

Research Article

Multivariable Controller Design for the Cooling System of a PEM Fuel Cell by considering Nearly Optimal Solutions in a Multiobjective Optimization Approach

Alberto Pajares , Xavier Blasco , Juan M. Herrero , and Raúl Simarro

Instituto Universitario de Automática e Informática Industrial, Universitat Politècnica de València, Valencia, Spain

Correspondence should be addressed to Alberto Pajares; alpafer1@upv.es

Received 17 May 2020; Revised 10 July 2020; Accepted 21 July 2020; Published 6 August 2020

Guest Editor: Ramon Costa Castelló

Copyright © 2020 Alberto Pajares et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper presents a design for the multivariable control of a cooling system in a PEM (proton exchange membrane) fuel cell stack. This system is complex and challenging enough: interactions between variables, highly nonlinear dynamic behavior, etc. This design is carried out using a multiobjective optimization methodology. There are few previous works that address this problem using multiobjective techniques. Also, this work has, as a novelty, the consideration of, in addition to the optimal controllers, the nearly optimal controllers nondominated in their neighborhood (potentially useful alternatives). In the multiobjective optimization problem approach, the designer must make decisions that include design objectives; parameters of the controllers to be estimated; and the conditions and characteristics of the simulation of the system. However, to simplify the optimization and decision stages, the designer does not include all the desired scenarios in the multiobjective problem definition. Nevertheless, these aspects can be analyzed in the decision stage only for the controllers obtained with a much less computational cost. At this stage, the potentially useful alternatives can play an important role. These controllers have significantly different parameters and therefore allow the designer to make a final decision with additional valuable information. Nearly optimal controllers can obtain an improvement in some aspects not included in the multiobjective optimization problem. For example, in this paper, various aspects are analyzed regarding potentially useful solutions, such as (1) the influence of certain parameters of the simulator; (2) the sample time of the controller; (3) the effect of stack degradation; and (4) the robustness. Therefore, this paper highlights the relevance of this in-depth analysis using the methodology proposed in the design of the multivariable control of the cooling system of a PEM fuel cell. This analysis can modify the final choice of the designer.

1. Introduction

In control engineering, optimization tools are widely used, for example, in the design of control systems [1–3]. Normally, in these problems, there are several conflicting objectives to optimize (such as output errors, control effort, and robustness). Therefore, it is reasonable to solve these problems as multiobjective optimization problems (MOP [4–7]). Thus, the designer can analyze the controller trade-off for each design objective and can better choose the final controller.

In an MOP, nearly optimal solutions (also called approximate or ϵ -efficient solutions) have been studied by many authors so far [8–10]. These alternatives have a slightly

worse performances than the optimal solutions and may be useful to the designer. However, these alternatives are ignored in a classic MOP and considering all these solutions can slow down the algorithm and overcomplicate the decision stage. Among them, the solutions with similar performance to the optimal ones in the objective space and those that differ significantly in the parameter space (different neighborhoods) are the potentially useful alternatives for the designer [11–13]. We define the potentially useful solutions as the optimal and nearly optimal solutions nondominated in their neighborhood. These alternatives provide the designer with greater diversity without increasing excessively the number of possible alternatives [11]. The designer can then analyze these controllers (significantly

different from the alternatives obtained in a classical MOP) and make the final decision with higher quality information. Thus, it is possible to find nearly optimal controllers with better performances than the optimal ones in some features not included in the design objectives [14]. In this situation, the designer could choose a nearly optimal controller instead of an optimal one.

To formulate an MOP, the designer must define certain aspects. In the design of control systems, the MOP must contemplate aspects such as

- (i) Design objectives for the control.
- (ii) Parameters to be adjusted for the defined control structure.
- (iii) Process model and its operation point.
- (iv) Definition of the simulation: input signals, disturbances, and noise.
- (v) Simulation environment setup:
 - (a) Integration method.
 - (b) Sample time of the controller.
 - (c) Other less relevant aspects, a priori, such as the nature of the noise (if existing).

Firstly, the designer must choose the design objectives. These objectives are defined to measure certain characteristics of the control such as output errors, control efforts, or robustness. However, each of these characteristics can be analyzed using different indicators. For example, output errors can be measured by ITAE (integral time absolute error), IAE (integral absolute error), ISE (integral squared error), and ITSE (integral time square error) [15]. Including many of these indicators in the MOP would increase the number of objectives and this has two drawbacks: a greatly increased computational cost and a more complicated decision stage. Increasing the number of the design objectives increases optimal and nearly optimal solutions if a similar discretization is maintained. Each new solution generated must be compared with all the optimal solutions and nearly optimal ones to analyze their inclusion in the Pareto set (or nearly optimal set). Therefore, increasing the design objectives significantly increases the computational cost of the optimization process. Furthermore, this increase in solutions, together with the new design objectives added, makes the decision process and final decision more difficult for the designer. Therefore, the designer usually chooses only some of the indicators to define the MOP. The rest of the interesting indicators can be analyzed more cheaply in the decision stage [14].

In the design of controllers, another important element is the process model employed. This model may have uncertainties in its structure and/or in the value of its parameters. There are different methods to evaluate the impact of these uncertainties: such as Monte Carlo [16] and minimax [17]. Nevertheless, these methods increase the computational cost of the MOP [18].

The simulation environment setup is set in the definition of the MOP. However, there may be different setups valid for the designer: different numerical integration methods,

different sample times, different types of noises, etc. Considering all these design alternatives in the MOP is unapproachable.

The designer must therefore establish certain fundamental aspects to define the MOP, but there are other interesting aspects that are not included in the MOP due to various limitations (usually related to the computational cost). These aspects ignored in the optimization stage could be analyzed in the decision stage on the controllers obtained during optimization. In this scenario, the nearly optimal alternatives nondominated in their neighborhood have an especially relevant role. These controllers have similar performances to the optimal ones, but they have significantly different parameters (they are located in different neighborhoods). These controllers could produce an improvement (even significant) over the optimal ones in some aspect not included in the optimization process. For this reason, it can be very valuable to obtain these nearly optimal controllers. Depending on their behavior in the aspects not considered in the optimization, the final choice can be made for a nearly optimal controller.

Let us look at a specific example. We define an MOP for the adjustment of a proportional integral (PI) controller (parameters to be adjusted: gain K_c and integral time T_i) for a nonlinear nominal model of the process. The MOP is defined with two design objectives for changes in the setpoint: f_1 measures the output errors through the ITAE and f_2 measures the control effort through IAU (integral of absolute control effort). In the simulation environment, steps are introduced in the setpoint and a noise in the outputs (similar to the noise present in the actual process), and no additional disturbances are considered. Under this scenario, the set of optimal and nearly optimal controllers is shown in Figure 1.

There are a set of optimal controllers (SET_1) and a set of nearly optimal controllers nondominated in their neighborhood (SET_2 and SET_3) in various neighborhoods of the parameter space. SET_2 and SET_3 provide the designer with new alternatives significantly different from the optimal ones (but with similar performances). Thus, these controllers are potentially useful for the designer. From these alternatives, we select optimal controllers \mathbf{x}^1 and \mathbf{x}^2 and nearly optimal controllers \mathbf{x}^3 , \mathbf{x}^4 , and \mathbf{x}^5 (see Table 1).

The designer now wants to analyze the output errors of the selected alternatives with a new indicator, IAE. This new indicator has not been contemplated in the optimization stage of the MOP. The value of this indicator on the selected controllers can be seen in Table 1. The nearly optimal controllers \mathbf{x}^5 and \mathbf{x}^4 obtain a better IAE than the optimal controllers \mathbf{x}^1 and \mathbf{x}^2 , respectively. In fact, the nearly optimal controller \mathbf{x}^5 obtains a significant improvement over the optimal controller \mathbf{x}^1 . Additionally, the designer analyzes the robustness of the alternatives \mathbf{x}^2 and \mathbf{x}^4 . For this, we consider the uncertainty in the parameters of the nonlinear model. The controllers are evaluated on 50 random variations of the model parameters with a variation of $\pm 10\%$. Figure 2 shows the value of the design objectives for each of the 50 variations of the model. The blue points are the objective value obtained by each of the model variations on the optimal

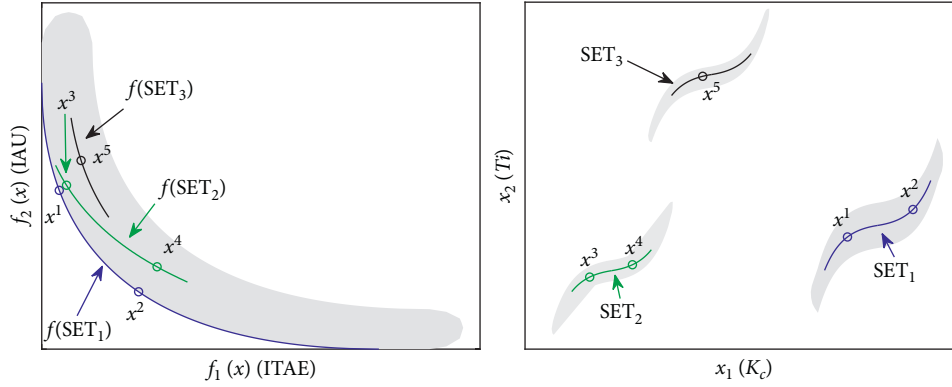


FIGURE 1: A multiobjective example: the optimal solutions are the ones in set SET_1 and the nearly optimal solutions nondominated in their neighborhood are the ones in SET_2 and SET_3 . The gray area represents the zone of the nearly optimal solutions.

TABLE 1: Selected controllers.

Controller	f_1 (ITAE)	f_2 (IAU)	IAE
\mathbf{x}^1	0.25	0.75	0.5
\mathbf{x}^2	0.75	0.25	0.78
\mathbf{x}^3	0.27	0.76	0.8
\mathbf{x}^4	0.8	0.3	0.77
\mathbf{x}^5	0.35	0.85	0.25

TABLE 2: Selected controllers.

