Self-Optimizing Control
and
NCO tracking

in the Context of Real-Time Optimization

DYCOPS 2010, Leuven

Johannes Jäschke, Sigurd Skogestad

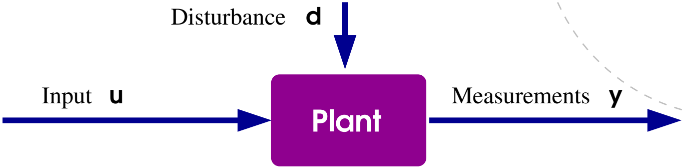
Norwegian University of Science and Technology (NTNU)
Trondheim

NTNU
Innovation and Creativity

NCO: Necessary Conditions of Optimality

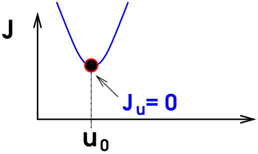
Outline

1. Introduction

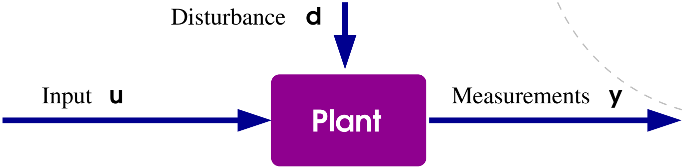


- Steady state optimization of continuous processes
- Objective:

$$\min_{\mathbf{u}} J(\mathbf{u}, \mathbf{d}) \quad \text{s.t.} \quad S(\mathbf{u}, \mathbf{d}) \leq 0.$$

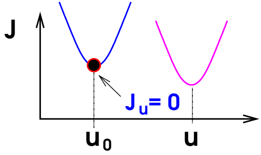


1. Introduction



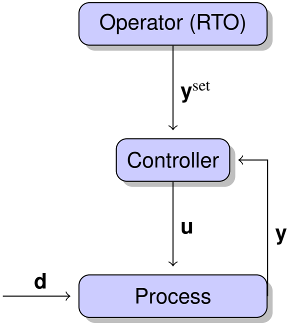
- Steady state optimization of continuous processes
- Objective:

$$\min_{\mathbf{u}} J(\mathbf{u},\mathbf{d}) \quad \text{s.t.} \quad S(\mathbf{u},\mathbf{d}) \leq 0.$$



Introduction – Conclusion preview

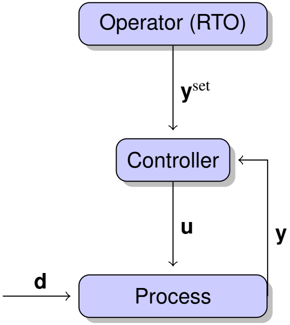
In practice



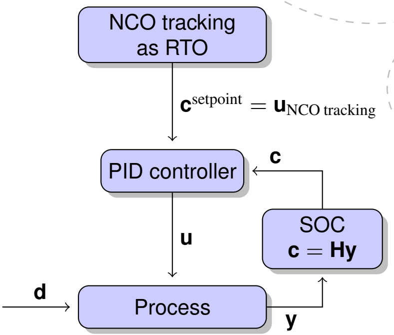
We propose.

Introduction – Conclusion preview

In practice



We propose.



2. NCO tracking for static optimization (François et al. 2005)

Optimization Problem

- Origin: Batch-to-Batch optimization

$$\min_{\mathbf{u}} J(\mathbf{u}, \mathbf{d}) \quad \text{s.t.} \quad S(\mathbf{u}, \mathbf{d}) = 0.$$

- Iteratively update the input $\mathbf{u}$

(Srinivasan 2002, François 2005, Srinivasan 2008)

2. NCO tracking for static optimization (François et al. 2005)

Optimization Problem

- Origin: Batch-to-Batch optimization

$$\min_{\mathbf{u}} J(\mathbf{u}, \mathbf{d}) \quad \text{s.t.} \quad S(\mathbf{u}, \mathbf{d}) = 0.$$

- Iteratively update the input **u** to
- satisfy the Necessary Conditions of Optimality (NCO)

$$\underbrace{S(\mathbf{u}, \mathbf{d}) = 0}_{\text{Active constraints}} \quad , \quad \underbrace{\left(\frac{\partial J(\mathbf{u}, \mathbf{d})}{\partial \mathbf{u}}\right)^{\mathbf{T}} + \left(\frac{\partial S(\mathbf{u}, \mathbf{d})}{\partial \mathbf{u}}\right)^{\mathbf{T}} \lambda = 0}_{\text{Sensitivities}}$$

(Srinivasan 2002, François 2005, Srinivasan 2008)

2. NCO tracking for static optimization (François et al. 2005)

Optimization Problem

- Origin: Batch-to-Batch optimization

$$\min_{\mathbf{u}} J(\mathbf{u}, \mathbf{d}) \quad \text{s.t.} \quad S(\mathbf{u}, \mathbf{d}) = 0.$$

- Iteratively update the input **u** to
- satisfy the Necessary Conditions of Optimality (NCO)

$$\underbrace{S(\mathbf{u}, \mathbf{d}) = 0}_{\text{Active constraints}} \quad , \quad \underbrace{\left(\frac{\partial J(\mathbf{u}, \mathbf{d})}{\partial \mathbf{u}}\right)^{\mathsf{T}} + \left(\frac{\partial S(\mathbf{u}, \mathbf{d})}{\partial \mathbf{u}}\right)^{\mathsf{T}} \lambda = 0}_{\text{Sensitivities}}$$

- Idea: Track the optimality conditions using measurements

(Srinivasan 2002, François 2005, Srinivasan 2008)

2. NCO tracking for static optimization (François et al. 2005)

Optimization Problem

- Origin: Batch-to-Batch optimization

$$\min_{\mathbf{u}} J(\mathbf{u},\mathbf{d}) \quad \text{s.t.} \quad S(\mathbf{u},\mathbf{d}) = 0.$$

- Iteratively update the input **u** to
- satisfy the Necessary Conditions of Optimality (NCO)

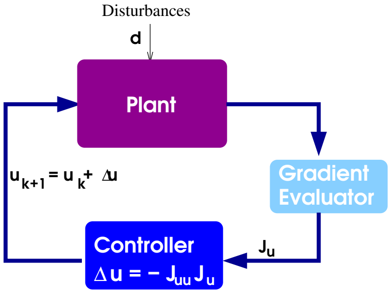
$$\underbrace{S(\mathbf{u},\mathbf{d}) = 0}_{\text{Active constraints}} \quad , \quad \underbrace{\left(\frac{\partial J(\mathbf{u},\mathbf{d})}{\partial \mathbf{u}}\right)^{\mathsf{T}} + \left(\frac{\partial S(\mathbf{u},\mathbf{d})}{\partial \mathbf{u}}\right)^{\mathsf{T}} \lambda = 0}_{\text{Sensitivities}}$$

- Idea: Track the optimality conditions using measurements
  — Measurements: Measured and estimated quantities

(Srinivasan 2002, François 2005, Srinivasan 2008)

NCO tracking for static optimization (François et al. 2005)

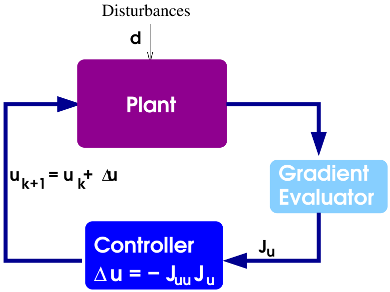
- Unconstrained optimization



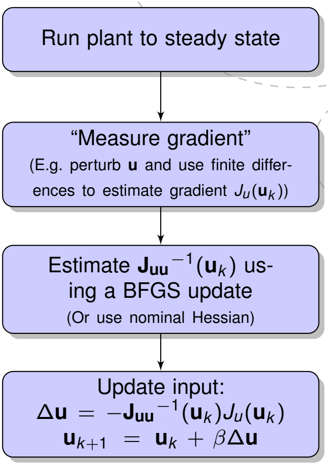
- Iteratively update **u**
- Push sensitivities J_u to zero.

