Milano, August 28th, 2011

Design of hierarchical and distributed MPC control systems with robustness tools

Marcello Farina, Bruno Picasso, Riccardo Scattolini
DEI-Politecnico di Milano

HD-MPC

SEVENTH FRAMEWORK PROGRAMME

Outline

1. Introduction

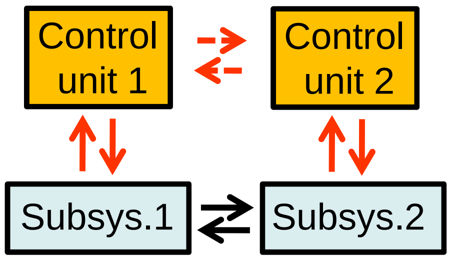
2. Hierarchical MPC systems
   - Basic architecture
   - Extensions (performance & reconfigurability)
   - Conclusions

3. Distributed MPC systems
   - A “tube-based”, non cooperative DMPC algorithm
   - Conclusions

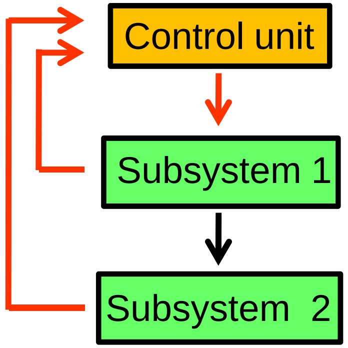
4. Concluding remark

Motivations

Motivations for **distributed** / **hierarchical** control:

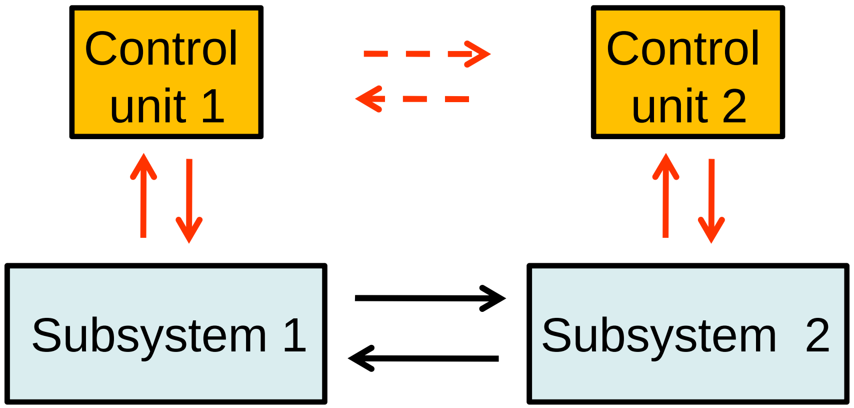


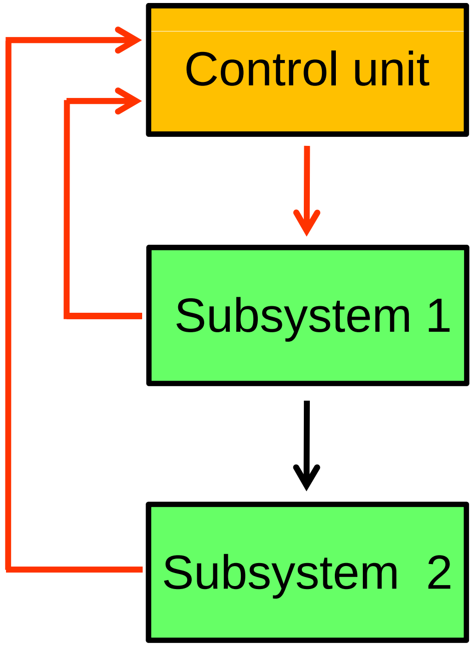
- Reduce the computational load
- Reduce the communication load
- Improve the robustness with respect to failures
  - in the transmission of information
  - in the central control unit
- Improve the modularity and the flexibility of the system
- Consider different goals at different time scales (Real-Time Optimization)
- Synchronize subsystems working at different time scales



There has hence been a long time interest for decentralized / distributed **[Siljak '78… '91]** and hierarchical control **[Mesarovic '70, Findeisen '80, …]** for **large-scale** and **complex systems**.
Recent contributions include: **[Engell '07, Tatjewski '08** and **Scattolini '09 -** *"An overview on distributed and hierarchical MPC"***]**).

The proposed approach





In both **distributed** and **hierarchical** structures, there are two possible approaches to the control synthesis allowing to deal with the interacting subsystems:

1. Game theory
2. **Robust control**

Outline

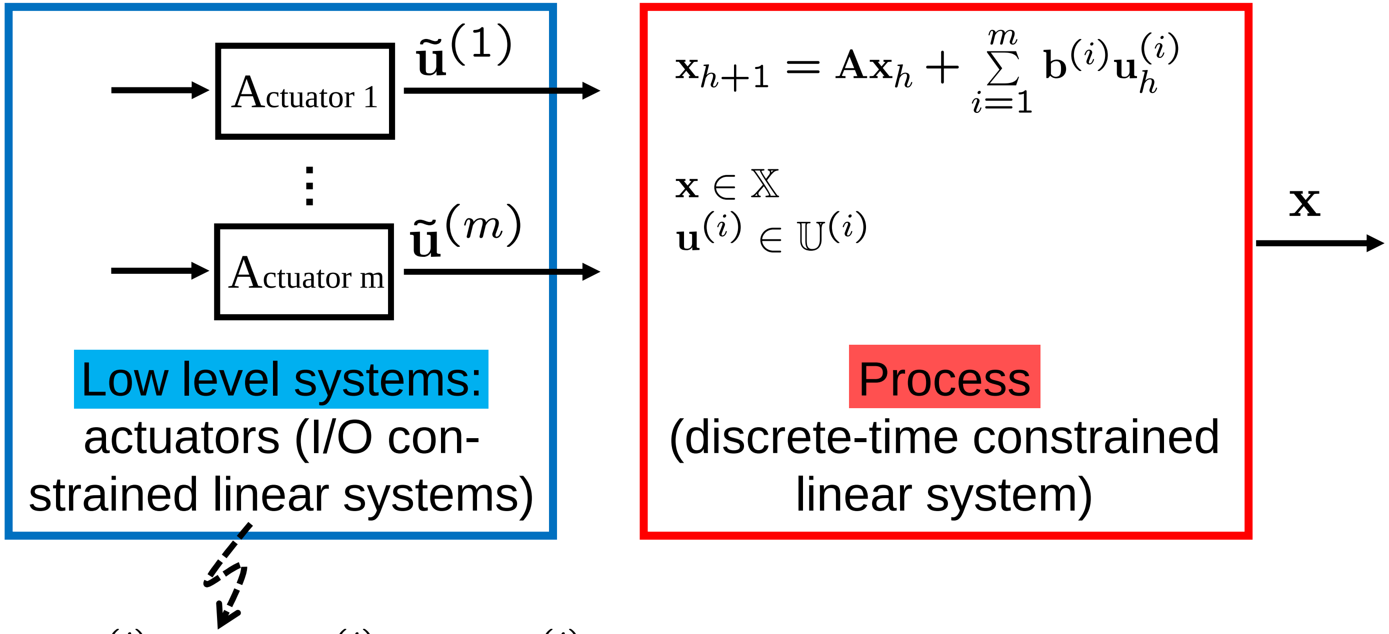
1. Introduction

2. **Hierarchical MPC systems**
   - **Basic architecture**
   - **Extensions (performance & reconfigurability)**
   - **Conclusions**

3. Distributed MPC systems
   - A “tube-based”, non cooperative DMPC algorithm
   - Conclusions

4. Concluding remark

Hierarchical MPC systems: basic architecture



$$\zeta_{h+1}^{(i)} = \mathbf{F}^{(i)}\zeta_h^{(i)} + \mathbf{G}^{(i)}\nu_h^{(i)}$$
$$\tilde{\mathbf{u}}_h^{(i)} = \mathbf{H}^{(i)}\zeta_h^{(i)}$$

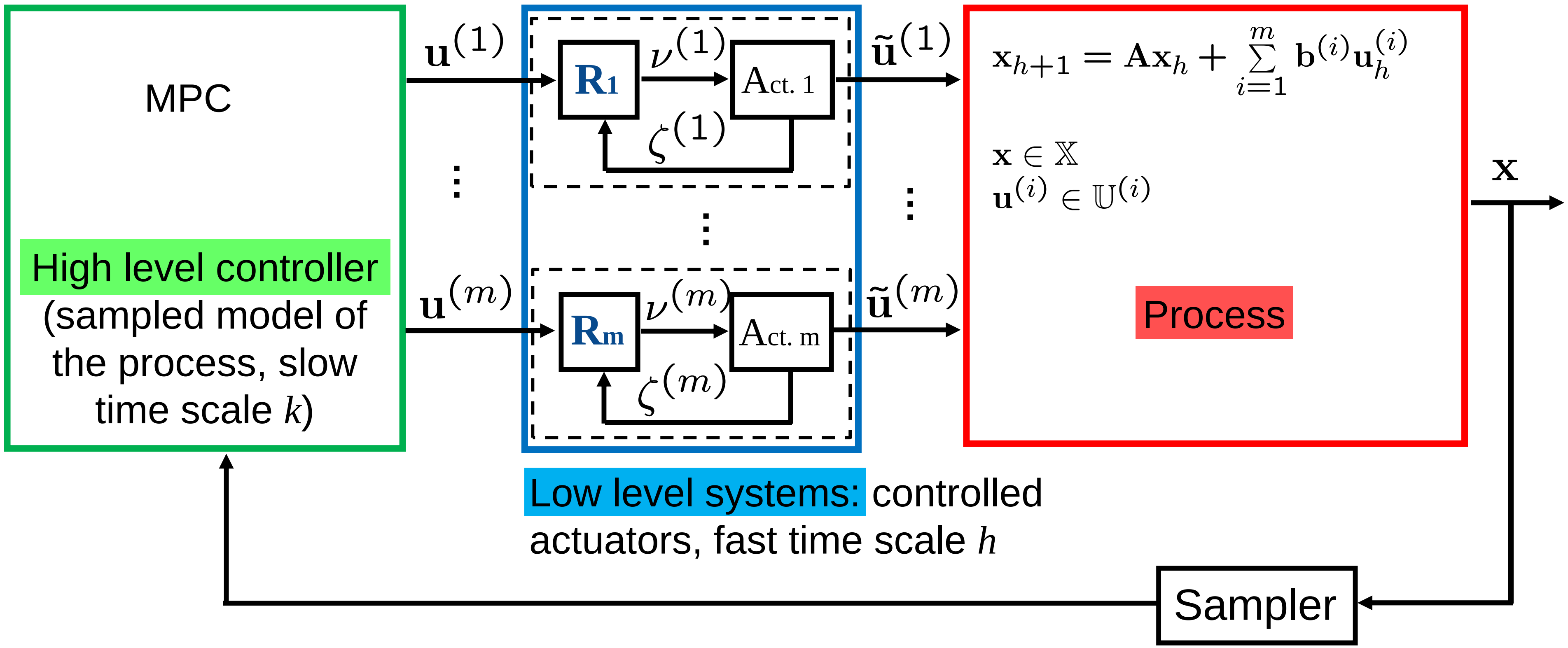
$$\zeta^{(i)} \in \mathcal{Z}^{(i)}$$
$$\nu^{(i)} \in \mathbb{V}^{(i)}$$

