Renewable Energy 180 (2021) 193–202

Contents lists available at ScienceDirect

Renewable Energy

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

Model predictive control based on deep learning for solar parabolic-trough plants

Sara Ruiz-Moreno *, José Ramón D. Frejo , Eduardo F. Camacho

Dept. de Ingeniería de Sistemas y Automática, University of Seville, Camino de los Descubrimientos, E-41092, Seville, Spain

ARTICLE INFO

Article history:
Received 26 March 2021
Received in revised form
30 June 2021
Accepted 16 August 2021
Available online 20 August 2021

Keywords:
Solar energy
Model predictive control
Parabolic-trough collector
Artificial intelligence

ABSTRACT

In solar parabolic-trough plants, the use of Model Predictive Control (MPC) increases the output thermal power. However, MPC has the disadvantage of a high computational demand that hinders its application to some processes. This work proposes using artificial neural networks to approximate the optimal flow rate given by an MPC controller to decrease the computational load drastically to a 3% of the MPC computation time. The neural networks have been trained using a 30-day synthetic dataset of a collector field controlled by MPC. The use of a different number of measurements as inputs to the network has been analyzed. The results show that the neural network controllers provide practically the same mean power as the MPC controller with differences under 0.02 kW for most neural networks, less abrupt changes at the output and slight violations of the constraints. Moreover, the proposed neural networks perform well, even using a low number of sensors and predictions, decreasing the number of neural network inputs to 10% of the original size.

1. Introduction

Nowadays, renewable energy sources (hydropower, bioenergy, wind power and solar energy) are unseating the fossil fuel [1]. Among them, solar energy is the most ancient and abundant [2], as it takes advantage of solar radiation. The different solar power generation technologies can be divided into two main groups according to the method of producing energy: converting the sunlight directly into electricity with Photovoltaic Cells (PV) or producing steam to drive turbine generators in Concentrating Solar Power (CSP) systems. CSP technology can include Thermal Energy Storage (TES), which gives them an advantage over PV. The main types of CSP are parabolic troughs, Fresnel collectors, tower plants and dish collectors [3]. This paper focuses on the control of parabolic-trough collectors.

Many control algorithms have been proposed for parabolic-trough collector fields, gathered in Refs. [4,5]. Among them, Model Predictive Control (MPC) [6] is widely used in the literature because it can deal with nonlinear behaviors and constraints, and the use of receding horizon allows it to take into account future outputs. More details of the control techniques applied to these types of plants can be found in section 2.3.

The main drawback of MPC is the high computational cost required to solve an optimization problem every few minutes or seconds. This paper proposes the use of Artificial Neural Networks (ANN) to overcome this drawback.

There are two main approaches to the application of neural networks to control systems:

- The most commonly used approach consists of using the ANN to model the behavior of the plant [7]. gathers a list of applications to renewable energy systems and, more specifically, some applications to parabolic trough collectors are the following. In Ref. [8], a Nonlinear Autoregressive Exogenous (NARX) neural network is applied to obtain a non-linear model of the plant [9]. use an Elman neural network to tune offline switching PIDs and in Ref. [10], an MPC controller is applied using a state-space neural network as a model. In Ref. [11] a neural network is used to estimate the optimal operating point in a real-time optimization scheme for renewable energy sources power.
- The second approach is to calculate the control signal directly. To the best of our knowledge, it has not been applied to solar thermal plants, but there are some applications in other fields [12]. approximate the output of a receding

* Corresponding author.
E-mail addresses: srmoreno@us.es (S. Ruiz-Moreno), jdominguez3@us.es (J.R.D. Frejo), efcamacho@us.es (E.F. Camacho).

https://doi.org/10.1016/j.renene.2021.08.058

horizon controller using feedforward neural networks applied to control the trajectory of a robot [13]. use a neural network to solve the optimization problem in MPC [14]. learn to approximate the output of an MPC controller applied to an energy management system in a smart building.

The main contribution of this work is to apply neural networks to directly approximate the output of a nonlinear MPC controller for the control of parabolic trough collector fields. The proposed method provides control signals that approach the strengths of MPC —such as optimality and compliance with constraints—, but with much faster implementation times. In most controllers, one of the control objectives is to smooth the control signal and reduce the slew rate, which increases the pump durability, lowers high-frequency noise and reduces failures in electronic systems. A minor contribution is to obtain less abrupt changes in the control signal.

The whole process can be divided into three main steps: First, an MPC controller is implemented to generate a dataset. Then, several artificial neural networks are trained offline to learn its outputs. Finally, the proposed NN controller is tested by simulation for the ACUREX plant, which was located at the Plataforma Solar de Almería (PSA).

This paper organizes as follows. Section 2 describes the model of a parabolic-trough collector and briefly summarizes its main control algorithms. Section 3 explains artificial neural networks and the training process. The problem formulation is presented in section 4, where the MPC algorithm and the neural network controller are explained. Section 5 shows the design of the selected ANN and the simulation results for ACUREX. Finally, some conclusions are given in section 6.

2. Parabolic-trough plant description

This section describes the parabolic-trough collector (PTC) field, which consists of a set of mirrors that concentrate the Direct Normal Irradiance (DNI) onto a tube that contains the Heat Transfer Fluid (HTF), usually synthetic oil or water. This oil is heated to produce steam, which is fed to a turbine moving a generator that produces electricity. One of the main characteristics of this type of plant is the capacity for energy storage by means of a Thermal Energy Storage (TES) system.

First, a brief description of the model used is given, together with its parameters and variables. Then, the required constraints and boundary equations are described. These are needed to take into account when controlling the plant. Finally, this section summarizes the main control approaches used for controlling parabolic-trough collector plants.

2.1. Model of the plant

This section gives a brief description of the mathematical model used, which is the distributed-parameter model, and the parameters and variables. Fig. 1 shows a scheme of one loop of collectors, which is divided into several segments of size Δl. Then, Table 1 contains the notation used for describing the system, where the superscripts f, m and a refer to fluid, metal and ambient.

Table 1
Parameters and variables description.

Symbol	Description	Units
t	Time	s
ΔT	Simulation time step	s
Δl	Segment length	m
C	Specific heat capacity	J/(kg °C)
ρ	Density	kg/m^3
A	Cross-sectional Area	m^2
T	Temperature	°C
η^{col}	Collector efficiency	–
G	Collector aperture	m
I	Direct solar irradiance	W/m^2
D^{m}	Outside diameter of the pipe	m
D^{f}	Inside diameter of the pipe	m
H^{t}	Metal-fluid heat transmission coefficient	(m^2 °C)
H^{l}	Thermal loss coefficient	W/(m^2 °C)
q	Flow rate	1/s

The system dynamics can be described by equations (1)–(3) [4]:

- The energy balance for the metal tube on segment i from instant $(k-1)\Delta T$ to $k\Delta T$:

$$T_i^{\text{m}}(k) = T_i^{\text{m}}(k-1) + \frac{\Delta T}{\rho^{\text{m}} C^{\text{m}} A^{\text{m}}}\Big(\eta_i^{\text{col}} G_i I_i(k) - \pi D^{\text{m}} H_i^{\text{l}}(k-1) \left(T_i^{\text{m}}(k-1) - T^{\text{a}}(k)\right) + -\pi D^{\text{f}} H_i^{\text{t}}(k-1)\Big(T_i^{\text{m}}(k-1) - T_i^{1\text{f}}(k-1)\Big)\Big) \tag{1}$$