NCO tracking for static optimization (François et al. 2005)

- Unconstrained optimization

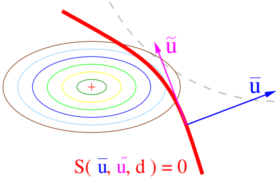


- Iteratively update **u**
- Push sensitivities J_u to zero.



NCO tracking for static optimization (François et al. 2005)

- Constraints: partition input space: $\bar{\mathbf{u}}$, $\tilde{\mathbf{u}}$



$$\frac{\partial J(\bar{\mathbf{u}}, \tilde{\mathbf{u}}, \mathbf{d})}{\partial \tilde{\mathbf{u}}} = 0$$

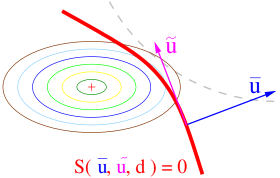
$$S(\bar{\mathbf{u}}, \tilde{\mathbf{u}}, \mathbf{d}) = 0$$

NCO tracking for static optimization (François et al. 2005)

- Constraints: partition input space: $\bar{\mathbf{u}}$, $\tilde{\mathbf{u}}$
- Constraint seeking inputs $\bar{\mathbf{u}}$:

$$\Delta\bar{\mathbf{u}} = -\left(\frac{\partial S(\bar{\mathbf{u}},\tilde{\mathbf{u}},\mathbf{d})}{\partial\bar{\mathbf{u}}}\right)^{-1} S_m$$

S_m: measured constraint



$$\frac{\partial J(\bar{\mathbf{u}},\tilde{\mathbf{u}},\mathbf{d})}{\partial\tilde{\mathbf{u}}} = 0$$

$$S(\bar{\mathbf{u}},\tilde{\mathbf{u}},\mathbf{d}) = 0$$

NCO tracking for static optimization (François et al. 2005)

- Constraints: partition input space: $\bar{\mathbf{u}}$, $\tilde{\mathbf{u}}$
- Constraint seeking inputs $\bar{\mathbf{u}}$:

$$\Delta\bar{\mathbf{u}} = -\left(\frac{\partial S(\bar{\mathbf{u}},\tilde{\mathbf{u}},\mathbf{d})}{\partial\bar{\mathbf{u}}}\right)^{-1} S_m$$

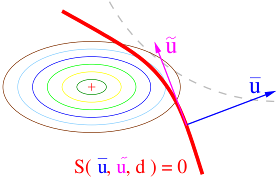
S_m: measured constraint

- Sensitivity seeking inputs $\tilde{\mathbf{u}}$

$$\Delta\tilde{\mathbf{u}} = -\left(\frac{\partial^2 J(\bar{\mathbf{u}},\tilde{\mathbf{u}},\mathbf{d})}{\partial\tilde{\mathbf{u}}^2}\right)^{-1}\left(\frac{\partial J(\bar{\mathbf{u}},\tilde{\mathbf{u}},\mathbf{d})}{\partial\tilde{\mathbf{u}}}\right)_m$$

$$= -\mathbf{J}_{\tilde{\mathbf{u}}\tilde{\mathbf{u}}}^{-1} J_{\tilde{\mathbf{u}}}$$

$J_{\tilde{\mathbf{u}}}$: measured gradient



$$\frac{\partial J(\bar{\mathbf{u}},\tilde{\mathbf{u}},\mathbf{d})}{\partial\tilde{\mathbf{u}}} = 0$$

$$S(\bar{\mathbf{u}},\tilde{\mathbf{u}},\mathbf{d}) = 0$$

NCO tracking for static optimization (François et al. 2005)

- Constraints: partition input space: $\bar{\mathbf{u}}$, $\tilde{\mathbf{u}}$
- Constraint seeking inputs $\bar{\mathbf{u}}$:

$$\Delta\bar{\mathbf{u}} = -\left(\frac{\partial S(\bar{\mathbf{u}},\tilde{\mathbf{u}},\mathbf{d})}{\partial\bar{\mathbf{u}}}\right)^{-1} S_m$$

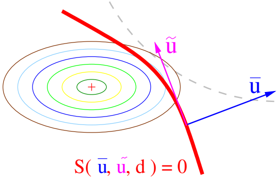
S_m: measured constraint

- Sensitivity seeking inputs $\tilde{\mathbf{u}}$

$$\Delta\tilde{\mathbf{u}} = -\left(\frac{\partial^2 J(\bar{\mathbf{u}},\tilde{\mathbf{u}},\mathbf{d})}{\partial\tilde{\mathbf{u}}^2}\right)^{-1}\left(\frac{\partial J(\bar{\mathbf{u}},\tilde{\mathbf{u}},\mathbf{d})}{\partial\tilde{\mathbf{u}}}\right)_m$$

$$= -\mathbf{J}_{\tilde{\mathbf{u}}\tilde{\mathbf{u}}}^{-1} J_{\tilde{\mathbf{u}}}$$

$J_{\tilde{\mathbf{u}}}$: measured gradient



$$\frac{\partial J(\bar{\mathbf{u}},\tilde{\mathbf{u}},\mathbf{d})}{\partial\tilde{\mathbf{u}}} = 0$$

$$S(\bar{\mathbf{u}},\tilde{\mathbf{u}},\mathbf{d}) = 0$$

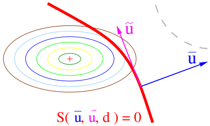
$$\boxed{\mathbf{u}^{new} = \mathbf{u}^{old} + \beta\Delta\mathbf{u}}$$

Step length parameter: β

NCO tracking for static optimization

Some comments on NCO tracking

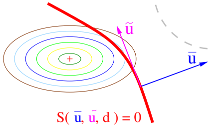
- $J_{\mathbf{u}}$ and $\Delta\mathbf{u}$ only defined at steady state
  - What about transients?



