A New Approach to Explicit MPC Using Self-Optimizing Control

Henrik Manum, Sridharakumar Narasimhan, Sigurd Skogestad

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

American Control Conference, Seattle, 2008

NTNU

Outline

1. Optimal operation paradigms
2. Self optimizing control
3. Explicit MPC
4. Link between the two
5. Examples
6. Extensions

Outline

1. Optimal operation paradigms
2. Self optimizing control
3. Explicit MPC
4. Link between the two
5. Examples
6. Extensions

Outline

1. Optimal operation paradigms
2. Self optimizing control
3. Explicit MPC
4. Link between the two
5. Examples
6. Extensions

Outline

1. Optimal operation paradigms
2. Self optimizing control
3. Explicit MPC
4. Link between the two
5. Examples
6. Extensions

Outline

1. Optimal operation paradigms
2. Self optimizing control
3. Explicit MPC
4. Link between the two
5. Examples
6. Extensions

Outline

1. Optimal operation paradigms
2. Self optimizing control
3. Explicit MPC
4. Link between the two
5. Examples
6. Extensions

Outline

Outline

Implementation of optimal operation using off-line computations [Narasimhan and Skogestad(2007)]

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 [Narasimhan and Skogestad(2007)]

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.

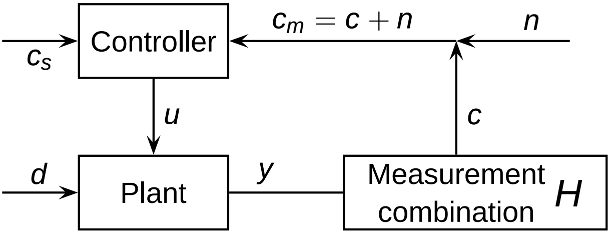
Example: Classical (implicit) MPC.

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.

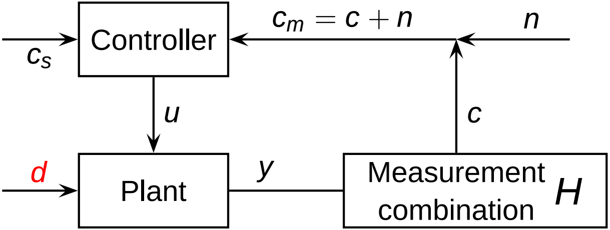
What variables should we control?



Self-optimizing control

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

What variables should we control?

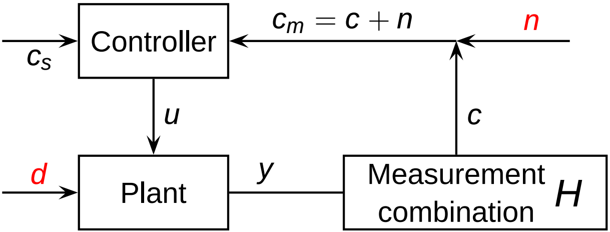


Self-optimizing control

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?

