Distributed Predictive Control and simplified implementations

Riccardo Scattolini

Dipartimento di Elettronica e Informazione
Politecnico di Milano

Industry Workshop on Hierarchical and Distributed
Model Predictive Control (HD-MPC)
Leuven, 24 June 2011

Outline and purpose of the presentation

Outline

- Motivations
- Decentralized MPC
- Distributed MPC
- Simplified distributed MPC

Purpose

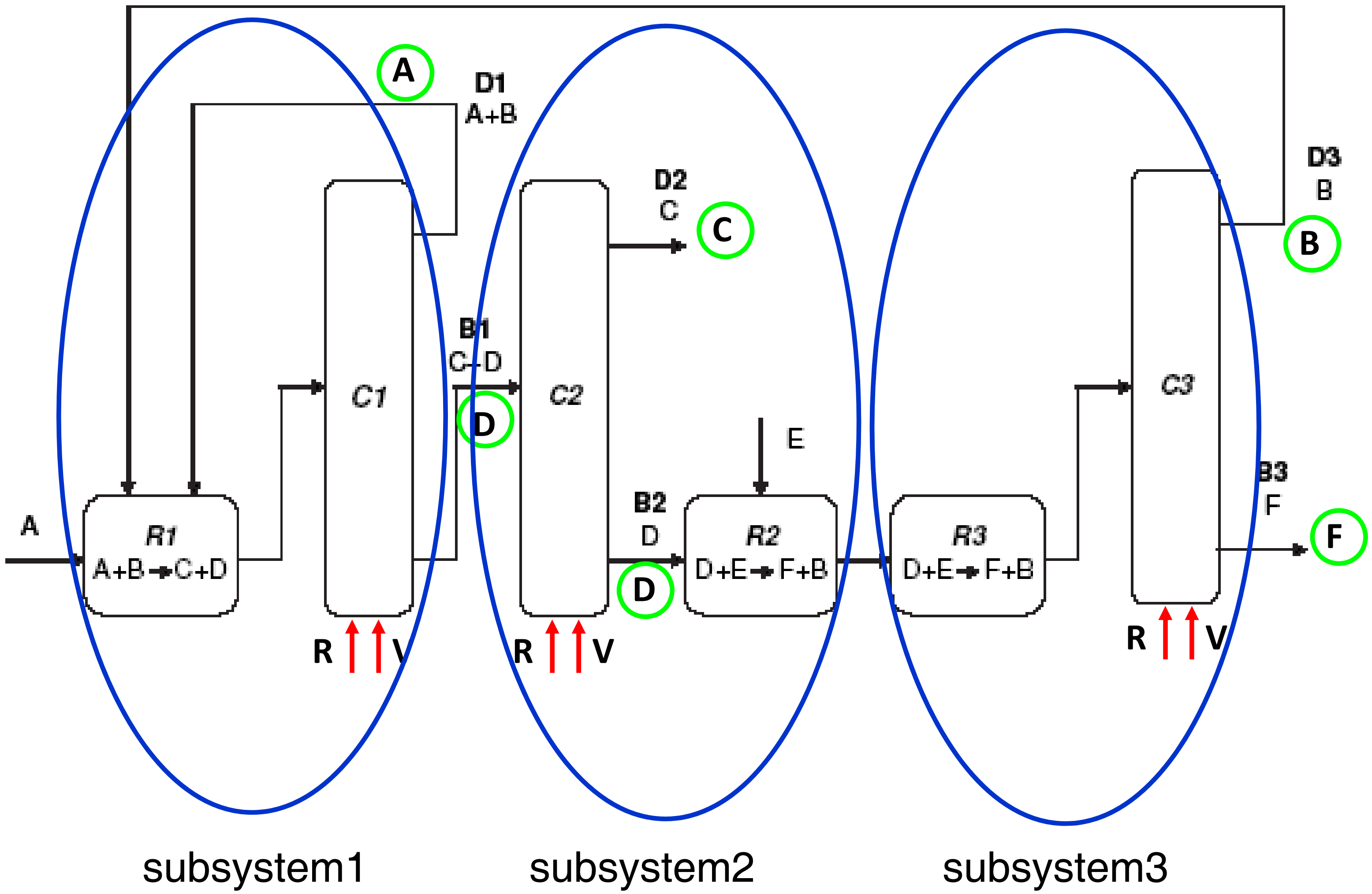
- To describe some “prototype” algorithms (main ideas, no formal proofs or theoretical results).
- To illustrate a simplified implementation
- To test the algorithms in significant cases

Motivations for decentralized/distributed control

Decomposition of a large scale MPC problem in smaller subproblems is useful to:

- Reduce the computational load
- Reduce the communication load
- Improve the robustness with respect to failures in the transmission of information and/or in the central control unit
- Improve the modularity and the flexibility of the system
- Synchronize subsystems working at different time scales

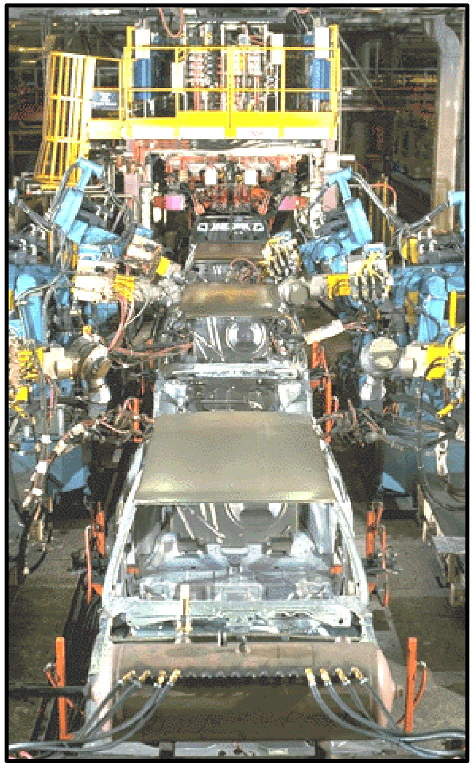
Distributed control of large scale systems



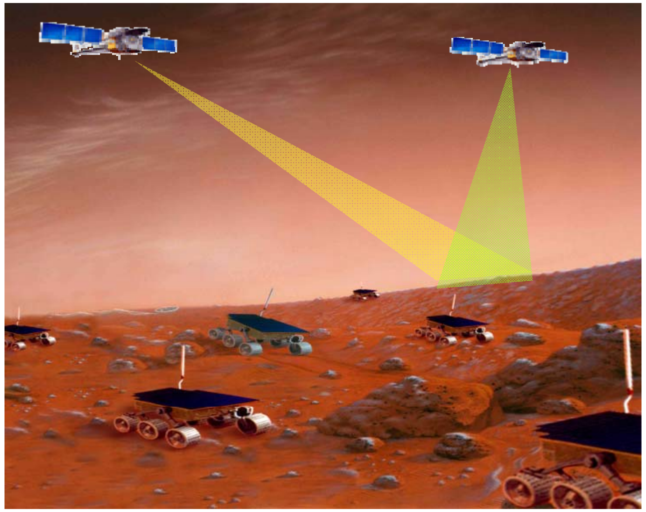
Coordination of independent systems

smart grids with
distributed generation

industrial systems



autonomous
vehicles



MPC basic formulation
state feedback

System $$x(k+1) = Ax(k) + Bu(k)$$

constraints $$x \in X,\; u \in U$$

X and U compact sets containing the origin

Auxiliary stabilizing control law $u = Kx$

Positively invariant set $X_f \subseteq X$ such that for any $x(\bar{k}) \in X_f$

