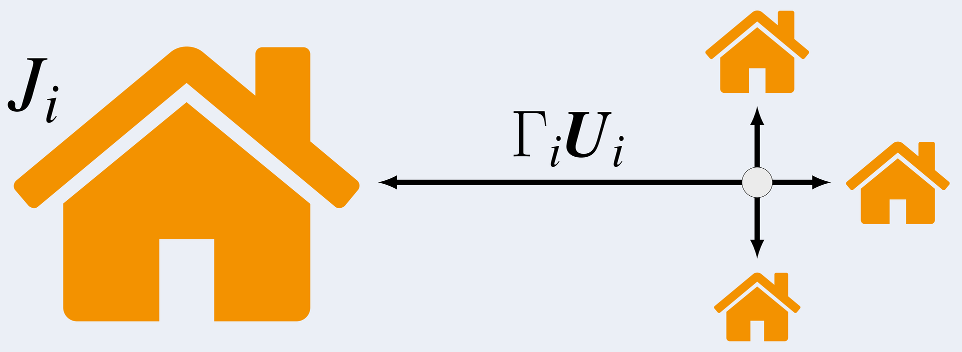
Expectation-Maximization Based Defense Mechanism for Distributed Model Predictive Control

Rafael Accácio Nogueira, Romain Bourdais, Simon Leglaive, Hervé Guéguen
IETR-CentraleSupélec, Rennes, France
rafael-accacio.nogueira@centralesupelec.fr

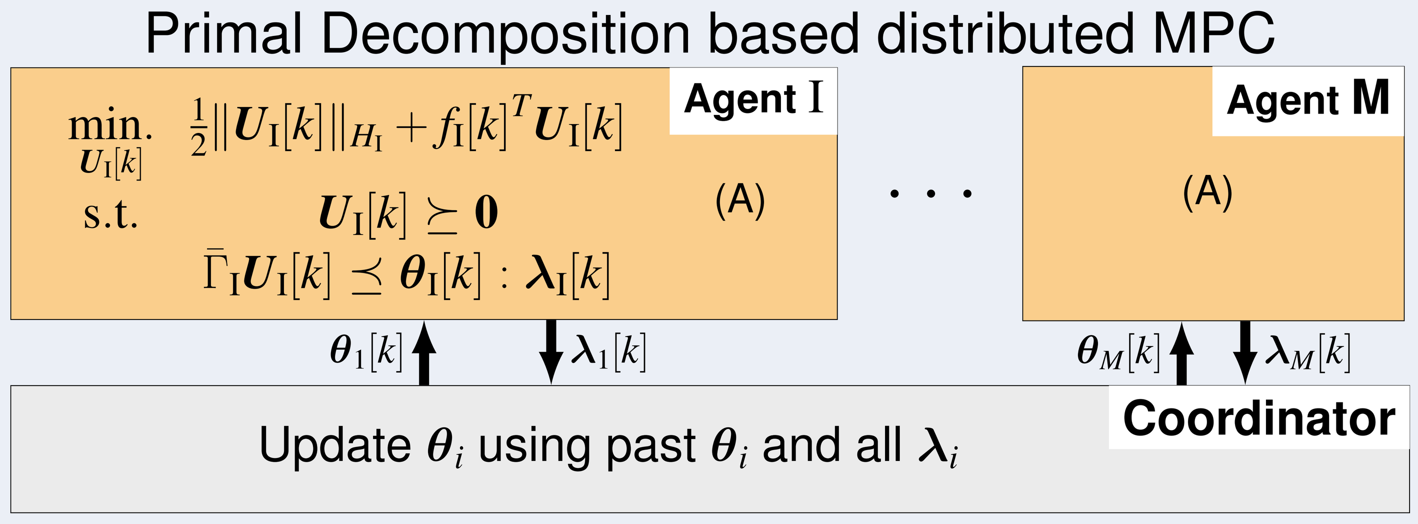
1. Challenge - False Data injection in dMPC exchange

- Decomposable quadratic objective $\sum_{i=1}^{M} J_i$
- Coupling constraint $\sum_{i=1}^{M} \Gamma_i \boldsymbol{U}_i[k] \preceq \boldsymbol{U}_{\max}$



Solution

Primal Decomposition based distributed MPC



Coordinator allocates $\boldsymbol{\theta}_i$
Agent has dissatisfaction $\boldsymbol{\lambda}_i$

What happens if an agent lies about $\boldsymbol{\lambda}_i$?