Controller	Initial noise		New noise	
	f_1 (ITAE)	f_2 (IAU)	f'_1 (ITAE)	f'_2 (IAU)
\mathbf{x}^1	0.25	0.75	0.26	0.77
\mathbf{x}^2	0.75	0.25	0.77	0.24
\mathbf{x}^3	0.27	0.76	0.25	0.76
\mathbf{x}^4	0.8	0.3	0.78	0.32
\mathbf{x}^5	0.35	0.85	0.36	0.84

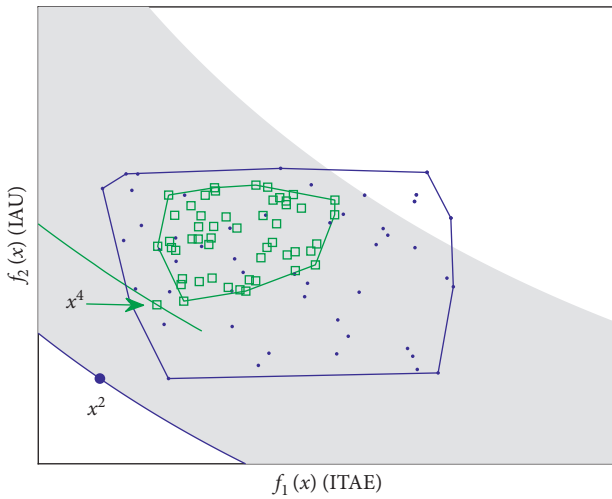


FIGURE 2: Degradation limits of \mathbf{x}^2 (optimal) and \mathbf{x}^4 (nearly optimal) controllers under uncertainty in the model. The uncertainty is measured by varying the model parameters by $\pm 10\%$.

controller \mathbf{x}^2 . The green squares are the objective value obtained by each of the variations of the model on the nearly optimal controller \mathbf{x}^4 . The nearly optimal controller \mathbf{x}^4 has a lower degradation than the optimal controller \mathbf{x}^2 . Finally, the designer analyzes the influence of the noise in the selected controllers. The controllers are evaluated on the design objectives by varying the type of noise (f'_1 and f'_2 in Table 2). The optimal alternative \mathbf{x}^1 is dominated by the nearly optimal controller \mathbf{x}^3 in this new scenario. So, \mathbf{x}^3 is nearly as good as \mathbf{x}^1 , and the dominance depends on the noise chosen for the simulations. Therefore, the nearly optimal alternatives bring new controllers potentially useful for the

designer. This diversity allows the designer to make a final decision with additional valuable information. In addition, the nearly optimal controllers present improvements in the three defined scenarios over the optimal controllers selected. Further, the nearly optimal solutions obtain similar performance in the design objectives compared to the optimal ones. This situation may lead the final choice of the designer towards a nearly optimal controller to the detriment of an optimal one.

In this work, we propose the design of a multivariable control system for the cooling circuit of a proton exchange membrane fuel cell (PEMFC [19, 20]) stack. The correct design of the stack cooling system is vital in the durability, cost, reliability, and energy efficiency of the stack [21–24]. The PEMFC stack is part of a microcombined heat and power (μ -CHP) system [25–28]. μ -CHP systems are co-generation systems. The main advantage of these systems is the use of the thermal energy produced in the generation of electrical energy. In this way, the efficiency of the system is increased. μ -CHP systems employ various technologies [29]. Among them, some authors agree that the most promising, due to their efficiency and low emissions, are technologies based on the fuel cell [30, 31]. The most common μ -CHP systems of this type are based on PEMFC stacks. These systems are sometimes used for the electrical and thermal supply of homes [32]. Nevertheless, it is necessary to advance in several technical aspects to improve the performance of these systems and reduce their costs. One of the most important work areas for improvement is the temperature control of the PEMFC [33, 34].

In this paper, a nonlinear model of a PEM fuel cell (Nedstack, model 2.0HP) is used which is able to produce up

to 2 kW of electrical energy and 3.3 kW of thermal energy. This stack is cooled by a liquid cooling system. This model is described in [35]. Using the methodology presented, nearly optimal controllers will be obtained. These controllers provide the designer with significantly different alternatives (in their parameters). Thanks to them, the designer can make a final decision using additional valuable information. Therefore, this work shows as a novelty the usefulness of considering the nearly optimal alternatives nondominated in their neighborhood in the design of a multivariable control system. The methodology proposed in the paper enables the maximum exploitation of a specific control technology by using valuable information in the tuning procedure.

This work is structured as follows. In Section 2, some basic definitions previously presented in the literature are described. In Section 3, the nevMOGA algorithm used in this work is described briefly. In Section 4, the design of a multivariable control system for the cooling system of a PEMFC stack is presented. Finally, the conclusions are commented in Section 5.

2. Background

The resolution of an MOP produces a set of optimal solutions (P_Q). There is also a set of nearly optimal solutions that could be interesting for the decision maker ($P_{Q,\varepsilon}$) and which are ignored in a classic MOP. Nevertheless, finding all of the nearly optimal solutions can considerably increase the number of alternatives. Among them, the solutions nondominated in their neighborhood (potentially useful, $P_{Q,n}$) provide the designer with alternatives that are close to the optimal ones in objective space, which differ significantly in parameter space. These alternatives maintain the diversity in parameter space without excessively increasing the number of possible alternatives. With them, the designer can make the final decision with the benefit of valuable additional information. In this section, these sets are defined.

A multiobjective optimization problem (a maximization problem can be converted into a minimization problem; for each of the objectives that have to be maximized, the transformation: $\max f_i(\mathbf{x}) = -\min(-f_i(\mathbf{x}))$ can be realised) can be defined as follows:

$$\begin{aligned} \min_{\mathbf{x} \in Q} \quad & \mathbf{f}(\mathbf{x}) \\ \text{subject to} \quad & \underline{x}_i \leq x_i \leq \bar{x}_i, \quad i = [1, \dots, k], \end{aligned} \quad (1)$$

where $\mathbf{x} = [x_1, \dots, x_k]$ is defined as a decision vector in the domain $Q \subset \mathfrak{R}^k$ and $\mathbf{f}: Q \rightarrow \mathfrak{R}^m$ is defined as the vector of objective functions $\mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), \dots, f_m(\mathbf{x})]$. \underline{x}_i and \bar{x}_i are the lower and upper bounds of each component of \mathbf{x} .

Definition 1. Dominance [36]: a decision vector \mathbf{x}^1 is dominated by any other decision vector \mathbf{x}^2 if $f_i(\mathbf{x}^2) \leq f_i(\mathbf{x}^1)$ for all $i \in [1, \dots, m]$ and $f_j(\mathbf{x}^2) < f_j(\mathbf{x}^1)$ for at least one $j, j \in [1, \dots, m]$. This is denoted as $\mathbf{x}^2 \leq \mathbf{x}^1$.

Definition 2. Pareto set: the Pareto set (denoted by P_Q) is the set of solutions in Q that is nondominated by another solution in Q :

$$P_Q := \{\mathbf{x} \in Q \mid \exists \mathbf{x}' \in Q: \mathbf{x}' \leq \mathbf{x}\}. \quad (2)$$

Definition 3. Pareto front: given a set of Pareto optimal solutions P_Q , the Pareto front $\mathbf{f}(P_Q)$ is defined as

$$\mathbf{f}(P_Q) := \{\mathbf{f}(\mathbf{x}) \mid \mathbf{x} \in P_Q\}. \quad (3)$$

Definition 4. $-\varepsilon$ -dominance [37]: define $\varepsilon = [\varepsilon_1, \dots, \varepsilon_m]$ as the maximum acceptable performance degradation. A decision vector \mathbf{x}^1 is $-\varepsilon$ -dominated by another decision vector \mathbf{x}^2 if $f_i(\mathbf{x}^2) + \varepsilon_i \leq f_i(\mathbf{x}^1)$ for all $i \in [1, \dots, m]$ and $f_j(\mathbf{x}^2) + \varepsilon_j < f_j(\mathbf{x}^1)$ for at least one $j, j \in [1, \dots, m]$. This is denoted by $\mathbf{x}^2 <_{-\varepsilon} \mathbf{x}^1$.

Definition 5. ε -efficiency [13]: the set of ε -efficient solutions (denoted by $P_{Q,\varepsilon}$) is the set of solutions in Q which are not $-\varepsilon$ -dominated by another solution in Q :

$$P_{Q,\varepsilon} := \{\mathbf{x} \in Q \mid \exists \mathbf{x}' \in Q: \mathbf{x}' \leq_{-\varepsilon} \mathbf{x}\}. \quad (4)$$

Definition 6. Neighborhood: define $\mathbf{n} = [n_1, \dots, n_k]$ as the maximum distance between neighboring solutions. Two decision vectors \mathbf{x}^1 and \mathbf{x}^2 are neighboring solutions ($\mathbf{x}^1 =_n \mathbf{x}^2$) if $|x_i^1 - x_i^2| < n_i$ for all $i \in [1, \dots, k]$.

Definition 7. n -dominance: a decision vector \mathbf{x}^1 is n -dominated by another decision vector \mathbf{x}^2 if they are neighboring solutions (Definition 6) and $\mathbf{x}^2 < \mathbf{x}^1$. This is denoted by $\mathbf{x}^2 \leq_n \mathbf{x}^1$.

Definition 8. n -efficiency [11]: the set of n -efficient solutions (denoted by $P_{Q,n}$) is the set of solutions of $P_{Q,\varepsilon}$ which are not n -dominated by another solution in $P_{Q,\varepsilon}$:

$$P_{Q,n} := \{\mathbf{x} \in P_{Q,\varepsilon} \mid \exists \mathbf{x}' \in P_{Q,\varepsilon}: \mathbf{x}' <_n \mathbf{x}\}. \quad (5)$$

The sets P_Q , $P_{Q,\varepsilon}$, and $P_{Q,n}$ may have infinite solutions. Therefore, obtaining these is often unapproachable computationally. Normally, the designer obtains the discrete sets $P_Q^* \subset P_Q$, $P_{Q,\varepsilon}^* \subset P_{Q,\varepsilon}$, and $P_{Q,n}^* \subset P_{Q,n}$, in such a way that appropriately characterize P_Q , $P_{Q,\varepsilon}$, and $P_{Q,n}$, respectively.

Figure 3 shows an example. There is a set of optimal solutions SET_1 located in the neighborhood₁. In addition, there is a set of nearly optimal solutions (gray area). Both sets form $P_{Q,\varepsilon}$. Therefore, if the designer considers the nearly optimal alternatives, he or she will obtain new alternatives significantly different from the optimal ones (neighborhood₂ and neighborhood₃ solutions). A knowledge of these neighborhoods enables the designer to make a more informed final decision. But adding all the nearly optimal solutions has two drawbacks: it slows down the algorithm and complicates the decision stage. Nevertheless, the nearly optimal solutions nondominated in their neighborhood provide the designer with diversity in the obtained set without excessively increasing the number of possible alternatives. Consequently, it avoids the two drawbacks mentioned previously. In this example, these alternatives are

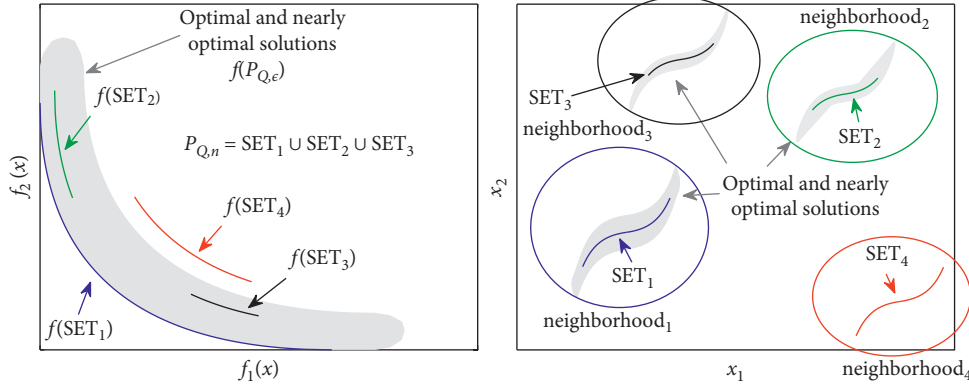


FIGURE 3: An MOP example. On the left, the objective space is shown, and on the right, the parameter space is shown. The optimal solutions are those in set SET_1 and the nearly optimal solutions nondominated in their neighborhood are the those in SET_2 and SET_3 . SET_4 does not contain any nearly optimal solution, and therefore, it is discarded.

