

Received December 1, 2020, accepted December 10, 2020, date of publication December 21, 2020, date of current version December 31, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3046352

# WH-MOEA: A Multi-Objective Evolutionary Algorithm for Wiener-Hammerstein System Identification. A Novel Approach for Trade-Off Analysis Between Complexity and Accuracy

J. ZAMBRANO<sup>1</sup>, (Member, IEEE), J. SANCHIS<sup>2</sup>, JUAN MANUEL HERRERO<sup>2</sup>, AND M. MARTÍNEZ<sup>2</sup>, (Member, IEEE)

<sup>1</sup>Grupo de Investigación en Interacción, Robótica y Automática (GIIRA), Universidad Politécnica Salesiana, Cuenca 010102, Ecuador  
<sup>2</sup>Instituto Universitario de Automática e Informática Industrial, Universitat Politècnica de València, 46022 Valencia, Spain

Corresponding author: J. Zambrano (jzambrano@ups.edu.ec)

This work was supported in part by the Ministerio de Ciencia, Innovación y Universidades, Spain, under Grant RTI2018-096904-B-I00-AR, and in part by the Salesian Polytechnic University of Ecuador through a Ph.D. scholarships granted to J. Zambrano.

**ABSTRACT** Several approaches have been presented to identify Wiener-Hammerstein models, most of them starting from a linear dynamic model whose poles and zeros are distributed around the static non-linearity. To achieve good precision in the estimation, the Best Linear Approximation (BLA) has usually been used to represent the linear dynamics, while static non-linearity has been arbitrarily parameterised without considering model complexity. In this paper, identification of Wiener, Hammerstein or Wiener-Hammerstein models is stated as a multiobjective optimisation problem (MOP), with a trade-off between accuracy and model complexity. Precision is quantified with the Mean-Absolute-Error (MAE) between the real and estimated output, while complexity is based on the number of poles, zeros and points of the static non-linearity. To solve the MOP, WH-MOEA, a new multiobjective evolutionary algorithm (MOEA) is proposed. From a linear structure, WH-MOEA will generate a set of optimal models considering a static non-linearity with a variable number of points. Using WH-MOEA, a procedure is also proposed to analyse various linear structures with different numbers of poles and zeros (known as design concepts). A comparison of the Pareto fronts of each design concept allows a more in-depth analysis to select the most appropriate model according to the user's needs. Finally, a complex numerical example and a real thermal process based on a Peltier cell are identified, showing the procedure's goodness. The results show that it can be useful to consider the simultaneously precision and complexity of a block-oriented model (Wiener, Hammerstein or Wiener-Hammerstein) in a non-linear process identification.

**INDEX TERMS** Wiener-Hammerstein identification, multiobjective optimisation, evolutionary algorithm, Pareto front.

## I. INTRODUCTION

Mathematical models are used in several engineering fields to describe the behaviour of dynamic systems. These models are based on a set of ordinary or partial differential equations [1], [2] associated with physical phenomena that occur in the system. However, it is not always possible in many practical applications to know the precise relationships that characterize a system. It is thus preferable to use simplified

models obtained experimentally from a set of input-output measurements. This way of building models is referred to in the literature as systems identification [3]–[5].

When the system to be modelled has non-linearities, the identification problem is not easy and selecting the model structure is the most challenging decision. When non-linearity is static, a good way to model nonlinear systems is through block-oriented models consisting of the interaction of linear time-invariant (LTI) dynamic subsystems and static nonlinear elements (NL) [6], [7]. The most common block-oriented models are the Wiener,

The associate editor coordinating the review of this manuscript and approving it for publication was Bing Li.

Hammerstein and Wiener-Hammerstein models. A Wiener model is formed by an LTI block followed by an NL block [8], while the permutation of these blocks results in a Hammerstein model [9], [10]. These two models are specific cases of the Wiener-Hammerstein structure in which the NL block is in the middle of two LTI blocks [11].

Several methods of identifying this type of models can now be found in the literature. For Wiener-Hammerstein models, the identification procedure generally begins with the estimation of the linear dynamics, and the main challenge is to split the LTI blocks in two. A non-exhaustive list of options includes, the “brute-force” method and the advanced method [12]; the scanning technique based on the estimation of the quadratic/cubic best linear approximation (QBLA/CBLA) [13]; the non-parametric approach based on the estimation of the QBLA [14]; fractional approach [15]; WH-EA: An Evolutionary Algorithm for Wiener-Hammerstein System Identification [16]; multi-stage identification [17], [18]; classification of poles and zeros using QBLA and manual tuning [19]; among others [20]–[23].

Most methods focus on obtaining nonlinear models with high precision using the BLA [7], [24], [25] as a starting point for nonlinear identification. However, it should be noted that the system under test must be excited with asymptotically normally distributed signals (Riemann equivalence class) to obtain the BLA [4], [27]. From a practical point of view, not all processes are enabled to handle this class of signals. An alternative approach based on a standard linear model and multi-step signals is proposed in [28].

The nature of block-oriented models forces the user to initially define the complexity of the dynamic part generally represented by some poles and zeros and also the complexity of the static nonlinear part. This is not an easy task, especially in the case of Wiener-Hammerstein models, where two LTI subsystems must be estimated. Some criteria are currently available such as Minimum Description Length (MDL) [30], Akaike information criterion (AIC) [29], Modified AIC and MDL criteria [31], and Final Prediction Error (FPE) [32], [33]. These help to define the complexity of the overall system dynamics. However, two or more different linear structures can give the same value to a specific criterion, or several criteria can offer opposite values for the same linear structure. To date, there is no recipe to precisely define the number of parameters to represent the static nonlinear part. This leads to several identification tests with different numbers of parameters until the desired accuracy is achieved. This task can be tedious and even worse if the linear structure is not well defined, since no matter how many parameters are added to the static non-linearity, the unmodelled dynamics will not be able to achieve good accuracy.

Of all these complications, the model’s applicability is an important issue when estimating Wiener-Hammerstein models. Although accuracy can be important when selecting a model, its complexity may also be a factor. For example, in control applications, such as Model Predictive Control

(MPC) [50], it is well known that controller performance will depend largely on the model quality. However, the higher the model complexity, the higher the computational cost required to calculate the control action. In terms of the control algorithm, there is a significant gap between the MPC based on Wiener models [51], [52] or Hammerstein models [53] and the MPC based on Wiener-Hammerstein models [54]. In this work we thus consider a trade-off analysis between accuracy and complexity in estimating the Wiener-Hammerstein model and the identification problem is stated as an MOP.

Unlike linear models, defining a suitable structure (complexity) in nonlinear models is not a simple task. For this reason, the use of multiobjective optimisation (MO) in nonlinear identification is not new. Thanks to MO, it is possible to generate a Pareto set to compare and analyse the complexity and accuracy of different models in order to represent the same process and avoid over-fitting problems. Applications of MO in nonlinear modelling includes the identification of: the Volterra series [43], radial-basis function (RBF) networks [44]–[46], Nonlinear Auto-Regressive eXogenous (NARX) models [47], [48], Nonlinear Auto-Regressive Moving Average models with eXogenous inputs (NARMAX) [49], and Wiener-Hammerstein models [26]. This last approach is based on genetic recombination and particle swarm optimisation. During the search process, the algorithm requires minimal user interaction, but even though a large number of poles and zeros are allowed for both LTI subsystems, good accuracy cannot be achieved since the optimisation problem does not use a linear approximation as a starting point. Furthermore, a polynomial is used for the static non-linearity, which is not recommended, since the sensitivity of the coefficients increases with the degree of the polynomial.

In this paper, a new Multiobjective Evolutionary Algorithm for Wiener-Hammerstein identification (WH-MOEA) is proposed based on *ev*-MOGA (epsilon-variable Multi-Objective Genetic Algorithm) [34] but using some new genetic operators and others inherited from WH-EA [16]. WH-MOEA genetic operators perform a smart distribution and fine-tune the linear dynamics while capturing the non-linearity. To tackle model complexity, two new genetic operators are incorporated into WH-MOEA to increase or decrease the number of points assigned to the static non-linearity.

The entire procedure includes the handling of several design concepts (i.e. alternative initial model structures with different number of poles and zeros) in separate optimisation trials. Nonlinear model complexity is thus handled naturally by the non-linearity with a variable number of points and by the different structures tested in several WH-MOEA runs. Due to the ad-hoc genetic operators of WH-MOEA, different initial linear structures can lead to nonlinear models that do not necessarily have a Wiener-Hammerstein structure, so that Wiener models and Hammerstein models can also be obtained. This may be attractive for the user since a Wiener-Hammerstein model obtained in an optimisation run can be compared to other Wiener or Hammerstein models from other optimisation tests. This procedure can give the

**TABLE 1. Features of the most relevant Wiener-Hammerstein model identification methods presented in the last decade.**

Identification method	Excitation signal	Initial model	linear	Two-step procedure	UIL	Additional requirements	Consider the model's complexity
Brute-force [12]	Gaussian	BLA		Yes	Very high		No
Advanced method [12]	Gaussian	BLA		No	High	Inverse of the BLA	No
Scanning technique [13]	Gaussian	BLA		Yes	High	QBL/CBLA	No
Biosocial culture [26]	Non-restricted	Not required		No	Low		Yes
Fractional approach [15]	Gaussian	BLA		Yes	Medium		No
Modified fractional approach [21]	Gaussian	BLA		Yes	Medium		No
Nonparametric QBLA [14]	Gaussian	BLA		Yes	High	QBLA	No
Multistage identification [17], [18]	Random sequence	No required		No	High		No
Classification of poles and zeros [19]	Gaussian	BLA		Yes	Very high	QBL/CBLA and manual tuning	No
<b>WH-MOEA</b>	<b>Multi-step</b>	<b>Standard model</b>		<b>No</b>	<b>Medium</b>		<b>Yes</b>

user a broader spectrum to decide on the best model. According to the available literature and estimation tools, this is currently only possible using a different method for each type of structure.

Compared with [26], this approach has two significant advantages. It uses a linear approximation as a starting point for nonlinear estimation. Although an additional step is required, knowledge of linear dynamics leads to better results. Also, different Pareto fronts (one for each initial linear structure selected) can be compared in the multiobjective space for an exhaustive trade-off analysis.

By way of summary, the novelty of this paper lies in two main aspects:

- 1) The development of a new multiobjective algorithm with specific genetic operators for the identification of Wiener-Hammerstein models and their specific cases, i.e., Wiener models and Hammerstein models.
- 2) Creation of a methodology that can compare different design concepts (alternatives to the dynamic part of the model) from an MO point of view (precision vs complexity) allowing the designer to analyse different model candidates in a more informed way and to choose the most suitable according to his/her preferences.

Thanks to this, the user will value the effect of adding or removing poles or zeros from a model. For example, a Wiener-Hammerstein model can be compared to models of similar structure but with a greater or lesser number of poles or zeros, or in case of subtracting complexity in the dynamic part, the model is no longer Wiener-Hammerstein and goes to a Wiener or a Hammerstein model. Comparing different structures can be interesting because differences in precision will justify whether or not to select the most complex structure. Also thanks to WH-MOEA, it will also be possible to compare several models with different complexity in the static nonlinear part.

WH-MOEA has certain advantages over other Wiener-Hammerstein identification methods. Table 1 summarises the most important features of several of the methods proposed in the last decade. As can be seen, most methods use BLA as a starting point for nonlinear estimation and are conditioned to use Gaussian excitation signals, and most methods have a high User Interaction Level (UIL). In these cases, user intervention is required to carry out intermediate procedures, analyse results and make decisions. This aspect is

even more critical in methods with a non-parametric approach or where the QBLA/CBLA is required.

In line with previous work [28], WH-MOEA uses multi-step signals, which, unlike Gaussian signals, are easy to design, and almost all processes are enabled to be excited by these signals. Even though estimation starts from a completely unknown static non-linearity, multi-step signals will help to highlight the effects of the non-linearity on the signals used for the identification process. On the other hand, WH-MOEA uses a standard linear model as a starting point for non-linear estimation. This model can easily be obtained from the step response, which is more practical, especially in industrial processes. Rigorous BLA estimation may require the use of more than one Gaussian excitation and each signal must be carefully designed with appropriate frequency content.

Regarding the UIL, WH-MOEA does not require intermediate procedures since the genetic operators are enabled to distribute the dynamics of the initial linear model when fine-tuning the location of the poles and zeros and capturing nonlinearity. Finally, assuming that the process under test is affected by a static non-linearity, the user does not need to enter the structure to be identified in the algorithm. WH-MOEA will find a set of models, which may have a Wiener, Hammerstein, or Wiener-Hammerstein structure.

The rest of this paper is organised as follows. In Section 2 the Wiener-Hammerstein formulation is presented together with the necessary assumptions, after which the multiobjective optimisation problem is stated. In Section 3 the evolutionary algorithm (WH-MOEA) is described in detail. Next, a guide is given for the application of this approach together with a summary of WH-MOEA parameters in Section 4. In Section 5 the application of this approach and results are presented, while the concluding remarks are reported in Section 6.

## II. WIENER-HAMMERSTEIN FORMULATION AND PROBLEM STATEMENT

### A. WIENER-HAMMERSTEIN SYSTEM

A Wiener-Hammerstein model is formed by a static nonlinear block ( $f(v(t), \rho_{nl})$ ) sandwiched between two LTI blocks  $G_w(s, \rho_w)$  and  $G_h(s, \rho_h)$ . The structure of a Wiener-Hammerstein model is shown in Figure 1, where  $u(t)$  and  $y(t)$  are the model input and output, respectively, while  $v(t)$  and  $w(t)$  are internal variables that cannot usually

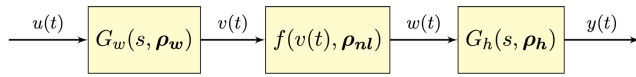


FIGURE 1. Wiener-Hammerstein model structure.

be measured. In general terms, a Wiener-Hammerstein model can be mathematically described by:

$$y(t, \rho) = G_h(s, \rho_h) f(G_w(s, \rho_w) u(t), \rho_{nl}) \quad (1)$$

$$\rho = [\rho_w, \rho_{nl}, \rho_h] \quad (2)$$

where vectors  $\rho_w$  and  $\rho_h$  contain the parameters of  $G_w$  and  $G_h$ , respectively, while vector  $\rho_{nl}$  contains the static non-linearity parameters. Let the LTI subsystems be represented as rational transfer functions in a continuous-time domain in factorised form (zero-pole-gain):

$$G_w(s) = \frac{\prod_{i=1}^{n_b} (s + z_{wi}) / (z_{wi})}{\prod_{i=1}^{n_a} (s + p_{wi}) / (p_{wi})} \quad (3)$$

$$G_h(s) = \frac{\prod_{i=1}^{n_d} (s + z_{hi}) / (z_{hi})}{\prod_{i=1}^{n_c} (s + p_{hi}) / (p_{hi})} \quad (4)$$

while static non-linearity is represented by a piece-wise linear function:

$$w(t) = f(v(t), \rho_{nl}). \quad (5)$$

In (3),  $-p_{wi}$  with  $i = 1 \dots n_a$  and  $-z_{wi}$  with  $i = 1 \dots n_b$  represent the poles and zeros of the front subsystem  $G_w$ , while in (4),  $-p_{hi}$  with  $i = 1 \dots n_c$  and  $-z_{hi}$  with  $i = 1 \dots n_d$  represent those of the  $G_h$  block. In both LTI subsystems  $s$  is the complex Laplace variable. Notice that static gains of both LTI subsystems have been normalized to 1 since the nonlinear block will absorb the real gains. On the other hand, in (5)  $\rho_{nl}$  contains the abscissas and ordinates that define the breakpoint locations of the piece-wise linear function. Under this configuration, vectors  $\rho_w$ ,  $\rho_h$ , and  $\rho_{nl}$  would be defined as follows:

$$\rho_w = [z_{w1}, z_{w2} \dots z_{wnb}, p_{w1}, p_{w2} \dots p_{wna}] \quad (6)$$

$$\rho_h = [z_{h1}, z_{h2} \dots z_{hnd}, p_{h1}, p_{h2} \dots p_{hnc}] \quad (7)$$

$$\rho_{nl} = [v_1, v_2, \dots, v_n, w_1, w_2, \dots, w_n] \quad (8)$$

where the pair  $(v_i, w_i)$  defines the location of a breakpoint and  $n$  is the number of breakpoints assigned to represent the static non-linearity.