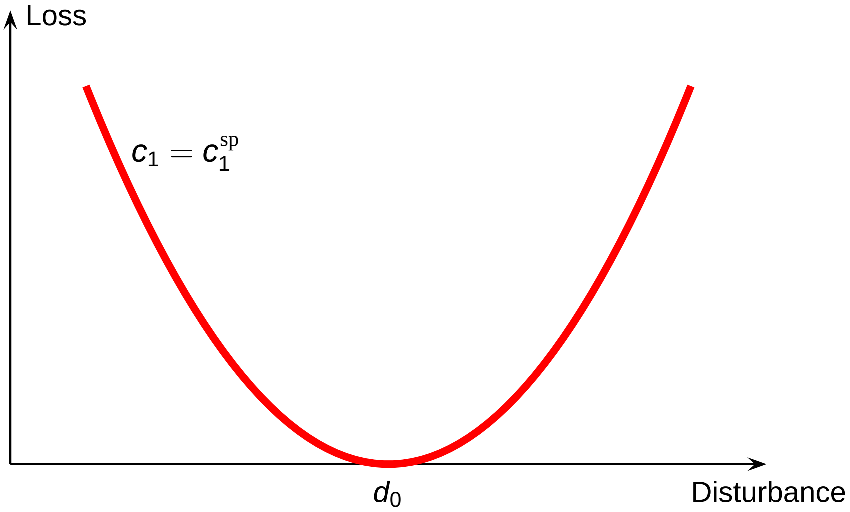


Self-optimizing control

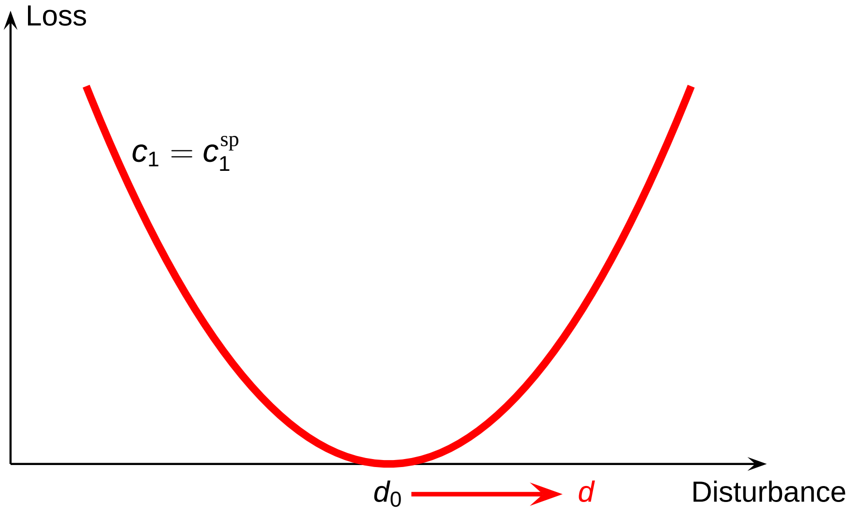
Choice of H such that acceptable operation is achieved with constant setpoints (c_s constant).

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

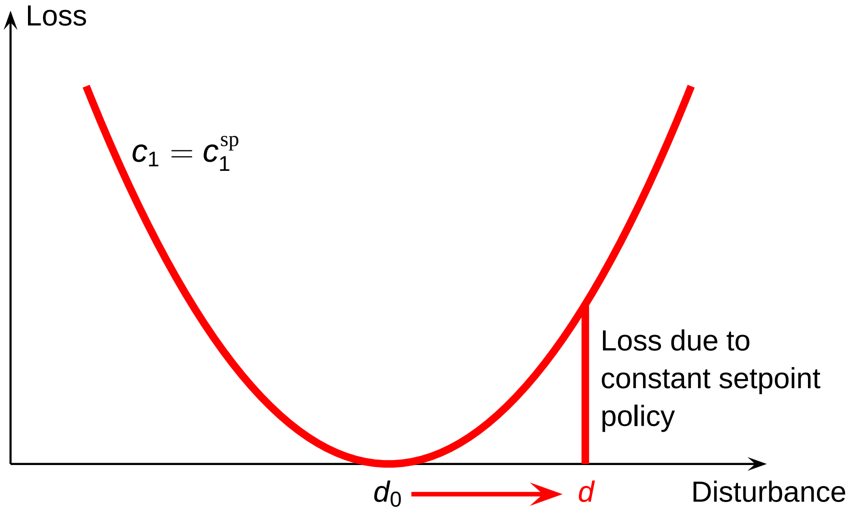
What variables should we control?



