Expectation-Maximization Based Defense Mechanism for Distributed Model Predictive Control

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

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

Abstract: Controlling large-scale systems sometimes requires decentralized computation. Communication among agents is crucial to achieving consensus and optimal global behavior. These negotiation mechanisms are sensitive to attacks on those exchanges. This paper proposes an algorithm based on Expectation Maximization to mitigate the effects of attacks in a resource allocation based distributed model predictive control. The performance is assessed through an academic example of the temperature control of multiple rooms under input power constraints.

Keywords: Distributed constrained control and MPC; Cyber security in networked control systems; Decentralized Control and Large-Scale Systems

1. INTRODUCTION

The information collected, processed, and transmitted within cyber-physical systems (CPS) impacts physical systems whose proper functioning is essential for society, like energy networks, industrial processes, water distribution, and others. Attacks in those critical infrastructures, such as Stuxnet, increased awareness for cyber-security of CPS.

Given these structures' complexity, scale, and geographic distribution, it is necessary to adopt decentralized control structures rather than monolithic ones. For these systems, a set of controllers are designed to collectively ensure the safe and efficient operation of the facilities.

This operation is done cooperatively, where the controllers exchange information. This approach can be used, i.e., for the voltage regulation of electrical networks, the production-consumption balance in these same networks, water distribution, and many other applications. Multi-agent systems (Kantamneni et al., 2015) and distributed Model-based Predictive Control (dMPC) (Maestre et al., 2014) are standard approaches to design these controllers.

A few articles have focused on safety in dMPC frameworks, presenting vulnerabilities of different frameworks, such as Lagrange-based decomposition (Velarde et al., 2017), Jacobi-Gauss decomposition (Chanfreut et al., 2018), and primal decomposition (Nogueira et al., 2021).

The authors in these works highlight the complexity of the problems. A modification of a local goal, the hacking of communication, or the failure of one of the agents are events that disrupt global behavior. If one of these agents is attacked, it can result in the violent destruction of the corrupted element (perhaps the whole system), or more subtly, it can lead to a deviant behavior that is more difficult to detect.

Some strategies are presented to mitigate the effects of such an event. For instance, dismissing extreme values as in Velarde et al. (2017) or using secure scenarios based on reliable historical data Maestre et al. (2021).

Nogueira et al. (2021) proposed a safe algorithm based on data reconstruction to mitigate the effects of a false data injection (FDI) in the communication between a coordinator and local constraint-free agents, which were bound by global equality constraints.

In this paper, we propose an extension of their approach by including local constraints and changing the global constraints. This extension profoundly changes the complexity, as the exchanges between the agents and the coordinator are no longer characterized by affine functions but by Piecewise Affine (PWA) functions. This fact leads not only to a combinatorial explosion but also to a parametric identification challenge. To overcome this problem, we propose using a learning method based on the Expectation Maximization (EM) algorithm (Dempster et al., 1977), which allows us to estimate the corruption mechanism and correct it if necessary.

This paper is organized as follows. First, Section 2 introduces the primal decomposition-based dMPC and its vulnerabilities. Then in Section 3, we study the local problems and propose a detection and mitigation mechanism based on the Expectation Maximization algorithm. Moreover, in Section 4, we illustrate the algorithm with an academic simulation and assess its performance. Finally, in Section 5, we conclude with an outlook of future works.

Notation: In this paper, $\|\cdot\|$, $\|\cdot\|_F$, and $\|\boldsymbol{v}\|_Y = \|Y^{\frac{1}{2}}\boldsymbol{v}\|$ represent the ℓ_2, Frobenius and weighted norms. $\text{Proj}^{\mathcal{T}}(\cdot)$ is the Euclidean projection onto $\mathcal{T}$. $\otimes$ ($\odot$) is the Kronecker (Hadamard) product. $\text{vec}(A)$ vectorizes matrix A. $n:i:j$ is a row vector builder with elements $\{n, n+i, \ldots, n+mi\}$, where $m = \text{truncate}(\frac{j-n}{i})$, and $n:j$ means $i=1$. $\mathbf{0}_m = 0_{m\times 1}$ ($\mathbf{1}_m = 1_{m\times 1}$), where $0_{m\times n}$ ($1_{m\times n}$) is a 0 (1) filled m-by-n matrix. $\mathbb{1}_{\{x\}}$ is the indicator function returning 1 if x is true and 0 otherwise. $\mathbb{S}^n$, and $\mathbb{S}^n_{++}$ ($\mathbb{S}^n_{+}$) are symmetric and positive (semi-)definite matrices of size n. $A^\dagger = (A^TA)^{-1}A^T$. A vector $\boldsymbol{v}_i$, corresponds to the i-th agent, and these vectors can be stacked in a vector $\boldsymbol{v}$. $\mathbb{E}[\underline{x}]$ is the expected value of random variable $\underline{x}$, and $\mathbb{P}(A \mid B)$ is the conditional probability of A given condition B. $\text{diag}(A_1, \ldots, A_N)$ corresponds to a block diagonal matrix.

2. PROBLEM STATEMENT

In this section, we present a dMPC algorithm based on the primal decomposition (Boyd et al., 2015) and a vulnerability of the decomposition, which can be exploited.

2.1 Model Predictive Control

Consider a system composed of discrete-time linear time-invariant agents $i \in \mathcal{M} = \{1:M\}$, modeled by

$$\begin{aligned} \boldsymbol{x}_i[k+1] &= A_i\boldsymbol{x}_i[k] + B_i\boldsymbol{u}_i[k] \\ \boldsymbol{y}_i[k+1] &= C_i\boldsymbol{x}_i[k] \end{aligned}, \tag{1}$$

with $\boldsymbol{x}_i[k] \in R^{n_x}$, $\boldsymbol{u}_i[k] \in R^{n_u}$ and $\boldsymbol{y}_i[k] \in R^{n_y}$.