SET_1 , SET_2 , and SET_3 . We believe these alternatives are potentially useful solutions—the best solutions in each neighborhood.

There are various algorithms designed to provide nearly optimal alternatives [12, 13]. Nevertheless, many do not take into account the space of the parameters when they discard solutions. Therefore, these algorithms cannot guarantee that the potentially useful solutions have not been discarded. However, the algorithm nevMOGA [11] takes into account the space of the parameters in their discretization, guaranteeing that the potentially useful alternatives are not ruled out. This algorithm has been evaluated on various examples, obtaining a good approximation to the set $P_{Q,n}$ in every case [14, 38].

3. Materials and Methods

In this work, we use the algorithm nevMOGA (multi-objective evolutionary algorithm available in Matlab Central: <https://www.mathworks.com/matlabcentral/fileexchange/71448-nevmoga-multiobjective-evolutionary-algorithm>) [11]. This algorithm is an evolutionary algorithm based on the algorithm ev-MOGA [39]. nevMOGA provides the designer with a discrete set of optimal and nearly optimal solutions nondominated in their neighborhood (potentially useful solutions) $P_{Q,n}^*$ (Definition 8). nevMOGA has four populations:

- (1) $P(t)$ is the main population. This population must converge towards $P_{Q,n}^*$, and not only towards P_Q , to achieve diversity in the set found. The number of individuals in this population is $Nind_P$.
- (2) $Front(t)$ is the archive where a discrete approximation of the Pareto front (P_Q^*) is stored. Its size is variable but bounded, depending on the number of boxes (divisions for each dimension, parameter n_box) previously defined by the designer.
- (3) $SubFront(t)$ is the archive where a discrete approximation of the nearly optimal solutions nondominated in their neighborhood ($P_{Q,n}^* \setminus P_Q^*$) is stored. The size of this population varies.

Nevertheless, the size of this population is limited and based on the number of boxes (divisions for each dimension, parameter n_box).

- (4) $G(t)$ is an auxiliary population. The population $G(t)$ stores the new individuals generated in each iteration. The size of this population is $Nind_{GA}$, which must be multiple of 4.

Figure 4 shows a flowchart of nevMOGA. First, populations $Front$ and $SubFront$ are initialized (empty sets). Then, the population P is created randomly. The designer can initially define part or all of the population P with an initial population. Later the populations $Front$ and $SubFront$ are updated from individuals of P . Then, in each iteration, the following is done: (1) create the evolved subpopulation G (by crossing and mutating individuals from $Front$, $SubFront$ and P); (2) update the populations $Front$ and $SubFront$ if necessary; (3) update the population P when the populations $Front$ and $SubFront$ change.

The parameter ϵ defines the size of the area of nearly optimal solutions (maximum degradation acceptable to the designer, Definition 4), and its definition is necessary for the use of nevMOGA. In addition, the parameter n (neighborhood, Definition 6) defines from which value we consider two significantly different solutions (in the decision space), and their definition is recommended. If the knowledge necessary for its definition is unavailable, there is a simple procedure for calculating this parameter from the ϵ parameter and a reference solution [38]. So, a very large ϵ or a very small n can lead to an excessive number of solutions, slowing down the optimization process and complicating the decision stage. However, a very small ϵ or a very large n can lead to obtaining a very small number of solutions, discarding potentially useful nearly optimal alternatives.

4. Results and Discussion

In this section, the new approach to the design of a multivariable control system for the cooling system of a PEMFC stack is used. The PEMFC stack can be part of a cogeneration system, for example, the μ -CHP system. The main advantage

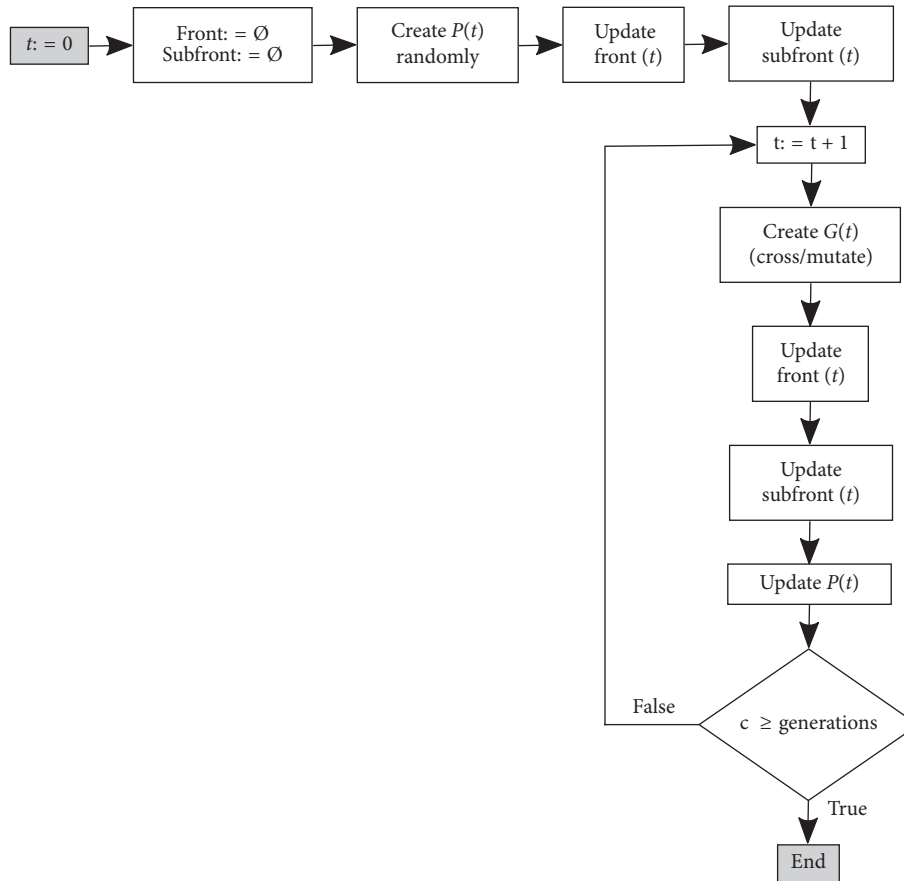


FIGURE 4: Flowchart of nevMOGA algorithm.

of these systems is that the use of the thermal energy produced in the generation of electrical energy increases overall efficiency. An accurate temperature control of the stack is necessary to improve the behavior of this type of system. Therefore, in this section, the control is designed using the new methodology.

The μ -CHP system used in this work is shown in Figure 5. The electric load demands electrical power to the PEMFC, requiring an electric current i . This current simulates the electrical demand of a house. To generate this current, the stack must be supplied with hydrogen (H_2) and oxygen (Air). In addition to the mentioned electrical energy, the stack generates thermal energy. Two water cooling circuits extract the heat, and the system consists of primary and secondary circuits, coupled by a heat exchanger. The heat generated by the stack is extracted by water from the primary circuit (with flow F_{w1}) at a temperature $T_{w_{out}}$ (water outlet temperature of the stack) and transferred to the secondary circuit (with flow F_{w2}) through the heat exchanger. The heat finally arrives in the hot water tank (Tank) for use (heating and hot water). The primary circuit consists of Pump 1 that propels the water in the primary circuit, regulating the flow of water that passes through the stack (F_{w1}). If F_{w1} increases, more heat is extracted and the stack cools down. The secondary circuit consists of a hot water tank and Pump 2. The water flow rate of the secondary circuit (F_{w2}) is varied using Pump 2. If this flow rate

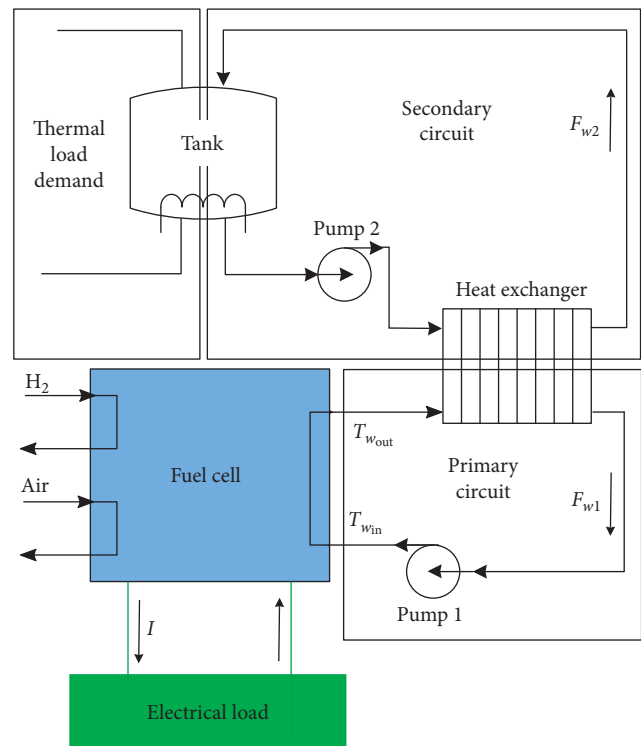


FIGURE 5: Diagram of the PEMFC stack cooling system.

decreases, the amount of heat transferred through the heat exchanger also decreases, and as a result, less heat passes from the primary circuit to the secondary circuit. The water temperature at the stack inlet ($T_{w_{in}}$) then increases.

In this paper, a nonlinear model of a PEM fuel cell is used. This model (available in <https://riunet.upv.es/handle/10251/118336>) is described in [35]. This model has been made in first principles and has more than 30 equations. This model simplifies some little relevant aspects of the cooling system. However, it is a complex model with high nonlinearities.

