Distributed MPC: a comprehensive overview and some recent advances

Gabriele Pannocchia
University of Pisa

May 31, 2016
MPC is the most successful advanced control technique applied in the process industries.

- Closed-loop stability
- Constraint satisfaction
- Robustness against modeling errors and disturbances
- Setpoint tracking (often hierarchically)

MPC enhances the process profitability.
Large scale systems: some examples

Large industrial plants

Power generation networks

Systems of Systems
MPC for large scale systems: different approaches

One or more MPCs?

☑ One **centralized MPC**
  • Pros: global *(plant-wide)* optimality, stability
  • Cons: limited flexibility, high computational cost

☑ Several **decentralized MPCs**
  • Pros: lower computational cost, high modularity
  • Cons: global suboptimality, stability issues

Middle field...

☑ Distributed **MPC**
  • Pros: high modularity, stability
  • Pros/Cons: global optimality (high computational cost)
Outline

1. Comprehensive overview of (linear) distributed MPC algorithms
   • Models
   • Decentralized, Cooperative and Non-Cooperative MPC
   • Design and properties of Cooperative MPC
   • Cooperative MPC for tracking
2. Advances in cooperative MPC for tracking
   • Proposed approaches
   • Application results
3. Conclusions
Part I
A comprehensive overview of distributed MPC
Models for distributed MPC - I

Overall DLTI system

\[ x^+ = A x + B u \]
\[ y = C x \]

- \( x \in \mathbb{R}^n \): current state
- \( x^+ \in \mathbb{R}^n \): successor state
- \( u \in \mathbb{R}^m \): manipulated input
- \( y \in \mathbb{R}^p \): controlled output

Local DLTI subsystems

\[ x_i^+ = A_i x_i + B_i u_i + \sum_{j \in \mathcal{N}_i} B_{ij} u_j \]
\[ y_i = C_i x_i \]

- \( x_i, x_i^+ \in \mathbb{R}^{n_i} \): current/successor \( i \)-th state
- \( u_i \in \mathbb{R}^{m_i} \): manipulated \( i \)-th input
- \( y_i \in \mathbb{R}^{p_i} \): controlled \( i \)-th output

- \( M \): number of subsystems
- \( \mathcal{N}_i \): set of neighbors of subsystem \( i \)
Interconnected systems and neighbors definition

Overall system

- **S₁**
  - \( \mathcal{N}_1 = \{2, 3\} \)
- **S₂**
  - \( \mathcal{N}_2 = \{1\} \)
- **S₃**
  - \( \mathcal{N}_3 = \{2\} \)

Connections:
- **u₁** → **S₁**
- **u₂** → **S₂**
- **u₃** → **S₃**
- **y₁** ← **S₁**
- **y₂** ← **S₂**
- **y₃** ← **S₃**
Why not coupling through states?

☑️ The state of each subsystem includes all modes from local and neighboring inputs to local output.

- An example of 2 subsystems "coupled by states":

\[
\begin{align*}
\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^+ &= \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} B_{11} & 0 \\ 0 & B_{22} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \\
\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} &= \begin{bmatrix} C_{11} & 0 \\ 0 & C_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}
\end{align*}
\]

- Equivalent subsystems coupled by inputs:

Subsystem 1: \( x_1 \leftarrow \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \), \( A_1 \leftarrow \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix} \), \( B_1 \leftarrow \begin{bmatrix} B_{11} \\ 0 \end{bmatrix} \), \( B_{12} \leftarrow \begin{bmatrix} 0 \\ B_{22} \end{bmatrix} \), \( C_1 \leftarrow \begin{bmatrix} C_{11} & 0 \end{bmatrix} \)

Subsystem 2: \( x_2 \leftarrow \begin{bmatrix} x_2 \end{bmatrix} \), \( A_2 \leftarrow \begin{bmatrix} A_{22} \end{bmatrix} \), \( B_2 \leftarrow \begin{bmatrix} B_{22} \end{bmatrix} \), \( B_{21} \leftarrow [] \), \( C_2 \leftarrow \begin{bmatrix} C_{22} \end{bmatrix} \)

