Detection and Mitigation of Corrupted Information in Distributed Model Predictive Control Based on Resource Allocation

Rafael Accacio Nogueira, Romain Bourdais, Hervé Guéguen

Detection and Mitigation of Corrupted Information in Distributed Model Predictive Control Based on Resource Allocation

Rafael Accácio Nogueira, Romain Bourdais and Hervé Guéguen

***Abstract*— In distributed predictive control structures, communication among agents is required to achieve a consensus and approach an optimal global behavior. Such negotiation mechanisms are sensitive to attacks on these exchanges. This paper proposes a monitoring scheme that detects and mitigates these attacks' effects in a resource allocation framework. The performance of the proposed method is illustrated through simulations of the temperature control of multiple rooms under power scarcity.**

I. INTRODUCTION

Recent performance objectives require systems to be driven not in isolation but in a coordinated way, emphasizing large systems. These systems cover many applications, such as energy distribution systems, traffic management in Smart City environments, coordinated control of intelligent building systems, and many others. Many works are built around model predictive control [1] to integrate optimality and constraints.

Furthermore, distributed model predictive control (DMPC) [2] techniques are a promising way to handle the optimization problem's complexity. In these structures, there is no longer a single controller for all systems. Instead, we use a set of local communicating controllers. These strategies thus reduce the computing burden while increasing confidentiality.

Many works use distributed optimization techniques, such as Lagrangian relaxation [3], Alternating Direction Method of Multipliers (ADMM) [4], primal decomposition [5], dual decomposition [6], [7], [8], and others [9], [10]. In these methods, local agents interact with a coordinator who uses an iterative process to ensure convergence towards the solution of the initial problem.

Usually, it is assumed that all agents work in perfect cooperation. However, when it is not the case, these uncooperative behavior have pernicious effects on the overall system, and their impact can be studied. The cause of this disruptive behavior can be either involuntary due to hacking or malfunctioning, or voluntary, by developing selfish behavior. Recent work has begun to explore this issue. In the article [6], the authors are interested in the vulnerabilities induced when distributed predictive control is built on dual decomposition. They analyze the impact of the deception depending on where it occurs: either in the followed references or directly in the local cost functions or coupling constraints. The same authors propose defense strategies against these attacks, either by using secure scenarios based on reliable historical data [7] or by ignoring extreme values of control signals [6]. Then [9] extends the initial work to analyze the vulnerabilities of the Jacobi-Gauss decomposition method.

Another way of dealing with these changes in behavior can be using robust distributed control principles, coupled with hierarchical identification of the attack [11], or the introduction of probabilistic models that implement a resilient strategy if the information exchanged is outside the confidence interval [12].

In this work, we analyze the exchange among agents controlled by DMPC using primal decomposition, which is perfectly adapted for agents that share resources. More specifically, we are interested when malicious agent steers these exchanges. By exploiting the nominal structure that characterizes the communication between the agents and the coordinator, we propose a monitoring scheme that detects an attack, and if necessary, corrects it.

The remainder of this paper is organized as follows. First, in Section II, the primal decomposition-based DMPC is introduced. In Section III, we discuss a model of the agents' selfish behavior that exploits the vulnerabilities of this DMPC structure. Then, in Section IV, we discuss the structure of the DMPC and how we can exploit it to construct a defense scheme to counteract the selfish agent. At last, we present a particular mechanism to detect the agents' selfish behavior and mitigate its effects. Moreover, in Section V, an application is given to illustrate and evaluate the algorithm's performance. Finally, in Section VI, we conclude, and we give an outlook of future works.

II. PRELIMINARIES AND PROBLEM STATEMENT

Notation: In this paper, $\|\cdot\|$ and $\|\cdot\|_F$ represent the ℓ_2 and Frobenius norms. $\|\boldsymbol{v}\|_Y$ is the weighted norm, $\|Y^{\frac{1}{2}}\boldsymbol{v}\|$. $P_{\mathsf{T}}(\cdot)$ is the Euclidean projection onto set T. $\otimes$ represents the Kronecker product. $\mathbf{1}_{m,n}$ is a $m \times n$ matrix filled with 1. I_c is a $c \times c$ identity matrix. π_v denotes the number of elements in v. A vector $\boldsymbol{v}_i$, correspond to the i-th agent, and these vectors can be stacked in a vector $\boldsymbol{v}$.

A. Model Predictive Control

Our primary purpose is to control a system composed of M subsystems using MPC. The dynamics of the state $\boldsymbol{x}_i(k)$ of i-th agent w.r.t input $\boldsymbol{u}_i(k)$ are described by the following linear discrete-time systems:

$$\boldsymbol{x}_i(k+1) = A_i\boldsymbol{x}_i(k) + B_i\boldsymbol{u}_i(k) \tag{1}$$

The authors are with IETR-CentraleSupélec, 35510 Cesson-Sévigné, Ille-et-Vilaine, France
{rafael-accacio.nogueira, romain.bourdais, herve.gueguen} @centralesupelec.fr

The M subsystems are coupled under linear input constraints. We assume as an interesting case when these constraints prevent the subsystems from meeting the systems' needs. Consequently, the constraints will always be active, yielding the same results from equality constraints [13]:

$$\sum_{i=1}^{M} \Gamma_i \boldsymbol{u}_i(k) = \boldsymbol{u}_{\max} \tag{2}$$

where $\Gamma_i : \mathbb{R}^{\pi_{\boldsymbol{u}_i(k)} \times \pi_{\boldsymbol{u}_i(k)}}$ and $\boldsymbol{u}_{\max} : \mathbb{R}^{\pi_{\boldsymbol{u}_i(k)} \times 1}$.