4.1. MOP. The methodology proposed in the paper enables the maximum exploitation of a specific control technology by using valuable information in the tuning procedure. For the design of the control system, a multiloop PI control was chosen because of its easier implementation and maintenance. The other control structures can be tuned with the same methodology. The RGA technique is used to establish the loop pairing [40]. Since the system is nonlinear, the static gains matrix is determined at three operating points corresponding to $T_{w_{out}} = 65^\circ\text{C}$ and $T_{w_{in}} = 60^\circ\text{C}$ and currents of $I = 140\text{ A}$, 170 A , and 200 A . For this, variations of 0.3 l/min are introduced in the flow rates of each inlet independently, at each operating point. Thus, the static gains represented by $K_{140\text{ A}}$, $K_{170\text{ A}}$, and $K_{200\text{ A}}$ are obtained, giving rise to the matrices $\text{RGA}_{140\text{ A}}$, $\text{RGA}_{170\text{ A}}$, and $\text{RGA}_{200\text{ A}}$, respectively:

$$\begin{aligned} \begin{bmatrix} T_w \\ T_w \end{bmatrix} &= K_{170\text{ A}} \begin{bmatrix} F_1 \\ F_2 \end{bmatrix} = \begin{bmatrix} -0.894 & -0.7112 \\ 0.197 & -0.743 \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \end{bmatrix}, \\ \text{RGA}_{170\text{ A}} &= \begin{bmatrix} 0.826 & 0.174 \\ 0.174 & 0.826 \end{bmatrix}, \\ K_{200\text{ A}} &= \begin{bmatrix} -0.735 & -0.415 \\ 0.099 & -0.433 \end{bmatrix} \rightarrow \text{RGA}_{200\text{ A}} = \begin{bmatrix} 0.886 & 0.115 \\ 0.115 & 0.886 \end{bmatrix}, \\ K_{140\text{ A}} &= \begin{bmatrix} -1.1 & -1.263 \\ 0.375 & -1.313 \end{bmatrix} \rightarrow \text{RGA}_{140\text{ A}} = \begin{bmatrix} 0.753 & 0.247 \\ 0.247 & 0.753 \end{bmatrix}. \end{aligned} \quad (6)$$

By observing and comparing the gains of the matrices $K_{140\text{ A}}$ and $K_{200\text{ A}}$, the static nonlinearity of the process is evident, with variations in the gains of up to 300%. The RGA matrices clearly show that the most suitable pairing is that of the main diagonal. Therefore, the control structure is defined as a multiloop PI control which uses the following loop pairing scheme: output $T_{w_{out}}$ is controlled by F_{w1} and $T_{w_{in}}$ by F_{w2} (see Figure 6). That is to say,

$$\begin{aligned} F_{w2}(s) &= Kc_1 \left(e_1(s) + \frac{1}{Ti_1} \frac{1}{s} e_1(s) \right), \\ F_{w1}(s) &= Kc_2 \left(e_2(s) + \frac{Ts}{Ti_2} \frac{1}{s} e_2(s) \right), \end{aligned} \quad (7)$$

where Kc_1 and Kc_2 are the proportional gains, Ti_1 and Ti_2 are the integral time constants in seconds, and $e_1 = r_{T_{w_{in}}} - T_{w_{in}}$ and $e_2 = r_{T_{w_{out}}} - T_{w_{out}}$ are the output errors, where $r_{T_{w_{in}}}$

and $r_{T_{w_{out}}}$ are the setpoints for $T_{w_{in}}$ and $T_{w_{out}}$ respectively. The actuators F_{w1} and F_{w2} must meet the restrictions $1.85 \leq F_{w1} \leq 6.9$ and $2.1 \leq F_{w2} \leq 9.5$, and for this the PI incorporates an antiwindup mechanism.

For tuning the controller of this system, the water temperature of the tank is maintained constant at 55°C . Thus, the secondary circuit of the μ -CHP system is simplified. The output temperatures of the real system have an associated noise. This noise is filtered to prevent it from spreading to system control actions. Similarly, for the simulation of the system, a noise similar to the real process is introduced. This noise will also be filtered so that the control actions of the system are less influenced by it.

Design objectives are evaluated throughout a defined test. This test has two changes of the current demanded at 500 and 1500 seconds (see Figure 7). In addition, as recommended by the manufacturer, the water outlet temperature from the stack ($T_{w_{out}}$) should be 65°C for the optimal stack operation. There must be a 5°C gradient between the temperature of the water input and output of the stack. Therefore, the system references are constant throughout the entire test: $r_{T_{w_{out}}} = 65^\circ\text{C}$ and $r_{T_{w_{in}}} = 60^\circ\text{C}$. In this way, it is possible to operate at the optimum operating point suggested by the manufacturer. Therefore, the performance of the controllers is evaluated on the rejection of disturbances (current demands). The heat demand is not evaluated because its influence on temperature changes has much slower dynamics than those produced by current changes. Additionally, its effect is filtered by the capacity of the secondary tank and the heat exchanger to the primary thermal circuit. Therefore, the most critical disturbance is the change in the current demand, and presumably, the disturbance produced by the demanded heat will be easily rejected with a valid controller for the rejection of the current effect.

The controllers obtained are evaluated by means of objectives that measure output errors and control efforts. The output errors and control effort objectives will presumably be in conflict. Therefore, it is valuable to study these as independent objectives to analyze the trade-off between each. However, the system consists of two outputs and two control actions. It seems reasonable to add the output error objectives (in both outputs) and control effort (in both control actions), as they have the same relative importance and equal magnitude. Thus, we simplify the optimization process and the decision stage. f_{11} is the average absolute error in stack temperature $T_{w_{out}}$, in $^\circ\text{C}$. The objective f_{12} is the average absolute error in stack inlet water temperature $T_{w_{in}}$, in $^\circ\text{C}$. The objective f_{21} is the average absolute value of the rate of change of the control action F_{w1} , in $(\text{l/min})/\text{s}$. The objective f_{22} is the average absolute value of the rate of change of the control action F_{w2} , in $(\text{l/min})/\text{s}$. The design objectives have been defined as integrals divided by the time to obtain an average measurement. In this way, the objectives have some physical sense, and their interpretation can be easier for the designer. Furthermore, the greater physical sense enables the designer to define the epsilon parameter (maximum

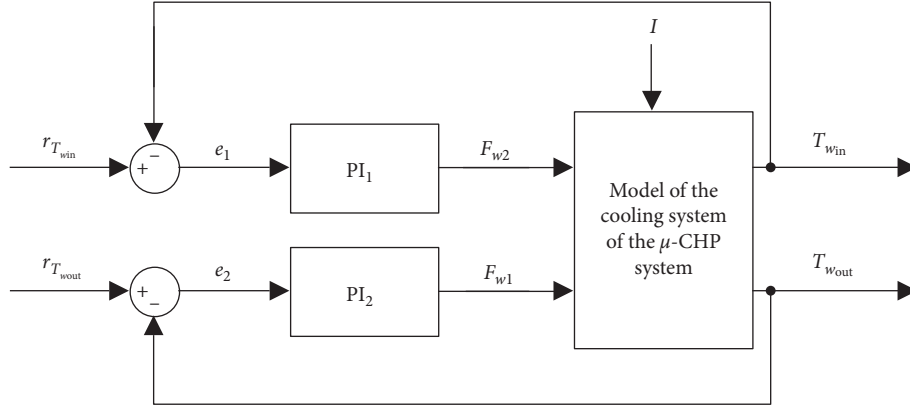


FIGURE 6: PI controller pairing scheme used to control the cooling system of the μ -CHP system.

degradation over design objectives) in a simpler way. Therefore, the first objective f_1 is defined as the aggregation of the average error in both outputs ($T_{w,out}$ and $T_{w,in}$). The second objective f_2 is defined as the aggregation of the integral of the control action derivative in both control actions (F_{w1} and F_{w2}). Therefore, the objective f_1 measures the rejection of disturbances, while f_2 measures the control effort. The MOP is defined as

$$\min_{\mathbf{x}} \mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}) \quad f_2(\mathbf{x})], \quad (8)$$

where

$$\begin{aligned} f_1(\mathbf{x}) &= f_{11}(\mathbf{x}) + f_{12}(\mathbf{x}), \\ f_2(\mathbf{x}) &= f_{21}(\mathbf{x}) + f_{22}(\mathbf{x}), \end{aligned} \quad (9)$$

where

$$\begin{aligned} f_{11}(\mathbf{x}) &= \frac{\int_{t_i}^{t_f} |e_1| dt}{t_f - t_i}, \\ f_{12}(\mathbf{x}) &= \frac{\int_{t_i}^{t_f} |e_2| dt}{t_f - t_i}, \\ f_{21}(\mathbf{x}) &= \frac{\int_{t_i}^{t_f} |dF_{w2}/dt| dt}{t_f - t_i}, \\ f_{22}(\mathbf{x}) &= \frac{\int_{t_i}^{t_f} |dF_{w1}/dt| dt}{t_f - t_i}, \end{aligned} \quad (10)$$

subject to

$$\begin{aligned} \underline{\mathbf{x}} &\leq \mathbf{x} \leq \bar{\mathbf{x}}, \\ f_j(\mathbf{x}) &< 0.375, \quad \text{for } j = 11, 12, \\ f_j(\mathbf{x}) &< 0.01, \quad \text{for } j = 21, 22, \end{aligned} \quad (11)$$

where

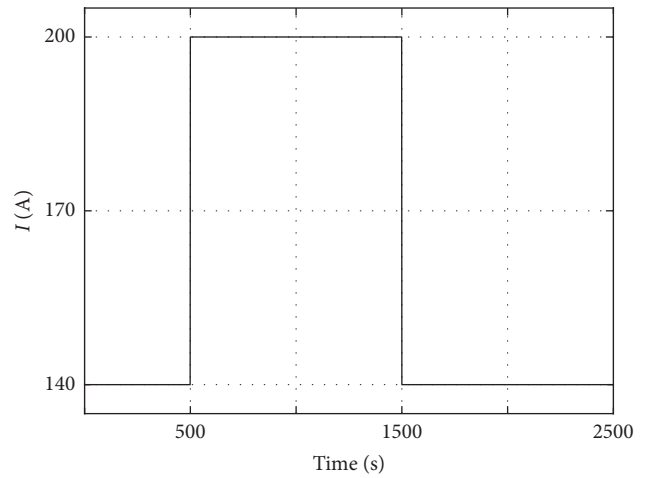


FIGURE 7: Current applied for the design of controller for the PEMFC cooling system.

$$\begin{aligned} \mathbf{x} &= [Kc_1, Ti_1, Kc_2, Ti_2] \\ \underline{\mathbf{x}} &= [-10, 1, -10, 1], \\ \bar{\mathbf{x}} &= [-0.1, 250, -0.1, 250], \\ t_i &= 500 \text{ sec}, \\ t_f &= 2500 \text{ sec}. \end{aligned} \quad (12)$$

The constraints (equation (11)) have been chosen to obtain the set of solutions in the designer's region of interest. Thus, we achieve an improvement in the pertinency of the set, discarding undesirable solutions. Furthermore, the bounds of the decision space (equation (12)) have been defined to find practical/realizable controllers.

Once the optimization problem is defined, the two main parameters of nevMOGA (ϵ and \mathbf{n}) must be defined. In this MOP, the design objectives make physical sense (f_1 measures the average error in $^{\circ}\text{C}$ and f_2 measures the average variations in control actions). Therefore, it is easier to define the ϵ parameter (maximum acceptable degradation). ϵ_1 and ϵ_2 maintain the units of the design objectives f_1 and f_2 respectively. In this problem, $\epsilon_1 = 0.05^{\circ}\text{C}$ and

$\varepsilon_2 = 0.0125$ (l/min)/sec have been defined for f_1 and f_2 , respectively, ($\varepsilon = [0.05 \ 0.0125]$). We then chose $\mathbf{n} = [0.5 \ 10 \ 0.5 \ 10]$ (neighborhood) based on the previously defined search space (approximately 4–5% of the search space). To optimize the defined MOP, nevMOGA is used with the following configuration:

- (i) $\text{Nind}_{GA} = 16$
- (ii) $\text{Nind}_p = 250$
- (iii) Generations = 2500
- (iv) $\mathbf{n_box} = [20 \ 20]$

These parameters have been defined to obtain an adequate distribution in the objective space (divisions for each dimension, parameter $\mathbf{n_box}$), a sufficient number of new candidate solutions (Nind_{GA} and Generations), and an adequate number of individuals Nind_p of the population $P(t)$ (population to explore the search space). For the definition of the remaining parameters, the values suggested in [41] for the original algorithm (ev-MOGA) are used.

