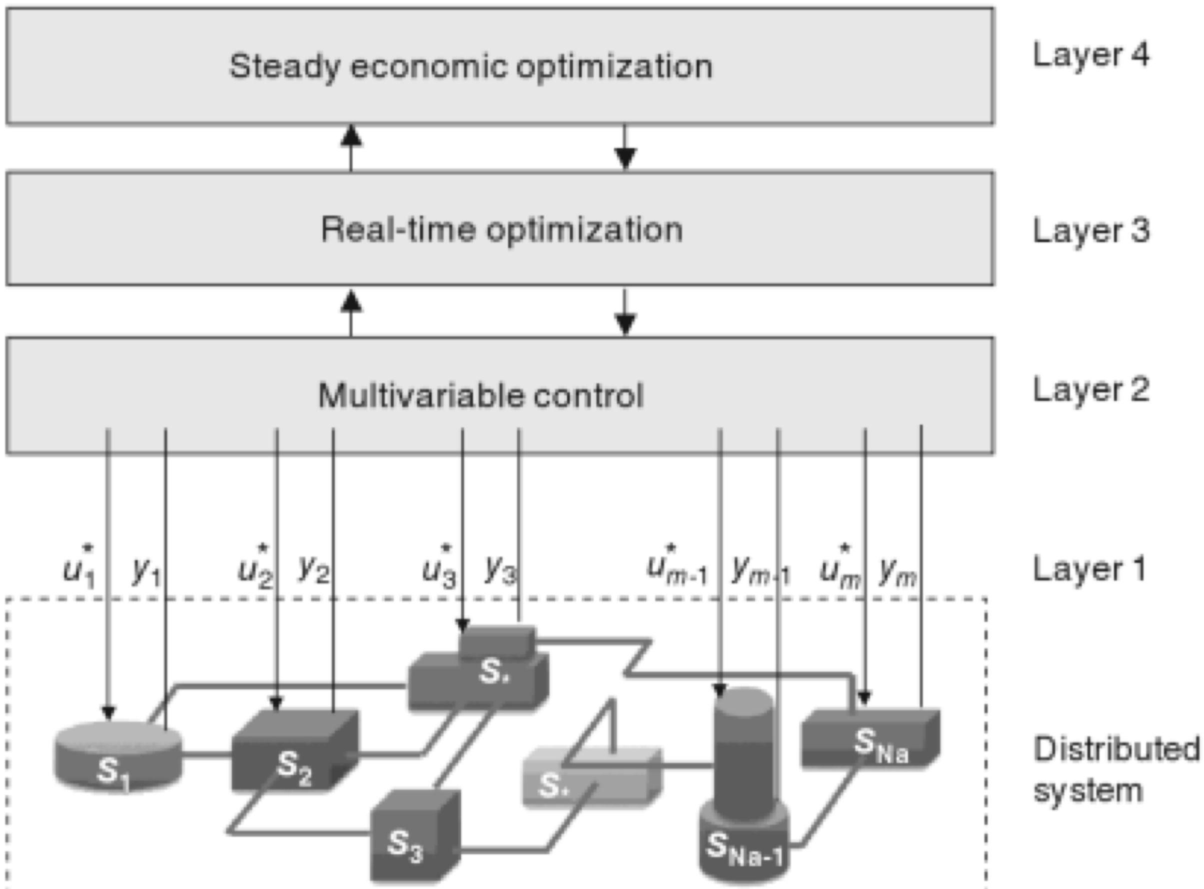
Distributed Real-Time Control Systems

Module 16

Distributed Control

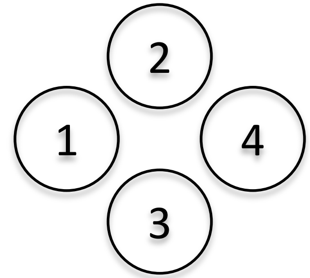
Distributed System Topologies