A known formulation of the MPC structure [1], [6], [7], [14], [9] with finite prediction horizon N_p is the following:

Problem 1: Global MPC Problem.

$$\underset{\boldsymbol{u}_i(k:k+N_p-1|k)}{\text{minimize}} \quad \overbrace{\sum_{i=1}^{M} \overbrace{\sum_{j=1}^{N_p} \|\boldsymbol{v}_i(k+j|k)\|^2_{Q_i} + \|\boldsymbol{u}_i(k+j-1|k)\|^2_{R_i}}^{J_i(k)}}^{J_G(k)}$$

$$\text{subject to} \qquad (1) \text{ and } (2) \begin{cases} \forall i \in \{1, \ldots, M\} \\ \forall j \in \{1, \ldots, N_p\} \end{cases}$$

with symmetric weight matrices $Q_i \geq 0, \quad R_i > 0$. $\boldsymbol{v}_i(k)$ represents a control objective. It can either be $\boldsymbol{v}_i(k) = \boldsymbol{w}_i(k) - \boldsymbol{x}_i(k)$ for reference tracking, where $\boldsymbol{w}_i(k)$ is a state reference, or $\boldsymbol{v}_i(k) = \boldsymbol{x}_i(k)$ for disturbance rejection.

The optimal value of the problem 1 is denoted by $J^\star$, and the optimal control sequences are represented by $\boldsymbol{u}_i^\star(k : k+N_p-1|k)$. At each time k, the problem is solved, and the $\boldsymbol{u}_i^\star(k|k)$ are applied in each respective i subsystem, following a receding horizon strategy.

One can see that if the subsystems were not coupled by (2), the overall system could be decomposed into M parts, solvable in parallel. Multiple decomposition methods solve this problem [4], [5], [8], [9]. Still, since we are interested in resource constraints and the dual decomposition does not enforce local feasibility [15], the primal decomposition is chosen.

B. Distributed Model Predictive Control

The technique consists of decomposing the *coupling constraints* (or *complicating constraints* [15]) of the original optimization problem into local versions with additional variables that are shared among them, negotiating the value of these variables until a consensus is reached.

Problem 1 is decomposed into multiple subproblems (3a), solvable in parallel, and a *master problem* (3b), which is equivalent to the original problem and uses information of the subproblems [15]:

$$\left.\begin{aligned} J_i^\star(\boldsymbol{\theta}_i(k)) = \underset{\boldsymbol{u}_i(k:k+N_p-1|k)}{\text{minimize}} \; & J_i(k) \\ \text{s.t.} \quad & (1) \\ & \Gamma_i \boldsymbol{u}_i(k) = \boldsymbol{\theta}_i(k) : \boldsymbol{\lambda}_i(k) \end{aligned}\right\} \begin{matrix} \forall i \in \{1, \ldots, M\} \\ \forall j \in \{1, \ldots, N_p\} \end{matrix} \tag{3a}$$

$$\begin{aligned} J^\star = \underset{\boldsymbol{\theta}(k:k+N_p-1|k)}{\text{minimize}} \; & \sum_{i=1}^{M} J_i^\star(\boldsymbol{\theta}_i(k)) \\ \text{s.t.} \quad & \sum_{i=1}^{M} \boldsymbol{\theta}_i(k) = \boldsymbol{u}_{\max} \end{aligned} \tag{3b}$$

The subproblems (3a) are formed by the local objectives $J_i(k)$ and a set of local constraints, with a sequence of allocations $\boldsymbol{\theta}_i(k : k+N_p-1|k)$ and associated sequence of dual variables (Lagrange multipliers) $\boldsymbol{\lambda}_i(k : k+N_p-1|k)$. For brevity's sake, we drop the $(k : k+N_p-1|k)$ sequence notation, using only where pertinent.

The variables $\boldsymbol{\theta}_i$ represent the resource or the "quantity" allocated for each subproblem; thus, the names "quantity decomposition" and "resource allocation" are also given for this decomposition [10].

The *master problem* shown in (3b) can be solved using an iterative method that updates the allocation sequence $\boldsymbol{\theta}_i$.

