Explicit solutions for self-optimizing control

Henrik Manum

Department of Chemical Engineering
Norwegian University of Science and Technology
N-7491 Trondheim

Bratislava, 19 Feb 2010

NTNU

Outline

- Self-optimizing control and links with explicit MPC
- Efficient on-line computation of constrained optimal control
- Self-optimizing control with constraints (“constrained SOC”)
- Example: Ammonia production
- How does constrained SOC fit into the control hierarchy?

Outline

- Self-optimizing control and links with explicit MPC
- Efficient on-line computation of constrained optimal control
- Self-optimizing control with constraints (“constrained SOC”)
- Example: Ammonia production
- How does constrained SOC fit into the control hierarchy?

Outline

- Self-optimizing control and links with explicit MPC
- Efficient on-line computation of constrained optimal control
- Self-optimizing control with constraints ("constrained SOC")
- Example: Ammonia production
- How does constrained SOC fit into the control hierarchy?

Outline

- Self-optimizing control and links with explicit MPC
- Efficient on-line computation of constrained optimal control
- Self-optimizing control with constraints (“constrained SOC”)
- Example: Ammonia production
- How does constrained SOC fit into the control hierarchy?

Outline

- Self-optimizing control and links with explicit MPC
- Efficient on-line computation of constrained optimal control
- Self-optimizing control with constraints (“constrained SOC”)
- Example: Ammonia production
- How does constrained SOC fit into the control hierarchy?

Outline

- Self-optimizing control and links with explicit MPC
- Efficient on-line computation of constrained optimal control
- Self-optimizing control with constraints ("constrained SOC")
- Example: Ammonia production
- How does constrained SOC fit into the control hierarchy?

Outline

- Self-optimizing control and links with explicit MPC
- Efficient on-line computation of constrained optimal control
- Self-optimizing control with constraints (“constrained SOC”)
- Example: Ammonia production
- How does constrained SOC fit into the control hierarchy?

Implementation of optimal operation using off-line computations

Paradigm 1

On-line optimizing control where measurements are primarily used to update the model. With arrival of new measurements, the optimization problem is resolved for the inputs.

Paradigm 2

Pre-computed solutions based on off-line optimization. Typically, the measurements are used to (indirectly) update the inputs using feedback control schemes. Focus of this work.

Implementation of optimal operation using off-line computations

Paradigm 1

On-line optimizing control where measurements are primarily used to update the model. With arrival of new measurements, the optimization problem is resolved for the inputs.

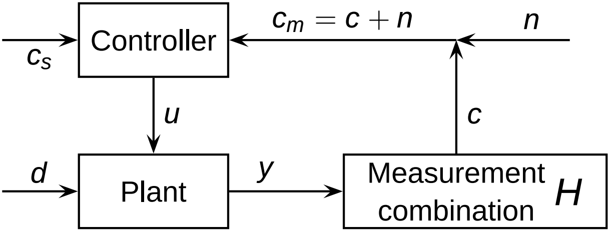
Example: Classical (implicit) MPC.

Paradigm 2

Pre-computed solutions based on off-line optimization. Typically, the measurements are used to (indirectly) update the inputs using feedback control schemes. Focus of this work.

Examples: Explicit MPC and self-optimizing control.

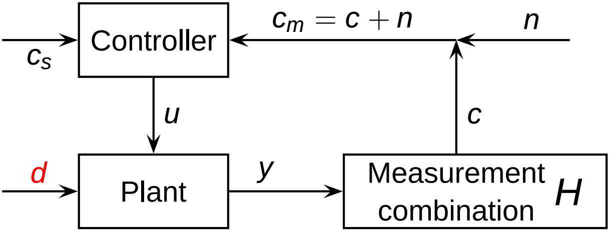
What variables should we control?



Self-optimizing control

Choice of H such that acceptable operation is achieved with constant setpoints (c_s constant).

What variables should we control?

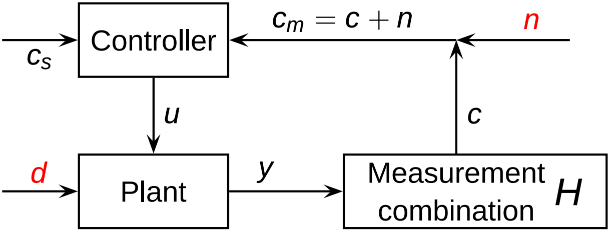


Self-optimizing control

Choice of H such that acceptable operation is achieved with constant setpoints (c_s constant).

- Optimal c_s is invariant with respect to disturbances d

What variables should we control?

