ISA Transactions xxx (xxxx) xxx

Contents lists available at ScienceDirect

ISA Transactions

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

Research article

# System identification and fault reconstruction in solar plants via extended Kalman filter-based training of recurrent neural networks☆

Sara Ruiz-Moreno [a],*,[1], Alberto Bemporad [a], Antonio Javier Gallego [b], Eduardo Fernández Camacho [b]

[a] IMT School for Advanced Studies Lucca, 55100, Lucca, Italy
[b] Dept. de Ingeniería de Sistemas y Automática, University of Seville, Camino de los Descubrimientos, no number E-41092, Seville, Spain

ARTICLE INFO

*Keywords:*
Extended Kalman filter
Recurrent neural network
Fault diagnosis
Solar energy
Deep learning

ABSTRACT

This article proposes using the extended Kalman filter (EKF) for recurrent neural network (RNN) training and fault estimation within a parabolic-trough solar plant. The initial step involves employing an RNN to model the system. Given the challenge of fault discernibility in the collectors, parallel EKFs are employed to reconstruct the parameters of the faults. The parameters are used independently to estimate the system output, and the type of fault is isolated based on the estimation errors using another feedforward neural network. To evaluate the effectiveness of the methodology, simulations are conducted on a loop of the ACUREX plant with irradiances from sunny and cloudy days. The results reveal a fault classification accuracy of approximately 90% and a fault reconstruction error below 3%, with even better accuracies in the cloudy dataset than in the sunny dataset.

## 1. Introduction

In modern industries, many autonomous systems are integrated into processes, resulting in an increasing number of sensors and actuators that may fail unexpectedly. However, this progress also brings with it concerns about safety and reliability [1], a focus that is also crucial for today's companies [2]. Therefore, there is a need to automatically detect faults and failures in the system [3]. To this end, the field of fault detection and diagnosis (FDD) aims to determine the occurrence of a fault and reveal some relevant information about it. Fault diagnosis is further categorized into fault isolation, which locates and assesses the type of fault, and fault identification, which determines its magnitude.

FDD is a rapidly expanding field with diverse applications across various industrial sectors. For instance, Molinié et al. [4] propose an unsupervised clustering method to identify anomalies in industrial systems. Their approach involves recursively partitioning a space using an integrity criterion. In another study, Chen et al. [5] introduce an interpretable mechanism based on convolutional neural networks with score-weighted class activation to classify faults in climate control systems. Abdel Karim et al. [6] apply FDD to communication networks using power-line communications and residuals. In the field of photovoltaics, Hajji et al. [7] apply and compare different artificial neural networks (ANNs) to diagnose several typical faults. The work by Kumar and Devakumar [8] focuses on diagnosing sensor faults by estimating state variables with recurrent neural networks (RNNs) and obtaining residuals. In the event of a fault, a new ANN classifies it into different sensor conditions. Rodríguez et al. [9] use digital twins in a solar cooling plant, incorporating a neuro-fuzzy system to detect faults and an RNN to identify them. Additionally, a Kalman filter with the Rauch-Tung-Striebel (RTS) smoother is used by Bidou et al. [10] to estimate failure and restart times in heat sources.

As introduced earlier, the FDD field is enriched by the use of artificial intelligence (AI), with artificial neural networks being one of its well-known components. The use of AI is well developed in systems engineering, with applications such as intrusion monitoring [11], digital twins in Fresnel plants [12], reinforcement learning for adaptive control of solar collector plants [13], or Lagrange multipliers initialization for computational reduction in distributed model predictive control (MPC) [14]. The continuous evolution of technology has allowed the

☆ This research was supported by the European Union's Horizon 2020 research and innovation program under the ERC Advanced Grant agreement No 789051, and by the Spanish Ministry of Science and Innovation (Grant n. FPU20/01958).

* Corresponding author.
*E-mail addresses:* srmoreno@us.es (S. Ruiz-Moreno), alberto.bemporad@imtlucca.it (A. Bemporad), agallego2@us.es (A.J. Gallego), efcamacho@us.es (E.F. Camacho).
[1] IMT Permanent address: Dept. de Ingeniería de Sistemas y Automática, University of Seville, Camino de los Descubrimientos, no number E-41092, Seville, Spain.

https://doi.org/10.1016/j.isatra.2025.01.002
Received 21 March 2024; Received in revised form 2 January 2025; Accepted 2 January 2025
Available online 8 January 2025

**Nomenclature**

**Parameters and variables**

| | |
|---|---|
| $\alpha$(t) | Fault multiplier (–) |
| $\delta_s$ | Declination (°) |
| $\omega_s$(t) | Hourly angle (°) |
| $\rho(T)$ | Density (kg/m$^3$) |
| $A$ | Pipe cross-sectional area (m$^2$) |
| $C(T)$ | Specific heat capacity (J/ (kg °C)) |
| $G$ | Collector aperture (m) |
| $H_l(T)$ | Thermal loss coefficient (W/ (m$^2$ °C)) |
| $H_t(T)$ | Convective heat transfer coefficient (W/ (m$^2$ °C)) |
| $K_{opt}$ | Optical efficiency (–) |
| $L$ | Tube perimeter (m) |
| $n_o(t)$ | Geometric efficiency (–) |
| $q(t)$ | Flow rate (m$^3$/s) |
| $S$ | Total area of the field (m$^2$) |
| $t$ | Time (s) |
| $T(t, x)$ | Temperature (°C) |
| $x$ | Space (m) |

**Subscripts**

| | |
|---|---|
| $a$ | Ambient |
| f | Fluid |
| in | Input |
| m | Metal |
| mean | Mean between input and output |
| ref | Reference |

development of many different neural network architectures. Notably, recurrent neural networks are models that can capture unknown dynamics well, although their peculiarity of incorporating hidden states makes them much more challenging to train. While stochastic gradient descent (SGD) [15] is the most commonly used algorithm, alternative approaches offer different advantages. In this study, an RNN is trained using the Extended Kalman Filter (EKF) due to its faster performance, as demonstrated by Trebatický and Pospíchal [16] and by Bemporad [17] for nonlinear systems within the framework of MPC.

Solar energy has garnered substantial attention due to the growing interest in renewable energy sources [18]. This research delves explicitly into the realm of thermal solar plants, focusing on parabolic trough collectors (PTCs). These collectors reflect solar radiation to heat a fluid to produce thermal energy. The application of FDD techniques to thermal solar plants is a broad research topic to be explored, with most applications in the existing literature focusing on water systems and small-scale plants. Notably, most applications center around fault detection or isolation rather than identifying collector module parameters. For example, the work by Schmelzer et al. [19] detects faults in a combi system by analyzing fractional solar consumption, and Brenner et al. [20] estimate mirror soiling in a PTC plant using feedforward neural networks. A related application of machine learning is presented in [21], where information from the defocusing mechanism of the plant feeds into an ANN classifier that detects and categorizes faults in the collector area. Despite these efforts, there remains a need for focused exploration into fault identification of collector module parameters in thermal solar plants. Most of the research in FDD is focused on systems with many outputs and subsystems. In the case of PTCs, the only output is the temperature, and the faults are highly correlated, which makes them difficult to isolate. This work uses neural networks to distinguish faults when the plant models cannot by including them in a new fault detection and reconstruction methodology. Specifically, RNNs are helpful to model the complex internal dynamics of the system. An EKF was chosen to train the RNN and estimate faults with the RNN trained because of its fastness and ability to reject measurements with significant errors.

The methodology proposed in this work consists of two steps: First, an RNN is trained using an EKF to model the dynamics of a PTC system, taking into account several faults introduced into the plant, benefiting from the fact that RNNs are time-aware and take into account the activations from previous data. In this step, the EKF helps capture the system dynamics and adapt the RNN to fault conditions. In this type of plants, the faults are highly coupled and different faults can produce the same changes in the output. For this reason, three EKFs governed by a feedforward neural network are applied in parallel, each dedicated to diagnosing a single fault. These filters utilize the previously trained RNN model to estimate the values of the faults present in the PTC system. The flowcharts in Fig. 1 describe the complete process. To the best of the authors' knowledge, this is the first time a combination of EKF and RNN is applied to PTC plants and fault reconstruction. This work proposes a new FDD methodology applied to solar plants using RNNs and EKFs and is not published elsewhere. The primary contributions of this research are outlined as follows:

- Use of an EKF-trained RNN to model the outlet and intermediate temperatures in one collector loop of a PTC plant with batch learning and taking into account fault parameters as inputs to the model.
- Fault estimation achieved through parameter reconstruction with EKF in the optical efficiency, flow rate, and thermal losses.
- Fault classification with an ANN governing a set of EKFs and selection of inputs based on output estimation error.

The remainder of this paper is as follows. Section 2 presents an overview of the system. The training of the neural network and the FDD process are described in Sections 3 and 4, respectively. Section 5 describes the evaluation metrics used in this paper. Some simulation results, focusing on the estimation and classification errors, are presented in Section 6. Finally, Section 7 elaborates on the principal findings of the study and draws conclusions.

## 2. System description

A PTC plant is a type of solar thermal system that comprises loops of parabolic mirrors designed to concentrate solar rays onto a focal line through which a heat transfer fluid (HTF) circulates. This fluid, typically water or oil, is heated to generate thermal energy. The heated HTF is then usually transported to a steam generator to drive a turbine, as illustrated in Fig. 2 [22].

The loops of a PTC system are described by the distributed parameter model, which accounts for the energy balances in the tubes and the HTF as follows [23]:

$$\rho_m C_m A_m \frac{\partial T_m}{\partial t} = \alpha_{K_{opt}} I K_{opt} n_o G - L H_t (T_m - T_f) - \alpha_{H_l} H_l G (T_m - T_a) \quad (1)$$

$$\rho_f C_f A_f \frac{\partial T_f}{\partial t} + \alpha_q \rho_f C_f q \frac{\partial T_f}{\partial x} = L H_t (T_m - T_f) \quad (2)$$

The model has been adapted to incorporate the parameters $\alpha$ representing the faults introduced into the system. These faults include optical efficiency faults $\alpha_{K_{opt}}$, flow rate faults $\alpha_q$, and thermal loss faults $\alpha_{H_l}$ [24]. It is presupposed that the metal's temperature remains constant radially, and the dimensions of the reflector and receiver remain the same along the loop, except for the passive parts (those not exposed to solar radiation), resulting in a uniform local concentration ratio.