Due to the form of the constraints, we use the projected sub-gradient method whose recurrence equation is:

$$\boldsymbol{\theta}^{(p+1)} = P_{\mathsf{H}}(\boldsymbol{\theta}^{(p)} - \rho^{(p)} \boldsymbol{g}^{(p)}) \tag{4}$$

where $\mathsf{H} = \{\boldsymbol{\theta} \mid \sum_{i=1}^{M} \boldsymbol{\theta}_i = \boldsymbol{u}_{\max}\}$, $\boldsymbol{g}^{(p)}$ is a sub-gradient of $J^\star(\boldsymbol{\theta}^{(p)})$ at the instant p and $\rho^{(p)}$ is an iteration step, well-chosen, so the method converges.

The sum $\sum_{i=1}^{M} \boldsymbol{\theta}_i$ can also be represented by the matrix multiplication $I_c^M \boldsymbol{\theta}$, where $I_c^M = \mathbf{1}_{M,1} \otimes I_c$. Where $c = \pi_{\boldsymbol{u}_i(k:k+N_p-1|k)} = N_p \pi_{\boldsymbol{u}_i(k)}$.

Assuming *strong duality* holds, we can use the sensitivity analysis of the problem [13, § 5.6.2], and we can conclude that the opposite of the sequences of optimal dual variables, $-\boldsymbol{\lambda}_i^\star$, which are $\boldsymbol{\theta}_i^{(p)}$ dependent, is a sub-gradient of $J_i^\star(\boldsymbol{\theta}_i^{(p)})$, which can be used in (4) to solve the problem (3b).

Applying the Euclidean projection onto H [16] and using $-\boldsymbol{\lambda}_i^\star(\boldsymbol{\theta}_i^{(p)})$ in (4) results in the complete expression for the allocation's update [10, §VI-C]:

$$\boldsymbol{\theta}_i^{(p+1)} = \boldsymbol{\theta}_i^{(p)} + \rho \left(\boldsymbol{\lambda}_i^\star(\boldsymbol{\theta}_i^{(p)}) - I_c^M (I_c^{M\mathrm{T}} I_c^M)^{-1} I_c^{M\mathrm{T}} \boldsymbol{\lambda}^\star(\boldsymbol{\theta}^{(p)}) \right) \tag{5}$$

In each step (p), the subproblems receive a sequence of allocation of the total resources. Then they return their corresponding sequence of dual variables so the *master problem* can be solved by updating the allocations, recommencing the negotiation. Once a consensus is reached, the negotiation is finished, each subsystem takes the last sequence of inputs $\boldsymbol{u}_i^\star(k : k+N_p-1|k)$ calculated and applies the first element $\boldsymbol{u}_i^\star(k|k)$, following a receding horizon strategy.

Delegating the iterative process of allocation update to an agent with the *coordinator*'s role, we have the scheme in Fig. 1 that illustrates the negotiation. Observe that each block *negot* solves (5) for a respective agent i. This way, the only interaction that the *coordinator* has with the subsystems is via the variables $\boldsymbol{\lambda}_i^{(p)}$ and $\boldsymbol{\theta}_i^{(p+1)}$, increasing the privacy of the subsystems.

Algorithm 1 resumes the distributed control problem solved to calculate the optimal input sequence at each time k using quantity decomposition.

III. Attack in DMPC scheme

As expected [15], [10], this decomposition method works well when each agent cooperatively calculates its $\boldsymbol{\lambda}_i$ correctly. Here we study the effects when an ill-intentioned agent exploits the scheme for its interest.

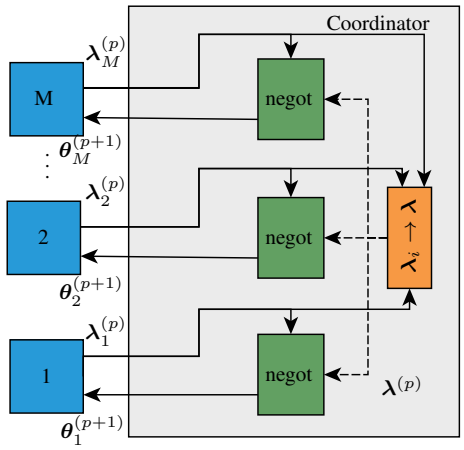

Fig. 1. Scheme of DMPC using a *coordinator* and M agents.

Algorithm 1: Quantity decomposition based DMPC.

Coordinator initializes $\boldsymbol{\theta}^{(0)}$
$p := 0$
repeat
  Subsystems solve (3a), and send $\boldsymbol{\lambda}_i^\star(\boldsymbol{\theta}^{(p)})$
  Coordinator updates allocations (5)
  $p := p + 1$
until $\|\boldsymbol{\theta}^{(p)} - \boldsymbol{\theta}^{(p-1)}\| \leq \epsilon$

[6], [7], [9], [12] present 4 types of attacks, which can be divided into 2 principal groups: changes in the optimization parameters (*selfish attack* - multiply the objective function by a scalar α, *fake reference*, and *fake constraints*) and nonagreed control (*liar agent*). In the decomposition scheme used in this work, the coordinator allocates the resources. So we can discard the last kind of attack.