Each agent's input $\boldsymbol{u}_i[k]$ is constrained by

$$\mathbf{0}_{n_u} \preceq \boldsymbol{u}_i[k] \preceq \boldsymbol{u}_{\max}, \tag{2}$$

with $\boldsymbol{u}_{\max} \in R^{n_u}$. The agents are coupled by global input constraints with weighting matrices $\Gamma_i \in \mathbb{S}^{n_u}_+$:

$$\sum_{i\in\mathcal{M}} \Gamma_i\boldsymbol{u}_i[k] \preceq \boldsymbol{u}_{\max}. \tag{3}$$

The overall system is controlled by a Model-based Predictive Control (MPC), which computes the optimal input for a finite prediction horizon $\mathcal{H} = \{1:N_p\}$ by solving the following problem:

$$\begin{aligned} &\overbrace{\phantom{\sum_{i\in\mathcal{M}}\sum_{j\in\mathcal{H}}\|\boldsymbol{v}_i[k+j|k]\|^2_{Q_i} + \|\boldsymbol{u}_i[k+j-1|k]\|^2_{R_i}}}^{J[k]} \\ &\underset{\boldsymbol{u}[k:k+N_p-1|k]}{\text{minimize}} \quad \sum_{i\in\mathcal{M}}\sum_{j\in\mathcal{H}} \overbrace{\|\boldsymbol{v}_i[k+j|k]\|^2_{Q_i} + \|\boldsymbol{u}_i[k+j-1|k]\|^2_{R_i}}^{J_i[k]} \\ &\text{subject to} \quad (1), (2) \text{ and } (3) \quad \left.\begin{matrix}\forall i \in \mathcal{M} \\ \forall j \in \mathcal{H}\end{matrix}\right\}, \end{aligned} \tag{4}$$

where $Q_i \in \mathbb{S}_+$, $R_i \in \mathbb{S}_{++}$, and $\boldsymbol{v}_i[k]$ is a control objective. $J^\star[k]$ denotes the optimal value of the objective function for the optimal control sequences $\boldsymbol{u}^\star_i[k:k+N_p-1|k]$. The problem in (4) is solved at each time k, and the $\boldsymbol{u}^\star_i[k|k]$ are applied in their respective subsystem i, following a receding horizon strategy (RHS).

Since the prediction horizon and the number of agents can be large, solving (4) at each time k can be rather challenging due to computational costs. Another issue that can arise is the fact that the complete information $\mathcal{I}_i = \{A_i, B_i, C_i, Q_i, R_i, \Gamma_i, \boldsymbol{v}_i\}$ from all the subsystems is needed to solve the problem, which can be viewed negatively from a confidentiality point of view.

So, we use the *primal decomposition* method, sometimes called *quantity decomposition* or *resource allocation*, with which we can profit from the parallelism, while avoiding the use of the complete information $\mathcal{I}_i$.

2.2 Primal Decomposition based dMPC

The technique decomposes the monolithic MPC in (4) into M modified MPC problems (5a) (solvable in parallel by each subsystem) called *local problems*, and a *master problem* (5b), which is equivalent to the global problem and is solved by a coordinator.

$$J^\star_i(\boldsymbol{\theta}_i[k]) = \begin{array}{ll}\underset{\boldsymbol{u}_i[k:k+N_p-1|k]}{\text{minimize}} & J_i[k] \\ \text{subject to} & (1)\ \&\ (2) \\ & \Gamma_i\boldsymbol{u}_i[k] \preceq \boldsymbol{q}_i[k] : \boldsymbol{d}_i[k]\end{array}\left.\begin{matrix}\\ \\ \\ \end{matrix}\right\} \begin{matrix}\forall i \in \mathcal{M} \\ \forall j \in \mathcal{H}\end{matrix} \tag{5a}$$

$$J^\star = \begin{array}{ll}\underset{\boldsymbol{q}_i[k:k+N_p-1|k]}{\text{minimize}} & \sum_{i\in\mathcal{M}} J^\star_i(\boldsymbol{q}_i[k]) \\ \text{subject to} & \sum_{i\in\mathcal{M}} \boldsymbol{q}_i[k] \preceq \boldsymbol{u}_{\max} \\ & \boldsymbol{q}_i[k] \succeq \mathbf{0}_{n_u}, \forall i \in \mathcal{M}\end{array}\left.\begin{matrix}\\ \\ \\ \end{matrix}\right\} \forall j \in \mathcal{H} \tag{5b}$$

The modified MPC problems (5a) are created by exchanging the $\boldsymbol{u}_{\max}$ in (3) by vectors $\boldsymbol{q}_i[k]$, which correspond to the quantity of the total resource $\boldsymbol{u}_{\max}$ allocated to agent i in time k for each prediction j, thus the names *resource allocation* and *quantity decomposition*. This new set of constraints have associated dual variables $\boldsymbol{d}_i[k] \succeq \mathbf{0}_{n_u}$. The sequences $\boldsymbol{q}_i[k:k+N_p-1|k]$ and $\boldsymbol{d}_i[k:k+N_p-1|k]$ can be aggregated in vectors $\boldsymbol{\theta}_i[k] = \mathbf{1}_{N_p} \otimes \boldsymbol{q}_i[k]$, and $\boldsymbol{\lambda}_i[k] = \mathbf{1}_{N_p} \otimes \boldsymbol{d}_i[k]$.

