The Netherlands
Research School for
Transport, Infrastructure
and Logistics
TRAIL

Multi-Agent Control of Transportation Networks

Rudy Negenborn

DCSC Lunch Colloquium – December 7, 2005

Delft Center for Systems and Control

TU Delft
Delft University of Technology

Overview

1. Transportation networks
2. Single versus multi-agent control
3. Model predictive control
   - Single versus multi-agent
   - Subnetwork models
   - Obtaining agreement
   - Communication and decision making
4. Example: load-frequency control
5. Future plans

1. Transportation networks

What?

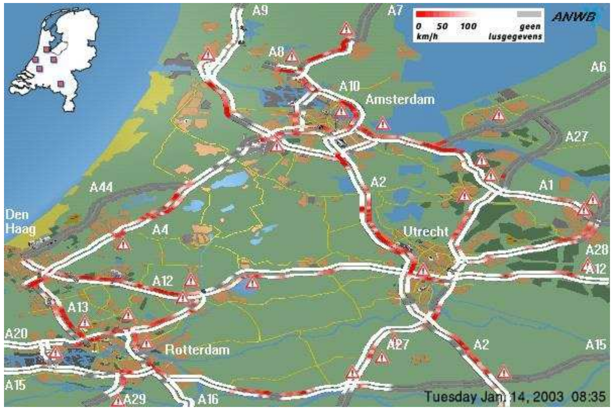
road traffic networks, power distribution networks,

water networks, railway networks, gas networks, etc.

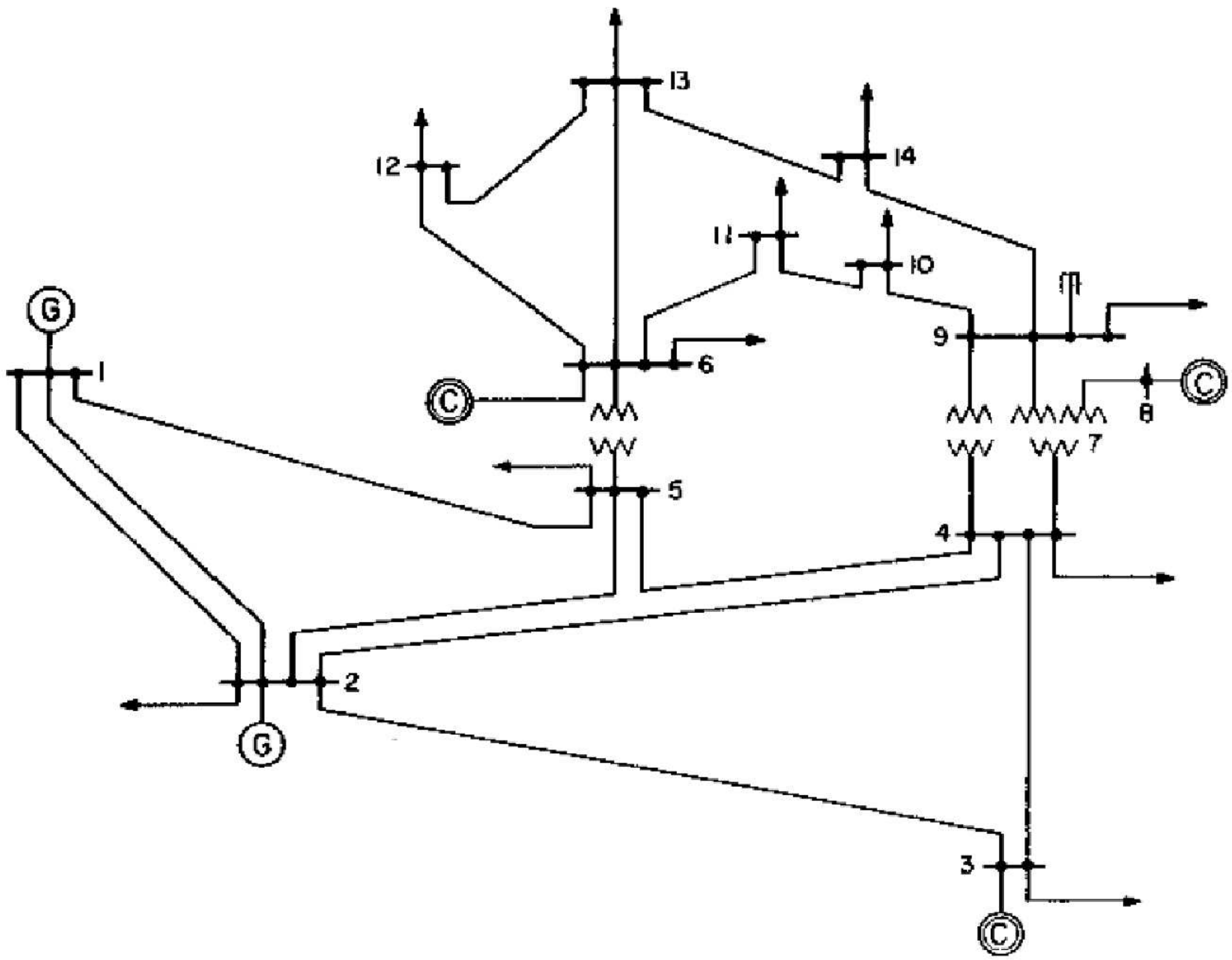
Why?

Efficient operation crucial for modern economy, environment, safety, ...

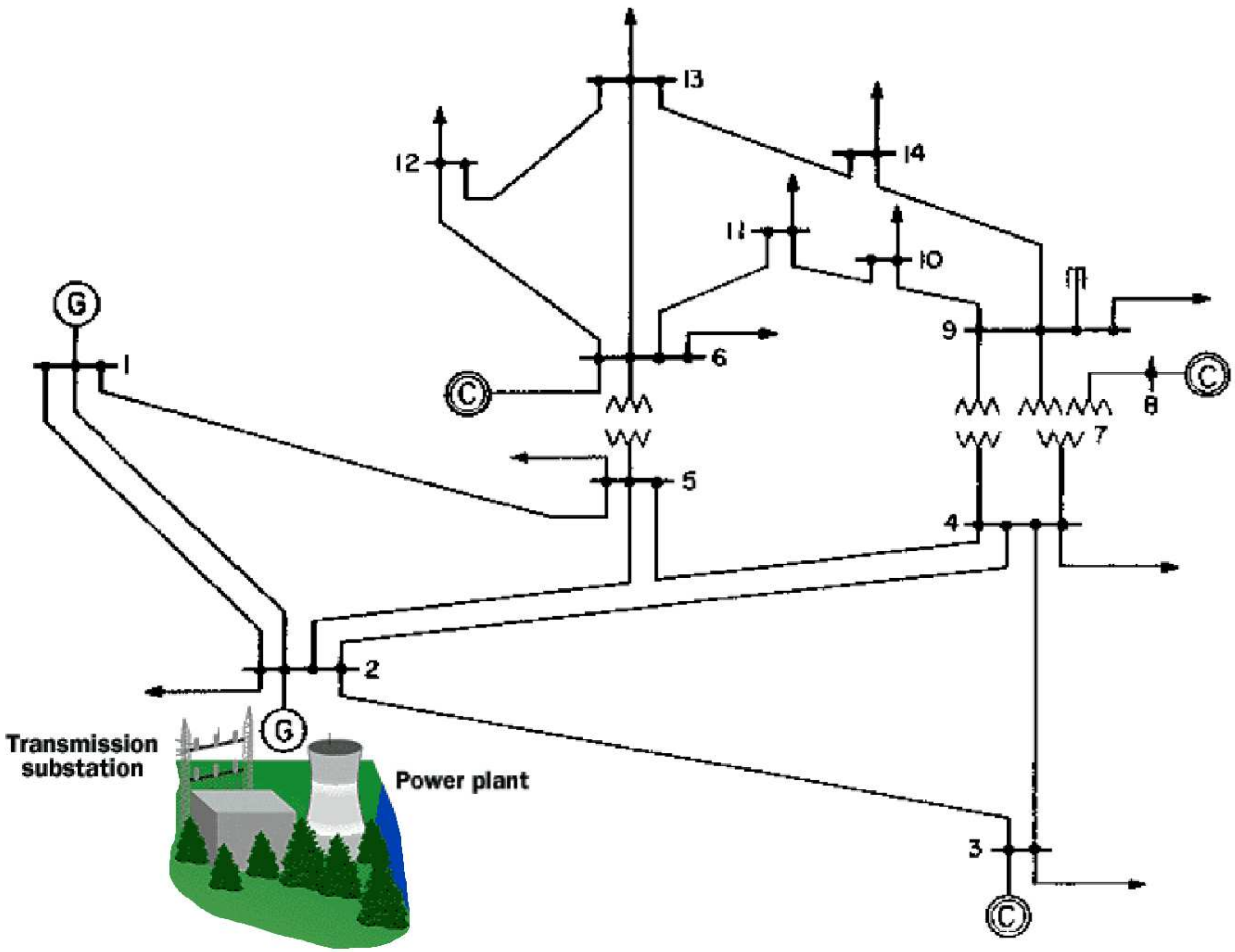
Expanding network expensive → improve control



