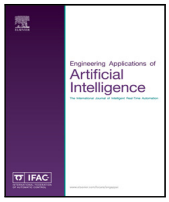
Engineering Applications of Artificial Intelligence 118 (2023) 105666

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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



# A fast implementation of coalitional model predictive controllers based on machine learning: Application to solar power plants

Eva Masero *, Sara Ruiz-Moreno, José Ramón D. Frejo, José M. Maestre, Eduardo F. Camacho

*Departament of Ingeniería de Sistemas y Automática, Universidad de Sevilla, C/ Camino de los Descubrimientos s/n., Sevilla 41092, Spain*

ARTICLE INFO

*Keywords:*
Neural networks
Artificial intelligence
Non-linear model predictive control
Coalitional control
Multi-agent systems
Solar thermal applications

ABSTRACT

This article proposes a real-time implementation of distributed model predictive controllers to maximize the thermal energy generated by parabolic trough collector fields. For this control strategy, we consider that each loop of the solar collector field is individually managed by a controller, which can form coalition with other controllers to attain its local goals while contributing to the overall objective. The formation of coalitions is based on a market-based mechanism in which the heat transfer fluid is traded. To relieve the computational burden online, we propose a learning-based approach that approximates optimization problems so that the controller can be applied in real time. Finally, simulations in a 100-loop solar collector field are used to assess the coalitional strategy based on neural networks in comparison with the coalitional model predictive control. The results show that the coalitional strategy based on neural networks provides a reduction in computing time of up to 99.74% and a minimal reduction in performance compared to the coalitional model predictive controller used as the baseline.

## 1. Introduction

The transition to low-carbon energy systems is changing the global energy landscape. Renewable energy accounted for almost 30% of the global power mix in 2021 and is expected to increase in the near future (McKinsey&Company, 2022). Since solar power is one of the most abundant and cost competitive energy sources, many researchers have paid special attention to it over the past few decades (Camacho et al., 2012; Kabir et al., 2018; Hayat et al., 2019). Electricity can be generated directly from sunlight using photovoltaic cells (PV) or indirectly by concentrating solar power (CSP) systems, which employ solar radiation and a heat transfer fluid (HTF) to produce steam that drives a turbine generator. For electricity providers, CPS systems lead to greater grid stability because they can produce in zero- or low-sunlight conditions using thermal energy storage (TES) systems. Another benefit of CPS systems is its use for industrial activities that require super-heated water (Tasmin et al., 2022). Among CPS technologies (Islam et al., 2018), we can find parabolic dishes (Bianchini et al., 2019), solar power towers (Awan et al., 2020), linear fresnel collectors (Pulido-Iparraguirre et al., 2019), and parabolic trough collector fields (Yılmaz and Mwesigye, 2018). In this work, we focus on the latter and maximize the thermal energy generated despite daylight changes by controlling inlet valves at the beginning of the collector loops. Other control objectives such as maximizing electrical power and tracking an outlet temperature setpoint could be also considered.

A comprehensive overview of control techniques for controlling solar power plants can be found in the surveys presented by Camacho et al. (2007a,b). Among all of them, model predictive control (MPC), which optimizes the control input over a finite horizon using a model to predict the system evolution, is second to none due to its remarkable characteristics such as the possibility to handle constraints, delays, and multi-variable systems (Mayne, 2014). Much research has been dedicated to the applications of MPC to solar collector plants. For example, Alsharkawi and Rossiter (2016) detail a dual MPC based on a linear model with disturbance rejection, Vasallo et al. (2017) proposes an economic MPC to address the scheduling problem in CSP plants with TES, and Pipino et al. (2020) employs an MPC based on a linear parameter varying model. Due to the non-linear dynamics of plants, non-linear model predictive control (NMPC) approaches are also found in the literature. Flores et al. (2005) proposes a fuzzy NMPC to account for complex requirements, Gálvez-Carrillo et al. (2009) combines the NMPC with Smith predictor to tackle dead times, and Falugi and Mayne (2011) present a tube-based model predictive control to deal with non-linear dynamics and unstructured uncertainty.

Most studies focus on optimizing the HTF flow of the solar collector field using a centralized setting. However, centralized approaches present limitations in large-scale fields due to the complexity of the corresponding models and the high computing time requirements. An alternative is to use distributed MPC (DMPC) approaches, which divide

* Corresponding author.
*E-mail address:* evamasero@us.es (E. Masero).

https://doi.org/10.1016/j.engappai.2022.105666
Received 12 August 2022; Received in revised form 7 November 2022; Accepted 21 November 2022
Available online 12 December 2022

the overall control problem into sub-problems solved by local controllers that can coordinate with each other (Negenborn and Maestre, 2014). Several distributed strategies that consider valves at the beginning of loops have been explored. For instance, De Araújo Elias et al. (2019) proposes a distributed MPC architecture for large-scale fields in parallel configuration, where the field is divided into sub-fields that are locally controlled and can be disconnected to prevent heat losses. In (Navas et al., 2018), a non-cooperative distributed approach in which the flow rate of each loop is determined by a local MPC is presented, and Frejo and Camacho (2020) details a logic-based distributed MPC that provides performance close to the centralized MPC with less computational burden. Nevertheless, the fixed cooperation structure between controllers may limit the scalability due to increasing cooperation efforts and may not provide the best adaptation to disturbances as dusty collectors and partial clouds, which generate high sunlight differences between loops in large-size solar fields. A further strategy to overcome this drawback is coalitional model predictive control (Baldivieso-Monasterios and Trodden, 2021; Chanfreut et al., 2021), which fosters time-varying groups of local agents that coordinate their control actions to improve performance with an adequate degree of cooperation. This strategy results in an online partitioning problem to adapt the control structure to changing operating conditions (Barreiro-Gomez et al., 2019). The partition can be obtained in a fully distributed way or by a supervisor with overall information of the system. Regarding the former, (Fele et al., 2018) decides the cooperation structure autonomously by a negotiation process between local agents. An example of hierarchical structure can be found in (Masero et al., 2021), where a supervisor layer sets coalitions and the best moment to implement them. Applications of coalitional control cover fleet of robots (Xiao and Chen, 2021), electricity networks (La Bella et al., 2022), wind farms (Ye et al., 2019), telecommunication networks (Masero et al., 2020), vehicle platooning (Maxim and Caruntu, 2022), water networks (Maestre et al., 2021), microgrids (Ananduta and Ocampo-Martinez, 2021), and solar power plants (Masero et al., 2022), among others.

Likewise, to handle real-time requirements, one may need to employ methods as explicit MPC (Alessio and Bemporad, 2009) or machine learning (Drgoňa et al., 2018). This latter group includes artificial neural networks (ANNs), which can approximate the MPC policy in a compact form. Depending on the training criteria, learning can be classified as: (i) supervised if label data are mapped to obtain known outputs (Bemporad, 2022), (ii) unsupervised if patterns are extracted from untagged data, *e.g.*, for fault diagnosis (Vaish et al., 2021), and (ii) reinforcement when learning stems from the feedback provided by interaction with the environment (Pane et al., 2019). Machine learning techniques are present in vast applications such as robotics (Soriano et al., 2020), physical problems (Edalatifar et al., 2022), medicine (Ngiam and Khor, 2019), education (Kuleto et al., 2021), mechanical structure (Safa and Kachitvichyanukul, 2019; Shariati et al., 2019), energy efficiency of buildings (Yang et al., 2020), thermal energy (Azadeh et al., 2011; Lee et al., 2022), etc. However, few studies about machine learning for controlling solar plants are found in the literature. In (Sun et al., 2017), it is designed a controller based on artificial neural networks that approximates the PI controller of a residential solar CV system; Ceusters et al. (2021) presents a reinforcement learning strategy that outperforms centralized MPC to manage multi-energy systems; and Ruiz-Moreno et al. (2021) proposes an artificial neural network to approximate the local NMPC controller of a loop in solar collector fields.

To cover the gap of studies for controlling solar collector plants, this article presents supervised learning control based on coalitional model predictive control to maximize thermal energy in solar collector fields. The hierarchical coalitional NMPC used as a baseline is proposed in (Masero et al., 2022), where the solar field partition is selected online by an MPC-based supervisory layer that groups controllers offering and demanding HTF in a market, following an auction-based mechanism.

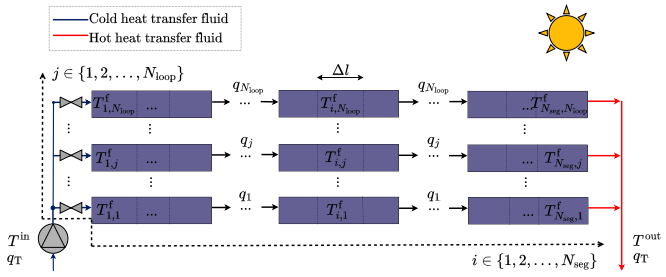



**Fig. 1.** Diagram of a parabolic-trough solar collector field.

Once the partition is selected, each coalition is controlled by an MPC controller. With respect to (Masero et al., 2022), the current work replaces all MPC optimizations with artificial neural networks to relieve the computational burden. Therefore, the partition is selected by a learning-based supervisory layer, and then each coalition is controlled by an artificial neural network that calculates its control input to achieve a performance similar to that obtained with model predictive controllers. The main contribution of this work is the novel combination of artificial neural networks in a distributed coalition setting, which allows the proposed strategy to substantially reduce computational requirements while adapting to current operating conditions. The motivation of this work stems from the need for achieving real-time optimizations to control solar collector plants. Finally, we test the proposed method with a simulated 100-loop solar collector field, which is an extension of the ACUREX field located in the Plataforma Solar de Almería (Yebra et al., 2010).

*Index of contents.* Section 2 presents the distributed parameter model and constraints of solar collector fields. Section 3 introduces the control objective, the coalitional NMPC problem, and the method for establishing the field partition. Section 4 explains the proposed algorithm based on artificial neural networks. Section 5 contains the simulation results and discussion. Finally, Section 6 concludes with the main findings.