☑️ Models obtained from ID naturally have input couplings.
Taxonomy

Based on dynamics and objective of local controllers

- **Centralized MPC**: a single MPC computes all inputs to optimize a global objective
- **Decentralized MPC**: each MPC computes its local input, disregarding interacting dynamics, to optimize a local objective
- **Non-cooperative MPC**: each MPC computes its local input, considering interacting dynamics, to optimize a local objective
- **Cooperative MPC**: each MPC computes its local input, considering interacting dynamics, to optimize a global objective
Decentralized MPC

Model (decentralized)

\[ x_i^{+} = A_i x_i + B_i u_i + \sum_{j \in \mathcal{N}_i} B_{ij} u_j \]
\[ y_i = C_i x_i \]

Cost (local)

\[ V_i = \sum_{k=0}^{N-1} (x_i(k)^T Q_i x_i(k) + u_i(k)^T R_i u_i(k)) + x_i(N)^T P_i x_i(N) \]

Reference Scheme
Non-cooperative Distributed MPC

Model (interacting)
\[ x_i^+ = A_i x_i + B_i u_i + \sum_{j \in \mathcal{N}_i} B_{ij} u_j \]
\[ y_i = C_i x_i \]

Cost (local)
\[ V_i = \sum_{k=0}^{N-1} (x_i(k)^T Q_i x_i(k) + u_i(k)^T R_i u_i(k)) + x_i(N)^T P_i x_i(N) \]

Reference Scheme
Cooperative Distributed MPC

Model (interacting)
\[
x_i^+ = A_i x_i + B_i u_i + \sum_{j \in \mathcal{N}_i} B_{ij} u_j
\]
\[
y_i = C_i x_i
\]

Cost (global)
\[
V = \sum_{j=1}^{M} \rho_j V_i = \sum_{k=0}^{N-1} (x(k)^T Q x(k) + u(k)^T R u(k)) + x(N)^T P x(N)
\]

Reference Scheme
Properties of different architectures

- **Decentralized** MPC
  - Due to neglected dynamics, **stability** (as well feasibility, tracking, etc.) are not guaranteed
  - Possible remedies are based on robust (tube) MPC paradigms [Alessio et al., 2011; Riverso et al., 2013]

- **Noncooperative** distributed MPC
  - Due to local objectives, **convergence** of iterations and **stability** is not guaranteed
  - When convergence occurs, **Nash equilibrium** is reached. Still, stability may not hold [Rawlings and Mayne, 2009]

- **Cooperative** distributed MPC
  - **Convergence** of iterations and **stability** is guaranteed [Rawlings and Mayne, 2009]
  - **Global optimality** can be guaranteed [Stewart et al., 2011]
Cooperative distributed MPC - I

FHOCOCP and cooperative iterations

Each local MPC, knowing candidate sequences of all other MPCs, solves the FHOCOCP

\[ \mathbb{P}_i \left( x, \{ u_j \}_{j \neq i} \right) : \min_{u_i} V(x, u) \quad \text{s.t.} \]

\[ u_i \in \mathcal{U}_i \left( x, \{ u_j \}_{j \neq i} \right), \quad x(N) \in X_f \]

Input feasibility space

\[ \mathcal{U}_i \left( x, \{ u_j \}_{j \neq i} \right) = \{ u_i | u_i(k) \in \mathcal{U}_i, x(k) \in X \} \]

Cooperative iterations, given \( u_i^0 \): solution to \( \mathbb{P}_i \left( x, \{ u_j^{[q-1]} \}_{j \neq i} \right) \)

\[ u_i^{[q]} = w_i u_i^0 + (1 - w_i) u_i^{[q-1]} \]

Keep track of overall state dynamics
Cooperative distributed MPC - II

Main properties

- **Feasibility** of each iterate
  \[ u_i^{[q-1]} \in \mathcal{U}_i^N \Rightarrow u_i^{[q]} \in \mathcal{U}_i^N, \text{ for all } i = 1, \ldots, M \text{ and } q \in \mathbb{I}_{\geq 0} \]

- **Cost decrease** at each iteration
  \[ V(x(0), u^{[q]}) \leq V(x(0), u^{[q-1]}) \text{ for all } q \in \mathbb{I}_{\geq 0} \]

- **Cost convergence** to centralized optimum
  \[ \lim_{q \to \infty} V(x(0), u^{[q]}) = \min_{u \in \mathcal{U}^N} V(x(0), u) \]