$$x(k) \in X_f \quad ,\; k > \bar{k}$$

$$u(k) = Kx(k) \in U \quad ,\; k > \bar{k}$$

MPC basic formulation - 2

At any time k find the sequence

$$u(k), u(k+1), \ldots, u(k+N-1)$$

minimizing the cost function ($Q>0$, $R>0$)

$$J = \sum_{i=0}^{N-1} [x'(k+i)Qx(k+1) + u'(k+i)Ru(k+i)] + V_f(x(k+N))$$

subject to

$$x(k+i) \in X, \; u(k+i) \in U$$

$$x(k+N) \in X_f$$

Then use only the first element of the optimal control sequence. This implicitly defines the MPC time-invariant control law

$$u = \kappa_{MPC}(x)$$

MPC basic formulation - 3

Typical choices – 1

auxiliary control law $u=K_{LQ}x$

$$K_{LQ} = -\left(R + B'PB\right)^{-1} B'PA$$

$$P = A'P + PA + Q - A'PB\left(R + B'PB\right)^{-1} B'PA \ , \ P > 0$$

terminal cost $V_f=x'Px$

terminal set $X_f = \left\{x \mid x'Px \leq \alpha\right\} \subseteq X$

P is in general a matrix without a block diagonal structure

Typical choices - 2

auxiliary control law $u=0$

Terminal cost $V_f=0$

Terminal set $\{0\}$

Robust MPC
"tube-based" approach

W.Langson, I. Chryssochoos, S.V. Rakovic, D.Q. Mayne: "Robust model predictive control using tubes", *Automatica*, Vol. 40, n.1, 2004.

System $x(k+1) = Ax(k) + Bu(k) + w(k)$

Constraints $x \in X,\ u \in U,\ w \in W$

W is a compact and convex set containing the origin

Nominal model $x_n(k+1) = Ax_n(k) + Bu_n(k)$

Consider the control law $u(k)=u_n(k)+K(x(k)-x_n(k))$

where K is a stabilizing gain, i.e. $A_K=A+BK$ is stable

Robust MPC

"tube-based" approach – 2

Now let $e(k)=x(k)-x_n(k)$,

$$e(k+1) = A_K e(k) + w(k)$$

and define E as a "disturbance invariant set" such that

$$A_K E \oplus W \subseteq E$$

If the following constraints are fulfilled

$$x_n \in X_n = X \ominus E,\ u_n \in U_n = U \ominus KE$$

one has $\forall e(0) \in E,\ e(k) \in E$, $x(k) \in X$ and $u(k) \in U$

Robust MPC
"tube-based" approach - 3

For the nominal model solve the MPC problem

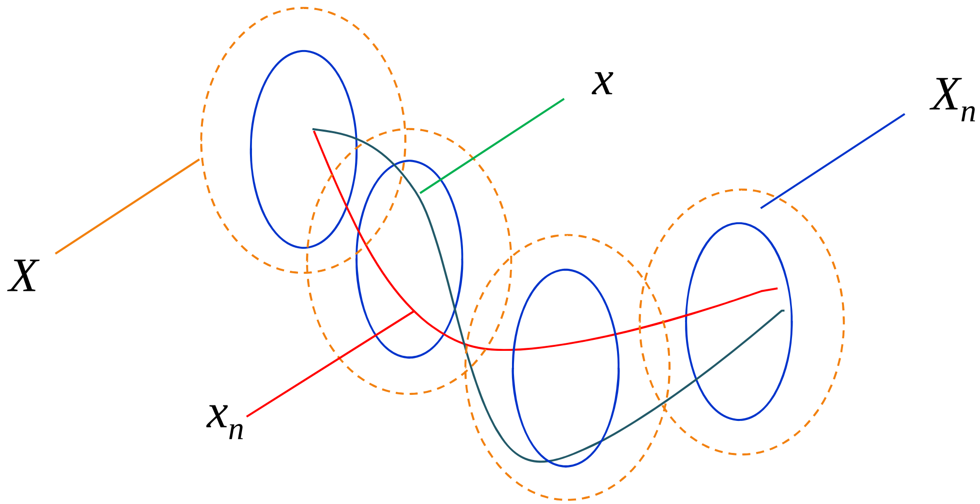
$$\min_{u_n} J = \sum_{i=0}^{N-1} [x_n'(k+i) Q x_n(k+i) + u_n'(k+i) R u_n(k+i)] + V_f(x_n(k+N))$$

$$x_n(k+1) = A x_n(k) + B u_n(k)$$

$$x_n(k+i) \in X_n$$

$$u_n(k+i) \in U_n$$

$$x_n(k+N) \in X_{fn} \subseteq X \ominus E$$



Robust MPC

"tube-based" approach – an extension

D.Q. Mayne, M.M. Seron, S.V. Rakovic: "Robust model predictive control of linear systems with bounded disturbances", *Automatica*, Vol. 41, n.2, 2005.

At any time instant k it is possible to optimize also with respect to the initial state $x_n(k)$

$$\min_{u_n, x_n(k)} J = \sum_{i=0}^{N-1} [x_n'(k+i) Q x_n(k+i) + u_n'(k+i) R u_n(k+i)] + V_f(u_n(k+N))$$

$$x_n(k+1) = A x_n(k) + B u_n(k)$$

$$x(k) - x_n(k) \in E$$

$$x_n(k+i) \in X_n$$

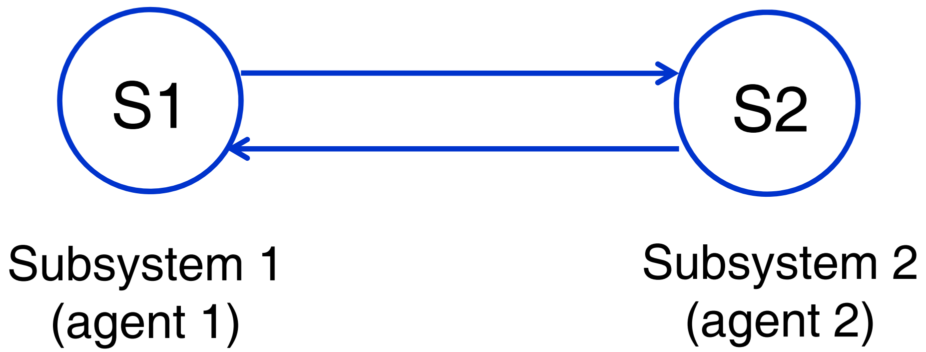
$$u_n(k+i) \in U_n$$

$$x_n(k+N) \in X_{fn}$$

This leads to the control law $u(k)=K_{MPC}(x(k))$ instead of $u(k)=K_{MPC}(x(k),x_n(k))$