Figure 8 shows the discrete set $P_{Q,n}^*$ obtained by nevMOGA. In the figure, to show the decision variables, we use the level diagram (LD (available in Matlab Central: <https://es.mathworks.com/matlabcentral/fileexchange/62224-interactive-tool-for-decision-making-in-multiobjective-optimization-with-level-diagrams>) [42, 43]) visualization tool, using 2-norm ($\|\cdot\|_2$). The objective space is shown in an x - y graph because it is an MOP with only two objectives, and so the trade-off on the design objectives can be analyzed in a simpler way than if we use LD on the objectives space. As seen in the figure, nevMOGA has been able to find a large number of nearly optimal controllers nondominated in their neighborhood with similar performances to the optimal ones. These controllers provide the designer with valuable information to make a final decision with still greater criterion—as will be shown below.

In all MOPs, the final decision is always a subjective decision based on the designer's preferences, knowledge, and previous experience. All the optimal solutions are equally valid but have different trade-offs between the design objectives. In this paper, we choose different optimal alternatives with different trade-offs (different areas of the Pareto front) that could be chosen by the designer (depending on his/her preferences). In this way, we validate the methodology independently of the specific preferences of the designer in question. There is no unique procedure for the designer's final decision. However, our procedure consists of the following: (1) we choose an optimal solution in a certain area of interest for the designer and (2) we select significantly different solutions but with similar performance in the design objectives. On these alternatives, we analyze new indicators not included in the design objectives. Thus, significantly different solutions with similar performance in the design objectives obtain a significant improvement in the new indicators analyzed with respect to the initially chosen optimal solution. This is valuable information for the designer before the final decision.

To carry out the analysis before the final decision, we chose three optimal controllers (\mathbf{x}^{A1} , \mathbf{x}^{B1} , and \mathbf{x}^{C1}) in three

different zones in the objective space. \mathbf{x}^{A1} is a fast controller, that is, with little error in the output in exchange for aggressive control action. Controller \mathbf{x}^{C1} is slow, that is, it produces more output error in exchange for smooth control action. Finally, \mathbf{x}^{B1} is a balanced controller. In addition, we choose the nearly optimal controllers \mathbf{x}^{A2} , \mathbf{x}^{B2} , and \mathbf{x}^{C2} . These controllers obtain a similar performance to \mathbf{x}^{A1} , \mathbf{x}^{B1} , and \mathbf{x}^{C1} respectively, being significantly different in their parameters.

Let us now look at the fast controllers (\mathbf{x}^{A1} and \mathbf{x}^{A2} , see Table 3). Figure 9 shows the behavior of the system with both controllers. The objective values of both controllers are observed in Table 4. The error in $T_{w_{in}}$ (f_{11}) is greater for the nearly optimal controller \mathbf{x}^{A2} . However, in the output $T_{w_{out}}$ (f_{12}), the opposite occurs, and the optimal controller \mathbf{x}^{A1} produces a greater error. With respect to the control effort, \mathbf{x}^{A1} controller is softer for F_{w1} (f_{22}) and more aggressive for F_{w2} (f_{21}) than the controller \mathbf{x}^{A2} (see Table 4). Therefore, although \mathbf{x}^{A1} slightly dominates \mathbf{x}^{A2} , there is no significant improvement (small behavioral differences). So, we look for new indicator to make a better informed final decision.

To study the selected controllers, we will make an analysis in four different scenarios:

- (i) Increasing the sample time (initially $T_s = 0.1$ seconds).
- (ii) Electrical degradation of the fuel cell.
- (iii) Changes in the noise introduced in the system outputs (change in the seed, that is, in the sequence of noise values or in the amplitude).
- (iv) Uncertainty in the model.

The choice of the sample time may vary the set of optimal controllers (Pareto front). The sample time refers to the controller sample time (it is a parameter of the real-time implementation). Therefore, we will analyze how the increase of this parameter affects the performance of the controllers obtained (through the design objectives). Suppose we increase the sample time to $T_s = 0.4$ seconds. Considering the dynamics of the process, this sample time is still perfectly valid. In this scenario, the nearly optimal controller \mathbf{x}^{A2} dominates the optimal controller \mathbf{x}^{A1} (the dominance is reversed). The objective values in this new scenario are observed in Table 5.

The electrical power provided by the stack also depends on its degradation. The manufacturer provides a voltage-current characteristic curve for certain stack operating temperatures. These curves are valid at the beginning of the stack life. However, after hours of operation, the electrical power generated by the stack decreases. The characteristic curve degrades, providing less voltage with the same demanded current, which translates into less electrical energy [44, 45]. For example, in [45], it is said that the degradation of the stack is approximately 0.3–3% in the voltage provided for every 1000 hours of operation. Therefore, in this analysis, we will study how this degradation of the stack affects the \mathbf{x}^{A1} and \mathbf{x}^{A2} controllers. This study is carried out by reducing a percentage (degradation) of the voltage provided by the stack. With a degradation greater than or

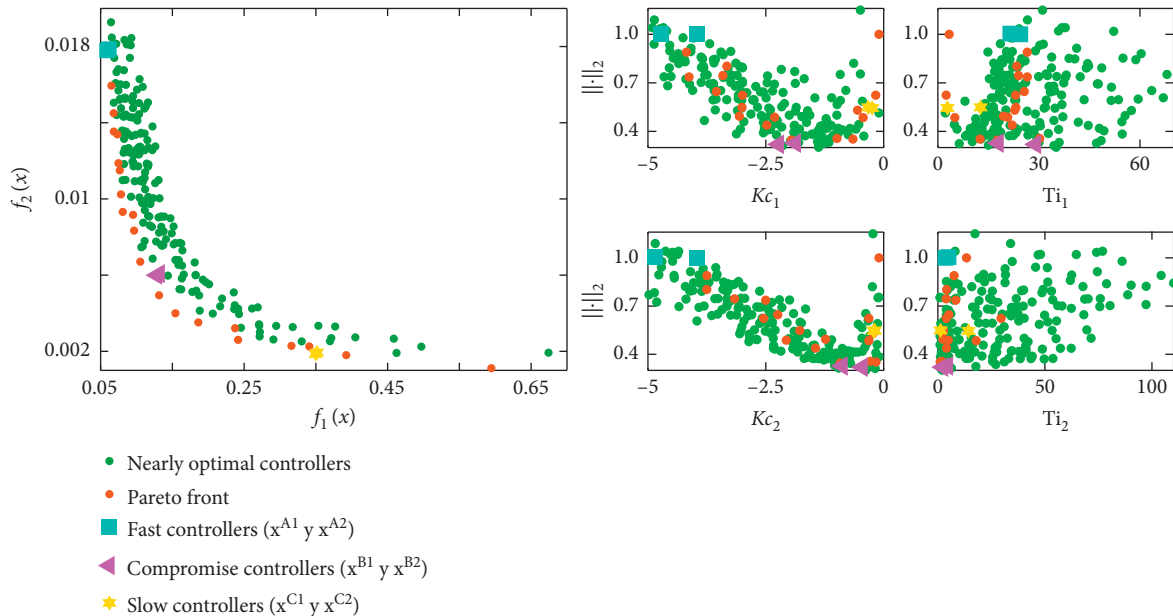


FIGURE 8: Set of optimal and nearly optimal controllers nondominated in the neighborhood ($P_{Q,n}$) obtained for the control design of the cooling system. Decision variables are shown using the LD visualization tool, using the 2-norm ($\|\cdot\|_2$).

TABLE 3: Fast (x^{A1} y x^{A2}), compromise (x^{B1} y x^{B2}), and slow controllers (x^{C1} y x^{C2}).

Controller	Kc_1	Ti_1	Kc_2	Ti_2
x^{A1}	-4.73	24.47	-3.95	3.55
x^{A2}	-3.97	21.51	-4.85	4.96
x^{B1}	-2.23	28.98	-0.46	2.18
x^{B2}	-1.87	17.87	-0.88	4.45
x^{C1}	-0.24	2.86	-0.21	14.31
x^{C2}	-0.33	12.67	-0.19	1.23

equal to 6%, the optimal controller x^{A1} no longer dominates the nearly optimal controller x^{A2} (see Table 5). In this scenario, the x^{A2} controller provides softer control actions. Therefore, after hours of stack operation, the nearly optimal controller x^{A2} is not worse (the controller with the highest objective value being understood as worse) than the optimal controller x^{A1} .

The output temperatures of the real system have noise. This noise is filtered to prevent it from spreading to system control actions. This noise has been included in the simulated system using white noise. However, the noise introduced has a random component depending on its seed. This value modifies the sequence of the noise values at each moment, maintaining its nature and amplitude. Therefore, in this analysis, we will study how the random component of noise affects the performance of the controllers (changing the seed). Suppose that the seed of the noise is randomly modified. In this new scenario, the nearly optimal controller x^{A2} dominates the optimal controller x^{A1} (the dominance is reversed, see Table 5). Therefore, in this new scenario (just as valid as the initial scenario for the designer), the x^{A2} controller could be

optimal (and x^{A1} nearly optimal), and therefore, both controllers are equally good.

The objective value obtained depends on the noise seed. A statistical study of this aspect is then appropriate. This analysis can be made by performing more simulations with different noise seeds. This analysis can be carried out in the decision stage only for the optimal and nearly optimal solutions (much less expensive than making the analysis in the optimization stage). This analysis has been carried out on controllers x^{A1} and x^{A2} . The controllers have been evaluated on 250 randomly obtained noise seeds. 10% of the noise seeds cause the dominance to be reversed, that is, the x^{A2} controller dominates the x^{A1} controller. In addition, the dominance between both controllers disappears with 17% of the seeds. In this analysis, x^{A1} is a more robust controller for seed changes. Despite this, it is shown that changes in the noise seed can cause a change in the Pareto set obtained.

Finally, we will analyze how the uncertainty of the model affects the controllers studied. To do this, we made 50 variations of the parameter's value of the model. A variation of $\pm 15\%$ is carried out on all the parameters of the model (30 parameters described in [35]). The controllers will be evaluated through the design objectives on each of the 50 variations of the model. In this way, we measure the robustness of the controllers.

We analyze how the uncertainty of the model affects the controllers studied. To do this, we made 50 variations of the parameters of the model with a variation of $\pm 15\%$. This variation is carried out on all the parameters of the model (30 parameters described in [35]). The controllers will be evaluated through the design objectives on each of the 50 variations of the model. In this way, we measure the robustness of the controllers.