Control goal: state-feedback stabilization

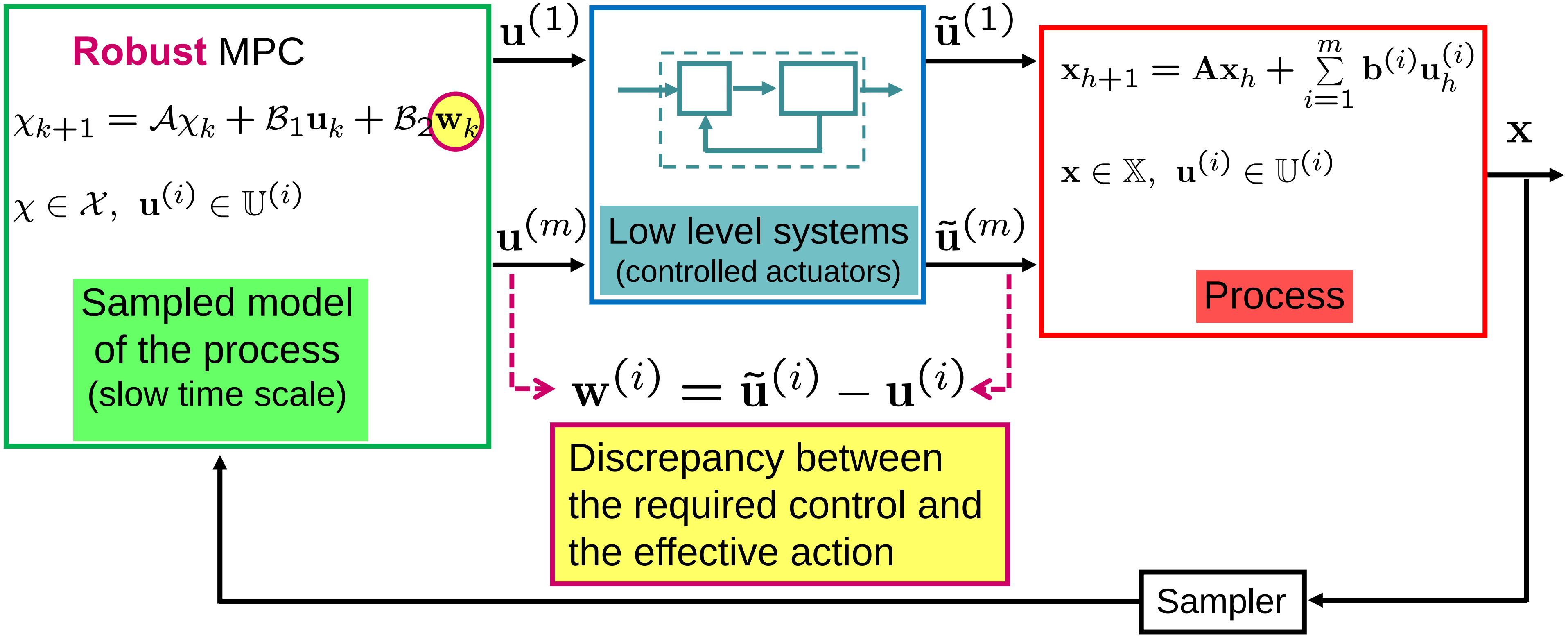
Typical structure in many control applications:

- Process control **[Skogestadt '00]**
- Automotive **[Brahma et al. '00]**
- Production planning **[Golenko-Ginzburg et al. '93]**

A two-layer hierarchical (cascade) control system:

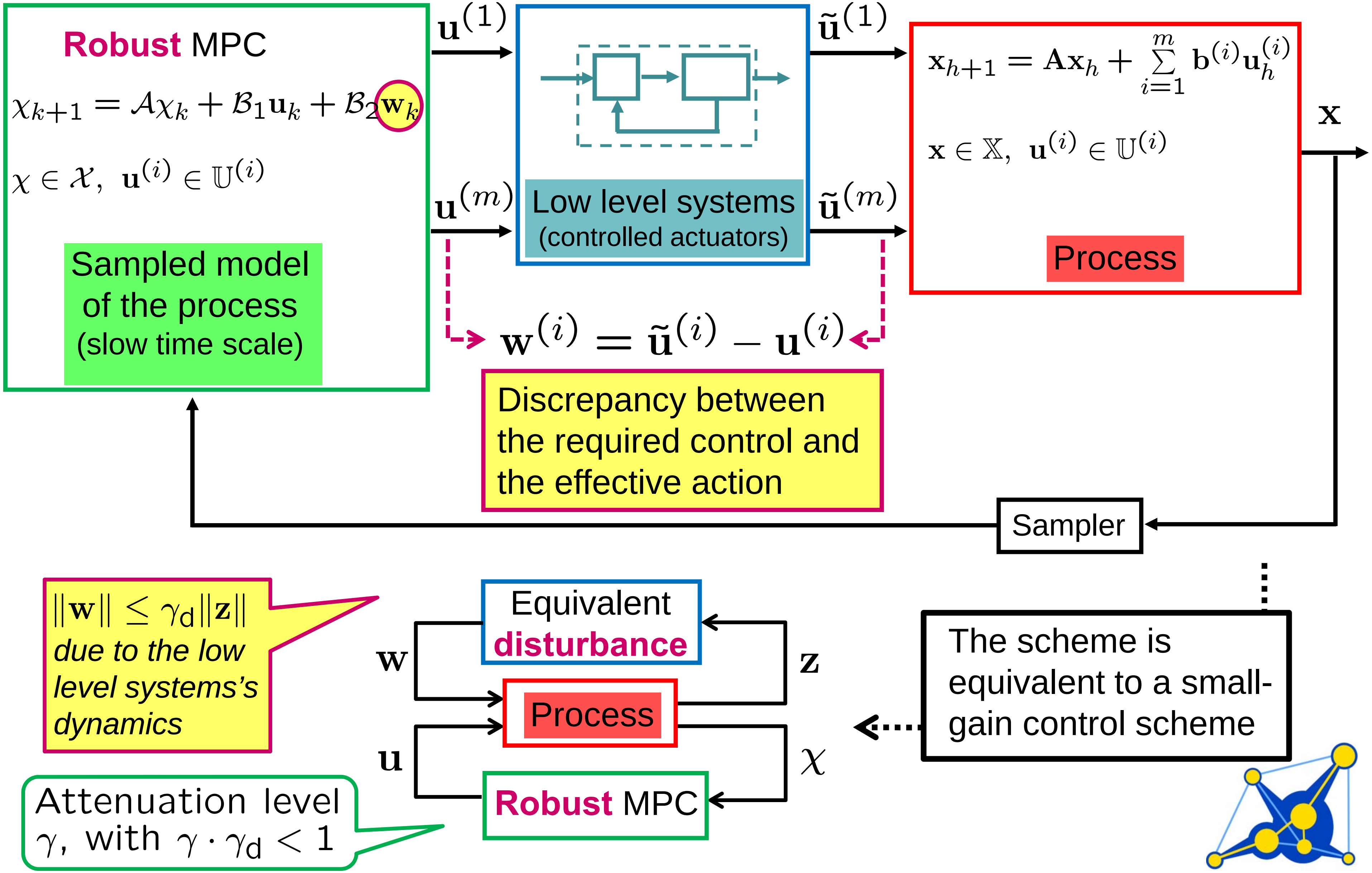


<u>**To be designed:**</u> **MPC** (high-level controller) and
$\mathbf{R}_i$'s (low-level regulators)



Robust MPC
$\chi_{k+1} = \mathcal{A}\chi_k + \mathcal{B}_1\mathbf{u}_k + \mathcal{B}_2\mathbf{w}_k$
$\chi \in \mathcal{X},\ \mathbf{u}^{(i)} \in \mathbb{U}^{(i)}$
Sampled model of the process (slow time scale)
$\mathbf{u}^{(1)}$
$\mathbf{u}^{(m)}$
Low level systems
(controlled actuators)
$\tilde{\mathbf{u}}^{(1)}$
$\tilde{\mathbf{u}}^{(m)}$
$\mathbf{x}_{h+1} = \mathbf{A}\mathbf{x}_h + \sum_{i=1}^{m} \mathbf{b}^{(i)}\mathbf{u}_h^{(i)}$
$\mathbf{x} \in \mathbb{X},\ \mathbf{u}^{(i)} \in \mathbb{U}^{(i)}$
Process
$\mathbf{x}$
$\mathbf{w}^{(i)} = \tilde{\mathbf{u}}^{(i)} - \mathbf{u}^{(i)}$
Discrepancy between the required control and the effective action
Sampler

Hierarchical MPC systems: the robust control approach



High level robust MPC design

Two possible scenarios:

1. The high level unit can <u>*simulate*</u> the low level actuators
   $\Rightarrow$ the disturbance $\mathbf{w}$ is predicted (γ_d is locally available)

2. The disturbance $\mathbf{w}$ is not predictable by the high level unit
   but γ_d is globally available

<u>Main result</u>

In both cases, a **robust MPC** controller is designed so that:

- The high level controller is robustly stabilizing in the slow time scale k ;

- Convergence to the equilibrium for the overall control system is guaranteed in the fast time scale h .

High level robust MPC design

Scenario 1 (w is predictable):

$$\min_{\mathcal{F}} \; J(\chi, \mathcal{F}, N_p)$$

subject to the dynamics, the constraints + a suitable auxiliary law

where

$$J = \sum_{j=0}^{N_p-1} (\|\mathbf{z}_{k+j}\|^2_{Q_z} - \gamma^2 \|\mathbf{w}_{k+j}\|^2_{Q_w}) + V_f(\chi_{k+N_p})$$

and

$$\mathcal{F} = \begin{bmatrix} \mathbf{u}_k & \mathbf{u}_{k+1} & \cdots & \mathbf{u}_{k+N_c-1} \end{bmatrix}$$

is a sequence of control **values**

Features: the high and low level designs are only partially decoupled but a global γ_d is not needed and the optimization is less demanding

Scenario 2 (only γ_d is known):

$$\min_{\mathcal{F}} \max_{\mathcal{D}} \; J(\chi, \mathcal{F}, \mathcal{D}, N_p)$$

subject to the dynamics, the constraints + a suitable auxiliary law

where J is as above

and

$$\mathcal{D} = \begin{bmatrix} \mathbf{w}_k & \mathbf{w}_{k+1} & \cdots & \mathbf{w}_{k+N_p-1} \end{bmatrix}$$

is a sequence of disturbance values

$$\mathcal{F} = \begin{bmatrix} \mathbf{u}_k & \mathbf{u}_{k+1}(\cdot) & \cdots & \mathbf{u}_{k+N_c-1}(\cdot) \end{bmatrix}$$

is a sequence of control **policies**

Features: the high and low level designs are decoupled but γ_d and minmax optimization are needed

Comments

- Perfect reference tracking of the low level systems – i.e., *frequency decoupling* between the inner and outer loops – is not assumed: the low level dynamics is fully taken into account