## 2. Solar collector field description

The parabolic-trough solar collector field is composed of a series of parallel loops formed by concave mirrors that lead the solar direct normal irradiance (DNI) in their focal line tube, as shown in Fig. 1. The heat transfer fluid (HTF) that runs through the tube is heated up and led to the heat exchanger to produce steam.

### 2.1. Non-linear system dynamics

We consider that the set of loops $\mathcal{L} \triangleq \{1, \ldots, N_{\text{loop}}\}$ is divided into $S \triangleq \{1, \ldots, N_{\text{seg}}\}$ segments of length $\Delta l$ (see Fig. 1) to describe the field dynamics using a distributed parameter model whose variables are listed in Table 1. The model, which was previously proposed in (Camacho et al., 2012), is obtained by the following iterative two-stage process with finite differences:

1. Calculation of the HTF and metal tube temperatures of all segments $i \in S$ of loop $j \in \mathcal{L}$, assuming the fluid is in steady state:

$$\begin{aligned} T^{\text{f}}_{i,j}(k) = \; & T^{1\text{f}}_{i,j}(k-1) \\ & + \frac{\pi D^{\text{f}} H^{\text{t}}_{i,j}(k-1)\Delta t}{\rho^{\text{f}}_{i,j}(k-1) C^{\text{f}}_{i,j}(k-1) A^{\text{f}}} \left(T^{\text{m}}_{i,j}(k-1) - T^{1\text{f}}_{i,j}(k-1)\right), \\ T^{\text{m}}_{i,j}(k) = \; & T^{\text{m}}_{i,j}(k-1) + \frac{\Delta t}{\rho^{\text{m}} C^{\text{m}} A^{\text{m}}} \Big( \eta^{\text{col}}_{i,j} G_{i,j} I_{i,j}(k) \\ & - \pi D^{\text{m}} H^{\text{l}}_{i,j}(k-1)\left(T^{\text{m}}_{i,j}(k-1) - T^{\text{a}}(k)\right) \\ & - \pi D^{\text{f}} H^{\text{t}}_{i,j}(k-1)\left(T^{\text{m}}_{i,j}(k-1) - T^{1\text{f}}_{i,j}(k-1)\right) \Big), \end{aligned} \tag{1}$$

where superscripts $(\text{f}, \text{m})$ correspond to the fluid and the metal tube, respectively, and $k \in \mathbb{N}_+$ is the discrete-time instant, being the continuous-time instant $t = k\Delta t$.

**Table 1**
Summary of model parameters and variables.

| Symbol | Description | Unit |
|---|---|---|
| $\eta^{\text{col}}$ | Efficiency of collectors | – |
| $G$ | Aperture of collectors | m |
| $I$ | Solar direct normal irradiance | W/m$^2$ |
| $D^{\text{m}}$ | Outside diameter of the pipe | m |
| $D^{\text{f}}$ | Inside diameter of the pipe | m |
| $H^{\text{t}}$ | Coef. of transmission metal-fluid | W/(m$^2$ °C) |
| $H^{\text{l}}$ | Coef. of thermal losses | W/(m$^2$ °C) |
| $q$ | Flow rate | l/s |
| $\Delta t$ | Discretization time of the model | s |
| $\Delta l$ | Length of segments | m |
| $\rho^{\text{m}}, \rho^{\text{f}}$ | Densities | kg/m$^3$ |
| $C^{\text{m}}, C^{\text{f}}$ | Specific heat capacities | J/(kg °C) |
| $A^{\text{m}}, A^{\text{f}}$ | Cross-sectional areas | m$^2$ |
| $T^{\text{m}}, T^{\text{f}}, T^{\text{a}}$ | Temperatures | °C |

2. Correcting the HTF temperature considering the net energy carried by the fluid:

$$T^{\text{lf}}_{i,j}(k) = T^{\text{f}}_{i,j}(k) - \frac{q_j(k)\Delta t}{\Delta l\, A^{\text{f}}}\left(T^{\text{f}}_{i,j}(k) - T^{\text{f}}_{i-1,j}(k)\right). \tag{2}$$

We consider that the initial-segment temperature of each loop $j$ corresponds to the field input temperature ($T^{\text{f}}_{1,j}(k) = T^{\text{in}}(k)$ for all $k$). Moreover, disturbances such as solar DNI $I_{i,j}(k)$, inlet temperature of the field $T^{\text{in}}(k)$, and ambient temperature $T^{\text{a}}(k)$ are assumed to be measurable or estimated. The outlet temperature of the field is determined by considering the final-segment temperature of all the loops:

$$T^{\text{out}}(k) = \frac{\sum_{j\in\mathcal{L}} q_j(k)\; T^{\text{f}}_{N_{\text{seg}},j}(k)}{q_{\text{T}}(k)}, \tag{3}$$

with $q_{\text{T}}(k) = \sum_{j\in\mathcal{L}} q_j(k)$ being the total flow rate.

### 2.2. Collector field constraints

The thermal energy generated by the solar collector field depends, among other aspects, on the DNI received by the loops, their reflectivity, and their flow rates, which are locally restricted as follows:

$$q_j^{\min} \le q_j(k) \le q_j^{\max}, \tag{4}$$

where $q_j^{\max}$ is the maximum flow rate that relies on the highest allowed pressure drop, and $q_j^{\min} > 0$ is the minimum value, based on the Reynolds number, to ensure turbulent flow and obtain a homogeneous temperature of the HTF along the pipe. All loops are coupled by the total flow rate $q_{\text{T}}(k)$, which is also upper limited by $q_{\text{T}}^{\max}$.

Another aspect that can limit the flow rate of loops is the constraint of keeping the field outlet temperature (4) within its operational limits. Therefore, the outlet temperature of each loop is bounded by

$$T^{\text{f,min}} \le T^{\text{f}}_{N_{\text{seg}},j}(k) \le T^{\text{f,max}}, \tag{5}$$

where $T^{\text{f,min}}$ and $T^{\text{f,max}}$ are the minimum and maximum temperature values, respectively.

## 3. Coalitional control problem

We aim to maximize the thermal power generated by the solar collector field while maintaining the outlet temperature and the flow rate within its limits despite the variability of solar irradiance and reflectivity differences between loops. We consider flow rates of loops to be locally managed by an individual control agent that can cooperate in a dynamic manner with other loop controllers – yielding the so-called *coalitions* – to reduce their local costs while contributing to the global objective.

By cooperating, the control input of each coalition is the flow rate $q_c(k) = \sum_{j\in c} q_j(k)$, in which the thermal dynamics is assumed to be slower than the hydraulic dynamics. The coalition flow rate is also lower and upper bounded by $q_c^{\min} = \sum_{j\in c} q_j^{\min}$ and $q_c^{\max} = \sum_{j\in c} q_j^{\max}$, respectively.

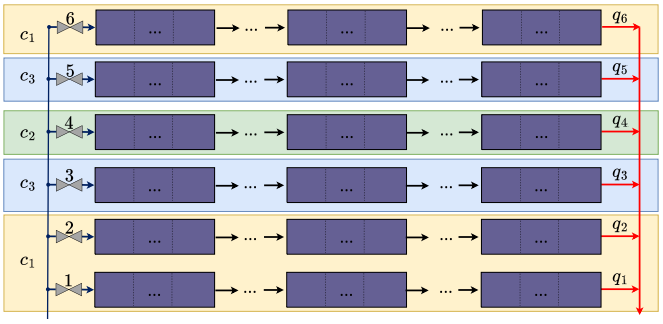



**Fig. 2.** Example of a solar collector field partitioned into a set of coalitions, where coalition $c_1$ is formed by loops 1, 2 and 6; $c_2$ is formed by loop 4; and $c_3$ is formed by loops 3 and 5.

**Definition 1** (*Cooperation Partition*)**.** A cooperation partition $\mathcal{P}(k)$ organizes the set of loops $\mathcal{L} = \{1, \dots, N_{\text{loop}}\}$ into a set of $C$ non-empty and non-overlapping coalitions of loops:

$$\mathcal{P}(k) \triangleq \{c_1, c_2, \dots, c_C\},$$

which cover all loops of the field: $\bigcup_{c\in\mathcal{P}(k)} c = \mathcal{L}$.

By definition, the number of coalitions can vary from $C = 1$, giving the grand coalition $c_1 = \mathcal{L}$, to $C = N_{\text{loop}}$, which gives the decentralized setting with each $c_j = j$ for $j \in \mathcal{L}$. The cardinality of a coalition $|c|$ denotes the size coalition, *i.e.,* the number of loops that cooperate in $c$. As an example, Fig. 2 shows a cooperation partition of a six-loop field at a time instant $k_1$, *i.e.,* $\mathcal{P}(k_1) = \{c_1, c_2, c_3\} = \{\{1,2,6\}, \{4\}, \{3,5\}\}$.