The identification problem is presented as a search problem in which the elements of vectors  $\rho_w$ ,  $\rho_h$ , and  $\rho_{nl}$  must be found from input and output measurements. It should be taken into account that parameters  $n_a, n_b, n_c, n_d$  and  $n$  define the model complexity but are unknown beforehand. They will be known after the optimisation algorithm has distributed the dynamics of the initial linear model, whose number of poles ( $n_{poles} = n_a + n_c$ ) and zeros ( $n_{zeros} = n_b + n_d$ ) is known in advance. Since parameter  $n$  is variable, the user must indicate the minimum and maximum number of points allowed ( $n_{min} < n < n_{max}$ ) for the nonlinear block.

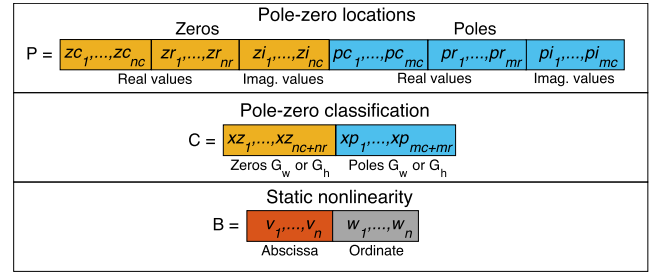


FIGURE 2. Structure of an individual with its coded genetic information.

Thanks to the unified approach proposed in [28], the formulation presented here is also applicable to Wiener and Hammerstein models, which are specific cases of Wiener-Hammerstein structure when one of the two LTI blocks lacks dynamics. In the case of Hammerstein models  $G_w = 1$  and  $\rho_w$  do not exist, while neither do  $G_h = 1$  and  $\rho_h$  in Wiener models. In any case, the user will not need to specify the model structure to be identified, but rather the algorithm will decide the best structure (Wiener, Hammerstein or Wiener-Hammerstein) from the measured data set.

## B. GENETIC CODING AND SEARCH SPACE FOR NONLINEAR OPTIMISATION

This approach uses a multiobjective optimisation algorithm to solve the identification problem detailed in the previous section. This algorithm is based on a population of individuals in which each individual represents a solution of the problem. All the individuals contain coded genetic information on the structure and parameters of a model. This genetic information is composed of three segments (Figure 2). The first one,  $P$ , contains information on the pole and zero locations. A second segment  $C$  contains binary information classifying poles and zeros, i.e. which ones belong to  $G_w$  and which to  $G_h$ . Segment  $B$  contains breakpoint coordinates representing the static non-linearity and varies in size between individuals. It can also change from generation to generation. This variation will be exploited to generate a set of optimal solutions (models of different complexity).

In  $P$ , the field  $zr_1, \dots, zr_{nr}$  contains the real zeros, while fields  $zc_1, \dots, zc_{nc}$  and  $zi_1, \dots, zi_{nc}$  contain the real and imaginary parts of the complex zeros, respectively. In the same way,  $pr_1, \dots, pr_{mr}$  contains the real poles of the model, while fields  $pc_1, \dots, pc_{mc}$  and  $pi_1, \dots, pi_{mc}$  contain the real and imaginary parts of the complex poles, respectively. The vector  $C$  contains a binary code that indicates how the poles and zeros of the initial linear model are distributed between the two LTI subsystems. Field  $xz_1, \dots, xz_{nc+nr}$  indicates how the zeros should be distributed, while field  $xp_1, \dots, xp_{mc+mr}$  indicates the poles' distribution. Values of  $mr, nr, mc$  and  $nc$  are obtained directly from the structure of the initial linear model and indicate the number of real poles and zeros as well as the number of pairs of complex conjugate poles and zeros respectively. The vector  $B$  contains information on the location of the breakpoints used to represent the



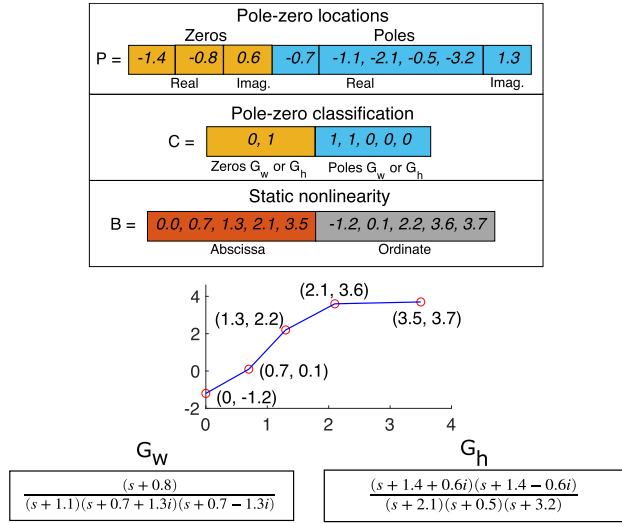


FIGURE 3. Example of an encoded individual and the resulting nonlinear model.

static non-linearity. The pairs  $(v_i, w_i)$  with  $i = 1 \dots n$ , indicate the coordinates in the two-dimensional  $v - w$  plane of the  $n$  breakpoints.

All the parameters of vector  $\rho$  are encoded in vectors  $P$ ,  $B$  and  $C$ . Figure 3 shows an example of how an individual has been coded to represent a nonlinear model in which the initial linear model has six poles (four reals and two complexes) and three zeros (one real and two complexes). According to this structure,  $mr = 4$ ,  $nr = 1$ ,  $mc = 1$  and  $nc = 1$ . On the other hand, it is assumed that static non-linearity is represented with five points ( $n = 5$ ). Notice the correspondence between vectors  $C$  and  $P$ : if an element of vector  $C$  has the value 1, the corresponding zero (or pole) of  $P$  is located in the input LTI subsystem  $G_w(s)$ , while a 0 value in  $C$  indicates that it is located in the output block  $G_h(s)$ . In Figure 3, the first element of  $C$  is zero, so the first one of  $P$  belongs to  $G_h(s)$ . Since the first element of  $P$  contains the real part of a complex zero, the output LTI subsystem must include this zero  $-1.4 + 0.6i$  and its conjugate  $-1.4 - 0.6i$ . Since the second element of  $C$  is 1 then the second one of  $P$  is located in the input LTI subsystem. Under this same logic, the poles are distributed between both LTI subsystems.

As generations go by, the genetic information of individuals is modified by genetic operations to find the best solutions to the identification problem. A modification in the genetic information of  $P$  implies an exploration of new locations for the poles and zeros, while a change in the genetic information of  $C$  means that a new distribution of poles and zeros is tested. On the other hand, a modification in  $B$  may imply a variation in the number of breakpoints or a coordinates change of the existing ones. In any case, individual modifications must respect a search space, which must be adequately defined to facilitate the convergence of the algorithm.

New pole and zero locations are bounded by  $P^{min}$  and  $P^{max}$  (components of vector  $P$  are element-wise bounded by vector  $P^{min}$  and  $P^{max}$ ). Bounds in  $P^{min}$  and  $P^{max}$  are defined around

the location of each pole or each zero of the initial linear model. They can be set according to the sensitivity of poles and zeros, i.e. the closer to the imaginary axis, the smaller the search space, and conversely the farther to the imaginary axis, the larger the bounds. By way of illustration, assuming that the poles and zeros in the example of Figure 3 belong to the initial linear model, vectors  $P^{min}$  and  $P^{max}$  could be set as follows:

$$P^{min} = [-1.7, -1.1, 0.3, -1.0, -1.4, -2.6, -0.8, -4.2, 1.0] \tag{9}$$

$$P^{max} = [-1.1, -0.5, 0.9, -0.4, -0.8, -1.6 - 0.2, -2.2, 1.6] \tag{10}$$

In this case, bounds have been set so that the pole in  $-3.2$  can move  $\pm 1$  around its value, the pole in  $-2.1$  can move  $\pm 0.5$  around its value and all other poles and zeros have a freedom of movement of  $\pm 0.3$ . Since there is no recipe to define the bounds precisely, others could be set for this example. However, it should be noted that too large bounds could cause the algorithm to converge more slowly, while too small bounds could cause an ineffective exploration.

Regarding parameter bounds of non-linearity, WH-MOEA takes the approach defined in [28], where a common search space is defined to face the uncertainty of the location of the static non-linearity around the dynamics. The bounds will be vertically defined by the minimum ( $y_{min}$ ) and maximum ( $y_{max}$ ) values of the output signal, while the horizontal bounds will be given by the minimum ( $u_{min}$ ) and maximum ( $u_{max}$ ) values of the multi-step input signal. To prevent disorderly horizontal movement of breakpoints, movement of each breakpoint will be constrained by the position of the neighbouring breakpoints. In addition, to avoid an overlap, a minimum distance between the breakpoints will be considered through the user-defined parameter  $\alpha$ . The minimum and maximum bounds that define the vertical and horizontal search space of each breakpoint are expressed through:

$$W^{min}(i) = y_{min} \text{ for } i = 1 \dots n \tag{11}$$

$$W^{max}(i) = y_{max} \text{ for } i = 1 \dots n \tag{12}$$

$$V^{min}(i) = \begin{cases} u_{min} & \text{if } i = 1 \\ v_{i-1} + \alpha & \text{if } i = 2 \dots n \end{cases} \tag{13}$$

$$V^{max}(i) = \begin{cases} v_{i+1} - \alpha & \text{if } i = 1 \dots n - 1 \\ u_{max} & \text{if } i = n \end{cases} \tag{14}$$

where vectors  $W^{min}$  and  $W^{max}$  define the vertical search space (y-coordinate) of the breakpoints and vectors  $V^{min}$  and  $V^{max}$  define horizontal ones (x-coordinate). The entire search space of vector  $B$  is then defined as:

$$B^{min} = [V^{min}, W^{min}] \tag{15}$$

$$B^{max} = [V^{max}, W^{max}] \tag{16}$$

On the other hand, the binary code contained in vector  $C$  is generated randomly but subject to the following considerations:

- The number of poles assigned to a subsystem must always be greater than or equal to the number of zeros assigned to the same subsystem, i.e., the resulting system cannot be improper. Therefore  $n_a \geq n_b$  and  $n_c \geq n_d$ .
- The sum of the poles distributed between both subsystems must be equal to the number of poles of the initial linear model ( $n_a + n_c = n_{poles}$ ).
- The sum of the zeros distributed between both subsystems must be equal to the number of zeros of the initial linear model ( $n_b + n_d = n_{zeros}$ ).

**C. MULTIOBJECTIVE OPTIMISATION PROBLEM STATEMENT**

A MOP with  $m$  objectives to minimise can be stated as follows:

$$\min_x f(x) \tag{17}$$

where  $f(x) = [f_1(x), f_2(x) \dots f_m(x)]$  is the vector-valued objective function and  $x$  is the decision variable in the search space  $D$ .

Since a MOP usually involves conflicting objectives, there is no single solution that minimises all the objectives. Instead, there will be a set of optimal solutions, known as non-dominated solutions or Pareto solutions.

*Definition 1 (Pareto Optimality [35]):* An objective vector  $f(x^2)$  is Pareto optimal if there is not another objective vector  $f(x^1)$  such that  $f_i(x^1) \leq f_i(x^2)$  for all  $i \in [1, 2 \dots m]$  and  $f_j(x^1) < f_j(x^2)$  for at least one  $j, j \in [1, 2 \dots m]$ .

*Definition 2 (Dominance [36]):* An objective vector  $f(x^1)$  is dominated by another objective vector  $f(x^2)$  if  $f_i(x^2) \leq f_i(x^1)$  for all  $i \in [1, 2 \dots m]$  and  $f_j(x^2) < f_j(x^1)$  for at least one  $j, j \in [1, 2 \dots m]$ . This is denoted as  $f(x^2) \leq f(x^1)$ .

The Pareto set (set of optimal solutions) and its corresponding Pareto front are therefore defined as follows:

*Definition 3 (Pareto set,  $X_p$ ):* The Pareto set is the set of solutions in  $D$  that are not dominated by another solution in  $D$ .

$$X_p := \{x \in D \mid \nexists x' \in D : f(x') \leq f(x)\}$$

*Definition 4 (Pareto Front,  $f(X_p)$ ):* Given a Pareto set,  $X_p$ , the Pareto front is defined as

$$f(X_p) := \{f(x) \mid x \in X_p\}.$$

Usually,  $X_p$  contains an infinite number of solutions and, for this reason, it is not possible to get it completely. Therefore, a discrete set  $X_p^* \subset X_p$  such that  $f(X_p^*)$  characterises  $f(X_p)$  is obtained. Note that the set  $X_p^*$  is not unique.

For the identification problem, the decision variable  $x$  is a vector formed by the concatenation of vectors  $P$ ,  $B$ , and  $C$ . Therefore, the optimisation problem would be stated as follows:

$$\min_x \{f_1(x) \quad f_2(x)\} \tag{18}$$

$$\text{subject to: } P^{min} \leq P \leq P^{max} \tag{19}$$

$$B^{min} \leq B \leq B^{max} \tag{20}$$

$$n_a \geq n_b \tag{21}$$

$$n_c \geq n_d \tag{22}$$

$$n_a + n_c = n_{poles} \tag{23}$$

$$n_b + n_d = n_{zeros} \tag{24}$$

$$x = [P, B, C] \tag{25}$$

where

$$f_1(x) = \frac{\sum |y - y_r|}{N} \tag{26}$$

$$f_2(x) = n_a + n_c + n_b + n_d + n \tag{27}$$

Notice how  $f_1$  is related to model accuracy quantified by the MAE between the estimated model output ( $y$ ) and the real output ( $y_r$ ) for a set of  $N$  samples, whilst  $f_2$  represents complexity model, measured by the number of poles, zeros and points of the static non-linearity.