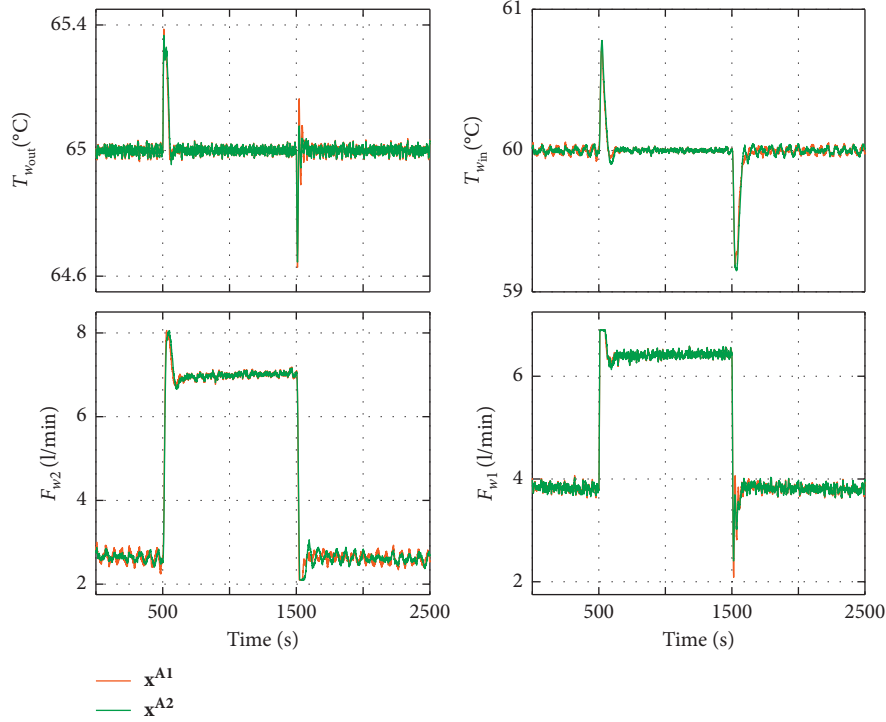


FIGURE 9: Response of \mathbf{x}^{A1} (optimal) and \mathbf{x}^{A2} (nearly optimal) controllers. Although $f(\mathbf{x}^{A1}) < f(\mathbf{x}^{A2})$, there is no clear advantage of the \mathbf{x}^{A1} controller over \mathbf{x}^{A2} in their response.

TABLE 4: Objective value of the fast, compromise, and slow controllers.

Cont.	$f_{11}(\mathbf{x})$	$f_{12}(\mathbf{x})$	$f_1(\mathbf{x})$	$f_{21}(\mathbf{x})$	$f_{22}(\mathbf{x})$	$f_2(\mathbf{x})$
\mathbf{x}^{A1}	0.042	0.017	0.059	0.0095	0.0083	0.0178
\mathbf{x}^{A2}	0.046	0.016	0.062	0.0081	0.0098	0.0179
\mathbf{x}^{B1}	0.083	0.045	0.128	0.0047	0.0013	0.0060
\mathbf{x}^{B2}	0.089	0.043	0.132	0.0040	0.0021	0.0061
\mathbf{x}^{C1}	0.14	0.21	0.35	0.0012	0.00062	0.0018
\mathbf{x}^{C2}	0.25	0.10	0.35	0.0011	0.00089	0.0019

TABLE 5: Objective values of \mathbf{x}^{A1} and \mathbf{x}^{A2} controllers in different scenarios.

Scenario	Controller	$f_1(\mathbf{x})$	$f_2(\mathbf{x})$
Ts = 0.4 seconds	\mathbf{x}^{A1}	0.0654	0.409
	\mathbf{x}^{A2}	0.0613	0.396
Degradation of the stack by 6%	\mathbf{x}^{A1}	0.0614	0.175
	\mathbf{x}^{A2}	0.0637	0.174
Change in the seed of the noise	\mathbf{x}^{A1}	0.0651	0.18
	\mathbf{x}^{A2}	0.063	0.179

Figure 10 shows the degradation limits of the \mathbf{x}^{A1} and \mathbf{x}^{A2} controllers over the 50 variations of the plant. The degradation of \mathbf{x}^{A2} is practically included within the degradation limits of \mathbf{x}^{A1} . Therefore, the nearly optimal controller \mathbf{x}^{A2} is more robust and shows less variability due to uncertainty in the model parameters.

Therefore, after studying the fast controllers (\mathbf{x}^{A1} and \mathbf{x}^{A2}), \mathbf{x}^{A2} may be preferred by the designer. In different scenarios (sample time, stack degradation, random noise change, and robustness analysis), this controller produces a

better (or equal) performance than \mathbf{x}^{A1} . Thus, in this case, considering the nearly optimal controller \mathbf{x}^{A2} has been very useful for the designer. Not considering this controller (obtaining only the optimal controllers) would mean ignoring relevant information that enables the designer to make a more informed decision.

Let us now analyze the compromise controllers (\mathbf{x}^{B1} and \mathbf{x}^{B2} in Figure 8). Figure 11 shows the system responses for both controllers. The controllers and their objective values are shown in Tables 3 and 4, respectively. The optimal controller \mathbf{x}^{B1} shows a smaller error on $T_{w_{in}}$ (f_{11}) with more aggressive control actions on F_{w2} (f_{21}). However, the nearly optimal controller \mathbf{x}^{B2} shows smaller errors on $T_{w_{out}}$ (f_{12}) with more aggressive control actions on F_{w1} (f_{22}). Again, although \mathbf{x}^{B1} dominates slightly \mathbf{x}^{B2} , no clear improvement is seen in their response, and therefore, it may be useful to evaluate their performance on the alternative indicators in order to make a more informed final decision.

Firstly, as above, the effect of increasing the sample time is studied. If the sample time is increased to $T_s = 0.4$

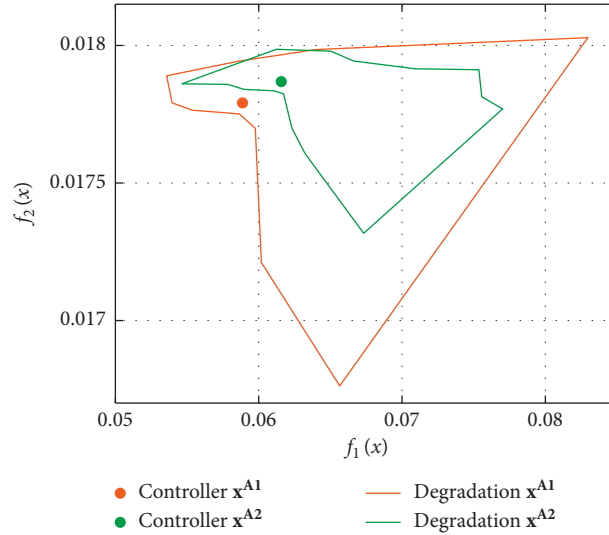


FIGURE 10: Degradation limits of x^{A1} (optimal) and x^{A2} (nearly optimal) controllers due to uncertainty in the model. The model parameters are varied by $\pm 15\%$.

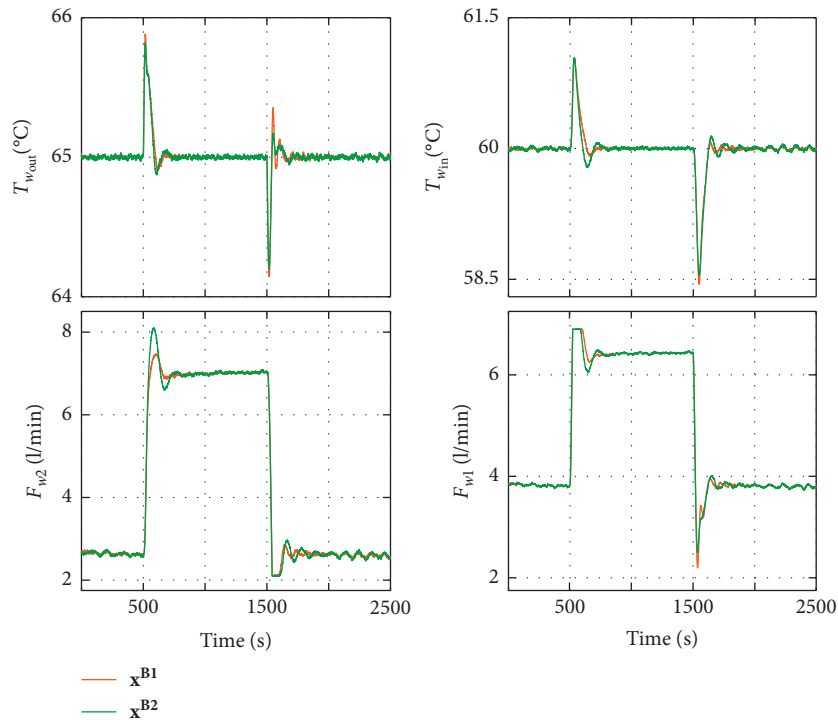


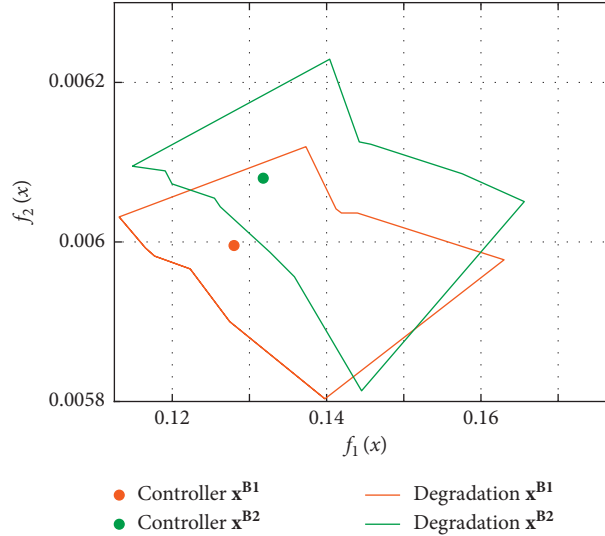
FIGURE 11: Response of x^{B1} (optimal) y x^{B2} (nearly optimal) controllers. Although $f(x^{B1}) < f(x^{B2})$, there is no clear advantage of the x^{B1} controller over x^{B2} in their response.

seconds, the optimal controller x^{B1} still dominates the nearly optimal controller x^{B2} (see Table 6). Secondly, we analyze how the degradation of the stack affects the controllers studied. In this case, with a degradation greater than or equal to 8%, x^{B2} dominates x^{B1} (the dominance is reversed, see Table 6). Therefore, after hours of stack operation, the nearly optimal controller x^{B2} is better than the optimal controller x^{B1} . Thirdly, we analyze the influence of the noise introduced in the system outputs on the controllers studied. Suppose we slightly decrease the amplitude of the noise introduced on

$T_{w_{out}}$. In this scenario, x^{B1} does not dominate x^{B2} (see Table 6). Similarly, if we slightly increase the amplitude of the noise introduced on $T_{w_{in}}$, again, the x^{B2} controller is not dominated by x^{B1} . Finally, we analyze the robustness of the compromise controllers (performed in the same way as in the previous comparison). In this situation, the optimal controller x^{B1} seems more robust than the nearly optimal controller x^{B2} (see Figure 12). Therefore, after making this detailed study of the compromise controllers (x^{B1} and x^{B2}), we can conclude that x^{B2} , in certain scenarios, is just as good

TABLE 6: Objective values of \mathbf{x}^{B1} and \mathbf{x}^{B2} controllers in different scenarios.