- The energy balance on the HTF for segment i:

$$T_i^{\text{f}}(k) = T_i^{\text{f}}(k-1) + \frac{\pi D^{\text{f}} H_i^{\text{t}}(k-1)\Delta T}{\rho_i^{\text{f}}(k-1) C_i^{\text{f}}(k-1) A^{\text{f}}}\Big(T_i^{\text{m}}(k-1) - T_i^{1\text{f}}(k-1)\Big) \tag{2}$$

- An auxiliary temperature, which allows incorporating the temperature of the adjacent segment in the energy balance:

$$T_i^{1\text{f}}(k) = T_i^{1\text{f}}(k-1) - \frac{q(k)\Delta T}{A^{\text{f}}}\Big(T_i^{\text{f}}(k) - T_{i-1}^{\text{f}}(k)\Big) \tag{3}$$

In solar fields, the Direct Normal Irradiance (DNI) $I_i(k)$ cannot be manipulated, so it acts as a disturbance that has to be estimated along the prediction horizon. More details about this model can be found in Refs. [15,16].

2.2. Constraints and boundary equations

A boundary equation is defined for the temperature of the first segment of the loop, where $T^{\text{in}}(k)$ is the inlet temperature, which acts as a disturbance that must be measured or estimated:

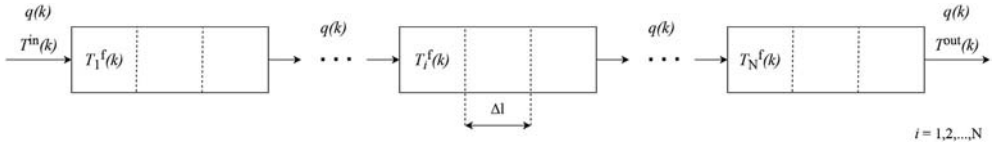

Fig. 1. Schematic of one loop of the collector field. $T^{\text{in}}(k)$ and $T^{\text{out}}(k)$ are the inlet and outlet temperatures, $T_i^{\text{f}}(k)$ is the fluid temperature in segment i and $q(k)$ is the flow rate.

$$T_1^{\mathrm{f}}(k) = T^{\mathrm{in}}(k) \tag{4}$$

Also, other two constraints must be considered:

- Constraints on the flow rate based on the minimum Reynolds number required to guarantee turbulent flow in the pipes and the maximum pressure drop, where $q^{\min}$ is the maximum flow rate and $q^{\max}$ is the minimum flow rate:

$$q^{\min} \leq q(k) \leq q^{\max} \tag{5}$$

- Constraints on the outlet temperature, where $T^{\mathrm{f,\ min}}$ is the minimum temperature of the HTF and $T^{\mathrm{f,\ max}}$ is the maximum:

$$T^{\mathrm{f,\ min}} \leq T_N^{\mathrm{f}}(k) \leq T^{\mathrm{f,\ max}} \tag{6}$$

2.3. Main controllers

Many control architectures, described by Ref. [17], have been used in parabolic trough collectors with the general purpose of controlling the flow rate and the HTF temperature: Feedforward (FF), Proportional-Integral-Derivative (PID), Cascade Control (CC), Adaptive Control (AC), Gain Scheduling (GS), Internal Model Control (IMC), Time Delay Compensation (TDC), Optimal Control (OC), Robust Control (RC), Non-linear Contol (NC), Model Predictive Control (MPC), Fuzzy Logic Control (FLC) and Neural Network Controllers (NNC). One of the most important is Model Predictive Control.

Among the recent works on this subject, the following articles should be noted [18]. propose an observer-based MPC, apply it to ACUREX and compare it to a Gain Scheduling Generalized Predictive Control (GS-GPC) and PID [19]. control the outlet temperature of the solar collector field using Filtered Dynamic Matrix Control (FDMC) [20]. apply MPC to the ACUREX plant manipulating the oil temperature and analyze the difference between manipulating each valve individually and the total oil flow [21]. present a strategy to manipulate the flow on the loops and obtain a thermal balance of the field [15]. compare different MPC approaches: local, distributed and centralized; and also compare different objectives: power maximization, temperature maximization, temperature minimization and no-valves situation [16]. control the plant with coalitional model predictive control [22]. take into consideration the effect of degraded collectors and their nonuniformity and propose using an improved thermal-hydraulic model of the field and an optimization strategy. In Ref. [23], an adaptive control method is implemented by solving two consecutive QP (Quadratic Programming) problems: the first one for establishing a momentaneous model and the second one for solving MPC. More economical approaches are [24], where the return of investment (ROI) is maximized, and [25], where the operation is optimized using a three-layer hierarchical control strategy to reduce the operating costs.

3. Artificial neural networks

Artificial Neural Networks (ANN) are models designed to emulate the human brain. Its origin takes back to 1943, when [26] modeled a neural network using electrical circuits. They have been widely used for modeling relationships between data and finding patterns. Its applications are categorized in regression, classification and data processing, many of them described by Ref. [27].

An Artificial Neural Network is a function $f_{\mathrm{NN}}(\cdot)$ formed by weighted sums of functions $g(\cdot)$ (called activation functions) that

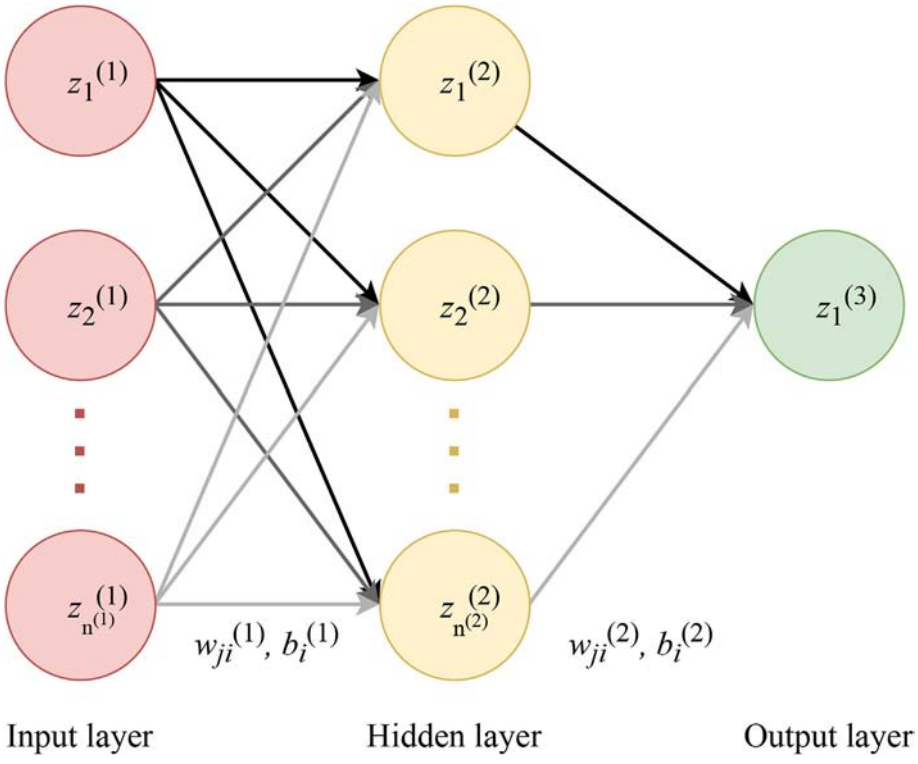

Fig. 2. Diagram of a Multilayer Perceptron with one hidden layer. $z_i^{(l)}$ is the output of node i and layer l, $w_{ji}^{(l)}$ is the weight vector betwen neurons j and j, and $b_i^{(l)}$ is the bias unit.