Although we could make the same analysis from the mentioned works, we are interested in the coordinator's point of view, so any of these attacks will reflect as a change on the $\boldsymbol{\lambda}_i$ received. Therefore, we propose that any selfish agent sends a corrupted

$$\tilde{\boldsymbol{\lambda}}_i = \gamma_i(\boldsymbol{\lambda}_i) \tag{6}$$

to the coordinator instead of sending the agreed $\boldsymbol{\lambda}_i$.

We give a unidimensional example where $\gamma_i(\lambda_i) = \tau_i\,\lambda_i$ to illustrate such an attack. Here, 4 agents negotiate with the coordinator, and agent 1 attacks the system ($\tau_1 \neq 1$).

In Fig. 2, we see that when $\tau_i > 1$, agents 1's local cost $J_1^\star$ decreases while all other costs, including the overall $J^\star$, increase. This attack is comparable to the *selfish attack* portrayed in [6]. This decrease in the cost justifies the attack since the attacking agent has more comfort than all others.

On the other hand, when τ_1 tends to 0, $J_1^\star$ increases and all others J_i decreases, while still degrading the overall cost $J^\star$. Such an agent could be considered as a benevolent agent or an agent attacked by a malevolent one.

From this variation in the values of $\boldsymbol{\lambda}_1$ caused by τ_1, we can interpret its role in the negotiation: the values of $\boldsymbol{\lambda}_i$ represent the dissatisfaction with the given allocation $\boldsymbol{\theta}_i$.

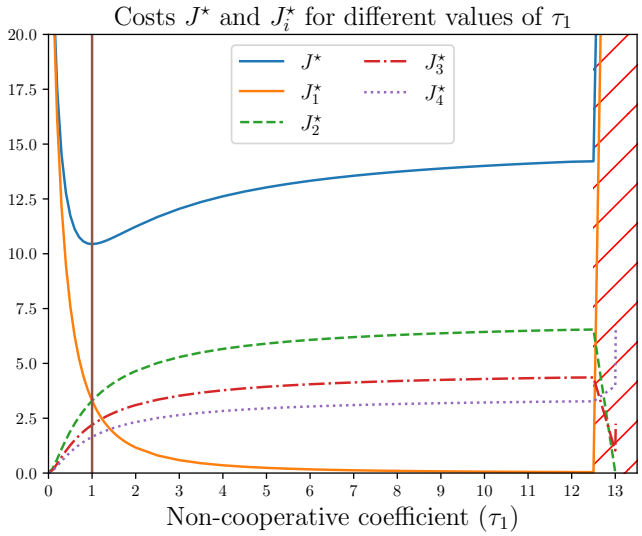

Fig. 2. Change of $J^\star$ with respect to the non-cooperative coefficient τ_1.

Since the negotiation (5) finds its stability when $\boldsymbol{\lambda}^{(p)} - I_c^M (I_c^{M\mathrm{T}} I_c^M)^{-1} I_c^{M\mathrm{T}} \boldsymbol{\lambda}^{(p)} = \mathbf{0}$, that means when all $\boldsymbol{\lambda}_i$ are equal to the mean of the $\boldsymbol{\lambda}_i$. We can interpret that the coordinator's role is to minimize the overall dissatisfaction. This way, the selfish agent can lie about its dissatisfaction (increasing $\boldsymbol{\lambda}_i$ by using an adequate $\gamma_i(\cdot)$), driving the negotiation to a value of $\boldsymbol{\theta}_i$ that "satisfies" it more (lower optimal value $J_i^\star$).

Another effect we can expect from the observation of (5) is that the negotiation may not converge for some values of $\boldsymbol{\lambda}_i$. We can find those values by the analysis of the eigenvalues of the iterative process. This effect is illustrated in the hatched area in Fig. 2.

IV. Secure DMPC based on resource allocation

As seen, a malicious agent can deviate the allocations for its benefit, driving the negotiation or even destabilizing it. Hence, it is needed to find a way to lessen the effects caused by this agent. To fill this gap, we propose a detection and mitigation mechanism to reduce the effects of any agent malfeasance in the negotiation. However, before presenting the mechanism, we need to analyze the problem structure to sustain the proposition.

A. Quadratic Case — Formal Analysis

Another known form to represent the problems (3a) is using matrix representation [9]:

$$\begin{aligned} \underset{\boldsymbol{U}_i(k)}{\text{minimize}} \quad & \overbrace{\frac{1}{2}\boldsymbol{U}_i(k)^T H_i \boldsymbol{U}_i(k) + \boldsymbol{f}_i(k)^T \boldsymbol{U}_i(k)}^{J_i(\boldsymbol{\theta}_i)} \\ \text{s.t.} \quad & \Theta_i \boldsymbol{U}_i(k) = \boldsymbol{\theta}_i : \boldsymbol{\lambda}_i \end{aligned} \tag{7}$$

If we take reference tracking, for instance, we have:

$$\begin{aligned} H_i &= \mathcal{D}_i^T \bar{Q}_i \mathcal{D}_i + \bar{R}_i \\ \boldsymbol{f}_i(k) &= \mathcal{D}_i^T \bar{Q}_i (\mathcal{M}_i \boldsymbol{x}_i(k) - \boldsymbol{W}_i(k)) \end{aligned} \tag{8}$$

The input and setpoint predictions for times k to $k+N_p$ calculated in time k are adequately stacked in vectors $\boldsymbol{U}_i(k)$ and $\boldsymbol{W}_i(k)$. $\mathcal{M}_i$ and $\mathcal{D}_i$ are the prediction matrices of the MPC. $\bar{Q}_i$, $\bar{R}_i$, and Θ_i are block diagonal matrices built repeating N_p times Q_i, R_i, and Γ_i respectively.

