



HAL
open science

Expectation-Maximization Based Defense Mechanism for Distributed Model Predictive Control

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

► **To cite this version:**

Rafael Accacio Nogueira, Romain Bourdais, Simon Leglaive, Hervé Guéguen. Expectation-Maximization Based Defense Mechanism for Distributed Model Predictive Control. 9th IFAC Conference on Networked Systems (NecSys22), Jul 2022, Zürich, Switzerland. hal-04628841

HAL Id: hal-04628841

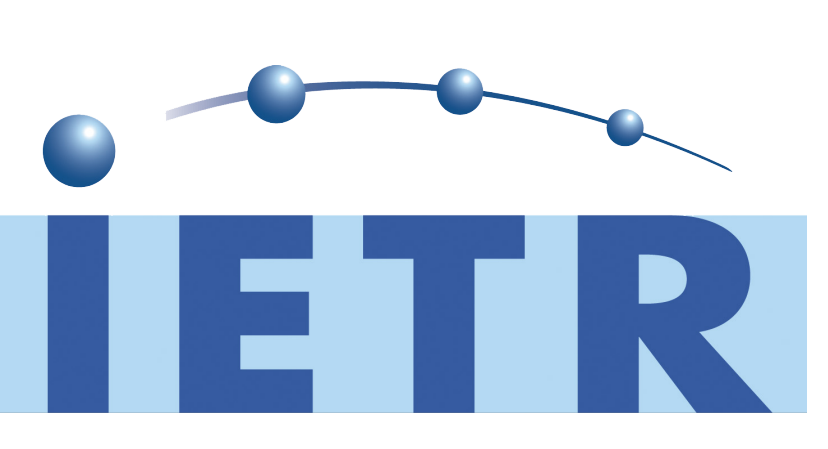
<https://hal.science/hal-04628841>

Submitted on 28 Jun 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Expectation-Maximization Based Defense Mechanism for Distributed Model Predictive Control