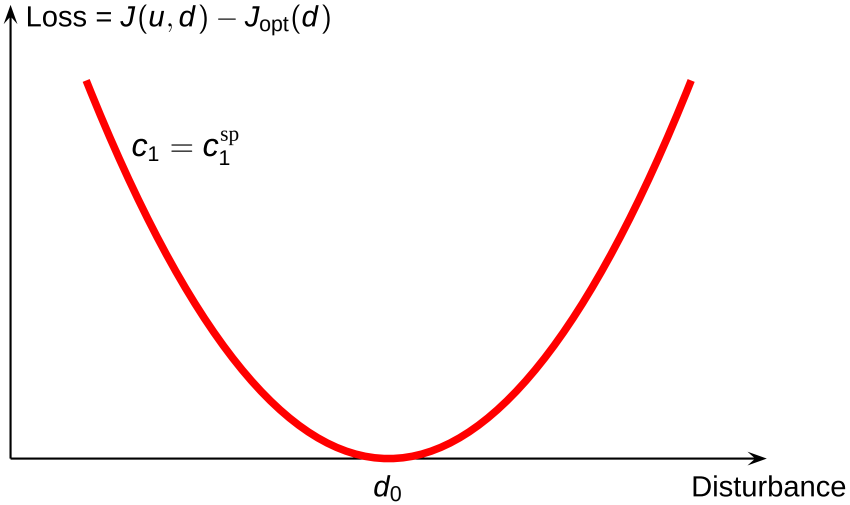


Self-optimizing control

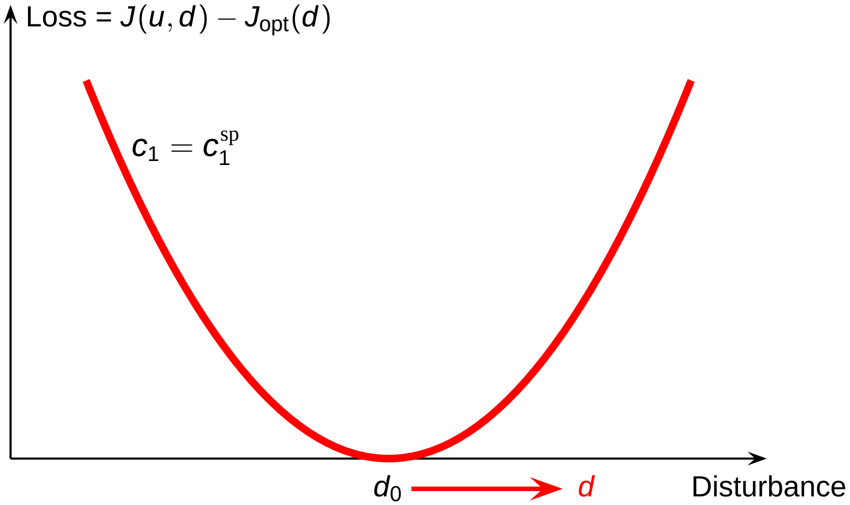
Choice of H such that acceptable operation is achieved with constant setpoints (c_s constant).

- Optimal c_s is invariant with respect to disturbances d
- Insensitive to measurement errors n

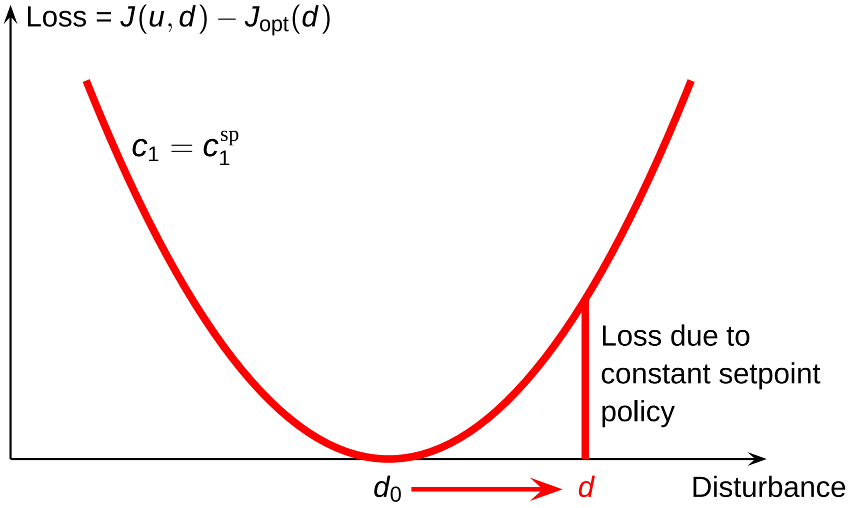
What variables should we control?

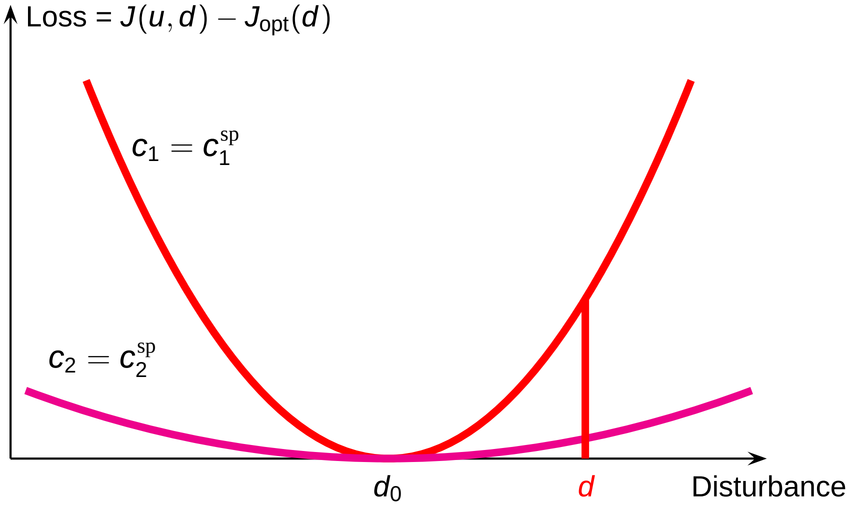


What variables should we control?