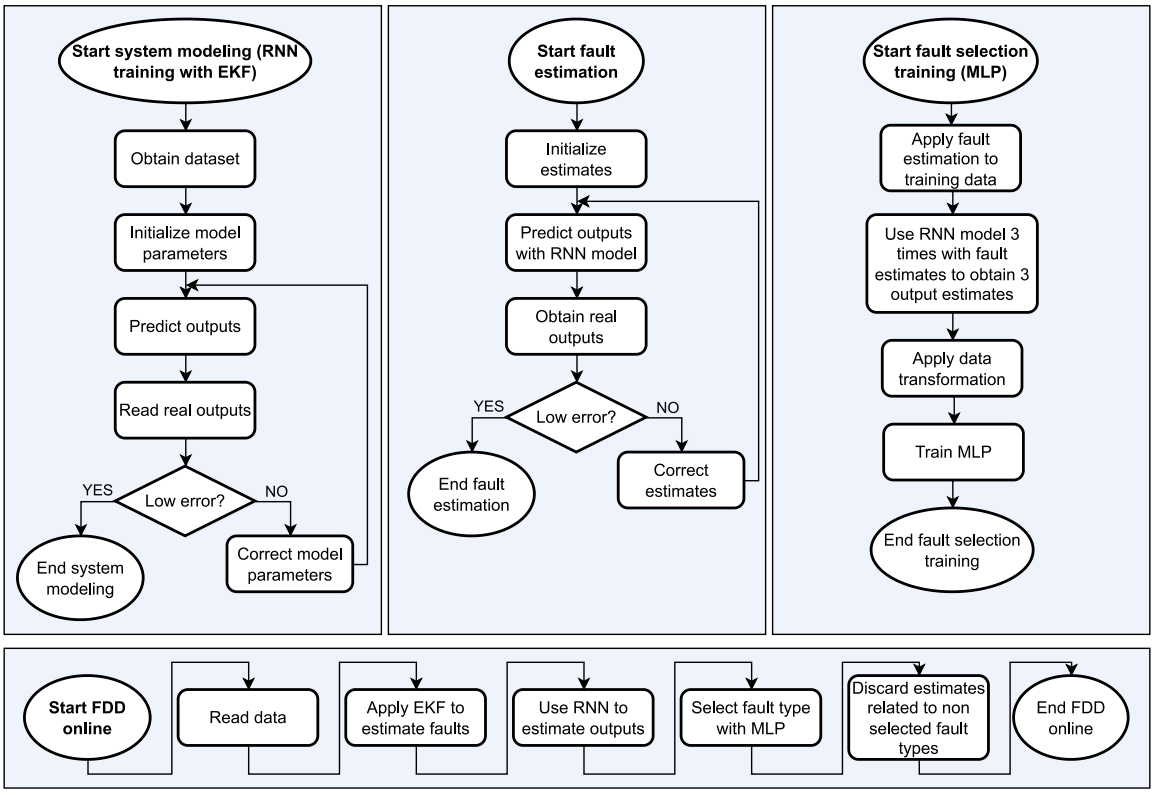



**Fig. 1.** Flowchart of the proposed methodology.

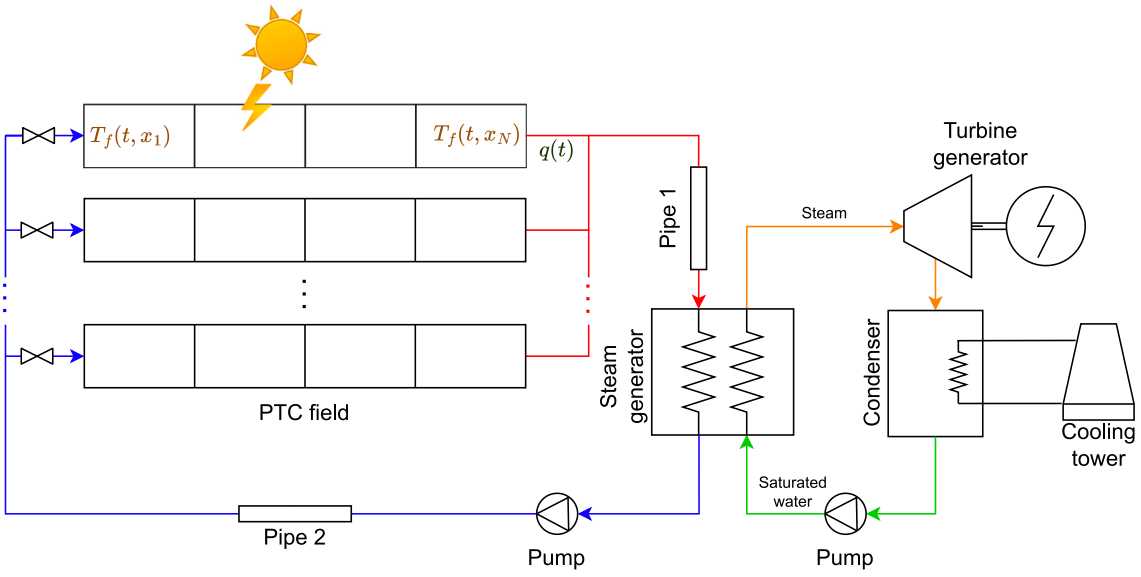



**Fig. 2.** General scheme of a PTC plant.

### 2.1. Faults considered

The three types of faults considered in this work, located in the collector area, are described as follows:

- Faults in the optical efficiency, denoted as $\alpha_{K_{\text{opt}}} < 1$. Since optical efficiency is influenced by tube absorptance, reflectivity, interception factor, and soiling of the reflectors, these faults encompass coating, breakage, tube deterioration, mirror defects, dirt, degradation, and corrosion.
- Faults in the flow rate, expressed as $\alpha_q \neq 1$, are related to flowmeter failures and loop imbalances.
- Faults in the thermal loss, indicated by $\alpha_{H_l} > 1$, are linked to pressure drop in the tubes resulting from wear, insulation, dirt accumulation, and pipe breakage.

The parameters represent deviations from the nominal values and signify various issues that can affect the performance of the PTCs, although these faults only affect the model of the loop and not the rest of the subsystems. To adapt this methodology to a real plant, three paths

**Fig. 3.** ACUREX collector loops.

**Table 1**
Parameters of the ACUREX plant.

| Parameter | Value |
|---|---|
| $\rho_m$ | 7800 kg/m$^3$ |
| $C_m$ | 550 J/kg°C |
| $A_m$ | $2.4806 \cdot 10^{-4}$ m$^2$ |
| $G$ | 1.82 m |
| $L$ | $7.98 \cdot 10^{-2}$ m |
| $A_f$ | $5.0671 \cdot 10^{-4}$ m$^2$ |
| $S$ | 2672 m$^2$ |

could be taken: first, using a digital twin that accurately reproduces the system and from which to take the data for training the neural networks. Secondly, one can provoke these faults in the actual plant in an instrumented loop to gather the necessary data (by partially covering the mirrors, modifying the insulation of the pipes, or mismatching the flow rate). Finally, this information can be gathered from large-scale plants where knowledge of failures is acquired retrospectively as analysis and maintenance tasks are performed periodically. In this case, the advantage of the proposed methodology would be to detect these failures with greater immediacy.

### 2.2. Case study

The methodology was tested in simulation on one loop of ACUREX [25], an experimental, 1-MW PTC plant that was situated at the Plataforma Solar de Almería. Fig. 3 depicts some of the collectors within ACUREX. The plant is comprised of 10 loops of single-axis collectors aligned in an east–west orientation, with each loop containing 12 modules grouped into 4 collectors. The loops are composed of an active part of 142 m and a passive part of 30 m. ACUREX serves as a practical and representative case for evaluating the proposed methodology.

Table 1 contains the parameters of the ACUREX plant, and the thermal loss coefficient $H_l$ and convective thermal exchange within the inner tube $H_t$ are given by, respectively [26]:

$$H_l = 0.00249\left(T_f - T_a\right) - 0.06133 \tag{3}$$

$$H_t = q^{0.8}(2.17 \cdot 10^6 - 5.01 \cdot 10^4 T_f + 4.53 \cdot 10^2 T_f^2 - 1.64 T_f^3 + 2.1 \cdot 10^{-3} T_f^4) \tag{4}$$

The geometric efficiency [27,28] is derived from the correlation between the radiation beam's direction and the perpendicular vector of the mirror. As the sun is tracked in elevation, the geometric efficiency is calculated using the following equation [29]:

$$n_o = \left(1 - \cos^2(\delta_s)\sin^2(\omega_s)\right)^{\frac{1}{2}} \tag{5}$$

The plant is equipped with a sun tracking system that precisely controls the mirror rotation around an axis aligned parallel to the pipe, enhancing the geometric efficiency for solar radiation capture and utilization [26].

The HTF used in the system is Therminol 55 thermal oil. The specific heat capacity $C_f$ and density $\rho_f$ of the HTF characterize its thermophysicial properties and are, respectively:

$$C_f = 3.478 T_f + 1820 \tag{6}$$

$$\rho_f = -0.672 T_f + 903 \tag{7}$$

The loop is discretized longitudinally into 172 segments, each with a length of 1 m. The model is computed with an integration step of 0.25 s. Initially, the metal temperature is calculated. Subsequently, the fluid temperature is obtained under the assumption of steady-state conditions. Finally, the fluid temperature is corrected by considering the energy transfer between each fluid control volume. For further details, the reader is referred to [26].

### 2.3. Flow-rate controller

The flow rate is manipulated using valves to control the outlet temperature and track a reference. The concentrated parameter model of the plant, which describes the variation of the internal energy of the fluid in steady state, is used to implement the following feedforward controller:

$$q = \frac{n_o K_{opt} S I - H_l A (T_{mean} - T_a)}{P_{cp}(T_{ref} - T_{in})} \tag{8}$$

where $P_{cp} = \rho_m C_m$.

The flow rate is constrained within the range $[0.2, 1.2]$ l/s and the controller sampling time is 39 s. This control strategy effectively adjusts the flow rate per loop to attain the target outlet temperature, considering the objectives of this work. It is important to note that the primary focus of this study is not on optimizing control actions but rather on implementing an FDD strategy, so no other control technique is applied. In practice, an advanced control technique like MPC would be used. It is not relevant for this work because the ranges in which the control signal varies are similar for both controllers.

