Expectation-Maximization Based Defense Mechanism for Distributed Model Predictive Control

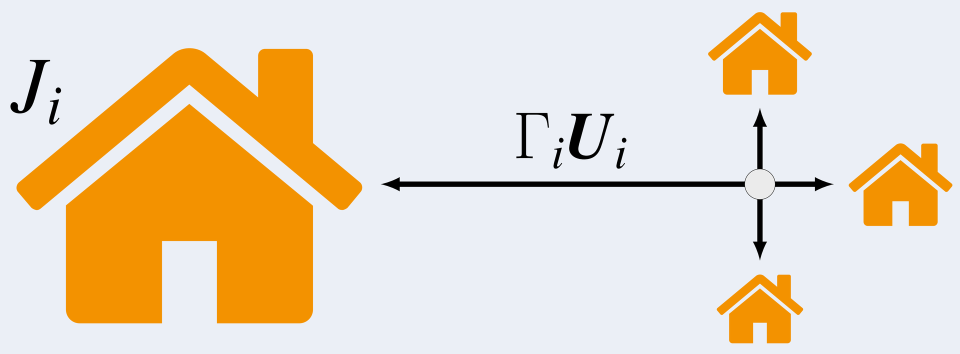
Rafael Accácio Nogueira, Romain Bourdais, Simon Leglaive, Hervé Guéguen

IETR-CentraleSupélec, Rennes, France

rafael-accacio.nogueira@centralesupelec.fr

1. Challenge - False Data injection in dMPC exchange

- Global decomposable quadratic objective $\sum_{i=1}^{M} J_i$
- Global coupling constraint $\sum_{i=1}^{M} \Gamma_i \boldsymbol{U}_i \leq \boldsymbol{U}_{\max}$



Agent 1

$$\begin{aligned} \min_{\boldsymbol{U}_1[k]} \quad & \tfrac{1}{2}\|\boldsymbol{U}_1[k]\|_{H_1} + f_1[k]^T \boldsymbol{U}_1[k] \\ \text{s.t.} \quad & \bar{\Gamma}_1 \boldsymbol{U}_1[k] \preceq \boldsymbol{\theta}_1[k] : \boldsymbol{\lambda}_1 \\ & \boldsymbol{U}_1[k] \succeq \boldsymbol{0} \end{aligned} \quad \text{(A)}$$

. . .

Agent M

(A)

$\boldsymbol{\theta}_1[k]$ ↑ ↓ $\tilde{\boldsymbol{\lambda}}_1[k]$ $\qquad$ $\boldsymbol{\theta}_M[k]$ ↑ ↓ $\tilde{\boldsymbol{\lambda}}_M[k]$

Coordinator

$$\boldsymbol{\theta}[k]^{(p+1)} = \mathrm{Proj}^{\mathcal{S}}(\boldsymbol{\theta}[k]^{(p)} + \rho^{(p)} \boldsymbol{\lambda}[k]^{(p)}) \quad \text{(B)}$$

What happens if an agent lies?