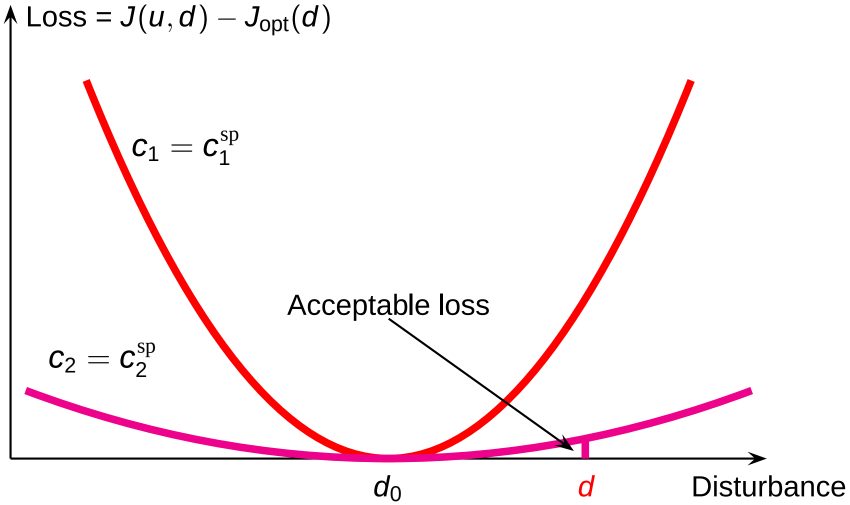
What variables should we control?

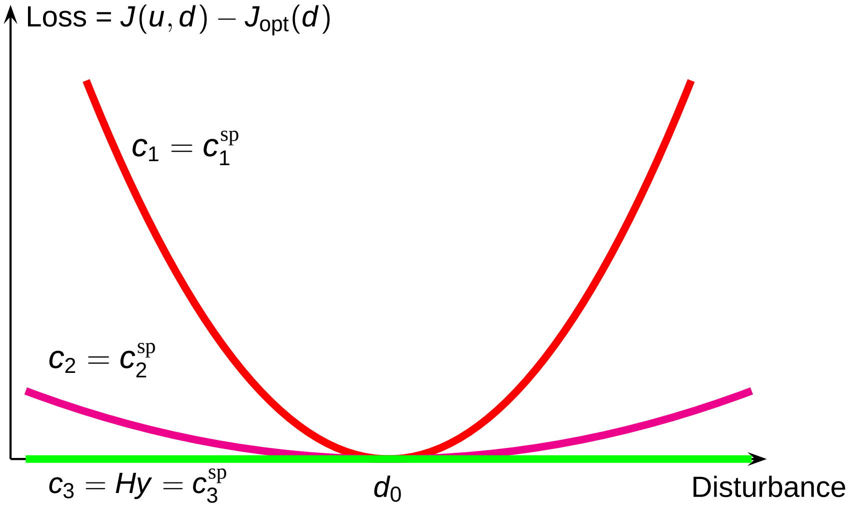




$\text{Loss} = J(u,d) - J_{opt}(d)$
$c_1 = c_1^{sp}$
$c_2 = c_2^{sp}$
d_0
d
Disturbance

What variables should we control?




Loss = $J(u,d) - J_{\text{opt}}(d)$
$c_1 = c_1^{\text{sp}}$
$c_2 = c_2^{\text{sp}}$
$c_3 = Hy = c_3^{\text{sp}}$
d_0
Disturbance

Theorem (Nullspace method for QP)

- *Consider the quadratic problem*

$$\min_u J(u,d) = \begin{bmatrix} u \\ d \end{bmatrix}' \begin{bmatrix} J_{uu} & J_{ud} \\ \star & J_{dd} \end{bmatrix} \begin{bmatrix} u \\ d \end{bmatrix} \qquad (1)$$

- *In addition there are n_y independent measurements*

$$y = G^y u + G^y_d d$$

- *If $n_y \geq n_u + n_d$ there exists an H such that the combinations $c = Hy$ are invariant to the disturbances*
- *H may be found from $HF = 0$, where $F = \frac{\partial y^{\mathrm{opt}}}{\partial d} = -(G^y J_{uu}^{-1} J_{ud} - G^y_d)$*

Alstad and Skogestad, *Ind. Eng. Chem. Res.*, 2007

For a given $x(t)$, one solves the quadratic problem

$$\min_{U=(u_0,u_1,\cdots,u_{N-1})} J(U,x(t)) = x_N^{\mathrm{T}} P x_N + \sum_{k=0}^{N-1} \left[x_k^{\mathrm{T}} Q x_k + u_k^{\mathrm{T}} R u_k \right]$$

subject to

$$\begin{aligned} x_0 &= x(0) \\ x_{k+1} &= Ax_k + Bu_k, \quad k = 0, 1, \cdots, N-1 \\ y_k &= Cx_k, \quad k = 0, 1, \cdots, N \end{aligned}$$

Link between linear-quadratic control and self-optimizing control

Let

$$d = x_0 \quad \text{and} \quad y = \begin{bmatrix} u \\ x \end{bmatrix}$$

The optimal combination

$$c = Hy$$

can be written as the feedback law

$$c = u - (Kx + g)$$

and H (or K) can be obtained from nullspace method

Efficient on-line computation of constrained optimal control

- The results in the following slides are taken from:
  Baotić et al: “Efficient on-line implementation of constrained optimal control”, SIAM Journal of Control and Optimization, 2008.

Efficient on-line computation of constrained optimal control

- The results in the following slides are taken from: Baotić et al: "Efficient on-line implementation of constrained optimal control", SIAM Journal of Control and Optimization, 2008.
- Here we focus on linear-quadratic finite horizon optimal control, i.e. problems that can be written on the form

$$J^*(x) = \frac{1}{2}x'Yx + \min_U \frac{1}{2}U'HU + x'FU \quad \text{s.t. } M^uU \leq M + M^x x \qquad \text{(MPC)}$$

Efficient on-line computation of constrained optimal control

- The results in the following slides are taken from: Baotić et al: “Efficient on-line implementation of constrained optimal control”, SIAM Journal of Control and Optimization, 2008.
- Here we focus on linear-quadratic finite horizon optimal control, i.e. problems that can be written on the form

$$J^*(x) = \frac{1}{2}x'Yx + \min_U \frac{1}{2}U'HU + x'FU \quad \text{s.t. } M^u U \leq M + M^x x \qquad \text{(MPC)}$$

  - Solution: $u(x) = F_i x + G_i, \quad \forall x \in P_i, \quad i = 1, \ldots, N_P$.

