Hindawi
Complexity
Volume 2020, Article ID 7132349, 23 pages
https://doi.org/10.1155/2020/7132349

Research Article

A Unified Approach for the Identification of Wiener, Hammerstein, and Wiener–Hammerstein Models by Using WH-EA and Multistep Signals

J. Zambrano,[1] **J. Sanchis**,[2] **J. M. Herrero**,[2] **and M. Martínez**[2]

[1] *Universidad Politécnica Salesiana, Cuenca, Ecuador*
[2] *Instituto Universitario de Automática e Informática Industrial, Universitat Politécnica de Valéncia, Valéncia, Spain*

Correspondence should be addressed to J. Zambrano; jzambranoa@ups.edu.ec

Received 1 July 2019; Revised 3 January 2020; Accepted 11 January 2020; Published 21 February 2020

Academic Editor: Marcelo Messias

Wiener, Hammerstein, and Wiener–Hammerstein structures are useful for modelling dynamic systems that exhibit a static type nonlinearity. Many methods to identify these systems can be found in the literature; however, choosing a method requires prior knowledge about the location of the static nonlinearity. In addition, existing methods are rigid and exclusive for a single structure. This paper presents a unified approach for the identification of Wiener, Hammerstein, and Wiener–Hammerstein models. This approach is based on the use of multistep excitation signals and WH-EA (an evolutionary algorithm for Wiener–Hammerstein system identification). The use of multistep signals will take advantage of certain properties of the algorithm, allowing it to be used as it is to identify the three types of structures without the need for the user to know a priori the process structure. In addition, since not all processes can be excited with Gaussian signals, the best linear approximation (BLA) will not be required. Performance of the proposed method is analysed using three numerical simulation examples and a real thermal process. Results show that the proposed approach is useful for identifying Wiener, Hammerstein, and Wiener–Hammerstein models, without requiring prior information on the type of structure to be identified.

1. Introduction

Block-oriented models are a class of nonlinear model consisting of an interaction of linear-time invariant (LTI) dynamic subsystems and nonlinear (NL) static elements [1, 2]. This interaction between basic blocks is not restricted to serial connections, and blocks can be used more than once. Therefore, block-oriented models comprise a series of structures that can be useful when modelling dynamic systems affected by memoryless nonlinearities. An overview of the most common block-oriented model structures can be found in [3]. The simplest structures within this class of model are the Wiener (LTI-NL) and Hammerstein (NL-LTI) models. Generalisations of these basic models lead to more complex structures known as Wiener–Hammerstein (LTI-NL-LTI) and Hammerstein–Wiener (NL-LTI-NL) models.

Static nonlinearities are very common in real systems, and for this reason, block-oriented models have great potential within the identification of nonlinear systems. This class of models has been successfully used in practice including biological processes [4–6]; chemical processes [7–9]; electronic systems [10, 11]; and others [12, 13]. Motivation for block-oriented models includes the identification of systems, and many reports of control applications using these types of models can be found in the literature [14–18].

In the context of block-oriented models, LTI subsystems are typically represented by pulse transfer models, impulse response models, and state space models, while nonlinearity blocks are usually parameterized with polynomials, piece wise functions, splines, neural networks, basis functions, and wavelets, among others. When LTI subsystems and NL blocks are represented with one of these forms, the model is parametric. However, block-oriented models can also be represented in a nonparametric form [19] or in a combined way [20]. This paper is concerned with parametric block-oriented models based on serial connections with a single nonlinear block, i.e., Wiener, Hammerstein, and Wiener–Hammerstein models. The latter structure provides higher modelling capabilities than Wiener and Hammerstein; however, its identification scenario is more complex due to the presence of two LTI blocks.

Knowledge of linear dynamics is a good starting point for identifying block-oriented models [3]. In this context, the *Best Linear Approximation* (BLA) of a nonlinear system is preferred [21–24]. The BLA can greatly simplify the identification problem in Wiener and Hammerstein models. In both cases, once linear dynamics are obtained, static nonlinearity can be obtained by solving an optimisation problem that is linear in the parameters. However, a more efficient estimate of Wiener and Hammerstein models may require a whole parameter refitting using nonlinear optimisation. Several identification algorithms for Wiener and Hammerstein models have been developed in the literature, and the works of Giri and Bai [2] and dos Santos et al. [25] show a good selection of algorithms.

In the case of Wiener–Hammerstein models, identification is not so easy since the BLA must be divided to establish the dynamics of both LTI blocks. Several methods to split the BLA can be found in the literature [26–33]. After the front and back dynamics of a Wiener–Hammerstein model have been classified, a fine-tuning of model parameters must be made to achieve an efficient estimate. Usually, considerable user interaction is demanded—at least two procedures are required from the BLA. In addition, the final model estimated greatly depends on these previous stages (a poor division of the BLA will obviously lead to a poor estimate). To minimise these drawbacks, a new methodology called WH-EA (evolutionary algorithm for Wiener–Hammerstein system identification) was introduced in [34]. This algorithm enables estimating all the parameters of a Wiener–Hammerstein model with a single procedure from the BLA. With WH-EA, a good estimate does not depend on intermediate procedures since the evolutionary algorithm looks for the best BLA partition, while the locations of the poles and zeros are fine-tuned and nonlinearity is captured simultaneously.

Although state-of-the-art methods for BLA splitting offer their own advantages and disadvantages, they make the identification of Wiener–Hammerstein models a subjective task with an acceptable degree of maturity. However, from a practical point of view, obtaining the BLA can be a complex and sometimes unfeasible task. On one side, multiple realisations—each with a large amount of data—may be required to obtain the BLA. In real processes with slow dynamics, experiments for obtaining the BLA would require too much time, so it would be impractical. On the other hand, excitation signals used to obtain the BLA must belong to the Riemann equivalence class of asymptotically normally distributed excitation signals [35]. The most common signals of this type are the Gaussian noise sequences and random-phased multisines [22, 36] and not all real processes can be excited with this kind of inputs.

Beyond problems derived from BLA attainment, several methods for Wiener, Hammerstein, and Wiener–Hammerstein model identification can be found in the literature [3]. Almost all have in common that they use the BLA as a starting point—although this has not been used much for Wiener and Hammerstein models. Although the three model structures are differentiated by how the dynamics are distributed around static nonlinearity, to date, there is no method to identify any of the three models without distinction. Existing methods have been developed independently and exclusively for a single structure. That is, one for identifying Wiener cannot be used to identify Hammerstein models and vice versa. If there is uncertainty about the location of the dynamics and the static nonlinearity, the user would be obliged to make separate estimates of Wiener, Hammerstein, and Wiener–Hammerstein using three different identification methods. After this tedious task, the performance of the models obtained should be compared to select the appropriate one.

At first glance, it would seem that existing Wiener–Hammerstein identification methods could easily overcome this drawback, since the Wiener and Hammerstein models are specific cases of the Wiener–Hammerstein structure where the dynamics have been distributed to only one side of the static nonlinearity. However, this situation must be handled carefully. Existing methods to identify Wiener–Hammerstein models address the problem of identification as an optimisation problem. In that case, to achieve a good convergence, it is necessary to define appropriately the range where the static nonlinearity will be captured. Since static nonlinearity is located in different positions with respect to the dynamics, the nonlinearity bounds are different for the three types of structures. Note that it is not the same to capture the static nonlinearity before or after a dynamic block given that the domain and codomain of the nonlinear function will change notably. Defining a very small search space will result in the nonlinearity not being properly captured. On the other hand, a search space too broad will cause a slow convergence, or worse, the algorithm could get stuck at a local minimum. Therefore, without beforehand information on the process structure, the search space of static nonlinearity could be defined incorrectly.

Difficulties in estimating the BLA in practical applications and the need to know, prior to the selection of an identification method, if static nonlinearity is in front, behind, or in the middle of the dynamics have been the principal motivations to present this work. The aim is to create a unified approach to estimate Wiener, Hammerstein, and Wiener–Hammerstein models without the need for the user to know a priori the structure of the process under test. This unified approach is based on the use of WH-EA without any modification. However, for WH-EA can identify any of the three structures without distinction, an effective and common search space for static nonlinearity is stated. This search space will be useful for any possible structure without the need for their dimensions to change as WH-EA distributes the dynamics. It must be taken into account that it is not an oversized search space, rather it is a search space with optimal dimensions to capture static nonlinearity regardless of the distribution of the dynamics.

In the case of Wiener, Hammerstein, and Wiener–Hammerstein models, a search space for static nonlinearity can be defined using information from the input and output dataset used during the identification procedure. However, when an arbitrary excitation signal (e.g., a Gaussian signal) is applied, the process structure must be known to define an effective search space. Since this work assumes that the process structure is unknown, from the input and output dataset a common search space useful for the three structures will be defined. As it will be seen in Section 2.4, this common search space will be possible as long as the applied excitation signal leads the output of the process to steady state and for this reason, multistep signals will be used. In addition, multistep excitation will enable an effective exploration of different process operation zones highlighting existing nonlinearities (not possible if Gaussian signals are used to excite the process). Note that Gaussian signals are useful for capturing the dynamic behavior of a system, however, these types of signals are not suitable for highlighting static non-linearites - as in the case of saturations.

Since this approach will not use Gaussian-type signals, an initial linear model obtained using standard linear identification methods can be used instead of the BLA. In noisy environments and under the effect of nonlinearity, the initial linear model will be a biased version of the real process dynamics. As it will be shown, the potentiality of WH-EA is exploited to refine the location of the poles and zeros of this initial model while they are distributed around nonlinear block (which is also captured simultaneously).

Unlike other proposals where a single type of model is addressed, this paper presents a new approach that allows the identification of any of the three types of block-oriented models indistinctly without any beforehand information about the type of structure. This approach is useful for identifying nonlinear systems, where it is known a priori that the system is affected by a static nonlinearity but its location with respect to dynamics is unknown. From the core idea of this paper, other derived novelties are highlighted below:

(i) In this proposal, the BLA is not used. This is a significant advantage since the estimation of the BLA can be impractical in many real applications due to the execution time required to excite the process under test.

(ii) This approach uses multistep excitation signals. This type of signals allows nonlinearities to emerge better. This is very useful since nonlinear estimation starts from dynamics already known.

(iii) Thanks to the normalisation of the dynamics and the use of multistep signals a common search space for the three types of models can be stated. This search space is not dependent on any parameter provided by the user, such as it was the case of the Ω parameter in [34].

(iv) The estimation is done in continuous time, which gives the user a clearer view of the process behaviour under test.

The rest of this paper is organised as follows. Section 2 presents the identification framework that has been divided into five parts. First, the structure of the Wiener, Hammerstein, and Wiener–Hammerstein models and their mathematical formulation are described. The initial linear model estimation and selection of its structure is then addressed, followed by the optimisation problem statement. The search space for static nonlinearity is then analysed for these three types of models when creating a common search space. Finally, several aspects related to multistep inputs, such as excitation signals, are presented. Section 3 presents an abstract of WH-EA including codification of the individuals, its genetic operators, and details of how the algorithm works. To end, in Section 4, the presented methodology is applied to three numerical examples and a real application (a thermal process). Finally, concluding remarks are presented in Section 5.

2. Identification Framework

2.1. Structure of Nonlinear Models and Problem Formulation. All the three block-oriented models treated in this work have a single nonlinear element. In the case of the Wiener–Hammerstein models, two LTI blocks $G_w(s, \boldsymbol{\rho}_w)$ and $G_h(s, \boldsymbol{\rho}_h)$ surround the nonlinear element $f(v(t), \boldsymbol{\rho}_{nl})$. Wiener and Hammerstein models are specific cases of Wiener–Hammerstein models when one of the linear blocks lacks dynamics. If dynamics are present only at the input linear block, the resulting model is known as a Wiener model. When dynamics are present only at the output block, the resulting model is known as a Hammerstein model.

In this paper, the block-model structure of the process to be identified is unknown, so the most general form is considered as a starting point for the problem formulation. Let us represent Wiener, Hammerstein, and Wiener–Hammerstein models by

$$y(t, \boldsymbol{\rho}) = G_h(s, \boldsymbol{\rho}_h) f(G_w(s, \boldsymbol{\rho}_w) u(t), \boldsymbol{\rho}_{nl}), \tag{1}$$

$$\boldsymbol{\rho} = [\boldsymbol{\rho}_w, \boldsymbol{\rho}_{nl}, \boldsymbol{\rho}_h], \tag{2}$$

where $u(t)$ and $y(t)$ are the model input and output, respectively, and vectors $\boldsymbol{\rho}_w$ and $\boldsymbol{\rho}_h$ contain the parameters of the dynamic blocks, while $\boldsymbol{\rho}_{nl}$ contains the parameters of static nonlinearity.

Notice that equations (1) and (2) correspond to the formulation of Wiener–Hammerstein models or generic case. In the case of Wiener models, vector $\boldsymbol{\rho}_h$ does not exist and $G_h = 1$, whereas in the case of Hammerstein models, vector $\boldsymbol{\rho}_w$ does not exist and $G_w = 1$.

This paper establishes a common framework for the identification of Wiener, Hammerstein, and Wiener–Hammerstein models that is only possible under certain constraints that are detailed in Section 2.4. For all three cases, the identification problem starts from (1) and is addressed as a classification problem. The evolutionary algorithm will determine if there are dynamics distributed between the two blocks or if the dynamics are present just in one of them.

For the two LTI blocks to be parameterized, both LTI subsystems are represented in the continuous time domain as rational transfer functions in factorised form (zero-pole-gain):

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

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

where $-p_{w_i}$ with $i = 1, \ldots, n_a$ and $-z_{w_i}$ with $i = 1, \ldots, n_b$ represent front LTI poles and zeros, respectively. In a similar way, $-p_{h_i}$ with $i = 1, \ldots, n_c$ and $-z_{h_i}$ with $i = 1, \ldots, n_d$ represent poles and zeros of the back LTI one. Static gains of each linear block are represented by K_w and K_h, while s is the complex Laplace variable. Considering this, let us define vectors $\boldsymbol{\rho}_w$ and $\boldsymbol{\rho}_h$ as

$$\boldsymbol{\rho}_w = \left[K_w, z_{w_1}, z_{w_2}, \ldots, z_{w_{nb}}, p_{w_1}, p_{w_2}, \ldots, p_{w_{na}}\right], \quad (5)$$

$$\boldsymbol{\rho}_h = \left[K_h, z_{h_1}, z_{h_2}, \ldots, z_{h_{nd}}, p_{h_1}, p_{h_2}, \ldots, p_{h_{nc}}\right]. \quad (6)$$

Notice that poles and zeros in (3) and (4) are not restricted to be real, since $-p_{w_i}$, $-p_{h_i}$, $-z_{w_i}$, and $-z_{h_i}$ can also represent complex poles or zeros, respectively.