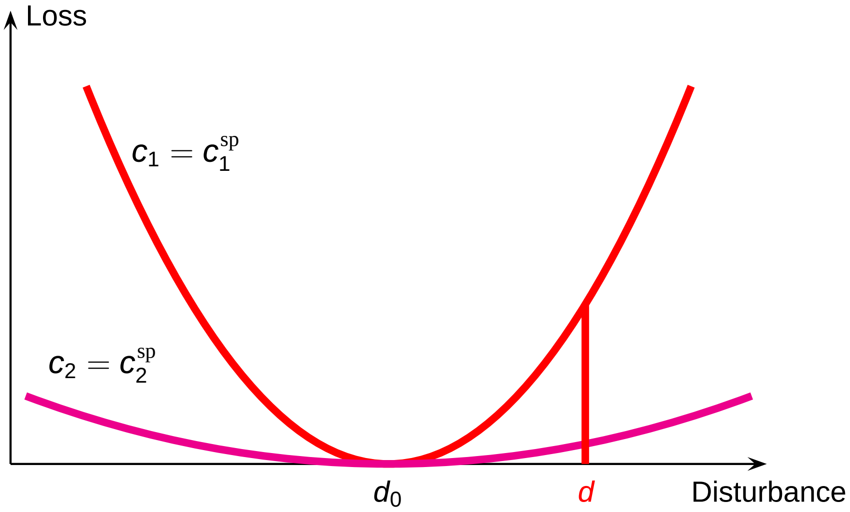
What variables should we control?



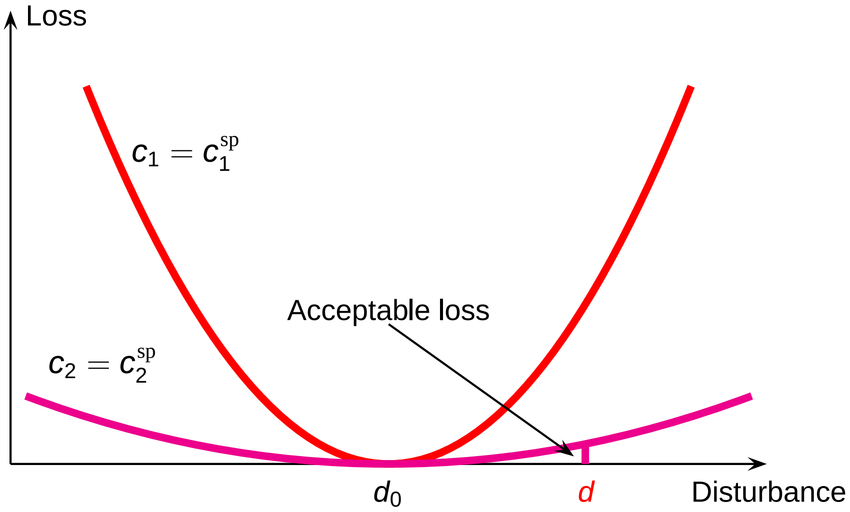
What variables should we control?



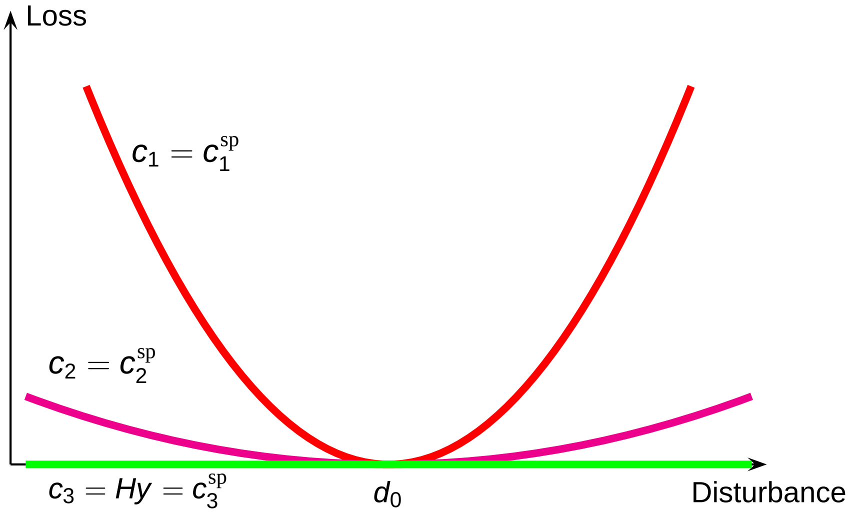
What variables should we control?



What variables should we control?



What variables should we control?