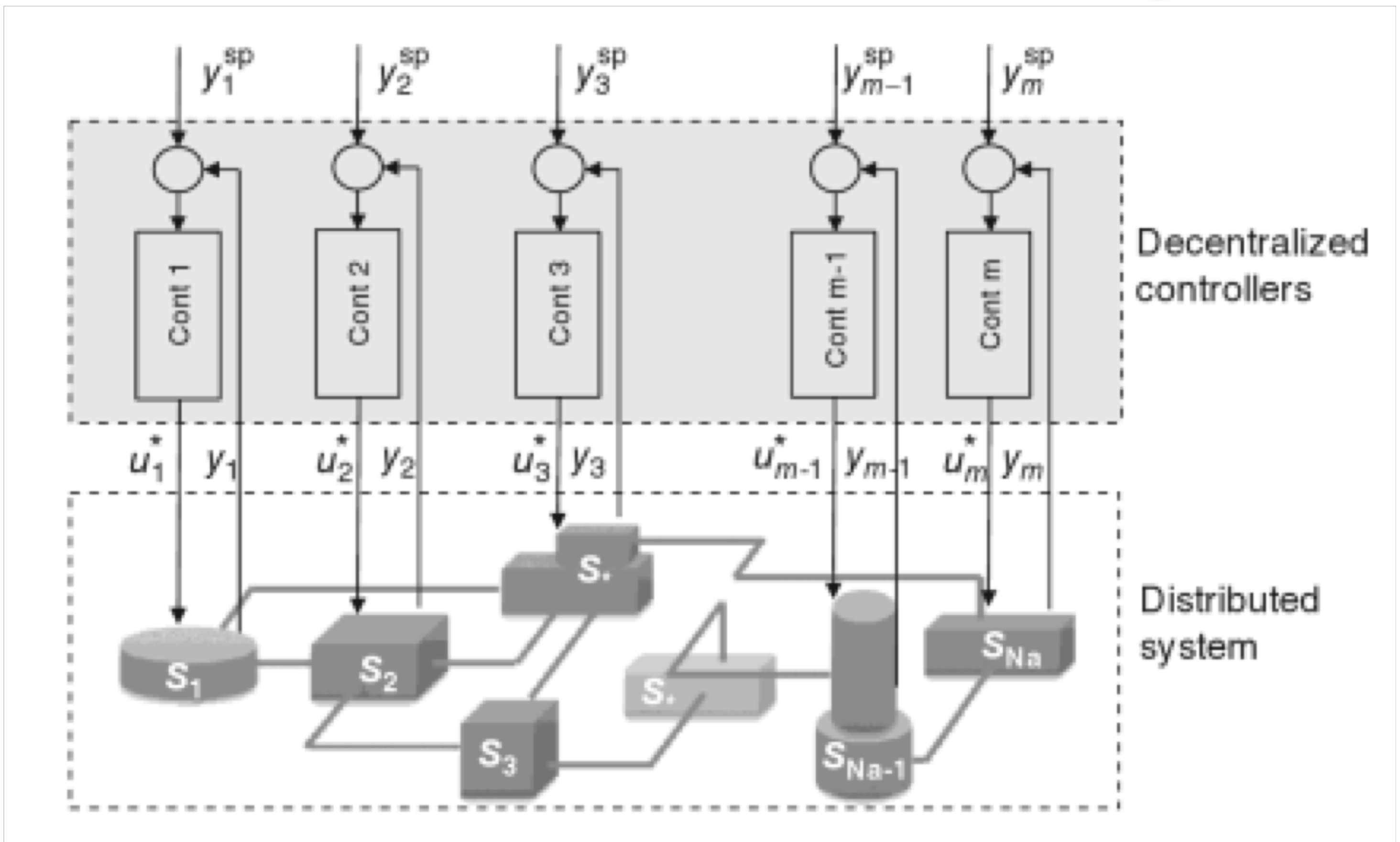
Centralized Control



Distributed Systems Topologies

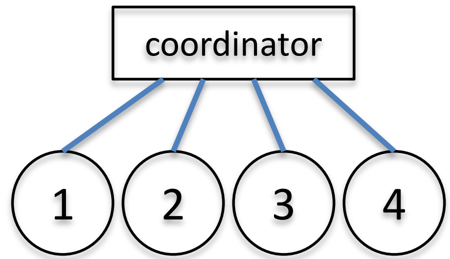
Decentralized
Non-cooperative/Competitive