are inspired by the functioning of the human brain. It is formed by neurons (also called nodes), each of them with inputs emulating the dendrites and an output emulating the axon. Each neuron computes a linear regression problem and the activation function translates it to an active or non-active state. The use of several nodes gives rise to more complex functions. Neural networks are distributed in three types of layers: input layers, output layers and hidden layers (all layers between the input and the output, whose information is not directly related to the known data). According to the number of layers, neural networks can be shallow —they have only one hidden layer— or deep —they have two or more hidden layers.

The Multilayer Perceptron (MLP) is one of the most useful neural networks in function approximation. It is a feedforward neural network, which means that all neurons in a layer are connected to the neurons in the layer before and there are not cycles between them [28]. Each neuron of the network computes equation (7), giving $z_i^{(l)}$ as output. The superscript (l) indicates the layer, which is composed of $n^{(l)}$ neurons. $w_{ji}^{(l-1)}$ is the kernel between neurons j and i of layer $l-1$ and $b_i^{(l-1)}$ is the bias unit of neuron i in layer $l-1$. The activation functions are usually the same for each layer.

$$z_i^{(l)} = g^{(l)}\left(\sum_{j=1}^{n^{(l-1)}} w_{ji}^{(l-1)} z_j^{(l-1)} + b_i^{(l-1)}\right) \tag{7}$$

The general scheme of an MLP is shown in Fig. 2, where there is only one hidden layer (colored in yellow). The input layer is colored in red, and it has as many nodes as input variables. The output layer (in green) has as many nodes as the outputs of the neural network.

The activation function [29] defines the output of a node and allows the constraints in a certain range. The original activation function was a relay, but it is preferred to use other analog functions with smooth gradients, which prevents them from jumps in the output. Different types of activation functions allow multi-output and facilitate the learning process. Fig. 3 shows the most common activation functions. Non-linear activation functions allow creating non-linear regression when using multiple layers, whereas linear activation functions create simple linear combinations of linear functions, which produces linear regression. These functions are generally used in the output layer, as there is no need to saturate the output data. Also, the hyperbolic tangent function is zero-centered and has a steeper derivative than the sigmoid function, making it more efficient.

The parameters of the neural network are determined with an iterative process. First, they need to be initialized. Then, the output error is calculated and the weights are adjusted until fulfilling a given criterium. When a network has hidden layers, the errors must be calculated indirectly, starting from the output error. This is accomplished with an algorithm called back-propagation [30]. Different learning rules can be used according to this method. One of the most efficient back-propagation algorithms for medium-size neural networks is the Levenberg-Marquardt [31,32], which uses a sum of squared errors as a loss function.

The equations of neural networks solve a fitting problem. Consequently, any MLP can approximate continuous functions when having sufficient nodes. This is determined after a trial-and-error process: it starts with a few layers and neurons, a neural network is trained and the number of nodes is increased until founding a correct fitting.

The training process is composed of different steps, as follows:

- Pre-processing the data: Normalizing input and output variables leads to faster learning. The input data is usually scaled between −1 and 1 or between 0 and 1.
- Defining the subsets: The data is divided into training set —used for adjusting the parameters—, validation set —used to validate the behavior of the network and adjust some parameters— and test set —used for estimating the functioning of the network when fed with new data.
- Specifying the architecture and parameters of the network: This step must be repeated until finding a network that performs well.
- Learning: The weights are calculated.
- Evaluating the network: The Pearson correlation coefficient (R) and the Mean Squared Error (MCE) are calculated for each subset. If these values are not good enough, the process must be repeated from the third step.

4. Problem formulation

This section poses the formulation of the problem. First, the control strategy is described, including the cost function and calculations made to obtain it. Then, the use of the neural network to substitute the MPC controller is explained.

4.1. Model predictive control strategy

Model Predictive Control (MPC) [6] is a control method that uses a dynamic model of the system to predict the future outputs. Based on this model, an optimization problem is solved to obtain the best future control signals by minimizing a cost function. MPC has been successfully applied to this plant on several occasions [15,16,33]. In this work, we implement an MPC controller, whose inputs and outputs will serve to train an artificial neural network. The MPC strategy applied in this paper is analogous to the strategies applied in Refs. [15,16], but applied to only one loop. More details about the problem formulation can be found in the referred articles.

In this work, the control variable is the flow-rate circulating through the pipes. Regarding the objective function, commercial

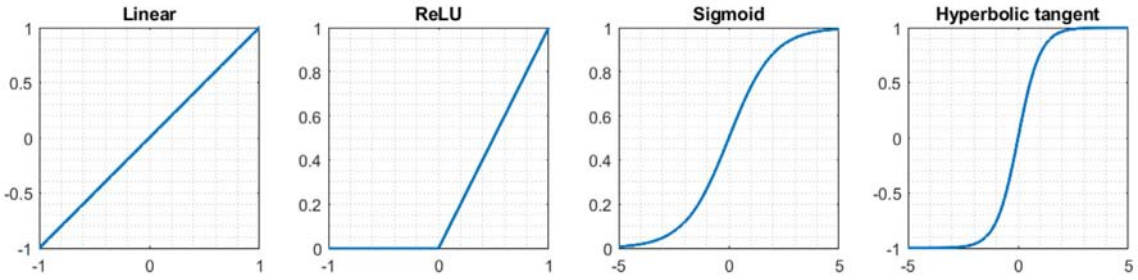

Fig. 3. Most known activation functions.

plants usually operate at the maximum temperature to increase the efficiency of the power cycle. This also implies an increase in the thermal losses of the field, affecting the overall efficiency as indicated by Ref. [25]. The main control objective in this work is then to maximize the net thermal power provided by the plant while fullfilling the constraints and minimizing the control effort.

