MECH 6v29.002 – Model Predictive Control

L20 – Decentralized MPC

Outline

- Motivation for decentralized and distributed MPC
- Control Architectures
- Unconstrained MPC Problem Formulation
  - Centralized MPC
  - Decentralized MPC
  - Noncooperative Distributed MPC
  - Cooperative Distributed MPC

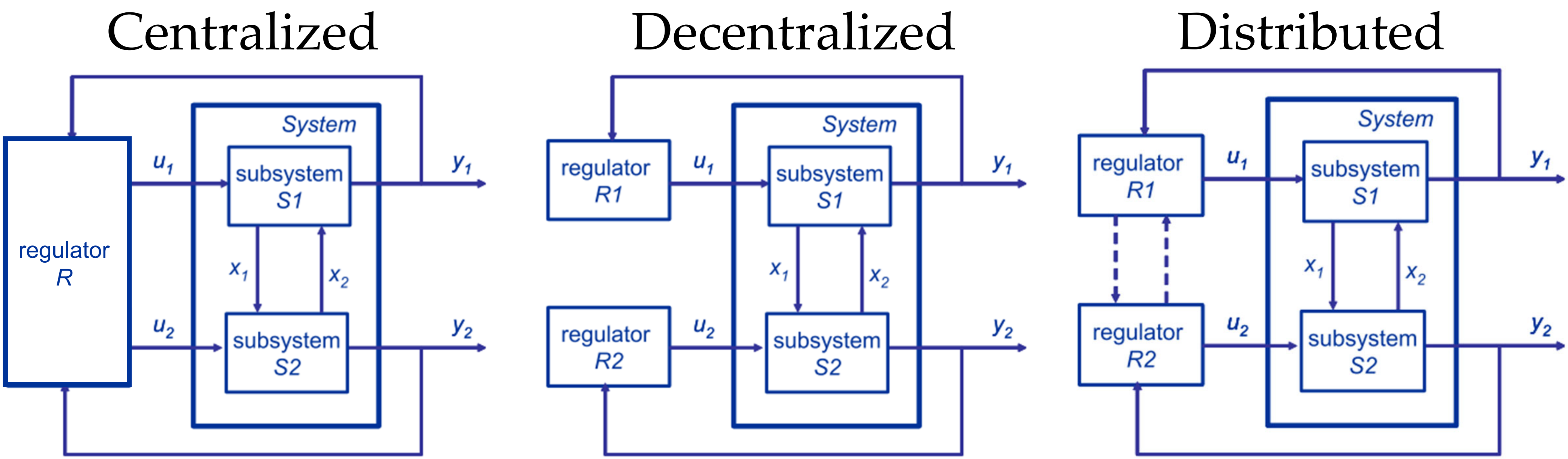
11/06	Decentralized and Distributed MPC	HW #4
11/13	Explicit and Hybrid MPC	
11/20	No Lectures (Fall Break)	
11/27	Project Presentations	
12/04	No Lectures (Last Week of classes)	Project Report

Motivation

- So far we have studied centralized MPC
  - One controller that takes all available information about the system (we assume full state feedback), uses a complete model of the system, and determines optimal values for all inputs
  - Generally thought of as the "ideal" control structure in terms of performance (minimizing cost function)
  - May not be "ideal" for complex systems. Why?
    - Computation time could be too large if there are many states and inputs
    - Dynamics could span multiple timescales resulting in the need for fast update rates and long prediction horizons
    - Information may not be centrally available (controller might only have access to local information)
    - Closed-loop system might need robustness (prevent issues from propagating to other parts of the system through the controller)

Control Architectures

- There are many controller architectures to choose from based on the specific needs of an application [1]