## 3. Recurrent neural network

A recurrent neural network is a type of ANN that incorporates delayed feedback loops in its layers [30]. Similar to multilayer perceptrons, RNNs comprise an input layer, an output layer, and one or more hidden layers. Each hidden layer contains a varying number of neurons with an activation function that produces nonlinear effects. RNNs are time-aware and can capture the system dynamics by considering previous data in the sequence as inputs. They can be described by the state-space model:

$$\begin{aligned} x(k+1) &= f_x(x(k), u(k), \theta_x) \\ \hat{z}(k) &= f_z(x(k), u(k), \theta_z) \end{aligned} \tag{9}$$

where $u \in \mathbb{R}^{n_u}$ represents the input, $\hat{z} \in \mathbb{R}^{n_z}$ is the predicted output, $x \in \mathbb{R}^{n_x}$ is the state vector and $\theta_x \in \mathbb{R}^{n_{\theta_x}}$, and $\theta_z \in \mathbb{R}^{n_{\theta_z}}$ are the trainable model parameters.

More explicity, as done in [17], Eq. (9) can be expressed by:

$$\begin{cases} v_1^x(k) = A_1^x \begin{bmatrix} v_{L_x}^x(k-1) \\ u(k) \end{bmatrix} + b_1^x \\ v_2^x(k) = A_2^x f_1^x(v_1^x(k)) + b_2^x \\ \quad \vdots \\ v_{L_x}^x(k) = A_{L_x}^x f_{L_x-1}^x(v_{L_x-1}^x(k)) + b_{L_x}^x \\ x(k+1) = v_{L_x}^x(k) \end{cases} \tag{10}$$

$$\begin{cases} v_1^z(k) = A_1^z \begin{bmatrix} v_{L_x}^x(k) \\ u(k) \end{bmatrix} + b_1^z \\ v_2^z(k) = A_2^z f_1^z(v_1^z(k)) + b_2^z \\ \quad \vdots \\ v_{L_z}^z(k) = A_{L_z}^z f_{L_z-1}^z(v_{L_z-1}^z(k)) + b_{L_z}^z \\ \hat{z}(k) = v_{L_z}^z(k) \end{cases} \tag{11}$$

where $\theta_x = \left(A_1^x, b_1^x, \ldots, A_{L_x}^x, b_{L_x}^x\right)$ and $\theta_z = \left(A_1^z, b_1^z, \ldots, A_{L_z}^z, b_{L_z}^z\right)$, $L_x$ and $L_z$ are the number of hidden layers in the state-update and output functions, $v_i^x \in \mathbb{R}^{n_i^x}$ and $v_i^z \in \mathbb{R}^{n_i^z}$ are values associated with the neurons, $f_i^x : \mathbb{R}^{n_i^x} \to \mathbb{R}^{n_{i+1}^x}$ and $f_i^z : \mathbb{R}^{n_i^z} \to \mathbb{R}^{n_{i+1}^z}$ are the activation functions, $A_i^x \in \mathbb{R}^{n_i^x \times n_{i-1}^x}$ and $A_i^z \in \mathbb{R}^{n_i^z \times n_{i-1}^z}$ are the weight matrices, and $b_i^x \in \mathbb{R}^{n_i^x}$ and $b_i^z \in \mathbb{R}^{n_i^z}$ are the bias terms.

### 3.1. Training with EKF

Typically, the parameters of ANNs are updated using gradient descent (GD) methods [31,32]. However, in this work, the supervised training of the RNN is performed using the EKF algorithm to achieve much faster and less expensive computations, as emphasized by Bemporad [17].

For the specific case of modeling a system with faults $\alpha$, the input vector is extended to include them, treating them as inputs to the system. To obtain training and test data, these faults must be intentionally introduced and known to the training algorithm. The model can be rewritten in the form of:

$$\begin{aligned} x(k+1) &= f_x(x(k), \tilde{u}(k), \theta_x(k)) + \eta_x(k) \\ \hat{z}(k) &= f_z(x(k), \tilde{u}, \theta_z) + \eta_z(k) \\ \alpha(k+1) &= \alpha(k) + \eta_\alpha(k) \\ \theta(k+1) &= \theta(k) + \eta_\theta(k) \end{aligned} \tag{12}$$

where $\tilde{u}(k) = [u(k), \alpha(k)]^T$ and $\theta(k) = [\theta_x(k), \theta_z(k)]^T$ and $\eta_x(k) \in \mathbb{R}^{n_x}$, $\eta_z(k) \in \mathbb{R}^{n_z}$, $\eta_\alpha(k) \in \mathbb{R}^{n_\alpha}$ and $\eta_\theta(k) \in \mathbb{R}^{n_\theta}$ are white noise vectors. Since noise estimation [33,34] is not part of the scope of this work, noises are assumed to be white. The covariance matrices are $Q_x(k)$, $Q_z(k)$, $Q_\alpha(k)$ and $Q_\theta(k)$, respectively.

The state vector $\hat{x}(k)$ and the RNN coefficients $\hat{\theta}(k)$ are estimated with the EKF updates [35] by augmenting the state vector with the parameter vector using the RNN model of Eqs. (10) and (11) as the transition and observation models $f_x$ and $f_z$ and computing the derivatives at each step, in accordance with the following:

$$H(k) = \left[\begin{matrix} \frac{\partial f_z}{\partial x} & 0 & \frac{\partial f_z}{\partial \theta_z} \end{matrix}\right]\Bigg|_{\hat{\theta}(k|k-1), \hat{x}(k|x-1), \tilde{u}(k)} \tag{13a}$$

$$K(k) = P(k|k-1)H(k)^T[H(k)P(k|k-1)H(k)^T + Q_z(k)]^{-1} \tag{13b}$$

$$e(k) = z(k) - f_z(\hat{x}(k|k-1), \tilde{u}(k), \hat{\theta}_z(k|k-1)) \tag{13c}$$

$$\begin{bmatrix} \hat{x}(k|k) \\ \hat{\theta}(k|k) \end{bmatrix} = \begin{bmatrix} \hat{x}(k|k-1) \\ \hat{\theta}(k|k-1) \end{bmatrix} + K(k)e(k) \tag{13d}$$

$$P(k|k) = (I - K(k)H(k))P(k|k-1) \tag{13e}$$

$$\begin{bmatrix} \hat{x}(k+1|k) \\ \hat{\theta}(k+1|k) \end{bmatrix} = \begin{bmatrix} f_x(\hat{x}(k|k), \tilde{u}(k), \hat{\theta}_x(k|k)) \\ \hat{\theta}(k|k) \end{bmatrix} \tag{13f}$$

$$A(k) = \begin{bmatrix} \frac{\partial f_x}{\partial x} & \frac{\partial f_x}{\partial \theta_x} & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix}\Bigg|_{\hat{\theta}(k|k), \hat{x}(k|k), \tilde{u}(k)} \tag{13g}$$

$$P(k+1|k) = A(k)P(k|k)A(k)^T + \begin{bmatrix} Q_x(k) & 0 \\ 0 & Q_\theta(k) \end{bmatrix} \tag{13h}$$

The initial covariance matrix $P(0|-1)$ is chosen to incorporate $\ell_2$-regularization by analogy with Newton's method as stated by [17]:

$$P(0|-1) = \begin{bmatrix} \frac{1}{N\rho_x} I & 0 \\ 0 & \frac{1}{N\rho_\theta} \end{bmatrix} \tag{14}$$

where $N$ is the number of instances in the training set and $\rho_x$ and $\rho_\theta$ are the penalizing terms on $\frac{1}{2}\|x\|_2^2$ and $\frac{1}{2}\|\theta\|_2^2$.

The EKF is applied by processing the data for several epochs, each one corresponding to a one-day simulation. This process involves comparing the estimation error of the RNN and the correct output on a test set. The stopping criteria are selected to ensure that the number of epochs falls within a specified range, and the mean squared error (MSE) on the validation set is neither constant nor increasing for a certain number of epochs. To recover the initial state and covariance matrix after each epoch, the extended Rauch-Tungg-Striebel smoother was applied [36]. This smoother first applies the EKF to the new data using the previous state and covariance matrix and then performs smoothing.

For the present problem, the input vector contains the inputs to the concentrated parameter model, so $u = (T_{\text{in}}, T_a, I \cdot n_o, q)^T$, the outputs are the temperatures measured at the central point of every collector, the outlet temperature as $z = (T_1, T_2, T_3, T_4, T_{\text{out}})^T$, and the fault vector contains the three aforementioned parameters $\alpha = (\alpha_{K_{\text{opt}}}, \alpha_q, \alpha_{H_1})^T$.

## 4. Fault detection and estimation

Once the system has been modeled, a second EKF is applied online to identify the values of the faults using the RNN trained as described in Section 3. In this case, the parameters $\theta$ of the model in Eq. (12) are fixed, while the inputs $\alpha$ are estimated. Since the three fault parameters are highly correlated, three parallel EKFs are applied, one for each type of fault, using the RNN model of Eqs. (10) and (11) as the transition and observation models $f_x$ and $f_z$. For each fault $\alpha_i \in \alpha$, the new EKF process is given by:

$$H(k) = \left[\begin{matrix} \frac{\partial f_z}{\partial x} & \frac{\partial f_z}{\partial \alpha} \end{matrix}\right]\Bigg|_{\hat{\alpha}_i(k|k-1), \hat{x}(k|x-1), \tilde{u}(k)} \tag{15a}$$

$$K(k) = P(k|k-1)H(k)^T[H(k)P(k|k-1)H(k)^T + Q_z(k)]^{-1} \tag{15b}$$

$$e(k) = z(k) - f_z(\hat{x}(k|k-1), \tilde{u}(k), \theta_z(k|k-1)) \tag{15c}$$

$$\begin{bmatrix} \hat{x}(k|k) \\ \hat{\alpha}_i(k|k) \end{bmatrix} = \begin{bmatrix} \hat{x}(k|k-1) \\ \hat{\alpha}_i(k|k-1) \end{bmatrix} + K(k)e(k) \tag{15d}$$

$$P(k|k) = (I - K(k)H(k))P(k|k-1) \tag{15e}$$

$$\begin{bmatrix} \hat{x}(k+1|k) \\ \hat{\alpha}_i(k+1|k) \end{bmatrix} = \begin{bmatrix} f_x(\hat{x}(k|k), \tilde{u}(k), \theta_x(k|k)) \\ \hat{\alpha}_i(k|k) \end{bmatrix} \tag{15f}$$

$$A(k) = \begin{bmatrix} \frac{\partial f_x}{\partial x} & \frac{\partial f_x}{\partial \alpha_i} \\ 0 & I \end{bmatrix}\Bigg|_{\theta(k|k), \hat{x}(k|k), \tilde{u}(k)} \tag{15g}$$

$$P(k+1|k) = A(k)P(k|k)A(k)^T + \begin{bmatrix} Q_x(k) & 0 \\ 0 & Q_\alpha(k) \end{bmatrix} \tag{15h}$$

During plant operation, three estimations are made, each assuming that the rest of the fault parameters are set to 1 (i.e., no more faults). In this context, to decide which is the correct one, each estimate $\alpha_i(k)$ is used to feed the neural network, resulting in different output vector estimates $\hat{z}_i(k)$ for every fault $i \in \{K_{\text{opt}}, q, H_1\}$. For instance, $\hat{x}^{K_{\text{opt}}}(k)$ and $\hat{z}^{K_{\text{opt}}}(k)$ are the estimated state and output, assuming there is only a fault in the optical efficiency $K_{\text{opt}}$. The entire process is illustrated in Fig. 4. This iterative estimation process occurs during the dynamic operation of the plant.