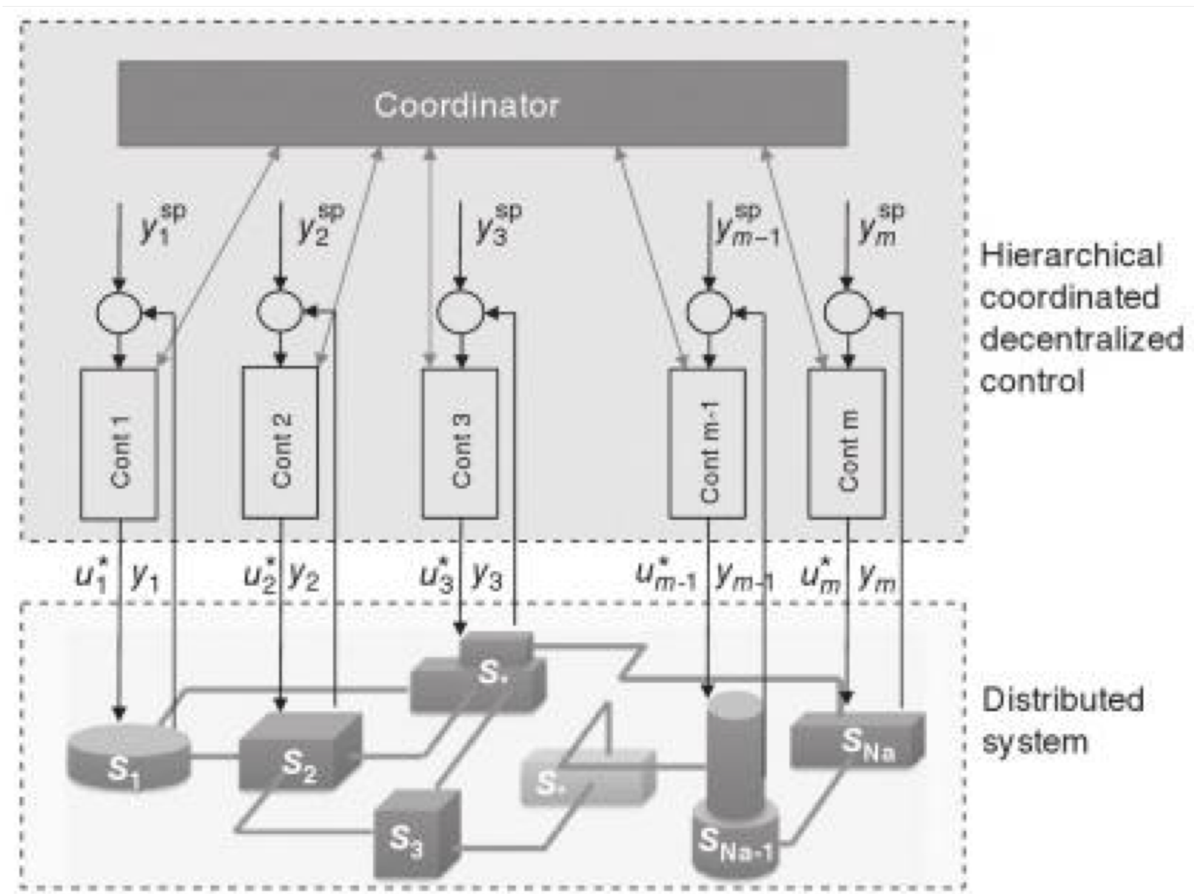




Distributed Systems Topologies

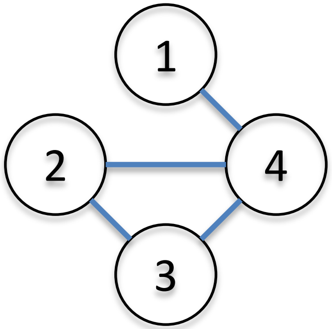
Hierarchical Coordinated

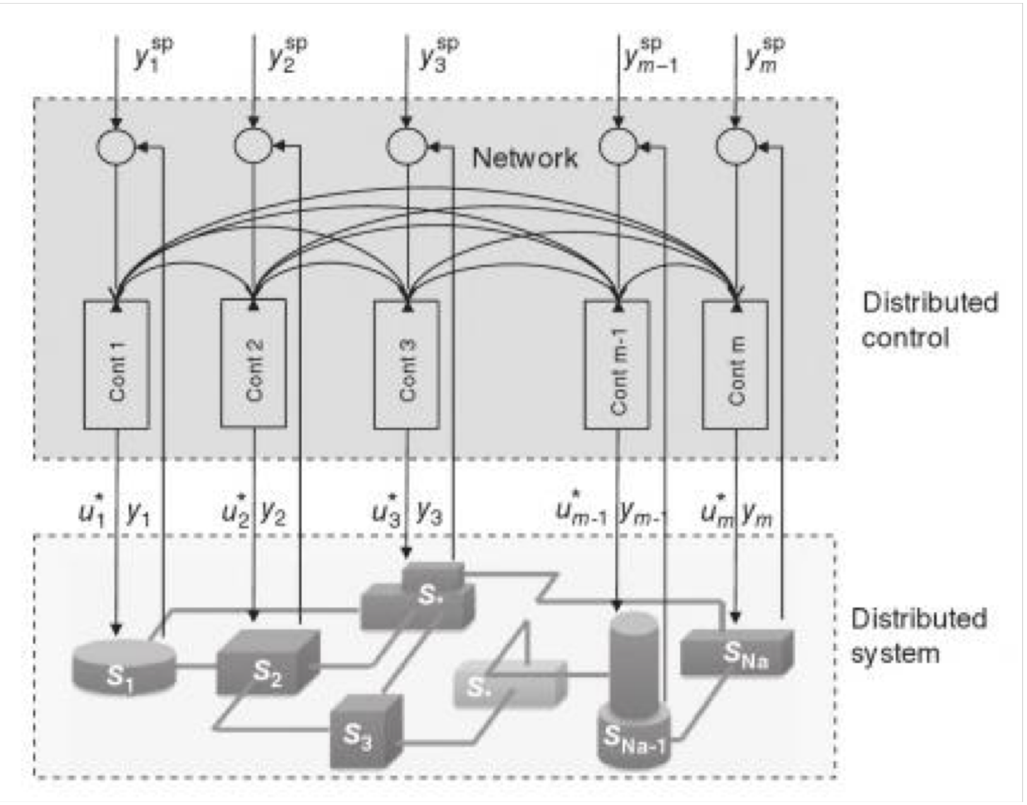




Distributed Systems Topologies

Networked
Distributed
Cooperative

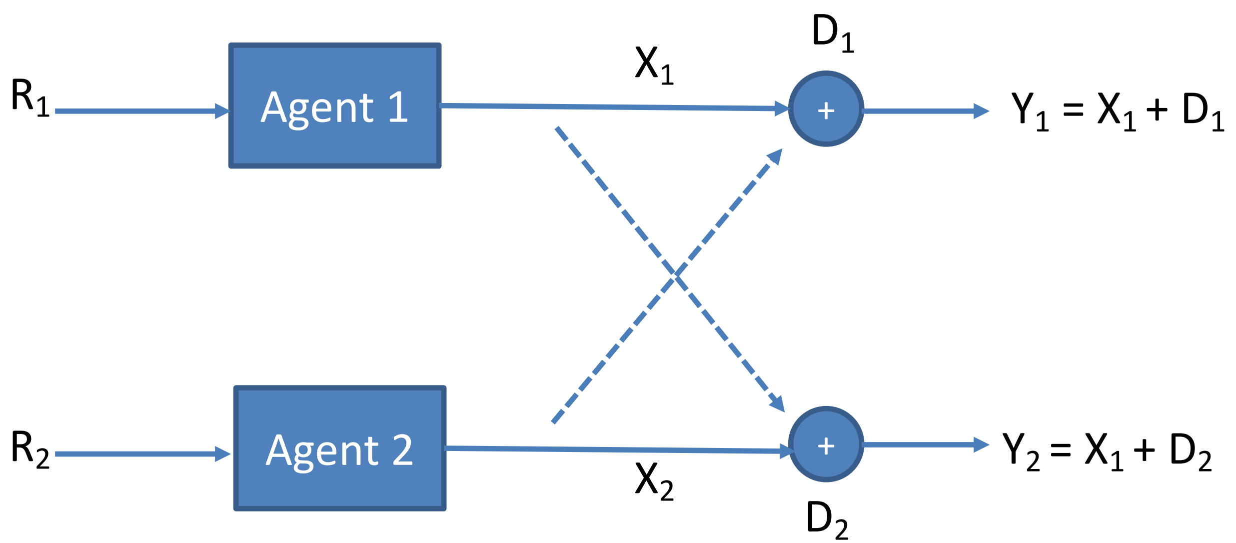




Distributed Non-Cooperative Control

Distributed Non-Cooperative Control

- In distributed non-cooperative control, each agent is just concerned in achieving its own objective.
- If the actions of other agents interfere, they are dealt as if they were an unknown disturbance.