NCO tracking for static optimization

Some comments on NCO tracking

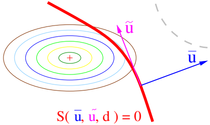
- $J_{\mathbf{u}}$ and $\Delta\mathbf{u}$ only defined at steady state
  - What about transients?
- Gradient is generally difficult to measure
  - Finite difference
  - Model



NCO tracking for static optimization

Some comments on NCO tracking

- $J_{\mathbf{u}}$ and $\Delta \mathbf{u}$ only defined at steady state
  - What about transients?
- Gradient is generally difficult to measure
  - Finite difference
  - Model



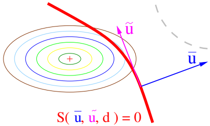
Strengths:

- Converges to the optimum after few iterations
- No knowledge about disturbance required

NCO tracking for static optimization

Some comments on NCO tracking

- $J_{\mathbf{u}}$ and $\Delta\mathbf{u}$ only defined at steady state
  - What about transients?
- Gradient is generally difficult to measure
  - Finite difference
  - Model



Strengths:

- Converges to the optimum after few iterations
- No knowledge about disturbance required

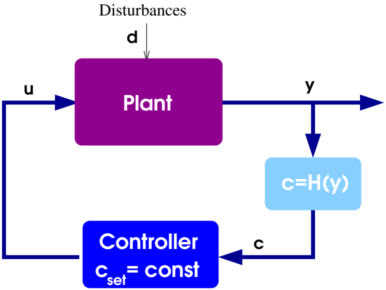
Weaknesses

- Existing knowledge about disturbances is not used
- Online (intermediate/transient) measurements not used
- Discrete input updates
  - Active constraints satisfied iteratively (⇔ Feedback)

3. Self-optimizing control

$$\min_{\mathbf{u}} J(\mathbf{u},\mathbf{d}) \quad \text{s.t.} \quad S(\mathbf{u},\mathbf{d}) = 0.$$

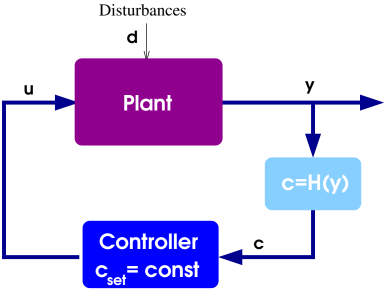
- Active constraints $S(\mathbf{u},\mathbf{d}) = 0$ controlled (by PI controller)



3. Self-optimizing control

$$\min_{\mathbf{u}} J(\mathbf{u}, \mathbf{d}) \quad \text{s.t.} \quad S(\mathbf{u}, \mathbf{d}) = 0.$$

- Active constraints $S(\mathbf{u}, \mathbf{d}) = 0$ controlled (by PI controller)

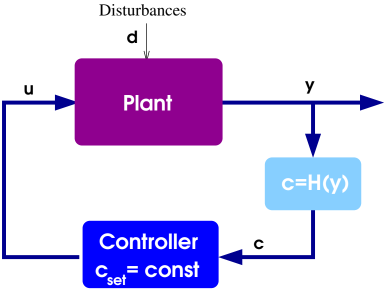


- SOC addresses the question: How to select **H**?

3. Self-optimizing control

$$\min_{\mathbf{u}} J(\mathbf{u}, \mathbf{d}) \quad \text{s.t.} \quad S(\mathbf{u}, \mathbf{d}) = 0.$$

- Active constraints $S(\mathbf{u}, \mathbf{d}) = 0$ controlled (by PI controller)



- SOC addresses the question: How to select **H**?
- **y** instant online measurements

Self-optimizing control

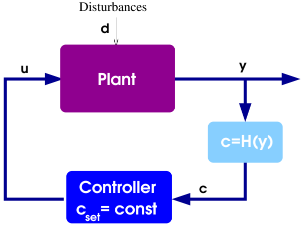
Definition (Skogestad (2000))

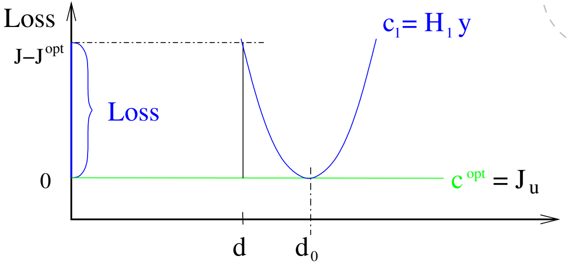
Self-optimizing control is when we can achieve an acceptable loss with constant setpoint values for the controlled variables

Self-optimizing control

Definition (Skogestad (2000))

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

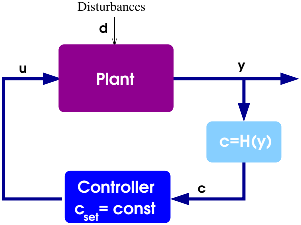


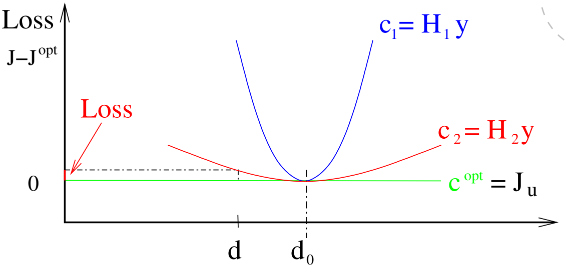


Self-optimizing control

Definition (Skogestad (2000))

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





Self-optimizing control

Interpretation: Find good and simple approximation to J_u using online measurements **y**

Self-optimizing control

Interpretation: Find good and simple approximation to J_u
using online measurements $\mathbf{y}$

- Ideal controlled variable: Gradient J_u

Self-optimizing control

Interpretation: Find good and simple approximation to J_u using online measurements $\mathbf{y}$

- Ideal controlled variable: Gradient J_u
- Single measurements:

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

Self-optimizing control

Interpretation: Find good and simple approximation to J_u
using online measurements $\mathbf{y}$

- Ideal controlled variable: Gradient J_u
- Single measurements:

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

- Combinations of measurements:

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

e. g. ratio control

Self-optimizing control

Null space method

$$\min_{\mathbf{u}} J(\mathbf{u},\mathbf{d}) = [\mathbf{u}\,\mathbf{d}] \begin{bmatrix} \mathbf{J_{uu}} & \mathbf{J_{ud}} \\ \mathbf{J_{ud}}^{\mathbf{T}} & \mathbf{J_{dd}} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{d} \end{bmatrix}$$

- Linear measurement model $\mathbf{y} = \mathbf{G}^y\mathbf{u} + \mathbf{G}_d^y\mathbf{d}$
- Linear Measurement combinations $\mathbf{c} = \mathbf{H}\mathbf{y}$

Self-optimizing control

Null space method

$$\min_{\mathbf{u}} J(\mathbf{u},\mathbf{d}) = [\mathbf{u}\,\mathbf{d}] \begin{bmatrix} \mathbf{J_{uu}} & \mathbf{J_{ud}} \\ \mathbf{J_{ud}}^{\mathbf{T}} & \mathbf{J_{dd}} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{d} \end{bmatrix}$$

- Linear measurement model $\mathbf{y} = \mathbf{G}^y\mathbf{u} + \mathbf{G}_d^y\mathbf{d}$
- Linear Measurement combinations $\mathbf{c} = \mathbf{H}\mathbf{y}$

Theorem

Given a sufficient number of measurements ($n_y \geq n_u + n_d$) and no measurement noise, select $\mathbf{H}$ such that

$$\mathbf{H}\mathbf{F} = 0$$

where

$$\mathbf{F} = \frac{\partial \mathbf{y}^{opt}}{\partial \mathbf{d}}$$

Controlling $\mathbf{c} = \mathbf{H}\mathbf{y}$ to zero yields locally zero loss from optimal operation.

