



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

Identification of nonlinear processes based on Wiener-Hammerstein models and heuristic optimization.

Doctoral Thesis

Author: Julio César Zambrano Abad

Advisors: Dr. Javier Sanchis Saez
Dr. Juan Manuel Herrero Durá

Valencia, June 2021

I wish to express my sincere gratitude to my advisors Prof. Javier Sanchis and Prof. Juan Manuel Herrero, to whom I owe much of my learning. I thank them for giving me the opportunity to access all their knowledge and experience during my doctoral studies and during the development of this thesis.

I would also like to thank my wife Mary for patiently supporting me all these years. I also thank and dedicate this work to my sons Julio Alejandro and Juan David, for being the greatest motivation to face all my challenges. Finally, I thank my parents Julio and Mariana for building my character and for teaching me to fight for my dreams.

Abstract

In several engineering fields, mathematical models are used to describe the behaviour of systems, processes or phenomena. Nowadays, there are several techniques or methods for obtaining mathematical models. Because of their versatility and simplicity, system identification methods are often preferred. Generally, systems identification methods require defining a structure and estimating computationally the parameters that make it up, using a set of procedures y measurements of the system's input and output signals. In the context of nonlinear system identification, a significant challenge is the structure selection. In the case that the system to be identified presents a static type of nonlinearity, block-oriented models consisting of the interaction of linear time-invariant (LTI) dynamic subsystems and static nonlinear elements (NL), can be useful to define a suitable structure. However, the designer may face a certain degree of uncertainty when selecting the block-oriented model in accordance with the real system. In addition to this inconvenience, the estimation of some block-oriented models is not an easy task, as is the case with the Wiener-Hammerstein models consisting of a NL block in the middle of two LTI subsystems.

The presence of two LTI subsystems in the Wiener-Hammerstein models is what mainly makes their estimation difficult. Generally, the identification procedure begins with the estimation of the linear dynamics, and the main challenge is to split this dynamic between the two LTI block. Usually, this implies a high user interaction to develop several procedures, and the final model estimated mostly depends on these previous stages.

The aim of this thesis is to contribute to the identification of the Wiener-Hammerstein models. This contribution is based on the presentation of two new algorithms to address specific aspects that have not been addressed in the identification of this type of model. The first algorithm, called WH-EA (An Evolutionary Algorithm for Wiener-Hammerstein System Identification), allows estimating all the parameters of a Wiener-Hammerstein model with a single procedure from a linear dynamic model. With WH-EA, a good estimate does not depend on intermediate procedures since the evolutionary algorithm looks for the best dynamic division, while the locations of the poles and zeros are fine-tuned, and nonlinearity is captured simultaneously. Another significant advantage of this algorithm is that under specific considerations and using a suitable excitation signal; it is possible to create a unified approach that also allows the identification of Wiener and Hammerstein models which are particular cases of the Wiener-Hammerstein model when one of its LTI blocks lacks dynamics. What is interesting about this unified approach is that with the same algorithm, it is possible to identify Wiener, Hammerstein, and Wiener-Hammerstein models without the user specifying in advance the type of structure to be identified.

The second algorithm called WH-MOEA (Multi-objective Evolutionary Algorithm for Wiener-Hammerstein identification), allows to address the identification problem as a Multi-Objective Optimisation Problem (MOOP). Based on this algorithm, a new approach for the identification of Wiener-Hammerstein models is presented considering a compromise between the accuracy achieved and the model complexity. With this approach, it is possible to compare several models with different performances, including as an identification target the number of parameters that the estimated model may have. The contribution of this approach is based on the fact that in many engineering problems the design requirements and user's preferences do not always point to the accuracy of the model as a single objective, but many times the complexity is also a predominant factor in decision-making.

Resumen

En muchos campos de la ingeniería los modelos matemáticos son utilizados para describir el comportamiento de los sistemas, procesos o fenómenos. Hoy en día, existen varias técnicas o métodos que pueden ser usadas para obtener estos modelos. Debido a su versatilidad y simplicidad, a menudo se prefieren los métodos de identificación de sistemas. Por lo general, estos métodos requieren la definición de una estructura y la estimación computacional de los parámetros que la componen utilizando un conjunto de procedimientos y mediciones de las señales de entrada y salida del sistema. En el contexto de la identificación de sistemas no lineales, un desafío importante es la selección de la estructura. En el caso de que el sistema a identificar presente una no linealidad de tipo estático, los modelos orientados a bloques que consisten en la interacción de subsistemas dinámicos lineales invariantes en el tiempo (LTI) y elementos estáticos no lineales (NL), pueden ser útiles para definir adecuadamente una estructura. Sin embargo, el diseñador puede enfrentarse a cierto grado de incertidumbre al seleccionar el modelo orientado a bloques adecuado en concordancia con el sistema real. Además de este inconveniente, se debe tener en cuenta que la estimación de algunos modelos orientados a bloques no es sencilla, como es el caso de los modelos de Wiener-Hammerstein que consisten en un bloque NL en medio de dos subsistemas LTI.

La presencia de dos subsistemas LTI en los modelos de Wiener-Hammerstein es lo que principalmente dificulta su estimación. Generalmente, el procedimiento de identificación comienza con la estimación de la dinámica lineal, y el principal desafío es dividir esta dinámica entre los dos bloques LTI. Por lo general, esto

implica una alta interacción del usuario para desarrollar varios procedimientos, y el modelo final estimado depende principalmente de estas etapas previas.

El objetivo de esta tesis es contribuir a la identificación de los modelos de Wiener-Hammerstein. Esta contribución se basa en la presentación de dos nuevos algoritmos para atender aspectos específicos que no han sido abordados en la identificación de este tipo de modelos. El primer algoritmo, denominado WH-EA (Algoritmo Evolutivo para la identificación de sistemas de Wiener-Hammerstein), permite estimar todos los parámetros de un modelo de Wiener-Hammerstein con un solo procedimiento a partir de un modelo dinámico lineal. Con WH-EA, una buena estimación no depende de procedimientos intermedios ya que el algoritmo evolutivo simultáneamente busca la mejor distribución de la dinámica, ajusta con precisión la ubicación de los polos y los ceros y captura la no linealidad estática. Otra ventaja importante de este algoritmo es que bajo consideraciones específicas y utilizando una señal de excitación adecuada, es posible crear un enfoque unificado que permite también la identificación de los modelos de Wiener y Hammerstein, que son casos particulares del modelo de Wiener-Hammerstein cuando uno de sus bloques LTI carece de dinámica. Lo interesante de este enfoque unificado es que con un mismo algoritmo es posible identificar los modelos de Wiener, Hammerstein y Wiener-Hammerstein sin que el usuario especifique de antemano el tipo de estructura a identificar.

El segundo algoritmo llamado WH-MOEA (Algoritmo evolutivo multi-objetivo para la identificación de modelos de Wiener-Hammerstein), permite abordar el problema de identificación como un Problema de Optimización Multiobjetivo (MOOP). Sobre la base de este algoritmo se presenta un nuevo enfoque para la identificación de los modelos de Wiener-Hammerstein considerando un compromiso entre la precisión alcanzada y la complejidad del modelo. Con este enfoque es posible comparar varios modelos con diferentes prestaciones incluyendo como un objetivo de identificación el número de parámetros que puede tener el modelo estimado. El aporte de este enfoque se sustenta en el hecho de que en muchos problemas de ingeniería los requisitos de diseño y las preferencias del usuario no siempre apuntan a la precisión del modelo como un único objetivo, sino que muchas veces la complejidad es también un factor predominante en la toma de decisiones.

Resum

En molts camps de l'enginyeria els models matemàtics són utilitzats per a descriure el comportament dels sistemes, processos o fenòmens. Hui dia, existeixen diverses tècniques o mètodes que poden ser usades per a obtenir aquests models. A causa de la seua versatilitat i simplicitat, sovint es prefereixen els mètodes d'identificació de sistemes. En general, aquests mètodes requereixen la definició d'una estructura i l'estimació computacional dels paràmetres que la componen utilitzant un conjunt de procediments i mesuraments dels senyals d'entrada i eixida del sistema. En el context de la identificació de sistemes no lineals, un desafiament important és la selecció de l'estructura. En el cas que el sistema a identificar presente una no linealitat de tipus estàtic, els models orientats a blocs que consisteixen en la interacció de subsistemes dinàmics lineals invariants en el temps (LTI) i elements estàtics no lineals (NL), poden ser útils per a definir adequadament una estructura. No obstant això, el dissenyador pot enfrontar-se a cert grau d'incertesa en seleccionar el model orientat a blocs adequat en concordança amb el sistema real. A més d'aquest inconvenient, s'ha de tindre en compte que l'estimació d'alguns models orientats a blocs no és senzilla, com és el cas dels models de Wiener-Hammerstein que consisteixen en un bloc NL enmig de dos subsistemes LTI.

La presència de dos subsistemes LTI en els models de Wiener-Hammerstein és el que principalment dificulta la seua estimació. Generalment, el procediment d'identificació comença amb l'estimació de la dinàmica lineal, i el principal desafiament és dividir aquesta dinàmica entre els dos blocs LTI. En general, això implica una alta interacció de l'usuari per a desenvolupar diversos

procediments, i el model final estimat depèn principalment d'aquestes etapes prèvies.

L'objectiu d'aquesta tesi és contribuir a la identificació dels models de Wiener-Hammerstein. Aquesta contribució es basa en la presentació de dos nous algorismes per a atendre aspectes específics que no han sigut adreçats en la identificació d'aquesta mena de models. El primer algorisme, denominat WH-EA (Algorisme Evolutiu per a la identificació de sistemes de Wiener-Hammerstein), permet estimar tots els paràmetres d'un model de Wiener-Hammerstein amb un sol procediment a partir d'un model dinàmic lineal. Amb WH-EA, una bona estimació no depèn de procediments intermedis ja que l'algorisme evolutiu simultàniament busca la millor distribució de la dinàmica, afina la ubicació dels pols i els zeros i captura la no linealitat estàtica. Un altre avantatge important d'aquest algorisme és que sota consideracions específiques i utilitzant un senyal d'excitació adequada, és possible crear un enfocament unificat que permet també la identificació dels models de Wiener i Hammerstein, que són casos particulars del model de Wiener-Hammerstein quan un dels seus blocs LTI manca de dinàmica. L'interessant d'aquest enfocament unificat és que amb un mateix algorisme és possible identificar els models de Wiener, Hammerstein i Wiener-Hammerstein sense que l'usuari especifique per endavant el tipus d'estructura a identificar.

El segon algorisme anomenat WH-MOEA (Algorisme evolutiu multi-objectiu per a la identificació de models de Wiener-Hammerstein), permet abordar el problema d'identificació com un Problema d'Optimització Multiobjectiu (MOOP). Sobre la base d'aquest algorisme es presenta un nou enfocament per a la identificació dels models de Wiener-Hammerstein considerant un compromís entre la precisió aconseguida i la complexitat del model. Amb aquest enfocament és possible comparar diversos models amb diferents prestacions incloent com un objectiu d'identificació el nombre de paràmetres que pot tindre el model estimat. L'aportació d'aquest enfocament se sustenta en el fet que en molts problemes d'enginyeria els requisits de disseny i les preferències de l'usuari no sempre apunten a la precisió del model com un únic objectiu, sinó que moltes vegades la complexitat és també un factor predominant en la presa de decisions.

Contents

Abstract	iii
1 Introduction	1
1.1 Block-oriented models	2
1.2 Wiener-Hammerstein models	4
1.3 Motivation and objectives of the thesis	7
1.4 Overview and Outline	10
2 WH-EA: evolutionary algorithm for Wiener-Hammerstein system identification	13
2.1 Introduction	14
2.2 Discrete-time formulation	15
2.3 The BLA of a Wiener-Hammerstein system	16
2.4 Genetic coding and search space	18
2.5 Optimisation problem statement	22
2.6 Genetic operations	23
2.7 WH-EA description	31
2.8 Application of WH-EA and results	35

3	A unified approach for the identification of Wiener, Hammerstein, and Wiener-Hammerstein models.	49
3.1	Introduction	50
3.2	Continuous-time formulation for the unified approach	52
3.3	Initial linear model.	54
3.4	Optimisation problem statement.	55
3.5	Genetic coding.	56
3.6	Search space for poles and zeros	58
3.7	Search space for static nonlinearity	58
3.8	The multi-step signal	64
3.9	Application examples	68
4	WH-MOEA: A Multi-objective Evolutionary Algorithm for Wiener-Hammerstein system identification. A novel approach for trade-off analysis between complexity and accuracy.	87
4.1	Introduction	88
4.2	Preliminary considerations.	90
4.3	Multi-objective optimization problem statement	93
4.4	Wiener-Hammerstein Multi-objective Evolutionary Algorithm (WH-MOEA).. .	95
4.5	Using WH-MOEA to compare design concepts: a procedure	101
4.6	Evaluation of WH-MOEA	104
4.7	Application of WH-MOEA and trade-off analysis using design concepts	112
5	Conclusions and future research directions	133
5.1	Conclusions.	134
5.2	Future research directions	139
	Bibliography	141

Chapter 1

Introduction

The opening section of this chapter provides a brief introduction to block-oriented models and a literature review on Wiener-Hammerstein models, pointing out the main characteristics of the existing identification methods. Next, thesis motivation and objectives are presented. Finally, a brief description of the structure of the thesis can be found in the fourth section.

1.1 Block-oriented models

Non-linearities are present to a greater or lesser extent in all real processes. For weak non linearities, linear models can be successfully used to forecast the evolution of variables or to design control schemes. Currently, a lot of methods to build linear models can be found in the literature [91, 70, 62, 109, 65]. However, against hard non linearities, linear models just can be used in a specific operation range. If process is controlled in a wide operating range, cause-effect relationships should be represented by nonlinear models. One alternative is the use of rigorous first-principles formulation [24, 50] as nonlinear models. Nevertheless in most cases this formulation can be a very challenging task. Another option is process identification using soft computing methods. Nonlinear system identification has attracted considerable interest of researchers over the past few years. Nowadays, nonlinear identification is an open research topic where some benchmark problems have been proposed [95, 117, 96, 76] and real measurement data are available for testing and validate different nonlinear identification methods (e.g. DaISy database [16]).

One of the most challenging problems regarding nonlinear system identification is the selection of a suitable model structure. Currently there are several structures based on Neural Networks [61], block-oriented models [35, 47], Volterra series [21], NARMAX models [10], fuzzy models [79], among others. A review of black box methods to nonlinear identification can be found in [105].

Block-oriented models are a class of nonlinear models used to represent dynamic systems affected by static non-linearities. These models consist of the interaction of Linear Time-Invariant (LTI) subsystems and nonlinear (NL) static elements. Today, a wide range of such structures can be found in the literature. An overview of the most common block-oriented model structures can be found in [101]. Study of block-oriented models not only addresses parameter estimation. These nonlinear structures have found an important place within advanced control strategies such as Model Predictive Control (MPC), however, the direct use of these models within this strategy leads to a non-convex optimisation problem. In this sense, several contributions have been made to solve this inconvenience. [36, 60, 58, 123, 59].

In the context of block-oriented models, LTI subsystems are typically represented by rational transfer functions, impulse response models, auto-regressive models, Box–Jenkins (BJ) models, Output Error (OE) models and state-space models; while non-linearity blocks are usually parameterized with polynomials, piece-wise functions, splines, and neural networks. 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 non-parametric form [37] or in a combined way [71]. On the other hand, block-oriented models are not only useful to single-input, single-output (SISO) systems. There are currently several approaches to modelling multivariable systems using these structures [38, 124, 107, 1, 115, 51, 8].

The interaction between LTI subsystems and NL elements is not restricted to serial connections since blocks can be used more than once. Therefore, block-oriented models comprise a large diversity of structures that can be useful when modelling dynamic systems affected by memory-less non-linearities. The simplest structures within this class of models 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. These four structures are based on serial connections and their great usefulness for modelling real systems is reflected in the large number of studies developed in comparison with block-oriented models based on feedback connections [85, 83] and parallel branches [97, 98, 100].

In series or parallel block-oriented models, the connection of two blocks gives rise to an internal variable. Generally these variables do not have a real physical representation or they are not accessible for measurement. In Wiener models the internal variable can be estimated from the excitation signal, once the linear dynamics of the system are known. On the other hand, in Hammerstein models, it is also possible to estimate the internal variable from the system's output signal, however, this implies assuming that the LTI block is invertible, which is not recommended, as it reduces the model's flexibility and it is impractical.

Rigorously, in a practical context, all the parameters of a block oriented model might be estimated from input and output measurements. Without an a priori knowledge of the system under test, the designer may face several inconveniences. The main dilemma will surely be the selection of the structure. On the other hand, the way in which each of the blocks of the model should be parameterised is also often questioned. This last issue can be solved by selecting the identification method to be used. Generally, each identification method implicitly brings with it a certain way of parameterizing the blocks. However, to select the estimation method the designer must be clear about the structure of the system. In this regard, some authors have concentrated efforts on proposing techniques to detect the appropriate structure from linear approximations of the nonlinear system [94, 57]. Although these techniques do not accurately state whether the system under test corresponds

to a specific structure, through their use it is possible to rule out a set of structures based on certain patterns. It should also be noted that the structure detection through these techniques is an additional procedure prior to using any identification method. This obviously demands more user interaction to estimate a block-oriented model.

This dissertation focuses on parametric block-oriented models based on serial connections containing a single nonlinear block, i.e., Wiener, Hammerstein, and Wiener-Hammerstein models. A Wiener model consists of a dynamic LTI subsystem connected in series with a nonlinear static element. This structure was introduced in 1958 by Norbert Wiener [116]. Various applications of Wiener's models on real processes can be found in the literature including biological systems [46, 54], chemical processes such as pH [49, 36, 112] or distillation columns [125, 11].

On the other hand, a Hammerstein model consists of a nonlinear static element connected in series with a dynamic LTI subsystem. This structure was introduced in 1930 by the mathematician A. Hammerstein [40], however, the first identification algorithm was presented in 1966 by Narendra and Gallman [75]. Like Wiener's models, Hammerstein's models have also been widely used to model real processes including distillation columns [25], heat ex-changers [86, 20], biological systems [14, 18] and solid oxide fuel cells [48].

The Wiener and Hammerstein models are specific cases of the Wiener-Hammerstein model when one of the LTI blocks lacks dynamics. Identification of Wiener-Hammerstein is more complex than models with just a single LTI block and several stages may be required. This complexity has attracted the interest of researchers and for this reason, a large number of methods for identifying these models can be found in the literature. A review of some identification methods for Wiener-Hammerstein models is presented in the following section.

1.2 Wiener-Hammerstein models

A Wiener-Hammerstein (WH) model (Figure 1.1) consists of the cascading of an LTI input subsystem (LTI_1), a nonlinear element, and an LTI output subsystem (LTI_2). Wiener-Hammerstein models have proved to be able to describe non-linear dynamics of several systems such as: a paralyzed skeletal muscle [6, 5, 13], a limb reflex control system [19], a DC-DC converter [78] a heat exchanger system with a superheater-desuperheater in a boiler system [41] and a thermal process [7], among others [53].

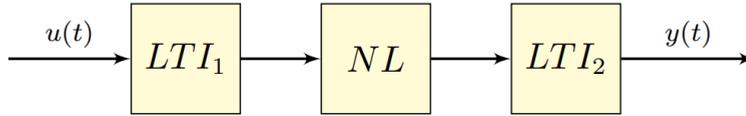


Figure 1.1: Wiener-Hammerstein model.

Early studies related to the estimation of WH structures were mainly based on correlation analysis and date back to the 1970s [31, 22, 9]. From that time to date, the interest of researchers has been latent and several identification methods for this structure have been presented. This interest has been based on the degree of complexity involved in estimating it. The main problem lies in the presence of two LTI blocks, which implies that the general dynamics of the system must be distributed. On the other hand, the internal variables, i.e, the input and output of the nonlinear block, generally are not accessible for measurement.

Knowledge of linear dynamics can be a good starting point to identify a Wiener-Hammerstein model [101]. In this regard, the Best Linear Approximation or BLA of a nonlinear system [64, 23, 92, 101] can be used. For the specific case of Wiener-Hammerstein models, the BLA does not provide information about the dynamics of each LTI block. Therefore, all BLA-based Wiener-Hammerstein identification methods have concentrated their efforts on the BLA division to generate initial estimates and avoid sub-optimal local minima in an optimisation procedure. In [102] both LTI subsystems are initialised with all possible BLA partitions and least squares optimisation is applied to fit the non-linearity. Although identification results are very good, the number of possible partitions (combinations) grows with the number of poles and zeros of the BLA and therefore the computational cost required for this method can be very high.

To avoid multiple BLA divisions, in [56, 102] an “advanced” method is proposed where both LTI subsystems are over-parameterised with all poles and zeros of the BLA. This method is formulated as a linear-in-the-parameters total-least-squares problem for which the back LTI subsystem is inverted and basis functions are used to represent both linear subsystems. To minimise the effect of over-parameterisation, an order reduction technique is applied. However, since the formulation is based on neglecting the effect of disturbances, the solution is in general not consistent if there is noise on the output. In addition the BLA is required to be invertible.

Another approach to initialise Wiener-Hammerstein models is presented by [113], where the poles and/or zeros of the BLA are classified by using a nonlinear transformation of the input and the output residuals (Quadratic/Cubic BLA). On the same context of QBLA/CBLA and in line with “brute-force” method, [114] propose a scanning technique for a rapid evaluation of all possible BLA partitions between both LTI blocks of the Wiener-Hammerstein system. With this evaluation the vast majority of possible partitions are discarded. Both proposals based on QBLA/CBLA show excellent results and overcome some disadvantages of “brute-force” and “advanced” methods, however the QBLA/CBLA estimation can be difficult (high variance) since it is estimated from the BLA residuals.

A more robust method based on QBLA/CBLA is proposed by [99]. Unlike the two previous proposals the BLA is splitted in a non-parametric framework. This avoids mainly the parameterization of the QBLA that can be tedious given that the number of poles and zeros tends to be high. Once the front and the back dynamics of the Wiener-Hammerstein model have been identified, a parameterization of both LTI blocks is required in an additional step. This step can be complicated because a linear phase shift can be present in the non-parametric estimate.

To avoid QBLA/CBLA estimation, in [110] a fractional model parameterization based in multiplicities (powers) of poles and zeros is presented. The problem is formulated in the frequency domain and fractional exponents indicate the partitioning of the poles and zeros between the two linear parts after a optimisation problem is solved. Once the poles and zeros of the BLA have been classified, both LTI blocks must be parameterized in an additional step.

The fractional approach is discussed in detail in [34]. Here it is pointed out that the formulation becomes ill-conditioned for some configurations of the poles and zeros of the linear dynamic parts. To solve this drawback a expanded fractional approach which is exact for integer values of the exponents is introduced. In addition, a novel formulation for a faster and more reliable convergence to the solution when using Newton-type methods is proposed.

In [42] and [72], another alternative for estimating Wiener-Hammerstein models is proposed. This alternative is based on a multi-stage identification strategy, in which the static nonlinear characteristic and the linear dynamic blocks are estimated separately. The identification procedure starts with the non-parametric estimation of the static non-linearity. In a next step, both LTI

blocks represented by their impulse responses are estimated assuming that the order of each subsystem is known.

A different identification alternative is proposed by [74]. This approach is based on genetic recombination and particle swarm optimisation. During the search process, the algorithm requires minimal user interaction, but even though a large number of poles and zeros are allowed for both LTI subsystems, good accuracy cannot be achieved since the optimisation problem does not use a linear approximation as a starting point. Furthermore, a polynomial is used for the static non-linearity, which is not recommended, since the sensitivity of the coefficients increases with the degree of the polynomial. An important feature of this approach is that the identification problem is addressed as a multi-objective optimization problem (MOOP), where the accuracy and complexity of the models are penalised.

Almost all the identification methods mentioned in this section, including the evolutionary algorithm proposed as part of this research work, have been evaluated on a well-known Wiener-Hammerstein benchmark [95]. A comparison of the results can be seen in Subsection 2.8.2.

1.3 Motivation and objectives of the thesis

As mentioned above, in the case of Wiener-Hammerstein models, identification is not so easy since the linear dynamics must be divided in two. Table 1.1 summarises the most important features of the methods addressed in the previous section. As can be seen, most methods use BLA as a starting point for nonlinear estimation and are conditioned to use Gaussian excitation signals, and most methods have a high User Interaction Level (UIL). In these cases, user intervention is required to carry out intermediate procedures, analyse results and make decisions. This aspect is even more critical in methods with a non-parametric approach or where the QBLA/CBLA is required.

Although state of 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

Table 1.1: Features of the most relevant Wiener-Hammerstein model identification methods presented in the last decade.

Identification method	Excitation signal	Initial linear model	UIL	Additional requirements
Brute-force [102]	Gaussian	BLA	Very high	
Advanced method [102]	Gaussian	BLA	High	Inverse of the BLA
Scanning technique [114]	Gaussian	BLA	High	QBL/CBLA
Biosocial culture [74]	Non-restricted	Not required	Low	
Fractional approach [110]	Gaussian	BLA	Medium	
Modified fractional approach [34]	Gaussian	BLA	Medium	
Nonparametric QBLA [99]	Gaussian	BLA	High	QBLA
Multistage identification [42, 72]	Random sequence	No required	High	
Classification of poles and zeros [113]	Gaussian	BLA	Very high	QBL/CBLA and manual tuning

normally distributed excitation signals [84]. The most common signals of this type are the Gaussian noise sequences and random-phased multisines [93, 108] 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 [101]. 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 non-linearity, 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 non-linearity, the user would be obliged to make independent and 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 non-linearity. 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 cases, to achieve a good convergence it is necessary to define appropriately the range where the static non-linearity will be captured. Since static non-linearity is located in different positions respect to the dynamics, the non-linearity bounds are different for the three types of structures. Note that it is not the same to capture the static non-linearity before or after a dynamic block given that the domain and co-domain of the nonlinear function will change notably. Defining a very small search space will result in the non-linearity 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 non-linearity could be defined incorrectly.

This thesis aims to contribute in the field of nonlinear identification, specifically, all efforts have been concentrated on estimation of Wiener-Hammerstein models and their specific cases which are called Wiener models and Hammerstein models. The objectives are based on the development of new ways to identifying these structures, for which the following issues are considered:

- The current methods for identifying Wiener-Hammerstein models generally require a considerable user interaction. In addition, the quality of the final results depends on intermediate procedures which may be subject to errors. The initial objective of this research work is to develop an evolutionary algorithm to estimate all the parameters of a Wiener-Hammerstein model in a single procedure. This implies that the algorithm must have the ability to split the BLA, refine the position of the poles and zeros and capture static non-linearity at the same time. With this algorithm the user interaction will be minimal and the final results will not depend much on intermediate procedures.
- Although the Wiener and Hammerstein models are specific cases of the Wiener-Hammerstein model, to date there is no common method that can identify these three types of structures. In addition, not all processes are enabled to be excited with signals that allow the BLA be obtained. For this reason, a tool to identify Wiener, Hammerstein and Wiener-Hammerstein models without the need for the user to specify in advance the type of structure to be identified will be developed. On the

other hand, taking advantage of the characteristics of the evolutionary algorithm developed, it is proposed to start from an initial linear model obtained from a simple step response. Obviously, this model will be affected by static nonlinearity which is not the case with BLA, however, the evolutionary algorithm will refine the position of the poles and zeros as the dynamics are distributed and non-linearity is captured.

- Currently, most block oriented identification methods are focused on obtaining a model with high precision. However, in practical applications, precision should not be the only criteria for model selection. In that sense, this thesis also addresses the complexity of the model as a criterion for selecting the best model. To this end, an approach based on a multi-objective evolutionary algorithm will be presented.

1.4 Overview and Outline

Figure 1.2 shows a graphical abstract illustrating the core structure of this thesis which is based on three contributions that have been developed in the field of Wiener-Hammerstein model identification. Around these contributions the chapters of this dissertation have been organised as follows:

In Chapter 2, an evolutionary algorithm for Wiener-Hammerstein system identification (WH-EA) is presented. Since the BLA model is used as starting point, the system to be identified must be excited with Gaussian type signals or equivalent. Thanks to customised genetic operators, WH-EA estimates all the parameters of a Wiener-Hammerstein model in a single procedure, i.e, no user interaction is required to develop intermediate steps. In other words, WH-EA is able to look for the best BLA split, capturing at the same time the process static non-linearity with high precision. Furthermore, to correct possible errors in BLA estimation, the locations of poles and zeros are subtly modified within an adequate search space to allow a fine-tuning of the model. The performance of the algorithm is analysed by using a demonstration example and a nonlinear system identification benchmark [95]. This chapter is a literal transcription with minor modifications of the following research publication [122]:

- J. Zambrano, J. Sanchis, J. Herrero, and M. Martínez, “WH-EA: an evolutionary algorithm for Wiener-Hammerstein system identification”, *Complexity*, Volume 2018, 2018

As mentioned above, the use of Gaussian signals in real applications may be impractical. This drawback led research to a new approach based on multi-step

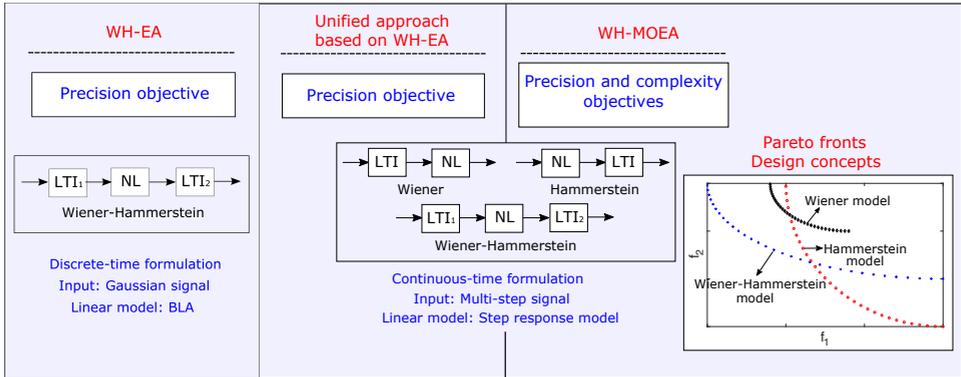


Figure 1.2: Graphical abstract of the thesis

excitation signals and an initial linear model obtained with standard linear identification techniques. Multi-step signals are more practical and easier to design than Gaussian signals, however this is not the only advantage. Thanks to the use of multi-step signals, WH-EA can be used to estimate Wiener, Hammerstein and Wiener-Hammerstein models without the user anticipating the type of structure to be estimated. Details about this approach which has been called *the unified approach*, as well as their evaluation on three numerical simulation examples and a real thermal process, are presented in Chapter 3. It must be remarked that this chapter is a literal transcription with minor modifications of the following research publication [121]:

- J. Zambrano, J. Sanchis, J. Herrero, and M. Martínez, “A unified approach for the identification of Wiener, Hammerstein, and Wiener–Hammerstein models by using WH-EA and multistep signals”, *Complexity*, Volume 2020, 2020.