Notice that the matrices H_i are not only symmetric positive definite, but they are also time-invariant, unlike the $\boldsymbol{f}_i(k)$, which depend on $x_i(k)$ and $\boldsymbol{W}_i(k)$.

Observe that since $J_i(\boldsymbol{\theta}_i)$ is quadratic, we can get an explicit solution for its dual variables $\boldsymbol{\lambda}_i$, which are affine with respect to $\boldsymbol{\theta}_i$:

$$\boldsymbol{\lambda}_i = -P_i\boldsymbol{\theta}_i - \boldsymbol{s}_i(k) \tag{9}$$

where $P_i = \left(\Theta_i H_i^{-1}\Theta_i^{\mathrm{T}}\right)^{-1}$ and $\boldsymbol{s}_i(k) = P_i\Theta_i H_i^{-1}\boldsymbol{f}_i(k)$. We can observe that P_i are symmetric and depend only on Θ_i and H_i, which are time-invariant.

B. Detection and mitigation

In this secure scheme, the exchange between coordinator and agents is divided into two parts: first, to detect any misbehavior, and second, the negotiation itself, which limits the effects of eventual attacks.

Assumption 1: $\gamma_i(\cdot)$ is the same during the negotiation phase for a given time k (it does not depend on p).

Assumption 2: We suppose the agent chooses a linear function such as

$$\tilde{\boldsymbol{\lambda}}_i = \gamma_i(\boldsymbol{\lambda}_i) = T_i(k)\boldsymbol{\lambda}_i = -T_i(k)P_i\boldsymbol{\theta}_i - T_i(k)\boldsymbol{s}_i(k), \tag{10}$$

and we define $\tilde{P}_i(k) = T_i(k)P_i$ and $\tilde{\boldsymbol{s}}_i(k) = T_i(k)\boldsymbol{s}_i(k)$.

Given that P_i does not change from time to time, we can use the relation between $\boldsymbol{\theta}_i$ and $\boldsymbol{\lambda}_i$, shown in (9), to find estimates $\widehat{\tilde{P}}_i(k)$ such as:

$$\tilde{\boldsymbol{\lambda}}_i = \gamma_i(\boldsymbol{\lambda}_i(\boldsymbol{\theta}_i)) = -\widehat{\tilde{P}}_i(k)\boldsymbol{\theta}_i - \widehat{\tilde{\boldsymbol{s}}}_i(k) \tag{11}$$

Remark 1: If the estimation does not converge, necessarily there has been a change in P_i since the relation between $\boldsymbol{\lambda}_i$ and $\boldsymbol{\theta}_i$ has ceased to be affine.

If we estimate $\widehat{\tilde{P}}_i(k)$ for two different times k and they differ, then there has been a change in behavior in agent i.

Assumption 3: We have access to the nominal value of P_i, denoted $\bar{P}_i$, from reliable attack-free historical data.

Using this strategy, we can detect a deviation from nominal behavior using $E_i(k) = \|\widehat{\tilde{P}}_i(k) - \bar{P}_i\|_F$, where $\|\cdot\|_F$ is the Frobenius norm. Let $d_i \in \{0,1\}$ be an indicator that detects the attack in agent i. If the disturbance $E_i(k)$ respects an arbitrary bound

$$E_i(k) \leq \epsilon_P, \tag{12}$$

then $d_i = 0$, and no attack is detected. Otherwise, $d_i = 1$, and a change in behavior of agent i is detected.

If the attack is detected and we want to counteract the change in $\boldsymbol{\lambda}_i$, one strategy would be to recover $\boldsymbol{\lambda}_i$ from an inverse of $\gamma_i(\cdot)$.

Assumption 4: We suppose $\tilde{\boldsymbol{\lambda}}_i = \boldsymbol{0}$ only if $\boldsymbol{\lambda}_i = \boldsymbol{0}$, which implies $T_i(k)$ invertible.

Using these assumptions, we can try to estimate the inverse of $T_i(k)$ as in

$$\widehat{T_i(k)^{-1}} = \bar{P}_i\widehat{\tilde{P}}_i(k)^{-1}, \tag{13}$$

and from (9), we can derive a method to reconstruct $\boldsymbol{\lambda}_i$:

$$\boldsymbol{\lambda}_{i\mathrm{rec}} = \widehat{T_i(k)^{-1}}\tilde{\boldsymbol{\lambda}}_i = -\bar{P}_i\boldsymbol{\theta}_i - \widehat{T_i(k)^{-1}}\widehat{\tilde{\boldsymbol{s}}}_i(k). \tag{14}$$

Notice that we also need $\widehat{\tilde{\boldsymbol{s}}}_i(k)$ to use this reconstruction.