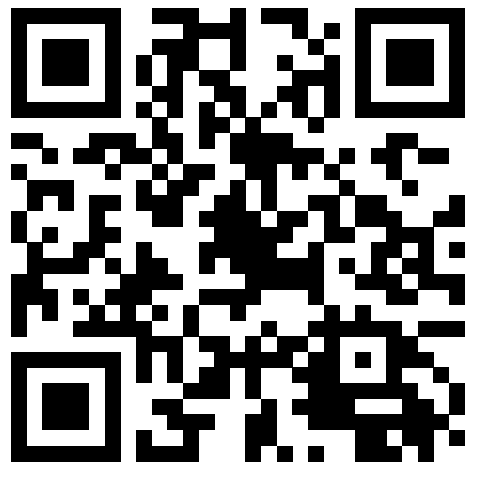


Rafael Accácio Nogueira, Romain Bourdais, Simon Leglaive, Hervé Guéguen

IETR-CentraleSupélec, Rennes, France

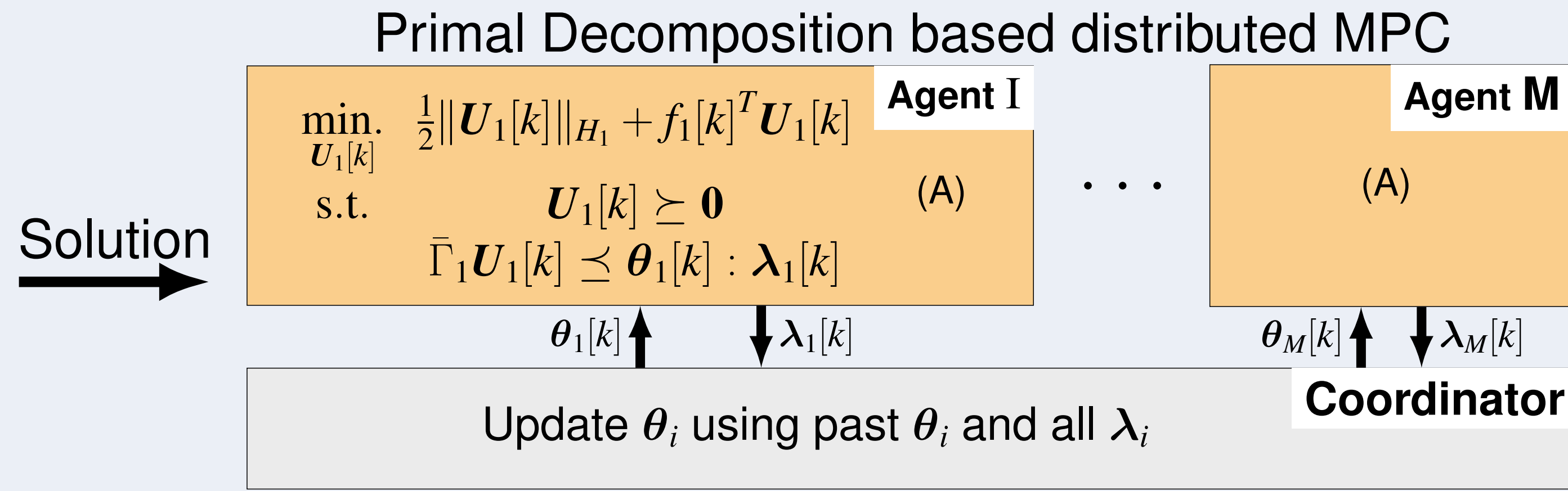
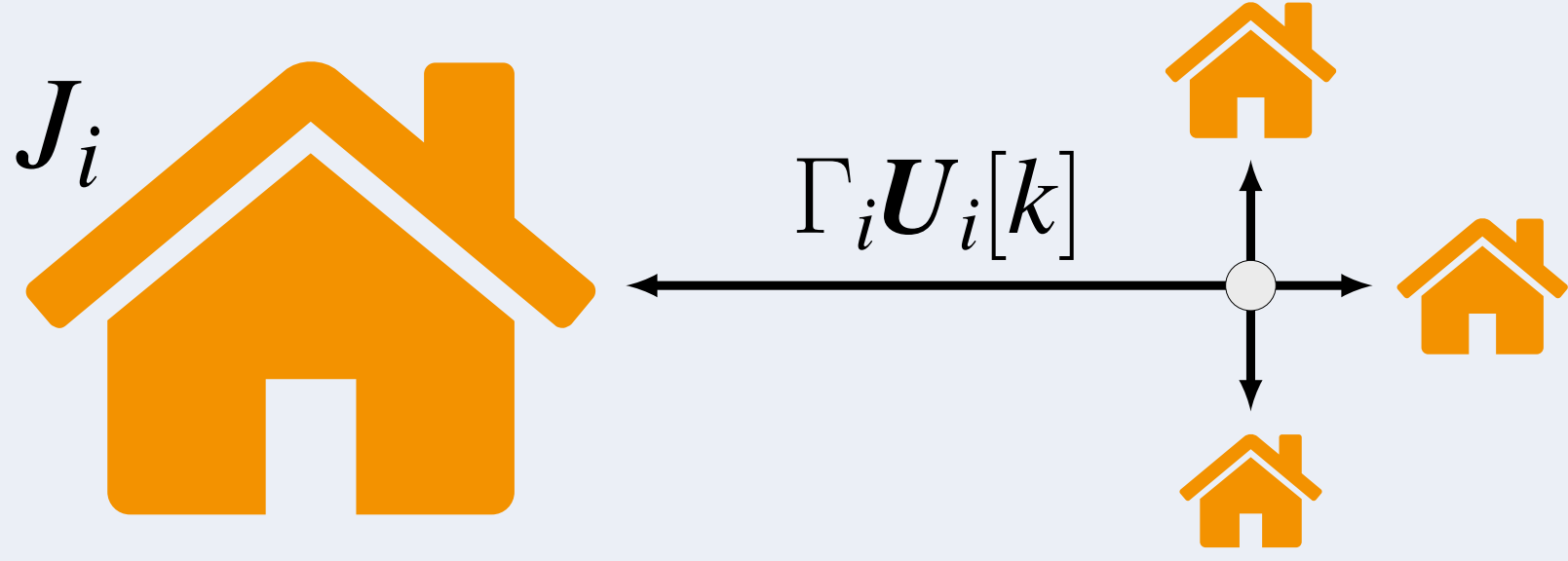
rafael-accacio.nogueira

@centralesupelec.fr



1. Challenge - False Data injection in dMPC exchange

- ▶ Decomposable quadratic objective $\sum_{i=1}^M J_i$
- ▶ Coupling constraint $\sum_{i=1}^M \Gamma_i U_i[k] \leq U_{\max}$



Coordinator allocates θ_i
Agent has dissatisfaction λ_i

What happens if an agent lies about λ_i ?