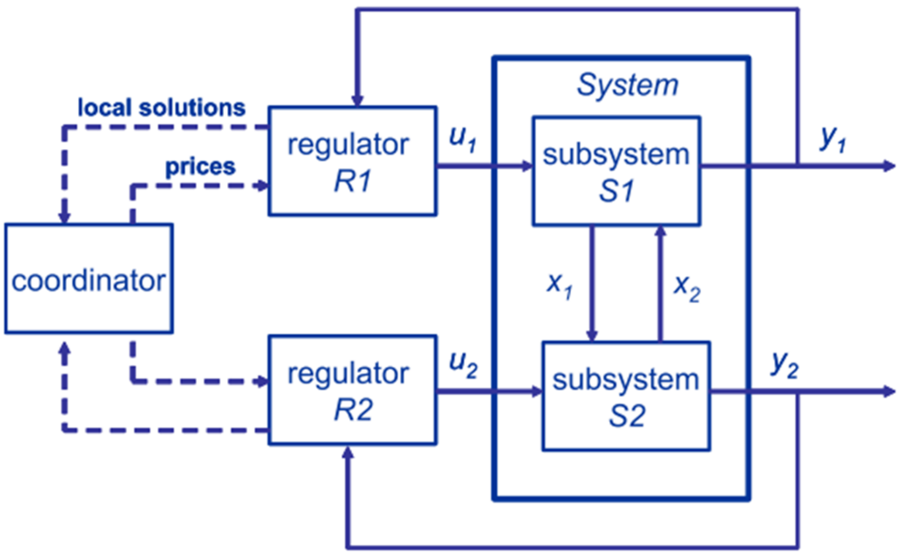


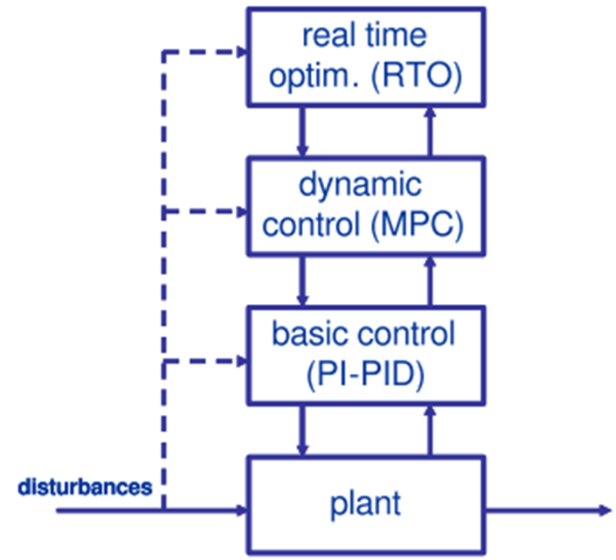
[1] R. Scattolini, "Architectures for distributed and hierarchical Model Predictive Control - A review," Journal of Process Control, 2009.

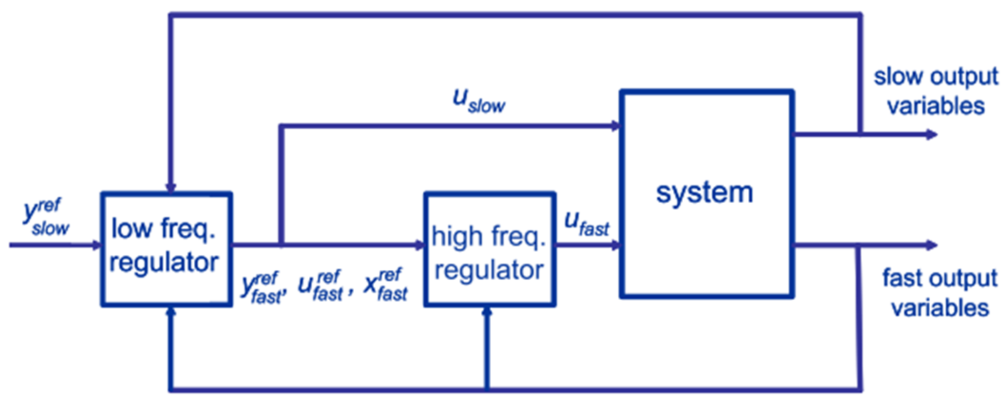
Control Architectures (cont.)

- There are many controller architectures to choose from based on the specific needs of an application [1]

Hierarchical



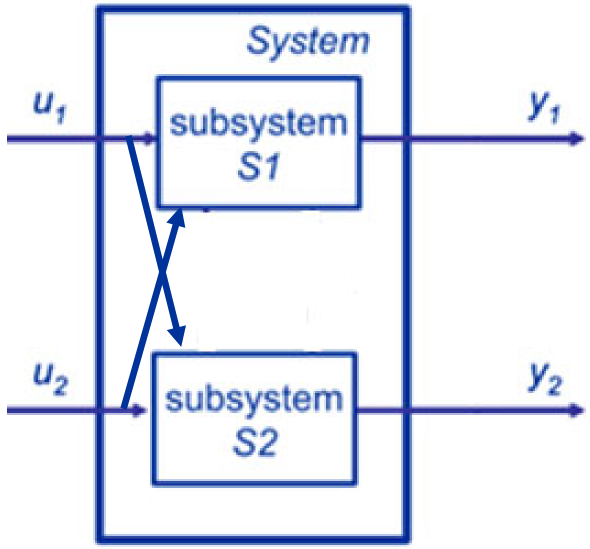




[1] R. Scattolini, "Architectures for distributed and hierarchical Model Predictive Control - A review," Journal of Process Control, 2009.