- **Stability**, for any finite \( q \), is proved via suboptimal MPC arguments

Can reduce computational requirements
Tracking in centralized MPC

Equilibrium target

Any equilibrium solves:

\[
\begin{bmatrix}
A - I & B & 0 \\
C & 0 & -I
\end{bmatrix}
\begin{bmatrix}
x_s \\
u_s \\
y_s
\end{bmatrix} = \begin{bmatrix} 0 \\
0 \end{bmatrix}
\]

Given desired setpoint \( y_t \), the optimal equilibrium is:

\[
\min_{x_s, u_s, y_s} V_{ss}(y_s, y_t) \quad \text{s.t. above constraint and } x_s \in \mathbb{X}, \quad u_s \in \mathbb{U}
\]

Tracking MPC problem

Deviation variables: \( \tilde{x} = x - x_s^0, \quad \tilde{u} = u - u_s^0 \)

\[
\mathbb{P}(\tilde{x}) : \quad V^0(\tilde{x}) = \min_{\tilde{u}} \left\{ V(\tilde{x}(0), \tilde{u}) \mid \tilde{u} \in \tilde{U}_N(\tilde{x}) \right\}
\]

Single step approaches are also possible [Limon et al., 2008]
Distributed cooperative MPC for tracking - I

Single step approach [Ferramosca et al., 2013]

**Centralized** cost function, including centralized target

\[
V_t(x, u, x_s, u_s, y_s) = \sum_{k=0}^{N-1} \ell(x(k) - x_s, u(k) - u_s) + V_f(x(N) - x_s)
\]

\[+ V_{ss}(y_s, y_t)\]

s.t. \(x(0) = x\)

\(x(k+1) = Ax(k) + Bu(k)\)

\[
\begin{bmatrix}
A - I & B & 0 \\
C & 0 & -I
\end{bmatrix}
\begin{bmatrix}
x_s \\
u_s \\
y_s
\end{bmatrix} = \begin{bmatrix}0 \\0\end{bmatrix}
\]
FHOCP and cooperative iterations

Each local MPC, knowing candidate sequences of all other MPCs, solves the FHOCP

\[
P_i \left( x, \{u_j\}_{j \neq i} \right) : \min_{u_i, x_s, u_s, y_s} V_t \left( x, u, x_s, u_s, y_s \right) \quad \text{s.t.} \quad u_i \in \mathcal{U}_i \left( x, \{u_j\}_{j \neq i} \right), \quad (x(N), y_s) \in \Omega
\]

\( \Omega \) is an invariant set for tracking [Ferramosca et al. 2013]

Cooperative iterations, given \( u_i^0 : \text{ solution to } P_i \left( x, \{u_j\}_{j \neq i} \right) \)

\[
u_i^{[q]} = w_i u_i^0 + (1 - w_i) u_i^{[q-1]} \quad \text{for } q = 1, 2, \ldots
\]

Need to keep track of centralized state dynamics
Part II

Advances in cooperative MPC for tracking
Some reminders of graph theory

**Useful definitions**

- A graph $G = (\mathcal{V}, \mathcal{E})$ : set of vertices $\mathcal{V}$ and edges $\mathcal{E}$
- A **directed graph** is composed by oriented edges
- **Inlet** star: $S_{i}^{IN} = \{v_j \in \mathcal{V} \mid (v_j, v_i) \in \mathcal{E}\}$
- **Outlet** star: $S_{i}^{OUT} = \{v_j \in \mathcal{V} \mid (v_i, v_j) \in \mathcal{E}\}$

**An example**

\[
\begin{align*}
x_1^+ &= A_1 x_1 + B_1 u_1 \\
x_2^+ &= A_2 x_2 + B_2 u_2 + B_{21} u_1 \\
x_3^+ &= A_3 x_3 + B_3 u_3 + B_{31} u_1 + B_{32} u_2
\end{align*}
\]

\[
S_1^{IN} = \emptyset, \quad S_1^{OUT} = \{2, 3\}, \quad S_2^{IN} = \{1\}, \quad S_2^{OUT} = \{3\}, \quad S_3^{IN} = \{1, 2\}, \quad S_3^{OUT} = \emptyset
\]
The augmented system - I