Static nonlinearity can also be represented in different ways. In this case, piecewise functions are used as WH-EA uses them:

$$w(t) = f(v(t), \boldsymbol{\rho}_{nl}), \quad (7)$$

where $v(t)$ is the input signal to the nonlinear block, while $\boldsymbol{\rho}_{nl}$ contains the abscissas and ordinates which define the breakpoint locations of the piecewise function. Notice that for a Wiener system, $y(t) = w(t)$, while for a Hammerstein system, $v(t) = u(t)$.

The problem formulation is completed by the following assumptions:

A1. The model to be identified corresponds to a Wiener, Hammerstein, or Wiener–Hammerstein system, where the structure is unknown but the general dynamics must be known.

A2. There is no cancellation of poles and zeros and the location of the poles is consistent with a stable system.

A3. The system under test will be identified from an input/output dataset, where the input excitation signal $u(t)$ is a multistep signal (see Section 2.5 for more details), while the measured output $y(t)$ may be corrupted by stationary additive noise $n(t)$:

$$y(t) = y_0(t) + n(t). \quad (8)$$

2.2. Initial Linear Model. In our context of Wiener–Hammerstein models, obtaining a perfect linear dynamic model in the presence of noise and nonlinearities is not an easy task; however, gathering an overview of system dynamics can be a good starting point. The BLA is an option that has been used generally in the estimation of Wiener–Hammerstein models. From a theoretical point of view, the fastest and most robust method to find the BLA hides the effect of noise and nonlinearities, and so the dynamics can be captured with great precision [22]. However, from a practical point of view, obtaining the BLA is not always possible or may require the use of multiple realisations, especially when the robust method is used.

In practical applications, the BLA can present a lack of accuracy, once its poles/zeros have been distributed and the nonlinearity captured, and refinement of the dynamics is always possible to improve model preciseness. In this work, it is assumed that the initial linear model is not perfect but it can be fine-tuned during estimation. The initial model can be obtained as usual from the response to a step signal. The process under test can be excited with a small amplitude step avoiding excitation of the nonlinearity. Due to its static nature, any process operating point can be selected to inject the step signal. Estimated models in different operation zones will give similar dynamics but with different static gains (it is advisable to avoid zones near operation limits since the nonlinearity can be stronger due to saturation phenomena).

The purpose of this paper is not to discuss methods for linear system identification. For a direct estimation in continuous time and to obtain models with better precision, *simple refined instrumental variable method for continuous-time models* (SRIVC) has been used [37, 38], available in the CONtinuous-Time System IDentification (CONTSID) toolbox for Matlab [39–41].

Since that initial linear model estimation is based on a step response, it is assumed that a small amount of data will be used. For this reason, the *Modified minimum Description Length* (MDL) criterion has been used to select the best linear structure [42]:

$$\text{MDL} = \frac{V(\boldsymbol{\rho}\mathbf{l}, \mathbf{Z})}{N_L} e^{p_c (n_{\rho l}, N_L)}, \quad (9)$$

where $\boldsymbol{\rho}\mathbf{l}$ are the estimated model parameters, $\mathbf{Z}$ is a two-dimensional vector containing the input/output data, $n_{\rho l}$ is the number of parameters in the estimated model, N_L represents the amount of data used for estimation, $V(\boldsymbol{\rho}\mathbf{l}, \mathbf{Z})$ is the quadratic-like cost function depending on the difference between measurements and model (ϵ), computed using (10), and $p_c(n_{\rho l}, N_L)$ is the term known as the corrected penalty and is computed using (11).

$$V(\boldsymbol{\rho}\mathbf{l}, \mathbf{Z}) = \left(\frac{1}{2}\right) \epsilon^T \epsilon, \quad (10)$$

$$p_c(n_{\rho l}, N_L) = \frac{\ln(N_L)(n_{\rho l} + 1)}{N_L - n_{\rho l} - 2}. \quad (11)$$

2.3. Optimisation Problem Statement. From the initial linear model and an input/output dataset $\{u(t), yr(t)\}_{t=1}^{N}$, WH-EA is used to find the best set of parameters that represent the nonlinear model. The procedure includes the refinement of the initial linear model, the characterisation of static nonlinearity, and the pole/zero distribution of the initial linear model around the static nonlinearity. The best set of parameters is assigned to a model of Wiener, Hammerstein, or Wiener–Hammerstein. For this purpose, the identification problem is stated as an optimisation problem based on a prediction-error method and the mean absolute error criterion (notice that any other criterion could be used in the proposed method, such quadratic or maximum error criteria):

$$\varepsilon_{\mathrm{NL}}(t,\boldsymbol{\rho}) = yr(t) - y(t,\boldsymbol{\rho}), \tag{12}$$

$$J(\boldsymbol{\rho}) = \frac{1}{N}\sum_{t=1}^{N} \mathrm{abs}\left(\varepsilon_{\mathrm{NL}}(t,\boldsymbol{\rho})\right). \tag{13}$$

The solution of the optimisation problem is written as

$$\widehat{\boldsymbol{\rho}} = \arg\min_{\boldsymbol{\rho}} J(\boldsymbol{\rho}), \tag{14}$$

where $\widehat{\boldsymbol{\rho}}$ contains the best set of parameters to represent the nonlinear model.

2.4. Search Space for Static Nonlinearity. In the present approach, the problem of identifying Wiener, Hammerstein, and Wiener–Hammerstein models is addressed as an optimisation problem that is solved with WH-EA. For the algorithm to converge successfully and the best model to be estimated, it is necessary to define a suitable search space for static nonlinearity. The minimum and maximum values of the input and output signals of the nonlinear block give a clear idea of the domain and codomain of the static nonlinearity; therefore, from this information, it is possible to define its search space. However, it is necessary to point out that the minimum and maximum values of the input signal to the nonlinear block depend on the excitation signal used and the location of the nonlinearity around the dynamics of the process, while the minimum and maximum values of the output signal depend on the input signal to the block and the nonlinearity itself. This can be clearly seen in Figure 1, where a Gaussian signal has been used to excite three models containing the same dynamics and the same static nonlinearity. These models differ only in the distribution of the dynamic that has intentionally been handled to give rise to the three structures that are addressed in this paper.

In the case of Wiener and Wiener–Hammerstein models, the limits that define the horizontal search space of the nonlinear static function are affected by the static gain and the dynamics of the linear input block. Since the linear blocks of these two models are different, the limits are also different. For example, for the Wiener–Hammerstein model defined in Figure 1, the limits that horizontally define the search space for static nonlinearity are −1.68 and 1.85, while for the Wiener model, the limits are −1.10 and 1.12. In the case of the Hammerstein model, these limits could be obtained directly from the minimum and maximum values of the excitation signal (−2.38 and 2.43). It is evident that the limits that horizontally define the search space are different for the three types of models. This difference is also reflected in the vertical limits—even though the three models have the same static nonlinearity. The fact that there are different search spaces makes it necessary to know a priori the process structure under test in order to define an adequate search space. If it is not possible to know the process structure, an oversized search space could be defined; however, this will surely complicate the convergence of any search algorithm.

This section shows how to create a unified search space for the three types of models. This search space is independent of the distribution of the dynamics, so the search algorithm will not be restricted to estimating a certain structure. In other words, thanks to the creation of this unified search space, WH-EA will be able to estimate Wiener, Hammerstein, and Wiener–Hammerstein models without the need for the user to specify a priori the process structure.

For a better understanding, prior to explaining how to create a unified search space, in the first instance, it is shown how to determine the search space of the three types of models assuming an arbitrary excitation signal (e.g., a Gaussian signal). For all three cases, it is assumed that the intermediate signals are not measurable and the dynamic blocks are nonreversible, and therefore the only way to determine the search space is by using information from the input and output data and from the initial linear model.

Let us assume, for the three cases of analysis, that the input signal $u(t)$ is bounded by a maximum value $u_{\max}$ and a minimum value $u_{\min}$ with a mean value of u_{mean}. In the same way, the output $y(t)$ is bounded by a maximum value $y_{\max}$ and a minimum value $y_{\min}$, and it has a mean y_{mean}.

2.4.1. Search Space in Wiener–Hammerstein Models. In a Wiener–Hammerstein model, the excitation signal $u(t)$ enters the first LTI block ($G_w(s)$). This block will produce an output $v(t)$ with mean v_{mean}, bounded by a maximum value $v_{\max}$ and minimum value $v_{\min}$. The relationship between minimum and maximum values of signals $u(t)$ and $v(t)$ is determined by

$$v_{\min} = \Omega * u_{\min}, \tag{15}$$

$$v_{\max} = \Omega * u_{\max}, \tag{16}$$

where Ω is a scaling factor depending on the block $G_w(s)$. Without loss of generality, the static gain of $G_w(s)$ can be normalised to one, since the real gain can be absorbed by static nonlinearity. Under this normalisation, it can be assumed that there will be no offset between input and output signals since $G_w(s)$ is an LTI subsystem; therefore, $v_{\mathrm{mean}} = u_{\mathrm{mean}}$. The same can be applied to the output block $G_h(s)$; therefore, $w_{\mathrm{mean}} = y_{\mathrm{mean}}$, where w_{mean} is the mean of the signal $w(t)$.

The search space for static nonlinearity in a Wiener–Hammerstein model will be horizontally bounded by $v_{\min}$ and $v_{\max}$. To compute these parameters, $u_{\min}$ and $u_{\max}$ can be obtained directly from the input signal $u(t)$, whereas Ω would be a user-defined parameter. Selection of this parameter will be addressed later in this same section.

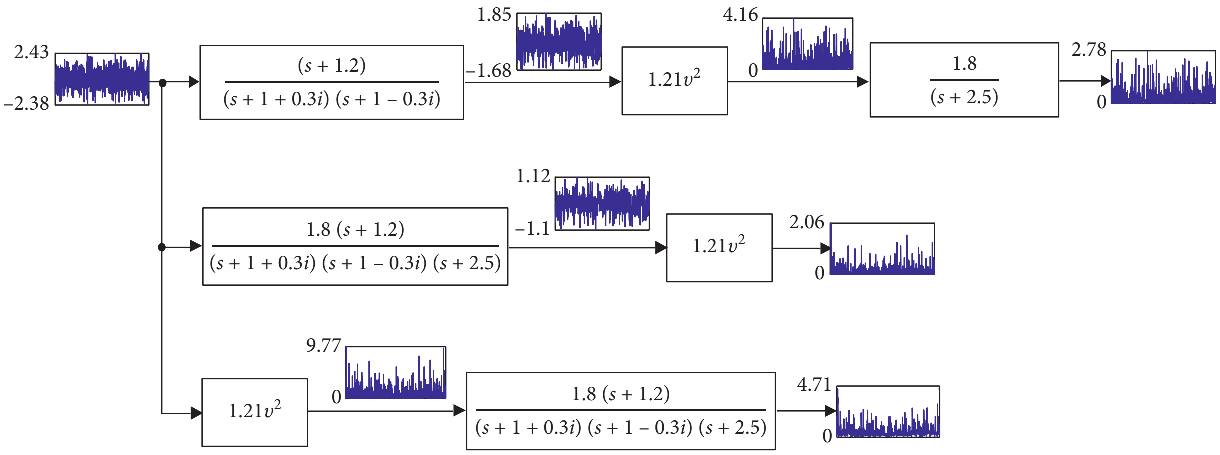

2.43
−2.38
$\frac{(s+1.2)}{(s+1+0.3i)(s+1-0.3i)}$
1.85
−1.68
$1.21v^2$
4.16
0
$\frac{1.8}{(s+2.5)}$
2.78
0
$\frac{1.8(s+1.2)}{(s+1+0.3i)(s+1-0.3i)(s+2.5)}$
1.12
−1.1
$1.21v^2$
2.06
0
$1.21v^2$
9.77
0
$\frac{1.8(s+1.2)}{(s+1+0.3i)(s+1-0.3i)(s+2.5)}$
4.71
0

Figure 1: Example of search space limits for static nonlinearity in Wiener, Hammerstein, and Wiener–Hammerstein models when they are excited with a Gaussian signal.

Search space for nonlinearity is vertically delimited by $w_{\min}$ and $w_{\max}$ corresponding to the minimum and maximum values of signal $w(t)$. The linear dynamic model complements the information required to find this values. The static gain of this model corresponds to the slope of the straight line (K_{NL}) passing through the point $(v_{\text{mean}}, w_{\text{mean}})$ and the extreme points $(v_{\min}, w_{\min})$ and $(v_{\max}, w_{\max})$. Therefore, $w_{\min}$ and $w_{\max}$ can be found using (17) and (18) (it should be noted that if the linear dynamic model has negative static gain, the search space for static nonlinearity would be delimited by the coordinate pair $(v_{\min}, w_{\max})$ and $(v_{\max}, w_{\min})$). The search space for the static nonlinearity of a Wiener–Hammerstein model is illustrated in Figure 2.

$$w_{\min} = y_{\text{mean}} + K_{\text{NL}}(\Omega u_{\min} - u_{\text{mean}}), \quad (17)$$

$$w_{\max} = y_{\text{mean}} + K_{\text{NL}}(\Omega u_{\max} - u_{\text{mean}}). \quad (18)$$

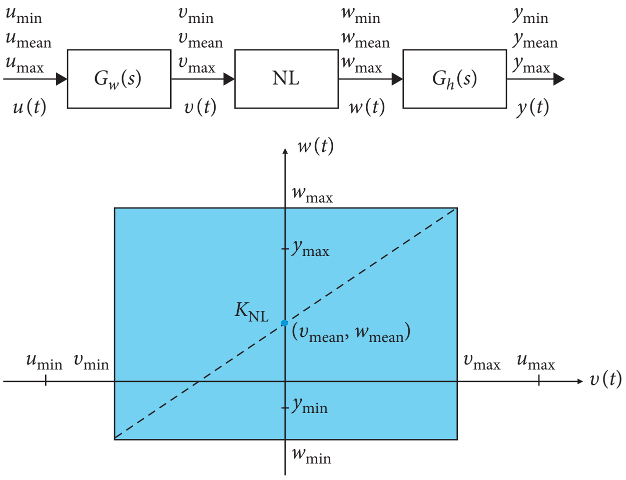


Figure 2: Search space for nonlinear static function in a Wiener–Hammerstein model.

2.4.2. Search Space in Wiener Models. In this case, the input signal $u(t)$ produces an output $v(t)$ and nonlinearity search space is horizontally bounded by $(v_{\min}, v_{\max})$, whereas vertical bounds will be given by $y_{\min}$ and $y_{\max}$.

