Annual Reviews in Control 53 (2022) 338–355

Contents lists available at ScienceDirect

Annual Reviews in Control

journal homepage: www.elsevier.com/locate/arcontrol

Review article

Cyber-security in networked and distributed model predictive control

T. Arauz, P. Chanfreut, J.M. Maestre *

Department of Ingeniería de Sistemas y Automática, Universidad de Sevilla, Camino de los Descubrimientos, 41092 Sevilla, Spain

ARTICLE INFO

Keywords:
Cyber-security
Distributed control
Model predictive control
Learning

ABSTRACT

Distributed model predictive control (DMPC) schemes have become a popular choice for networked control problems. Under this approach, local controllers use a model to predict its subsystem behavior during a certain horizon so as to find the sequence of inputs that optimizes its evolution according to a given criterion. Some convenient features of this method are the explicit handling of constraints and the exchange of information between controllers to coordinate their actuation and minimize undesired mutual interactions. However, we find that schemes have been developed naively, presenting flaws and vulnerabilities that malicious entities can exploit to gain leverage in cyber-attacks. The goal of this work is to raise awareness about this issue by reviewing the vulnerabilities of DMPC methods and surveying defense mechanisms. Finally, several examples are given to indicate how these defense mechanisms can be implemented in DMPC controllers.

1. Introduction

The last years have witnessed a growing interest in distributed control methods due to their superior scalability in large-scale applications such as smart grids (Qi, Liu, & Christofides, 2011; Yazdanian & Mehrizi-Sani, 2014), water systems (Negenborn, van Overloop, Keviczky, & De Schutter, 2009), and traffic control (De Oliveira & Camponogara, 2010). This approach considers the overall system as an aggregation of smaller *pieces*, i.e., subsystems, which are locally managed by control units referred to as agents (Kordestani, Safavi, & Saif, 2021; Scattolini, 2009), whose combined decisions determine the overall performance due to the subsystems' coupling, e.g., in the control objectives and the system constraints. Moreover, this decomposition may be the only choice regarding the control architecture in applications where the implementation of a centralized controller becomes unfeasible due to the problem size or the existence of multiple independent decision-making entities.

From the multiple distributed control approaches, we are specially interested in distributed model predictive control (DMPC) (Christofides, Scattolini, de la Pena, & Liu, 2013; Negenborn & Maestre, 2014), which presents several advantages for networked control applications. In particular, predictive controllers employ a model of the system to predict its evolution over a given horizon and build an optimization problem to find the most appropriate sequence of control actions to steer its evolution according to a given criterion (Camacho & Alba, 2013). Being a computer-based approach, the method is constantly increasing the size of the problems it can handle due to the advances in information and communication technologies. In addition, it is possible to handle constraints, transport delays, and other complicating issues in an explicit fashion, which are very convenient features for industrial applications (Qin & Badgwell, 2003). Another remarkable characteristic of the DMPC family is that there are methods available to cluster controllers (Baldivieso-Monasterios & Trodden, 2021; Fele, Maestre, & Camacho, 2017; Maxim & Caruntu, 2021; Riverso, Boem, Ferrari-Trecate, & Parisini, 2016), thus providing a useful mechanism to group healthy agents. See Chanfreut, Maestre and Camacho (2021) for a survey on clustering methods where it can be seen that the interaction between controllers can be exploited to boost scalability and flexibility, and also to deal with unpredicted changes in the inter-agents communication network.

A critical issue in DMPC and any other distributed systems is that of the coordination of the agents' decisions (Rawlings & Stewart, 2008). It is well known that the controllers' attitude, which may not be willing to cooperate, and their knowledge of the overall system have a high influence on the local decisions, and hence, on the global performance (Farina & Scattolini, 2012; Mc Namara, Negenborn, De Schutter, & Lightbody, 2012; Worthmann, Kellett, Braun, Grüne, & Weller, 2015). In particular, the information available to each agent is often restricted to a local level, which implies some degree of uncertainty regarding the impact of their own and their neighbors' actions. The latter can be alleviated through communication among controllers, which allows negotiating the control actions and even attaining optimal (centralized) performance (Doan, Keviczky, & De Schutter, 2011; Giselsson, Doan, Keviczky, De Schutter, & Rantzer, 2013; Venkat, Hiskens, Rawlings, & Wright, 2008). To this end, agents interact with their physical environment, communicate local data, and update their neighboring

* Corresponding author.
E-mail address: pepemaestre@us.es (J.M. Maestre).

https://doi.org/10.1016/j.arcontrol.2021.10.005
Received 1 July 2021; Received in revised form 5 October 2021; Accepted 6 October 2021
Available online 14 November 2021

information, therefore requiring connectivity, reliability and security of their local equipment and the communication network. Regarding the latter, three different security goals are often defined (Zeldovich, 2014): confidentiality, i.e., to maintain the secrecy of the important data, integrity, i.e., to guarantee the fidelity of the data, and availability, i.e., to ensure the accessibility of the data at the right time.

Nevertheless, as stated by Peter Deutsch in the *eight fallacies of distributed computing* (Rotem-Gal-Oz, 2006), the above-mentioned requirements are likely to fail at some point in the long-term. For this reason, numerous distributed control applications that rely on a set of cyber–physical components, including hardware and software units, can become contributing factors to faults and attacks (Ding, Han, Wang, & Ge, 2019; Ding, Han, Xiang, Ge, & Zhang, 2018; Sánchez, Rotondo, Escobet, Puig, & Quevedo, 2019). On the one hand, networked systems may suffer unpredicted non-malicious faults, leading to intermittent communication and information losses. For example, Olfati-Saber and Murray (2004) and Savino et al. (2015) analyze multi-agent consensus problems where the data exchange is affected by communication delays and switching topologies; Schiffer, Dörfler, and Fridman (2017) addresses the frequency control problem in a power network operating in presence of links failures, packet losses, and delays; and Li, Bian, Li, Xu, and Wang (2020) deal with a multi-vehicle system connected by wireless channels which dynamically form and break. Also, the number and configuration of the agents may experience changes in time, as a consequence of the possible incoming and outgoing subsystems and the turning on/off of certain system processes (Riverso et al., 2016; Riverso, Farina, & Ferrari-Trecate, 2014), resulting in changes of topology that should be handled by the controller.

On the other hand, the vulnerabilities of distributed systems make them potential targets for cyber-attackers, thus introducing further security concerns. See for example Mo et al. (2011), which discusses security challenges and advances in smart grids, and emphasizes that there exists a wide variety of motivations to launch an attack in this type of applications, from economic reasons, e.g., when the attacker seeks to reduce its costs at the expense of others subsystems, to blackmailing users by controlling their access to electricity. Additionally, as pointed out in Humayed, Lin, Li, and Luo (2017), attacks may originate at different system components and propagate within the system, which renders their detection and identification difficult. In this regard, attackers can alter measurement and actuation signals, and corrupt the information transmitted through the network, hence threatening both the interactions of local controllers with their physical subsystems and with other controllers. The risks involved by the presence of attackers compromising any of these properties go beyond the system performance, because they can bring stability issues, thus highlighting the need of control structures able to detect and mitigate the effect of possible attacks. Finally, note that these treats and their harmful consequences are very real. Well-known malicious cyber-attacks such as *Stuxnet* (Kushner, 2013) and *Crash Override* (Bindra, 2017) are powerful reminders in this regard. Other common environments and situations where cyber-attacks are very present are industrial systems (Bhamare et al., 2020; Schwab & Poujol, 2018; Thames & Schaefer, 2017) and computer networks (Lavrov, Volosiuk, Pasko, Gonchar, & Kozhevnikov, 2018; Wu & Irwin, 2016). In this respect, Jang-Jaccard and Nepal (2014) also provide an overview of potential vulnerabilities in hardware, software, and network systems, while discussing new threats and attacks patterns associated with emerging technologies such as social media and cloud computing.

For all the above-mentioned reasons, cyber-security has become a recent field of interest for DMPC, and different algorithms have already been developed to deal with cyber-attacks within this framework. For example, Velarde, Maestre, Ishii, and Negenborn (2018) consider insider attacks and present a resilient DMPC negotiation procedure. Similarly, Ananduta, Maestre, Ocampo-Martinez, and Ishii (2020) propose an active method to deal with adversarial agents within the DMPC algorithm. Attacks on communication channels have also been addressed. For instance, Liu and Bai (2018) presented an iterative DMPC where data injection attacks are considered.

Despite their sources of vulnerability, DMPC schemes also possess some unique features to deal with cyber-attacks. To begin with, robust MPC formulations can enhance safety margins even in case of cyber-attacks, e.g., the tree-based MPC presented by Pierron, Arauz, Maestre, Cetinkaya, and Maniu (2020) computes the input sequence considering all possible jamming attacks scenarios. Likewise, the calculation of a sequence of inputs can also be used to mitigate issues such as losses in the information packages exchanged between controllers and actuators (Quevedo & Nešić, 2010; Sun, Zhang, & Shi, 2019). Another relevant feature of the MPC framework comes from the model employed to generate the controller's predictions, which establishes clear analytical relationships between the problem variables that can be used to detect anomalies, e.g., by exploiting analytical redundancy. Likewise, there is a recent bloom of predictive control methods where learning algorithms play a significant role. This feature is particularly interesting in this context because it enhances the set of detection and identification tools that agents can employ to trigger their defensive countermeasures (Chen, Wu & Christofides, 2020; Wu et al., 2018). Finally, being a computer-based approach, DMPC algorithms can be easily extended to incorporate defense mechanisms that already exist in the literature for other control methods.

The rest of the paper is organized as follows. Section 2 presents a general description of the structure of distributed systems and its underlying communication network, along with two widely used DMPC approaches. Section 3 summarizes the cyber attacks that can be found in the DMPC framework. In Section 4, different defense mechanisms that DMPC can use are outlined distinguishing between prevention, detection, and mitigation actions. Finally, concluding remarks and future research prospects are indicated in Section 5.

2. Problem setting

The overall system is generally controlled by a set of M local agents, hereafter $\mathcal{A} = \{\mathcal{A}_1, \dots, \mathcal{A}_M\}$, which correspondingly manage a set of coupled subsystems $\mathcal{S} = \{\mathcal{S}_1, \dots, \mathcal{S}_M\}$, i.e., subsystem $\mathcal{S}_i$ is assigned to control agent $\mathcal{A}_i$, for all $i \in [1, \dots, M]$. In this regard, it is assumed that $\mathcal{A}_i$ measures the local variables and manages the inputs of subsystem $\mathcal{S}_i$. In turn, the local controllers in $\mathcal{A}$ are interconnected by a data network that allows them to communicate and perform coordinated tasks. As shown in Fig. 1, the overall structure can be modeled as the combination of two graphs, one associated with the network of the agents, i.e., $\mathcal{G}_{\mathrm{a}} = (\mathcal{A}, \mathcal{E}_{\mathrm{a}})$, another with the system dynamics, i.e., $\mathcal{G}_{\mathrm{s}} = (\mathcal{S}, \mathcal{E}_{\mathrm{s}})$. The edges in $\mathcal{E}_{\mathrm{a}}$ represent communication links that allow the agents to exchange data and coordinate their actuation, whereas the edges in $\mathcal{E}_{\mathrm{s}}$ model the coupling between subsystems. That is, an edge in $\mathcal{E}_{\mathrm{s}}$ from subsystems $\mathcal{S}_j$ to $\mathcal{S}_i$ is associated with the coupling effect that j has on i, i.e., $A_{ij}x_j + B_{ij}u_j$. Therefore, whenever sets $\mathcal{E}_{\mathrm{a}}$ and $\mathcal{E}_{\mathrm{s}}$ do not coincide, not all coupled controllers can communicate, which restricts the coordination capacity of the distributed system. Finally, notice that we have implicitly assumed that there exist additional communication connections that allow the agents to acquire information from the system and to send commands to the corresponding actuators.

The system dynamics are modeled mathematically to predict the overall behavior as an aggregation of the subsystems' evolution. In particular, for state-space representations, the dynamics of each $\mathcal{S}_i$ are commonly described by a model of the form:

$$x_i(k+1) = f_i\left(x_i(k), u_i(k), [x_j(k), u_j(k)]_{j\in\mathcal{N}_i}, d_i(k)\right), \tag{1}$$

where $x_i(k)$, $u_i(k)$ denote respectively the state, and input of subsystem i, $x_i(k+1)$ is its successor state, $d_i(k)$ represents the external state-disturbances on $\mathcal{S}_i$, and $f_i(\cdot)$ is a function defined accordingly.[1]

[1] Note that $\mathcal{S}_i$ refers to subsystem i and, analogously, $\mathcal{A}_i$ denotes agent i.

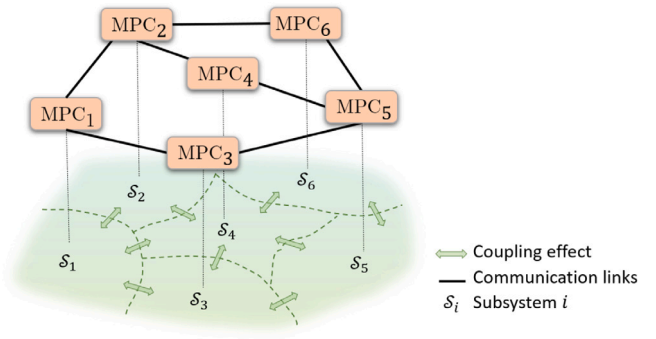

Fig. 1. Scheme of a distributed system composed of 6 subsystems that are assigned to a set of distributed MPC agents. The network of agents, whose links are represented in black solid lines, is modeled by graph $\mathcal{G}_a = (\mathcal{A}, \mathcal{E}_a)$, while the subsystems interactions can be described by another graph $\mathcal{G}_s = (\mathcal{S}, \mathcal{E}_s)$, where the edges in $\mathcal{E}_s$ connect dynamically coupled subsystems. In addition, the communication connections allowing the interaction between agents and their corresponding subsystems are indicated with dotted lines.

Likewise, $\mathcal{N}_i$ contains the set of neighbors whose states and inputs affect the dynamics of subsystem i. In case of linear representations, model (1) can be rewritten as

$$\begin{aligned} x_i(k+1) &= A_{ii}x_i(k) + B_{ii}u_i(k) + w_i(k), \\ w_i(k) &= \sum_{j\in\mathcal{N}_i} \left[A_{ij}x_j(k) + B_{ij}u_j(k)\right] + d_i(k), \end{aligned} \tag{2}$$

where w_i is a vector of *disturbances* that captures both the possible external perturbations and the coupling effect of neighboring subsystems, i.e., $\sum_{j\in\mathcal{N}_i}\left[A_{ij}x_j(k) + B_{ij}u_j(k)\right]$, where $\mathcal{N}_i = \{j \in \mathcal{S}\setminus\{i\} \mid [A_{ij}, B_{ij}] \neq \mathbf{0}\}$.

However, system states may be unavailable for controllers, requiring the presence of observers to provide an estimate of the state from the system outputs. Hence, system dynamics may also include an extra equation:

$$y(k) = Cx(k), \tag{3}$$

that relates the measurement received by the controller from the sensors $y(k)$ with system state $x(k)$. In addition, as will be seen in Sections 3 and 4 , these observers can become the target of cyber-attackers (e.g., Sahoo, Mishra, Peng, & Dragičević, 2018), but also provide means to detect them (e.g. Barboni, Rezaee, Boem, & Parisini, 2020). Nevertheless, it can also be assumed that $C = I$ (identity matrix of corresponding dimensions) in (3), meaning that the system states can be perfectly measured, i.e., $y(k) = x(k)$, so that observers are not required. Indeed, this assumption is followed in this work to provide a clearer exposition of the problem, but notice that results can be directly extended to the general case just including the extra equation (3) and the corresponding observer.

2.1. DMPC algorithms

Predictive controllers repeatedly solve a constrained optimization problem based on a model of the system to find the sequence of inputs that minimizes a performance index (Camacho & Alba, 2013). Also, the agents can handle different constraints and objectives while accessing to different sets of data (Christofides et al., 2013; Negenborn & Maestre, 2014). For simplicity, let us consider that the overall objective $J(x, \mathbf{u})$ adds subsystem's goals regarding their state and input trajectories, e.g.,

$$J(x,\mathbf{u}) = \sum_{i\in[1,M]} \sum_{n=0}^{N_h-1} \left(\|x_i(n) - x_{\text{ref},i}\|^2_{Q_i} + \|u_i(n)\|^2_{R_i} \right), \tag{4}$$

where $x = [x_i]_{i\in[1,M]}$ represents the global state, N_h is the prediction horizon, $x_{\text{ref},i}$ denotes the state reference for subsystem i, and $Q_i \geq 0$ and $R_i > 0$ are weighing matrices. Additionally, $\mathbf{u}$ represents the sequence of global inputs for a prediction horizon of N_h time steps, i.e., $\mathbf{u} = [\mathbf{u}]_{i\in[1,M]}$, with $\mathbf{u}_i = [u_i^T(0), u_i^T(1), \dots, u_i^T(N_h-1)]^T$. Also, let us assume the centralized MPC problem is given by

$$\begin{aligned} &\min_{\mathbf{u}} J(x,\mathbf{u}) \\ &\text{s.t.} \quad C_{in}\mathbf{u} \leq c_{in}, && \text{(5a)} \\ &\qquad C_{eq}\mathbf{u} = c_{eq}, && \text{(5b)} \end{aligned}$$

where C_{in}, C_{eq}, c_{in}, and c_{eq} are respectively matrices and vectors defining affine constraints on the optimization variable $\mathbf{u}$. Note (5a) and (5b) respectively include inequality and equality constraints that can bound both the inputs and the states, possibly coupling the local inputs trajectories $\mathbf{u}_i$.