2. Attack and consequences

- $\boldsymbol{\lambda}_i$ is the dissatisfaction of i to allocation $\boldsymbol{\theta}_i$
- Attacker increases $\boldsymbol{\lambda}_i$ using function $\gamma(\cdot)$
- $\uparrow$ dissatisfaction == $\uparrow$ allocation

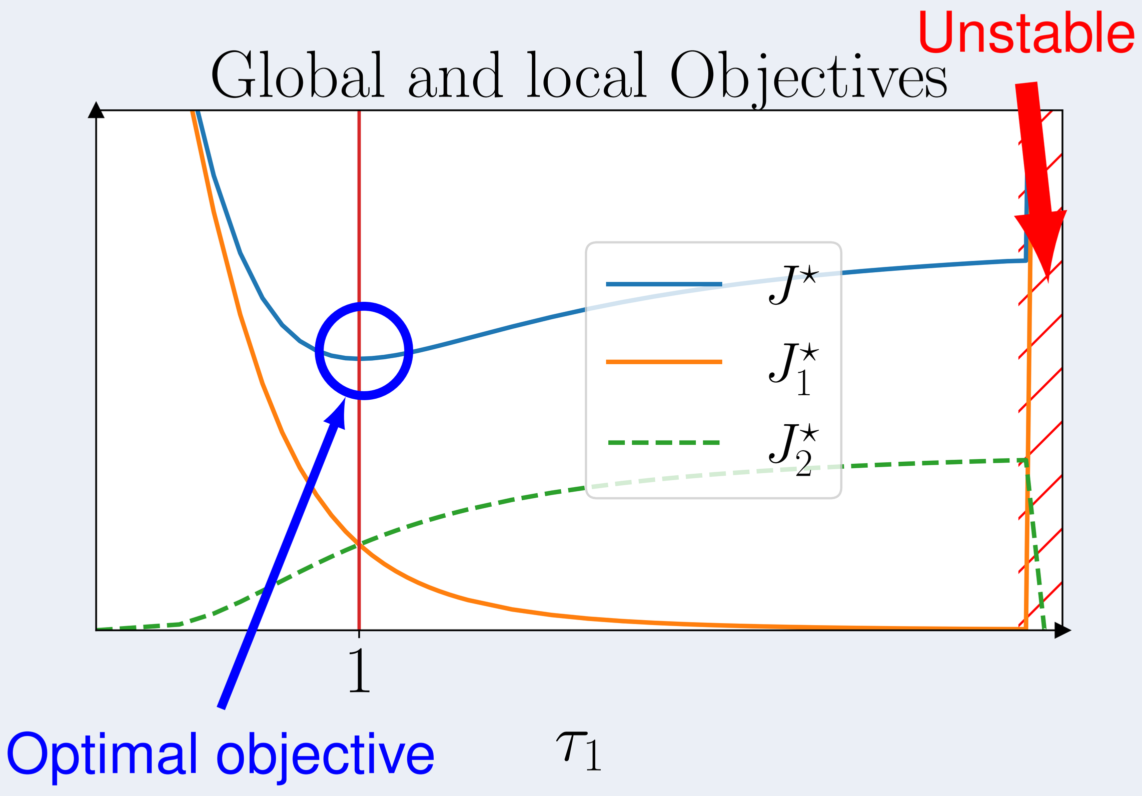
Remark
Attacker says it is satisfied **only when it is**

Assumption
Attacker chooses an *invertible linear function*

$$\tilde{\boldsymbol{\lambda}}_i = \gamma_i(\boldsymbol{\lambda}_i) = T_i[k]\boldsymbol{\lambda}_i,$$

- Effects of cheating matrix $T_i[k]$
  - Increase on global objective
  - Destabilization

Example $T_1[k] = \tau_1 I$



Can we mitigate the effects?