Distributed Non-Cooperative Control

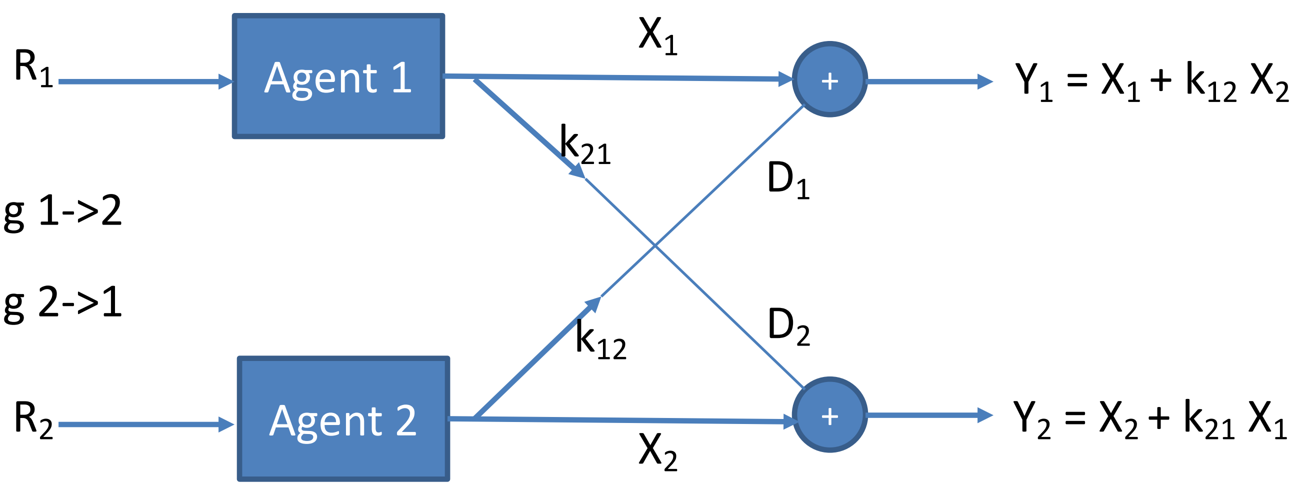
- Let us assume a simple linear coupling:

$$D_1 = X_1 + k_{12}X_2$$
$$D_2 = X_2 + k_{21}X_1$$

k_{12}, k_{21} – coupling gains

k_{21} - Coupling 1->2

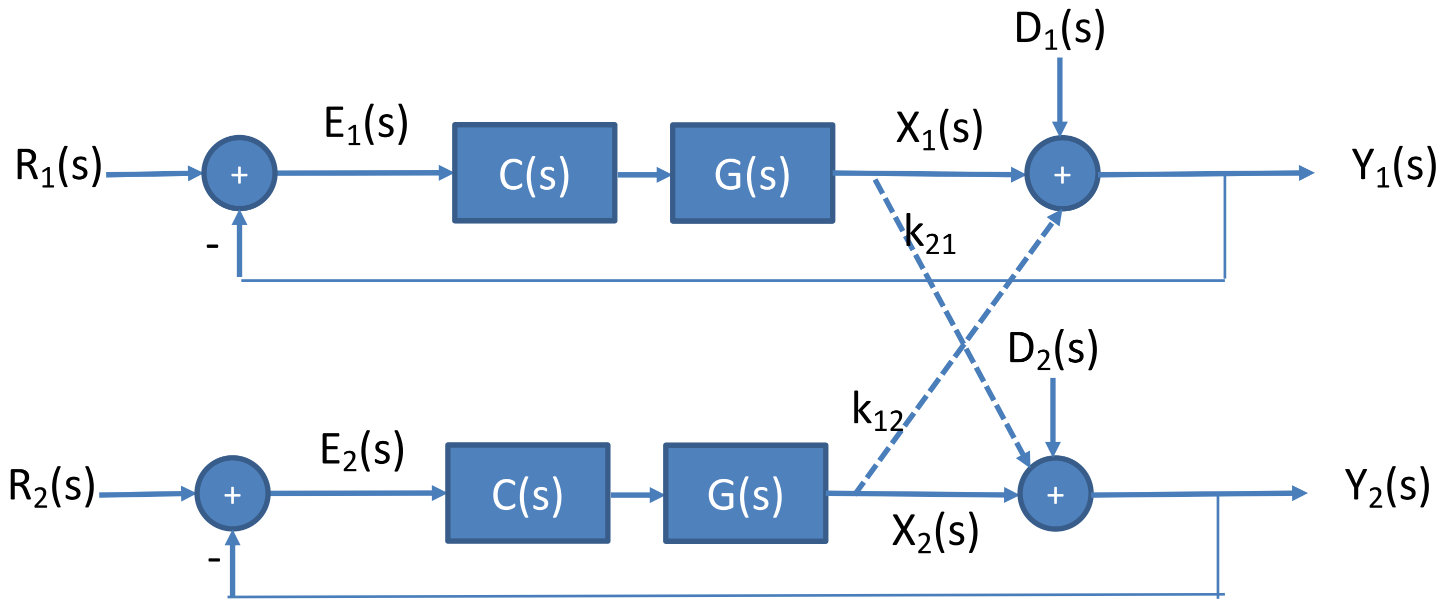
k_{12} - Coupling 2->1



How do the two agents interfere?