For their further use throughout this survey, this subsection introduces two popular distributed MPC algorithms, which attain the optimal centralized MPC solution by performing iterative distributed negotiations.

2.1.1. Dual decomposition

Dual decomposition methods separate global optimization problems into smaller components by using Lagrange multipliers to enforce the coupling constraints satisfaction (Boyd, Parikh, & Chu, 2011; Cheng, Forbes, & Yip, 2007; Rantzer, 2009; Zhu & Martínez, 2015). Let us consider a simplification of problem (5), i.e.,

$$\begin{aligned} &\min_{[\mathbf{u}_i]_{i\in[1,M]}} \sum_{i\in[1,M]} J_i(x_i,\mathbf{u}_i) \\ &\text{s.t.} \quad C_{in}\mathbf{u} = \sum_{i\in[1,M]} C_{i,in}\mathbf{u}_i \leq c_{in}, && \text{(6a)} \\ &\qquad C_{eq}\mathbf{u} = \sum_{i\in[1,M]} C_{i,eq}\mathbf{u}_i = c_{eq}, && \text{(6b)} \end{aligned}$$

where the objective function is separable. Notice that constraints (5a) and (5b) have been rewritten as a summation on variables $\mathbf{u}_i$, being $C_{i,in}$ and $C_{i,eq}$ matrices computed accordingly. By forming the Lagrangian of problem (6), i.e.,

$$\begin{aligned} &L(x,\mathbf{u},\lambda) = \\ &\sum_{i\in[1,M]} J_i(x_i,\mathbf{u}_i) + \mu\left(\sum_{i\in[1,M]} C_{i,in}\mathbf{u}_i - c_{in}\right) + \lambda\left(\sum_{i\in[1,M]} C_{i,eq}\mathbf{u}_i - c_{eq}\right), \end{aligned} \tag{7}$$

where $\mu \geq 0$ and λ are Lagrange multipliers, constraints (6a) and (6b) can be relaxed, allowing for the distribution of the problem among the set of local agents. In this regard, sub-gradient based methods where the agents iteratively optimize inputs sequences $\mathbf{u}_i$ and exchange their solutions to update μ and λ are extensively used. In particular, let superscript p denote any iteration, and consider some initial prices λ^0 and μ^0. Then, at each p, any agent i computes sequence $\mathbf{u}_i^p$ by solving the following local problem

$$\mathbf{u}_i^p = \arg\min_{\mathbf{u}_i} J_i(x_i,\mathbf{u}_i) + \mu^p C_{i,in}\mathbf{u}_i + \lambda^p C_{i,eq}\mathbf{u}_i \tag{8}$$

Subsequently, the Lagrange multipliers are updated in the direction of the sub-gradient of the dual problem, i.e.,

$$\begin{aligned} \mu^{p+1} &= \mu^p + \gamma_{eq}^p \left(\sum_{i\in[1,M]} C_{i,eq}\mathbf{u}_i - c_{eq}\right), \\ \lambda^{p+1} &= \max\left(\mathbf{0},\ \lambda^p + \gamma_{in}^p \left(\sum_{i\in[1,M]} C_{i,in}\mathbf{u}_i - c_{in}\right)\right), \end{aligned} \tag{9}$$

where $\gamma_{in}^p, \gamma_{eq}^p > 0$ are the step sizes of the pth iteration. Note that outcome of this procedure relies on continuous and reliable communication among agents, since the variables associated with the coupling constraints need to be shared to evaluate (9).

Considering the above, dual decomposition methods can be used to coordinate a set of agents with coupled dynamics (Farokhi, Shames,

& Johansson, 2014; Ma, Anderson, & Borrelli, 2011). In particular, consider a set of input-coupled linear agents that seek to solve (5), with $J(x,\mathbf{u})$ being the quadratic index in (4). Also, notice that, in this case, (4) admits the following decomposition:

$$J(x,\mathbf{u}) = \sum_{i\in[1,M]} J_i(x_i,\mathbf{u}_i,[\mathbf{u}_j]_{j\in\mathcal{N}_i}). \tag{10}$$

That is, there may be *shared* optimization variables that prevents the objective function from being separable as in (6), e.g., $\mathbf{u}_j$ for $j\in\mathcal{N}_i$ is both present in $J_i(\cdot)$ and $J_j(\cdot)$. To this end, assume that any agent i optimizes an augmented vector $\mathbf{u}_i^{\mathrm{a}} = \left[\mathbf{u}_i;[\mathbf{u}_j^i]_{j\in\mathcal{N}_i}\right]$, where $[\mathbf{u}_j^i]_{j\in\mathcal{N}_i}$ represents a local duplicate of sequences $[\mathbf{u}_j]_{j\in\mathcal{N}_i}$. Overall coordination is achieved if all agents reach a consensus on their corresponding coupled inputs, which is translated into a set of equality constraints

$$\mathbf{u}_i - \mathbf{u}_i^j = 0, \quad \forall i\in[1,M], j\in\mathcal{N}_i. \tag{11}$$

Since (5a), (5a) and (11) can be rewritten as inequality and equality constraints on variables $\mathbf{u}_i^{\mathrm{a}}$, i.e., $\sum_{i\in[1,M]} C_{i,\mathrm{in}}^{\mathrm{a}}\mathbf{u}_i^{\mathrm{a}} \le 0$, $\sum_{i\in[1,M]} C_{i,\mathrm{eq}}^{\mathrm{a}}\mathbf{u}_i^{\mathrm{a}} = 0$, and the objective (10) equals $J(x,\mathbf{u}) = \sum_{i\in[1,M]} J_i(x_i,\mathbf{u}_i^{\mathrm{a}})$, the overall optimization can be formulated as the class of problems (6). In particular, each agent i would iteratively optimize sequence $\mathbf{u}_i^{\mathrm{a}}$, and the Lagrange-prices update equations become functions on the augmented vectors of all the agents. The update of the Lagrange prices can be performed by a central coordinator, which transmits the new values to the agents for the inputs' optimization. Note that in this case the optimization problems still need to be solved in parallel by the agents at a local level; hence, the algorithm implementation retains a significant distributed nature. Likewise, this update can also be performed directly by the local agents as long as each one knows the neighbors that it needs to coordinate with, i.e., starting from common λ^0 and μ^0, all agents could iteratively compute (9) after receiving all their neighbors' optimized input sequences, and then use the resulting prices in their optimization problems.

This approach has been widely used in the literature and it is possible to find several relevant modifications with respect to the formulation presented above. For example, an augmented Lagrangian approach is employed by Mc Namara, Negenborn, De Schutter, Lightbody, and McLoone (2016) for frequency regulation in power grids. Also, authors as Hammami, Maraoui, and Bouzrara (2020) have extended the method to consider nonlinear dynamics. Finally, other authors like Giselsson et al. (2013) explore how to improve the convergence rate by using accelerated gradient methods.

2.1.2. Cooperation-based MPC

In cooperation-based MPC (Stewart, Venkat, Rawlings, Wright, & Pannocchia, 2010; Venkat et al., 2008), centralized behavior is attained by the use of a plant-wide performance function as control objective for all the agents. That is, all agents $\mathcal{A}_i$ optimize their inputs sequences $\mathbf{u}_i$ so as to minimize global index $J(x,\mathbf{u})$, thus taking into account the effect of the local actions on the neighboring subsystems. As described in Stewart et al. (2010) and Venkat et al. (2008), the distributed optimization is of the Gauss–Jacobi type and is performed through an iterative procedure where the controllers adapt their local input sequences to the expected neighboring actions. In particular, consider problem (5) and note that the objective function can be rewritten as $J(x,\mathbf{u}_i,[\mathbf{u}_j]_{j\in[1,M]\setminus\{i\}})$ by simply considering the definition of $\mathbf{u}$, i.e., $\mathbf{u} = [\mathbf{u}_i]_{i\in[1,M]}$. Then, at each iteration p of the distributed optimization, each agent i solves the following problem

$$\mathbf{u}_i^* = \arg\min_{\mathbf{u}_i} \; J(x,\mathbf{u}_i,[\mathbf{u}_j]_{j\in[1,M]\setminus\{i\}}^{p-1})$$

$$\text{s.t.} \;\; C_{i,\mathrm{in}}\mathbf{u}_i + \sum_{j\in[1,M]\setminus\{i\}} C_{j,\mathrm{in}}\mathbf{u}_j^{p-1} \le c_{\mathrm{in}}, \tag{12a}$$

$$C_{i,\mathrm{eq}}\mathbf{u}_i + \sum_{j\in[1,M]\setminus\{i\}} C_{j,\mathrm{eq}}\mathbf{u}_j^{p-1} = c_{\mathrm{eq}}, \tag{12b}$$

where neighboring inputs obtained at iteration $p-1$ are considered as constant parameters, i.e., the only variable in (12) is the local input sequence $\mathbf{u}_i$, which optimizes the global evolution. Notice that the most recent values of neighboring inputs available for each agent correspond to the previous iteration step solutions $\mathbf{u}_j^{p-1}$ because all agents compute $\mathbf{u}_i^*$ simultaneously. Once all agents have solved (12), the solution of the $(p+1)$th iteration is obtained as a convex combination of $\mathbf{u}_i^*$ and $\mathbf{u}_i^p$, i.e.,

$$\mathbf{u}_i^{p+1} = w_i\mathbf{u}_i^* + (1-w_i)\mathbf{u}_i^p, \tag{13}$$

where $0 < w_i \le 1$ for all $i\in[1,M]$, and $\sum_{i\in[1,M]} w_i = 1$. Subsequently, the intended local actions $\mathbf{u}_i^{p+1}$ are shared, and the process is iteratively repeated until convergence is attained, or until a maximum number of iterations is reached.

See Venkat et al. (2008) for a detailed presentation of the above-described algorithm and for illustrative results on a power system application. Additionally, Stewart, Wright, and Rawlings (2011) extend this optimization method to nonlinear systems, and in Ferramosca, Limón, Alvarado, and Camacho (2013) it is used to implement in a distributed manner the MPC for tracking described in Limón, Alvarado, Alamo, and Camacho (2008). Other application examples are given in Subramanian, Rawlings, Maravelias, Flores-Cerrillo, and Megan (2013), which exploits the results in Stewart et al. (2011) for supply chain optimization; and in Jia, Meng, Wu, Sun, and Dong (2020), which introduces a cooperative economic MPC inspired in Venkat et al. (2008) for optimal load frequency control in large-scale power networks.

2.2. Main elements of DMPC approaches and cyber-security

Distributed predictive controllers comprise different elements, which also represent sources of vulnerability. The main ones are summarized as follows:

- *Vulnerabilities in the problem formulation:* here, we list weak spots in the problem setup that can disrupt DMPC algorithms:
  - *Optimization function*: controllers compute input actions by optimizing the previously defined overall objective (4). Indeed, when no constraints are active, it is this cost function that governs the response of the controller. Therefore, any modification at a local level in the cost function may steer the negotiation and alter the resulting input sequences optimized by the agents.
  - *Constraints*: states and inputs are usually restricted either by physical limits or by safety conditions. Also, coupling constraints are also relevant in this context. These constraints define the domain of the feasible input sequences and therefore any manipulation in this regard has the potential to modify the result of the distributed optimization. Likewise, the convergence of the algorithm can be endangered if agents perform local optimizations using incompatible sets of constraints.
  - *Negotiation process between agents*: agents are required to exchange information following a certain negotiation process. This process is also a vulnerable spot of the system. In particular, attackers can use the knowledge of the negotiation process to gain advantage with respect to agents that are complying with the DMPC algorithm.
- *Vulnerabilities in the communication process:* two main weak spots can be identified regarding the process of data exchange:
  - *Communication network*: distributed agents need to exchange information to coordinate their input sequences. Likewise, controllers can use a network to send input values and receive measurements. This is a clear source of vulnerability and the target of cyber-threats such as replay attacks and jamming attack.

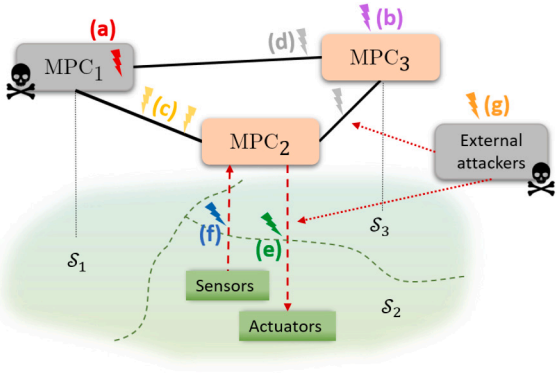

Fig. 2. Representation of the vulnerabilities to cyber-attacks in a distributed system with 3 MPC agents. Inter-controller communication is threatened by malicious and infected agents ((a) and (b)) that may corrupt the shared data and perform disruptive and/or deceptive actions ((c) and (d)). Likewise, the interactions of the network of agents with their physical subsystems can be attacked ((e) and (f)), thus affecting the control decisions and the system performance. Also, the system can suffer the effects of external attackers that may jeopardize both the inter-agent communication and the local operations (g). For the sake of clarity, controllers' interactions with sensors and actuators are only represented for agent 2.

  - *Protocols*: inter agent communication is carried out following a common protocol. Again, this is another weak spot that can be exploited, disrupting normal communications, and hence, the operation of the system.

- *Vulnerabilities in the physical components*: sensors, actuators, and even the controller itself are vulnerable elements, because they can be physically manipulated. Moreover, their normal operation can also be disrupted via software, e.g., by viruses.

Finally, note that cyber-attacks can cause different consequences as their effect spread: performance degradation, violation of the constraints, loss of recursive feasibility, etc. Therefore, in this study, we focus on the classification of the attack types rather than on their consequences.

3. Cyber attacks in DMPC

Many DMPC methods have been developed, but insufficient attention has been paid to their vulnerabilities, e.g., against malicious agents and external attacks. The aim of this section is to survey different types of cyber-attacks considering their corresponding source of vulnerability from a physical viewpoint. Fig. 2 illustrates this criterion and clearly shows different weak spots for DMPC algorithms in the control infrastructure. Special attention is paid to communication links because they are required for inter-agent communication and they may be used as well by local controllers to exchange information with sensors and actuators. Other vulnerable spots are the software that runs in the agents, which may be corrupted due to viruses and malicious software, and the hardware of the controllers and the instrumentation. Nevertheless, we consider that the last case is out of the scope of this work because attacks to hardware require the physical presence of the attacker, and therefore they can be considered acts of sabotage rather than cyber-attacks. Also, their consequences are similar to the attacks considered in the other categories. Therefore, we will focus on the malicious behavior that stems from software and communication links.

3.1. The communication network

The communication network is required to share information between different agents and system components, representing a significant source of vulnerability. In particular, the channel attacked can be inter-agent (for coordination purposes as (c) in Fig. 2) and intra-agent (to exchange data between the controller and its actuators and sensors such as (e) and (f) in Fig. 2). While there are many examples of intra-agent attacks in the MPC literature, e.g., Qin et al. (2020), Sun and Yang (2019) and Wang, Gao and Qiu (2016) ((e) in Fig. 2) and Chen, Wu et al. (2020), Liu and Bai (2018) and Yang et al. (2019) ((f) in Fig. 2), inter-agent attacks are less common and may appear presented as insider attacks (Feng & Ishii, 2020; Kikuchi, Cetinkaya, Hayakawa, & Ishii, 2017).

Also, most cyber-attacks in communication networks can be categorized depending on the effect produced in the system (Dibaji et al., 2019; Teixeira, Shames, Sandberg, & Johansson, 2015). In the first place, there are *deception attacks*, which corrupt the data transmitted through the communication channel, e.g., false-data injection and replay attacks. The second category is that of *disruption attacks*, which interrupt the regular network operation either by blocking the channel or by capturing the signals, e.g., denial of service (DoS) and jamming attacks. These attacks compromise different security goals: integrity is violated by deception attacks, and availability by disruption attacks. However, note that these two categories are not mutually exclusive, i.e., attacks can have multiple consequences and be included in both categories.

Deception and disruption attacks are presented below along with the corresponding attack model definitions for each of them and examples of the literature. In addition, some MPC works are summarized in Table 1 and classified according to this criterion.

Remark 1. Another type of malicious behavior is that of eavesdropping, for it compromises the confidentiality in the communication channel without altering the data. This category has been omitted because eavesdropping does not affect system performance.

3.1.1. Deception attacks

Deception attacks modify the data transferred via the communication network and are also known as *false data injection (FDI) attacks* (e.g., Barboni et al., 2020; Braun et al., 2020; Kushal, Lai, & Illindala, 2018; Li, Zhou, Li, Li, & Lu, 2019; Tian & Peng, 2020; Wu et al., 2018). They can also be distinguished considering the signal corrupted, e.g., state and input signals.

Regardless of the signal attacked, deception attacks can be modeled as (Chamanbaz et al., 2019; Dibaji et al., 2019; Gallo, Turan, Boem, Parisini, & Ferrari-Trecate, 2020):

$$g_{\text{DecAtt}}(k) = g(k) + \gamma(k) a_g(k) \tag{14}$$

where $g(k)$ represents either the state ($x(k)$) or the input ($u(k)$), and $g_{\text{DecAtt}}(k)$ is the complete signal including the data injected by the attacker, i.e., $x_{\text{DecAtt}}(k)$ or $u_{\text{DecAtt}}(k)$. Here, $a_g(k)$ is the data injected from the attacker ($a_x(k)$ or $a_u(k)$), and $\gamma(k) = \{0, 1\}$ denotes if an attack occurs ($\gamma(k) = 1$) or not ($\gamma(k) = 0$). The precise definition of γ leads to slightly different models. For example, $\gamma(k)$ can be a Bernoulli-distributed white sequence (Wang, Song, Liu & Zhang, 2016), and it can be extended in a diagonal matrix $\Gamma(k) = diag\{\gamma_1(k), \gamma_2(k), \dots \gamma_m(k)\}$ to model a different effect for each signal element in case of attack (Qin et al., 2020). Finally, some studies employ a simpler model that ignores the probability of occurrence of the attacks (Braun et al., 2020; Chamanbaz et al., 2019), so that model (14) becomes:

$$g_{\text{DecAtt}}(k) = \begin{cases} g(k) & \text{In case of no attack,} \\ g_a(k) \quad (\text{or} \neq g(k)) & \text{In case of attack.} \end{cases} \tag{15}$$

Some examples of state signals being attacked ((f) in Fig. 2) are given by Chen, Wu et al. (2020) and Wu et al. (2018), in which the closed-loop system is destabilized via cyber-attacks on sensor measurements.