The *master problem* finds the optimal allocation sequences $\boldsymbol{\theta}^\star_i$ by using a projected sub-gradient method:

$$\boldsymbol{\theta}[k]^{(p+1)} = \text{Proj}^{\mathcal{S}}(\boldsymbol{\theta}[k]^{(p)} - \rho^{(p)}\boldsymbol{g}[k]^{(p)}), \tag{6}$$

where $\mathcal{S} = \{\boldsymbol{\theta} \mid I^M_c\boldsymbol{\theta} \preceq \boldsymbol{U}_{\max}\ \&\ \boldsymbol{\theta} \succeq \mathbf{0}\}$, being $c = N_pn_u$, $I^M_c = \mathbf{1}_M \otimes I_c$, $\boldsymbol{U}_{\max} = \mathbf{1}_{N_p} \otimes \boldsymbol{u}_{\max}$, (p) is a given step in the iterative process and $\boldsymbol{g}^{(p)}[k]$ is a sub-gradient of the objective function of problem in (5b) in step (p).

From Boyd et al. (2015), it is known that $-\boldsymbol{\lambda}_i[k]$ are sub-gradients of objective $J[k]$. Plugging it in $\boldsymbol{g}[k]^{(p)}$ we have

$$\boldsymbol{\theta}[k]^{(p+1)} = \text{Proj}^{\mathcal{S}}(\boldsymbol{\theta}[k]^{(p)} + \rho^{(p)}\boldsymbol{\lambda}[k]^{(p)}), \tag{7}$$

which henceforth is referred to as the *negotiation*.

Once the *negotiation* for a given time k converges, the $\boldsymbol{u}^\star_i[k|k]$ found in the last step of the *negotiation* are applied in their corresponding agent and then the RHS is followed.

The algorithm to find the optimal $\boldsymbol{\theta}^\star_i[k]$ can be summarized in Algorithm 1, and the exchange between coordinator and agents can be seen in Fig. 1.

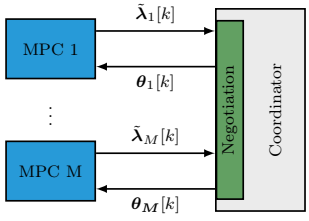

Fig. 1. Exchange between coordinator and agents.

Algorithm 1: Quantity decomposition negotiation.

$p := 0$
Coordinator initializes $\boldsymbol{\theta}^{(p)}$
repeat
  Subsystems solve (5a), and send $\boldsymbol{\lambda}_i^\star(\boldsymbol{\theta}_i^{(p)})$
  Coordinator updates allocations (7)
  $p := p + 1$
until $\|\boldsymbol{\theta}^{(p)} - \boldsymbol{\theta}^{(p-1)}\| \leq \epsilon$

One can observe that each agent only sends $\boldsymbol{\lambda}_i[k]$ instead of using $\mathcal{I}_i$. An interpretation of $\boldsymbol{\lambda}_i[k]$ is the dissatisfaction of agent i with the resource $\boldsymbol{\theta}_i[k]$ allocated for it, where $\boldsymbol{\lambda}_i[k] = \mathbf{0}_c$ means total satisfaction. The coordinator trusts in the authenticity of the $\boldsymbol{\lambda}_i[k]$ received to update the allocations. However, if false data is injected, the *negotiation* can be driven by a malicious agent, taking advantage of this trust to favor itself, harm others or even destabilize the *negotiation*, as shown in Nogueira et al. (2021).

Our main objective is to mitigate the effects of a given configuration where agents send untrustworthy

$$\tilde{\boldsymbol{\lambda}}_i = \gamma_i(\boldsymbol{\lambda}_i).$$

3. TOWARDS A SAFE DMPC

Here we focus on the local problems in (5a), and study how their structure can contribute for a safe dMPC algorithm.

3.1 Local Problems — Formal Analysis

The *local problems* (5a) can be rewritten in a equivalent (same solution) constrained *quadratic program* (QP) form:

$$\underset{\boldsymbol{U}_i[k]}{\text{minimize}} \quad J_i(\boldsymbol{\theta}_i) = \frac{1}{2}\boldsymbol{U}_i[k]^T H_i \boldsymbol{U}_i[k] + \boldsymbol{f}_i[k]^T \boldsymbol{U}_i[k] \quad (8a)$$

$$\text{subject to} \quad \bar{\Gamma}_i \boldsymbol{U}_i[k] \preceq \boldsymbol{\theta}_i[k] : \boldsymbol{\lambda}_i[k] \quad (8b)$$

$$\boldsymbol{U}_i[k] \succeq \mathbf{0}_c, \quad (8c)$$

where $\boldsymbol{U}_i[k]$ stacks the input prediction sequences, and $\bar{\Gamma}_i = I_{N_p} \otimes \Gamma_i$. The values for H_i and $\boldsymbol{f}_i[k]$ vary depending on the control objective $\boldsymbol{v}_i[k]$. The approach presented in this paper works for linear control objectives such as $\boldsymbol{v}_i[k] = \boldsymbol{y}_i[k]$ and $\boldsymbol{v}_i[k] = \boldsymbol{y}_i[k] - \boldsymbol{w}_i[k]$. The approach depends only on the fact that $H_i \in \mathbb{S}_+$ does not vary w.r.t. k, while $\boldsymbol{f}_i[k]$ does. Those facts are instrumental for the results in the following sections.

We can get an explicit solution for the dual variables $\boldsymbol{\lambda}_i[k]$, which are Piecewise Affine (PWA) functions w.r.t. $\boldsymbol{\theta}_i[k]$:

$$\boldsymbol{\lambda}_i[k] = -P_i^n \boldsymbol{\theta}_i[k] - \boldsymbol{s}_i^n[k], \text{ if } G^n[k]\boldsymbol{\theta}_i[k] \preceq \boldsymbol{b}^n[k], \quad (9)$$

with $n \in \{1 : N\}$.