This reconstructed $\boldsymbol{\lambda}_{i\mathrm{rec}}$ can be used in (5). Observe that, as (14) does not depend on $\tilde{\boldsymbol{\lambda}}_i$, the rest of the negotiation process takes place without taking the attacking agent's responses into account.

In case no attack is detected, the coordinator can use the $\tilde{\boldsymbol{\lambda}}_i$ during the negotiation phase.

This mechanic of detecting and choosing which version of $\boldsymbol{\lambda}$ to use during the negotiation, corresponds to the inclusion of a supervisor for each agent (Fig. 3),

Observe in Fig. 3 that the coordinator sends $\mathring{\boldsymbol{\theta}}_i$ to the agents. These values may be the ones from the negotiation or other. The reason to send different values is discussed in the following subsection.

C. Considerations about parameter estimation

As seen, we need to estimate $\tilde{P}_i(k)$ and $\tilde{\boldsymbol{s}}_i(k)$. This estimation is achieved by the relation between $\boldsymbol{\theta}_i$ and $\boldsymbol{\lambda}_i$ shown in (9). As we suppose there is no noise in the communication, we propose to use Recursive Least Squares (RLS) with a forgetting coefficient ϕ to find simultaneously unbiased estimates of $\tilde{P}_i(k)$ and $\tilde{\boldsymbol{s}}_i(k)$.

If we try to estimate during the negotiation, the estimation will fail since consecutive values of $\boldsymbol{\lambda}_i^p$ and $\boldsymbol{\theta}_i^p$ are necessarily linearly dependent (5), and estimators become badly scaled. This fact is known and is described as low input excitation [17, §5]. As a counter-measure, to enrich the input excitation, the coordinator sends a sequence of random values of $\boldsymbol{\theta}_i$ until the estimation converges. It then resumes the typical negotiation, eventually using the mitigation mechanism if an attack is detected.

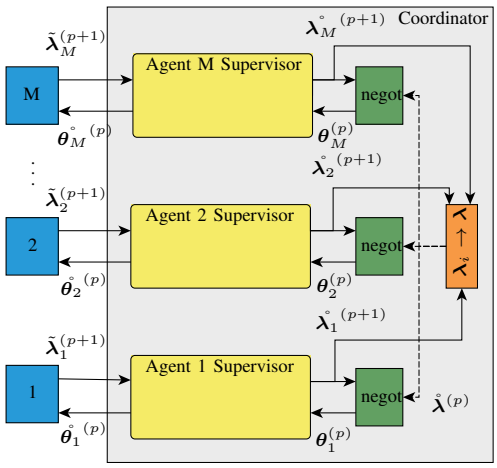

Fig. 3. Scheme for Secure DMPC

Assumption 5: Since P_i is expected to be symmetric (9), we suppose that the attacker chooses a $T_i(k)$ that does not change the structure of the resulting matrix, so it can not be discovered. In this case, we assume $\tilde{P}_i(k)$ symmetric and invertible.

As $\tilde{P}_i(k)$ is symmetric, we estimate only the upper triangle, reducing the number of estimated parameters from $\pi_{\tilde{P}_i(k)}$ to $\frac{\pi_{\tilde{P}_i(k)}+\sqrt{\pi_{\tilde{P}_i(k)}}}{2}$, and consequently the length of the estimation sequence [17].

We stack the elements of $\widehat{\tilde{P}}_i(k)$ and $\widehat{\tilde{s}}_i(k)$ estimated in a step h in vectors η_i^h. The estimation converges when $\|\eta_i^h - \eta_i^{h-1}\| \leq \epsilon$, with ϵ arbitrarily small.

D. Secure DMPC

After all the reflections about parameter estimation and the detection and mitigation mechanism, we can finally propose a secure DMPC based on the reconstruction of $\boldsymbol{\lambda}_i$.

Algorithm 2 summarizes the process used to find the optimal inputs $\boldsymbol{u}_i^\star(k|k)$ to be applied at each time k. We can see the two phases: the detection phase, where the coordinator detects if the system is attacked and by which agent. And the second phase, where the usual negotiation in algorithm 1 takes place, using different values of $\boldsymbol{\lambda}_i$ depending on if the respective agent is an attacker.

In the next section, we present an example to illustrate the performance of the mechanism.

Algorithm 2: Secure DMPC.