Table 1
Classification of MPC-based schemes working in presence of different types of cyber-attacks and faults. Note that the letters between brackets in the right column refers to the attacks illustrated in Fig. 2.

Insider attacks			Ananduta, Maestre, Ocampo-Martinez, and Ishii (2018, 2019), Ananduta et al. (2020), Chanfreut, Maestre, and Ishii (2018), Maestre, Trodden, and Ishii (2018), Maestre, Velarde, Ishii, and Negenborn (2021), Tanaka and Gupta (2016), Tiwari et al. (2017), Velarde, Maestre, Ishii, and Negenborn (2017), Velarde et al. (2018), Wang and Ishii (2019)	(a), (b)
Communication attacks	Deception	Input signal	Barboni, Boem, and Parisini (2018), Braun, Albrecht, and Lucia (2020), Chamanbaz, Dabbene, and Bouffanais (2019), Franze, Lucia, and Tedesco (2021), Qin, Zhao, Huang, Tian, and Zhou (2020), Xu, Yuan, Yang, and Zhou (2021)	(e)
		Measurements	Abdelwahab, Lucia, and Youssef (2020), Bagherzadeh and Lucia (2019), Chamanbaz et al. (2019), Chen, Wu et al. (2020), Chen, Zhang, Ni and Wang (2020), Franze et al. (2021), Franzè, Tedesco, and Famularo (2020), Franzè, Tedesco, and Lucia (2019), Liu and Bai (2018), Liu, Chen and Li (2020), Liu, Chen, Li and Wan (2020), Liu, Song, Wei, and Huang (2017), Wang, Song, Liu and Zhang (2016), Wu et al. (2018), Xu et al. (2021)	(f)
	Disruption	Input signal	Liu, Wang and Geng (2020), Lješnjanin, Quevedo, and Nešić (2014), Mishra, Chatterjee, and Quevedo (2017), Mishra, Quevedo, and Chatterjee (2016), Pierron et al. (2020), Qiu, Yang, and Zhu (2021), Quevedo, Mishra, Findeisen, and Chatterjee (2015), Quevedo and Nešić (2010, 2012), Sun and Yang (2019), Sun et al. (2019), Wang, Gao and Qiu (2016)	(e)
		Measurements	Qiu et al. (2021), Quevedo and Ahlén (2008), Yang, Li, Dai, and Xia (2019)	(f)
Other system faults			Boem, Gallo, Raimondo, and Parisini (2019), Boem, Riverso, Ferrari-Trecate, and Parisini (2018), Ferranti, Wan, and Keviczky (2019), Jiang and Yu (2012), MacGregor and Cinar (2012), Moradmand, Ramezani, Nezhad, and Sardashti (2019), Naghavi, Safavi, and Kazerooni (2014), Raimondo, Marseglia, Braatz, and Scott (2013), Riverso et al. (2016), Zafra-Cabeza, Maestre, Ridao, Camacho, and Sánchez (2011), Zarei, Gupta, Ramirez, and Martinez-Rodrigo (2019), Zhang, Xie, and Lian (2020)	

Also, Wang, Song, Liu and Zhang (2016) assume that the signal received by the controller is corrupted by randomly occurring deception attacks. On the other hand, similar attacks against input signal (type (e) of Fig. 2) are reported by Braun et al. (2020) and Qin et al. (2020). Finally, attacks to state and input signals are considered by Chamanbaz et al. (2019), who only assumes that both types of attack cannot happen simultaneously.

Replay attacks can be considered as a particular case of deception attacks because they also alter the data. In these attacks, there is an initial period where the attacker gains access to a system component to record data during a certain amount of time (Yaghooti, Romagnoli, & Sinopoli, 2021). Next, the attack is carried out by injecting the previously stored data in the communication channel (Mo & Sinopoli, 2009). Thus, data confidentiality and integrity are compromised. These attacks can also be modeled using (14) and (15) considering the attacked data equals to previous values. For example, Franzè et al. (2020) consider replay attacks on state measurements where a state value sequence is recorded during a time interval.

Other important types of deception attacks are the so-called *covert* and *stealth* attacks (Ferrari & Teixeira, 2020; Pasqualetti, Dörfler, & Bullo, 2013; Sánchez et al., 2019). The former type has been studied by authors like de Sá, da Costa Carmo, and Machado (2017), and requires to perform a system identification to gain knowledge about the target system, e.g., by eavesdropping, in order to attack using FDI. For example, as described by Smith (2011), the covert attacker can inject false data on the control commands received by the actuators and simultaneously modify the system output feedback in order to make the attack undetectable for the controller. Similarly, in stealth attacks, the attacker corrupts the system measurements using data compatible with the system equations to avoid the triggering of detection mechanisms (Dán & Sandberg, 2010; Pasqualetti et al., 2013). See for example the work of Sahoo et al. (2018), who introduces a cooperative strategy for distributed microgrids applications working in presence of stealthy attackers. Stealth attacks also depend on the information the attackers have on the system to mimic its signals. In this regard, besides using a model of the system as in Smith (2011), attackers can intercept the measurements and control signals and use deep learning techniques to generate stealthy attacks able to deceive the anomaly detectors (Feng, Li, Zhu, & Chana, 2017).

3.1.2. Disruption attacks

Disruption attacks disturb the system performance by blocking or capturing signals, e.g., by performing DoS and *jamming attacks* (Cetinkaya, Ishii, & Hayakawa, 2019; Peng & Sun, 2020; Pierron et al., 2020; Wakaiki, Cetinkaya, & Ishii, 2019; Wang, Gao & Qiu, 2016; Xiao, Ge, Han, & Zhang, 2020).

Similarly to deception attacks, disruption attacks can be classified according to the signal attacked. However, the attack model can always be defined as:

$$g_{\text{DisAtt}}(k) = \begin{cases} g(k) & \text{In case of no attack,} \\ 0 & \text{In case of attack,} \end{cases} \tag{16}$$

where $g(k)$ represents state ($x(k)$) or input ($u(k)$), and $g_{\text{DisAtt}}(k)$ is the complete signal (i.e., $x_{\text{DisAtt}}(k)$ or $u_{\text{DisAtt}}(k)$).

Another possibility for defining the attack model is to consider a Boolean/binary indicator $\nu(k)$ for attacks:

$$g_{\text{DisAtt}}(k) = \nu(k)g(k), \tag{17}$$

where $\nu(k)$ can be formulated as a stochastic process where $\nu(k) = 0$ in case of attack and $\nu(k) = 1$ otherwise. For instance, Sun et al. (2019) use this approach to indicate the status of the communication channel of the controller–actuator network (ON or OFF). Yang et al. (2019) also consider the probability distribution of the attacks.

The duration of these attacks has to be also taken into account. In particular, it may be convenient to consider limits on the consecutive time steps that the communication channel can be blocked (Quevedo & Nešić, 2012). For example, it can be assumed that the attacker is only capable of disrupting the channel a limited number of times due to its finite power energy, and thus the prediction horizon can be set

greater than that limit as proposed by Sun et al. (2019). Also, Feng, Cetinkaya, Ishii, Tesi, and De Persis (2020) and Sun and Yang (2019) constrain the action of the attacker in frequency. To this end, a new variable is defined to account for the starting instant of a sequence of the DoS attacks based on the work of De Persis and Tesi (2015). Sun et al. (2019) consider an attacker that launches adversarial jamming signals blocking the communication channel between the controller and the actuator ((e) in Fig. 2). Likewise, Yang et al. (2019) assume the channel between the controller and actuators to be reliable, but not the feedback channels between sensors and the controller, which are subject to DoS attacks ((f) in Fig. 2).

Additionally, note that packet losses can also be due to unreliable transmissions, and not only due to malicious attacks. Indeed, $\nu(k)$ in (17) can also be used to describe packet dropouts (Pierron et al., 2020; Quevedo & Nešić, 2010; Wang, Gao & Qiu, 2016). For example, authors as Cetinkaya, Ishii, and Hayakawa (2015) present a probabilistic characterization for packet exchange failures from random losses and jamming attacks by defining an overall packet drop ratio.

Finally, notice that these deception and disruption attacks, which compromise the interactions between the controller and the system, may equally affect DMPC schemes (Chen, Zhang et al., 2020; Yang et al., 2019). In the DMPC context, these attacks in turn lead to a threat to the reliability and availability of the information exchanged between agents, and thus to the overall coordination. For example, in Chen, Zhang et al. (2020), the authors consider a power system composed of four locally managed generation units that may suffer FDI attacks on the measurement signals and DoS attacks on the communication links between units. See also Tiwari et al. (2017), which formulates a control problem as a game between two players, one representing a reliable MPC controller and the other an attacker, where both implement actions by using randomized strategies.

Example 1. The goal of this example is to illustrate the effect of basic deception and disruption attacks within a dual-decomposition DMPC scheme. To this end, we consider the four-tanks system of Fig. 3 (Johansson, 2000), and assume that it is partitioned into two subsystems i.e., $\mathcal{S}_1$ and $\mathcal{S}_2$, that are locally managed by a corresponding controller, i.e., agents 1 and 2 (Alvarado et al., 2011). In particular, these agents can act respective on pumps u_1 and u_2, which, along with the three-way valves, regulate the flows between the storage tank and the 4 smaller tanks that comprise the system. Additionally, the tanks at the top discharge water into the tanks at the bottoms, thus coupling the evolution of their water levels. The subsystems dynamics are described by a model of the form of (2) with matrices:

$$A_{11} = \begin{bmatrix} 0.9705 & 0.0205 \\ 0 & 0.9792 \end{bmatrix}, \quad A_{22} = \begin{bmatrix} 0.9961 & 0.0195 \\ 0 & 0.9802 \end{bmatrix},$$
$$B_{11} = \begin{bmatrix} 0.0068 \\ 0 \end{bmatrix}, \quad B_{12} = \begin{bmatrix} 0.0011 \\ 0.0137 \end{bmatrix}, \tag{18}$$
$$B_{21} = \begin{bmatrix} 0.0002 \\ 0.0160 \end{bmatrix} \text{ and } B_{22} = \begin{bmatrix} 0.0091 \\ 0 \end{bmatrix},$$

where the state of each subsystem is defined by the relative water levels (h_j) of its corresponding tanks to height $h_j^0 = 0.65$ m for all $j = 1, 2, 3, 4$.

Every time step, the agents negotiate their actions to minimize a quadratic global function as (4) with weighting matrices

$$Q_1 = Q_2 = \begin{bmatrix} 1 & 0 \\ 0 & 5 \end{bmatrix}, \quad R_1 = R_2 = 0.01, \tag{19}$$

prediction horizon $N_h = 5$, and the origin as state reference. In particular, at each iteration of the negotiation procedure, both agents optimize the augmented input vectors $\mathbf{u}_1^a = [\mathbf{u}_1; \mathbf{u}_2^1]$ and $\mathbf{u}_2^a = [\mathbf{u}_1^2; \mathbf{u}_2]$, while constraint $\mathbf{u}_1^a - \mathbf{u}_2^a = 0$ is enforced by the introduction of the Lagrange multipliers (see Section 2.1.1). Additionally, the following constraints on the state and inputs must be satisfied: $-0.45 \leq x_i \leq 0.71$, for $i = 1, 2$, $-3.26 \leq u_1 \leq 3.26$, and $-4 \leq u_2 \leq 4$.

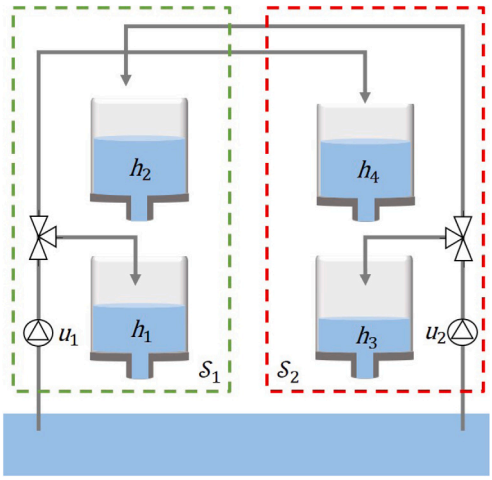

Fig. 3. Diagram of the four-tank system.

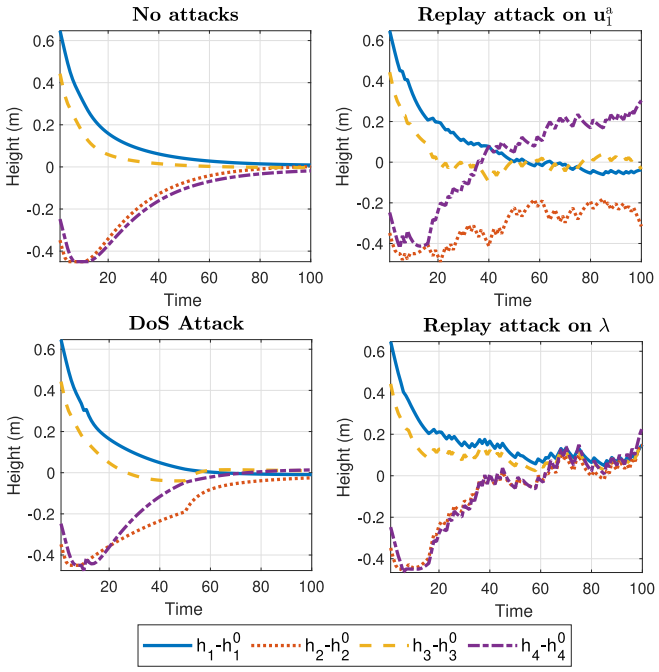

Fig. 4. State evolution of the four-tank system under different disruption and deception attacks. As a reference, the top-left subplot shows the behavior when both agents follow duly the distributed optimization. While the DoS attack (bottom-left) only generates a small deviation from the optimal evolution, the replay attacks (top-right and bottom-right) cause a much greater impact, specially in the water levels of tanks 2 and 4.

For the introduction of the attacks, let us consider the following assumptions on the agents communication. At each iteration, after optimizing their variables $\mathbf{u}_1^a$ and $\mathbf{u}_2^a$, agents 1 and 2 communicate their solutions to a coordinator, which updates the Lagrange prices according to (9). Then, the coordinator communicate to the agents the new multipliers, which again optimize the input sequences. The process is repeated until convergence is attained or a maximum of 100 iterations is reached. The impact on the state evolution caused by different deception and disruption attacks can be seen in Fig. 4. In particular, Fig. 4-(bottom, left) illustrates the effect of a DoS attack, where agent 1 disrupts the distributed negotiation from time instant 10 to 50. During this period, the input sequence received by the coordinator is $\mathbf{u}_1^a = \mathbf{0}$, with $\mathbf{0}$ being a null vector of corresponding dimensions. Note that the effect of the DoS attack strongly depends on

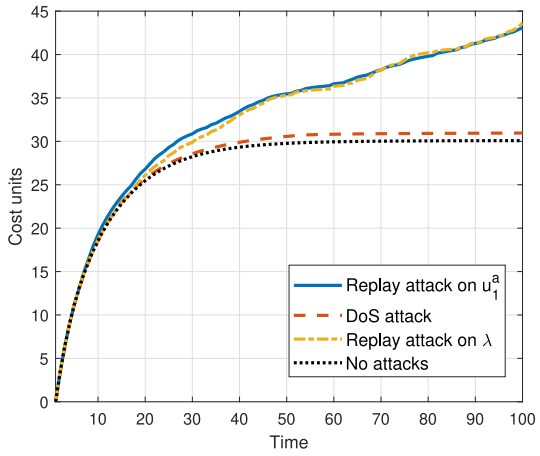

Fig. 5. Cumulative performance cost for the four-tank system under the effect of the attacks in Fig. 4. In accordance with Fig. 4, the DoS attack causes a slight increase of the cumulative costs, with respect to the no-attacks simulation. However, a significant loss of performance is observed for the replay attacks on the input communicated by agent 1 and on the Lagrange price transmitted by the coordinator.

the time period in which it is active. Indeed, if it is performed when the system has reached the steady state and the optimal input $\mathbf{u}_1$ is null or close to be null, then its effect can be negligible. By contrast, if there is a notable mismatch between corrupted signal, i.e., $\mathbf{0}$, and the real one, its impact on the system performance is greater. In this regard, notice that we have not performed any exhaustive or analytic search on the period that maximizes the consequences of the implemented DoS attack. Additionally, Fig. 4-(top, right) shows the results when agent 1 also acts maliciously and implements a replay attack on $\mathbf{u}_1^a$. We assume that agent 1 records its communicated input sequences at any iteration performed during the first 5 time steps. Then, from that step on, with a probability of 0.3, it selects randomly certain instants in which it alters the optimization by, instead of communicating a reliable $\mathbf{u}_1^a$, sharing a false input sequence that was previously transmitted. Similarly, Fig. 4-(bottom, right) shows the state evolution when a replay attack on the communicated Lagrange prices is introduced. In this case, we assume that any of the agents act maliciously, but that there exists an external attacker that alters the prices that go from the coordinator to the agents. Fig. 5 provides the corresponding global cumulative costs, which are calculated as the sum of $\ell(k) = \sum_{i=1,2} \left(\|x_i(k+1)\|_{Q_i}^2 + \|u_i(k)\|_{R_i}^2 \right)$, for all time steps k of the simulation. Note that all of these attacks threaten the Lagrange prices update and the local optimizations, thus altering the implemented actions and hence the system performance. Finally, notice also that the DoS attack is easier to achieve even though it may not have a big impact on cost.