### 3.1. Coalitional NMPC problem

Given a partition $\mathcal{P}(k)$ at time instant $k$, the control objective of each coalition $c \in \mathcal{P}(k)$ is to solve:

$$Q_c^*(k) = \arg\min_{Q_c} J_c(\cdot,\cdot,\cdot), \tag{6}$$

subject to dynamics: (1)–(2), constraints: (4), (5), and the maximum coalition HTF flow:

$$q_c^{\max} = \frac{q_{\text{T}}^{\max}\,|c|}{N_{\text{loop}}}. \tag{7}$$

Here, $Q_c^*(k)$ is the optimal flow rate sequence over the control horizon $N_{\text{u}}$:

$$Q_c^*(k) \triangleq [q_c^*(k), q_c^*(k+1), \dots, q_c^*(k+N_{\text{u}}-1)],$$

and function $J_c(\cdot,\cdot,\cdot)$ is defined as a three-criterion cost that is evaluated along the prediction horizon $N_{\text{p}} > N_{\text{u}}$:

$$\begin{aligned} J_c(\cdot,\cdot,\cdot) = & \sum_{n=1}^{N_{\text{p}}} \left(-W_c(k+n) + \alpha_c \Psi_c\left(T^{\text{f}}_{N_{\text{seg}},j}(k+n)\right)\right) \\ & + \beta_c \sum_{n=0}^{N_{\text{p}}-1} \sum_{j\in c} \left(q_j(k+n) - q_j(k+n-1)\right)^2, \end{aligned} \tag{8}$$

where the control input is assumed to remain constant beyond $k + N_{\text{u}}$ to reduce the computation burden. The first argument of $J_c(\cdot,\cdot,\cdot)$ is the coalition's thermal power, which is described as

$$\begin{aligned} W_c(k+n) &= W_c^{\text{out}}(k+n) - W_c^{\text{in}}(k+n) \\ &= \sum_{j\in C} W_j^{\text{out}}(k+n) - \sum_{j\in C} W_j^{\text{in}}(k+n), \end{aligned}$$

with the output and input thermal powers of each loop $j \in c$ being, respectively,

$$W_j^{\text{out}}(k+n) = \rho^{\text{f}}_{N_{\text{seg}},j}\left(T^{\text{f}}_{N_{\text{seg}},j}(k+n)\right) C^{\text{f}}_{N_{\text{seg}},j}\left(T^{\text{f}}_{N_{\text{seg}},j}(k+n)\right) q_j(k+n) \; T^{\text{f}}_{N_{\text{seg}},j}(k+n),$$
$$W_j^{\text{in}}(k+n) = \rho^{\text{f}}_{1,j}\left(T^{\text{f}}_{1,j}(k+n)\right) C^{\text{f}}_{1,j}\left(T^{\text{f}}_{1,j}(k+n)\right) q_j(k+n)\, T^{\text{f}}_{1,j}(k+n)$$

The second and third terms of $J_c(\cdot)$ penalize, respectively, the violation of the outlet temperature and the control efforts with weights $\alpha_c, \beta_c \in \mathbb{R}_{>0}$. In particular, $\Psi_c\left(T^{\text{f}}_{N_{\text{seg}},j}(k+n)\right)$ is calculated as

$$\sum_{j\in C} \max\left(\frac{T^{\text{f}}_{N_{\text{seg}},j}(k+n) - T^{\text{f,max}}}{T^{\text{f,max}}}, \frac{T^{\text{f,min}} - T^{\text{f}}_{N_{\text{seg}},j}(k+n)}{T^{\text{f,max}}}, 0\right)^2,$$

to determine the maximum temperature violation of all loops within coalition $c$. Note that the outlet temperature of each coalition can be computed as

$$T_c^{\text{out}}(k) = \frac{\sum_{j\in c} q_j(k)\; T^{\text{f}}_{N_{\text{seg}},j}}{q_c(k)}.$$

### 3.2. Market-based mechanism

Local constraints divide the total flow rate, but some loops could achieve the optimum without activating their HTF constraints, while others could enhance their performance by increasing their flows over their values. For that reason, individual agents can take advantage of cooperating. That is, the key point of cooperation is to benefit from the HTF flow as a shared resource.

We consider a dynamic field partition $\mathcal{P}(k)$ based on coalitions of agents that supply and demand HTF in a market, as proposed by Masero et al. (2022). Therefore, the set of loops $\mathcal{L}$ can be classified into two disjoint subsets of supply loops $\mathcal{L}_s$ and demand loops $\mathcal{L}_d$ in terms of their utility gain to HTF flow changes, defined as

$$U_c(\cdot,\cdot,\cdot) = -J_c(\cdot,\cdot,\cdot). \tag{9}$$

The market-based mechanism employed to determine the field partition in one iteration is outlined below (Masero et al., 2022):

1. Start with a decentralized partition $\mathcal{P}_0 = \mathcal{L}$, *i.e.*, $c_j = j$.
2. Calculate the utility gain (9) of each coalition considering that its HTF flow increases and decreases a quantum: $q_c \pm \Delta q_c$.
3. Classify as demand (supply) agents $d \in \mathcal{L}_d$ ($s \in \mathcal{L}_s$) those with higher utility gains when their flow increases (decreases) and sort them in descending (ascending) order according to utility gain (loss).
4. Following an auction-based approach, the supply loop with the highest utility gain forms a coalition with the highest bidder to share its HTF surplus. After that, the supplier with the second-highest utility gain is clustered with the second-highest bidder, and so on. Therefore, coalitions are composed of top-winning pairs to least-losing pairs, as shown in Fig. 3.
5. Compute the utility gain (9) of the resulting coalitions and evaluate the surpluses of the demand and supply agents as $(U_c - U_d)$ and $(U_c - U_s)$, respectively. If any surplus is below a given threshold $\varphi \in \mathbb{R}_{0+}$, the formation of a coalition is not worth the computational burden.
6. The partition $\mathcal{P}_1$ is composed of the resulting coalitions:

$$\mathcal{P}_1 = \{c_1, c_2, \ldots, c_C\} = \{\{d_1, s_1\}, \{d_2, s_2\}, \ldots, \{d_C, s_C\}\}.$$

The above procedure can be repeated $M \in \mathbb{N}_+$ iterations to find a better partition $\mathcal{P}(k) = \mathcal{P}_M$. To this end, the resulting coalitions can also be classified into supply and demand entities. Additionally, a maximum coalition size of $|c|^{\max} \in \mathbb{N}_+$ is set to limit the computational burden, which increases with the coalition size.

**Remark 1.** To calculate the utility gain $U_c$ in coalitions of size $|c| \geq 2$, it is necessary to know how the HTF flow $\Delta q_c$ is distributed between its loops. To this end, let $\lambda_c = [\lambda_j]_{j\in c} \in \mathbb{R}^{|c|}$ with $\lambda_j \in (0,1)$ denote this flow distribution and obtain the utility gain $U_c^{\text{d}}$ of a demand coalition:

$$\{U_c^{\text{d}}, \lambda_c^{\text{d}}\} = \arg\min_{\lambda^{\text{d}}} J_c\left(\cdot,\cdot,\sum_{j\in c}(q_j(k) + \lambda_j^{\text{d}}\Delta q_c)\right), \tag{10}$$

and that of a supply coalition:

$$\{U_c^{\text{s}}, \lambda_c^{\text{s}}\} = \arg\min_{\lambda^{\text{s}}} J_c\left(\cdot,\cdot,\sum_{j\in c}(q_j(k) - \lambda_j^{\text{s}}\Delta q_c)\right), \tag{11}$$

subject to the same equations as (6), and $\sum_{j\in c}\lambda_j = 1$.

## 4. Learning-based control algorithm

To enjoy the online computing time relief of machine learning techniques, the control policy of coalitional model predictive controllers is approximated by using artificial neural networks to achieve real-time optimizations. In particular, we propose a hierarchical learning-based coalitional control algorithm by clustering loops in such a way that the resulting thermal energy is maximized.

**Description 1.** *Fig. 3 illustrates a learning-based control scheme for coalitional problems with a maximum coalition size of* $|c|^{\max} = 3$. *The scheme is divided mainly into two blocks:*

1. *The first block represents the supervisory layer, where the partition* $\mathcal{P}_M$ *is obtained in* $M = 2$ *iterations. Starting with the decentralized partition* $\mathcal{P}_0$*, pairs of supply and demand agents are merged in the order established by the market-based mechanism to obtain* $\mathcal{P}_1$*. Afterwards, the utility gains* $(U_c^{\text{d}}, U_c^{s})$ *of the coalitions are obtained by applying two neural networks instead of solving the coalitional NMPC problems (10) and (11). In the second iteration, we obtain the final partition* $\mathcal{P} = \mathcal{P}_2$ *by applying the market-based approach again.*
2. *The second block illustrates the lower control layer, where each coalitional controller* $c \in \mathcal{P}$ *solves its problem by a neural network according to its coalition size.*

In the following sections, we provide details on the design of these artificial neural networks and summarize the hierarchical coalitional control algorithm.

### 4.1. Neural networks description

An artificial neural network is a general function approximator that emulates the functioning of the human brain. In this work, we propose the use of multilayer perceptrons, which are ANNs formed by three types of layers: input, output, and hidden layers. These ANNs are types of feedforward neural networks, which means that all neurons in a layer are connected to all neurons in the previous layer without cycles between them (Fine, 2006). Each layer is formed by one or more neurons that implement a linear regression problem. Neurons contain transfer functions on their outputs to represent their activation state and impose constraints (Ramachandran et al., 2017). The linear regression problem and the activation function $a_n^{(l)}$ of neuron $n$ in layer $l$ is given by

$$a_n^{(l)} = g^{(l)}\left(z_n^{(l)}\right), \quad \text{with} \quad z_n^{(l)} = \sum_{m=1}^{L^{(l-1)}}\left(w_{nm}^{(l-1)} a_m^{(l-1)} + b_n^{(l-1)}\right),$$

where each layer is formed by $L^{(l)}$ layers; $w_{nm}^{(l-1)}$ is the kernel between neurons $n$ and $m$ of layer $(l-1)$; and $b_n^{(l-1)}$ is the bias of neuron $n$ in layer $l-1$.