It should also be taken into account that the metrics defining the objectives  $f_1$  and  $f_2$  are independent of those that can be used in the estimation and selection of linear structures, which, as explained in Subsection IV-B, is a preliminary step to multiobjective optimisation.

**III. WIENER HAMMERSTEIN MULTIOBJECTIVE EVOLUTIONARY ALGORITHM (WH-MOEA)**

WH-MOEA is an improvement of WH-EA algorithm [16] to address the identification of Wiener, Hammerstein and Wiener-Hammerstein models under a multiobjective optimisation approach. WH-MOEA adopts some genetic operators and coding from the WH-EA algorithm, whilst the structure and functioning are acquired from ev-MOGA algorithm [34]. Arguably, WH-MOEA is therefore an elitist multiobjective evolutionary algorithm based on the concept of  $\epsilon$ -dominance [37] for identification of block oriented models (Wiener, Hammerstein, and Wiener-Hammerstein).

Balancing convergence and diversity is guaranteed in WH-MOEA thanks to features inherited from ev-MOGA. Basically, ev-MOGA tries to ensure that  $X_p^*$  converges toward the Pareto set in a smart distributed manner along the Pareto front with limited memory resources. To do that, 1) the prevalence of dominant solutions in the population  $Pop$  and the archive  $A$  with respect to the dominated solutions is guaranteed, and 2) the objective space is split into a fixed number of boxes, and only one solution can be stored in each box. This avoids the need to use other clustering techniques to obtain adequate distributions, and so considerably reduces the computational cost [34], [37].

**A. WH-MOEA ALGORITHM DESCRIPTION**

The WH-MOEA algorithm is made up of two populations ( $Pop$  and  $G$ ) and one archive ( $A$ ):

- 1)  $Pop$  is the main population. It explores the searching space during the algorithm iterations. Its size is  $N_{indp}$ .

- 2) Archive  $A$  is used to store the Pareto front approximation  $f(X_p^*)$ . Its size  $Nind_A$  is variable but bounded because, as already mentioned, the target space is divided into a finite number of boxes where, at most, there can only be one solution.
- 3) Auxiliary population  $G$  is used to store new solutions created in each iteration of the optimisation process. Its size is  $Nind_G$ , which must be an even number.

The pseudocode of the WH-MOEA algorithm is shown in Algorithm 1, whilst its main steps are detailed as follows:

---

**Algorithm 1** Pseudocode of WH-MOEA
 

---

- 1: Initialise population  $Pop(0)$ ;
  - 2: Initialise archive  $A$  as empty;
  - 3: Evaluate objectives  $f$  for all individuals in  $Pop(0)$ ;
  - 4:  $A(0) = store(A(0), Pop(0))$ ;
  - 5: **for**  $g = 1$  to  $MaxGen$  **do**
  - 6:    $G(g) = create(A(g - 1), Pop(g - 1))$ ;
  - 7:   Evaluate objectives  $f$  for all individuals in  $G(g)$ ;
  - 8:    $A(g) = store(A(g - 1), G(g))$ ;
  - 9:    $Pop(g) = update(Pop(g - 1), G(g))$ ;
  - 10: **end for**
  - 11: Print solution  $A(MaxGen)$
- 

**Step 1.**  $Pop(0)$  is initialised with  $Nind_p$  individuals (candidate solutions). First individual is coded from the initial linear model, as indicated in [16] and [28]. Next, this first individual undergoes all mutation operations  $Nind_p - 1$  times to give rise to the rest of the population.

**Step 4.** Function *store* checks individuals in  $Pop(g)$  that might be included in the archive  $A(g)$  taking into account  $\epsilon$ -dominance concept.

**Step 6.** Function *create* creates new individuals of  $G(g)$  by using procedure and genetic operators described in Subsection III-B.

**Step 8.** Function *store* checks individuals in  $G(g)$  that might be included in the archive  $A(g)$  taking into account  $\epsilon$ -dominance concept. Besides, individuals from  $A(g)$ , which are  $\epsilon$ -dominated by individuals in  $G(g)$ , will be eliminated.

**Step 10.** Function *update* updates  $Pop(g)$  with individuals from  $G(g)$ . Every individual  $G(g)$  is compared with an individual that is randomly selected from the individuals in  $Pop(g)$ . The individual who is not dominated will be the one who remains at  $Pop(g)$ .

## B. DESCRIPTION OF CREATE FUNCTION AND GENETIC OPERATORS

Function *create* creates  $G(g)$  by using Algorithm 2. This procedure is repeated  $Nind_G/2$  times until  $G(g)$  is filled. A new individual is created by altering the genetic information of an existing individual in the main population  $Pop(g)$ , which is selected at random with  $r_1$ . The genetic alteration is given by the mutations M.1, M.2, M.3, M.4, M.5, and the crossover

---

**Algorithm 2** Pseudocode of *create* Function
 

---

- 1: Random selection of a individual ( $r_1$ ) of  $Pop(g)$ ;
  - 2: Random selection of a individual ( $r_2$ ) of  $A(g)$ ;
  - 3: Compute  $\gamma(g)$ ;
  - 4: **if**  $r_{pzn} \leq \gamma(g)$  **then**
  - 5:   Compute  $\eta(g)$ ;
  - 6:   **if**  $r_{mm} \leq \eta_{min} + \eta(g)$  **then**
  - 7:     Mutation M.2
  - 8:   **else**
  - 9:     Mutation M.3
  - 10:   **end if**
  - 11: **else**
  - 12:   Mutation M.1;
  - 13: **end if**
  - 14: **if**  $r_c \leq \xi$  **then**
  - 15:   Mutation M.4;
  - 16: **end if**
  - 17: **if**  $r_{nl} \leq P_{nl}$  **then**
  - 18:   Mutation M.5;
  - 19: **end if**
  - 20: **if**  $r_{cr} \leq P_{cr}$  **then**
  - 21:   Crossover C.3;
  - 22: **end if**
- 

---

**Algorithm 3** Pseudocode of Mutation M.5
 

---

- 1: Compute  $P_{as}$
  - 2: **if**  $P_{as} \leq 0.5$  **and**  $size(\mathbf{B}/2) < n_{max}$  **then**
  - 3:   Increase a breakpoint
  - 4: **else if**  $P_{as} > 0.5$  **and**  $size(\mathbf{B}/2) > n_{min}$  **then**
  - 5:   Compute the slopes of all segments
  - 6:   Find the two consecutive segments with the most similar slopes
  - 7:   Compute the difference ( $D_s$ ) between these two slopes
  - 8:   **if**  $|D_s| \leq 1.0 \times 10^{-3}$  **then**
  - 9:     Remove the breakpoint that is common to both segments
  - 10:   **end if**
  - 11: **end if**
- 

C.3. In this last genetic operation the individual selected from  $Pop(g)$  and an individual from the archive  $A(g)$  randomly selected by  $r_2$  exchange information.

Each genetic operation acts on a certain portion of an individual's information. As can be seen in Algorithm 2, to expand diversity and avoid premature convergence, not all genetic operations are applied at the same time. A random process and control parameters  $\gamma(g)$ ,  $\eta_{min}$ ,  $\eta(g)$ ,  $\xi$ ,  $P_{cr}$ , and  $P_{nl}$  decide which genetic operators should be applied. The random number  $r_{pzn} \in (0, 1]$  is compared with  $\gamma(g)$  to decide if the location of a pole/zero is mutated with M.1 or whether the location of a breakpoint is mutated with either M.2 or M.3. This last selection depends on the random number  $r_{mm} \in (0, 1]$  and the control parameter  $\eta(g)$ . On the other

**Algorithm 4** Pseudocode of Crossover C.3

---

```

1: Compute  $C_{ga}$ 
2: if  $C_{ga} = 1$  then
3:   Individuals  $r_1$  and  $r_2$  cross all genetic information
   contained in  $P$ 
4: else if  $C_{ga} = 2$  then
5:   Individuals  $r_1$  and  $r_2$  cross all genetic information
   contained in  $B$ 
6: else if  $C_{ga} = 3$  then
7:   Individuals  $r_1$  and  $r_2$  cross all genetic information
   contained in  $C$ 
8: end if

```

---

hand, parameter  $\xi$  defines the probability that an information alteration will occur in  $C$  (binary code for the classification of the dynamics). This alteration is handled by mutation M.4, and also its occurrence depends randomly on the value of  $r_c \in (0,1]$ . Similarly, mutation M.5 and crossover C.3 are randomly selected through random numbers  $r_{nl} \in (0, 1]$  and  $r_{cr} \in (0, 1]$  respectively. Parameter  $P_{nl}$  defines the probability that mutation M.5 occurs, while parameter  $P_{cr}$  defines the probability that crossover C.3 occurs.

Control parameters adjust the probabilities of mutation and crossover. The parameters  $\gamma(g)$  and  $\eta(g)$  change through generations, while fixed parameters  $\eta_{min}$ ,  $\xi$ ,  $P_{cr}$ , and  $P_{nl}$  are considered tuning parameters of the algorithm. However, through examples where WH-MOEA and its precursors have been used, it has been possible to establish through trial and error appropriate values depending on the results obtained. These values have been used in both identification problems addressed in this paper.

During the first generations,  $\gamma(g)$  is close to one which gives a high probability of a mutation occurring on the segment that contains genetic information on the position of the breakpoints; the probability of a mutation occurring on the segment containing genetic information on the location of the poles and zeros is therefore low. Parameter  $\gamma(g)$  decreases as the generations pass so that in the last generations the algorithm will modify both portions of genetic information with equal probability. The way the  $\gamma(g)$  parameter works is justified by the fact that static non-linearity is entirely unknown, so the algorithm should focus more on this portion of genetic information during the first generations. As non-linearity takes shape, the algorithm will increase the probability that new positions for the poles and zeros will also be explored, so that the dynamics will be refined. Similarly, the control parameter  $\eta(g)$  allows the probability of selecting between M.2 and M.3 mutations to be variable. During the first generations, mutation M.3 is not so necessary since this genetic operation concentrates breakpoints in the curvatures to achieve higher accuracy. As long as the non-linearity does not take shape, the concentration of points in the curves will have no significant effect. The mutations M.1, M.2, M.3, and M.4 are inherited from the WH-EA, therefore further information on how these genetic operations work as well as

the calculation of the parameters  $\gamma(g)$  and  $\eta(g)$  can be found in [16]. In this section, a brief description of these genetic operations and new genetic operators (mutation M.5 and crossover C.3) are briefly explained.

- Mutation M.1. A gene (element) of  $P$  is randomly selected to be modified. This modification involves exploring a new position for the corresponding pole or zero within the search space. The following mathematical expression describes this genetic operation:

$$p^{new} = p^{act} + N_{zp}(0, \sigma^2(g)) \quad (28)$$

where,  $N_{zp}(0, \sigma^2(g))$  is a random number with Gaussian distribution and variable standard deviation  $\sigma^2(g)$ .  $p^{act}$  contains the current value of the selected gene to be mutated, while  $p^{new}$  contains the result of the mutation. In each generation  $g$ , the standard deviation is calculated with the following expressions:

$$\sigma^2(g) = \frac{\Delta_s}{100} \left( \frac{\sigma_{ini}^2}{\sqrt{1 + g * \sigma_{ratio}^2}} \right) \quad (29)$$

$$\sigma_{ratio}^2 = \frac{(\sigma_{ini}^2 / \sigma_{end}^2)^2 - 1}{MaxGen - 1} \quad (30)$$

As generations go by,  $\sigma^2(g)$  will reduce its value from an initial value ( $\sigma_{ini}^2$ ) to a final value ( $\sigma_{end}^2$ ). In the last generations, mutations on  $P$  will be more subtle to achieve a fine-tuning of the parameters. The rate of decrease of the standard deviation ( $\sigma_{ratio}^2$ ) depends on  $\sigma_{ini}^2$ ,  $\sigma_{end}^2$  and the predefined number of algorithm generations ( $MaxGen$ ). Both  $\sigma_{ini}^2$  and  $\sigma_{end}^2$  are user-defined parameters that must be configured before the execution of the algorithm.  $\Delta_s$  is a measure of the scanning space that can be calculated with the upper and lower limit of the search space of the corresponding pole or zero.

- Mutation M.2. A pair of  $B$  genes are randomly selected to be modified. These genes are matched and represent the coordinates of a breakpoint. The new values of the selected genes are calculated with the same procedure used in the mutation M.1. Two random numbers with Gaussian distribution ( $N_v(0, \sigma^2(g))$  and  $N_w(0, \sigma^2(g))$ ) are thus required.  $N_v(0, \sigma^2(g))$  is used to mutate the gene of the abscissa of the selected breakpoint, while  $N_w(0, \sigma^2(g))$  is used to mutate the gene of the ordinate of the same breakpoint. As in mutation M.1,  $\sigma^2(g)$  varies from  $\sigma_{ini}^2$  to  $\sigma_{end}^2$  to control the aggressiveness of the mutations and fine-tune the breakpoints.
- Mutation M.3. This genetic operation is applied to  $B$  and allows the breakpoints to jump to each other so that they can concentrate on the curvatures. The two genes that define the position of a breakpoint are randomly selected. The selected breakpoint must jump to a new location which will be in a segment defined by two other breakpoints. This segment is also chosen randomly. The new abscissa is computed as the midpoint between the two breakpoints that form the segment over which



the point will jump, while the new ordinate will be calculated using a quadratic interpolation and information on the breakpoints near the location where the jump occurred. Quadratic interpolation will help to make a smooth transition from a point when it jumps to a specific segment.