Distributed Non-Cooperative Control

- Assume identical systems
- Let us analyse the coupled system dynamics:



$$X_1 = CG\Big(R_1 - (X_1 + k_{12}X_2)\Big)$$
$$X_2 = CG\Big(R_2 - (X_2 + k_{21}X_1)\Big)$$
$$\Longrightarrow$$
$$X_1 = \frac{CG}{1+CG}(R_1 - k_{12}X_2)$$
$$X_2 = \frac{CG}{1+CG}(R_2 - k_{21}X_1)$$

Distributed Non-Cooperative Control

$$X_1 = \frac{CG}{1+CG}(R_1 - k_{12}X_2)$$
$$X_2 = \frac{CG}{1+CG}(R_2 - k_{21}X_1)$$

Replacing the right hand side of the second equation in X_2 of the first equation:

$$X_1 = \frac{CG}{1+CG}R_1 - \left(\frac{CG}{1+CG}\right)^2 k_{12}R_2 + \left(\frac{CG}{1+CG}\right)^2 k_{12}k_{21}X_1$$

$$X_1\left(1 - \left(\frac{CG}{1+CG}\right)^2 k_{12}k_{21}\right) = \frac{CG}{1+CG}R_1 - \left(\frac{CG}{1+CG}\right)^2 k_{12}R_2$$

$$X_1 = \frac{CG(1+CG)}{(1+CG)^2 - (CG)^2 k_{12}k_{21}}R_1 - \frac{(CG)^2 k_{12}}{(1+CG)^2 - (CG)^2 k_{12}k_{21}}R_2$$

Distributed Non-Cooperative Control

$$X_1 = \frac{CG(1+CG)}{(1+CG)^2 - (CG)^2 k_{12}k_{21}} R_1 - \frac{(CG)^2 k_{12}}{(1+CG)^2 - (CG)^2 k_{12}k_{21}} R_2$$

(Analogous for X2)

The poles of the coupled system are the roots of the characteristic polynomial:

$$(1+CG)^2 - (CG)^2 k_{12}k_{21} = 0$$

Unless the coupling gains are null, the poles of the coupled system can be quite different from the poles of the individual systems.

Distributed Non-Cooperative Control

HOMEWORK 1:

Compute the poles of the coupled system corresponding to your project and conclude about its dynamical properties. Is it stable? will it oscillate ?

Assumptions:

1 – C is a PI controller

2 – G is first order system

$$C(s) = K_p\left(1 + \frac{K_i}{s}\right) = \frac{K_p(s + K_i)}{s}$$

$$G(s) = \frac{K_0}{1 + s\tau}$$

HOMEWORK 2:

Compute the transfer functions Y_1/R_1 and Y_1/R_2 and determine if, in steady state, the Y_1 can track the reference R_1 without error and reject the influence of the reference R_2.

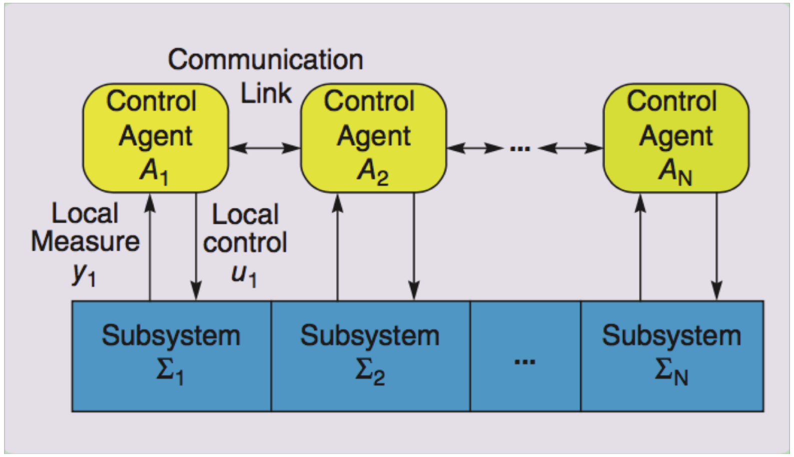
Same **assumptions** as Homework 1

Distributed Cooperative Control

Distributed Cooperative Control

In distributed cooperative control, **multiple local control agents** act in a **coordinated way** to perform a given task.

Coordination is important when the **subsystems** controlled by each agent **interact** due to **physical coupling**.

