Expectation-Maximization Based Defense Mechanism for Distributed Model Predictive Control

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

IETR-CentraleSupélec

35510 Cesson-Sévigné, Ille-et-Vilaine, France

{rafael-accacio.nogueira, romain.bourdais, simon.leglaive, herve.gueguen}@centralesupelec.fr

1. Context - False Data injection in dMPC exchange

- Cyber-Physical Systems
- Large Scale

MPC

- Linear Model
- Linear Control Objective (e.g. $(\boldsymbol{w}[k]-\boldsymbol{y}[k])\to 0$)
- Linear Input Constraints

→

MPC
Quadratic Program (QP)

$$\begin{aligned}\underset{\boldsymbol{U}_i[k]}{\text{minimize}}\quad & \tfrac{1}{2}\|\boldsymbol{U}[k]\|_H + f[k]^T\boldsymbol{U}[k]\\ \text{subject to}\quad & \bar{\Gamma}\boldsymbol{U}[k]\preceq \boldsymbol{U}_{\max}\\ & \boldsymbol{U}[k]\succeq \boldsymbol{0}\end{aligned}$$

HARD TO COMPUTE

→

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

$\cdots$

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

$$\boldsymbol{\theta}[k]^{(p+1)} = \mathrm{Proj}^{\mathcal{S}}(\boldsymbol{\theta}[k]^{(p)} + \rho^{(p)}\boldsymbol{\lambda}[k]^{(p)}) \qquad (2)$$

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

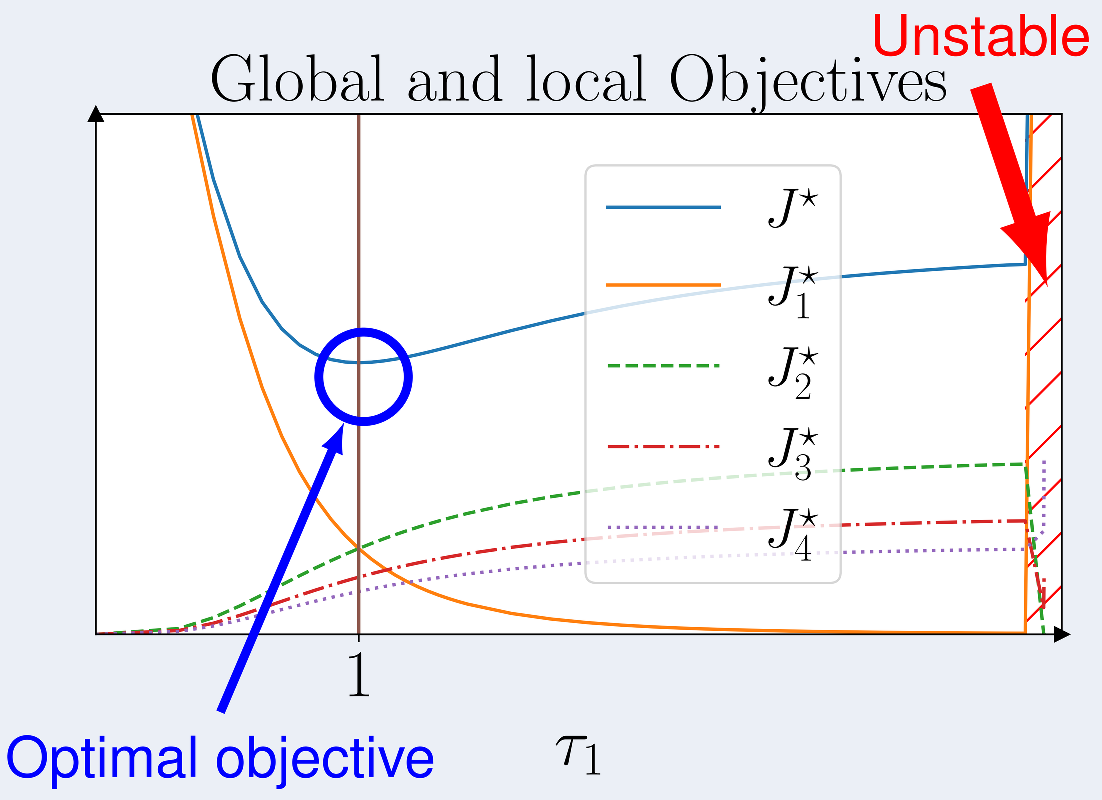
The attacker chooses a linear function

$$\tilde{\boldsymbol{\lambda}}_i = \gamma_i(\boldsymbol{\lambda}_i) = T_i[k]\boldsymbol{\lambda}_i, \qquad (3)$$