- Mutation M.4. This genetic operation is applied to **C** and generates new combinations of poles and zeros to give rise to the dynamics of the two LTI subsystems. Each time this mutation is required, a new binary code subject to (21)-(24) is randomly generated in **C**.
- Mutation M.5. This genetic operation is applied to **B**. Under certain conditions it can increase or reduce the number of breakpoints used to represent static non-linearity. The way this genetic operation works is synthesised in Algorithm 3. A possible increase or decrease is determined by the random number  $P_{as} \in (0, 1]$ . An increase of one breakpoint will occur as long as the individual selected to mutate does not contain in **B** the maximum amount of breakpoints allowed ( $n_{max}$ ). If an increase is required, a new breakpoint will appear on a randomly selected segment. As with mutation M.3, the abscissa of the new breakpoint will be calculated as the midpoint between the two breakpoints that define the selected segment, while the ordinate will be calculated using a quadratic interpolation and the coordinates of three neighbouring breakpoints. For a decrease to occur, two conditions must be met in addition to  $P_{as} > 0.5$ . The first condition is that **B** must not contain the minimum number of breakpoints allowed ( $n_{min}$ ), while, the second condition allows a decrease to occur as long as a breakpoint is located on a straight line, that is, in a redundant position. To verify this condition, the slopes of all segments joining two breakpoints will be calculated. The slope of each segment will be compared with the slope of the segment on the right. The two consecutive segments with the smallest difference in slope will contain the breakpoint that is likely to be eliminated. These two slopes will be compared again; if the absolute value of the difference is close to zero, the breakpoint that is common to the two segments will be removed. Closeness to zero is quantified by a fixed value of  $1.0 \times 10^{-3}$ . This procedure aims to eliminate unnecessary breakpoints.
- Crossover C.3. This genetic operation allows an individual from the main population  $Pop(g)$  to exchange genetic information with an individual from the archive  $A(g)$  (front of Pareto), for which two random integers are generated. The first integer ( $r_1 \in [1, size(Pop(g))]$ ) is used to randomly select the individual of  $Pop(g)$ , while the second integer ( $r_2 \in [1, size(A(g))]$ ) is used to randomly select the individual of  $A(g)$ . Each individual of both  $Pop(g)$  and  $A(g)$  has three portions of genetic information (**P**, **B**, and **C**), however, the exchange of information between the two individuals will be of only one portion, for which another random integer is generated ( $C_{ag} \in [1, 3]$ ). This number will decide which

portion of the genetic information will be exchanged (see Algorithm 4).

#### IV. USING WH-MOEA TO COMPARE DESIGN CONCEPTS: A PROCEDURE

##### A. COMPARING DESIGN CONCEPTS BY THE MOP APPROACH

When dealing with a MOP design problem, it is quite common to consider different alternatives (design concepts), i.e. ideas on how to solve the problem. It is possible to define an independent optimisation problem for each of these design concepts [38] and when they share the same objectives they can easily be compared directly in the objective space by simply comparing their respective Pareto fronts. For example in [39], this idea is used to compare alternative slide model control structures, in [40] to compare different loop pairings and in [41] to analyse the performance of different battery models.

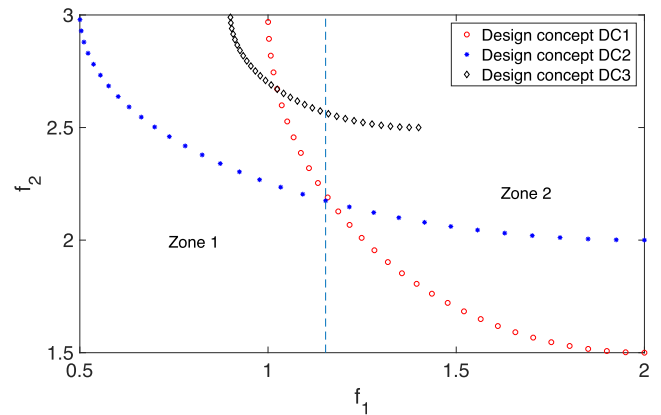


FIGURE 4. Comparison of three design concepts in a bi-objective space.

Figure 4 shows an example in which the Pareto fronts of three design concepts are compared in a two-dimensional objective space. Notice that as design concept DC2 completely dominates design concept DC3, concept DC2 will be preferred over concept DC3. However, DC2 does not completely dominate concept DC1 or vice versa. If the designer has a preference for solutions in Zone 1, solutions from concept DC2 would be selected, as in this zone concept DC2 dominates concept DC1. Whilst if Zone 2 was preferred, the designer should choose solutions from concept DC1. This illustrative example shows how interesting it is to use the MOP optimisation approach when comparing possible design concepts. In the end, the designer would choose the final solution taking into account the dominance between concepts and his/her specific preferences.

In this paper, an initial linear model is used as a starting point for nonlinear estimation. Most of the time, the model is selected from a ranking based on a specific criterion. However, under the assumption that there are any other useful criteria giving more candidates, the idea of design concepts arises so more than one initial linear model should be tested. Then, each linear structure will give rise to a design concept.

TABLE 2. WH-MOEA tuning parameters.

Parameter	Description
$MaxGen$	Maximum number of generations.
$Nind_P$	Population size.
$Nind_G$	Size of the auxiliary population $G$ .
$n_{box_1}$	Number of divisions for the objective $f_1$ .
$n_{box_2}$	Number of divisions for the objective $f_2$ .
$n_{max}$	Maximum number of breakpoints
$n_{min}$	Minimum number of breakpoints
$\alpha$	Minimum distance between two breakpoints on the abscissa axis
$\sigma_{ini}^2, \sigma_{end}^2$	Initial and final standard deviations for control of the aggressiveness of mutations M.1 and M.2
$\eta_{min}$	Minimum probability for selection of mutation M.2. The maximum probability for this selection is 1.
$\xi$	Probability to modify with mutation M.4 the genetic information related to the classification of poles and zeros.
$P_{nl}$	Probability to modify with mutation M.5 the number of breakpoints for static non-linearity
$P_{cr}$	Probability of crossover between an individual of $Pop$ and an individual of $A$

Furthermore, if you sort the list of initial candidates by a criterion, several models with similar values could be obtained and therefore one might add other criterion to perform the initial selection. In these case, it is worth to evaluate more than one linear structure.

**B. PROCEDURE DEFINITION**

Using WH-MOEA as a tool for nonlinear identification can not only get a specific model, but also a set of models with different features for effective decision-making based on particular needs. The suggested procedure is defined as follows:

*Step 1 (Establish a set of Candidates Using a Standard Linear Identification Method):* Since WH-MOEA can fine-tune the initial linear model, a classical step test can be used. The ranking can be made according to the structure of the models (number of poles and zeros) and one or more performance criteria. For example, the mean squared error (MSE) could be used; however, to avoid over-modelling the MDL criteria or any others, mentioned in Section I, can be useful.

*Step 2 (Choose the Design Concepts):* From this set of candidates, you have to select some initial linear structures to be tested. Each one will correspond to a design concept. It may be that two or more structures present similar values to a performance criterion or you may have two or more conflicting candidates for the best of their respective criteria. However, other less complex linear structures can also be selected. A comparison of the Pareto fronts of various design concepts will help in adequate decision-making.

*Step 3 (Prepare a Multi-Step Excitation Signal and Performs an Experiment With the Process Under Identification):* The input signal must ensure that the system output reaches steady state. For static non-linearity to be appropriately captured, different step amplitudes must allow a scan of the entire process operating range. A minimum distance between two consecutive steps must be considered [28] to highlight static non-linearity. Finally, excite the process under identification with this signal and record its output for a given sampling period.

*Step 4 (Set the Bounds):* From the minimum and maximum values of the input ( $u_{min}$  and  $u_{max}$ ) and output ( $y_{min}$  and  $y_{max}$ ) signals, define the search space bounds for static non-linearity using equations (11) to (14).

*Step 5 (Run WH-MOEA for Each Design Concept):* Table 2 shows the list of WH-MOEA parameters that must be previously assigned to carry out each run. The bounds  $P^{min}$  and  $P^{max}$  must also be previously defined as a function of the sensitivity of the poles and zeros. At the end of each WH-MOEA execution a Pareto front is obtained.

*Step 6 (Perform a Decision Making Analysis):* Each Pareto front contains a set of models with different performances. For trade-off analysis, draw all the Pareto fronts on the same graph. Evaluate selected models on the estimation and validation data sets for a final decision.

**V. APPLICATION OF WH-MOEA PROCEDURE, RESULTS AND DISCUSSION**

To show the effectiveness and usefulness of the methodology presented, two application examples are included in this section. The first consists of a numerical example, a process with a pure Wiener-Hammerstein structure, while the second is a real thermal process based on a Peltier cell. WH-MOEA parameters were set to the same values for both identification problems:  $\xi = 0.25$ ;  $P_{nl} = 0.33$ ;  $P_{cr} = 0.33$  and  $\eta_{min} = 0.30$ . The initial and final standard deviations for mutations were set to 20 and 1, respectively.

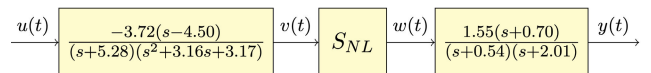


FIGURE 5. Numerical example: Wiener-Hammerstein structure.

**A. APPLICATION 1: NUMERICAL EXAMPLE**

Figure 5 shows the Wiener-Hammerstein process used for this example. Static non-linearity ( $S_{NL}$ ) consists of a sigmoid hyperbolic tangent function “*tansig*” (31), which symmetrically saturates large values of the independent variable. Also, Gaussian noise with a power of  $-40db$  was added on the system output to emulate a real situation.

$$S_{NL} = 50[1 + \text{tansig}(0.07(v(t) - 50))] \tag{31}$$

*Step 1 (Establish a set of Candidates Using a Standard Linear Identification Method):* First, an experiment for initial linear identification was designed. Since input range was between 0 and 100 units a constant signal with amplitude

50 units was injected into the system until it reached steady-state. On this operating point, the process was excited with a small input change of 2.5 units and a duration of 15s (Figure 6). During the experiment, input and output data were recorded with a period of 50ms. Offset was removed from the data to avoid problems with initial conditions.

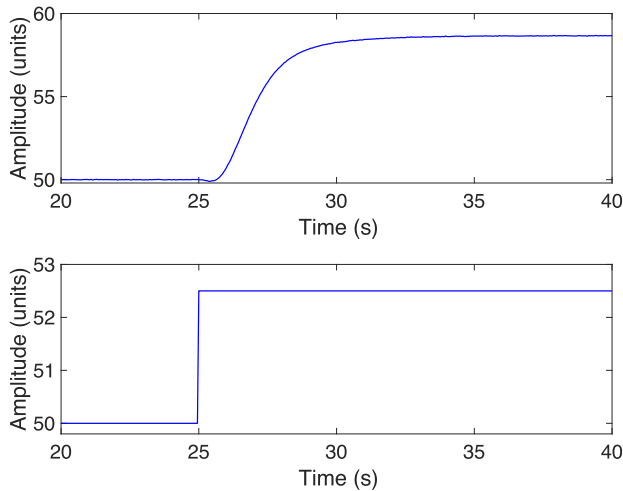


FIGURE 6. Input and output data for linear estimations of WH process. Top: Simulated system output. Bottom: Input signal.

TABLE 3. Set of candidates of initial linear models for WH process. Model  $G_{2z/6p}$  was excluded since its MDL value was very high.

Model	MSE	MDL
$G_{3p}$	$6.318 \times 10^{-3}$	$3.412 \times 10^{-3}$
$G_{1z/3p}$	$2.283 \times 10^{-4}$	$1.254 \times 10^{-4}$
$G_{2z/3p}$	$2.186 \times 10^{-4}$	$1.220 \times 10^{-4}$
$G_{4p}$	$1.732 \times 10^{-3}$	$9.539 \times 10^{-4}$
$G_{1z/4p}$	$2.201 \times 10^{-4}$	$1.228 \times 10^{-4}$
$G_{2z/4p}$	$1.486 \times 10^{-4}$	$8.506 \times 10^{-5}$
$G_{3z/4p}$	$9.315 \times 10^{-5}$	$5.479 \times 10^{-5}$
$G_{5p}$	$5.504 \times 10^{-4}$	$3.092 \times 10^{-4}$
$G_{1z/5p}$	$2.114 \times 10^{-4}$	$1.201 \times 10^{-4}$
$G_{2z/5p}$	$8.143 \times 10^{-5}$	$4.781 \times 10^{-5}$
$G_{3z/5p}$	$8.139 \times 10^{-5}$	$4.850 \times 10^{-5}$
$G_{6p}$	$4.580 \times 10^{-4}$	$2.610 \times 10^{-4}$
$G_{1z/6p}$	$4.721 \times 10^{-4}$	$3.028 \times 10^{-4}$
$G_{3z/6p}$	$8.126 \times 10^{-5}$	$4.928 \times 10^{-5}$

Fifteen initial linear models were estimated from 3<sup>rd</sup> to 6<sup>th</sup> order considering only strictly proper systems and no more than three zeros. For each structure, a linear model was estimated using the Matlab System Identification Toolbox [42] with the *tfest* command. The MSE and the modified MDL were calculated on the estimation data set. The results of these estimates are shown in Table 3. According to the modified MDL criteria, the best model is two zeros - five poles ( $G_{2z/5p}$ ) which is consistent with the actual dynamics of the WH process under consideration.

**Step 2 (Choose the Design Concepts):** According to the results in Table 3, the linear structure  $G_{2z/5p}$  might be considered for nonlinear estimation; however, as mentioned in Section IV, others can also be considered. Models with six poles would not be a good option since there are other

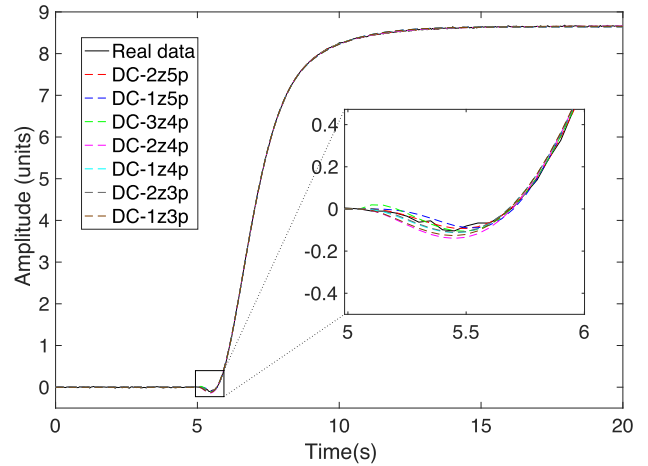


FIGURE 7. Performance of linear models (design concepts) chosen for nonlinear estimation of WH process. To distinguish between models, a magnification is depicted on a portion of data.

less complex models that can achieve similar performance. As model  $G_{3z/5p}$  has a zero in  $-119$ , far from the system dominant dynamics, all the models more complex than  $G_{2z/5p}$  would be discarded.

The effect of non-minimum phase zero in this example is remarkable. The MSE of model  $G_{1z/3p}$  (4 zero-pole) is better than that of model  $G_{4p}$  having the same number of parameters. It is even better than more complex models without zeros, such as models  $G_{5p}$  and  $G_{6p}$ . For this reason, models without zeros will not be considered for nonlinear estimation.

According to this analysis, linear structures  $G_{1z/3p}$ ,  $G_{2z/3p}$ ,  $G_{1z/4p}$ ,  $G_{2z/4p}$ ,  $G_{3z/4p}$ ,  $G_{1z/5p}$ , and  $G_{2z/5p}$  would be good options for nonlinear estimation. Each of these structures will give rise to a design concept for nonlinear estimation. Table 4 shows the poles and zeros of each design concept, while Figure 7 shows a comparison of their performance. For convenience sake, from now on each design concept will use a nomenclature as indicated in the first column of Table 4. This nomenclature is preceded by the characters “DC” followed by an indication of the number of poles and zeros of the linear structure.