Self-optimizing control

Proof

$$\mathbf{F} = \frac{\partial \mathbf{y}^{opt}}{\partial \mathbf{d}}$$

$$\mathbf{y}^{opt}(\mathbf{d}) - \mathbf{y}^{opt}(\mathbf{d}_0) = \mathbf{F}(\mathbf{d} - \mathbf{d}_0)$$

Self-optimizing control

Proof

$$\mathbf{F} = \frac{\partial \mathbf{y}^{opt}}{\partial \mathbf{d}}$$

$$\mathbf{y}^{opt}(\mathbf{d}) - \mathbf{y}^{opt}(\mathbf{d}_0) = \mathbf{F}(\mathbf{d} - \mathbf{d}_0)$$

Using $\mathbf{c} = \mathbf{H}\mathbf{y}$:

$$\underbrace{\mathbf{c}^{opt}(\mathbf{d}) - \mathbf{c}^{opt}(\mathbf{d}_0)}_{\Delta \mathbf{c}^{opt}} = \mathbf{H}\mathbf{F}(\mathbf{d} - \mathbf{d}_0)$$

Since $\mathbf{HF} = 0$, we have that $\Delta \mathbf{c} = 0$ □

Self-optimizing control

Proof

$$\mathbf{F} = \frac{\partial \mathbf{y}^{opt}}{\partial \mathbf{d}}$$

$$\mathbf{y}^{opt}(\mathbf{d}) - \mathbf{y}^{opt}(\mathbf{d}_0) = \mathbf{F}(\mathbf{d} - \mathbf{d}_0)$$

Using $\mathbf{c} = \mathbf{H}\mathbf{y}$:

$$\underbrace{\mathbf{c}^{opt}(\mathbf{d}) - \mathbf{c}^{opt}(\mathbf{d}_0)}_{\Delta \mathbf{c}^{opt}} = \mathbf{H}\mathbf{F}(\mathbf{d} - \mathbf{d}_0)$$

Since $\mathbf{HF} = 0$, we have that $\Delta \mathbf{c} = 0$ □

Obtaining F

- Assume set of disturbances $\mathbf{d}$
- Numerically find $\mathbf{F} = \frac{\Delta \mathbf{y}^{opt}}{\Delta \mathbf{d}}$
- From $\mathbf{F} = -\mathbf{G}^y \mathbf{J}_{\mathbf{uu}}{}^{-1} \mathbf{J}_{\mathbf{ud}} + \mathbf{G}^y_d$ where $\mathbf{J}_{\mathbf{uu}} = \frac{\partial^2 J}{\partial \mathbf{u}^2}$ and $\mathbf{J}_{\mathbf{ud}} = \frac{\partial J}{\partial \mathbf{d}}$

4. Properties of NCO tracking and SOC

Self-optimizing control

- Procedure for finding $\mathbf{c} = \mathbf{H}\mathbf{y}$

NCO tracking

- Controlled variable: $\mathbf{J_u}$

4. Properties of NCO tracking and SOC

Self-optimizing control

- Procedure for finding $\mathbf{c} = \mathbf{Hy}$
- J_u and Jacobian not measured

NCO tracking

- Controlled variable: $\mathbf{J_u}$
- J_u and Jacobian are measured

4. Properties of NCO tracking and SOC

Self-optimizing control

- Procedure for finding $\mathbf{c} = \mathbf{Hy}$
- J_u and Jacobian not measured
- Important **d** known a priori

NCO tracking

- Controlled variable: $\mathbf{J_u}$
- J_u and Jacobian are measured
- No assumption disturbances

4. Properties of NCO tracking and SOC

Self-optimizing control

- Procedure for finding $\mathbf{c} = \mathbf{Hy}$
- J_u and Jacobian not measured
- Important $\mathbf{d}$ known a priori
- $\mathbf{F} = \frac{\partial \mathbf{y}^{opt}}{\partial \mathbf{d}}$ from disturbance model

NCO tracking

- Controlled variable: $\mathbf{J_u}$
- J_u and Jacobian are measured
- No assumption disturbances
- No model needed

4. Properties of NCO tracking and SOC

Self-optimizing control

- Procedure for finding $\mathbf{c} = \mathbf{Hy}$
- J_u and Jacobian not measured
- Important **d** known a priori
- $\mathbf{F} = \frac{\partial \mathbf{y}^{opt}}{\partial \mathbf{d}}$ from disturbance model
- Active constraints controlled by Feedback (PI)

NCO tracking

- Controlled variable: $\mathbf{J_u}$
- J_u and Jacobian are measured
- No assumption disturbances
- No model needed
- Active constraints controlled by input updates

4. Properties of NCO tracking and SOC

Self-optimizing control

- Procedure for finding $\mathbf{c} = \mathbf{H}\mathbf{y}$
- J_u and Jacobian not measured
- Important $\mathbf{d}$ known a priori
- $\mathbf{F} = \frac{\partial \mathbf{y}^{opt}}{\partial \mathbf{d}}$ from disturbance model
- Active constraints controlled by Feedback (PI)
- Local (linearized at nominal point)

NCO tracking

- Controlled variable: $\mathbf{J_u}$
- J_u and Jacobian are measured
- No assumption disturbances
- No model needed
- Active constraints controlled by input updates
- Local, moves with operating point

4. Properties of NCO tracking and SOC

Self-optimizing control

- Procedure for finding $\mathbf{c} = \mathbf{H}\mathbf{y}$
- J_u and Jacobian not measured
- Important $\mathbf{d}$ known a priori
- $\mathbf{F} = \frac{\partial \mathbf{y}^{opt}}{\partial \mathbf{d}}$ from disturbance model
- Active constraints controlled by Feedback (PI)
- Local (linearized at nominal point)
- Continuous input change (by e. g. PI-control)

NCO tracking

- Controlled variable: $\mathbf{J_u}$
- J_u and Jacobian are measured
- No assumption disturbances
- No model needed
- Active constraints controlled by input updates
- Local, moves with operating point
- Iterative input change at sampling times

4. Properties of NCO tracking and SOC

Self-optimizing control

- Procedure for finding $\mathbf{c} = \mathbf{Hy}$
- J_u and Jacobian not measured
- Important **d** known a priori
- $\mathbf{F} = \frac{\partial \mathbf{y}^{opt}}{\partial \mathbf{d}}$ from disturbance model
- Active constraints controlled by Feedback (PI)
- Local (linearized at nominal point)
- Continuous input change (by e. g. PI-control)

$\Rightarrow$ Lower control layer

NCO tracking

- Controlled variable: $\mathbf{J_u}$
- J_u and Jacobian are measured
- No assumption disturbances
- No model needed
- Active constraints controlled by input updates
- Local, moves with operating point
- Iterative input change at sampling times

4. Properties of NCO tracking and SOC

Self-optimizing control

- Procedure for finding $\mathbf{c} = \mathbf{Hy}$
- J_u and Jacobian not measured
- Important $\mathbf{d}$ known a priori
- $\mathbf{F} = \frac{\partial \mathbf{y}^{opt}}{\partial \mathbf{d}}$ from disturbance model
- Active constraints controlled by Feedback (PI)
- Local (linearized at nominal point)
- Continuous input change (by e. g. PI-control)