3.2. The software

The software that each agent runs to control its corresponding subsystem also represents a source of vulnerability for the system. In particular, the software can be corrupted due to any external virus causing the unexpected behavior of the attacked agent. *Insider attacks* are also grouped in this category. In this situation, the attacker is one of the system agents that becomes malicious, and its identification is usually more difficult than when the attack comes from an external agent. To illustrate this issue, consider the special case of a Byzantine agent, which sends different information to its neighbors, so it can appear as both compliant and non-compliant to their corresponding detection systems.

In Fig. 2, these attacks happen at (a) and (b). Besides, attacks at (c) and (d) can be originated by any external attacker such as (g) that strikes out at the data in the communication channel, or any internal agent whose software has been attacked such as (a) or (b) that exchanges wrong data with the rest of the agents. Likewise, it is also common to make some assumption regarding the number of malicious entities in the network. To this end, the concepts of *f-local* and *f-global* are frequently used to set an upper bound on the number of attackers. In particular, an *f-local* assumption establishes that there are at most f malicious agents in the neighborhood of an agent. Likewise, the *f-global* assumption sets an analogous bound for malicious agents in the overall network.

In this context, attackers can have rational incentives to send false information to the rest of agents affecting the overall coordination. For example, this is the case presented in Tanaka and Gupta (2016), where a socially efficient implementation of MPC for load frequency control in the presence of self-interested power generators is addressed. Particularly, the system operator tries to implement an MPC searching an aggregated social cost minimization. However, it has to deal with the fact that every participant has its own strategy that may affect the future state of other participants. Therefore, they consider as attackers one or many adversarial agents that misreport their private parameters to maximize their own profits. The same type of motivation is studied by Velarde et al. (2017) and Velarde et al. (2018) who consider a Lagrange-based DMPC scheme where malicious agents modify their local optimization problems (objective functions and constraints) to steer the distributed negotiation according to their self-interest. In particular, instead of (4), the optimization problem objective for the attacked agent becomes

$$J_{\text{malicious}}(x,\mathbf{u}) = \sum_{i\in[1,M]\setminus\{a\}} \sum_{n=0}^{N_h-1} \left(\|x_i(n) - x_{\text{ref},i}\|_{Q_i}^2 + \|u_i(n)\|_{R_i}^2 \right) + \alpha \sum_{n=0}^{N_h-1} \left(\|x_a(n) - x_{\text{ref},a}\|_{Q_a}^2 + \|u_a(n)\|_{R_a}^2 \right). \tag{20}$$

In the so-called *selfish attack*, the malicious agent (agent a) seeks to promote only its local control goals, by introducing the false weighting coefficient $\alpha > 1$ in the objective function minimized by the attacker, so that the terms associated with its local benefit receive a greater weight. A similar effect can be attained if the attacker applies a coefficient to the Lagrangian prices of its coupled variables. In addition, the attacker (agent a) can set a *false reference* $x_{\text{ref},a}$ to steer the negotiation process and improve its original cost function, which contains its true preferences. Also, in the so-called *fake constraints* attack, the malicious agent alters the constraint sets used in its local optimizations. Remarkably, Chanfreut et al. (2018) extend these attacks within the cooperative distributed MPC algorithm of Venkat et al. (2008). The transversality of these attacks between the previously DMPC methods suggests that it should be possible to transpose them to other algorithms with ease. Deep down, the vulnerability boils down to the inherent assumption that there are guarantees regarding the integrity of the local optimization problems and the compliance of the agents regarding the DMPC algorithm.

Maestre et al. (2018) also deal with *non-compliant* controllers within the distributed tube-based MPC scheme introduced in Trodden and Maestre (2017). In this context, the agents share data to dynamically adjust the *size* of coupling uncertainty, and commit themselves to implement local actions that do not exceed certain limits on their mutual disturbances. The *non-compliant* controllers represent malicious or faulty agents that violate this commitment, thus endangering theoretical properties of robustness and stability. Similarly, Ananduta et al. (2018, 2019, 2020) consider *liar agents* that unilaterally deviate from coordinated action values, e.g., by recalculating its control signal after the consensus in the dual-decomposition DMPC is attained (Velarde et al., 2018).

Remark 2. From a communication viewpoint, these attacks can also be considered as deception attacks. Indeed, the consequences of the liar agents of Ananduta et al. (2018, 2019, 2020) are modeled using (15) which corresponds to deception attacks.

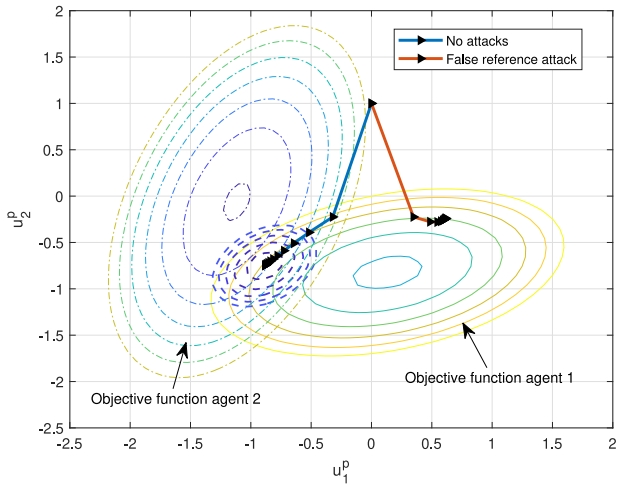

Fig. 6. Effect of a *false reference* attack (Chanfreut et al., 2018; Velarde et al., 2017) on the negotiation procedure described in Section 2.1.2. The figure shows the inputs computed by agents 1 and 2 at each iteration, until an agreement is reached. The blue dashed lines represent the level curves of the global objective function, whose minimum is reached when there are no attackers (blue solid line). Additionally, the colored level curves correspond to the local objective functions of agents 1 and 2. Note that in case of agent 1, this function is given by (23), whereas for agent 2 it is defined analogously. Finally, the red solid line illustrates the impact of the attack on the negotiation, which shows a deviation of the inputs towards points of lower local cost for the attacker, i.e., agent 1. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Finally, there are other attacks that are applicable in this context although they have not been directly considered in the DMPC literature yet. That is the case of Trodden, Maestre, and Ishii (2020), who consider that the attacker can hijack a *portion* of the input set of an agent. Also, Romagnoli, Griffioen, Krogh, and Sinopoli (2020) and Romagnoli, Krogh, and Sinopoli (2019a, 2019b) consider the situation where the controller's software can be infected by a virus that disrupts its behavior so becoming it unpredictable within a certain bounded set.

Table 1 provides a classification of references that consider the above-mentioned types of attacks in the context of MPC. Additionally, a list of works that deal with system faults, such as faulty subsystems (e.g., Boem et al., 2019; Riverso et al., 2016) and sensors and actuators faults (e.g., Ferranti et al., 2019; Raimondo et al., 2013), is also given.

Example 2. This example is devoted to illustrate the effects of the software-based attacks presented in Chanfreut et al. (2018) and Velarde et al. (2017). In particular, we focus on the *false reference* attack and consider a cooperative MPC framework where the agents negotiate by implementing the algorithm in Venkat et al. (2008) (see Section 2.1.2). Consider an academic example with two input-coupled subsystems defined by (2) with

$$A_{11} = \begin{bmatrix} 1 & 0.5 \\ 0 & 1 \end{bmatrix}, \; A_{22} = \begin{bmatrix} 1 & 0 \\ -0.5 & 1 \end{bmatrix}, B_{11} = \begin{bmatrix} -1 \\ 1 \end{bmatrix}, \; B_{22} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \\ B_{12} = \begin{bmatrix} 0.3 \\ -0.2 \end{bmatrix} \text{ and } B_{21} = \begin{bmatrix} 0.1 \\ -0.5 \end{bmatrix}. \tag{21}$$

Additionally, let the objective function be defined as in (4), with $Q_1 = Q_2 = \mathbf{I}_2$ and $R_1 = R_2 = 0.5$, being $\mathbf{I}_2$ the identity matrix of dimensions 2×2. In case of a false reference attack, the attacker, say agent 1, alters its own objective function such that instead of (4) it optimizes (20), which particularized to the current system becomes the following index:

$$J_{\text{falseRef}}(x, \mathbf{u}) = \sum_{n=0}^{N_h-1} \left(\|x_1(n) - x_{\text{ref},1}^{\text{f}}\|_{Q_1}^2 + \|u_1(n)\|_{R_1}^2 \right) + \sum_{n=0}^{N_h-1} \left(\|x_2(n) - x_{\text{ref},2}\|_{Q_2}^2 + \|u_2(n)\|_{R_2}^2 \right) \tag{22}$$

where $x_{\text{ref},1}^{\text{f}}$ denotes the false reference. This misleading variable can be optimized by the attacker to benefit from its neighbors. In particular, Fig. 6 shows the impact on the negotiation procedure when the agent 1 iteratively optimizes $x_{\text{ref},1}^{\text{f}}$ to minimize its local costs, i.e.,

$$J_1(x, \mathbf{u}) = \sum_{n=0}^{N_h-1} \left(\|x_1(n) - x_{\text{ref},1}\|_{Q_1}^2 + \|u_1(n)\|_{R_1}^2 \right), \tag{23}$$

where $x_{\text{ref},1}$ remains being its real reference. Considering a prediction horizon of $N_h = 1$, this figure illustrates the solutions obtained for $\mathbf{u}_1^p$ and $\mathbf{u}_2^p$ along the iterations when the system state is $x_1 = [0.5,\ 0]^{\text{T}}$ and $x_2 = [-0.5,\ 0]^{\text{T}}$, the initial solutions are $\mathbf{u}_1^0 = 0$ and $\mathbf{u}_2^0 = 1$, and the constraints are defined as $-5 \leq x_1, x_2, u_1, u_2 \leq 5$.

Note that, if $x_{\text{ref},1}$ is a new variable, then, the solution $\mathbf{u}_1^p$ at any iteration p (see problem (12) and Eq. (13)) can be formulated as

$$\mathbf{u}_1^p = \kappa(x, \mathbf{u}_1^{p-1}, \mathbf{u}_2^{p-1}, x_{\text{ref},1}^{\text{f}}), \tag{24}$$

where κ is a function defined accordingly. For the sake of clarity, let us omit the dependence on the state and input trajectories and use $\mathbf{u}_1^p(x_{\text{ref},1}^{\text{f}})$. To obtain the optimal false reference, agent 1 first computes the neighbor's action $\mathbf{u}_2^p$, so that the global solution can be written as $\mathbf{u}^p(x_{\text{ref},1}^{\text{f}}) = [\mathbf{u}_1^p(x_{\text{ref},1}^{\text{f}}); \mathbf{u}_2^p]$, and solves the following optimization problem:

$$\min_{x_{\text{ref},1}^{\text{f}}} J_1(x, \mathbf{u}(x_{\text{ref},1}^{\text{f}})) \tag{25}$$

subject to state constraints of subsystem 1. Subsequently, the attacker optimizes (22) using as false reference the solution of (25). That is, the attacker tries to gain advantage by computing a false reference that leads to an input sequence $\mathbf{u}_1^p$ to minimize further local cost $J_1(x, \mathbf{u}_1^p, \mathbf{u}_2^p)$. Once sequence $\mathbf{u}_1^p$ is computed, it is exchanged with its neighbor, which will use it to find $\mathbf{u}_2^{p+1}$. The latter is repeated at each iteration of the negotiation, which deviates the agents from the optimal global performance expected from the algorithm in Venkat et al. (2008). Note that this situation is similar to that of Stackelberg games, where there is a leader and a follower, and the leader anticipates and takes advantage of the reactions of the follower.

3.3. Cyber-attacks consequences in DMPC

The aforementioned attacks translate into different consequences for the DMPC controllers, especially depending on the specific elements attacked (recall Section 2.2).

- *Vulnerabilities in the problem formulation*:
  - The optimization function can be manipulated by a selfish insider attacker causing an increase of overall costs. However, a purely malicious external attacker can provoke worse consequences leading to a loss of theoretical properties such as recursive feasibility.
  - Constraints can be easily violated in case of attack. For instance, deception attacks on input signals can steer the system out of admissible states, and the same holds for disruption attacks. Similarly, malicious agents may attain constraint violations as well.
  - The negotiation process can be altered mainly by insider attackers, causing consequences such as the loss of optimality or convergence in the distributed optimization.

Table 2
Classification of MPC-based methods according to whether they describe detection, identification and/or mitigation mechanisms to counteract the effect of attacks.

Detection		Abdelwahab et al. (2020), Ananduta et al. (2018, 2019, 2020), Barboni et al. (2018), Boem et al. (2019), Chamanbaz et al. (2019), Chen, Wu et al. (2020), Chen, Zhang et al. (2020), Franzè et al. (2020, 2019), Maestre et al. (2018), Mo, Weerakkody, and Sinopoli (2015), Mo et al. (2015), Qiu et al. (2021), Riverso et al. (2016), Sun and Yang (2019), Sun et al. (2019), Wang, Gao and Qiu (2016), Wu et al. (2018), Xu et al. (2021)
Identification		Abdelwahab et al. (2020), Ananduta et al. (2018, 2019, 2020), Boem et al. (2019), Braun et al. (2020), Chen, Wu et al. (2020), Chen, Zhang et al. (2020), Franzè et al. (2020, 2019), Maestre et al. (2018), Qiu et al. (2021), Riverso et al. (2016), Wu et al. (2018)
Mitigation	Active	Abdelwahab et al. (2020), Ananduta et al. (2018, 2019, 2020), Boem et al. (2019), Chen, Zhang et al. (2020), Franzè et al. (2020, 2019), Liu, Wang and Geng (2020), Lješnjanin et al. (2014), Mishra et al. (2017), Pierron et al. (2020), Qiu et al. (2021), Quevedo and Ahlén (2008), Quevedo and Nešić (2010, 2012), Riverso et al. (2016), Sun and Yang (2019), Sun et al. (2019), Wang, Gao and Qiu (2016), Xu et al. (2021), Yang et al. (2019)
	Passive	Ananduta et al. (2018, 2019, 2020), Braun et al. (2020), Chamanbaz et al. (2019), Chen, Wu et al. (2020), Liu and Bai (2018), Liu, Chen, Li and Wan (2020), Liu et al. (2017), Maestre et al. (2018, 2021), Mishra et al. (2016), Pierron et al. (2020), Qin et al. (2020), Quevedo et al. (2015), Romagnoli et al. (2019b), Sun et al. (2019), Tanaka and Gupta (2016), Tiwari et al. (2017), Velarde et al. (2017, 2018), Wang and Ishii (2019), Wang, Song, Liu and Zhang (2016), Wu et al. (2018), Xu et al. (2021), Zafra-Cabeza et al. (2011)

- *Vulnerabilities in the communication process*: the communication network is the target of deception and disruption attacks. Likewise, communication protocols can be altered by external and insider attackers. In both cases, the consequences can be loss of optimality, degradation of system performance, constraints violation, and loss of theoretical properties such as recursive feasibility and robustness.
- *Physical components and their software*: they can also be cyber-attacked by malicious external and insider programs, and also directly manipulated. The consequences can be any of the above mentioned ones.

Finally, note that cyber-attacks in a given element of the DMPC can be originated by different types of attacks causing diverse consequences. Besides, note that the same type of attacks can cause distinct consequences depending on the particular attack point. Moreover, an attack can provoke many consequences as its effect propagates throughout the system due to the close connection between all DMPC elements. On the other hand, consequences also depend on the features of the specific system, e.g., the more unstable system is, the more harmful consequences can be.

4. Cyber-defense mechanisms

Mechanisms for managing cyber-attacks can be organized in three categories, namely, prevention, detection and mitigation measures (Cardenas et al., 2009). These three categories are presented in the next subsections, along with some examples of algorithms from the literature. MPC-based strategies are also summarized in Table 2 and Fig. 7. However, before introducing the corresponding subsections, some comments should be made regarding the close relationship between fault tolerant control (FTC) methods and cyber-defense mechanisms. Both approaches share a common objective because FTC also tries to preserve the stability of the system and maintain an acceptable level of performance in the event of system component malfunctions (Jiang & Yu, 2012). Since cyber-attacks generate non-compliant behavior in system components, it is not surprising that there are significant overlappings in the methods employed in both frameworks. As a matter of a fact, Wang, Gao and Qiu (2016) consider both random failures and packet dropouts, which can be the result of faults and attacks, within a fault-tolerant predictive control scheme that uses a double layer architecture. Also, both attacks and faults can be dealt with robust control methods, e.g., Naghavi et al. (2014) presented a decentralized fault tolerant MPC to address unknown interconnection effects and changes in model dynamics. Likewise, the distributed MPC strategy of Riverso et al. (2016) integrates a distributed fault detection architecture to unplug faulty subsystems that can be easily extrapolated for isolated malicious agents. Another interesting work is that of Raimondo et al. (2013), who propose the use of active isolation mechanisms by modifying the input sequence. In addition, Ferranti et al. (2019) deal with actuator jamming faults and Zarei et al. (2019) with switching issues in power grids. Therefore, the reader is also referred to the rich FTC literature to find other methods that can be suitable for cyber-defense. See for example the surveys of Jiang and Yu (2012) and MacGregor and Cinar (2012).