Key observations

- The evolution of $i$-th subsystem is **influenced** by inputs of subsystems in its **inlet star**
- The input of $i$-th subsystem **influences** evolution of subsystems in its **outlet star**

\[
\begin{align*}
x_i^+ &= A_i x_i + B_i u_i + \sum_{k \in S_i^{IN}} B_{ik} u_k \\
x_j^+ &= A_j x_j + B_{ji} u_i + \left( B_j u_j + \sum_{k \in S_j^{IN} \setminus \{i\}} B_{jk} u_k \right), & j \in S_i^{OUT}
\end{align*}
\]

- Other subsystems are **not affected** by the $i$-th subsystem input

\[
\begin{align*}
x_j^+ &= A_j x_j + \left( B_j u_j + \sum_{k \in S_j^{IN}} B_{jk} u_k \right), & j \notin S_i^{OUT}
\end{align*}
\]
The augmented system - II

Augmented i-th subsystem

Augmented inlet star

$S_i^{IN} \leftarrow S_i^{IN} \cup S_i^{OUT} \cup \left( \bigcup_{j \in S_i^{OUT} \setminus \{i\}} S_j^{IN} \right)$

Augmented local variables and matrices

$\tilde{x}_i = \begin{bmatrix} x_i \\ [x_j]_{j \in S_i^{OUT}} \end{bmatrix}, \quad \tilde{u}_i = [u_k]_{k \in S_i^{IN}}, \quad \tilde{y}_i = \begin{bmatrix} y_i \\ [y_j]_{j \in S_i^{OUT}} \end{bmatrix}$

$\bar{A}_i = \text{diag}\left\{ A_i, \{A_j\}_{j \in S_i^{OUT}} \right\}, \quad \bar{B}_i = \begin{bmatrix} B_i \\ [B_{ji}]_{j \in S_i^{OUT}} \end{bmatrix}, \quad \bar{B}_i^{IN} = \text{hor}\{B_{ik}\}_{k \in S_i^{IN}}, \text{hor}\{B_{jk}\}_{j \in S_i^{OUT}, k \in S_i^{IN}}$

Final augmented dynamics

$\tilde{x}_i^+ = \bar{A}_i \tilde{x}_i + \bar{B}_i \tilde{u}_i + \bar{B}_i^{IN} \tilde{u}_i$

$\bar{y}_i = \bar{C}_i \tilde{x}_i$
Augmented system based cooperative MPC for tracking - I

Local (augmented system) cost function

The global cost function can be reduced to a local (augmented) cost:

$$V_{ti}(\cdot) = \sum_{k=0}^{N-1} \ell_i(\bar{x}_i(k) - \bar{x}_s, u_i(k) - u_s) + \bar{V}_{fi}(\bar{x}_i(k) - \bar{x}_s) + \|y_s - y_t\|^2_T$$

s.t.

$$\bar{x}_i(0) = \bar{x}_i$$

$$\bar{x}_i(k + 1) = \bar{A}_i \bar{x}_i(k) + \bar{B}_i u_i(k) + \bar{B}_i^{IN} \bar{u}_i(k)$$

Local (augmented) cost functions:

$$\bar{\ell}_i(\bar{x}_i, u_i) = \frac{1}{2} (\bar{x}_i^T \bar{Q}_i \bar{x}_i + u_i^T \bar{R}_i u_i)$$

$$\bar{V}_{fi}(\bar{x}_i) = \frac{1}{2} \bar{x}_i^T \bar{P}_i \bar{x}_i$$

Discarded terms in the global function are not affected by $u_i$.
Augmented system based cooperative MPC for tracking - II

FHOCP and cooperative iterations

☑ Each local MPC, knowing candidate sequences of all MPCs in its inlet star, solves the FHOCP

\[ P_i \left( \bar{x}_i, \{u_j\}_{j \in S_i^{IN}} \right): \quad \min_{u_i, x_s, u_s, y_s} V_{ti}(u_i, x_s, u_s, y_s) \quad \text{s.t.} \]

\[ u_i \in \tilde{U}_i \left( \bar{x}_i, \{u_j\}_{j \in S_i^{IN}} \right), \quad (\bar{x}_i(N), \bar{y}_si) \in \tilde{\Omega}_i \]