Efficient on-line computation of constrained optimal control

- The results in the following slides are taken from: Baotić et al: “Efficient on-line implementation of constrained optimal control”, SIAM Journal of Control and Optimization, 2008.
- Here we focus on linear-quadratic finite horizon optimal control, i.e. problems that can be written on the form

$$J^*(x) = \frac{1}{2}x'Yx + \min_U \frac{1}{2}U'HU + x'FU \quad \text{s.t. } M^u U \leq M + M^x x \qquad \text{(MPC)}$$

  - Solution: $u(x) = F_i x + G_i, \quad \forall x \in P_i, \quad i = 1, \ldots, N_P$.
- Goal: Efficient implementation of the solution to (MPC).

Descriptor functions

Definition (PWA descriptor function)

A scalar continuous real-valued PWA function $f : X_f \mapsto \mathbb{R}$,

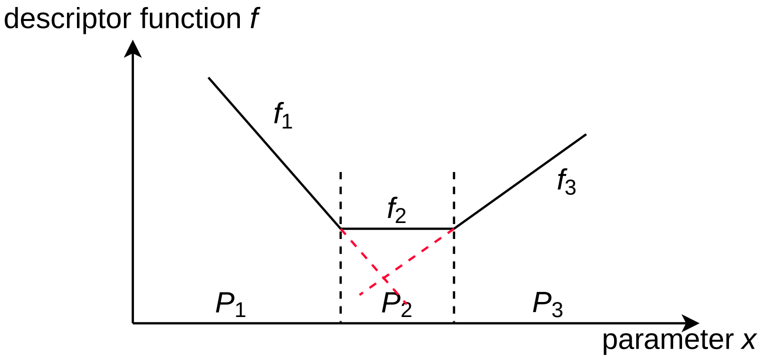
$$f(x) := f_i(x) := a_i' x + b_i \quad \text{if} \quad x \in P_i, \tag{2}$$

with $a_i \in \mathbb{R}^{n_x}$, $b_i \in \mathbb{R}$, *is called a descriptor function if*

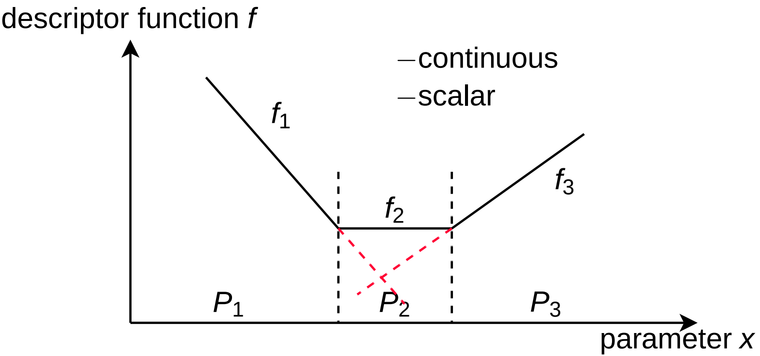
$$a_i \neq a_j, \quad \forall j \in C_i, \quad i = 1, \ldots, N_p, \tag{3}$$

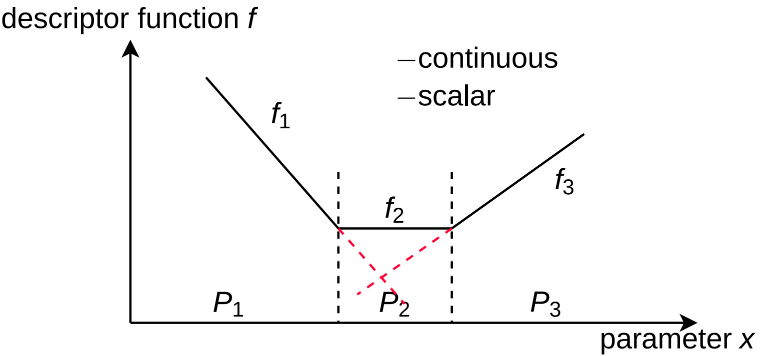
where $\cup_i P_i = X_f \subset \mathbb{R}^{n_x}$, *and* C_i *is the list of neighbors of* P_i.

Descriptor functions



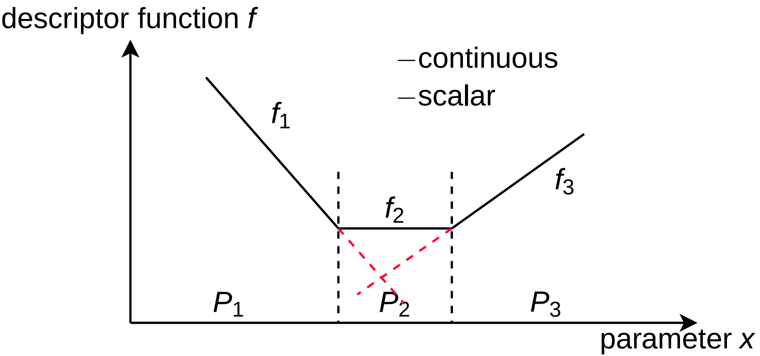
Descriptor functions





Necessary information:

- List of neighbors C_i to each polytope P_i
- List of “correct” signs of corresponding function $f_i - f_j$



Necessary information:

- List of neighbors C_i to each polytope P_i
- List of “correct” signs of corresponding function $f_i - f_j$

From this we can make a global algorithm for the point location problem.