4.1. Prevention measures

Prevention measures aim to guarantee confidentiality, integrity and availability of the data exchanged by discouraging and hindering the attackers' attempts to penetrate in the control system. The category is broad and ranges from good practices (promoting the rational use of passwords, installing and update firewalls, antivirus and any other relevant software, etc.) to other more advanced methods such as cryptography and cloud-based MPC (Darup, 2020). Regarding the latter, it is worth to mention that the risk introduced from the use of an external communication channel may be compensated by the ease of update of cloud-based software. Also, encrypting the data helps to preserve confidentiality and becomes essential in this context. For example, Alexandru, Morari, and Pappas (2018), Darup, Redder, Shames, Farokhi, and Quevedo (2017) employ a cloud-based MPC architecture using a homomorphic cryptosystem. Likewise, Darup, Redder, and Quevedo (2018) transform the controller using homomorphic encryption so that the MPC computes encrypted actions based on encrypted system states without intermediate decryption. This is useful, for example, to discourage replay attacks.

Another relevant technology in this context is blockchain (Wei, Wu, Long, & Lin, 2019), which generates a chain of data packages known as *blocks* that comprise multiple transactions. These blocks can be validated by the network, so that the integrity of data can be ensured discouraging malicious entities (Nofer, Gomber, Hinz, & Schiereck, 2017). Indeed, blockchain has been used in distributed algorithms. For instance, Wu, Zhang, and Sun (2021) present a blockchain-based multi-time-scale autonomous system within energy distribution networks where MPC is employed for trading. In particular, blockchain was used for recording transactions of the clearing process in a time sequence of blocks based on the consensus and encryption mechanism. Blockchain has also been applied to rate agents in distributed systems. For example, reputation is defined by Kang et al. (2019) and Lei, Zhang, Xu, and Qi (2018) in terms of credibility to decide who is responsible to append the next block. It is straightforward to extend this idea in the context of DMPC to detect and isolate malicious controllers.

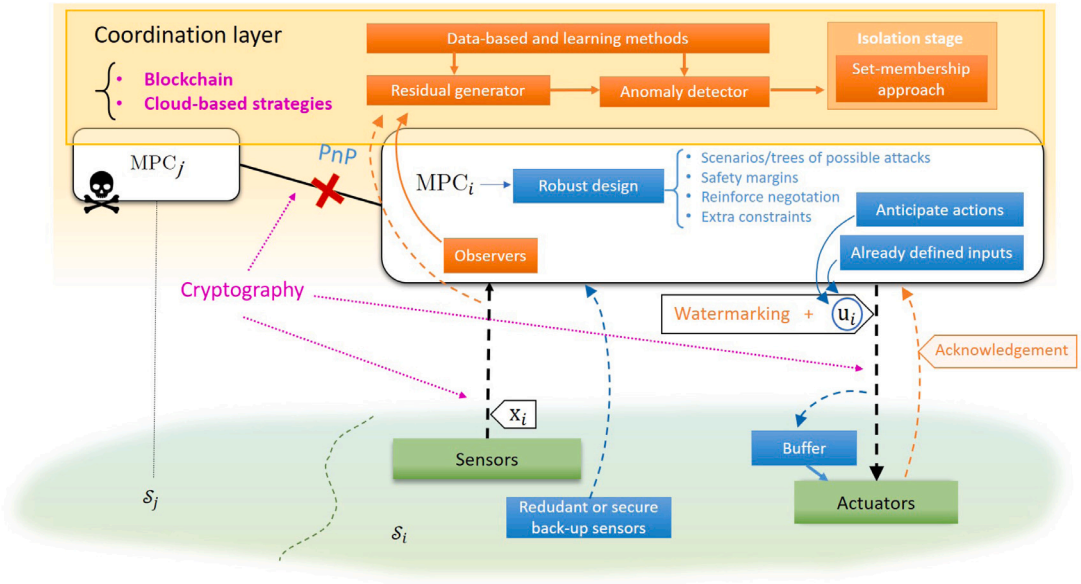

Fig. 7. Diagram of prevention (purple), detection (orange) and mitigation (blue) mechanisms that can be used to protect DMPC schemes. Note that the methods included in the coordination layer or *supervisor* can also be implemented at the controllers level, and thus they could also be integrated within the MPC_i box. For this reason, the boxes associated with the supervisor and the control agents overlap. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Finally, it is worth analyzing whether the DMPC scheme implemented offers incentives for misbehavior. For example, the two popular schemes presented in Section 2 allow malicious entities to improve their cost at the expense of other subsystems. In this regard, there may be other alternatives within the literature that are naturally robust to this type of misbehavior. For example, Maestre, Muñoz De La Peña and Camacho (2011) and Maestre, De La Pena, Camacho and Alamo (2011) propose DMPC methods based on game theory where agents make proposals that can be accepted or rejected by neighboring subsystems. Therefore, it is in the best interest of the proposer agent that the proposal reflects its true preferences regarding the sequences for the shared input signals. From this viewpoint, these schemes discourage malicious behavior, although they do not yield optimal performance because the coordination is attained in few communication rounds with a much more restricted information exchange. The works of Maestre, Muñoz De La Peña and Camacho (2011) and Maestre, De La Pena, Camacho and Alamo (2011) were later enhanced by Francisco, Mezquita, Revollar, Vega, and De Paz (2019) and Masero, Francisco, Maestre, Revollar, and Vega (2021), who include an additional fuzzy layer so as to merge proposal considering their impact on key performance indicators. Again, this practice promotes that only input sequences that lead to an improvement of the overall cost are actually taken into consideration. In addition, the negotiation process can be reinforced to encourage agents to share information truthfully searching the global optimization. Tanaka and Gupta (2016) present a real-time monetary compensation schemes along with MPC-based for load frequency control to promote self-interested power generators to report their parameters truthfully.

4.2. Detection and isolation measures

Detection and isolation become relevant when the attacker has overcome the prevention measures and the system is actually under attack. Also, many algorithms rely on attack detection to trigger active defensive strategies (Franzè et al., 2020; Maestre et al., 2018) and Qin et al. (2020), although the isolation step is not always required (Sun & Yang, 2019; Sun et al., 2019; Wang, Gao & Qiu, 2016).

Detection is always based in the occurrence of a measurable event that allows the system to raise the *under-attack* flag. It must be noted that detection mechanisms are not perfect, i.e., false (negative and positive) detections will occur with a probability that depends on how the corresponding thresholds are configured. Additionally, these strategies can be broadly divided into two categories: active and passive (Chamanbaz et al., 2019). Active strategies manipulate the control system, e.g., by including extra input signals, to observe the corresponding state responses so as to learn more about the attack, whereas passive strategies do not affect the system.

In their simplest form, the detection and isolation can be based on exceptions returned by the control system, e.g., time-outs, absence of acknowledgment frames (Sun & Yang, 2019), etc. Besides that simple approach, two families of methods can be found in the literature, namely, analytical and learning approaches.

4.2.1. Analytical detection approaches

Analytical approaches exploit the model of the system and the bounds of its signals to find out whether an attack happened. Many attack detection methods are based in observers that provide expected state values for the plant. Expected and measured states and outputs are then compared to form a residual, which is used by an anomaly detector to determine whether there is an attack (Cui et al., 2012). For instance, Chamanbaz et al. (2019) present an FDI attack detector for non-linear systems along with an MPC controller. In particular, an additional constraint is added to the MPC to force the system to remain in a neighborhood of a properly designed reference trajectory. The tracking error is then used as the residual, which feeds a nonparametric cumulative sum anomaly detector. Barboni et al. (2020) design a distributed covert attack detection algorithm of linear large-scale interconnected systems, which is based on using for each subsystem both a decentralized observer, whose state estimation is decoupled from its neighbors, and a distributed observer, which computes a state estimate based on the communicated neighboring estimations. The possible inconsistencies in the measurements from neighbors are revealed by exploiting the cooperation between the two observers. Riverso et al. (2016) design a fault detector for faulty subsystems based on residual computation which can also be used for attack detection. Each subsystem is equipped with a local nonlinear estimator whose estimation is used for computing the residual. If any component of this residual exceeds a determined threshold, the subsystem is marked as faulty. Similarly, Boem et al. (2019) present a fault detection and isolation procedure for FTC, where the local passive fault detection is

based on computing a residual error between a nominal or expected state and the real state. Also, Braun et al. (2020) introduce a scalable hierarchical attack identification method for systems of interconnected nonlinear systems with coupled dynamics or constraints. After the attack detection, its propagation through the network is approximated and used to formulate a quadratic program that determines the attack signal that best explains the observed network evolution.

These residual tests can be complemented with set-membership strategies to isolate malicious agents. Maestre et al. (2018) present a DMPC approach with robustness against noncompliant controllers, where agents exchange information regarding the bounds of their local disturbance sets, which are optimized in each step to reduce mutual disturbances. These exchanged sets are employed to perform noncompliance detection and isolation bounds on expected and real disturbance values. Similarly, Franzè et al. (2020, 2019) use set-membership tests for attack detection in networked multi-agent systems and Qiu et al. (2021) design a cyber-attack localization method using set-membership estimation. Likewise, Raimondo et al. (2013) present an active fault detection and isolation method based on observer design using residuals computation.

Another control-theoretic method of active detection is *physical watermarking* (Yaghooti et al., 2021), which consists on injecting a known noisy input to the system. If the corresponding effect in the system dynamics is not found in the output, it is derived that there is an attack (Cayre, Fontaine, & Furon, 2005). For example, Mo et al. (2015) present a watermarking algorithm where an optimal watermarked signal is designed as a random noise of known distribution, and an anomaly detector based on residuals is used for attack detection. Similarly, Abdelwahab et al. (2020) design a watermarking technique for detecting replay attacks where watermarked control signals are obtained by randomly dropping the last computed command input. Likewise, Ferrari and Teixeira (2020) propose a switching multiplicative watermarking scheme for detection of stealthy attacks on sensor measurements.

Finally, note that there are other detection and isolation methods in the literature that have not been applied yet to DMPC schemes, but they could be transposed. For instance, Isozaki et al. (2015) present a detection algorithm for voltage control that considers multiple features of the measured signal. Chakhchoukh and Ishii (2016) propose to run multiple robust estimators with different breakdown points to improve the detection of cyber-attacks in state estimation. Nishino and Ishii (2014) consider distributed detection methods of cyber-attacks and faults for power systems by grouping of buses in the system. Finally, Cetinkaya, Arcaini, Ishii, and Hayakawa (2020) present an identification method for jamming attacks based on a search-based approach that uses multi-objective genetic algorithms.

4.2.2. Learning detection approaches

Learning methods can be fed with data such as performance indicators and residuals. For example, the detection stages of Chen, Wu et al. (2020) and Wu et al. (2018) are both based on machine learning. In particular, a neural network (NN)-based detection system is built by Wu et al. (2018), whereas Chen, Wu et al. (2020) develop data-based cyber-attack detectors using sensor data to identify the presence of cyber-attacks as well as to differentiate between the different types of cyber-attacks. In addition, Ananduta et al. (2018, 2019, 2020) design a local identification mechanism based on hypothesis testing with Bayesian inference that is used to decide the connections with neighbors.

Finally, there are other learning approaches that can also be applied for detection mechanism, such as the warm-start algorithm presented by Chanfreut, Sánchez-Amores, Maestre, and Camacho (2021), where an off-line database of the optimized pairs between states and their corresponding Lagrange multipliers is created for a dual-decomposition DMPC scheme, and the machine learning outlier detection method designed by Chakhchoukh, Liu, Sugiyama, and Ishii (2016) for stealthy FDI attacks in state estimation.

Example 3. This example is presented to illustrate the performance of an identification mechanism based on the work of Ananduta et al. (2020). The mechanism is designed independently of the MPC algorithm, but the MPC optimization problem of each agent is accordingly adapted to penalize neighbors with higher probability of being attackers based on the results of the identification mechanism. Therefore, each agent learns over time which are the attackers and proceeds to exclude them accordingly.

We consider a networked system composed of 5 interconnected agents, and focus on agent 5, which is connected to all other agents (1 to 4). Also, it is assumed that the maximum number of malicious agents in the neighborhood of an agent is 2. In particular, agents 1 and 3 are malicious in this example. When an attack takes place, the malicious neighbor deceives agent 5 and applies signals that differ from the agreed value. Since agent 5 can only compute a residual of the aggregate impact of its neighbors, isolating the attacks sources becomes a challenging issue.

The strategy used by agent 5 for attack isolation is based on testing eleven different hypotheses using Bayesian inference. The hypotheses tested cover the different possibilities of insider attacks, which range from no malicious neighbors, to all the combinations where there are one or two malicious agents. Initially, agent 5 considers that the probability of receiving an attack from any of its neighbors is 0.2, and assigns uniform probability for all the hypotheses, i.e, $P_{\mathrm{H}_i}(0) = 1/11$. Regarding the detection stage, the anomaly detection has probability of false positive of 0.05, and a probability of false positive of 0.1. Also, the actual attack rate of malicious agents is $P_{\mathrm{a}} = 0.25$.

At each simulation step, agent 5 negotiates with a random subset of neighbors, $S(k)$, for which the probability of attack is computed as

$$P_{\mathrm{S}}(k) = 1 - (1 - P_{\mathrm{a}})^{s_{\mathrm{m}}(k)} \tag{26}$$

where $s_{\mathrm{m}}(k)$ represents the number of malicious agents in $S(k)$. Agent 5 computes whether it receives an attack and updates the probability of all hypothesis using the Bayesian inference, which is defined as

$$P(H_i|E) = \frac{P(E|H_i)P(H_i)}{P(E)}, \tag{27}$$

where H_i stands for hypothesis i and E denotes the detection/no detection event. Also, $P(H_i)$ is the estimate of the probability of the hypothesis H_i, $P(E)$ is the probability of observing E, $P(H_i|E)$ is the probability of H_i given E, and $P(E|H_i)$ is the probability of observing E given H_i. This way, the hypothesis probabilities are updated by considering data from current and previous instants, and agent 5 can learn to identify malicious agents and decide which neighbors can remain for further negotiation rounds.

Fig. 8 presents the probability evolution of all hypotheses considered in this example. As can be seen, hypothesis H_6 reaches the value 1 while all the others become 0, indicating that agents 1 and 3 are identified as adversarial, just as expected.

4.3. Mitigation measures

Different strategies can be implemented to mitigate the impact of attacks. Inspired by the FTC framework, we classify mitigation algorithms as *active* and *passive* approaches. In particular, in active methods controllers that respond to failures can change their setup, whereas passive approaches are those that employ a failure-proof design (Jiang & Yu, 2012). In other words, active strategies implement extra actions after the attack occurrence whereas passive strategies are preconfigured robust controllers against attacks.

Remark 3. Passive control strategies can also be seen as prevention measures. However, they have been introduced in the mitigation category because that is their primary goal.

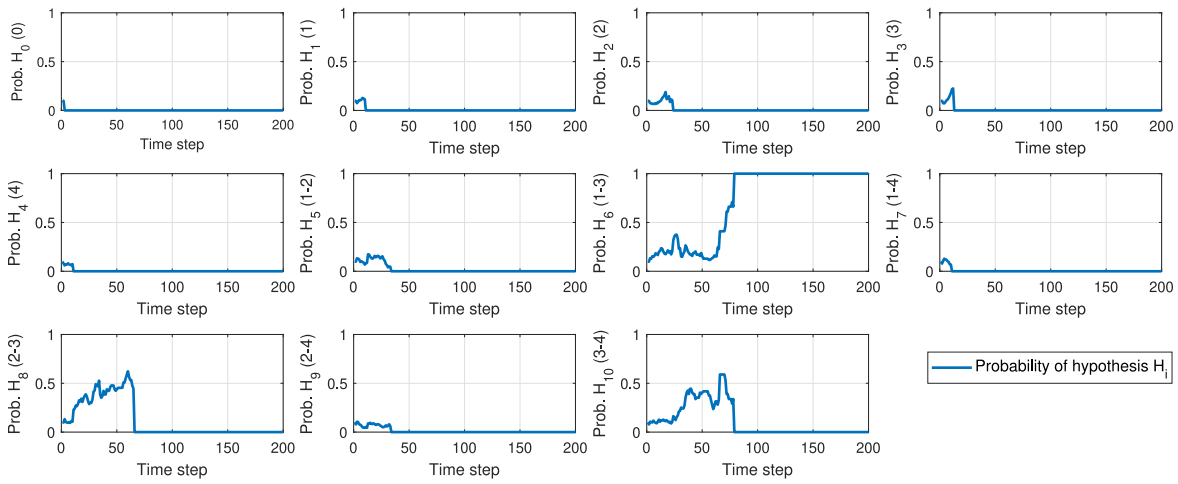

Fig. 8. Evolution of the probabilities assigned to each of the different hypothesis H_i. The labels of the y-axis indicate between brackets which agents are considered as malicious controllers in each of the hypothesis. Note that the probability of agents 1 and 3 being malicious reaches and remains at the value of 1 from time step 80, i.e., agent i is able to detect the attackers.

4.3.1. Active mitigation strategies

Active strategies react in case of attack and let the system work in nominal conditions otherwise. Therefore, attack detection becomes essential for them. Let us examine next how these methods can be used to mitigate cyber-attacks.