Neural networks can be used to solve regression and classification problems. In particular, this work uses multilayer perceptrons, *i.e.*, five regression ANNs and one classification ANN. Neural networks are trained with a back-propagation process (Rumelhart et al., 1986) in

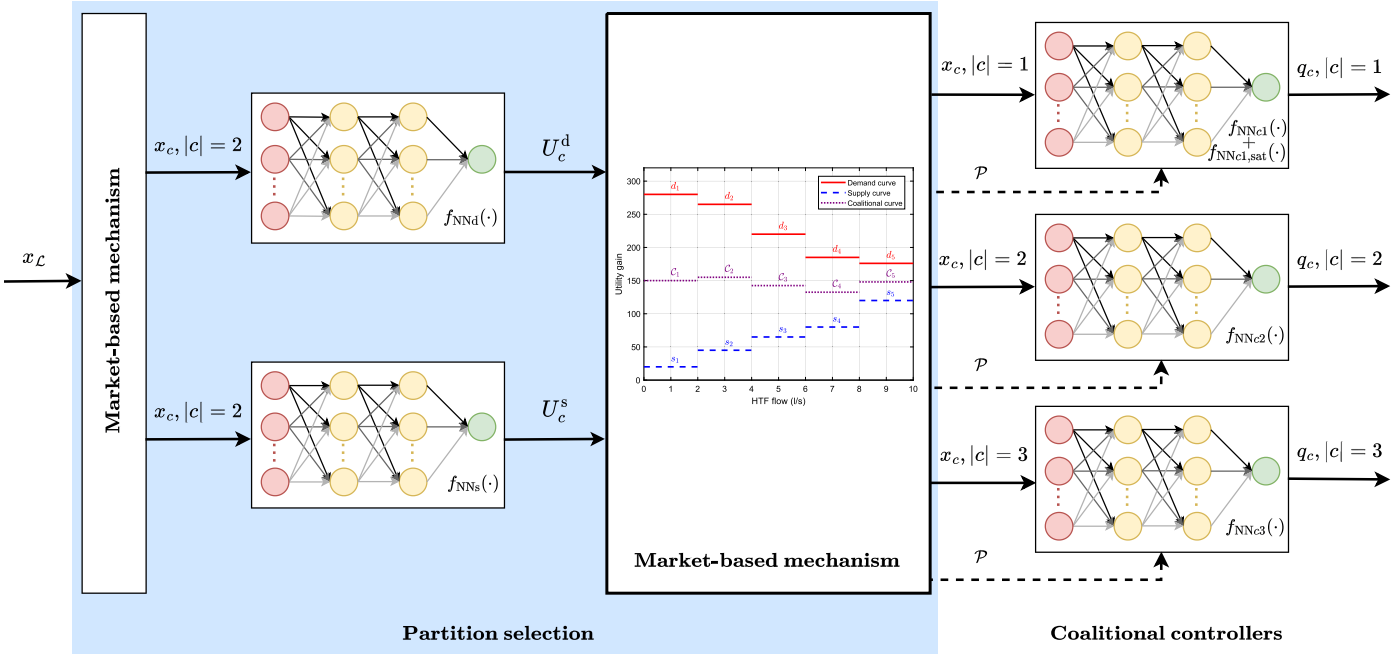



**Fig. 3.** Scheme of the proposed learning-based control approach for a maximum coalition size of $|c|^{\max} = 3$.

which the error of each neuron is obtained by starting from the output error and using gradients, *i.e.,* propagating the error from one layer to the previous one. In this work, regression ANNs use the Levenberg–Marquardt back-propagation algorithm (Marquardt, 1963), which takes the sum of squared errors as a loss function and performs a fast training by approximating the Hessian matrix. The classification ANN uses the scaled conjugate gradient algorithm (Møller, 1993), which performs a search along conjugate directions to obtain the gradients, and the loss function is the cross entropy.

Before training the ANNs, the data is usually scaled in the range $[-1, 1]$ as

$$x' = \frac{2\,(x - x_{\min})}{x_{\max} - x_{\min}} - 1,$$

where $x$ is the original data vector, $x'$ is the vector of normalized data, and $x_{\max}$ and $x_{\min}$ are, respectively, the maximum and minimum values of the original vector. Moreover, the dataset is divided into three subsets: training (for adjusting the parameters), validation (for evaluating the behavior while tuning some hyperparameters), and test (for estimating the behavior with new data) subsets. The mean squared error of the ANN in the validation set is monitored during the training process and, at a certain point in the training, it starts to increase. To avoid overfitting, when the error has increased for a certain number of iterations (validation checks), the training is stopped, and the weights obtained when the error was minimal are taken. Once trained, ANNs are evaluated offline in the test set to compare their behavior with different architectures. This training process is carried out by trial and error, changing the architecture of the neural network until the desired performance is obtained in the three subsets.

The proposed algorithm employs several neural networks for controlling the solar plant in real time, as depicted in Fig. 3. The ones that compute the utility gain of supply and demand coalitions are:

$$\begin{aligned} U_c^{\text{d}} &= f_{\text{NNd}}\big(x_c(k)\big), \\ U_c^{\text{s}} &= f_{\text{NNs}}\big(x_c(k)\big), \end{aligned} \tag{12}$$

where the input $x_c(k)$ gathers the flow rate $q_c(k-1)$, the overall temperatures $T^{\text{a}}(k), T^{\text{in}}(k)$, and $T^{\text{out}}(k)$, the solar DNI every six segments and two-time instants, *i.e.,* $I_{\{1:6:N_{\text{seg}}\},j}(k+n)$, with $n = 1, 3, 5, \ldots, N_{\text{p}}$, and the flow and metal temperatures of each loop $j \in c$ in the middle of collectors $T^{\text{f}}_{S_{col}/2,j}(k)$ and $T^{\text{m}}_{S_{col}/2,j}(k)$ for all $col \in \{1, \ldots, N_{\text{col}}\}$, being $S_{col}$ the number of segments in each collector.

The rest of the neural networks compute the coalition flow rate $q_c(k)$ according to its coalition size:

$$\begin{aligned} q_c(k) &= f_{\text{NNc1}}\big(x_c(k)\big), \\ q_c(k) &= f_{\text{NNc2}}\big(x_c(k)\big), \\ q_c(k) &= f_{\text{NNc3}}\big(x_c(k)\big), \end{aligned} \tag{13}$$

for all $c \in \mathcal{P}(k)$ with $|c| = 1$, $|c| = 2$, and $|c| = 3$, respectively.

When a loop does not form a coalition, its flow rate is usually saturated. Although neural networks are good function approximators, they tend to avoid saturations. For that reason, an extra classification neural network is added to the output of $f_{\text{NNc1}}(\cdot)$. This saturation neural network classifies the signal in saturated / no-saturated:

$$\delta(k) = f_{\text{NNc1,sat}}\big(x_c(k)\big), \tag{14}$$

where $\delta(k) \in [0, 1]$ indicates if the flow rate must be saturated. In particular, $\delta(k) > 0.8$ saturates the output of $f_{\text{NNc1}}(\cdot)$ to its nearest limit.

### 4.2. Coalitional ANNs algorithm

This section summaries the coalitional control algorithm based on neural networks for a fast implementation of coalitional NMPC. The supervisory layer decides the field partition $\mathcal{P}(k)$ every $T_{\text{up}} \in \mathbb{N}_+$ time instants depending on the state and disturbances of the solar collector field:

**Algorithm 1 (Supervisory layer).**

**Initial data:** $T_{\text{up}}$, $M$, $|c|^{\max}$, $\varphi$.
**if** mod $(k, T_{\text{up}}) = 0$ **then**:

1: Estimate or measure flow rate $q_j(k-1)$ for all $j \in \mathcal{L}$ (*e.g.*, with $q_{\text{T}}(k-1)$ and the position of the valve through a hydraulic model), temperatures $T^{\text{a}}(k)$, $T^{\text{f}}_{i,j}(k)$ $\forall i \in S$, and solar DNI in the prediction horizon (*e.g.*, following (Cao and Lin, 2008; Aguilar-López et al., 2022)).
2: Set as the initial partition the one formed by decentralized agents $\mathcal{P}_0 = \bigcup_{j \in \mathcal{L}} c_j$.
3: **for** $m = 1$ to $M$ iteration **do**:

- Evaluate the utility gain $U_c$ of each coalition and classify it as a supply or demand entity.
- Sort demand and supply coalitions in descending order of their utility gains.
- Merge one supply and one demand coalition in the order established by the market mechanism if the resulting surpluses exceed the threshold $\varphi$ and the resulting size is below $|c|^{\max}$.
- The resulting coalitions yield partition $\mathcal{P}_m$.

4: **end for**
5: Finally, send $\mathcal{P}(k) = \mathcal{P}_M$ to the lower control layer.
**end if**

In the lower control layer, coalitions $c \in \mathcal{P}(k)$ compute and implement their control inputs:

**Algorithm 2 (Lower control layer).**

At each time instant $k$:
1: Coalition $c \in \mathcal{P}(k)$ receives its forecasted solar DNI in the prediction horizon $N_p$, and measures/estimates $T^f_{i,j}(k), \forall i \in S$ of any loop $j \in C$.
2: Compute the coalition flow rate $q_c(k)$ by (13).
3: Saturate $q_j(k)$ of all $j \in c$ if necessary to satisfy the local and coalitional HTF flow constraints (4) and (7).
4: Apply to the coalition its final HTF flow $q_c(k)$.

## 5. Simulation results

The parabolic-trough solar collector field considered in this study is based on the ACUREX plant, which was operating in the Plataforma Solar de Almería (Gálvez-Carrillo et al., 2009; Gallego et al., 2022). The field consists of $N_{\text{loop}} = 100$ parallel loops of length $d_{\text{loop}} = 147$ m, which are discretized into $N_{\text{seg}} = 174$ segments of length $\Delta l = 1$ m. Each loop $j \in \{1, \dots, N_{\text{loop}}\}$ is composed of twelve single-axis collectors aligned East–West, the active part is 144 m while the passive part is 30 m. In the ACUREX plant, the HTF was Therminol 55 thermal oil, whose density and specific heat capacity were defined, respectively, as:

$$\begin{aligned} \rho^f_{i,j}\left(T^f_{i,j}(k)\right) &= 903 - 0.672\,T^f_{i,j}(k), \\ C^f_{i,j}\left(T^f_{i,j}(k)\right) &= 1820 + 3.478\,T^f_{i,j}(k). \end{aligned}$$

The coefficients of metal-fluid transmission and thermal losses are calculated by:

$$\begin{aligned} H^t_{i,j}\left(q_i(k), T^f_{i,j}(k)\right) &= q_i^{0.8}(k)\,\Phi\left(T^f_{i,j}(k)\right), \\ H^l_{i,j}\left(T^f_{i,j}(k), T^a(k)\right) &= 0.00249\left(T^f_{i,j}(k) - T^a(k)\right) - 0.06133, \end{aligned}$$

with $\Phi\left(T^f_{i,j}(k)\right)$ obtained according to (Camacho et al., 2012) as:

$$\begin{aligned} \Phi\left(T^f_{i,j}(k)\right) = \; & 2.17e6 - 5.01e4\,T^f_{i,j}(k) + 4.53e2\,T^f_{i,j}(k)^2 \\ & - 1.64\,T^f_{i,j}(k)^3 + (2.1e-3)\,T^f_{i,j}(k)^4. \end{aligned}$$