YES! If we estimate $T_i[k]$ and invert it
But how?

3. Estimating cheating matrix $T_i[k]$

Local problems (A) are **QP**
Explicit Solution with **PWA form w.r.t** $\boldsymbol{\theta}_i$:

$$\boldsymbol{\lambda}_i[k] = -P_i^n \boldsymbol{\theta}_i[k] - \boldsymbol{s}_i^n[k], \text{ if } G_i^n[k]\boldsymbol{\theta}_i[k] \preceq \boldsymbol{b}_i^n[k] \quad \text{(B)}$$

with $n \in \{1:N\}$. $G_i^n[k]$ and $\boldsymbol{b}_i^n[k]$ define regions.

Remark
Sensibilities P_i^n are **time invariant**.

Another assumption
In Region 1 **local constraints are active**:

$$\boldsymbol{\lambda}_i[k] = -P_i^1 \boldsymbol{\theta}_i[k] - \boldsymbol{s}_i^1[k], \text{ if } G_i^1[k]\boldsymbol{\theta}_i[k] \preceq \boldsymbol{b}_i^1[k] \quad \text{(C)}$$

and $\boldsymbol{\theta}_i = \mathbf{0}$ belongs to it

Attacker **modifies sensibility** $\tilde{P}_i[k] = T_i[k]\bar{P}_i$ and $\tilde{\boldsymbol{s}}_i[k] = T_i[k]\boldsymbol{s}_i[k]$

If we can know **nominal** $\bar{P}_i^1$,
by estimating $\tilde{P}_i[k]$, we can find $T_i[k]^{-1}$:

$$\widehat{T_i[k]^{-1}} = \bar{P}_i^1 \widehat{\tilde{P}_i^1[k]}^{-1} \quad \text{(D)}$$

But how can we estimate the $\tilde{P}_i^1[k]$?

Enter Expectation Maximization

- Classify data in regions (latent variables)
- Estimates parameters using weighted LS

EM needs minimally excited inputs $\boldsymbol{\theta}_i$ and $\tilde{\boldsymbol{\lambda}}_i$.

- ~~During negotiation~~ (time dependence)
- Solution: estimate in a separate phase
  - Generate independent points near $\boldsymbol{\theta}_i = \mathbf{0}$ Artificial Scarcity Sampling

4. Expectation Maximization

- Regions are indexed by $z \in \mathcal{Z} = \{1:Z\}$
- Gaussian mixture (mean (B) and $\Sigma \to 0$)
- Parameters $\mathcal{P} = \{\mathcal{P}^z \mid z \in \mathcal{Z}\}$, with $\mathcal{P}^z = (\tilde{P}^z, \tilde{\boldsymbol{s}}^z, \pi^z)$.
- Observations $o \in \mathcal{O} = \{1:O\}$ of $(\boldsymbol{\theta}_i, \boldsymbol{\lambda}_i)$ stacked as $(\underline{\Theta}, \underline{\Lambda})$ with corresponding $\underline{Z}$

Algorithm 1: Expectation Maximization
Initialize parameters $\mathcal{P}_{\text{new}}$
repeat
  $\mathcal{P}_{\text{cur}} \leftarrow \mathcal{P}_{\text{new}}$
  E step:
    Evaluate $\zeta_{zo}(\mathcal{P}_{\text{cur}}) = \mathbb{P}(\underline{z}_o = z \mid \underline{\boldsymbol{\lambda}}_o, \underline{\boldsymbol{\theta}}_o; \mathcal{P}_{\text{cur}})$
  M step:
    Reestimate parameters using:
    $\mathcal{P}_{\text{new}} = \arg\max_{\mathcal{P}} \mathbb{E}_{\zeta_{zo}(\mathcal{P}_{\text{cur}})} [\ln \mathbb{P}(\underline{\Theta}, \underline{\Lambda}, \underline{Z}; \mathcal{P})]$
until $\mathcal{P}_{\text{cur}}$ *converges*