More contributions on Wiener-Hammerstein models identification is presented in Chapter 4. In this chapter, the identification problem of Wiener-Hammerstein models and their specific cases is addressed as a multi-objective optimization problem (MOOP) managing a trade-off between accuracy and model complexity. To solve the MOOP, a new multi-objective evolutionary algorithm (MOEA) is proposed (WH-MOEA). From a linear structure, WH-MOEA will generate a set of optimal models considering a static non-linearity described by a variable number of points. Using WH-MOEA, a procedure is also proposed to analyse various initial linear structures with different numbers of poles and zeros (known as design concepts). A comparison

of the Pareto fronts of each design concept allows a more in-depth analysis to select the most appropriate model according to the user's needs. As in the previous chapters, this chapter is a literal transcription with minor modifications of the following research publication [120]:

- J. Zambrano, J. Sanchis, J. Herrero, and M. Martínez, “WH-MOEA: A Multi-objective Evolutionary Algorithm for Wiener-Hammerstein system identification. A novel approach for trade-off analysis between complexity and accuracy”, in *IEEE Access*, vol. 8, pp. 228655-228674, 2020, doi: 10.1109/ACCESS.2020.3046352.

Finally, the most important conclusions of this research work are reported in Chapter 5. In addition, based on the research carried out, this chapter highlights some future research work on the identification of Wiener-Hammerstein models.

WH-EA: evolutionary algorithm for Wiener-Hammerstein system identification

This chapter describes the structure of WH-EA and its way of working to estimate all the parameters of a Wiener-Hammerstein model. In the first section, a brief introduction to the algorithm and some guidelines are given. The second section presents the discrete-time formulation of the Wiener-Hammerstein model. The third section presents background information on the BLA. The fourth section presents the genetic coding of the individuals and the search space to explore possible solutions to the optimisation problem. The fifth section below shows how the optimization problem is stated. In the sixth section, the customised genetic operations used by the evolutionary algorithm are described in detail, while a comprehensive description of the evolutionary algorithm is given in the following section. The last section presents an evaluation of the algorithm, for which a numerical example and a well-known benchmark for the identification of Wiener-Hammerstein models have been considered.

2.1 Introduction

In this chapter an elitist evolutionary algorithm (WH-EA) for the identification of Wiener-Hammerstein models is presented. This algorithm evolves a population of NP individuals and each individual contains genetic information related to the parameters of a Wiener-Hammerstein model. Like any other evolutionary algorithm, WH-EA is inspired by biological evolution over generations. Starting from an initial population, new generations are created using information of the current generation g and performing crossover and/or mutation operations and selection based on the fitness of the new individuals.

Like almost all Wiener-Hammerstein model identification methods, this algorithm requires a linear model that represents the general dynamics of the system. In this case the BLA will be used. The intention of this algorithm is to estimate in a single procedure all the parameters of a Wiener-Hammerstein model. This is, WH-EA will be enabled to look for the best BLA split capturing at the same time the process static nonlinearity with high precision. Furthermore, to correct possible errors in BLA estimation, the location of poles and zeros are subtly modified within an adequate search space to allow a fine-tuning of the model. Unlike other identification methods, WH-EA does not require a user interaction to perform and make decisions in intermediate procedures.

The following issues were considered in the creation of this algorithm:

- A Wiener-Hammerstein system must be estimated from an input/output data set of N samples ($\{u(t), y(t)\}_{t=1}^N$). For further generalisation it is assumed that the internal variables of the system are not known.
- The BLA is estimated in the first instance.
- The poles and zeros of the BLA must be classified to find the dynamics of the front and back of the Wiener-Hammerstein model.
- The pole-zero locations of the BLA can change moderately to improve modelling errors.
- Without loss of generality, it is possible to model a Wiener-Hammerstein system considering that both linear blocks have unit gain.
- The nonlinear static function is modelled as a piece-wise function represented by a set of n breakpoints.

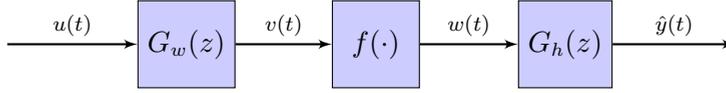


Figure 2.1: Wiener-Hammerstein model.

2.2 Discrete-time formulation.

A Wiener-Hammerstein model consists of two LTI subsystems $G_w(z)$ and $G_h(z)$ surrounding a static nonlinear function $f(v(t))$ (Figure 2.1). Both LTI subsystems can be represented in the discrete time domain as rational transfer functions in factorized form:

$$v(t) = G_w(z)u(t) = K_w \frac{\prod_{i=1}^{n_b} (z - z_{w_i}) / (1 - z_{w_i})}{\prod_{i=1}^{n_a} (z - p_{w_i}) / (1 - p_{w_i})} u(t) \quad (2.1)$$

$$\hat{y}(t) = G_h(z)w(t) = K_h \frac{\prod_{i=1}^{n_d} (z - z_{h_i}) / (1 - z_{h_i})}{\prod_{i=1}^{n_c} (z - p_{h_i}) / (1 - p_{h_i})} w(t) \quad (2.2)$$

where z is the discrete-time operator, $K_w, p_{w_1} \dots p_{w_{n_a}}$ and $z_{w_1} \dots z_{w_{n_b}}$ represent the static gain, poles and zeros of the front LTI block respectively, while $K_h, p_{h_1} \dots p_{h_{n_c}}$ and $z_{h_1} \dots z_{h_{n_d}}$ represent the static gain, poles and zeros of the back LTI block respectively.

The nonlinearity can be represented as a linear combination of a finite set (M) of basis functions:

$$w(t) = f(v(t)) = \sum_{m=1}^M \beta_m f_m(v(t)) \quad (2.3)$$

where $v(t)$ and $w(t)$ are the input and output of the static nonlinearity, β_m are weighting parameters to be estimated and f_m are basis functions.

From (2.1), (2.2) and (2.3), the output of the Wiener-Hammerstein model is analytically related to the input through the following expression:

$$\hat{y}(t, \theta) = G_h(z, \theta_h) f(\theta_{NL}, G_w(z, \theta_w) u(t)) \quad (2.4)$$

where:

$$\boldsymbol{\theta}_w = [K_w, z_{w_1}, z_{w_2} \dots z_{w_n}, p_{w_1}, p_{w_2} \dots p_{w_n}] \quad (2.5)$$

$$\boldsymbol{\theta}_{NL} = [\beta_1, \beta_2 \dots, \beta_m] \quad (2.6)$$

$$\boldsymbol{\theta}_h = [K_h, z_{h_1}, z_{h_2} \dots z_{h_n}, p_{h_1}, p_{h_2} \dots p_{h_n}] \quad (2.7)$$

$$\boldsymbol{\theta} = [\boldsymbol{\theta}_w, \boldsymbol{\theta}_{NL}, \boldsymbol{\theta}_h] \quad (2.8)$$

The challenge is to find the best $\boldsymbol{\theta}$ so that the predicted output $\hat{y}(t, \boldsymbol{\theta})$ is as close as possible to the measured output $y(t)$.

As a complement to the formulation presented, the following assumptions are made about the system:

Assumption 1 The nonlinear system to be identified can be described by (2.4).

Assumption 2 The Wiener-Hammerstein system will be identified from an input/output data set $\{u(t), y(t)\}_{t=1}^N$. The input signal $u(t)$ is Gaussian or equivalent, while the measured output $y(t)$ may be corrupted by stationary additive noise $n(t)$. It is further assumed that the noise is independent of the input excitation signal:

$$y(t) = y_0(t) + n(t) \quad (2.9)$$

Assumption 3 There is no cancellation of poles and zeros and all poles of both LTI subsystems must be within the unit circle.

Assumption 4 Non-linearity is static and its current output $w(t)$ only depends on the current input $v(t)$ (i.e., the nonlinearity has no memory).

2.3 The BLA of a Wiener-Hammerstein system.

The BLA of a nonlinear system for a given class of excitation signals is a linear model that minimises the expected mean square error between the true output of the nonlinear system and the output of the linear model [93]:

$$G_{BLA}(z) = \arg \min_{G(z)} E[|y(t) - G(z)u(t)|^2] \quad (2.10)$$

where $u(t)$ is the input that excites the nonlinear system, $y(t)$ is the measured output and E is the expectation operator. An alternative way to obtain the BLA of a nonlinear system is in a non-parametric framework:

$$G_{BLA}(j\omega_k) = \frac{S_{yu}(j\omega_k)}{S_{uu}(j\omega_k)} \quad (2.11)$$

where $S_{yu}(j\omega_k)$ is the cross-power spectrum between the output $y(t)$ and the input $u(t)$, and $S_{uu}(j\omega_k)$ is the auto power spectral density of $u(t)$ [23, 84].

The BLA depends on the excitation power spectrum (bandwidth and amplitude level) and excitation probability density function. Therefore obtaining the BLA is restricted to the type of input signal that excites the process. Most estimation methods to obtaining the BLA use Gaussian noise signals or equivalent [90].

When a nonlinear system is excited with a Gaussian signal or equivalent, according to Bussgang's theorem [12], the non-linearity can be replaced by a constant (K_{NL}). Therefore, in the specific case of a Wiener-Hammerstein system the BLA can be defined by the following expression:

$$G_{BLA}(z) = K_{NL}G_{wh}(z) \quad (2.12)$$

where $G_{wh}(z)$ represents the dynamics of the nonlinear system:

$$G_{wh}(z) = \frac{\prod_{i=1}^{n_b+n_d} (z - z_i)/(1 - z_i)}{\prod_{i=1}^{n_a+n_c} (z - p_i)/(1 - p_i)} u(t) \quad (2.13)$$

It is evident that $p_1 \dots p_{n_a+n_c}$ and $z_1 \dots z_{n_b+n_d}$ are the poles and zeros that must be assigned to $G_w(z)$ and $G_h(z)$. Although the BLA does not provide information to distinguish the dynamics between both LTI subsystems, knowledge of the overall dynamics of a Wiener-Hammerstein system, is a good starting point to identify such systems.

2.4 Genetic coding and search space

Theoretically in a Wiener-Hammerstein model the pole-zero locations of $G_w(z)$ and $G_h(z)$ subsystems correspond to the pole-zero locations of the BLA, however it is well known that once the BLA has been divided a re-fit can be used to improve the modelling error. In this regard, the proposed algorithm considers that while the BLA is divided and non-linearity is captured the pole-zero locations can change subtly.

Changing locations of poles and zeros of the BLA to improve modelling error implies new estimates around the known values. These locations for both linear subsystems are coded in a single vector as follows:

$$\mathbf{P} = [z_{c_1}, \dots, z_{c_{nc}}, z_{r_1}, \dots, z_{r_{nr}}, z_{i_1}, \dots, z_{i_{nc}}, \dots, p_{c_1}, \dots, p_{c_{mc}}, p_{r_1}, \dots, p_{r_{mr}}, p_{i_1}, \dots, p_{i_{mc}}] \quad (2.14)$$

where $z_{r_1}, \dots, z_{r_{nr}}$ and $p_{r_1}, \dots, p_{r_{mr}}$ contains the locations of the real zeros and poles, respectively; $z_{c_1}, \dots, z_{c_{nc}}$ and $z_{i_1}, \dots, z_{i_{nc}}$ contains the real and imaginary parts of complex conjugate zeros, respectively; while $p_{c_1}, \dots, p_{c_{mc}}$ and $p_{i_1}, \dots, p_{i_{mc}}$ contains the real and imaginary parts of complex conjugate poles, respectively. The values of nc , nr , mc and mr depend on the number of zeros and poles (real and/or complex conjugates) of the BLA.

Poles and zeros contained in 2.14 must be classified to obtain the dynamics of the front and back blocks of a Wiener-Hammerstein model. This classification is performed using a binary vector:

$$\mathbf{C} = [xz_1, \dots, xz_{nc+nr}, xp_1, \dots, xp_{mc+mr}] \quad (2.15)$$

The first part of the vector, $\mathbf{C} (xz_1, \dots, xz_{nc+nr})$, is associated with $z_{c_1}, \dots, z_{c_{nc}}$, $z_{r_1}, \dots, z_{r_{nr}}$ and indicates the zeros classification, while its second part, $\mathbf{C} (xp_1, \dots, xp_{mc+mr})$, is associated with $p_{c_1}, \dots, p_{c_{mc}}$, $p_{r_1}, \dots, p_{r_{mr}}$ indicating the poles classification. Note that imaginary parts are not considered for classification since they are already associated with their corresponding real parts. It is assumed that if $xz_{i-th} = 1$, the corresponding $i - th$ element of \mathbf{P} with $i = 1, \dots, nc + nr$ (i.e., a real zero or a pair of complex conjugated zeros) will belong to the subsystem $G_w(z)$, otherwise it will belong to the subsystem $G_h(z)$. In the same way this correspondence can be applied to classify the poles using xp_1, \dots, xp_{mc+mr} .

For example, if a nonlinear system is approximated by a BLA with four poles: $p_{1,2} = -0.32 \pm 0.77i$, $p_3 = -0.11$, $p_4 = 0.17$ and three zeros: $z_{1,2} = 1.41 \pm 0.56i$, $z_3 = 1.1$, then $nc = 1$, $nr = 1$, $mc = 1$ and $mr = 2$, \mathbf{P} would be structured as: $[1.41, 1.1, 0.56, -0.32, -0.11, 0.17, 0.77]$ and vector \mathbf{C} should contain five elements whose values switch between zero and one as the algorithm evolves. By way of illustration if $\mathbf{C} = [1, 0, 0, 1, 1]$, then $G_w(z)$ would have two zeros and two poles: $z_{1,2} = 1.41 \pm 0.56i$, $p_3 = -0.11$, $p_4 = 0.17$, while $G_h(z)$ would have a zero and two poles: $z_3 = 1.1$, $p_{1,2} = -0.32 \pm 0.77i$.

The location of the breakpoints used to capture static non-linearity is coded as follows:

$$\mathbf{B} = [v_1, \dots, v_n, w_1, \dots, w_n] \quad (2.16)$$

where the pair (v_i, w_i) defines the location of a breakpoint and n is the number of user-defined breakpoints to represent the static nonlinearity.

WH-EA is based on stochastic population of candidate solutions (individuals). Each individual contains genetic information related to:

- The pole/zero locations in the Z-plane of the linear subsystems (\mathbf{P})
- The breakpoint coordinates representing the nonlinear static function (\mathbf{B})
- The pole/zero classification for blocks $G_w(z)$ and $G_h(z)$ (\mathbf{C})

such that any Wiener-Hammerstein model can be easily described from this coded information. Recall that gains from linear blocks are assumed to be 1 and that parameters n_a, n_b, n_c, n_d will be implicitly optimised and they will depend on the structure of vector \mathbf{C} .

As generations go by, the genetic information of each individual can change depending on the crossover and mutation operations, as well as the selection process. Each gene in an individual has a search space defined for a minimum and a maximum value. The new locations for poles and zeros are explored within a search space bounded by the user-defined vectors \mathbf{P}^{min} and \mathbf{P}^{max} . Each element of vector \mathbf{P} corresponds to one element of vector \mathbf{P}^{min} and one element of vector \mathbf{P}^{max} . Therefore vectors \mathbf{P} , \mathbf{P}^{min} , and \mathbf{P}^{max} have the same size. Elements of vectors \mathbf{P}^{min} and \mathbf{P}^{max} are defined around the location of each pole or each zero of the BLA. For example, if the BLA had two real poles, one at 0.70 and the other at 0.87, the search space for exploring

new locations for these poles could be defined with $\mathbf{P}^{min} = [0.6, 0.82]$ and $\mathbf{P}^{max} = [0.8, 0.92]$. These bounds imply that the algorithm will try to tune the pole at 0.7 by exploring new locations between 0.6 and 0.8. On the other hand, the pole at 0.87 could be refined by exploring new locations between 0.82 and 0.92. There is no recipe to define the vectors \mathbf{P}^{min} , and \mathbf{P}^{max} , however it should be noted that a very large search space would allow a good exploration but the algorithm would converge more slowly. On the other hand, a very small search space could lead to ineffective exploration for the refinement of the poles and zeros. It should also be taken into account that in discrete time the poles cannot be outside the unitary circle and that small changes in the positions of the poles and zeros in discrete time imply large changes in continuous time due to the exponential relationship between these two representations.

The genetic information contained in \mathbf{C} determines the distribution of the poles and zeros between the two LTI subsystems. This information is generated randomly and can change from generation to generation, but taking into account the following considerations:

- The number of poles assigned to a subsystem must always be greater or equal than the number of zeros assigned to the same subsystem, i.e., the resulting system cannot be improper.
- The sum of the poles distributed between both subsystems must be equal to the number of poles of the BLA.
- The sum of the zeros distributed between both subsystems must be equal to the number of zeros of the BLA.

As the dynamics is distributed between the two LTI subsystems and the locations of the poles and zeros are refined, static nonlinearity must be captured. For this, the genetic information of individuals contained in \mathbf{B} must be changed allowing the exploration of new locations for breakpoints. To facilitate the convergence of the algorithm this exploration must be done within a suitable search space. In a Wiener-Hammerstein system, 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, and therefore, from this information it is possible to define its search space.

The search space for static nonlinearity can be defined horizontally by the minimum (v_{min}) and maximum (v_{max}) amplitude of $v(t)$ (the output of $G_w(z)$) and vertically by the minimum (w_{min}) and maximum (w_{max}) amplitude of $w(t)$ (the input to $G_h(z)$). Both $v(t)$ and $w(t)$ are unknown signals, however their

minimum and maximum values can be obtained from the input ($u(t)$) and output ($y(t)$) signals and the BLA.

Equations 2.17 and 2.18 show the relationship between minimum and maximum values of signals $u(t)$ and $v(t)$. In these equations Ω is a scaling factor depending on the input LTI 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 and considering that $G_w(s)$ is an LTI subsystem, it can be assumed that there will be no offset between input and output signals, therefore the mean values of $u(t)$ and $v(t)$ will be equal ($v_{mean} = u_{mean}$). The same can be applied to the output block $G_h(s)$, therefore $w_{mean} = y_{mean}$, where w_{mean} and y_{mean} are the mean values of $w(t)$ and $y(t)$, respectively.

$$v_{min} = \Omega * u_{min} \tag{2.17}$$

$$v_{max} = \Omega * u_{max} \tag{2.18}$$

Search space for nonlinearity is vertically delimited by w_{min} and w_{max} . The BLA 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_{mean}, w_{mean}) and the extreme points (v_{min}, w_{min}) and (v_{max}, w_{max}) . Therefore, w_{min} and w_{max} can be found using (2.19) and (2.20).¹ The search space for the static nonlinearity of a Wiener-Hammerstein model is illustrated in Figure 2.2.

$$w_{min} = y_{mean} + K_{NL}(\Omega u_{min} - u_{mean}) \tag{2.19}$$

$$w_{max} = y_{mean} + K_{NL}(\Omega u_{max} - u_{mean}) \tag{2.20}$$

As can be seen from 2.17 to 2.18, the search space for static nonlinearity depends on the minimum and maximum values of the input and output signals, the static gain of the BLA and Ω which is a user-defined parameter. Since both linear subsystems will be estimated with unit gain, neither of these will amplify their input signals, therefore $v_{min} > u_{min}$ and $v_{max} < u_{max}$. For these two conditions to be met Ω must be less than one. In the same way it must be observed that $w_{min} < y_{min}$ and $w_{max} > y_{max}$. If Ω is less than one, the first pair of conditions will always be met, however there is no guarantee that the second pair of conditions will be met. Since it is possible to perform this

¹It should be noted that if the linear dynamic model has negative static gain, the search space for static non-linearity would be delimited by the coordinate pair (v_{min}, w_{max}) and (v_{max}, w_{min}) .

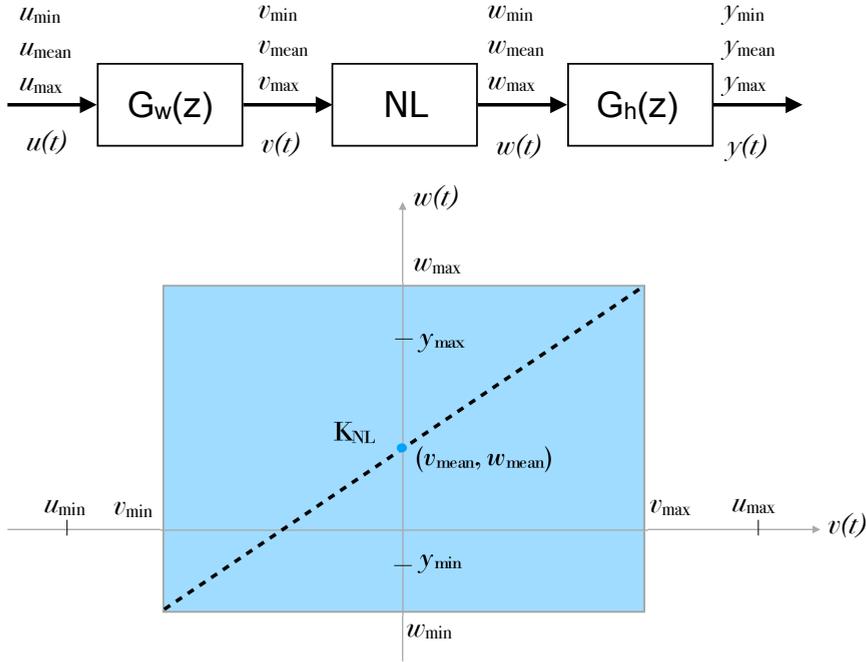


Figure 2.2: Search space for nonlinear static function in a Wiener-Hammerstein model.

check prior to the execution of the algorithm, if the second pair of conditions are not met Ω must be increased but considering that it must be less than one. It should also be taken into account that if the Ω is too large the search space will be larger than necessary, so the algorithm will cost more to estimate static non-linearity.

2.5 Optimisation problem statement

WH-EA makes it possible to find the best parameter set of a Wiener-Hammerstein model, for which an optimisation problem is stated based on a prediction-error method and the typical mean-squared error criterion: ²:

²Although any other criteria can be used in the proposed method, such as the mean absolute or maximum error criteria.

$$\varepsilon(t, \boldsymbol{\theta}) = y(t) - \hat{y}(t, \boldsymbol{\theta}) \quad (2.21)$$

$$J(\boldsymbol{\theta}) = \frac{1}{N} \sum_{t=1}^N \varepsilon^2(t, \boldsymbol{\theta}). \quad (2.22)$$

Where $y(t)$ is the actual measured output and $\hat{y}(t, \boldsymbol{\theta})$ is the estimated model output. According to the genetic coding presented in Subsection 2.4, $\boldsymbol{\theta} = [\mathbf{P}, \mathbf{B}, \mathbf{C}]$. Under this coding the solution of the optimisation problem is stated as:

$$\hat{\boldsymbol{\theta}} = \arg \min_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) \quad (2.23)$$

where $\hat{\boldsymbol{\theta}}$ contains the genetic information from the best individual at the end of generations.

2.6 Genetic operations

Customised mutation and crossover operators will be developed taking in mind the problem at hand: to identify all parameters of the Wiener-Hammerstein model in a single optimisation trial. Figure 2.3 shows the structure of an individual as well as the genetic operators developed on each piece of genetic information. Note that i and g have been introduced into the formulation. Subscript i represents an individual in the population, while the superscript g indicates the current population.

The specific mutation and crossover operators designed are randomly selected to maintain a balance between exploration and exploitation of the search space. Mutation operations are used to maintain genetic diversity, while crossover operations allow genetic information from the best individuals to be combined and disseminated throughout the generations. Further details on how the algorithm works will be given in Section 2.7.

<p>Pole-zero locations</p> <p><i>Zeros</i></p> $P_i^g = \left\{ \begin{array}{ c c c } \hline zc_1, \dots, zc_{nc} & zr_1, \dots, zr_{nr} & zi_1, \dots, zi_{nc} \\ \hline \text{Real values} & & \text{Imag. values} \\ \hline \end{array} \right\} \dots$ <p><i>Poles</i></p> $\dots \left\{ \begin{array}{ c c c } \hline pc_1, \dots, pc_{mc} & pr_1, \dots, pr_{mr} & pi_1, \dots, pi_{mc} \\ \hline \text{Real values} & & \text{Imag. values} \\ \hline \end{array} \right\}$	<p>Mutation M.1</p> <p>Crossover C.1</p>
<p>Static nonlinearity</p> $B_i^g = \left\{ \begin{array}{ c c } \hline v_1, \dots, v_n & w_1, \dots, w_n \\ \hline \text{Abscissa} & \text{Ordinate} \\ \hline \end{array} \right\}$	<p>Mutation M.2</p> <p>Mutation M.3</p> <p>Crossover C.2</p>
<p>Pole-zero classification</p> $C_i^g = \left\{ \begin{array}{ c c } \hline xz_1, \dots, xz_{nc+nr}, & xp_1, \dots, xp_{mc+mr} \\ \hline \text{Zeros } G_w \text{ or } G_h & \text{Poles } G_w \text{ or } G_h \\ \hline \end{array} \right\}$	<p>Mutation M.4</p>

Figure 2.3: Structure of an individual and genetic operations performed on each piece of genetic information.

2.6.1 Location in the Z-plane of poles and zeros.

Both operations used on this portion of genetic information produce an offspring vector \tilde{P}^g , which directly inherits from its parent P_i^g all the genetic information except in a gene. This gene will be selected using a random integer number $r_{zp} \in [1, \dots, nr + nc + mr + mc]$ and modified according to the corresponding genetic operator *Mutation M.1* or *Crossover C.1*.

Mutation M.1. The selected gene is mutated to explore in an individualised way new pole-zero locations of the BLA. A new location \tilde{P}_j^g is determined by a random number N_{zp} with Gaussian distribution:

$$\tilde{P}_j^g = \begin{cases} P_{i,j}^g + N_{zp}(0, \sigma^2(g)) & \text{if } (j = r_{zp}) \\ P_{i,j}^g & \text{otherwise} \end{cases} \quad (2.24)$$

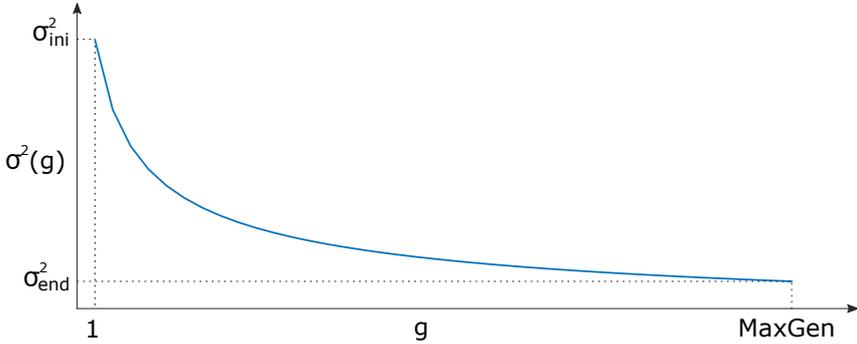


Figure 2.4: Variation of standard deviation over generations to control the aggressiveness of mutation M.1

where $j = 1 \dots nr + nc + mr + mc$. $P_{i,j}^g$ and \tilde{P}_j^g represent the j -th elements of vectors \mathbf{P}_i^g and $\tilde{\mathbf{P}}^g$, respectively.

The new locations for poles and zeros are explored within a search space defined by \mathbf{P}^{min} and \mathbf{P}^{max} , therefore $P_j^{min} \leq \tilde{P}_j^g \leq P_j^{max}$, where P_j^{min} and P_j^{max} are the j -th elements of vectors \mathbf{P}^{min} and \mathbf{P}^{max} , respectively.

Aggressiveness of mutations can be controlled through the standard deviation:

$$\sigma^2(g) = \frac{\Delta_s}{100} \left(\frac{\sigma_{ini}^2}{\sqrt{1 + g * \sigma_{ratio}^2}} \right) \quad (2.25)$$

$$\sigma_{ratio}^2 = \frac{(\sigma_{ini}^2 / \sigma_{end}^2)^2 - 1}{MaxGen - 1} \quad (2.26)$$

where $MaxGen$ is the predefined number of algorithm generations; σ_{ratio}^2 is the rate at which the standard deviation will decrease from σ_{ini}^2 to σ_{end}^2 as the generations pass (see Figure 2.4); Δ_s is a parameter to control the interval over the selected gene will be able to move. For this mutation $\Delta_s = P_{r_{zp}}^{max} - P_{r_{zp}}^{min}$. Variation of $\sigma^2(g)$ will allow mutations to be more subtle in the last generations to achieve a fine-tuning of the corresponding parameters.