The ambient temperature $T^a$ is considered constant for simplicity, and the values of the main variables and constraints are summarized in Table 2. Moreover, we consider the output temperature of the steam generator to drop 90 °C with respect to the inlet temperature and a time constant of 600 s:

$$\frac{T^{\text{in}}(s)}{\hat{T}^{\text{out}}(s)} = \frac{1}{600s + 1}, \tag{15}$$

where $\hat{T}^{\text{out}}(s) = T^{\text{out}}(s) - 90$. Applying the z-transform to (15) with a discretization step $\Delta t = 0.5$ s, the following discrete-time inlet temperature is obtained:

$$T^{\text{in}}(k) = 0.999167\,T^{\text{in}}(k-1) + 8.33e-4\left(T^{\text{out}}(k-1) - 90\right).$$

**Table 2**
Model parameters and constraints of the 100-loop collector field.

| Symbol | Value | Units |
|---|---|---|
| $\rho^m$ | 7800 | kg/m$^3$ |
| $C^m$ | 550 | J/(kg °C) |
| $A^f$ | $7.55e-4$ | m$^2$ |
| $D^m$ | 0.031 | m |
| $D^f$ | 0.0254 | m |
| $q_j^{\min}$ | 0.2 | l/s |
| $q_j^{\max}$ | 1 | l/s |
| $q_T^{\max}$ | 65 | l/s |
| $T^a$ | 25 | °C |
| $T^{f,\min}$ | 220 | °C |
| $T^{f,\max}$ | 300 | °C |

### 5.1. Simulation parameters and defocus algorithm

We perform simulations of length $N_{\text{sim}} = 120$ min in MATLAB® R2020a with a PC Intel® Core™ i7 – 8700 CPU at 3.20 GHz and 16 GB RAM. The `Optimization Toolbox` and `Deep Learning Toolbox` are, respectively, used to implement the control strategy under two different techniques:

- **Coal. NMPC** denotes the non-linear coalitional MPC strategy proposed in (Masero et al., 2022), which solves the optimization problem (6) with selected horizons $N_p = 12$ and $N_u = 10$.
- **Coal. ANN** represents the proposed coalitional algorithm based on artificial neural networks.

Both approaches select the partition every $T_{\text{up}} = 5$ time instants after two iterations $M = 2$, and use $|c|^{\max} = 3$ and $\Delta q_c = 0.5$ l/s due to their good trade-off between performance and computational burden in simulation. To compare these approaches, we use as key performance indicators: (i) the average thermal energy produced $\bar{E}$ [KWh] during the simulation time, (ii) the maximum computing time of the overall controller $\tau^{\max}_{\text{ctr}} = \max\left(\tau_{\text{ctr}}(k)\right)$ along the simulation, and (iii) the maximum computing time to solve a coalition problem, also considering the time spent to determine the partition:

$$\tau^{\max}_c = \max\left(\tau^{\max}_c(k) + \frac{\tau_{\mathcal{P}}(k)}{|\mathcal{P}(k)|}\right), \; \forall k \in N_{\text{sim}}, \tag{16}$$

where $\tau^{\max}_c(k)$ is the maximum time to solve a coalition problem at $k$, $\tau_{\mathcal{P}}(k)$ refers to the time required to select the field partition at $k$, and $|\mathcal{P}(k)|$ denotes the number of coalitions in partition $\mathcal{P}(k)$.

In simulations, we initially consider $T^{\text{in}}, T^f_{1,j}, T^{1f}_{1,j} = 155$ °C and $T^m_{1,j} = 165$ °C, $\forall j \in \mathcal{L}$, and also some simplifications such as the precomputation of thermal losses $H^l$ for a given set of flow rates and temperatures, and the selection of the controller sampling time as $\Delta t_{\text{ctr}} = \gamma \Delta t$, *i.e.*, a multiple of the model discretization time $\Delta t$ being $\gamma \in \mathbb{N}_+$. Tuning parameters $\alpha_j = 0.008$ and $\beta_j = 50$ for all $j \in c$ are designed in such a way that, respectively, penalize the violations of the outlet temperature and the valve control effort without strongly affecting the performance of the controllers. Additionally, we use a simple defocus mechanism for the last nine collectors of each loop to prevent HTF degradation from exceeding the maximum temperature $T^{f,\max}$. Hence, the efficiency of these collectors $\eta^{\text{col}}$ is based on the defocusing angle $\theta$ (Goswami et al., 2000), which depends on the outlet temperature of the corresponding loop. Let $\eta^{\text{col}}_0$ be the collector efficiencies when defocused, the algorithm implemented can be summarized as follows:

$\eta^{\text{col}}_0 = \eta^{\text{col}}$
**if** $T^f_{N_{\text{seg}},j} > 290$°C **then**
  $\theta = 1.6°$, $\eta^{\text{col}} \leftarrow 0.75\,\eta^{\text{col}}_0$
**end if**
**if** $T^f_{N_{\text{seg}},j} > 292$°C **then**
  $\theta = 2.25°$, $\eta^{\text{col}} \leftarrow 0.5\,\eta^{\text{col}}_0$
**end if**
**if** $T^f_{N_{\text{seg}},j} > 294$°C **then**

**Table 3**
Pearson correlation coefficients of the training process.

(a) $f_{\text{NNd}}(\cdot)$

| Neurons | $R$ (tr.) | $R$ (val.) | $R$ (test) | $R$ (test2) |
|---|---|---|---|---|
| 15 | 99.472 | 99.416 | 99.105 | 90.782 |
| 20 | 98.894 | 99.108 | 99.008 | 83.163 |
| 30 | 99.110 | 99.146 | 98.170 | 75.053 |
| 50 | 99.181 | 99.553 | 98.847 | 76.420 |
| 15-5 | 99.339 | 99.181 | 99.258 | 92.313 |
| 15-10 | 87.207 | 84.822 | 87.273 | 46.021 |
| **20-4** | **99.291** | **99.509** | **99.419** | **93.234** |
| 30-15 | 99.254 | 99.441 | 99.133 | 87.511 |
| 50-25 | 99.385 | 99.201 | 99.071 | 85.981 |
| 15-10-5 | 98.207 | 99.110 | 98.026 | 75.416 |
| 15-10-10 | 99.392 | 99.387 | 99.149 | 85.705 |
| 30-15-10 | 97.973 | 97.919 | 97.666 | 68.491 |

(b) $f_{\text{NNs}}(\cdot)$

| Neurons | $R$ (tr.) | $R$ (val.) | $R$ (test) | $R$ (test2) |
|---|---|---|---|---|
| 15 | 99.636 | 99.735 | 99.532 | 90.035 |
| 20 | 99.654 | 99.856 | 99.682 | 90.541 |
| 30 | 91.499 | 91.883 | 91.197 | 73.925 |
| 50 | 99.533 | 99.436 | 99.841 | 77.243 |
| 15-5 | 99.863 | 99.914 | 99.915 | 98.191 |
| 15-10 | 99.729 | 99.920 | 99.943 | 95.688 |
| 20-4 | 84.283 | 83.435 | 83.098 | 60.846 |
| **30-15** | **99.878** | **99.827** | **99.949** | **97.386** |
| 50-25 | 99.821 | 99.606 | 99.917 | 90.622 |
| 15-10-5 | 99.836 | 99.863 | 99.847 | 92.830 |
| 15-10-10 | 99.809 | 99.956 | 99.934 | 95.933 |
| 30-15-10 | 99.900 | 99.768 | 99.949 | 95.895 |

(c) $f_{\text{NNc1}}(\cdot)$

| Neurons | $R$ (tr.) | $R$ (val.) | $R$ (test) | $R$ (test2) |
|---|---|---|---|---|
| 15 | 99.960 | 99.971 | 99.973 | 99.256 |
| 20 | 99.930 | 99.975 | 99.980 | 99.670 |
| 30 | 99.964 | 99.979 | 99.952 | 99.399 |
| 50 | 99.981 | 99.984 | 99.975 | 99.204 |
| 15-5 | 99.743 | 99.745 | 99.735 | 99.721 |
| 15-10 | 99.947 | 99.973 | 99.907 | 99.786 |
| 20-4 | 99.950 | 99.979 | 99.980 | 99.798 |
| 30-15 | 99.952 | 99.969 | 99.942 | 99.834 |
| 50-25 | 99.954 | 99.965 | 99.979 | 99.766 |
| 15-10-5 | 99.987 | 99.991 | 99.985 | 99.547 |
| **15-10-10** | **99.977** | **99.968** | **99.964** | **99.864** |
| 30-15-10 | 99.961 | 99.979 | 99.983 | 99.781 |

(d) $f_{\text{NNc1,sat}}(\cdot)$

| Neurons | $R$ (tr.) | $R$ (val.) | $R$ (test) | $R$ (test2) | MSEsat |
|---|---|---|---|---|---|
| 15 | 90.166 | 90.264 | 90.53 | 99.851 | $1.261e-10$ |
| 20 | 87.408 | 87.185 | 87.947 | 99.851 | $1.308e-10$ |
| 30 | 90.633 | 90.627 | 90.628 | 99.851 | $1.185e-10$ |
| 50 | 99.795 | 88.236 | 87.352 | 99.850 | $1.286e-10$ |
| **15-5** | **91.011** | **90.670** | **90.873** | **99.856** | $1.252e-10$ |
| 15-10 | 88.184 | 88.661 | 88.805 | 99.852 | $1.285e-10$ |
| 20-4 | 89.281 | 89.331 | 89.168 | 99.851 | $1.261e-10$ |
| 30-15 | 88.672 | 88.337 | 87.880 | 99.851 | $1.282e-10$ |
| 50-25 | 88.180 | 88.235 | 88.167 | 99.851 | $1.300e-10$ |
| 15-10-5 | 87.851 | 87.938 | 87.864 | 99.852 | $1.290e-10$ |
| 15-10-10 | 91.440 | 90.535 | 91.395 | 99.851 | $1.173e-10$ |
| 30-15-10 | 86.821 | 87.357 | 86.78 | 99.852 | $1.302e-10$ |
| 50-30-15 | 89.861 | 89.072 | 89.687 | 99.850 | $1.243e-10$ |

(*continued on next column*)

$\theta = 2.75°$, $\eta^{\text{col}} \leftarrow 0.25\ \eta_0^{\text{col}}$
**end if**
**if** $T^{\text{f}}_{N_{\text{seg}},j} > 295°$ **then**
$\theta = 5°$, $\eta^{\text{col}} \leftarrow 0\ \eta_0^{\text{col}}$
**end if**

Finally, the control model uses $\Delta l = 6\,\text{m}$ and $\Delta t = 3\,\text{s}$ to speed up the computing time, and the controllers use a sampling time $\Delta t_{\text{ctr}} = \gamma \Delta t = 60\,\text{s}$, with $\gamma = 20$.