- Even in the absence of perfect frequency decoupling, the robust control approach allows one to largely decouple the control designs at the **high** and at the **low** level

- Perfect reference tracking of the low level systems – i.e., *frequency decoupling* between the inner and outer loops – is not assumed: the low level dynamics is fully taken into account

- Even in the absence of perfect frequency decoupling, the robust control approach allows one to largely decouple the control designs at the **high** and at the **low** level

Example:

Robust high-level MPC ——
vs
non-Robust algorithm - - -
(neglected low level dynamics)

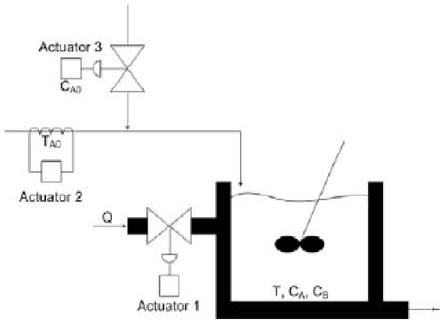
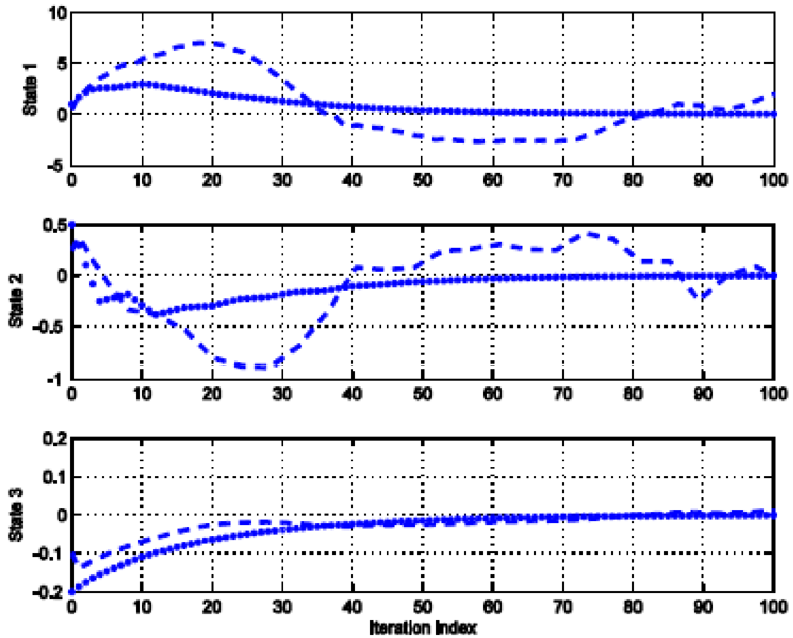
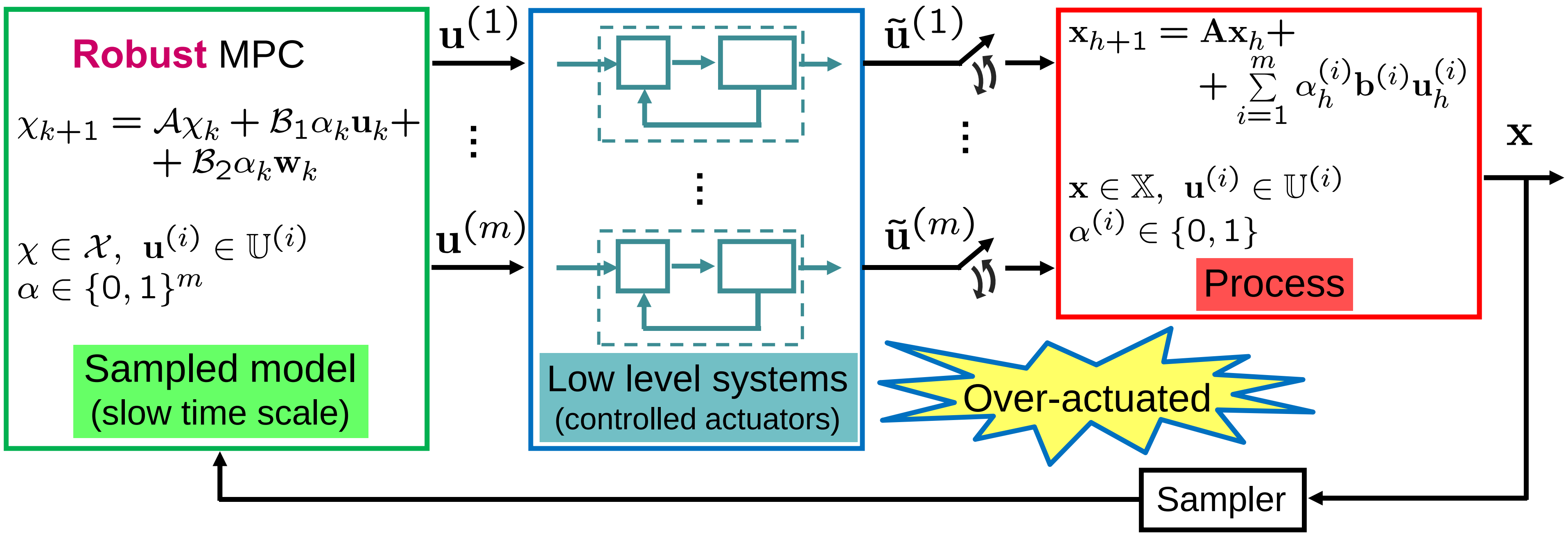


Figure 1: CSTR schematic plant.

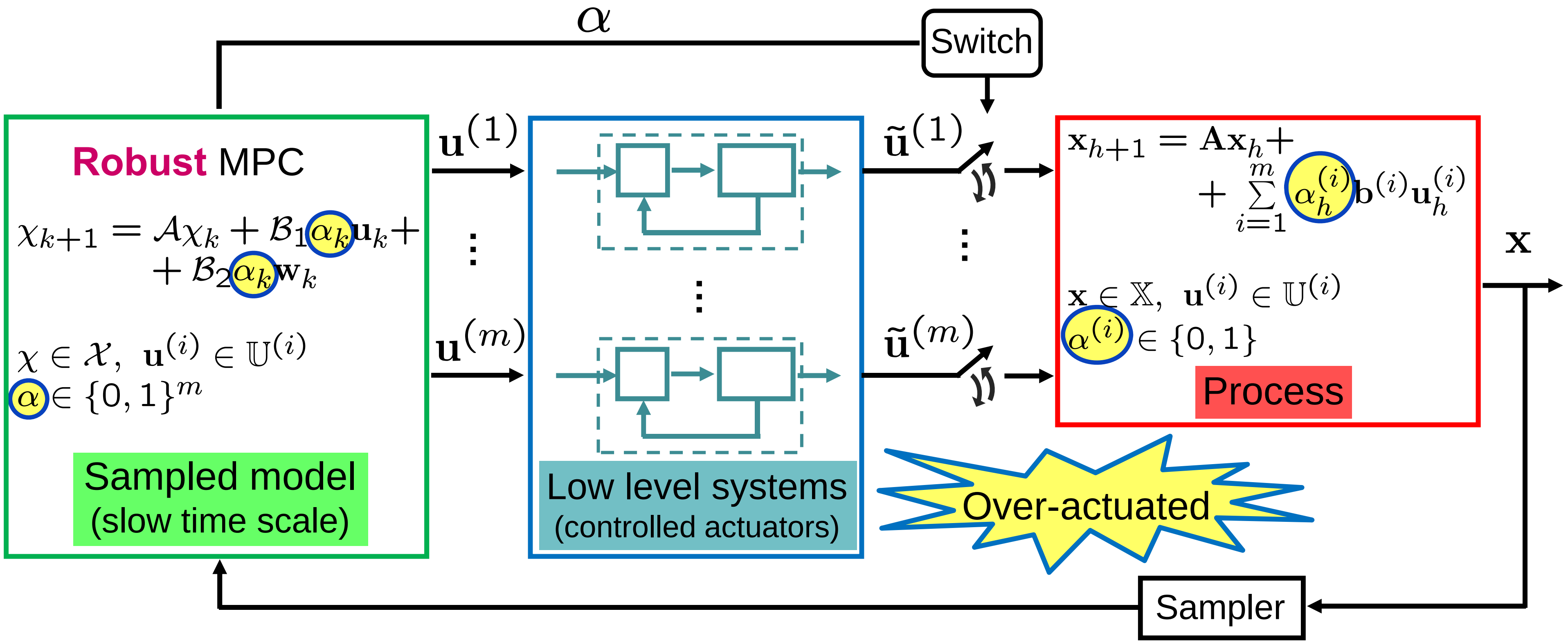


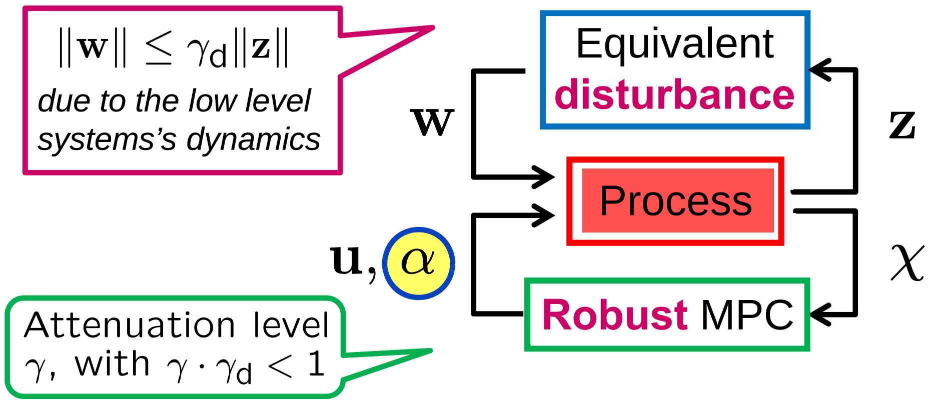


Related works, e.g.:

- Load sharing **[Eitelberg '99]**
- Fault tolerance **[Mhaskar et al. '05** (with MPC)**, Casavola et al. '07]**
- …