Scenario	Controller	$f_1(\mathbf{x})$	$f_2(\mathbf{x})$
Ts = 0.4 seconds	\mathbf{x}^{B1}	0.127	0.146
	\mathbf{x}^{B2}	0.134	0.148
Degradation of the stack by 8%	\mathbf{x}^{B1}	0.152	0.0563
	\mathbf{x}^{B2}	0.152	0.0546
Decrease noise over $T_{w_{\text{out}}}$	\mathbf{x}^{B1}	0.127	0.0585
	\mathbf{x}^{B2}	0.131	0.0587

FIGURE 12: Degradation limits of \mathbf{x}^{B1} (optimal) y \mathbf{x}^{B2} (nearly optimal) controllers due to uncertainty in the model. The model parameters are varied by $\pm 15\%$.

as the optimal controller \mathbf{x}^{B1} . However, in this case, the preference for the nearly optimal controller \mathbf{x}^{B2} over \mathbf{x}^{B1} is unclear. Again, this analysis is very valuable for the designer before making the final decision.

Let us now study the slow controllers (\mathbf{x}^{C1} and \mathbf{x}^{C2} , see Figure 8). The objective values of both controllers are observed in Table 4. Figure 13 shows the outputs and control actions of both controllers. Both controllers have a significantly different behavior despite having a similar performance (objective values). Therefore, it does not seem reasonable to select the final solution only from the results of the optimization (obtained for a specific scenario). It seems reasonable to let the designer choose between these solutions by consulting additional information that has not been taken into account in the optimization. The error in $T_{w_{\text{in}}}$ is greater for the nearly optimal controller \mathbf{x}^{C2} (f_{11}). However, for the output $T_{w_{\text{out}}}$, the opposite occurs, and the optimal controller \mathbf{x}^{C1} has a greater error (f_{12}). With respect to the control effort, the \mathbf{x}^{C1} controller is smoother for F_{w1} (f_{22}) and more aggressive for F_{w2} (f_{21}) than the \mathbf{x}^{C2} controller (see Table 4). Therefore, \mathbf{x}^{C1} slightly dominates \mathbf{x}^{C2} , but significant differences in their behavior invite us to conduct a deeper study of both controllers with the goal of making a more informed decision.

Firstly, we analyze how the increase of the sample time affects the performance of the controllers obtained,

analogously to the study of previous controllers. Again, suppose we increase the sample time to $T_s = 0.4$ seconds. In this scenario, the optimal controller \mathbf{x}^{C1} does not dominate the nearly optimal controller \mathbf{x}^{C2} (see Table 7). Secondly, we analyze how the degradation of the stack affects the slow controllers. In this study, \mathbf{x}^{C2} is dominated by \mathbf{x}^{C1} independently of the degradation of the stack (see Table 7). Therefore, the degradation of the stack does not affect the dominance of these controllers. Thirdly, suppose that the seed of the noise is randomly modified. In this scenario, \mathbf{x}^{C1} does not dominate \mathbf{x}^{C2} (see Table 7). Therefore, in another scenario (just as valid as the initial scenario, for the designer), the \mathbf{x}^{C2} controller could be optimal, and therefore, both controllers can be equally good. The same study made on the controllers \mathbf{x}^{A1} and \mathbf{x}^{A2} is carried out with controllers \mathbf{x}^{C1} and \mathbf{x}^{C2} . In this case, 55% of the noise seeds (of 250 seeds randomly obtained) make the dominance disappear and there is no seed that reverses dominance. Thus, it is shown that varying the noise seed varies the Pareto set obtained. Finally, we analyze the robustness of the slow controllers (in the same way as in the previous comparisons). In this context, \mathbf{x}^{C1} seems to be a more robust controller than \mathbf{x}^{C2} (see Figure 14). Therefore, after this deep study of the slow controllers, the nearly optimal controller \mathbf{x}^{C2} could be optimal in another scenario. In fact, in certain scenarios, both controllers can be considered equally good. Again,

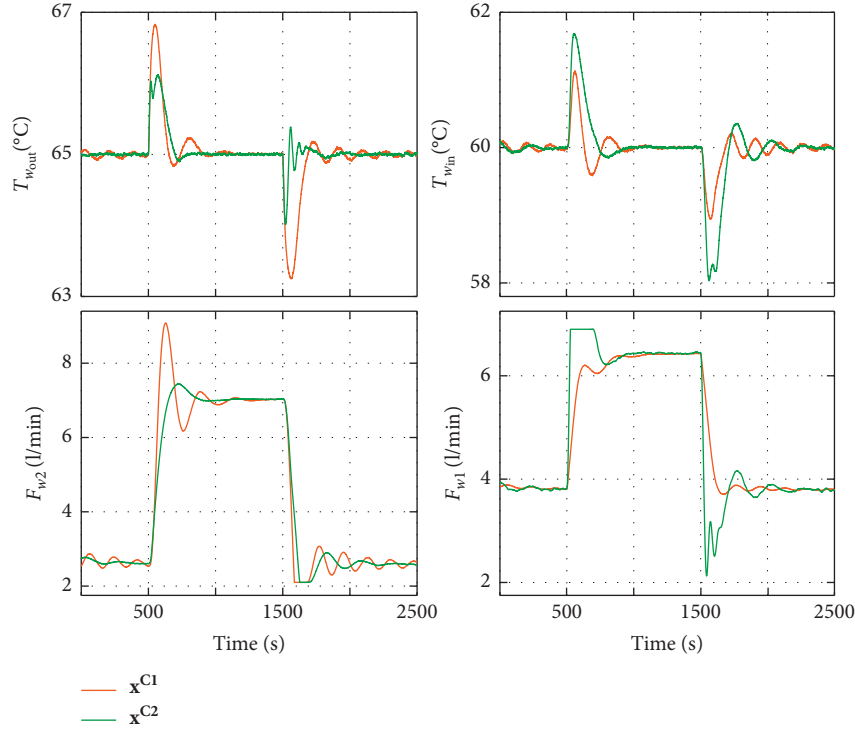


FIGURE 13: Response of \mathbf{x}^{C1} (optimal) and \mathbf{x}^{C2} (nearly optimal) controllers. Both responses are significantly different, although it has similar performance.

TABLE 7: Objective values of \mathbf{x}^{C1} and \mathbf{x}^{C2} controllers in different scenarios.

Scenario	Controller	$f_1(\mathbf{x})$	$f_2(\mathbf{x})$
Ts = 0.4 seconds	\mathbf{x}^{C1}	0.351	0.0626
	\mathbf{x}^{C2}	0.351	0.0636
Degradation of the stack by 2%	\mathbf{x}^{C1}	0.351	0.019
	\mathbf{x}^{C2}	0.363	0.0196
Change in the seed of the noise	\mathbf{x}^{C1}	0.354	0.0187
	\mathbf{x}^{C2}	0.352	0.0194

this information is very useful for the designer before making a final decision.

Thus, in this section, a deep study has been carried out for the design of the control of the inlet and outlet water temperatures of the PEMFC stack. Using the methodology raised, the utility of considering nearly optimal controllers nondominated in their neighborhood has been demonstrated. These controllers could be equal (or better) than the optimal ones in different scenarios (sample time, stack degradation, noise change, and robustness analysis). Analyzing all these scenarios in the design objectives is unapproachable. However, the designer can analyze them at the decision stage. In this context, the nearly optimal solutions nondominated in their neighborhood are very relevant alternatives for the designer. Thanks to obtaining these alternatives, controllers that are significantly different to the optimal ones have been obtained. Due to this difference (in parameters), some of them show improvements in different scenarios not studied in the optimization stage. Thanks to the analysis made in the decision stage, the designer can

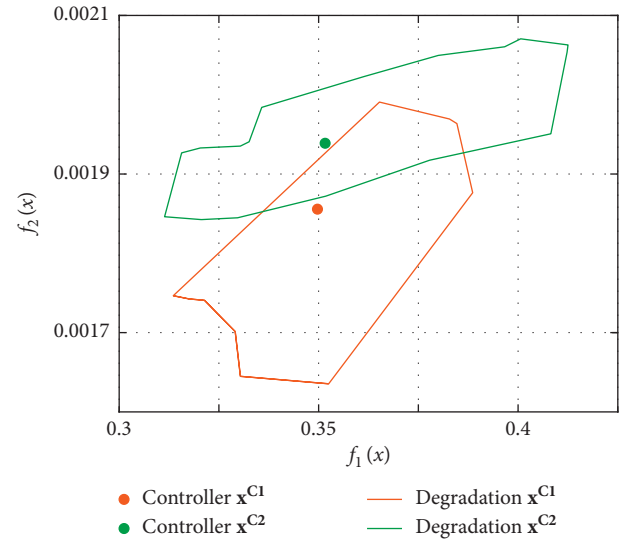


FIGURE 14: Degradation limits of \mathbf{x}^{C1} (optimal) and \mathbf{x}^{C2} (nearly optimal) controllers due to uncertainty in the model. The model parameters are varied by $\pm 15\%$.

carry out the final decision with valuable additional information.

5. Conclusions

In this paper, the design of the multivariable control system for the cooling circuit of a PEMFC stack has been shown. This system is complex and challenging enough: interactions

between variables, highly nonlinear dynamic behavior, etc. In the design, in addition to the set of optimal controllers under a multiobjective approach, the set of nearly optimal controllers potentially useful has been taken into account. Various aspects are analyzed on the potentially useful solutions, for example, the influence of certain parameters of the simulator on the Pareto front. This is an aspect little considered in the literature, and in this work, it has been shown how this can vary the Pareto set obtained (for instance, when the seed of the noise is changed). In addition, the effect of fuel cell degradation and robustness have also been analyzed. As observed in this work, this could be valuable information for the designer before the final decision. Therefore, this document highlights the usefulness of this methodology in the in-depth analysis of several of the aspects that can influence the tuning of a control structure, especially in the design of the multivariable control of complex systems such as the cooling system of a PEM fuel cell.

Including all the interesting aspects for the designer in the optimization stage is generally computationally unapproachable and complicates analysis of the results. However, in this work, we have analyzed these aspects (not contemplated in the optimization stage) in the decision making phase. Thus, the designer can consider these aspects without excessively increasing the computational cost of MOP resolution.

In this context, the diversity in the set of controllers obtained takes a still more relevant role. Different controllers can provide an improvement (even significant) in several interesting aspects not included in the MOP. Therefore, in this situation, nearly optimal controllers nondominated in their neighborhood play an essential role. These controllers provide the designer with solutions with similar performances to the optimal ones, but with significantly different characteristics. Consequently, these controllers provide greater diversity in the set, without excessively increasing the number of solutions obtained.

In this paper, we have analyzed various interesting scenarios for the designer using the obtained set of controllers $P_{Q,n}^*$. Firstly, with an increased sampling time, nearly optimal controllers can improve the optimal controllers; this means that the way the simulation itself is carried out can vary the result obtained in the resolution of the MOP. Secondly, we have analyzed how the degradation of the stack affects the performance of the controllers. In this scenario, nearly optimal controllers can again improve the optimal ones. The designer, with this analysis, can opt for nearly optimal controllers instead of controllers that are optimal only at the beginning of the stack's life. In addition, we have analyzed the effect of the seed used for the generation of the introduced noise. A change in this seed can cause the nearly optimal alternatives to improve the performance of the optimal ones. Generally, the seed of noise is a parameter not chosen by the designer but by the simulation tool. This implies that both scenarios (each with a different noise seed) are equally valid for the designer, and therefore, both types of controllers (optimal and nearly optimal) are also equally

valid. Finally, a robustness analysis of the controllers obtained has been carried out. An assessment of the impact of the uncertainties in the parameters of the nonlinear model has been carried out. In some cases, the nearly optimal solutions are more robust.

