.2]' \]
\[ HF = 0 \rightarrow h_1 f_1 + h_2 f_2 = 0.25 h_1 - 0.2 h_2 = 0 \]
Choose \( h_1 = 1 \rightarrow h_2 = \frac{0.25}{0.2} = 1.25 \)

Conclusion: \( c = \text{hr} + 1.25 \; v \)
Control \( c = \text{constant} \rightarrow \text{hr increases when v decreases (OK uphill!)} \)
Exact local method for $H$

$$
\min_H \left\| J_{uu}^{1/2} (HG_y)^{-1} H \begin{bmatrix} F W_d & W_{ny} \end{bmatrix} \right\|_2
$$

- “Minimize” in Maximum gain rule (maximize $S_1 G J_{uu}^{-1/2}, G=HG_y$)
- “Scaling” $S_1$
- “=0” in nullspace method (no noise)

Analytical solution:

$$
H = G_y^T (Y Y^T)^{-1} \text{ where } Y = \begin{bmatrix} F W_d & W_{ny} \end{bmatrix}
$$
Unconstrained degrees of freedom in practice

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

$$\min_H \| J_{uu}^{1/2} \left( H G^y \right)^{-1} H [FW_d \ W_n^y] \|_2$$

1. The optimal value of $c$ should be insensitive to disturbances
   - Small $HF = \frac{dc_{opt}}{dd}$

2. The value of $c$ should be sensitive to the inputs ("maximum gain rule")
   - Large $G = HG^y = \frac{dc}{du}$
   - Equivalent: Want flat optimum

Note: Must also find optimal setpoint for $c=CV_1$
Example: CO2 refrigeration cycle

\[ J = W_s \text{ (work supplied)} \]
\[ \text{DOF} = u \text{ (valve opening, z)} \]

Main disturbances:
\[ d_1 = T_H \]
\[ d_2 = T_{CS} \text{ (setpoint)} \]
\[ d_3 = UA_{loss} \]

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

Linear “maximum gain” analysis of controlled variables for CO2 case

<table>
<thead>
<tr>
<th>Variable (y)</th>
<th>Nom.</th>
<th>G</th>
<th>$\Delta y_{\text{opt}}(d_i)$</th>
<th>$d_i$ (T)</th>
<th>$d_i$ (T)</th>
<th>$d_i$ (U)</th>
<th>$\Delta y_{\text{opt}}$</th>
<th>n</th>
<th>Span y</th>
<th>$G'$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_h/T_2$ (bar °C$^{-1}$)</td>
<td>0.32</td>
<td>-0.291</td>
<td>0.140</td>
<td>-0.047</td>
<td>0.003</td>
<td>0.174</td>
<td>0.0033</td>
<td>1.177</td>
<td>0.25</td>
<td></td>
</tr>
<tr>
<td>$P_h$ (bar)</td>
<td>97.61</td>
<td>-78.85</td>
<td>48.3</td>
<td>-15.5</td>
<td>31.0</td>
<td>59.4</td>
<td>1.0</td>
<td>60.4</td>
<td>1.31</td>
<td></td>
</tr>
<tr>
<td>$T_2$ (°C)</td>
<td>35.5</td>
<td>36.7</td>
<td>16.27</td>
<td>-2.93</td>
<td>7.64</td>
<td>18.21</td>
<td>1</td>
<td>19.2</td>
<td>1.91</td>
<td></td>
</tr>
<tr>
<td>$T_2 - T_H$ (°C)</td>
<td>3.62</td>
<td>24</td>
<td>4.10</td>
<td>-1.92</td>
<td>5.00</td>
<td>6.75</td>
<td>1.5</td>
<td>8.25</td>
<td>2.91</td>
<td></td>
</tr>
<tr>
<td>$\gamma$</td>
<td>0.34</td>
<td>1</td>
<td>0.15</td>
<td>-0.04</td>
<td>0.18</td>
<td>0.24</td>
<td>0.05</td>
<td>0.29</td>
<td>3.45</td>
<td></td>
</tr>
<tr>
<td>$V_I$ (m$^3$)</td>
<td>0.07</td>
<td>0.03</td>
<td>-0.02</td>
<td>0.005</td>
<td>-0.03</td>
<td>0.006</td>
<td>0.001</td>
<td>0.007</td>
<td>4.77</td>
<td></td>
</tr>
<tr>
<td>$T_3$ (°C)</td>
<td>25.5</td>
<td>60.14</td>
<td>8.37</td>
<td>0.90</td>
<td>3.18</td>
<td>9.00</td>
<td>1</td>
<td>10.0</td>
<td>6.02</td>
<td></td>
</tr>
<tr>
<td>$P_h,\text{combine}$ (bar)</td>
<td>97.61</td>
<td>-592.0</td>
<td>-23.1</td>
<td>-23.1</td>
<td>3.91</td>
<td>33.0</td>
<td>9.53</td>
<td>42.5</td>
<td>13.9</td>
<td></td>
</tr>
<tr>
<td>$m_{\text{geo}}$ (kg)</td>
<td>4.83</td>
<td>-11.18</td>
<td>0.151</td>
<td>-0.136</td>
<td>0.119</td>
<td>0.235</td>
<td>0.44</td>
<td>0.675</td>
<td>16.55</td>
<td></td>
</tr>
</tbody>
</table>