2. Attack and consequences

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

Assumption 1

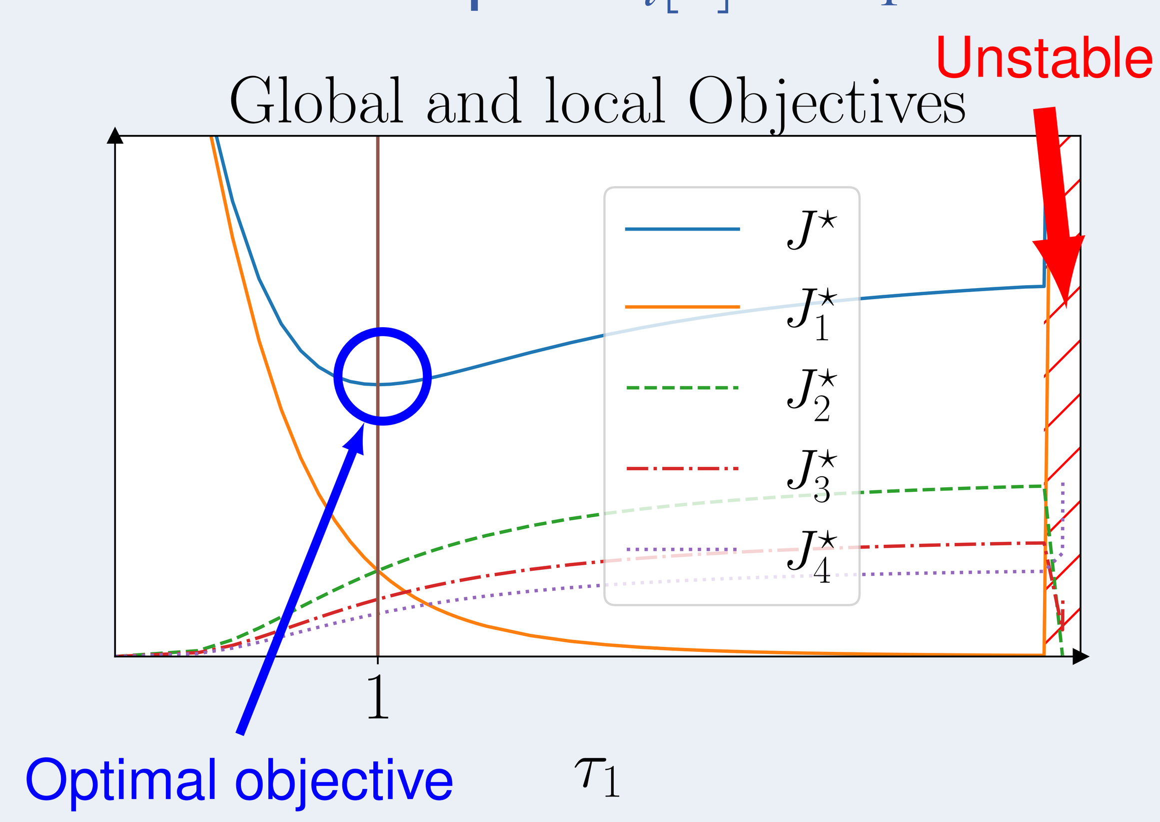
The attacker chooses a linear function

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

$\tilde{\boldsymbol{\lambda}}_i = \boldsymbol{0}$ *only if* $\boldsymbol{\lambda}_i = \boldsymbol{0} \rightarrow T_i[k]$ *is invertible.*

- Effects
  - Increase on global objective
  - Destabilization

Example $T_i[k] = \tau_1 I$



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** $\boldsymbol{\theta}_i$:

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

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

Remark 1

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

Assumption 2

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

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

Assumption 3

$\boldsymbol{\theta}_i = \boldsymbol{0}$ belongs to Region 1

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

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

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

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

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 $\boldsymbol{\theta}_i = \boldsymbol{0}$ Artificial Scarcity Sampling

4. Expectation Maximization

- Regions are indexed by $z \in \mathcal{Z} = \{1:Z\}$
- Gaussian mixture (mean (C) and $\Sigma \rightarrow 0$)
- 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 $(\boldsymbol{\theta}_i, \boldsymbol{\lambda}_i)$

Algorithm 1: Expectation Maximization

Initialize parameters $\mathcal{P}_{\text{new}}$

repeat

$\quad \mathcal{P}_{\text{cur}} \leftarrow \mathcal{P}_{\text{new}}$

$\quad$ **E step:**

$\quad\quad$ Evaluate $\zeta_{zo}(\mathcal{P}) = \mathbb{P}(\underline{z}_o = z | \underline{\boldsymbol{\lambda}}_o, \underline{\boldsymbol{\theta}}_o; \mathcal{P})$

$\quad$ **M step:**

$\quad\quad$ 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 to a local maximum*

5. Secure dMPC

Modified negotiation (some additional steps):

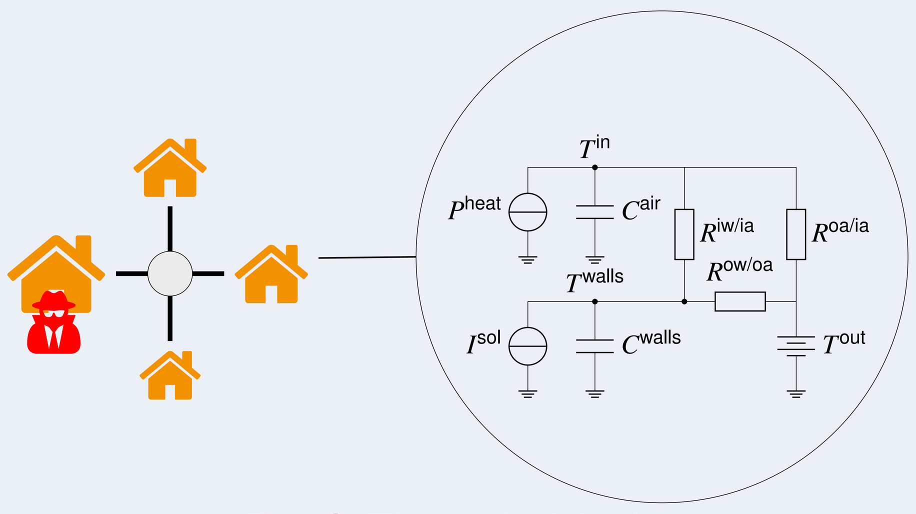
1. **Detection Phase**
   1.1 Estimate sensibility $\widehat{\tilde{P}_i^1}[k]$
   - Artificial Scarcity Sampling + EM