It is well known that the identification of Wiener models is not as complex as the identification of Wiener–Hammerstein models. In a Wiener identification, once the linear block is known, the signal $v(t)$ can be obtained directly; therefore, to define the search space for static nonlinearity, it would not be necessary to use (15) and (16). However, estimation of the intermediate variable $v(t)$ can be useful when dealing with Wiener models. This approach assumes that the distribution of the dynamics around nonlinearity is unknown; therefore, it is not possible to estimate $v(t)$, rather it is necessary to establish a search space for nonlinearity that is common for all three structures.

To define the horizontal bounds of the search space of a Wiener model, without loss of generality, we could follow the same guidelines that were followed for Wiener–Hammerstein models, that is, values of $v_{\min}$ and $v_{\max}$ can be determined with (15) and (16), considering the scaling factor Ω. The static nonlinearity search space for a Wiener model is shown in Figure 3. The extremes of the search space give rise to the straight line whose slope K_{NL} must match the static gain of the initial linear dynamic model.

2.4.3. Search Space in Hammerstein Models. In a Hammerstein model, the input signal $u(t)$ enters the nonlinear block; therefore, $u_{\min}$ and $u_{\max}$ horizontally define the search space for static nonlinearity, while vertical bounds are defined by $w_{\min}$ and $w_{\max}$. To estimate the intermediate variable $w(t)$, the dynamic block needs to be invertible, which is impossible from a practical point of view. Furthermore, our approach assumes that pole/zero distribution around nonlinearity is unknown; therefore, for the sake of establishing a common search space for the three structures, $w_{\min}$ and $w_{\max}$ can be determined following the same procedure that was used for Wiener–Hammerstein models. The search space for the static nonlinearity of a Hammerstein model is illustrated in Figure 4.

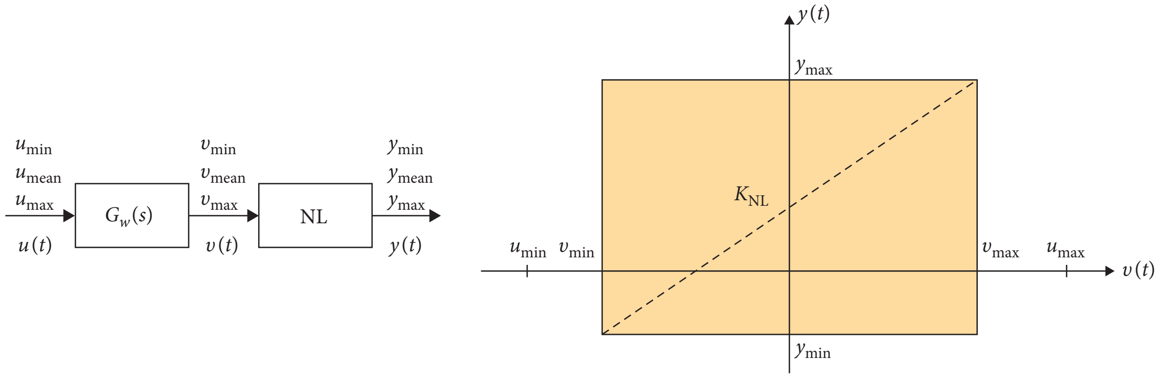


Figure 3: Search space for nonlinear static function in a Wiener model.

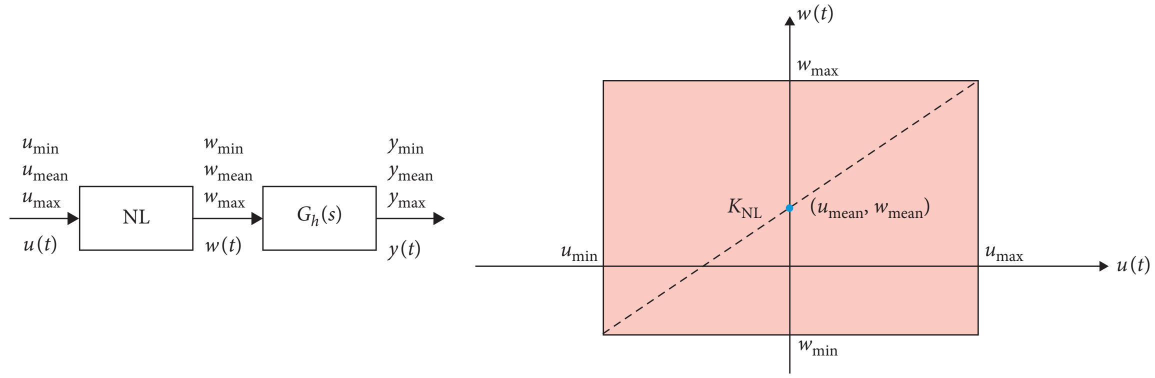


Figure 4: Search space for nonlinear static function in a Hammerstein model.

2.4.4. A Common Search Space Definition for Wiener, Hammerstein, and Wiener–Hammerstein Models. According to previous sections, when an arbitrary signal $u(t)$ excites the system (for example, a Gaussian signal), horizontal and vertical limits that define the search space for static nonlinearity are different for the three types of models. This fact implies that the identification algorithm should change the search space over which the nonlinearity is captured at the same time that distributes the dynamics. To solve this drawback, a common and fixed search space for the three types of structures will be defined. To achieve a common search space, it is necessary that both horizontal and vertical limits of the search space for each model are the same. Therefore, it is necessary that in the Wiener and Wiener–Hammerstein models, when an excitation signal $u(t)$ is applied, the dynamics and static gain of the $G_w(s)$ block cause an output signal $v(t)$ whose minimum and maximum values are equal to the minimum and maximum values of $u(t)$, respectively. That is, $v_{\min} = u_{\min}$ and $v_{\max} = u_{\max}$.

With the static gain of $G_w(s)$ normalised to one, the amplitude of the signal $v(t)$ will only be affected by the dynamics present in this linear block. The effect of the dynamics present $G_w(s)$ on $v(t)$ is represented by the Ω factor. According to (15) and (16), so that $v_{\min} = u_{\min}$ and $v_{\max} = u_{\max}$, Ω must be one. However, Ω cannot take any value without taking into account the input signal. For example, if a Gaussian signal is used to excite the system, the output of $G_w(s)$ will be modified in amplitude and the corresponding minimum and maximum values of $u(t)$ and $v(t)$ will be different. However, if an input causes the output of $G_w(s)$ in a Wiener or Wiener–Hammerstein model to reach steady state, both amplitudes will be coincident since $G_w(s)$ has unity gain.

Similarly, vertical bounds must be coincident to achieve a common search space for the three types of models. For this to occur, amplitudes of $w(t)$ and $y(t)$ must be equal. If $u(t)$ brings to $y(t)$ at steady state, normalising the static gain of $G_h(s)$ to 1 would mean that vertical bounds coincide for the three cases. A good option to obtain the output of a dynamic system at steady state is to apply step inputs with sufficient duration. Figure 5 shows how the horizontal and vertical limits that give rise to the search space of static nonlinearity are the same for the three types of models. The models used are the same as in Figure 1; the difference is that the static gains of the dynamic blocks in Figure 5 have been normalised to 1. A multistep signal has been used to excite the three models. The step duration ensures that the response of the three models reaches steady state at each step change. As can be seen, the limits that define horizontally the search space of the three models are 0 and 4.68. One great advantage of having a unified search space is that these limits can be obtained directly from the minimum and maximum values of the input signal. Similarly, the limits that vertically define the search space are the same for the three models (0 and 26.5). These limits can be obtained directly from the minimum and maximum values of the output signal.

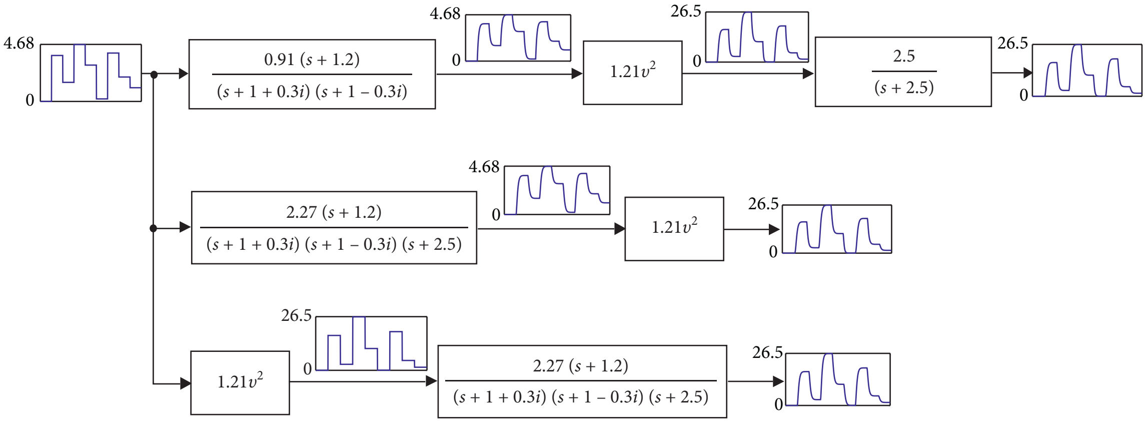


FIGURE 5: Example of search space limits for static nonlinearity in Wiener, Hammerstein, and Wiener–Hammerstein models when they are excited with a multistep signal.

A good option to obtain the output of a dynamic system at steady state is to apply step inputs of sufficient duration. In Section 2.5 more details on how to design this signal will be given.

With the above discussion, the common search space for the three types of models can be constructed directly from input $u(t)$ and output $y(t)$ measurements. If the gain of the initial linear model is positive, the search space will be defined by coordinates $(u_{\min}, y_{\min})$ and $(u_{\max}, y_{\max})$, while if it is negative, the search space will be defined by $(u_{\min}, y_{\max})$ and $(u_{\max}, y_{\min})$.

2.5. The Multistep Signal. Previous sections state that the process under test must be excited with an input based on steps. Step duration must be long enough for the process to reach steady state. Since it is intended to capture a nonlinearity that is present throughout the entire process operating range, it will be necessary to design a multistep signal with different amplitudes.

For this aim, three important aspects must be considered: step duration; number and amplitude of steps; and the minimum difference between two consecutive steps. All the steps of the excitation signal can have a fixed duration, based on the process dynamics under test. This duration can be easily established based on the initial tests in which the initial linear model was obtained.

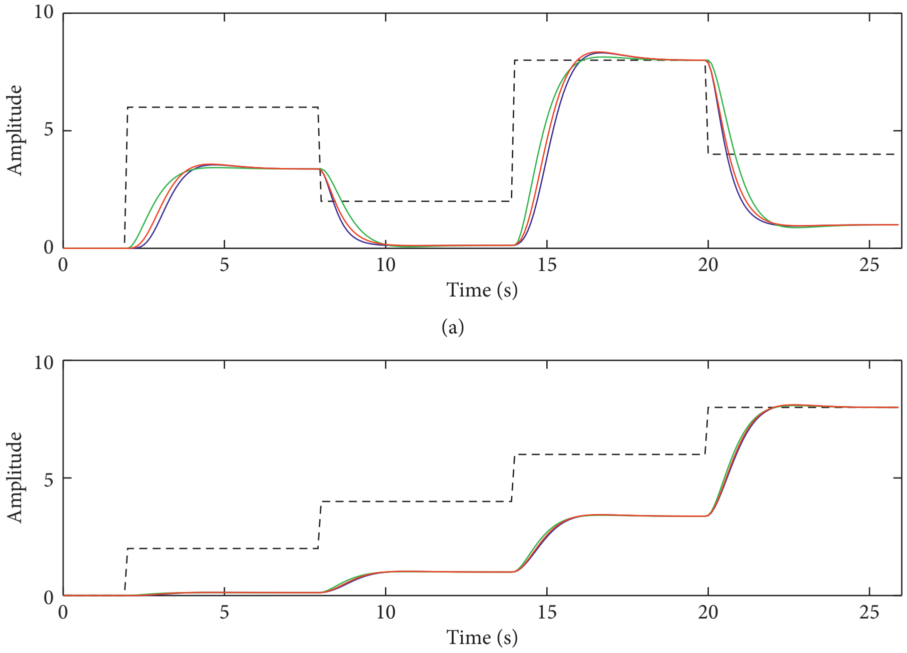
A very small amount of data could mean that nonlinearity is not captured correctly and the dynamics will not be distributed properly. A large amount of data would lead to a satisfactory estimate, but could demand an important computational cost. How much data need to be used for identification of a nonlinear model deserves debate, and a vast majority of nonlinear model identification methods require a large volume of input and output data.

Calculating the static nonlinearity with precision will lead to a good dynamic classification. Therefore, an effective exploration of the entire process operating range will be required, and step amplitudes must change within input limits by varying randomly, and the number of changes will depend on the desired precision. Furthermore, the minimum difference between two consecutive steps should also be considered when designing the multistep signal. Amplitude changes of the steps will give rise to transitory stages, which contain information to classify the dynamics. If they are very small, these transitory intervals will not contain substantial information for the classification. A suitable scenario to classify the dynamics is achieved when the nonlinearity is visible. Therefore, amplitude changes of the steps must be large to highlight nonlinearity.

Figure 6 reflects this fact through a numerical simulation example. Four operating points of the system are explored for two scenarios (large/small step input changes) where the same static nonlinearity and the same dynamic have been considered. The nonlinearity consists of a cubic function $(1/64x^3)$, while the dynamic is formed with three poles $(-2.4;\ -1.5 + 0.856i;\ -1.5 - 0.856i)$ and a real zero (-1.56). For each case, three simulations were executed corresponding to Wiener, Hammerstein, and Wiener–Hammerstein models and dynamic blocks were normalised with unit static gain (for the Wiener–Hammerstein model, the zero and the complex poles were placed before the nonlinearity, while the remaining pole was placed afterwards).

In Figure 6(b), no difference between the responses can be seen when the excitation signal has small amplitude changes. Conversely, Figure 6(a) shows a marked difference between responses when the excitation signal has larger amplitude changes. Table 1 shows the differences between responses as mean absolute error (MAE) and reveals the advantage of using excitation signals with large amplitude changes. This means that the identification algorithm has more information to distinguish if the dynamics are in front, behind, or distributed on both sides of the nonlinearity. Notice how low MAE_{ss} values imply no significant difference between the structures formed as the algorithm cannot split the dynamics properly.

Multistep signals are ideal to highlight the nonlinearities of a process; however, this type of signal has some limitations that must be evaluated by the user prior to the estimation of a process. A multistep signal is enabled to excite the dominant dynamics of a process. In contrast, a well-designed Gaussian signal or equivalent is enabled to excite all the oscillatory



(a)

(b)

Figure 6: Responses of Wiener (blue), Hammerstein (green), and Wiener–Hammerstein (red) models when they are excited with a multistep signal (dashed) with large (a) and small (b) amplitude changes.