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

- Effects
  - Increase on global objective
  - Destabilization

Example $T_i[k] = \tau_1 I$



Can we mitigate the effects?

What if we estimate $T_i[k]$ and invert it?

Problem: How to estimate it?

3. Estimating cheating matrix $T_i[k]$

Local pbs. (1) are **QP** → **Explicit Solution PWA form w.r.t $\boldsymbol{\theta}_i$**:

$$\boldsymbol{\lambda}_i[k] = -P_i^n\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i^n[k], \text{ if } G_i^n[k]\boldsymbol{\theta}_i[k]\preceq \boldsymbol{b}_i^n[k] \quad (4)$$

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**:*

$$\boldsymbol{\lambda}_i[k] = -P_i^1\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i^1[k], \textit{ if } G_i^1[k]\boldsymbol{\theta}_i[k]\preceq \boldsymbol{b}_i^1[k] \quad (5)$$

Assumption 3

$\theta_i = \boldsymbol{0}$ *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 $T_i[k]^{-1}$:

$$\widehat{T_i[k]^{-1}} = \bar{P}_i^1\widehat{\tilde{P}_i^1[k]}^{-1} \qquad (6)$$

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

Enter Expectation Maximization

- Classify data in regions
- Estimates parameters using weights

EM estimates $\tilde{P}_i^1[k]$ if we provide minimally excited inputs $\boldsymbol{\theta}_i$ and $\tilde{\boldsymbol{\lambda}}_i$.

- Minimally excited inputs
  - ~~Estimation negotiation~~ (time dependence)
  - Solution: 2 phases Detection and Negotiation

4. Expectation Maximization

- Each zone has a different $z\in\mathcal{Z}=\{1:Z\}$
- Gaussian mixture (mean (4) and $\Sigma\to 0$)
- Parameters $\mathcal{P}=\{\mathcal{P}^z \mid z\in\mathcal{Z}\}$, with $\mathcal{P}^z=(\tilde{\boldsymbol{P}}^z,\tilde{\boldsymbol{s}}^z,\pi^z)$.
- Generate O observations close to $\boldsymbol{0}$

Algorithm 1: Expectation Maximization

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

repeat

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

E step:

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

M step:

Reestimate parameters using:

$$\mathcal{P}_{\text{new}} = \arg\max_{\mathcal{P}}.\ \mathbb{E}_{\zeta_{zo}(\mathcal{P}_{\text{cur}})}\left[\ln\mathbb{P}(\underline{\Theta},\underline{\Lambda},\underline{Z};\mathcal{P})\right]$$

until $\mathcal{P}_{\text{cur}}$ *converges to 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 $\boldsymbol{\lambda}_i$

$$\boldsymbol{\lambda}_{i\text{rec}} = \widehat{T_i[k]^{-1}}\tilde{\boldsymbol{\lambda}}_i \qquad (7)$$