2. Attack and consequences

- ▶ λ_i is the dissatisfaction of i to allocation θ_i
- ▶ Attacker increases λ_i using function $\gamma(\cdot)$
- ▶ \uparrow dissatisfaction == \uparrow allocation

Remark

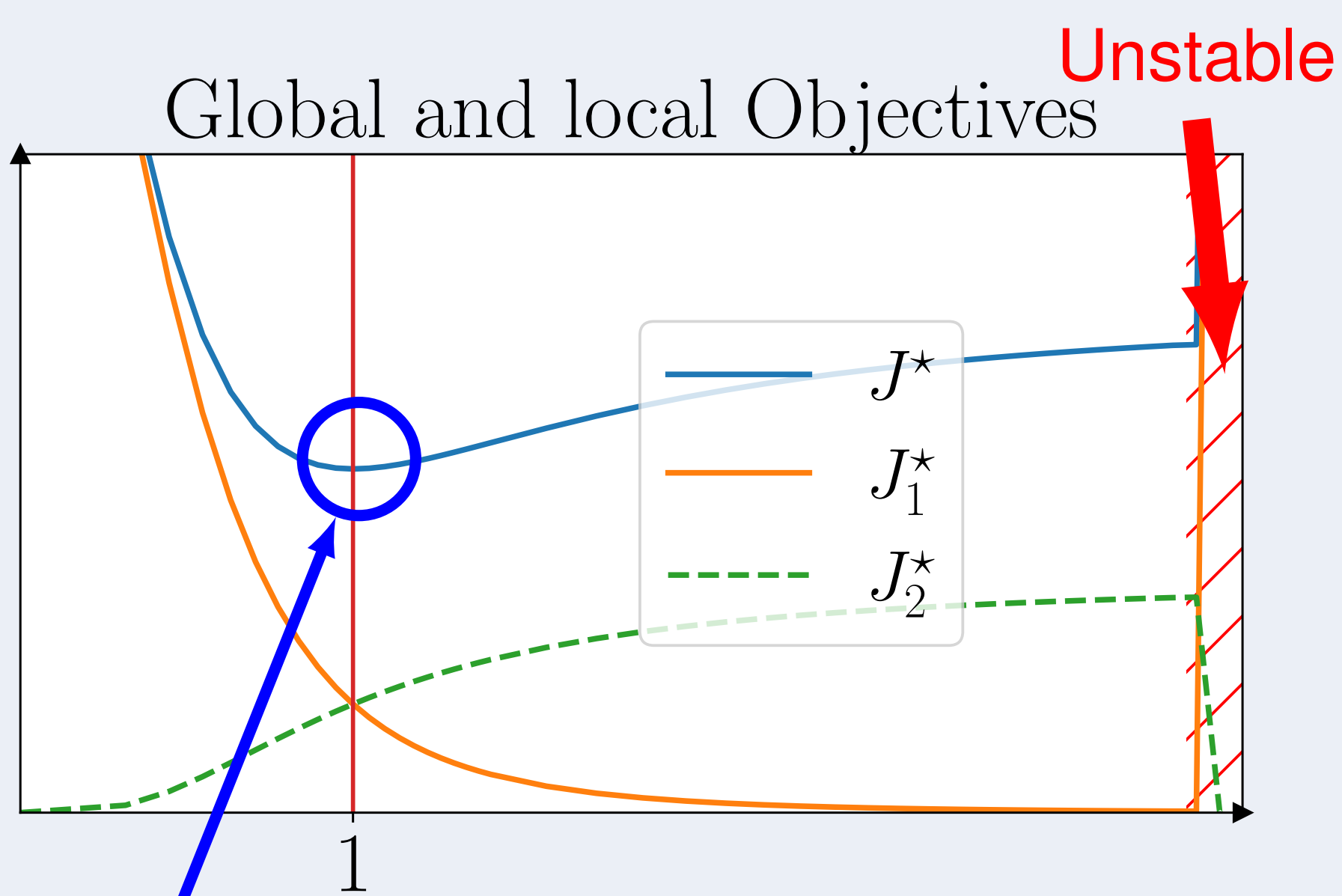
Attacker says it is satisfied only when it is Assumption