Table 1: Model comparison calculated as MAE when they are excited with large amplitude changes (MAE_{ls}) or low ones (MAE_{ss}).

Model comparison	MAE_{ls}	MAE_{ss}
Wiener-Hammerstein	0.2396	0.0342
Wiener-Wiener-Hammerstein	0.0895	0.0126
Hammerstein-Wiener-Hammerstein	0.1653	0.0235

modes of a process. The persistence of the Gaussian signal enables capturing all the dynamics; however, from a practical point of view, there are two important aspects that must be considered. Precision is not the only criterion to consider for the selection of a model; it is also necessary to consider its complexity. For example, for practical control applications, a model with an excessive number of poles and zeros is not always necessary, and in many cases, only the dominant dynamic is required. On the other hand, to excite all the oscillating modes of a process, the Gaussian signal must be of a long duration. For this reason, its use is impractical in real processes with relatively slow dynamics. Table 2 shows a comparison of the characteristics of a Gaussian signal and a multistep signal.

The issues of using Gaussian signals in processes with slow dynamics are further aggravated when the BLA is required, as its estimation may require multiple realisations. The proposed unified approach, besides enabling estimation of Wiener, Hammerstein, and Wiener–Hammerstein models without a priori information from the user, provides a practical alternative to estimate processes where the BLA estimation is not possible, either because they are slow dynamically, or they are not enabled to handle Gaussian signals.

Table 2: Comparison of characteristics between a multistep signal and a Gaussian signal.

Features	Multistep signal	Gaussian signal
Frequency content.	Low.	High.
Applicability on real processes.	Highly applicable.	It is not always possible.
Information on static nonlinearity.	High information content.	Lower information content, especially if the nonlinearities are at the extremes of the process operation range.
Information on the dynamics.	Less information content. Ideal to estimate the dominant dynamics of a process.	High content if the signal is well designed. Ideal to estimate all the dynamics of a process.
Complexity.	Easy design.	Not so simple to design. The bandwidth must be selected carefully.
Duration.	Do not need to be so extensive.	Must be extensive to excite all the oscillatory modes of a process.

3. WH-EA Abstract

WH-EA is an elitist evolutionary algorithm inspired by biological evolution over generations. It is based on a population of NP individuals who evolve through the generations and compete with each other for their survival. This section presents the main components of the algorithm; nonetheless, complete information can be found in [34].

3.1. Structure of Individuals and Genetic Operators. Each individual contains coded information ($\boldsymbol{\theta}_i^g = [\mathbf{P}_i^g, \mathbf{B}_i^g, \mathbf{C}_i^g]$) representing a possible solution for the optimisation problem. It is comprised of three genetic portions: location of poles and zeros ($\mathbf{P}_i^g$); location of the breakpoints ($\mathbf{B}_i^g$) to capture the static nonlinearity; and the classification of poles and zeros ($\mathbf{C}_i^g$). The algorithm contains customised mutation and crossover operations, performed on each piece of genetic information. Figure 7 shows the structure of an individual and the genetic operations that apply to each part. The subscript i identifies an individual in the population, while the superscript g indicates the current generation.

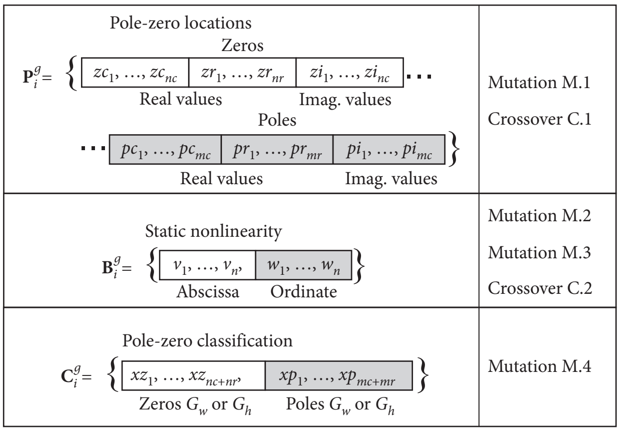


Figure 7: Structure of an individual and genetic operations performed on each piece of genetic information.

Location of poles and zeros for each individual is encoded in a single vector. Its elements $zr_1, \ldots, zr_{nr}$ and $pr_1, \ldots, pr_{mr}$ contain the location of real poles and zeros, respectively. Real and imaginary parts of complex zeros are coded in $zc_1, \ldots, zc_{nc}$ and $zi_1, \ldots, zi_{nc}$, while complex poles are coded in $pc_1, \ldots, pc_{mc}(x)$ and $pi_1, \ldots, pi_{mc}$, respectively. Values of mr, nr, mc, and nc depend on 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.

Figure 8 depicts the correspondence that exists between an encoded individual and the resulting nonlinear model. In the example, an initial linear model of four zeros and six poles has been considered, while to capture the static nonlinearity, four points have been assigned. From the initial linear model, $nc = 1$ and $nr = 2$ since there is a pair of complex zeros and two real zeros. In the same way, $mc = 1$ and $mr = 4$ since there is a pair of complex poles and four real poles. The binary code contained in $\mathbf{C}_i^g$ indicates how the poles and zeros of the initial linear model are distributed around the static nonlinearity.

3.1.1. Pole-Zero Locations. Since the initial linear model is not perfect, the location of the poles and zeros must be refined with new estimates around the known values. To explore new locations in the S-plane, two genetic operations are carried out. In both operations, an offspring is created from all the genetic information of his parent except in one gene. The modified gene is formed depending on the selected genetic operation. These operations work as follows:

(i) *Mutation M.1.* The modified gene is determined by a random number with Gaussian distribution ($N_{zp}(0, \sigma^2(g))$) that is generated within the corresponding search space. The search space is defined by the user specifically for each gene. The standard deviation to generate the random number is variable throughout the generations. This deviation decreases from σ_{ini}^2 to σ_{end}^2 as the algorithm evolves. This enables controlling the aggressiveness of the mutations, that is, in the final generations, the mutations will be more subtle to achieve a fine-tuning of the parameters. Initial and final values of the standard deviation are defined by the user and are expressed as a percentage ratio of the search space of each gene.

(ii) *Crossover C.1.* The modified gene is formed by crossing genetic information between the father and the best individual in the population. Crossing is determined by an average between the values of the corresponding genes.

The refit of locations of poles and zeros is carried out within a search space defined by the user. This search space must be bounded around each pole or each zero based on a minimum and maximum value. For example, a real pole at -0.5 with bounds of $\pm\, 0.1$ may move between -0.4 and -0.6. Bound selection depends on how close the pole or zero is in relation to the imaginary axis. The poles and zeros closest to the origin are more dominant than those that are further away; therefore, they are more sensitive to changes. Since only fine-tuning is performed at the location of the poles and zeros, smaller dimensions have been selected for the poles most attached to the origin, while the farthest poles have greater freedom of movement since they are less sensitive.

There is no recipe for precisely defining these bounds; however, it should be taken into account that large bounds will allow a better exploration but at the cost of the algorithm converging more slowly.

3.1.2. Static Nonlinearity. Information regarding static nonlinearity is also encoded in a single vector. This information contains the abscissas ($v_1, \ldots, v_n$) and ordinates ($w_1, \ldots, w_n$) of the breakpoints that define the piecewise linear function. The user-defined parameter n indicates the number of breaking points used to capture the static nonlinearity.

To capture nonlinearity with great precision, two mutations plus one crossover are executed on this portion of genetic information. In both mutations, an offspring is created from all the genetic information of his parent except in two genes. These two modified genes correspond to an abscissa and its corresponding ordinate that describe the breaking point position. On the other hand, the crossover

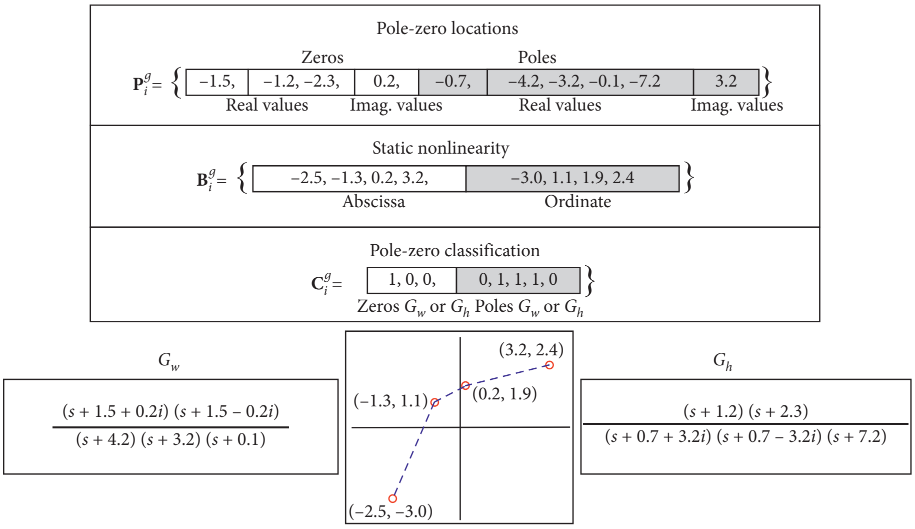

Pole-zero locations
Zeros
Poles
$\mathbf{P}_i^g = \{$ −1.5, | −1.2, −2.3, | 0.2, | −0.7, | −4.2, −3.2, −0.1, −7.2 | 3.2 $\}$
Real values
Imag. values
Real values
Imag. values
Static nonlinearity
$\mathbf{B}_i^g = \{$ −2.5, −1.3, 0.2, 3.2, | −3.0, 1.1, 1.9, 2.4 $\}$
Abscissa
Ordinate
Pole-zero classification
$\mathbf{C}_i^g = \{$ 1, 0, 0, | 0, 1, 1, 1, 0 $\}$
Zeros G_w or G_h Poles G_w or G_h
G_w
$\frac{(s + 1.5 + 0.2i)(s + 1.5 - 0.2i)}{(s + 4.2)(s + 3.2)(s + 0.1)}$
(3.2, 2.4)
(−1.3, 1.1)
(0.2, 1.9)
(−2.5, −3.0)
G_h
$\frac{(s + 1.2)(s + 2.3)}{(s + 0.7 + 3.2i)(s + 0.7 - 3.2i)(s + 7.2)}$

FIGURE 8: Example of an encoded individual and the resulting nonlinear model.

operation generates an offspring taking all the genetic information of his parent except in one gene. The modified gene corresponds to the ordinate of a breaking point. The three genetic operations applied to this portion of genetic information work as follows:

(i) *Mutation M.2.* The two genes that correspond to the coordinates of a breaking point change randomly. For this, two random numbers $((N_v(0, \sigma^2(g))$ and $N_w(0, \sigma^2(g)))$ with Gaussian distribution are generated within the corresponding search space. To avoid overlapping points, the search space for the genes corresponding to the abscissa is determined by the position of the neighbouring points. The search space for ordinates is the same for all points and corresponds to the codomain of the nonlinear function that is determined as indicated in Section 2.4. To control the aggressiveness of the mutations, standard deviations for generating random numbers are variable throughout the generations as in Mutation M.1. The initial and final values of both standard deviations are user-defined parameters that are also expressed as a percentage ratio of the search space of each gene.

(ii) *Mutation M.3.* This mutation allows concentrating as many points as possible in places where there are curvatures. The gene corresponding to the abscissa is modified in such a way that the points can jump between them. The new value of the abscissa is calculated as the midpoint between the two breakpoints that form the segment over which the point will jump. This segment is determined randomly. On the other hand, the gene corresponding to the ordinates is modified using a quadratic interpolation and information of the points near the location where the jump occurred. Quadratic interpolation will cause a smooth transition of a point when it jumps from one segment to another.

(iii) *Crossover C.2.* The modified gene corresponding to the ordinate of a point is formed by crossing genetic information between the father and the best individual in the population. Crossing is determined by an average between the values of the corresponding genes.

Parameter α indicates how close the break points can be located. This parameter is considered in the aforementioned genetic operations to prevent break points from overlapping. This is an important situation to consider since interpolation methods require that there are no break points with equal values on the abscissa axis; however, α must be small enough to allow break points to join in the curvatures.

The ratio between the horizontal size of the search space and the number of break points $((V_{\max} - V_{\min})/n)$ gives an idea of how small α should be. If α would take the value of this ratio, a uniform horizontal distribution of the break points along the search space would be achieved; however, what is required is that the points be grouped into the curvatures. In this sense, α must take a lower value than this ratio. In any case, it should be taken into account that it is better to select a small α value because if the break points are not required to be so close together, the corresponding genetic operations will be responsible for separating them, while if a large α value is selected and the break points require grouping, the algorithm will not be able to do so and precision will be lost in capturing static nonlinearity.

3.1.3. Pole-Zero Classification. This portion of genetic information contains binary coded information that enables the distribution of the dynamics. The binary codification will indicate whether the identified system is a Wiener, Hammerstein, or Wiener–Hammerstein structure. In this last case, the code will also indicate how the poles and zeros have been distributed between the two LTI subsystems. The first part of the binary vector $(xz_1, \ldots, xz_{nc+nr})$ is used for zero classifications, while the second part $(xp_1, \ldots, xp_{mc+mr})$ is used for pole classifications.

Pole and zero classifications are based on a correspondence between $\mathbf{P}_i^g$ and $\mathbf{C}_i^g$. When an element of $\mathbf{C}_i^g$ is equal to 1, the corresponding pole or zero of $\mathbf{P}_i^g$ will be located in the LTI subsystem prior to nonlinearity, whereas if it takes the value of 0, it will be located in the LTI subsystem that follows the nonlinearity. The genetic operation applied to this information portion works as follows:

(i) *Mutation M.4.* To test different dynamic structures as the algorithm evolves, this genetic operation randomly modifies all the genes, i.e., each time the mutation M.4 is required, a new binary code is generated. This code is generated considering that the resulting linear subsystems cannot be improper; in addition, in case there is a dynamic distribution, the sum of the zeros and the sum of the poles between the two subsystems must be equal to the number of zeros and poles of the initial linear model, respectively.

3.2. Algorithm Description. WH-EA is based on three stages: initialization of the population; offspring generation; and population update. From an initial population of *NP* individuals, the algorithm evolves based on the mutation and crossover operators described in Section 3.1. In each generation, there will be an individual with the best fitness ($\boldsymbol{\theta}_{\text{best}}^g$). When the algorithm obtains the last generation (MaxGen), the individual with the best fitness will be the solution to the optimisation problem. Algorithm 1 shows a pseudocode of WH-EA, while a detail of the three algorithm stages is indicated below.