Nullspace method for QP problems

The nullspace method in [Alstad and Skogestad(2007)] is restated for QP's

Theorem (Nullspace method for QP)

- *Consider the quadratic problem*

$$\min_u J = \begin{bmatrix} u & d \end{bmatrix} \begin{bmatrix} J_{uu} & J_{ud} \\ J_{ud}^{\mathrm{T}} & J_{dd} \end{bmatrix} \begin{bmatrix} u \\ d \end{bmatrix} \tag{1}$$

  If there exists $n_y \geq n_u + n_d$ independent measurements, then the optimal solution to (1) has the property that there exists variable combinations $c = Hy$ that are invariant to the disturbances d.

- *H may be found from $HF = 0$, where $F = \frac{\partial y^{opt}}{\partial d^{\mathrm{T}}}$*

Classical (implicit) MPC

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

$$\min_{U} J(U, x(t)) = x_{t+N_y}^{\mathrm{T}} P x_{t+N_y} + \sum_{k=0}^{N_y-1} \left[x_{t+k}^{\mathrm{T}} Q x_{t+k} + u_{t+k}^{\mathrm{T}} R u_{t+k} \right]$$

subject to

- Output and input constraints.
- Discrete model

$$\begin{aligned}
x_t &= x(t) & \\
x_{t+k+1} &= A x_{t+k} + B u_{t+k}, & k \geq 0 \\
y_{t+k} &= C x_{t+k}, & k \geq 0 \\
u_{t+k} &= K x_{t+k}, & N_u \leq k \leq N_y
\end{aligned}$$

Explicit MPC [Bemporad et al.(2002)]

- The “classical” MPC problem can, by substitution, be written as a quadratic problem:

$$\min_{U} J(U, x(t)) = \begin{bmatrix} U^{\mathrm{T}} & x(t)^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} H & F \\ H & Y \end{bmatrix} \begin{bmatrix} U \\ x(t) \end{bmatrix}$$

$$\text{s.t. } GU \leq W + Ex(t)$$

- The initial state $x(t)$ is considered to be a parameter and a parametric program is solved.
- The solution of the parametric program gives regions in the state space.
- Given an algorithm for deciding the current region (i), one implements a continuous piece-wise affine control law

$$u = F^i x + g^i.$$

Link between explicit MPC and self-optimizing control

Let

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

The optimal combination

$$c = Hy$$

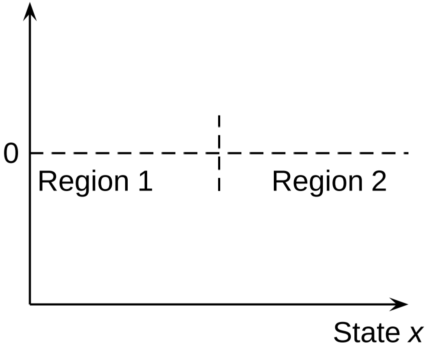
can be written as the feedback law

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

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

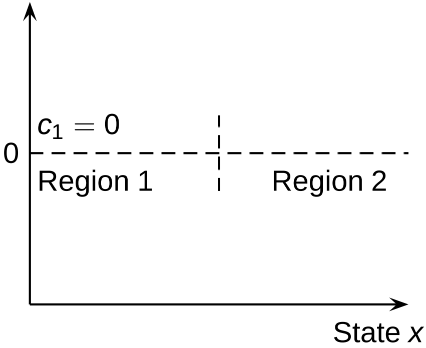
When to switch region?

- The invariants can be used to track region changes
- By monitoring neighboring regions we switch regions when $c_i - c_j$ changes sign



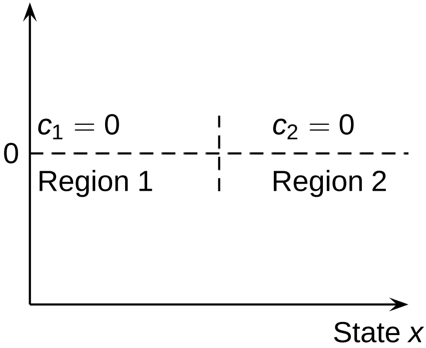
When to switch region?