**Table 3** (*continued*).

(e) $f_{\text{NNc2}}(\cdot)$

| Neurons | $R$ (tr.) | $R$ (val.) | $R$ (test) | $R$ (test2) |
|---|---|---|---|---|
| 15 | 99.729 | 99.090 | 99.779 | 96.723 |
| 20 | 99.640 | 99.646 | 99.679 | 97.920 |
| 30 | 99.599 | 99.627 | 99.721 | 97.834 |
| 15-5 | 99.673 | 99.822 | 99.745 | 98.217 |
| 15-10 | 99.418 | 99.636 | 99.282 | 98.021 |
| 20-4 | 99.597 | 99.817 | 99.567 | 98.310 |
| **30-15** | **99.672** | **99.759** | **99.601** | **98.741** |
| 50-25 | 99.954 | 99.965 | 99.979 | 99.766 |
| 15-10-5 | 99.987 | 99.991 | 99.985 | 99.547 |
| 15-10-10 | 99.977 | 99.968 | 99.964 | 99.864 |
| 30-15-10 | 99.961 | 99.979 | 99.983 | 99.781 |

(f) $f_{\text{NNc3}}(\cdot)$

| Neurons | $R$ (tr.) | $R$ (val.) | $R$ (test) | $R$ (test2) |
|---|---|---|---|---|
| 15 | 99.858 | 99.866 | 99.886 | 96.236 |
| 20 | 99.876 | 99.851 | 99.826 | 94.869 |
| 30 | 99.850 | 99.793 | 99.883 | 91.595 |
| 50 | 99.795 | 99.933 | 99.786 | 94.453 |
| 15-5 | 99.853 | 99.869 | 99.770 | 96.733 |
| 15-10 | 99.879 | 99.891 | 99.873 | 94.828 |
| **20-4** | **99.879** | **99.891** | **99.873** | **97.119** |
| 30-15 | 99.788 | 99.707 | 99.876 | 93.940 |
| 50-25 | 99.806 | 99.935 | 99.804 | 95.355 |
| 15-10-5 | 99.858 | 99.785 | 99.875 | 94.786 |
| 15-10-10 | 99.802 | 99.909 | 99.909 | 95.233 |
| 30-15-10 | 99.822 | 99.903 | 99.907 | 96.043 |

### 5.2. Neural network: Training and design

The neural networks selected for controlling are multilayer perceptrons trained with Levenberg–Marquardt back-propagation. Different numbers and sizes of hidden layers are tested, stopping the training when the neural network achieved a mean squared error under $10^{-9}$ in its output or six validation checks. The activation functions are hyperbolic tangent sigmoid for all layers except for a linear function in the output, and the maximum number of epochs is $10^3$. The damping factor $\mu$ of the Levenberg–Marquardt algorithm has an increasing ratio of $10^{-1}$, a decreasing ratio of 10, and a maximum value of $10^{10}$. The saturation neural network is trained with scaled-conjugate back-propagation and the stopping criteria are to achieve mean squared errors under $10^{-15}$ and six validation checks. The activation functions are hyperbolic tangent sigmoid for all layers, except for a softmax in the last one (as it is used for classification). The maximum number of epochs is $10^3$, and the parameters of the back-propagation algorithm are $\sigma = 5e-5$ and $\lambda = 5e-7$.

Fig. 4 displays a scheme of the offline training. We performed eight simulations (a total of sixteen simulated hours) to obtain training datasets using different solar DNI profiles. For each simulation, the optimization problems are solved using the SQP technique (*e.g.*, via `fmincon` in MATLAB). Other techniques such as (Riedmiller and Braun, 1993; Ghalambaz et al., 2021) could be employed to solve the optimization problems. Note that all profiles are characterized by moving clouds, and loops have reflectivity differences because some of them are considered dusty or in maintenance. The following datasets are obtained and divided into a training set of 70%, a validation set of 15% and a test set of 15% for the training process of the ANNs:

- $f_{\text{NNd}}(\cdot)$ dataset: 3077 instances.
- $f_{\text{NNs}}(\cdot)$ dataset: 3077 instances.
- $f_{\text{NNc1}}(\cdot)$ dataset: 56065 instances.
- $f_{\text{NNc1,sat}}(\cdot)$ dataset: 56065 instances.
- $f_{\text{NNc2}}(\cdot)$ dataset: 14030 instances.
- $f_{\text{NNc3}}(\cdot)$ dataset: 3965 instances.

For each neural network, different architectures are implemented in open loop and compared using the Pearson correlation coefficient ($R$).

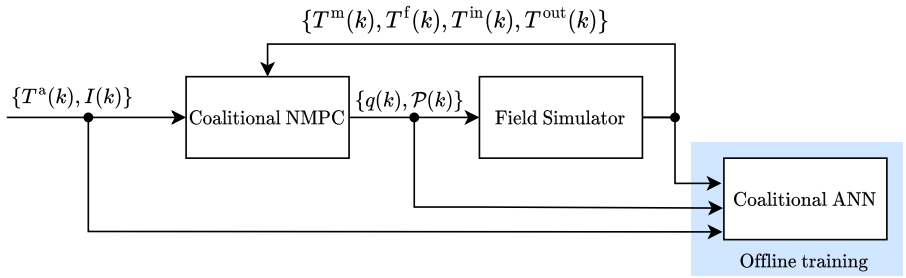



**Fig. 4.** Offline training of coalitional ANN based on coalitional NMPC.

**Table 4**
Numerical results comparison between the proposed coalitional ANN (C. ANN) and the original coalitional NMPC (C. NMPC).

| | DNI profile #1 | | | DNI profile #2 | | | DNI profile #3 | | | DNI profile #4 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | C. NMPC | C. ANN | Reduc. ANN | C. NMPC | C. ANN | Reduc. ANN | C. NMPC | C. ANN | Reduc. ANN | C. NMPC | C. ANN | Reduc. ANN |
| $\bar{E}$ [KWh] | 20 250 | 20 142 | 0,53% | 18 399 | 18425.2 | −0.142% | 15 286 | 15273.4 | 0,082% | 20 052 | 20 022 | 0,150% |
| $\tau_{ctr}^{max}$ [s] | 580.2 | 2.81 | 99.51% | 544.5 | 2.76 | 99.49% | 516.0 | 2.53 | 99.51% | 584.5 | 2.68 | 99.54% |
| $\tau_{c}^{max}$ [s] | 29.9 | 0.101 | 99.663% | 28.5 | 0.173 | 99.34% | 27.0 | 0.071 | 99.74% | 24.6 | 0.078 | 99.68% |
| $\|c\| = 1$ [%] | 34.92 | 12.92 | – | 50.38 | 36.75 | – | 89.79 | 72.21 | – | 62.63 | 58.54 | – |
| $\|c\| = 2$ [%] | 59.58 | 75.33 | – | 40.75 | 53.75 | – | 7.58 | 21.92 | – | 33.75 | 34.58 | – |
| $\|c\| = 3$ [%] | 5.50 | 11.75 | – | 8.88 | 9.50 | – | 2.63 | 5.88 | – | 3.63 | 6.88 | – |

Table 3 shows all the results obtained for the three subsets (tr.:training, val.: validation, and test). The column $R$ (test2) corresponds to an extra evaluation of the obtained flow rate by performing open-loop simulations using three new solar DNI profiles. Note that the column *Neurons* indicates the number of neurons in each layer separated by a dash. In Tables 3a–3f, although the $R$ values do not differ much between subsets, we observe a slight decay in some ANNs due to the use of different solar DNI profiles. The selected architectures, which are marked in bold, are those that maintain a high value of $R$ in the 'test2' subset.

Table 3d shows the $R$ coefficients of the saturation ANN in every subset. Afterwards, the evaluation with new data is performed with the combination of $f_{\text{NN}c1}(\cdot)$ and $f_{\text{NN}c1,\text{sat}}(\cdot)$ in open loop, and the value of $R$ (test2) is obtained from the resulting flow rate with saturation. We also take account of saturated instances from the original controller, and compute the mean squared error (MSEsat) between the real output and the corresponding flow rate from the combination of both neural networks. Note that the first three $R$ columns of Table 3d (those of the training, validation, and test subsets) refer to the binary saturation value, while $R$ (test2) and MSEsat refers to the flow rate obtained from the combination of the two neural networks. The $R$ values are around 90% for the saturations, but increase to more than 99% when combining $f_{\text{NN}c1}(\cdot)$ and $f_{\text{NN}c1,\text{sat}}(\cdot)$, with an MSEsat around $10^{-10}$.

### 5.3. Results and discussion

For simulations, we use several two-hour solar DNI profiles – different from the ones used to train the neural networks – to evaluate the proposed learning-based strategy.

Table 4 displays a numerical comparison of the results obtained by implementing the coalitional NMPC and coalitional ANN approaches with four solar DNI profiles. Specifically, we show the average thermal energy ($\bar{E}$) generated by the field in two-hour simulations, the maximum time to solve the coalitional control (16), and also the average number of loops in coalitions of size $|c| = 1$, $|c| = 2$, and $|c| = 3$ throughout the simulations. As observed, the proposed coalitional ANN strategy attains an average thermal energy reduction of up to 0.746% with respect to the coalitional NMPC. However, by implementing the coalitional ANN, a major reduction in the computing time is achieved in all cases (see Fig. 7). For example, with profile #1, the Coal. NMPC approach obtains a maximum time per coalition of 29.9 s, while the Coal. ANN approach achieves a maximum of 0.101 s. Moreover, whereas the maximum computing time of the Coal. NMPC controller is 580.2 s, the Coal. ANN controller yields 2.81 s. Since the sampling time of the controller is 60 s, and most commercial solar plants are equipped with an overall controller, our proposed coalitional ANN approach is suitable for real-time implementation because $\tau_{ctr}^{max} < 60$ s, unlike the coalitional NMPC. Finally, note that the proposed learning-based strategy is implemented in closed loop but trained in open loop, dragging the possible errors that can arise with respect to the coalitional NMPC.