Crossover C.1. The selected gene is formed using genetic information from the parent, $P_{i,j}^g$, combined with the corresponding genetic information from the best individual, $P_{best,j}^g$, in the current population:

$$\tilde{P}_j^g = \begin{cases} \frac{P_{i,j}^g + P_{best,j}^g}{2} & \text{if } (j = r_{zp}) \\ P_{i,j}^g & \text{otherwise} \end{cases} \quad (2.27)$$

where $j = 1 \dots nr + nc + mr + mc$.

2.6.2 Nonlinear static function

As the algorithm evolves, points for nonlinear static function must be located adequately in the $v - w$ plane. Here any type of interpolation can be used to capture the static non linearity. To achieve a good fit, mutations M.1 and M.2 plus a crossover operation are used. Both mutations used on this portion of genetic information produce a offspring vector $\tilde{\mathbf{B}}^g$, which directly inherits from its parent \mathbf{B}_i^g all the genetic information except in two genes. This pair of genes represents the coordinates of the point that will be modified. Unlike mutation operations, the crossover operation generates a offspring with a single modified gene which corresponds to the ordinate of a point. Given the correspondence between the abscissa and the ordinate of a point, for the three operations a single integer random number $r_{bp} \in [1, n]$ will allow to select the gene(s) to be modified.

Mutation M.2. This genetic operation allows to explore in the $v - w$ plane new positions for the points. The mutation in both genes is handled by random numbers (N_v, N_w) with Gaussian distribution:

$$\tilde{B}_j^g = \begin{cases} B_{i,j}^g + N_v(0, \sigma^2(g)) & \text{if } (j = r_{bp}) \\ B_{i,j}^g + N_w(0, \sigma^2(g)) & \text{if } (j = n + r_{bp}) \\ B_{i,j}^g & \text{otherwise} \end{cases} \quad (2.28)$$

with $j = 1 \dots 2n$. $B_{i,j}^g$ and \tilde{B}_j^g represent the $j - th$ elements of vectors \mathbf{B}_i^g and $\tilde{\mathbf{B}}^g$, respectively. To avoid overlapping points, bounds for mutations on the abscissa axis are set depending on the selected point to mutate ($\tilde{B}_{r_{bp}}^g$) and the location of its neighbours, according to:

- $v_{min} < \tilde{B}_{r_{bp}}^g < B_{i,r_{bp}+1}^g - \alpha$; if $r_{bp} = 1$
- $B_{i,r_{bp}-1}^g + \alpha < \tilde{B}_{r_{bp}}^g < B_{i,r_{bp}+1}^g - \alpha$; if $r_{bp} = 2 \dots n - 1$

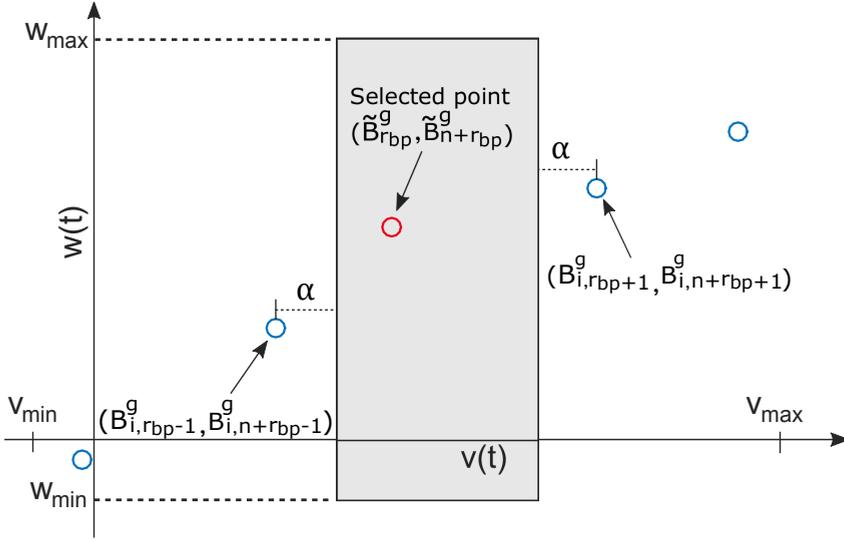


Figure 2.5: Bounds for mutation M.2. Grey area indicates the feasible space for breakpoint defined by $(\tilde{B}_{r_{bp}}^g, \tilde{B}_{n+r_{bp}}^g)$.

- $B_{i,r_{bp}-1}^g + \alpha < \tilde{B}_{r_{bp}}^g < v_{max}$; if $r_{bp} = n$

where α is a user-defined parameter that indicates how close the points can be located. To achieve a good fit of the non linearity α must be small, relative to the search space on the abscissa axis defined by v_{min} and v_{max} . Note that the horizontal boundaries for the endpoints are delimited by v_{min} and v_{max} . These two values define the horizontal size of the search space where static nonlinearity will be captured. Bounds for mutations on ordinate axis are fixed and equal for all points. This allows each point to move freely throughout the search space on the ordinate axis defined by w_{min} and w_{max} . The vertical and horizontal bounds for a selected point are illustrated in Figure 2.5. When mutation M.2 is required, the selected point can be changed to a new position within the grey rectangle.

To achieve a fine-tuning of all breakpoints, mutations' aggressiveness can be controlled through the variable standard deviation $\sigma^2(g)$ (2.25) as in mutation

M.1. Note that $\Delta_s = w_{max} - w_{min}$ is constant for all mutations over ordinate axis while for abscissa axis mutations, Δ_s can be calculated as:

$$\Delta_s = \begin{cases} B_{i,r_{bp}+1}^g - \alpha - v_{min}; & \text{if } (r_{bp} = 1) \\ B_{i,r_{bp}+1}^g - B_{i,r_{bp}-1}^g - 2\alpha; & \text{if } (r_{bp} = 2 \dots n - 1) \\ v_{max} - B_{i,r_{bp}-1}^g - \alpha; & \text{if } (r_{bp} = n) \end{cases} \quad (2.29)$$

Mutation M.2 has great potential to explore the searching space. This genetic operation will locate points where there are slope changes. During the first generations, it is useful to shape non-linearity, while in the last ones, it allows a refinement. However, when a point is located where there is a slope change, it could be kept in this location until the end of generations, especially when there are abrupt changes in the slope. Because jumps between points are not allowed with mutation M.2, when a point is kept in a place where there is a significant change of slope, one or more points would remain trapped to the left or right of it. This would lead to having redundant points in a segment that would not require so many, or worse, to having a segment (curvature) that would not contain enough points. To avoid this drawback, the exploration in the search space is complemented with mutation M.3.

Mutation M.3. This genetic operation is used to concentrate as many points as possible on the curvatures that non linearity can have. Therefore it will be required that each point can be displaced on the abscissa axis by jumping one or more positions of the other points. Let's define a segment as the horizontal space between two consecutive points (so for n points there will be $n - 1$ segments), then a random integer number $r_s \in [1, n - 1]$ will indicate to which segment the selected point will move. The first half of the offspring vector is found using the following expression:

$$\tilde{B}_j^g = \begin{cases} \frac{B_{i,r_s}^g + B_{i,r_s+1}^g}{2} & \text{if } (j = r_{bp}) \\ B_{i,j}^g & \text{otherwise} \end{cases} \quad (2.30)$$

with $j = 1 \dots n$. Note that if $r_{bp} = r_s$ or $r_{bp} = r_s - 1$ the corresponding point will not make a jump but it will be located at the midpoint between its current position and the position of the point on the right or left respectively. As in the mutation M.2 to prevent points from getting too close together the α

parameter is also used in this mutation therefore a jump is conditioned to the space available in the selected segment to accommodate a new point. Minimum space should be 2α . If this condition is not met r_s must be re-generated to randomly search for another segment.

To provide a smooth transition between adjacent segments, gene mutation corresponding to the position on the ordinates axis is performed using a quadratic interpolation. To do that, three neighbouring points are required. The second half of the offspring vector is found using the following expression:

$$\tilde{B}_j^g = \begin{cases} \left[(\tilde{B}_{j-n}^g)^2, \tilde{B}_{j-n}^g, 1 \right] * \begin{bmatrix} k_2 \\ k_1 \\ k_0 \end{bmatrix} & \text{if } (j = n + r_{bp}) \\ B_{i,j}^g & \text{otherwise} \end{cases} \quad (2.31)$$

with $j = n + 1 \dots 2n$. \tilde{B}_{j-n}^g is the v -coordinate of the selected point to mutate which can be found with 2.30, k_0 , k_1 and k_2 are the coefficients of the quadratic polynomial Ψ defined by three adjacent points selected once the new v -coordinate of the point that is mutating is known. The three adjacent points can be selected directly when a point has mutated to the first or last segment,

$$\Psi = \begin{cases} \text{if } (r_s = 1) : \\ f((B_{i,1}^g, B_{i,n+1}^g); (B_{i,2}^g, B_{i,n+2}^g); (B_{i,3}^g, B_{i,n+3}^g)) \\ \text{if } (r_s = n - 1) : \\ f((B_{i,n}^g, B_{i,2n}^g); (B_{i,n-1}^g, B_{i,2n-1}^g); (B_{i,n-2}^g, B_{i,2n-2}^g)) \end{cases} \quad (2.32)$$

whilst if the point has mutated to a non-extreme segment, the three adjacent points can be selected using the two points that define that segment plus one on its right or left. For more effective exploration a random number $r_{3p} \in (0, 1]$ is used for selection:

$$\Psi = \begin{cases} \text{if } (r_{3p} \leq 0.5) : \\ f((B_{i,r_s-1}^g, B_{i,n+r_s-1}^g); (B_{i,r_s}^g, B_{i,n+r_s}^g); (B_{i,r_s+1}^g, B_{i,n+r_s+1}^g)) \\ \text{otherwise} : \\ f((B_{i,r_s}^g, B_{i,n+r_s}^g); (B_{i,r_s+1}^g, B_{i,n+r_s+1}^g); (B_{i,r_s+2}^g, B_{i,n+r_s+2}^g)) \end{cases} \quad (2.33)$$

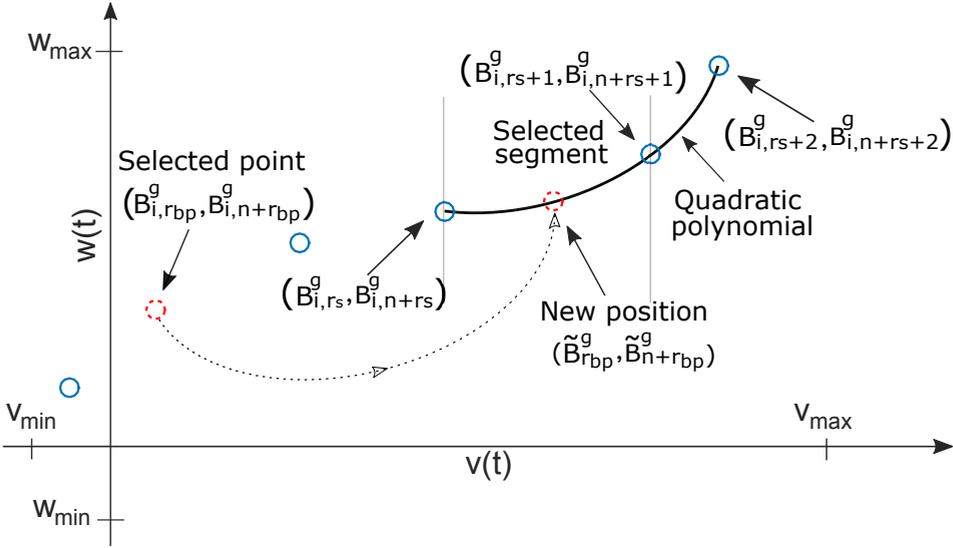


Figure 2.6: Mutation M.3 with $r_{bp} = 2$, $r_s = 4$ and $r_{3p} > 0.5$. Jump to the selected segment (dashed line). Quadratic polynomial (solid line).

Figure 2.6 illustrates how a jump occurs with mutation M.3. Noticed that the new ordinate is calculated according to the polynomial formed by the two points of the segment plus a point to the right, e.i. $r_{3p} > 0.5$. After a jump has occurred, an ascending reordering of the points with respect to the abscissa values is necessary.

Crossover C.2. This genetic operation works just like crossover C.1 and is applied only to vary the position of a point on the ordinate axis:

$$\tilde{B}_j^g = \begin{cases} \frac{B_{i,j}^g + B_{best,j}^g}{2} & \text{if } (j = r_{bp}) \\ B_{i,j}^g & \text{otherwise} \end{cases} \quad (2.34)$$

with $j = n \dots 2n$. $B_{best,j}^g$ is the j -th element of vector \mathbf{B}_{best}^g , which corresponds to the individual in the current population g with the best fitness value.

Pole-zero classification

Due to the stochastic nature of evolutionary algorithms, the binary values of 2.15 will change as the algorithm evolves and generating different structures of $G_w(z)$ and $G_h(z)$. The evolution of this piece of genetic information is handled by a simple mutation operator.

Mutation M.4. Unlike the previous ones, this operator generates a new vector $\tilde{\mathbf{C}}^g$ that depends entirely on the effects of mutation, meaning that for this piece of genetic information there is no information exchange between generations. This allows free testing of different structures for $G_w(z)$ and $G_h(z)$ to avoid premature convergence. When this operation is required a random process will generate the mutation vector:

$$\tilde{C}_j^g = \begin{cases} 1 & \text{if } N_c \leq 0.5 \\ 0 & \text{otherwise} \end{cases} \quad (2.35)$$

with $j = 1 \dots nr + mr$. \tilde{C}_j^g is the j -th element of vector $\tilde{\mathbf{C}}^g$. N_c is a random number with standard uniform distribution on the open interval (0,1). Note that the structure of $\tilde{\mathbf{C}}^g$ is built under two considerations: the LTI subsystems can not be improper and the sum of zeros and the sum the poles between both subsystems must be equal to the number of zeros and poles of the BLA respectively.

2.7 WH-EA description

Algorithm 1 shows a pseudocode of main steps performed in WH-EA. Details of each of these steps are given below:

Initialise the population. The initial population $[\mathbf{P}^0 \ \mathbf{B}^0 \ \mathbf{C}^0]$ contains NP individuals generated from a reference individual. This individual takes the first place in the population and its genetic information is structured as follows:

- The \mathbf{P} portion is directly coded with the location of the poles and zeros of the BLA.
- The \mathbf{B} portion contains the abscissa and ordinate of the n breakpoints assigned to represent the static nonlinearity. The n points are uniformly

Algorithm 1 Pseudocode of WH-EA.

```

1: Initialise the population;
2: Evaluate fitness of all population;
3: for  $g = 1$  to  $MaxGen$  do
4:   Find  $\theta_{best}^g$ 
5:   Random selection of a individual ( $r_1$ );
6:   Compute  $\gamma(g)$ ;
7:   if  $r_{pznl} \leq \gamma(g)$  then
8:     Compute  $\tilde{B}^g$  using Algorithm 2;
9:   else
10:    Compute  $\tilde{P}^g$  using Algorithm 3;
11:   end if
12:   if  $r_c \leq \xi$  then
13:     Compute  $\tilde{C}^g$  using Mutation M.4;
14:   end if
15:   Update population;
16: end for
17: Print  $\theta_{best}^{MaxGen}$ 

```

distributed between v_{min} and v_{max} and located on the diagonal line joining the ends of the search space (see Figure 2.2).

- The C portion is completed with randomly generated binary information but taking into account the considerations indicated in Section 2.4).

The rest of the population is completed by individuals who are a mutated version of the reference individual. For this, the M.1, M.2 and M.4 mutations are applied at the same time to the corresponding portions of the genetic information of the reference individual. For every time these three mutations are applied a new individual will be generated, i.e. this process must be carried out $NP-1$ times to complete the population. Since the genes to be mutated are selected at random and since the mutations are based on genetic modifications that are also random, this process guarantees an initial population that is rich in diversity.

Evaluate fitness. Performance of each individual in the population is defined by a fitness criterion which can be calculated using (2.22), where θ is obtained from the encoded information in $[P_i^g, B_i^g, C_i^g]$.

The offspring. Once population has been initialised, for each generation a random integer number $r_1 \in [1, NP]$ will be used to select the parent from which an offspring \tilde{P}^g will be generated. As can be seen in Algorithm 1, not all genetic operators are applied at the same time to generate a offspring -this can help to expand diversity and avoid premature convergence.

One or two pieces of the offspring genetic information will be randomly selected for modification according to their respective genetic operators. A random number $r_{pznl} \in (0, 1]$ choose between modifying the portion related to static nonlinearity using Algorithm 2 or the portion of genetic information related to pole/zero locations using Algorithm 3. The probability for this selection is handled by the control parameter $\gamma(g)$ defined as:

$$\gamma(g) = \frac{\gamma_{ini}}{\sqrt{1 + g * \gamma_{rat}}} \quad (2.36)$$

$$\gamma_{rat} = \frac{(\gamma_{ini}/\gamma_{end})^2 - 1}{MaxGen - 1} \quad (2.37)$$

where γ_{rat} is the rate at which the probability $\gamma(g)$ will decrease from initial probability γ_{ini} to final probability γ_{end} as generations pass, therefore, $0 < \gamma_{fin} < \gamma_{ini} \leq 1$. If $\gamma_{ini} = 1$ the probability of modifying the genetic information of nonlinearity in the first generations will be high, while the probability of modifying the location of poles and zeros will be low. On the other hand if and $\gamma_{end} = 0.5$, in the final generations the algorithm will modify with equal probability both portions of genetic information. The selection of these values is justified by the fact that pole/zero locations are known and they will only be fine-tuned within a suitable search space to amend possible errors in the BLA estimation, whereas that nonlinearity is completely unknown, so the algorithm should focus more on this portion of genetic information during first generations.

Variation of genetic information corresponding to the classification of poles and zeros for both LTI subsystems is handled by a comparison between a random number $r_c \in (0, 1]$ and the probability $\xi \in (0, 1]$. The value of probability ξ is defined by the user and will be constant throughout the evolution of the algorithm. Figure 2.7 shows the behaviour of the control parameters (probabilities) used to select the portions of genetic information that will be modified in each generation.

Algorithm 2 is used to modify the genetic information related to nonlinear static function. 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. Probability of selecting M.2 or M.3 is variable with respect to the generations. During first generations, mutation M.3 is not necessary, since the nonlinearity can be captured thanks to the two-dimensional points movements due to mutation M.2 and crossover C.2 operations. Since it is very likely

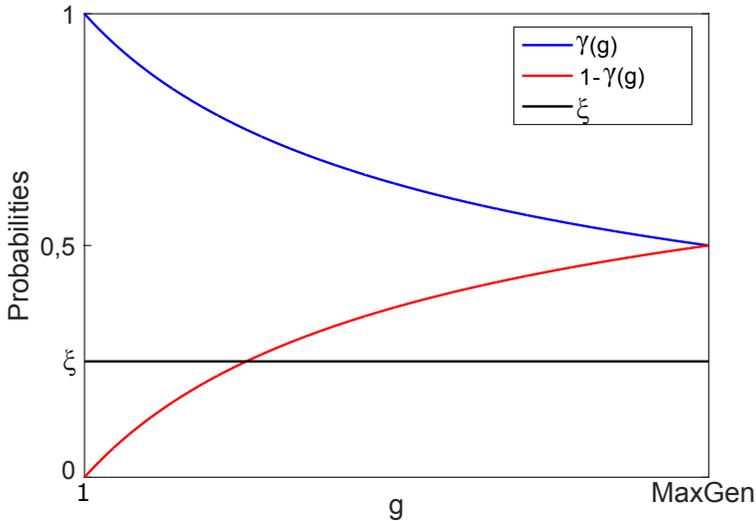


Figure 2.7: Control parameters of WH-EA for selection of the genetic information to be modified in generation g .

that non-linearity includes one or more curvatures, as the algorithm evolves mutation M.3 will be required to concentrate as many points as possible on these curvatures. The variable probability for selection between both mutations is defined by:

$$\eta(g) = (1 - \eta_{min}) - (1 - \eta_{min}) \frac{g}{MaxGen} \quad (2.38)$$

where $\eta_{min} \in (0, 0.5]$ is a user-defined parameter indicating the minimum probability with which the mutation M.2 can be selected. Note that according to (2.38) and Algorithm 2, the maximum probability is one and occurs in the first generation. As the algorithm evolves this probability will decrease linearly until it reaches η_{min} in the last generation. When Algorithm 2 is required a random number $r_{nmc} \in (0, 1]$ will allow to select either a mutation or crossover C.2. If a mutation is selected, a new random number $r_{mm} \in (0, 1]$ will allow to select between mutation M.2 or mutation M.3.

On the other hand Algorithm 3 is used to modify the genetic information related to pole/zero locations using mutation M.1 or crossover C.1. The control parameter $\delta_{zp} \in (0, 1]$ indicates the probability with which each genetic operation will be used. Since crossover C.1 causes offsprings inherit

Algorithm 2 Modify two-dimensional points for nonlinear function.

```
1: if  $r_{nmc} \leq \delta_{nl}$  then  
2:   Compute  $\eta(g)$   
3:   if  $r_{mm} \leq \eta_{min} + \eta(g)$  then  
4:     Mutation M.2;  
5:   else  
6:     Mutation M.3;  
7:   end if  
8: else  
9:   Crossover C.2;  
10: end if
```

Algorithm 3 Modify polo/zero locations.

```
1: if  $r_{lmc} \leq \delta_{zp}$  then  
2:   Mutation M.1;  
3: else  
4:   Crossover C.1;  
5: end if
```

genetic information from the best individual, a small value of δ_{zp} may lead to premature convergence, whereas a value closer to one will cause the algorithm to converge very slowly. When Algorithm 3 is required a random number $r_{lmc} \in (0, 1]$ will determine the genetic operation to be used.

Update. It is based on a competition between the generated offspring and the individuals of the population. The contestant with the best fitness will be the one who wins the competition. From a randomly selected individual, the offspring starts to compete until defeating an individual, when this happens the descendant will take his place in the population and the algorithm continues with the next generation. If the offspring comes to compete with all individuals and could not win, this will be discarded and the algorithm will pass to the next generation.

2.8 Application of WH-EA and results.

WH-EA was tested on a numerical example and on the benchmark for nonlinear system identification in (SYSID'09) [95], where a Wiener-Hammerstein system is selected as test object. The benchmark is not intended as a competition, but as a tool to compare the possibilities of different methods to deal with this specific nonlinear structure.

For both cases the BLA was estimated with the Matlab System Identification Toolbox [63] using a Box-Jenkins (BJ) structure. Besides, trends and means were only removed for the BLA identification. The following parameters of the algorithm were set in common for both estimates: $\xi = 0.25$; $\delta_{zp} = 0.75$; $\delta_{nl} = 0.75$; $\eta_{min} = 0.35$, in addition, initial and final standard deviations for mutations were set to 20 and 1 respectively.

2.8.1 Numerical example

A Wiener-Hammerstein system with the following structure was designed³:

$$G'_w(z) = \frac{0.1190}{(z - 0.9048)} \quad (2.39)$$

$$w'(t) = 0.45 \text{tansig}(2.80v'(t)) \quad (2.40)$$

$$G'_h(z) = -0.01426 \frac{(z - 1.0510)(z + 1)}{(z - 0.9746 + 0.03656i)(z - 0.9746 - 0.03656i)} \quad (2.41)$$

A Gaussian excitation signal of 6 dB was filtered with a cut-off frequency of 6 Hz and used as input signal. The system was simulated and 120000 input/output samples were recorded and separated in two parts: the estimation data set $t_n \in [1001, 70000]$ for identification purposes and the test data set $t_n \in [71001, 120000]$ for validation purposes (in both data sets first 1000 samples were ignored to avoid transient effects). Furthermore additive white Gaussian noise with a Signal-to-Noise Ratio (SNR) of 45.32 dB was added to the output.

The identification of the BLA was carried out and the model obtained was expressed in factored form:

$$G_{BLA} = -1e^{-3} \frac{(z - 1.0508)(z + 0.9631)}{(z - 0.9749 + 0.0366i)(z - 0.9749 - 0.0366i)(z - 0.9045)} \quad (2.42)$$

The Root Mean Square of the error ($eRMS$) obtained with this linear model on test data was of 0.0414.

³where tansig is the hyperbolic tangent sigmoid transfer function.

According to the BLA structure, vector \mathbf{P}_1^0 was coded with $nc = 0$, $nr = 2$, $mc = 1$ and $mr = 1$ as follows:

$$\mathbf{P}_1^0 = [1.0508, -0.9631, 0.9749, 0.9045, 0.0366] \quad (2.43)$$

The search space for nonlinear function was defined with $\Omega = 0.28$, while \mathbf{P}^{min} and \mathbf{P}^{max} were defined as follows:

$$\mathbf{P}^{min} = [1.0408, -1.0631, 0.9649, 0.8945, 0.0266] \quad (2.44)$$

$$\mathbf{P}^{max} = [1.0608, -0.8631, 0.9849, 0.9145, 0.0466] \quad (2.45)$$

According to \mathbf{P}^{min} and \mathbf{P}^{max} the zero at -0.9631 can be refined considering that its location can change ± 0.1 from its own value, while the zero at 1.0508 and all the poles can be refined considering bounds of ± 0.01 from its own value. The limits of the zero at -0.9631 were defined larger since it influences slightly the dynamics and should have freedom of movement during tuning.

WH-EA was executed 3 times with $MaxGen = 5.0E6$ and different number of points was chosen for the nonlinearity. For all trials, the algorithm was initialised with $6.0E4$ individuals and the minimum distance between two points was set to $\alpha = (v_{max} - v_{min})/6n$. For each estimated Wiener-Hammerstein model the $eRMS$ on the test data was computed, in addition, the Normalised Root Mean Square Error ($NRMSE$) criterion was used to quantify the goodness of fit between real and captured non linearity. The results are reported in Table 2.1, for all cases piecewise linear interpolation was used to connect the n breakpoints.

Table 2.1: Performance of the numerical example estimation using different numbers of breakpoints (n) to represent static nonlinearity.

n	NRMSE (%)	eRMS
8	99.179	$1.183E - 3$
10	99.396	$1.109E - 3$
12	99.564	$1.044E - 3$

The poles and zeros of the BLA were correctly classified in the three tests carried out. As can be seen in Table 2.1, the $eRMS$ of the Wiener-Hammerstein models decreased as the quality of the captured nonlinearity increases. A reasonable model was scored with 12 points

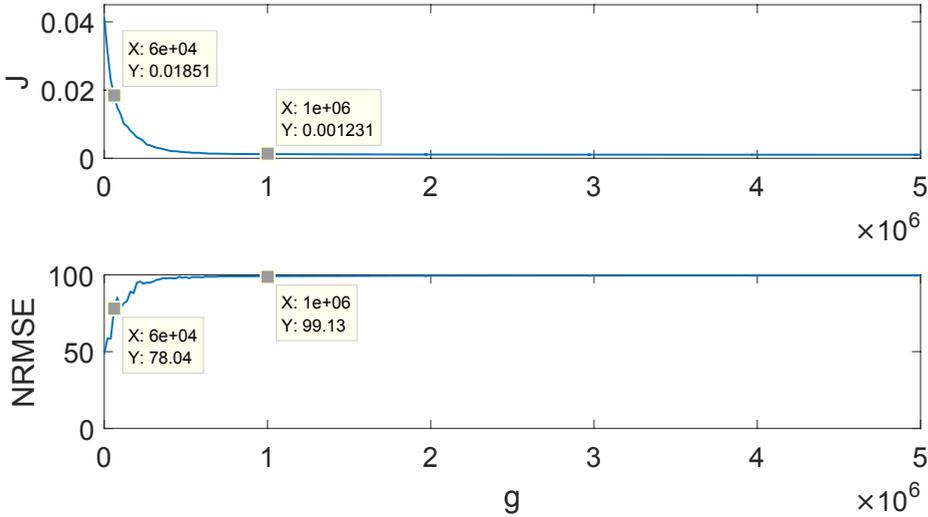


Figure 2.8: Convergence graph (top) and NRMSE of the captured non linearity (bottom) for Wiener-Hammerstein estimation with $n=12$.

considering that the RMS noise was $9.98E - 4$. A convergence graph for this model is shown in Figure 2.8, during WH-EA execution, at $6e4th$ generation the poles and zeros of the BLA were correctly classified and from there the best individual of each generation conserved the genetic information for this classification. Since the noise RMS is known, at $g = 1.0E6$ the performance of the model was good enough, so the algorithm could have been stopped. Anyway, $5.0E6$ generations have been allowed in order to demonstrate the great precision that the algorithm can achieve.

In 2.9 pole/zero locations of BLA model, obtained Wiener-Hammerstein model and real system are compared. Notice how WH-EA has moved initial locations trying to get to the true values improving modelling error.

On the other hand, a graphical comparison between real and captured non linearity is shown in 2.10. Linear subsystems of the estimated Wiener-Hammerstein model are represented by (2.46) and (2.47), while the ordered pairs for the nonlinear static function are shown in 2.2.

$$G_w(z) = \frac{0.0259}{z - 0.9048} \quad (2.46)$$

Table 2.2: Coordinates of the estimated nonlinearity with $n = 12$.

i	1	2	3	4	5	6
v_i	-0.9694	-0.5794	-0.3716	-0.2479	-0.1251	0.1038
w_i	-0.3303	-0.3250	-0.2894	-0.2354	-0.1415	0.1178
i	7	8	9	10	11	12
v_i	0.1996	0.2893	0.3363	0.4739	0.7648	1.0808
w_i	0.2031	0.2558	0.2754	0.3126	0.3309	0.3317

$$G_h(z) = -0.01960 \frac{(z - 1.0512)(z + 0.9713)}{(z - 0.9746 + 0.0365i)(z - 0.9746 - 0.0365i)} \quad (2.47)$$

2.8.2 Nonlinear system identification benchmark

The system to be modelled is an electronic nonlinear circuit with a Wiener-Hammerstein structure (see 2.11). This system was built by [111] and presented as a benchmark problem for system identification by [95].