The cost function $J(k_c)$ is given by equation (8) for a certain time $t = k_c\Delta T_c$, where k_c is the controller time-step and ΔT_c is the controller step size. It includes a term for the minimization of the thermal power $W(k_c)$, a second term to limit the outlet temperature using soft constraints and a third term that penalizes the flow-rate variation. ψ and ε are tuning parameters.

$$J(k_c) = \left(-W(k_c) + \psi \max\left(\frac{T_N^f(k_c) - T^{f,\max}}{T^{f,\max}}, \frac{T^{f,\min} - T_N^f(k_c)}{T^{f,\max}}, 0 \right)^2 + \varepsilon (q(k_c) - q(k_c - 1))^2 \right) \tag{8}$$

The net thermal power is computed by the difference between the output and input thermal powers for the collector field, $W^{out}(k_c)$ and $W^{in}(k_c)$, which can be approximated by the net power of the loop using equation (9).

$$W(k_c) = W^{out}(k_c) - W^{in}(k_c) \approx W_{loop}^{out}(k_c) - W_{loop}^{in}(k_c) \tag{9}$$

where $W^{out}(k_c)$ and $W^{in}(k_c)$ are the output and input thermal powers for the field and $W_{loop}^{out}(k_c)$ and $W_{loop}^{in}(k_c)$ are the output and input thermal powers of the control loop, being:

$$W_{loop}^{out}(k_c) = \rho_N^f(k_c) C_N^f(k_c) q(k_c) T_N^f(k_c) \tag{10}$$

$$W_{loop}^{in}(k_c) = \rho_1^f(k_c) C_1^f(k_c) q(k_c) T_1^f(k_c) \tag{11}$$

The optimal solution is given by the flow-rate values that maximize the thermal power for a given DNI profile. This is accomplished by optimizing the cost function in equation (8) along the prediction horizon, as in equation (12).

$$\begin{aligned} &\min_{q_t(k_c)} && \sum_{k=k_c}^{k_c+N_p} J(k) \\ &\text{s.t.} && q^{\min} < q(k_c) < q^{\max} \ \forall \ k_c \end{aligned} \tag{12}$$

where $q_t(k_c) = [q(k_c), q(k_c + 1), \ldots, q(k_c + N_u - 1)]$ and contains the flow-rate values throughout the control horizon, N_p is the prediction horizon, and N_u is the control horizon.

This problem has been solved using Sequential Quadratic Programming (SQP). As in Ref. [15], four initial points have been taken: the profile of the previous time step shifted, the lower and upper bounds and one random point.

The main problem of MPC is its computational requirements. It needs a long time to calculate the optimal solution, which makes it complicated to implement it in real-time on large solar plants and with long control horizons. To solve this problem, this work proposes in section 4.2 to use artificial neural networks that learn offline the output of the controller and apply it in real-time.

4.2. Neural network controller

This section aims to describe the implementation of the neural network controller based on MPC. The ultimate goal is to use the neural network to substitute the MPC controller in real-time operation. To do this, it will be trained using the inputs and outputs of the MPC controller.

The MPC controller obtains, for each time-step, a value for the flow-rate given some inputs: the flow rate in the previous time-step, the inlet temperature, the outlet temperature, the ambient temperature, the fluid temperature on each segment, the metal temperature on each segment and the predicted irradiance on each segment along the prediction horizon. This can be expressed as a function $f(\cdot)$ (equation (13)). The aim of this work is to find a function that approximates the output of the MPC controller obtaining a similar controller without its computational requirements (i.e. without the need to solve an optimization in real-time).

$$q(k) = f\Big(q(k-1), T^{in}(k), T^{out}(k), T^a(k), T_i^f(k), T_i^m(k), I_i(k), \ldots, I_i(k + N_p - 1)\Big) \tag{13}$$

The inputs to the function in equation (13) can be collected in a vector and used as input to the neural network $z_1^{(1)}(k) = [q(k-1), T^{in}(k), T^{out}(k), T^a(k), T_i^f(k), T_i^m(k), I_i(k), \ldots, I_i(k + N_p - 1)]$. The objective is to find the neural network whose output best matches the output of the MPC controller (equation (14)).

$$\hat{q}(k) = f_{NN}\left(z_1^{(1)}(k)\right) \tag{14}$$

This work is composed of two stages. First, an offline training of the ANN (Fig. 4) is done using the outputs of an MPC controller applied to a simulator of the plant with a dataset of radiations and ambient temperature. As the MPC controller calculates a series of future outputs along the control horizon, only the first one is used to feed the neural network —the one corresponding to the consequent instant. On the other hand, the entire set of inputs is used to fed to the neural network. Once trained, the network is implemented online to substitute the MPC controller in real-time (Fig. 5).

5. Results

5.1. Case study

The considered collector field was ACUREX, located at the Plataforma Solar de Almería (PSA), Spain. As aforementioned, this plant has been widely used in the literature. In particular, the different experiments of this work are carried out by simulating one loop of the plant.

The field is composed of East-West aligned single axis parabolic-trough collectors. Each loop is 174 m long and is discretized into $N = 174$ segments of $\Delta l = 1$. The loop has 12 collectors in series, being segments 37, 42, 79, 96 and 133, 138 passive parts. The active part is the one that receives solar irradiance and is 144 m long, whereas the passive part is 30 m long.

The HTF used is Therminol VP-1, with density $\rho_i^f(k)$ and specific heat capacity $C_i^f(k)$ for each time instant are computed for each segment i using equations (15) and (16).

$$\rho_i^f(k) = 903 - 0.672 T_i^f(k) \tag{15}$$

$$C_i^f(k) = 1820 - 3.478 T_i^f(k) \tag{16}$$

The coefficient of transmission metal-fluid and the coefficient of thermal losses are also computed according to equations (17) and

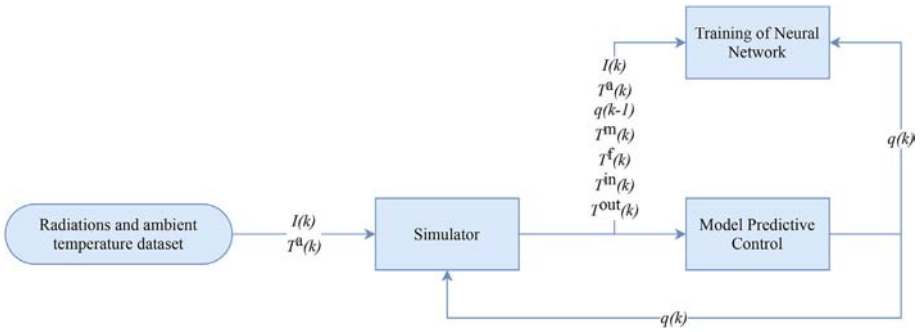

Fig. 4. Offline training of the control algorithm with the notation given by Table 1.

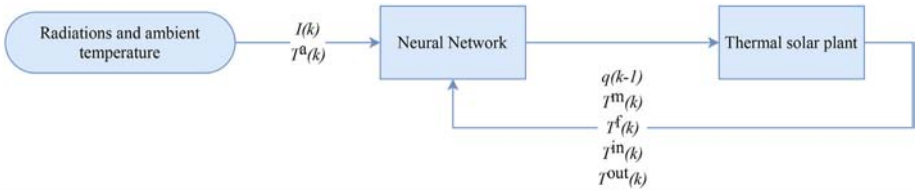

Fig. 5. Online implementation of the control algorithm with the notation given by Table 1.