### 4.1. Fault selection with neural network

In order to select which of the three estimates is the correct one, a fault detection phase is implemented based on the previous estimates using the EKF. For this purpose, we trained a second feedforward neural network classifier. It is constructed on the basic structure of the multilayer perceptron (MLP) with input-transformation blocks added at the input, as depicted in Fig. 5. The output of the ANN represents the likelihood of each fault and the case without fault in a vector $\hat{y}(k) \in \mathbb{R}^{n_\alpha+1}$. In this case, $\hat{y}(k) = (\hat{y}_{\text{faultless}}(k), \hat{y}_{K_{\text{opt}}}(k), \hat{y}_q(k), \hat{y}_{H_1}(k))$.

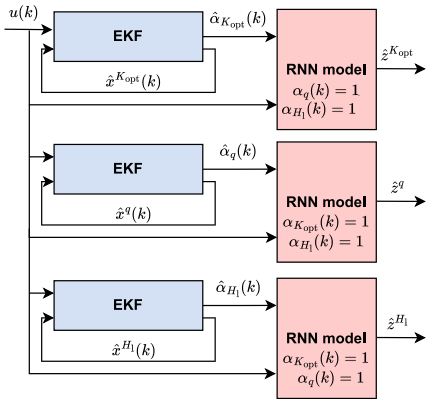



**Fig. 4.** Scheme of the fault parameters and output estimates. An EKF is applied three times to independently estimate the faults and feed the RNN model, assuming the remaining faults are 1. This is performed to obtain the different output estimates that will be used to isolate the correct fault and discard the rest.

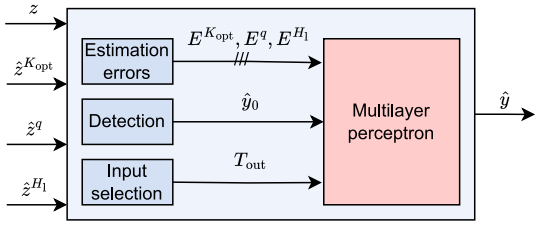



**Fig. 5.** Scheme of the fault detection phase.

This fault detection phase aids in determining the most plausible fault scenario based on the estimates generated during plant operations.

The first block obtains the errors $E^{K_{\text{opt}}}$, $E^q$ and $E^{H_1}$ over the actual outputs as $E^i(k) = \hat{z}^i(k)/z(k)$, where $E^i(k) \in \mathbb{R}^{n_z}$. The second block performs a preliminary fault detection by selecting the type of fault corresponding to the minimum estimation error $E^i(k)$ in the vector $\hat{y}_0 \in \mathbb{R}^{n_\alpha+1}$. If the estimated fault is approximately 1 (by 5%), it is considered that there is no fault. The third block selects the system outputs that will be used as inputs to the MLP. For this problem, only the outlet temperature $T_{\text{out}}$ was considered. The ANN underwent training utilizing the scaled conjugate gradient backpropagation algorithm [37].

## 5. Evaluation

The methodology is evaluated according to (1) fault detection and isolation capabilities and (2) fault reconstruction capabilities. For the first purpose, the classification accuracy and F1-scores are computed as:

$$Acc = \frac{TN + TP}{TP + FN + TN + FP} \tag{16}$$

$$F1 = 2 \cdot \frac{Rec \cdot Pre}{Rec + Pre}, \text{ where } \begin{cases} Pre &= \frac{TP}{FP+TP} \\ Rec &= \frac{TP}{FN+TP} \end{cases} \tag{17}$$

where TP is the correct identification of faults, TN is the correct identification of the absence of faults, FP is the incorrect identification of faults, and FN is the failure to identify actual faults.

The fault reconstruction effectiveness is evaluated with the average estimation error $E_i^i(k)$ obtained for the actual fault $i$ with its corresponding EKF, as described in Section 4.1.

## 6. Simulation results

The methodology was tested by simulating the ACUREX plant. For this purpose, two datasets were created: one obtained from synthetic irradiances representing sunny days and the other consisting of real irradiance profiles obtained from the Plataforma Solar de Almería data with different types of clouds. All computations were performed in MATLAB R2020b with Intel® Core™ i7-9700F CPU at 3 GHz and 16 GB RAM using CasADi [38] for automatic differentiation.

### 6.1. Sunny dataset

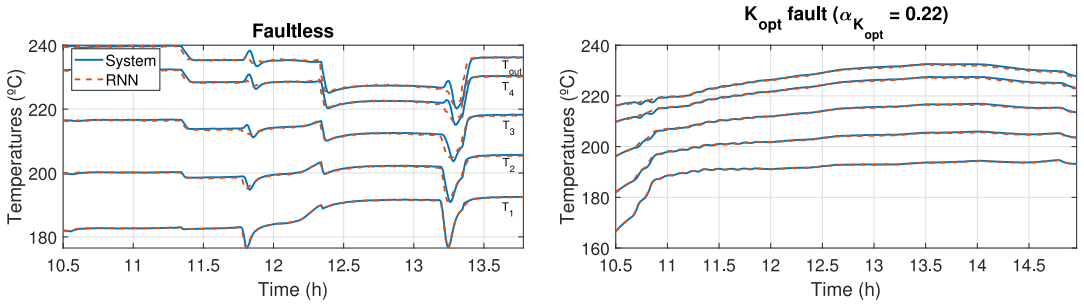
To create the dataset, 450 single-day simulations were run, collecting data every 39 s with random inputs, disturbances, and fault values. All the simulations utilized synthetic irradiance profiles of one day of various random shapes, assuming sunny days. The peak irradiances of these profiles ranged from 750 W/m$^2$ to 1000 W/m$^2$. The reference temperatures varied from 200 °C to 300 °C, with steps up to 15 °C every hour. The fault values were fixed throughout the day and selected so that only one type of fault co-occurred. Specifically, faults in the optical efficiency ranged from 0.1 to 0.9, faults in the flow rate were from 0.5 to 1.5, and faults in the thermal losses ranged from 1.1 to 1.2. At the end of the day, the fault was restored to its nominal value and a new time series began with a new fault, a new reference temperature, and a new irradiance profile. With this, it is assumed that the system is not working during shutdowns of the plant, and an operator will fix the failure. The dataset was divided into training, validation, and test sets of 320, 80, and 50 instances, respectively.

The learning process was carried out for the entire batch with a minimum of 15 epochs and a maximum of 5000 epochs, stopping training when the error remained constant or increased for four consecutive steps. The activation functions employed were hyperbolic tangent across all layers, excluding the final layer, which utilizes a linear function. The rest of the RNN parameters and hyperparameters were selected in a process of trial and error by comparing the mean squared error on the validation set, and the data were normalized. The selected RNN has $n_x = 2$, $L_x = 2$ with 20 and 10 neurons in each layer, and directly a linear connection for the output with $L_y = 1$ and one neuron. Additionally, $Q_x(k) = 0.1I$, $Q_z(k) = 100I$, and $\rho_x = \rho_\theta = 0.001$.

Fig. 6 shows the results of four random, closed-loop experiments of the validation set with the selected RNN architecture, one corresponding to each type of fault. This neural network was trained with 22 epochs in 32.5890 min, achieving average MSEs of $2.0638 \cdot 10^{-3}$ on the training set, $1.6604 \cdot 10^{-3}$ on the validation set and $8.2602 \cdot 10^{-4}$ on the test set. The graphs show the good adaptation of the neural network in its five outputs with minimal deviations, even when fluctuations occur. This confirms the behavior of this RNN as a model of the plant.

Once the neural network was trained, the second EKF was applied with the components of $P(0|-1) = diag(10^2, 10^2, 10^{-2}, 10^{-2}, 10^{-2})$, $Q_x = 10^{-8}I$, $Q_\alpha = diag(10^{-3}, 10^{-3}, 10^{-3})$ and $Q_z = 10^{-4}$. The estimates were smoothed with a low-pass filter of 15 min before being passed to the feedforward neural network. Fig. 7 shows an example of the three estimates in a case with a fault of $\alpha_q = 0.625$. Judging from the graph, the estimates only approached the real value for the flow rate estimator, as one would expect, with a small error. The estimates obtained considering only the optical efficiency and thermal losses are affected by the flow rate fault, producing wrong values and evidencing the importance of a decoupling strategy. Since the faults are strongly coupled, the next step needed is to isolate the faults and select the correct estimate, discarding the other two. It is also important to notice that the perturbations in the temperature do not affect the EKF estimations. Regarding time consumption, the mean time of each iteration of the EKF was 0.013 s.