Attacker chooses an *invertible* linear function

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

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

Example $T_1[k] = \tau_1 I$



Optimal objective τ_1

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 θ_i :

$$\lambda_i[k] = -P_i^n \theta_i[k] - s_i^n[k], \text{ if } G_i^n[k] \theta_i[k] \leq b_i^n[k] \quad (B)$$

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

Remark

Sensibilities P_i^n are time invariant.

Another assumption

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

$$\lambda_i[k] = -P_i^1 \theta_i[k] - s_i^1[k], \text{ if } G_i^1[k] \theta_i[k] \leq b_i^1[k] \quad (C)$$

and $\theta_i = \mathbf{0}$ belongs to it

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

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

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

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 θ_i and $\tilde{\lambda}_i$.

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

4. Expectation Maximization

- ▶ Regions are indexed by $z \in \mathcal{Z} = \{1 : Z\}$
- ▶ Gaussian mixture (mean (B) and $\Sigma \rightarrow O$)
- ▶ Parameters $\mathcal{P} = \{\mathcal{P}^z \mid z \in \mathcal{Z}\}$, with $\mathcal{P}^z = (\tilde{P}^z, \tilde{s}^z, \pi^z)$.
- ▶ Observations $o \in \mathcal{O} = \{1 : O\}$ of (θ_i, λ_i) stacked as (Θ, Λ) with corresponding \underline{Z}

Algorithm 1: Expectation Maximization

Initialize parameters \mathcal{P}_{new}

repeat

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

E step:

Evaluate $\zeta_{z_o}(\mathcal{P}_{\text{cur}}) = \mathbb{P}(z_o = z \mid \Lambda_o, \Theta_o; \mathcal{P}_{\text{cur}})$

M step:

Reestimate parameters using:

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

until \mathcal{P}_{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 $\|\widehat{\tilde{P}_i^1[k]} - \tilde{P}_i^1\|_F \geq \epsilon_P$