**Step 3 (Prepare a Multi-Step Excitation Signal and Performs an Experiment With the Process Under Identification):** Two multi-step signals were designed, one for estimation and one for validation purposes (see Figure 8). The amplitude of the steps in both signals was handled randomly, lasting 25s each. The minimum step amplitude was set to 15 units. The estimation signal was designed with 50 steps, while the validation signal was designed with 30 steps. The proposed WH process was simulated twice in Matlab using estimation and validation multi-step inputs. In both experiments, process input and output were recorded with a sampling period of 50ms.

**Step 4 (Set the Bounds):** From the estimation data set, the minimum and maximum values of the input and output signals were obtained ( $u_{min} = 0$ ,  $u_{max} = 100$ ,  $y_{min} = -0.024$  and  $y_{max} = 99.933$ ). The vertical search space for

TABLE 4. Selected initial linear models (design concepts) for nonlinear identification of WH process.

Design Concept	Model	Poles	Zeros
DC – 2z5p	$G_{2z/5p}$	$[-5.873; -1.812; -0.550; -1.609 \pm 0.893i]$	$[4.459; -0.719]$
DC – 1z5p	$G_{1z/5p}$	$[-0.729; -2.168 \pm 8.699i; -1.238 \pm 1.470i]$	$[4.896]$
DC – 3z4p	$G_{3z/4p}$	$[-0.974; -0.451; -1.465 \pm 1.308i]$	$[8.880; 5.081; -0.504]$
DC – 2z4p	$G_{2z/4p}$	$[-0.794; -0.249; -1.289 \pm 1.400i]$	$[3.750; -0.256]$
DC – 1z4p	$G_{1z/4p}$	$[-33.92; -0.732; -1.250 \pm 1.491i]$	$[4.089]$
DC – 2z3p	$G_{2z/3p}$	$[-0.733; -1.254 \pm 1.492i]$	$[27.52; 4.193]$
DC – 1z3p	$G_{1z/3p}$	$[-0.726; -1.214 \pm 1.473i]$	$[3.866]$

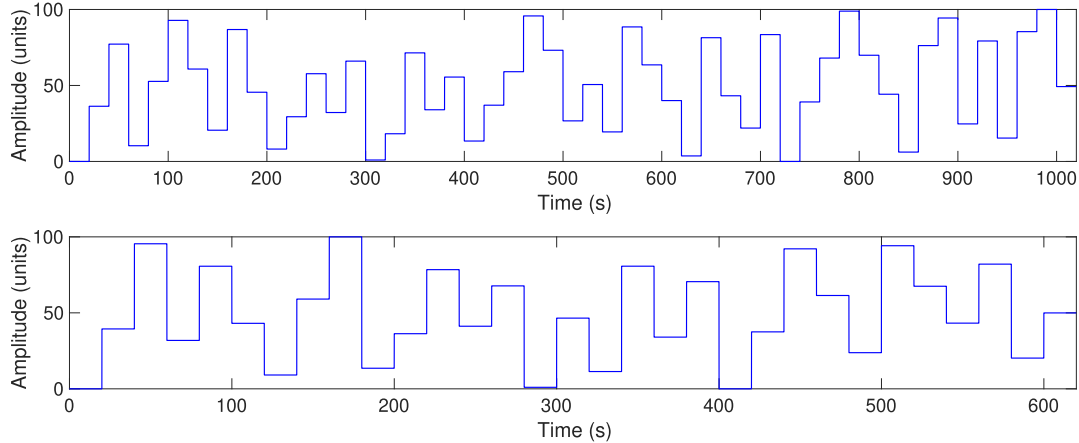


FIGURE 8. Estimation (top) and validation (bottom) input signals for nonlinear identification of WH process.

all breakpoints was defined with  $y_{min}$  and  $y_{max}$  using (11) and (12) respectively:

$$W^{min}(i) = -0.024 \text{ for } i = 1 \dots n \quad (32)$$

$$W^{max}(i) = 99.933 \text{ for } i = 1 \dots n \quad (33)$$

whilst  $u_{min}$  and  $u_{max}$  values were used internally by the algorithm to define the variable limits  $V^{min}$  and  $V^{max}$  using (13) and (14) respectively. It should be noted that the horizontal search space of each breakpoint is variable and depends on the position of the neighbouring breakpoints.

Step 5 (Run WH-MOEA for Each Design Concept): According to selected design concepts (Table 4), seven bi-objective optimisation problems were stated as in Subsection II-C. WH-MOEA was thus executed seven times, each one fed with a different initial linear model (design concept). The configuration parameters in Table 5 were used for all the executions. Table 6 shows how the first individual in the population ( $P_1^0$ ) was coded for each design concept and the lower ( $P^{min}$ ) and upper ( $P^{max}$ ) exploration bounds to refine the locations of the poles and zeros (defined individually around each pole or zero). It should be borne in mind that there is no recipe to set precisely the bounds but, the higher the interval the more exploration (slower algorithm convergence). A good alternative is to set the bounds based on the sensitivity of each pole or zero. Those closest to the imaginary axis are more sensitive to changes so their exploration bounds may be small, while a greater degree of freedom can be given to the poles and zeros furthest from the imaginary axis.

After the seven WH-MOEA runs, a total of 82 different models were obtained. Table 7 shows the number of models

TABLE 5. Summary of WH-MOEA parameter settings for nonlinear estimation of WH process.

Parameter	Value	Parameter	Value
MaxGen	$2.0 \times 10^6$	$n_{box2}$	16
Nind <sub>p</sub>	$5.0 \times 10^4$	$n_{max}$	26
Nind <sub>G</sub>	4	$n_{min}$	10
$n_{box1}$	50	$\alpha$	0.641

obtained in each design concept. This table also specifies how each model has been labelled.

Step 6 (Perform a Decision Making Analysis): To make an effective analysis of the results obtained, the Pareto fronts of all design concepts must be compared in a two-dimensional objective space. According to the number of design concepts stated for this example, seven Pareto fronts were obtained. Given the high precision achieved by the models of design concept DC – 2z5p, the corresponding Pareto front will be analysed separately. Figure 9 shows a graph of this Pareto front which contains 14 Wiener-Hammerstein models with the same dynamic structure. The distribution of the poles and zeros of these 14 models is consistent with the real dynamics of the proposed example. Undoubtedly, the fact of having the same structure as the real example has meant that the precision of all these models cannot be achieved by any model of the other fronts, even by the models of design concept DC – 3z4p, which has the same complexity level devoted to representing the dynamic part. In this case, the nonlinear estimates have highlighted the different degrees of accuracy that models of these design concepts can achieve. This difference is not so apparent on inspecting Table 3, which contains the results of the linear estimates, i.e. the MAE achieved



**TABLE 6.** Coding of the first individual in the population and exploration bounds to refine the poles and zeros (design concepts for identification of the numerical example).

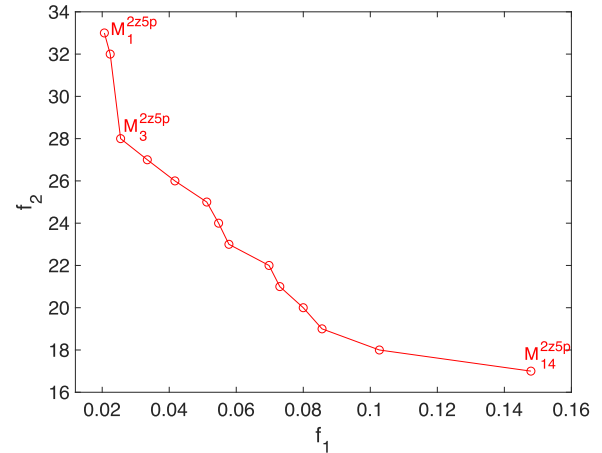
<i>DC – 2z5p</i>	
<i>nc = 0, nr = 2, mc = 1, mr = 3</i>	
$P^{min}$	[3.959, -0.819, -1.709, -6.373, -2.112, -0.650, 0.793]
$P_1^0$	[4.459, -0.719, -1.609, -5.873, -1.812, -0.550, 0.893]
$P^{max}$	[4.959, -0.619, -1.509, -5.372, -1.512, -0.450, 0.993]
<i>DC – 1z5p</i>	
<i>nc = 0, nr = 1, mc = 2, mr = 1</i>	
$P^{min}$	[4.396, -2.468, -1.438, -0.929, 7.699, 1.170]
$P_1^0$	[4.896, -2.168, -1.238, -0.729, 8.699, 1.470]
$P^{max}$	[5.396, -1.868, -1.038, -0.529, 9.699, 1.770]
<i>DC – 3z4p</i>	
<i>nc = 0, nr = 3, mc = 1, mr = 2</i>	
$P^{min}$	[7.880, 4.581, -0.604, -1.765, -1.174, -0.551, 1.008]
$P_1^0$	[8.880, 5.081, -0.504, -1.465, -0.974, -0.451, 1.308]
$P^{max}$	[9.880, 5.580, -0.404, -1.165, -0.774, -0.351, 1.608]
<i>DC – 2z4p</i>	
<i>nc = 0, nr = 2, mc = 1, mr = 2</i>	
$P^{min}$	[3.250, -0.356, -1.589, -0.994, -0.349, 1.100]
$P_1^0$	[3.750, -0.256, -1.289, -0.794, -0.249, 1.400]
$P^{max}$	[4.250, -0.156, -0.989, -0.594, -0.149, 1.700]
<i>DC – 1z4p</i>	
<i>nc = 0, nr = 1, mc = 1, mr = 2</i>	
$P^{min}$	[3.589, -1.550, -40.92, -0.932, 1.192]
$P_1^0$	[4.089, -1.250, -33.92, -0.732, 1.492]
$P^{max}$	[4.589, -0.950, -26.92, -0.532, 1.792]
<i>DC – 2z3p</i>	
<i>nc = 0, nr = 2, mc = 1, mr = 1</i>	
$P^{min}$	[20.52, 3.693, -1.554, -0.933, 1.192]
$P_1^0$	[27.52, 4.193, -1.254, -0.733, 1.492]
$P^{max}$	[34.52, 4.693, -0.954, -0.533, 1.792]
<i>DC – 1z3p</i>	
<i>nc = 0, nr = 1, mc = 1, mr = 1</i>	
$P^{min}$	[3.366, -1.514, -0.926, 1.173]
$P_1^0$	[3.866, -1.214, -0.726, 1.473]
$P^{max}$	[4.366, -0.914, -0.526, 1.773]

**TABLE 7.** Number of models obtained for each design concept.

Design Concept	Number of models	Labels
DC-2z5p	14	$[M_1^{2z5p}, M_2^{2z5p}, \dots, M_{14}^{2z5p}]$
DC-1z5p	10	$[M_1^{1z5p}, M_2^{1z5p}, \dots, M_{10}^{1z5p}]$
DC-3z4p	12	$[M_1^{3z4p}, M_2^{3z4p}, \dots, M_{12}^{3z4p}]$
DC-2z4p	14	$[M_1^{2z4p}, M_2^{2z4p}, \dots, M_{14}^{2z4p}]$
DC-1z4p	13	$[M_1^{1z4p}, M_2^{1z4p}, \dots, M_{13}^{1z4p}]$
DC-2z3p	10	$[M_1^{2z3p}, M_2^{2z3p}, \dots, M_{10}^{2z3p}]$
DC-1z3p	10	$[M_1^{1z3p}, M_2^{1z3p}, \dots, M_{10}^{1z3p}]$

by initial models  $G_{2z/5p}$  and  $G_{3z/4p}$  is  $8.143 \times 10^{-5}$  and  $9.315 \times 10^{-5}$  while their MDL is  $4.781 \times 10^{-5}$  and  $5.479 \times 10^{-5}$ , respectively. Apparently, it seems that both models have similar performances. However, with nonlinear estimations, the importance of selecting the appropriate structure has been highlighted.

Looking at Figure 9, it can be clearly seen that the best model in terms of accuracy is  $M_1^{2z5p}$ . The precision (objective  $f_1$ ) achieved by this model is  $2.065 \times 10^{-2}$ , while its complexity level (objective  $f_2$ ) is 33 (2 zeros, 5 poles and 26 points for the static non-linearity). This model would undoubtedly be



**FIGURE 9.** Pareto front corresponding to the design concept *DC – 2z5p*.

the best option in terms of precision, but it should be borne in mind that it is also the most complex model.

A great advantage of this approach is that the Pareto fronts allow the performance of different models to be compared. In addition, in each Pareto front it is also possible to analyse the contribution of the number of breakpoints in terms of precision. Looking at the Pareto front in Figure 9 it can be seen that the accuracy achieved by model  $M_3^{2z5p}$  is very close to that achieved by model  $M_1^{2z5p}$ . Considering the precision between  $M_1^{2z5p}$  and  $M_3^{2z5p}$  is 100%, improvement of  $M_1^{2z5p}$  concerning  $M_3^{2z5p}$  is only 3.85%. In this regard,  $M_3^{2z5p}$  would be preferred since it has five points less.

Figure 10 compares the other Pareto fronts. Analysing these fronts one can notice that the best models in terms of precision are on the front of design concept *DC – 3z4p*, However, it should be noted that the models of this design concept have the same complexity in the dynamic part as the models of the design concept *DC – 2z5p*, which achieved better accuracy. It would therefore not make sense to select models from design concept *DC – 3z4p*, as there will always be an equally or less complex model of design concept *DC – 2z5p* which will always be more accurate.

Something similar occurs with models belonging to design concepts *DC – 1z5p* and *DC – 2z4p*, all of which have the same complexity in the dynamic part; however, the difference in accuracy is evident. This would exclude all models that have one zero and five poles from the selection since there will always be a model with the same complexity in the dynamic part that has better performance in terms of accuracy.