Decentralized
and
distributed MPC

A simple example for decentralized/distributed MPC



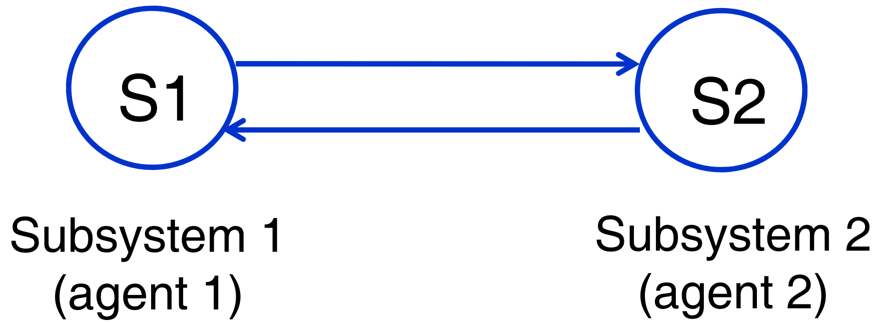
$$S1:\ x_1(k+1) = A_{11}x_1(k) + A_{12}x_2(k) + B_{11}u_1(k) + B_{12}u_2(k)$$

$$S2:\ x_2(k+1) = A_{21}x_1(k) + A_{22}x_2(k) + B_{21}u_1(k) + B_{22}u_2(k)$$

$$x_1 \in X_1,\ x_2 \in X_2,\ u_1 \in U_1, u_2 \in U_2$$

polytopic sets containing the origin

Special cases

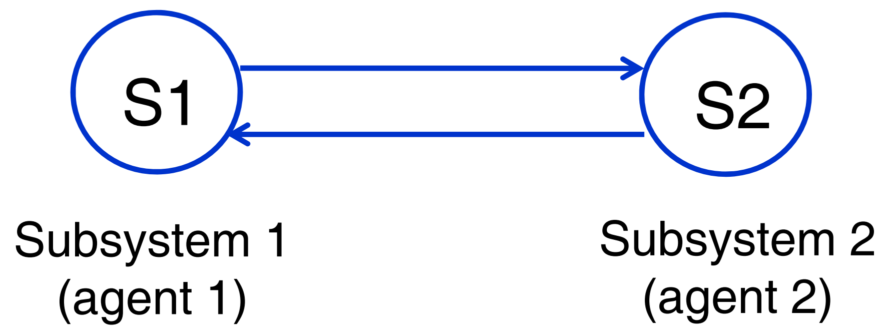


Input-coupled systems

$$S1: \; x_1(k+1) = A_{11}x_1(k) + B_{11}u_1(k) + B_{12}u_2(k)$$
$$S2: \; x_2(k+1) = A_{22}x_2(k) + B_{21}u_1(k) + B_{22}u_2(k)$$

It is possible to reformulate the general system in this way, but a nonminimal representation is obtained

Special cases



State-coupled systems

$$S1:\ x_1(k+1) = A_{11}x_1(k) + A_{12}x_2(k) + B_{11}u_1(k)$$
$$S2:\ x_2(k+1) = A_{21}x_1(k) + A_{22}x_2(k) + B_{22}u_2(k)$$

It is possible to reformulate the general system in this way, but additional delays have to be forced into some input couplings

A remark

Let the original system be described in continuous time

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix}$$

with “sparse” matrices, for example with most of the elements of A_{12}, A_{21}, B_{12}, B_{21} equal to zero (or zero).
Its discretization with the *ZOH* formula, with the Tusting rule or with the backward Euler method leads to a discrete time system with “full” matrices A and B.

The forward Euler method preserves the stucture of the matrices A and B, but can lead to unstable discrete time models even though the original continuous time system is stable.

Is it worth developing new distributed MPC algorithms for continuous time systems?

General MPC problem

$$\min_{u_1,u_2} J = \sum_{j=0}^{N-1} [x_1'(k+j)Q_1 x_1(k+j) + x_2'(k+j)Q_2 x_2(k+j) +$$
$$+ u_1'(k+j)R_1 u_1(k+j) + u_2'(k+j)R_2 u_2(k+j)] +$$
$$+ x_1'(k+N)P_1 x_1(k+N) + x_2'(k+N)P_2 x_2(k+N)$$

under the dynamic constraints, the previous state and control constraints, the terminal constraints

$$x_1(k+N) \in X_{f1} \ , \ x_2(k+N) \in X_{f2}$$

and the additional "mixed" (linear) constraints

$$\begin{bmatrix} H_{x1} & H_{x2} & H_{u1} & H_{u2} \end{bmatrix} \begin{bmatrix} x_1(k+j) \\ x_2(k+j) \\ u_1(k+j) \\ u_2(k+j) \end{bmatrix} \leq c \ , \ j = 0,\ldots,N$$

General MPC problem
remarks

1. the cost function is formally separable

$$J = \sum_{j=0}^{N-1} [x_1'(k+j)Q_1 x_1(k+j) + u_1'(k+j)R_1 u_1(k+j)] + x_1'(k+N)P_1 x_1(k+N) +$$

$$+ \sum_{j=0}^{N-1} [x_2'(k+j)Q_2 x_2(k+j) + u_2'(k+j)R_2 u_2(k+j)] + x_2'(k+N)P_2 x_2(k+N)$$

$$= J_1 + J_2$$

however, in view of the dynamic constraints

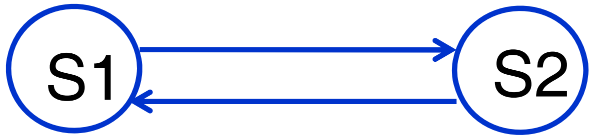
$$J_1 = J_1(x_1, x_2, u_1, u_2) \ , \ J_2 = J_2(x_1, x_2, u_1, u_2)$$

2. ***the terminal cost has a particular structure:***

$$\begin{bmatrix} x_1'(k+N) & x_2'(k+N) \end{bmatrix} \begin{bmatrix} P_1 & 0 \\ 0 & P_2 \end{bmatrix} \begin{bmatrix} x_1(k+N) \\ x_2(k+N) \end{bmatrix}$$

Decentralized/distributed MPC

two main approaches



Subsystem 1 (agent 1) Subsystem 2 (agent 2)