(18).

$$H_i^t(k) = q_k^{0.8}\left(2.17\cdot 10^6 - 5.01\cdot 10^4 T_i^f(k) + 4.53\cdot 10^2 T_i^f(k)^2 - 1.64 T_i^f(k)^3 + 2.1\cdot 10^{-3} T_i^f(k)^4\right) \tag{17}$$

$$H_i^l(k) = 0.00249\left(T_i^f(k) - T^a(k)\right) - 0.06133 \tag{18}$$

The rest of the values are shown in Table 2, as well as the operational constraints. Later in this document, it will be shown that small violations of the outlet temperature constraints are possible with the proposed method. Although these occur in just a few moments, they could be avoided by augmenting the lower bound.

In solar collectors, the disturbances are the ambient temperature and the effective DNI. The ambient temperature is considered constant and equal to 25 °C. A 30-days DNI profile has been used, in which different synthetic clouds have been included to obtain a heterogeneous dataset. Fig. 6 shows the first day of the DNI used for training. Another one-day profile was used for validation, represented in Fig. 7.

The inlet temperature can be expressed as a first order system with a time constant of 10 min as in equation (19), where $\hat{T}^{out}(s) = T^{out}(s) - 90$ to take into account the temperature fall in the steam generator.

$$\frac{T^{in}(s)}{\hat{T}^{out}(s)} = \frac{1}{600s + 1} \tag{19}$$

The discretization of this equation leads to equation (20). A discretization time of 0.5 s was used, which is the simulation time step, also used by Refs. [15,16,34].

$$T^{in}(k) = 0.999167 T^{in}(k-1) + 0.000833\left(T^{out}(k-1) - 90\right) \tag{20}$$

The last measured value of the outlet temperature $T^{out}(k_c)$ is used for all the prediction horizon and the parameters of the MPC controller are given by Table 3. To speed up the computation time, although the simulation model considers a length of the segments is $\Delta l = 1$ m and the discretization time of the model is $\Delta T = 0.5$ s, the MPC controller uses a length $\Delta l_c = 6$ m and a discretization time $\Delta T_{m,c} = 3$ s.

Table 2
Parameters and constraints of ACUREX.

Symbol	Value	Units
ΔT	0.5	s
ρ^m	7800	kg/m^3
C^m	550	J/(kg °C)
D^m	0.031	m
D^f	0.0254	m
A^m	$2.48 \cdot 10^{-4}$	m^2
A^f	$7.55 \cdot 10^{-4}$	m^2
q^{min}	0.2	l/s
q^{max}	1.5	l/s
$T^{f,\,min}$	220	°C
$T^{f,\,max}$	300	°C

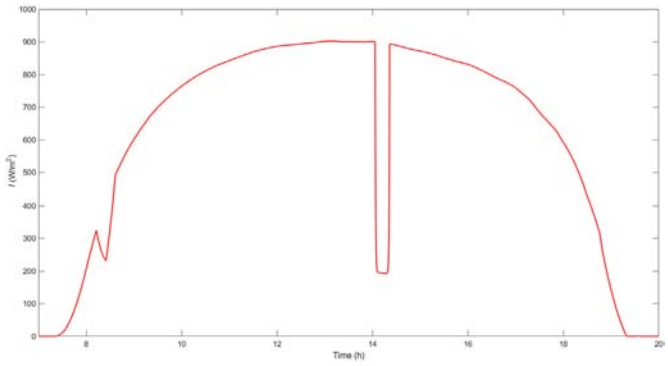

Fig. 6. First day of the DNI profile used for training.

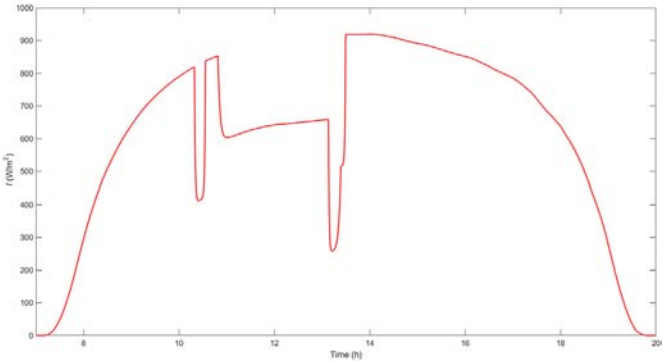

Fig. 7. DNI profile used for validation.

Table 3
Parameters of the MPC controller.

Symbol	Value	Units
ΔT_c	60	s
$\Delta T_{m,\,c}$	3	s
Δl_c	6	m
N_p	12	–
N_u	10	–
ψ	45	–
ε	3	–

5.2. Neural network design

This section is dedicated to explaining the design and training of the neural network. On the one hand, a description of the data and its processing is given. On the other hand, the architecture and inputs of the neural network are explained.

The 30-days dataset has been divided into a training set (70%), a validation set (15%) and a test set (15%), randomly selected. The total number of samples is 16867. Afterward, the data is scaled in the range [−1, 1]. This is done to avoid inappropriate saturations caused by the use of the activation functions. With high input values, the weight vectors would have to be very small reducing gradients excessively and slowing down the training [7,35].

Several neural networks have been trained using different numbers and sizes of hidden layers. The training was stopped once the neural network achieved a mean squared error under 10^{-9}. The activation functions used are hyperbolic tangent for all layers except for the last one (the output layer), where a linear function was used. The parameters are given by Table 4.

Different choices have been used for the selection of the segments in which measurements of temperatures and irradiation are taken and for the number of predictions used. According to this, eight different cases have been investigated:

1. Temperatures and irradiance every six segments.
2. Irradiance every six segments and temperatures at the center of each collector.
3. Irradiance every six segments, predictions at instants 1,4,8 and 12, and temperatures at the center of each collector.
4. Irradiance every six segments, predictions at instants 1 and 12, and temperatures at the center of each collector.
5. Irradiance every six segments, prediction horizon of 6, and temperatures at the center of each collector.
6. Irradiance every six segments, prediction horizon of 3, and temperatures at the center of each collector
7. Irradiance every six segments, prediction horizon of 1, and temperatures at the center of each collector.
8. Temperature and irradiance in the first collector, without inlet and outlet temperatures, and prediction horizon of 1.

5.3. Simulation results

This section presents the results obtained with the different neural networks applied to the plant. First, different neural networks are trained using temperatures and irradiance every six segments and one architecture is selected to be trained with the rest of the options. Then, the neural networks are tested with a validation profile.

Table 5 gathers the results of the neural networks used for case 1 (temperatures and irradiance every six segments and the entire prediction horizon). The Pearson correlation coefficient R has been calculated in open loop for the three subsets. After that, the neural networks have been used to control the plant and the mean power obtained is represented in the table. One advantage of neural networks is the obtaining of a smooth output to the detriment of slight violations of the constraints. For that reason, the table includes the accumulated absolute difference between the value of the flow rate in an instant and the next one, which we define as Accumulated Absolute Control Increment (AACI) in equation (21), and the Mean Squared Constraint Violation (MSCV), a measurement of the differences between T^{out} and its limit when constraint violations occur between the hours 8:00 and 19:00 –times between power on and off are not considered–, defined in equation (22), where n_s is the number of samples between those instants.

$$\mathrm{AACI} = \sum_k |q(k) - q(k-1)| \tag{21}$$

$$\mathrm{MSCV} = \frac{1}{n_s} \sum_{k\Delta T=8}^{k\Delta T=19} \left(\max \left(T^{f,\,min} - T^{out}(k), T^{out}(k) - T^{f,\,max}, 0 \right) \right)^2 \tag{22}$$

From the table, it can be extracted that either a shallow network or deep networks with three hidden layers obtain smoother outputs than MPC. Although the network with only one layer obtains more power –surpassing even the MPC controller–, it is due to the violations of the constraints. For that reason, the selected neural network for the next experiments is the third one (two hidden layers, with 15 neurons in the first one and 10 neurons in the second one), with lower violations of the constraints.