   2.2 Use adequate λ_i to update θ_i (2)

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

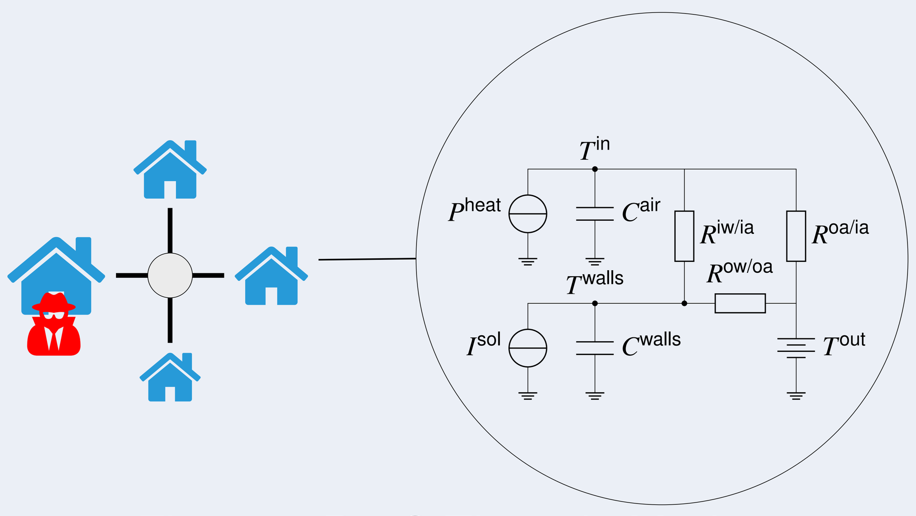


Figure: 3R-2C Thermic Model.

Table: Objective functions J_i (% error)

Agent	Nominal	Selfish	+ Correction
I	35008.7 (0.0)	21969.6 (−40.0)	35008.7 (−0.0)
II	29495.3 (0.0)	38867.4 (30.0)	29495.4 (0.0)
III	24808.7 (0.0)	33266.4 (30.0)	24808.7 (0.0)
IV	23457.8 (0.0)	31511.0 (30.0)	23457.8 (0.0)
Global	112770.6 (0.0)	125614.4 (10.0)	112770.6 (−0.0)