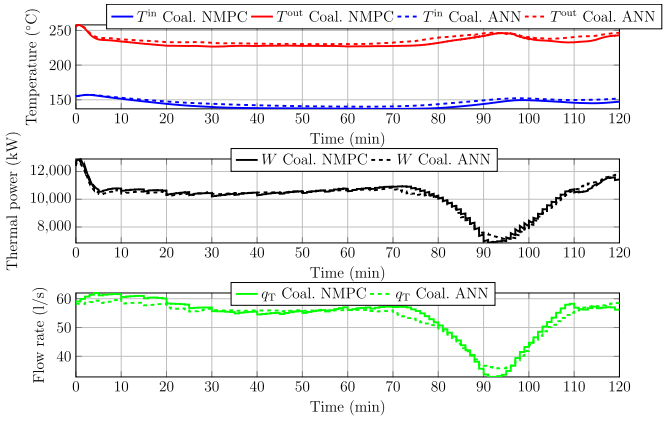



**Fig. 5.** Evolution of temperatures, thermal power, and total flow rate of the 100-loop field with solar DNI profile #1.

As illustrative results, we show the continuous-time evolution of the outlet and input temperatures ($T^{in}$ and $T^{out}$ [°C]), the thermal power generated ($W$ [kW]) and the total flow rate ($q_T$ [l/s]) by implementing the coalitional NMPC and coalitional ANN with solar DNI profiles #1 and #3. This comparison is displayed in Figs. 5 and 6, which show a similar evolution between both control strategies but with slight changes due to the selection of unequal partitions $\mathcal{P}(k)$ throughout the simulation.

Lastly, we also discuss the shortcomings of the proposed method. In the first place, constraint violations are not taken directly into account when using neural networks. For that reason, we saturate the ANNs output within their limits, but other solutions can be contemplated, *e.g.*, to increase the penalty of the MPC constraints within a more compact temperature range. Second, neural networks may fail to provide suitable outputs when the operating conditions are very far from those used in the training process.

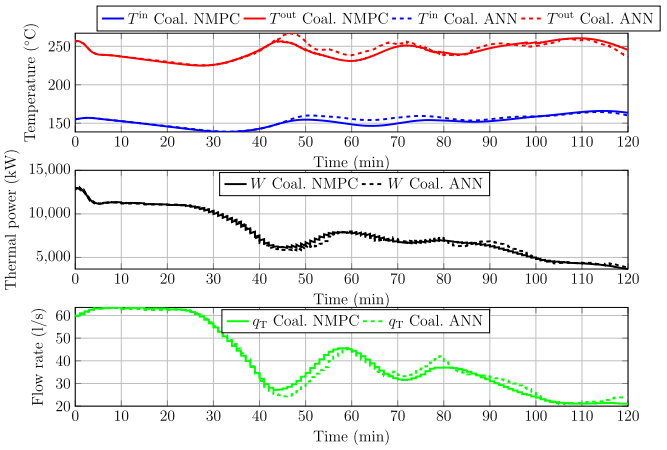



**Fig. 6.** Evolution of temperatures, thermal power, and total flow rate of the 100-loop field with solar DNI profile #3.

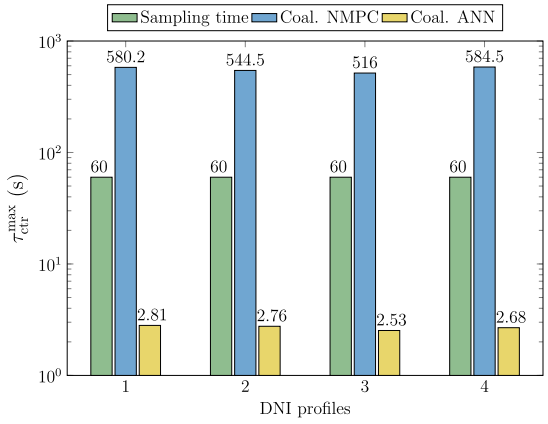



**Fig. 7.** Comparison of the maximum computing times obtained with Coalitional NMPC and Coalitional ANN, for a controller's sampling time of 60 s.

## 6. Conclusions

We present a hierarchical coalitional non-linear model predictive control (NMPC) based on artificial neural networks (ANNs) to maximize the thermal energy provided by solar collector fields. Depending on operation conditions, the supervisory layer partitions the field online into clusters of loops that trade the heat transfer fluid. Cooperating agents coordinate their control actions to reduce their local costs whilst contributing to the achievement of global objective. In addition to this adaptive capacity of the control system to tackle disturbances, another objective is that the controller is able to apply the control actions in real time. As shown in the simulation results, the current implementation allows real-time executions of the proposed coalitional ANN strategy. Therefore, our contribution upgrades the scalability of coalitional controllers in large-scale solar power plants.

Future investigations are the inclusion of a detailed hydraulic model, the design of neural networks that include constraint satisfaction, the analysis of the controller robustness with respect to changes in the model, and the application of proposed strategy to other potential distributed non-linear systems such as microgrids.

## CRediT authorship contribution statement

**Eva Masero:** Conceptualization, Methodology, Investigation, Formal analysis, Software, Writing – original draft, Writing – review & editing, Validation, Data curation, Visualization. **Sara Ruiz-Moreno:** Conceptualization, Methodology, Investigation, Software, Writing – original draft, Writing – review & editing, Validation. **José Ramón D. Frejo:** Conceptualization, Methodology, Validation, Writing – review & editing, Supervision, Project administration. **José M. Maestre:** Methodology, Writing – review & editing, Supervision, Project administration, Funding acquisition. **Eduardo F. Camacho:** Conceptualization, Methodology, Validation, Supervision, Project administration, Funding acquisition.

## 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.

## Data availability

Data will be made available on request.

## Acknowledgments

Financial support for this research was provided by the European Research Council (ERC-AdG) under the H2020 programme for the project OCONTSOLAR (Grant no. 789051), by the MCIN/AEI, Spain /10.13039/501100011033 for the project C3PO-R2D2 (Grant no. PID2020-119476RB-I00), and by the Spanish Ministry of Science and Innovation (Grant nos. IJC2018-035395-I, FPU18/04476, and FPU20/01958).

## References

Aguilar-López, J.M., García, R.A., Sánchez, A.J., Gallego, A.J., Camacho, E.F., 2022. Mobile sensor for clouds shadow detection and direct normal irradiance estimation. Sol. Energy 237, 470–482.

Alessio, A., Bemporad, A., 2009. A survey on explicit model predictive control. In: Nonlinear Model Predictive Control. Springer, pp. 345–369.

Alsharkawi, A., Rossiter, J.A., 2016. Dual mode MPC for a concentrated solar thermal power plant. IFAC-PapersOnLine 49 (7), 260–265.

Ananduta, W., Ocampo-Martinez, C., 2021. Event-triggered partitioning for non-centralized predictive-control-based economic dispatch of interconnected microgrids. Automatica 132, 109829.

Awan, A.B., Mouli, K.V.C., Zubair, M., 2020. Performance enhancement of solar tower power plant: a multi-objective optimization approach. Energy Convers. Manage. 225, 113378.

Azadeh, A., Sheikhalishahi, M., Asadzadeh, S., 2011. A flexible neural network-fuzzy data envelopment analysis approach for location optimization of solar plants with uncertainty and complexity. Renew. Energy (ISSN: 0960-1481) 36 (12), 3394–3401.

Baldivieso-Monasterios, P.R., Trodden, P.A., 2021. Coalitional predictive control: consensus-based coalition forming with robust regulation. Automatica 125, 109380.

Barreiro-Gomez, J., Ocampo-Martinez, C., Quijano, N., 2019. Time-varying partitioning for predictive control design: density-games approach. J. Process Control 75, 1–14.

Bemporad, A., 2022. A piecewise linear regression and classification algorithm with application to learning and model predictive control of hybrid systems. IEEE Trans. Automat. Control.

Bianchini, A., Guzzini, A., Pellegrini, M., Saccani, C., 2019. Performance assessment of a solar parabolic dish for domestic use based on experimental measurements. Renew. Energy 133, 382–392.

Camacho, E.F., Berenguel, M., Rubio, F., Martínez, D., 2012. Control of solar energy systems. In: Advances in Industrial Control, Springer London, ISBN: 9780857299161, URL: https://books.google.es/books?id=LrExNjtUh0kC.

Camacho, E.F., Rubio, F.R., Berenguel, M., Valenzuela, L., 2007a. A survey on control schemes for distributed solar collector fields. Part I: modeling and basic control approaches. Sol. Energy 81 (10), 1240–1251.

Camacho, E.F., Rubio, F.R., Berenguel, M., Valenzuela, L., 2007b. A survey on control schemes for distributed solar collector fields. Part II: advanced control approaches. Sol. Energy 81 (10), 1252–1272.

Cao, J., Lin, X., 2008. Application of the diagonal recurrent wavelet neural network to solar irradiation forecast assisted with fuzzy technique. Eng. Appl. Artif. Intell. 21 (8), 1255–1263.

Ceusters, G., Rodríguez, R.C., García, A.B., Franke, R., Deconinck, G., Helsen, L., Nowé, A., Messagie, M., Camargo, L.R., 2021. Model-predictive control and reinforcement learning in multi-energy system case studies. Appl. Energy 303, 117634.

Chanfreut, P., Maestre, J.M., Camacho, E.F., 2021. A survey on clustering methods for distributed and networked control systems. Annu. Rev. Control 52, 75–90.