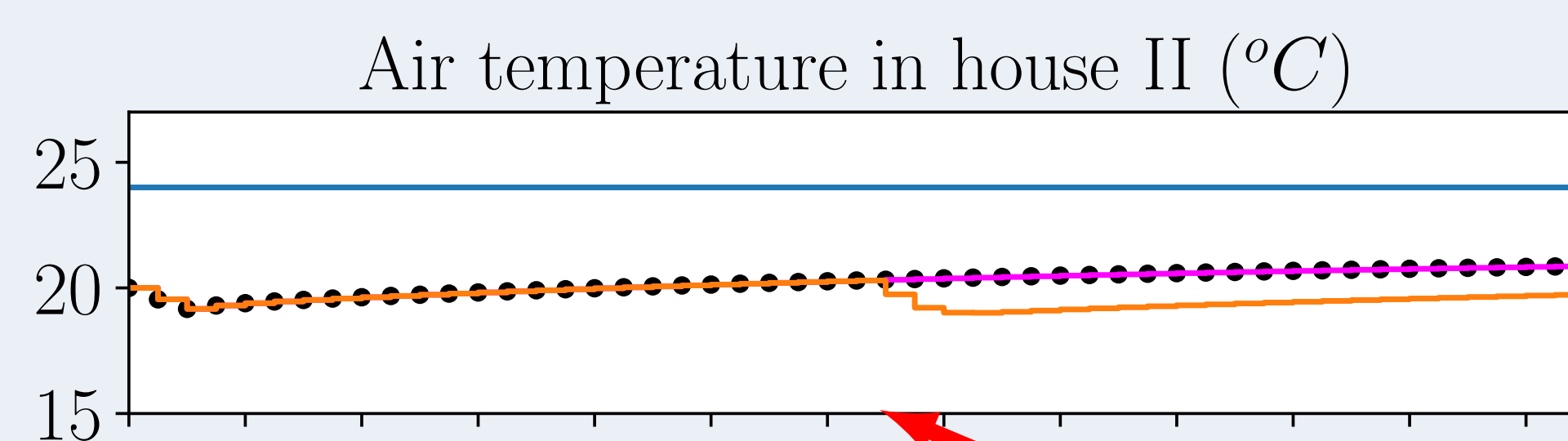
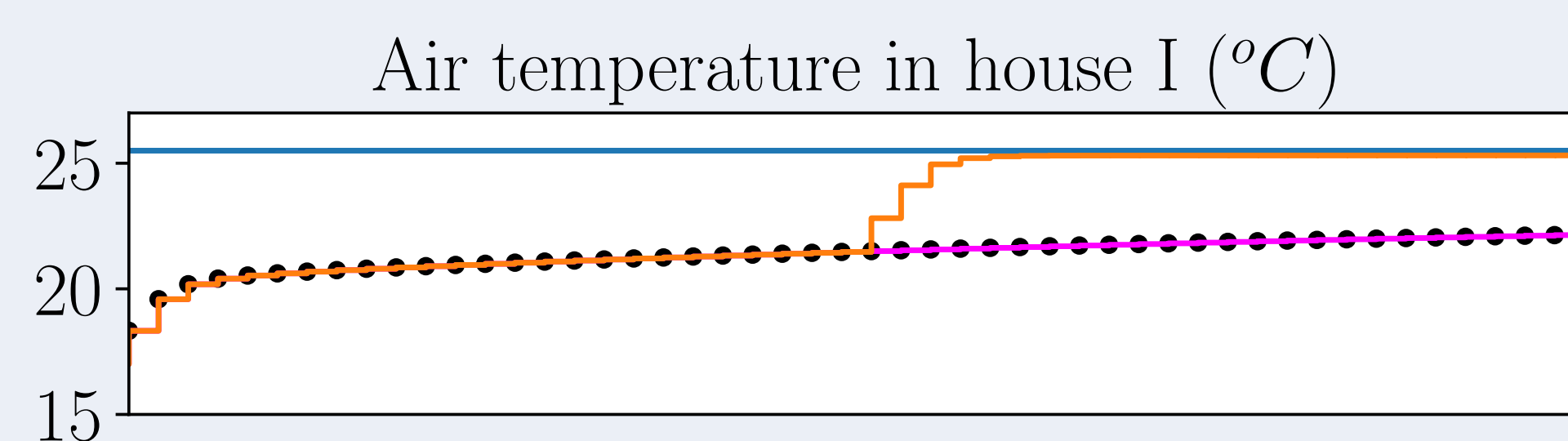