Nullspace method: $c = p_{h,\text{combine}} = h_1p_h + h_2T_2 = p_h + kT_2; k = -8.53 \text{ bar/K}$
Refrigeration cycle: Proposed control structure

CV1 = Room temperature
CV2 = “temperature-corrected high CO2 pressure”
Summary Step 3. What should we control (CV₁)?

Selection of primary controlled variables \( c = CV₁ \)

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

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

“\textit{We want to find a function} \( c \) \textit{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
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 =mid ranging)
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)
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

\[
\text{max } y \\
\text{s.t. } y \leq y_{\text{max}} \\
u \leq u_{\text{max}}
\]

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

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

\(s.t. = \) subject to
\(y = CV = \) controlled variable
Optimization with PI-controller

\[
\begin{align*}
\text{min } u \\
\text{s.t. } y & \geq y^{\text{min}} \\
u & \geq u^{\text{min}}
\end{align*}
\]

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

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

s.t. = subject to
\(y = CV = \text{controlled variable}\)
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 ($u_1, u_2$)

Example: Room heating with 4 MVs

MV-MV switching

MV:
1. AC (expensive cooling)
2. CW (cooling water; cheap)
3. HW (hot water, quite cheap)
4. Electric heat, EH (expensive)
Simulation PI-control: Setpoint changes temperature
Example: Room heating with 4 MVs

MV-MV switching

MV:  
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 concentration (Max. 0.001)

Disturbances: Variation in sugar feed ($F_1$) and concentration of impurity in sugar
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 $y_{2s} = 0.001$. 

CV-CV switching
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)

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

- 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
Distillation Case Study

- Separation of 3 components

RTO Problem

$$\min_u \quad \text{Cost}^{\text{opt}} = p_F F + p_V (VB1 + VB2) - p_{D1} D1 - p_{D2} D2 - p_{X2} B2$$

- $X_b \geq 0.95$
- $X_t \geq 0.95$
- $X_x \geq 0.95$
- $VB1 \leq 4.080$ [kmol/min]
- $VB2 \leq 2.405$ [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.

--- standard MPC controlling concentrations and 1 tray temp.
--- MPC Controlling SOC variables

[Graph showing profit profile with annotations for standard MPC and MPC controlling SOC variables, with a disturbance occurring at a specific time and an RTO update at another time.]
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

Plant gradient

NCO: Necessary conditions of Optimality
Combination of self-optimizing control and NCO tracking

- 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

Jäschke & Skogestad (2011)
• 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

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

Data based approach
Plant gradient based method

Real-time optimization
(SRTO/DRTO/HRTO)

Self-optimizing control
(in MPC/PID layer)

Process

Measurements

Exremely slow (days)

Slow (hour)

Fast (minute)

Model free

RTO:
Detailed model
(online)

SOC:
Detailed model
(offline)

• Thank you!
• Next slide
  
  Red box = bad,
  green box = good,
  No box = neutral
<table>
<thead>
<tr>
<th>Cost Measured</th>
<th>self-optimizing control$^1$</th>
<th>extremum seeking control$^2$</th>
<th>new proposed method (Feedback RTO)</th>
<th>Static RTO</th>
<th>Hybrid RTO</th>
<th>economic MPC/Dynamic RTO$^3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>