Unconstrained MPC Problem Formulation

- System comprised of two subsystems
  - Dynamically decoupled
  - Coupled by inputs



- Subsystem 1

$$x_1^+ = A_1 x_1 + B_{11} u_1 + B_{12} u_2$$
$$y_1 = C_1 x_1$$

- Subsystem 2

$$x_2^+ = A_2 x_2 + B_{22} u_2 + B_{21} u_1$$
$$y_2 = C_2 x_2$$

- Complete system

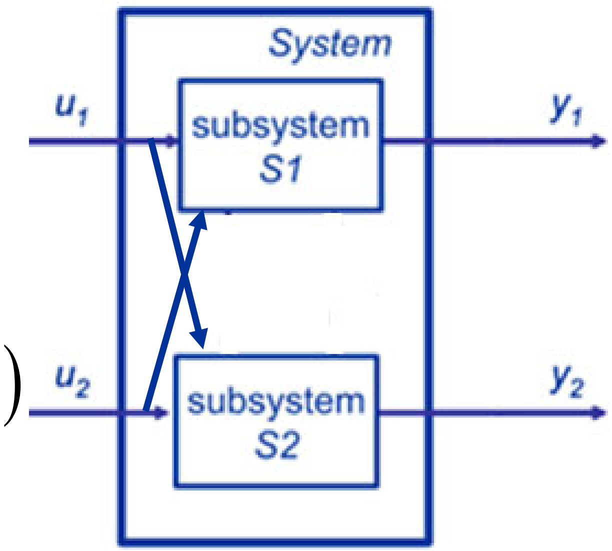
$$x^+ = \begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix} x + \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} u \qquad x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix},\ u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix},\ y = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$$

$$y = \begin{bmatrix} C_1 & 0 \\ 0 & C_2 \end{bmatrix} x \qquad C_1 = I, C_2 = I \quad \text{(assume state feedback)}$$

Unconstrained MPC Problem Formulation

- Subsystem 1 local cost function

$$V_1\left(x_1(0),U_1,U_2\right)=\sum_{k=0}^{N-1}\ell_1\left(x_1(k),u_1(k)\right)+V_{1f}\left(x_1(N)\right)$$



- Subsystem 2 local cost function

$$V_2\left(x_2(0),U_1,U_2\right)=\sum_{k=0}^{N-1}\ell_2\left(x_2(k),u_2(k)\right)+V_{2f}\left(x_2(N)\right)$$

$$U_i=\left[u_i^T(0)\quad u_i^T(1)\qquad u_i^T(N-1)\right]^T \qquad i\in\{1,2\}$$

$$\ell_i\left(x_i(k),u_i(k)\right)=\frac{1}{2}x_i^T(k)Q_i x_i(k)+\frac{1}{2}u_i^T(k)R_i u_i(k)$$

$$V_{i,f}\left(x_i(N)\right)=\frac{1}{2}x_i^T(N)P_{i,f}x_i(N) \qquad A_i^T P_{i,f}A_i-P_{i,f}=-Q_{i,f}$$

$$\left(\text{Assume } A_i \text{ is stable}\right)$$

- Complete system cost function

$$V\left(x_1(0),x_2(0),U_1,U_2\right)=\rho_1 V_1\left(x_1(0),U_1,U_2\right)+\rho_2 V_2\left(x_2(0),U_1,U_2\right)$$

$$0<\rho_1,\rho_2 \quad \rho_1+\rho_2=1$$

Centralized MPC

$$\min_{U_1,U_2} V\left(x_1(0),x_2(0),U_1,U_2\right)$$

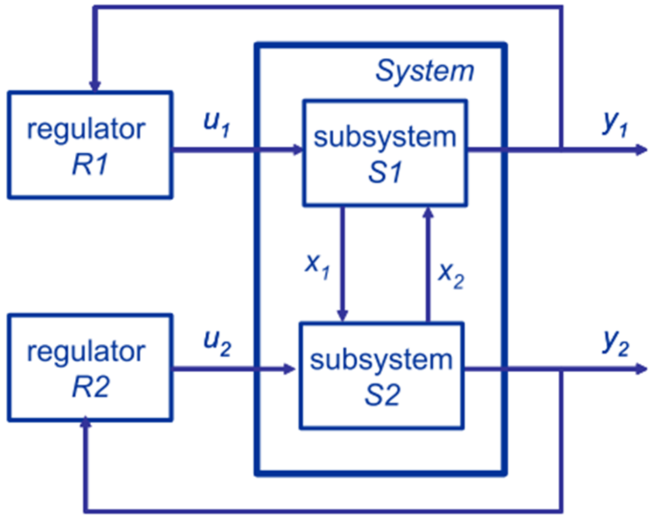
$$s.t.$$

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} \begin{bmatrix} u_1(k) \\ u_2(k) \end{bmatrix}, \; k \in \{0,1,\ldots,N-1\}$$