In the same way, the models of design concepts *DC – 1z4p* and *DC – 2z3p* have equal complexity in the dynamic part, so it would be interesting to compare them. At first glance, the most complex model of each design concept ( $M_1^{1z4p}$  and  $M_1^{2z3p}$ ) perform identically. However, as the number of static nonlinearity points decreases, the models of the *DC – 2z3p* Pareto set become more accurate. It is thus not practical to select models from *DC – 1z4p*. Even if the precision of

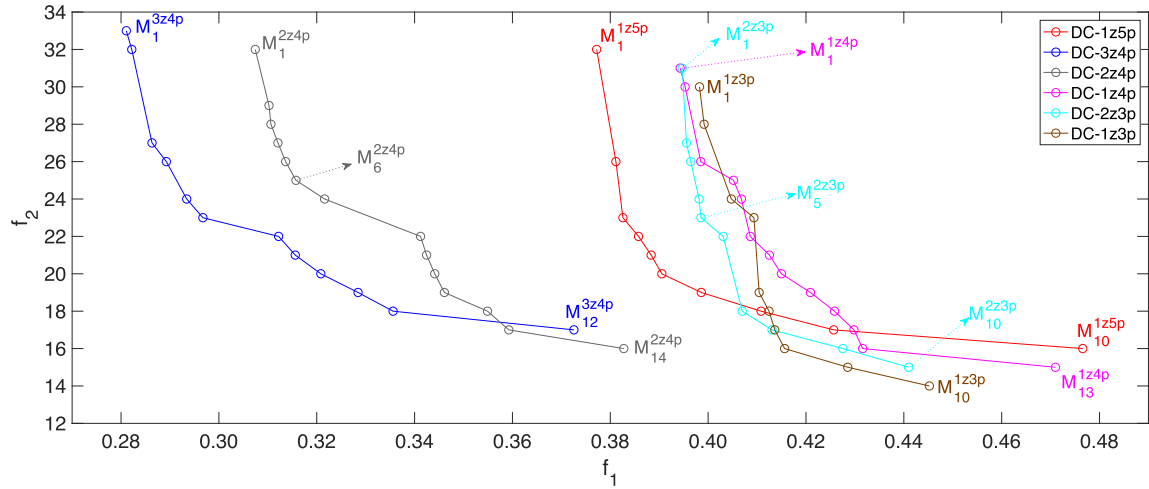


FIGURE 10. Comparison of Pareto fronts corresponding to design concepts  $DC - 1z5p$ ,  $DC - 3z4p$ ,  $DC - 2z4p$ ,  $DC - 1z4p$ ,  $DC - 2z3p$ , and  $DC - 1z3p$ .

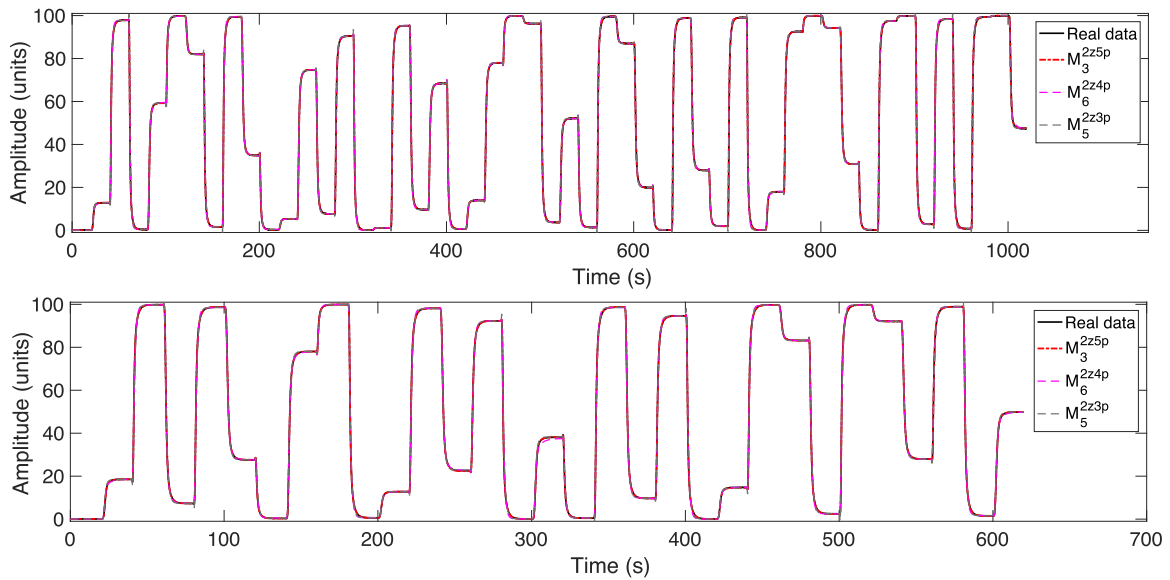


FIGURE 11. Performance of models  $M_3^{2z5p}$ ,  $M_6^{2z4p}$ , and  $M_5^{2z3p}$  on the estimation (top) and validation (bottom) data sets.

models of this front with a complexity level less than 21 were analysed, their performance would be worse than models of design concept  $DC - 1z3p$ , whose dynamic structure has one pole less.

According to the analysis presented, there are some design concepts whose models do not offer good performance. However, before discarding them, it would be essential to review their dynamic distribution. For example from the previous analysis, models from design concept  $DC - 3z4p$  are not good candidates since other models have better precision and the same complexity in the dynamic part. Conversely, if models with a Wiener or Hammerstein structure had appeared as a result of WH-MOEA execution, any of them would have been very attractive especially for control applications.<sup>1</sup>

<sup>1</sup>This is not the case in this application example since the 84 estimated models have a Wiener-Hammerstein structure

In this regard, the fronts from which models could be selected to represent the system would be those of design concepts  $DC - 2z5p$ ,  $DC - 2z4p$ ,  $DC - 2z3p$  and  $DC - 1z3p$ . If high accuracy were required, the  $M_3^{2z5p}$  model would be a very good option, while in case of requiring models with less complex dynamics,  $M_6^{2z4p}$  and  $M_5^{2z3p}$  would be good candidates. Analysing the Pareto Front of design concept  $DC - 2z4p$ , as the rate of increase in accuracy vs complexity in the  $M_6^{2z4p}$  to  $M_1^{2z4p}$  models decreases considerably, it does not make much sense to extend the points in static non-linearity. The same conclusion can be reached when analysing the Pareto front of design concept  $DC - 2z3p$ , in which the lines that join  $M_5^{2z3p}$  with  $M_1^{2z3p}$  are almost vertical, showing that the higher complexity model gives approximately the same precision. Table 8 shows the complexity and accuracy achieved by the three models indicated. The accuracy reached in the validation data set is also calculated ( $f_{1v}$ ).

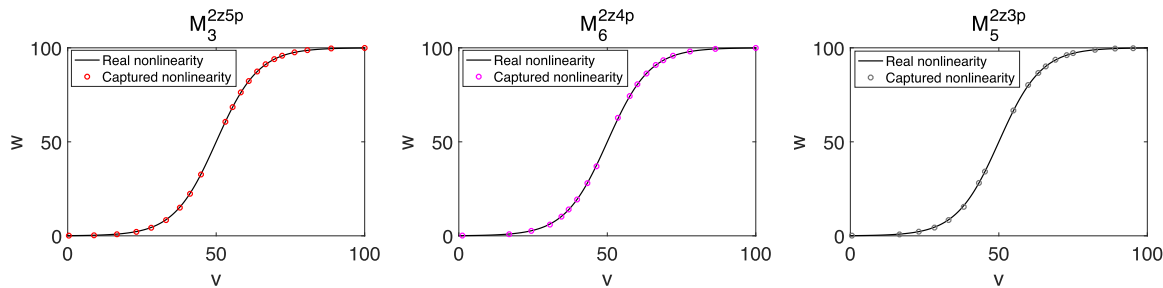


FIGURE 12. Comparison between real non-linearity and captured non-linearities for models  $M_3^{2z5p}$ ,  $M_6^{2z4p}$ , and  $M_5^{2z3p}$ .

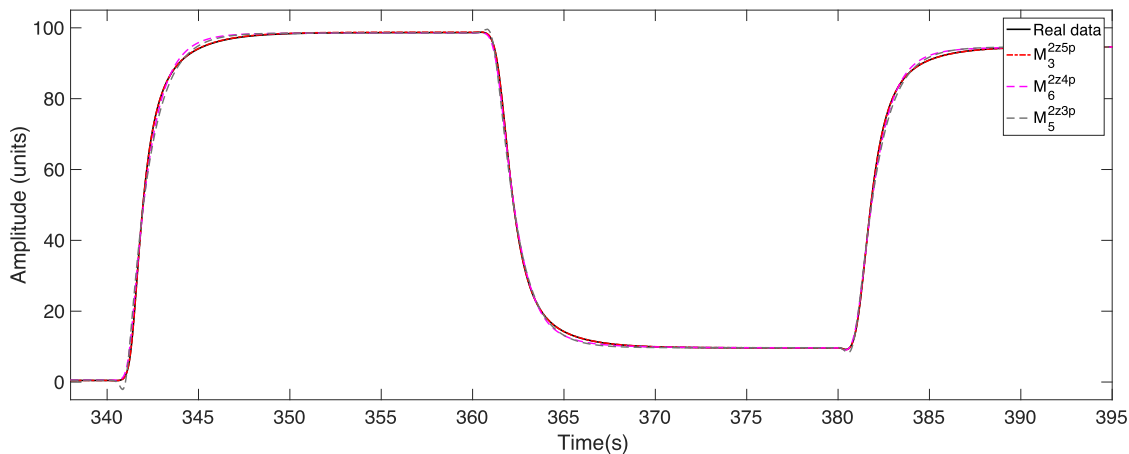


FIGURE 13. Enlargement of the transient stage of  $M_3^{2z5p}$ ,  $M_6^{2z4p}$ , and  $M_5^{2z3p}$  models with validation data.

TABLE 8. Details of the precision and complexity of models  $M_3^{2z5p}$ ,  $M_6^{2z4p}$ , and  $M_5^{2z3p}$  ( $m_{pz}$  indicates the total number of poles and zeros).

Model	$f_1$	$f_{1v}$	$f_2$	n	$m_{pz}$
$M_3^{2z5p}$	$2.550 \times 10^{-2}$	$4.510 \times 10^{-2}$	28	21	7
$M_6^{2z4p}$	$3.157 \times 10^{-1}$	$3.916 \times 10^{-1}$	25	19	6
$M_5^{2z3p}$	$3.985 \times 10^{-1}$	$4.827 \times 10^{-1}$	23	18	5

Figure 11 shows their responses compared with the process under identification. Figure 12 shows the breakpoint locations representing static non-linearity.

As can be seen in Figure 12, distribution of the points in the three models is effective and the non-linearity has been satisfactorily captured. This is reflected in Figure 11, where there are no visible inaccuracies in steady state behaviour. Differences can be seen on the transients due to non-modelled dynamics in models  $M_6^{2z4p}$  and  $M_5^{2z3p}$ . This is shown in greater detail in Figure 13, which highlights the differences. Notice that model  $M_3^{2z5p}$  reproduces the most accurate process dynamics.

Although model  $M_6^{2z4p}$  can be seen to lack a pole, its performance could be considered acceptable for certain uses. The effect of the non-modelled dynamics is more apparent in model  $M_5^{2z3p}$ , which tries to reproduce the real dynamics with the effect of the non-minimum phase.

**B. APPLICATION 2: THERMAL PROCESS**

The second example consists of a lab-scale thermal process based on a Peltier cell. The input system is the voltage ( $u_d$ )

TABLE 9. Ranking of estimated linear models for the thermal process. Models  $G_{4p}$  and  $G_{5p}$  have been excluded due to their MDL high values.

Structure	MSE ( $^{\circ}C$ )	MDL
$G_{2p}$	$3.994 \times 10^{-3}$	13.981
$G_{1z/2p}$	$3.991 \times 10^{-3}$	13.972
$G_{3p}$	$3.982 \times 10^{-3}$	13.940
$G_{1z/3p}$	$2.577 \times 10^{-3}$	9.020
$G_{1z/4p}$	$2.575 \times 10^{-3}$	9.014
$G_{1z/5p}$	$2.576 \times 10^{-3}$	9.018

applied to the Peltier cell actuator, which can vary between 0 and 4.5 Vdc. The temperature gradient between the hot and cold surfaces ( $\Delta T$ ) was considered as the output of the system. Further information on the process structure, lab test-bench, and the linear model estimation procedure can be found in previous work [28]. This non-linear real identification procedure is shown below:

*Step1 (Establish a Ranking of Candidates Using a Standard Linear Identification Method):* The results of the linear estimation are summarised in Table 9. As in the numerical example, the MSE and the modified MDL criterion were calculated for each model.

*Step 2 (Choose the Design Concepts):* According to the modified MDL criterion and the MSE values, the best linear model is  $G_{1z/4p}$  (four poles and one zero). However, the MDL of models  $G_{1z/3p}$  and  $G_{1z/5p}$  is very similar. Model  $G_{1z/3p}$  has one pole less than the best model, so it would be interesting to compare the results obtained with the nonlinear estimation from these two linear structures. On the other hand, model

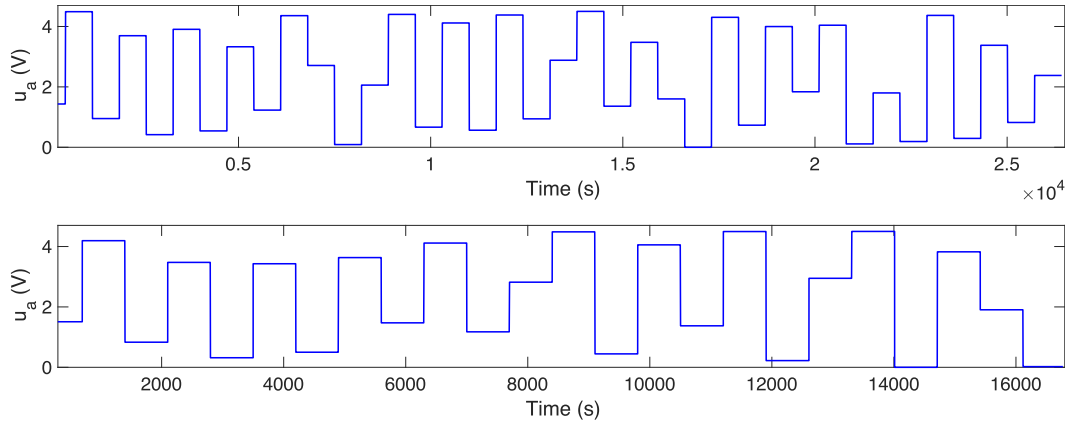


FIGURE 14. Input signals for nonlinear identification of the thermal process. Top: signal for estimation. Bottom: signal for validation.

TABLE 10. Selected initial linear models (design concepts) for nonlinear identification of thermal process.

Design Concept	Model	Poles	Zeros
DC-1z4p	$G_{1z/4p}$	[-4.461; -2.523; -0.340; -0.014]	[-0.015]
DC-1z3p	$G_{1z/3p}$	[-1.493; -0.349; -0.014]	[-0.016]
DC-1z2p	$G_{1z/2p}$	[-0.269; -0.012]	[-0.013]

TABLE 11. Summary of WH-MOEA parameter settings for nonlinear estimation of the thermal process.

Parameter	Value	Parameter	Value
MaxGen	$1.2 \times 10^6$	n_box2	10
Nind <sub>P</sub>	$4.0 \times 10^4$	n_max	14
Nind <sub>G</sub>	4	n_min	4
n_box1	50	α	0.05

$G_{1z/5p}$  will be excluded as there are two less complex models with slightly higher performance.

The remaining models  $G_{2p}$ ,  $G_{1z/2p}$ , and  $G_{3p}$  have similar performance and any of these could be a good option for obtaining less complex nonlinear models. In this case, the linear structure  $G_{1z/2p}$  will be selected. The location of the poles and zeros of the three selected structures for nonlinear estimation are shown in Table 10. As in the numerical example, each of these structures gives rise to a different design concept.