⇒ Lower control layer

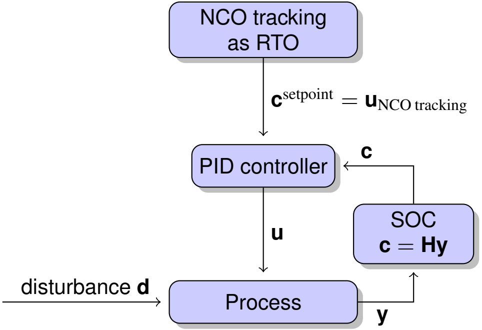
NCO tracking

- Controlled variable: $\mathbf{J_u}$
- J_u and Jacobian are measured
- No assumption disturbances
- No model needed
- Active constraints controlled by input updates
- Local, moves with operating point
- Iterative input change at sampling times

⇒ RTO layer

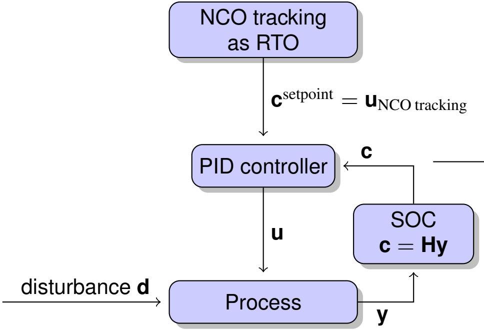
5. Use methods together

- Control active constraints $S(\mathbf{u}, \mathbf{d}) = 0$ using feedback, e.g. PI-control
- Separate layers:



5. Use methods together

- Control active constraints $S(\mathbf{u}, \mathbf{d}) = 0$ using feedback, e.g. PI-control
- Separate layers:



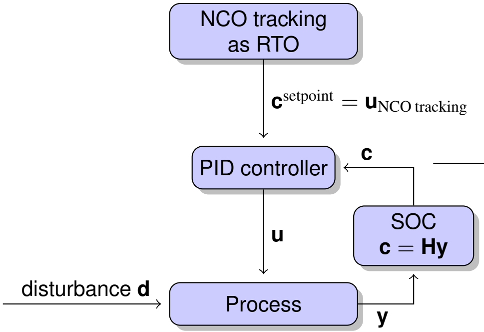
Slow time scale,
Discrete updates,

$$\Delta \mathbf{c} = -J_{cc}^{-1} J_c$$

$$\mathbf{c}_{k+1} = \mathbf{c}_k + \beta \Delta \mathbf{c}$$

5. Use methods together

- Control active constraints $S(\mathbf{u}, \mathbf{d}) = 0$ using feedback, e.g. PI-control
- Separate layers:



Slow time scale,
Discrete updates,
$\Delta \mathbf{c} = -J_{cc}^{-1} J_c$

$\mathbf{c}_{k+1} = \mathbf{c}_k + \beta \Delta \mathbf{c}$

Fast time scale,
Continuous updates,
Control $\mathbf{c} = \mathbf{H}\mathbf{y}$

Use methods together

Combines the advantages

- Smooth inputs **u**

Use methods together

Combines the advantages

- Smooth inputs **u**
- Expected disturbances rejected fast by SOC (lower layer)

Use methods together

Combines the advantages

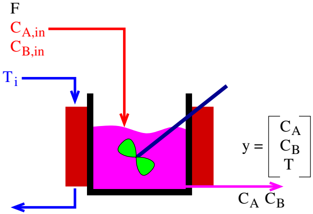
- Smooth inputs **u**
- Expected disturbances rejected fast by SOC (lower layer)
- Unexpected disturbances rejected on a slow time scale by NCO tracking (RTO layer)

Use methods together

Combines the advantages

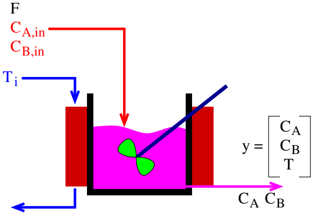
- Smooth inputs **u**
- Expected disturbances rejected fast by SOC (lower layer)
- Unexpected disturbances rejected on a slow time scale by NCO tracking (RTO layer)
- Gradient measurements not required so frequently.

6. CSTR Example (Economou 1986, Alstad 2005)



$$A \rightleftharpoons B$$

6. CSTR Example (Economou 1986, Alstad 2005)



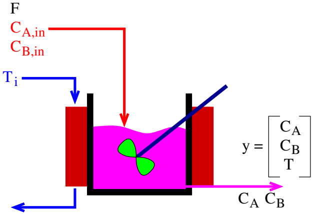
$$A \rightleftharpoons B$$

Disturbance (d):

Feed Concentration $C_{A,in}$

Feed Concentration $C_{B,in}$

6. CSTR Example (Economou 1986, Alstad 2005)



$$A \rightleftharpoons B$$

Disturbance (d):

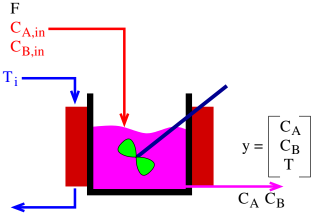
Feed Concentration $C_{A,in}$

Feed Concentration $C_{B,in}$

Input (u):

Jacket temperature T_i

6. CSTR Example (Economou 1986, Alstad 2005)



$$A \rightleftharpoons B$$

Disturbance (**d**):

Feed Concentration $C_{A,in}$

Feed Concentration $C_{B,in}$

Input (**u**):

Jacket temperature T_i

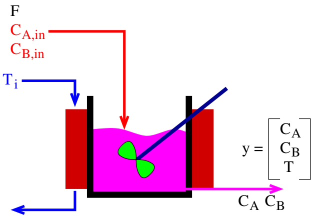
Measurements (**y**):

Concentration C_A

Concentration C_B

Temperature T

6. CSTR Example (Economou 1986, Alstad 2005)



$$A \rightleftharpoons B$$

$$\frac{dC_A}{dt} = \frac{1}{\tau}(C_{A,in} - C_A) - r$$
$$\frac{dC_B}{dt} = \frac{1}{\tau}(C_{B,in} - C_B) + r$$
$$\frac{dT}{dt} = \frac{1}{\tau}(T_i - T) + \frac{-\Delta H_{rx}}{\rho C_p} r$$

$$r = k_1 C_A - k_2 C_B$$
$$k_1 = K_1 e^{\frac{-E_1}{RT}}$$
$$k_2 = K_2 e^{\frac{-E_2}{RT}}$$

Disturbance (d):

Feed Concentration	$C_{A,in}$
Feed Concentration	$C_{B,in}$

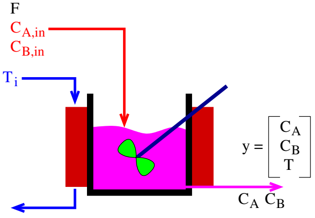
Input (u):

Jacket temperature	T_i

Measurements (y):

Concentration	C_A
Concentration	C_B
Temperature	T

6. CSTR Example (Economou 1986, Alstad 2005)



$$A \rightleftharpoons B$$

$$\frac{dC_A}{dt} = \frac{1}{\tau}(C_{A,in} - C_A) - r$$
$$\frac{dC_B}{dt} = \frac{1}{\tau}(C_{B,in} - C_B) + r$$
$$\frac{dT}{dt} = \frac{1}{\tau}(T_i - T) + \frac{-\Delta H_{rx}}{\rho C_p} r$$

$$r = k_1 C_A - k_2 C_B$$
$$k_1 = K_1 e^{\frac{-E_1}{RT}}$$
$$k_2 = K_2 e^{\frac{-E_2}{RT}}$$