- Centralized MPC is considered “optimal” and will be used as a benchmark for decentralized and distributed MPC
- All other control architectures can, at best, achieve the same performance but will generally be “suboptimal”
- Centralized MPC is more complex because it has both input trajectories as decision variables
- Decentralized and distributed MPC should have lower complexity which should result in lower computational cost.
- Stability of this centralized MPC formulation is guaranteed through the choice of terminal cost

Decentralized MPC

- Each controller optimizes its own local objective with no information about the actions of the other subsystem
- Controller i optimization problem



$$\min_{U_i} V_i\left(x_i(0), U_i\right) = \sum_{k=0}^{N-1} \ell_i\left(x_i(k), u_i(k)\right) + V_{i,f}\left(x_i(N)\right)$$

$$s.t.$$

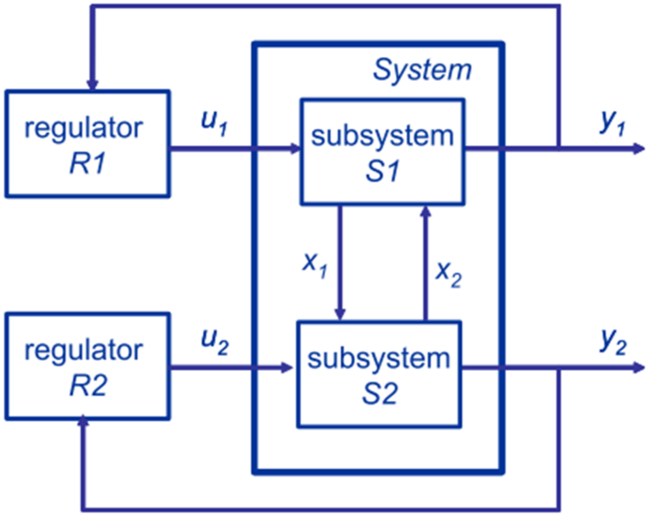
$$x_i(k+1) = A_i x_i(k) + B_{ii} u_i(k),\ k \in \{0,1,\ldots,N-1\}$$

Does not account for the affect of the other input $+B_{ij}u_j$

- Without constraints, we know that the optimal solution is

$$u_i(0) = K_i x_i(0)$$

Decentralized MPC (cont.)



- This results in the closed-loop dynamics

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} A_1 + B_{11}K_1 & B_{12}K_2 \\ B_{21}K_1 & A_2 + B_{22}K_2 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix}$$

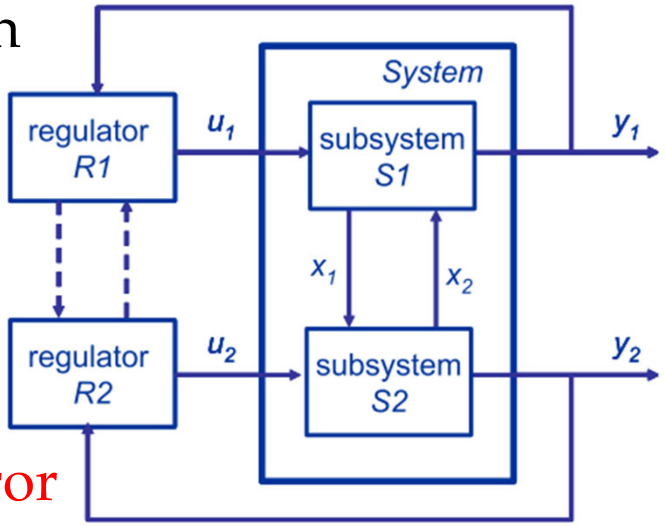
- We know that the diagonal terms are stable

$$A_1 + B_{11}K_1, \quad A_2 + B_{22}K_2$$

- Strong coupling (interactions) between the two subsystems can cause the closed-loop system to be unstable based on $B_{12}K_2, \; B_{21}K_1$
- Decentralized MPC summary
  - Reduced computational cost
  - Only uses local information (zero communication overhead)
  - Despite closed-loop stable subsystems, the overall closed-loop system could be unstable
  - Even if stable, the control performance could be poor