1. Robust MPC

the effect of the other subsystem is viewed as a disturbance to be rejected

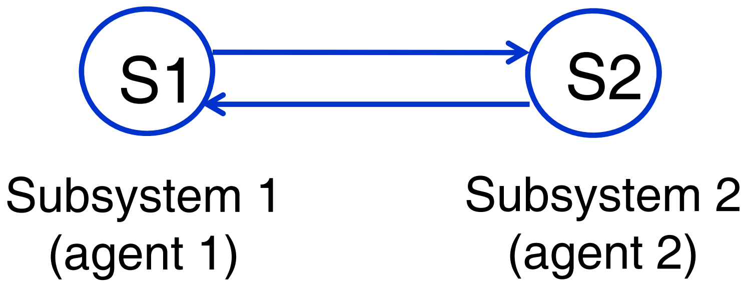
w_1

$$S1: \; x_1(k+1) = A_{11}x_1(k) + A_{12}x_2(k) + B_{11}u_1(k)$$

$$S2: \; x_2(k+1) = A_{21}x_1(k) + A_{22}x_2(k) + B_{22}u_2(k)$$

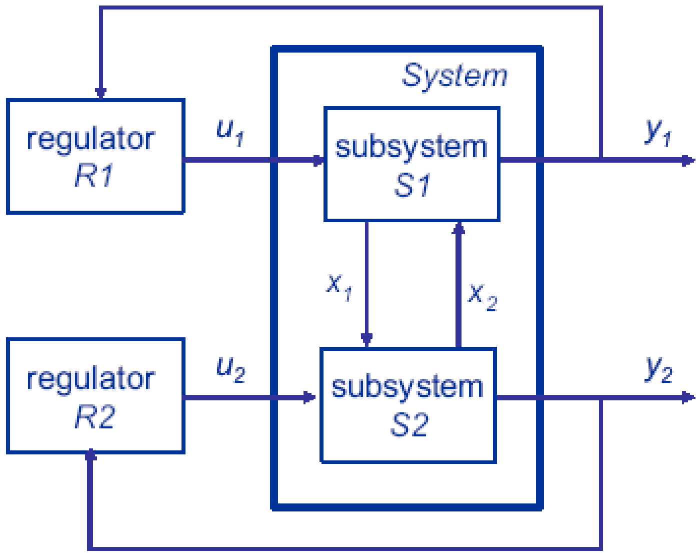
w_2

Decentralized/distributed MPC
two main approaches



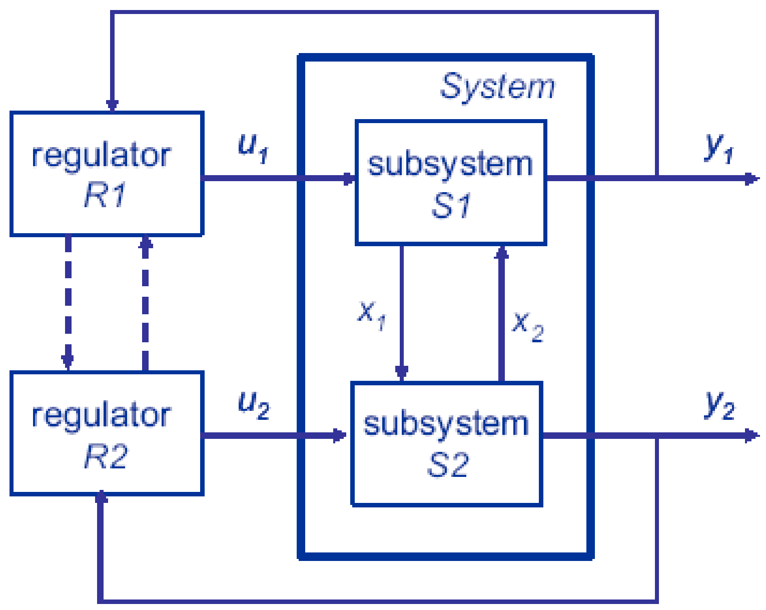
2. “Game-theory” MPC
   the two agents play a game, which can be either cooperative or not (Pareto or Nash equilibra are searched for depending on the local or global objective of each agent, or player)

Decentralized control



- Inputs and outputs are grouped into non overlapping pairs
- Local regulators have limited information on the input /state/outputs
- Decentralization is independent of the complexity of the local regulators (which can rely on the model of the whole system)
- Some systems cannot be stabilized with a decentralized control structure (presence of fixed modes)
- Very few MPC algorithms with guaranteed properties are available

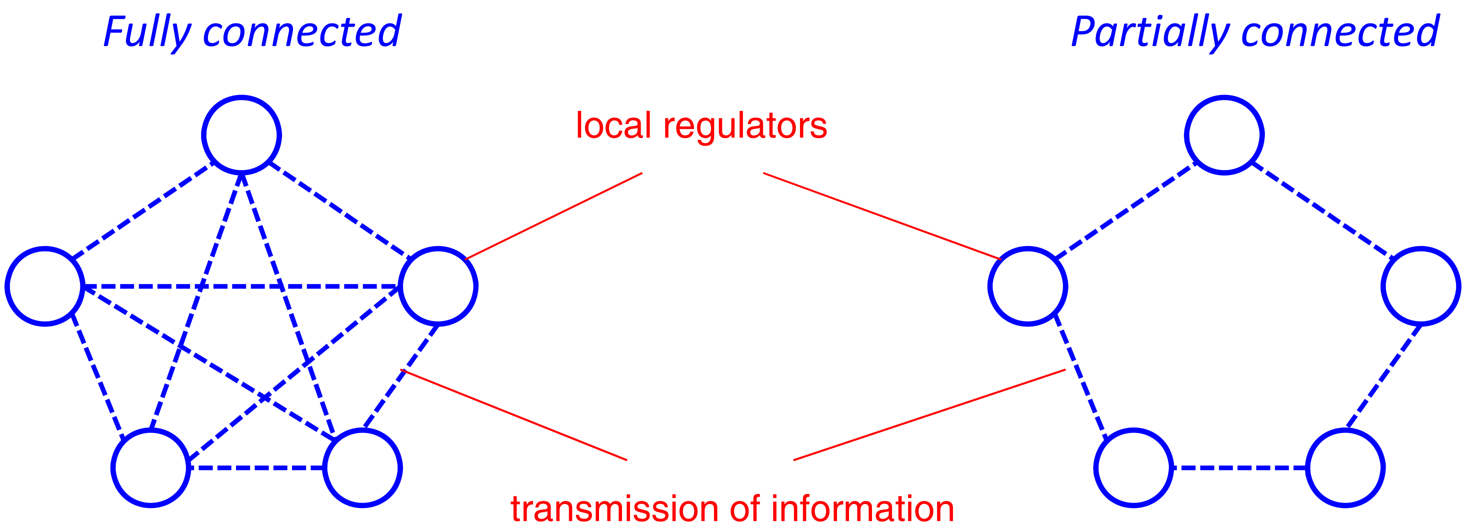
Distributed control



- Information is transmitted among local regulators
- It can concern future (predicted) control sequences or state trajectories
- In the first case, the local regulators must know the whole model of the system

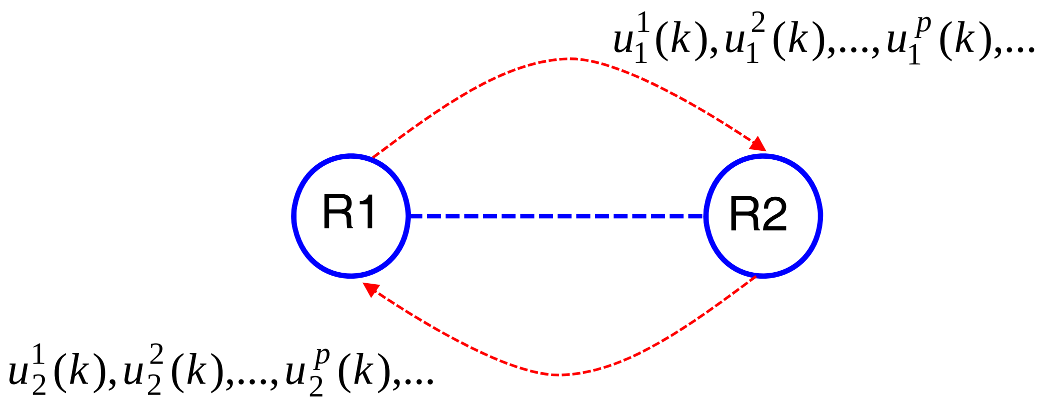
Distributed control algorithms
Properties