- The invariants can be used to track region changes
- By monitoring neighboring regions we switch regions when $c_i - c_j$ changes sign



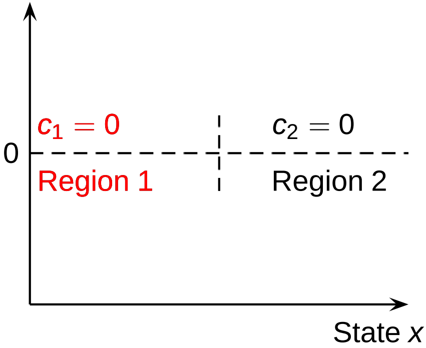
When to switch region?

- The invariants can be used to track region changes
- By monitoring neighboring regions we switch regions when $c_i - c_j$ changes sign



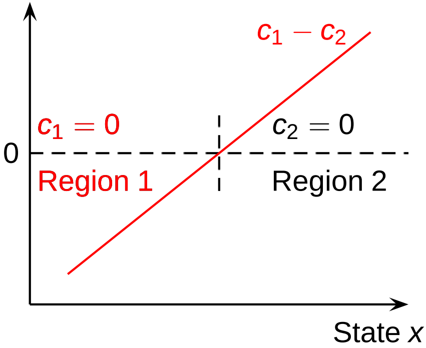
When to switch region?

- The invariants can be used to track region changes
- By monitoring neighboring regions we switch regions when $c_i - c_j$ changes sign



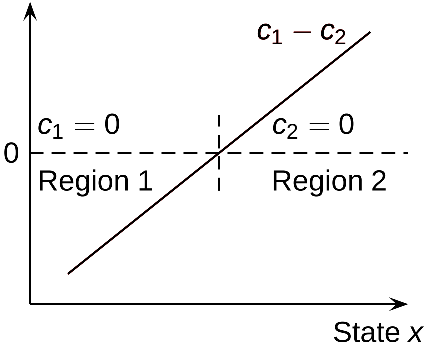
When to switch region?

- The invariants can be used to track region changes
- By monitoring neighboring regions we switch regions when $c_i - c_j$ changes sign



When to switch region?

- The invariants can be used to track region changes
- By monitoring neighboring regions we switch regions when $c_i - c_j$ changes sign



Examples

- We will now show 3 examples. These are based on the explicit MPC paper by [Bemporad et al.(2002)].

Examples

- We will now show 3 examples. These are based on the explicit MPC paper by [Bemporad et al.(2002)].
- Problem formulation:

$$\min_{u_t, u_{t+1}, \cdots, u_{t+N_u-1}} x_{t+N_u}^{\mathrm{T}} P x_{t+N_u} + \sum_{k=0}^{N_u-1} x_{t+k}^{\mathrm{T}} Q x_{t+k} + u_{t+k}^{\mathrm{T}} R u_{y+k}$$

$$\text{s.t. } x_k = x(t)$$

$$x_{k+1} = Ax_k + Bu_k$$

input constraints

state constraints

Examples

- We will now show 3 examples. These are based on the explicit MPC paper by [Bemporad et al.(2002)].
- Problem formulation:

$$\min_{u_t, u_{t+1}, \cdots, u_{t+N_u-1}} x_{t+N_u}^{\mathrm{T}} P x_{t+N_u} + \sum_{k=0}^{N_u-1} x_{t+k}^{\mathrm{T}} Q x_{t+k} + u_{t+k}^{\mathrm{T}} R u_{y+k}$$

$$\begin{aligned} \text{s.t. } x_k &= x(t) \\ x_{k+1} &= A x_k + B u_k \\ &\text{input constraints} \\ &\text{state constraints} \end{aligned}$$

- Examples:
  1. SISO system $y = \frac{1}{s^2+3s+2} u$ subject to input constraint.
  2. Same system with additional state constraint.
  3. Double integrator $y = \frac{1}{s^2} u$ with input constraint.