☑ Input feasibility space

\[ \tilde{U}_i \left( \bar{x}_i, \{u_j\}_{j \in S_i^{IN}} \right) = \{u_i \mid u_i(k) \in U_i, \bar{x}_i(k) \in \bar{X}_i\} \]

☑ Cooperative iterations:

\[ u_i^{[q]} = w_i u_i^0 + (1 - w_i) u_i^{[q-1]} \]

Need to keep track of augmented state dynamics only
Overall algorithm

**Require:** Augmented subsystems, \( S_i^{IN} \forall i = 1 \ldots M \), tolerance \( \epsilon \), maximum no. cooperative iterations \( q_{\text{max}} \), convex combination weights \( w_i > 0 \), such that \( \sum_{i=1}^{M} w_i = 1 \).

1: Set \( q \leftarrow 0 \) and \( e_i \leftarrow 2\epsilon \).
2: \textbf{while} \( q < q_{\text{max}} \) and \( \exists i \) such that \( e_i > \epsilon \) \textbf{do}
3: \quad \( q \leftarrow q + 1 \)
4: \quad \textbf{for} \( i = 1 \) \textbf{to} \( M \) \textbf{do}
5: \quad \quad \text{Solve problem} \( \mathbb{P}_i \) \text{ to obtain the optimal input sequence} \( u_i^0(x) \) \text{ and the centralized state-steady triple} \( (x_s, u_s, y_s) \).
6: \quad \quad \textbf{if} \( q = 1 \) \textbf{then}
7: \quad \quad \quad \quad u_i^{[q-1]} = \begin{bmatrix} u_{s_i}^T & \ldots & u_{s_i}^T \end{bmatrix}^T
8: \quad \quad \textbf{end if}
9: \quad \quad \text{Define new iterate:} \( u_i^{[q]} = w_i u_i^0 + (1 - w_i) u_i^{[q-1]} \).
10: \quad \text{Compute convergence error:} \( e_i = \frac{||u_i^{[q]} - u_i^{[q-1]}||}{1 + ||u_i^{[q]}||} \).
11: \quad \textbf{end for}
12: \textbf{end while}
13: \textbf{return} \quad \text{Overall solution} \( u = (u_1^{[q]}, u_2^{[q]}, \ldots, u_M^{[q]}) \).
A two-step variant

Step 1
☑ Solve the centralized target problem to obtain \((x_s, u_s, y_s)\)

Step 2
☑ Each local MPC, knowing candidate sequences of all MPCs in its inlet star, solves the FHOCP with known \((x_s, u_s, y_s)\)
☑ Cooperative iterations:
\[
u_i^{[q]} = w_i u_i^0 + (1 - w_i) u_i^{[q-1]}\]
### Complexity of different methods

**Three alternatives**

- **DMPC0**: available method [Ferramosca et al., 2013]
- **DMPC1**: proposed method (single step)
- **DMPC2**: proposed method (two steps)

<table>
<thead>
<tr>
<th></th>
<th>DMPC0</th>
<th>DMPC1</th>
<th>DMPC2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Prediction model</strong></td>
<td>Centralized</td>
<td>Augmented</td>
<td>Augmented</td>
</tr>
<tr>
<td><strong>Target calc. (TC)</strong></td>
<td>Embedded</td>
<td>Embedded</td>
<td>Separate</td>
</tr>
<tr>
<td><strong>TC decision var.</strong></td>
<td></td>
<td></td>
<td>$(x_s, u_s, y_s)$</td>
</tr>
<tr>
<td><strong>OCP decision var.</strong></td>
<td>$(u_i, x, x_s, u_s, y_s)$</td>
<td>$(u_i, \bar{x}_i, x_s, u_s, y_s)$</td>
<td>$(u_i, \bar{x}_i)$</td>
</tr>
</tbody>
</table>
Application example - I