Disruption attacks are simple and the most commonly used mechanism to deal with them takes advantage of the predictive features of MPCs. In particular, a buffer can store the last successfully received input sequence and, in case of attack, the corresponding input element can be applied (Ljeśnjanin et al., 2014; Qiu et al., 2021; Quevedo & Nešić, 2010, 2012; Sun et al., 2019; Wang, Gao & Qiu, 2016). In addition, Sun and Yang (2019) combine this buffer with an event-triggered process for transmission. In absence of DoS attacks, control data is updated at each instant and an acknowledgment is received by the controller to confirm that the transmission was successful. If the acknowledgment is missing, it is assumed that a DoS attack is present and retransmission attempts occur following the sampling interval of the control unit until the controller receives an acknowledgment that indicates lack of DoS. In addition, Quevedo and Nešić (2012) assume that these acknowlegements can also be disrupted and design a packetized MPC controller with an actuator buffer to deal with DoS in input signals. Similarly, Pierron et al. (2020) compute the input sequence by using a tree-based MPC strategy that considers all possible scenarios for the prediction horizon. This way, the input sequence is robustified against all possibilities of packet losses, and in case of loss, the corresponding input signal is taken from the last input tree received. In addition, Franzè et al. (2019) add an auxiliary nominal MPC scheme in charge of keeping system dynamical behavior at an admissible level until the communication channel is restored in case of all input signals from the last sequence stored have already been applied. Likewise, Mishra et al. (2017) consider a buffer in the actuator side and present three transmission protocols depending upon the availability of storage and computation facilities at the actuator.

Observers can also be used to mitigate attacks. For example, Qin et al. (2020) use a Luenberger observer to estimate the states of the plant with noisy sensor measurements, and after that, a resilient Luenberger observer is designed using the invariant set theory in the presence of attacks to input signals. Then, an output feedback MPC strategy is presented to handle these attacks. Yang et al. (2019) design a distributed stochastic MPC that uses an observer in the control side to reconstruct the state signals when networks suffer from DoS attacks. Also, a distributed nonlinear observer approach is designed in Cecilia, Sahoo, Dragicevic, Costa-Castello, and Blaabjerg (2021) that estimates the system states even in the presence of false data. This estimation is used to detect the attack presence and also to reconstruct the attacked signal. Additionally, Hu et al. (2020) present an observer-based dynamic event-triggered control under DoS attacks, where the observer is constructed to deal with the unavailability of full-state information. Quevedo and Ahlén (2008) design a state estimation method for wireless sensor networks over fading channels causing random packet loss by using a time-varying Kalman Filter along with a controller endowed with predictive control elements.

To mitigate deception attacks in measurement signals, some secure back-up and redundant sensors can be designed. This is proposed, for example by Wu et al. (2018), who present a Lyapunov-based MPC method that utilizes state measurements from secure, redundant sensors in case a sensor tamper cyber-attack is detected, and by Chen, Wu et al. (2020), who replace the attacked sensors by secure back-up sensors for an economic MPC.

In case of insider attackers, a possible strategy after their identification is to disconnect them from the system, i.e., the *Plug-and-Play* (PnP) architecture. Riverso et al. (2016) propose a DMPC strategy in a PnP framework for faulty subsystems identification and unplugging to avoid the propagation of the fault. Similarly, Boem et al. (2019) design a tube-based MPC scheme that, after fault detection and isolation, allows the possible disconnection of faulty subsystems and the local controllers reconfiguration. Ananduta et al. (2018, 2019, 2020) also design a local mechanism where each agent disconnects from neighbor agents after their identification as attackers for its local optimization in a DMPC scheme.

Other possible mitigation measures apply different predefined actions depending on the particular attack affecting in the system. This is the case of Franzè et al. (2020), who define a resilient DMPC scheme against replay attacks for multi-agent network systems. In particular, the system is topologically described by a leader–follower digraph and set-theoretic receding horizon control ideas are exploited to implement specific control actions to avoid the possible domino effects from the attack.

Finally, some maintenance and system enhancements can be performed to mitigate the possible damage that attacks cause. For instance, Zafra-Cabeza et al. (2011) present a hierarchical DMPC approach using a risk management strategy for irrigation canals where mitigation actions are executed if risk occurrence are expected. The considered risk factors are unexpected changes in demand, failures in operation or maintenance costs, which can also be considered as caused by attacks.

Other strategies that could also be extended to DMPC schemes are presented by Cetinkaya et al. (2019), who provide an overview on the

developed control and communication techniques to achieve resiliency against DoS attacks. Similarly, Kikuchi et al. (2017) develop a new consensus framework based on stochastic communication protocols to deal with jamming attacks in the communication network between agents.

4.3.2. Passive mitigation strategies

Passive strategies are based on a robust control designed to deal with possible attacks ensuring system safety, usually without the necessity of attack detection and identification.

For instance, the consequences of attacks can be mitigated by using scenario-based techniques in a DMPC, so that the input sequence is computed accounting for scenarios of nominal operation (Maestre et al., 2021). Similarly, Pierron et al. (2020) employ a tree-based MPC to deal with jamming attacks where the robust input sequence is computed by considering all possible scenarios of packet losses for the prediction horizon. Additionally, Braun et al. (2020) design a robust nonlinear MPC (NMPC) setup that integrates the contract-based distributed NMPC along with a multi-stage NMPC for systems of interconnected nonlinear subsystems with coupled dynamics or constraints. In particular, local controllers exchange sensitivity information about their coupling variables which is used to approximate the propagation of the attack through the network. Reachable sets and resulting contracts are approximated with multi-stage NMPC, but to decrease the size of the scenario tree, branching is only applied up to some earlier stage.

Also, the MPC controller can be designed by including some extra features. For example, in Wang, Song, Liu and Zhang (2016), a MPC-based static output feedback controller is designed by using linear matrix inequality constraints to consider randomly occurring deception attacks. Chamanbaz et al. (2019) augment the MPC controller with additional constraints to ensure the actual output trajectory remains within a specified time-invariant neighborhood of the reference trajectory to deal with FDI attacks on input and measurement channels. Liu and Bai (2018) use an iterative DMPC algorithm to design distributed controllers based on a cooperative control strategy to address time-varying delayed input states. Mishra et al. (2016) and Quevedo et al. (2015) present a stochastic MPC whose cost function explicitly accounts for random packet dropouts. Sun et al. (2019) robustify the dual-mode MPC by including a terminal constraint set which is designed considering DoS attacks. Ananduta et al. (2018, 2019, 2020) robustify the controllers via constraints with a stochastic method in which probabilistic bounds are computed for possible disturbances and attacks. This way, the decisions obtained for the DMPC are robustly feasible against most of the attacks with high confidence.

In case of insider attacks, a mitigation measure is to reinforce the negotiation process. Velarde et al. (2018) present a Lagrange-based DMPC based on dual-decomposition where the two extreme control actions are dismissed in the consensus approach. Similarly, Wang and Ishii (2019) study the problem of resilient consensus in multi-agent networks applying a DMPC scheme where the resilient update rules consider the presence of malicious agents in the network. The total number of malicious neighbors of an agent is bounded using the *f-local* model. Before the local update, each node removes the f largest and f smallest values from its neighbors, which are considered as possibly malicious.

Other possibility to gain robustness is to readjust the system constraint sets by considering the possible effects of the attacks. For example, Qin et al. (2020) tighten the constraints of the nominal system in the MPC optimization. Maestre et al. (2018) present a robust DMPC for non-compliant agents where mutual disturbance sets are minimized by local optimization to provide robustness.

Many other developed algorithms are based in strategies that could also be applied with MPC, e.g., software rejuvenation. This strategy consists on periodical resets of the control software to a secure version. The software refresh frequency is established to guarantee system safety in case of attack (Aung & Park, 2004). This way, the control strategy is robustly designed. In particular, Romagnoli et al. (2019a) design a software rejuvenation strategy using invariant sets, and Arauz, Maestre, Romagnoli, Sinopoli and Camacho (2021) present an extended version for discrete-time systems. Besides, Griffioen, Romagnoli, Krogh, and Sinopoli (2019) apply the software rejuvenation strategy for secure networked control which differs from the previous works in the necessity of the network connection for the proper system recovery. Another algorithm that could be used in DMPC is the approach presented by Feng and Ishii (2020) for searching consensus of multi-agent systems under DoS attacks, which is based on dynamic quantization. Finally, different strategies have been implemented in the literature to mitigate the impact of attacks in the context of networked control. For example, Farraj, Hammad, and Kundur (2017) propose an adaptative cyber-enabled parametric feedback linearization control scheme to deal with cyber-attacks on data integrity and availability. Yang, Xu, Xia, and Zhang (2020) use a predictive control method to ensure system stability under DoS and FDI attacks. Sharma, Singh, Lin, and Foruzan (2017) propose a consensus based leader–follower distributed control scheme and an agent-based distributed optimal control scheme, both considering the presence of cyber-attacked misbehaving agents. In addition, many works deal with packet losses in a general sense and not only caused by attacks. For example, Arauz, Maestre, Cetinkaya and Camacho (2021) use the strategy presented by Cetinkaya et al. (2015) for packet dropouts due to unreliable transmissions and jamming attacks to design a resilient feedback controller. Likewise, Alonso, Ho, and Maestre (2021) present a distributed linear quadratic regulator robust to communication dropouts.

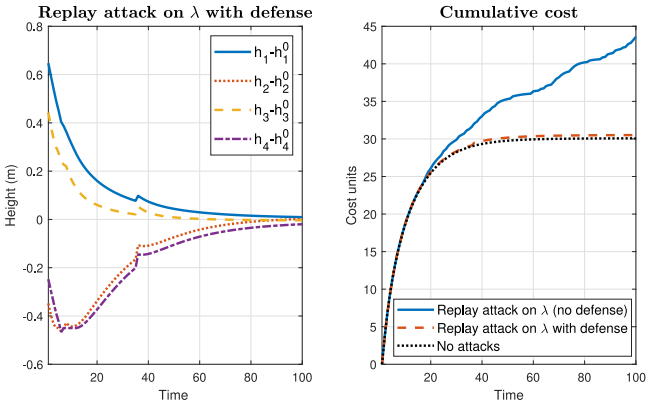

Fig. 9. State evolution and cumulative cost when the defense mechanism described in Example 4 is introduced to reduce the impact of the replay attack on λ implemented in Example 1. The figure shows how this neural-network-based defense allows to practically avoid the impact of the attacker that alters the prices sent by the coordinator. As can be seen, it leads to a cumulative cost similar to that of the optimal performance, thus also reflecting that the multipliers provided by model (28) are close to the optimal ones.

Example 4. We introduce a defense mechanism that uses the predictive ability of neural networks to provide an optimal warm start for Lagrange prices (Chanfreut et al., 2021) to alleviate the effects of the replay attack on the Lagrange multipliers implemented in Example 1 (see bottom-right subplot at Fig. 4). Note that the problem setting remains the same as in Example 1, i.e., the subsystems are modeled by (18), and the quadratic objective function defined by (19) with the prediction horizon $N_h = 5$ is optimized subject to the corresponding state and input constraints. For this setting, we compute a model of the form

$$\lambda_{NN} = f_{NN}(x_1, x_2), \tag{28}$$

where sub-index 'NN' stands for neural network. Function $f_{NN}(x_1, x_2)$ was trained from a database composed of a set of 1500 global states and

their associated optimal multipliers, which were generated simulating a compliant dual-decomposition based negotiation on the four-tank system. To this end, the function `train` from Matlab Deep Learning Toolbox was used, with the Levenberg–Marquardt algorithm, and a division of the data into 70% for training, 15% for validation and 15% for testing.

Assuming the global state is known by the coordinator layer, model (28) can be used as an *oracle* to detect the presence of attackers and react to them (see Fig. 9). In particular, we consider that, after implementing the negotiation procedure at each time instant, the coordinator checks if the resulting Lagrange price, say $\lambda^{\bar{p}}$, where $\bar{p}$ is the index of the last performed iteration, differs significantly from the forecast λ_{NN}. For the results in Fig. 9, this condition has been formulated as $\|\lambda^{\bar{p}} - \lambda_{NN}\|_\infty > \delta$, where δ has been set to 0.15. An attack is detected if the previous inequality holds. As a mitigation mechanism, both agents take λ_{NN}, recalculate their solution $\mathbf{u}_1^a$ and $\mathbf{u}_2^a$ with the new prices, and implement the corresponding actions.

5. Conclusions

The evolution of technology is leading towards a world of pervasive connectivity where control systems are expected to play a major role. In this context, DMPC algorithms are likely to become essential because they enhance their capabilities with the progress of technology and they offer a means to coordinate control actions in order to attain optimal performance in a scalable manner. However, an interconnected world will also offer significant opportunities for cyber-attacks, which may have devastating consequences if they affect critical infrastructures. Even nowadays we can find every now and then headlines in major newspapers that show how severe cyber-threats can become, with some notorious attacks disrupting applications such as nuclear plants, power grids, smart buildings and autonomous cars, to make a few examples.

In this article we have reviewed the most vulnerable spots in the control infrastructure that can be exploited to attack DMPC methods. In addition, we have seen that the algorithms have inherent vulnerabilities because their design is usually based on the assumption that every controller in the network will be compliant with the algorithm employed. To deal with these issues, we have presented detection and mitigation mechanisms that can be used to make these schemes resilient. In particular, we have seen that learning methods can make a difference in defensive tasks because of their superior flexibility to adapt to the nominal operation conditions.

Future work on this area could focus on the following topics:

- **Vulnerability assessment of DMPC protocols:** the tens of schemes in the DMPC literature should be inspected searching for vulnerabilities. In particular, it should be explored whether an abnormal behavior can be generated whenever one or more agents are not following the method. Also, it should be explored whether the methods offer incentives for misbehavior, i.e., whether one agent can improve its costs by manipulating the information exchanged with other agents. In this way, it is possible to build attack models that can help improve detection and mitigation mechanisms. Additionally, MPC controllers can collaborate with decision entities of different nature, such as human beings (Van Overloop, Maestre, Sadowska, Camacho, & De Schutter, 2015), which opens-up new vulnerabilities due to the increased difficulty to characterize the attackers' behavior.
- **Theoretical properties:** while properties such as robustness and stability are always of interest, we believe that other properties of interest can emerge in the context of cyber-security. To being with, protection comes at a price and it is interesting to quantify the loss of performance of cyber-defense methods. Likewise, it can be interesting to obtain guarantees regarding the time that a hijacked control system can resist before events such as constraint violation or an irreversible stability loss can occur.
- **Resilient and flexible control strategies:** positive network externalities are related to the possibility of failure and misbehavior. While the predictive control framework possess strong theoretical results regarding robustness, it needs to expand its toolbox for dynamic networked environments where agents can be plugged and unplugged, and possibly change their behavior due to cyber-threats. How to isolate malicious agents and cluster healthy ones so as to obtain the best achievable performance is an interesting open problem for the DMPC community. Additionally, the systems may integrate subsystems with heterogeneous dynamics and agents with different computation and communication equipment, leading to further challenges to detect and mitigate attacks.
- **Benchmarks and performance indicators:** an essential task in this context is the validation of strategies by using properly designed testbeds. In particular, this is necessary to correctly compare the impact of different attack models and the mitigation and detection power of new cyber-defense mechanisms. To this end, it is also necessary to define a set of standardized performance indicators and properties of interest.
- **Learning methods:** As our examples have shown, the recent bloom of learning methods for MPC can find multiple applications here, hopefully with strong statistical guarantees that allow to quantify their detection and mitigation power.
- **Blockchain:** the application of blockchain based technologies seems an unstoppable trend that have penetrated in very close fields such as consensus. For this reason, we believe that it is a matter of time that blockchain DMPC methods are proposed. A particularly relevant question here is whether the overhead generated by the use of this technology can become an issue for schemes that require hundreds of iterations before convergence is attained.

Finally, we would like to stress some of the limitations of this work. In the first place, our survey is by no means exhaustive because nowadays there are literally thousands of cyber-security articles yearly published out there, not to mention that the methods that are employed present relevant overlaps with other well established areas, e.g., switching systems, game theory, and networked and fault tolerant control. Likewise, the examples given are meant to be of academic use, i.e., they intend to be illustrative enough to help the reader understand how attacks can affect system performance and the extent to which cyber-defense mechanisms can relieve these issues. Nevertheless, these issues can become research opportunities for the predictive control community because we have seen that some attack models and defense methods have not been considered nor validated yet in DMPC methods. In this regard, we hope to have given here a starting point and a roadmap that can provide help and guidance for future works.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This paper has received funding from the Spanish Training Program for Academic Staff under Grants (FPU19/00127 and FPU17/02653), the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (OCONTSOLAR, grant agreement No 789051), the project GESVIP (ref. US-1265917), and MCIN/AEI/ 10.13039/501100011033 under project C3PO-R2D2 (PID2020-119476RB-I00).

References

Abdelwahab, A., Lucia, W., & Youssef, A. (2020). Set-theoretic control for active detection of replay attacks with applications to smart grid. In *2020 IEEE conference on control technology and applications* (pp. 1004–1009). IEEE.

Alexandru, A. B., Morari, M., & Pappas, G. J. (2018). Cloud-based MPC with encrypted data. In *2018 IEEE conference on decision and control* (pp. 5014–5019). IEEE.

Alonso, C. A., Ho, D., & Maestre, J. M. (2021). Distributed linear quadratic regulator robust to communication dropouts. arXiv preprint arXiv:2103.03967.

Alvarado, I., Limon, D., De La Peña, D. M., Maestre, J. M., Ridao, M., Scheu, H., et al. (2011). A comparative analysis of distributed MPC techniques applied to the HD-MPC four-tank benchmark. *Journal of Process Control*, *21*(5), 800–815.

Ananduta, W., Maestre, J. M., Ocampo-Martinez, C., & Ishii, H. (2018). Resilient distributed energy management for systems of interconnected microgrids. In *2018 IEEE conference on decision and control* (pp. 3159–3164). IEEE.