Extensions: performance, control allocation problem

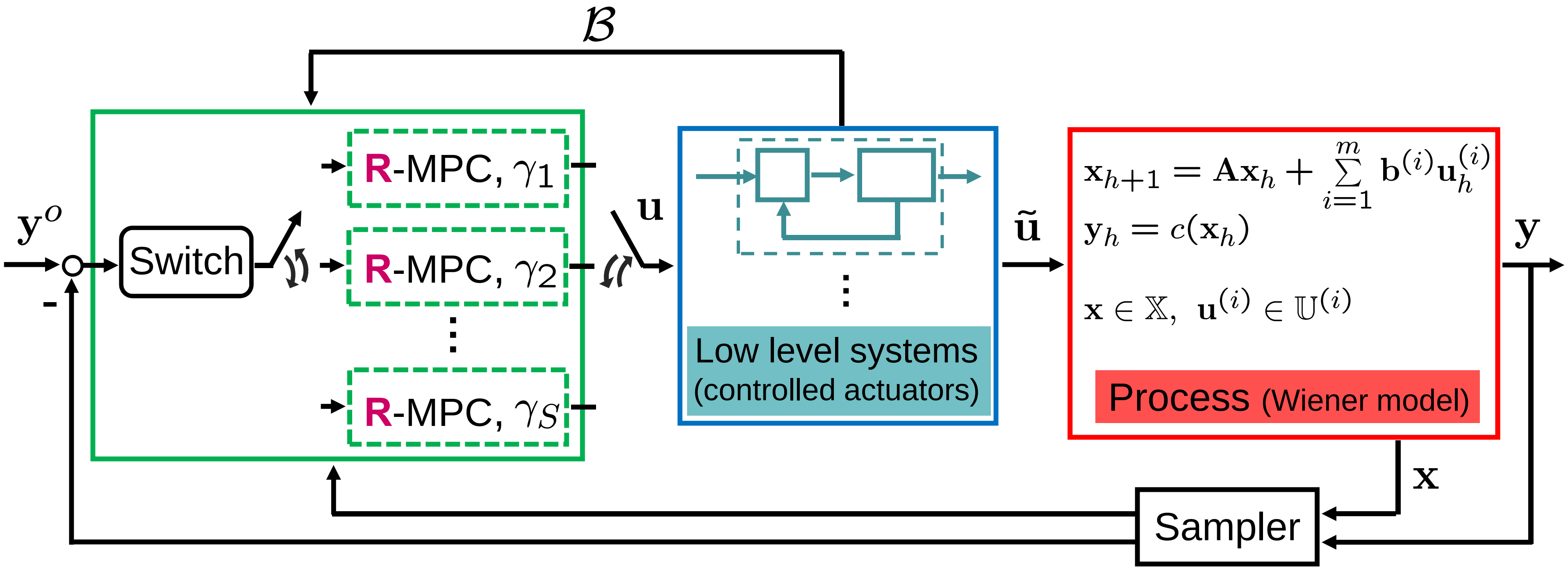




Features:

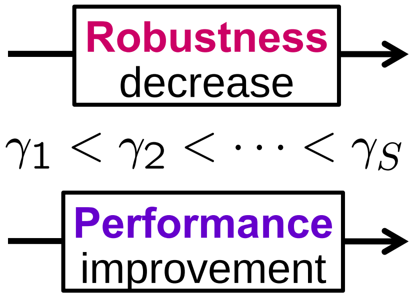
- Control load can be balanced
- The optimization is a Mixed Integer Quadratic Programming problem because of the presence of the boolean variable α

Extensions: performance, high level switching controller



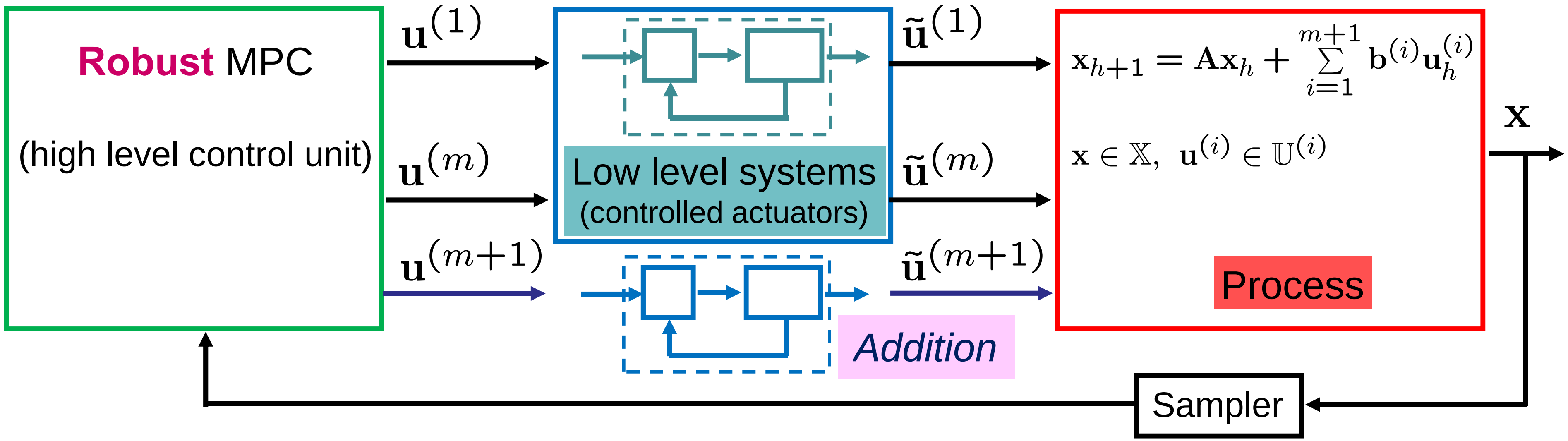
Performance vs robustness :

- Less robustness (a larger γ_i) enforces a faster response of the low level systems, thus it ensures better performance
- Feasibility (i.e., the small-gain condition) is guaranteed: if the actuators are not fast enough, an alert signal $\mathcal{B}$ is sent to the high level which switches to a more robust (smaller γ_i) mode

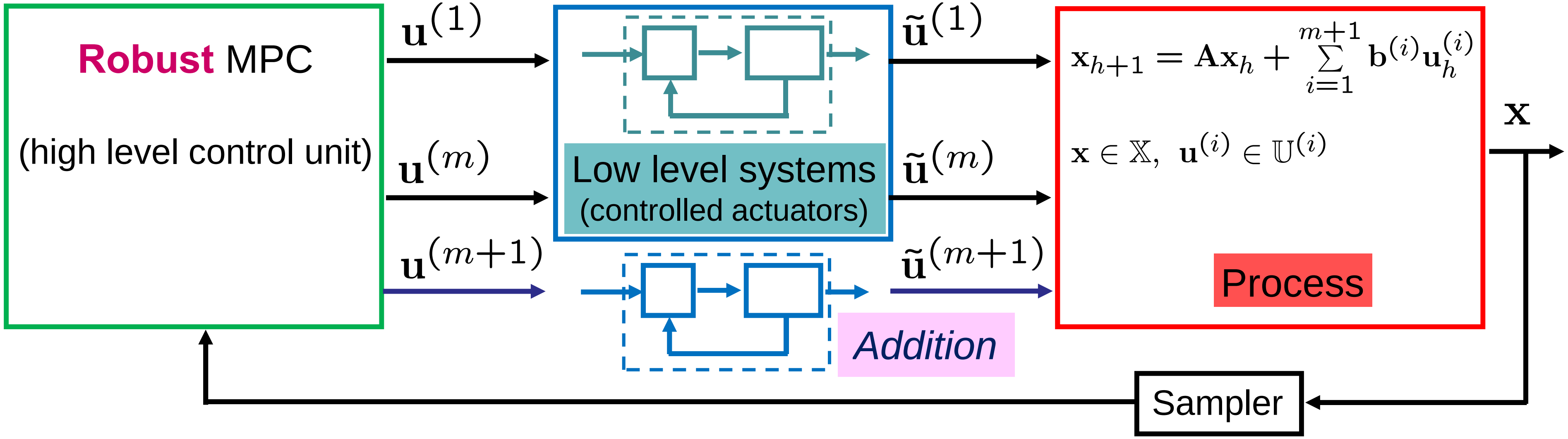


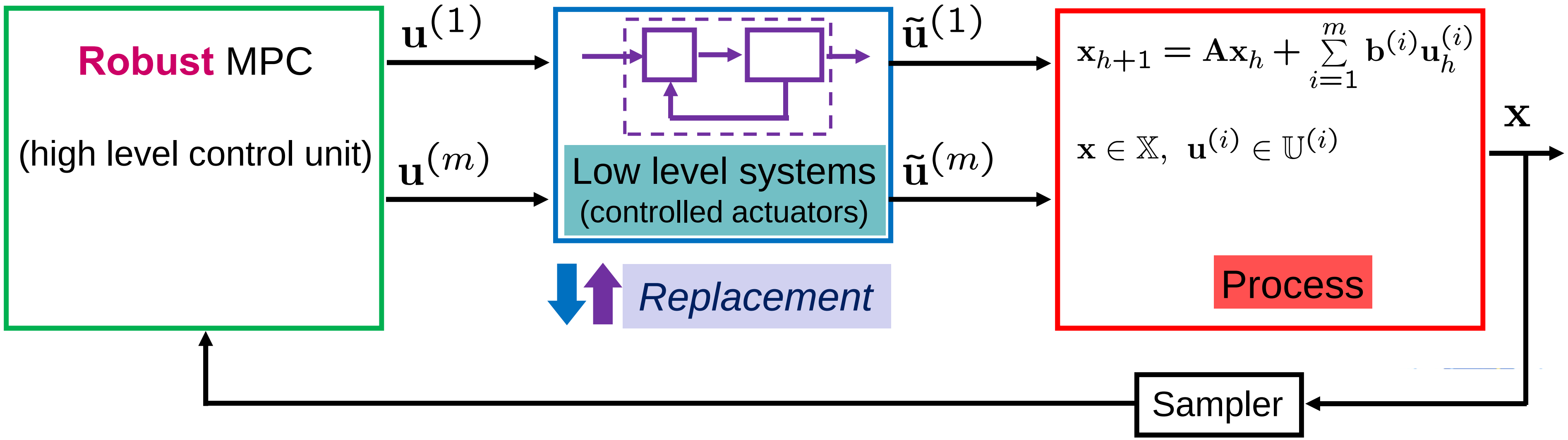
A hybrid system: stability is ensured by a sufficiently large *average dwell-time*

Extensions: reconfigurability (plug & play [Stoustrup '09])



Extensions: reconfigurability (plug & play [Stoustrup '09])





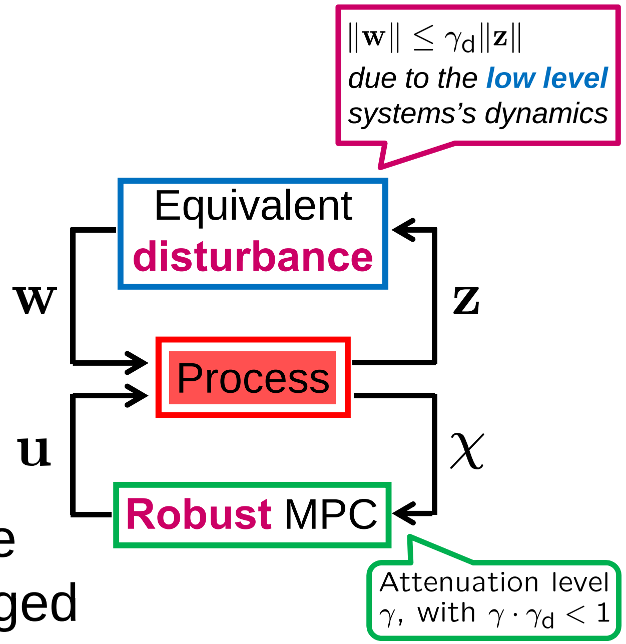
Extensions: reconfigurability (plug & play)

Should one completely re-design the high level control unit ?

In the MPC approach reconfigurability is achieved
if the auxiliary law can be kept unchanged

Main idea:

- The gain γ_d is an abstraction of the low level system
- Different low level configurations characterized by the same (or similar) γ_d can be considered
- Actuators can be substituted/added provided that γ_d does not change. Otherwise a new "attenuation constraint" is added to the MPC problem
- In both cases (actuator substitution and addition) the auxiliary control law can be left (essentially) unchanged



Thus, reconfigurability properties are achieved !