- *Fully connected*: all-to-all communication. Information is transmitted from any local regulator to all the others
- *Partially connected*: neighbor-to-neighbor communication. Information is transmitted among the local regulators of subsystems with a direct dynamic influence



Distributed control algorithms
properties

- *Iterative*: multiple transmissions among local regulators within each sampling time
- *Non iterative:* only one iteration within the sampling time



distributed MPC algorithms based on the *game theory* approach are *iterative*, since they must "negotiate" their control action

distributed MPC approaches based on *robust control* are *non iterative*

Distributed control algorithms
properties

- *Cooperating*: each local regulator minimizes a global cost function
- *Independent*: each local regulator minimizes a local cost function

$$J = \sum_{j=0}^{N-1} [x_1'(k+j)Q_1 x_1(k+j) + u_1'(k+j)R_1 u_1(k+j)] + x_1'(k+N)P_1 x_1(k+N) +$$

$$+ \sum_{j=0}^{N-1} [x_2'(k+j)Q_2 x_2(k+j) + u_2'(k+j)R_2 u_2(k+j)] + x_2'(k+N)P_2 x_2(k+N)$$

$$= J_1 + J_2$$

Cooperating

both MPC1 and MPC2 aim at minimizing J_1+J_2

Independent

MPC1 minimizes J_1
MPC2 minimizes J_2

distributed MPC algorithms based on the *game theory* approach can be *cooperating* or *independent*

distributed MPC approaches based on *robust control* are *independent*

Distributed algorithms
an additional classification

The distributed algorithms described in the literature can be divided in two main classes:

- distributed optimization approaches: the goal is to decompose the large-scale optimization problem into smaller subproblems, possibly with a central coordinator (hierarchical structure)

- distributed control algorithms: all the computations are distributed over the local MPC regulators (“flat” structure)

This classification tends to be obsolete, new distributed optimization algorithms do not need a central coordinator

Decentralized and distributed MPC

some “prototype” algorithms

In the following the goal is to present some approaches followed in the development of decentralized and distributed algorithms, focusing on the main ideas rather than on the formal proofs of the corresponding theoretical results

Decentralized MPC

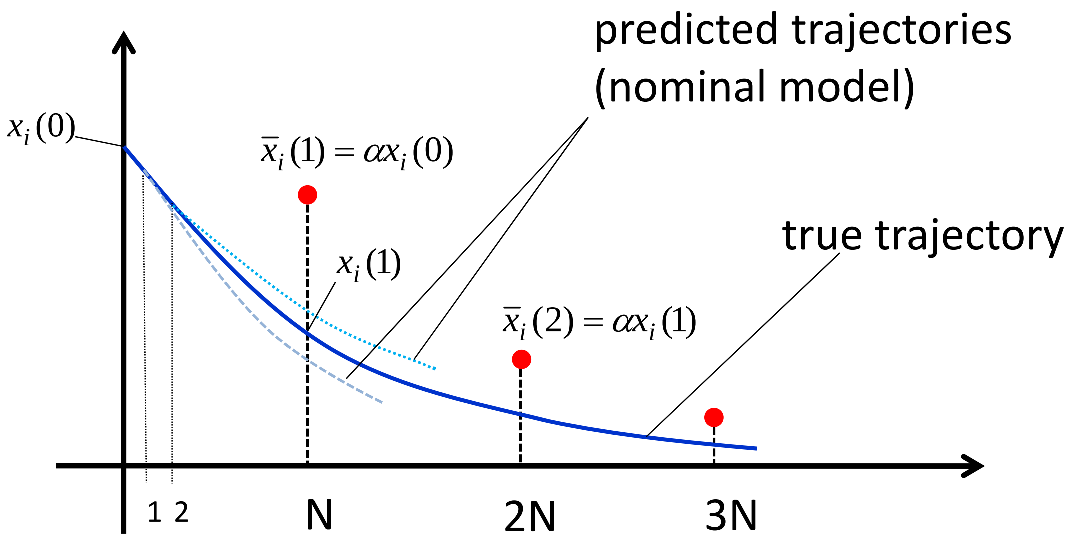
inspired by: L. Magni, R. Scattolini: "Stabilizing Decentralized Model Predictive Control of Nonlinear Systems", *Automatica*, Vol. 42, n. 7, pp. 1231-1236, 2006.

$$S1: \; x_1(k+1) = A_{11}x_1(k) + A_{12}x_2(k) + B_{11}u_1(k) + w_1(k)$$

$$S2: \; x_2(k+1) = A_{21}x_1(k) + A_{22}x_2(k) + B_{22}u_2(k) + w_2(k)$$

asymptotically decaying disturbances

Each local MPC regulator solves a local optimization problem where an additional ***contraction constraint*** is added to force the state trajectory to converge to the origin (despite the effect of the mutual influences and of the disturbances)



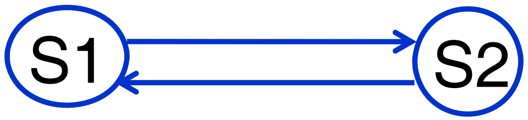
i=1,2

A robust approach (in decentralized control the robust approach seems to be the only possibility)

Distributed MPC

an iterative, non cooperative algorithm

inspired by Rawlings and Mayne: “Model predictive control theory and design”, Nob Hill Pub, 2009



Subsystem 1 (agent 1)

Subsystem 2 (agent 2)

$$S1: \; x_1(k+1) = A_{11}x_1(k) + B_{11}u_1(k) + B_{12}u_2(k)$$

$$S2: \; x_2(k+1) = A_{22}x_2(k) + B_{21}u_1(k) + B_{22}u_2(k)$$

$$J = \rho_1 \left[\sum_{j=0}^{N-1} [x_1^{'}(k+j)Q_1x_1(k+j) + u_1^{'}(k+j)R_1u_1(k+j)] + x_1^{'}(k+N)P_1x_1(k+N) \right] +$$

$$+ \rho_2 \left[\sum_{j=0}^{N-1} [x_2^{'}(k+j)Q_2x_2(k+j) + u_2^{'}(k+j)R_2u_2(k+j)] + x_2^{'}(k+N)P_2x_2(k+N) \right]$$

$$= \rho_1 J_1(x_1, u_1) + \rho_2 J_2(x_2, u_2) \; , \; \rho_1, \rho_2 > 0 \; , \; \rho_1 + \rho_2 = 1$$

Distributed MPC

an iterative noncooperative algorithm

- Inside the sampling time k, many iterations (negotiation) $p=1,2,\ldots,$ are performed between the local subsystems

- At any iteration p, subsystem i, $i=1,2$, solves its own optimization problem by minimizing $J_i(x_i,u_i)$ with respect to the sequence $u_i(k), \ldots, u_i(k+N-1)$ and using the sequence

$$u_j^{p-1}(k),\, u_j^{p-1}(k+1),\, \ldots,\, u_j^{p-1}(k+N-1)$$

computed by the other subsystem at the previous iteration

- Letting $u_i^o(k),\, u_i^o(k+1),\, \ldots,\, u_i^o(k+N-1)$ be the optimal sequence, it is set

$$u_i^p(k+l) = w_i u_i^o(k+l) + (1-w_i) u_i^{p-1}(k+l)\ ,\ l=0,\ldots,N-1,\ \ 0<w_i<1$$