Noncooperative Distributed MPC

- Controller i minimizes its own local cost function assuming the input trajectory from the other controller(s) is known
  - Both controllers are greedy
- Controllers communicate planned control trajectories
- Unlike decentralized MPC, there is no model error
  - Poor performance is cause by greedy behavior that results in a Nash equilibrium (not model error)
- Controller i optimization problem



$$\min_{U_i} V_i\left(x_i(0), U_i, U_j\right) = \sum_{k=0}^{N-1} \ell_i\left(x_i(k), u_i(k)\right) + V_{i,f}\left(x_i(N)\right)$$

$$s.t.$$

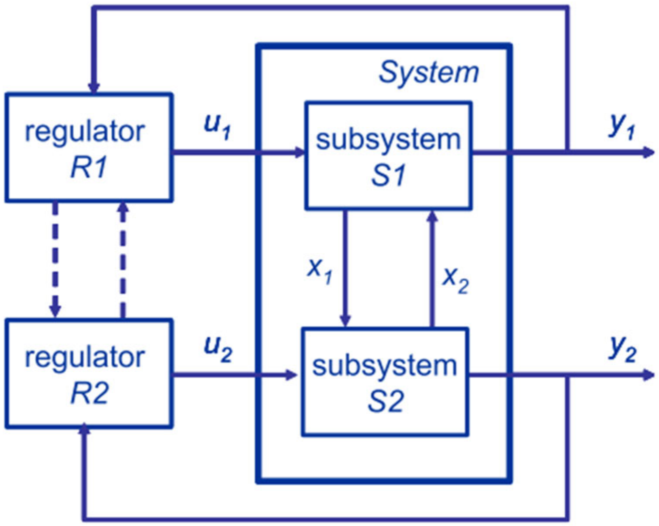
$$x_i(k+1) = A_i x_i(k) + B_{ii} u_i(k) + B_{i,j} u_j(k), \; k \in \{0,1,\dots,N-1\}$$

Noncooperative Distributed MPC (cont.)

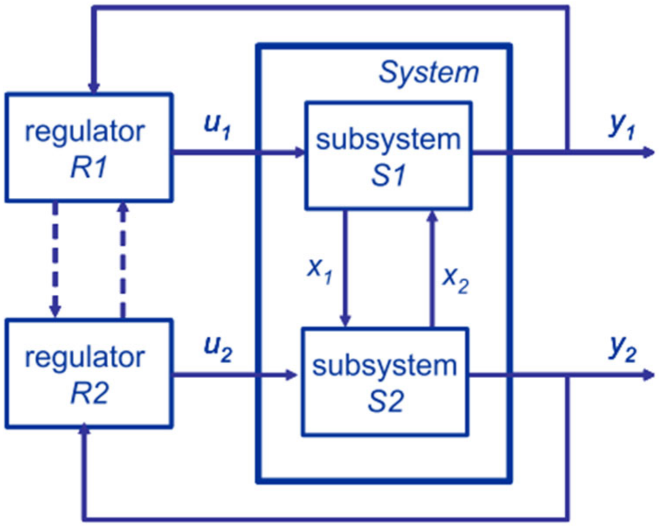
- Can use batch approach to solve for optimal control trajectory

$$U_i(0) = K_i x_i(0) + L_{ij} U_j$$

  - Optimal control sequence still depends linearly on measured state but also on the other controller's input trajectory
- Need to
  - Initialize both input trajectories
  - Communicate initial trajectories
  - Solve for optimal trajectory
  - Communicate
  - Repeat until convergence
- Need to analyze/guarantee convergence



Noncooperative Distributed MPC (cont.)

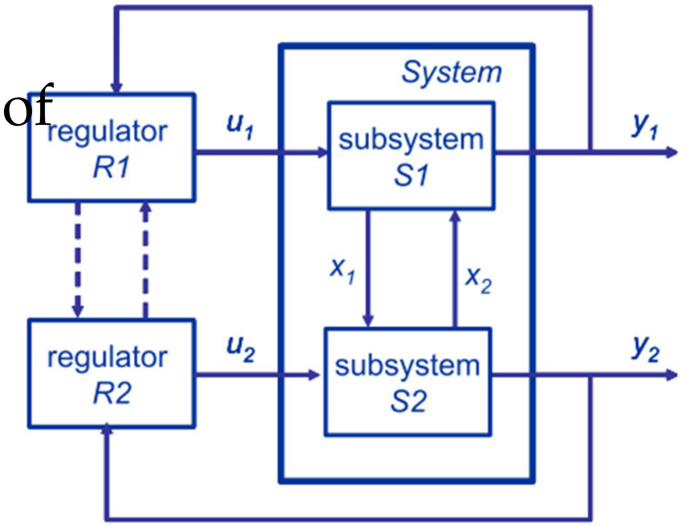


- Control update and convergence analysis
  - At every time-step, the two controllers will communicate and iteratively update their control sequence based on the other controllers input trajectory
  - Let p be the index for these iterations
  - Let U_i^p be the input trajectory at iteration p
  - We still have $U_i^0 = K_i x_i(0) + L_{ij} U_j^p$
  - But to guarantee convergence, we are not going to use this solution directly (full step)
  - Instead, we will take a convex combination of the current optimal solution U_i^0 and U_i^p