The halfspaces defined by the pairs $(G^n[k], \boldsymbol{b}^n[k])$ represent the combinations of active constraints (8b) for a given time k, which vary w.r.t. $\boldsymbol{v}_i[k]$.

The P_i^n are constructed using H_i and $\bar{\Gamma}_i$; and $\boldsymbol{s}_i^n$ are constructed using H_i, $\boldsymbol{f}_i[k]$ and $\bar{\Gamma}_i$. The specific construction depends on the active constraints in (8b). For example, if all constraints are active, we have $P_i^{\text{ac}} = (\bar{\Gamma}_i H_i^{-1} \bar{\Gamma}_i^{\text{T}})^{-1}$ and $\boldsymbol{s}_i^{\text{ac}}[k] = P_i^{\text{ac}} \bar{\Gamma}_i H_i^{-1} \boldsymbol{f}_i[k]$. Futhermore, if all constraints are inactive, we have $P_i^{\text{in}} = 0_{c\times c}$ and $\boldsymbol{s}_i^{\text{in}} = \mathbf{0}_c$.

Remark 1. The $\boldsymbol{s}_i^n[k]$ depend on time k, while P_i^n do not.

Challenge 1. It is important to note that since $\boldsymbol{v}_i[k]$ is unknown by the coordinator, it cannot anticipate the partition of the space for each agent.

Challenge 2. For the same reason, the values of $\boldsymbol{s}_i[k]$ are also unknown.

As each constraint can be active or inactive, for a given group of t constraints, we can have at most $N = 2^t$ different combinations of active sets.

Assumption 1. We assume that none of the active/inactive constraints combinations are redundant (no linear dependency), nor make the optimization infeasible (no empty intersection). Thus we always have N zones.

3.2 The Attack

We suppose the malicious agent chooses a function $\gamma_i(\cdot)$ to use throughout a given time k.

Assumption 2. $\gamma_i(\cdot)$ does not change during the *negotiation* phase for a given time k.

Assumption 3. The agent chooses a linear function

$$\tilde{\boldsymbol{\lambda}}_i = \gamma_i(\boldsymbol{\lambda}_i) = T_i[k]\boldsymbol{\lambda}_i, \quad (10)$$

where T is invertible.

Such attack function results in a PWA solution for $\tilde{\boldsymbol{\lambda}}_i[k]$:

$$\tilde{\boldsymbol{\lambda}}_i[k] = -\tilde{P}_i^n[k]\boldsymbol{\theta}_i[k] - \tilde{\boldsymbol{s}}_i^n[k], \text{ if } G^n[k]\boldsymbol{\theta}_i[k] \preceq b^n[k], \quad (11)$$

with $n \in \{1 : N\}$, $\tilde{P}_i^n[k] = T_i[k]P_i^n$ and $\tilde{\boldsymbol{s}}_i^n[k] = T_i[k]\boldsymbol{s}_i^n[k]$. Observe that as the function is applied only in $\boldsymbol{\lambda}_i[k]$, it does not affect the zones' hyperplanes.

We fix the index for the zones where all constraints are active to 1, now referenced to as zones 1 or the 1-zones.

3.3 Detection and mitigation

Supposing we have a sequence of $\boldsymbol{\theta}_i[k]$ in a zone j, we can estimate the parameters for this zone with

$$\tilde{\boldsymbol{\lambda}}_i[k] = \gamma_i(\boldsymbol{\lambda}_i(\boldsymbol{\theta}_i[k])) = -\widehat{\tilde{P}_i^j}[k]\boldsymbol{\theta}_i[k] - \widehat{\tilde{\boldsymbol{s}}_i^j}[k]. \quad (12)$$

Assumption 4. The nominal value of P_i^j for this given j-zone, denoted $\bar{P}_i^j$, is available from reliable attack-free historical data.

Since P_i^j do not change w.r.t. time, we can detect a deviation from nominal behavior using $E_i[k] = \|\widehat{\tilde{P}_i^j}[k] - \bar{P}_i^j\|_F$. Let $\mathfrak{D}_i$ be an indicator

$$\mathfrak{D}_i = \mathbb{1}_{\{E_i[k] \geq \epsilon_P\}}, \quad (13)$$

which detects the attack in agent i if the disturbance $E_i[k]$ disrespects an arbitrary bound ϵ_P.

If an attack is detected, we can estimate the inverse of $T_i(k)$ with

$$\widehat{T_i(k)^{-1}} = \bar{P}_i^j \widehat{\tilde{P}_i^j}[k]^{-1}, \quad (14)$$

if $\widehat{\tilde{P}_i^j}[k]$ is invertible, and $\widehat{\tilde{P}_i^j}[k]$ is only invertible when all constraints are active, that means, when $j = 1$.

Assumption 5. For each agent i, at every time k, there is a corresponding 1-zone.

Moreover, from (9), we can reconstruct $\boldsymbol{\lambda}_i$:

$$\boldsymbol{\lambda}_{i\text{rec}} = \widehat{T_i[k]^{-1}}\tilde{\boldsymbol{\lambda}}_i \tag{15}$$

In order to estimate $\widehat{\tilde{P}_i^1}[k]$, we must have enough observed points in the 1-zones, so we propose to generate points surrounding arbitrary $\bar{\boldsymbol{\theta}}_i$ in the 1-zones.