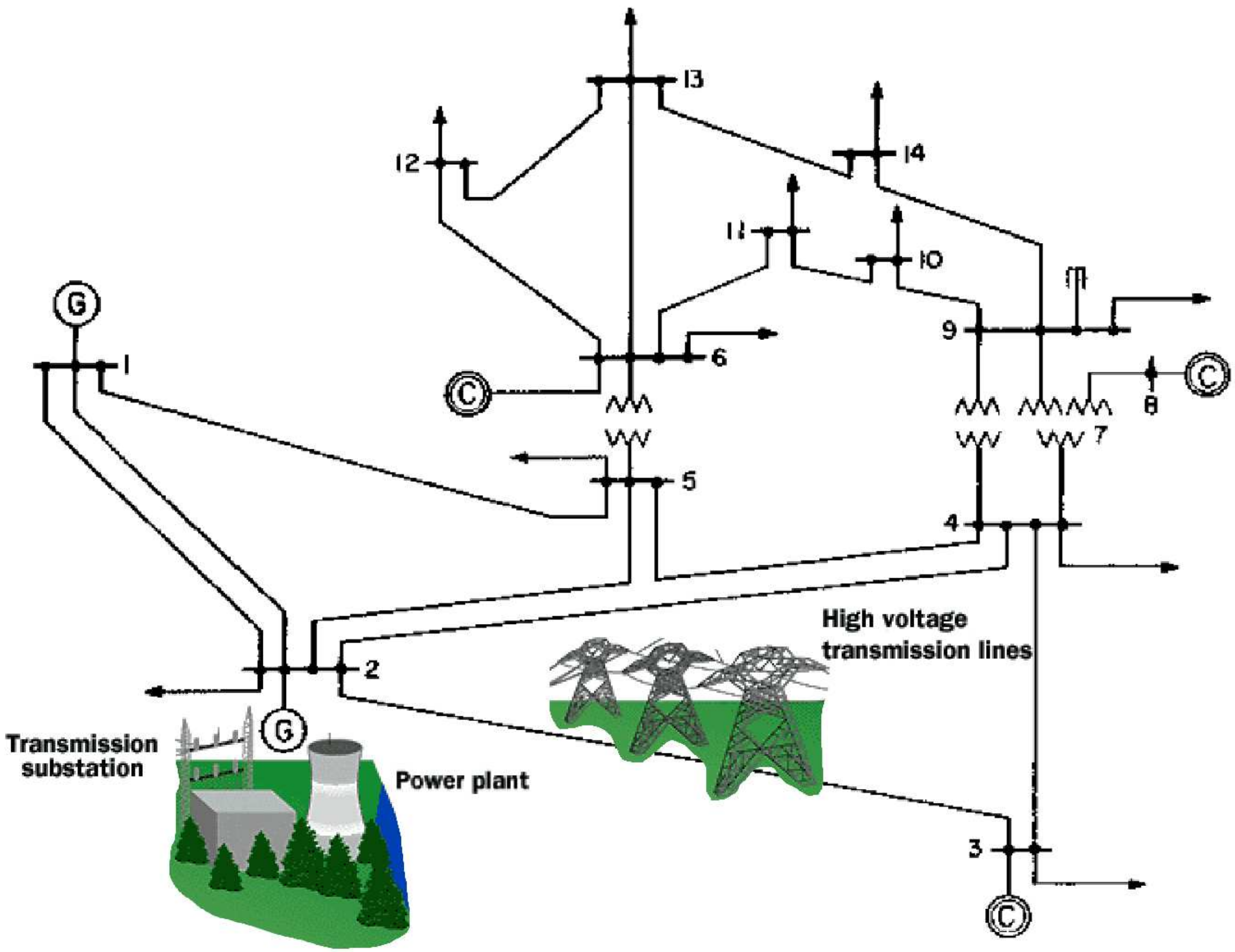
1.1. Example: power network



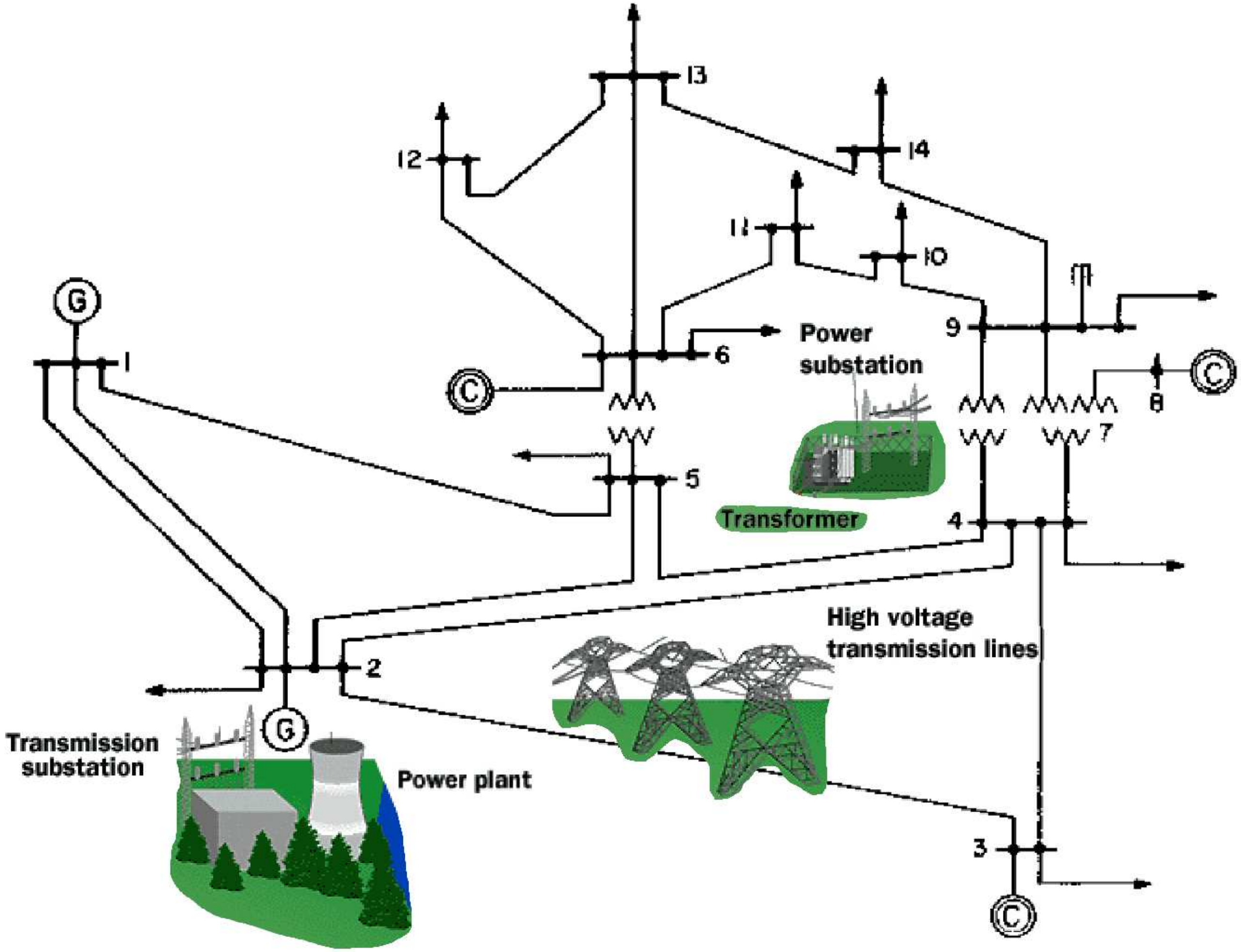
1.1. Example: power network



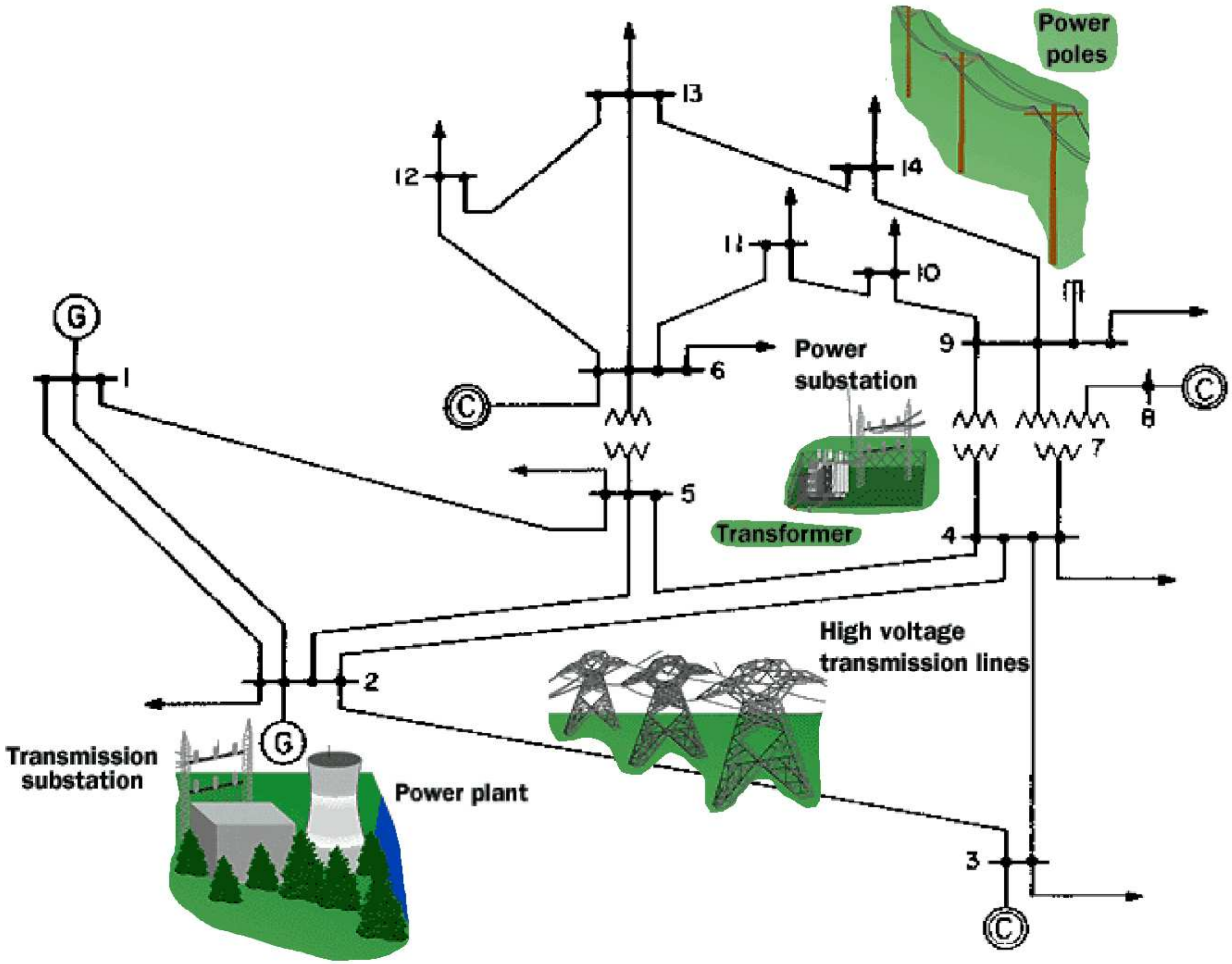
1.1. Example: power network



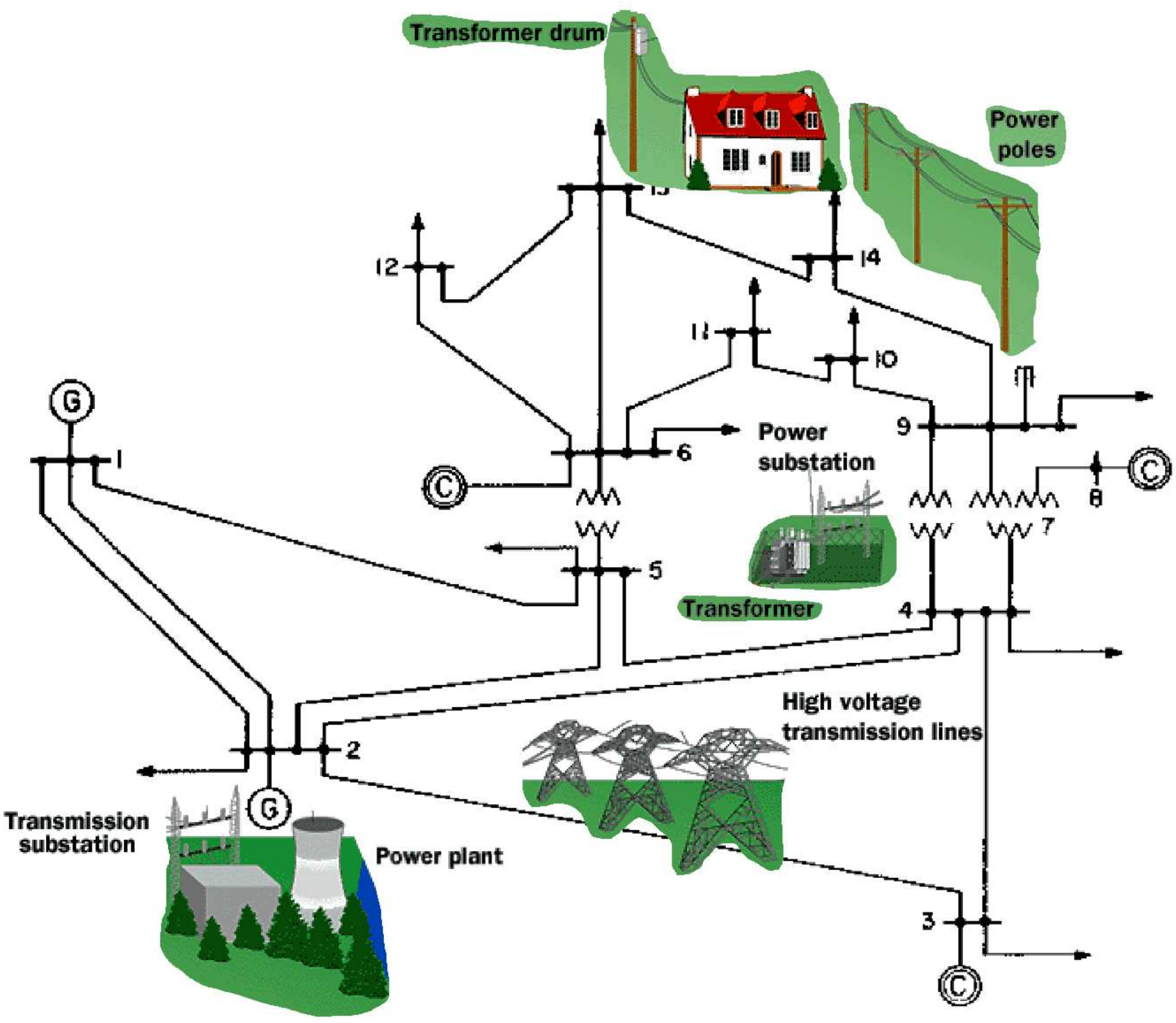
1.1. Example: power network



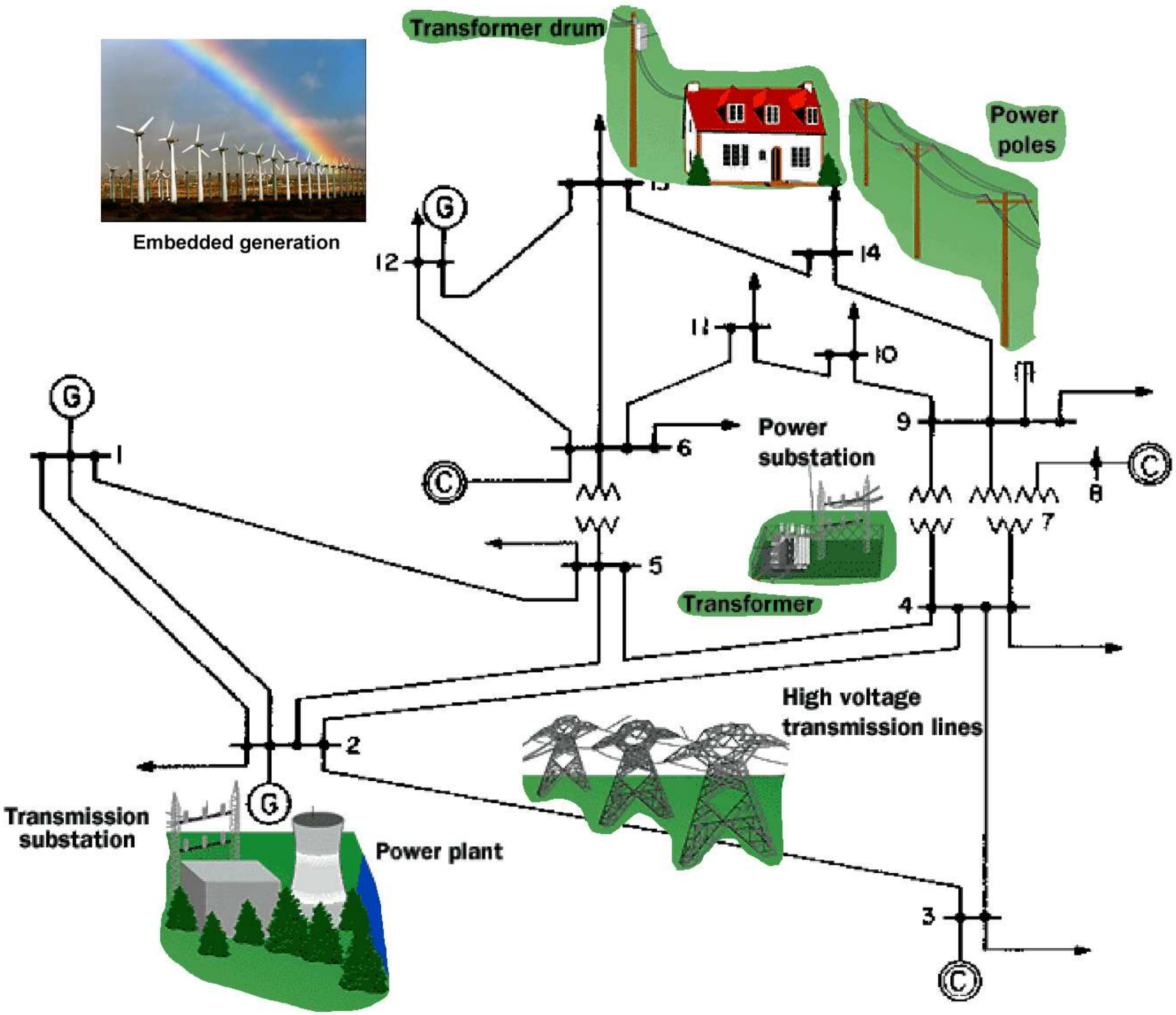
1.1. Example: power network



1.1. Example: power network



1.1. Example: power network

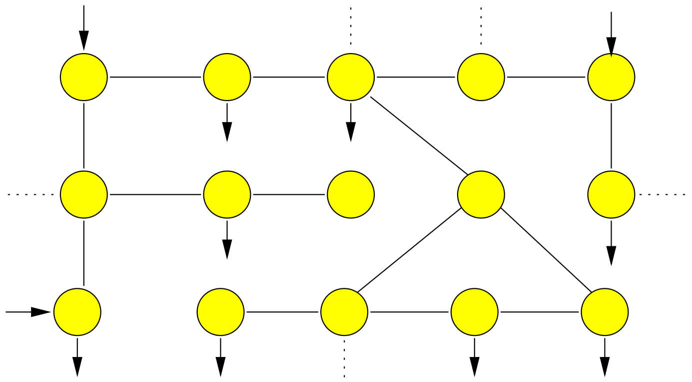


1.2. Unified representation

Consider transportation networks in a unified way.

In general, in a transportation network,

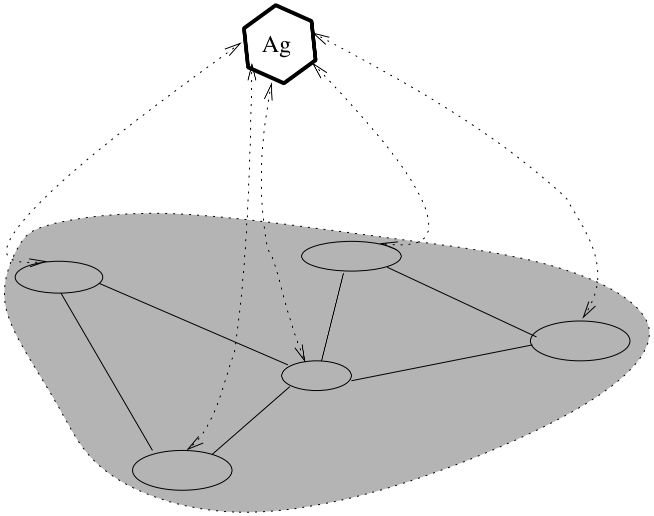
- there is **commodity**,
- which enters at **sources**,
- which **flows** over the network,
- to leave at **sinks**,
- while **components** influence the flow,
- which is restricted by **interconnections**.



Control goals:

- avoiding congestion,
- maximizing throughput,
- minimizing control actions,
- ...

2.1. Single-agent control



One control agent makes all decisions.

- Consider the whole network at once.
- Whole network controlled by a single agent,
  - Access to all actuators and sensors.

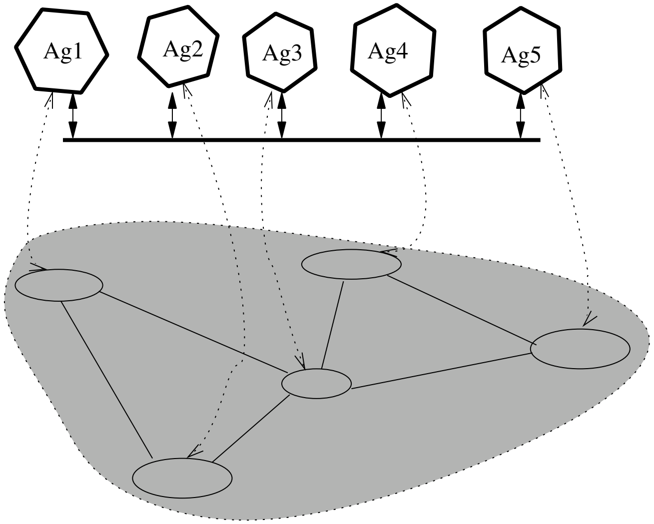
Issues with **single-agent** control:

- **Undesirable properties** with respect to