$$U_i^{p+1} = w_i U_i^0 + (1 - w_i) U_i^p, \;\; 0 < w_i < 1, \;\; w_1 + w_2 = 1$$

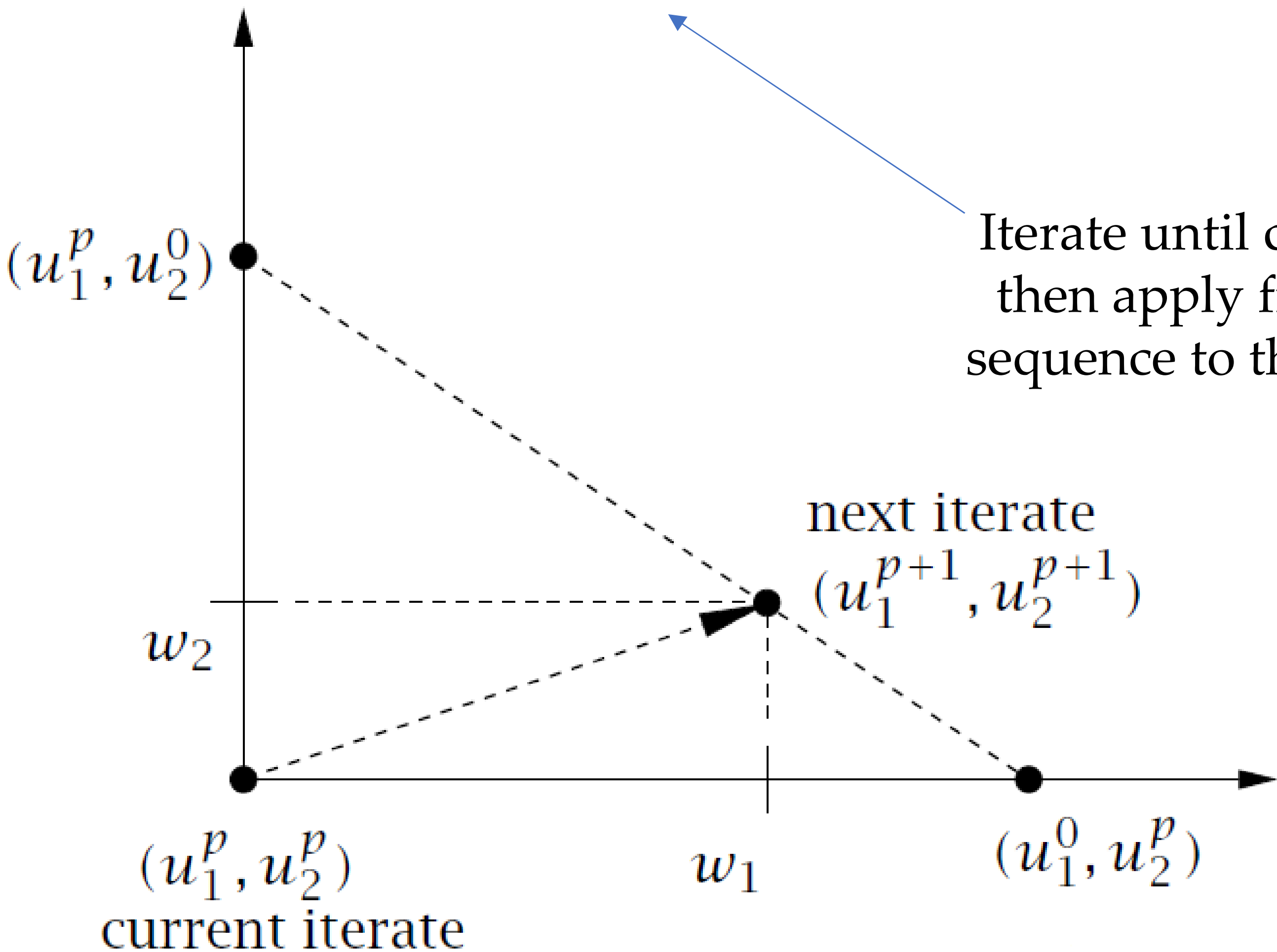
Noncooperative Distributed MPC (cont.)