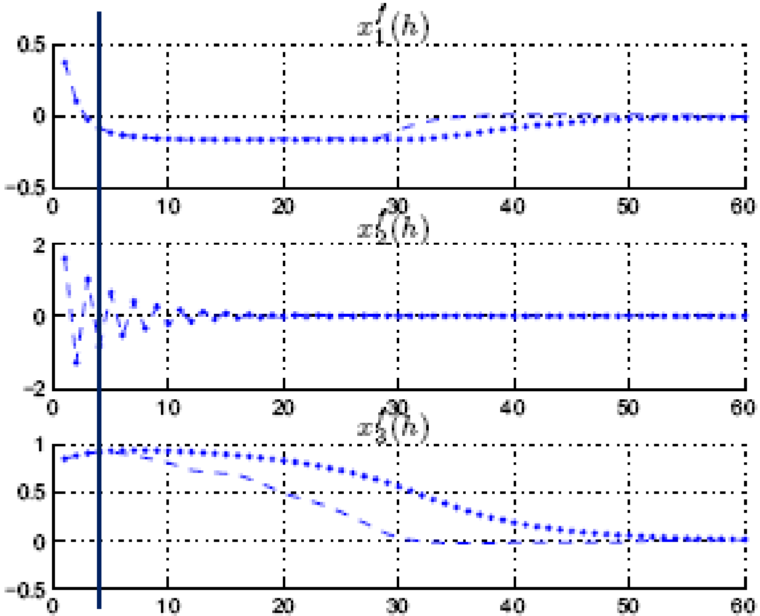
Remark: the resulting control system switches among different stable configurations. Stability is preserved if proper *dwell-time* is guaranteed

Process (basic configuration):

$$x^f(h+1) = \begin{bmatrix} 0.4 & 0 & 0 \\ 0 & -0.8 & 0 \\ 0 & 0 & 1.1 \end{bmatrix} x^f(h) + \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} u_1^f(h) + \begin{bmatrix} 3 \\ 1 \\ 0 \end{bmatrix} u_2^f(h).$$

Actuators (low level system gain): $\gamma_{\mathrm{d}} = 0.161$.

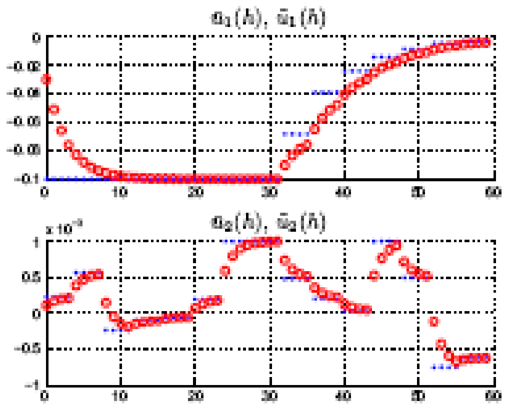
Actuator addition: At time $h = 4$, a new actuator is added and $\gamma_{\mathrm{d}} = 0.963 > 0.161$ (the supplementary "attenuation constraint" is needed in MPC).

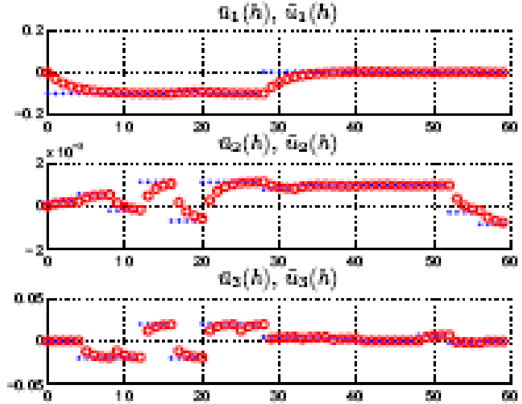


Time: h=4

State trajectories: basic configuration (dots) and with the added actuator (dashed line).

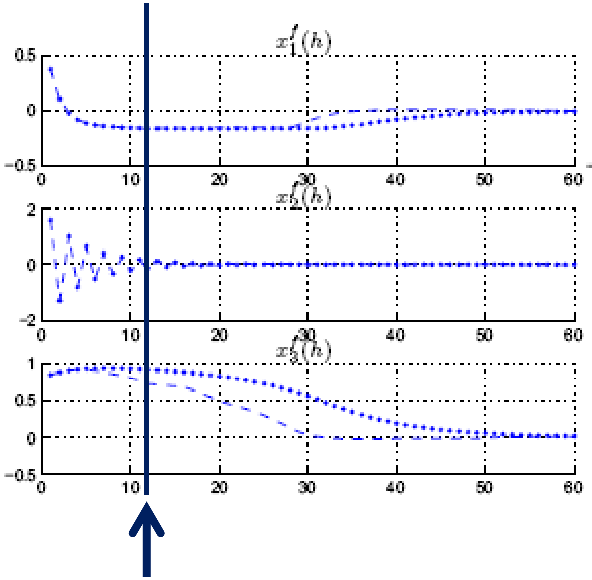
Control reference vs **effective control action**



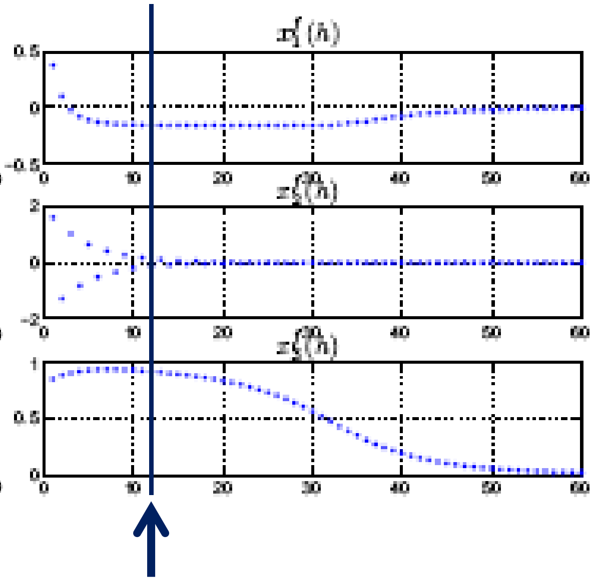


Example

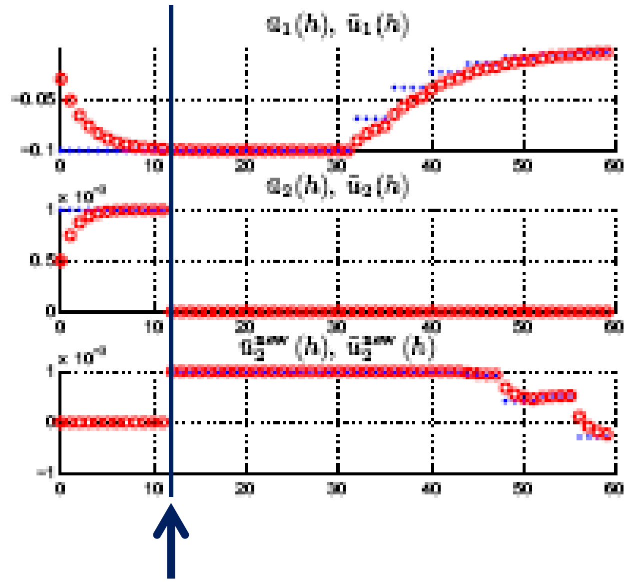
Actuator replacement: At time $h = 12$, the second actuator is replaced with one guaranteeing a better attenuation level ($\gamma_{\mathrm{d}}^{\mathrm{new}} = 0.118 < 0.161$).



Time: *h=12*



Time: *h=12*



Time: *h=12*

State trajectories:

basic configuration, and added actuator

with the replaced actuator

Control reference vs **effective control action**

Conclusions

A robust MPC approach has been presented for the design of two-layer hierarchical control systems

- For constrained **linear discrete-time** systems
- The robust control approach allows to :
  - ➢ largely **decouple** the design at the two levels
  - ➢ to abstract subsystems with their *gain* and thus to obtain versatility resulting in numerous extensions (**reconfigurability**, **control allocation** problems, switching control for **performance** improvements)
- **Convergence results** have been established

Papers:

- B. Picasso, D. De Vito, R. Scattolini, P. Colaneri. An MPC approach to the design of two layer hierarchical control systems. *Automatica, Vol.46(5), pp. 823-831*, 2010.
- B. Picasso, C. Romani, R. Scattolini. Tracking control of Wiener models with hierarchical and switching MPC. *Submitted.*
- D. De Vito, B. Picasso, R. Scattolini. On the design of reconfigurable two layer hierarchical control systems with MPC. In *Proceedings of the American Control Conference, Baltimore, pp. 4704-4712*, 2010.

Outline

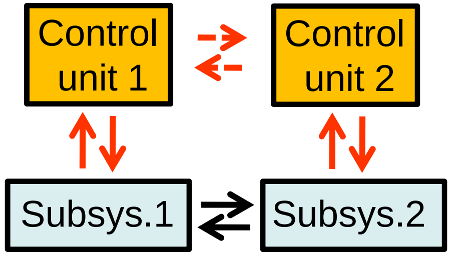
1. Introduction

2. Hierarchical MPC systems
   - Basic architecture
   - Extensions (performance & reconfigurability)
   - Conclusions

3. Distributed MPC systems
   - **A “tube-based”, non cooperative DMPC algorithm**
   - **Conclusions**

4. Concluding remark

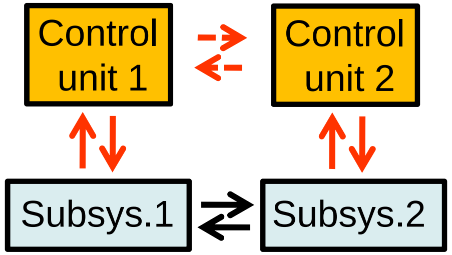
Distributed MPC



Distributed-MPC methods can be classified **[Scattolini '09]** according to:

- Communication protocols
  - Neighbor-to-neighbor
  - All-to-all
- Number of iteration to achieve a solution (at each step)
  - Iterative algorithms
  - Non-iterative algorithms
- Cost function to be optimized
  - Cooperative algorithms (common goal)
  - Non-cooperative algorithms (temperature control, ecc…)

Distributed MPC



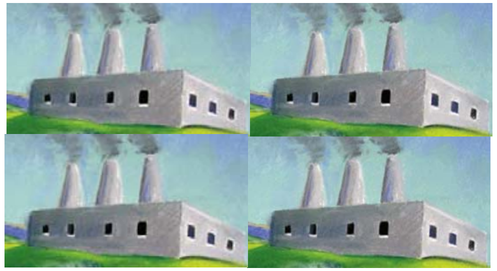
Distributed-MPC methods can be classified **[Scattolini '09]** according to:

– Communication protocols
  - Neighbor-to-neighbor
  - All-to-all
– Number of iteration to achieve a solution (at each step)
  - Iterative algorithms
  - Non-iterative algorithms
– Cost function to be optimized
  - Cooperative algorithms (common goal)
  - Non-cooperative algorithms (temperature control, ecc…)

Most common approaches:

- **Decentralized** MPC:
  [Magni-Scattolini '06, Raimondo et al. '07] (ISS perspective) **[Alessio-Bemporad '08], [Barcelli-Bemporad '09]**
- **Distributed** MPC:
  [Dunbar '07] (non-iterative, non cooperative, neighbor-to-neighbor communication);
  [Liu et al. '09-'10] (iterative, cooperative);
  [Venkat et al. '08, Stewart et al.'10] (possibly iterative, cooperative, output feedback MPC with all-to-all communication);
  [Maestre,'09]: (game theory-based, cooperative, iterative approach for linear systems).