The first linear dynamic system $G_1(s)$ is designed as a third order Chebyshev filter (pass-band ripple of 0.5 dB and cut off frequency of 4.4 kHz). The second linear dynamic system $G_2(s)$ is a third order inverse Chebyshev filter (stop-band attenuation of 40 dB starting at 5 kHz). This system has a transmission zero in the frequency band of interest. This can complicate the identification significantly, because the inversion of such a characteristic is difficult. The system was excited with a filtered Gaussian signal (cut-off frequency 10 kHz). Data used for estimation corresponds to interval $t_n \in [1, 100000]$ whereas test data corresponds to the remaining part $t_n \in [101001, 188000]$. In order to analyse the performance of estimation methods the mean value of the simulation error (μ), the standard deviation of the error (std), and the root mean square value of the error ($eRMS$) must be calculated on test and estimation data [95].

Since first 5000 data samples just contain quantization noise, a set of 95000 input/output data $t_n = 5001, \dots, 100000$ was used to estimate the BLA. Multiple simulations were performed considering different combinations of poles and zeros for the input/output model and for the noise model. For each

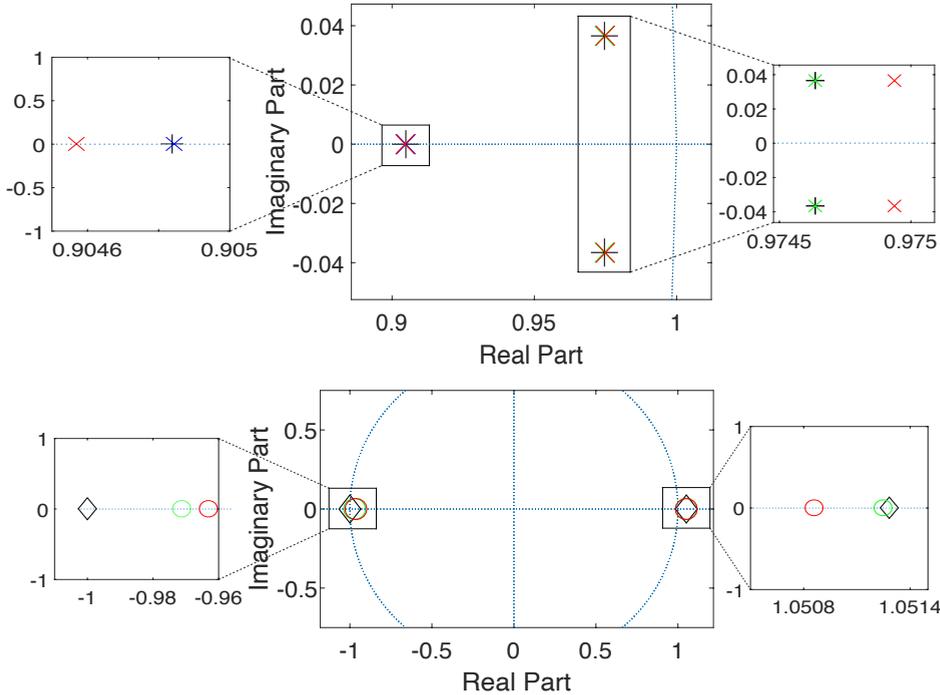


Figure 2.9: Top: poles of the real system (black +), the BLA (red x) and the estimated G_w (blue x) and G_h (green x) models. Bottom: zeros of the real system (black \diamond), the BLA (red o) and the estimated G_h model (green o).

BJ model the $eRMS$ on test data set $t_n = 101001, \dots, 188000$ was computed. The BLA was obtained with 6 poles, 5 zeros and one sample delay for the input/output model and 3 zeros and 3 poles for the noise model. The BLA is fully described with $K_{NL} = 0.7840$ and the pole-zero pattern is shown in Figure 2.12. The $eRMS$ of this linear model was of $56.159mV$ on test data and $43.143mV$ after removing trends and means. According to the BLA structure, vector \mathbf{P}_1^0 was coded with $nc = 1$, $nr = 4$, $mc = 2$ and $mr = 2$ as follows:

$$\mathbf{P}_1^0 = [0.7605, -0.2733, 0, -3.4122, -30.2553, 0.6501, 0.7314, \dots, \dots, 0.8912, 0.8289, 0.7004, 0.4358, 0.1692]. \quad (2.48)$$

During BLA estimation stage, different noise models were tested and it was observed that all poles, real zeros within the unitary circle, and complex zeros

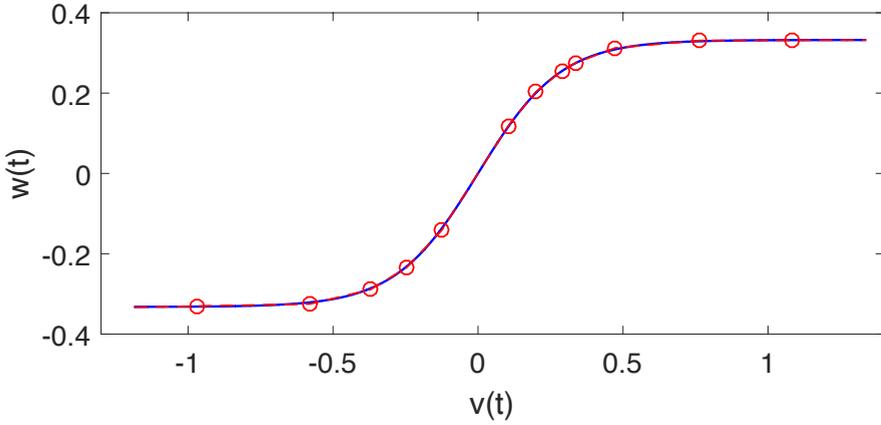


Figure 2.10: Comparison between true (solid-blue) and estimated nonlinearity defined as a piecewise linear function (dashed-red) using pairs $[v_i, w_i]$ (red circles).

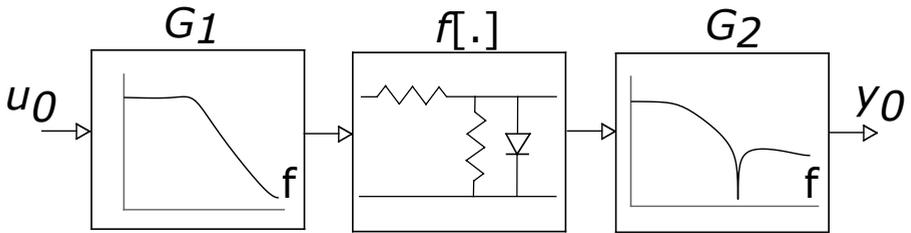


Figure 2.11: Wiener-Hammerstein benchmark.

as they are located correspond to the dominant dynamics of the system, while real zeros outside the unitary circle were more likely to vary their location. This information was used to define the search space for refining the location of poles and zeros:

$$\begin{aligned}
 \mathbf{P}^{min} = & [0.7355, -0.2983, -0.025, -4.4122, -40.2553, 0.6251, 0.7064, \dots \\
 & \dots, 0.8662, 0.8039, 0.6754, 0.4108, 0.1442].
 \end{aligned}
 \tag{2.49}$$

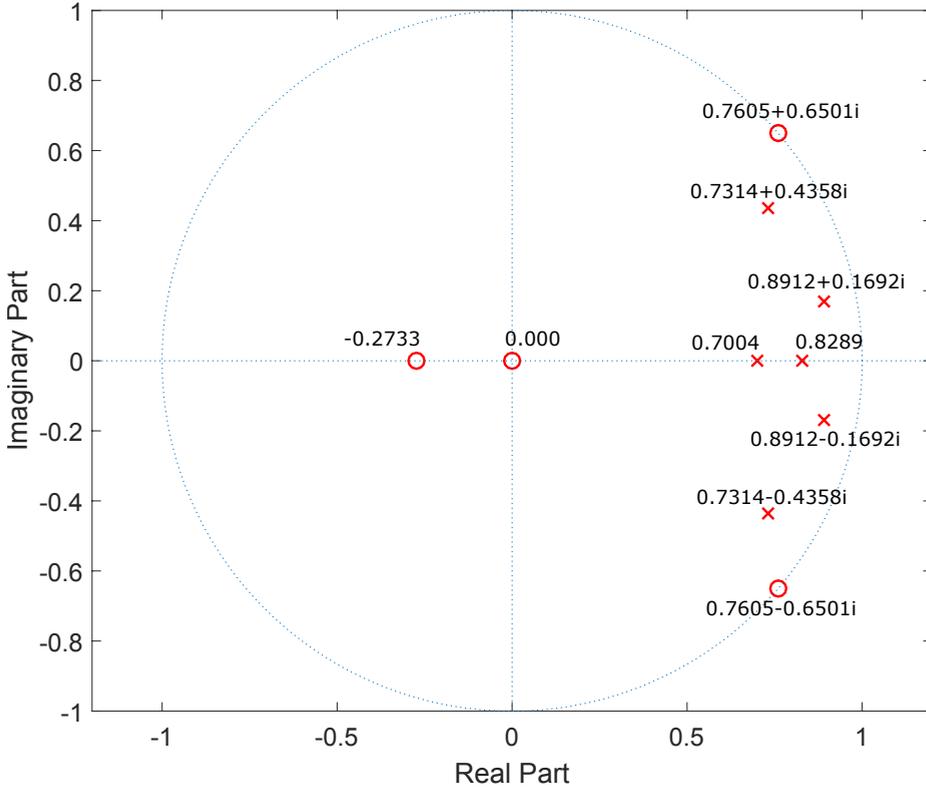


Figure 2.12: Poles(x) and zeros(o) of the BLA for the benchmark data (SYSID'09). Two real zeros fall outside the plot in -30.2553 and -3.4122 .

$$\begin{aligned}
 P^{max} = & [0.7855, -0.2483, 0.025, -2.4122, -20.2553, 0.6751, 0.7564, \dots \\
 & \dots, 0.9162, 0.8539, 0.7254, 0.4608, 0.1942].
 \end{aligned}
 \tag{2.50}$$

Bounds (2.49) and (2.50) limit search space in the system dominant dynamics between ± 0.025 , while for $z = -30.255$ and $z = -3.412$, limits are between ± 10 and ± 1 respectively.

Eight points were assigned to represent static nonlinearity ($n = 8$). The search space for this static nonlinearity was defined with $\Omega = 0.51$, while the minimum distance between two points was calculated with $\alpha = (v_{max} - v_{min})/10n$.

Table 2.3: Performance indicators of the estimated Wiener-Hammerstein model. All values are shown in mV.

	BLA		Wiener-Hammerstein	
	Estimation	Test	Estimation	Test
μ	-35.825	-35.951	$5.60E - 4$	$1.10E - 5$
std	42.108	43.143	$3.22E - 1$	$3.06E - 1$
$eRMS$	55.286	56.159	$3.22E - 1$	$3.06E - 1$

The algorithm was initialised with $5E3$ individuals and $3E7$ generations were executed. The performance of the estimated Wiener-Hammerstein model is shown in Table 2.3. In Figure 2.13 is depicted how the algorithm has distributed pole/zero locations for both linear subsystems. Notice how some of them were displaced to improve the modelling error. The output of the estimated Wiener-Hammerstein model, the linear model error and the nonlinear model error on test data are shown in Figure 2.14, while the DFT spectra of this signals is shown in Figure 2.15.

Final estimated linear blocks $G_w(z)$ and $G_h(z)$ are shown in (2.51) and (2.52) respectively, while coordinates for nonlinear function are shown in Table 2.4. The estimated Wiener-Hammerstein model contains 26 parameters of which 14 are used to represent the static non linearity (without end points since they can be located anywhere on their respective end segments, nevertheless, these segments slopes are taken into account). Figure 2.16 shows how mutation M.2 and mutation M.3 located the two-dimensional points to capture the nonlinearity. As expected, due to the effect of mutation M.3, most of them were concentrated on the curvature.

$$G_w(z) = 6.5e^{-4} \frac{(z + 0.0138)(z + 2.9034)(z + 26.76)}{(z - 0.7243)(z - 0.7324 + 0.4361i)(z - 0.7324 - 0.4361i)} \quad (2.51)$$

$$G_h(z) = 0.0120 \frac{(z + 0.2635)(z - 0.7575 + 0.6513i)(z - 0.7575 - 0.6513i)}{(z - 0.8191)(z - 0.8899 + 0.1688i)(z - 0.8899 - 0.1688i)} \quad (2.52)$$

In contrast to other methods which generate good initial estimates by splitting the poles and zeros of the BLA, WH-EA allows to identify Wiener-Hammerstein models avoiding high user interaction which is

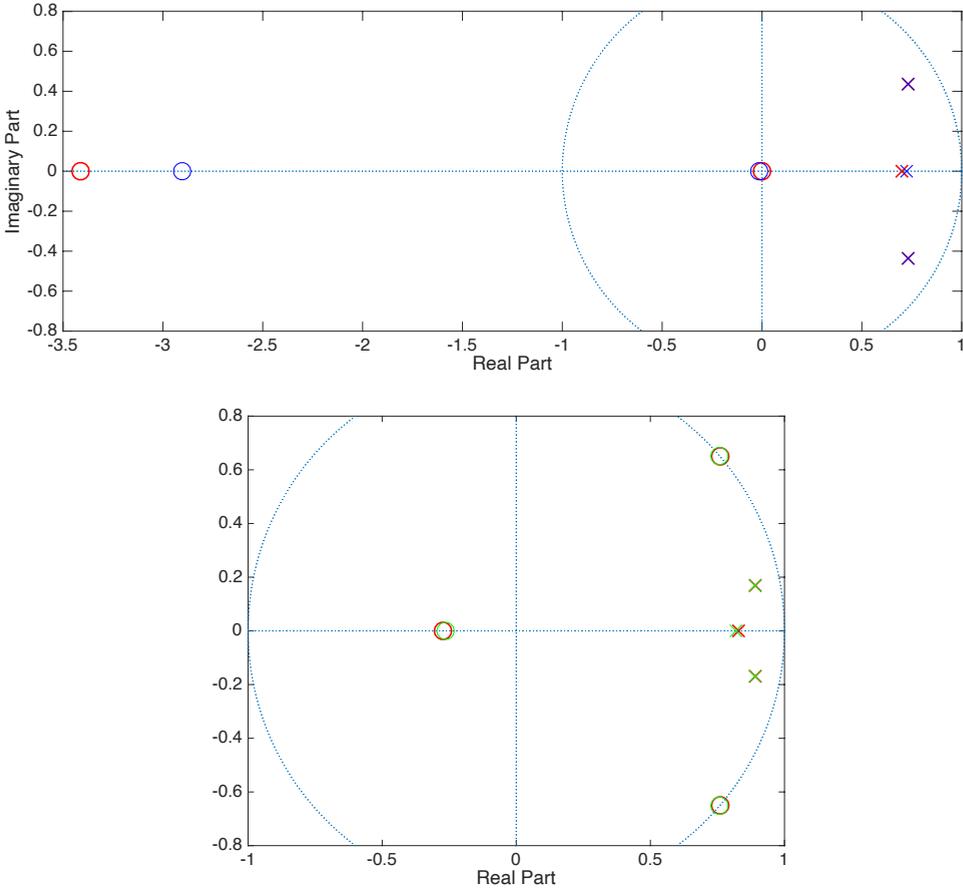


Figure 2.13: Top: poles (x) and zeros (o) of the linear subsystem G_w (blue) and BLA (red). Zero in $z = -30.2553$ belonging to BLA and its adjusted final value in $z = -26.7609$ fall outside the plot. Bottom: poles(x) and zeros (o) of the linear subsystem G_h (green) and BLA (red).

an advantage compared to methods using QBLA, where at least two intermediate procedures are required before fine-tuning all parameters of the Wiener-Hammerstein model.

The $eRMS$ of $0.306mV$ achieved with WH-EA on test data is quite acceptable considering that the RMS of the quantization noise is $0.189mV$. Respect to the initial model (BLA), the error was reduced by a factor of 183.52 thanks to the captured non linearity and the updated pole/zero locations.

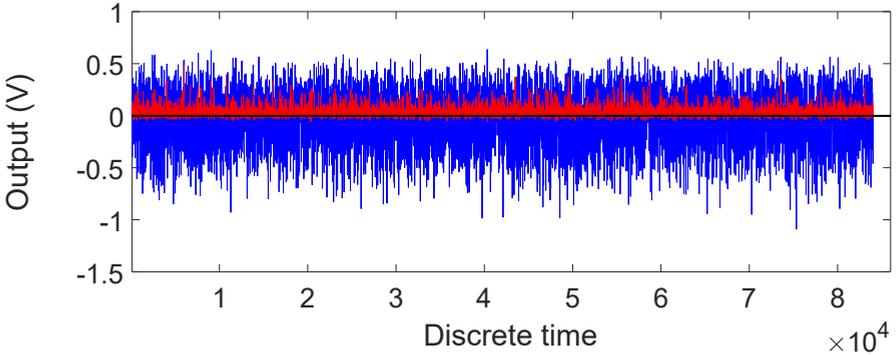


Figure 2.14: Model output (blue), simulation error of the BLA (red) and simulation error of the estimated Wiener-Hammerstein model (black).

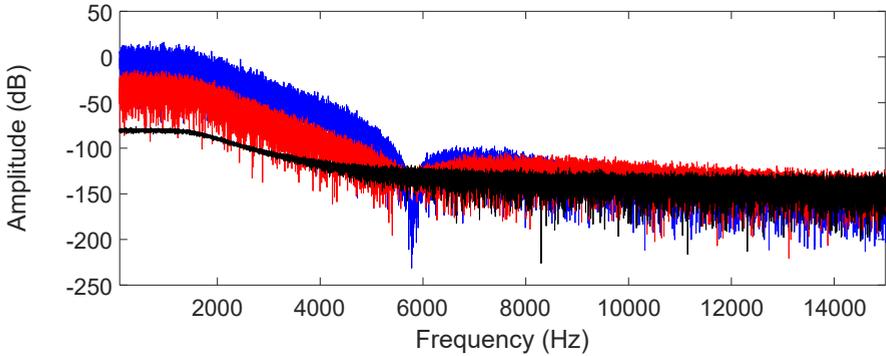


Figure 2.15: DFT spectra of the modelled output signal (blue), linear model error (red) and nonlinear model error (black).

Table 2.5 shows other proposals that have been tested on the benchmark. It can be appreciated that the *eRMS* of WH-EA is slightly higher than others, however not all estimated models have the same complexity. Some of them use complex models with a greater number of parameters processing raw data before identification whilst in this work, WH-EA is fed raw input/output data without pre-processing operations.

Compared WH-EA with the proposals of [113] and [110] whose models have the same complexity as the model estimated with WH-EA, the results are quite similar, however to obtain a good final model with these two proposals it is required that the BLA be estimated with high precision. In [110] at the BLA

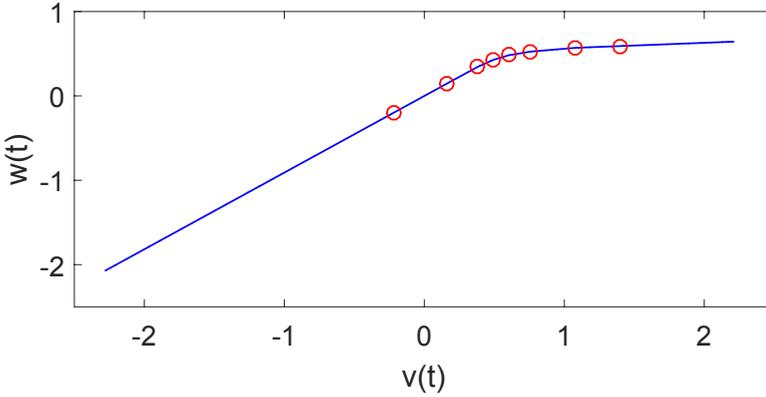


Figure 2.16: Captured non linearity as a piece-wise linear function with $n = 8$ by WH-EA from the benchmark data. Notice that the non linear block characterisation only needs 14 parameters since the first and last straight segments can be defined just with their angles.

Table 2.4: Non linearity coordinates ($n = 8$, 14 parameters) estimated by WH-EA from benchmark data.

i	1	2	3	4
v_i	-0.2168	0.1596	0.3819	0.4943
w_i	-0.1979	0.1440	0.3443	0.4276
i	5	6	7	8
v_i	0.6047	0.7596	1.0811	1.3953
w_i	0.4822	0.5248	0.5698	0.5901

division phase a false position of a pole or zero could cause the values of the fractional powers to be close to $1/2$ which would cause the user to make a bad decision and the BLA is badly divided. This problem is much more critical in [113] since the method is based on a graphical comparison between the poles and zeros of the BLA and the poles and zeros of the QBLA. With WH-EA this problem does not occur, since the evolutionary algorithm contemplates possible errors that can be made in the estimation of the BLA. During the algorithm evolution the binary code used for the classification of the poles and zeros of the BLA can be changed without user interaction as the false positions of the poles and zeros are corrected. This is an important advantage of WH-EA

Table 2.5: Performance measurements on benchmark data (SYSID'09). All the values are shown in mV. θ indicates the number of parameters used for the model.

Method/Technique	e RMS (mV)	θ
Nonparametric BLA, QBLA. ([99])	0.278	44
Classification of poles and zeros using QBLA. ([113])	0.286	26
Fractional model parameterization. ([110])	0.295	26
Advanced method. ([56, 102])	0.30	64
WH-EA.	0.306	26
Brute force method. ([102])	0.31	30
Scanning technique ([114])	0.370	-
Polynomial nonlinear state space. ([81])	0.42	797
Generalised Hammerstein-Wiener. ([118])	0.481	47
Incremental nonlinear optimisation. ([106])	0.679	25
LS-SVMs. ([26])	4.070	-
Biosocial culture. ([74])	8.546	34

since it is very likely that the BLA estimate is subject to errors due to noise and non-linearity effects. This has been experimentally demonstrated, for this reason many proposals carry out a final readjustment of the parameters of the Wiener-Hammerstein model.

Chapter 3

This chapter presents a unified approach for the identification of Wiener, Hammerstein, and Wiener–Hammerstein models. This approach is based on the use of multi-step excitation signals and WH-EA. The chapter begins with a brief introduction to the approach. The second section then presents the formulation of the problem of identification in continuous time. Since the BLA will not be used, the following section explains in detail the initial linear model used in this approach. Then, in the fourth section the optimisation problem is stated. The fifth section explains the genetic coding used by the algorithm. Sections six and seven then explain how to set up the search space for the dynamic and static parts, respectively. Section seven also explains how to achieve a common search space to identify models of Wiener, Hammerstein and Wiener-Hammerstein. This common search area is the key to the unified approach. Next, in section eight, several aspects related to multistep inputs, such as excitation signals, are presented. To end, in section nine, the presented methodology is applied to three numerical examples and a real application (a thermal process).

3.1 Introduction

In Chapter 2 an algorithm for identifying Wiener-Hammerstein models was introduced. As presented, this algorithm requires the user to define a search space where the static nonlinearity will be captured. This search space was defined considering that the nonlinear block is in the middle of two LTI subsystems, i.e., the search space restricted the formulation to a Wiener-Hammerstein model.

When dealing with processes that have a static nonlinearity, it is highly likely that the location of this nonlinearity around the dynamics will not be known, that is, there will be an uncertainty about whether the process to be estimated has the structure of Wiener, Hammerstein, or Wiener-Hammerstein. Given this uncertainty, the user could use three different estimation methods, one for each structure, and at the end compare the performance of the models obtained to select the best process representation.

This chapter describes how WH-EA can be used 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.

Both for Wiener, Hammerstein and Wiener-Hammerstein models, a search space for static nonlinearity can be defined using information from the input and output data set 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 data-set, a common search space useful for the three structures will be defined. As it will be seen in Subsection 3.7, 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, multi-step signals will be used. In addition, multi-step excitation will enable an effective exploration of different process operation zones highlighting existing non-linearities (not possible if Gaussian signals are used to excite the process). Note that Gaussian signals are more

useful for estimating process dynamics and not necessarily allow to emerge process non-linearities, especially when they are present at the extremes of its range of operation, as in the case of saturation for example.

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 of static nonlinearity (which is also captured simultaneously).

The proposal presented in this chapter is useful to identify Wiener, Hammerstein and Wiener-Hammerstein models without any beforehand information about the type of structure. This is highly attractive to identify dynamic processes where is known they are affected by an static nonlinearity but there is uncertainty about its location around the dynamics. From the core idea of this unified approach, other derived novelties are highlighted below:

- 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.
- This approach uses multi-step excitation signals. This type of signals allow non-linearities to emerge better. This is very useful since nonlinear estimation starts from dynamics already known.
- Thanks to the normalisation of the dynamics and the use of multi-step 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 that it is necessary to define it when using Gaussian type signals.
- The estimation is done in continuous-time, which gives the user a clearer view of the process behaviour under test. On the other hand, in continuous-time it is easier to differentiate between dominant and non-dominant dynamics. This can be of great help to define the search space for exploring new locations for poles and zeros.

3.2 Continuous-time formulation for the unified approach

All the three block-oriented models addressed with this approach 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 non-linear 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 approach it is assumed that the process to be identified is affected by a static nonlinearity, however the block-model structure is not precisely known, so the most general form is considered as a starting point for the problem formulation. Let's represent Wiener, Hammerstein, and Wiener-Hammerstein models by:

$$y(t, \boldsymbol{\rho}) = G_h(s, \boldsymbol{\rho}_h) f(G_w(s, \boldsymbol{\rho}_w) u(t), \boldsymbol{\rho}_{nl}) \quad (3.1)$$

$$\boldsymbol{\rho} = [\boldsymbol{\rho}_w, \boldsymbol{\rho}_{nl}, \boldsymbol{\rho}_h] \quad (3.2)$$

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

Notice that equations (3.1) and (3.2) 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 approach 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 3.7. For all three cases, the identification problem starts from (3.1) 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 parameterised, both LTI subsystems are represented in the continuous-time domain as rational transfer functions in factorised form (zero-pole-gain):

$$G_w(s) = K_w \frac{\prod_{i=1}^{n_b} (s + z_{w_i}) / (z_{w_i})}{\prod_{i=1}^{n_a} (s + p_{w_i}) / (p_{w_i})} \quad (3.3)$$

$$G_h(s) = K_h \frac{\prod_{i=1}^{n_d} (s + z_{h_i}) / (z_{h_i})}{\prod_{i=1}^{n_c} (s + p_{h_i}) / (p_{h_i})} \quad (3.4)$$

where $-p_{w_i}$ with $i = 1 \dots n_a$ and $-z_{w_i}$ with $i = 1 \dots n_b$ represent front LTI poles and zeros, respectively. In a similar way $-p_{h_i}$ with $i = 1 \dots n_c$ and $-z_{h_i}$ with $i = 1 \dots n_d$ 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's define vectors $\boldsymbol{\rho}_w$ and $\boldsymbol{\rho}_h$ as:

$$\boldsymbol{\rho}_w = [K_w, z_{w_1}, z_{w_2}, \dots, z_{w_{n_b}}, p_{w_1}, p_{w_2}, \dots, p_{w_{n_a}}] \quad (3.5)$$

$$\boldsymbol{\rho}_h = [K_h, z_{h_1}, z_{h_2}, \dots, z_{h_{n_d}}, p_{h_1}, p_{h_2}, \dots, p_{h_{n_c}}] \quad (3.6)$$

Notice that poles and zeros in (3.3) and (3.4) 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 non-linearity can also be represented in different ways. In this case, piece-wise functions are used as WH-EA uses them:

$$w(t) = f(v(t), \boldsymbol{\rho}_{nl}) \quad (3.7)$$

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 piece-wise 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 data set, where the input excitation signal $u(t)$ is a multi-step signal (see Subsection 3.7), while the measured output $y(t)$ may be corrupted by stationary additive noise $n(t)$:

$$y(t) = y_0(t) + n(t) \tag{3.8}$$

3.3 Initial linear model

In the context of Wiener-Hammerstein models, obtaining a perfect linear dynamic model in the presence of noise and non-linearities 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 non-linearities, and so the dynamics can be captured with great precision [93]. 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 non-linearity captured, and refinement of the dynamics is always possible to improve model preciseness. In this approach, 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 non-linearity. 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.¹

The purpose of this section 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*

¹It is advisable to avoid zones near operation limits since the non-linearity can be stronger due to saturation phenomena.

for continuous-time models (SRIVC) can be used [119, 33], available in the CONTinuous-Time System IDentification (CONTSID) toolbox for Matlab [30, 80, 32].