Disturbance (**d**):

Feed Concentration	$C_{A,in}$
Feed Concentration	$C_{B,in}$

Input (**u**):

Jacket temperature	T_i

Measurements (**y**):

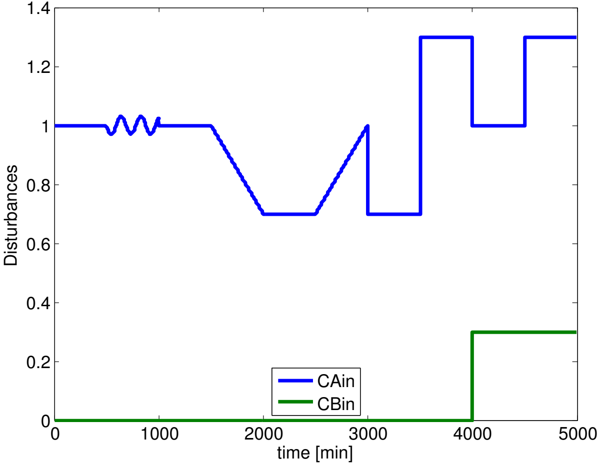
Concentration	C_A
Concentration	C_B
Temperature	T

Objective: Maximize Profit

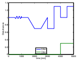
$$\max_{T_i} J = p_{C_B} C_B - (p_{T_i} T_i)^2$$

Noise: offset -0.1, std dev: 0.2

CSTR Example – Disturbances (d)



CSTR Example – Operation using NCO tracking



Run plant to steady state

↓

Perturb **u** and run plant to steady state to estimate $J_u(\mathbf{u}_k)$

↓

Estimate $\mathbf{J_{uu}}^{-1}(\mathbf{u}_k)$ using a BFGS update

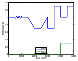
↓

Update input:

$$\Delta\mathbf{u} = -\mathbf{J_{uu}}^{-1}(\mathbf{u}_k)J_u(\mathbf{u}_k)$$
$$\mathbf{u}_{k+1} = \mathbf{u}_k + \beta\Delta\mathbf{u}$$

Sampling time	$T_S = 10$ min
Perturbation	$\Delta T_{pert} = 1$ K
Step size	$\beta = 0.4$

CSTR Example – Operation using NCO tracking



Run plant to steady state

↓

Perturb **u** and run plant to steady state to estimate $J_u(\mathbf{u}_k)$

↓

Estimate $\mathbf{J_{uu}}^{-1}(\mathbf{u}_k)$ using a BFGS update

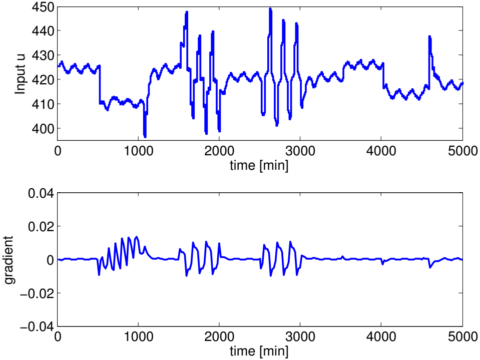
↓

Update input:

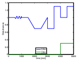
$$\Delta\mathbf{u} = -\mathbf{J_{uu}}^{-1}(\mathbf{u}_k)J_u(\mathbf{u}_k)$$
$$\mathbf{u}_{k+1} = \mathbf{u}_k + \beta\Delta\mathbf{u}$$

Sampling time	$T_S = 10$ min
Perturbation	$\Delta T_{pert} = 1$ K
Step size	$\beta = 0.4$

Input usage and gradient

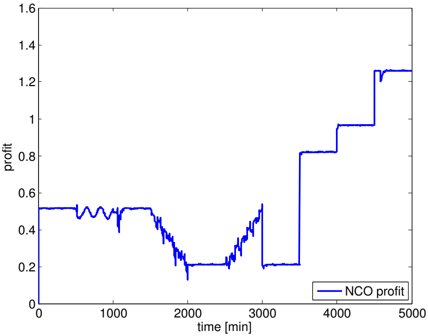


CSTR Example – Operation using NCO tracking

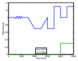


Instantaneous profit

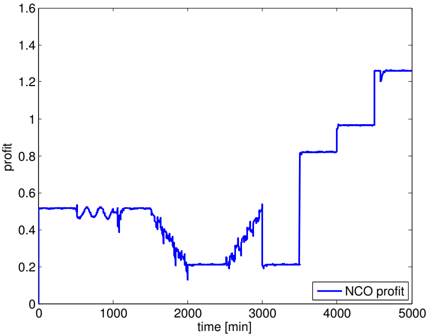
Concentrations and reactor temperature



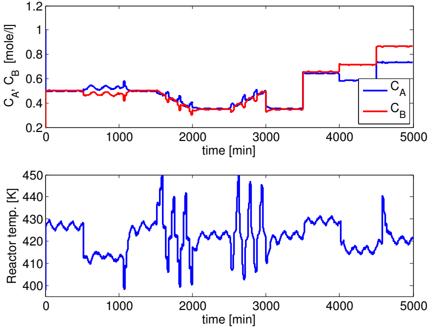
CSTR Example – Operation using NCO tracking



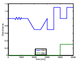
Instantaneous profit



Concentrations and reactor temperature



CSTR Example – Operation using SOC



Self-optimizing control

- Cost is not measured
- Select $\mathbf{H}$ such that $\mathbf{HF} = 0$

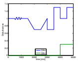
$$\mathbf{F} = \frac{\partial \mathbf{y}^{opt}}{\partial \mathbf{d}}$$

- $\mathbf{c} = \mathbf{Hy}$

Controlled variable:

$$\mathbf{c} = -0.769C_A + 0.639C_B + 0.005T$$

CSTR Example – Operation using SOC



Input and controlled variable **c**

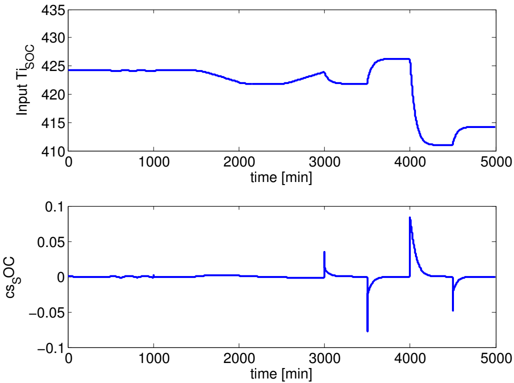
Self-optimizing control

- Cost is not measured
- Select **H** such that $\mathbf{HF} = 0$

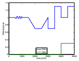
$$\mathbf{F} = \frac{\partial \mathbf{y}^{opt}}{\partial \mathbf{d}}$$