Distributed MPC
an iterative noncooperative algorithm

Main problems

- Inside the sampling time, convergence of the iterations is not guaranteed

- Even if convergence is obtained, the resulting control law can be not stabilizing

The adopted noncooperative approach leads to a Nash equilibrium and does not guarantee any stability result

Distributed MPC

*an iterative, **cooperative** algorithm*

inspired by Rawlings and Mayne: "Model predictive control theory and design", Nob Hill Pub, 2009

The only difference with respect to the previous iterative algorithm is that each subsystem minimizes the overall performance index

$$J = \rho_1 J_1(x_1, u_1) + \rho_2 J_2(x_2, u_2) \ , \ \rho_1, \rho_2 > 0 \ , \ \rho_1 + \rho_2 = 1$$

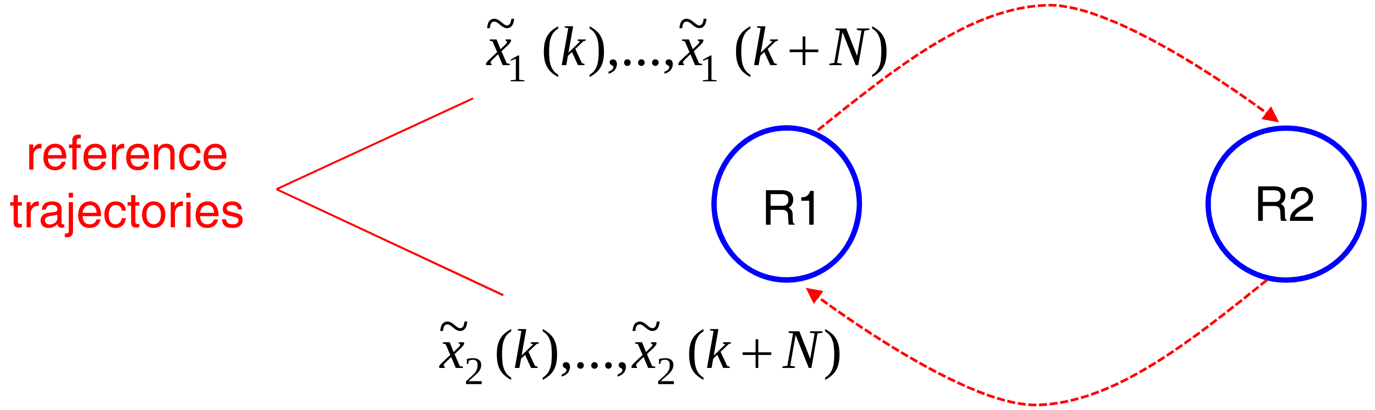
Closed-loop stability can be achieved even with a limited number of iterations *p* inside the sampling time

Distributed MPC

a noniterative, noncooperative algorithm with neighbor-to-neighbor communication (DPC)

inspired by Farina and Scattolini: "Distributed non-cooperative MPC with neighbor-to-neighbor communication ", *IFAC World Congress*, 2011

A robust approach is used



$$S1:\ x_1(k+1) = A_{11}x_1(k) + A_{12}\tilde{x}_2(k) + B_{11}u_1(k) + \underbrace{A_{12}\left(x_2(k) - \tilde{x}_2(k)\right)}_{w_1(k)}$$

$$S2:\ x_2(k+1) = A_{21}\tilde{x}_1(k) + A_{22}x_2(k) + B_{22}u_2(k) + \underbrace{A_{21}\left(x_1(k) - \tilde{x}_1(k)\right)}_{w_2(k)}$$

Distributed MPC
DPC - 2

Each subsystem *i=1,2* guarantees that $x_i(k) - \tilde{x}_i(k) \in \Phi_i$

Therefore $w_i(k) \in W_i = A_{ij}\Phi_j$

and the design problem is transformed in a couple of standard robust design problems for the two subsystems

$$S1:\ x_1(k+1) = A_{11}x_1(k) + A_{12}\tilde{x}_2(k) + B_{11}u_1(k) + w_1(k)\ ,\ w_1 \in W_1$$
$$S2:\ x_2(k+1) = A_{21}\tilde{x}_1(k) + A_{22}x_2(k) + B_{22}u_2(k) + w_2(k)\ ,\ w_2 \in W_2$$

The "tube approach" can be used to derive a distributed and stabilizing solution

Distributed MPC
DPC - 3

Assume that $x_i(k) - \tilde{x}_i(k) \in \Phi_i \;\; , \;\; \forall k \geq 0$

Define the nominal model for the i-th ($i=1,2$) subsystem

$$x_{n_i}(k+1) = A_{ii} x_{n_i}(k) + B_i u_{n_i}(k) + A_{ij} \tilde{x}_j(k)$$

and the control law $u_i(k)=u_{ni}(k)+K_i(x_i(k)-x_{ni}(k))$
where K_i is a stabilizing gain, i.e. $A_{iK}=A_{ii}+B_iK_i$ is stable, with the additional property that, letting $K=diag(K_i)$, $A+BK$ is stable

The “error” model is

$$e_i(k+1) = x_i(k+1) - x_{n_i}(k+1) = (A_{ii} + B_i K_i) e_i(k) + w_i(k)$$

For this system let E_i be a robust positive invariant set, i.e.

$$e_i(k) \in E_i \Rightarrow e_i(k+l) \in E_i \;\; , \;\; l > 0$$

and define the set Ω_i such that $\Omega_i + E_i \subseteq \Phi_i$

Distributed MPC

DPC - 4

The MPC problem for the *i-th* system at time *k* can be formulated as

$$\min_{u_{n_i}, x_{n_i}(k)} J_i = \sum_{l=0}^{N-1} [x_{n_i}'(k+l) Q_i x_{n_i}(k+l) + u_{n_i}'(k+l) R_i u_{n_i}(k+l)] + V_{fi}(x_{n_i}(k+N))$$

$$x_{n_i}(k+1) = A_{ii} x_{n_i}(k) + B_i u_{n_i}(k) + A_{ij} \tilde{x}_j(k)$$

$$x_i(k) - x_{n_i}(k) \in E_i$$

$$x_{n_i}(k+l) - \tilde{x}_i(k+l) \in \Omega_i \quad , \quad l = 0,1,\ldots$$

$$x_{n_i}(k+l) \in X_i \ominus E_i$$

$$u_{n_i}(k+l) \in U_i \ominus K_i E_i$$

$$x_{n_i}(k+N) \in X_{nfi}$$

The solution of this problem allows one to compute $x_{ni}(k+N)$, which is used to define the reference trajectory for the time instant $k+1$

$$\tilde{x}_i(k) \quad \tilde{x}_i(k+1) \ \ldots \ \tilde{x}_i(k+N-2) \quad \tilde{x}_i(k+N-1)$$

$$\tilde{x}_i(k+1) \quad \tilde{x}_i(k+2) \ \ldots \ \tilde{x}_i(k+N-1) \quad \tilde{x}_i(k+N) = x_{n_i}(k+N)$$