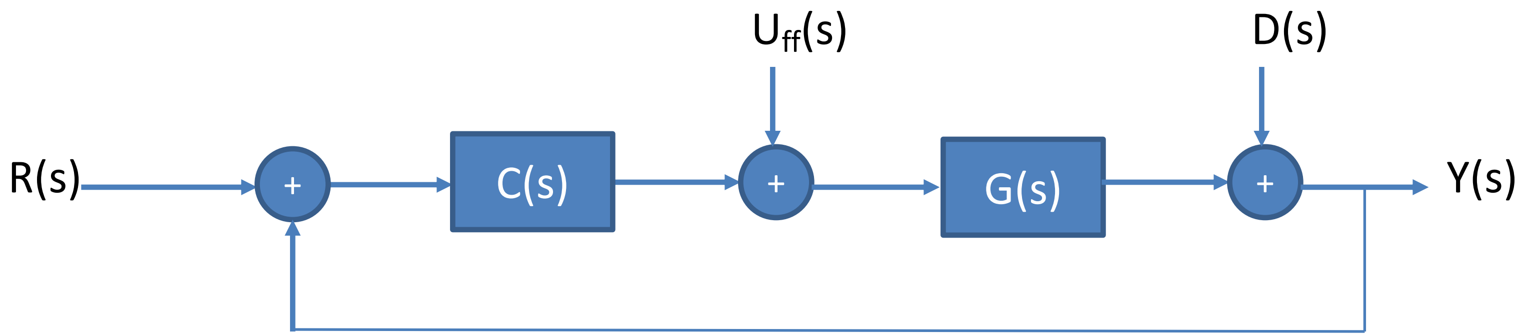


If each control agent tries to optimize a local cost in isolation, local decisions may conflict with each other, and may even drive the system to instability.

Control agents communicate with their neighbors and try to reach a consensus on the values of the manipulated variables at the beginning of each sampling interval

Feedforward from Accessible Disturbances

- Let us consider the following system with feedforward and output disturbance.

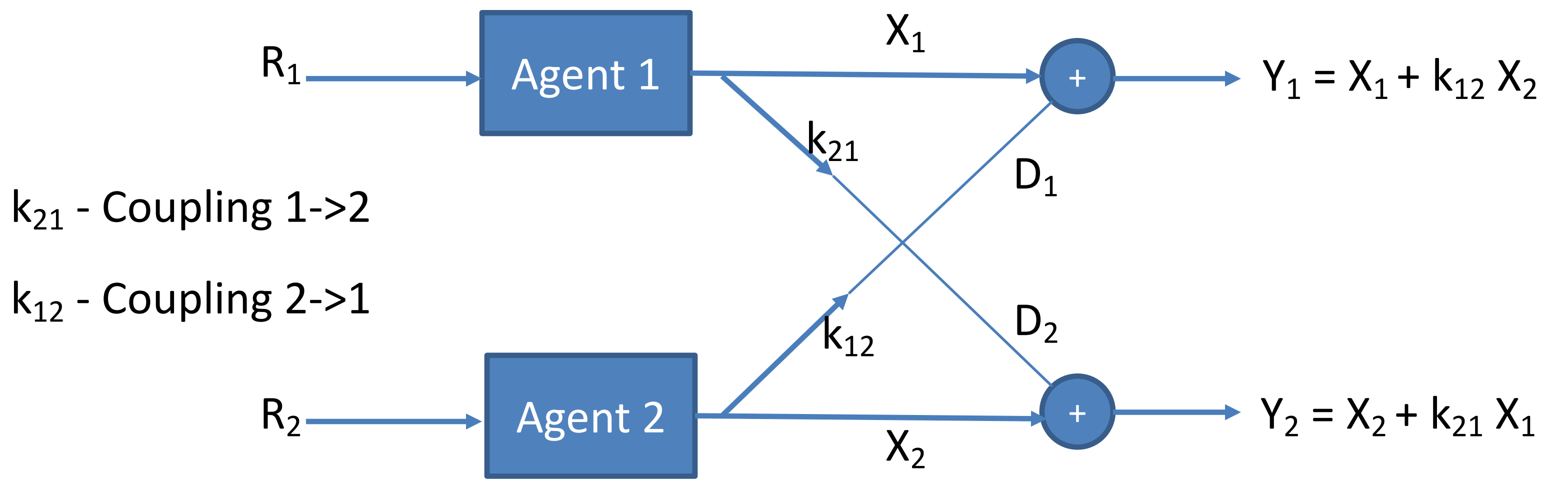


$$Y = \frac{CG}{1+CG}R + \frac{G}{1+CG}U_{ff} + \frac{1}{1+CG}D$$

(Laplace transform domain)

Feedforward from Accessible Disturbances

- If the actuating device **can communicate** its actuation intents to the other devices, and a **partial model** of the coupling is available, the other devices can **"prepare"** for the change by considering it a **known (accessible) disturbance**.