Assumption 6. Given the unconstrained solution of 8 $\mathring{\boldsymbol{U}}_i^\star[k] = -H_i^{-1}\boldsymbol{f}_i[k]$, we suppose $\bar{\Gamma}_i\mathring{\boldsymbol{U}}_i^\star[k] \succeq \mathbf{0}_c$ for all k, that means, the points $\boldsymbol{\theta}_i = \mathbf{0}_c$ are in the 1-zones.

Unfortunately, since we do not know the hyperplanes separating the different zones, we do not know how close of $\bar{\boldsymbol{\theta}}_i = \mathbf{0}_c$ we need to generate our points. So we generate points arbitrarily close to $\bar{\boldsymbol{\theta}}_i$ and use the Expectation Maximization (EM) algorithm, which can potentially identify the parameters of all N modes.

3.4 Expectation Maximization

As the method is the same for all agents and repeated each time k, we drop the subscript i and the time dependency $[k]$ to simplify the notation.

Given a set of parameters $\mathcal{P}$, a set of observable data $\mathcal{B}$ and a set of non-observable data $\mathcal{U}$, the main objective of the EM algorithm is to find estimators of $\mathcal{P}$ that maximize the log marginal likelihood of the observed data $\ln \mathbb{P}(\mathcal{B};\mathcal{P})$, for models with latent variables in $\mathcal{U}$.

Since maximizing $\ln \mathbb{P}(\mathcal{B};\mathcal{P})$ does not have an analytical solution, the algorithm solves the optimization problem iteratively. So, rather than finding the $\mathcal{P}$ that maximizes $\ln \mathbb{P}(\mathcal{B};\mathcal{P})$, we find the $\mathcal{P}$ that maximizes the expectation of the complete data log-likelihood $\ln \mathbb{P}(\mathcal{B},\mathcal{U};\mathcal{P})$ w.r.t. the posterior probabilities $\mathbb{P}(\mathcal{U}|\mathcal{B};\mathcal{P})$ calculated using a given set of parameter estimates $\mathcal{P}_{\text{cur}}$. These steps provide a convergence-guaranteed iterative algorithm ensuring a monotonic increase of the log-marginal likelihood at each iteration (Algorithm 2).

Algorithm 2: Expectation Maximization

Initialize parameters $\mathcal{P}_{\text{new}}$
repeat
  $\mathcal{P}_{\text{cur}} \leftarrow \mathcal{P}_{\text{new}}$
  E step:
    Evaluate $\mathbb{P}(\mathcal{U}|\mathcal{B};\mathcal{P}_{\text{cur}})$
  M step:
    Reestimate parameters using:

$$\mathcal{P}_{\text{new}} = \arg\max_{\mathcal{P}}. \ \mathbb{E}_{\mathbb{P}(\mathcal{U}|\mathcal{B};\mathcal{P}_{\text{cur}})}\left[\ln \mathbb{P}(\mathcal{B},\mathcal{U};\mathcal{P})\right] \tag{16}$$

until $\mathcal{P}_{\text{cur}}$ *converges*

For a group of O exchanges between an agent and the coordinator, we observe the input and response variables, identified as $\underline{\boldsymbol{\theta}}_o$ and $\underline{\boldsymbol{\lambda}}_o$, $o \in \mathcal{O} = \{1:O\}$, which can be organized in $\underline{\Theta}, \underline{\Lambda} \in \mathbb{R}^{c\times O}$. $(\underline{\Theta}, \underline{\Lambda})$ is our set of observable data $\mathcal{B}$.

As (11) gives us a multidimensional PWA function, we propose using a multidimensional expansion of the model referred to as *mixture of linear regressions* in Faria and Soromenho (2010) to map the relation between $\underline{\Theta}$ and $\underline{\Lambda}$.

We will call the model *mixture of affine regressions*, since our regressors have linear and constant terms:

$$\underline{\boldsymbol{\lambda}}_o = -\tilde{P}^z\underline{\boldsymbol{\theta}}_o - \tilde{\boldsymbol{s}}^z, \text{ if in zone } z, \tag{17}$$

with $z \in \mathcal{Z} = \{1:Z\}$.

Remark 2. Observe that the indices z, do not necessarily correspond to the original indices n.

As (17) is a PWA function whose modes depend on the unknown zone $z \in \mathcal{Z}$, we associate to each couple $(\underline{\boldsymbol{\lambda}}_o, \underline{\boldsymbol{\theta}}_o)$ a latent unobserved random variable $\underline{z}_o$ that indicates in which zone in $\mathcal{Z}$ the observable variables were obtained. These variables are organized in $\underline{Z} \in \mathbb{R}^{1\times O}$ which is our set of latent variables $\mathcal{U}$.

The latent variable $\underline{z}_o$ follows a categorical prior distribution, with associated probabilities $\Pi = \{\pi^z | z \in \mathcal{Z}\}$ such

$$\mathbb{P}(\underline{z}_o = z) = \pi^z \in [0,1], \qquad \sum_{z\in\mathcal{Z}} \pi^z = 1.$$

As parameters to estimate, we have $\mathcal{P} = \{\mathcal{P}^z \mid z \in \mathcal{Z}\}$, with $\mathcal{P}^z = (\tilde{P}^z, \tilde{\boldsymbol{s}}^z, \pi^z)$.