Detection Phase:
- $h := 0$
- **repeat**
  - Coordinator sets random $\boldsymbol{\theta}_i^{(h+1)}$
  - Subsystems solve (3a), and send $\boldsymbol{\lambda}_i^\star(\boldsymbol{\theta}^{(h)})$
  - Coordinator estimates $\widehat{\tilde{P}}_i(k)^{(h)}$ and $\widehat{\tilde{s}}_i(k)^{(h)}$
  - $h := h + 1$
- **until** $\|\eta_i^h - \eta^{h-1}\| \leq \epsilon$
- Coordinator computes d_i using (12)

Negotiation Phase:
- Coordinator initializes $\boldsymbol{\theta}^{(0)}$
- $p := 0$
- **repeat**
  - Subsystems solve (3a), and send $\boldsymbol{\lambda}_i^\star(\boldsymbol{\theta}^{(p)})$
  - Coordinator updates allocation (5) using adequate versions of $\boldsymbol{\lambda}_i$ for each agent: $\boldsymbol{\lambda}_i^\star(\boldsymbol{\theta}^{(p)})$, if $d_i = 0$ and $\boldsymbol{\lambda}_{i\text{rec}}$, if $d_i = 1$
  - $p := p + 1$
- **until** $\|\boldsymbol{\theta}^{(p)} - \boldsymbol{\theta}^{(p-1)}\| \leq \epsilon$

V. Example: Temperature Control

In this example, we want to control the temperature of 4 distinct rooms (called I, II, III, and IV) under power scarcity using quantity decomposition. The systems are modeled as continuous-time linear time-invariant systems using the 3R-2C model [18].

The state-space model of each subsystem is given by:

$$\begin{bmatrix} \dot{\boldsymbol{x}}_{Ai} \\ \dot{\boldsymbol{x}}_{Wi} \end{bmatrix} = \dot{\boldsymbol{x}}_i = A_{\mathrm{c}_i}\boldsymbol{x}_i + B_{\mathrm{c}_i}\boldsymbol{u}_i \qquad \boldsymbol{y}_i = C_{\mathrm{c}_i}\boldsymbol{x}_i \tag{15}$$

where

$$A_{\mathrm{c}_i} = \begin{bmatrix} -\frac{1}{C_{\mathrm{res}_i}Rf_i} - \frac{1}{C_{\mathrm{res}_i}Ri_i} & \frac{1}{C_{\mathrm{res}_i}Ri_i} \\ \frac{1}{Cs_iRi_i} & -\frac{1}{Cs_iRo_i} - \frac{1}{Cs_iRi_i} \end{bmatrix} \tag{16}$$
$$B_{\mathrm{c}_i} = \begin{bmatrix} \frac{10}{C_{\mathrm{res}_i}} & 0 \end{bmatrix}^{\mathrm{T}} \quad C_{\mathrm{c}_i} = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

We can see the meaning and the values of its parameters in tables I and II.

The states $\boldsymbol{x}_{Ai}$ and $\boldsymbol{x}_{Wi}$ represent the mean temperatures of the air and walls inside room i. The input $\boldsymbol{u}_i$ is the heating power for the corresponding room. The global coupling constraint is $\sum_{i=1}^{4} \boldsymbol{u}_i(k) = 4\text{kW}$.

The subsystems are discretized using the 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 5 hours:

1) Nominal behavior.
2) Agent I presents constant non-cooperative behavior $T_I(k) = 4\, I_{\sqrt{\pi_{P_I}}}$ for $k \geq 6$, without correction.
3) Agent I presents constant non-cooperative behavior $T_I(k) = 4\, I_{\sqrt{\pi_{P_I}}}$ for $k \geq 6$, with correction, $\epsilon_P = 10^{-4}$.

In Fig. 4, first, we compare the output of the agent I (air temperature in the room) with its reference (20°C), and then the decision variable $E_I(k)$ with the threshold ϵ_P. All the 3 scenarios above are represented with indices N (for nominal), S (for selfish), and C (for corrected).

Observe that in the nominal behavior, the reference w_I is not reached due to power scarcity since we deliberately set a total power not sufficient to satisfy the needs of each agent.

TABLE I
Model Parameters Meanings

Symbol	Meaning
C_{res_i}	Heat Capacity of Inside Air
Cs_i	Heat Capacity of External Walls
Rf_i	Resistance Between Inside and Outside Air (from windows)
Ri_i	Resistance Between Inside Air and Inside Walls
Ro_i	Resistance Between Outside Air and Outside Walls

TABLE II
Model Parameters Values

Symbol	I	II	III	IV	Unit
C_{res}	5	4	4.5	4.7	10^4J/K
Cs	8	7	9	6	10^4J/K
Rf	5	6	4	5	10^{-3}K/W
Ri	2.5	2.3	2	2.2	10^{-4}K/W
Ro	0.5	1	0.8	0.9	10^{-4}K/W

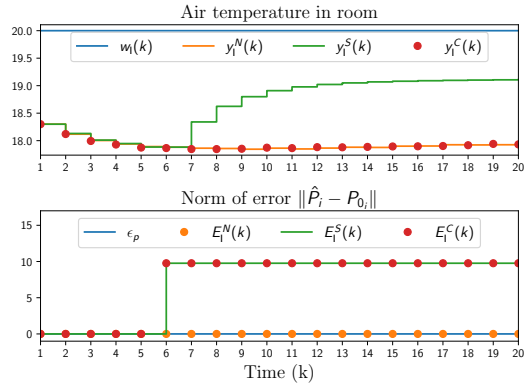

Fig. 4. Air temperature in room I and the decision variable $E_I(k)$ for different scenarios: nominal (N), with selfish behavior without correction (S) and with selfish behavior with correction (C)

TABLE III
COMPARISON OF COSTS J_i^N AND J_G^N

Agent	Nominal	Selfish	Selfish + correction
I	103	64	104
II	73	91	73
III	100	123	101
IV	132	154	131
Global	408	442	409

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, the tracking error $w_I - y_I$ is reduced but insufficient to attain the reference. In this case, the detection variable surpasses ϵ_P, $E_I^S = 9.762$, indicating the change of behavior of agent I.