- Control update and convergence analysis
  - Instead, we will take a convex combination of the current optimal solution U_i^0 and U_i^p



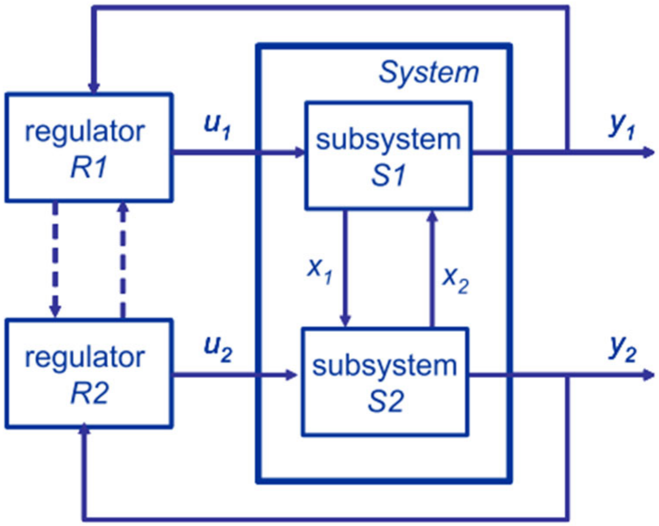
$$U_i^{p+1} = w_i U_i^0 + (1 - w_i) U_i^p, \quad 0 < w_i < 1, \quad w_1 + w_2 = 1$$

Iterate until convergence, then apply first input in sequence to the subsystem



Noncooperative Distributed MPC (cont.)

- Control update and convergence analysis
  - We can analyze if these iterations will converge
  - Assume both subsystem are stable
  - Assume cost matrices (Q's and R's) are positive definite



$$U_i^{p+1} = w_i U_i^0 + (1 - w_i) U_i^p$$

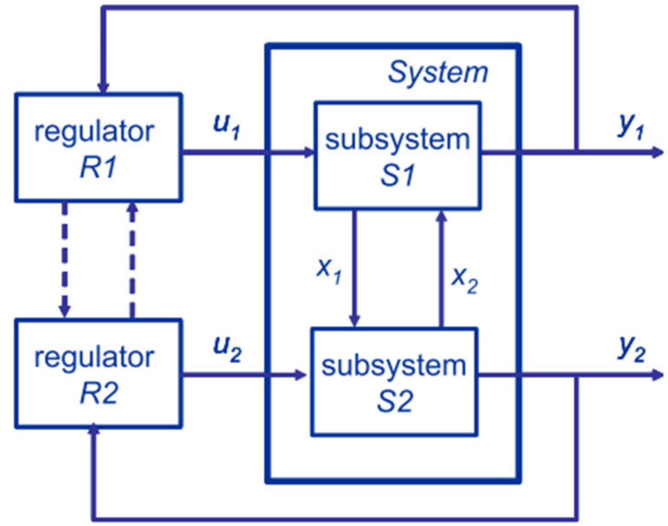
$$\begin{bmatrix} U_1 \\ U_2 \end{bmatrix}^{p+1} = \begin{bmatrix} w_1 I & 0 \\ 0 & w_2 I \end{bmatrix} \begin{bmatrix} U_1^0 \\ U_2^0 \end{bmatrix} + \begin{bmatrix} (1-w_1)I & 0 \\ 0 & (1-w_2)I \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \end{bmatrix}^p$$

$$U_i^0 = K_i x_i(0) + L_{ij} U_j^p$$

$$\begin{bmatrix} U_1 \\ U_2 \end{bmatrix}^{p+1} = \underbrace{\begin{bmatrix} w_1 K_1 & 0 \\ 0 & w_2 K_2 \end{bmatrix}}_{K} \begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix} + \underbrace{\begin{bmatrix} (1-w_1)I & w_1 L_{12} \\ w_2 L_{21} & (1-w_2)I \end{bmatrix}}_{L} \begin{bmatrix} U_1 \\ U_2 \end{bmatrix}^p$$

Noncooperative Distributed MPC (cont.)