Since $\boldsymbol{\theta}$ is our input, we consider a non-informative improper probability density function (Christensen et al., 2010)

$$\mathbb{P}(\underline{\boldsymbol{\theta}}_o) \propto 1.$$

Given the input and latent variables, the response variable $\underline{\lambda}_o$ is modeled as a multivariate normal random variable with probability density function

$$\mathbb{P}(\underline{\boldsymbol{\lambda}}_o|\underline{\boldsymbol{\theta}}_o, \underline{z}_o = z; \mathcal{P}^z) = \mathcal{N}(\underline{\boldsymbol{\lambda}}_o; f(\underline{\boldsymbol{\theta}}_o; \mathcal{P}^z), \Sigma^z), \tag{18}$$

where, following (17), the mean vector is defined by $f(\underline{\boldsymbol{\theta}}_o; (P, \boldsymbol{s}, \pi)) = -P\underline{\boldsymbol{\theta}}_o - \boldsymbol{s}$, and the covariance matrix Σ^z tends to 0.

We can calculate the posterior probabilities $\zeta_{zo}(\mathcal{P}) = \mathbb{P}(\underline{z}_o = z|\underline{\boldsymbol{\lambda}}_o, \underline{\boldsymbol{\theta}}_o; \mathcal{P})$, also called *responsibilities*:

$$\zeta_{zo}(\mathcal{P}) = \frac{\pi_z \mathcal{N}(\underline{\boldsymbol{\lambda}}_o; f(\underline{\boldsymbol{\theta}}_o; \mathcal{P}^z), \Sigma^z)}{\sum_{j=1}^{Z} \pi_j \mathcal{N}(\underline{\boldsymbol{\lambda}}_o; f(\underline{\boldsymbol{\theta}}_o; \mathcal{P}^j), \Sigma^j)}, \tag{19}$$

and then calculate the expectation of $\ln \mathbb{P}(\underline{\Theta}, \underline{\Lambda}, \underline{Z}; \mathcal{P})$ with respect to $\zeta_{zo}(\mathcal{P}_{\text{cur}})$ (Bishop, 2006, Chapter 9)

$$\mathbb{E}_{\zeta_{zo}(\mathcal{P}_{\text{cur}})}\left[\ln \mathbb{P}(\underline{\Theta}, \underline{\Lambda}, \underline{Z}; \mathcal{P})\right] = \sum_{o\in\mathcal{O}}\sum_{z\in\mathcal{Z}} \zeta_{zo}(\mathcal{P}_{\text{cur}})\alpha_{zo}, \tag{20}$$

where $\alpha_{zo} = \ln \pi_z + \ln \mathcal{N}(\underline{\boldsymbol{\lambda}}_o; f(\underline{\boldsymbol{\theta}}_o; \mathcal{P}^z), \Sigma^z)$.

Remark 3. The Σ^z used in (19) are updated using a technique called *Simulated annealing* (Ozerov and Fevotte, 2010), where they are initialized with arbitrarily significant values indicating the uncertainty of the parameters and it is reduced as the estimations converge.

Here we introduce a variable $\boldsymbol{\phi}^z = \left[\text{vec}(\tilde{P}^z)^T \ (\tilde{\boldsymbol{s}}^z)^T\right]^T$. We can find an optimal $\boldsymbol{\phi}^z$ for the problem in (16), by taking the gradients of (20) with respect to vectors $\boldsymbol{\phi}^z$ and making them vanish. Because of the multidimensional nature of the problem, some matrix operations are needed to synthesize the results. After those operations, we have a matricial solution that yields the optimal estimates $\boldsymbol{\phi}^z_{\text{new}}$:

$$\boldsymbol{\phi}^z_{\text{new}} = (\Xi^z\underline{\Omega})^\dagger \Xi^z \text{vec}(\underline{\Lambda}), \tag{21}$$

where $\underline{\Omega} = [(\Upsilon\underline{\Theta}\Delta) \odot Y; G]$, with matrices $\Upsilon = \mathbf{1}_c^T \otimes I_c$, $\Delta = I_O \otimes \mathbf{1}_c^T$, $Y = G \otimes \mathbf{1}_c$, $G = \mathbf{1}_O^T \otimes I_c$, and

$$\Xi^z = \text{diag}(\sqrt{\zeta(z_{z1}; \mathcal{P}_{\text{cur}})}I_c, \cdots, \sqrt{\zeta(z_{zO}; \mathcal{P}_{\text{cur}})}I_c).$$

Doing the same for π^z we get

$$\pi^z = \sum_{o \in \mathcal{O}} \frac{\zeta_{zo}(\mathcal{P}_{\text{cur}})}{O}.$$

As we can see, (21) is the solution of a weighted Least-Squares, with responsibilities as weights, which adjust the contribution of all observations to the regression models. We can see some similarities to the K-planes algorithm (see Bradley and Mangasarian (2000)), but EM is more compromising. Instead of affecting the observed data to a zone with 100% of certainty (*hard assignment*), EM uses each zone's responsibilities (*soft assignment*).

Once the estimates $\boldsymbol{\phi}_z^{\text{new}}$ converge, we can reconstruct the estimates $\tilde{P}^z$ and $\tilde{\boldsymbol{s}}^z$, and use them in our mitigation scheme proposed in §3.3. Since the z indices do not necessarily correspond to the indices n in (9) (due to initialization of $\mathcal{P}$), to find the z that corresponds to $n = 1$, we take the observation o for $\bar{\boldsymbol{\theta}} = \mathbf{0}_c$, which we know belongs to the 1-zone, and we find the most probable z with

$$\arg\max_z \ \zeta_{zo}(\mathcal{P}).$$

Remark 4. A discussion about the initialization and update of other parameters is beyond the scope of this article, so we refer the reader to Bishop (2006).

3.5 Safe dMPC Algorithm

Integrating the EM to our detection and mitigation mechanism we can propose in Algorithm 3 a secure dMPC, divided into two phases: detection and negotiation. The new algorithm corresponds to adding a supervision layer for each agent as in Fig. 2

Algorithm 3: Secure DMPC.