Ananduta, W., Maestre, J. M., Ocampo-Martinez, C., & Ishii, H. (2019). A resilient approach for distributed MPC-based economic dispatch in interconnected microgrids. In *2019 18th European control conference* (pp. 691–696). IEEE.

Ananduta, W., Maestre, J. M., Ocampo-Martinez, C., & Ishii, H. (2020). Resilient distributed model predictive control for energy management of interconnected microgrids. *Optimal Control Applications & Methods*, *41*(1), 146–169.

Arauz, T., Maestre, J. M., Cetinkaya, A., & Camacho, E. F. (2021). Model-based PI design for irrigation canals with faulty communication networks. In *2021 European control conference*. IEEE.

Arauz, T., Maestre, J., Romagnoli, R., Sinopoli, B., & Camacho, E. (2021). A linear programming approach to computing safe sets for software rejuvenation. *IEEE Control Systems Letters*, *6*, 1214—1219.

Aung, K. M. M., & Park, J. S. (2004). Software rejuvenation approach to security engineering. In *International conference on computational science and its applications* (pp. 574–583). Springer.

Bagherzadeh, M., & Lucia, W. (2019). A set-theoretic model predictive control approach for transient stability in smart grid. *IET Control Theory & Applications*, *14*(5), 700–707.

Baldivieso-Monasterios, P. R., & Trodden, P. A. (2021). Coalitional predictive control: Consensus-based coalition forming with robust regulation. *Automatica*, *125*, Article 109380.

Barboni, A., Boem, F., & Parisini, T. (2018). Model-based detection of cyber-attacks in networked MPC-based control systems. *IFAC-PapersOnLine*, *51*(24), 963–968.

Barboni, A., Rezaee, H., Boem, F., & Parisini, T. (2020). Detection of covert cyber-attacks in interconnected systems: A distributed model-based approach. *IEEE Transactions on Automatic Control*, *65*(9), 3728–3741.

Bhamare, D., Zolanvari, M., Erbad, A., Jain, R., Khan, K., & Meskin, N. (2020). Cybersecurity for industrial control systems: A survey. *Computers & Security*, *89*, Article 101677.

Bindra, A. (2017). Securing the power grid: Protecting smart grids and connected power systems from cyberattacks. *IEEE Power Electronics Magazine*, *4*(3), 20–27.

Boem, F., Gallo, A. J., Raimondo, D. M., & Parisini, T. (2019). Distributed fault-tolerant control of large-scale systems: An active fault diagnosis approach. *IEEE Transactions on Control of Network Systems*, *7*(1), 288–301.

Boem, F., Riverso, S., Ferrari-Trecate, G., & Parisini, T. (2018). Plug-and-play fault detection and isolation for large-scale nonlinear systems with stochastic uncertainties. *IEEE Transactions on Automatic Control*, *64*(1), 4–19.

Boyd, S., Parikh, N., & Chu, E. (2011). *Distributed optimization and statistical learning via the alternating direction method of multipliers*. Now Publishers Inc.

Braun, S., Albrecht, S., & Lucia, S. (2020). Hierarchical attack identification for distributed robust nonlinear control. In *Proc. of the 21st IFAC world congress* (pp. 6191–6198).

Camacho, E. F., & Alba, C. B. (2013). *Model predictive control*. Springer science & business media.

Cardenas, A., Amin, S., Sinopoli, B., Giani, A., Perrig, A., Sastry, S., et al. (2009). Challenges for securing cyber physical systems. In *Workshop on future directions in cyber-physical systems security*: *Vol. 5*, (1), Citeseer.

Cayre, F., Fontaine, C., & Furon, T. (2005). Watermarking security: Theory and practice. *IEEE Transactions on Signal Processing*, *53*(10), 3976–3987.

Cecilia, A., Sahoo, S., Dragicevic, T., Costa-Castello, R., & Blaabjerg, F. (2021). Detection and mitigation of false data in cooperative DC microgrids with unknown constant power loads. *IEEE Transactions on Power Electronics*.

Cetinkaya, A., Arcaini, P., Ishii, H., & Hayakawa, T. (2020). A search-based approach to identifying jamming attacks and defense policies in wireless networked control. In *2020 59th IEEE conference on decision and control* (pp. 5717–5724). IEEE.

Cetinkaya, A., Ishii, H., & Hayakawa, T. (2015). Event-triggered control over unreliable networks subject to jamming attacks. In *2015 54th IEEE conference on decision and control* (pp. 4818–4823). IEEE.

Cetinkaya, A., Ishii, H., & Hayakawa, T. (2019). An overview on denial-of-service attacks in control systems: Attack models and security analyses. *Entropy*, *21*(2), 210.

Chakhchoukh, Y., & Ishii, H. (2016). Enhancing robustness to cyber-attacks in power systems through multiple least trimmed squares state estimations. *IEEE Transactions on Power Systems*, *31*(6), 4395–4405.

Chakhchoukh, Y., Liu, S., Sugiyama, M., & Ishii, H. (2016). Statistical outlier detection for diagnosis of cyber attacks in power state estimation. In *2016 IEEE power and energy society general meeting* (pp. 1–5). IEEE.

Chamanbaz, M., Dabbene, F., & Bouffanais, R. (2019). A physics-based attack detection technique in cyber-physical systems: A model predictive control co-design approach. In *2019 Australian & New Zealand control conference* (pp. 18–23). IEEE.

Chanfreut, P., Maestre, J. M., & Camacho, E. F. (2021). A survey on clustering methods for distributed and networked control systems. *Annual Reviews in Control*, http://dx.doi.org/10.1016/j.arcontrol.2021.08.002.

Chanfreut, P., Maestre, J. M., & Ishii, H. (2018). Vulnerabilities in distributed model predictive control based on Jacobi-Gauss decomposition. In *2018 European control conference* (pp. 2587–2592). IEEE.

Chanfreut, P., Sánchez-Amores, A., Maestre, J. M., & Camacho, E. F. (2021). Distributed model predictive control based on dual decomposition with neural-network-based warm start. In *2021 European control conference*. IEEE.

Chen, S., Wu, Z., & Christofides, P. D. (2020). Cyber-attack detection and resilient operation of nonlinear processes under economic model predictive control. *Computers & Chemical Engineering*, *136*, Article 106806.

Chen, C., Zhang, K., Ni, M., & Wang, Y. (2020). Cyber-attack-tolerant frequency control of power systems. *Journal of Modern Power Systems and Clean Energy*.

Cheng, R., Forbes, J. F., & Yip, W. (2007). Price-driven coordination method for solving plant-wide MPC problems. *Journal of Process Control*, *17*(5), 429–438.

Christofides, P. D., Scattolini, R., de la Pena, D. M., & Liu, J. (2013). Distributed model predictive control: A tutorial review and future research directions. *Computers & Chemical Engineering*, *51*, 21–41.

Cui, S., Han, Z., Kar, S., Kim, T. T., Poor, H. V., & Tajer, A. (2012). Coordinated data-injection attack and detection in the smart grid: A detailed look at enriching detection solutions. *IEEE Signal Processing Magazine*, *29*(5), 106–115.

Dán, G., & Sandberg, H. (2010). Stealth attacks and protection schemes for state estimators in power systems. In *2010 first IEEE international conference on smart grid communications* (pp. 214–219). IEEE.

Darup, M. S. (2020). Encrypted model predictive control in the cloud. In *Privacy in dynamical systems* (pp. 231–265). Springer.

Darup, M. S., Redder, A., & Quevedo, D. E. (2018). Encrypted cloud-based MPC for linear systems with input constraints. *IFAC-PapersOnLine*, *51*(20), 535–542.

Darup, M. S., Redder, A., Shames, I., Farokhi, F., & Quevedo, D. (2017). Towards encrypted MPC for linear constrained systems. *IEEE Control Systems Letters*, *2*(2), 195–200.

De Oliveira, L. B., & Camponogara, E. (2010). Multi-agent model predictive control of signaling split in urban traffic networks. *Transportation Research Part C (Emerging Technologies)*, *18*(1), 120–139.

De Persis, C., & Tesi, P. (2015). Input-to-state stabilizing control under denial-of-service. *IEEE Transactions on Automatic Control*, *60*(11), 2930–2944.

Dibaji, S. M., Pirani, M., Flamholz, D. B., Annaswamy, A. M., Johansson, K. H., & Chakrabortty, A. (2019). A systems and control perspective of CPS security. *Annual Reviews in Control*, *47*, 394–411.

Ding, D., Han, Q.-L., Wang, Z., & Ge, X. (2019). A survey on model-based distributed control and filtering for industrial cyber-physical systems. *IEEE Transactions on Industrial Informatics*, *15*(5), 2483–2499.

Ding, D., Han, Q.-L., Xiang, Y., Ge, X., & Zhang, X.-M. (2018). A survey on security control and attack detection for industrial cyber-physical systems. *Neurocomputing*, *275*, 1674–1683.

Doan, M. D., Keviczky, T., & De Schutter, B. (2011). An iterative scheme for distributed model predictive control using Fenchel's duality. *Journal of Process Control*, *21*(5), 746–755.

Farina, M., & Scattolini, R. (2012). Distributed predictive control: A non-cooperative algorithm with neighbor-to-neighbor communication for linear systems. *Automatica*, *48*(6), 1088–1096.

Farokhi, F., Shames, I., & Johansson, K. H. (2014). Distributed MPC via dual decomposition and alternative direction method of multipliers. In *Distributed model predictive control made easy* (pp. 115–131). Springer.

Farraj, A., Hammad, E., & Kundur, D. (2017). A distributed control paradigm for smart grid to address attacks on data integrity and availability. *IEEE Transactions on Signal and Information Processing over Networks*, *4*(1), 70–81.

Fele, F., Maestre, J. M., & Camacho, E. F. (2017). Coalitional control: Cooperative game theory and control. *IEEE Control Systems Magazine*, *37*(1), 53–69.

Feng, S., Cetinkaya, A., Ishii, H., Tesi, P., & De Persis, C. (2020). Networked control under DoS attacks: Tradeoffs between resilience and data rate. *IEEE Transactions on Automatic Control*, *66*(1), 460–467.

Feng, S., & Ishii, H. (2020). Dynamic quantized consensus of general linear multi-agent systems under denial-of-service attacks. *IFAC-PapersOnLine*, *53*(2), 3533–3538.

Feng, C., Li, T., Zhu, Z., & Chana, D. (2017). A deep learning-based framework for conducting stealthy attacks in industrial control systems. arXiv e-prints, arXiv–1709.

Ferramosca, A., Limón, D., Alvarado, I., & Camacho, E. F. (2013). Cooperative distributed MPC for tracking. *Automatica*, *49*(4), 906–914.

Ferranti, L., Wan, Y., & Keviczky, T. (2019). Fault-tolerant reference generation for model predictive control with active diagnosis of elevator jamming faults. *International Journal of Robust and Nonlinear Control*, *29*(16), 5412–5428.

Ferrari, R. M., & Teixeira, A. M. (2020). A switching multiplicative watermarking scheme for detection of stealthy cyber-attacks. *IEEE Transactions on Automatic Control*.

Francisco, M., Mezquita, Y., Revollar, S., Vega, P., & De Paz, J. F. (2019). Multi-agent distributed model predictive control with fuzzy negotiation. *Expert Systems with Applications*, *129*, 68–83.

Franze, G., Lucia, W., & Tedesco, F. (2021). Resilient model predictive control for constrained cyber-physical systems subject to severe attacks on the communication channels. *IEEE Transactions on Automatic Control*.

Franzè, G., Tedesco, F., & Famularo, D. (2020). Resilience against replay attacks: A distributed model predictive control scheme for networked multi-agent systems. *IEEE/CAA Journal of Automatica Sinica*, *8*(3), 628–640.

Franzè, G., Tedesco, F., & Lucia, W. (2019). Resilient control for cyber-physical systems subject to replay attacks. *IEEE Control Systems Letters*, *3*(4), 984–989.

Gallo, A. J., Turan, M. S., Boem, F., Parisini, T., & Ferrari-Trecate, G. (2020). A distributed cyber-attack detection scheme with application to DC microgrids. *IEEE Transactions on Automatic Control*, *65*(9), 3800–3815.

Giselsson, P., Doan, M. D., Keviczky, T., De Schutter, B., & Rantzer, A. (2013). Accelerated gradient methods and dual decomposition in distributed model predictive control. *Automatica*, *49*(3), 829–833.

Griffioen, P., Romagnoli, R., Krogh, B. H., & Sinopoli, B. (2019). Secure networked control via software rejuvenation. In *2019 IEEE 58th conference on decision and control* (pp. 3878–3884). IEEE.

Hammami, D. E. H., Maraoui, S., & Bouzrara, K. (2020). Nonlinear distributed model predictive control with dual decomposition and event-based communication approach. *Transactions of the Institute of Measurement and Control*, *42*(15), 2929–2940.

Hu, S., Yue, D., Cheng, Z., Tian, E., Xie, X., & Chen, X. (2020). Co-design of dynamic event-triggered communication scheme and resilient observer-based control under aperiodic DoS attacks. *IEEE Transactions on Cybernetics*.

Humayed, A., Lin, J., Li, F., & Luo, B. (2017). Cyber-physical systems security–A survey. *IEEE Internet of Things Journal*, *4*(6), 1802–1831.

Isozaki, Y., Yoshizawa, S., Fujimoto, Y., Ishii, H., Ono, I., Onoda, T., et al. (2015). Detection of cyber attacks against voltage control in distribution power grids with pvs. *IEEE Transactions on Smart Grid*, *7*(4), 1824–1835.

Jang-Jaccard, J., & Nepal, S. (2014). A survey of emerging threats in cybersecurity. *Journal of Computer and System Sciences*, *80*(5), 973–993.

Jia, Y., Meng, K., Wu, K., Sun, C., & Dong, Z. Y. (2020). Optimal load frequency control for networked power systems based on distributed economic MPC. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*.

Jiang, J., & Yu, X. (2012). Fault-tolerant control systems: A comparative study between active and passive approaches. *Annual Reviews in Control*, *36*(1), 60–72.

Johansson, K. H. (2000). The quadruple-tank process: A multivariable laboratory process with an adjustable zero. *IEEE Transactions on Control Systems Technology*, *8*(3), 456–465.

Kang, J., Xiong, Z., Niyato, D., Ye, D., Kim, D. I., & Zhao, J. (2019). Toward secure blockchain-enabled internet of vehicles: Optimizing consensus management using reputation and contract theory. *IEEE Transactions on Vehicular Technology*, *68*(3), 2906–2920.

Kikuchi, K., Cetinkaya, A., Hayakawa, T., & Ishii, H. (2017). Stochastic communication protocols for multi-agent consensus under jamming attacks. In *2017 IEEE 56th annual conference on decision and control* (pp. 1657–1662). IEEE.

Kordestani, M., Safavi, A. A., & Saif, M. (2021). Recent survey of large-scale systems: Architectures, controller strategies, and industrial applications. *IEEE Systems Journal*.

Kushal, T. R. B., Lai, K., & Illindala, M. S. (2018). Risk-based mitigation of load curtailment cyber attack using intelligent agents in a shipboard power system. *IEEE Transactions on Smart Grid*, *10*(5), 4741–4750.

Kushner, D. (2013). The real story of Stuxnet. *IEEE Spectrum*, *50*(3), 48–53.

Lavrov, E. A., Volosiuk, A. A., Pasko, N. B., Gonchar, V. P., & Kozhevnikov, G. K. (2018). Computer simulation of discrete human-machine interaction for providing reliability and cybersecurity of critical systems. In *2018 third international conference on human factors in complex technical systems and environments* (pp. 67–70). IEEE.

Lei, K., Zhang, Q., Xu, L., & Qi, Z. (2018). Reputation-based byzantine fault-tolerance for consortium blockchain. In *2018 IEEE 24th international conference on parallel and distributed systems* (pp. 604–611). IEEE.

Li, K., Bian, Y., Li, S. E., Xu, B., & Wang, J. (2020). Distributed model predictive control of multi-vehicle systems with switching communication topologies. *Transportation Research Part C (Emerging Technologies)*, *118*, Article 102717.

Li, X.-M., Zhou, Q., Li, P., Li, H., & Lu, R. (2019). Event-triggered consensus control for multi-agent systems against false data-injection attacks. *IEEE Transactions on Cybernetics*, *50*(5), 1856–1866.

Limón, D., Alvarado, I., Alamo, T., & Camacho, E. F. (2008). MPC for tracking piecewise constant references for constrained linear systems. *Automatica*, *44*(9), 2382–2387.

Liu, A., & Bai, L. (2018). Distributed model predictive control for wide area measurement power systems under malicious attacks. *IET Cyber-Physical Systems: Theory & Applications*, *3*(3), 111–118.

Liu, Y., Chen, Y., & Li, M. (2020). Dynamic event-based model predictive load frequency control for power systems under cyber attacks. *IEEE Transactions on Smart Grid*, *12*(1), 715–725.

Liu, Y., Chen, Y., Li, M., & Wan, Z. (2020). MPC for the cyber-physical system with deception attacks. In *2020 Chinese control and decision conference* (pp. 3847–3852). IEEE.

Liu, S., Song, Y., Wei, G., & Huang, X. (2017). RMPC-based security problem for polytopic uncertain system subject to deception attacks and persistent disturbances. *IET Control Theory & Applications*, *11*(10), 1611–1618.

Liu, F., Wang, C., & Geng, Q. (2020). Observer-based MPC for NCS with actuator saturation and DoS attacks via interval type-2 T–S fuzzy model. *IET Control Theory & Applications*, *14*(20), 3537–3546.