Since that initial linear model estimation is based on a step response, it is assumed that a small amount of data will be used. In this regard, the *Modified minimum Description Length* (MDL) criteria is a good option to select the best linear structure [17]:

$$MDL = \frac{V(\boldsymbol{\rho}l, \mathbf{Z})}{N_L} e^{p_c(n_{\rho l}, N_L)} \quad (3.9)$$

where $\boldsymbol{\rho}l$ 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, $V(\boldsymbol{\rho}l, \mathbf{Z})$ is the quadratic-like cost function depending on the difference between measurements and model ($\boldsymbol{\epsilon}$), computed using (3.10), and $p_c(n_{\rho l}, N_L)$ is the term known as the corrected penalty and is computed using (3.11).

$$V(\boldsymbol{\rho}l, \mathbf{Z}) = \frac{1}{2} \boldsymbol{\epsilon}^T \boldsymbol{\epsilon} \quad (3.10)$$

$$p_c(n_{\rho l}, N_L) = \frac{\ln(N_L)(n_{\rho l} + 1)}{N_L - n_{\rho l} - 2} \quad (3.11)$$

3.4 Optimisation problem statement

From the initial linear model and an input/output data set $\{u(t), yr(t)\}_{t=1}^N$, 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 non-linearity, and the pole/zero distribution of the initial linear model around the static non-linearity. 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:²

²Notice that any other criterion could be used in the proposed method, such quadratic or maximum error criteria.

$$\varepsilon_{NL}(t, \boldsymbol{\rho}) = yr(t) - y(t, \boldsymbol{\rho}) \quad (3.12)$$

$$J(\boldsymbol{\rho}) = \frac{1}{N} \sum_{t=1}^N abs(\varepsilon_{NL}(t, \boldsymbol{\rho})) \quad (3.13)$$

The solution of the optimisation problem is written as:

$$\hat{\boldsymbol{\rho}} = \arg \min_{\boldsymbol{\rho}} J(\boldsymbol{\rho}) \quad (3.14)$$

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

3.5 Genetic coding

WH-EA uses a coding in which a Wiener-Hammerstein model is represented by three portions of genetic information. The first one, \mathbf{P} , contains information about pole and zero locations. A second segment \mathbf{C} contains binary information classifying poles and zeros, i.e. which ones belong to G_w and which to G_h . Segment \mathbf{B} contains breakpoint coordinates representing the static non-linearity. This coding was defined in detail in Chapter 2, however, it should be noted that the formulation of the identification problem was done in discrete time.

The identification problem for this unified approach is formulated in continuous time, however, this does not have any implication that the genetic coding must be changed. One of the advantages of WH-EA is that it can be easily used without any alteration to identify models in continuous and discrete time.

In this sense, all the parameters of vector $\boldsymbol{\rho}$ are encoded in vectors \mathbf{P} , \mathbf{B} and \mathbf{C} . Figure 3.1 shows an example of how an individual has been coded to represent a nonlinear model in continuous time, where the initial linear model has six poles (four reals and two complexes) and three zeros (one real and two complexes). According to this structure, $mr = 4$, $nr = 1$, $mc = 1$ and $nc = 1$. On the other hand, it is assumed that static non-linearity is represented with five points ($n = 5$). Notice the correspondence between vectors \mathbf{C} and \mathbf{P} : if an element of vector \mathbf{C} has the value 1, the corresponding zero (or pole) of \mathbf{P} is located in the input LTI subsystem $G_w(s)$, while a 0 value in \mathbf{C} indicates that it is located in the output block $G_h(s)$. In Figure 3.1, the first element of \mathbf{C} is zero, so the first one of \mathbf{P} belongs to $G_h(s)$. Since the first element

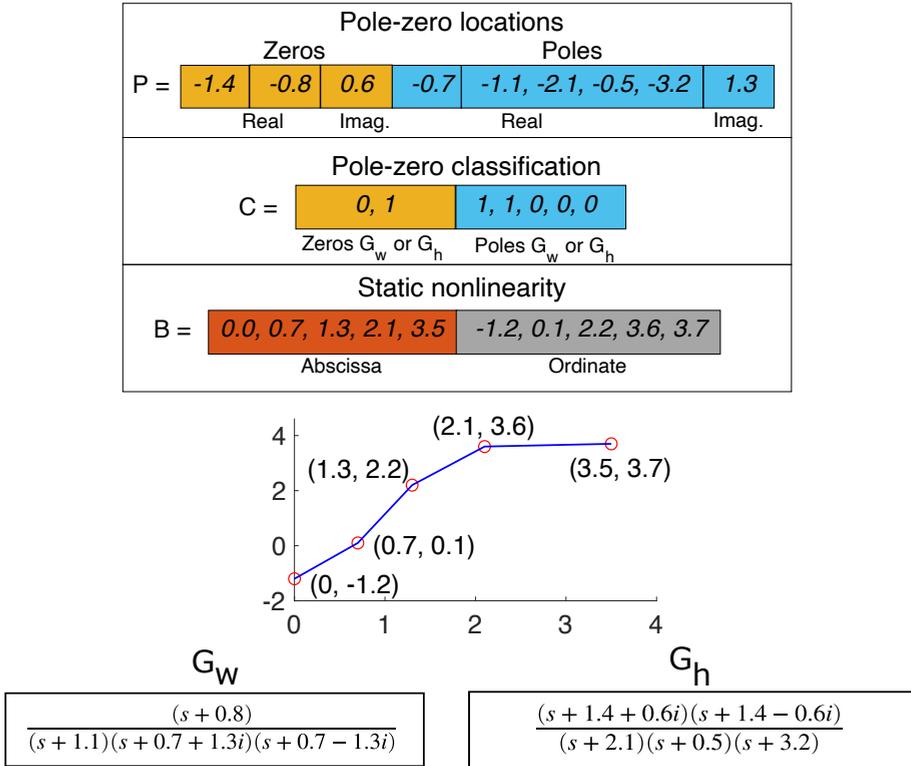


Figure 3.1: Example of an encoded individual and the resulting nonlinear model.

of \mathbf{P} contains the real part of a complex zero, the output LTI subsystem must include this zero $-1.4 + 0.6i$ and its conjugate $-1.4 - 0.6i$. Since the second element of \mathbf{C} is 1 then the second one of \mathbf{P} is located in the input LTI subsystem. Under this same logic, the poles are distributed between both LTI subsystems.

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

3.6 Search space for poles and zeros

New pole and zero locations are bounded by \mathbf{P}^{min} and \mathbf{P}^{max} (components of vector \mathbf{P} are element-wise bounded by vector \mathbf{P}^{min} and \mathbf{P}^{max}). Bounds in \mathbf{P}^{min} and \mathbf{P}^{max} are defined around the location of each pole or each zero of the initial linear model. They can be set according to the sensitivity of poles and zeros. That is, closest to the imaginary axis, smaller the search space, and conversely farthest to the imaginary axis, larger bounds. By way of illustration, assuming that the poles and zeros in the example of Figure 3.1 belong to the initial linear model, vectors \mathbf{P}^{min} and \mathbf{P}^{max} could be set as follows:

$$\mathbf{P}^{min} = [-1.7, -1.1, 0.3, -1.0, -1.4, -2.6, -0.8, -4.2, 1.0] \quad (3.15)$$

$$\mathbf{P}^{max} = [-1.1, -0.5, 0.9, -0.4, -0.8, -1.6, -0.2, -2.2, 1.6] \quad (3.16)$$

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

3.7 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 non-linearity. 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 non-linearity, and 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 non-linearity 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 non-linearity itself. This can be clearly seen in Figure 3.2, where a Gaussian signal has been used to excite three models containing the same dynamics and the same static non-linearity. 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 chapter.

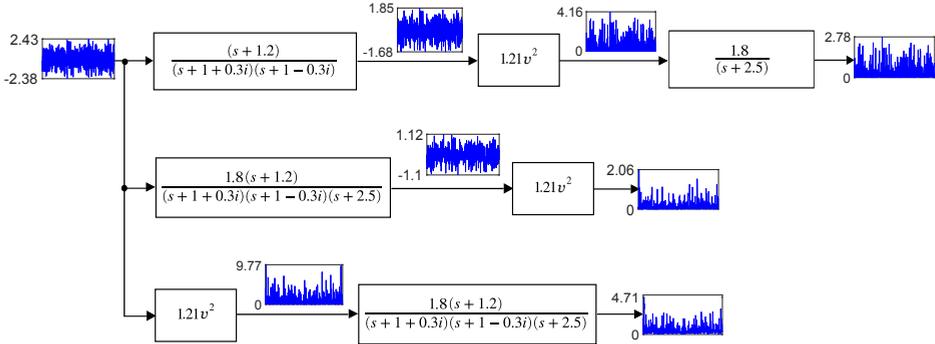


Figure 3.2: Example of search space limits for static non linearity in Wiener, Hammerstein, and Wiener-Hammerstein models when they are excited with a Gaussian signal.

In the case of Wiener and Wiener-Hammerstein models, the limits that define the horizontal search space of the non-linear 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 3.2 the limits that horizontally define the search space for static non-linearity 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 non-linearity. 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 Wiener and Hammerstein models assuming an arbitrary excitation signal (e.g., a Gaussian signal). The search space for static non-linearity of the Wiener-Hammerstein models was addressed in Section 2.4. This search space was defined as a function of the minimum and maximum values of the intermediate signals $v(t)$ and $w(t)$. These variables are not measurable, however, through (2.17), (2.18), (2.19), and (2.20) it is possible to find the corresponding minimum and maximum values.

For Wiener and Hammerstein models, it will also be assumed that the intermediate signals are not measurable and the dynamic blocks are non-reversible. 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. As in Wiener-Hammerstein models, for both cases addressed in this section it will be assumed 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} , a minimum value y_{min} and it has a mean y_{mean} .

3.7.1 Search space in Wiener models

In this case, the input signal $u(t)$ produces an output $v(t)$ and non-linearity search space is horizontally bounded by (v_{min}, v_{max}) whereas that 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 non-linearity it would not be necessary to use (2.17) and (2.18). 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 non-linearity is unknown, therefore it is not possible to estimate $v(t)$, rather it is necessary to establish a search space for non-linearity 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 (2.17) and (2.18), considering the scaling factor Ω . The static non-linearity search space for a Wiener model is shown in Figure 3.3. The

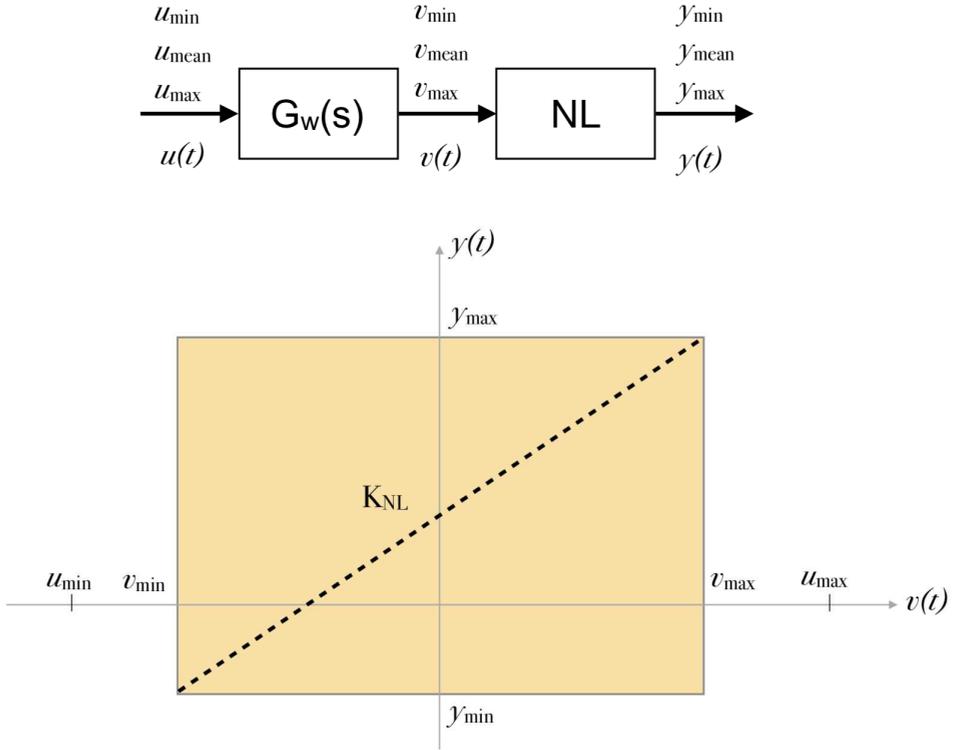


Figure 3.3: Search space for nonlinear static function in a Wiener model.

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.

3.7.2 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 non-linearity, 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. Further, our approach assumes that pole/zero distribution around non-linearity 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

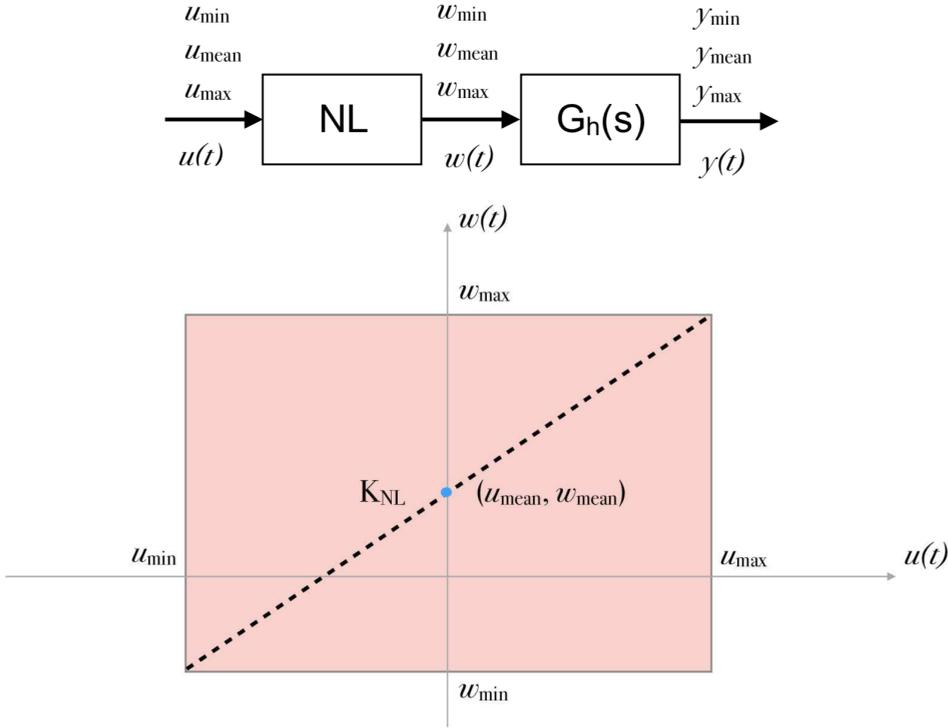


Figure 3.4: Search space for nonlinear static function in a Hammerstein model.

Wiener-Hammerstein models. The search space for the static non-linearity of a Hammerstein model is illustrated in Figure 3.4.

3.7.3 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 non-linearity are different for the three types of models. This fact implies that the identification algorithm should change the search space over which the non-linearity 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 causes 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 $G_w(s)$ normalized 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 (2.17) and (2.18), 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 3.5 shows how the horizontal and vertical limits that give rise to the search space of static non-linearity are the same for the three types of models. The models used are the same as in Figure 3.2, the difference is that the static gains of the dynamic blocks in Figure 3.5 have been normalised to 1. A multi-step signal has been used to excite the three models. The steps duration ensure 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.

A good option to obtain the output of a dynamic system at steady state is to apply step inputs of sufficient duration. In Subsection 3.8 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 $y(t)$

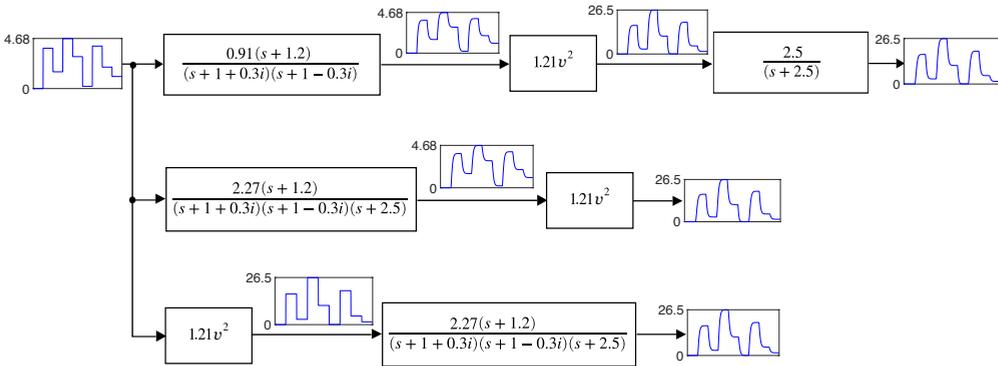


Figure 3.5: Example of search space limits for static non linearity in Wiener, Hammerstein, and Wiener-Hammerstein models when they are excited with a multi-step signal.

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}) .

3.8 The multi-step 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 non-linearity that is present throughout the entire process operating range, it will be necessary to design a multi-step signal with different amplitudes.

For this aim, three important aspects must be considered: number and amplitude of steps; minimum step amplitude and step duration. 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 non-linearity 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 needs to be used for identification of a nonlinear model deserves debate, and the vast majority of nonlinear model identification methods require a large volume of input and output data.

Calculating the static non-linearity 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. Further, minimum step amplitude should also be considered when designing the multi-step 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 non-linearity is visible. Therefore, amplitude changes of the steps must be large to highlight non-linearity.

Figure 3.6 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 non-linearity and the same dynamic have been considered. The non-linearity 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 non-linearity, while the remaining pole was placed afterwards).

In the lower graph of Figure 3.6, no difference between the responses can be seen when the excitation signal has small amplitude changes. Conversely, the upper graph shows a marked difference between responses when the excitation signal has larger amplitude changes. Table 3.1 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 non-linearity. Notice how low \mathbf{MAE}_{ss} values imply no significant difference between the structures formed as the algorithm cannot split the dynamics properly.

Multi-step signals are ideal to highlight the non-linearities 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 multi-step 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 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

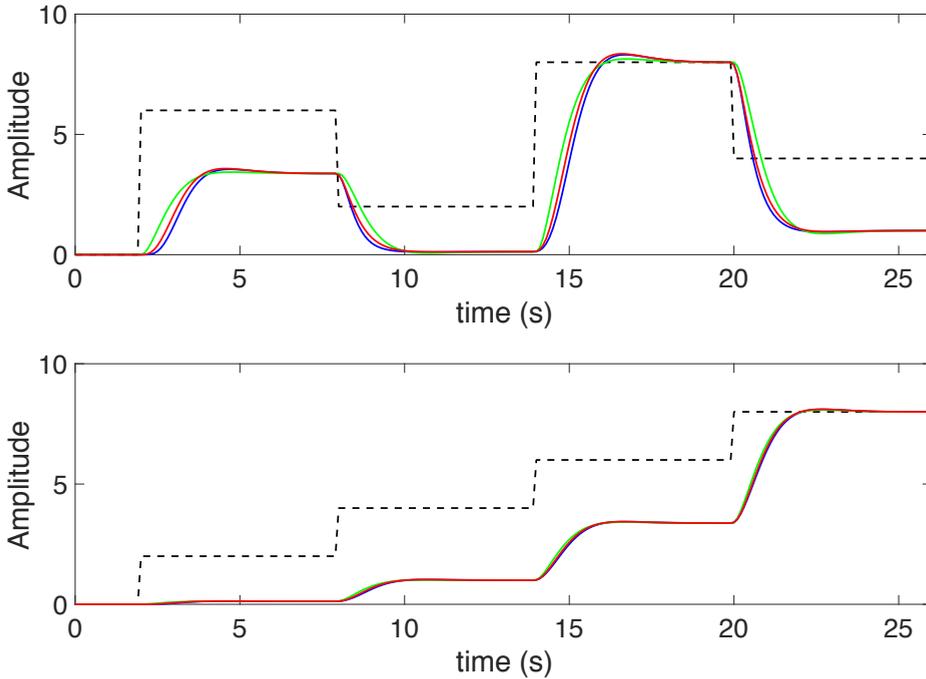


Figure 3.6: Responses of Wiener (blue), Hammerstein (green), and Wiener-Hammerstein (red) models when they are excited with a multi-step signal (dashed) with large (top) and small (bottom) amplitude changes.

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 3.2 shows a comparison of the characteristics of a Gaussian signal and a multi-step 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

Table 3.1: 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

Table 3.2: Comparison of characteristics between a multi-step signal and a Gaussian signal.

Features	Multi-step signal	Gaussian signal
Frequency content.	Low.	High.
Applicability on real processes.	Highly applicable.	It is not always possible.
Information on static non-linearity.	High information content.	Lower information content, especially if the non-linearities 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.

where the BLA estimation is not possible, either because they are slow dynamically, or they are not enabled to handle Gaussian signals.

3.9 Application examples

This 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 non-linearity were interpolated to define a piece-wise linear function. WH-EA parameters were set the same for the four identification problems: $\xi = 0.25$; $\delta_{zp} = 0.75$; $\delta_{nl} = 0.75$; $\eta_{min} = 0.35$. In addition, initial and final standard deviations for mutations were set to 20 and 1 respectively.

3.9.1 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 3.7). Static nonlinearity consists of a sigmoid hyperbolic tangent function "*tansig*" (3.17), which symmetrically saturates large values of the independent variable.

$$S_{NL} = 50(1 + \text{tansig}(0.05(x - 50))) \quad (3.17)$$

The LTI subsystem used for this example has unitary static gain. Although the methodology proposed in this chapter enables the identification of block-oriented models with a static non-linearity 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 20s. 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 -30db was added on the system output. Input and output data were sampled with a period of 10ms. Figure 3.8 depicts the excitation signal and the response of the simulated system.

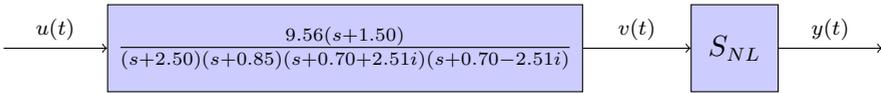


Figure 3.7: Numerical example 1: Wiener structure.

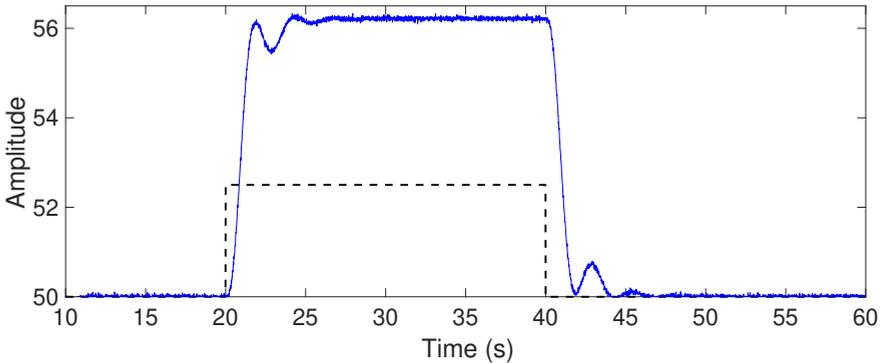


Figure 3.8: Input (black dashed-line) and output (blue solid-line) data for initial linear model estimation (numerical example 1).

The data obtained with the first input change $t(15 \dots 35)$ was used to estimate the initial linear model, while the data belonging to the second input change $t(36 \dots 56)$ was 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) data sets. In addition, for better support in selecting the best structure, the modified MDL criterion was calculated using (3.9).

The results of the estimates are shown in Table 3.3. According to lowest value of MDL criterion, the best estimated model was of four poles and one zero (3.18), 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.³

³From the model of four poles and a zero in cases where the MSE decreases, this decrease is practically negligible.

Table 3.3: Ranking of estimated linear models for numerical example 1 (Wiener system). Models 4p, 2z/5p, 3z/5p, 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

$$G_{w_{ine}}(s) = \frac{23.96(s + 1.52)}{(s + 2.51)(s + 0.85)(s + 0.70 + 2.51i)(s + 0.70 - 2.51i)} \quad (3.18)$$

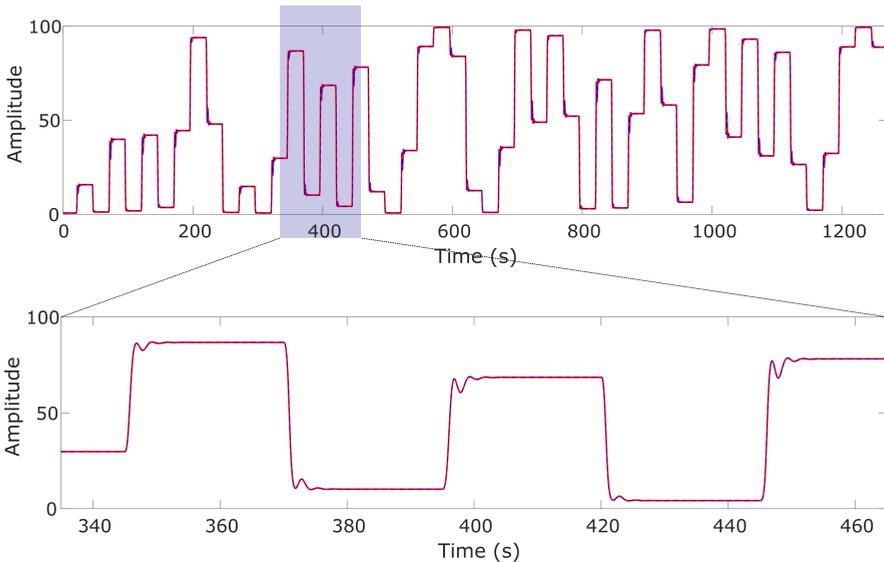
For nonlinear identification, two multi-step inputs were generated, one for estimation and another for validation purposes. Both signals were designed with 50 steps of random amplitude to explore the entire operating range of the input process (0-100%). Each step had a time duration of 25s and the minimum step amplitude 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 10ms. From the estimation data set, the minimum and maximum values of the input and output signals were obtained to define the search space for the static non-linearity. 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 non-linearity was defined with (u_{min}, y_{min}) and (u_{max}, y_{max}) .

Once the search space for static non-linearity was defined, WH-EA was configured according to the data of Table 3.4. In addition, \mathbf{P}_1^0 was coded with $nc = 0$, $nr = 1$, $mc = 1$, $mr = 2$, and pole/zero locations of (3.18). Further, 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

Table 3.4: 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 non-linearity	18
α	Minimum distance between two points	0.9

**Figure 3.9:** Validation results for numerical example 1: real output (red solid-line) and model output (blue dashed-line).

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 data set (MAE_v) was $5.567E - 2$. Figure 3.9 depicts a comparison on the validation data set 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 3.10 shows how the 18 points representing the static non-linearity 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.

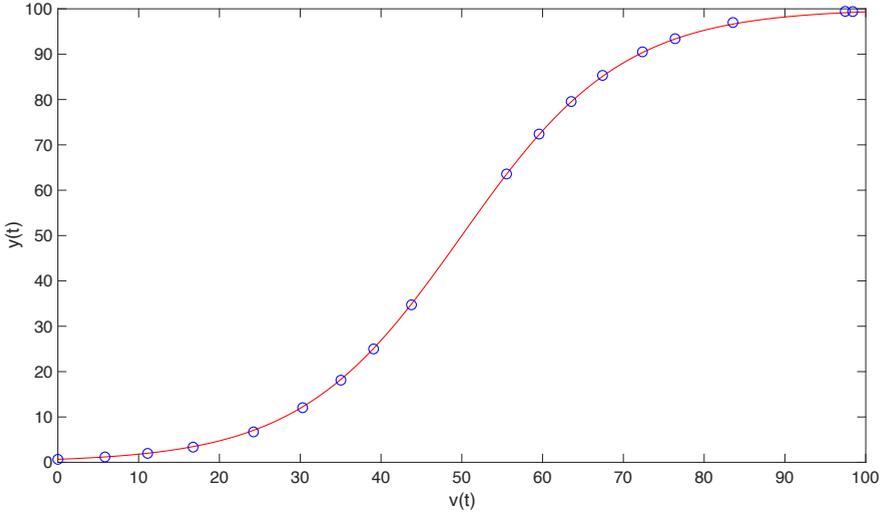


Figure 3.10: Numerical example 1: Comparison between real nonlinear function *tansig* (red) and captured non-linearity (blue circles).

3.9.2 Numerical Example 2

For this numerical example, the same linear subsystem and the same static nonlinearity of the previous numerical example were used, however, the blocks were permuted to give rise to a Hammerstein model (see Figure 3.11). Similarly, the same input signals that were used in the previous example were used to excite this simulated model. With the corresponding data set 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 3.5. The best linear model (3.19) according to the *MDL* criterion was four poles and one zero corresponding to the order of the real system.

From the nonlinear estimation data set, the search space for static non-linearity 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 $y_{max} = 99.45$.

$$Gh_{lne}(s) = \frac{23.87(s + 1.53)}{(s + 2.51)(s + 0.85)(s + 0.70 + 2.51i)(s + 0.70 - 2.51i)} \quad (3.19)$$

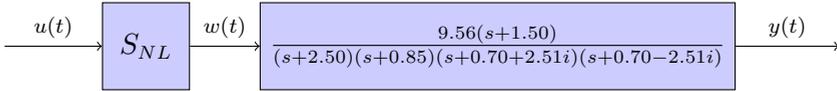


Figure 3.11: Numerical example 2: Hammerstein structure.

Table 3.5: Ranking of estimated linear models for numerical example 2 (Hammerstein system). Models 4p, 3z/5p, 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

For non-linear estimation, WH-EA was executed considering the configuration parameters of Table 3.4. In addition, P_1^0 was coded with $nc = 0$, $nr = 1$, $mc = 1$, $mr = 2$, and pole/zero locations of (3.19), were used. Further, 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, this is, a Hammerstein model was obtained. The value reached for the objective function (MAE_e) was $4.328E-2$, while the absolute error on the validation data set (MAE_v) was $6.526E-2$. Figure 3.12 depict a comparison on the validation data set between the simulated output generated by the numerical example and the output of the estimated model. On the other hand, Figure 3.13 shows how the 18 points were distributed within the search space to capture the static non-linearity. As in the previous case, the real nonlinear function was introduced in this graph to visualise the precision achieved with WH-EA.