Detection Phase:

- Coordinator sends sequence of $\boldsymbol{\theta}_i^o$, $o \in \mathcal{O}$
- Subsystems solve (5a), and send $\tilde{\boldsymbol{\lambda}}_i^o$, $o \in \mathcal{O}$
- Coordinator estimates $\widehat{\tilde{P}_i^1}[k]$ and $\widehat{\tilde{\boldsymbol{s}}_i^1}[k]$ with EM
- Coordinator computes $\mathfrak{D}_i$ using (13)

Negotiation Phase:

- Apply Algorithm 1 with adequate versions of $\boldsymbol{\lambda}_i^{(p)}$: $\tilde{\boldsymbol{\lambda}}_i^{(p)}$, if $\mathfrak{D}_i = 0$ and $\boldsymbol{\lambda}_{i\text{rec}}$, if $\mathfrak{D}_i = 1$ (15)

4. EXAMPLE: TEMPERATURE CONTROL

The system consists of 4 distinct rooms (I to IV), which we want to control the air temperatures inside each one of them. The systems are modeled as continuous-time linear time-invariant systems using the 3R-2C model, with parameters in Tables 1 and 2, and dynamics

$$\begin{aligned} \dot{\boldsymbol{x}}_i(t) &= A_{ci}\boldsymbol{x}_i(t) + B_{ci}\boldsymbol{u}_i(t) \\ \boldsymbol{y}_i(t) &= C_{ci}\boldsymbol{x}_i(t) \end{aligned}, \tag{22}$$

with

$$A_{\text{c}_i} = \begin{bmatrix} -\dfrac{1}{C_i^{\text{walls}} R_i^{\text{oa/ia}}} - \dfrac{1}{C_i^{\text{walls}} R_i^{\text{iw/ia}}} & \dfrac{1}{C_i^{\text{walls}} R_i^{\text{iw/ia}}} \\ \dfrac{1}{C_i^{\text{air}} R_i^{\text{iw/ia}}} & -\dfrac{1}{C_i^{\text{air}} Ro_i} - \dfrac{1}{C_i^{\text{air}} R_i^{\text{iw/ia}}} \end{bmatrix},$$

$$B_{\text{c}_i} = \begin{bmatrix} \dfrac{10}{C_i^{\text{walls}}} & 0 \end{bmatrix}^{\text{T}} \quad C_{\text{c}_i} = [1 \ 0]$$

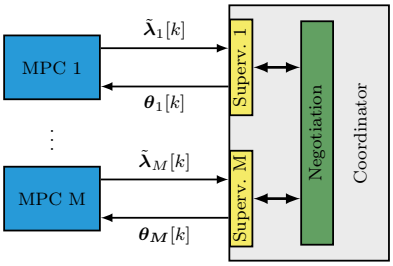

Fig. 2. Exchange between agents in secure dMPC.

where $\boldsymbol{x}_i = [x_{Ai}^{\text{T}} \ x_{Wi}^{\text{T}}]^{\text{T}}$. x_{Ai} and x_{Wi} are the mean temperatures of the air and walls inside room i. $\boldsymbol{u}_i$ is the input (the heating power) for the corresponding room. The inputs are constrained by $\sum_{i=1}^{4} \boldsymbol{u}_i(t) \preceq 4\text{kW}$.

The subsystems are discretized using zero-order hold discretization method with sampling time $T_s = 0.25\text{h}$ and the quantity decomposition-based dMPC is implemented using prediction horizon $N_p = 4$.

Three scenarios are simulated for a period of 12.5h:

(1) Nominal behavior.
(2) Agent I presents non-cooperative behavior for $k \geq 25$.
(3) Agent I presents non-cooperative behavior for $k \geq 25$, with correction.

For scenarios (2) and (3), Agent I uses

$$T_I = \begin{bmatrix} 14.43288267 & 0. & 0. & 0. \\ 0. & 13.4590903 & 0. & 0. \\ 0. & 0. & 6.93065061 & 0. \\ 0. & 0. & 0. & 3.4447393 \end{bmatrix}.$$

In Fig. 3, first, we compare the air temperature in room I with its reference ($w_I = 25.5°\text{C}$), and then the decision variable $E_I(k)$ with the threshold ϵ_P. Observe that the reference w_I is not reached in the nominal behavior (in magenta), due to power constraints by which the systems are influenced. As expected, the decision variable lies under the threshold $\epsilon_P = 10^{-4}$ with values of order $E_I^N(k) \approx 10^{-10}$.

When the agent presents a selfish behavior (in orange), the tracking error $w_I - y_I$ is reduced, almost attaining the reference. In this case, the detection variable surpasses ϵ_P, $E_I^S \approx 200$, indicating the change of behavior.

Table 1. Model Parameters

Symbol	Meaning
C_i^{air}	Heat Capacity of Inside Air
C_i^{walls}	Heat Capacity of External Walls
$R_i^{\text{iw/ia}}$	Resist. Between Inside Air and Inside Walls
$R_i^{\text{ow/oa}}$	Resist. Between Outside Air and Outside Walls
$R_i^{\text{oa/ia}}$	Resist. Between Inside and Out. Air (from windows)

Table 2. Parameters for each agent

Symbol	I	II	III	IV	Unit
C^{walls}	5.4	4.9	4.7	4.7	10^4J/K
C^{air}	7.5	8.4	8.2	7.7	10^4J/K
$R^{\text{oa/ia}}$	5.2	4.6	4.9	5.4	10^{-3}K/W
$R^{\text{iw/ia}}$	2.3	2.4	2.3	2.9	10^{-4}K/W
$R^{\text{ow/oa}}$	1.5	0.6	0.7	0.7	10^{-4}K/W