Motivation



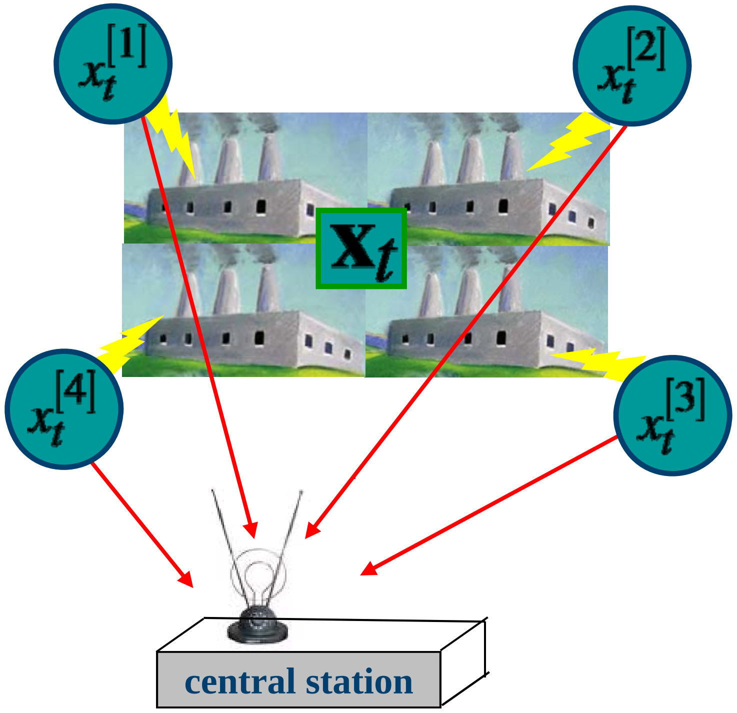
The large-scale system evolves according to the centralized dynamical model:

$$\mathbf{x}_{t+1} = \mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{u}_t$$

$\mathbf{x}_t \in \mathbb{X}$ **constrained state**

$\mathbf{u}_t \in \mathbb{U}$ **constrained input**

Motivation



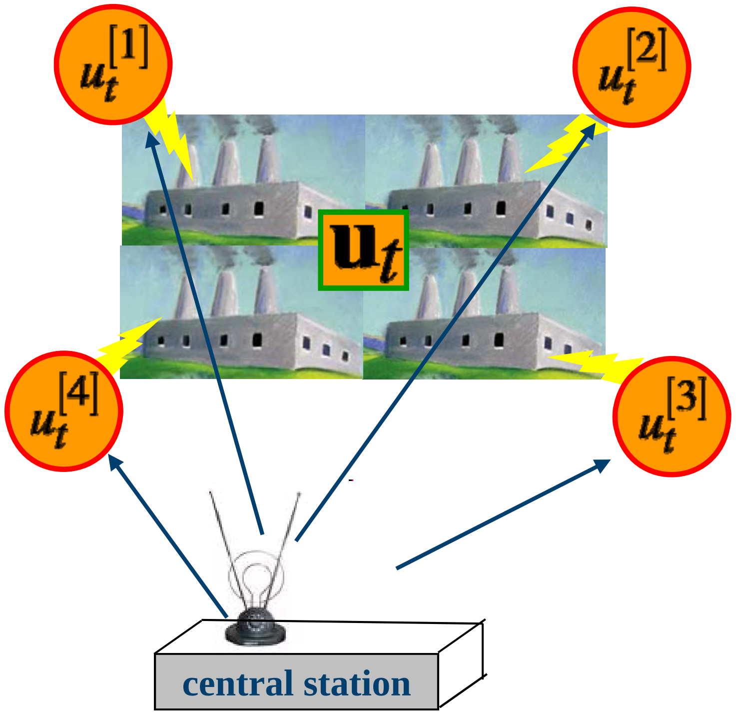
The large-scale system evolves according to the centralized dynamical model:

$$\mathbf{x}_{t+1} = \mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{u}_t$$

$\mathbf{x}_t \in \mathbb{X}$ **constrained state**

$\mathbf{u}_t \in \mathbb{U}$ **constrained input**

Motivation

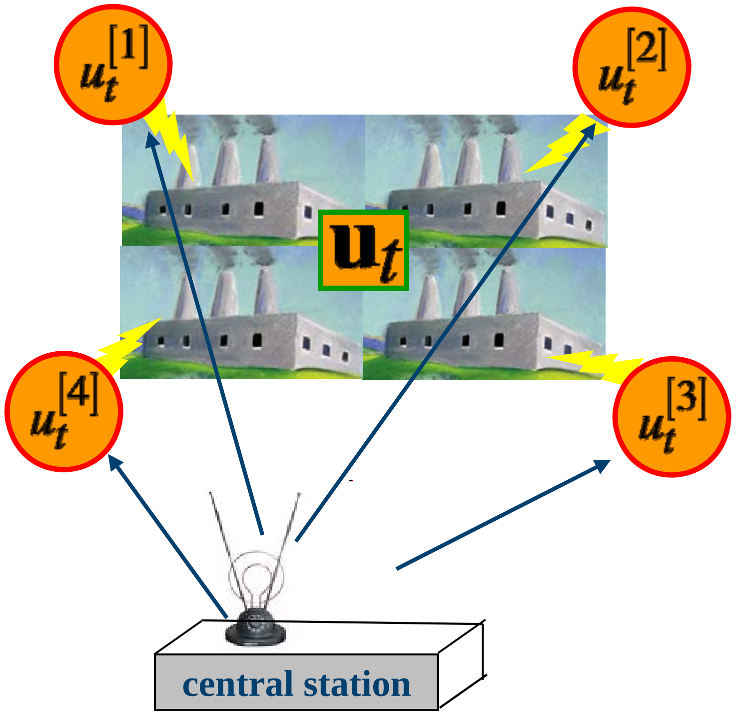


The large-scale system evolves according to the centralized dynamical model:

$$\mathbf{x}_{t+1} = \mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{u}_t$$

$\mathbf{x}_t \in \mathbb{X}$ **constrained state**

$\mathbf{u}_t \in \mathbb{U}$ **constrained input**



The large-scale system evolves according to the centralized dynamical model:

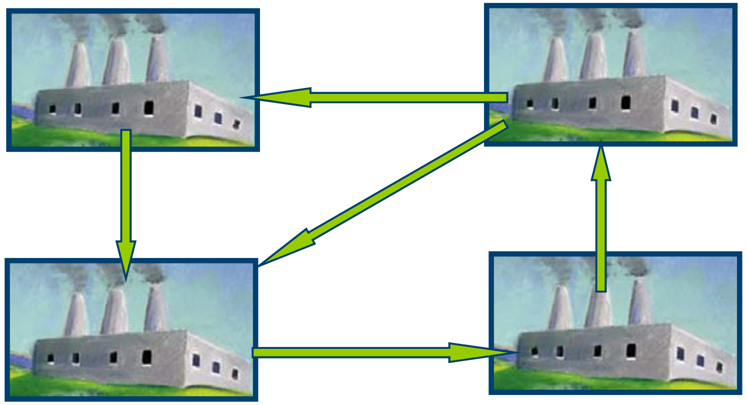
$$\mathbf{x}_{t+1} = \mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{u}_t$$

$\mathbf{x}_t \in \mathbb{X}$ **constrained state**

$\mathbf{u}_t \in \mathbb{U}$ **constrained input**

Aims:
- develop a control algorithm for the process
- use *model predictive control* for optimality and to handle constraints
- solve in parallel 4 small scale optimization problems instead of one large problem
- exploit a neighbor-to-neighbor communication protocol

Assumptions



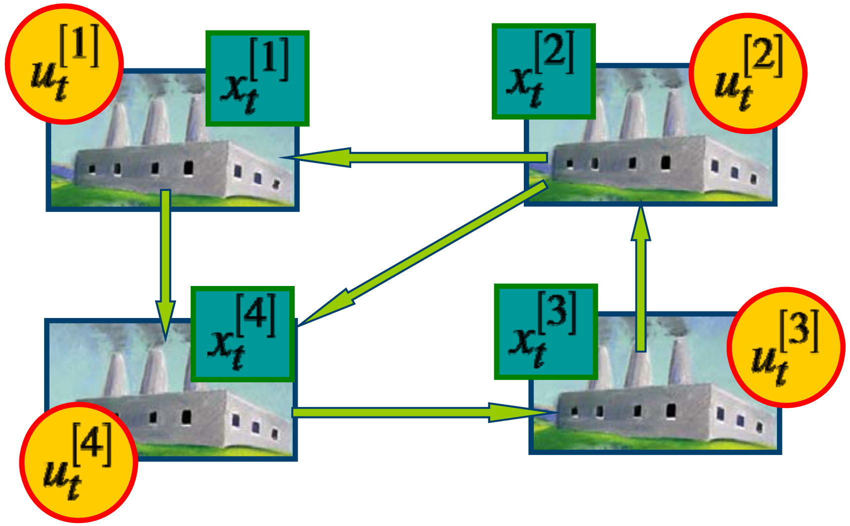
We partition the system into a ***graph*** of interconnected M (here M=4) low-order models.

$$\mathbf{A} = \begin{bmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ A_{21} & A_{22} & A_{23} & A_{24} \\ A_{31} & A_{32} & A_{33} & A_{34} \\ A_{41} & A_{42} & A_{43} & A_{44} \end{bmatrix}$$

$$\mathscr{S} : \mathbf{x}_{t+1} = \mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{u}_t$$

$$\mathbf{B} = \begin{bmatrix} B_1 & 0 & 0 & 0 \\ 0 & B_2 & 0 & 0 \\ 0 & 0 & B_3 & 0 \\ 0 & 0 & 0 & B_4 \end{bmatrix}$$

Assumptions



We partition the system into a ***graph*** of interconnected M (here M=4) low-order models.

$$\mathbf{A} = \begin{bmatrix} A_{11} & A_{12} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & A_{22} & A_{23} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & A_{33} & A_{34} \\ A_{41} & A_{42} & \mathbf{0} & A_{44} \end{bmatrix}$$

$$\mathscr{S} : \mathbf{x}_{t+1} = \mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{u}_t$$

$$\mathbf{B} = \begin{bmatrix} B_1 & 0 & 0 & 0 \\ 0 & B_2 & 0 & 0 \\ 0 & 0 & B_3 & 0 \\ 0 & 0 & 0 & B_4 \end{bmatrix}$$

$$\mathscr{S}_i : x_{t+1}^{[i]} = A_{ii} x_t^{[i]} + B_i u_t^{[i]} + \sum_{j \neq i} A_{ij} x_t^{[j]}$$