5. Secure dMPC

Modified negotiation (some additional steps):

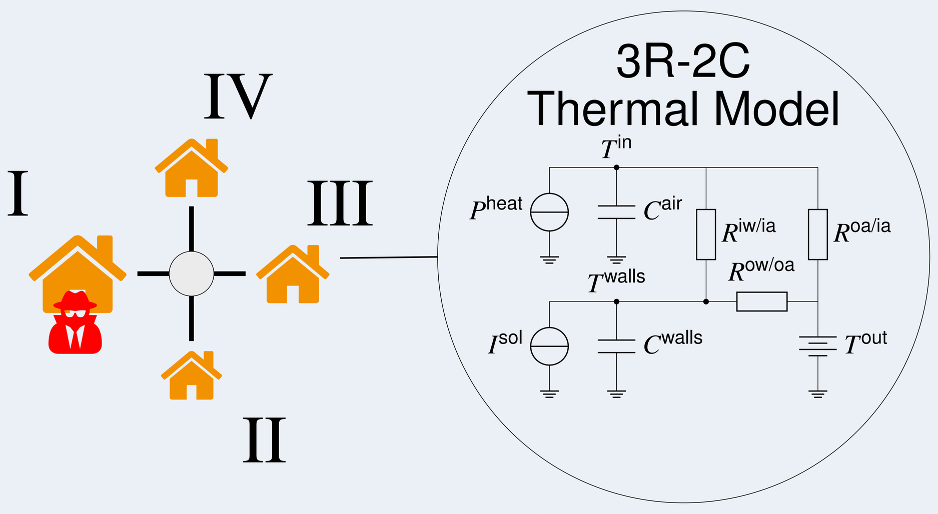
1. **Detection Phase**
   1.1 Estimate sensibility $\widehat{\tilde{P}_i^1[k]}$
     - Artificial Scarcity Sampling + EM

   1.2 Detect attack if $\|\widehat{\tilde{P}_i^1[k]} - \bar{P}_i^1\|_F \geq \epsilon_P$
2. **Negotiation Phase**
   2.1 If detected reconstruct $\boldsymbol{\lambda}_i$

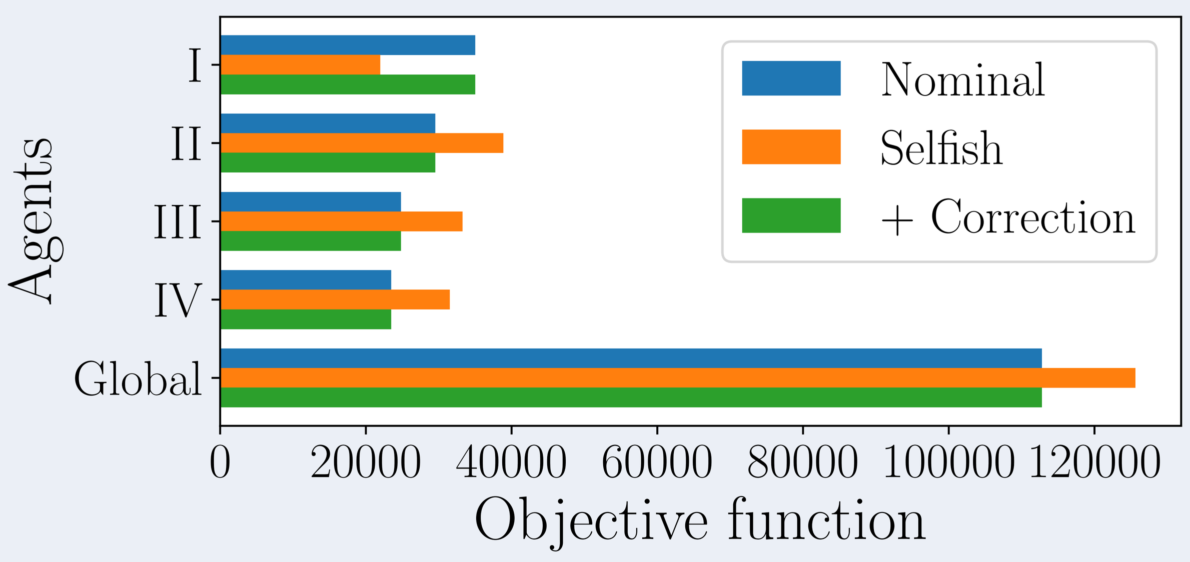
$$\boldsymbol{\lambda}_{i\text{rec}} = \widehat{T_i[k]^{-1}} \tilde{\boldsymbol{\lambda}}_i \quad \text{(E)}$$