   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 $\boldsymbol{\lambda}_i$

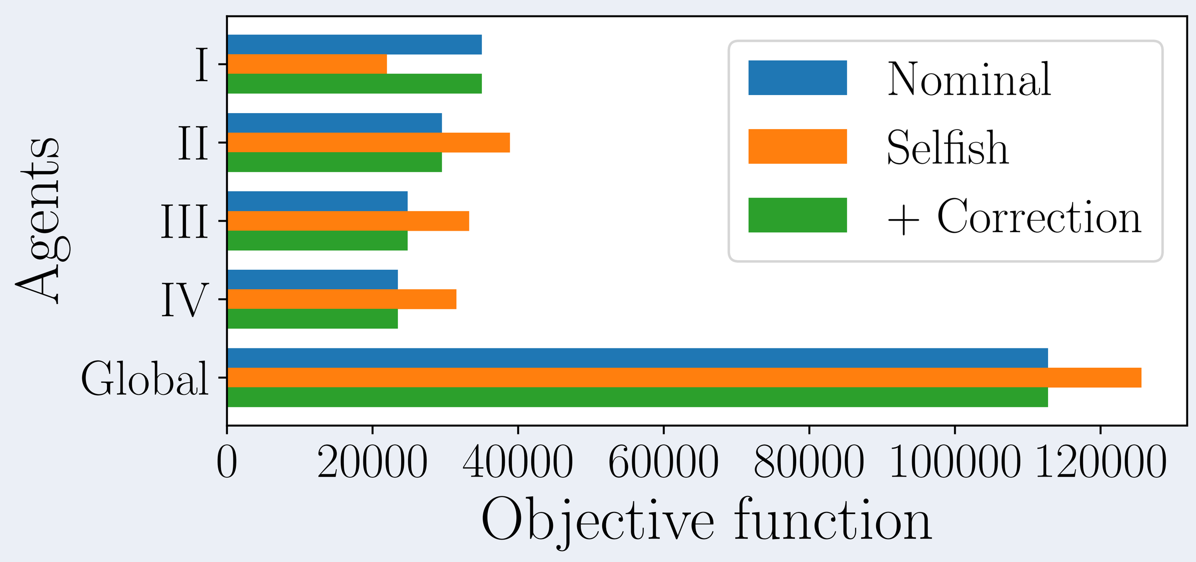
$$\boldsymbol{\lambda}_{i\text{rec}} = \widehat{T_i[k]^{-1}} \tilde{\boldsymbol{\lambda}}_i \quad \text{(F)}$$

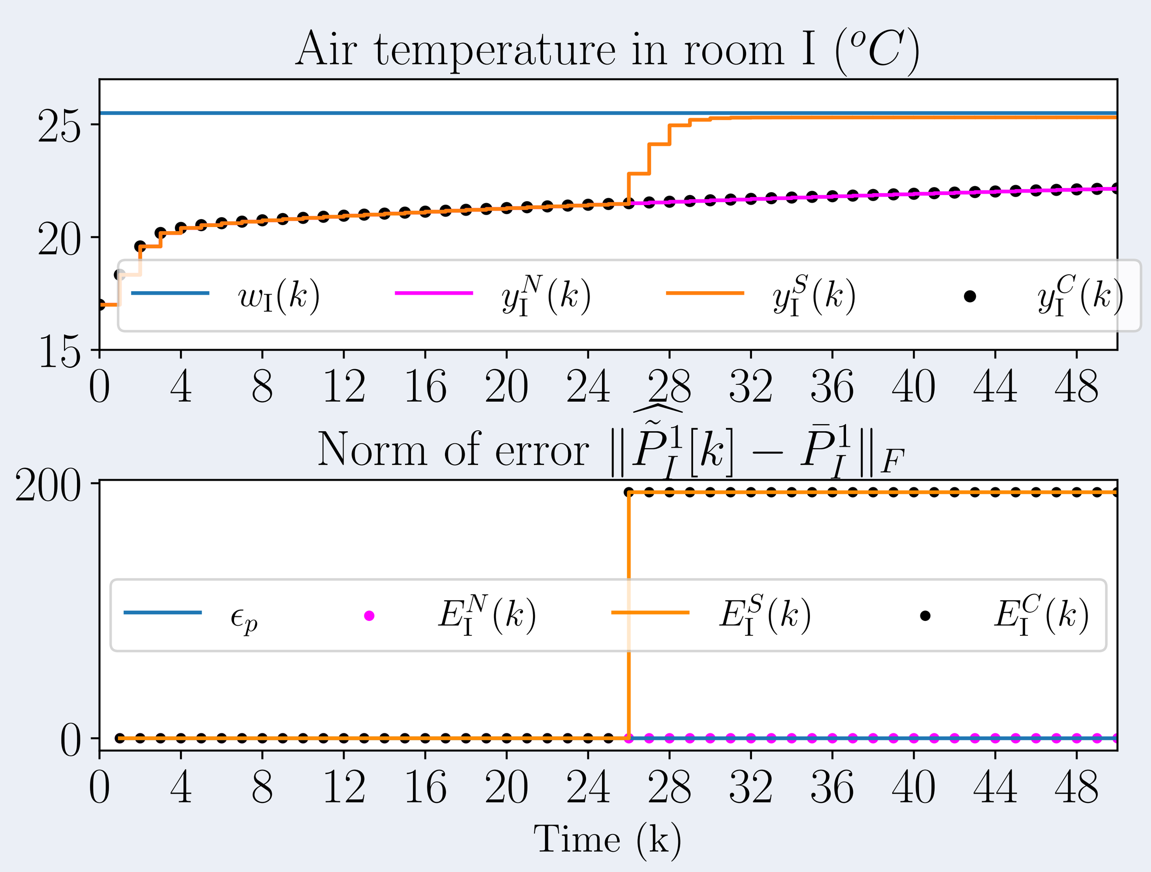
   2.2 Use adequate λ_i to update θ_i (B)