- $\mathbf{c} = \mathbf{Hy}$

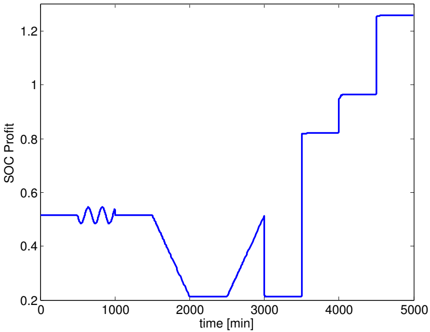
Controlled variable:
$\mathbf{c} =$
$-0.769 C_A + 0.639 C_B + 0.005 T$



CSTR Example – Operation using SOC

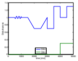


Instantaneous Profit

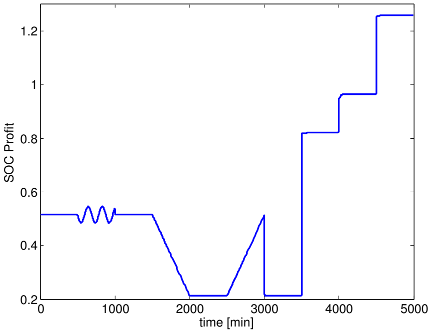


Concentrations and reactor temperature

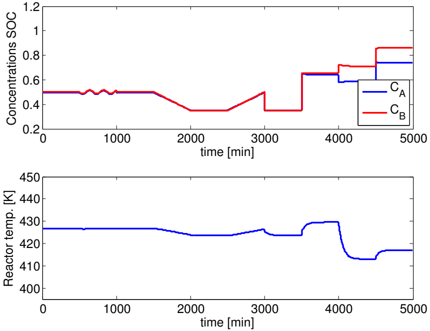
CSTR Example – Operation using SOC



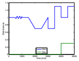
Instantaneous Profit



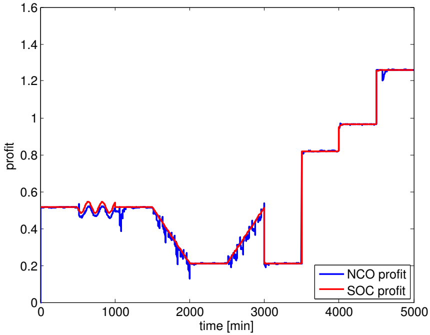
Concentrations and reactor temperature



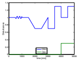
CSTR Example – Operation using SOC



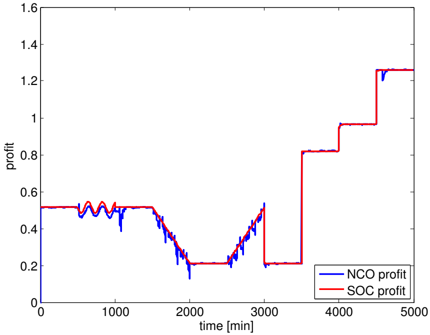
Comparing instantaneous profit



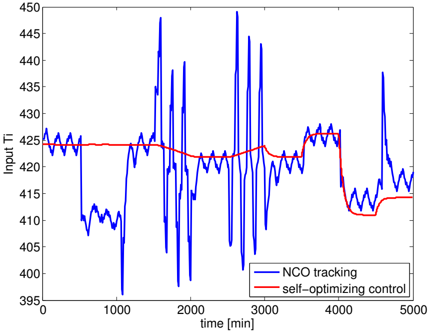
CSTR Example – Operation using SOC



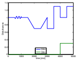
Comparing instantaneous profit



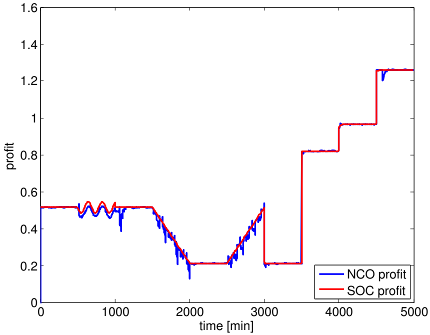
Comparing input usage



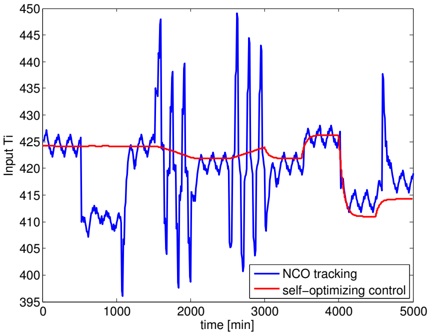
CSTR Example – Operation using SOC



Comparing instantaneous profit



Comparing input usage



Winner so far: SOC

CSTR Example – Unexpected disturbance in E_2

- New disturbance: Activation Energy E_2 changes $+10\%$

$$k_2 = K_2 e^{\frac{-E_2}{RT}}$$

- Reaction rate:

$$r = k_1 C_A - k_2 C_B$$

- Favours formation of product B

CSTR Example – Unexpected disturbance in E_2

- New disturbance: Activation Energy E_2 changes $+10\%$

$$k_2 = K_2 e^{\frac{-E_2}{RT}}$$

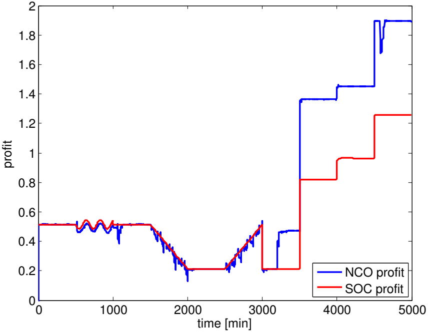
- Reaction rate:

$$r = k_1 C_A - k_2 C_B$$

- Favours formation of product B
- Not taken into account when calculating $\mathbf{F} = \frac{\partial \mathbf{y}^{opt}}{\partial \mathbf{d}}$

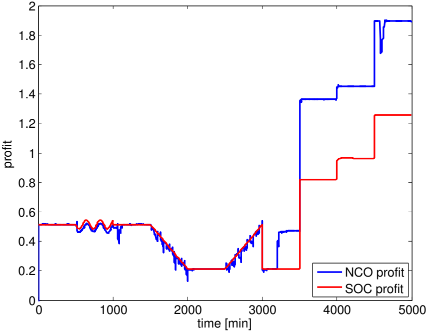
CSTR Example – Unexpected disturbance in E_2