Distributed MPC
DPC - 5

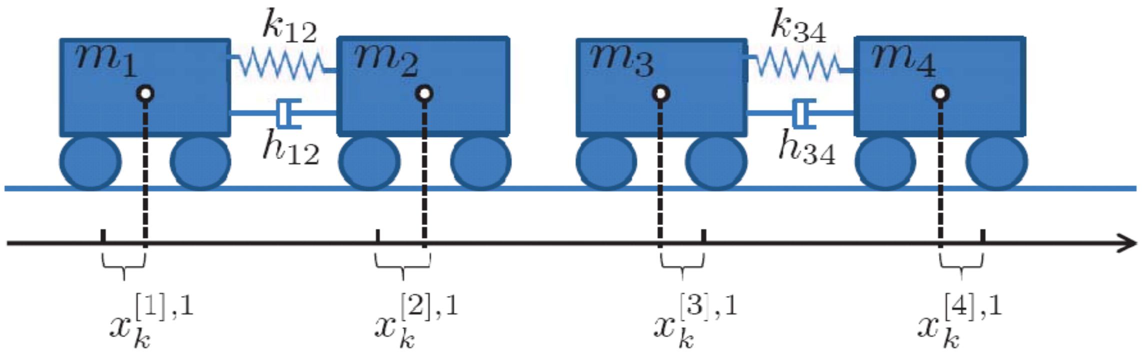
Properties

A proper choice of the design parameters allows one to guarantee:

- *Feasibility:* at any time instant (provided that an initial feasible solution is available at *k=0, as standard in MPC)*
- *Convergence:* of the state trajectories to the origin
- *Mixed constraints:* it is possible to include in the problem formulation joint constraints on the states of different systems without destroying the other properties
- *Local knowledge:* any subsystem does not need to know the model of the other subsystems
- *Communication requirements:* Neighbor-to-neighbor communication, noniterative scheme
- *Scalability (**plug and play**):* as the number of subsystems grows, the information to be stored, processed and transmitted by the subsystems not directly connected to the new ones remains constant

Distributed MPC

DPC - 6



Scope: move the trucks from the initial conditions $x_0^{[i]}$ to the equilibrium condition $[0\ 0]^T$ subject to constraints:

$$|u_k^{[i]}| \leq 1 \text{ for } i = 1, \ldots, 3 \text{ and } |u_k^{[4]}| \leq 2$$

$$|x_k^{[i],1} - x_k^{[i+1],1}| \leq 2.1$$

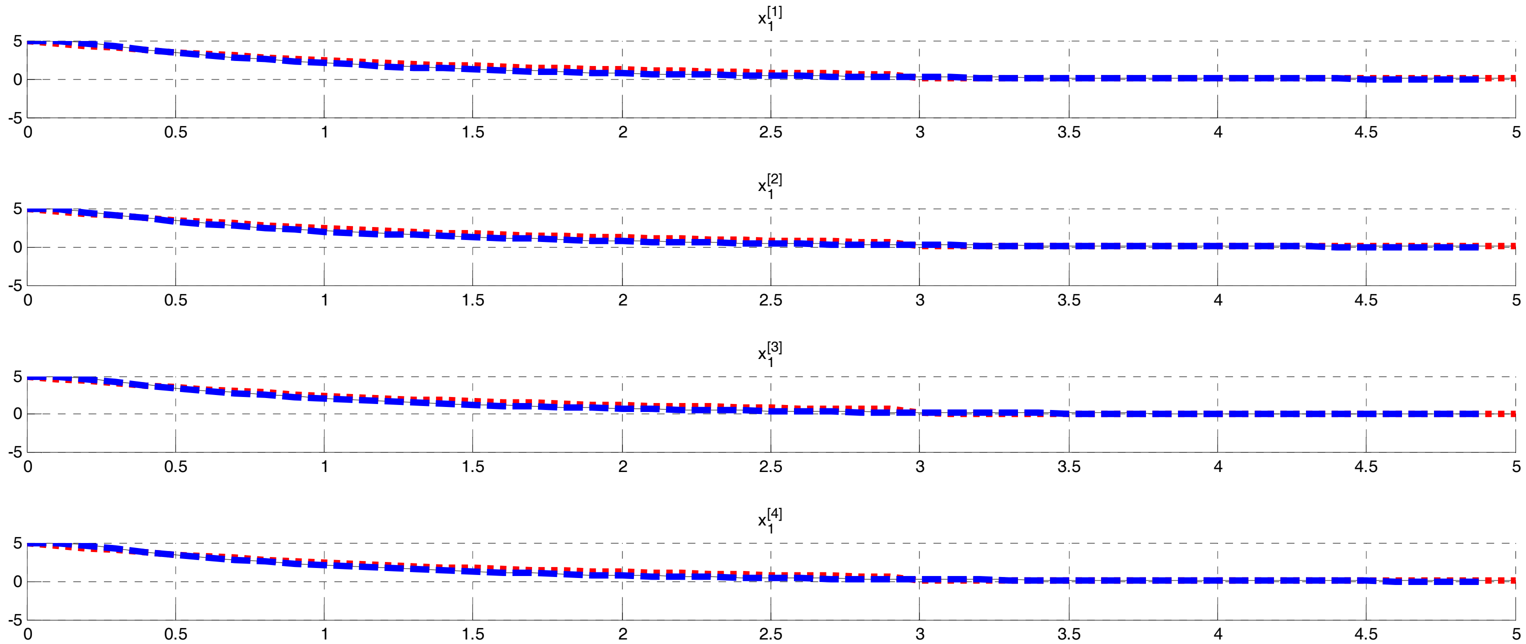
Data:

$$m_1 = 3,\ m_2 = 2,\ m_3 = 3,\ m_4 = 6$$

$$x_0^{[i]} = [5\,,0]^T$$

Distributed MPC
DPC - 7

Position of the charts (dotted red line – reference)



DPC

a simplified implementation

- The main drawback of DPC is its heavy off-line design (computation of the sets E_i, Ω_i, Φ_i)
- A simple, yet effective in many cases, implementation consists in simply using the nominal model and the prediced state trajectories of the neighborhods without any "set constrains", but with restriced state and control constraints

$$\min_{u_{n_i}} J_i = \sum_{l=0}^{N-1} [x_{n_i}'(k+l) Q_i x_{n_i}(k+l) + u_{n_i}'(k+l) R_i u_{n_i}(k+l)] + V_{fi}(x_{n_i}(k+N))$$