De Araújo Elias, T., Costa Mendes, P.R., Elias Normey-Rico, J., 2019. Mixed logical dynamical nonlinear model predictive controller for large-scale solar fields. Asian J. Control 21 (4), 1881–1891.

Drgoňa, J., Picard, D., Kvasnica, M., Helsen, L., 2018. Approximate model predictive building control via machine learning. Appl. Energy 218, 199–216.

Edalatifar, M., Ghalambaz, M., Tavakoli, M.B., Setoudeh, F., 2022. New loss functions to improve deep learning estimation of heat transfer. Neural Comput. Appl. 1–18.

Falugi, P., Mayne, D.Q., 2011. Tube-based model predictive control for nonlinear systems with unstructured uncertainty. In: Proceedings of the 50th Conference on Decision and Control and European Control Conference. IEEE, pp. 2656–2661.

Fele, F., Debada, E., Maestre, J.M., Camacho, E.F., 2018. Coalitional control for self-organizing agents. IEEE Trans. Automat. Control 63 (9), 2883–2897.

Fine, T.L., 2006. Feedforward Neural Network Methodology. Springer Science & Business Media.

Flores, A., Sáez, D., Araya, J., Berenguel, M., Cipriano, A., 2005. Fuzzy predictive control of a solar power plant. IEEE Trans. Fuzzy Syst. 13 (1), 58–68.

Frejo, J.R.D., Camacho, E.F., 2020. Centralized and distributed model predictive control for the maximization of the thermal power of solar parabolic-trough plants. Sol. Energy 204, 190–199.

Gallego, A.J., Sánchez, A.J., Camacho, E.F., 2022. Aplicaciones de control predictivo en plantas solares CCP. Revista Iberoamericana Automática E Informática Ind. 19 (3), 309–317.

Gálvez-Carrillo, M., De Keyser, R., Ionescu, C., 2009. Nonlinear predictive control with dead-time compensator: application to a solar power plant. Sol. Energy 83 (5), 743–752.

Ghalambaz, M., Jalilzadeh, Y.R., Davami, A.H., 2021. Building energy optimization using butterfly optimization algorithm (BOA). Therm. Sci. (00), 306.

Goswami, D.Y., Kreith, F., Kreider, J.F., 2000. Principles of Solar Engineering. CRC Press.

Hayat, M.B., Ali, D., Monyake, K.C., Alagha, L., Ahmed, N., 2019. Solar energy—A look into power generation, challenges, and a solar-powered future. Int. J. Energy Res. 43 (3), 1049–1067.

Islam, M.T., Huda, N., Abdullah, A., Saidur, R., 2018. A comprehensive review of state-of-the-art concentrating solar power (CSP) technologies: Current status and research trends. Renew. Sustain. Energy Rev. 91, 987–1018.

Kabir, E., Kumar, P., Kumar, S., Adelodun, A.A., Kim, K.-H., 2018. Solar energy: Potential and future prospects. Renew. Sustain. Energy Rev. 82, 894–900.

Kuleto, V., Ilić, M., Dumangiu, M., Ranković, M., Martins, O.M., Păun, D., Mihoreanu, L., 2021. Exploring opportunities and challenges of artificial intelligence and machine learning in higher education institutions. Sustainability 13 (18), 10424.

La Bella, A., Klaus, P., Ferrari-Trecate, G., Scattolini, R., 2022. Supervised model predictive control of large-scale electricity networks via clustering methods. Optim. Control Appl. Methods 43 (1), 44–64.

Lee, D., Ooka, R., Matsuda, Y., Ikeda, S., Choi, W., 2022. Experimental analysis of artificial intelligence-based model predictive control for thermal energy storage under different cooling load conditions. Sustainable Cities Soc. 103700.

Maestre, J.M., Lopez-Rodriguez, F., Muros, F.J., Ocampo-Martinez, C., 2021. Modular feedback control of networked systems by clustering: A drinking water network case study. Processes 9 (2), 389.

Marquardt, D.W., 1963. An algorithm for least-squares estimation of nonlinear parameters. J. Soc. Ind. Appl. Math. 11 (2), 431–441.

Masero, E., Fletscher, L.A., Maestre, J.M., 2020. A coalitional model predictive control for the energy efficiency of next-generation cellular networks. Energies 13 (24), 6546.

Masero, E., Maestre, J.M., Camacho, E.F., 2022. Market-based clustering of model predictive controllers for maximizing collected energy by parabolic-trough solar collector fields. Appl. Energy 306, 117936.

Masero, E., Maestre, J.M., Ferramosca, A., Francisco, M., Camacho, E.F., 2021. Robust coalitional model predictive control with predicted topology transitions. IEEE Trans. Control Netw. Syst. 8 (4), 1869–1880.

Maxim, A., Caruntu, C.-F., 2022. Coalitional distributed model predictive control strategy for vehicle platooning applications. Sensors 22 (3), 997.

Mayne, D.Q., 2014. Model predictive control: recent developments and future promise. Automatica 50 (12), 2967–2986.

McKinsey&Company, 2022. Global energy perspective 2022. URL: https://www.mckinsey.com/industries/oil-and-gas/our-insights/global-energy-perspective-2022. (Accessed 30 June 2022).

Møller, M.F., 1993. A scaled conjugate gradient algorithm for fast supervised learning. Neural Netw. (ISSN: 0893-6080) 6 (4), 525–533.

Navas, S.J., Rubio, F.R., Ollero, P., Lemos, J.M., 2018. Optimal control applied to distributed solar collector fields with partial radiation. Sol. Energy 159, 811–819.

Negenborn, R.R., Maestre, J.M., 2014. Distributed model predictive control: an overview and roadmap of future research opportunities. IEEE Control Syst. Mag. 34 (4), 87–97.

Ngiam, K.Y., Khor, W., 2019. Big data and machine learning algorithms for health-care delivery. Lancet Oncol. 20 (5), e262–e273.

Pane, Y.P., Nageshrao, S.P., Kober, J., Babuška, R., 2019. Reinforcement learning based compensation methods for robot manipulators. Eng. Appl. Artif. Intell. 78, 236–247.

Pipino, H.A., Morato, M.M., Bernardi, E., Adam, E.J., Normey-Rico, J.E., 2020. Nonlinear temperature regulation of solar collectors with a fast adaptive polytopic LPV MPC formulation. Sol. Energy 209, 214–225.

Pulido-Iparraguirre, D., Valenzuela, L., Serrano-Aguilera, J.-J., Fernández-García, A., 2019. Optimized design of a linear fresnel reflector for solar process heat applications. Renew. Energy 131, 1089–1106.

Ramachandran, P., Zoph, B., Le, Q.V., 2017. Searching for activation functions. arXiv preprint arXiv:1710.05941.

Riedmiller, M., Braun, H., 1993. A direct adaptive method for faster backpropagation learning: The RPROP algorithm. In: IEEE International Conference on Neural Networks. IEEE, pp. 586–591.

Ruiz-Moreno, S., Frejo, J.R.D., Camacho, E.F., 2021. Model predictive control based on deep learning for solar parabolic-trough plants. Renew. Energy 180, 193–202.

Rumelhart, D.E., Hinton, G.E., Williams, R.J., 1986. Learning representations by back-propagating errors. Nature 323 (6088), 533–536.

Safa, M., Kachitvichyanukul, V., 2019. Moment rotation prediction of precast beam to column connections using extreme learning machine. Struct. Eng. Mech. Int'L J. 70 (5), 639–647.

Shariati, M., Mafipour, M.S., Mehrabi, P., Zandi, Y., Dehghani, D., Bahadori, A., Shariati, A., Trung, N.T., Salih, M., Poi-Ngian, S., 2019. Application of extreme learning machine (ELM) and genetic programming (GP) to design steel-concrete composite floor systems at elevated temperatures. Steel Compos. Struct. 33 (3), 319–332.

Soriano, L.A., Zamora, E., Vazquez-Nicolas, J., Hernández, G., Barraza Madrigal, J.A., Balderas, D., 2020. PD control compensation based on a cascade neural network applied to a robot manipulator. Front. Neurorobotics 14, 577749.

Sun, Y., Li, S., Lin, B., Fu, X., Ramezani, M., Jaithwa, I., 2017. Artificial neural network for control and grid integration of residential solar photovoltaic systems. IEEE Trans. Sustain. Energy 8 (4), 1484–1495.

Tasmin, N., Farjana, S.H., Hossain, M.R., Golder, S., Mahmud, M.P., 2022. Integration of solar process heat in industries: a review. Clean Technol. 4 (1), 97–131.

Vaish, R., Dwivedi, U., Tewari, S., Tripathi, S.M., 2021. Machine learning applications in power system fault diagnosis: Research advancements and perspectives. Eng. Appl. Artif. Intell. 106, 104504.

Vasallo, M.J., Bravo, J.M., Cojocaru, E.G., Gegúndez, M.E., 2017. Calculating the profits of an economic MPC applied to CSP plants with thermal storage system. Sol. Energy 155, 1165–1177.

Xiao, H., Chen, C.P., 2021. Time-varying nonholonomic robot consensus formation using model predictive based protocol with switching topology. Inform. Sci. 567, 201–215.

Yang, S., Wan, M.P., Chen, W., Ng, B.F., Dubey, S., 2020. Model predictive control with adaptive machine-learning-based model for building energy efficiency and comfort optimization. Appl. Energy 271, 115147.

Ye, L., Zhang, C., Tang, Y., Zhong, W., Zhao, Y., Lu, P., Zhai, B., Lan, H., Li, Z., 2019. Hierarchical model predictive control strategy based on dynamic active power dispatch for wind power cluster integration. IEEE Trans. Power Syst. 34 (6), 4617–4629.

Yebra, L., Berenguel, M., Bonilla, J., Roca, L., Dormido, S., Zarza, E., 2010. Object-oriented modelling and simulation of ACUREX solar thermal power plant. Math. Comput. Model. Dyn. Syst. 16 (3), 211–224.

Yılmaz, İ.H., Mwesigye, A., 2018. Modeling, simulation and performance analysis of parabolic trough solar collectors: a comprehensive review. Appl. Energy 225, 135–174.