Descriptor functions

Lemma

- *The PWA control law can be used as a (vector-valued) PWA descriptor function.*
- *By taking inner product with a "random" vector w we can make a scalar-valued PWA descriptor function.*

Descriptor functions

Lemma

- *The PWA control law can be used as a (vector-valued) PWA descriptor function.*
- *By taking inner product with a "random" vector w we can make a scalar-valued PWA descriptor function.*

- Variable combinations $\mathrm{inv}^i = H^i y - c_s^i$ from the "nullspace method" can also be used as a scalar PWA descriptor function (by inner product with vector w).

Constrained self-optimizing control

- We consider the following problem:

$$\min_u \frac{1}{2} \begin{bmatrix} u \\ d \end{bmatrix}' \begin{bmatrix} J_{uu} & J_{ud} \\ \star & J_{dd} \end{bmatrix} \begin{bmatrix} u \\ d \end{bmatrix} \qquad \text{(QP)}$$
$$\text{s.t. } M^u u \leq M + M^d d$$

- In addition we have measurements on the form $y = G^y u + G^y_d d$

Constrained self-optimizing control

- We consider the following problem:

$$\min_u \frac{1}{2} \begin{bmatrix} u \\ d \end{bmatrix}' \begin{bmatrix} J_{uu} & J_{ud} \\ \star & J_{dd} \end{bmatrix} \begin{bmatrix} u \\ d \end{bmatrix} \qquad \text{(QP)}$$
$$\text{s.t. } M^u u \leq M + M^d d$$

- In addition we have measurements on the form $y = G^y u + G^y_d d$
- Goal: Find a “self-optimizing” implementation of (QP).
  - Nullspace method
  - Region detection with scalar PWA descriptor function

Constrained self-optimizing control

1: Define objective function and constraints (optimal operation)

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

Constrained self-optimizing control

1: Define objective function and constraints (optimal operation)

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

2: For $d = d_0$ find nominal optimal point u_0

Constrained self-optimizing control

1: Define objective function and constraints (optimal operation)

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

2: For $d = d_0$ find nominal optimal point u_0

3: Approximate the problem as a constrained QP around (u_0, d_0)

$$\min_u \frac{1}{2} \begin{bmatrix} u \\ d \end{bmatrix}' \begin{bmatrix} J_{uu} & J_{ud} \\ \star & J_{dd} \end{bmatrix} \begin{bmatrix} u \\ d \end{bmatrix} \quad \text{s.t. } M^u u \leq M + M^d d$$

Constrained self-optimizing control

1: Define objective function and constraints (optimal operation)

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

2: For $d = d_0$ find nominal optimal point u_0

3: Approximate the problem as a constrained QP around (u_0, d_0)

$$\min_u \frac{1}{2} \begin{bmatrix} u \\ d \end{bmatrix}' \begin{bmatrix} J_{uu} & J_{ud} \\ \star & J_{dd} \end{bmatrix} \begin{bmatrix} u \\ d \end{bmatrix} \quad \text{s.t. } M^u u \leq M + M^d d$$

4: Solve this problem *parametrically*

Constrained self-optimizing control

1: Define objective function and constraints (optimal operation)

$$\min_u J(x,u,d) \quad \text{s.t. } f(x,u,d) = 0, \quad g(x,u,d) \le 0$$

2: For $d = d_0$ find nominal optimal point u_0

3: Approximate the problem as a constrained QP around (u_0, d_0)

$$\min_u \frac{1}{2} \begin{bmatrix} u \\ d \end{bmatrix}' \begin{bmatrix} J_{uu} & J_{ud} \\ \star & J_{dd} \end{bmatrix} \begin{bmatrix} u \\ d \end{bmatrix} \quad \text{s.t. } M^u u \le M + M^d d$$

4: Solve this problem *parametrically*

5: In each region (in the disturbance space), use the nullspace method to find controlled variables $\mathrm{inv}^i := H^i y - c_s^i$, using available measurements $y = G^y u + G_d^y d$.

Constrained self-optimizing control

1: Define objective function and constraints (optimal operation)

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

2: For $d = d_0$ find nominal optimal point u_0

3: Approximate the problem as a constrained QP around (u_0, d_0)

$$\min_u \frac{1}{2} \begin{bmatrix} u \\ d \end{bmatrix}' \begin{bmatrix} J_{uu} & J_{ud} \\ \star & J_{dd} \end{bmatrix} \begin{bmatrix} u \\ d \end{bmatrix} \quad \text{s.t. } M^u u \leq M + M^d d$$

4: Solve this problem *parametrically*

5: In each region (in the disturbance space), use the nullspace method to find controlled variables $\mathrm{inv}^i := H^i y - c_s^i$, using available measurements $y = G^y u + G_d^y d$.

6: Based on the invariants, make a *scalar PWA descriptor function* for region detection.