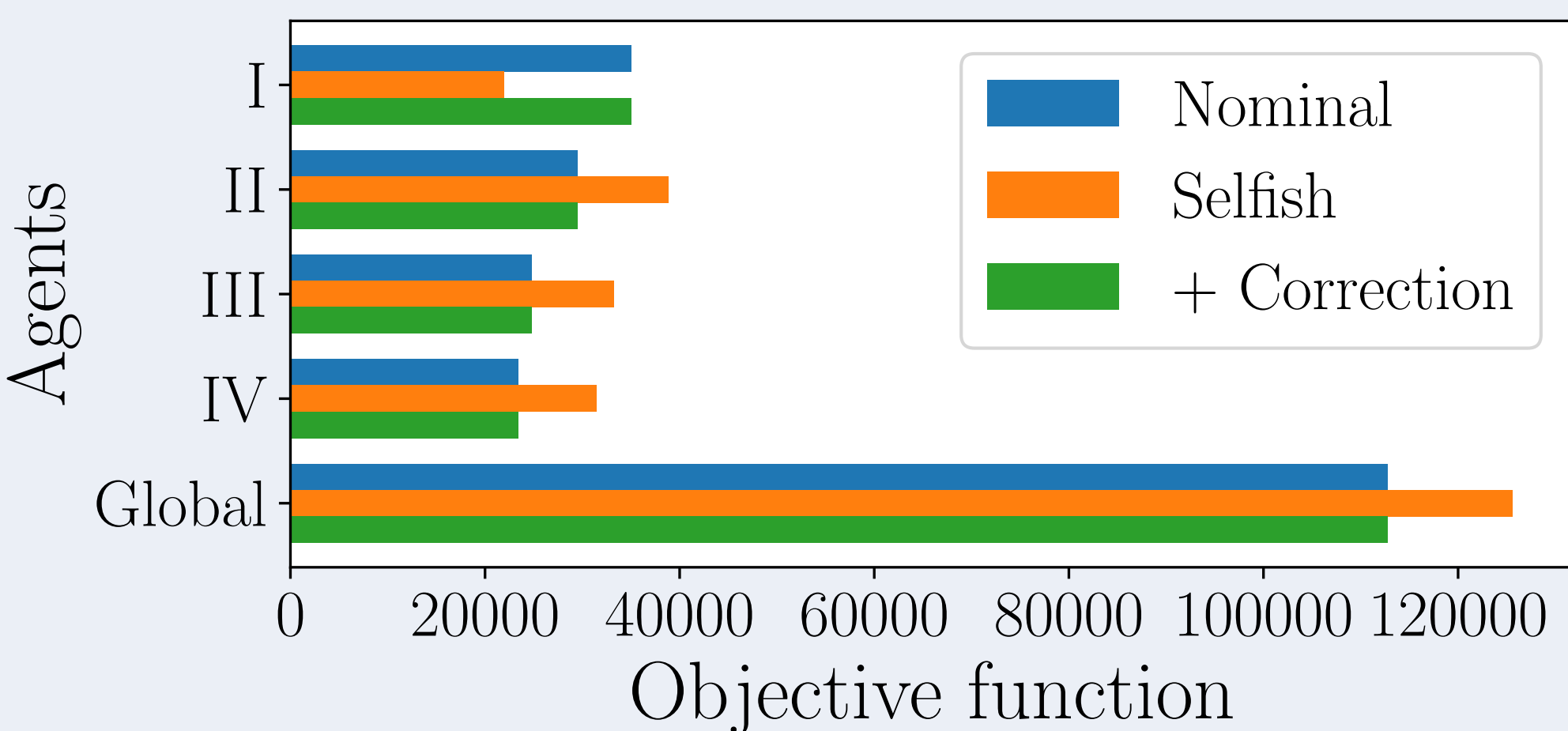
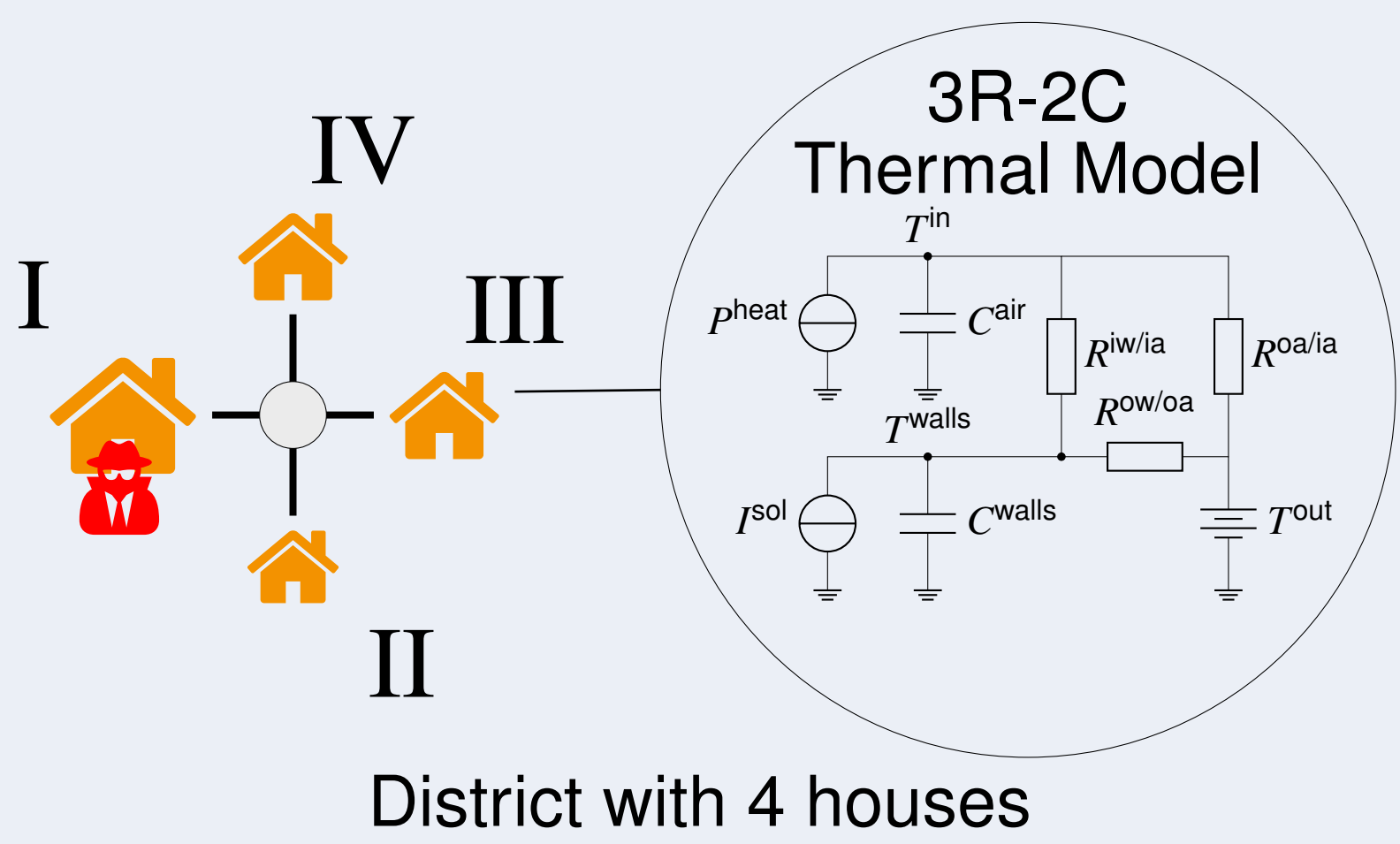
2. Negotiation Phase