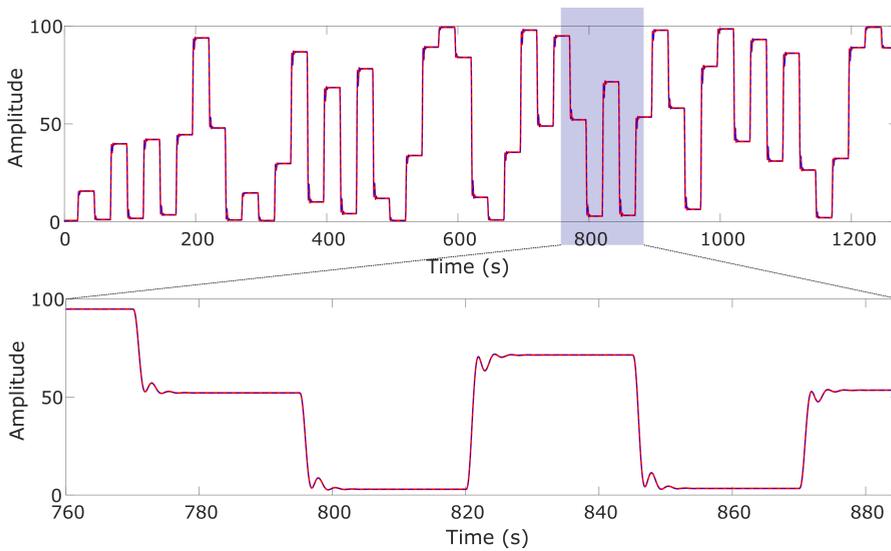


Figure 3.12: Validation results for numerical example 2: real output (red solid-line) and model output (blue dashed-line).

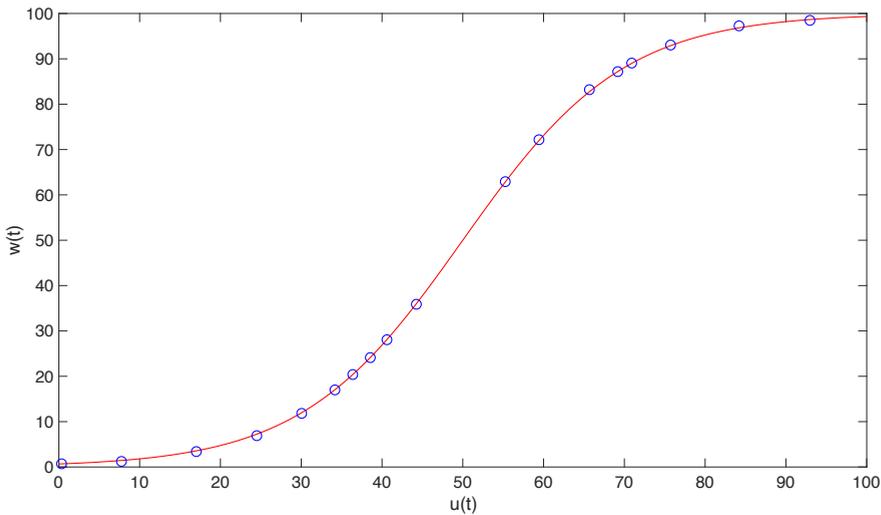


Figure 3.13: Comparison between real nonlinear function *tansig* (red) and captured non-linearity (blue circles). Results obtained on the numerical example 2.

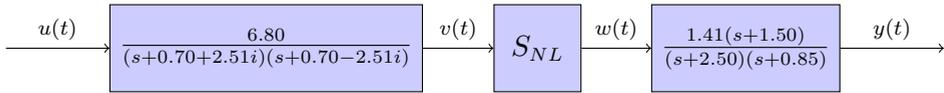


Figure 3.14: Numerical example 3: Wiener-Hammerstein structure.

Table 3.6: Ranking of estimated linear models for numerical example 3 (Wiener-Hammerstein system). Models 4p, 1z/5p, 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

3.9.3 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 3.14). The excitation signals and the procedures for linear and non-linear estimation were the same as those used in the previous examples. Ranking of linear estimates are shown in Table 3.6. As in the previous cases, the best linear model according to the MDL criterion was of four poles and one zero (3.20), which is consistent with the dynamics of the real system even though for this case the dynamics was distributed around the static non-linearity. This demonstrates the great effectiveness of the linear estimation method used in this approach.

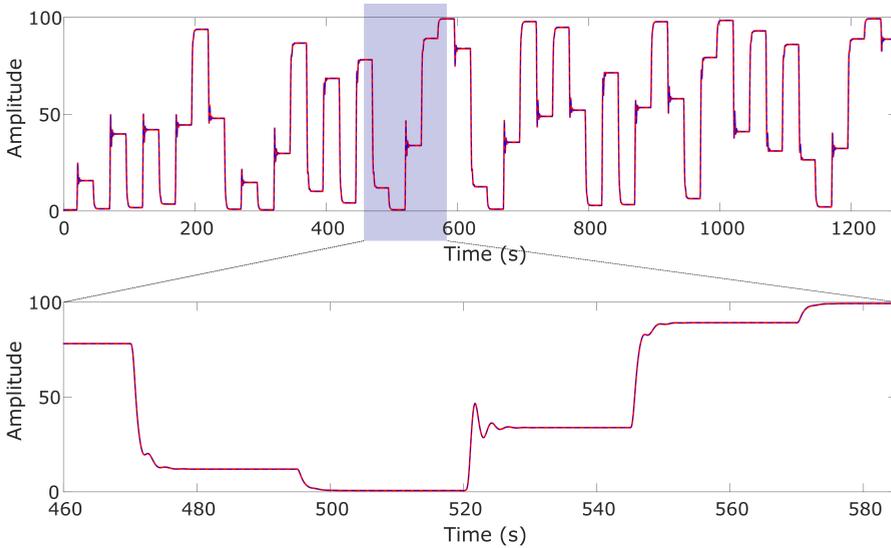


Figure 3.15: Validation results for numerical example 3: real output (red) and model output (blue dashed line).

$$Gwh_{lne}(s) = \frac{24.195(s + 1.549)}{(s + 2.598)(s + 0.854)(s + 0.709 + 2.507i)(s + 0.709 - 2.507i)} \quad (3.20)$$

As can be seen, the linear models obtained in (3.18), (3.19) and (3.20) 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 non-linearity. This corroborates what was indicated in Section 3.8. A step signal with a small amplitude change is useful for linear estimation, however, for nonlinear estimation, it is necessary that the non-linearity 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 3.4. In addition, the first individual of the population (\mathbf{P}_1^0) was coded with with $nc = 0$, $nr = 1$, $mc = 1$, $mr = 2$, and pole/zero locations of (3.20). According to the minimum and maximum values of the input and output signals, the search space of the static non-linearity was defined with the coordinates: (0, 0.5546) and (100, 99.443).

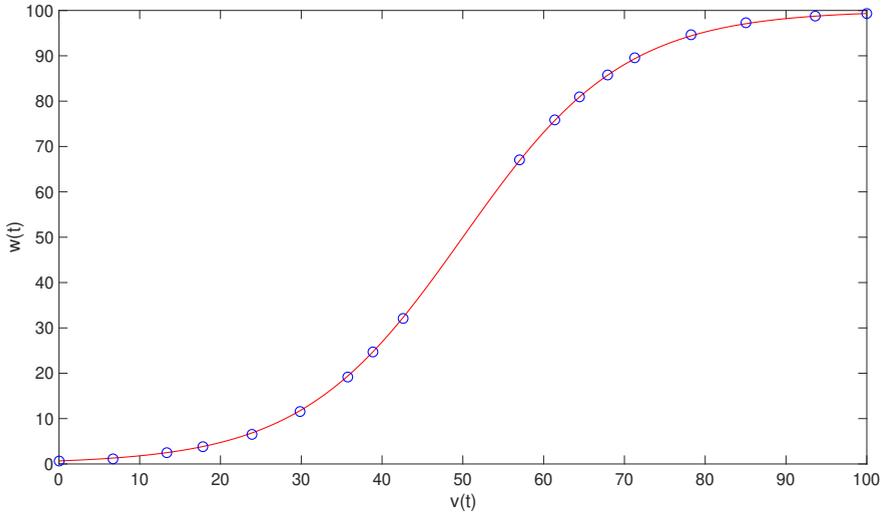


Figure 3.16: Example 3: Comparison between real nonlinear function *tansig* (red) and captured non-linearity (blue circles).

At the end of the generations, a Wiener-Hammerstein model was obtained and the dynamics of both LTI subsystems was consistent with the real system. The value reached for the objective function (MAE_e) was $3.768E - 2$, while the absolute error on the validation data set (MAE_v) was $5.117E - 2$. Figure 3.15 depict a comparison on the validation data set between the simulated output generated by the numerical example and the output of the estimated model. On the other hand, Figure 3.16 shows a comparison between the real and estimated non-linearity.

3.9.4 Thermal process identification

The real process used to validate this approach 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 3.17 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

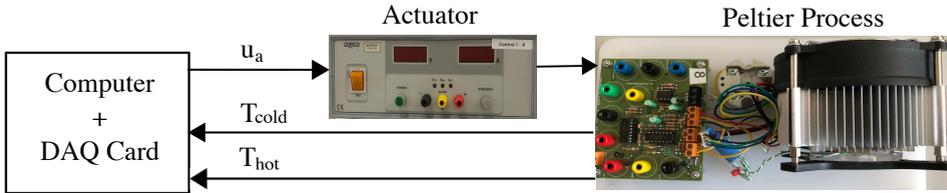


Figure 3.17: The real process based on a Peltier cell.

voltage signal u_a ($0 \dots 4.5Vdc$) 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 100ms 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 surfaces ($\Delta T = T_{cold} - T_{hot}$).

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 (2.25V). The first step had a positive amplitude of 0.225V (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 700s. The applied input signal and the response of the system are shown in Figure 3.18.

The data obtained with the first step change $t(0 \dots 750)$ was used to estimate the initial linear model, while the data belonging to the second input change $t(751 \dots 1451)$ was used for validation. To avoid problems with initial conditions, offset was removed from data sets. Different linear models were estimated from second to fifth order. Results of the estimates are shown in Table 3.7. For each estimated model the *MDL* criterion, error on estimation, and error on validation data sets were computed. Models 2z/3p, 3z/4p, 4z/5p were not considered, since they were of non-minimum 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.

The best structure according to the *MDL* criterion was four poles and one zero (3.21). Figure 3.19 shows a comparison between the estimated model output and the real process output. This comparison has been made considering the validation data set.

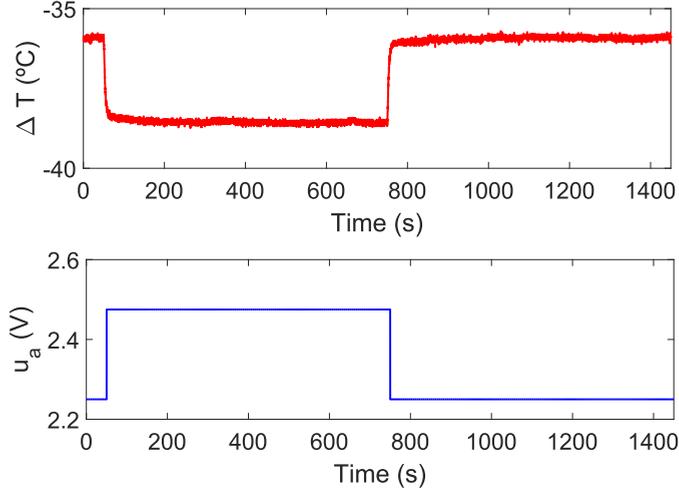


Figure 3.18: Input and output data for estimation of initial linear model (thermal process). Top: excitation signal (blue). Bottom: output signal of the thermal process. (red).

Table 3.7: Ranking of estimated linear models for the thermal process. Models 4p and 5p have been excluded due to their *MDL* high values.

Structure	MSE_e ($^{\circ}C$)	MSE_v ($^{\circ}C$)	<i>MDL</i>
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

$$G_{lm}(s) = \frac{-41.186(s + 0.015)}{(s + 4.473)(s + 2.519)(s + 0.3406)(s + 0.014)} \quad (3.21)$$

For the nonlinear identification, two multi-step signals were generated, one for identification and another for validation purposes (see Figure 3.20). 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 700s and the amplitude changes were handled randomly within the entire

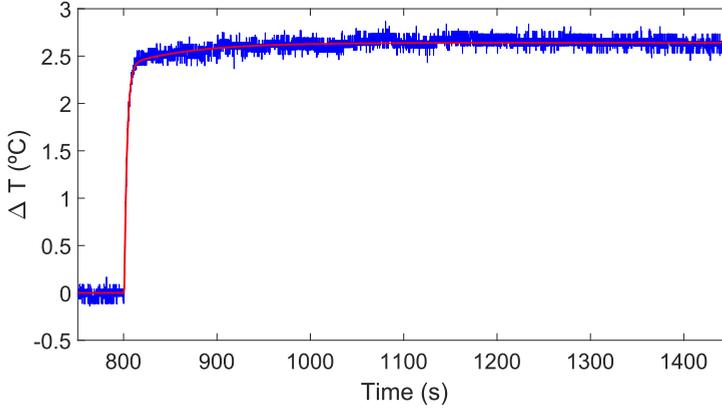


Figure 3.19: Comparison between the estimated model output (red) and the real process output (blue).

range of the actuator u_a ($0 \dots 4.5v$) whereas the minimum step amplitude was constrained to be greater than $1.5V$. 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 3.8 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:

$$\mathbf{P}_1^0 = [-0.015, -4.473, -2.519, -0.3406, -0.014]. \quad (3.22)$$

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 data set: $u_{min} = 0V$, $u_{max} = 4.5000V$, $y_{min} = -53.957^\circ C$, $y_{max} = -0.1810^\circ C$. Since the static gain of the estimated initial linear model is negative, the search space for static non-linearity 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 decided to divide the initial linear model into two linear subsystems, therefore the best structure to represent the thermal process corresponds to a Wiener-Hammerstein model. Table 3.9 presents the coordinates of the nine points that were assigned to the static non-linearity,

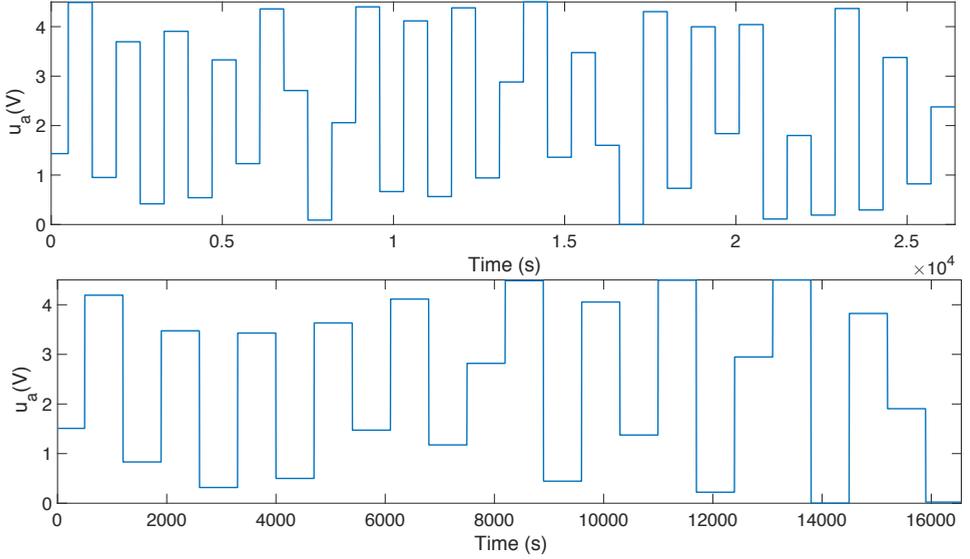


Figure 3.20: Input signals for nonlinear identification of the Peltier process. Top: signal for estimation. Bottom: signal for validation.

Table 3.8: 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

while a plot of this non-linearity is presented in Figure 3.21. Equations (3.23) and (3.24) show the two resulting subsystems, while performance of the WH identified model on estimation and validation data sets is shown in Figure 3.22. To quantify the accuracy of the estimated model, the MAE was calculated on the estimation and validation data sets with values of 0.1435 and 0.2184 respectively. To calculate both errors, the first 3000 samples of the data sets were not considered to avoid the transient effects.

Table 3.9: Non-linearity coordinates ($n = 9$) estimated by WH-EA from thermal process.

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

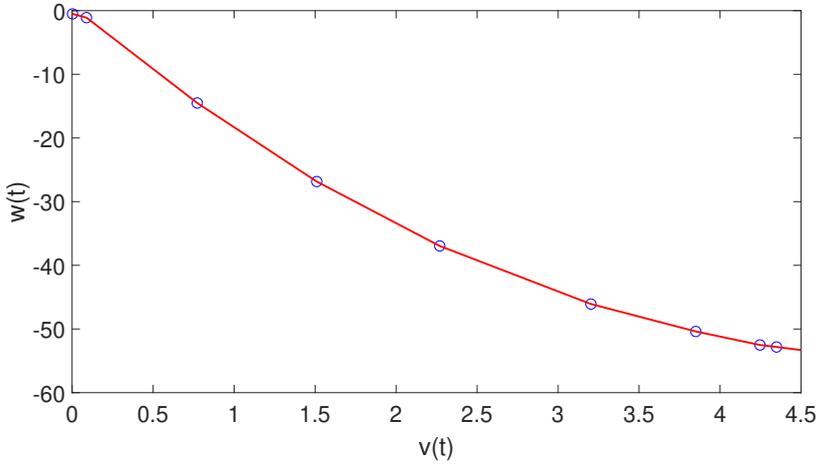


Figure 3.21: Captured non linearity as a piece-wise linear function (red) from the estimated breaking points (blue circles).

$$\hat{G}_w(s) = \frac{11.393}{(s + 4.394)(s + 2.593)} \quad (3.23)$$

$$\hat{G}_h(s) = \frac{0.293(s + 0.024)}{(s + 0.3263)(s + 0.022)} \quad (3.24)$$

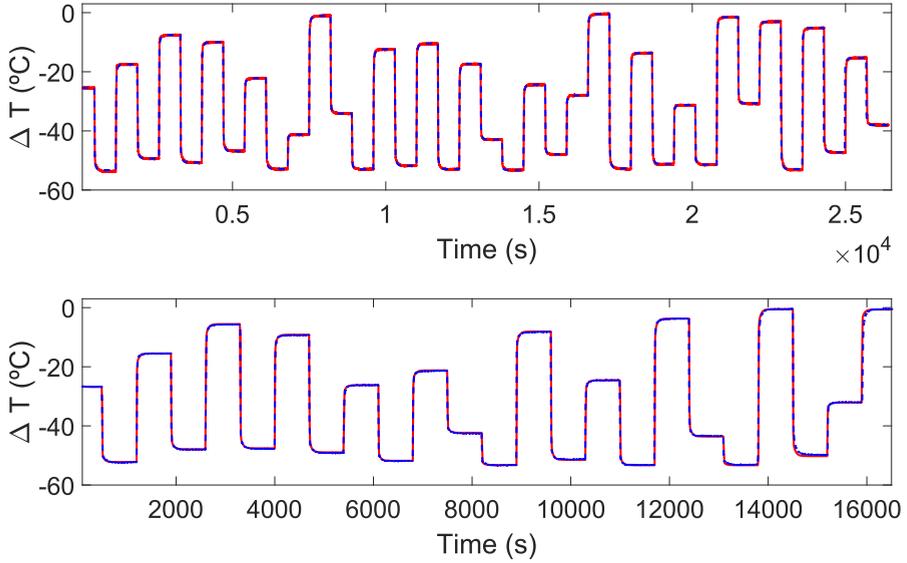


Figure 3.22: Comparison between thermal process output (red) and estimated model output (blue dashed line). Top: comparison on estimation data set. Bottom: comparison on validation data set.

3.9.5 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 non-linearity. 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 non-linearity. 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 non-linearity. A comparison of the errors obtained from the estimation and validation data sets 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 3.9, 3.12, and 3.15, 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 chapter, it also

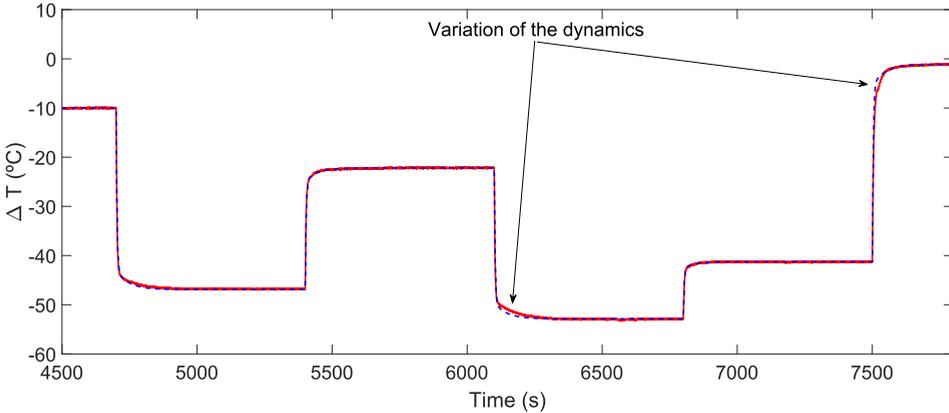


Figure 3.23: Variation of the dynamics detected on estimation data set for the thermal process. Red: Output of the thermal process. Blue dashed line: Output of the estimated model.

depends on the number of points assigned to capture the static non-linearity and the quality of the initial linear model.

The real process was estimated with 23 parameters: 5 for the linear dynamic part and 18 for static non-linearity. 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 $\hat{G}_w(s)$ has been separated from the slow dynamics of the Peltier cell $\hat{G}_h(s)$.

A great advantage of using multi-step 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 3.22, 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 3.23).⁴ 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 non-linearity that was not well captured.

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

⁴Variation of the dynamics shown in Figure 3.23 are not the only ones, other similar variations were detected over other portions of estimation and validation data.

static nonlinearity. It is evident that a hard non-linearity 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 piece-wise 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 non-linearity 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 $-53.957^{\circ}C$ and $-0.181^{\circ}C$, therefore the operating range was $53.776^{\circ}C$. For this operating range, the MAE between the real output and estimated output was $0.1435^{\circ}C$. 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 $-30dB$ was added to the output of each simulated model. The mean absolute value of this noise signal was $2.52E - 2$ 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 $60dB$. 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 multi-objective optimization approach.

Regarding computational cost of WH-EA⁵, a reference can be obtained. For example, in section 3.9.3 (Wiener-Hammerstein example), the average time to run a generation was 0.07s, 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% (0,99ms) 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.

⁵WH-EA was run on a computer with Intel Core I7 processor of 2.8GHz and 8,0Gb of RAM.

Chapter 4

This chapter presents a multi-objective approach to the identification of Wiener-Hammerstein models and their specific cases. The chapter begins with an introduction and some preliminary considerations. Next, in the third section the identification problem under a multi-objective approach is stated. The fourth section presents in detail the multi-objective optimisation algorithm developed to address the identification problem. The fifth section presents a procedure using the multi-objective algorithm considering various design concepts. In the sixth section the multi-objective algorithm is evaluated by identifying a simulated continuous stirred-tank reactor. Finally, in the seventh section the algorithm is tested again by estimating a numerical example and a thermal process, but this time including the complete procedure where several design concepts are considered.

4.1 Introduction

As seen in Chapter 1, a large number of methods for identifying Wiener-Hammerstein models can be found to date. Most of them starting from a linear dynamic model whose poles and zeros are distributed around the static non-linearity. To achieve good precision in the estimation, the BLA [64, 23, 101] has usually been used to represent the linear dynamics. However, in practical applications, especially those that involve automatic control, precision should not be the only criteria for model selection.

The nature of block-oriented models forces the user to initially define the complexity of the dynamic part generally represented by some poles and zeros and also the complexity of the static nonlinear part. This is not an easy task, especially in the case of Wiener-Hammerstein models, where two LTI subsystems must be estimated. Some criteria are currently available such as Minimum Description Length (MDL) [89], Akaike information criterion (AIC) [2], Modified AIC and MDL criteria [17], and Final Prediction Error (FPE) [4, 3]. These help to define the complexity of the overall system dynamics. However, two or more different linear structures can give the same value to a specific criterion, or several criteria can offer opposite values for the same linear structure. To date, there is no recipe to precisely define the number of parameters to represent the static nonlinear part. This leads to several identification tests with different numbers of parameters until the desired accuracy is achieved. This task can be tedious and even worse if the linear structure is not well defined, since no matter how many parameters are added to the static non-linearity, the unmodelled dynamics will not be able to achieve good accuracy.

Of all these complications, the model's applicability is an important issue when estimating Wiener-Hammerstein models. Although accuracy can be important when selecting a model, its complexity may also be a factor. For example, in control applications, such as Model Predictive Control (MPC) [88], it is well known that controller performance will depend largely on the model quality. However, the higher the model complexity, the higher the computational cost required to calculate the control action. In terms of the control algorithm, there is a significant gap between the MPC based on Wiener models [36, 77] or Hammerstein models [29] and the MPC based on Wiener-Hammerstein models [59].

Based on this background, this chapter considers a trade-off analysis between accuracy and complexity in estimating the Wiener-Hammerstein

model. Therefore, the identification problem is declared as a multi-objective optimization problem (MOOP).

Unlike linear models, defining a suitable structure (complexity) in nonlinear models is not a simple task. For this reason, the use of multi-objective optimisation (MOO) in nonlinear identification is not new. Thanks to MOO, it is possible to generate a Pareto set to compare and analyse the complexity and accuracy of different models in order to represent the same process and avoid over-fitting problems. Applications of MOO in nonlinear modelling includes the identification of: the Volterra series [67], radial-basis function (RBF) networks [43, 27, 87], Nonlinear Auto-Regressive eXogenous (NARX) models [39, 66], Nonlinear Auto-Regressive Moving Average models with eXogenous inputs (NARMAX) [28], and Wiener-Hammerstein models [74]. This last approach is based on genetic recombination and particle swarm optimisation. During the search process, the algorithm requires minimal user interaction, but even though a large number of poles and zeros are allowed for both LTI subsystems, good accuracy cannot be achieved since the optimisation problem does not use a linear approximation as a starting point. Furthermore, a polynomial is used for the static non-linearity, which is not recommended, since the sensitivity of the coefficients increases with the degree of the polynomial.

In this chapter, a new Multi-objective Evolutionary Algorithm for Wiener-Hammerstein identification (WH-MOEA) is proposed based on *ev*-MOGA (epsilon-variable Multi-Objective Genetic Algorithm) [44] but using some new genetic operators and others inherited from WH-EA [122]. WH-MOEA genetic operators perform a smart distribution and fine-tune the linear dynamics while capturing the non-linearity. To tackle model complexity, two new genetic operators are incorporated into WH-MOEA to increase or decrease the number of points assigned to the static non-linearity.

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

available literature and estimation tools, this is currently only possible using a different method for each type of structure.

Compared with [74], this approach has two significant advantages. It uses a linear approximation as a starting point for nonlinear estimation. Although an additional step is required, knowledge of linear dynamics leads to better results. Also, different Pareto fronts (one for each initial linear structure selected) can be compared in the multi-objective space for an exhaustive trade-off analysis.

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

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

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

4.2 Preliminary considerations

In this approach, the identification problem is presented as a search problem where the parameters of vectors $\boldsymbol{\rho}_w$, $\boldsymbol{\rho}_h$, and $\boldsymbol{\rho}_{nl}$, that were defined in Section 3.2, must be found from input and output measurements. It should be taken into account that parameters n_a , n_b , n_c , n_d and n define the model complexity but are unknown beforehand. They will be known after the optimisation algorithm has distributed the dynamics of the initial linear model, whose number of poles ($n_{poles} = n_a + n_c$) and zeros ($n_{zeros} = n_b + n_d$) is known in advance. Since parameter n is variable, the user must indicate the minimum

and maximum number of points allowed ($n_{min} < n < n_{max}$) for the nonlinear block.

Thanks to the unified approach presented in Chapter 3, the formulation presented here is also applicable for Wiener and Hammerstein models, which are specific cases of Wiener-Hammerstein structure when one of the two LTI blocks lacks dynamics. In the case of Hammerstein models $G_w = 1$ and ρ_w does not exist, whereas for Wiener models $G_h = 1$ and ρ_h neither does. In any case, the user will not need to specify the model structure to be identified, but rather the algorithm will decide the best structure (Wiener, Hammerstein or Wiener-Hammerstein) from the measured data set. However, it should be noted that the excitation signal used for non-linear estimation must be a multi-step signal, with the features specified in Section 3.8.

This approach uses a multi-objective optimisation algorithm to solve the identification problem detailed in Section 3.2. This algorithm is based on a population of individuals, and each individual represents a solution of the problem. All individuals contain coded genetic information corresponding to the structure and parameters of a model. This genetic information is composed of three segments. The first one, \mathbf{P} , contains information about pole and zero locations. A second segment \mathbf{C} contains binary information classifying poles and zeros, i.e. which ones belong to G_w and which to G_h . Segment \mathbf{B} contains breakpoint coordinates representing the static non-linearity. Its size is variable between individuals and it can also change from generation to generation. This variation will be exploited to generate a set of optimal solutions (models of different complexity).