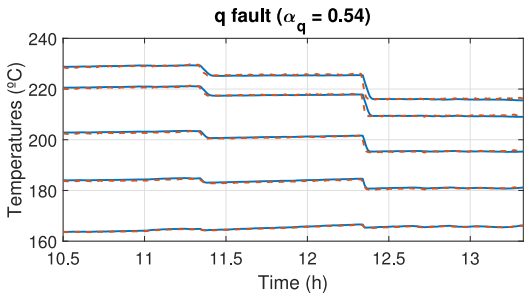
The second ANN was trained with the following hyperparameter values: $\lambda = 5 \cdot 10^{-7}$ that regulates the indefiniteness of the Hessian matrix, $\sigma = 5 \cdot 10^{-5}$ that determines changes in the weighting of the second derivative approximation, a maximum number of epochs of $4 \cdot 10^3$, a minimum gradient of $10^{-6}$ and a maximum of 6 validation checks. All layers have hyperbolic tangent, except for a softmax function in the last layer. The ANN consists of three hidden layers, with



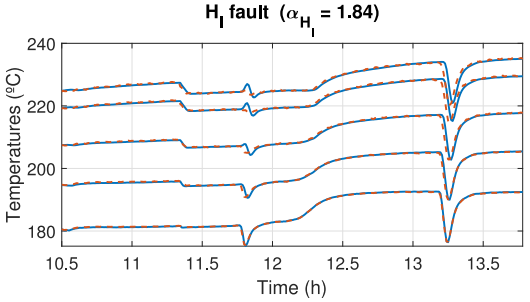

(a) Day without faults.




(b) Day with a fault in the optical efficiency.

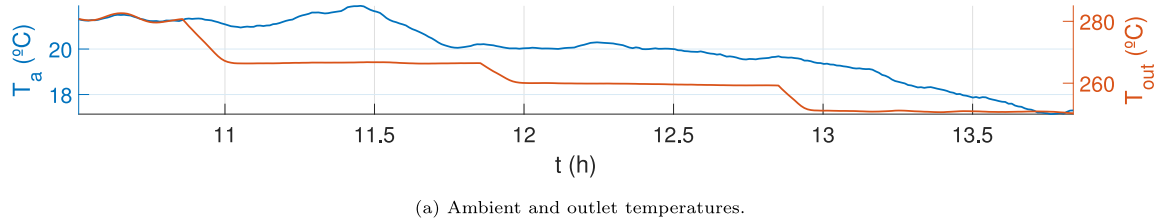



(c) Day with a fault in the flow rate.

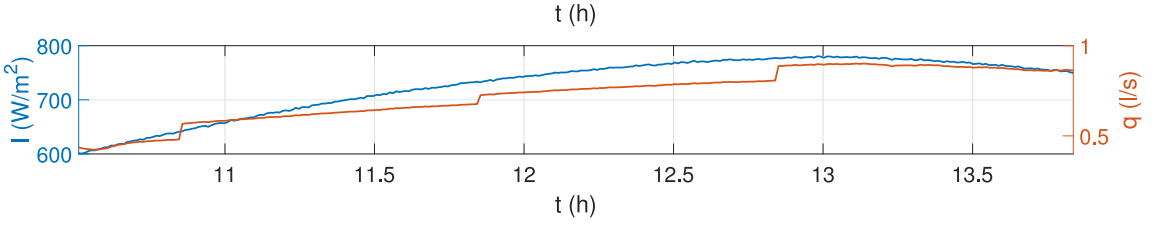



(d) Day with a fault in the thermal losses.

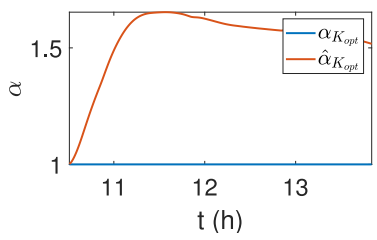
**Fig. 6.** Modeling results of the RNN on four of the experiments of the sunny validation set. Outlet temperature and temperatures at the center of each collector obtained from the sensors and the RNN.



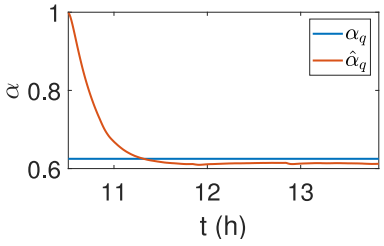

(a) Ambient and outlet temperatures.



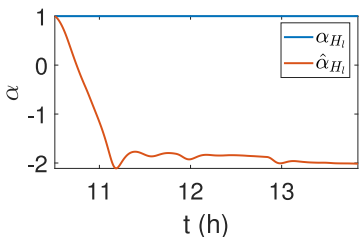

(b) Irradiance and flow rate.




(c) Estimates from the EKF assuming the fault is in the optical efficiency.



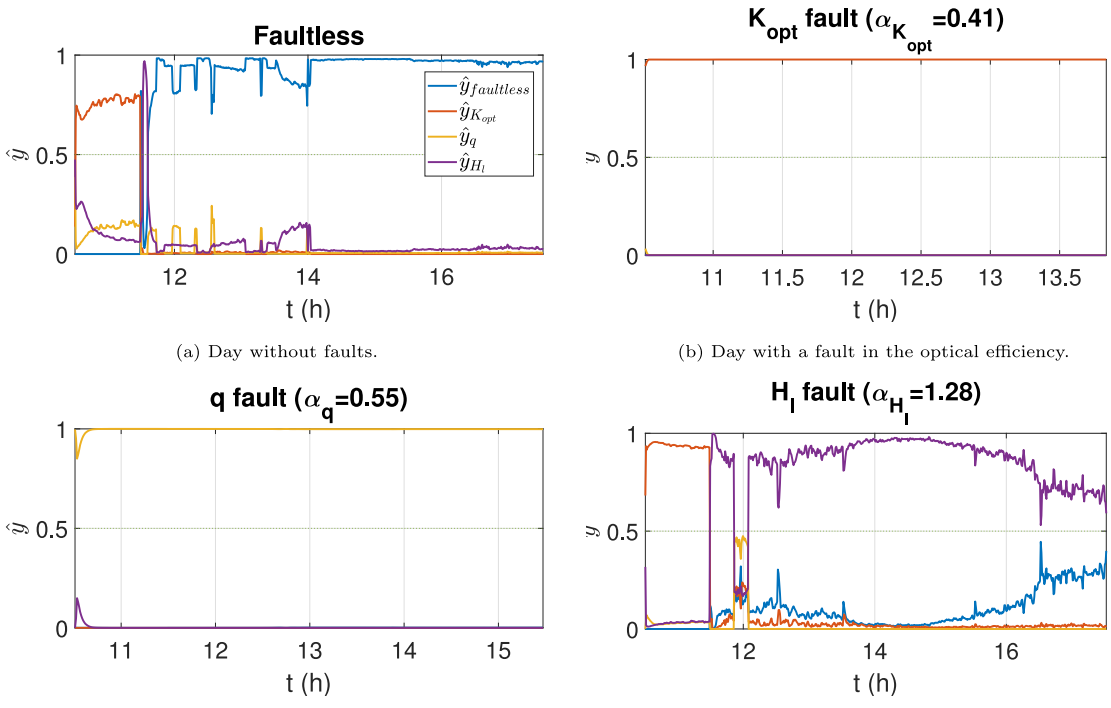

(d) Estimates from the EKF assuming the fault is in the flow rate.




(e) Estimates from the EKF assuming the fault is in the thermal losses.

**Fig. 7.** Simulation conditions and results of the parameter estimation with the three parallel EKFs in an experiment with $\alpha_q = 0.625$ of the sunny validation set.

200, 100, and 50 neurons in each one. Fig. 8 shows the outputs of the feedforward neural network in four different random experiments, each corresponding to a different faulty case. In these examples, the higher output corresponded to the correct type of fault most of the time, with




(a) Day without faults.

(b) Day with a fault in the optical efficiency.

(c) Day with a fault in the flow rate.

(d) Day with a fault in the thermal losses.

**Fig. 8.** Outputs of the feedforward neural network on four experiments of the sunny validation set.

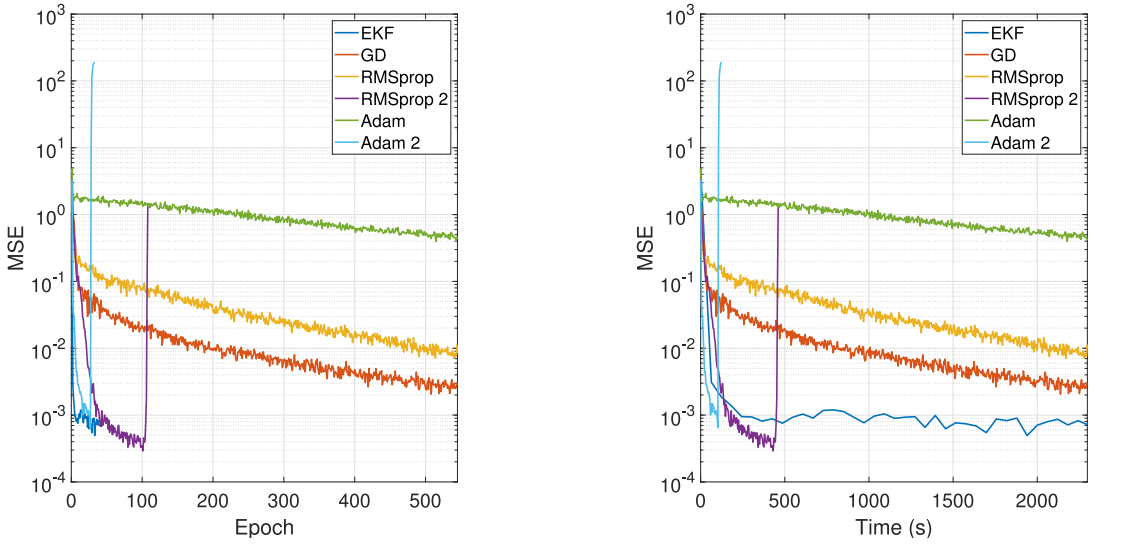
**Table 2**
Evaluation metrics of the methodology on the training, validation and test sets.

| | Without feedforward ANN | | | With feedforward ANN | | |
|---|---|---|---|---|---|---|
| | Training | Validation | Test | Training | Validation | Test |
| Average estimation error in the faulty parameter (%) | 2.07 | 2.37 | 2.77 | 2.07 | 2.37 | 2.77 |
| F1 in the faultless case (%) | 94.55 | 97.30 | 96.08 | 97.01 | 100.00 | 98.04 |
| F1 in the $K_{opt}$ fault (%) | 77.60 | 85.19 | 78.18 | 87.80 | 84.62 | 81.62 |
| F1 in the $q$ fault (%) | 73.91 | 77.42 | 76.40 | 88.89 | 92.31 | 82.76 |
| F1 in the $H_l$ fault (%) | 89.61 | 94.74 | 92.93 | 94.87 | 83.87 | 91.09 |
| Classification accuracy (%) | 84.06 | 88.75 | 86.00 | 92.19 | 90.00 | 88.50 |

some errors at the beginning of the day. To avoid these errors, alarms were not triggered immediately; instead, they were activated after data had been recorded during a specific time window. In this case, this activation was performed at the end of the day. In the four cases, the faults were correctly classified. There was a small degree of doubt in the faultless day and the day with a fault in $H_l$, but the neural network's output was able to distinguish the correct class effectively. The initial $K_{opt}$ classification during the first hour of the faultless day indicates the need not to trigger alarms immediately.