- Control update and convergence analysis
  - Convergence is governed by the eigenvalues of L



$$\begin{bmatrix} U_1 \\ U_2 \end{bmatrix}^{p+1} = \underbrace{\begin{bmatrix} w_1 K_1 & 0 \\ 0 & w_2 K_2 \end{bmatrix}}_{K} \begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix} + \underbrace{\begin{bmatrix} (1-w_1)I & w_1 L_{12} \\ w_2 L_{21} & (1-w_2)I \end{bmatrix}}_{L} \begin{bmatrix} U_1 \\ U_2 \end{bmatrix}^{p}$$

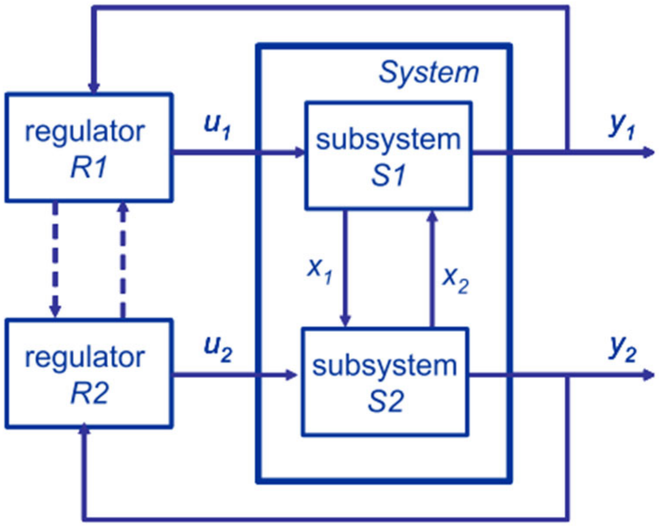
$$\begin{bmatrix} U_1 \\ U_2 \end{bmatrix}^{p+1} = L \begin{bmatrix} U_1 \\ U_2 \end{bmatrix}^{p} + K \begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix}$$

  - Assuming L is stable, the converged (steady-state) solution is

$$\begin{bmatrix} U_1 \\ U_2 \end{bmatrix}^{\infty} = (I-L)^{-1} K \begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix}$$

Noncooperative Distributed MPC (cont.)

- Control update and convergence analysis



$$\begin{bmatrix} U_1 \\ U_2 \end{bmatrix}^{\infty} = (I-L)^{-1} K \begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix}$$

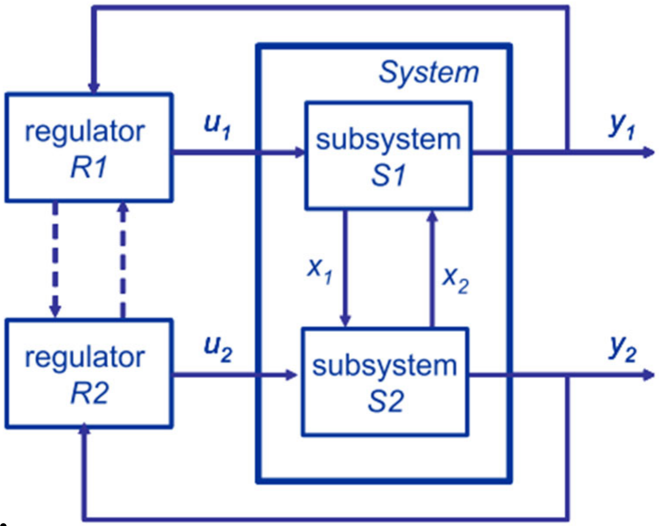
  - Note that

$$(I-L)^{-1} K = \begin{bmatrix} w_1 I & -w_1 L_{12} \\ -w_2 L_{21} & w_2 I \end{bmatrix}^{-1} \begin{bmatrix} w_1 K_1 & 0 \\ 0 & w_2 K_2 \end{bmatrix}$$

$$= \begin{bmatrix} I & -L_{12} \\ -L_{21} & I \end{bmatrix}^{-1} \begin{bmatrix} K_1 & 0 \\ 0 & K_2 \end{bmatrix}$$

  - The weights w_1 and w_2 do not affect the converged input trajectories (which is a good thing)
  - The converged input sequences $U_1^{\infty}, U_2^{\infty}$ correspond to a Nash equilibrium (which is likely different than the centralized MPC solution)
  - Nash equilibrium – neither controller can improve its performance given the other controller's trajectory

Noncooperative Distributed MPC (cont.)



- Main drawback of noncooperative distributed MPC
  - Nash equilibrium may not be stable
  - Nash equilibrium may be stable, but the closed-loop system is unstable
  - Nash equilibrium may be stable and the closed-loop system is stable
- Which case arises based on the unique combination of system dynamics and controller design
  - One has to perform this analysis for any time a system or controller parameter changes
- Next class, we will look at an example of each of these three cases and then show how cooperative MPC can overcome this limitation at the expense of slightly more information communication.