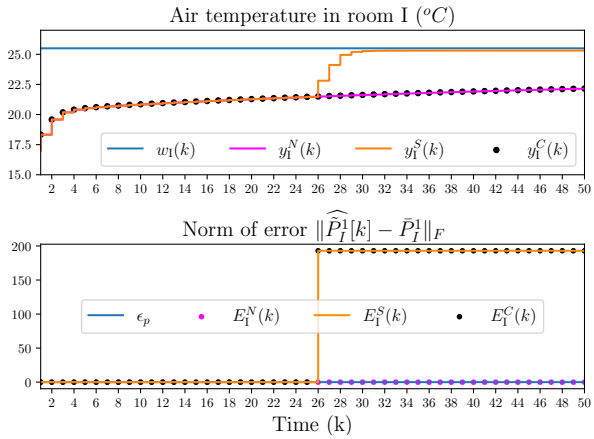

Fig. 3. Air temperature in room I and the decision variable $E_I[k]$ for three scenarios: nominal (N), selfish behavior (S), and selfish behavior with correction (C).

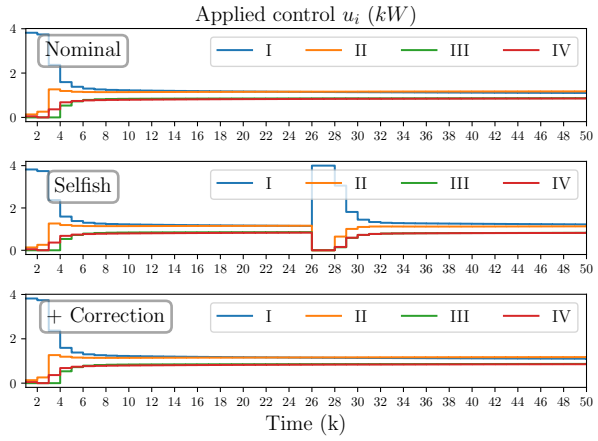

Fig. 4. Control applied in all rooms for the 3 scenarios.

When the correction is activated in the system, the temperatures approach their nominal value y_I^N. We can also illustrate the good performances of our proposition by comparing the inputs in Fig. 4. When room I is selfish, its control increases while other rooms' decrease. When the correction is activated, it approaches the nominal.

We can also evaluate the performance of the proposed mechanism by comparing the objective functions calculated using the period of simulation $N = 50$ for the three scenarios (Table 3). When agent I is selfish, we see the decrease in its objective ($\approx -40\%$), degrading the overall performance ($\approx +10\%$). When the correction mechanism is activated, the absolute percentual error is $|\frac{J^C - J^N}{J^N}| \leq 10^{-8}$.

Table 3. 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)

5. CONCLUSION AND FUTURE WORKS

In this paper, we proposed an algorithm for monitoring and correcting exchanges between agents in a resource-sharing framework. The first phase of the algorithm identifies the attacker by exploiting the exchange structure. After this identification, if necessary, it is possible to reconstruct the original mechanism and recover nominal optimality by inverting the attack. This principle should be generalized to cases when the attack is not entirely invertible, reconstructing by parts the original mechanism. Also, other decomposition structures and attack models need to be explored, which we plan to do shortly.

REFERENCES

Bishop, C.M. (2006). *Pattern Recognition and Machine Learning*. Springer Science and Business Media LLC.

Boyd, S., Xiao, L., Mutapcic, A., and Mattingley, J. (2015). Notes on decomposition methods. In S. University (ed.), *Notes for EE364B*.

Bradley, P. and Mangasarian, O. (2000). K-plane clustering. *Journal of Global Optimization*, 16(1), 23–32.

Chanfreut, P., Maestre, J.M., and Ishii, H. (2018). Vulnerabilities in distributed model predictive control based on Jacobi-Gauss decomposition. In *2018 European Control Conference (ECC)*, 2587–2592.

Christensen, R., Johnson, W., Branscum, A., and Hanson, T.E. (2010). *Bayesian ideas and data analysis: an introduction for scientists and statisticians*. CRC press.

Dempster, A.P., Laird, N.M., and Rubin, D.B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1), 1–22.

Faria, S. and Soromenho, G. (2010). Fitting mixtures of linear regressions. *Journal of Statistical Computation and Simulation*, 80(2), 201–225.

Kantamneni, A., Brown, L.E., Parker, G., and Weaver, W.W. (2015). Survey of multi-agent systems for microgrid control. *Engineering Applications of Artificial Intelligence*, 45, 192–203.

Maestre, J.M., Negenborn, R.R., et al. (2014). *Distributed Model Predictive Control made easy*, volume 69. Springer.

Maestre, J.M., Velarde, P., Ishii, H., and Negenborn, R.R. (2021). Scenario-based defense mechanism against vulnerabilities in lagrange-based dmpc. *Control Engineering Practice*, 114, 104879.

Nogueira, R.A., Bourdais, R., and Guéguen, H. (2021). Detection and mitigation of corrupted information in distributed model predictive control based on resource allocation. In *2021 5th Conference on Control and Fault-Tolerant Systems (SysTol)*, 329–334.

Ozerov, A. and Fevotte, C. (2010). Multichannel nonnegative matrix factorization in convolutive mixtures for audio source separation. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(3), 550–563.

Velarde, P., Maestre, J.M., Ishii, H., and Negenborn, R.R. (2017). Vulnerabilities in lagrange-based distributed model predictive control. *Optimal Control Applications and Methods*, 39(2), 601–621.