Table 2 summarizes some average estimation and classification errors computed for the training, validation, and test subsets at the end of each day, both using only the preliminary fault selection without feedforward neural network $\hat{y}_0$ and using the feedforward neural network $\hat{y}$. The estimation errors are computed for the actual fault type instead of the estimated fault type to allow a separate analysis between estimation and classification behavior. The accuracies and F1-scores improve with the feedforward ANN in all classes except for the thermal losses, although the accuracy is still higher with the ANN. The results show the good behavior of the FDD system, with accuracies and F1-scores around 90% and close to 100% in the faultless case. Both estimation errors are less than 3%, with even lower values in the test set than in the training set, which could be due to the randomness of the experiments. The results of the table also show the ability of the feedforward ANN to improve the classification accuracy around 2 points in validation and test subsets.

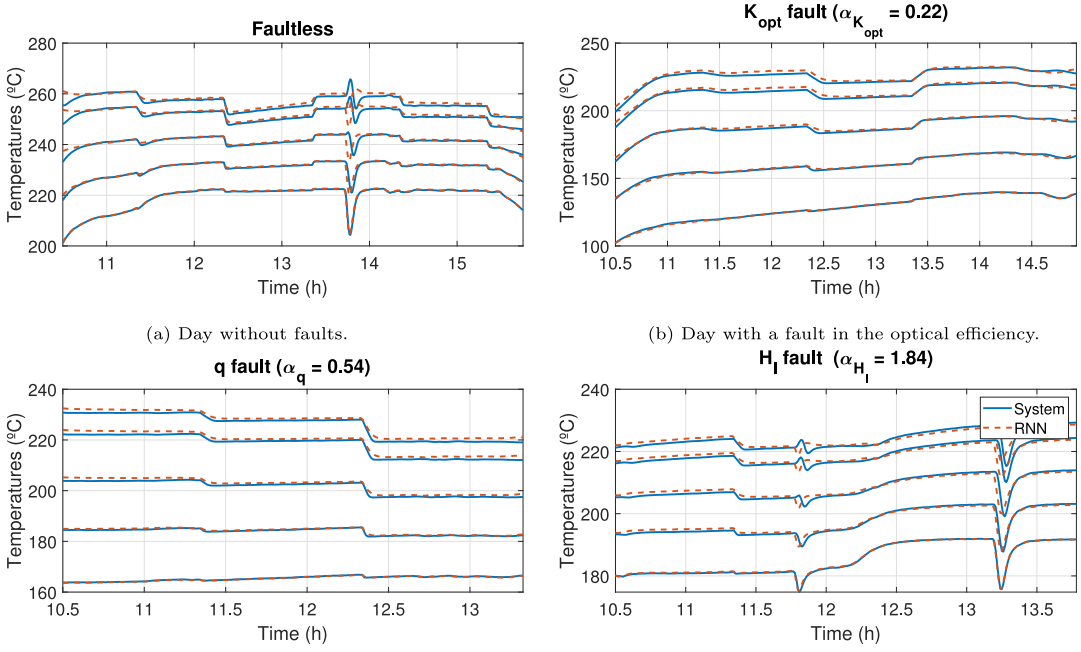
To prove that the training is faster with EKF than with GD, Fig. 9 shows the evolution of the training MSE with the number of epochs and the training time. The EKF algorithm is compared with (i) GD with a learning rate of $10^{-4}$, (ii) RMSprop with a learning rate of $10^{-5.2}$ and $\beta = 0.8$ with five initial steps of simple GD, (iii) RMSprop with a learning rate of $10^{-4}$ and $\beta = 0.9$, (iv) Adam with a learning rate of $10^{-6}$, $\beta_1 = 0.87$ and $\beta_2 = 0.65$, and (v) Adam with a learning rate of $10^{-3}$, $\beta_1 = 0.806$ and $\beta_2 = 0.999$ with five initial steps of gradient descent, all of them with gradient clipping. A random search was performed in a coarse-to-fine scheme to select the hyperparameters. It is worth noting that GD performs better than RMSprop and Adam. In the only experiments where RMSprop and Adam were faster than EKF and GD, the weights ended up unstabilizing, as one can see in the figure. This can be due to the nature of this specific problem, which does not benefit from adaptive learning rates provided by Adam and RMSprop. Although each epoch of the EKF is slower than GD, the number of epochs needed to obtain a low MSE and the total training time are much lower. A similar experiment was carried out by Bemporad [17] with Adam. There are several aspects that justify the fast training with EKF. First, gradient descent relies on first-order gradients, but EKF uses second-order information through a covariance matrix. This way, EKF can adapt its updates more intelligently based on the local curvature, allowing for more precise parameter updates and fewer iterations.




(a) Evolution of MSE with the number of epochs.

(b) Evolution of MSE with training time.

**Fig. 9.** Comparison of training with EKF and GD.




(a) Day without faults.

(b) Day with a fault in the optical efficiency.

(c) Day with a fault in the flow rate.

(d) Day with a fault in the thermal losses.

**Fig. 10.** Modeling results of the RNN on four of the experiments of the cloudy validation set. Outlet temperature and temperatures at the center of each collector obtained from the sensors and the RNN.

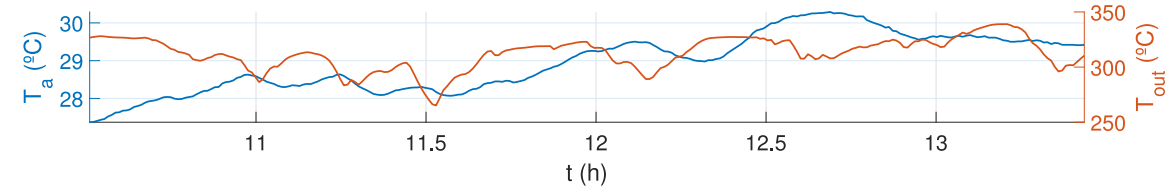
Moreover, the Kalman gain reduces the impact of noisy measurements, allowing a stable and rapid convergence.

### 6.2. Cloudy dataset

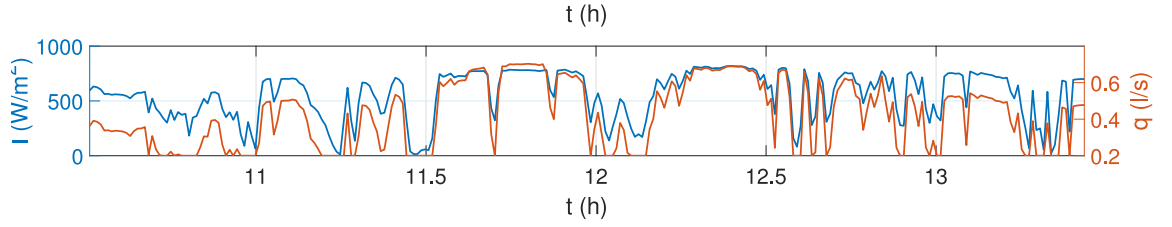
As done with the sunny dataset, 450 single-day simulations were run, collecting data every 39 s with random values of inputs, disturbances, and faults. These simulations were performed using irradiances from actual days when different types of clouds passed by. The reference temperatures and fault values were introduced under the same conditions as those used for the sunny dataset. The dataset was divided into training, validation and test sets of 320, 80, and 50, instances, respectively. Each subset was obtained from different irradiance profiles.

After trial and error, an RNN was trained using hyperbolic tangent and linear functions. The RNN has the same characteristics as the one used for the sunny dataset and was trained in 18 iterations and 27.1178 min. This neural network achieved average MSEs of $5.9542 \cdot 10^{-3}$ on the training set, $1.7679 \cdot 10^{-3}$ on the validation set and $3.6720 \cdot 10^{-3}$ on the test set. Fig. 10 shows the adaptation of the RNN to four experiments of the validation set. Again, the RNN performs satisfactorily and adapts well to the system output, although in this case the error is slightly greater than with the sunny day due to the complex dynamics introduced by the clouds.

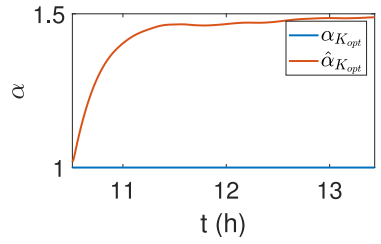
To estimate the fault parameters, the second EKF was applied with the components of $P(0|-1) = diag(10^2, 10^2, 10^{-2}, 10^{-2}, 10^{-2})$, $Q_x =$



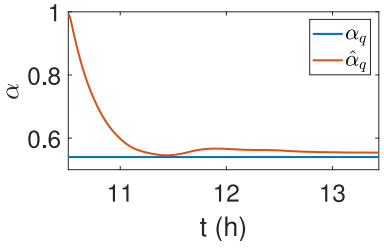

(a) Ambient and outlet temperatures.



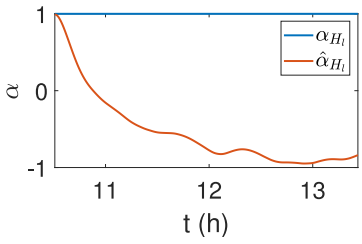

(b) Irradiance and flow rate.




(c) Estimates from the EKF assuming the fault is in the optical efficiency.




(d) Estimates from the EKF assuming the fault is in the flow rate.




(e) Estimates from the EKF assuming the fault is in the thermal losses.

**Fig. 11.** Simulation conditions and results of the parameter estimation with the three parallel EKFs in an experiment with $\alpha_q = 0.7$ of the cloudy validation set.