1: Define objective function and constraints (optimal operation)

$$\min_u J(x,u,d) \quad \text{s.t. } f(x,u,d)=0, \quad g(x,u,d)\le 0$$

2: For $d = d_0$ find nominal optimal point u_0

3: Approximate the problem as a constrained QP around (u_0, d_0)

$$\min_u \frac{1}{2}\begin{bmatrix} u \\ d \end{bmatrix}' \begin{bmatrix} J_{uu} & J_{ud} \\ \star & J_{dd} \end{bmatrix} \begin{bmatrix} u \\ d \end{bmatrix} \quad \text{s.t. } M^u u \le M + M^d d$$

4: Solve this problem *parametrically*

5: In each region (in the disturbance space), use the nullspace method to find controlled variables $\text{inv}^i := H^i y - c_s^i$, using available measurements $y = G^y u + G_d^y d$.

6: Based on the invariants, make a *scalar PWA descriptor function* for region detection.

Constrained self-optimizing control

1: Define objective function and constraints (optimal operation)

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

2: For $d = d_0$ find nominal optimal point u_0

3: Approximate the problem as a constrained QP around (u_0, d_0)

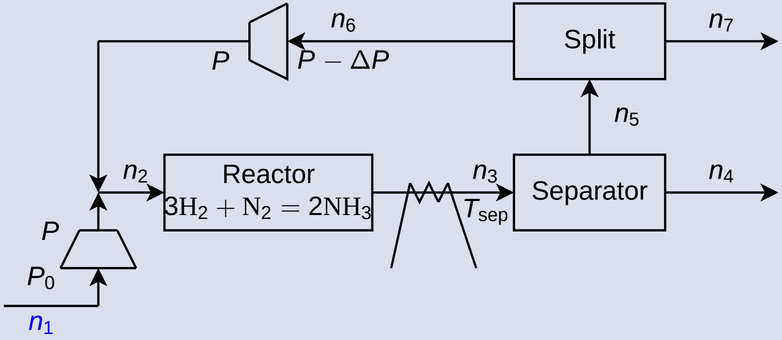
$$\min_u \frac{1}{2}\begin{bmatrix} u \\ d \end{bmatrix}' \begin{bmatrix} J_{uu} & J_{ud} \\ \star & J_{dd} \end{bmatrix} \begin{bmatrix} u \\ d \end{bmatrix} \quad \text{s.t. } M^u u \leq M + M^d d$$

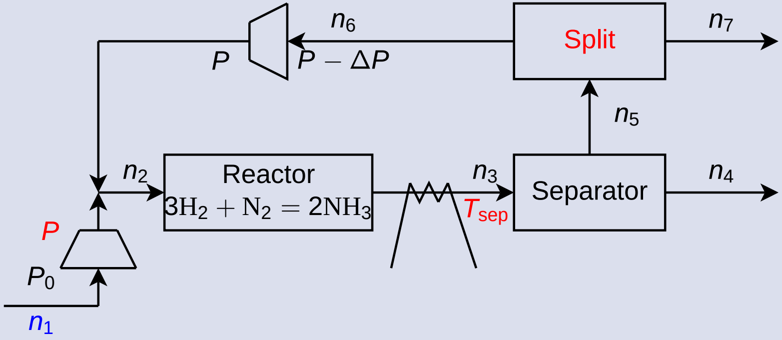
4: Solve this problem *parametrically*

5: In each region (in the disturbance space), use the nullspace method to find controlled variables $\mathrm{inv}^i := H^i y - c_s^i$, using available measurements $y = G^y u + G_d^y d$.

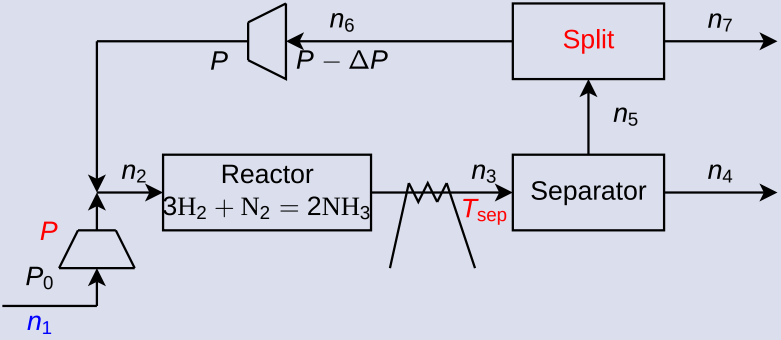
6: Based on the invariants, make a *scalar PWA descriptor function* for region detection.

Example: Ammonia production
Problem formulation



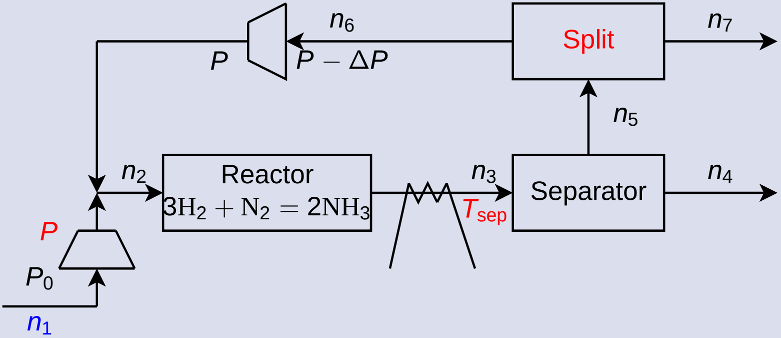

n_6
n_7
Split
P
$P - \Delta P$
n_5
n_2
Reactor
$3H_2 + N_2 = 2NH_3$
n_3
T_{sep}
Separator
n_4
P
P_0
n_1

PROBLEM FORMULATION



- cost function:
  $J = p_1 e' n_1 R T_1 \ln(P/P_0) + p_2 e' n_6 R T_{sep} \ln(P/(P - \Delta P)) + p_3 e' n_3 C p (T_0 - T_{sep})(\frac{T_0}{T_c} - 1) - p_4 n_{4,\mathrm{NH_3}}$