Step 3 (Prepare a Multi-Step Excitation Signal and Performs an Experiment With the Process Under Identification): Two multi-step signals were generated, one for estimation and another for validation purposes. The estimation signal was designed with 38 steps and the validation with 24 steps. The steps in both signals lasted 700s, and the amplitude changes were handled randomly within the entire range of variation of signal  $u_a$  (0 – 4.5V). To highlight the non-linearity of the process, the minimum step amplitude was constrained to 1.5V. The thermal process was excited with both signals separately. After extinguishing the transient stage of the first step, the input and output data were recorded with a sampling period of 100ms. The input signals recorded for both estimation and validation tests are shown in Figure 14.

TABLE 12. Coding of the first individual in the population and exploration bounds to refine the poles and zeros (design concepts for identification of thermal process).

DC - 1z4p	
$nc = 0, nr = 1, mc = 0, mr = 4$	
$P^{min}$	[-0.055, -4.561, -2.623, -0.440, -0.054]
$P_1^0$	[-0.015, -4.461, -2.523, -0.340, -0.014]
$P^{max}$	[0.025, -4.361, -2.423, -0.240, -0.010]
DC - 1z3p	
$nc = 0, nr = 1, mc = 0, mr = 3$	
$P^{min}$	[-0.056, -1.593, -0.449, -0.054]
$P_1^0$	[-0.016, -1.493, -0.349, -0.014]
$P^{max}$	[0.024, -1.393, -0.249, -0.010]
DC - 1z2p	
$nc = 0, nr = 1, mc = 0, mr = 2$	
$P^{min}$	[-0.053, -0.369, -0.052]
$P_1^0$	[-0.013, -0.269, -0.012]
$P^{max}$	[0.027, -0.169, -0.010]

Step 4 (Set the Bounds): The minimum and maximum values of the input and output signals were obtained from the estimation data set. These values were:  $u_{min} = 0V$ ,  $u_{max} = 4.50V$ ,  $y_{min} = -53.957^\circ C$  and  $y_{max} = -0.181^\circ C$ . As the previous example,  $y_{min}$  and  $y_{max}$  values were used to create  $W^{min}$  and  $W^{max}$ , respectively and  $u_{min}$  and  $u_{max}$  to define the variable limits  $V^{min}$  and  $V^{max}$ .

Step 5 (Run WH-MOEA for Each Design Concept): According to the selected design concepts, three bi-objective optimisation problems were stated, as in Subsection II-C. The configuration parameters in Table 11 were used for all three executions of WH-MOEA. For each design concept, the first individual in the population was coded using the poles and zeros of each initial linear model. The exploration bounds to refine poles and zeros were defined according to its location concerning the imaginary axis. In other words, the poles or zeros furthest from the imaginary axis had a greater exploration margin since they were less sensitive, while those closer to the imaginary axis had a smaller exploration margin. The codification of the first individual and the exploration bounds to refine the poles and zeros of each design concept are shown in Table 12.



TABLE 13. Optimisation results for thermal process.

$DC - 1z4p$			$DC - 1z3p$			$DC - 1z2p$		
Model	$f_1$	$f_2$	Model	$f_1$	$f_2$	Model	$f_1$	$f_2$
$M_1^{1z4p}$	0.1650	15	$M_1^{1z3p}$	0.1654	16	$M_1^{1z2p}$	0.2086	13
$M_2^{1z4p}$	0.1714	13	$M_2^{1z3p}$	0.1674	13	$M_2^{1z2p}$	0.2162	19
$M_3^{1z4p}$	0.1753	12	$M_3^{1z3p}$	0.1812	11	$M_3^{1z2p}$	0.2327	8
$M_4^{1z4p}$	0.2018	11	$M_4^{1z3p}$	0.1883	10	$M_4^{1z2p}$	0.8585	7
$M_5^{1z4p}$	0.2328	10	$M_5^{1z3p}$	0.2185	9			
$M_6^{1z4p}$	0.5999	9	$M_6^{1z3p}$	0.4371	8			

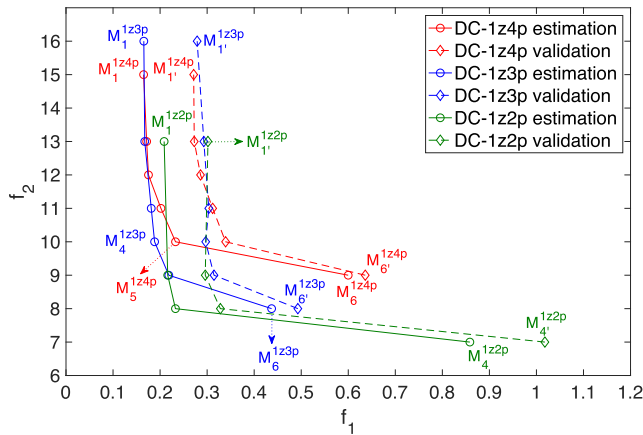


FIGURE 15. Comparison of Pareto fronts for nonlinear identification of thermal process.

After optimisation of the design concept  $DC - 1z4p$ , six Wiener-Hammerstein models were obtained. In all these models, the two fast dynamic poles were located at the front of the structure, whilst the two slow poles were located at the rear. This is consistent with the real process structure: fast poles are from the actuator dynamics while slow ones are from the Peltier cell dynamics. Design concept  $DC - 1z3p$  also resulted in six Wiener-Hammerstein models in which the fast actuator dynamics were separated from the slow process dynamics. Since this design concept does not consider the fastest actuator pole, a single-pole was placed at the front of the Wiener-Hammerstein structure. In the case of design concept  $DC - 1z2p$ , four Hammerstein models were obtained at the end of the generations. The resulting structure is consistent, since the design concept does not consider the two fast dynamic poles corresponding to the actuator.

Step 6 (Perform a Decision Making Analysis): Pareto fronts of the three design concepts are shown in Figure 15 and the model performance is quantified in Table 13. The best process representation was obtained by model  $M_1^{1z4p}$ , which has a complexity level of 15 (1 zero, 4 poles and 10 points to represent static non-linearity). Models  $M_2^{1z4p}$  and  $M_3^{1z4p}$  have similar performances to  $M_1^{1z4p}$  but with less complexity. ( $M_2^{1z4p}$  and  $M_3^{1z4p}$  represent static non-linearity with 2 and 3 less points respectively). Considering the improved precision from  $M_6^{1z4p}$  to  $M_1^{1z4p}$  as 100%, the improvement between  $M_2^{1z4p}$  and  $M_3^{1z4p}$  is only 0.89% at the cost of adding one more point to static non-linearity.

On the other hand, accuracy increased between  $M_1^{1z4p}$  and  $M_3^{1z4p}$  represents only 2.37% of the total, costing 3 points more in static non-linearity. It can therefore be concluded that it is useless to increase the static non-linearity of this design concept by more than 7 points because the improvement in precision is negligible.

The most significant increase in accuracy occurs when the complexity level changes from 9 to 10; in this case, an increase of one point in static non-linearity represents an 84.4% improvement in model accuracy. This implies that the performance of the  $M_6^{1z4p}$  model (4 points for static non-linearity) is poor and it should therefore be discarded from the selection process.

Regarding the design concept  $DC - 1z3p$ , the precision achieved by model  $M_2^{1z3p}$  (complexity level 13) is almost equal to that achieved by model  $M_1^{1z3p}$  (complexity level 16). It should be noted that an increase of more than nine points in static non-linearity does not mean a significant accuracy increase (between  $M_1^{1z3p}$  and  $M_2^{1z3p}$ , which represents only 0.73% of the total). In this design concept, the largest increase in accuracy occurs when the complexity level changes from 8 to 9 and is 90% of the total. This implies that model  $M_6^{1z3p}$ , which uses four points to represent static non-linearity, should be discarded due to its poor performance in terms of accuracy.

On comparing complexity level 10 models  $M_4^{1z3p}$  and  $M_5^{1z4p}$ , it can be seen that  $M_4^{1z3p}$  has better performance in terms of accuracy despite having a pole less than  $M_5^{1z4p}$ . This implies that it is more important to increase the number of points than to add the missing pole in models with insufficient static non-linearity points. As the number of points increases this effect is less noticeable. This can be checked by comparing models  $M_4^{1z4p}$  and  $M_3^{1z3p}$ , which have a smaller accuracy difference than the two models discussed above. Once the static non-linearity has been captured with the necessary number of points the effect of the missing pole in the second model concept is very slight, so that the most accurate model is that of the first design concept.

The Pareto front of the design concept  $DC - 1z2p$  shows the performances of four Hammerstein models. The most precise model is  $M_1^{1z2p}$  with a complexity level of 13 (10 points are used for static non-linearity). Although the complexity could have increased, the algorithm did not find any more models, since increasing static non-linearity by more than 10 points does not represent increased accuracy.

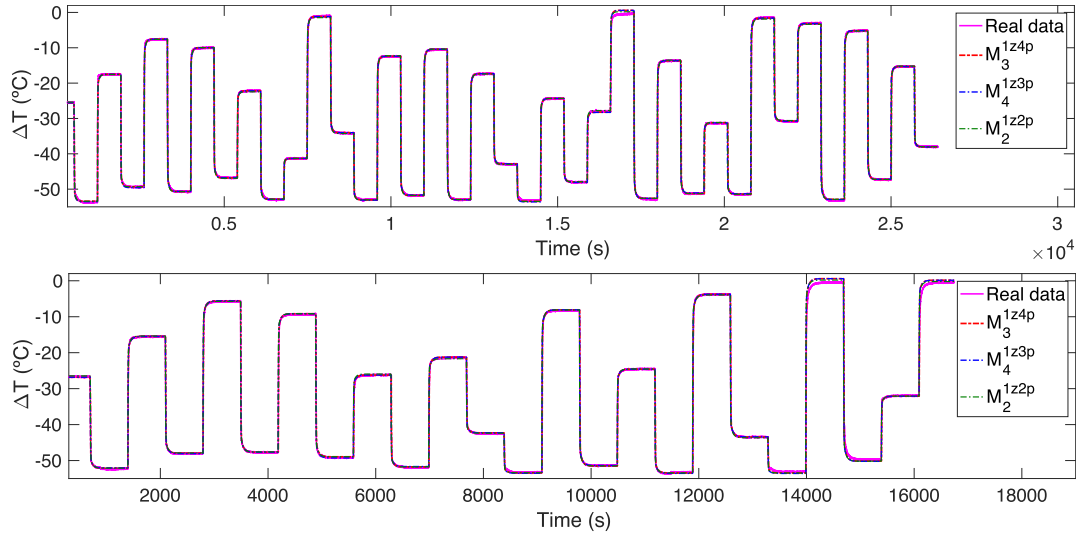


FIGURE 16. Performance of the models  $M_3^{1z4p}$ ,  $M_4^{1z3p}$ , and  $M_2^{1z2p}$  evaluated on the estimation (top) and validation (bottom) data sets.

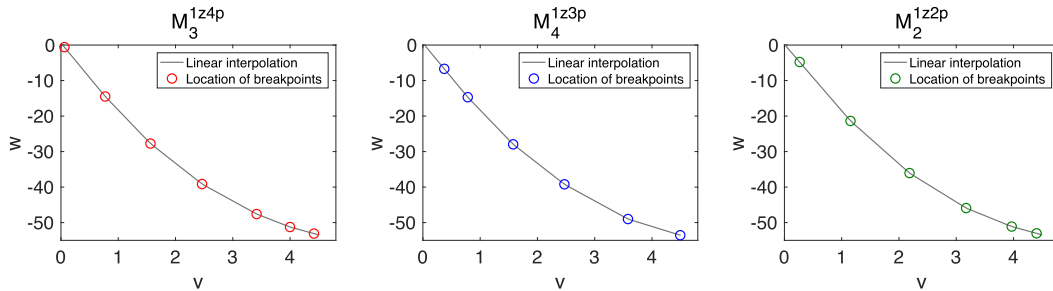


FIGURE 17. Static non-linearity of the thermal process captured by the models  $M_3^{1z4p}$ ,  $M_4^{1z3p}$ , and  $M_2^{1z2p}$ .

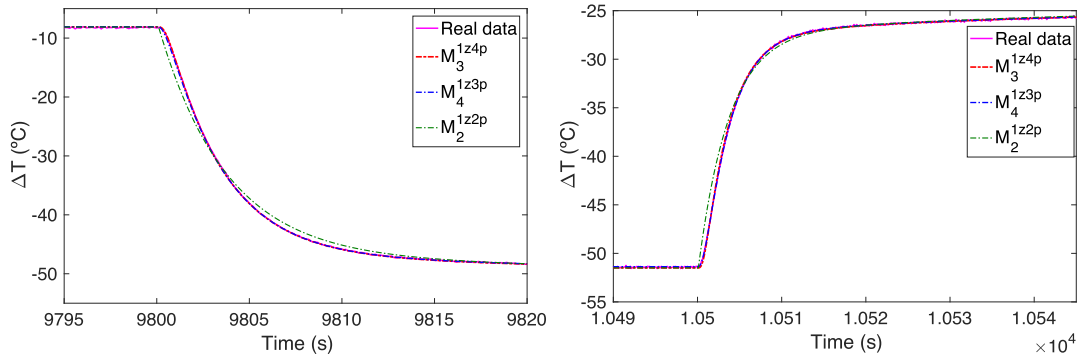


FIGURE 18. Enlargement illustrates the differences in the transient responses of  $M_3^{1z4p}$ ,  $M_4^{1z3p}$ , and  $M_2^{1z2p}$  for the validation input.

On comparing  $DC - 1z2p$  front with  $DC - 1z3p$  and  $DC - 1z4p$  fronts, one can see the effect of omitting the two fast actuator poles in the models. This shortcoming cannot even be compensated by increasing the static non-linearity points.

As final part of the analysis, the designer can evaluate the models of all fronts on a different data set (validation set) than the one used for estimation. A practical example of this is shown in Figure 15. The displacement of the three Pareto fronts (models represented by diamonds) shows lower accuracy than those achieved by the estimation data set.

The Pareto front of the first model concept can be seen to stand out more to the left, indicating that the difference between the four-pole and three-pole models has become wider in terms of precision. This particular situation can also give the designer clues to select a suitable model.

Thanks to this approach it has been possible to compare Wiener-Hammerstein and Hammerstein model structures. Clearly, more complex models will be more accurate, however it will always be important to quantify their differences and select the best, considering not only model accuracy but

also complexity. For example, if you require an adequate compromise between precision and complexity, model  $M_2^{1z2p}$  could be selected as well as  $M_4^{1z3p}$  and  $M_3^{1z4p}$ . Figure 15 shows a comparison of the performance of these models on the estimation and validation data sets, while Figure 17 shows how the break points were located to represent the static non-linearity of each model.