$10^{-8}I$, $Q_\alpha = diag(10^{-8}, 10^{-7}, 10^{-6})$ and $Q_z = 10^{-4}$ and the estimates were smoothed with another low pass filter of 15 min. An example of the parameter estimation is shown in Fig. 11, where the estimates in the faulty parameter approximate the correct value around hour 11:30. As expected, the other estimates deviate from the correct value due to the strong interconnection between the faults, but the estimates of the actual class approach the correct value. Again, a classification phase is necessary to isolate the fault type. Regarding time consumption, the mean time of each iteration of the EKF was 0.012 s.

The second feedforward ANN was trained with $\sigma = 5 \cdot 10^{-5}$, $\lambda = 5 \cdot 10^{-7}$, a maximum number of epochs of $10^3$, a minimum gradient of $10^{-6}$ and a maximum of 6 validation checks. It contains hyperbolic tangent functions in the first layers and a softmax function in the last one. The neural network is formed by two hidden layers, with 200 and 100 neurons in each one.

The average estimation and classification errors at the end of each day are summarized in Table 3. Although there is a loss of performance across the subsets, the results on the test set are even higher than on the sunny dataset since the variability of this dataset helped to isolate the faults. The accuracies obtained are over 90% in the experiments where the second ANN was applied and almost all F1-scores are above 90%, except for the flow rate fault on the test set. Moreover, the estimation errors were close to 2%, with a slightly higher value on the validation and test sets.

#### 6.2.1. Fault boundary

In this work, the faults are assumed to be greater than 10% for the optical efficiency and thermal losses and than 0.5 $m^3/s$ for the flow rate. This assumption is taken based on an intuitive approximation of the measurement systems. Minor faults would be confused with noise and errors in measurement instruments, and they have minimal effect on the outlet temperature. To adapt it to smaller deviations, it would be necessary to obtain a new dataset and retrain the system.

To provide an insight of this range of faults, we have carried out experiments with different fault sizes (with 10 experiments for each fault size). To obtain the preliminar classification with these new tests, we considered that there is no fault when the estimation from the Kalman filter is 1% around the nominal value. The average accuracies and estimation errors are shown in Figs. 12 and 13.

In the case of the estimation error, there is no loss of performance when including small faults, but there is a loss of accuracy. This is because the RNN is able to model the system and the faults, but the classifier neural network is adjusted to the range of faults with which it was trained.

## 7. Discussion and conclusions

In this paper, an EKF-based fault detection and diagnosis methodology has been presented and applied to a PTC system with both sunny and cloudy datasets. First, an RNN was implemented to model the system using EKF to obtain the parameters with an MSE of the order of $10^{-3}$ or even less in the three subsets of each dataset. A second EKF stage was performed by independently estimating the fault parameters with a percentage error of less than 3% in the sunny dataset and a maximum of 2.02% in the cloudy dataset. To select the correct type of fault, a feedforward neural network was applied, obtaining accuracies above 90%.

The choice of RNNs is well-suited theoretically to capture the system dynamics, as it learns sequential patterns. This design allows the model to capture long-term dependencies and obtain a fast representation of the system that can be combined with other methodologies. The tests demonstrate the effectiveness of the methodology using an RNN to model the dynamic behavior of the system, combined with the knowledge of the dynamics of solar radiation due to clouds. This leads to

**Table 3**
Evaluation metrics of the methodology on the training, validation and test sets of the cloudy dataset.

| | Without feedforward ANN | | | With feedforward ANN | | |
|---|---|---|---|---|---|---|
| | Training | Validation | Test | Training | Validation | Test |
| Average estimation error in the faulty parameter (%) | 1.45 | 2.02 | 1.68 | 1.45 | 2.02 | 1.68 |
| F1 in the faultless case (%) | 91.72 | 95.24 | 90.72 | 98.77 | 93.02 | 96.15 |
| F1 in the $K_{opt}$ fault (%) | 87.01 | 89.47 | 82.57 | 98.09 | 91.89 | 90.20 |
| F1 in the $q$ fault (%) | 80.75 | 90.48 | 73.68 | 98.16 | 92.68 | 88.89 |
| F1 in the $H_l$ fault (%) | 80.95 | 94.74 | 76.77 | 98.73 | 97.44 | 94.74 |
| Classification accuracy (%) | 85.00 | 92.50 | 81.00 | 98.44 | 93.75 | 92.50 |

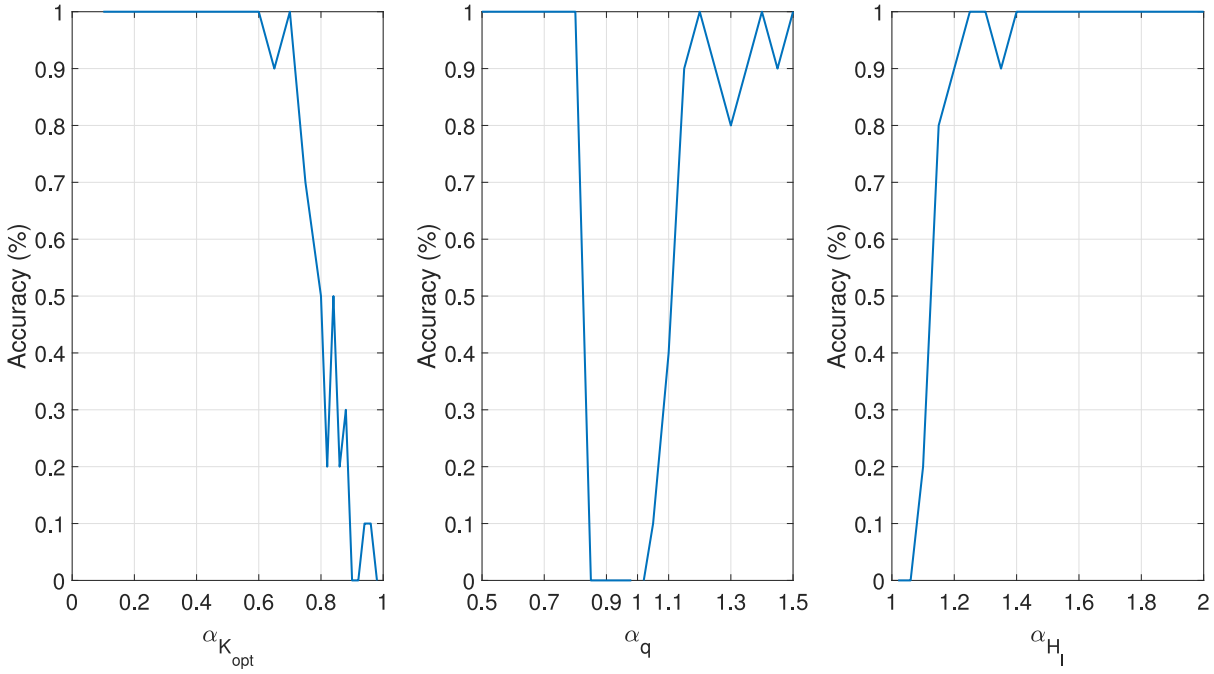



**Fig. 12.** Average accuracies with different fault sizes.

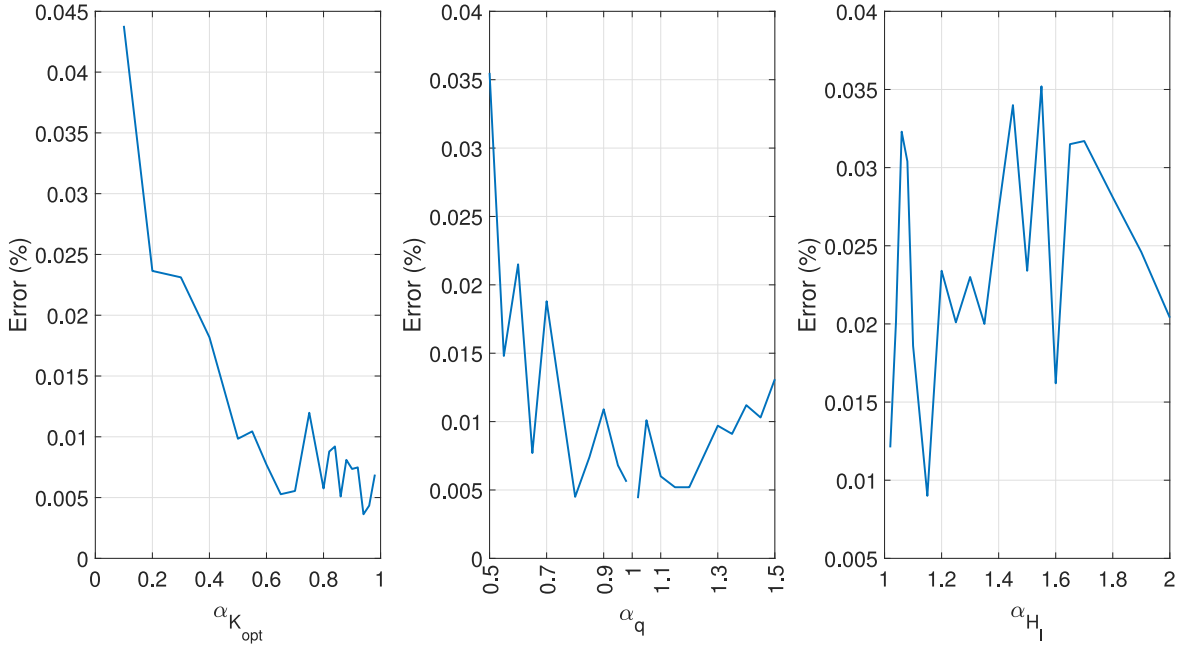



**Fig. 13.** Average estimation errors with different fault sizes.

results on the cloudy dataset that even outperform those on the sunny dataset. Furthermore, the accuracies and F1 scores obtained in these tests exceed those obtained in other previously published works using different machine learning-based approaches [24]. The depth of the ANNs aligns with universal approximation theory, which suggests that sufficiently deep networks can represent complex functions necessary for this task. In addition, EKF is helpful for this application because it is designed for systems with nonlinear dynamics by linearizing around the current estimate and aligns well with real-time data processing requirements.

The results obtained in this work prove the goodness of the methodology and open the way for possible future tests in an actual plant, leveraging the rapid training with EKF for a real-time learning approach. In experiments with commercial plants, real-time learning could be performed in parallel with plant operation, with each loop serving as a component for mini-batch training due to the high number of loops (for instance, 90 loops in Solacor 2 [39] and 282 in Mojave [40]).