3.2.1. Initialisation of the Population. In this stage, NP individuals with the genetic structure of Figure 7 are generated. The first individual of the population is built as follows:

(i) Poles and zeros of the initial linear model give rise to the genetic information portion corresponding to pole-zero locations ($\mathbf{P}_1^0$).

(ii) The n points used to capture the nonlinear function are distributed uniformly along the diagonal formed within the search space of the static nonlinearity (see Section 2.4 for more details). Distribution of these points gives rise to the genetic information portion of the static nonlinearity ($\mathbf{B}_1^0$).

(iii) A random binary code is generated to fill the genetic information portion corresponding to the classification of poles and zeros ($\mathbf{C}_1^0$).

Once the first individual has been structured, mutations M.1, M.2, and M.4 are applied successively on this individual to give rise to the new individuals that will occupy a place in the initial population.

3.2.2. Offspring Generation. In each generation g, an individual ($r_1 \in [1, \text{NP}]$) of the population is selected randomly. The selected individual is known as a parent ($\boldsymbol{\theta}_{r_1}^g = [\mathbf{P}_{r_1}^g, \mathbf{B}_{r_1}^g, \mathbf{C}_{r_1}^g]$), since it will give rise to an offspring who will inherit a large part of its genetic information. An offspring will differ to a lesser or greater extent from its parent depending on the genetic information portion selected to be modified and the genetic operation applied. The genetic information portions modified in the offspring are denoted by $\widetilde{B}^g$ (breakpoint positions), $\widetilde{\mathbf{P}}^g$ (pole-zero locations), and $\widetilde{\mathbf{C}}^g$ (pole-zero classification), respectively.

Selection between the genetic information portions corresponding to the locations of the breakpoints and locations of poles and zeros is handled randomly with $r_{pznl} \in (0, 1]$. In addition, probability for the selection is not fixed throughout the generations, and this is controlled with the parameter $\gamma(g)$. This parameter has an exponential behaviour that decreases as the generations go by. During the first generations, the probability of modifying the genetic information corresponding to the breakpoint locations is very high, and this is justified by the fact that the nonlinearity is not known, while the dynamics in a certain way are known and only require refinement. Genetic information corresponding to the locations of the breakpoints is modified through Algorithm 2, while the locations of the poles and zeros are modified with Algorithm 3.

After one of the two portions of genetic information has been modified, a random process handled by $r_c \in (0, 1]$ will determine whether the genetic information portion corresponding to the classification of poles and zeros will be modified by using Mutation M.4. The probability for this modification is controlled by the user through the parameter $\xi \in (0, 1]$.

To modify a genetic information portion (either $\mathbf{B}_{r_1}^g$ or $\mathbf{P}_{r_1}^g$), not all genetic operations are applied at the same time. In Algorithm 2, the control parameter $\delta_{nl} \in (0, 1]$ indicates the probability with which the mutation (either M.2 or M.3) or crossover C.2 will be used, while the random number $r_{nmc} \in (0, 1]$ enables the selection. In the same way, in Algorithm 3, the control parameter $\delta_{zp} \in (0, 1]$ indicates the probability with which each genetic operation will be used (either mutation M.1 or crossover C.1), while the random number $r_{lmc} \in (0, 1]$ enables the selection.

In Algorithm 2, the selection probability between mutations M.2 and M.3 is variable throughout the generations. This is justified by the fact that mutation M.2 enables an exploration in the search space to shape the static nonlinearity. This mutation is very useful in the first generations. While mutation M.3 enables an accumulation of points in the curvatures, M.3 is not useful during the first generations as nonlinearity curvatures have not yet taken shape. The variable probability of selection between the two mutations is controlled by the parameter $\eta(g)$. This parameter decreases linearly as the generations go by. Since mutation M.2 may also be useful in the final generations, the parameter $\eta_{\min} \in (0, 0.5]$ is included to control the selection probability of the two mutations. This is a user-defined parameter indicating the minimum probability with which the mutation M.2 can be selected.

3.2.3. Updation of the Population. Once an offspring has been generated, it must compete with the individuals of the

```
Initialise the population;
Evaluate fitness of all population;
for $g = 1$ to MaxGen do
    Find $\boldsymbol{\theta}^{g}_{\text{best}}$
    Random selection of an individual $(r_1)$;
    Compute $\gamma(g)$;
    if $r_{pznl} \le \gamma(g)$then
        Compute $\widetilde{\mathbf{B}}^{g}$ using Algorithm 2;
    else
        Compute $\widetilde{\mathbf{P}}^{g}$ using Algorithm 3;
    end if
    if $r_c \le \xi$then
        Compute $\widetilde{\mathbf{C}}^{g}$ using Mutation M.4;
    end if
    Update population;
end for
Print $\boldsymbol{\theta}^{\text{MaxGen}}_{\text{best}}$
```

Algorithm 1: Pseudocode of WH-EA.

```
if $r_{nmc} \le \delta_{nl}$ then
    Compute $\eta(g)$
    if $r_{mm} \le \eta_{\min} + \eta(g)$then
        Mutation M.2;
    else
        Mutation M.3;
    end if
else
    Crossover C.2;
end if
```

Algorithm 2: Modifying two-dimensional points for nonlinear function.

```
if $r_{lmc} \le \delta_{zp}$ then
    Mutation M.1;
else
    Crossover C.1;
end if
```

Algorithm 3: Modifying pole/zero locations.

population. The fittest contestant wins the competition. The first opponent will be chosen from the population on a random basis. If the offspring wins the competition, it will occupy the position of the defeated individual in the population and the algorithm continues with the next generation. If the offspring fails to defeat the selected individual, a new competition will be held with the next individual of the population. This process will be repeated until the offspring defeats an individual. If the offspring has competed with all the individuals of the population and has not been able to defeat any, this is discarded and the algorithm continues with the next generation.

4. Application Examples

The presented approach was validated with three numerical examples and a real thermal process. In each case, the initial linear model was identified with the Matlab CONTSID Toolbox using the command *srivc*. Nonlinear identification was executed in continuous time using WH-EA. For this, simulations of the dynamic models in the objective function were performed with the *lsim* command of Matlab, while points of nonlinearity were interpolated to define a piecewise linear function. WH-EA parameters were set the same for the four identification problems: $\xi = 0.25$; $\delta_{zp} = 0.75$; $\delta_{nl} = 0.75$; and $\eta_{\min} = 0.35$. In addition, initial and final standard deviations for mutations were set to 20 and 1, respectively.

4.1. Numerical Example 1.

4.1. Numerical Example 1. For the first numerical example, an LTI subsystem of four poles and one zero is connected in series to a static nonlinearity to give rise to a Wiener structure (see Figure 9). Static nonlinearity consists of a sigmoid hyperbolic tangent function “tansig” (19), which symmetrically saturates large values of the independent variable.

$$S_{\text{NL}} = 50\,(1 + \text{tansig}\,(0.05\,(x - 50))). \tag{19}$$

The LTI subsystem used for this example has unitary static gain. Although the methodology proposed in this paper enables the identification of block-oriented models with a static nonlinearity and LTI subsystems with any gain, unitary gains have been assumed simply for convenience. This will allow the captured static nonlinearity to be compared with the real nonlinear function to evaluate the precision that can be achieved with WH-EA.

To estimate the initial linear model, the simulated system took half their operation range (50%). From this point, the input was modified twice consecutively to give rise to two steps. Each step had a temporary duration of 20 s. The first step had a positive amplitude of 2.5%, while the next had the same amplitude but negative, forming a rectangular pulse. To emulate a real situation, Gaussian noise with a power of −30 dB was added on the system output. Input and output data were sampled with a period of 10 ms. Figure 10 depicts the excitation signal and the response of the simulated system.

The data obtained with the first input change $t\,(15,\ldots,35)$ were used to estimate the initial linear model, while the data belonging to the second input change $t\,(36,\ldots,56)$ were used for validation purposes. To avoid problems with initial conditions, offset was removed from data. Fourteen linear models were estimated from second to fifth order considering only strictly proper systems (number of zeros smaller than the number of poles). For each estimated model, the quadratic mean error (MSE) was calculated on the estimation (MSE_e) and validation (MSE_v) datasets. In addition, to select the best structure, the MDL criterion was calculated using (9).

The results of the estimates are shown in Table 3. According to lowest value of MDL criterion, the best

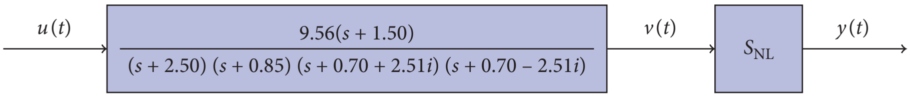

$u(t)$
$\frac{9.56(s+1.50)}{(s+2.50)(s+0.85)(s+0.70+2.51i)(s+0.70-2.51i)}$
$v(t)$
S_{NL}
$y(t)$

Figure 9: Numerical example 1: Wiener structure.

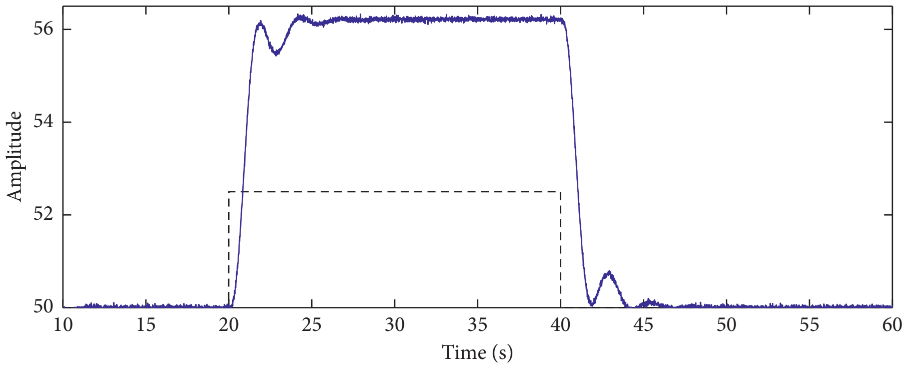


Figure 10: Input (black dashed line) and output (blue solid line) data for initial linear model estimation (numerical example 1).

estimated model was of four poles and one zero (20), which corresponds to the real structure of the linear dynamics that was used for the numerical example. This selection is corroborated by MSE$_e$ values (from the model of four poles and a zero in cases where the MSE decreases, this decrease is practically negligible).

$$G_{w_{lne}}(s)=\frac{23.96(s+1.52)}{(s+2.51)(s+0.85)(s+0.70+2.51i)(s+0.70-2.51i)}. \tag{20}$$

For nonlinear identification, two multistep inputs were generated, one for estimation and another for validation purposes. Both signals were designed with 50 steps of random amplitude to explore the entire operating range of the input process (0–100%). Each step had a time duration of 25 s, and the minimum difference between two consecutive steps was restricted so that it is not less than 18 units. The simulated system was excited with both signals separately. Input and output data for both cases were sampled with a period of 10 ms. From the estimation dataset, the minimum and maximum values of the input and output signals were obtained to define the search space for the static nonlinearity. These values were $u_{\min}=0$, $u_{\max}=100$, $y_{\min}=0.559$, and $y_{\max}=99.44$. Taking into account that the static gain of the estimated initial linear model is positive, the search space for static nonlinearity was defined with $(u_{\min}, y_{\min})$ and $(u_{\max}, y_{\max})$.

Once the search space for static nonlinearity was defined, WH-EA was configured according to the data of Table 4. In addition, $\mathbf{P}_1^0$ was coded with $nc=0$, $nr=1$, $mc=1$, $mr=2$, and pole/zero locations of (20). Furthermore, all bounds to search new pole-zero locations were set in ± 0.1.

At the end of the generations, a Wiener model was obtained, that is, WH-EA distributed the dynamics correctly without the need for the user to specify the type of structure to be identified. The value reached for the objective function (MAE$_e$) was $4.415E-2$, while the absolute error on the validation dataset (MAE$_v$) was 5.567×10^{-2}. Figure 11 depicts a comparison on the validation dataset between the Wiener system output and the estimated model one. A magnification on a portion of data is also shown to demonstrate the great precision of the estimated model. Figure 12 shows how the 18 points representing the static nonlinearity were located within the search space. To verify that it has been captured with great precision, the nonlinear function tansig was included in the graph.

Table 3: Ranking of estimated linear models for numerical example 1 (Wiener system). Models $4p$, $2z/5p$, $3z/5p$, and $4z/5p$ have been excluded due to their high MCL values.

Structure	MSE$_e$	MSE$_v$	MDL
$2p$	$4.61E-2$	$4.66E-2$	$2.34E-2$
$1z/2p$	$3.13E-2$	$3.19E-2$	$1.59E-2$
$3p$	$2.38E-3$	$2.49E-3$	$1.21E-3$
$1z/3p$	$1.50E-3$	$1.51E-3$	$7.71E-4$
$2z/3p$	$1.17E-3$	$1.18E-3$	$6.05E-4$
1z/4p	**1.01E − 3**	**9.81E − 4**	**5.20E − 4**
$2z/4p$	$1.01E-3$	$9.82E-4$	$5.22E-4$
$3z/4p$	$1.01E-3$	$9.92E-4$	$5.24E-4$
$5p$	$4.46E-2$	$4.42E-2$	$2.29E-2$
$1z/5p$	$1.01E-3$	$9.83E-4$	$5.22E-4$

Table 4: Summary of WH-EA parameter settings for numerical examples.

Parameter	Description	Value
MaxGen	Generations number	$5e6$
NP	Population size	$5e3$
n	Number of points to represent nonlinearity	18
α	Minimum distance between two points	0.9

4.2. Numerical Example 2. For this numerical example, the same linear subsystem and the same static nonlinearity of the

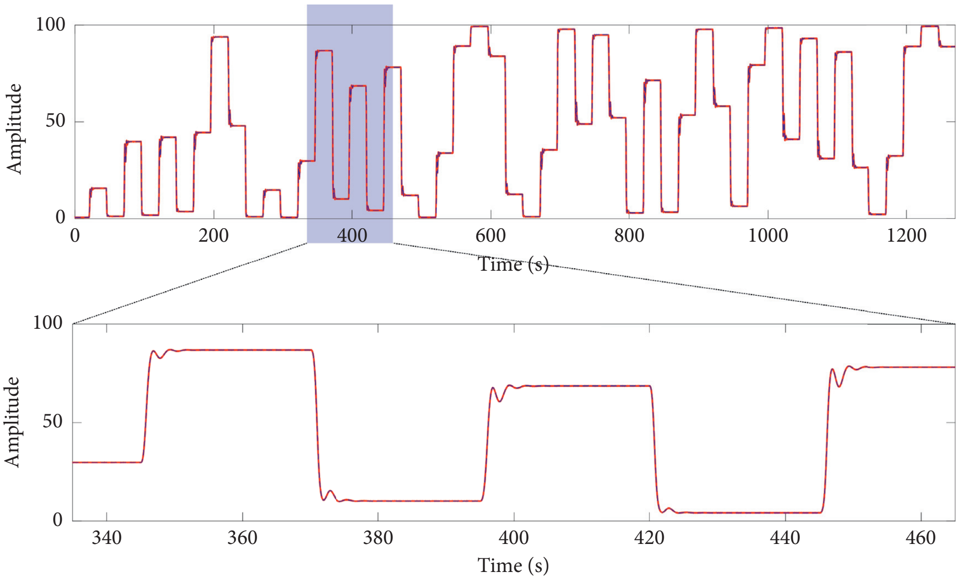


FIGURE 11: Validation results for numerical example 1: real output (red solid line) and model output (blue dashed line).