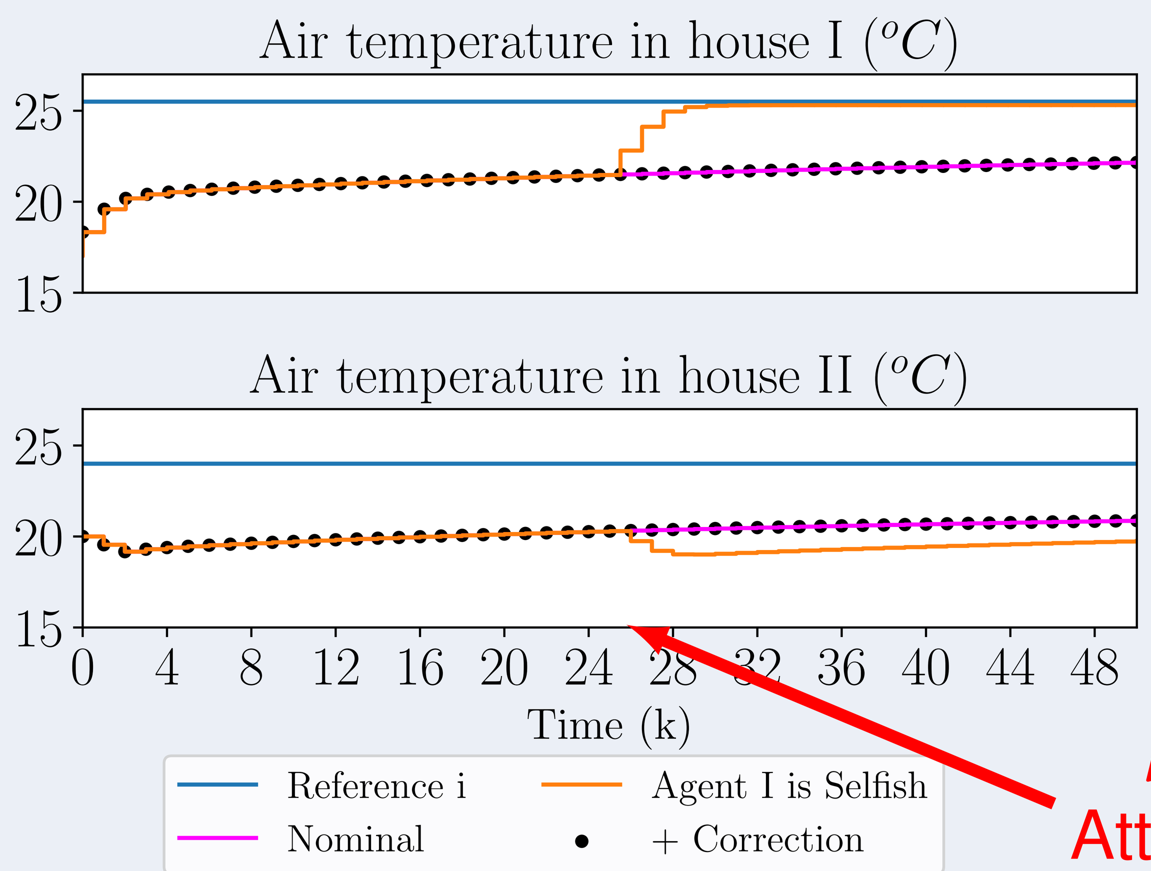
   2.2 Use adequate $\boldsymbol{\lambda}_i$ to update $\boldsymbol{\theta}_i$