One possible limitation of this work is the dependency of the modeled system on the controller used to generate the data. This work uses a feedforward controller to manipulate the flow rate and track the desired set-point. Different control techniques, such as MPC, may be used in actual plants. Although these controllers do not produce the same output, the control signal and the ranges in which it varies are representative. Even though, this could create dependencies between disturbances and the control actions that interfere with the model behavior. A deeper analysis of this dependency could be performed by studying the retraining of the neural networks in the case the plant controller is changed.

Future work could analyze the detectable range of faults similarly to the work in [21]. To adapt the methodology to smaller deviations, it would be necessary to obtain a new dataset and retrain the system. Moreover, although the method described only considers the possibility of separate faults, including coincident faults is fairly straightforward. It would be necessary to add four estimators and four outputs to the classifier, one for each possible fault combination, thus obtaining eight fault classes. The inputs and outputs of the MLP would be affected: its output vector would be $\hat{y}(k) = (\hat{y}_{\text{faultless}}(k), \hat{y}_{K_{\text{opt}}}(k), \hat{y}_{q}(k), \hat{y}_{H_{\text{l}}}(k), \hat{y}_{\text{K}_{\text{opt}}+q}(k), \hat{y}_{\text{K}_{\text{opt}}+H_{\text{l}}}(k), \hat{y}_{q+H_{\text{l}}}(k), \hat{y}_{\text{K}_{\text{opt}}+q+H_{\text{l}}})$. Instead of three EKFs, it would be necessary to have one EKF of each situation (one that considered that $\alpha_{K_{\text{opt}}} = 1$, another one that considered that $\alpha_{q} = 1$, another that considered that $\alpha_{H_{\text{l}}} = 1$ and another one that considered that neither of them is one). The dataset should be updated to include data with compound faults.

Regarding the application of this methodology to physical systems, commercial plants have many loops, which facilitates gathering data and augments the probability of finding faults. For example, Mojave has 282 loops instead of the 10 loops that ACUREX has. This means that if ACUREX needs 45 days to obtain 450 simulations, we would obtain 12 690 simulations in the same time. Collecting data from every loop over several months would cover the range of typical faults in the plant. On the other hand, this could also be addressed with digital twins that reproduce accurately the behavior of the plant. These aspects are proposed for future development, adapting this methodology to large-scale solar plants with a defocusing strategy and distinguishing faults on every collector, not only loop by loop. Moreover, future work will be applying other filter-based methods to improve the results of the EKF, like the unscented Kalman filter or the particle filter.

## CRediT authorship contribution statement

**Sara Ruiz-Moreno:** Writing – review & editing, Writing – original draft, Software, Methodology, Investigation. **Alberto Bemporad:** Writing – review & editing, Supervision, Software, Methodology, Conceptualization. **Antonio Javier Gallego:** Writing – review & editing, Supervision, Software, Conceptualization. **Eduardo Fernández Camacho:** Writing – review & editing, Supervision, Resources, 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.

## References

[1] Sanjuan A, Nejjari F, Sarrate R. Actuator fault estimation using optimization-based learning techniques for linear parameter varying systems with unreliable scheduling parameters. Eng Appl Artif Intell 2024;127:107247.
[2] Zafra-Cabeza A, Marquez JJ, Bordons C, Ridao MA. An online stochastic MPC-based fault-tolerant optimization for microgrids. Control Eng Pract 2023;130:105381.
[3] Isermann R. Fault-diagnosis systems: an introduction from fault detection to fault tolerance. Springer Science & Business Media; 2006.
[4] Molinié D, Madani K, Amarger V, Chebira A. Identifying the regions of a space with the self-parameterized recursively assessed decomposition algorithm (SPRADA). Mach Learn Knowl Extr 2023;5:979–1009.
[5] Chen K, Chen S, Zhu X, Jin X, Du Z. Interpretable mechanism mining enhanced deep learning for fault diagnosis of heating, ventilation and air conditioning systems. Build Environ 2023;237:110328.
[6] Karim AKA, Atoui MA, Degardin V, Laly P, Cocquempot V. Bus network decomposition for fault detection and isolation through power line communication. ISA Trans 2023;137:492–505.
[7] Hajji M, Yahyaoui Z, Mansouri M, Nounou H, Nounou M. Fault detection and diagnosis in grid-connected PV systems under irradiance variations. Energy Rep 2023;9:4005–17.
[8] Kumar SR, Devakumar J. Recurrent neural network based sensor fault detection and isolation for nonlinear systems: Application in PWR. Prog Nucl Energy 2023;163:104836.
[9] Rodríguez F, Chicaiza WD, Sánchez A, Escaño JM. Updating digital twins: Methodology for data accuracy quality control using machine learning techniques. Comput Ind 2023;151:103958.
[10] Bidou MS, Verron S, Perez L, Autrique L. Kalman smoother for detection of heat sources defects. In: 2022 international conference on control, automation and diagnosis. ICCAD, 2022, p. 1–6.
[11] Gañán FJ, Sanchez-Diaz JA, Tapia R, de Dios JRM, Ollero A. Efficient event-based intrusion monitoring using probabilistic distributions. In: 2022 IEEE international symposium on safety, security, and rescue robotics. SSRR, 2022, p. 211–6.
[12] Machado DO, Chicaiza WD, Escaño JM, Gallego AJ, de Andrade GA, Normey-Rico JE, Bordons C, Camacho EF. Digital twin of a Fresnel solar collector for solar cooling. Appl Energy 2023;339:120944.
[13] Pataro IML, Cunha R, Gil JD, Guzmán JL, Berenguel M, Lemos JM. Optimal model-free adaptive control based on reinforcement Q-Learning for solar thermal collector fields. Eng Appl Artif Intell 2023;126:106785.
[14] Chanfreut P, Sánchez-Amores A, Maestre JM, Camacho EF. Distributed model predictive control based on dual decomposition with neural-network-based warm start. In: 2021 European control conference. ECC, 2021, p. 1969–74.
[15] Ketkar N, Ketkar N. Stochastic gradient descent. In: Deep learning with python: a hands-on introduction. Springer; 2017, p. 113–32.
[16] Trebatickỳ P, Pospíchal J. Neural network training with extended kalman filter using graphics processing unit. In: International conference on artificial neural networks. 2008, p. 198–207.
[17] Bemporad A. Recurrent neural network training with convex loss and regularization functions by extended Kalman filtering. IEEE Trans Autom Control 2023;68(9):5661–8.
[18] Rakhshani E, Rouzbehi K, Sánchez AJ, Tobar AC, Pouresmaeil E. Integration of large scale PV-based generation into power systems: A survey. Energies 2019;12:1425.
[19] Schmelzer C, Georgii M, Orozaliev J, Vajen K. Fault detection for solar thermal systems–overall system evaluation or component-oriented approach. In: Proceedings of the ISES euroSun 2020 conference–13th international conference on solar energy for buildings and industry. presented at the euroSun 2020. 2020, p. 1–8.
[20] Brenner A, Kahn J, Hirsch T, Röger M, Pitz-Paal R. Soiling determination for parabolic trough collectors based on operational data analysis and machine learning. Sol Energy 2023;259:257–76.
[21] Ruiz-Moreno S, Gallego AJ, Camacho EF. Artificial neural network-based fault detection and isolation in a parabolic-trough solar plant with defocusing strategy. Sol Energy 2023;262:111909.
[22] Gallego AJ, Macías M, de Castilla F, Camacho EF. Mathematical modeling of the Mojave solar plants. Energies 2019;12:4197.
[23] Camacho EF, Berenguel M, Rubio FR, Martinez D. Control of solar energy systems. Springer-Verlag; 2012.
[24] Ruiz-Moreno S, Gallego AJ, Sanchez AJ, Camacho EF. A cascade neural network methodology for fault detection and diagnosis in solar thermal plants. Renew Energy 2023;211:76–86.

[25] Himour Y, Tadjine M, Boucherit MS. Nonlinear and infinite gain scheduling neural predictive control of the outlet temperature in a parabolic trough solar field: A comparative study. Eng Appl Artif Intell 2023;126:106862.
[26] Camacho EF, Berenguel M, Rubio FR. Advanced control of solar plants. Springer; 1997.
[27] Goswami DY, Kreith F, Kreider JF. Principles of solar engineering. CRC Press; 2000.
[28] Len AJG, Yebra LJ, Camacho EF, Sánchez AJ. Mathematical modeling of the parabolic trough collector field of the TCP-100 research plant. Linköpings universitet: Linköping University Electronic Press; 2018, p. 912–8.
[29] Österholm R, Pålsson J. Dynamic modelling of a parabolic trough solar power plant. 2014, p. 409–18.
[30] Tealab A. Time series forecasting using artificial neural networks methodologies: A systematic review. Future Comput Inform J 2018;3:334–40.
[31] Amari S. Backpropagation and stochastic gradient descent method. Neurocomputing 1993;5:185–96.
[32] Du S, Lee J, Li H, Wang L, Zhai X. Gradient descent finds global minima of deep neural networks. In: International conference on machine learning. 2019, p. 1675–85.
[33] Čurn J, Marinescu D, Lacey G, Cahill V. Estimation with non-white Gaussian observation noise using a generalised ENSEMBLE KALMAN filter. In: 2012 IEEE international symposium on robotic and sensors environments proceedings. 2012, p. 85–90.
[34] Stacey N, D'Amico S. Adaptive and dynamically constrained process noise estimation for orbit determination. IEEE Trans Aerosp Electron Syst 2021;57(5):2920–37.
[35] Ribeiro MI. Kalman and extended kalman filters: Concept, derivation and properties. Inst Syst Robot 2004;43:3736–41.
[36] Särkkä S, Svensson L. Bayesian filtering and smoothing. Vol. 17, Cambridge University Press; 2023.
[37] Møller MF. A scaled conjugate gradient algorithm for fast supervised learning. Neural Netw 1993;6:525–33.
[38] Andersson JAE, Gillis J, Horn G, Rawlings JB, Diehl M. CasADi – a software framework for nonlinear optimization and optimal control. Math Program Comput 2019;11:1–36.
[39] Solacor 2 project. 2021, URL https://solarpaces.nrel.gov/project/solacor-2.
[40] Mojave solar project. 2022, URL https://solarpaces.nrel.gov/project/mojave-solar-project.