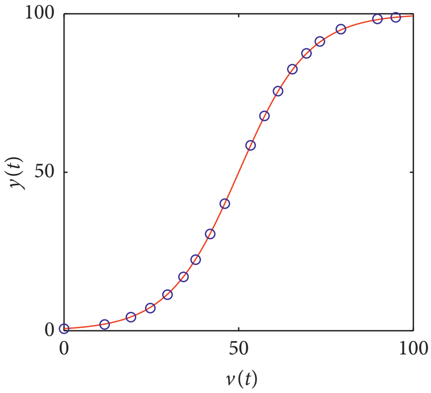


FIGURE 12: Numerical example 1: comparison between real nonlinear function tansig (red) and captured nonlinearity (blue circles).

previous numerical example were used; however, the blocks were permuted to give rise to a Hammerstein model (see Figure 13). Similarly, the same input signals that were used in the previous example were used to excite this simulated model. With the corresponding dataset for linear estimation, fourteen different linear models were tried. As in the previous case, only strictly proper models from second to fifth order were considered. The results of the estimates are shown in Table 5. The best linear model (21) according to the MDL criterion was four poles and one zero corresponding to the order of the real system.

$$G_{h_{lne}}(s) = \frac{23.87\,(s+1.53)}{(s+2.51)\,(s+0.85)\,(s+0.70+2.51i)\,(s+0.70-2.51i)}. \tag{21}$$

From the nonlinear estimation dataset, the search space for static nonlinearity was defined in the same way as it was for the previous numerical example. For this case, the minimum and maximum values of the input and output signals were $u_{\min} = 0$, $u_{\max} = 100$, $y_{\min} = 0.236$, and $y_{\max} = 99.45$.

For nonlinear estimation, WH-EA was executed considering the configuration parameters of Table 4. In addition, $\mathbf{P}_1^0$ was coded with $nc = 0$, $nr = 1$, $mc = 1$, and $mr = 2$, and pole/zero locations of (21) were used. Furthermore, all bounds to search new pole-zero locations were set in ± 0.1. At the end of the generations, WH-EA distributed the dynamics correctly, that is, a Hammerstein model was obtained. The value reached for the objective function (MAE_e) was 4.328×10^{-2}, while the absolute error on the validation dataset (MAE_v) was 6.526×10^{-2}. Figure 14 depicts a comparison on the validation dataset between the simulated output generated by the numerical example and the output of the estimated model. On the other hand, Figure 15 shows how the 18 points were distributed within the search space to capture the static nonlinearity. As in the previous case, the real nonlinear function was introduced in this graph to visualise the precision achieved with WH-EA.

4.3. Numerical Example 3. For this numerical example, a Wiener–Hammerstein model was constructed using the same dynamics and the same static nonlinearity of the previous examples. In this case, the front LTI subsystem was formed with the two complex poles and a gain of 6.80, while the back LTI subsystem was formed with the two real poles, the zero, and a gain of 1.41 (see Figure 16). The excitation signals and the procedures for linear and nonlinear estimation were the same as those used in the previous examples. Ranking of linear estimates is shown in Table 6. As in the previous cases, the best linear model according to the MDL criterion was of four poles and one zero (22), which is consistent with the dynamics of the real system even though for this case the dynamics was distributed around the static nonlinearity. This demonstrates the great effectiveness of the linear estimation method used in this approach.

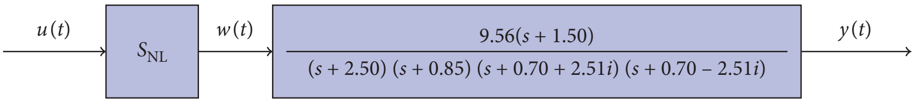

$u(t)$
S_{NL}
$w(t)$
$\frac{9.56(s+1.50)}{(s+2.50)(s+0.85)(s+0.70+2.51i)(s+0.70-2.51i)}$
$y(t)$

Figure 13: Numerical example 2: Hammerstein structure.

Table 5: Ranking of estimated linear models for numerical example 2 (Hammerstein system). Models $4p$, $3z/5p$, and $4z/5p$ have been excluded since their MDL values were very high.

Structure	MSE_e	MSE_v	MDL
$2p$	$4.57E-2$	$4.565E-2$	$2.32E-2$
$1z/2p$	$3.29E-2$	$3.29E-2$	$1.68E-2$
$3p$	$2.29E-3$	$2.48E-3$	$1.17E-3$
$1z/3p$	$1.38E-3$	$1.45E-3$	$7.08E-4$
$2z/3p$	$1.13E-3$	$1.14E-3$	$5.82E-4$
1z/4p	**9.45E − 4**	**9.59E − 4**	**4.85E − 4**
$2z/4p$	$9.43E-4$	$9.59E-4$	$4.86E-4$
$3z/4p$	$9.42E-4$	$9.60E-4$	$4.87E-4$
$5p$	$4.26E-2$	$4.28E-2$	$2.19E-2$
$2z/5p$	$9.46E-4$	$9.59E-4$	$4.89E-4$

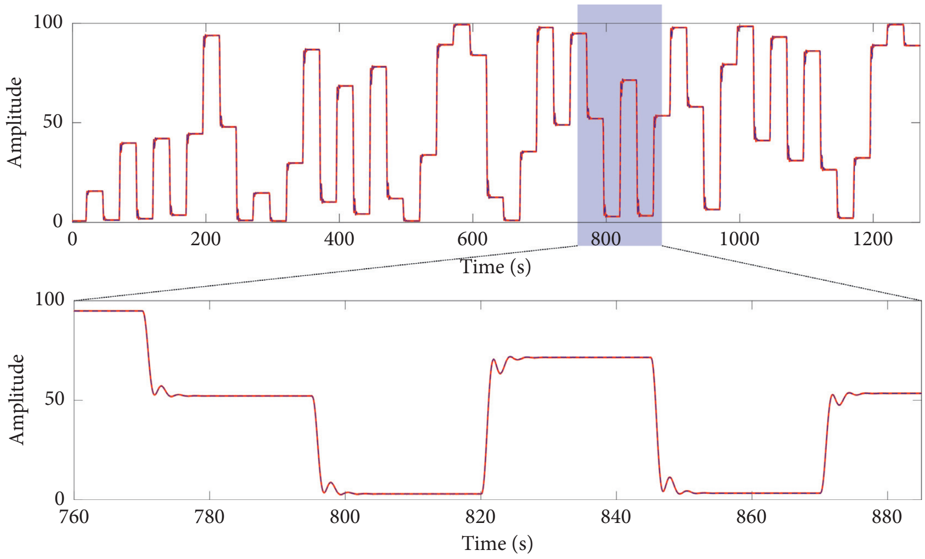


Figure 14: Validation results for numerical example 2: real output (red solid line) and model output (blue dashed line).

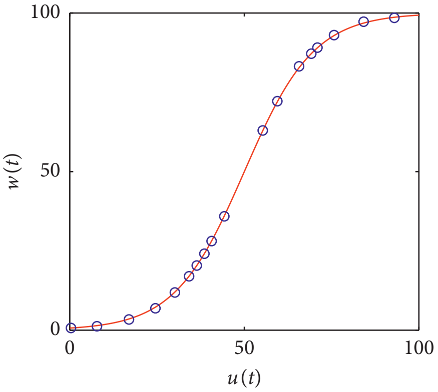


Figure 15: Comparison between real nonlinear function tansig (red) and captured nonlinearity (blue circles). Results obtained on numerical example 2.

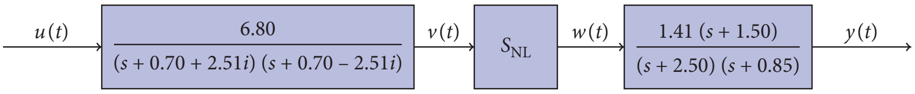


FIGURE 16: Numerical example 3: Wiener–Hammerstein structure.

TABLE 6: Ranking of estimated linear models for numerical example 3 (Wiener–Hammerstein system). Models $4p$, $1z/5p$, and $2z/5p$ have been excluded due to high MDL values.

Structure	MSE_e	MSE_v	MDL
$2p$	$4.53E-2$	$4.61E-2$	$2.30E-2$
$1z/2p$	$3.09E-2$	$3.11E-2$	$1.57E-2$
$3p$	$2.50E-3$	$2.32E-3$	$1.27E-3$
$1z/3p$	$1.50E-3$	$1.46E-3$	$7.68E-4$
$2z/3p$	$1.18E-3$	$1.11E-3$	$6.07E-4$
1z/4p	**1.02E − 3**	**9.67E − 4**	**5.23E − 4**
$2z/4p$	$1.02E-3$	$9.69E-4$	$5.25E-4$
$3z/4p$	$1.02E-3$	$9.70E-4$	$5.26E-4$
$5p$	$4.96E-2$	$4.96E-2$	$2.55E-2$
$3z/5p$	$1.01E-3$	$9.72E-4$	$5.28E-4$
$4z/5p$	$1.01E-3$	$9.73E-4$	$5.30E-4$

$$G_{wh_{lne}}(s) = \frac{24.195\,(s+1.549)}{(s+2.598)\,(s+0.854)\,(s+0.709+2.507i)\,(s+0.709-2.507i)}. \tag{22}$$

As can be seen, the linear models obtained in (20)–(22) differ very little from each other and are almost equal to the real dynamic model. This is because the step signal used for the three identification experiments has a small amplitude which hides the effect of nonlinearity. This corroborates what was indicated in Section 2.5. A step signal with a small amplitude change is useful for linear estimation; however, for nonlinear estimation, it is necessary that the nonlinearity is notorious, for this the amplitude changes of the step signal must be large.

As in the previous cases, WH-EA was configured with the parameters of Table 4. In addition, the first individual of the population $(\mathbf{P}_1^0)$ was coded with $nc = 0$, $nr = 1$, $mc = 1$, $mr = 2$, and pole/zero locations of (22). According to the minimum and maximum values of the input and output signals, the search space of the static nonlinearity was defined with the coordinates: $(0, 0.5546)$ and $(100, 99.443)$.

At the end of the generations, a Wiener–Hammerstein model was obtained and the dynamics of both LTI subsystems were consistent with the real system. The value reached for the objective function (MAE_e) was 3.768×10^{-2}, while the absolute error on the validation dataset (MAE_v) was 5.117×10^{-2}. Figure 17 depicts a comparison on the validation dataset between the simulated output generated by the numerical example and the output of the estimated model. On the other hand, Figure 18 shows a comparison between the real and estimated nonlinearity.

4.4. Thermal Process Identification. The real process used to validate the paper proposal consists of a lab scale thermal process based on a Peltier cell. Principle of operation of this device is based on nonlinear Peltier and Seebeck effects. Figure 19 shows the architecture of the system that was assembled to operate the process and acquire its output variables. As can be seen, a fan radiator has been coupled to the hot face of the Peltier cell. To measure the temperature of the cold face (T_{cold}), a type k thermocouple was used, while the temperature of the hot face (T_{hot}) was measured with an LM35 sensor. A power supply regulated with an external voltage signal u_a $(0, \ldots, 4.5V_{\mathrm{dc}})$ was used as an actuator to apply voltage to the Peltier cell. For all the experiments involved, the input/output process signals were sampled at 100 ms using a general purpose acquisition card with 12 bits A/D and D/A converters. The process was identified based on the input signal u_a and the temperature gradient between the cold and hot faces $(\Delta_T = T_{\mathrm{cold}} - T_{\mathrm{hot}})$.

A two-step signal was designed to identify and validate the initial linear model. This signal was injected after the process was taken to the middle of its operating range (2.25 V). The first step had a positive amplitude of 0.225 V (5% of the maximum voltage), while the next step had the same amplitude but negative. To ensure that the process reaches steady state, each step had a temporary duration of 700 s. The applied input signal and the response of the system are shown in Figure 20.

The data obtained with the first step change $t\,(0, \ldots, 750)$ were used to estimate the initial linear model, while the data belonging to the second input change $t\,(751, \ldots, 1451)$ were used for validation. To avoid problems with initial conditions, offset was removed from datasets. Different linear models were estimated from second to fifth order. Results of the estimates are shown in Table 7. For each estimated model, the MDL criterion, error on estimation, and error on validation datasets were computed. Models $2z/3p$, $3z/4p$, and $4z/5p$ were not considered, since they were of nonminimum phase, which is not consistent with the reality of the process. The rest of the discarded models had pole/zero cancellations or there were zeros far removed from the imaginary axis.

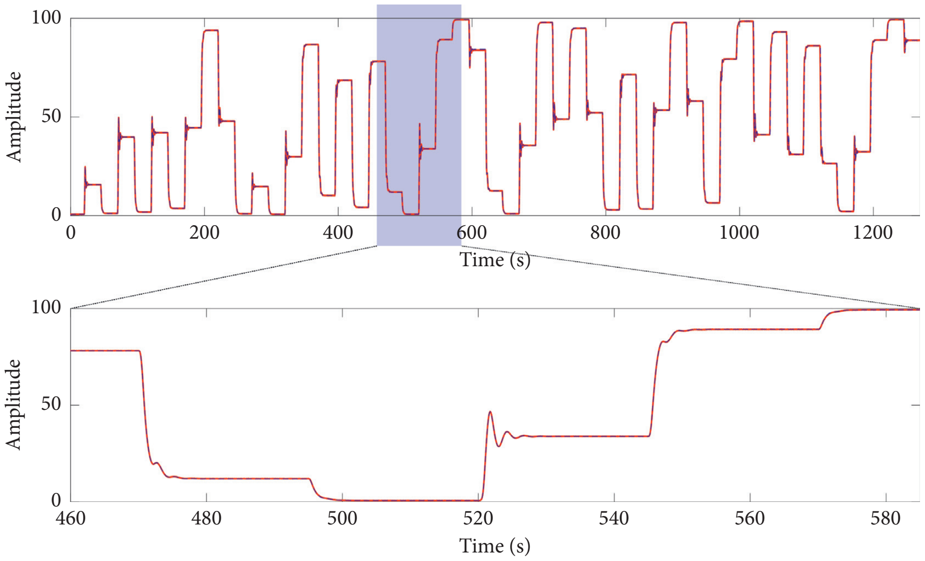


Figure 17: Validation results for numerical example 3: real output (red) and model output (blue dashed line).