Instantaneous profit



CSTR Example – Unexpected disturbance in E_2

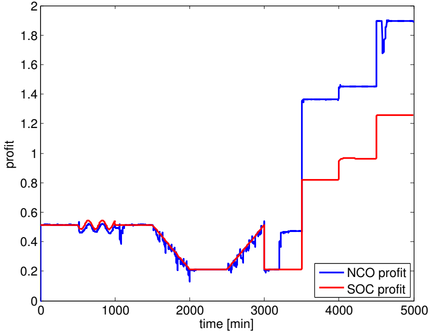
Instantaneous profit



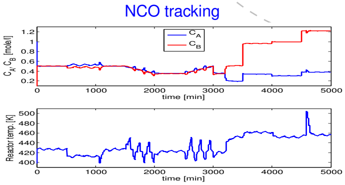
Reactor states

CSTR Example – Unexpected disturbance in E_2

Instantaneous profit

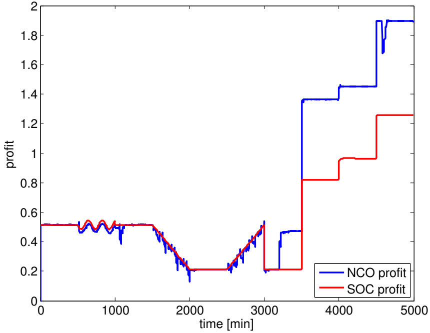


Reactor states

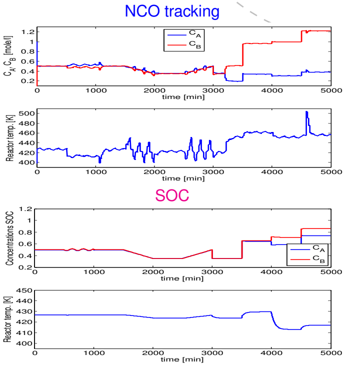


CSTR Example – Unexpected disturbance in E_2

Instantaneous profit

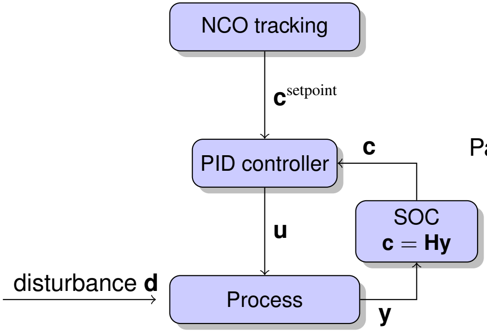


Reactor states



Winner this time: NCO tracking

CSTR Example – Combined method

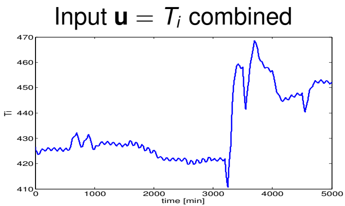


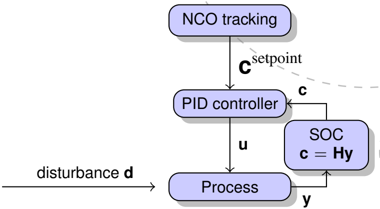
- NCO tracking in RTO layer
- **u**(*NCO*) = setpoint for **c**

Parameters:

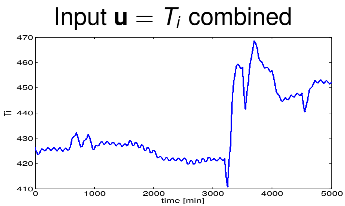
Sampling time:	$T_S^{combi} = 25$ min
Perturbation:	$\Delta \mathbf{c}_{pert} = 0.02$
Step size:	$\beta = 0.3$

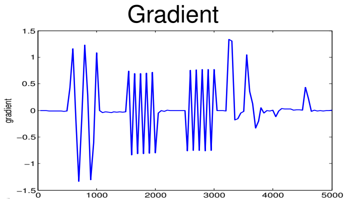
CSTR Example – Combined method

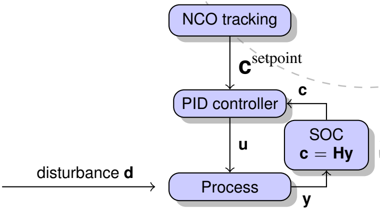




CSTR Example – Combined method

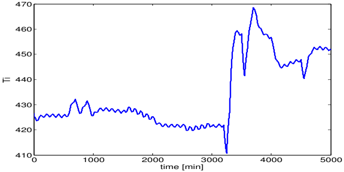


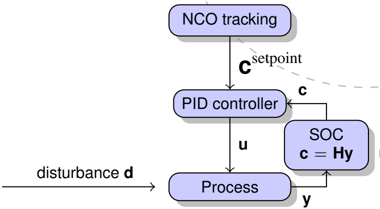




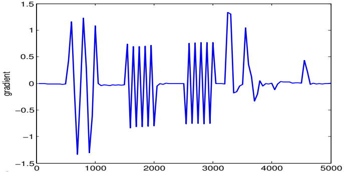
CSTR Example – Combined method

Input $\mathbf{u} = T_i$ combined

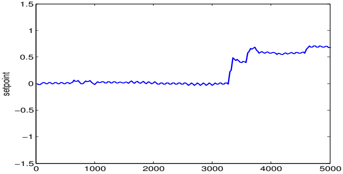




Gradient

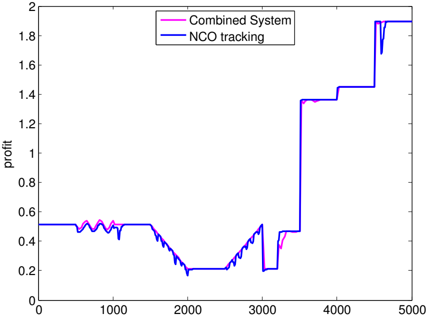


Setpoints for $\mathbf{c} = \mathbf{Hy}$



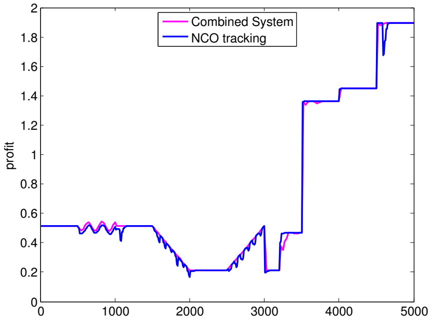
CSTR Example – Combined method

Profit

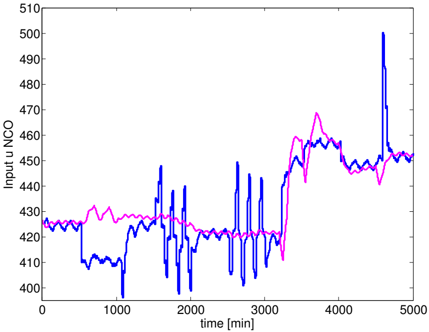


CSTR Example – Combined method

Profit



Input $\mathbf{u} = T_i$



7. Conclusions

NCO tracking and SOC

- Have the same purpose:

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

7. Conclusions

NCO tracking and SOC

- Have the same purpose:

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

- Are not competing methods

7. Conclusions

NCO tracking and SOC

- Have the same purpose:

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

- Are not competing methods
- Should be seen as complementary
  — NCO tracking as RTO
  — SOC used in the lower layer.

7. Conclusions

NCO tracking and SOC

- Have the same purpose:

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

- Are not competing methods
- Should be seen as complementary
  — NCO tracking as RTO
  — SOC used in the lower layer.

- Self-optimizing control can not replace RTO

7. Conclusions

NCO tracking and SOC

- Have the same purpose:

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

- Are not competing methods
- Should be seen as complementary
  — NCO tracking as RTO
  — SOC used in the lower layer.

- Self-optimizing control can not replace RTO
- Self-optimizing control layer reduces need for RTO/NCO tracking updates
  — less perturbations and discrete input changes

7. Conclusions

NCO tracking and SOC

- Have the same purpose:

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

- Are not competing methods
- Should be seen as complementary
  - — NCO tracking as RTO
  - — SOC used in the lower layer.
- Self-optimizing control can not replace RTO
- Self-optimizing control layer reduces need for RTO/NCO tracking updates
  - — less perturbations and discrete input changes
- Use SOC in the lower layer

Thank you