- 2.1 If detected reconstruct λ_i

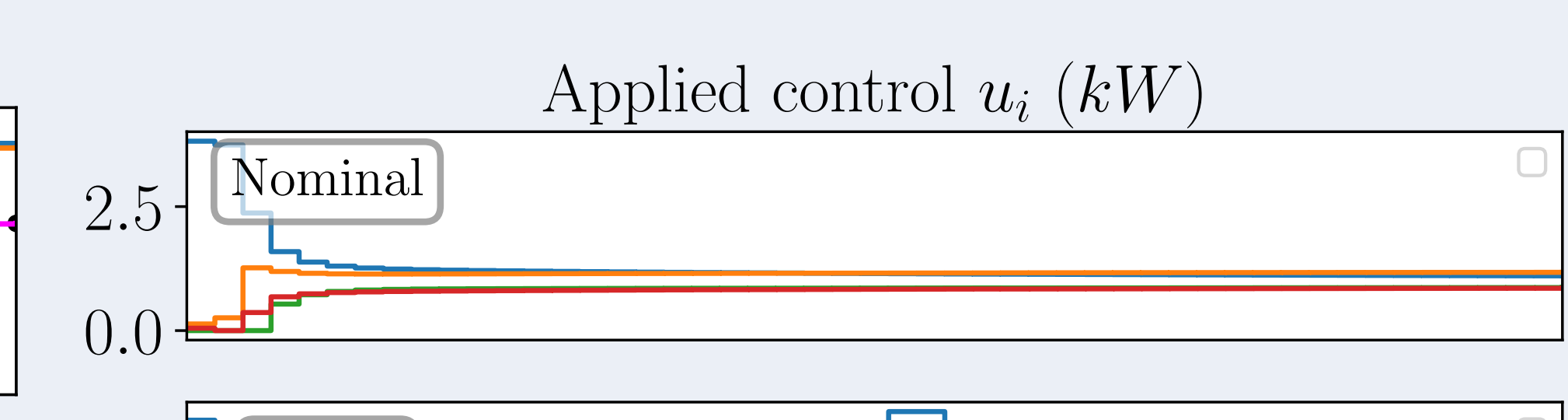
$$\lambda_{i\text{rec}} = \widehat{T_i[k]^{-1}} \tilde{\lambda}_i \quad (E)$$

- 2.2 Use adequate λ_i to update θ_i

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



Air temperature in houses I and II.



Control applied in all houses for the 3 scenarios.