Examples

- We will now show 3 examples. These are based on the explicit MPC paper by [Bemporad et al.(2002)].
- Problem formulation:

$$\min_{u_t, u_{t+1}, \cdots, u_{t+N_u-1}} x_{t+N_u}^{\mathrm{T}} P x_{t+N_u} + \sum_{k=0}^{N_u-1} x_{t+k}^{\mathrm{T}} Q x_{t+k} + u_{t+k}^{\mathrm{T}} R u_{y+k}$$

$$\text{s.t. } x_k = x(t)$$

$$x_{k+1} = A x_k + B u_k$$

input constraints

state constraints

- Examples:
  1. SISO system $y = \frac{1}{s^2+3s+2} u$ subject to input constraint.
  2. Same system with additional state constraint.
  3. Double integrator $y = \frac{1}{s^2} u$ with input constraint.

Examples

- We will now show 3 examples. These are based on the explicit MPC paper by [Bemporad et al.(2002)].
- Problem formulation:

$$\min_{u_t, u_{t+1}, \cdots, u_{t+N_u-1}} x_{t+N_u}^{\mathrm{T}} P x_{t+N_u} + \sum_{k=0}^{N_u-1} x_{t+k}^{\mathrm{T}} Q x_{t+k} + u_{t+k}^{\mathrm{T}} R u_{y+k}$$

$$\text{s.t. } x_k = x(t)$$

$$x_{k+1} = Ax_k + Bu_k$$

input constraints

state constraints

- Examples:
  1. SISO system $y = \frac{1}{s^2+3s+2} u$ subject to input constraint.
  2. Same system with additional state constraint.
  3. Double integrator $y = \frac{1}{s^2} u$ with input constraint.

Examples

- We will now show 3 examples. These are based on the explicit MPC paper by [Bemporad et al.(2002)].
- Problem formulation:

$$\min_{u_t, u_{t+1}, \cdots, u_{t+N_u-1}} x_{t+N_u}^{\mathrm{T}} P x_{t+N_u} + \sum_{k=0}^{N_u-1} x_{t+k}^{\mathrm{T}} Q x_{t+k} + u_{t+k}^{\mathrm{T}} R u_{y+k}$$

$$\text{s.t. } x_k = x(t)$$

$$x_{k+1} = A x_k + B u_k$$

input constraints

state constraints

- Examples:
  1. SISO system $y = \frac{1}{s^2+3s+2} u$ subject to input constraint.
  2. Same system with additional state constraint.
  3. Double integrator $y = \frac{1}{s^2} u$ with input constraint.

Examples

- We will now show 3 examples. These are based on the explicit MPC paper by [Bemporad et al.(2002)].
- Problem formulation:

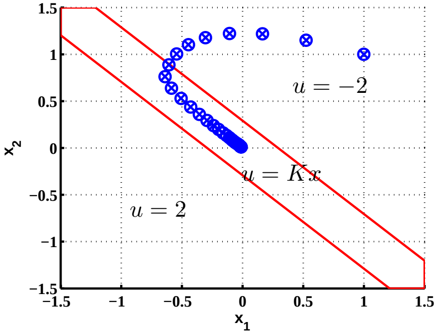
$$\min_{u_t, u_{t+1}, \cdots, u_{t+N_u-1}} x_{t+N_u}^{\mathrm{T}} P x_{t+N_u} + \sum_{k=0}^{N_u-1} x_{t+k}^{\mathrm{T}} Q x_{t+k} + u_{t+k}^{\mathrm{T}} R u_{y+k}$$

$$\begin{aligned} \text{s.t. } x_k &= x(t) \\ x_{k+1} &= A x_k + B u_k \\ &\text{input constraints} \\ &\text{state constraints} \end{aligned}$$