Fig. 8 compares the performance of the selected neural network and the MPC controller throughout the daylight hours of one of the days used for training. It should be noted that the curves corresponding to the artificial neural networks are much smoother than those obtained with MPC.

Regarding the calculation times of both controllers, Fig. 9 shows that the MPC controller is much slower than the neural network at

Table 4
Training parameters of the neural networks.

Range	Transfer function	μ	μ increase ratio	μ decrease ratio	Max μ	Max epochs	Min gradient	Max validation checks
[-1,1]	tansig pureling	10^{-3}	10^{-1}	10	10^{10}	10^{3}	10^{-9}	6

Table 5
Results of the controllers used for case 1 (temperatures and irradiance every six segments and the entire prediction horizon) with the first day of the training data.

Neurons	R (train)	R (validation)	R (test)	Mean power	AACI	MSCV
MPC	–	–	–	65.6607 kW	7.9309 l/s	0.0445
15	0.99720	0.99690	0.99651	65.6638 kW	2.8189 l/s	0.4706
15–5	0.99793	0.99678	0.99647	65.6515 kW	2.7983 l/s	0.1016
15–10	0.99791	0.99636	0.99679	65.6580 kW	2.9247 l/s	0.0740
20–10	0.99778	0.99664	0.99621	65.6508 kW	2.9479 l/s	0.1760
15-10-5	0.99771	0.99699	0.99686	65.6557 kW	2.8435 l/s	0.5388
15-10-10	0.99739	0.99685	0.99685	65.6619 kW	2.9479 l/s	0.5182

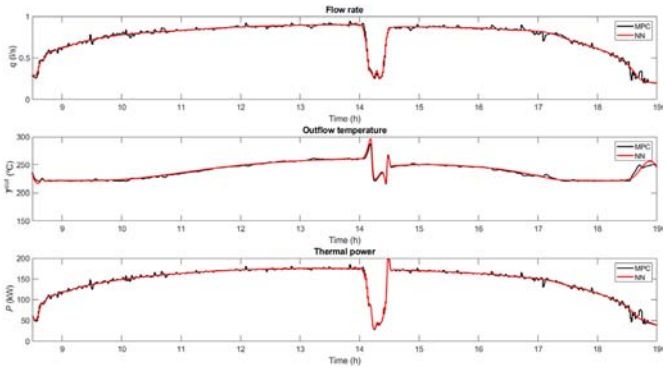

Fig. 8. Evolution of the flow rate, the outlet temperature and the thermal power for one day of the profile used for training.

all times. The mean time for the MPC controller is 2.3929 s with a standard deviation of 1.1418 s. In the case of the neural network, the mean time is 0.07 s and the standard deviation is 0.0087 s.

Subsequently, different neural networks have been trained for the rest of the cases, using for each one a lower number of inputs. All of them are designed using the same architecture –2 hidden layers, with 15 neurons in the first one and 10 in the second one– and the same parameters as in the first case. Fig. 10 shows the architecture of the selected neural network with the number of inputs indicated in each layer. The only difference with the neural networks of the rest of the cases is the size of the input layer.

Table 6 shows the results obtained for the neural networks trained for cases 2 to 8 (explained in section 5.2) using the 15-10 neural network. It also shows the Pearson correlation coefficient for the training, validation and test subsets, the mean power obtained in a one-day simulation, the accumulated absolute difference between the value of the flow rate in two consecutive instants and the MSCV. It can be extracted that, even eliminating some sensors and predictions, the artificial neural networks are capable of learning by

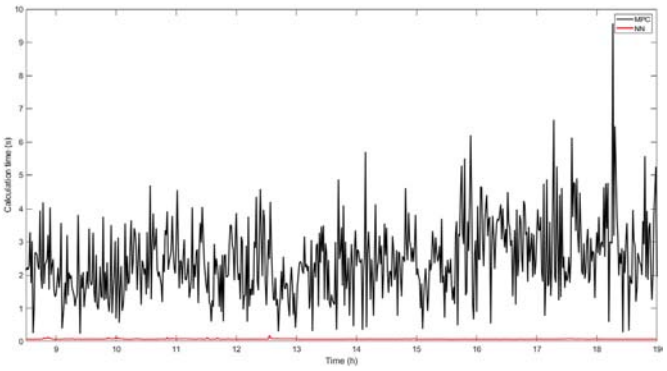

Fig. 9. Calculation times of MPC and the ANN with the first day of the profile used for training.

themselves thanks to the great behavior of the MPC controller used for training, which takes into account future predictions and estimations of the entire loop. The last two cases, in which no predictions are used, are the ones that obtain the worst performance, with great violations of the constraints. Even though, the mean power is near the one obtained with the MPC controller (65.6607 kW) and the flow rate remains smoother, which is desirable in these kinds of systems restricted by the physical capabilities of the valves.

To visualize how the neural network responds to different prediction horizons, Fig. 11 shows the MSCV and the q differences between two instants and their evolution when the prediction horizon decreases. It is worth noting that the neural networks are capable of replicating the behavior of MPC despite the decrease in the number of inputs, except for the case with only one prediction step. A slight decrease in the AACI is also visible as the prediction horizon decreases.

For validation, a new DNI profile is used, obtaining the results of Table 7. In all cases, the constraint violations increase, providing a slight increment in the mean power. For case 8, the neural network is unable to control the system, which is destabilized from hour 10:50. Note that this neural network has only five inputs and no predictions are made.

The overall performance for the first case is good and the AACI reduction is maintained, as it can be seen in Fig. 12. To visualize the ability of the neural networks to learn the prediction and estimations, Fig. 13 represents the results obtained for case 7 in validation. This is the worst neural network (not taking into account case 8), as the prediction horizon has been set to 1.

6. Conclusion

The use of MPC to control solar plants allows the maximization of thermal power while satisfying certain constraints and considering future solar radiation values. The drawback of MPC is the computational cost required to solve optimization problems in non-linear systems. This paper proposes using artificial neural networks to approximate the output of the model predictive controller.

In this work, a model of the ACUREX collector field has been used, together with two different DNI profiles. A dataset has been created out of a 30-days simulation of the plant controlled by MPC and another 1-day profile of irradiation has been used for validation purposes. These data have been used to train different architectures of neural networks and several tests have been carried out eliminating some inputs (from 410 inputs in case 1 to 41 inputs in case 7 or 5 inputs in case 8). From those tests, the following conclusions can be extracted:

- The value of the mean power obtained is similar using MPC and neural networks.
- The neural networks obtain much smoother values of the flow rate.