Lješnjanin, M., Quevedo, D. E., & Nešić, D. (2014). Packetized MPC with dynamic scheduling constraints and bounded packet dropouts. *Automatica*, *50*(3), 784–797.

Ma, Y., Anderson, G., & Borrelli, F. (2011). A distributed predictive control approach to building temperature regulation. In *Proceedings of the 2011 American control conference* (pp. 2089–2094). IEEE.

MacGregor, J., & Cinar, A. (2012). Monitoring, fault diagnosis, fault-tolerant control and optimization: Data driven methods. *Computers & Chemical Engineering*, *47*, 111–120.

Maestre, J. M., Muñoz De La Peña, D., & Camacho, E. F. (2011). Distributed model predictive control based on a cooperative game. *Optimal Control Applications & Methods*, *32*(2), 153–176.

Maestre, J., De La Pena, D. M., Camacho, E., & Alamo, T. (2011). Distributed model predictive control based on agent negotiation. *Journal of Process Control*, *21*(5), 685–697.

Maestre, J. M., Trodden, P. A., & Ishii, H. (2018). A distributed model predictive control scheme with robustness against noncompliant controllers. In *2018 IEEE conference on decision and control* (pp. 3704–3709). IEEE.

Maestre, J. M., Velarde, P., Ishii, H., & Negenborn, R. R. (2021). Scenario based defense mechanism against vulnerabilities in Lagrange-based DMPC. *Control Engineering Practice*.

Masero, E., Francisco, M., Maestre, J. M., Revollar, S., & Vega, P. (2021). Hierarchical distributed model predictive control based on fuzzy negotiation. *Expert Systems with Applications*, *176*, Article 114836.

Maxim, A., & Caruntu, C.-F. (2021). A coalitional distributed model predictive control perspective for a cyber-physical multi-agent application. *Sensors*, *21*(12), 4041.

Mc Namara, P., Negenborn, R. R., De Schutter, B., & Lightbody, G. (2012). Optimal coordination of a multiple HVDC link system using centralized and distributed control. *IEEE Transactions on Control Systems Technology*, *21*(2), 302–314.

Mc Namara, P., Negenborn, R. R., De Schutter, B., Lightbody, G., & McLoone, S. (2016). Distributed MPC for frequency regulation in multi-terminal HVDC grids. *Control Engineering Practice*, *46*, 176–187.

Mishra, P. K., Chatterjee, D., & Quevedo, D. E. (2017). Stabilizing stochastic predictive control under Bernoulli dropouts. *IEEE Transactions on Automatic Control*, *63*(6), 1579–1590.

Mishra, P. K., Quevedo, D. E., & Chatterjee, D. (2016). Dropout feedback parametrized policies for stochastic predictive controller. *IFAC-PapersOnLine*, *49*(18), 59–64.

Mo, Y., Kim, T. H.-J., Brancik, K., Dickinson, D., Lee, H., Perrig, A., et al. (2011). Cyber–physical security of a smart grid infrastructure. *Proceedings of the IEEE*, *100*(1), 195–209.

Mo, Y., & Sinopoli, B. (2009). Secure control against replay attacks. In *2009 47th annual Allerton conference on communication, control, and computing* (pp. 911–918). IEEE.

Mo, Y., Weerakkody, S., & Sinopoli, B. (2015). Physical authentication of control systems: Designing watermarked control inputs to detect counterfeit sensor outputs. *IEEE Control Systems Magazine*, *35*(1), 93–109.

Moradmand, A., Ramezani, A., Nezhad, H. S., & Sardashti, A. (2019). Fault tolerant Kalman filter-based distributed predictive control in power systems under governor malfunction. In *2019 6th international conference on control, instrumentation and automation* (pp. 1–6). IEEE.

Naghavi, S. V., Safavi, A. A., & Kazerooni, M. (2014). Decentralized fault tolerant model predictive control of discrete-time interconnected nonlinear systems. *Journal of the Franklin Institute*, *351*(3), 1644–1656.

Negenborn, R. R., & Maestre, J. M. (2014). Distributed model predictive control: An overview and roadmap of future research opportunities. *IEEE Control Systems Magazine*, *34*(4), 87–97.

Negenborn, R. R., van Overloop, P.-J., Keviczky, T., & De Schutter, B. (2009). Distributed model predictive control of irrigation canals. *Networks & Heterogeneous Media*, *4*(2), 359.

Nishino, H., & Ishii, H. (2014). Distributed detection of cyber attacks and faults for power systems. *IFAC Proceedings Volumes*, *47*(3), 11932–11937.

Nofer, M., Gomber, P., Hinz, O., & Schiereck, D. (2017). Blockchain. *Business & Information Systems Engineering*, *59*(3), 183–187.

Olfati-Saber, R., & Murray, R. M. (2004). Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control*, *49*(9), 1520–1533.

Pasqualetti, F., Dörfler, F., & Bullo, F. (2013). Attack detection and identification in cyber-physical systems. *IEEE Transactions on Automatic Control*, *58*(11), 2715–2729.

Peng, C., & Sun, H. (2020). Switching-like event-triggered control for networked control systems under malicious denial of service attacks. *IEEE Transactions on Automatic Control*, *65*(9), 3943–3949.

Pierron, T., Arauz, T., Maestre, J., Cetinkaya, A., & Maniu, C. S. (2020). Tree-based model predictive control for jamming attacks. In *2020 European control conference* (pp. 948–953). IEEE.

Qi, W., Liu, J., & Christofides, P. D. (2011). A distributed control framework for smart grid development: Energy/water system optimal operation and electric grid integration. *Journal of Process Control*, *21*(10), 1504–1516.

Qin, S. J., & Badgwell, T. A. (2003). A survey of industrial model predictive control technology. *Control Engineering Practice*, *11*(7), 733–764.

Qin, Y., Zhao, Y., Huang, K., Tian, Y.-C., & Zhou, C. (2020). Dynamic model predictive control for constrained cyber-physical systems subject to actuator attacks. *International Journal of Systems Science*, 1–11.

Qiu, Q., Yang, F., & Zhu, Y. (2021). Cyber-attack localisation and tolerant control for microgrid energy management system based on set-membership estimation. *International Journal of Systems Science*, *52*(6), 1206–1222.

Quevedo, D. E., & Ahlén, A. (2008). A predictive power control scheme for energy efficient state estimation via wireless sensor networks. In *2008 47th IEEE conference on decision and control* (pp. 1103–1108). IEEE.

Quevedo, D. E., Mishra, P. K., Findeisen, R., & Chatterjee, D. (2015). A stochastic model predictive controller for systems with unreliable communications. *IFAC-PapersOnLine*, *48*(23), 57–64.

Quevedo, D. E., & Nešić, D. (2010). Input-to-state stability of packetized predictive control over unreliable networks affected by packet-dropouts. *IEEE Transactions on Automatic Control*, *56*(2), 370–375.

Quevedo, D. E., & Nešić, D. (2012). Robust stability of packetized predictive control of nonlinear systems with disturbances and Markovian packet losses. *Automatica*, *48*(8), 1803–1811.

Raimondo, D. M., Marseglia, G. R., Braatz, R. D., & Scott, J. K. (2013). Fault-tolerant model predictive control with active fault isolation. In *2013 conference on control and fault-tolerant systems* (pp. 444–449). IEEE.

Rantzer, A. (2009). Dynamic dual decomposition for distributed control. In *2009 American control conference* (pp. 884–888). IEEE.

Rawlings, J. B., & Stewart, B. T. (2008). Coordinating multiple optimization-based controllers: New opportunities and challenges. *Journal of Process Control*, *18*(9), 839–845.

Riverso, S., Boem, F., Ferrari-Trecate, G., & Parisini, T. (2016). Plug-and-play fault detection and control-reconfiguration for a class of nonlinear large-scale constrained systems. *IEEE Transactions on Automatic Control*, *61*(12), 3963–3978.

Riverso, S., Farina, M., & Ferrari-Trecate, G. (2014). Plug-and-play model predictive control based on robust control invariant sets. *Automatica*, *50*(8), 2179–2186.

Romagnoli, R., Griffioen, P., Krogh, B. H., & Sinopoli, B. (2020). Software rejuvenation under persistent attacks in constrained environments. *IFAC-PapersOnLine*, *53*(2), 4088–4094.

Romagnoli, R., Krogh, B. H., & Sinopoli, B. (2019a). Design of software rejuvenation for CPS security using invariant sets. In *2019 American control conference* (pp. 3740–3745). IEEE.

Romagnoli, R., Krogh, B. H., & Sinopoli, B. (2019b). Safety and liveness of software rejuvenation for secure tracking control. In *2019 18th European control conference* (pp. 2215–2220). IEEE.

Rotem-Gal-Oz, A. (2006). Fallacies of distributed computing explained. URL http://www.rgoarchitects.com/Files/fallacies.pdf20.

de Sá, A. O., da Costa Carmo, L. F. R., & Machado, R. C. (2017). Covert attacks in cyber-physical control systems. *IEEE Transactions on Industrial Informatics*, *13*(4), 1641–1651.

Sahoo, S., Mishra, S., Peng, J. C.-H., & Dragičević, T. (2018). A stealth cyber-attack detection strategy for DC microgrids. *IEEE Transactions on Power Electronics*, *34*(8), 8162–8174.

Sánchez, H. S., Rotondo, D., Escobet, T., Puig, V., & Quevedo, J. (2019). Bibliographical review on cyber attacks from a control oriented perspective. *Annual Reviews in Control*, *48*, 103–128.

Savino, H. J., dos Santos, C. R., Souza, F. O., Pimenta, L. C., de Oliveira, M., & Palhares, R. M. (2015). Conditions for consensus of multi-agent systems with time-delays and uncertain switching topology. *IEEE Transactions on Industrial Electronics*, *63*(2), 1258–1267.

Scattolini, R. (2009). Architectures for distributed and hierarchical model predictive control–A review. *Journal of Process Control*, *19*(5), 723–731.

Schiffer, J., Dörfler, F., & Fridman, E. (2017). Robustness of distributed averaging control in power systems: Time delays & dynamic communication topology. *Automatica*, *80*, 261–271.

Schwab, W., & Poujol, M. (2018). *The state of industrial cybersecurity 2018*: *Trend study kaspersky reports 33*.

Sharma, D. D., Singh, S., Lin, J., & Foruzan, E. (2017). Agent-based distributed control schemes for distributed energy storage systems under cyber attacks. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, *7*(2), 307–318.

Smith, R. S. (2011). A decoupled feedback structure for covertly appropriating networked control systems. *IFAC Proceedings Volumes*, *44*(1), 90–95.

Stewart, B. T., Venkat, A. N., Rawlings, J. B., Wright, S. J., & Pannocchia, G. (2010). Cooperative distributed model predictive control. *Systems & Control Letters*, *59*(8), 460–469.

Stewart, B. T., Wright, S. J., & Rawlings, J. B. (2011). Cooperative distributed model predictive control for nonlinear systems. *Journal of Process Control*, *21*(5), 698–704.

Subramanian, K., Rawlings, J. B., Maravelias, C. T., Flores-Cerrillo, J., & Megan, L. (2013). Integration of control theory and scheduling methods for supply chain management. *Computers & Chemical Engineering*, *51*, 4–20.

Sun, Y.-C., & Yang, G.-H. (2019). Robust event-triggered model predictive control for cyber-physical systems under denial-of-service attacks. *International Journal of Robust and Nonlinear Control*, *29*(14), 4797–4811.

Sun, Q., Zhang, K., & Shi, Y. (2019). Resilient model predictive control of cyber–physical systems under dos attacks. *IEEE Transactions on Industrial Informatics*, *16*(7), 4920–4927.

Tanaka, T., & Gupta, V. (2016). Incentivizing truth-telling in MPC-based load frequency control. In *2016 IEEE 55th conference on decision and control* (pp. 1549–1555). IEEE.

Teixeira, A., Shames, I., Sandberg, H., & Johansson, K. H. (2015). A secure control framework for resource-limited adversaries. *Automatica*, *51*, 135–148.

Thames, L., & Schaefer, D. (2017). *Cybersecurity for industry 4.0*. Springer.

Tian, E., & Peng, C. (2020). Memory-based event-triggering H∞ load frequency control for power systems under deception attacks. *IEEE Transactions on Cybernetics*, *50*(11), 4610–4618.

Tiwari, A., Smolka, S. A., Esterle, L., Lukina, A., Yang, J., & Grosu, R. (2017). Attacking the V: On the resiliency of adaptive-horizon MPC. In *International symposium on automated technology for verification and analysis* (pp. 446–462). Springer.

Trodden, P. A., & Maestre, J. M. (2017). Distributed predictive control with minimization of mutual disturbances. *Automatica*, *77*, 31–43.

Trodden, P. A., Maestre, J., & Ishii, H. (2020). Actuation attacks on constrained linear systems: A set-theoretic analysis. *IFAC-PapersOnLine*, *53*(2), 6963–6968.

Van Overloop, P., Maestre, J., Sadowska, A. D., Camacho, E. F., & De Schutter, B. (2015). Human-in-the-loop model predictive control of an irrigation canal [applications of control]. *IEEE Control Systems Magazine*, *35*(4), 19–29.

Velarde, P., Maestre, J. M., Ishii, H., & Negenborn, R. R. (2017). Scenario-based defense mechanism for distributed model predictive control. In *2017 IEEE 56th annual conference on decision and control* (pp. 6171–6176). IEEE.

Velarde, P., Maestre, J. M., Ishii, H., & Negenborn, R. R. (2018). Vulnerabilities in Lagrange-based distributed model predictive control. *Optimal Control Applications & Methods*, *39*(2), 601–621.

Venkat, A. N., Hiskens, I. A., Rawlings, J. B., & Wright, S. J. (2008). Distributed MPC strategies with application to power system automatic generation control. *IEEE Transactions on Control Systems Technology*, *16*(6), 1192–1206.

Wakaiki, M., Cetinkaya, A., & Ishii, H. (2019). Stabilization of networked control systems under DoS attacks and output quantization. *IEEE Transactions on Automatic Control*, *65*(8), 3560–3575.

Wang, T., Gao, H., & Qiu, J. (2016). A combined fault-tolerant and predictive control for network-based industrial processes. *IEEE Transactions on Industrial Electronics*, *63*(4), 2529–2536.

Wang, Y., & Ishii, H. (2019). A distributed model predictive scheme for resilient consensus with input constraints. In *2019 IEEE conference on control technology and applications* (pp. 349–354). IEEE.

Wang, J., Song, Y., Liu, S., & Zhang, S. (2016). Security in H_2–sense for polytopic uncertain systems with attacks based on model predictive control. *Journal of the Franklin Institute*, *353*(15), 3769–3785.

Wei, L., Wu, J., Long, C., & Lin, Y.-B. (2019). The convergence of IoE and blockchain: Security challenges. *IT Professional*, *21*(5), 26–32.

Worthmann, K., Kellett, C. M., Braun, P., Grüne, L., & Weller, S. R. (2015). Distributed and decentralized control of residential energy systems incorporating battery storage. *IEEE Transactions on Smart Grid*, *6*(4), 1914–1923.

Wu, Z., Albalawi, F., Zhang, J., Zhang, Z., Durand, H., & Christofides, P. D. (2018). Detecting and handling cyber-attacks in model predictive control of chemical processes. *Mathematics*, *6*(10), 173.

Wu, C.-H. J., & Irwin, J. D. (2016). *Introduction to computer networks and cybersecurity*. CRC Press.

Wu, Y., Zhang, X., & Sun, H. (2021). A multi-time-scale autonomous energy trading framework within distribution networks based on blockchain. *Applied Energy*, *287*, Article 116560.

Xiao, S., Ge, X., Han, Q.-L., & Zhang, Y. (2020). Distributed resilient estimator design for positive systems under topological attacks. *IEEE Transactions on Cybernetics*.

Xu, Y., Yuan, Y., Yang, H., & Zhou, D. (2021). The safety region-based model predictive control for discrete-time systems under deception attacks. *International Journal of Systems Science*, 1–17.

Yaghooti, B., Romagnoli, R., & Sinopoli, B. (2021). Physical watermarking for replay attack detection in continuous-time systems. *European Journal of Control*, 0947–3580.

Yang, H., Li, Y., Dai, L., & Xia, Y. (2019). MPC-based defense strategy for distributed networked control systems under DoS attacks. *Systems & Control Letters*, *128*, 9–18.

Yang, H., Xu, H., Xia, Y., & Zhang, J. (2020). Stability analysis on networked control systems under double attacks with predictive control. *International Journal of Robust and Nonlinear Control*, *30*(4), 1549–1563.

Yazdanian, M., & Mehrizi-Sani, A. (2014). Distributed control techniques in microgrids. *IEEE Transactions on Smart Grid*, *5*(6), 2901–2909.

Zafra-Cabeza, A., Maestre, J., Ridao, M. A., Camacho, E. F., & Sánchez, L. (2011). A hierarchical distributed model predictive control approach to irrigation canals: A risk mitigation perspective. *Journal of Process Control*, *21*(5), 787–799.

Zarei, M. E., Gupta, M., Ramirez, D., & Martinez-Rodrigo, F. (2019). Switch fault tolerant model-based predictive control (MPC) of a VSC connected to the grid. *IEEE Journal of Emerging and Selected Topics in Power Electronics*.

Zeldovich, N. (2014). 6.858 computer systems security. In *MIT lecture notes, Fall*.

Zhang, L., Xie, W., & Lian, Y. (2020). Distributed fault detection of nonlinear process systems with senor failures. *IFAC-PapersOnLine*, *53*(2), 2544–2549.

Zhu, M., & Martínez, S. (2015). *Distributed optimization-based control of multi-agent networks in complex environments*. Springer.