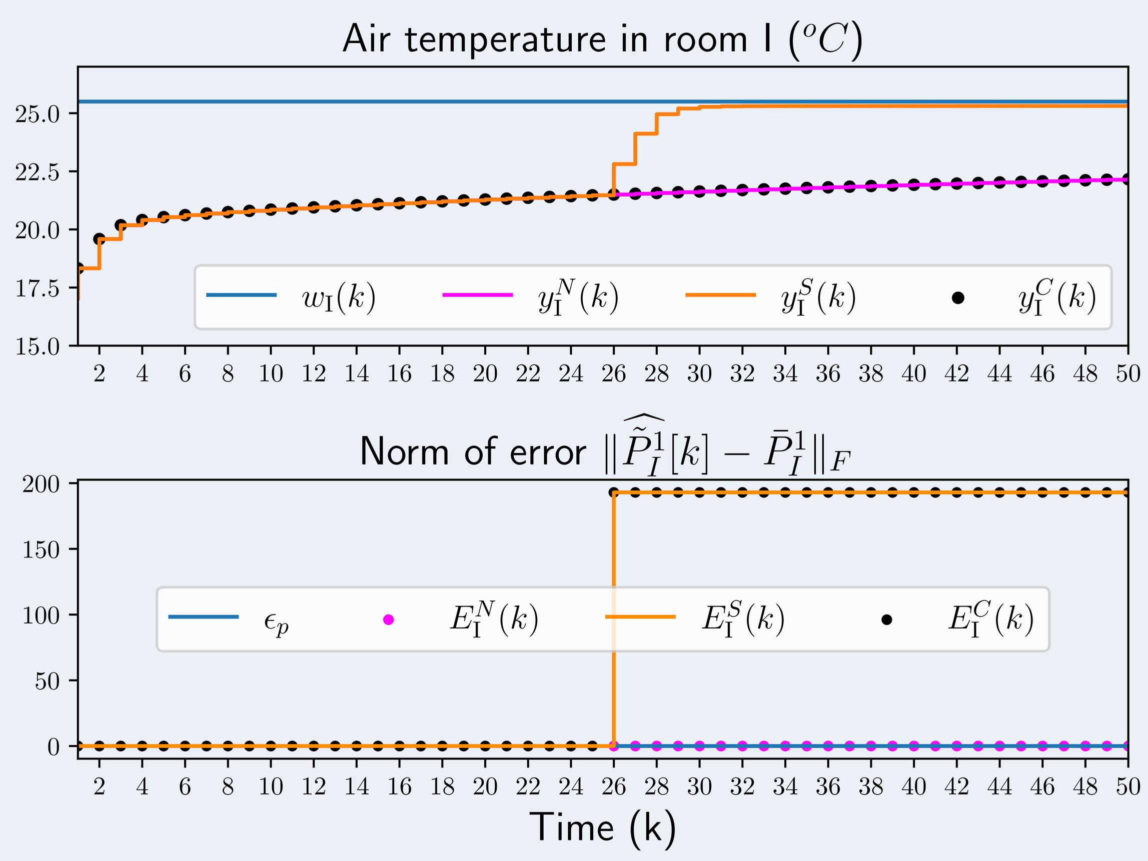


Figure: Air temperature in room I and the decision variable $E_I[k]$ for three scenarios: nominal (N), selfish behavior (S), and selfish behavior with correction (C).

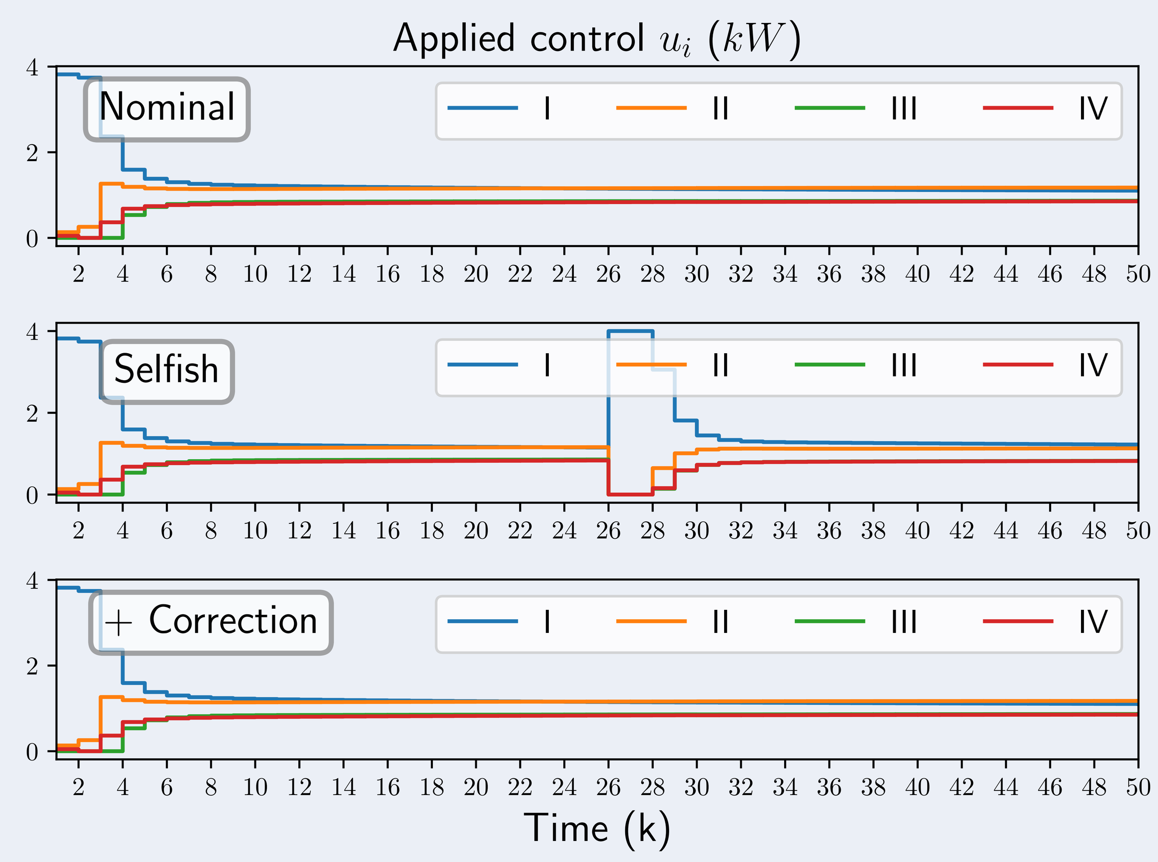


Figure: Control applied in all rooms for the 3 scenarios.

NecSys 22
Zurich, Switzerland
5-7 July 2022