  robustness, reliability, scalability, responsiveness.
- **Technical issues**:
  communication delays, computational requirements, ...
- **Commercial issues**:
  unavailability of information, restricted control access, ...

2.2. Multi-agent control

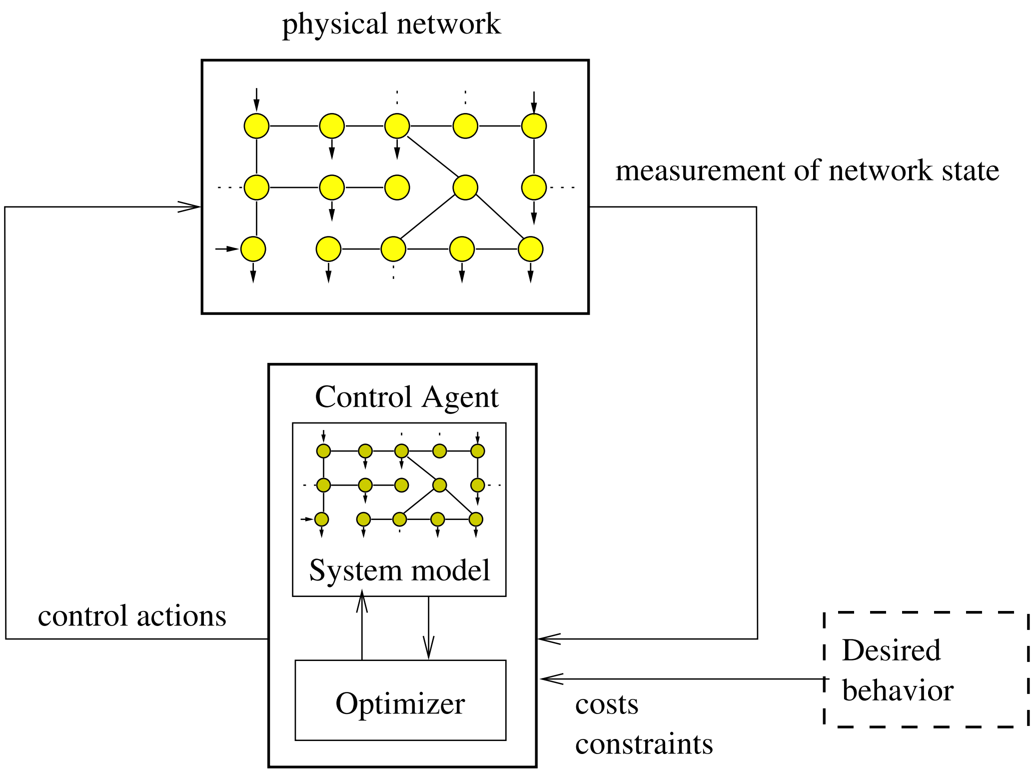
Instead of using one control agent,
use **multi-agent**, or **distributed**, approach.

- Consider subnetworks instead of overall network
- Each subnetwork controlled by single agent,
  - limited action capabilities,
  - limited informationg gathering,
  - limited processing skills.



Challenge for agents:
choose local actions with overall best performance

3.1. Model predictive control (MPC)



At each decision step, **solve optimization problem** that finds actions **over a horizon** giving best performance, subject to **model of system**, **objective function**, **constraints** on states, actions, and outputs.

3.1. Model predictive control (MPC)

After success in process industry, now also:
power, road traffic, and railway networks.

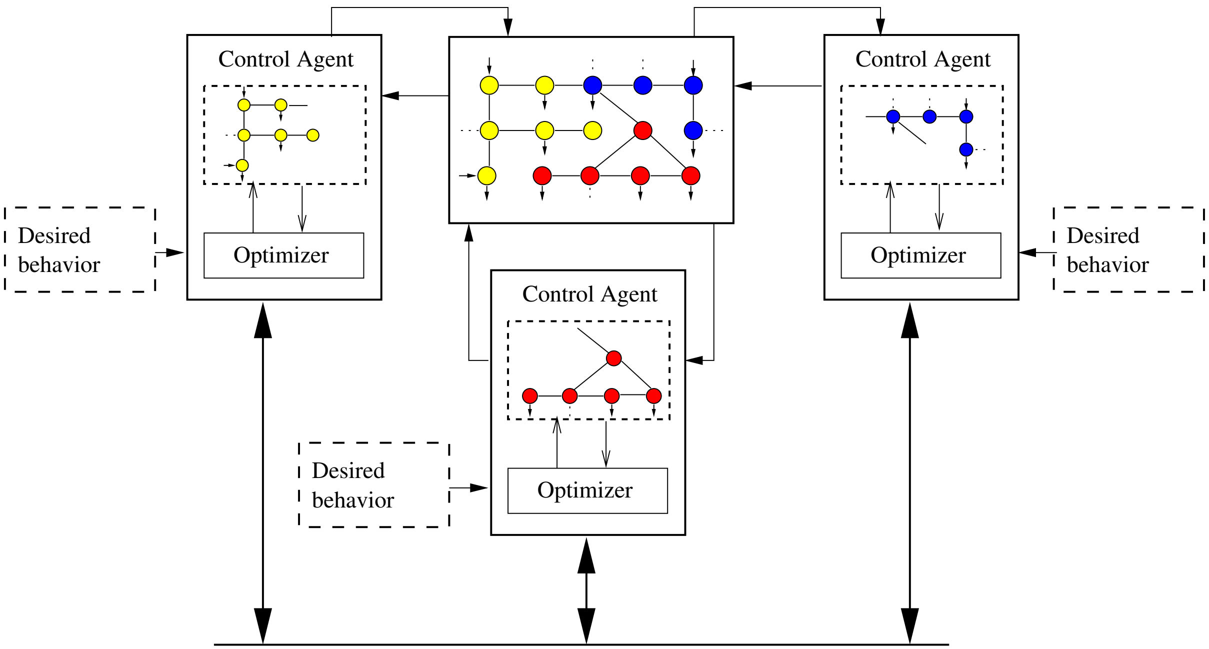
Main **advantages**:

- explicit way of integrating constraints,
- easy way of integrating forecasts,
- built in robustness against noise and modelling errors.