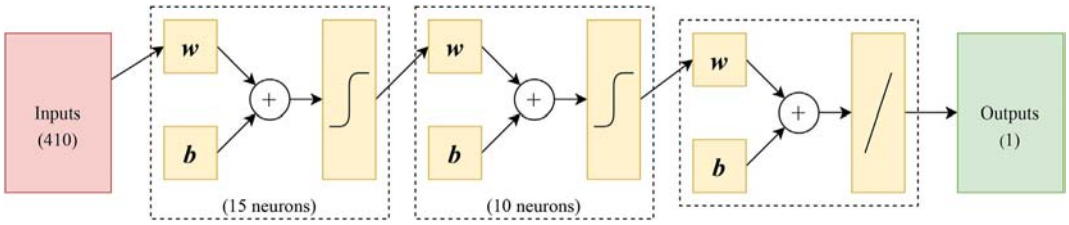

Fig. 10. Diagram of the selected artificial neural network for case 1 (temperatures and irradiance every six segments and the entire prediction horizon).

Table 6
Results of the controllers used for cases 2 to 8 with the first day of the training data (15-10 neurons).

Case	R (train)	R (validation)	R (test)	Mean power	AACI	T^{out} MSCV
2	0.99760	0.99719	0.99671	65.6581 kW	2.9096 l/s	0.1561
3	0.99768	0.99708	0.99704	65.6535 kW	2.9483 l/s	0.2810
4	0.99748	0.99647	0.99692	65.6629 kW	2.7363 l/s	0.7317
5	0.99772	0.99668	0.99677	65.6548 kW	2.7612 l/s	0.2239
6	0.99694	0.99663	0.99670	65.6599 kW	2.8096 l/s	0.0767
7	0.99686	0.99575	0.99649	65.6707 kW	2.7097 l/s	1.3840
8	0.99547	0.99542	0.99426	65.5358 kW	2.7726 l/s	26.0222

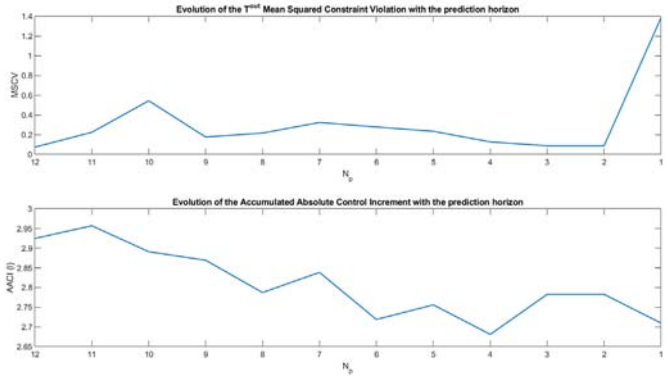

Fig. 11. Evolution of the MSCV and AACI with different prediction horizons for the first day of the profile used for training.

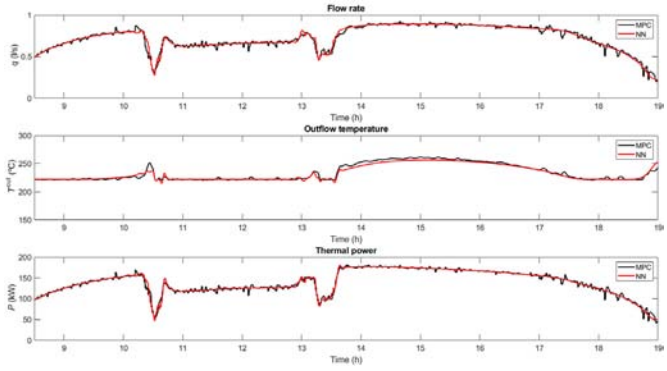

Fig. 12. Evolution of the flow rate, the outlet temperature and the thermal power for the profile used for validating case 1 (temperatures and irradiance every six segments and the entire prediction horizon).

Table 7
Results of the controllers used for cases 2 to 7 with the validation profile (15-10 neurons).

Neurons	Mean power	AACI	MSCV
MPC	64.0417 kW	9.9777 l/s	0.0033
case 1	64.0517 kW	3.8315 l/s	0.1490
case 2	64.0504 kW	3.5275 l/s	0.3651
case 3	64.0481 kW	3.2928 l/s	0.5636
case 4	64.0515 kW	3.0231 l/s	1.6368
case 5	64.0474 kW	3.4924 l/s	0.0393
case 6	64.0538 kW	3.2521 l/s	0.1317
case 7	64.0615 kW	2.9464 l/s	3.2281

- It is possible to find neural networks with very low violations in T^{out} constraints.
- The use of neural networks reduces the computation time significantly.

The reduction of computation times allows an implementation of the controller in real-time, as not only are they more constants, but also they are much lower than the sampling time of the controller and the computation times of the MPC controller. This also allows the application of these neural networks to plants with a great number of loops, where the implementation of MPC in real-time is much complicated. In this work, only one loop is considered and the time needed for calculating the MPC controller is around 3 s (even 7 or 9 s at some instants), whereas using the neural network, it is around 0.1 s. Scaling the plant to the real size, with 10 loops in the case of ACUREX or more than 100 in commercial plants, it would not be possible to obtain the optimal solution in real-time with MPC.

One limitation encountered when using neural networks is the violation of the constraints, as they are not directly taken into account. Although the total error of these violations is relatively small, it can be reduced by imposing harder constraints to the MPC controller with a smaller temperature range.

Future research will be the application of this methodology to a larger plant, other implementations combining the artificial neural networks with distributed MPC and coalitional control and the use of different types of learning techniques, such as non-supervised learning.

CRediT authorship contribution statement

Sara Ruiz-Moreno: Conceptualization, Methodology, Software, Investigation, Writing – original draft. **José Ramón D. Frejo:** Conceptualization, Methodology, Software, Writing – review & editing. **Eduardo F. Camacho:** Supervision, Writing – review &

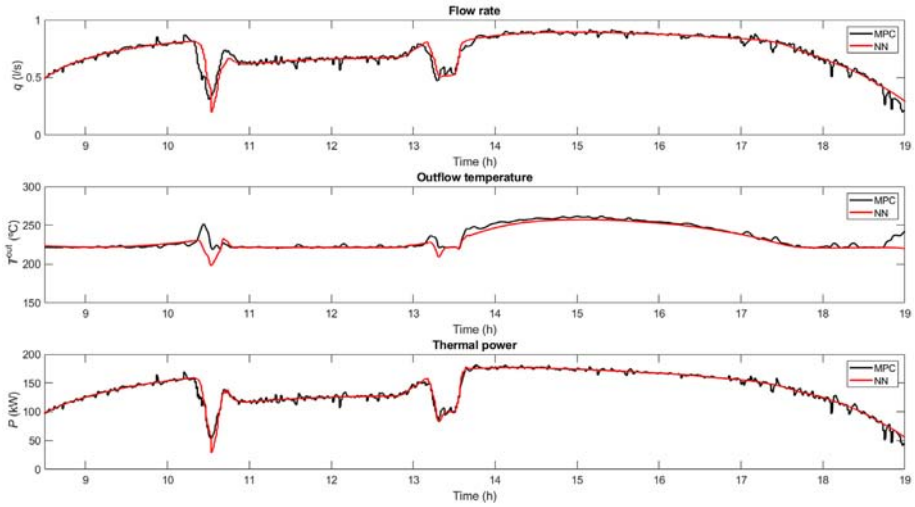

Fig. 13. Evolution of the flow rate, the outlet temperature and the thermal power for the profile used for validating case 7 (irradiance every six segments, prediction horizon of 1, and temperatures at the center of each collector).

editing, 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.