At first glance (Figure 16) it seems that the performances of the three selected models are the same; however an enlargement of the image (Figure 18) shows the existing differences mainly in the transient stage. Notice how model  $M_2^{1z2p}$  (green) has a small difference with the other two models with a Wiener-Hammerstein structure. If a simplified model is required,  $M_2^{1z2p}$  would therefore be a good option.

## VI. CONCLUSIONS

This paper describes a new method of identifying nonlinear block-oriented models (Wiener, Hammerstein or Wiener-Hammerstein) and demonstrates its effectiveness with a complex numerical example and a real thermal process. The procedure is based on WH-MOEA, a new multiobjective evolutionary optimisation algorithm which was formulated ad-hoc to manage this type of block-oriented model without previous knowledge about process structure. The method highlights the importance of generating a set of models with common targets and diverse performance -the generated Pareto fronts can compare and analyse the trade-off between precision and complexity. The procedure is therefore not focused on the selection of a specific model, but rather on showing the benefits of obtaining a wide range of models with different features and giving the engineer a chance to choose a final model according to his/her preferences. The WH-MOEA procedure has two important advantages: on one hand, by studying the Pareto front of a specific design concept, the influence of a variation in breakpoints for the static non-linearity on model precision can be analysed, which is more useful to the user than other methods, as it eliminates uncertainty about the number of breakpoints that must be assigned to represent static non-linearity. On the other hand, the design concept analysis can be used to compare several candidates. For example, simpler models than those recommended by the MDL criteria can be found with acceptable performance. Finally, the procedure described here contributes to solving the issue stated in many control problems, where design requirements and user preferences do not always point to model precision as the only objective, as model simplicity can also be an influential factor in decision-making.

## REFERENCES

- [1] J. DiStefano III, *Dynamic Systems Biology Modeling and Simulation*. New York, NY, USA: Academic, 2015.
- [2] R. S. Esfandiari and B. Lu, *Modeling and Analysis of Dynamic Systems*. Boca Raton, FL, USA: CRC Press, 2014.
- [3] L. Ljung, "System identification: Theory for the user," in *PTR Prentice-Hall Information and System Sciences Series*. Upper Saddle River, NJ, USA: Prentice-Hall, 1999.
- [4] R. Pintelon and J. Schoukens, *System Identification: A Frequency Domain Approach*. Hoboken, NJ, USA: Wiley, 2012.
- [5] J. Schoukens, R. Pintelon, and Y. Rolain, *Mastering System Identification in 100 Exercises*. Hoboken, NJ, USA: Wiley, 2012.
- [6] F. Giri and E.-W. Bai, *Block-Oriented Nonlinear System Identification*, vol. 1. Berlin, Germany: Springer, 2010.
- [7] M. Schoukens and K. Tiels, "Identification of nonlinear block-oriented systems starting from linear approximations: A survey," *CoRR*, vol. abs/1607.01217, 2016.
- [8] N. Wiener, "Nonlinear problems in random theory," in *Nonlinear Problems in Random Theory*, by Norbert Wiener. Cambridge, MA, USA: MIT Press, Aug. 1966, p. 142.
- [9] A. Hammerstein, "Nichtlineare integralgleichungen nebst anwendungen," *Acta Math.*, vol. 54, no. 1, pp. 117–176, 1930.
- [10] E.-W. Bai and D. Li, "Convergence of the iterative Hammerstein system identification algorithm," *IEEE Trans. Autom. Control*, vol. 49, no. 11, pp. 1929–1940, Nov. 2004.
- [11] S. Billings and S. Fakhouri, "Identification of nonlinear systems using correlation analysis and pseudorandom inputs," Dept. Control Eng., Univ. Sheffield, Sheffield, U.K., Res. Rep. ACSE Report 78, 1978.
- [12] J. Sjöberg, L. Lauwers, and J. Schoukens, "Identification of Wiener-Hammerstein models: Two algorithms based on the best split of a linear model applied to the SYSID'09 benchmark problem," *Control Eng. Pract.*, vol. 20, no. 11, pp. 1119–1125, 2012.
- [13] D. T. Westwick and J. Schoukens, "Initial estimates of the linear subsystems of Wiener-Hammerstein models," *Automatica*, vol. 48, no. 11, pp. 2931–2936, 2012.
- [14] M. Schoukens, R. Pintelon, and Y. Rolain, "Identification of Wiener-Hammerstein systems by a nonparametric separation of the best linear approximation," *Automatica*, vol. 50, no. 2, pp. 628–634, 2014.
- [15] L. Vanbeylen, "A fractional approach to identify Wiener-Hammerstein systems," *Automatica*, vol. 50, no. 3, pp. 903–909, 2014.
- [16] J. Zambrano, J. Sanchis, J. Herrero, and M. Martínez, "WH-EA: An evolutionary algorithm for Wiener-Hammerstein system identification," *Complexity*, vol. 2018, Feb. 2018, Art. no. 1753262.
- [17] Z. Hasiewicz, P. Wachel, G. Mzyk, and B. Kozdraś, "Multistage identification of Wiener-Hammerstein system," in *Proc. Polish Control Conf. Cham, Switzerland: Springer*, 2017, pp. 527–535.
- [18] G. Mzyk and P. Wachel, "Kernel-based identification of Wiener-Hammerstein system," *Automatica*, vol. 83, pp. 275–281, Sep. 2017.
- [19] D. T. Westwick and J. Schoukens, "Classification of the poles and zeros of the best linear approximations of Wiener-Hammerstein systems," *IFAC Proc.*, vol. 45, no. 16, pp. 470–475, 2012.
- [20] J. Guo and Y. Zhao, "Identification for Wiener-Hammerstein systems under quantized inputs and quantized output observations," *Asian J. Control*, pp. 1–10, 2019.
- [21] G. Giordano, S. Gros, and J. Sjöberg, "An improved method for Wiener-Hammerstein system identification based on the fractional approach," *Automatica*, vol. 94, pp. 349–360, Aug. 2018.
- [22] Y. Han and R. A. de Callafon, "Identification of Wiener-Hammerstein benchmark model via rank minimization," *Control Eng. Pract.*, vol. 20, no. 11, pp. 1149–1155, 2012.
- [23] A. H. Tan, H. K. Wong, and K. Godfrey, "Identification of a Wiener-Hammerstein system using an incremental nonlinear optimisation technique," *Control Eng. Pract.*, vol. 20, no. 11, pp. 1140–1148, 2012.
- [24] L. Ljung, "Estimating linear time-invariant models of nonlinear time-varying systems," *Eur. J. Control*, vol. 7, no. 2, pp. 203–219, 2001.
- [25] M. Enqvist and L. Ljung, "Linear approximations of nonlinear FIR systems for separable input processes," *Automatica*, vol. 41, no. 3, pp. 459–473, 2005.
- [26] A. Naitali and F. Giri, "Wiener-Hammerstein system identification—an evolutionary approach," *Int. J. Syst. Sci.*, vol. 47, no. 1, pp. 45–61, 2016.
- [27] J. Schoukens, J. Lataire, R. Pintelon, and G. Vandersteen, "Robustness issues of the equivalent linear representation of a nonlinear system," in *Proc. IEEE Instrum. Meas. Technol. Conf.*, May 2008, pp. 332–335.
- [28] J. Zambrano, J. Sanchis, J. Herrero, and M. Martínez, "A unified approach for the identification of Wiener, Hammerstein, and Wiener-Hammerstein models by using WH-EA and multistep signals," *Complex*, vol. 2020, pp. 1–23, Feb. 2020.
- [29] H. Akaike, "A new look at the statistical model identification," in *Selected Papers of Hirotugu Akaike*. Springer, 1974, pp. 215–222.
- [30] J. Rissanen, "Modeling by shortest data description," *Automatica*, vol. 14, no. 5, pp. 465–471, 1978.

- [31] F. D. Ridder, R. Pintelon, J. Schoukens, and D. P. Gillikin, "Modified AIC and MDL model selection criteria for short data records," *IEEE Trans. Instrum. Meas.*, vol. 54, no. 1, pp. 144–150, Feb. 2005.
- [32] H. Akaike, "Fitting autoregressive models for prediction," *Ann. Inst. Stat. Math.*, vol. 21, no. 1, pp. 243–247, 1969.
- [33] H. Akaike, "Autoregressive model fitting for control," in *Selected Papers of Hirotugu Akaike*. New York, NY, USA: Springer, 1998, pp. 153–170.
- [34] J. Herrero, G. Reynoso-Meza, M. Martínez, X. Blasco, and J. Sanchis, "A smart-distributed Pareto front using the ev-MOGA evolutionary algorithm," *Int. J. Artif. Intell. Tools*, vol. 23, no. 2, 2014, Art. no. 1450002.
- [35] K. M. Miettinen, *Nonlinear Multiobjective Optimization*, 1st ed. Boston, MA, USA: Kluwer, 1999.
- [36] C. A. Coello and G. B. Lamont, "Applications of multi-objective evolutionary algorithms," in *Advances in Natural Computation*, vol. 1. Singapore: World Scientific, 2004.
- [37] M. Laumanns, L. Thiele, K. Deb, and E. Zitzler, "Combining convergence and diversity in evolutionary multiobjective optimization," *Evol. Comput.*, vol. 10, no. 3, pp. 263–282, 2002, doi: 10.1162/106365602760234108.
- [38] C. Mattson and A. Messac, "Concept selection in n-dimension using s-pareto frontiers and visualization," in *Proc. 9th AIAA/ISSMO Symp. Multidisciplinary Anal. Optim.*, 2002, p. 5418.
- [39] A. Susperregui, J. M. Herrero, M. I. Martínez, G. Tapia-Otaegui, and X. Blasco, "Multi-objective optimisation-based tuning of two second-order sliding-mode controller variants for DFIGs connected to non-ideal grid voltage," *Energies*, vol. 12, no. 19, p. 3782, 2019.
- [40] V. Huilcapí, X. Blasco, J. M. Herrero, and G. Reynoso-Meza, "A loop pairing method for multivariable control systems under a multi-objective optimization approach," *IEEE Access*, vol. 7, pp. 81994–82014, 2019.
- [41] A. Pajares, X. Blasco, J. M. Herrero, and R. Simarro, "Using a multiobjective approach to compare multiple design alternatives—An application to battery dynamic model tuning," *Energies*, vol. 10, no. 7, p. 999, 2017.
- [42] L. Ljung, *System Identification Toolbox for Use with MATLAB*. Natick, MA, USA: The MathWorks, Inc., 2007.
- [43] S. M. R. Loghmanian, R. Yusof, and M. Khalid, "Nonlinear dynamic system identification using Volterra series: Multi-objective optimization approach," in *Proc. 4th Int. Conf. Modeling, Simulation Appl. Optim.*, Apr. 2011, pp. 1–5.
- [44] T. Hatanaka, N. Kondo, and K. Uosaki, "Multi-objective structure selection for RBF networks and its application to nonlinear system identification," in *Multi-Objective Machine Learning*. Berlin, Germany: Springer, 2006, pp. 491–505.
- [45] P. M. Ferreira and A. E. C. M. Ruano Fonseca, "Evolutionary multiobjective design of radial basis function networks for greenhouse environmental control," *IFAC Proc. Volumes*, vol. 38, no. 1, pp. 63–68, 2005.
- [46] S. N. Qasem and S. M. H. Shamsuddin, "Radial basis function network based on multi-objective particle swarm optimization," in *Proc. 6th Int. Symp. Mechatronics Appl.*, Mar. 2009, pp. 1–6.
- [47] F. Hafiz, A. Swain, and E. Mendes, "Multi-objective evolutionary framework for non-linear system identification: A comprehensive investigation," *Neurocomputing* vol. 386, pp. 257–280, Apr. 2020.
- [48] S. Loghmanian, M. Reza, R. Ahmad, and H. Jamaluddin, "Multi-objective optimization of NARX model for system identification using genetic algorithm," in *Proc. 1st Int. Conf. Comput. Intell., Commun. Syst. Netw.*, Jul. 2009, pp. 196–201.
- [49] C. M. Fonseca and J. P. Fleming, "Non-linear system identification with multiobjective genetic algorithms," *IFAC Proc. Volumes*, vol. 29, no. 1, pp. 1169–1174, 1996.
- [50] S. J. Qin and T. A. Badgwell, "A survey of industrial model predictive control technology," *Control Eng. Pract.*, vol. 11, no. 7, pp. 733–764, 2003.
- [51] J. C. Gomez, A. Jutan, and E. Baeyens, "Wiener model identification and predictive control of a pH neutralisation process," *IEE Proc. - Control Theory Appl.*, vol. 151, no. 3, pp. 329–338, May 2004.
- [52] S. J. Norquay, A. Palazoglu, and J. Romagnoli, "Model predictive control based on Wiener models," *Chem. Eng. Sci.*, vol. 53, no. 1, pp. 75–84, 1998.
- [53] K. P. Fruzzetti, A. Palazoglu, and K. A. McDonald, "Nonlinear model predictive control using Hammerstein models," *J. Process Control*, vol. 7, no. 1, pp. 31–41, 1997.
- [54] Ławrynczuk, M., "Nonlinear predictive control of dynamic systems represented by Wiener–Hammerstein models," *Nonlinear Dyn.*, vol. 86, no. 2, pp. 1193–1214, 2016.



**J. ZAMBRANO** (Member, IEEE) received the B.Sc. degree in electronic engineering from the Universidad Politécnica Salesiana, Ecuador, and the M.Sc. degree in automation and industrial control from the Escuela Superior Politécnica del Litoral, Ecuador, in 2007 and 2011, respectively. He is currently pursuing the Ph.D. degree with the Universitat Politècnica de València (UPV), Spain. He is also an Associate Professor with the Universidad Politécnica Salesiana (UPS). His main research interests include evolutionary optimization applied to identification and process control.



**J. SANCHIS** received the B.Sc. and Ph.D. degrees in computer science from the Universitat Politècnica de València, Spain, in 1993 and 2002, respectively. He is currently a Professor with the Department of Systems Engineering and Control, Universitat Politècnica de València. He has more than 15 years of experience applying soft computing methods to engineering problems related to process control and optimization. His main research interests are multivariable predictive control, constrained process control, and computational intelligence methods for systems engineering and process control.



**JUAN MANUEL HERRERO** received the B.S. and Ph.D. degrees in control systems engineering from the Universitat Politècnica de València (UPV), in 1999 and 2006, respectively. He is currently an Associate Professor with the Department of Systems Engineering and Automation, UPV. His main research interests are multivariable predictive control, process optimization, and computational intelligence methods for control engineering.



**M. MARTÍNEZ** (Member, IEEE) received the B.Sc. and Ph.D. degrees from the Universitat Politècnica de València, Spain. He is currently a Professor with the Department of Systems Engineering and Control, Universitat Politècnica de València, Spain. He has three years of industrial experience in control engineering in cement plants. He has published several chapters of control books, and numerous papers in conference proceedings and journals. His research interests are applied adaptive control, model predictive control, and evolutionary optimization applied to identification and process control.

...