E.g., **for transportation networks**:

- include capacity limits on links,
- profiles of demands,
- maximum queue lengths,
- ...

3.2. Multi-agent MPC



Each agent has **limited action, information, and processing** skills, while **subnetworks influence each other** $\longrightarrow$ **uncertainty in predictions** over horizon

Communication and **cooperation** to reduce uncertainty

3.2. Multi-agent MPC

Agent controlling subnetwork i has to solve the following **local optimization problem**:

Find actions for subnetwork i **over a horizon** that

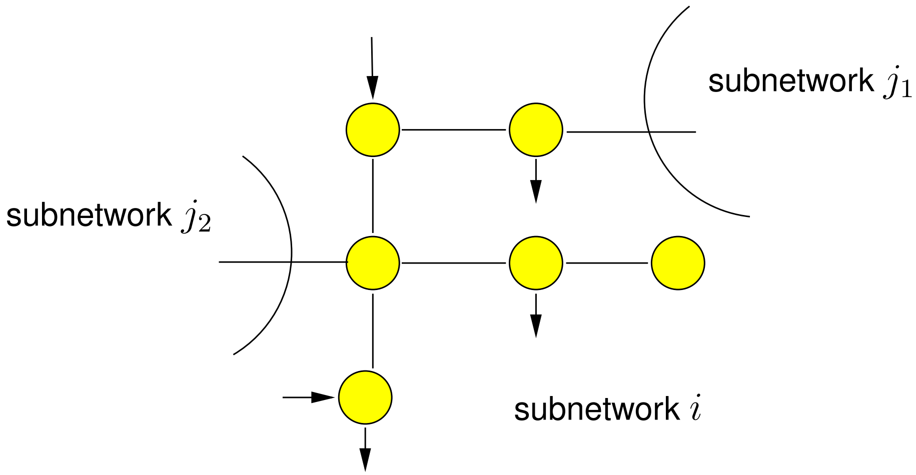
minimize a **local objective function**
based on local information of subnetwork i alone,

subject to:

the **dynamics of subnetwork** i,
and possibly additional constraints,
over the horizon.

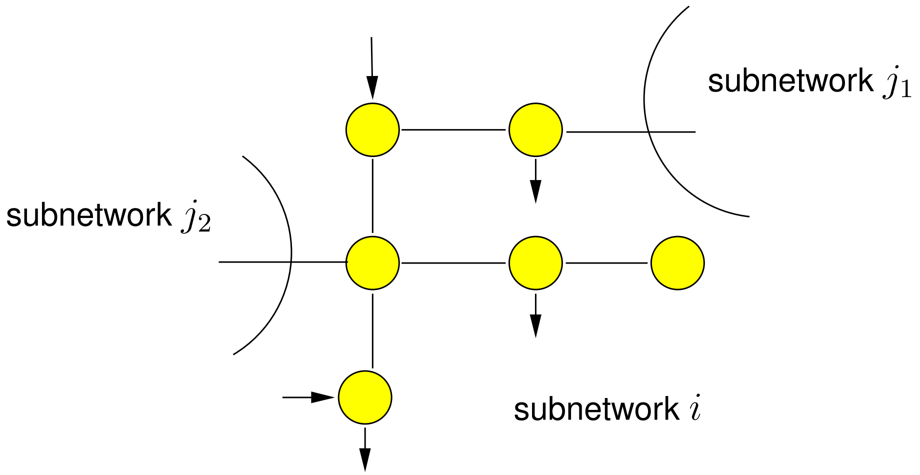
3.3. Subnetwork models

Agent controlling subnetwork i uses a **subnetwork model**.



3.3. Subnetwork models

Agent controlling subnetwork i uses a **subnetwork model**.

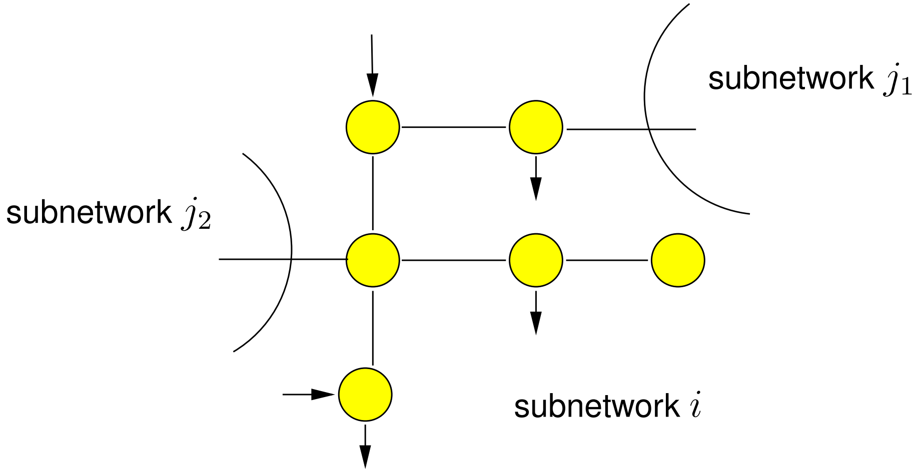


$$x^i_{k+1} = f^i(x^i_k, u^i_k, d^i_k, \ldots)$$
$$\ldots$$

- x^i_k are states
- u^i_k are actions
- d^i_k are disturbances

3.3. Subnetwork models

Agent controlling subnetwork i uses a **subnetwork model**.



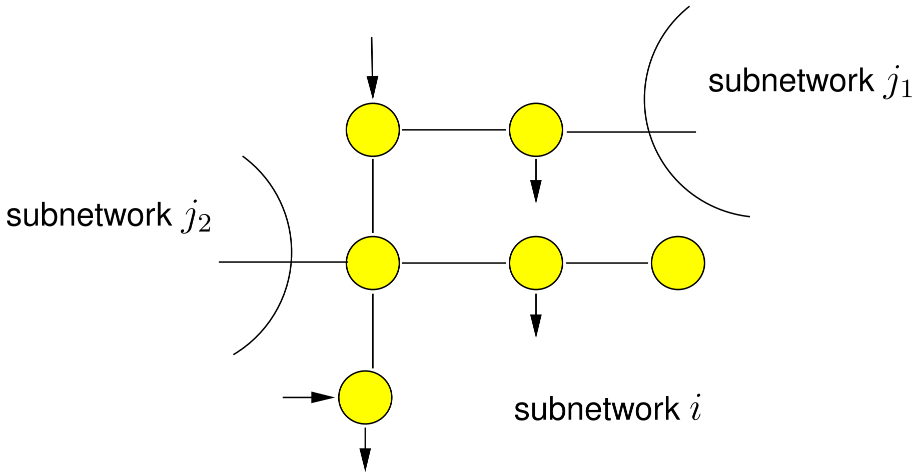
$$x_{k+1}^i = f^i(x_k^i, u_k^i, d_k^i, \ldots)$$
$$\ldots$$

- x_k^i are states
- u_k^i are actions
- d_k^i are disturbances

Subnetwork i has
neighbors $j \in N_i = \{j_1, \ldots, j_{m_i}\}$

3.3. Subnetwork models

Agent controlling subnetwork i uses a **subnetwork model**.



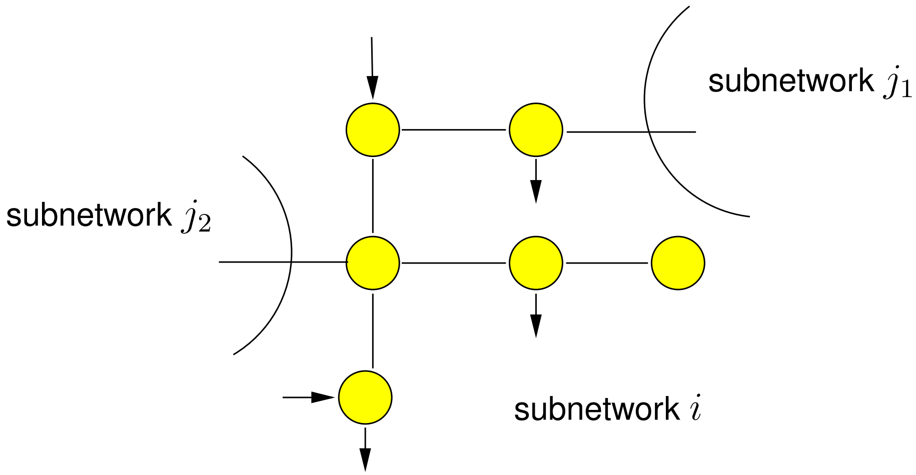
$$
\begin{aligned}
x^i_{k+1} &= f^i(x^i_k, u^i_k, d^i_k, w^{j_1\,i}_{\text{in},k}, \ldots, w^{j_{m_i}\,i}_{\text{in},k}) \\
w^{j\,i}_{\text{out},k} &= C^{j\,i}[(x^i_k)^T \ (u^i_k)^T \ (d^i_k)^T]^T \qquad \forall j \in N_i
\end{aligned}
$$

Subnetwork i has **neighbors** $j \in N_i = \{j_1, \ldots, j_{m_i}\}$

- x^i_k are states
- u^i_k are actions
- d^i_k are disturbances
- $w^{j\,i}_{\text{in},k}$ are **internetwork inputs**
- $w^{j\,i}_{\text{out},k}$ are **internetwork output**
- $C^{j\,i}$ is the internetwork output selection matrix

3.3. Subnetwork models

Agent controlling subnetwork i uses a **subnetwork model**.



$$x^i_{k+1} = f^i(x^i_k, u^i_k, d^i_k, w^{j_1 i}_{\text{in},k}, \ldots, w^{j_{m_i} i}_{\text{in},k})$$
$$w^{ji}_{\text{out},k} = C^{ji}[(x^i_k)^T \ (u^i_k)^T \ (d^i_k)^T]^T \qquad \forall j \in N_i$$

Subnetwork i has
neighbors $j \in N_i = \{j_1, \ldots, j_{m_i}\}$

Internetwork variables w_k:
internetwork input and output variables
Local variables z_k:
all other variables

- x^i_k are states
- u^i_k are actions
- d^i_k are disturbances
- $w^{ji}_{\text{in},k}$ are **internetwork inputs**
- $w^{ji}_{\text{out},k}$ are **internetwork output**
- C^{ji} is the internetwork output selection matrix

3.4. Local objective function

Agent controlling subnetwork i uses a **local objective function**,

$$J^i_{\text{local}} = \sum_{s=0}^{N-1} \text{cost based on variables per prediction step } s.$$

Notation: tilde ˜ over variable to represent variable over horizon, e.g.,

$$J^i_{\text{local}}(\tilde{z}^i_k) = \sum_{s=0}^{N-1} ||z^i_{k+s}||^2.$$

Objective-function distinction based on type of variables involved:

- **any variable**, whether internetwork variable or not,
  $J^i_{\text{local}}(\tilde{z}_k, \tilde{w}_k)$.
- **only local variables**, so no internetwork variables,
  $J^i_{\text{local}}(\tilde{z}_k)$.

3.4. Local optimization problem

Thus, the optimization problem becomes:

$$\min_{\tilde{z}_k^i} J^i_{\text{local}}(\tilde{z}_k)$$

subject to

$$\begin{aligned}
x^i_{k+1} &= f^i(x^i_k, u^i_k, d^i_k, w^{j_1 i}_{\text{in},k}, \ldots, w^{j_{m_i} i}_{\text{in},k}) \\
w^{ji}_{\text{out},k} &= C^{ji}[(x^i_k)^T \ (u^i_k)^T \ (d^i_k)^T]^T \qquad \forall j \in N_i \\
&\vdots \qquad \vdots \\
x^i_{k+N} &= f^i(x^i_{k+N-1}, u^i_{k+N-1}, d^i_{k+N-1}, w^{j_1 i}_{\text{in},k+N-1}, \ldots, w^{j_{m_i} i}_{\text{in},k+N-1}) \\
w^{ji}_{\text{out},k+N} &= C^{ji}[(x^i_{k+N})^T \ (u^i_{k+N})^T \ (d^i_{k+N})^T]^T \qquad \forall j \in N_i
\end{aligned}$$

and perhaps additional constraints.

3.4. Local optimization problem

In the **overall** network internetwork variables between subnetworks i and j satisfy: **interconnecting constraints**

$$
\begin{aligned}
\tilde{w}^{ji}_{\text{in},k} &= \tilde{w}^{ij}_{\text{out},k} \\
\tilde{w}^{ji}_{\text{out},k} &= \tilde{w}^{ij}_{\text{in},k}
\end{aligned}
$$

Note that the agent controlling subnetwork i **does not know** the values of $\tilde{w}^{ij}_{\text{in},k}$ and $\tilde{w}^{ij}_{\text{out},k}$.