A triple effect evaporator process

Subsystem 1

Subsystem 2

Subsystem 3

F, x_F, T_F

Q

M_1, x_1, T_1

L_1

V_1

M_2, x_2, T_2

L_2

V_2

M_3, x_3, T_3

L_3

V_3
### Identified linear model

<table>
<thead>
<tr>
<th></th>
<th>$L_1$</th>
<th>$Q_1$</th>
<th>$V_1$</th>
<th>$L_2$</th>
<th>$V_2$</th>
<th>$L_3$</th>
<th>$V_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M_1$</td>
<td>-0.04797 (\frac{1}{z-2.717})</td>
<td>-</td>
<td>0.0339 (\frac{1}{z-2.717})</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$T_1$</td>
<td>-</td>
<td>0.564 (\frac{1}{z-2.509})</td>
<td>-</td>
<td>0.1745 (\frac{1}{z-2.509})</td>
<td>0.009394 (\frac{1}{z-2.549})</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$\chi_1$</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$M_2$</td>
<td>0.05726 (\frac{1}{z-2.716})</td>
<td>-</td>
<td>-</td>
<td>0.07207 (\frac{1}{z-2.716})</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$T_2$</td>
<td>0.008029z-0.01856 (\frac{1}{z^2-4.913z+6.029})</td>
<td>0.089z-0.02579 (\frac{1}{z^2-4.913z+6.029})</td>
<td>0.2431z-0.6396 (\frac{1}{z^2-4.913z+6.029})</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$\chi_2$</td>
<td>-0.01418 (\frac{1}{z-2.604})</td>
<td>-</td>
<td>0.01038 (\frac{1}{z-2.604})</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$M_3$</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.07503 (\frac{1}{z-2.712})</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$T_3$</td>
<td>0.001138z+0.03875 (\frac{1}{z^2-4.898z+5.986})</td>
<td>-0.02526z+0.3423 (\frac{1}{z^2-4.898z+5.986})</td>
<td>0.06671z-0.127 (\frac{1}{z^2-4.898z+5.986})</td>
<td>0.09903z-0.2557 (\frac{1}{z-2.712})</td>
<td>2.472z-6.521 (\frac{1}{z^2-4.898z+5.986})</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$\chi_3$</td>
<td>-0.01013z+0.01865 (\frac{1}{z^2-5.241z+6.864})</td>
<td>-0.004064z+0.005355 (\frac{1}{z^2-5.241z+6.864})</td>
<td>0.02242z+0.6029 (\frac{1}{z^2-5.241z+6.864})</td>
<td>-0.22242z+0.6029 (\frac{1}{z^2-5.241z+6.864})</td>
<td>0.01244z-0.02893 (\frac{1}{z^2-5.241z+6.864})</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

#### Neighbors and local (augmented) systems

- **Neighbors sets:** $\mathcal{N}_1 = \emptyset$, $\mathcal{N}_2 = \{1\}$, $\mathcal{N}_3 = \{1, 2\}$

  $\bar{x}_1 = (x_1, x_2, x_3)$, $\bar{x}_2 = (x_2, x_3)$, $\bar{x}_3 = x_3$

- **Augmented systems:**

  $S_{1}^{IN} = \{2, 3\}$, $S_{2}^{IN} = \{1, 3\}$, $S_{3}^{IN} = \{1, 3\}$
Comparison of computation time

- Simulations performed in Matlab, MacBook Pro (3 GHz Intel Core i7, 16 GB RAM)
- Horizon N=100, QP solved (non condensed form) using quadprog (interior-point)
- Tested for 8 hours (480 samples) with two setpoint changes
Comparison of closed-loop outputs
Application example - V

Comparison of closed-loop inputs

Subsystem 1

Subsystem 2

Subsystem 3

Comparison of closed-loop inputs
Part III
Conclusions
Summary

- Presented a comprehensive overview on linear distributed MPC
- Focus on cooperative algorithms, which share compelling properties with centralized architectures
- Discussed available distributed MPC for tracking algorithms
- Proposed novel distributed MPC for tracking approaches that rely "as local as possible" information instead of plant-wide state
Conclusions - II

Take home messages

- Distributed MPC is a solid and reliable alternative to centralized MPC
- Cooperative architectures should be preferred
- Distributed MPC is preferable with respect to centralized MPC for organizational reasons, not computational

Research directions

- Nonlinear distributed MPC
- Reconfigurability and reliability with respect to communication disruptions
- Distributed economic MPC
I would like to thank…

• **Matteo Razzanelli** (PhD student, University of Pisa)
• Prof. John Bagterp Jorgensen (DTU)
• Prof. James B. Rawlings (Univ. of Wisconsin)
• Prof. David Q. Mayne (Imperial College)
References