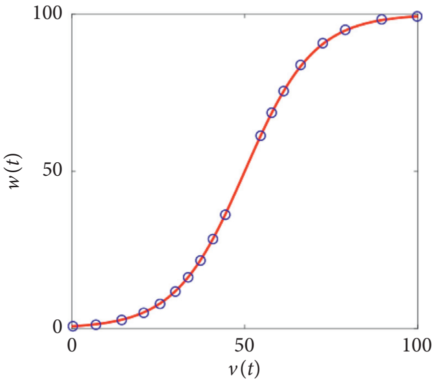


Figure 18: Example 3: comparison between real nonlinear function tansig (red) and captured nonlinearity (blue circles).

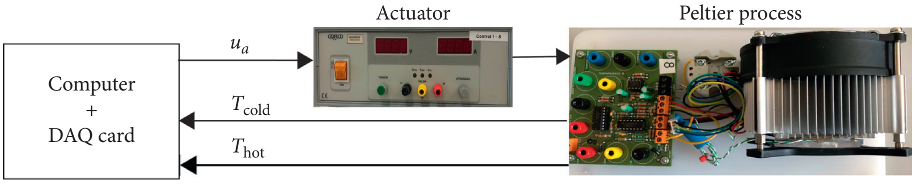


Figure 19: The real process based on a Peltier cell.

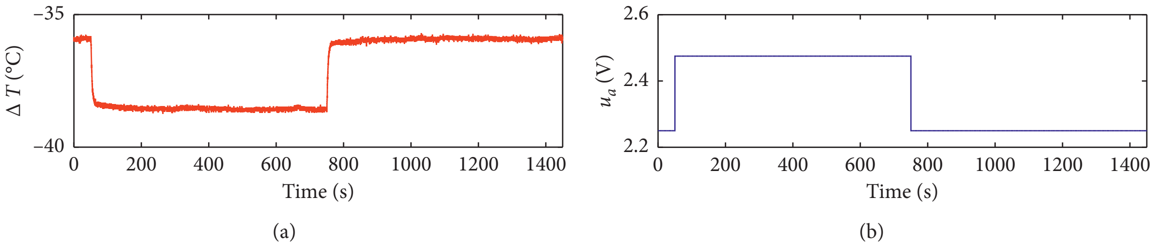


Figure 20: Input and output data for estimation of initial linear model (thermal process). (a) Excitation signal (blue). (b) Output signal of the thermal process (red).

Table 7: Ranking of estimated linear models for the thermal process. Models $4p$ and $5p$ have been excluded due to their MDL high values.

Structure	$\mathrm{MSE}_e\,(C)$	$\mathrm{MSE}_v\,(C)$	MDL
$2p$	$3.994E-3$	$4.475E-3$	13.981
$1z/2p$	$3.991E-3$	$4.473E-3$	13.972
$3p$	$3.982E-3$	$4.470E-3$	13.940
$1z/3p$	$2.577E-3$	$2.853E-3$	9.020
1z/4p	**2.575E − 3**	**2.852E − 4**	**9.014**
$1z/5p$	$2.576E-3$	$2.854E-4$	9.018

The best structure according to the MDL criterion was four poles and one zero (23). Figure 21 shows a comparison between the estimated model output and the real process output. This comparison has been made considering the validation dataset.

$$G_{lm}(s) = \frac{-41.186\,(s+0.015)}{(s+4.473)\,(s+2.519)\,(s+0.3406)\,(s+0.014)}. \tag{23}$$

For the nonlinear identification, two multistep signals were generated, one for identification and another for validation purposes (see Figure 22). The estimation signal was designed with 38 steps, while the validation one was designed with 24 steps. The temporary duration of the steps in both signals was 700 s, and the amplitude changes were handled randomly within the entire range of the actuator $u_a\,(0,\dots,4.5v)$, whereas the minimum difference between two consecutive steps was constrained to be greater than 1.5 V. Both signals were injected separately to the process, and the input and output data were recorded after the transient corresponding to the first step was extinguished.

WH-EA was configured with the parameters of Table 8, and according to the linear model structure, vector $\mathbf{P}_1^0$ was coded with $nc = 0$, $nr = 1$, $mc = 0$, and $mr = 4$, as follows:

$$\mathbf{P}_1^0 = [-0.015, -4.473, -2.519, -0.3406, -0.014]. \tag{24}$$

The bounds to explore new locations of poles and zeros were set to ±0.03 for poles/zeros close to the imaginary axis, while all other bounds were set to ±0.1. As in the numerical examples, the minimum and maximum values of the input and output signal were extracted from the estimation dataset: $u_{\min} = 0$, $u_{\max} = 4.5000$, $y_{\min} = -53.957$, and $y_{\max} = -0.1810$. Since the static gain of the estimated initial linear model is negative, the search space for static nonlinearity was defined with $(0, -0.1810)$ and $(4.5000, -53.957)$.

With this information, WH-EA was parameterized and executed. At the end of generations, the algorithm divided the dynamics into two linear subsystems, therefore the best structure to represent the thermal process was a Wiener-Hammrestein model.. Table 9 presents the coordinates of the nine points that were assigned to the static nonlinearity, while a plot of this nonlinearity is presented in Figure 23. Equations (25) and (26) show the two resulting subsystems, while performance of the WH identified model on estimation and validation datasets is shown in Figure 24. To quantify the accuracy of the estimated model, the MAE was calculated on the estimation and validation datasets with values of 0.1435 and 0.2184, respectively. To calculate both errors, the first 3000 samples of the datasets were not considered to avoid the transient effects.

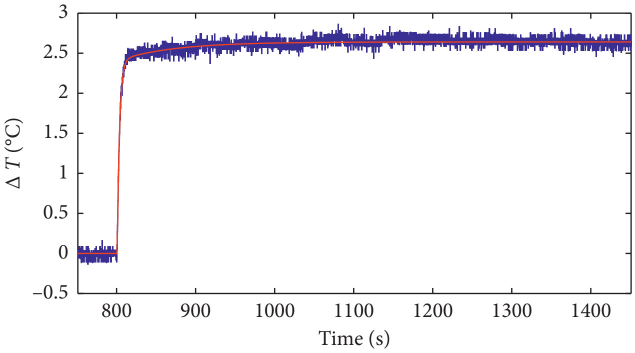


Figure 21: Comparison between the estimated model output (red) and the real process output (blue).

$$\widehat{G}_w(s) = \frac{11.393}{(s+4.394)\,(s+2.593)}, \tag{25}$$

$$\widehat{G}_h(s) = \frac{0.293\,(s+0.024)}{(s+0.3263)\,(s+0.022)}. \tag{26}$$

4.5. Discussion. The results obtained from the numerical examples show the effectiveness of the method to distribute the poles and zeros of the initial linear model around the static nonlinearity. For all three cases, a nonlinear model of 41 parameters has been estimated: 5 parameters for the linear dynamic model and 36 parameters for static nonlinearity. This number of parameters is due to the complexity of nonlinear function tansig, which was introduced intentionally to demonstrate the potential of the WH-EA genetic operators when capturing the nonlinearity. A comparison of the errors obtained from the estimation and validation datasets shows that these are very similar for each case. This shows that estimated models have a good predictive capacity, which can also be verified in Figures 11, 14, and 17, where the output of the estimated model has been compared with data not used in the identification procedure. However, it must be taken into account that the accuracy of the estimated model, as in all identification methods, depends on the amount of input and output data that feeds the procedure. In the specific case of the models addressed in this paper, it also depends on the number of points assigned to capture the static nonlinearity and the quality of the initial linear model.

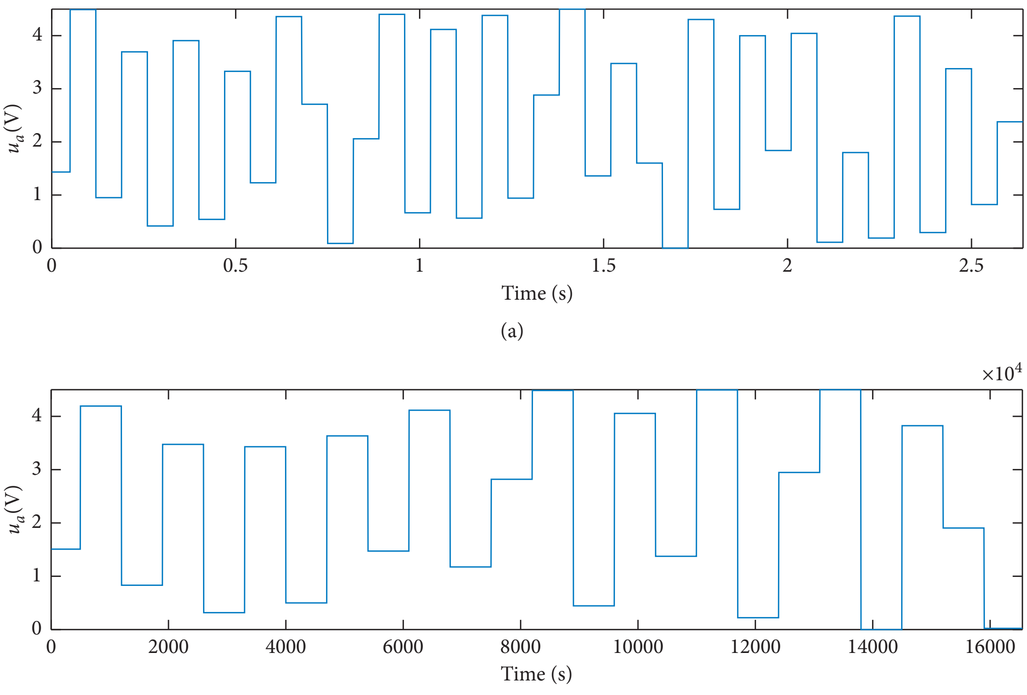


Figure 22: Input signals for nonlinear identification of the Peltier process. (a) Signal for estimation. (b) Signal for validation.

Table 8: Summary of WH-EA parameter settings for nonlinear identification of the thermal process.

Parameter	Description	Value
MaxGen	Generations number	$2e6$
NP	Population size	$5e3$
n	Number of points to represent nonlinearity	9
α	Minimum distance between two points	0.075

Table 9: Nonlinearity coordinates $(n = 9)$ estimated by WH-EA from thermal process.

i	1	2	3	4	5	6	7	8	9
v_i	0.0027	0.0887	0.7726	1.5100	2.2692	3.2030	3.8501	4.2461	4.3485
w_i	−0.5143	−1.0959	−14.5081	−26.8456	−36.9569	−46.0773	−50.3862	−52.5096	−52.8295

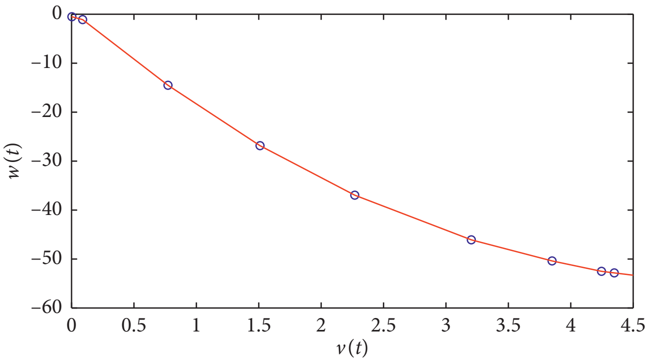
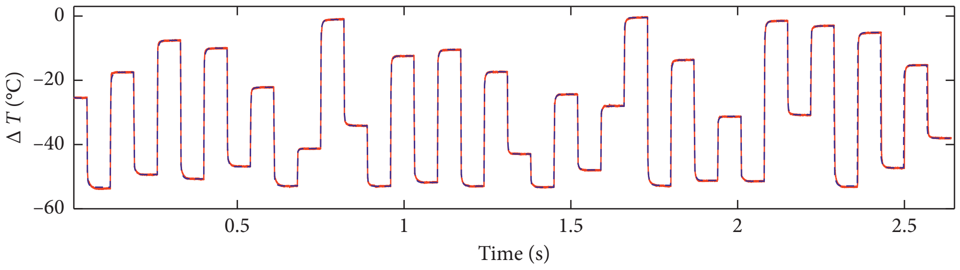


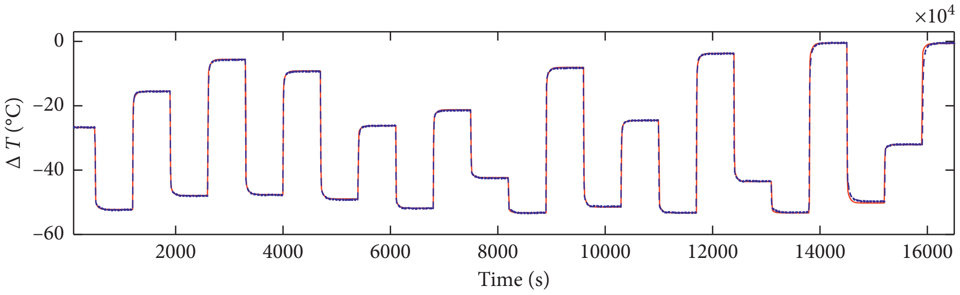
Figure 23: Captured nonlinearity as a piecewise linear function (red) from the estimated breaking points (blue circles).

The real process was estimated with 23 parameters: 5 for the linear dynamic part and 18 for static nonlinearity. The results obtained are very coherent given the structure of the thermal process. A Wiener–Hammerstein model has been estimated, where the fast dynamic of the actuator $\widehat{G}_w(s)$ has been separated from the slow dynamics of the Peltier cell $\widehat{G}_h(s)$.

A great advantage of using multistep signals for estimation of this type of models is that one can have a better panorama to analyse the graphical results. For example, an extended visual exploration of the results shown in Figure 24 showed that the process presents small changes in the dynamics, probably due to thermal drifts and other phenomena that may occur in real processes (see Figure 25) (variation of the dynamics shown in Figure 25 are not the only ones; other similar variations were detected over other portions of estimation and validation data). With a Gaussian excitation signal, in the event of a discrepancy between real output and model output, it would not be so easy to determine if this lack of precision is due to unmodeled dynamics, variation of the dynamics, or static nonlinearity that was not well captured.