DPC: the robust control approach

Large-scale system: $\mathbf{x}_{t+1} = \mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{u}_t$

Graph of interconnected M low-order subsystems:

$$x_{t+1}^{[i]} = A_{ii}x_t^{[i]} + B_i u_t^{[i]} + \sum_{j \neq i} A_{ij} x_t^{[j]}$$

$x_t^{[i]} \in \mathbb{X}_i$ local state constraints

$u_t^{[i]} \in \mathbb{U}_i$ local input constraints

$h_s^{[i]}(x_t^{[i]}, \mathbf{x}_t) \leq 0$

DPC: the robust control approach

Large-scale system: $\mathbf{x}_{t+1} = \mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{u}_t$

Graph of interconnected M low-order subsystems:

$$x_{t+1}^{[i]} = A_{ii}x_t^{[i]} + B_i u_t^{[i]} + \sum_{j \neq i} A_{ij} x_t^{[j]}$$

$x_t^{[i]} \in \mathbb{X}_i$ local state constraints

$u_t^{[i]} \in \mathbb{U}_i$ local input constraints

$h_s^{[i]}(x_t^{[i]}, \mathbf{x}_t) \leq 0$

Each subsystem i

- has a reference trajectory $\tilde{x}_t^{[i]}$ and guarantees that $x_t^{[i]} - \tilde{x}_t^{[i]} \in \mathcal{E}_i$
- transmits, at each time, the nominal trajectory $\tilde{x}_t^{[i]}$ to its neghbors

$$x_{t+1}^{[i]} = A_{ii}x_t^{[i]} + B_i u_t^{[i]} + \sum_{j \neq i} A_{ij}\tilde{x}_t^{[j]} + \sum_{j \neq i} A_{ij} \overbrace{(x_t^{[j]} - \tilde{x}_t^{[j]})}^{\in \mathcal{E}_j}$$

Large-scale system: $\mathbf{x}_{t+1} = \mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{u}_t$

Graph of interconnected M low-order subsystems:

$$x_{t+1}^{[i]} = A_{ii}x_t^{[i]} + B_i u_t^{[i]} + \sum_{j \neq i} A_{ij} x_t^{[j]}$$

$x_t^{[i]} \in \mathbb{X}_i$ local state constraints

$u_t^{[i]} \in \mathbb{U}_i$ local input constraints

$h_s^{[i]}(x_t^{[i]}, \mathbf{x}_t) \leq 0$

Each subsystem i

- has a reference trajectory $\tilde{x}_t^{[i]}$ and guarantees that $x_t^{[i]} - \tilde{x}_t^{[i]} \in \mathscr{E}_i$
- transmits, at each time, the nominal trajectory $\tilde{x}_t^{[i]}$ to its neghbors

$$x_{t+1}^{[i]} = A_{ii}x_t^{[i]} + B_i u_t^{[i]} + \sum_{j \neq i} A_{ij}\tilde{x}_t^{[j]} + \boxed{\sum_{j \neq i} A_{ij}\overbrace{(x_t^{[j]} - \tilde{x}_t^{[j]})}^{\in \mathscr{E}_j}}$$

$$= A_{ii}x_t^{[i]} + B_i u_t^{[i]} + \sum_{j \neq i} A_{ij}\tilde{x}_t^{[j]} + w_t^{[i]}$$

$w_t^{[i]} \in \mathbb{W}_i = \bigoplus_{j \neq i} A_{ij}\mathscr{E}_j$

constrained disturbance

DPC relies on the solution of *M* robust MPC problems (*i*-DPC) with the tube-based approach presented in **[Mayne, Seron, Raković, *Automatica*, 2005]**

i-th “perturbed” model:

$$x_{t+1}^{[i]} = A_{ii}x_t^{[i]} + B_i u_t^{[i]} + \sum_{j\neq i} A_{ij}\tilde{x}_t^{[j]} + w_t^{[i]}$$

i-th nominal model:

$$\hat{x}_{t+1}^{[i]} = A_{ii}\hat{x}_t^{[i]} + B_i \hat{u}_t^{[i]} + \sum_{j\neq i} A_{ij}\tilde{x}_t^{[j]}$$

Assign $u_t^{[i]} = \hat{u}_t^{[i]} + K_i^{aux}(x_t^{[i]} - \hat{x}_t^{[i]})$

Define $z_t^{[i]} = x_t^{[i]} - \hat{x}_t^{[i]}$

$$\Rightarrow \quad z_{t+1}^{[i]} = (A_{ii} + B_i K_i^{aux}) z_t^{[i]} + w_t^{[i]} \qquad w_t^{[i]} \in \mathbb{W}_i$$

If $(A_{ii}+B_i K^{aux}_i)$ is as. stable, there exists a RPI (robust positively invariant) set Z_i for all *i*. Therefore

$$x_t^{[i]} - \hat{x}_t^{[i]} \in Z_i \quad \Rightarrow \quad x_k^{[i]} - \hat{x}_k^{[i]} \in Z_i \text{ for all } k \geq t$$

MAIN UNDERLYING IDEA

Guarantee that

$$\hat{x}^{[i]}_{t+k} - \tilde{x}^{[i]}_{t+k} \in E_i,\ k = 0,\ldots,N-1$$

$$x^{[i]}_t - \hat{x}^{[i]}_t \in Z_i$$

Guaranteed by suitable constraints in the optimization problem

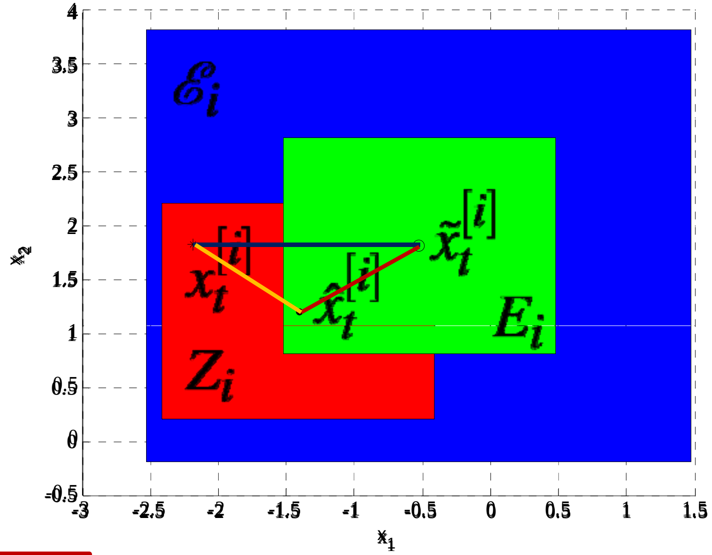
where $E_i \oplus Z_i \subseteq \mathscr{E}_i$

At time t:

$$x^{[i]}_t - \tilde{x}^{[i]}_t = (x^{[i]}_t - \hat{x}^{[i]}_t) + (\hat{x}^{[i]}_t - \tilde{x}^{[i]}_t) \in \mathscr{E}_i$$

$$w^{[i]}_t = \sum_{j \neq i} A_{ij}(x^{[j]}_t - \tilde{x}^{[j]}_t) \in \bigoplus_{j \neq i} A_{ij}\mathscr{E}_j = \mathbb{W}_i$$

$$x^{[i]}_{t+1} - \hat{x}^{[i]}_{t+1} \in Z_i$$



By induction:

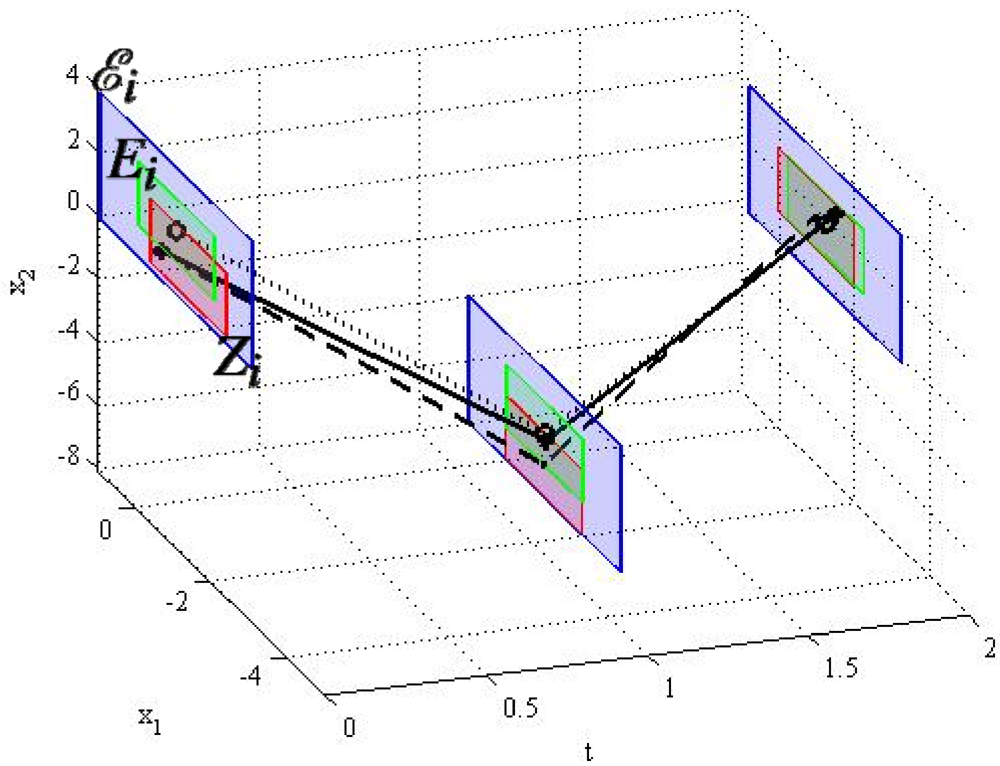
$$x^{[i]}_{t+k} - \hat{x}^{[i]}_{t+k} \in Z_i,\ k = 1,\ldots,N$$

DPC: the robust control approach

—— : $x_t^{[i]}$

- - - : $\hat{x}_t^{[i]}$

----- : $\tilde{x}_t^{[i]}$



Online phase

- Solve M tube-based robust MPC problems (i-DPC), with dynamic constraints:

$$\hat{x}_{t+1}^{[i]} = A_{ii}\hat{x}_t^{[i]} + B_i\hat{u}_t^{[i]} + \sum_{j\neq i} A_{ij}\tilde{x}_t^{[j]}$$

- Coupling variables are the reference trajectories $\tilde{x}_k^{[i]}$ (known in all the prediction horizon $k=t,\ldots,t+N-1$)

- Further constraint on the solution of the i-DPC:

$$\hat{x}_{t+k}^{[i]} - \tilde{x}_{t+k}^{[i]} \in E_i,\ k = 0,\ldots,N-1$$

$$x_t^{[i]} - \hat{x}_t^{[i]} \in Z_i$$

- Solution: $\hat{x}_{t/t}^{[i]}, \{\hat{u}_{k/t}^{[i]}\}_{k=t}^{t+N-1}$

→ input to the real system: $u_t^{[i]} = \hat{u}_{t/t}^{[i]} + K_i^{aux}(x_t^{[i]} - \hat{x}_{t/t}^{[i]})$