Acknowledgement

The authors gratefully acknowledge the financial support by the European Research Council [Advanced Grant OCONTSOLAR number 789051] and the Spanish Ministry of Science, Innovation, and Universities [Grant number IJC2018-035395-I].

References

[1] G. Destouni, H. Frank, Renewable energy, Ambio 39 (Suppl 1) (2010) 18–21.
[2] Z. Şen, Solar energy in progress and future research trends, Prog. Energy Combust. Sci. 30 (4) (2004) 367–416.
[3] H. Zhang, J. Baeyens, J. Degrève, G. Cacères, Concentrated solar power plants: review and design methodology, Renew. Sustain. Energy Rev. 22 (2013) 466–481.
[4] E.F. Camacho, F. Rubio, M. Berenguel, L. Valenzuela, A survey on control schemes for distributed solar collector fields. part i: modeling and basic control approaches, Sol. Energy 81 (2007) 1240–1251.
[5] E.F. Camacho, F.R. Rubio, M. Berenguel, L. Valenzuela, A survey on control schemes for distributed solar collector fields. part ii: advanced control approaches, Sol. Energy 81 (10) (2007) 1252–1272.
[6] E.F. Camacho, C. Bordons, Model Predictive Control, Springer Science & Business Media, 2013.
[7] S.A. Kalogirou, Artificial neural networks in renewable energy systems applications: a review, Renew. Sustain. Energy Rev. 5 (4) (2001) 373–401.
[8] M. Berenguel, M.R. Arahal, E.F. Camacho, Modelling the free response of a solar plant for predictive control, Contr. Eng. Pract. 6 (10) (1998) 1257–1266.
[9] A.L. Cardoso, J. Henriques, A. Dourado, Fuzzy supervisor and feedforward control of a solar power plant using accessible disturbances, in: 1999 European Control Conference, ECC, 1999, pp. 1711–1716.
[10] P. Gil, J. Henriques, P. Carvalho, H. Duarte-Ramos, A. Dourado, Adaptive neural model-based predictive control of a solar power plant, in: Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02 (Cat. No.02CH37290), vol. 3, 2002, pp. 2098–2103, 3.
[11] A. Kebir, L. Woodward, O. Akhrif, Real-time optimization of renewable energy sources power using neural network-based anticipative extremum-seeking control, Renew. Energy 134 (2019) 914–926.
[12] T. Parisini, R. Zoppoli, A receding-horizon regulator for nonlinear systems and a neural approximation, Automatica 31 (10) (1995) 1443–1451.
[13] B.M. Åkesson, H.T. Toivonen, A neural network model predictive controller, J. Process Contr. 16 (9) (2006) 937–946.
[14] B. Karg, S. Lucia, Deep learning-based embedded mixed-integer model predictive control, in: 2018 European Control Conference (ECC), IEEE, 2018, pp. 2075–2080.
[15] J.R.D. Frejo, E.F. Camacho, Centralized and distributed model predictive control for the maximization of the thermal power of solar parabolic-trough plants, Sol. Energy 204 (2020) 190–199.
[16] E. Masero, J.R.D. Frejo, J.M. Maestre, E.F. Camacho, A light clustering model predictive control approach to maximize thermal power in solar parabolic-trough plants, Sol. Energy 214 (2021) 531–541.
[17] E.F. Camacho, M. Berenguel, F.R. Rubio, D. Martinez Plaza, Control of Solar Energy Systems, Springer, 2012.
[18] A.J. Gallego, F. Fele, E.F. Camacho, L. Yebra, Observer-based model predictive control of a parabolic-trough field, Sol. Energy 97 (2013) 426–435.
[19] D.M. Lima, J.E. Normey-Rico, T.L.M. Santos, Temperature control in a solar collector field using filtered dynamic matrix control, ISA (Instrum. Soc. Am.) Trans. 62 (2016) 39–49, sl: Control of Renewable Energy Systems.
[20] S.J. Navas, F.R. Rubio, P. Ollero, J.M. Lemos, Optimal control applied to distributed solar collector fields with partial radiation, Sol. Energy 159 (2018) 811–819.
[21] A.J. Sánchez, A.J. Gallego, J.M. Escaño, E.F. Camacho, Thermal balance of large scale parabolic trough plants: a case study, Sol. Energy 190 (2019) 69–81.
[22] L. Ma, T. Zhang, X. Zhang, B. Wang, S. Mei, Z. Wang, X. Xue, Optimization of parabolic trough solar power plant operations with nonuniform and degraded collectors, Sol. Energy 214 (2021) 551–564.
[23] H.A. Pipino, M.M. Morato, E. Bernardi, E.J. Adam, J.E. Normey-Rico, Nonlinear temperature regulation of solar collectors with a fast adaptive polytopic lpv mpc formulation, Sol. Energy 209 (2020) 214–225.
[24] E.F. Camacho, A.J. Gallego, Advanced control strategies to maximize roi and the value of the concentrating solar thermal (cst) plant to the grid, in: M.J. Blanco, L.R. Santigosa (Eds.), Advances in Concentrating Solar Thermal Research and Technology, Woodhead Publishing Series in Energy, Woodhead Publishing, 2017, pp. 311–336.
[25] E.F. Camacho, A.J. Gallego, Optimal operation in solar trough plants: a case study, Sol. Energy 95 (2013) 106–117.
[26] W.S. McCulloch, W. Pitts, A logical calculus of the ideas immanent in nervous activity, Bull. Math. Biophys. 5 (4) (1943) 115–133.
[27] B. Widrow, D.E. Rumelhart, M.A. Lehr, Neural networks: applications in industry, business and science, Commun. ACM 37 (3) (1994) 93–106.
[28] T.L. Fine, Feedforward Neural Network Methodology, Springer Science & Business Media, 2006.
[29] P. Ramachandran, B. Zoph, Q. V. Le, Searching for Activation Functions, arXiv preprint arXiv:1710.05941.
[30] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning representations by back-propagating errors, nature 323 (6088) (1986) 533–536.
[31] K. Levenberg, A method for the solution of certain non-linear problems in least squares, Q. Appl. Math. 2 (2) (1944) 164–168.
[32] D.W. Marquardt, An algorithm for least-squares estimation of nonlinear parameters, J. Soc. Ind. Appl. Math. 11 (2) (1963) 431–441.
[33] E.F. Camacho, M. Berenguel, F.R. Rubio, Advanced Control of Solar Plants, Springer, 1997.
[34] A.J. Gallego, E.F. Camacho, Estimation of effective solar irradiation using an unscented kalman filter in a parabolic-trough field, Sol. Energy 86 (12) (2012) 3512–3518 (solar Resources).
[35] J. Sola, J. Sevilla, Importance of input data normalization for the application of neural networks to complex industrial problems, IEEE Trans. Nucl. Sci. 44 (3) (1997) 1464–1468.