6. Example | 4 distinct rooms | 3 scenarios (Nominal, Selfish, Selfish + Correction)

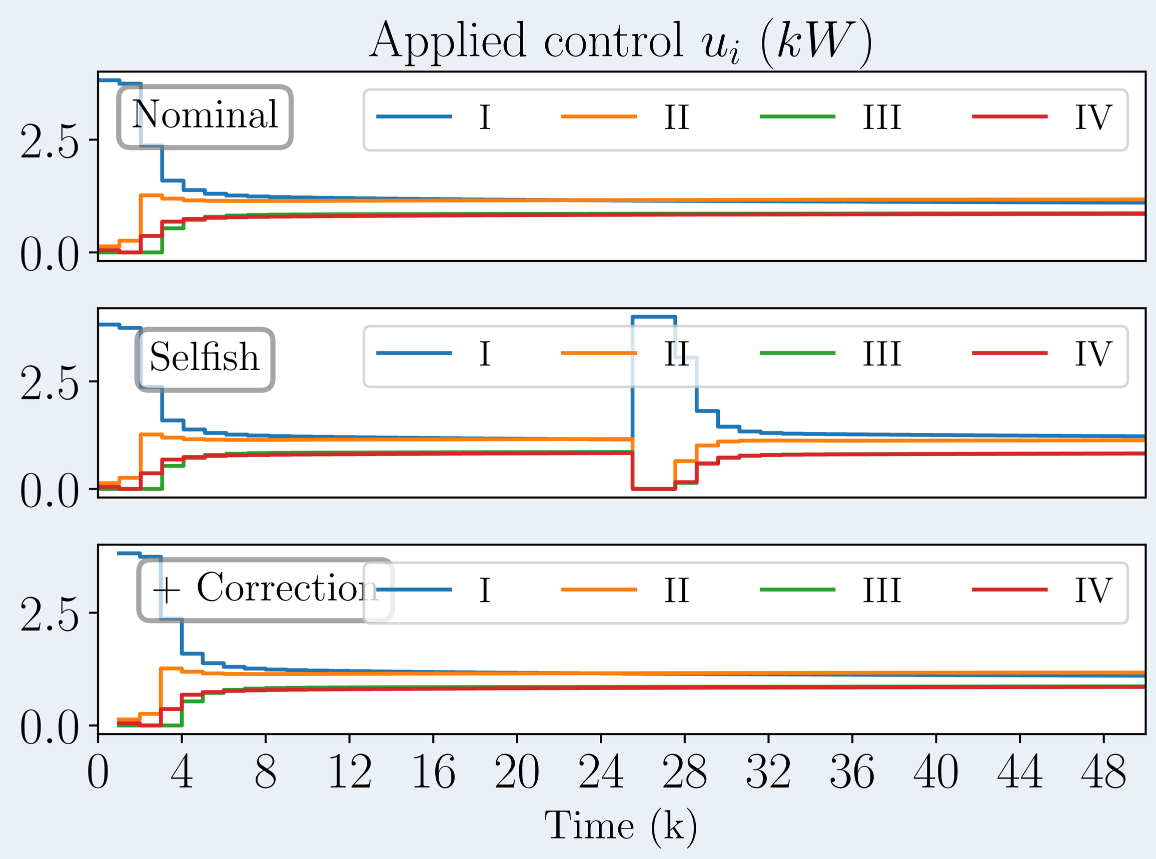


3R-2C Thermal Model.





Air temperature in room I and the decision variable $E_I[k]$ for three scenarios: nominal (N), selfish behavior (S), and selfish behavior with correction (C).



Control applied in all rooms for the 3 scenarios.

NecSys 22

Zurich, Switzerland

5-7 July 2022