→ reference trajectory update: $\tilde{x}_{t+N}^{[i]} = \hat{x}_{t+N/t}^{[i]}$

The optimization problem at time *t*

Given - the ref. trajectory of i: $\tilde{x}_k^{[i]}, k = t, \ldots, t+N-1$

- the ref. trajectories of its neighbors: $\tilde{x}_k^{[j]}, k = t, \ldots, t+N-1$

$$\min_{\hat{x}_t^{[i]}, \{\hat{u}_k^{[i]}\}_{k=t}^{t+N-1}} \sum_{k=t}^{t+N-1} l_i(\hat{x}_k^{[i]}, \hat{u}_k^{[i]}) + V_i^F(\hat{x}_{t+N}^{[i]})$$

subject to

$$\hat{x}_{t+1}^{[i]} = A_{ii}\hat{x}_t^{[i]} + B_i\hat{u}_t^{[i]} + \sum_{j \neq i} A_{ij}\tilde{x}_t^{[j]}$$

$$x_t^{[i]} - \hat{x}_t^{[i]} \in Z_i$$

$$\hat{x}_k^{[i]} - \tilde{x}_k^{[i]} \in E_i \qquad k = t, \ldots, t+N-1$$

$x_k^{[i]} \in \mathbb{X}_i$	→	$\hat{x}_k^{[i]} \in \hat{\mathbb{X}}_i$	$\hat{\mathbb{X}}_i \oplus Z_i \subseteq \mathbb{X}_i$	**local state constraint**
$u_k^{[i]} \in \mathbb{U}_i$	→	$\hat{u}_k^{[i]} \in \hat{\mathbb{U}}_i$	$\hat{\mathbb{U}}_i \oplus KZ_i \subseteq \mathbb{U}_i$	**input constraint**
$h_s^{[i]}(x_k^{[i]}, \mathbf{x}_k) \leq 0$	→	$\hat{h}_s^{[i]}(\hat{x}_k^{[i]}, \tilde{\mathbf{x}}_k) \leq 0$		**coupled state constraint**
$x_{t+N}^{[i]} \in \mathbb{X}_i^F$	→	$\hat{x}_{t+N}^{[i]} \in \hat{\mathbb{X}}_i^F$	$\hat{\mathbb{X}}_i^F \oplus Z_i \subseteq \mathbb{X}_i^F$	**terminal constraint**

Offline phase

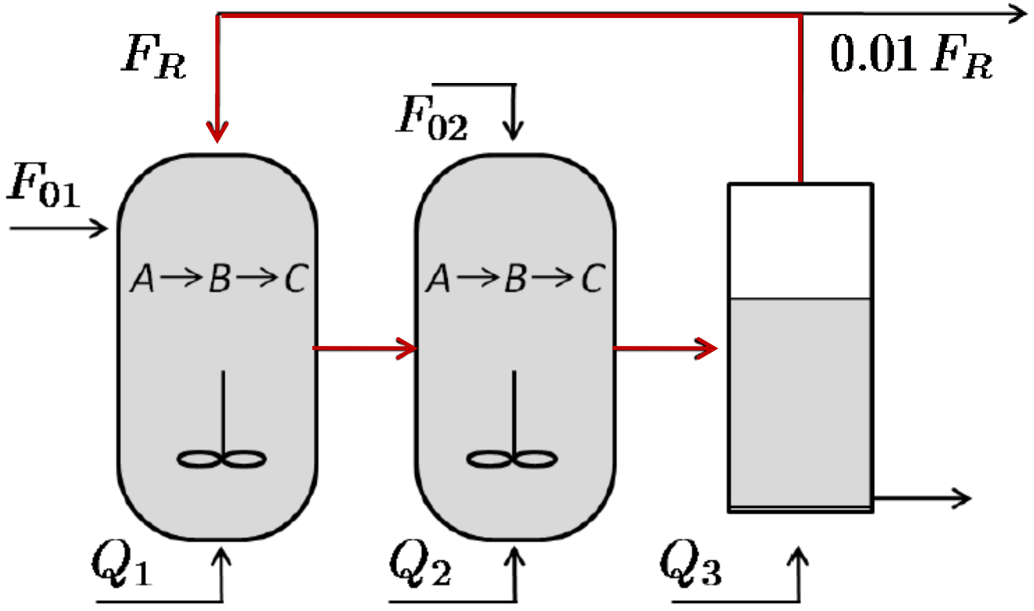
1. Assign suitable decentralized stabilizing auxiliary control law.

2. Define suitable *i*-DPC optimization problem cost functions.

3. Define the sets $\mathcal{E}_i, E_i, Z_i$.

4. Initialize the reference trajectory and the set a suitable value for the prediction horizon *N*.

Example: Chemical plant – reactor/separator process [Liu et al. 2010]



The model is developed under the assumption of hydraulic equilibrium

States for each subsystem:

- x_{Ai} : Concentration of compound *A*
- x_{Bi} : Concentration of compound *B*
- T_i : Temperature of subsystem *i*

Inputs for each subsystem:

- Q_i :Heat

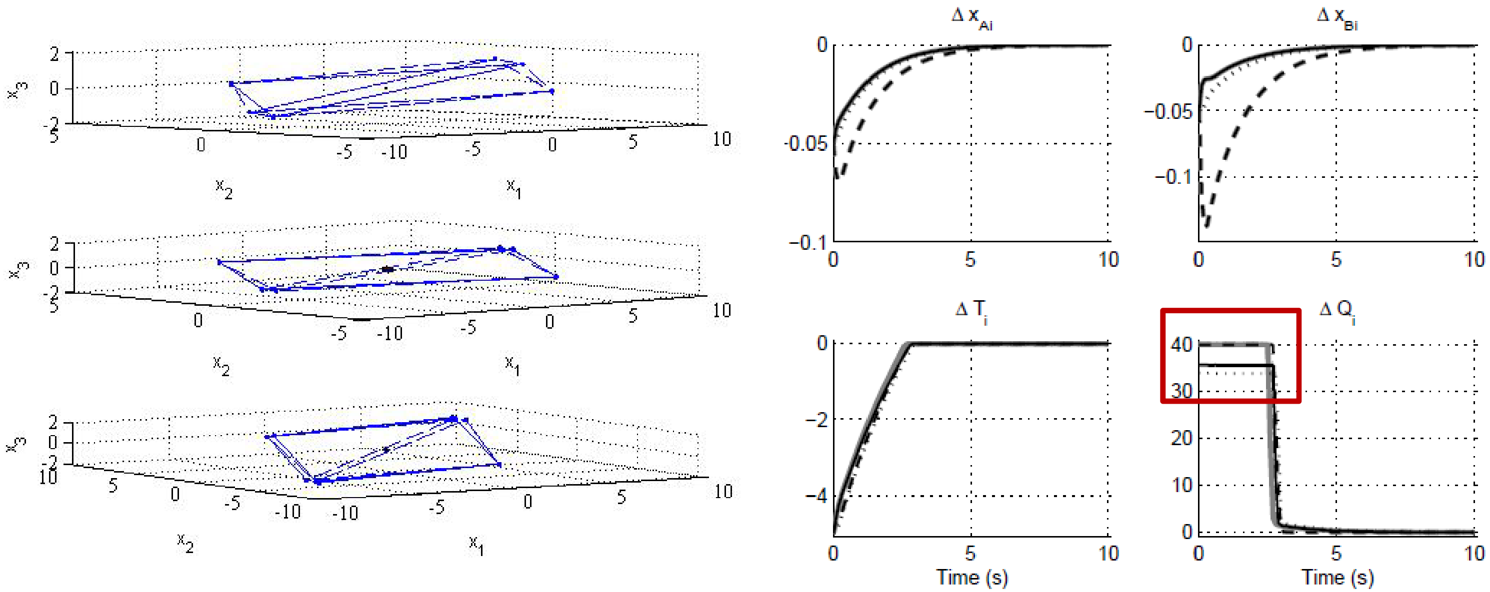
We use the linearized model around a given equilibrium point

Example: Chemical plant – reactor/separator process

We study the response of linearized model to a perturbation of magnitude

$$\begin{bmatrix} \Delta x_{Ai} \\ \Delta x_{Bi} \\ \Delta T_i \end{bmatrix} = \begin{bmatrix} -0.05 \\ -0.05 \\ -5 \end{bmatrix}$$

Input constraints: $0 \leq Q_i \leq 50$ ⇒ $-10 \leq \Delta Q_i \leq 40$



Conclusions

A distributed predictive control algorithm has been presented

- For **linear discrete-time** systems
- A **large scale control problem** has been subdivided into *M* low order, almost independent subproblems
- **Non cooperative algorithm**: each subsystem minimizes a local cost function
- **Neighbor-to-neighbor transmission** is required: low transmission burden
- Only **local knowledge on the systems** dynamics is required
- The algorithm is highly **scalable**: transmission, memory and computational loads do not grow.
- **Constraints** on state and input variables (local and global) can be handled
- **Convergence results** can be established

Conclusions

Advances:

- Efficient algorithms for the initialization of DPC
- Output feedback DPC
- Extension for coping with non input-decoupled systems (B is not block diagonal)

Wide area of application of DPC:

- Independent systems with coupled constraints (e.g., transportation network)
- Cascade systems (e.g., simplified model of an HPV)
- Chemical plants with relevant couplings and feedbacks

Future developments:

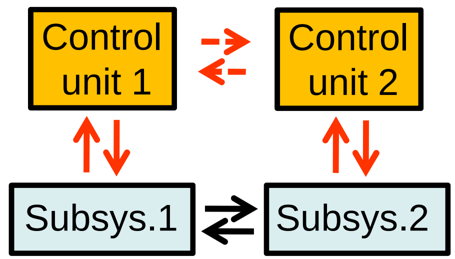
- Explore applications in a plug-and-play architecture
- DPC for tracking

Papers:

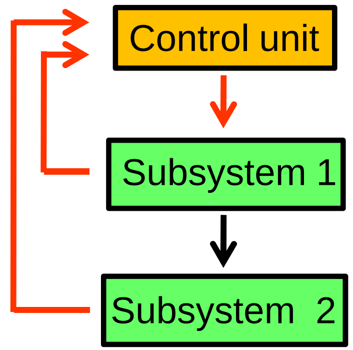
- M. Farina, R. Scattolini. Distributed non-cooperative MPC with neighbor-to-neighbor communication. *Proceedings of the IFAC World Conference, 2011*.
- M. Farina, R. Scattolini. Distributed predictive control: a non-cooperative algorithm with neighbor-to-neighbor communication for linear systems. Submitted.
- M. Farina, R. Scattolini. An output feedback distributed predictive control algorithm. To appear in *Proceedings of the IEEE Conference on Decision and Control 2011*.

Concluding remark

Motivations for **distributed** / **hierarchical** control:



- Reduce the computational load
- Reduce the communication load
- Improve the robustness with respect to failures
  - in the transmission of information
  - in the central control unit
- Improve the modularity and the flexibility of the system
- Consider different goals at different time scales (Real-Time Optimization)
- Synchronize subsystems working at different time scales



Both for distributed and hierarchical control systems, robust control turns out to be a suitable tool to deal with the main issues concerned with large-scale and complex systems.