3.4. Local optimization problem

In the **overall** network internetwork variables between subnetworks i and j satisfy: **interconnecting constraints**

$$\tilde{w}^{ji}_{\text{in},k} = \tilde{w}^{ij}_{\text{out},k}$$
$$\tilde{w}^{ji}_{\text{out},k} = \tilde{w}^{ij}_{\text{in},k}$$

Note that the agent controlling subnetwork i **does not know** the values of $\tilde{w}^{ij}_{\text{in},k}$ and $\tilde{w}^{ij}_{\text{out},k}$.

For making predictions of subnetwork i agent is missing information to make sure the interconnecting constraints are satisfied:

$$\tilde{w}^{ji}_{\text{in},k} = \textbf{??}$$
$$\textbf{??} = \tilde{w}^{ji}_{\text{out},k} \quad \text{for } j \in N_i.$$

3.4. Local optimization problem

In the **overall** network internetwork variables between subnetworks i and j satisfy: **interconnecting constraints**

$$\tilde{w}^{ji}_{\text{in},k} = \tilde{w}^{ij}_{\text{out},k}$$
$$\tilde{w}^{ji}_{\text{out},k} = \tilde{w}^{ij}_{\text{in},k}$$

Note that the agent controlling subnetwork i **does not know** the values of $\tilde{w}^{ij}_{\text{in},k}$ and $\tilde{w}^{ij}_{\text{out},k}$.

For making predictions of subnetwork i agent is missing information to make sure the interconnecting constraints are satisfied:

$$\tilde{w}^{ji}_{\text{in},k} = \textbf{??}$$
$$\textbf{??} = \tilde{w}^{ji}_{\text{out},k} \quad \text{for } j \in N_i.$$

$\longrightarrow$ ignore, estimate, negotiate, learn, ..., the unknowns.

3.4. Obtaining agreement

A **negotiation** approach:

obtain **agreement** on internetwork variables **through iterations**.

All agents:

- compute both optimal local variables and optimal internetwork variables;
- send out the values of internetwork variables to neighboring agents;
- re-optimize given received information and make new offer.

Additional cost term to encourage convergence:

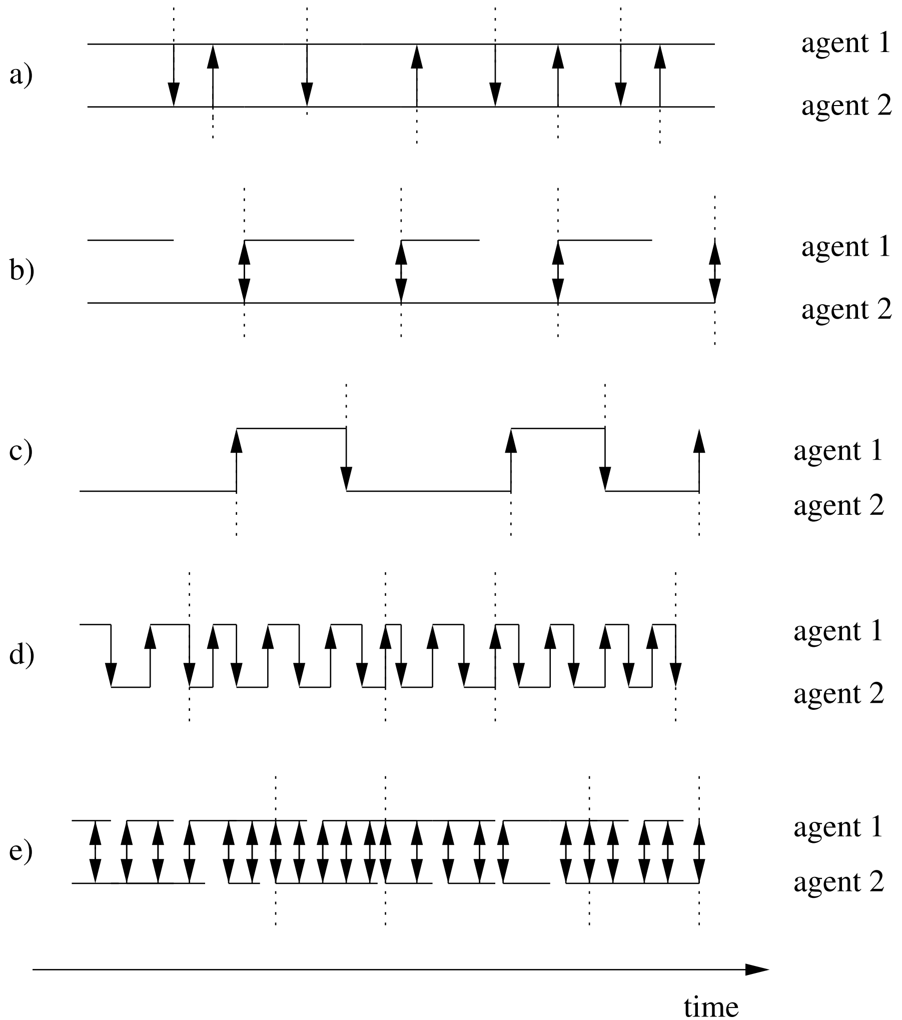
$$\min_{\tilde{z}_k^i, \tilde{w}_k^i} J_{\text{local}}^i(\tilde{z}_k^i) + \sum_{j \in N_i} J_{\text{inter}}^i(\tilde{w}_k^{ji}),$$

subject to the dynamics of subnetwork i over the horizon.

3.4. Obtaining agreement

Different types of
communication and decision making schemes

serial versus parallel,
single versus multiple iterations.



3.5. Serial versus parallel

Serial additional cost term:

$$J^i_{\text{inter}}(\tilde{w}^{ji}_k) = \left[(\lambda^{ji}_s)^T \; (-\lambda^{ij}_s)^T \right] \begin{bmatrix} \tilde{w}^{ji}_{\text{in},k} \\ \tilde{w}^{ji}_{\text{out},k} \end{bmatrix}$$

$$+ \frac{c}{2} \left\| \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \tilde{w}^{ij}_{\text{in, prev},k} \\ \tilde{w}^{ij}_{\text{out, prev},k} \end{bmatrix} - \begin{bmatrix} 0 & I \\ I & 0 \end{bmatrix} \begin{bmatrix} \tilde{w}^{ji}_{\text{in},k} \\ \tilde{w}^{ji}_{\text{out},k} \end{bmatrix} \right\|^2 .$$

parameters:

- $\lambda^{ji}_s, \lambda^{ij}_s$: Lagrangian multipliers,
- c : penalty to force internetwork inputs to equal internetwork outputs,
- $\tilde{w}^{ij}_{\text{in,prev},k}, \tilde{w}^{ij}_{\text{out,prev},k}$: information received from agent j at current or previous iteration.

3.5. Serial versus parallel

Parallel additional cost term:

$$J^{i}_{\text{inter}}(\tilde{w}^{ji}_{k}) = \left[(\lambda^{ji}_{s})^T \; (-\lambda^{ij}_{s})^T \right] \begin{bmatrix} \tilde{w}^{ji}_{\text{in},k} \\ \tilde{w}^{ji}_{\text{out},k} \end{bmatrix} + \frac{c}{2} \left\| \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \tilde{w}^{ij}_{\text{in,prev},k} \\ \tilde{w}^{ij}_{\text{out,prev},k} \end{bmatrix} - \begin{bmatrix} 0 & I \\ I & 0 \end{bmatrix} \begin{bmatrix} \tilde{w}^{ji}_{\text{in},k} \\ \tilde{w}^{ji}_{\text{out},k} \end{bmatrix} \right\|^2 + \frac{b-c}{2} \left\| \begin{bmatrix} \tilde{w}^{ji}_{\text{in},k} \\ \tilde{w}^{ji}_{\text{out},k} \end{bmatrix} - \begin{bmatrix} \tilde{w}^{ji}_{\text{in,prev},k} \\ \tilde{w}^{ji}_{\text{out,prev},k} \end{bmatrix} \right\|^2 .$$

parameters:

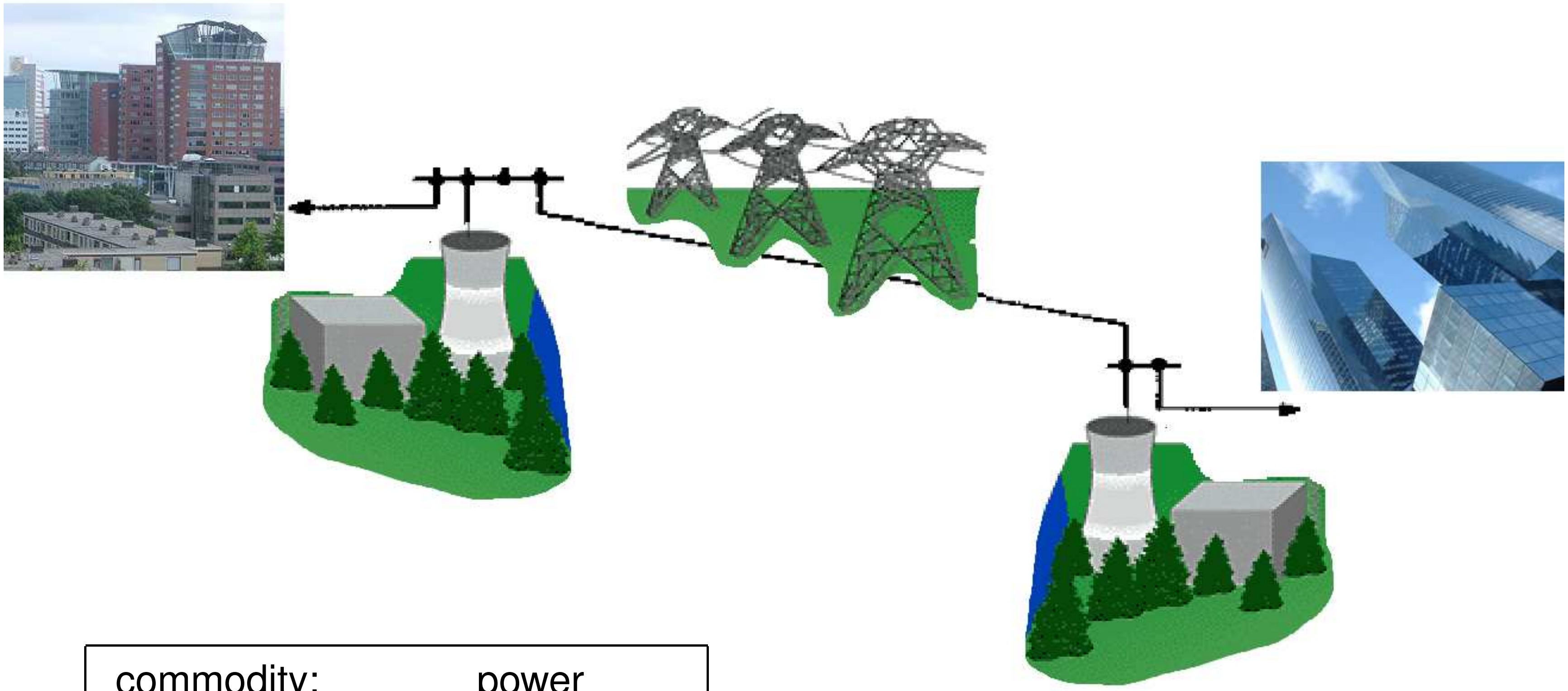
- $\lambda^{ji}_{s}, \lambda^{ij}_{s}$: Lagrangian multipliers,
- b : part of penalty on changes per iteration of internetwork variables agent i,
- c : penalty to force internetwork inputs to equal internetwork outputs,
- $\tilde{w}^{ij}_{\text{in,prev},k}, \tilde{w}^{ij}_{\text{out,prev},k}$: information received from agent j at previous iteration only,
- $\tilde{w}^{ji}_{\text{in,prev},k}, \tilde{w}^{ji}_{\text{out,prev},k}$: internetwork variables of agent i of previous iteration.