$$x_{n_i}(k+1) = A_{ii} x_{n_i}(k) + B_i u_{n_i}(k) + A_{ij} \tilde{x}_j(k)$$

$$x_{ni}(k) = x_i(k)$$

$$x_{n_i}(k+l) \in X_i$$

$$u_{n_i}(k+l) \in U_i$$

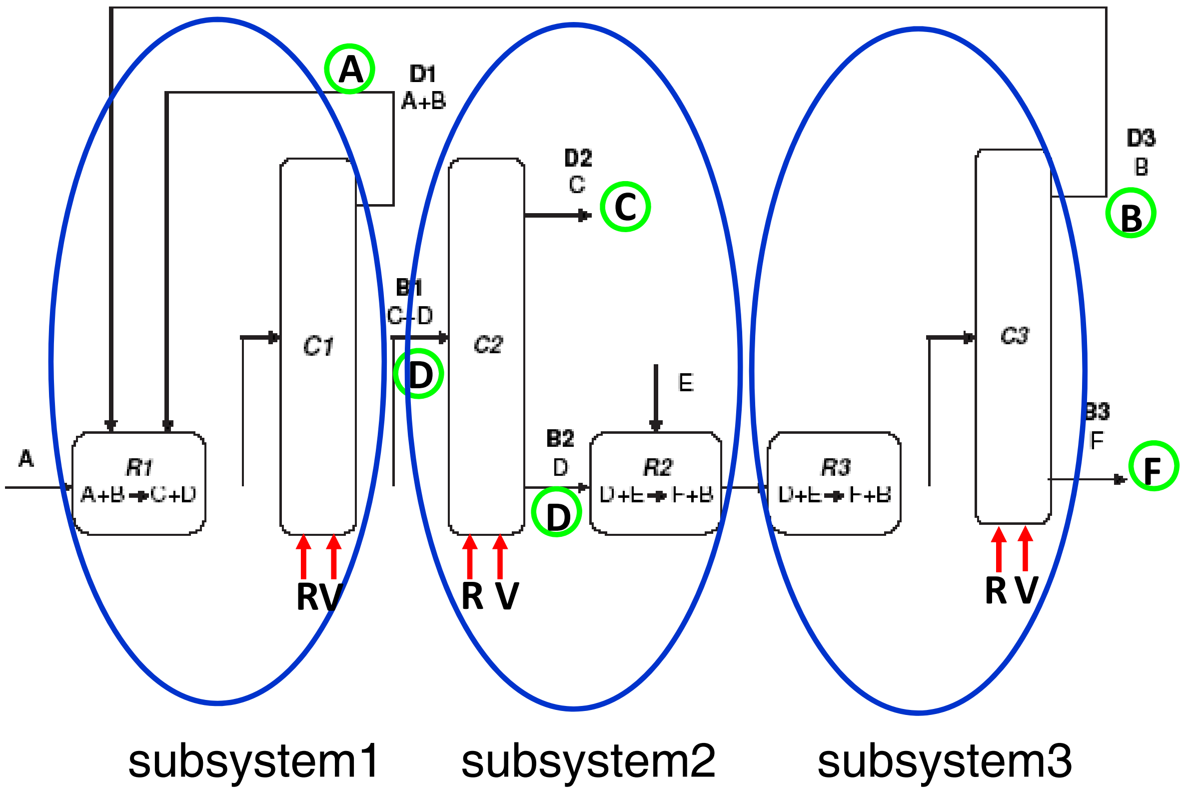
$$x_{n_i}(k+N) \in X_{nfi}$$

$$u_i(k) = u_{ni}(k)$$

X_i, U_i, X_{nfi} are "reduced sets"

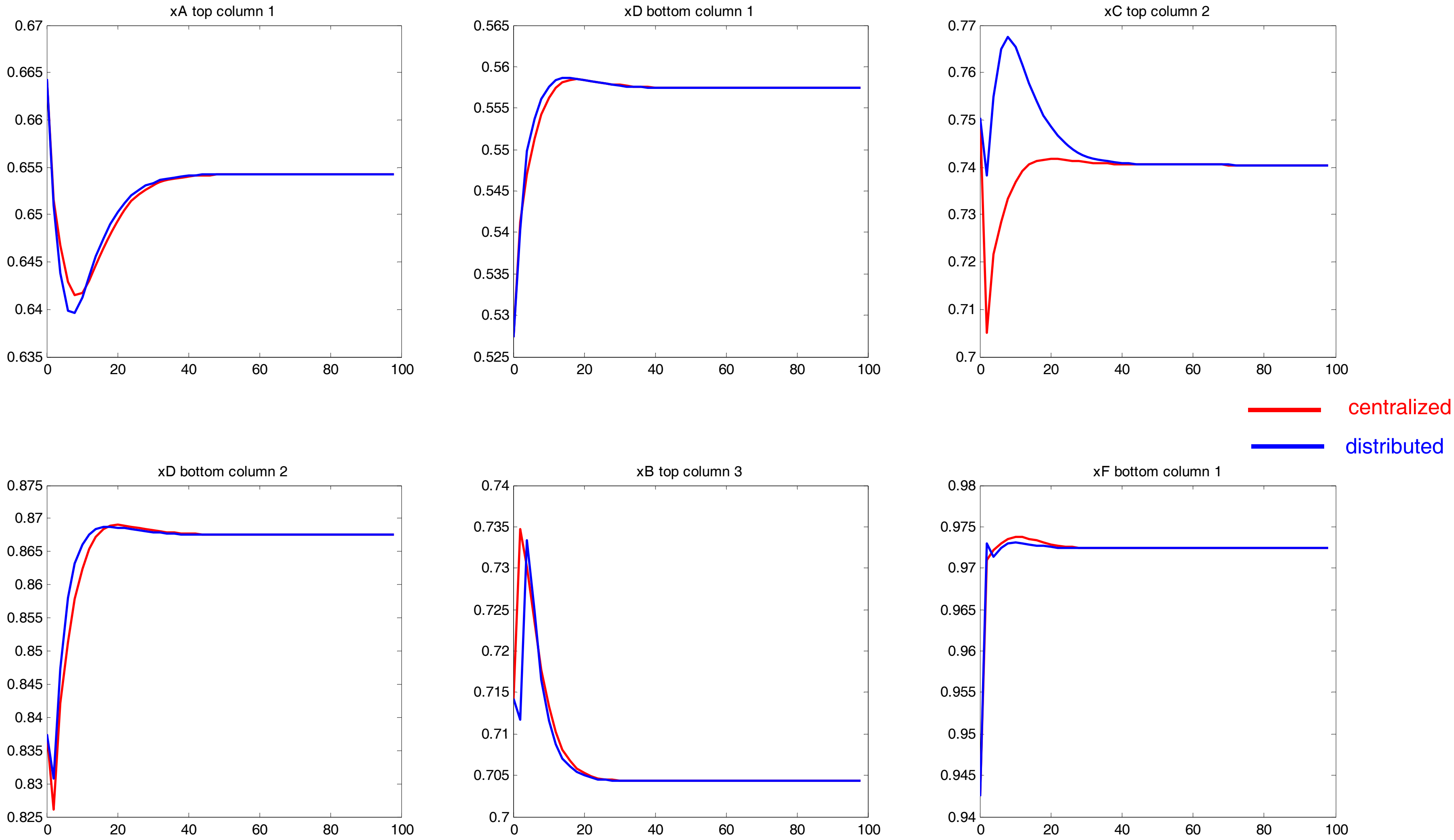
no stability properties
standard MPC implementation

simplified distributed MPC



- 187 state variables (62 – 58 – 67)
- 6 inputs and 6 outputs (2 for each subsystem)
- stabilization problem (initial state $\neq$ equilibrium state)
- reference trajectories generated from the current state to the equilibrium as exponentials
- linear centralized and distributed MPC

Simulation results



Thanks to

- Daniele Balzaretti
- Marcello Farina
- Bruno Picasso
- Carlo Savorgnan
- The HD-MPC European project and all its participants

  …

- *and to all of you for your attention*