When the correction is activated in the system, we see that the corrected y_I^C approaches the nominal value y_I^N, illustrating the good performances of our proposition.

We can also evaluate the performance of the proposed mechanism by comparing the local and global costs calculated using the initial cost function presented in (7) using N as the total period of simulation, $N = 20$. The same 3 scenarios are compared in table III.

As in Section III, when agent I is selfish, we see the decline of its cost at the expense of increasing all other costs. This increase in cost degrades the global objective. When the correction mechanism is activated, the differences between costs are minimal, and the global cost stays close to the nominal value, highlighting the mechanism's performance.

VI. CONCLUSION AND FUTURE WORKS

In this paper, an algorithm for monitoring and correcting exchanges between agents in a resource-sharing system has been proposed. The algorithm exploits the particular structure of exchanges, part of which must be constant over time. The first phase consists of identifying this constant part and checking if an attacker has modified it. From this identification, it is possible to reconstruct the original mechanism and find the centralized optimality. This principle should be generalized to other types of decomposition structures, and this is what we plan to do in the near future.

VII. ACKNOWLEDGMENTS

The authors would like to acknowledge C. R. Sorgho for her preliminary results.

REFERENCES

[1] E. F. Camacho and C. Bordons, *Model Predictive Controllers*. London: Springer London, 2007, pp. 13–30.

[2] J. M. Maestre, R. R. Negenborn *et al.*, *Distributed Model Predictive Control made easy*. Springer, 2014, vol. 69.

[3] R. Bourdais, H. Guéguen, and A. Belmiloudi, "Distributed Model Predictive Control for a class of hybrid system based on lagrangian relaxation," *IFAC Proceedings Volumes*, vol. 45, no. 9, pp. 46–51, 2012.

[4] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, *Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers*. Now Publishers Inc., 2011, vol. 3, no. 1. [Online]. Available: https://ieeexplore.ieee.org/document/8186925

[5] R. Paulen, S. Nazari, S. A. Shahidi, C. Sonntag, and S. Engell, "Primal and dual decomposition for distributed MPC - theory, implementation, and comparison in a SoS simulation framework," in *2016 24th Mediterranean Conference on Control and Automation (MED)*, June 2016, pp. 286–291.

[6] P. Velarde, J. M. Maestre, H. Ishii, and R. R. Negenborn, "Vulnerabilities in lagrange-based distributed model predictive control," *Optimal Control Applications and Methods*, vol. 39, no. 2, pp. 601–621, sep 2017.

[7] ——, "Scenario-based defense mechanism for distributed model predictive control," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*. IEEE, Dec 2017, pp. 6171–6176.

[8] P. Pflaum, M. Alamir, and M. Y. Lamoudi, "Comparison of a primal and a dual decomposition for distributed MPC in smart districts," in *2014 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, Nov 2014, pp. 55–60.

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

[10] G. Cohen, "Optimization by decomposition and coordination: A unified approach," *IEEE Transactions on Automatic Control*, vol. 23, no. 2, pp. 222–232, 1978.

[11] S. Braun, S. Albrecht, and S. Lucia, "Hierarchical attack identification for distributed robust nonlinear control," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 6113–6120, 2020, 21th IFAC World Congress. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405896320322916

[12] W. Ananduta, J. M. Maestre, C. Ocampo-Martinez, and H. Ishii, "Resilient distributed model predictive control for energy management of interconnected microgrids," *Optimal Control Applications and Methods*, vol. 41, no. 1, pp. 146–169, 2020.

[13] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[14] D. Simon, J. Löfberg, and T. Glad, "Reference tracking mpc using terminal set scaling," in *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, 2012, pp. 4543–4548.

[15] S. Boyd, L. Xiao, A. Mutapcic, and J. Mattingley, "Notes on decomposition methods," in *Notes for EE364B*, S. University, Ed., 2015.

[16] H. Ouyang, "Projecting onto intersections of halfspaces and hyperplanes," 2020.

[17] K. Åström and B. Wittenmark, *Adaptive Control*, ser. Addison-Wesley series in electrical and computer engineering: Control engineering. Addison-Wesley, 1989. [Online]. Available: https://books.google.fr/books?id=VJ0eAQAAIAAJ

[18] M. Gouda, S. Danaher, and C. Underwood, "Building thermal model reduction using nonlinear constrained optimization," *Building and Environment*, vol. 37, no. 12, pp. 1255 – 1265, 2002. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0360132301001214