Regarding parameter bounds of non-linearity, WH-MOEA takes the approach defined in Chapter 3, where a common search space is defined to face the uncertainty of the location of the static non-linearity around the dynamics. The bounds will be vertically defined by the minimum (y_{min}) and maximum (y_{max}) values of the output signal, while the horizontal bounds will be given by the minimum (u_{min}) and maximum (u_{max}) values of the multi-step input signal. To prevent disorderly horizontal movement of breakpoints, movement of each breakpoint will be constrained by the position of the neighbouring breakpoints. In addition, to avoid an overlap, a minimum distance between breakpoints will be considered through the user-defined parameter α . The minimum and maximum bounds that define the vertical and horizontal search space of each breakpoint are expressed through:

$$\mathbf{W}^{min}(\mathbf{i}) = y_{min} \quad \text{for } i = 1 \dots n \quad (4.1)$$

$$\mathbf{W}^{max}(\mathbf{i}) = y_{max} \quad \text{for } i = 1 \dots n \quad (4.2)$$

$$\mathbf{V}^{min}(\mathbf{i}) = \begin{cases} u_{min} & \text{if } i = 1 \\ v_{i-1} + \alpha & \text{if } i = 2 \dots n \end{cases} \quad (4.3)$$

$$\mathbf{V}^{max}(\mathbf{i}) = \begin{cases} v_{i+1} - \alpha & \text{if } i = 1 \dots n - 1 \\ u_{max} & \text{if } i = n \end{cases} \quad (4.4)$$

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

$$\mathbf{B}^{min} = [\mathbf{V}^{min}, \mathbf{W}^{min}] \quad (4.5)$$

$$\mathbf{B}^{max} = [\mathbf{V}^{max}, \mathbf{W}^{max}] \quad (4.6)$$

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

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

4.3 Multi-objective optimization problem statement

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

$$\min_{\mathbf{x}} \mathbf{f}(\mathbf{x}) \quad (4.7)$$

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

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

Definition 4.1 (*Pareto optimality [69]*): An objective vector $\mathbf{f}(\mathbf{x}^2)$ is Pareto optimal if there is not another objective vector $\mathbf{f}(\mathbf{x}^1)$ such that $f_i(\mathbf{x}^1) \leq f_i(\mathbf{x}^2)$ for all $i \in [1, 2 \dots m]$ and $f_j(\mathbf{x}^1) < f_j(\mathbf{x}^2)$ for at least one $j, j \in [1, 2 \dots m]$.

Definition 4.2 (*Dominance [15]*): An objective vector $\mathbf{f}(\mathbf{x}^1)$ is dominated by another objective vector $\mathbf{f}(\mathbf{x}^2)$ if $f_i(\mathbf{x}^2) \leq f_i(\mathbf{x}^1)$ for all $i \in [1, 2 \dots m]$ and $f_j(\mathbf{x}^2) < f_j(\mathbf{x}^1)$ for at least one $j, j \in [1, 2 \dots m]$. This is denoted as $\mathbf{f}(\mathbf{x}^2) \preceq \mathbf{f}(\mathbf{x}^1)$.

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

Definition 4.3 (*Pareto set, \mathbf{X}_p*): The Pareto set is the set of solutions in D that are not dominated by another solution in D .

$$\mathbf{X}_p := \{\mathbf{x} \in D \mid \nexists \mathbf{x}' \in D : \mathbf{f}(\mathbf{x}') \preceq \mathbf{f}(\mathbf{x})\}$$

Definition 4.4 (*Pareto front, $\mathbf{f}(\mathbf{X}_p)$*): Given a Pareto set, \mathbf{X}_p , the Pareto front is defined as

$$\mathbf{f}(\mathbf{X}_p) := \{\mathbf{f}(\mathbf{x}) \mid \mathbf{x} \in \mathbf{X}_p\}.$$

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

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

$$\min_{\mathbf{x}} \{[f_1(\mathbf{x}) \quad f_2(\mathbf{x})]\} \quad (4.8)$$

subject to:

$$\mathbf{P}^{min} \leq \mathbf{P} \leq \mathbf{P}^{max} \quad (4.9)$$

$$\mathbf{B}^{min} \leq \mathbf{B} \leq \mathbf{B}^{max} \quad (4.10)$$

$$n_a \geq n_b \quad (4.11)$$

$$n_c \geq n_d \quad (4.12)$$

$$n_a + n_c = n_{poles} \quad (4.13)$$

$$n_b + n_d = n_{zeros} \quad (4.14)$$

$$\mathbf{x} = [\mathbf{P}, \mathbf{B}, \mathbf{C}] \quad (4.15)$$

where

$$f_1(\mathbf{x}) = \frac{\sum |y - y_r|}{N} \quad (4.16)$$

$$f_2(\mathbf{x}) = n_a + n_c + n_b + n_d + n \quad (4.17)$$

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

It should also be taken into account that the metrics defining the objectives f_1 and f_2 are independent of those that can be used in the estimation and selection of linear structures, which, as explained in Subsection 4.5.2, is a preliminary step to multi-objective optimisation.

4.4 Wiener-Hammerstein Multi-objective Evolutionary Algorithm (WH-MOEA).

WH-MOEA is an improvement of WH-EA algorithm [122] to address the identification of Wiener, Hammerstein and Wiener-Hammerstein models under a multi-objective optimisation approach. WH-MOEA adopts some genetic operators and coding from the WH-EA algorithm, whilst the structure and functioning are acquired from ev-MOGA algorithm [44]. Arguably, WH-MOEA is therefore an elitist multi-objective evolutionary algorithm based on the concept of ϵ -dominance [55] for identification of block oriented models (Wiener, Hammerstein, and Wiener-Hammerstein).

Balancing convergence and diversity is guaranteed in WH-MOEA thanks to features inherited from ev-MOGA. Basically, ev-MOGA tries to ensure that \mathbf{X}_p^* converges toward the Pareto set in a smart distributed manner along the Pareto front with limited memory resources. To do that, 1) the prevalence of dominant solutions in the population Pop and the archive A with respect to the dominated solutions is guaranteed, and 2) the objective space is split into a fixed number of boxes, and only one solution can be stored in each box. This avoids the need to use other clustering techniques to obtain adequate distributions, and so considerably reduces the computational cost [55, 44].

4.4.1 WH-MOEA algorithm description.

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

1. Pop is the main population. It explores the searching space during the algorithm iterations. Its size is $Nind_P$.
2. Archive A is used to store the Pareto front approximation $\mathbf{f}(\mathbf{X}_p^*)$. Its size $Nind_A$ is variable but bounded because, as already mentioned, the target space is divided into a finite number of boxes where, at most, there can only be one solution.
3. Auxiliary population G is used to store new solutions created in each iteration of the optimisation process. Its size is $Nind_G$, which must be an even number.

The pseudocode of the WH-MOEA algorithm is shown in Algorithm 4, whilst its main steps are detailed as follows:

Algorithm 4 Pseudocode of WH-MOEA.

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

Step 1. $Pop(0)$ is initialised with $Nind_P$ individuals (candidate solutions).

First individual is coded from the initial linear model, as indicated in [122] and [121]. Next, this first individual undergoes all mutation operations $Nind_P - 1$ times to give rise to the rest of the population.

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

Step 6. Function *create* creates new individuals of $G(g)$ by using procedure and genetic operators described in Subsection 4.4.2.

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

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

4.4.2 Description of create function and genetic operators.

Function *create* creates $G(g)$ by using Algorithm 5. This procedure is repeated $Nind_G/2$ times until $G(g)$ is filled. A new individual is created by altering the genetic information of an existing individual in the main population $Pop(g)$, which is selected at random with r_1 . The genetic alteration is given by the mutations M.1, M.2, M.3, M.4, M.5, and the crossover C.3. In this last genetic operation the individual selected from $Pop(g)$ and an individual from the archive $A(g)$ randomly selected by r_2 exchange information.

Algorithm 5 Pseudocode of *create* function.

```

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

```

Algorithm 6 Pseudocode of Mutation M.5.

```

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

```

Each genetic operation acts on a certain portion of an individual's information. As can be seen in Algorithm 5, to expand diversity and avoid premature convergence, not all genetic operations are applied at the same time. A random process and control parameters $\gamma(g)$, η_{min} , $\eta(g)$, ξ , P_{cr} , and P_{nl} decide which genetic operators should be applied. The random number $r_{pzn} \in (0, 1]$ is compared with $\gamma(g)$ to decide if the location of a pole/zero is mutated with M.1 or whether the location of a breakpoint is mutated with either M.2 or M.3. This last selection depends on the random number $r_{mm} \in (0, 1]$ and the control parameter $\eta(g)$. On the other hand, parameter ξ defines the probability that an information alteration will occur in \mathcal{C} (binary code for the classification of the dynamics). This alteration is handled by mutation M.4, and also its occurrence depends randomly on the value of $r_c \in (0, 1]$. Similarly, mutation M.5 and crossover C.3 are randomly selected through random numbers $r_{nl} \in (0, 1]$ and $r_{cr} \in (0, 1]$ respectively. Parameter P_{nl} defines the probability that mutation

Algorithm 7 Pseudocode of Crossover C.3.

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

M.5 occurs, while parameter P_{cr} defines the probability that crossover C.3 occurs.

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

During the first generations, $\gamma(g)$ is close to one which gives a high probability of a mutation occurring on the segment that contains genetic information on the position of the breakpoints; the probability of a mutation occurring on the segment containing genetic information on the location of the poles and zeros is therefore low. Parameter $\gamma(g)$ decreases as the generations pass so that in the last generations the algorithm will modify both portions of genetic information with equal probability. The way the $\gamma(g)$ parameter works is justified by the fact that static non-linearity is entirely unknown, so the algorithm should focus more on this portion of genetic information during the first generations. As non-linearity takes shape, the algorithm will increase the probability that new positions for the poles and zeros will also be explored, so that the dynamics will be refined. Similarly, the control parameter $\eta(g)$ allows the probability of selecting between M.2 and M.3 mutations to be variable. During the first generations, mutation M.3 is not so necessary since this genetic operation concentrates breakpoints in the curvatures to achieve higher accuracy. As long as the non-linearity does not take shape, the concentration of points in the curves will have no significant effect. The mutations M.1, M.2, M.3, and M.4 are inherited from the WH-EA, therefore further information on how these genetic operations work as well as the calculation of the parameters $\gamma(g)$ and $\eta(g)$ can be found in Chapter 2. In this section, a brief summary of these genetic operations is presented. In addition, the new genetic operators (mutation M.5 and crossover C.3) are briefly explained.

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

$$p^{new} = p^{act} + N_{zp}(0, \sigma^2(g)) \quad (4.18)$$

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

In each generation g , the standard deviation is calculated with the following expressions:

$$\sigma^2(g) = \frac{\Delta_s}{100} \left(\frac{\sigma_{ini}^2}{\sqrt{1 + g * \sigma_{ratio}^2}} \right) \quad (4.19)$$

$$\sigma_{ratio}^2 = \frac{(\sigma_{ini}^2 / \sigma_{end}^2)^2 - 1}{MaxGen - 1} \quad (4.20)$$

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

- Mutation M.2. A pair of \mathbf{B} genes are randomly selected to be modified. These genes are matched and represent the coordinates of a breakpoint. The new values of the selected genes are calculated with the same procedure used in the mutation M.1. Two random numbers with Gaussian distribution ($N_v(0, \sigma^2(g))$ and $N_w(0, \sigma^2(g))$) are thus required. $N_v(0, \sigma^2(g))$ is used to mutate the gene of the abscissa of the selected breakpoint, while $N_w(0, \sigma^2(g))$ is used to mutate the gene of the ordinate of the same breakpoint. As in mutation M.1, $\sigma^2(g)$ varies from σ_{ini}^2 to σ_{end}^2 to control the aggressiveness of the mutations and fine-tune the breakpoints.

- Mutation M.3. This genetic operation is applied to \mathbf{B} and allows the breakpoints to jump to each other so that they can concentrate on the curvatures. The two genes that define the position of a breakpoint are randomly selected. The selected breakpoint must jump to a new location which will be in a segment defined by two other breakpoints. This segment is also chosen randomly. The new abscissa is computed as the midpoint between the two breakpoints that form the segment over which the point will jump, while the new ordinate will be calculated using a quadratic interpolation and information on the breakpoints near the location where the jump occurred. Quadratic interpolation will help to make a smooth transition from a point when it jumps to a specific segment.
- Mutation M.4. This genetic operation is applied to \mathbf{C} and generates new combinations of poles and zeros to give rise to the dynamics of the two LTI subsystems. Each time this mutation is required, a new binary code subject to (4.11)-(4.14) is randomly generated in \mathbf{C} .
- Mutation M.5. This genetic operation is applied to \mathbf{B} . Under certain conditions it can increase or reduce the number of breakpoints used to represent static non-linearity. The way this genetic operation works is synthesised in Algorithm 6. A possible increase or decrease is determined by the random number $P_{as} \in (0, 1]$. An increase of one breakpoint will occur as long as the individual selected to mutate does not contain in \mathbf{B} the maximum amount of breakpoints allowed (n_{max}). If an increase is required, a new breakpoint will appear on a randomly selected segment. As with mutation M.3, the abscissa of the new breakpoint will be calculated as the midpoint between the two breakpoints that define the selected segment, while the ordinate will be calculated using a quadratic interpolation and the coordinates of three neighbouring breakpoints. For a decrease to occur, two conditions must be met in addition to $P_{as} > 0.5$. The first condition is that \mathbf{B} must not contain the minimum number of breakpoints allowed (n_{min}), while, the second condition allows a decrease to occur as long as a breakpoint is located on a straight line, that is, in a redundant position. To verify this condition, the slopes of all segments joining two breakpoints will be calculated. The slope of each segment will be compared with the slope of the segment on the right. The two consecutive segments with the smallest difference in slope will contain the breakpoint that is likely to be eliminated. These two slopes will be compared again; if the absolute value of the difference is close to zero, the breakpoint that is common to the two segments will be removed. Closeness to zero is quantified by a fixed value of $1E - 3$. This procedure aims to eliminate unnecessary breakpoints.

- Crossover C.3. This genetic operation allows an individual from the main population $Pop(g)$ to exchange genetic information with an individual from the archive $A(g)$ (front of Pareto), for which two random integers are generated. The first integer ($r_1 \in [1, size(Pop(g))]$) is used to randomly select the individual of $Pop(g)$, while the second integer ($r_2 \in [1, size(A(g))]$) is used to randomly select the individual of $A(g)$. Each individual of both $Pop(g)$ and $A(g)$ has three portions of genetic information (**P**, **B**, and **C**), however, the exchange of information between the two individuals will be of only one portion, for which another random integer is generated ($C_{ag} \in [1, 3]$). This number will decide which portion of the genetic information will be exchanged (see Algorithm 7).

4.5 Using WH-MOEA to compare design concepts: a procedure

4.5.1 Comparing design concepts by the MOP approach

When dealing with a MOP design problem, it is quite common to consider different alternatives (design concepts), i.e. ideas on how to solve the problem. It is possible to define an independent optimisation problem for each of these design concepts [68] and when they share the same objectives they can easily be compared directly in the objective space by simply comparing their respective Pareto fronts. For example in [104], this idea is used to compare alternative slide model control structures, in [45] to compare different loop pairings and in [82] to analyse the performance of different battery models.

Figure 4.1 shows an example in which the Pareto fronts of three design concepts are compared in a two-dimensional objective space. Notice that as design concept DC2 completely dominates design concept DC3, concept DC2 will be preferred over concept DC3. However, DC2 does not completely dominate concept DC1 or vice versa. If the designer has a preference for solutions in Zone 1, solutions from concept DC2 would be selected, as in this zone concept DC2 dominates concept DC1. Whilst if Zone 2 was preferred, the designer should choose solutions from concept DC1. This illustrative example shows how interesting it is to use the MOP optimisation approach when comparing possible design concepts. In the end, the designer would choose the final solution taking into account the dominance between concepts and his/her specific preferences.

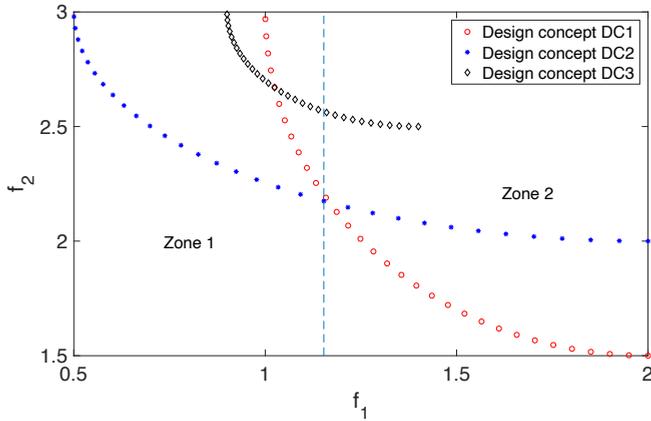


Figure 4.1: Comparison of three design concepts in a bi-objective space.

In this approach, an initial linear model is used as a starting point for nonlinear estimation. Most of the time, the model is selected from a ranking based on a specific criterion. However, under the assumption that there are any other useful criteria giving more candidates, the idea of design concepts arises so more than one initial linear model should be tested. Then, each linear structure will give rise to a design concept. Furthermore, if you sort the list of initial candidates by a criterion, several models with similar values could be obtained and therefore one might add other criterion to perform the initial selection. In these case, it is worth to evaluate more than one linear structure.

4.5.2 Procedure definition

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

Step 1 - Establish a set of candidates using a standard linear identification method:

Since WH-MOEA can fine-tune the initial linear model, a classical step test can be used. The ranking can be made according to the structure of the models (number of poles and zeros) and one or more performance criteria. For example, the mean squared error (MSE) could be used; however, to avoid

Table 4.1: WH-MOEA tuning parameters

Parameter	Description
$MaxGen$	Maximum number of generations.
$Nind_P$	Population size.
$Nind_G$	Size of the auxiliary population G .
n_box_1	Number of divisions for the objective f_1 .
n_box_2	Number of divisions for the objective f_2 .
n_{max}	Maximum number of breakpoints
n_{min}	Minimum number of breakpoints
α	Minimum distance between two breakpoints on the abscissa axis
$\sigma_{ini}^2, \sigma_{end}^2$	Initial and final standard deviations for control of the aggressiveness of mutations M.1 and M.2
η_{min}	Minimum probability for selection of mutation M.2. The maximum probability for this selection is 1.
ξ	Probability to modify with mutation M.4 the genetic information related to the classification of poles and zeros.
P_{nl}	Probability to modify with mutation M.5 the number of breakpoints for static non-linearity
P_{cr}	Probability of crossover between an individual of Pop and an individual of A

over-modelling the MDL criteria or any others, mentioned in Section 4.1, can be useful.

Step2 - Choose the design concepts:

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

Step3 - Prepare a multi-step excitation signal and performs an experiment with the process under identification:

The input signal must ensure that the system output reaches steady state. For static non-linearity to be appropriately captured, different step amplitudes must allow a scan of the entire process operating range. A minimum distance between two consecutive steps must be considered [121] to highlight static non-linearity. Finally, excite the process under identification with this signal and record its output for a given sampling period.

Step 4 - Set the bounds:

From the minimum and maximum values of the input (u_{min} and u_{max}) and output (y_{min} and y_{max}) signals, define the search space bounds for static non-linearity using equations (4.1) to (4.4).

Step 5 - Run WH-MOEA for each design concept:

Table 4.1 shows the list of WH-MOEA parameters that must be previously assigned to carry out each run. The bounds \mathbf{P}^{min} and \mathbf{P}^{max} must also be previously defined as a function of the sensitivity of the poles and zeros. At the end of each WH-MOEA execution a Pareto front is obtained.

Step 6 - Perform a decision making analysis:

Each Pareto front contains a set of models with different performances. For trade-off analysis, draw all the Pareto fronts on the same graph. Evaluate selected models on the estimation and validation data sets for a final decision.

4.6 Evaluation of WH-MOEA

In this section WH-MOEA will be evaluated on a non-linear system consisting of a continuous stirred-tank reactor (CSTR). In this application, a trade-off analysis will not be performed by testing different design concepts. The intention of this section is only to test the potential of the algorithm to generate a set of models with different complexities and accuracies framed within a Pareto front.

The schematic diagram of the CSTR process is shown in Figure 4.2. In this process, an irreversible exothermic reaction between two chemical species (A and B) takes place. The reaction can be written as $A \rightarrow B$. It is assumed that the liquid volume V in the reactor is kept constant by an overflow line and the system is cooled by a single coolant stream. The process model is described by the following two nonlinear differential equations [52, 103, 73]:

$$\dot{C}_A = \frac{q}{V}(C_{Af} - C_A) - k_0 C_A \exp\left(\frac{-E}{RT}\right) \quad (4.21)$$

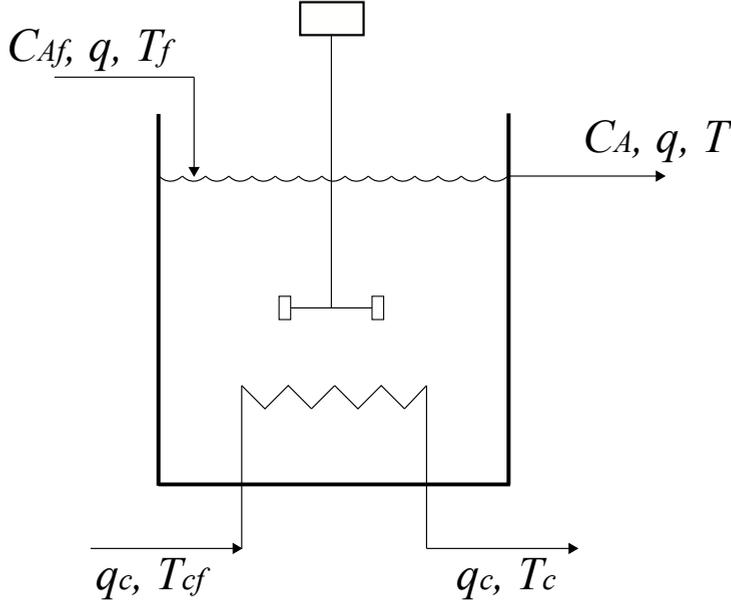


Figure 4.2: Schematic diagram of CSTR process

$$\begin{aligned} \dot{T} = & \frac{q}{V}(T_f - T) + \frac{-\Delta H k_0 C_A}{\rho C_p} \exp\left(\frac{-E}{RT}\right) + \dots \\ & \dots \frac{\rho_c C_{pc}}{\rho C_p V} q_c \left[1 - \exp\left(\frac{-hA}{q_c \rho_c C_{pc}}\right)\right] (T_{cf} - T) \quad (4.22) \end{aligned}$$

The concentration of the outlet fluid C_A and the reactor temperature T vary depending on the coolant flow rate q_c . The rest of parameters are considered constant and their values are defined in Table 4.2.

The CSTR is a complex nonlinear process presenting multiplicity behaviour with respect to the coolant flow rate. For the nominal values in Table 4.2, there are two stable operation regions (Figure 4.3). The CSTR can be brought to the lower stable region with the initial conditions: $q_c = 103.41 \text{ l/min}$, $C_A = 8.36E - 2 \text{ mol/l}$, $T = 440.2 \text{ K}$, while the upper region can be reached with: $q_c = 103.41 \text{ l/min}$, $C_A = 0.950 \text{ mol/l}$, $T = 353.20 \text{ K}$. The CSTR is unstable if C_A is between 0.14 mol/l and 0.92 mol/l . In this chapter, the CSTR is modelled

Table 4.2: CSTR parameter values.

Symbol	Description	Value	Unit
q	Process flow rate	100	$l\ min_{-1}$
C_{Af}	Feed concentration	1	mol/l
T_{cf}	Feed temperature	350	K
T_f	Inlet temperature	350	K
V	CSTR volume	100	l
hA	Heat transfer	$7.0E5$	$cal\ min^{-1}\ K^{-1}$
k_0	Frequency factor	$7.2E10$	min^{-1}
E/R	Activation energy	$9.95E3$	K
ΔH	Heat of reaction	$-2E5$	cal/mol
ρ, ρ_c	Liquid densities	$1E3$	g/l
C_p, C_{pc}	Specific heats	1	$cal\ g^{-1}\ K^{-1}$

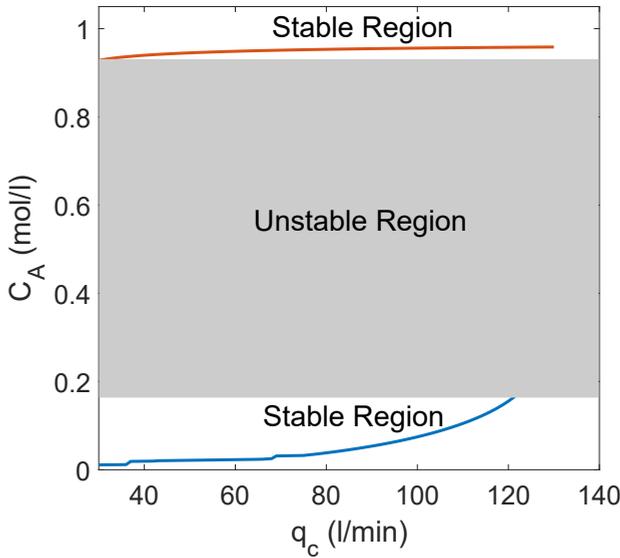


Figure 4.3: CSTR operating ranges

on the stable upper region, where there are very soft changes in dynamics and a non-linearity of static type predominates.

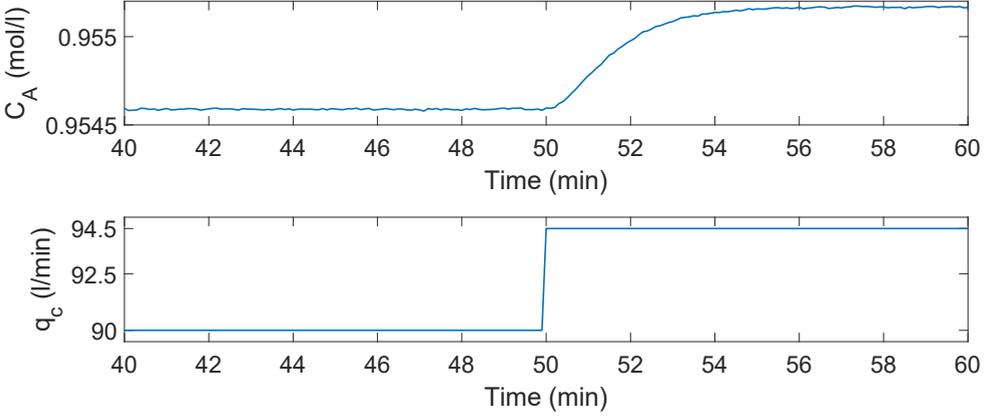


Figure 4.4: Input and output data for linear estimations (CSTR process). Top: Output concentration C_A . Bottom: coolant flow rate q_c .

To get data of the CSTR process, a model was developed in Simulink using equations 4.21 and 4.22 and parameters of Table 4.2. In addition, Gaussian noise with a power of $-110db$ was added on the system output. To analyse the structure of the linear dynamics, the concentration C_A was brought to $0.954mol/l$ with $q_c = 90l/min$. This operating point was arbitrarily selected within the region of interest. On this operating point, a change in q_c to $94.5l/min$ was simulated with a step signal. Input (q_c) and output (C_A) data were recorded with a sampling period of $0.1min$. Figure 4.4 shows the input and output data set used for the linear estimates.

To avoid problems with initial conditions, offset was removed from data. Different linear structures were estimated from first to third order considering only strictly proper systems. The results of the estimates are shown in Table 4.3.

To show the potential of WH-MOEA to generate a Pareto front, only one linear structure will be selected for the nonlinear estimation (in this case, the best model according to the MDL criteria, which according to Table 4.3 is G_{2p} (two-poles)). Figure 4.5 shows the performance of this model taking as a reference the output of the simulated process. The transfer function of this model is shown below:

$$G1_{CSTR} = \frac{1.757E - 4}{(s + 1.302)(s + 1.048)} \quad (4.23)$$

Table 4.3: Ranking of estimated linear models for CSTR process

Structure	MSE	MDL
G_{1p}	3.2028E-10	1.7363E-10
G_{2p}	1.0975E-11	6.1697E-12
$G_{1z/2p}$	1.0975E-11	6.3430E-12
G_{3p}	1.0922E-11	6.3434E-12
$G_{1z/3p}$	1.0989E-11	6.5710E-12
$G_{2z/3p}$	1.0908E-11	6.8674E-12

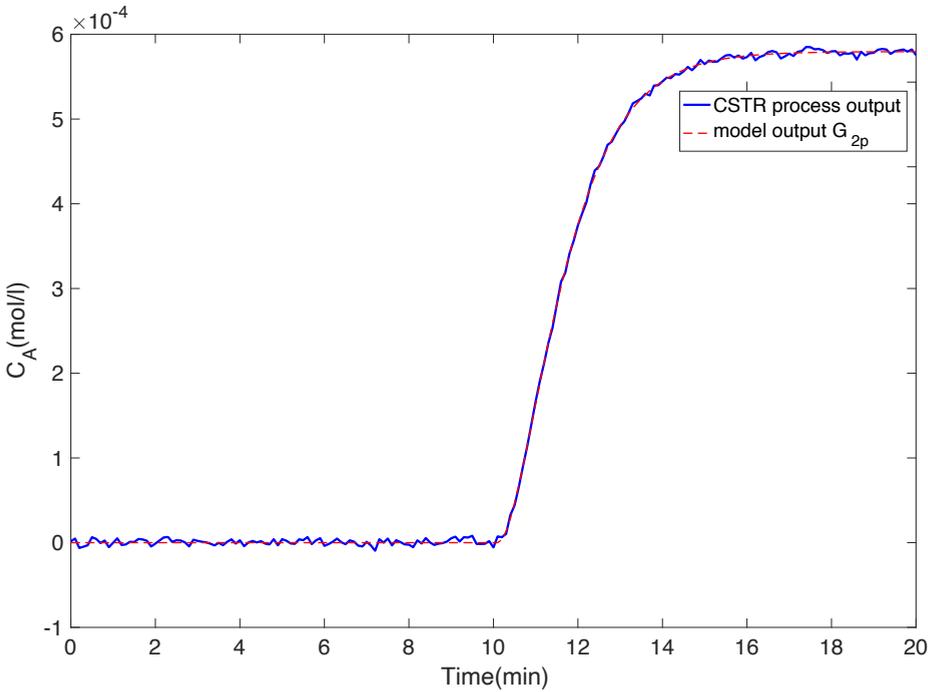


Figure 4.5: Performance of best linear model (G_{2p}) for CSTR process.