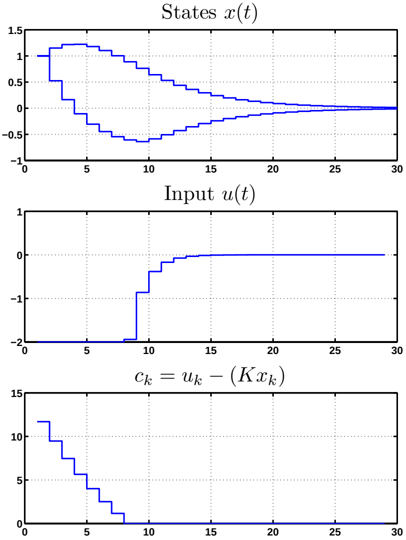
- Examples:
  1. SISO system $y = \frac{1}{s^2+3s+2} u$ subject to input constraint.
  2. Same system with additional state constraint.
  3. Double integrator $y = \frac{1}{s^2} u$ with input constraint.

Example 1: Simple SISO system

SISO with input constraint.



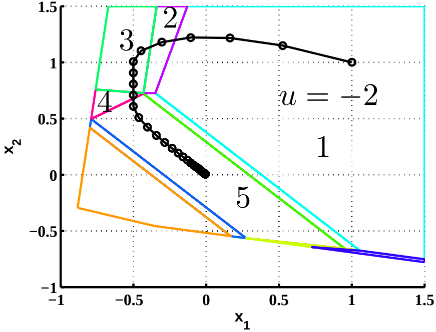
State space partition and simulation from $x_0 = (1, 1)$.



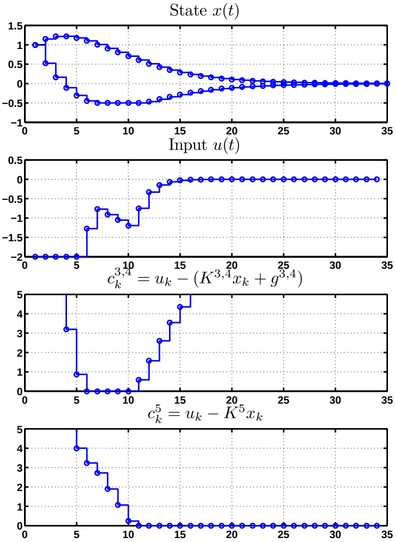
Example 2: Same system with state constraint

Additional constraint

$x_{k+1,k+2} \geq -0.5.$

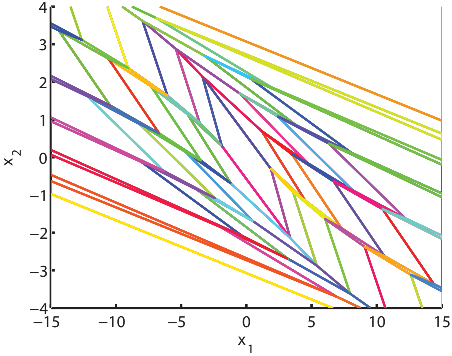


State space partition and simulation from $x_0 = (1, 1)$.



Example 3: Double integrator $y = \frac{1}{s^2}u$ with $N_u = 6$

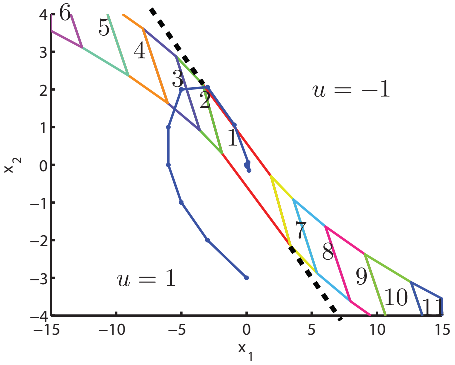
- We merge all regions were the first input is the same.



Bemporad et al.: 57 regions (from 73)

Example 3: Double integrator $y = \frac{1}{s^2}u$ with $N_u = 6$

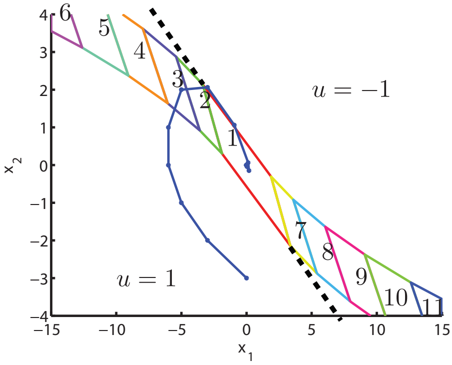
- We merge all regions were the first input is the same.