In summary, it is worth considering these additional solutions as they can provide new and perfectly valid design alternatives. This analysis has a computationally acceptable cost because it is only applied to a limited number of solutions—the set of optimal and nearly optimal controllers obtained. Its incorporation into the MOP statement would entail a high computational cost that may not be assumable. All of these analyses can move the designer's final choice towards a nearly optimal rather than an optimal controller.

Thanks to the methodology presented, the designer can make a final decision with additional valuable information (ignored in the classic MOP) by obtaining potentially useful new controllers. Therefore, this work has revealed the relevance of the nearly optimal alternatives nondominated in their neighborhood in the design of multivariable control systems. Given the usefulness of the approach proposed in this work on the controller design, as future work, we plan to use this methodology in system identification where nearly optimal models can be potentially useful for the designer.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This study was supported in part by the Ministerio de Ciencia, Innovación y Universidades (Spain) (grant no. RTI2018-096904-B-I00) and by the Generalitat Valenciana regional government through project AICO/2019/055.

References

- [1] R. C. Dorf and R. H. Bishop, *Modern Control Systems*, Pearson, Upper Saddle River, NJ, USA, 2011.
- [2] P. Mercader, C. Cánovas, and A. Baños, "Multivariable PID control of a boiler," *Revista Iberoamericana de Automática e Informática Industrial*, vol. 16, no. 1, pp. 15–25, 2019.
- [3] S. Skogestad and I. Postlethwaite, *Multivariable Feedback Control: Analysis and Design*, vol. 2, Wiley, New York, NY, USA, 2007.
- [4] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*, vol. 16, John Wiley & Sons, Hoboken, NJ, USA, 2001.
- [5] N. Gunantara, "A review of multi-objective optimization: methods and its applications," *Cogent Engineering*, vol. 5, no. 1, Article ID 1502242, 2018.

- [6] K. Miettinen, *Nonlinear Multiobjective Optimization*, vol. 12, Springer Science & Business Media, Berlin, Germany, 2012.
- [7] G. Reynoso-Meza, X. Blasco, J. Sanchis, and J. M. Herrero, *Controller Tuning with Evolutionary Multiobjective Optimization: A Holistic Multiobjective Optimization Design Procedure*, Springer, Berlin, Germany, 2017.
- [8] A. Engau and M. M. Wiecek, "Generating ϵ -efficient solutions in multiobjective programming," *European Journal of Operational Research*, vol. 177, no. 3, pp. 1566–1579, 2007.
- [9] P. Loridan, " ϵ -solutions in vector minimization problems," *Journal of Optimization Theory and Applications*, vol. 43, no. 2, pp. 265–276, 1984.
- [10] D. J. White, "Epsilon efficiency," *Journal of Optimization Theory and Applications*, vol. 49, no. 2, pp. 319–337, 1986.
- [11] A. Pajares, X. Blasco, J. M. Herrero, and G. Reynoso-Meza, "A multiobjective genetic algorithm for the localization of optimal and nearly optimal solutions which are potentially useful: nevMOGA," *Complexity*, vol. 2018, Article ID 1792420, 22 pages, 2018.
- [12] O. Schütze, M. Vasile, and C. A. C. Coello, "Approximate solutions in space mission design," in *Proceedings of the 2008 International Conference on Parallel Problem Solving from Nature*, pp. 805–814, Dortmund, Germany, September 2008.
- [13] O. Schütze, M. Vasile, and C. A. C. Coello, "Computing the set of epsilon-efficient solutions in multiobjective space mission design," *Journal of Aerospace Computing, Information, and Communication*, vol. 8, no. 3, pp. 53–70, 2011.
- [14] A. Pajares, X. Blasco, J. M. Herrero, and G. Reynoso-Meza, "A new point of view in multivariable controller tuning under multiobjective optimization by considering nearly optimal solutions," *IEEE Access*, vol. 7, pp. 66435–66452, 2019.
- [15] Y. K. Soni and R. Bhatt, "BF-PSO optimized PID controller design using ISE, IAE, IATE and MSE error criteria," *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, vol. 2, no. 7, pp. 2333–2336, 2013.
- [16] J. M. Hammersley, "Monte Carlo methods for solving multivariable problems," *Annals of the New York Academy of Sciences*, vol. 86, no. 3, pp. 844–874, 1960.
- [17] A. Fredriksson, A. Forsgren, and B. Hårdemark, "Minimax optimization for handling range and setup uncertainties in proton therapy," *Medical Physics*, vol. 38, no. 3, pp. 1672–1684, 2011.
- [18] J. Lee and G. E. Johnson, "Optimal tolerance allotment using a genetic algorithm and truncated Monte Carlo simulation," *Computer-Aided Design*, vol. 25, no. 9, pp. 601–611, 1993.
- [19] J. M. Andújar and F. Segura, "Fuel cells: history and updating. a walk along two centuries," *Renewable and Sustainable Energy Reviews*, vol. 13, no. 9, pp. 2309–2322, 2009.
- [20] V. Mehta and J. S. Cooper, "Review and analysis of PEM fuel cell design and manufacturing," *Journal of Power Sources*, vol. 114, no. 1, pp. 32–53, 2003.
- [21] A. De las Heras, F. J. Vivas, F. Segura, M. J. Redondo, and J. M. Andújar, "Air-cooled fuel cells: keys to design and build the oxidant/cooling system," *Renewable Energy*, vol. 125, pp. 1–20, 2018.
- [22] S. G. Kandlikar and Z. Lu, "Thermal management issues in a PEMFC stack—a brief review of current status," *Applied Thermal Engineering*, vol. 29, no. 7, pp. 1276–1280, 2009.
- [23] G. Petrecca, *Energy Conversion and Management*, Springer, Berlin, Germany, 2014.
- [24] Q. Yan, H. Toghiani, and H. Causey, "Steady state and dynamic performance of proton exchange membrane fuel cells (PEMFCs) under various operating conditions and load changes," *Journal of Power Sources*, vol. 161, no. 1, pp. 492–502, 2006.
- [25] R. Beith, *Small and Micro Combined Heat and Power (CHP) Systems: Advanced Design, Performance, Materials and Applications*, Elsevier, Amsterdam, Netherlands, 2011.
- [26] M. M. Maghanki, B. Ghobadian, G. Najafi, and R. J. Galogah, "Micro combined heat and power (MCHP) technologies and applications," *Renewable and Sustainable Energy Reviews*, vol. 28, pp. 510–524, 2013.
- [27] P. Martínez, M. Serra, and R. Costa-Castelló, "Modeling and control of HTPEMFC based combined heat and power for confort control," in *Proceedings of the 2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, IEEE, Limassol, Cyprus, September 2017.
- [28] D. A. Notter, K. Kouravelou, T. Karachalios, M. K. Daletou, and N. T. Haberland, "Life cycle assessment of PEM FC applications: electric mobility and μ -CHP," *Energy & Environmental Science*, vol. 8, no. 7, pp. 1969–1985, 2015.
- [29] S. Martinez, G. Michaux, P. Salagnac, and J.-L. Bouvier, "Micro-combined heat and power systems (micro-CHP) based on renewable energy sources," *Energy Conversion and Management*, vol. 154, pp. 262–285, 2017.
- [30] T. Elmer, M. Worall, S. Wu, and S. B. Riffat, "Fuel cell technology for domestic built environment applications: state of-the-art review," *Renewable and Sustainable Energy Reviews*, vol. 42, pp. 913–931, 2015.
- [31] A. Hawkes, I. Staffell, D. Brett, and N. Brandon, "Fuel cells for micro-combined heat and power generation," *Energy & Environmental Science*, vol. 2, no. 7, pp. 729–744, 2009.
- [32] H. R. Ellamla, I. Staffell, P. Bujlo, B. G. Pollet, and S. Pasupathi, "Current status of fuel cell based combined heat and power systems for residential sector," *Journal of Power Sources*, vol. 293, pp. 312–328, 2015.
- [33] S. Strahl and R. Costa-Castelló, "Temperature control of open-cathode PEM fuel cells," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 11088–11093, 2017.
- [34] G. Zhang and S. G. Kandlikar, "A critical review of cooling techniques in proton exchange membrane fuel cell stacks," *International Journal of Hydrogen Energy*, vol. 37, no. 3, pp. 2412–2429, 2012.
- [35] S. Navarro Giménez, J. M. Herrero Durá, F. X. Blasco Ferragud, and R. Simarro Fernández, "Control-oriented modeling of the cooling process of a PEMFC-based μ -CHP system," *IEEE Access*, vol. 7, pp. 95620–95642, 2019.
- [36] V. Pareto, *Manual of Political Economy*, Oxford University Press, Oxford, UK, 1971.
- [37] O. Schütze, C. A. C. Coello, and E.-G. Talbi, "Approximating the ϵ -efficient set of an MOP with stochastic search algorithms," in *Proceedings of the 2007 Mexican International Conference on Artificial Intelligence*, pp. 128–138, Guadalajara, México, 2007.
- [38] A. Pajares Ferrando, *Development of a multi-objective optimization methodology considering nearly optimal solutions. Application to problems in control engineering*, Ph.D. thesis, Departamento Ingeniería Sistemas Automática, Universitat Politècnica de València, Valencia, Spain, 2019.
- [39] J. M. Herrero, S. García-Nieto, X. Blasco, V. Romero-García, J. V. Sánchez-Pérez, and L. M. Garcia-Raffi, "Optimization of sonic crystal attenuation properties by ev-moga multi-objective evolutionary algorithm," *Structural and Multidisciplinary Optimization*, vol. 39, no. 2, pp. 203–215, 2009.

- [40] E. Bristol, "On a new measure of interaction for multivariable process control," *IEEE Transactions on Automatic Control*, vol. 11, no. 1, pp. 133-134, 1966.
- [41] J. M. Herrero Durá, *Identificación robusta de sistemas no lineales mediante algoritmos evolutivos*, Ph.D. thesis, Universitat Politècnica de València, Valencia, Spain, 2006.
- [42] X. Blasco, J. M. Herrero, G. Reynoso-Meza, and M. A. M. Iranzo, "Interactive tool for analyzing multiobjective optimization results with level diagrams," in *Proceedings of the 2017 Genetic and Evolutionary Computation Conference Companion*, pp. 1689-1696, Berlin, Germany, July 2017.
- [43] X. Blasco, J. M. Herrero, J. Sanchis, and M. Martínez, "A new graphical visualization of n -dimensional Pareto front for decision-making in multiobjective optimization," *Information Sciences*, vol. 178, no. 20, pp. 3908-3924, 2008.
- [44] W. Schmittinger and A. Vahidi, "A review of the main parameters influencing long-term performance and durability of PEM fuel cells," *Journal of Power Sources*, vol. 180, no. 1, pp. 1-14, 2008.
- [45] I. Staffell, *Fuel cells for domestic heat and power: are they worth it?* Ph.D. thesis, University of Birmingham, Birmingham, UK, 2010.