3.5. Serial versus parallel

Update of the **Lagrangian multipliers**:

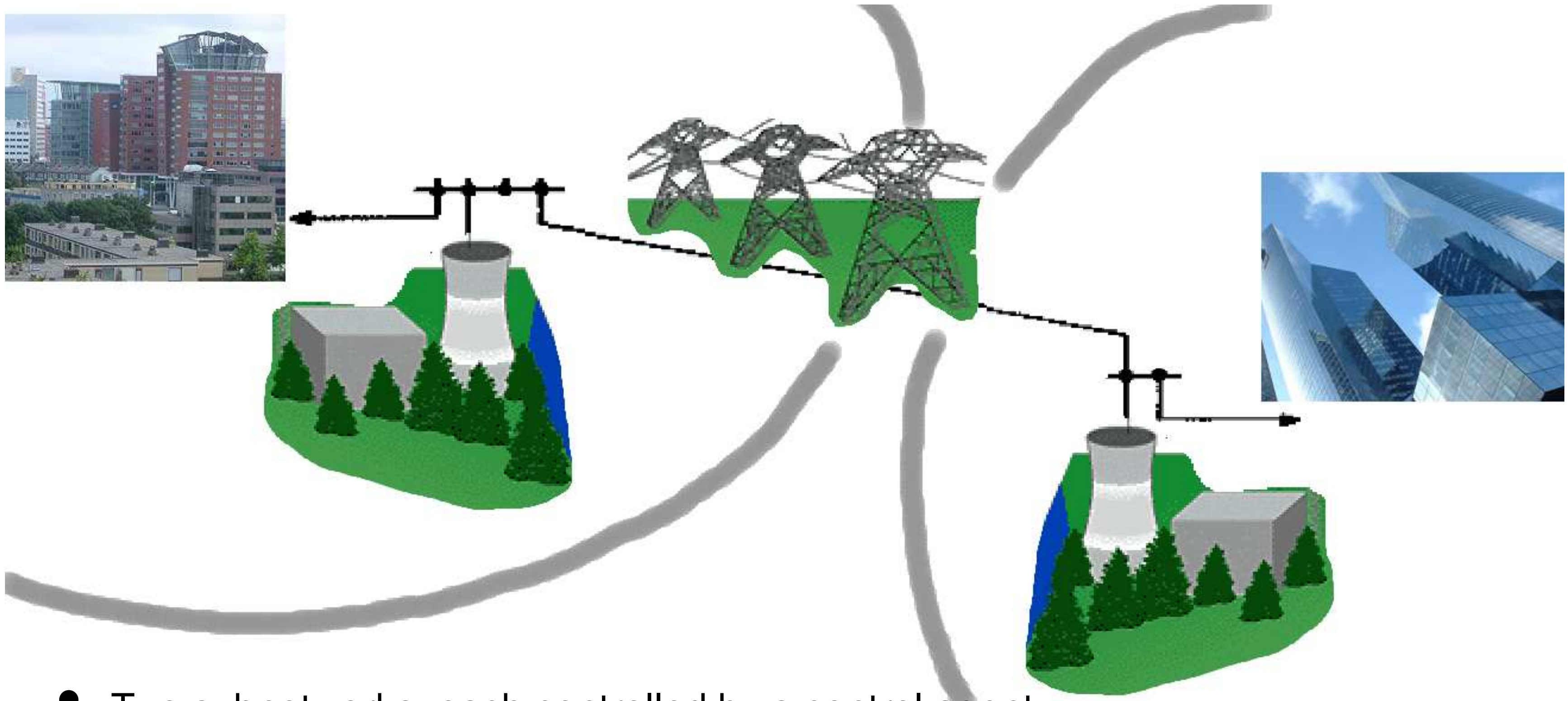
$$\lambda_{s+1}^{ji} = \lambda_{s}^{ji} + c(\tilde{w}_{\mathsf{in},s+1|k}^{ji} - \tilde{w}_{\mathsf{out},s+1|k}^{ij})$$
$$\lambda_{s+1}^{ij} = \lambda_{s}^{ij} + c(\tilde{w}_{\mathsf{in},s+1|k}^{ij} - \tilde{w}_{\mathsf{out},s+1|k}^{ji}).$$

4. Example: load-frequency control



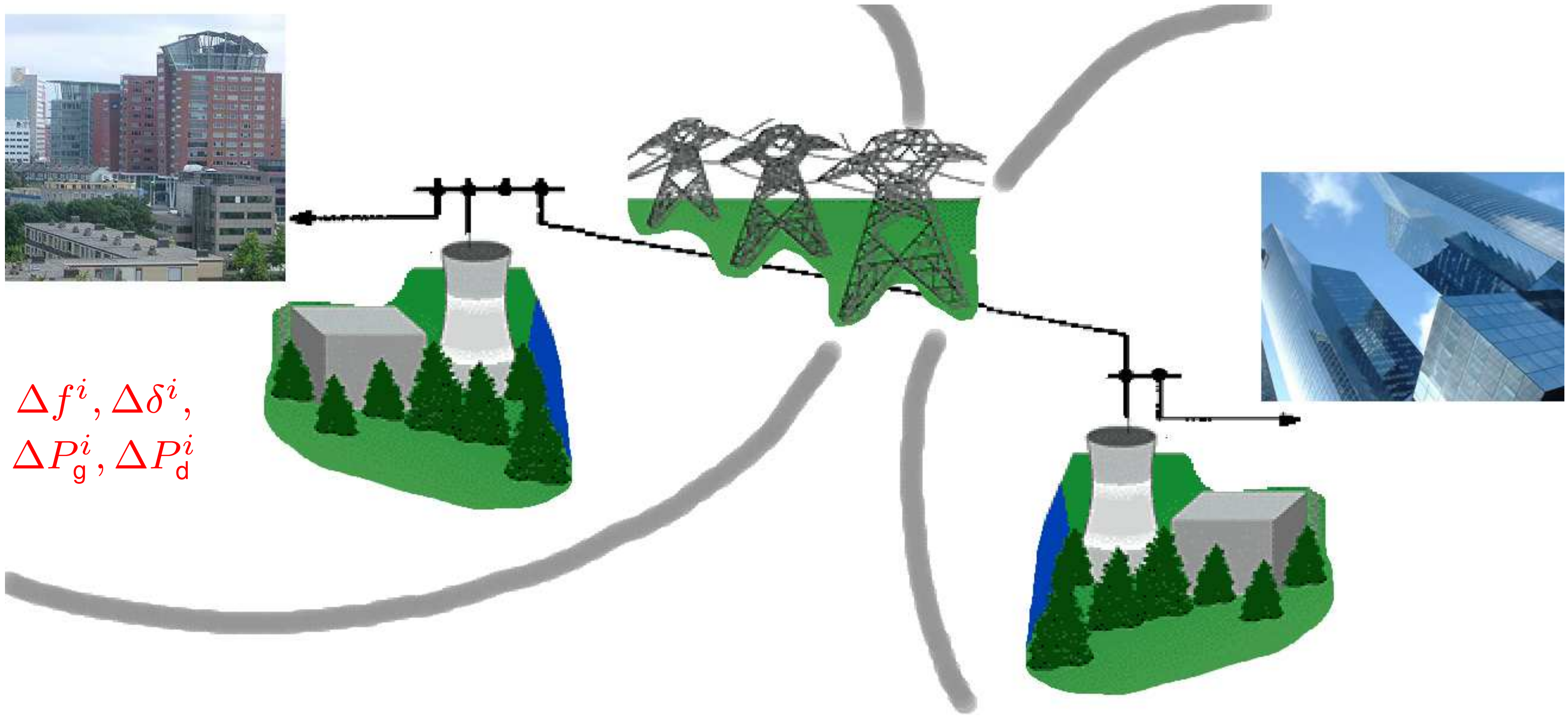
commodity:	power
sources:	generators
sinks:	loads
interconnections:	power lines

4. Example: load-frequency control



- Two subnetworks, each controlled by a control agent.
- Imbalance generation and consumption leads to *frequency deviations*.
- Main goal: maintain frequency deviation in each subnetwork close to zero.
- How? by controlling generation.

4. Example: load-frequency control



Δf: frequency deviation; $\Delta\delta$: angle deviation

ΔP_{d}: load deviation; ΔP_{g}: generation deviation

4. Example: load-frequency control

Agent controlling subnetwork i gets local objective function:

$$J^i_{\text{local}}(\tilde{z}^i_k) = \sum_{p=0}^{N-1} q^i_{\Delta f}(\Delta f^i_{k+p+1})^2 + q^i_{\Delta P_g}(\Delta P^i_{\text{g},k+p})^2,$$

where $q^i_{\Delta f} = 100$ and $q^i_{\Delta P_{\text{g}}} = 10$.

4. Example: load-frequency control

Subnetwork i can be modeled as:

$$\Delta\dot{\delta}^i = 2\pi\Delta f^i$$

$$\Delta\dot{f}^i = -\frac{1}{T_{Pi}}\Delta f^i + \frac{K_{Pi}}{T_{Pi}}\Delta P_{\mathrm{g}}^i - \frac{K_{Pi}}{T_{Pi}}\Delta P_{\mathrm{d}}^i + \frac{K_{Pi}K_{S_{ij}}}{2\pi T_{Pi}}(\Delta\delta^j - \Delta\delta^i).$$

Frequency deviation Δf^i in subnetwork i influenced by:

- consumption deviation ΔP_{d}^i within subnetwork,
- generation change ΔP_{g}^i within subnetwork,
- change in power flowing over interconnection to other subnetwork, $(K_{S_{ij}}/2\pi)(\Delta\delta^j - \Delta\delta^i)$.

Thus:

power flowing over interconnection depends on variable of both subnetworks.
agent i does not know $\Delta\delta^j$, agent j does not know $\Delta\delta^i$.

4. Example: load-frequency control

Continuous-time subnetwork model discretized into:

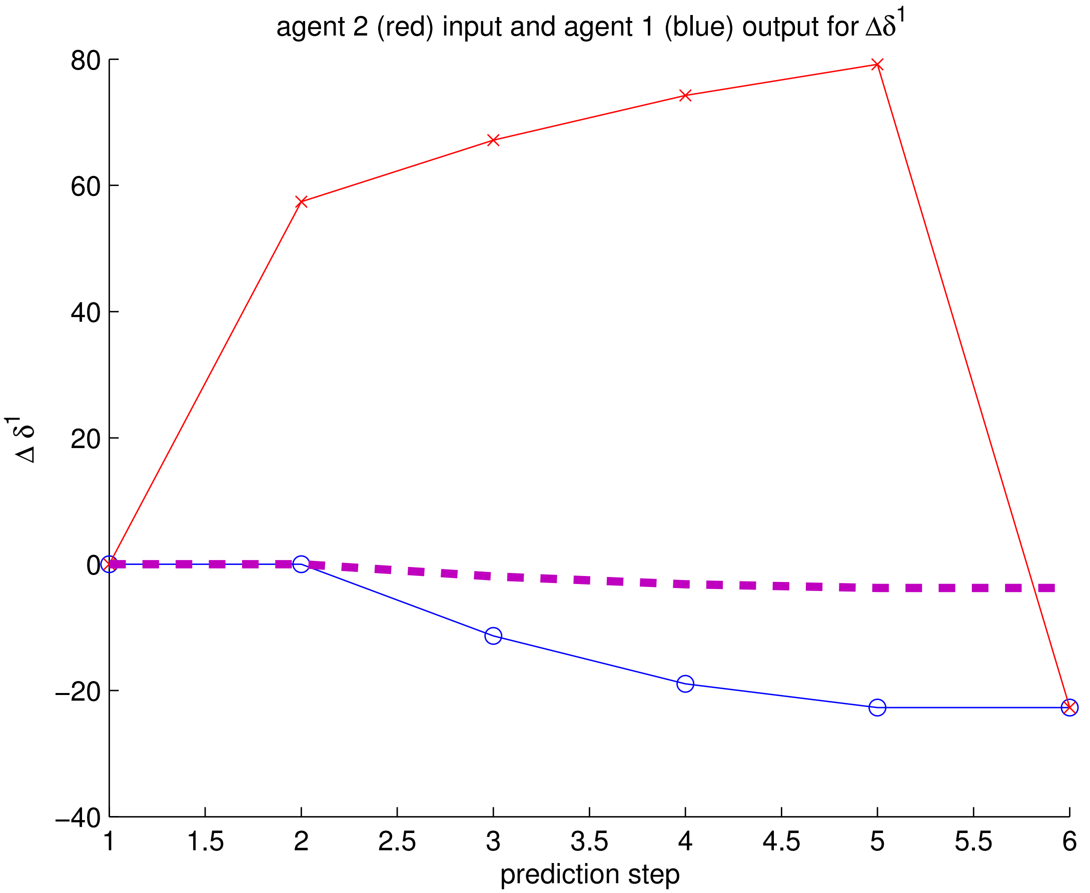
$$x^i_{k+1} = A^i x^i_k + B^i_1 u^i_k + B^i_2 d^i_k + B^i_3 w^{ji}_{\text{in},k}$$
$$w^{ji}_{\text{out},k} = C^{ji}[(x^i_k)^T \; (u^i_k)^T \; (d^i_k)^T]^T,$$

where for subnetwork i

- $w^{ji}_{\text{in},k}$ represents the internetwork input $\Delta\delta^j$ from subnetwork j,
- $w^{ji}_{\text{out},k}$ represents the internetwork output $\Delta\delta^i$ to subnetwork j.