11 regions

Example 3: Double integrator $y = \frac{1}{s^2}u$ with $N_u = 6$

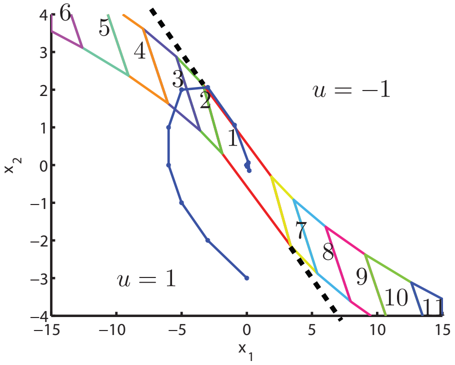
- We merge all regions were the first input is the same.
- The resulting merged regions were in this case not convex, but,. . .



11 regions

Example 3: Double integrator $y = \frac{1}{s^2}u$ with $N_u = 6$

- We merge all regions were the first input is the same.
- The resulting merged regions were in this case not convex, but,. . .
- . . . taking directions into account the “relevant” boundaries form a convex polyhedron.



11 regions

Summary of the examples

Examples 1,2

- The optimal solution for each active set can be used both for control and tracking the active region.

Summary of the examples

Examples 1,2

- The optimal solution for each active set can be used both for control and tracking the active region.

Example 3: Double integrator

- Problem: Not all merged regions were convex.
- Fix: Used the direction of the process and observed that the relevant boundaries make the regions convex.
- Storage
  - Original partition: 73 regions
  - Bemporad et. al., 2002: need to store 57 regions.
  - Our approach: 11 regions/boundaries needs to be stored.

Extensions

Extensions

1. Include measurement error in explicit MPC.
2. Extend the results to output feedback ($u = Ky$).
3. Explicit expressions for fixed low-order controllers, e.g., MIMO-PID

Extensions

Extensions

1. Include measurement error in explicit MPC.
2. Extend the results to output feedback ($u = Ky$).
3. Explicit expressions for fixed low-order controllers, e.g., MIMO-PID

Extensions

Extensions

1. Include measurement error in explicit MPC.
2. Extend the results to output feedback ($u = Ky$).
3. Explicit expressions for fixed low-order controllers, e.g., MIMO-PID

Extensions

Extensions

1. Include measurement error in explicit MPC.
2. Extend the results to output feedback ($u = Ky$).
3. Explicit expressions for fixed low-order controllers, e.g., MIMO-PID

Extensions

Extensions

1. Include measurement error in explicit MPC.
2. Extend the results to output feedback ($u = Ky$).
3. Explicit expressions for fixed low-order controllers, e.g., MIMO-PID

See IFAC World Congress, Seoul, July 2008

Conclusion

- MPC: Quadratic optimization problem at each time sample
- From link to self-optimizing control: Must exist invariants $c = Hy$
- Correspond to feedback law: $u = Kx + g$
- Gives new insights
  - Simple feedback solution to MPC must exist (explicit MPC)
  - Region changes identified by tracking variables c from neighboring regions
  - Low-order output feedback, etc.

References

- V. Alstad and S. Skogestad.
  Extended nullspace method for selecting measurement combinations as controlled variables for optimal steady-state operation.
  Submitted to Journal of Process Control, 2007.

- A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos.
  The explicit linear quadratic regulator for constrained systems.
  Automatica, 38:3–20, 2002.
  See also corrigendum 39(2003), pages 1845-1846.

- H. Manum, S. Narasimhan, and S. Skogestad.
  A new approach to explicit mpc using self-optimizing control. (Internal report.)
  Available at home page of Sigurd Skogestad (google!): http://www.nt.ntnu.no/users/skoge/publications/2007/.

- S. Narasimhan and S. Skogestad.
  Implementation of optimal operation using off-line calculations.
  In *Dycops*, 2007.