6. Example: Control of a heating network under power scarcity - 3 Scenarios (Nominal, Selfish, + Correction)



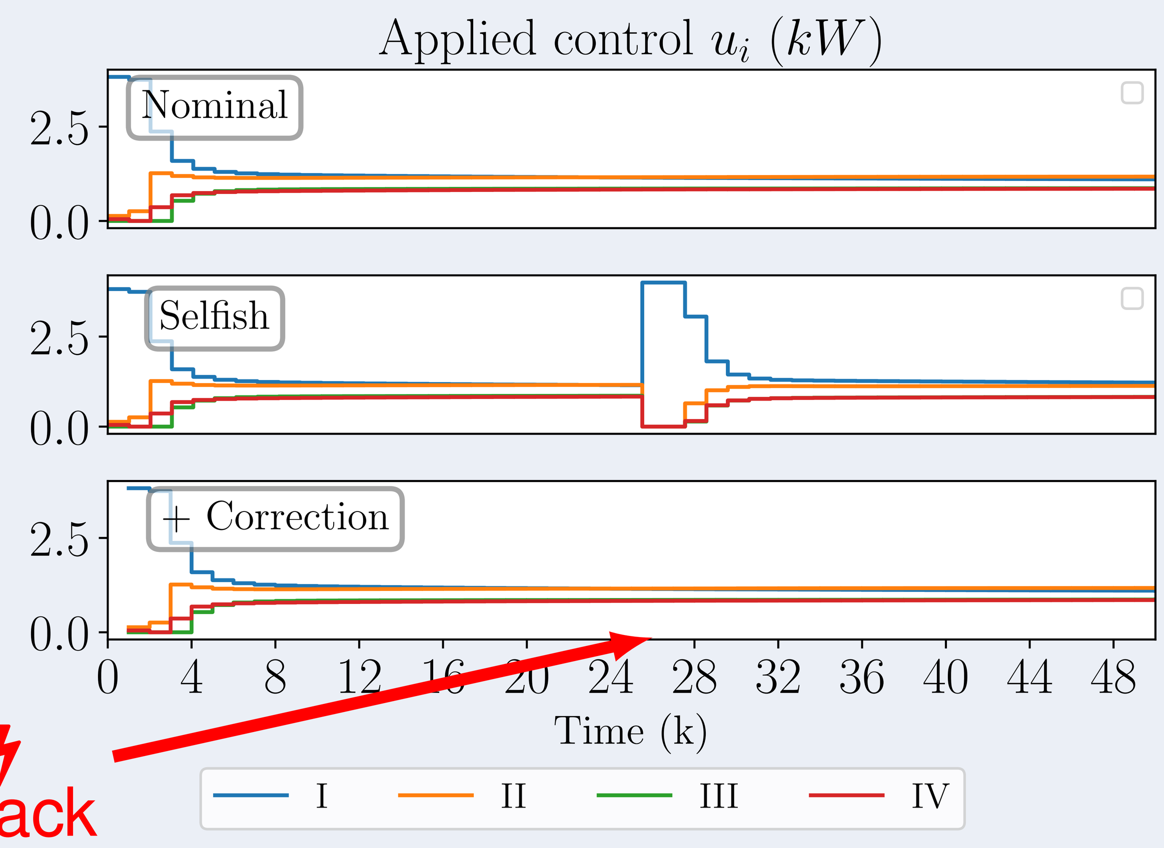
District with 4 houses





Air temperature in houses I and II.

Attack



Control applied in all houses for the 3 scenarios.

NecSys 22
Zurich, Switzerland
5-7 July 2022