(a)



(b)

Figure 24: Comparison between thermal process output (red) and estimated model output (blue dashed line). (a) Comparison on estimation dataset. (b) Comparison on validation dataset.

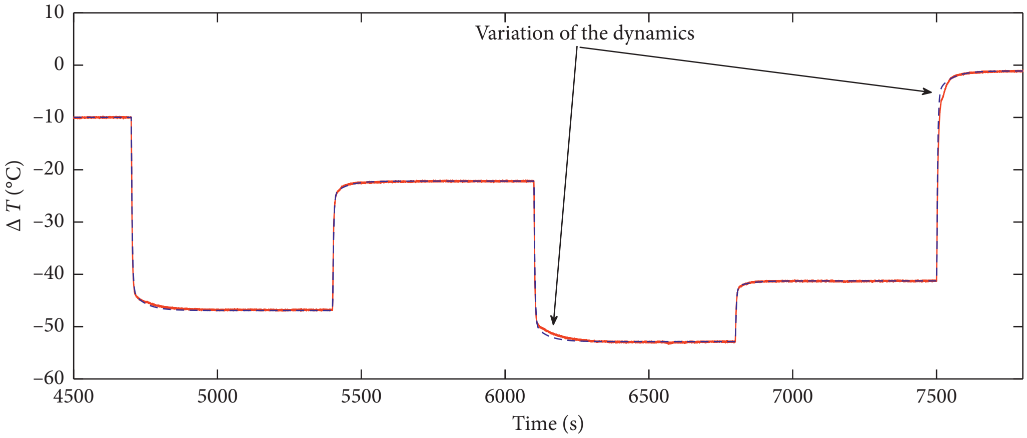


Figure 25: Variation of the dynamics detected on estimation dataset for the thermal process. Red: output of the thermal process. Blue dashed line: output of the estimated model.

The precision achieved in both the numerical examples and the real application depend to a large extent on the number of breakpoints used to capture the static nonlinearity. It is evident that a hard nonlinearity will require many points; however, this is not possible to determine until an initial estimate is made. After an initial estimation, the value reached by the objective function (index J) can give an idea of whether it is necessary to add more points to the piecewise function to reach a greater precision. This index can be compared with the process noise level or with the precision of the measuring instrument. If this information is not available, the precision of the nonlinear estimation can be evaluated with the index J and the range of the process output. Another way to establish if more points are required is through a visual comparison between the real and the modeled output. Since nonlinearity is static, the number of chosen points directly affects the steady-state error that may exist between the two outputs. This comparison is not possible when using Gaussian-type signals since these signals do not lead the system output to steady state.

In the case of the real application, the process output was bounded between −53.957°C and −0.181°C; therefore, the operating range was 53.776°C. For this operating range, the MAE between the real output and estimated output was 0.1435°C. As can be noted, the precision achieved with $n = 9$ was quite acceptable. Other estimates with a greater number of points were executed; however, the decrease in the error was negligible. In the case of the numerical examples, a noise signal of −30 dB was added to the output of each simulated model. The mean absolute value of this noise signal was 2.52×10^{-2}, and the MAE achieved by the three models is very close to the noise levels. It should be noted that in order to conclude that a good precision has been reached, the signal-to-noise ratio (SNR) must be considered. The three examples were excited with the same signal, and the SNR was approximately 60 dB. Although the results of the three numerical examples were quite acceptable, other estimates were made with the same algorithm configuration but with $n = 24$. The results obtained were slightly above those obtained with $n = 18$; however, it is very likely that the algorithm requires an increase in population size and generations to deal with more complex models. In this sense, it is not ruled out that in the numerical examples, it is possible to improve the accuracy of the models but surely a higher computational cost will be required for the algorithm execution and obviously the models will be more complex.

To date there is no recipe for assigning an optimal number of points for static nonlinearity. Since in the context of systems identification, precision and complexity are two conflicting objectives, a very interesting way to address this problem would be through a multiobjective optimisation approach.

Regarding computational cost of WH-EA (WH-EA was run on a computer with Intel Core I7 processor of 2.8 GHz and 8.0 Gb of RAM), a reference can be obtained. For example, in Section 4.3 (Wiener–Hammerstein example), the average time to run a generation was 0.07 s; however, it should be taken into consideration that the time required by the algorithm to execute all the tasks performed in a generation (mutations, crossovers, and selection) is only 1.42% (0, 99 ms) of the total time spent in a generation. The remaining 98.58% corresponds to the time it takes to evaluate the objective function. This evaluation involves an interpolation process and the simulation of one or two continuous LTI systems with a large amount of input data. It should be taken into account that the execution time of a generation is highly sensitive to the amount of data used for the nonlinear estimation. In both the numerical examples and the practical application, large amounts of data were used to demonstrate the great accuracy that can be achieved with WH-EA.

5. Conclusions

A unified approach to identify Wiener, Hammerstein, and Wiener–Hammerstein models has been presented. This paper shows that a smart parameter selection enables WH-EA to be used to independently identify Wiener, Hammerstein, and Wiener–Hammerstein models without a priori specification of the type of model to be estimated. This is highly attractive especially when a process must be identified and there is uncertainty about the distribution of the dynamics around nonlinearity. The performance of this approach has been evaluated with three numerical examples containing complex static nonlinearity and through a real process consisting of a lab scale thermal process based on a Peltier cell. The results show that WH-EA is enabled to estimate nonlinear models with good accuracy from an initial linear model that is not necessarily the BLA.

Data Availability

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

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This work was partially supported by projects DPI2015-71443-R and RTI2018-096904-B-I00 from the Spanish Ministry of Economy and Competitiveness and was also supported by Salesian Polytechnic University in Ecuador through a PhD scholarship granted to J. Z.

References

[1] A. Janczak, *Identification of Nonlinear Systems Using Neural Networks and Polynomial Models: A Block-Oriented Approach*, Vol. 310, Springer Science & Business Media, Berlin, Germany, 2004.

[2] F. Giri and E.-W. Bai, *Block-Oriented Nonlinear System Identification*, Vol. 1, Springer, Berlin, Germany, 2010.

[3] M. Schoukens and K. Tiels, “Identification of nonlinear block-oriented systems starting from linear approximations: a survey,” 2016, https://arxiv.org/abs/1607.01217.

[4] I. W. Hunter and M. J. Korenberg, “The identification of nonlinear biological systems: wiener and hammerstein cascade models,” *Biological Cybernetics*, vol. 55, no. 2-3, pp. 135–144, 1986.

[5] K. J. Hunt, M. Munih, N. d. N. Donaldson, and F. M. D. Barr, “Investigation of the hammerstein hypothesis in the modeling of electrically stimulated muscle,” *IEEE Transactions on Biomedical Engineering*, vol. 45, no. 8, pp. 998–1009, 1998.

[6] F. Le, I. Markovsky, C. T. Freeman, and E. Rogers, “Recursive identification of Hammerstein systems with application to electrically stimulated muscle,” *Control Engineering Practice*, vol. 20, no. 4, pp. 386–396, 2012.

[7] A. Kalafatis, N. Arifin, L. Wang, and W. R. Cluett, “A new approach to the identification of pH processes based on the wiener model,” *Chemical Engineering Science*, vol. 50, no. 23, pp. 3693–3701, 1995.

[8] J. C. Gómez and E. Baeyens, “Subspace-based identification algorithms for Hammerstein and Wiener models,” *European Journal of Control*, vol. 11, no. 2, pp. 127–136, 2005.

[9] Y. Zhu, “Distillation column identification for control using wiener model,” in *Proceedings of the 1999 American Control Conference (Cat. No. 99CH36251)*, vol. 5, pp. 3462–3466, IEEE, San Diego, CA, USA, June 1999.

[10] J. Oliver, R. Prieto, J. Cobos, O. Garcia, and P. Alou, “Hybrid Wiener-Hammerstein structure for grey-box modeling of DC-DC converters,” in *Proceedings of the 2009 Twenty-Fourth Annual IEEE Applied Power Electronics Conference and Exposition*, pp. 280–285, IEEE, Washington, DC, USA, February 2009.

[11] N. Patcharaprakiti, K. Kirtikara, V. Monyakul et al., “Modeling of single phase inverter of photovoltaic system using

Hammerstein-Wiener nonlinear system identification," *Current Applied Physics*, vol. 10, no. 3, pp. S532–S536, 2010.

[12] A. Haryanto and K.-S. Hong, "Maximum likelihood identification of Wiener–Hammerstein models," *Mechanical Systems and Signal Processing*, vol. 41, no. 1-2, pp. 54–70, 2013.

[13] O. P. Dewhirst, D. M. Simpson, N. Angarita, R. Allen, and P. L. Newland, "Wiener-Hammerstein parameter estimation using differential evolution," in *Proceedings of the International Conference on Bio-inspired Systems and Signal Processing*, pp. 271–276, Valencia, Spain, January 2010.

[14] K. P. Fruzzetti, A. Palazoğlu, and K. A. McDonald, "Nolinear model predictive control using Hammerstein models," *Journal of Process Control*, vol. 7, no. 1, pp. 31–41, 1997.

[15] J. C. Gómez, A. Jutan, and E. Baeyens, "Wiener model identification and predictive control of a pH neutralisation process," *IEE Proceedings—Control Theory and Applications*, vol. 151, no. 3, pp. 329–338, 2004.

[16] T. Patikirikorala, L. Wang, A. Colman, and J. Han, "Hammerstein-Wiener nonlinear model based predictive control for relative QoS performance and resource management of software systems," *Control Engineering Practice*, vol. 20, no. 1, pp. 49–61, 2012.

[17] S. Li and Y. Li, "Model predictive control of an intensified continuous reactor using a neural network wiener model," *Neurocomputing*, vol. 185, pp. 93–104, 2016.

[18] G. Shafiee, M. Arefi, M. Jahed-Motlagh, and A. Jalali, "Nonlinear predictive control of a polymerization reactor based on piecewise linear wiener model," *Chemical Engineering Journal*, vol. 143, no. 1-3, pp. 282–292, 2008.

[19] W. Greblicki and M. Pawlak, *Nonparametric System Identification*, Vol. 1, Cambridge University Press, Cambridge, UK, 2008.

[20] G. Mzyk, *Combined Parametric-Nonparametric Identification of Block-Oriented Systems*, Vol. 238, Springer, Berlin, Germany, 2014.

[21] J. Schoukens, J. Lataire, R. Pintelon, G. Vandersteen, and T. Dobrowiecki, "Robustness issues of the best linear approximation of a nonlinear system," *IEEE Transactions on Instrumentation and Measurement*, vol. 58, no. 5, pp. 1737–1745, 2009.

[22] J. Schoukens, R. Pintelon, and Y. Rolain, *Mastering System Identification in 100 Exercises*, John Wiley & Sons, Hoboken, NJ, USA, 2012.

[23] J. Schoukens and R. Pintelon, "Study of the variance of parametric estimates of the best linear approximation of nonlinear systems," *IEEE Transactions on Instrumentation and Measurement*, vol. 59, no. 12, pp. 3159–3167, 2010.

[24] H. K. R. Wong, *Study of the Best Linear Approximation of Nonlinear Systems with Arbitrary Inputs*, Ph.D. thesis, University of Warwick, Coventry, UK, 2013.

[25] P. L. dos Santos, J. A. Ramos, and J. M. de Carvalho, "Identification of a benchmark Wiener–Hammerstein: a bilinear and Hammerstein–bilinear model approach," *Control Engineering Practice*, vol. 20, no. 11, pp. 1156–1164, 2012.

[26] 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 Engineering Practice*, vol. 20, no. 11, pp. 1119–1125, 2012.

[27] D. T. Westwick and J. Schoukens, "Classification of the poles and zeros of the best linear approximations of Wiener-Hammerstein systems," *IFAC Proceedings Volumes*, vol. 45, no. 16, pp. 470–475, 2012a.

[28] A. H. Tan, H. K. Wong, and K. Godfrey, "Identification of a Wiener-Hammerstein system using an incremental nonlinear optimisation technique," *Control Engineering Practice*, vol. 20, no. 11, pp. 1140–1148, 2012.

[29] 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.

[30] D. T. Westwick and J. Schoukens, "Initial estimates of the linear subsystems of Wiener-Hammerstein models," *Automatica*, vol. 48, no. 11, pp. 2931–2936, 2012b.

[31] J. Sjöberg and J. Schoukens, "Initializing Wiener-Hammerstein models based on partitioning of the best linear approximation," *Automatica*, vol. 48, no. 2, pp. 353–359, 2012.

[32] L. Vanbeylen, "A fractional approach to identify Wiener-Hammerstein systems," *Automatica*, vol. 50, no. 3, pp. 903–909, 2014.

[33] 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, 2018.

[34] J. Zambrano, J. Sanchis, J. Herrero, and M. Martínez, "WH-EA: an evolutionary algorithm for Wiener-Hammerstein system identification," *Complexity*, vol. 2018, Article ID 1753262, 17 pages, 2018.

[35] R. Pintelon and J. Schoukens, *System Identification: A Frequency Domain Approach*, John Wiley & Sons, Hoboken, NJ, USA, 2012.

[36] K. Tiels, M. Schoukens, and J. Schoukens, "Initial estimates for Wiener-Hammerstein models using phase-coupled multisines," *Automatica*, vol. 60, pp. 201–209, 2015.

[37] P. Young and A. Jakeman, "Refined instrumental variable methods of recursive time-series analysis part I: single input, single output systems," *International Journal of Control*, vol. 29, no. 1, pp. 1–30, 1979.

[38] H. Garnier and P. C. Young, "The advantages of directly identifying continuous-time transfer function models in practical applications," *International Journal of Control*, vol. 87, no. 7, pp. 1319–1338, 2014.

[39] H. Gamier and M. Mensler, "Contsida continuous-time system identification toolbox for matlab®," in *Proceedings of the 1999 European Control Conference (ECC)*, pp. 3322–3327, IEEE, Karlsruhe, Germany, September 1999.

[40] A. Padilla, H. Garnier, and M. Gilson, "Version 7.0 of the CONTSID toolbox," *IFAC-PapersOnLine*, vol. 48, no. 28, pp. 757–762, 2015.

[41] H. Garnier, M. Mensler, and A. Richard, "Continuous-time model identification from sampled data: implementation issues and performance evaluation," *International Journal of Control*, vol. 76, no. 13, pp. 1337–1357, 2003.

[42] F. DeRidder, R. Pintelon, J. Schoukens, and D. P. Gillikin, "Modified AIC and MDL model selection criteria for short data records," *IEEE Transactions on Instrumentation and Measurement*, vol. 54, no. 1, pp. 144–150, 2005.