4. Example: load-frequency control

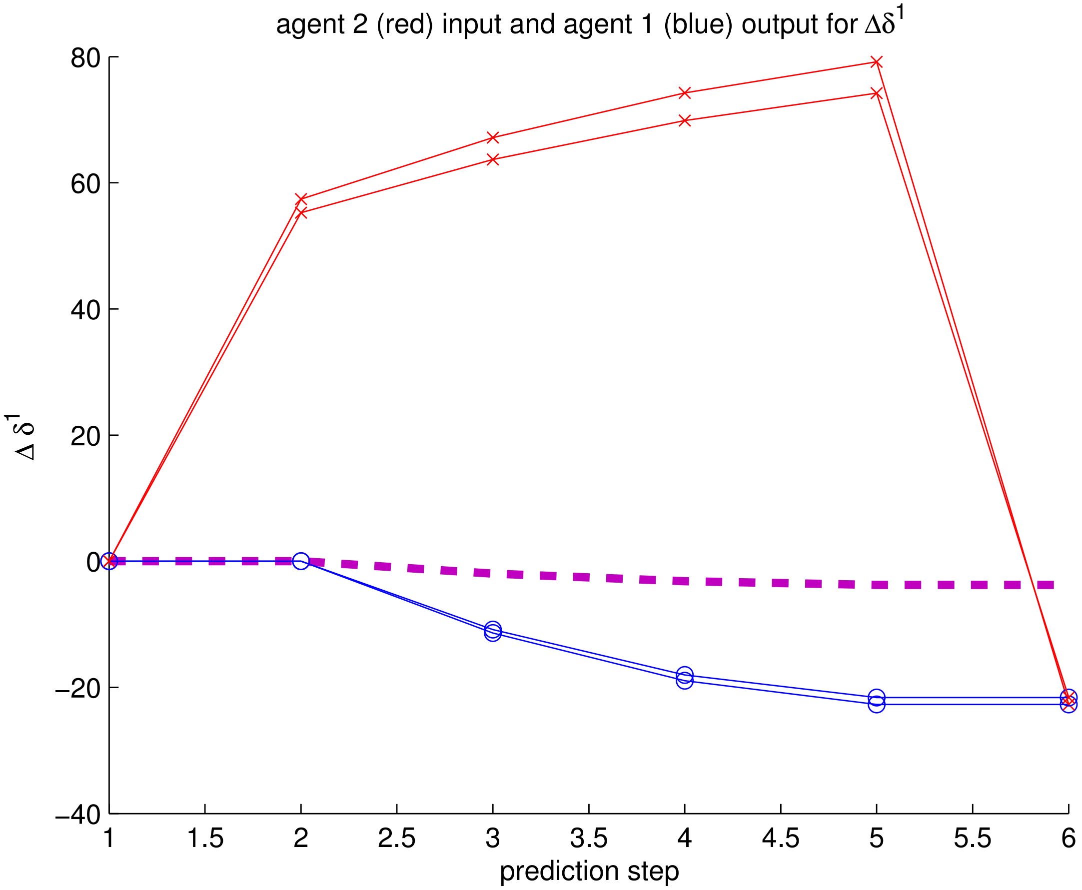
Example of serial run:



Iteration 1

4. Example: load-frequency control

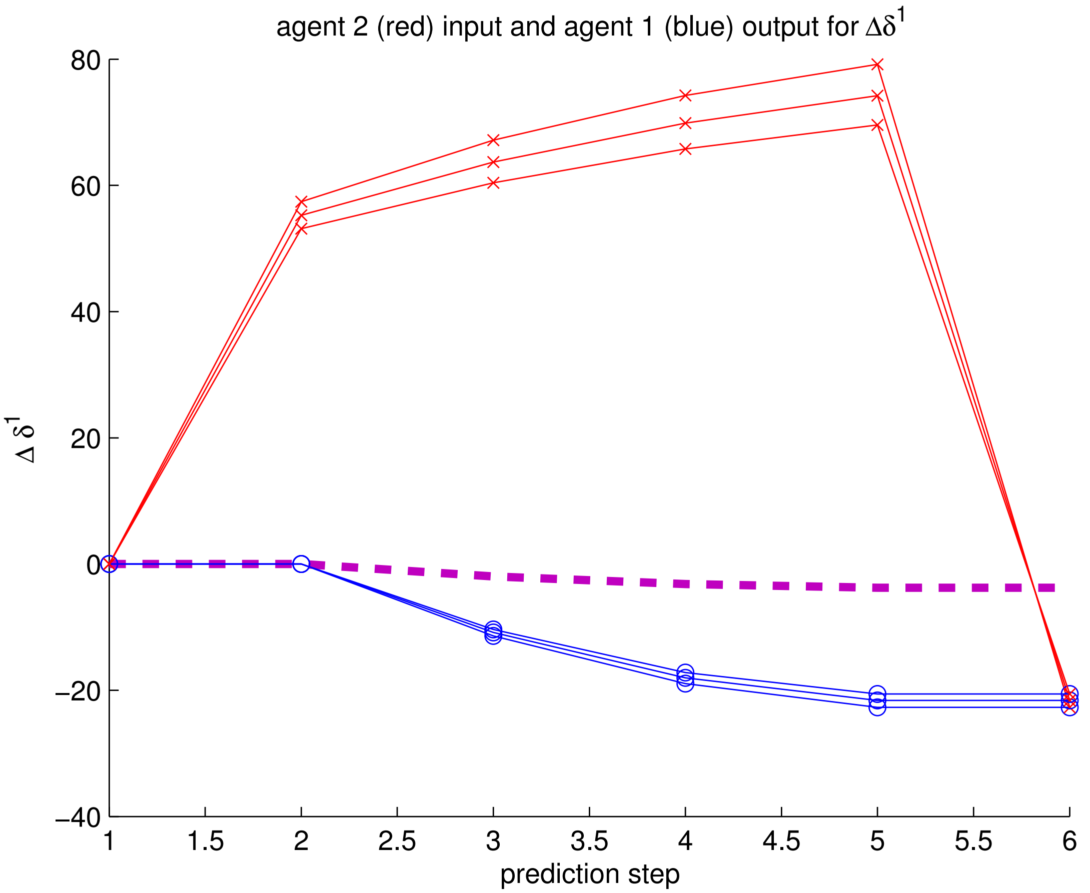
Example of serial run:



Iteration 2

4. Example: load-frequency control

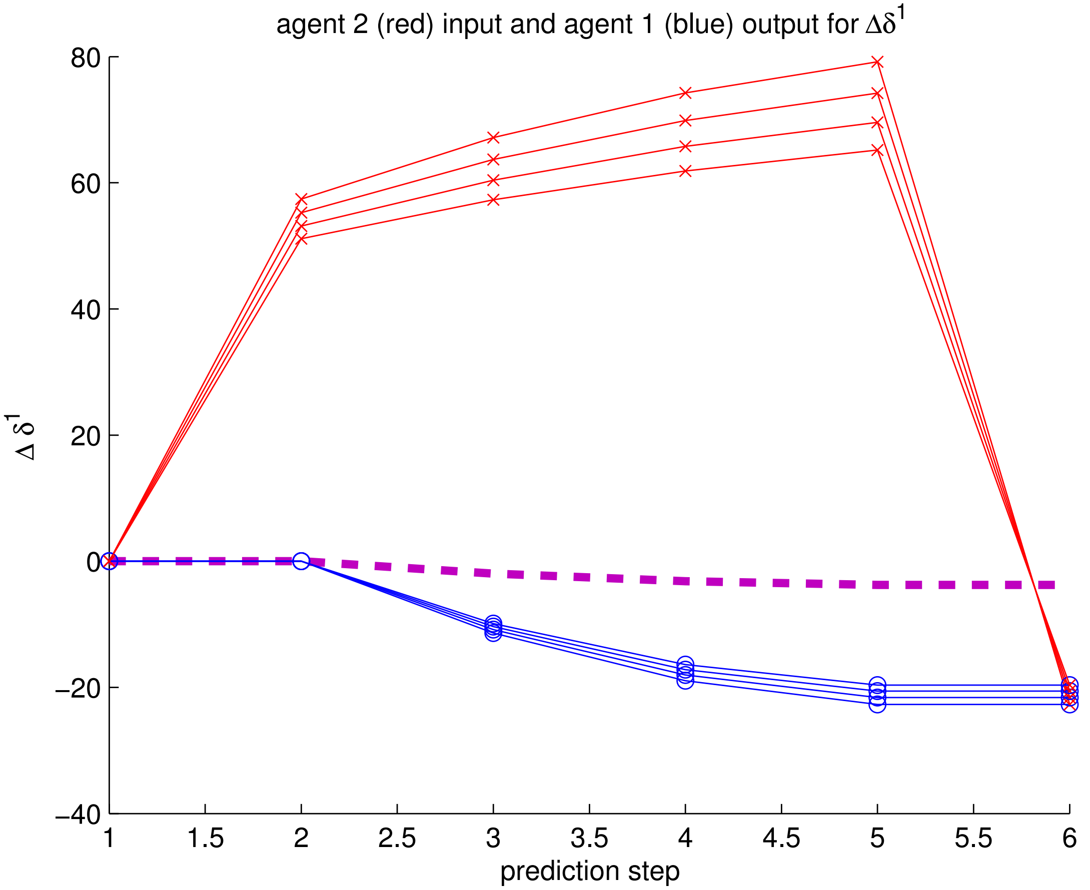
Example of serial run:



Iteration 3

4. Example: load-frequency control

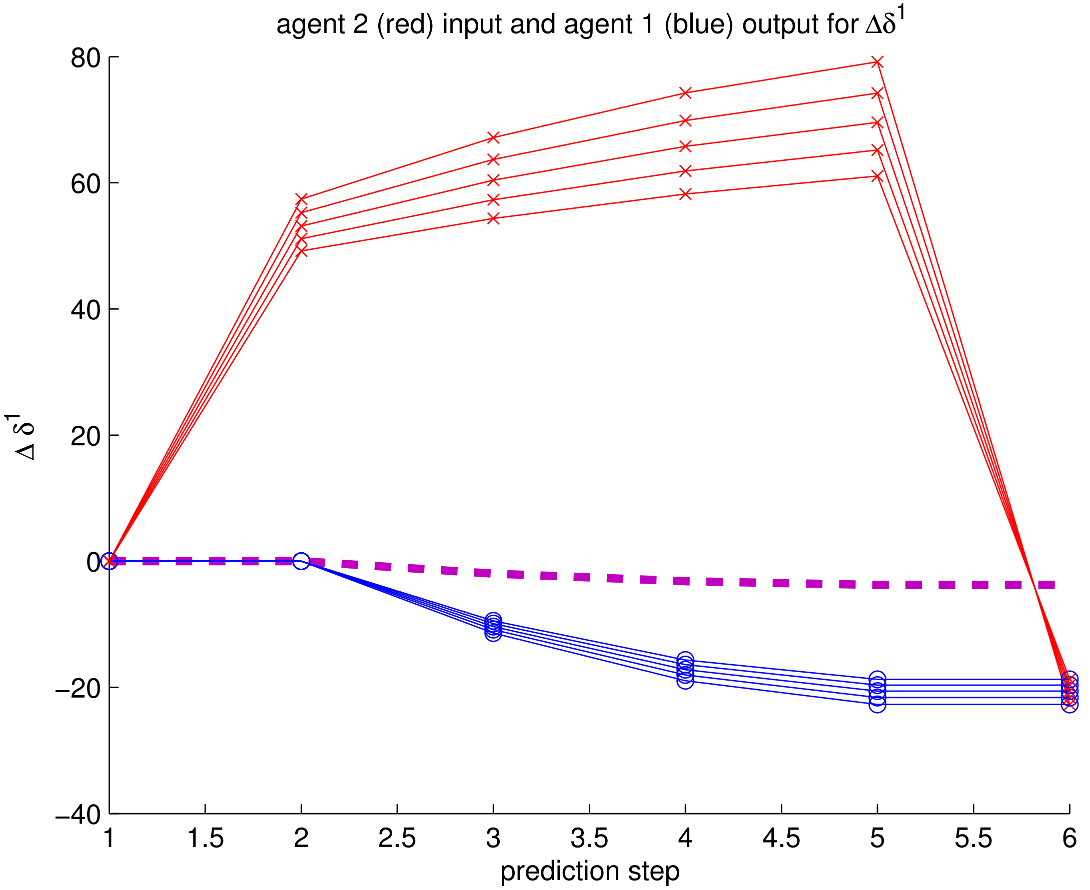
Example of serial run:



Iteration 4

4. Example: load-frequency control

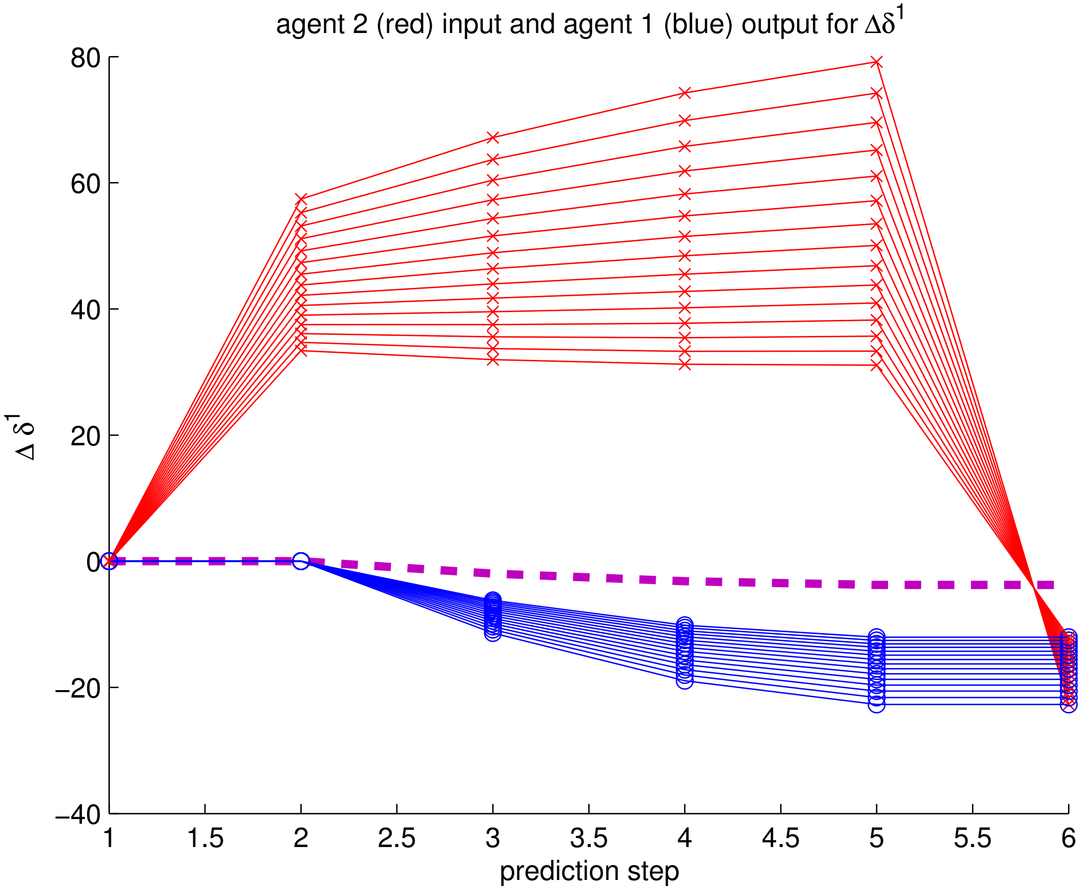
Example of serial run:



Iteration 5

4. Example: load-frequency control

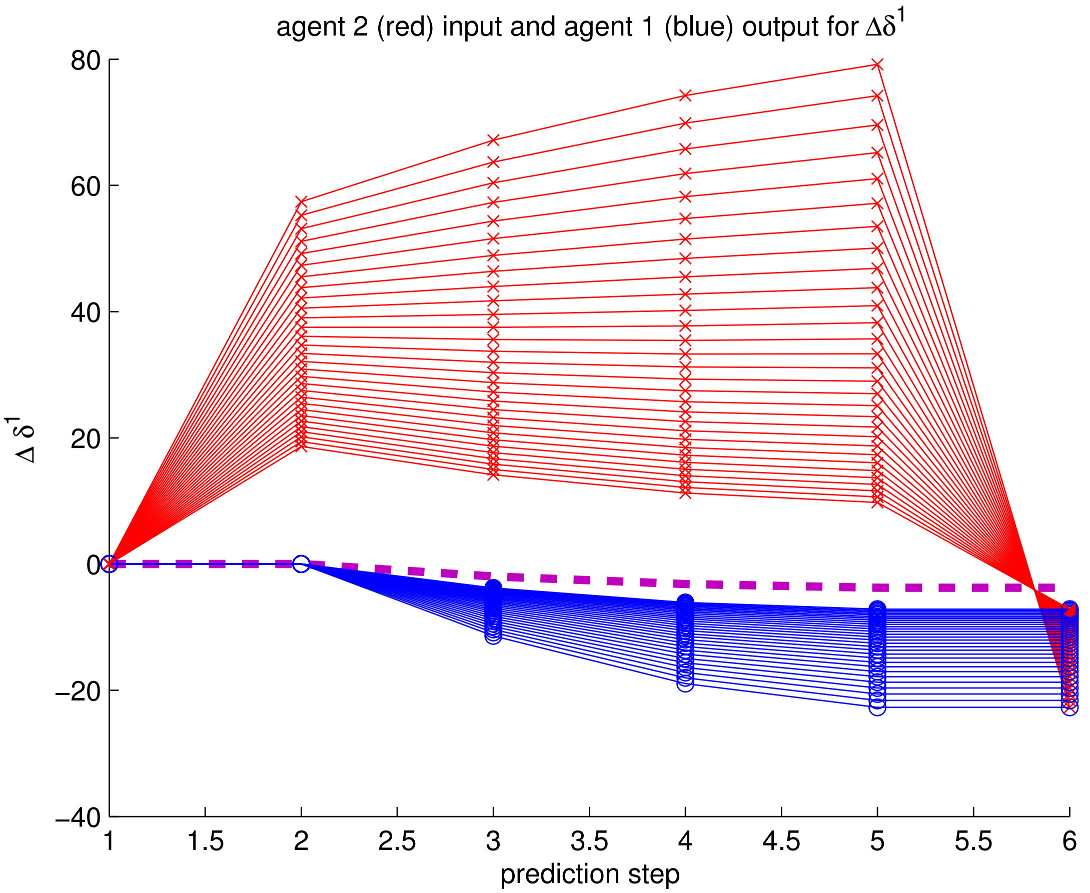
Example of serial run:



Iteration 15

4. Example: load-frequency control

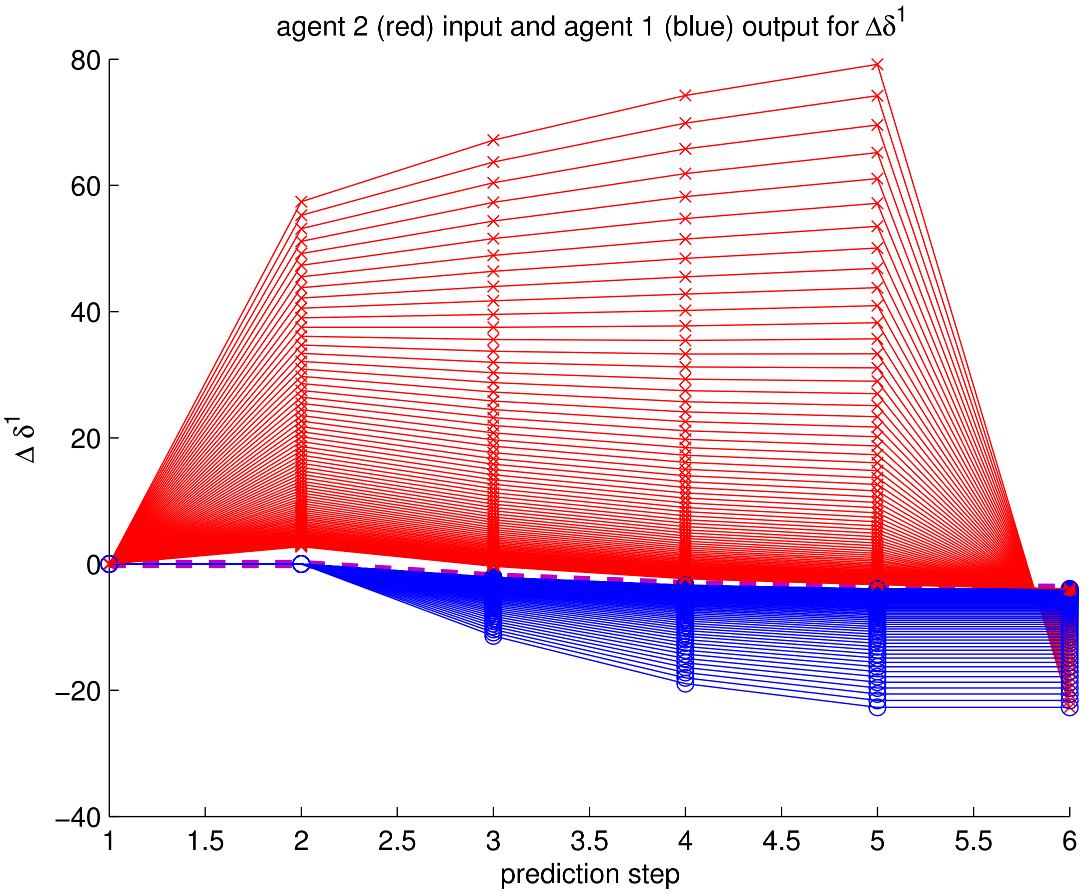
Example of serial run:



Iteration 30

4. Example: load-frequency control

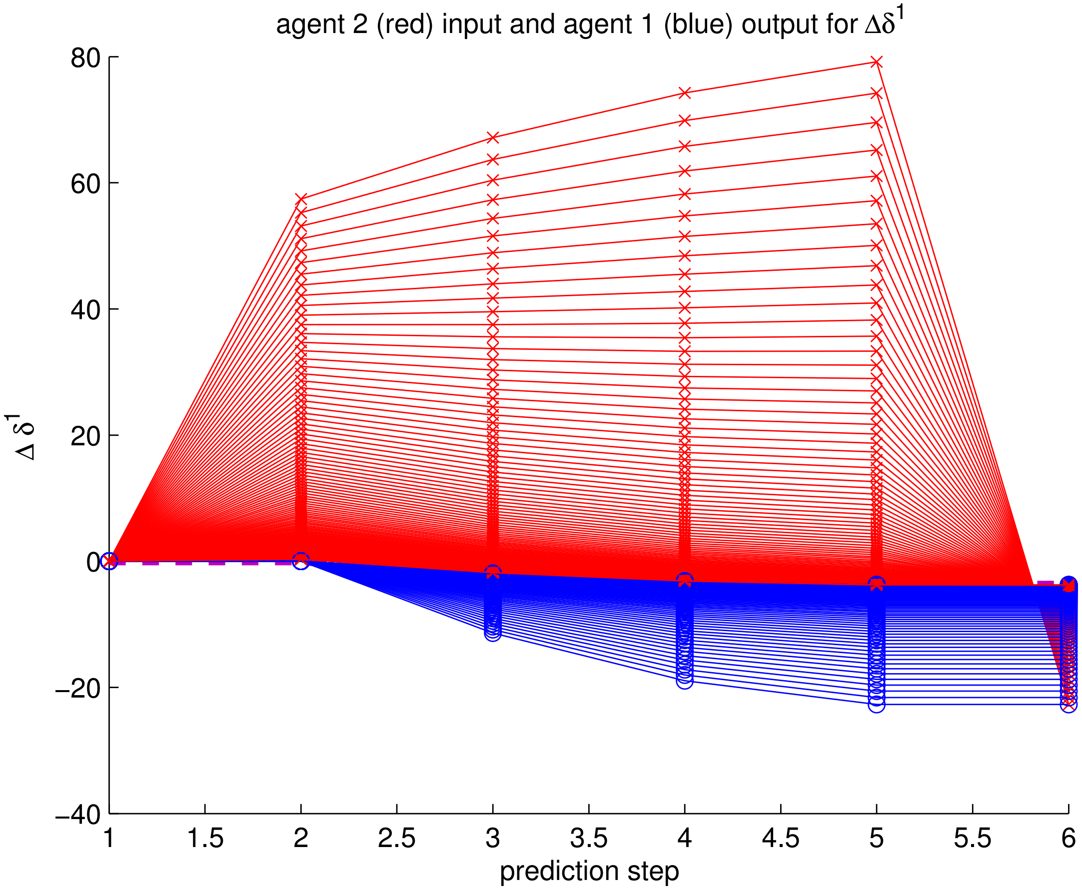
Example of serial run:



Iteration 80

4. Example: load-frequency control

Example of serial run:



Final

6. Future plans

- Alternatives to obtain agreement.
- More complex control architectures.
- Robustness and solution quality.
- Examples from other domains.