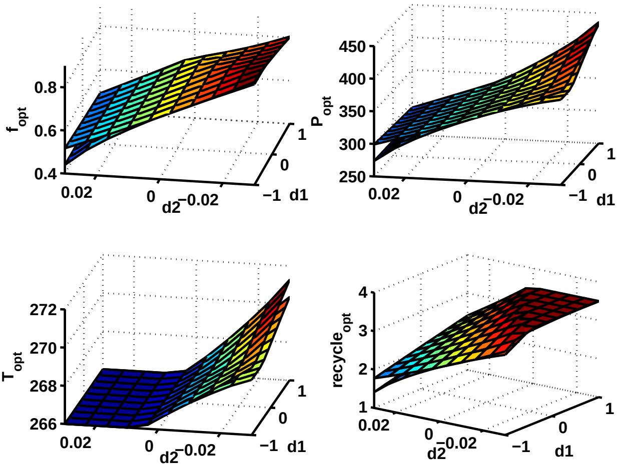
Problem formulation



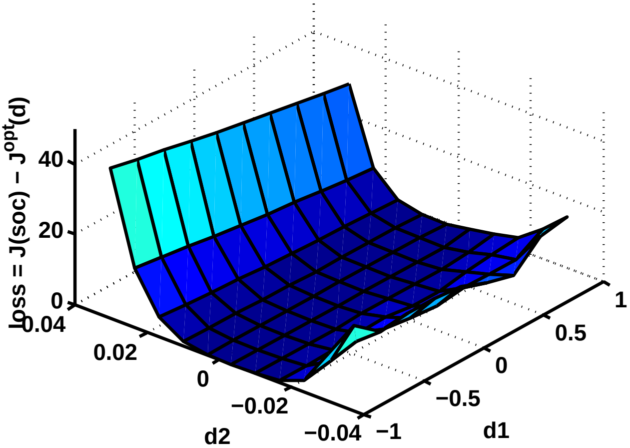
- cost function:
  $J = p_1 e' n_1 RT_1 \ln(P/P_0) + p_2 e' n_6 RT_{sep} \ln(P/(P - \Delta P)) + p_3 e' n_3 Cp(T_0 - T_{sep})(\frac{T_0}{T_c} - 1) - p_4 n_{4,\mathrm{NH_3}}$
- constraints: $T_{sep} \geq T_{sep}^{min}$, $e' n_6 \leq r_{max}$.

Example: Ammonia production

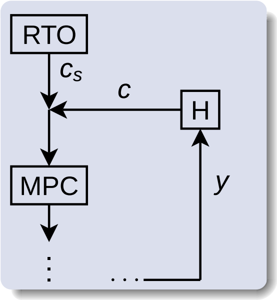
Results



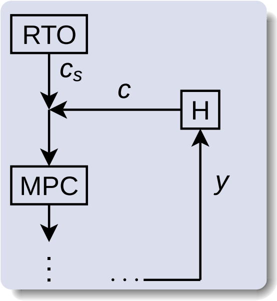
Example: Ammonia production
RESULTS



Degree of freedom: Controlled variables



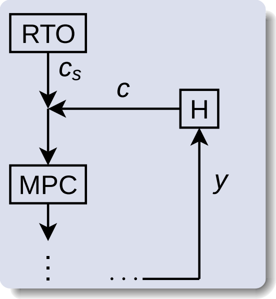
Degree of freedom: Controlled variables



Alt. 1 H fixed at “random” value, typically $c = (\bar{y}, \bar{u})$.

Alt. 2 H optimized, but constant.

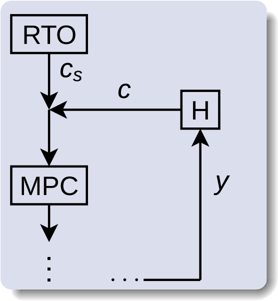
Alt. 3 H is a function of operating condition.



Alt. 1 H fixed at “random” value, typically $c = (\bar{y}, \bar{u})$.

Alt. 2 H optimized, but constant.

Alt. 3 H is a function of operating condition.

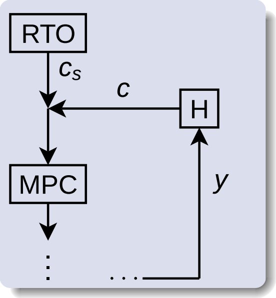


Alt. 1 H fixed at "random" value, typically $c = (\bar{y}, \bar{u})$.

Alt. 2 H optimized, but constant.

Alt. 3 H is a function of operating condition.

Degree of freedom: Controlled variables

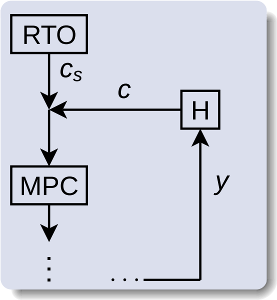


Alt. 1 H fixed at "random" value, typically $c = (\bar{y}, \bar{u})$.

Alt. 2 H optimized, but constant.

Alt. 3 H is a function of operating condition.

Degree of freedom: Controlled variables

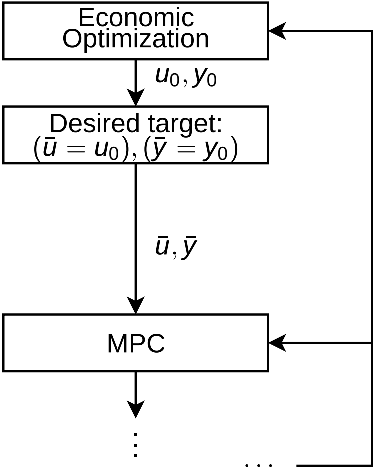


Alt. 1 H fixed at "random" value, typically $c = (\bar{y}, \bar{u})$.