Feedforward from Accessible Disturbances

- When a disturbance is accessible (can be **anticipated**) it can be compensated by a suitable feedforward term.

$$Y = \frac{CG}{1+CG}R + \frac{GU_{ff} + D}{1+CG}$$

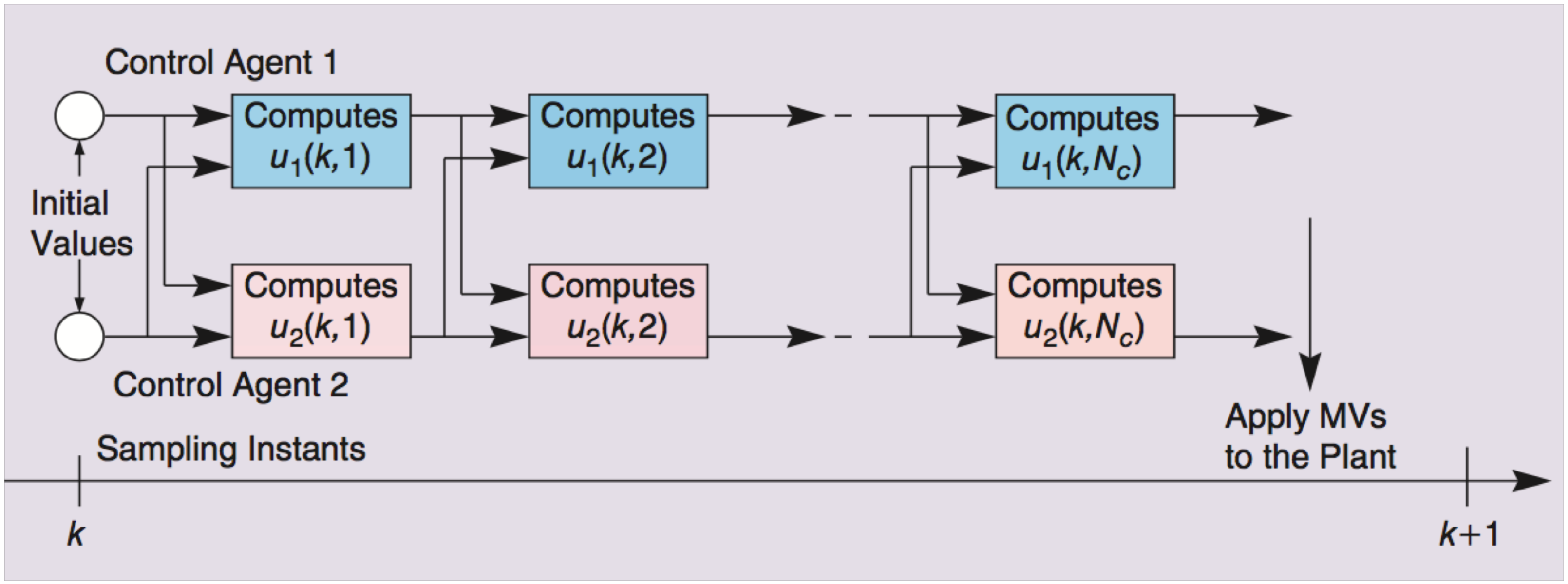
- Make the second term null by:

$$U_{ff} = -G^{-1}D$$

(Note: Again the compensating feedforward term requires a good model of G)

Feedforward from Accessible Disturbances

- A coordination procedure for 2 agents:



Agent 1 (current actuations u_1, u_2):

Compute d_1 due to u_2 :	$d_1 = f_1(u_2)$
Compute $y_{1'}$ due to d_1 :	$y_{1'} = y_1 + d_1$
Compute u_{1ff} to compensate d_1 :	$u_{1ff} = G^{-1} d_1$
Compute $u_{1'}$:	$u_{1'} = u_{1ff} + u_1$
Send $u_{1'}$ to agent 2	
If $u_{1'} == u_1$, we have converged.	

Feedforward from Accessible Disturbances

Synchronous coordination:

Agent 1 takes the lead (master)

	Agent 1		Agent 2
	Send u1	→	
			Receive u1
		←	Send u2
	Receive u2		Compute u2’
	Compute u1’		
(new cycle)	Send u1’	→	
			Receive u1’
		←	Send u2’
	Receive u2’		(…)
	(…)		

time

Feedforward from Accessible Disturbances

Asynchronous coordination: No explicit master. Data is sent by any agent whenever ready

Agent 1		Agent 2	
Send u1	→		
		Receive u1 Compute u2’ Send u2’	
	←		time
Receive u2 Compute u1’ Send u1’	→		
		Receive u1’ (…)	

Feedforward from Accessible Disturbances

- **<u>HOMEWORK 3</u>**
  - Sketch the pseudo-code for the coordination in the case of your project.
  - When to stop the iterations ?
  - Test with your system:
    - Does the system always converge?