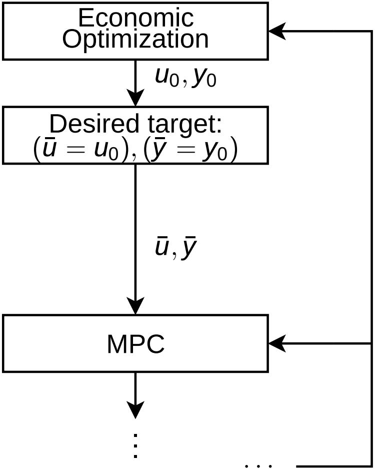
Alt. 2 H optimized, but constant.

Alt. 3 H is a function of operating condition.

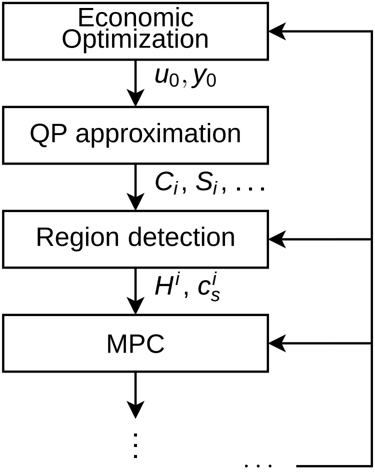
What should we do with constrained SOC?



Current situation.



Current situation.



Self-optimizing around (u_0, y_0) but also if active set changes

Conclusions

- Efficient implementation of MPC and nullspace method gives constrained SOC

Conclusions

- Efficient implementation of MPC and nullspace method gives constrained SOC
- The proposed method is exact for linear processes with quadratic objectives, but may be used for general plants and objectives.

Conclusions

- Efficient implementation of MPC and nullspace method gives constrained SOC
- The proposed method is exact for linear processes with quadratic objectives, but may be used for general plants and objectives.
- Discussed in report: Measured nonlinearities (such as active constraints) may be accounted for by adding extra disturbances.

– Target calculation:

$$\min_{x_s, u_s, \eta} \frac{1}{2}(\eta' Q_s \eta + (u_s - \bar{u}) R_s (u_s - \bar{u})) + q_s' \eta$$

subject to the constraints

$$\begin{bmatrix} I - A & -B & 0 \\ C & 0 & I \\ C & 0 & -I \end{bmatrix} \begin{bmatrix} x_s \\ u_s \\ \eta \end{bmatrix} \begin{Bmatrix} = \\ \geq \\ \leq \end{Bmatrix} \begin{bmatrix} Bd \\ \bar{y} - p \\ \bar{y} - p \end{bmatrix}$$

$$\eta \geq 0$$

$$u_{\min} \leq Du_s \leq u_{\max}, \; y_{\min} \leq Cx_s + p \leq y_{\max}$$

Taken from: J.B. Rawlings: *Tutorial Overview of Model Predictive Control*, IEEE Contr. Sys. Mag. June 2000

Linear-Quadratic Model Predictive Control

– Target calculation:

$$\min_{x_s, u_s, \eta} \frac{1}{2}(\eta' Q_s \eta + (u_s - \bar{u}) R_s (u_s - \bar{u})) + q_s' \eta$$

subject to the constraints

$$\begin{bmatrix} I - A & -B & 0 \\ C & 0 & I \\ C & 0 & -I \end{bmatrix} \begin{bmatrix} x_s \\ u_s \\ \eta \end{bmatrix} \left\{ \begin{matrix} = \\ \geq \\ \leq \end{matrix} \right\} \begin{bmatrix} Bd \\ \bar{y} - p \\ \bar{y} - p \end{bmatrix}$$

$$\eta \geq 0$$

$$u_{\min} \leq Du_s \leq u_{\max}, \; y_{\min} \leq Cx_s + p \leq y_{\max}$$

– Receding Horizon Controller (RHC):
Control the process towards (x_s, u_s) in a “constrained LQR”-way

Taken from: J.B. Rawlings: *Tutorial Overview of Model Predictive Control*, IEEE Contr. Sys. Mag. June 2000

Linear-Quadratic Model Predictive Control

– Target calculation:

$$\min_{x_s, u_s, \eta} \frac{1}{2}(\eta' Q_s \eta + (u_s - \bar{u}) R_s (u_s - \bar{u})) + q_s' \eta$$

subject to the constraints

$$\begin{bmatrix} I-A & -B & 0 \\ C & 0 & I \\ C & 0 & -I \end{bmatrix} \begin{bmatrix} x_s \\ u_s \\ \eta \end{bmatrix} \begin{Bmatrix} = \\ \geq \\ \leq \end{Bmatrix} \begin{bmatrix} Bd \\ \bar{y} - p \\ \bar{y} - p \end{bmatrix}$$

$$\eta \geq 0$$

$$u_{\min} \leq Du_s \leq u_{\max}, \; y_{\min} \leq Cx_s + p \leq y_{\max}$$

– Receding Horizon Controller (RHC):
Control the process towards (x_s, u_s) in a “constrained LQR”-way

– State estimator:
Need this to get $x(t)$, which is the “parameter” driving the RHC, and (p, d) which are giving integral action and driving the target calculator.

Taken from: J.B. Rawlings: *Tutorial Overview of Model Predictive Control*, IEEE Contr. Sys. Mag. June 2000