To start the identification process, two multi-step signals were generated, one for identification and another for validation purposes. A variation range for q_c between $40l/min$ and $200l/min$ was considered. Steps duration in both signals was $20min$ and their amplitude were handled randomly within the entire range of q_c . The minimum step amplitude was limited so that it is not

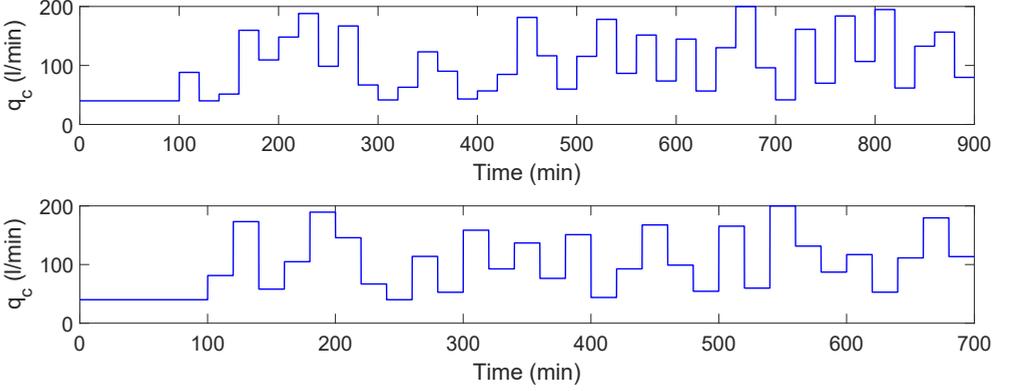


Figure 4.6: Estimation (top) and validation (bottom) input signals for nonlinear identification of CSTR process.

less than $10l/min$. The estimation and validation signals were designed with 40 and 30 different amplitudes respectively. To avoid transient effects, at the beginning of the input signals, 1000 samples with the lowest amplitude were added (first $100min$). The input signals for both estimation and validation are shown in Figure 4.6.

The CSTR process was excited with both signals separately. Input and output data for both cases were sampled with a period of $0.1min$. Next, the minimum and maximum values of the input and output signals from the estimation data set were obtained ($u_{min} = 40$, $u_{max} = 200$, $y_{min} = 0.9396$ and $y_{max} = 0.9614$). The vertical search space for all breakpoints was defined with y_{min} and y_{max} using (4.1) and (4.2) respectively:

$$\mathbf{W}^{min}(\mathbf{i}) = 0.9396 \quad for \quad i = 1 \dots n \quad (4.24)$$

$$\mathbf{W}^{max}(\mathbf{i}) = 0.9614 \quad for \quad i = 1 \dots n \quad (4.25)$$

whilst u_{min} and u_{max} values were used internally by the algorithm to define the variable limits \mathbf{V}^{min} and \mathbf{V}^{max} using (4.3) and (4.4) respectively.

The configuration parameters for WH-MOEA are shown in Table 4.4. In addition, initial and final standard deviations for mutations were set to 20 and 1, respectively. On the other hand, vector \mathbf{P}_1^0 was coded with $nc = 0$,

Table 4.4: Summary of WH-MOEA parameter settings for nonlinear estimation of the CSTR process

Parameter	Value	Parameter	Value
$MaxGen$	$1.0E6$	n_{min}	4
$Nind_P$	3.04	α	1.904
$Nind_{GA}$	4	ξ	0.25
n_box_1	50	P_{nl}	0.33
n_box_2	10	P_{cr}	0.33
n_{max}	14	η_{min}	0.30

Table 4.5: Optimisation results for CSTR process.

Model	f_1	f_2
M_1^{2p}	$1.026E - 4$	16
M_2^{2p}	$1.030E - 4$	14
M_3^{2p}	$1.166E - 4$	10
M_4^{2p}	$1.242E - 4$	9
M_5^{2p}	$1.332E - 4$	8
M_6^{2p}	$1.757E - 4$	7
M_7^{2p}	$2.683E - 4$	6

$nr = 0$, $mc = 0$, $mr = 2$, and pole locations presented in (4.23), while the bounds to explore new locations for both poles were set in ± 0.5 .

To compute the objective f_1 (MAE), the first 80 minutes were neglected to avoid transient effects. At the end of generations, 7 Wiener-Hammerstein models with the same dynamic distribution were obtained. The Pareto front obtained is shown in Figure 4.7, while the optimisation results are summarised in Table 4.5. In addition, this table also specifies how each model has been labeled.

According to the Figure 4.7, the most accurate model is M_1^{2p} with a MAE (f_1) of $1.026E - 4$. However, this model is also the most complex with $f_2 = 16$ (14 points used to represent the static non-linearity). On the other hand, the simplest model is M_7^{2p} with a MAE of $2.683E - 4$, but only 4 points used to represent the non-linearity.

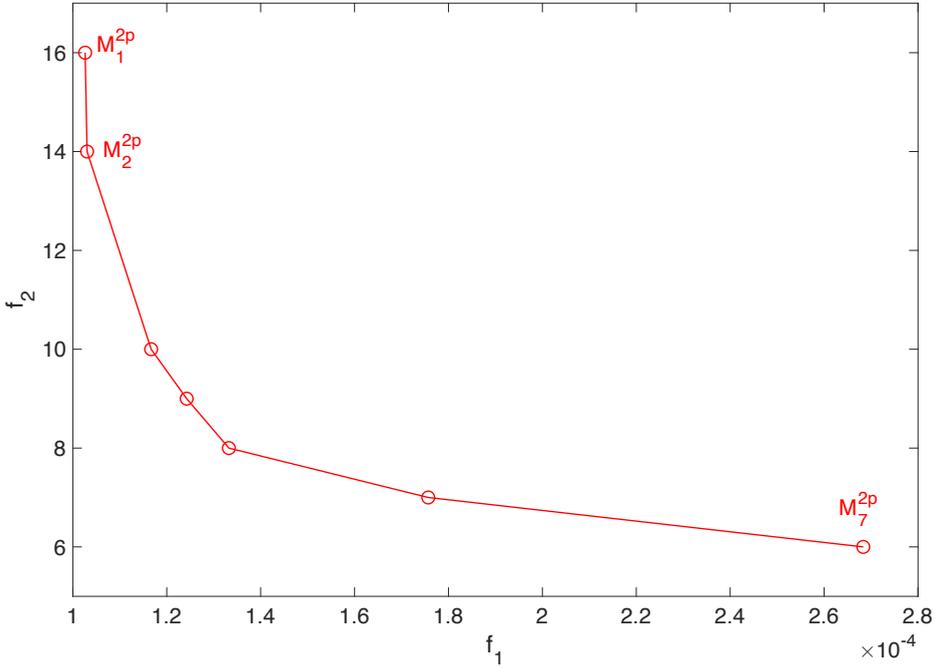


Figure 4.7: Pareto front for CSTR estimation.

The Pareto front obtained reveals the importance of using WH-MOEA. Notice that, models M_1^{2p} and M_2^{2p} have practically the same precision, however, M_2^{2p} has a lower level of complexity, as it represents the static non-linearity with two points less. Further, it can be concluded that it does not make sense to use more than 12 points to represent the static non-linearity because the increase in accuracy is negligible.

With other identification algorithms (including WH-EA), the choice of the number of points to represent the non-linearity can be an inconvenience, since in many identification problems it is not possible to know in advance its complexity. An exaggerated number of points will result in an over-parameterized model with a high degree of complexity. On the other hand, a model with few points will have insufficient accuracy. To solve this problem, the user must perform several optimization tests until finding the right number of points. With WH-MOEA, this uncertainty is naturally eliminated, because in a single optimization test the generated Pareto front will offer a set of models with different numbers of points to represent the static non-linearity. With a

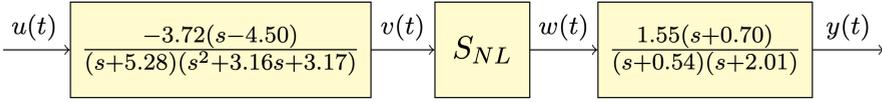


Figure 4.8: Numerical example: Wiener-Hammerstein structure.

simple analysis the user will be able to select the appropriate model taking into consideration the accuracy and model complexity.

4.7 Application of WH-MOEA and trade-off analysis using design concepts

To show the effectiveness and usefulness of this approach, two application examples are included in this section. The first consists of a numerical example, a process with a pure Wiener-Hammerstein structure, while the second is a real thermal process based on a Peltier cell.

WH-MOEA parameters were set to the same values for both identification problems: $\xi = 0.25$; $P_{nl} = 0.33$; $P_{cr} = 0.33$ and $\eta_{min} = 0.30$. The initial and final standard deviations for mutations were set to 20 and 1, respectively.

4.7.1 Application 1: Numerical example.

Figure 4.8 shows the Wiener-Hammerstein process used for this example. Static non-linearity (S_{NL}) consists of a sigmoid hyperbolic tangent function "tansig" (4.26), which symmetrically saturates large values of the independent variable. Also, Gaussian noise with a power of $-40db$ was added on the system output to emulate a real situation.

$$S_{NL} = 50[1 + \text{tansig}(0.07(v(t) - 50))] \quad (4.26)$$

Step 1 - Establish a set of candidates using a standard linear identification method:

First, an experiment for initial linear identification was designed. Since input range was between 0 and 100 units a constant signal with amplitude 50 units was injected into the system until it reached steady-state. On this operating point, the process was excited with a small input change of 2.5 units and a duration of 15s (Figure 4.9). During the experiment, input and output data

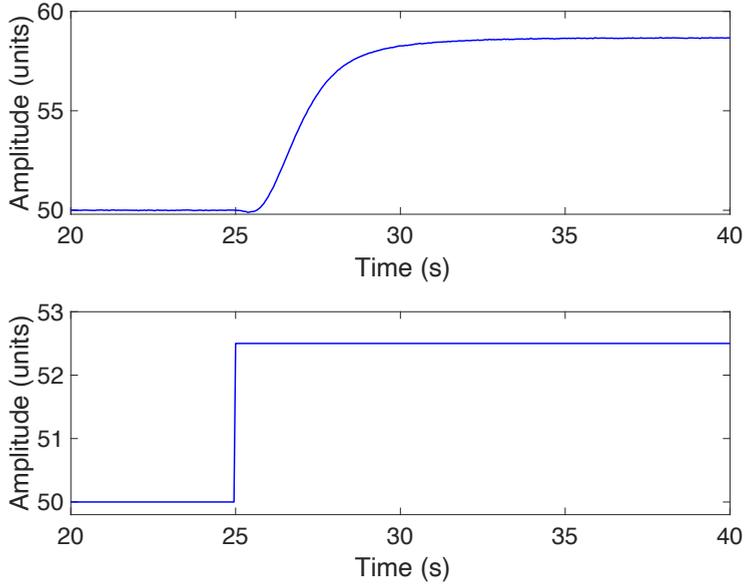


Figure 4.9: Input and output data for linear estimations of WH process. Top: Simulated system output. Bottom: Input signal.

were recorded with a period of $50ms$. Offset was removed from the data to avoid problems with initial conditions.

Fifteen initial linear models were estimated from 3^{rd} to 6^{th} order considering only strictly proper systems and no more than three zeros. For each structure, a linear model was estimated using the Matlab System Identification Toolbox [63] with the *tfest* command. The *MSE* and the modified *MDL* were calculated on the estimation data set. The results of these estimates are shown in Table 4.6. According to the modified MDL criteria, the best model is two zeros - five poles ($G_{2z/5p}$) which is consistent with the actual dynamics of the WH process under consideration.

Step2 - Choose the design concepts:

According to the results in Table 4.6, the linear structure $G_{2z/5p}$ might be considered for nonlinear estimation; however, as mentioned in Section 4.5, others can also be considered. Models with six poles would not be a good option since there are other less complex models that can achieve similar performance.

Table 4.6: Set of candidates of initial linear models for WH process. Model $G_{2z/6p}$ was excluded since its MDL value was very high.

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

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

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

According to this analysis, linear structures $G_{1z/3p}$, $G_{2z/3p}$, $G_{1z/4p}$, $G_{2z/4p}$, $G_{3z/4p}$, $G_{1z/5p}$, and $G_{2z/5p}$ would be good options for nonlinear estimation. Each of these structures will give rise to a design concept for nonlinear estimation. Table 4.7 shows the poles and zeros of each design concept, while Figure 4.10 shows a comparison of their performance. For convenience sake, from now on each design concept will use a nomenclature as indicated in the first column of Table 4.7. This nomenclature is preceded by the characters "DC" followed by an indication of the number of poles and zeros of the linear structure.

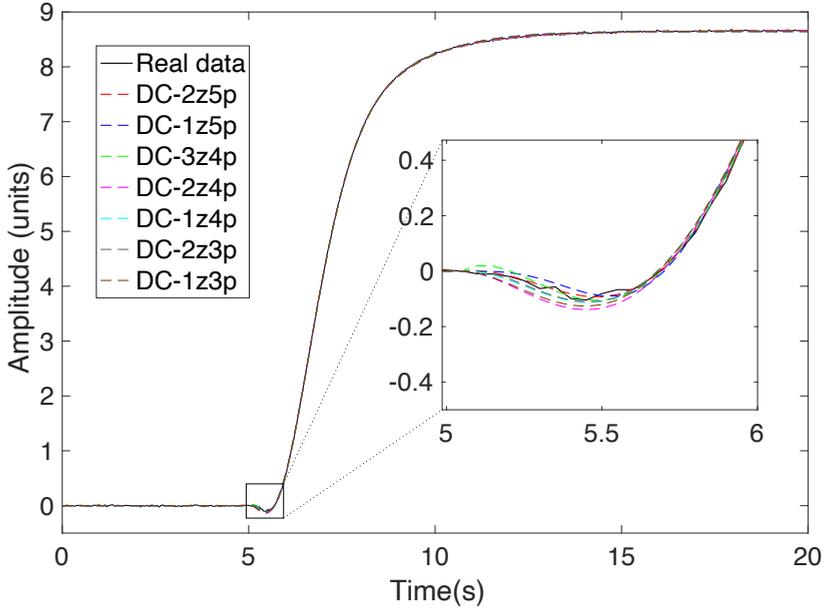


Figure 4.10: Performance of linear models (design concepts) chosen for nonlinear estimation of WH process. To distinguish between models, a magnification is depicted on a portion of data.

Step3 - Prepare a multi-step excitation signal and performs an experiment with the process under identification:

Two multi-step signals were designed, one for estimation and one for validation purposes (see Figure 4.11). The amplitude of the steps in both signals was handled randomly, lasting 25s each. The minimum step amplitude was set to 15 units. The estimation signal was designed with 50 steps, while the validation signal was designed with 30 steps. The proposed WH process was simulated twice in Matlab using estimation and validation multi-step inputs. In both experiments, process input and output were recorded with a sampling period of 50ms.

Step 4 - Set the bounds:

From the estimation data set, the minimum and maximum values of the input and output signals were obtained ($u_{min} = 0$, $u_{max} = 100$, $y_{min} = -0.024$ and $y_{max} = 99.933$). The vertical search space for all breakpoints was defined with y_{min} and y_{max} using (4.1) and (4.2) respectively:

Table 4.7: Selected initial linear models (design concepts) for nonlinear identification of WH process.

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

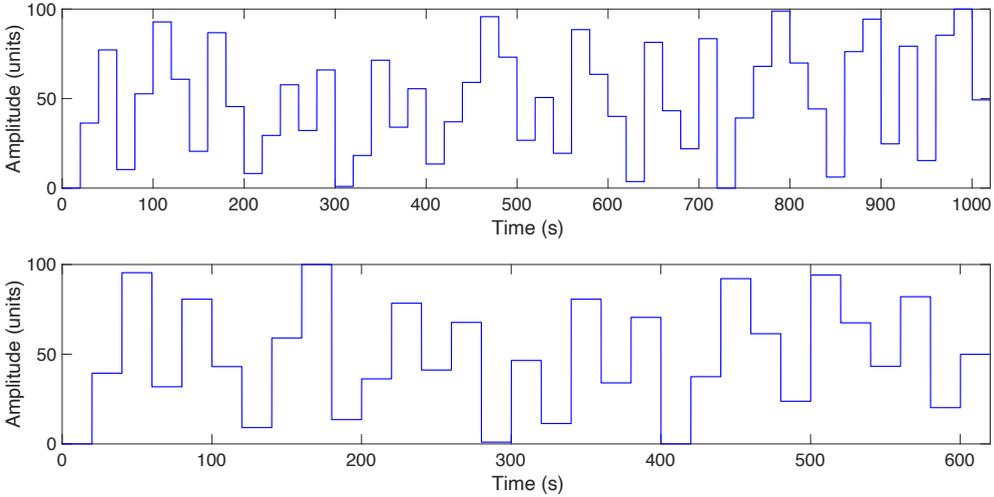


Figure 4.11: Estimation (top) and validation (bottom) input signals for nonlinear identification of WH process.

$$W^{min}(i) = -0.024 \quad \text{for } i = 1 \dots n \quad (4.27)$$

$$W^{max}(i) = 99.933 \quad \text{for } i = 1 \dots n \quad (4.28)$$

whilst u_{min} and u_{max} values were used internally by the algorithm to define the variable limits V^{min} and V^{max} using (4.3) and (4.4) respectively. It should be noted that the horizontal search space of each breakpoint is variable

Table 4.8: Summary of WH-MOEA parameter settings for nonlinear estimation of WH process.

Parameter	Value	Parameter	Value
$MaxGen$	$2.0E6$	n_box_2	16
$Nind_P$	$5.0E4$	n_{max}	26
$Nind_G$	4	n_{min}	10
n_box_1	50	α	0.641

and depends on the position of the neighbouring breakpoints.

Step 5 - Run WH-MOEA for each design concept:

According to selected design concepts (Table 4.7), seven bi-objective optimisation problems were stated as in Subsection 2.3. WH-MOEA was thus executed seven times, each one fed with a different initial linear model (design concept). The configuration parameters in Table 4.8 were used for all the executions. Table 4.9 shows how the first individual in the population (\mathbf{P}_1^0) was coded for each design concept and the lower (\mathbf{P}^{min}) and upper (\mathbf{P}^{max}) exploration bounds to refine the locations of the poles and zeros (defined individually around each pole or zero). It should be borne in mind that there is no recipe to set precisely the bounds but, the higher the interval the more exploration (slower algorithm convergence). A good alternative is to set the bounds based on the sensitivity of each pole or zero. Those closest to the imaginary axis are more sensitive to changes so their exploration bounds may be small, while a greater degree of freedom can be given to the poles and zeros furthest from the imaginary axis.

After the seven WH-MOEA runs, a total of 82 different models were obtained. Table 4.10 shows the number of models obtained in each design concept. This table also specifies how each model has been labelled.

Step 6 - Perform a decision making analysis:

To make an effective analysis of the results obtained, the Pareto fronts of all design concepts must be compared in a two-dimensional objective space. According to the number of design concepts stated for this example, seven Pareto fronts were obtained. Given the high precision achieved by the models of design concept $DC - 2z5p$, the corresponding Pareto front will be analysed separately. Figure 4.12 shows a graph of this Pareto Front which

Table 4.9: Coding of the first individual in the population and exploration bounds to refine the poles and zeros (design concepts for identification of the numerical example)

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

Table 4.10: Number of models obtained for each design concept.

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

contains 14 Wiener-Hammerstein models with the same dynamic structure. The distribution of the poles and zeros of these 14 models is consistent with the real dynamics of the proposed example. Undoubtedly, the fact of having the same structure as the real example has meant that the precision of all these models cannot be achieved by any model of the other fronts, even by the models of design concept $DC - 3z4p$, which has the same complexity level devoted to representing the dynamic part. In this case, the nonlinear estimates have highlighted the different degrees of accuracy that models of these design concepts can achieve. This difference is not so apparent on inspecting Table 4.6, which contains the results of the linear estimates, i.e. the MAE achieved by initial models $G_{2z/5p}$ and $G_{3z/4p}$ is $8.143E - 5$ and $9.315E - 5$ while their MDL is $4.781E - 5$ and $5.479E - 5$, respectively. Apparently, it seems that both models have similar performances. However, with nonlinear estimations, the importance of selecting the appropriate structure has been highlighted.

Looking at Figure 4.12, it can be clearly seen that the best model in terms of accuracy is M_1^{2z5p} . The precision (objective f_1) achieved by this model is $2.065E - 2$, while its complexity level (objective f_2) is 33 (2 zeros, 5 poles and 26 points for the static non-linearity). This model would undoubtedly be the best option in terms of precision, but it should be borne in mind that it is also the most complex model.

A great advantage of this approach is that the Pareto fronts allow the performance of different models to be compared. In addition, in each Pareto front it is also possible to analyse the contribution of the number of breakpoints in terms of precision. Looking at the Pareto front in Figure 4.12 it can be seen that the accuracy achieved by model M_3^{2z5p} is very close to that achieved by model M_1^{2z5p} . Considering the precision between M_{14}^{2z5p} and M_1^{2z5p} is 100%,

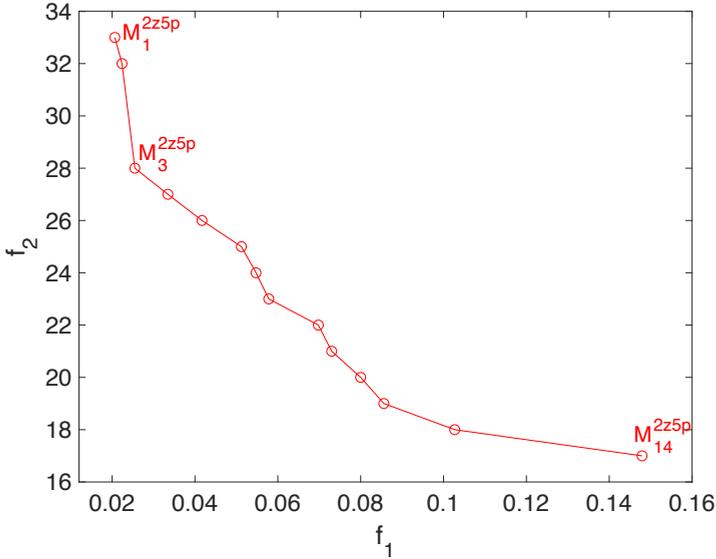


Figure 4.12: Pareto front corresponding to the design concept $DC - 2z5p$.

improvement of M_1^{2z5p} concerning M_3^{2z5p} is only 3.85%. In this regard, M_3^{2z5p} would be preferred since it has five points less.

Figure 4.13 compares the other Pareto fronts. Analysing these fronts one can notice that the best models in terms of precision are on the front of design concept $DC - 3z4p$, However, it should be noted that the models of this design concept have the same complexity in the dynamic part as the models of the design concept $DC - 2z5p$, which achieved better accuracy. It would therefore not make sense to select models from design concept $DC - 3z4p$, as there will always be an equally or less complex model of design concept $DC - 2z5p$ which will always be more accurate.

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

In the same way, the models of design concepts $DC - 1z4p$ and $DC - 2z3p$ have equal complexity in the dynamic part, so it would be interesting to compare

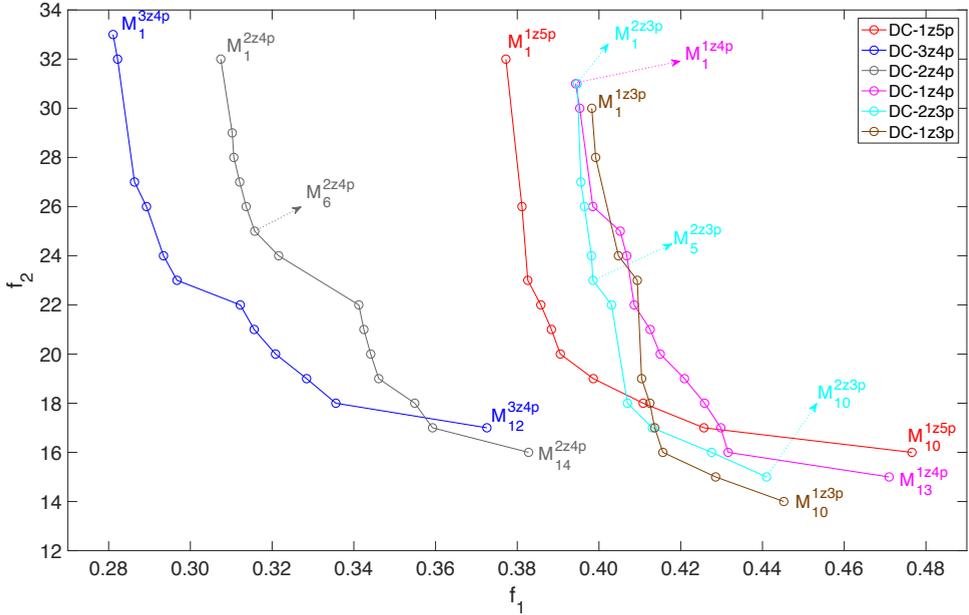


Figure 4.13: Comparison of Pareto fronts corresponding to design concepts $DC - 1z5p$, $DC - 3z4p$, $DC - 2z4p$, $DC - 1z4p$, $DC - 2z3p$, and $DC - 1z3p$

them. At first glance, the most complex model of each design concept (M_1^{1z4p} and M_1^{2z3p}) perform identically. However, as the number of static nonlinearity points decreases, the models of the $DC - 2z3p$ Pareto set become more accurate. It is thus not practical to select models from $DC - 1z4p$. Even if the precision of models of this front with a complexity level less than 21 were analysed, their performance would be worse than models of design concept $DC - 1z3p$, whose dynamic structure has one pole less.

According to the analysis presented, there are some design concepts whose models do not offer good performance. However, before discarding them, it would be essential to review their dynamic distribution. For example from the previous analysis, models from design concept $DC - 3z4p$ are not good candidates since other models have better precision and the same complexity in the dynamic part. Conversely, if models with a Wiener or Hammerstein structure had appeared as a result of WH-MOEA execution, any of them would have been very attractive especially for control applications¹.

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

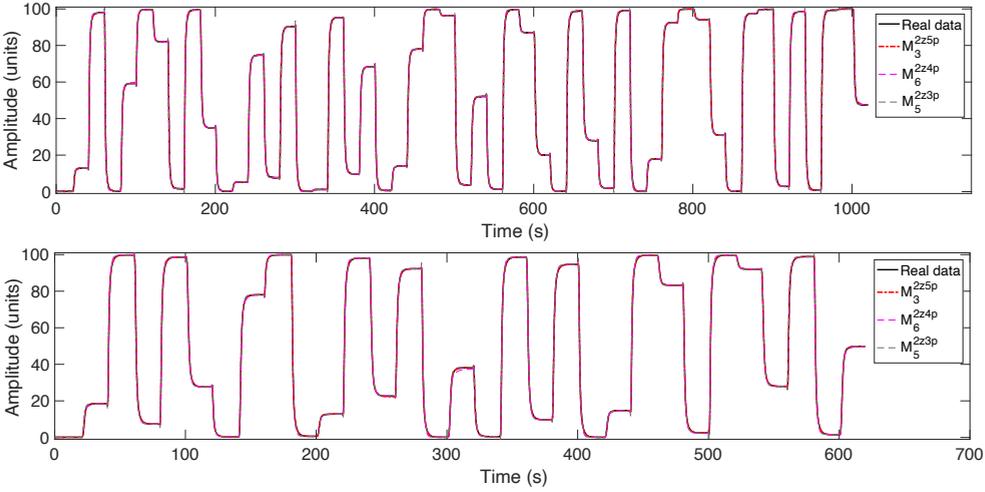


Figure 4.14: Performance of models M_3^{2z5p} , M_6^{2z4p} , and M_5^{2z3p} on the estimation (top) and validation (bottom) data sets.

In this regard, the fronts from which models could be selected to represent the system would be those of design concepts $DC - 2z5p$, $DC - 2z4p$, $DC - 2z3p$ and $DC - 1z3p$. If high accuracy were required, the M_3^{2z5p} model would be a very good option, while in case of requiring models with less complex dynamics, M_6^{2z4p} and M_5^{2z3p} would be good candidates. Analysing the Pareto Front of design concept $DC - 2z4p$, as the rate of increase in accuracy vs complexity in the M_6^{2z4p} to M_1^{2z4p} models decreases considerably, it does not make much sense to extend the points in static non-linearity. The same conclusion can be reached when analysing the Pareto front of design concept $DC - 2z3p$, in which the lines that join M_5^{2z3p} with M_1^{2z3p} are almost vertical, showing that the higher complexity model gives approximately the same precision. Table 4.11 shows the complexity and accuracy achieved by the three models indicated. The accuracy reached in the validation data set is also calculated (f_{1v}). Figure 4.14 shows their responses compared with the process under identification. Figure 4.15 shows the breakpoint locations representing static non-linearity.

As can be seen in Figure 4.15, distribution of the points in the three models is effective and the non-linearity has been satisfactorily captured. This is reflected in Figure 4.14, where there are no visible inaccuracies in steady state behaviour. Differences can be seen on the transients due to non-modelled dynamics in models M_6^{2z4p} and M_5^{2z3p} . This is shown in greater detail in Figure

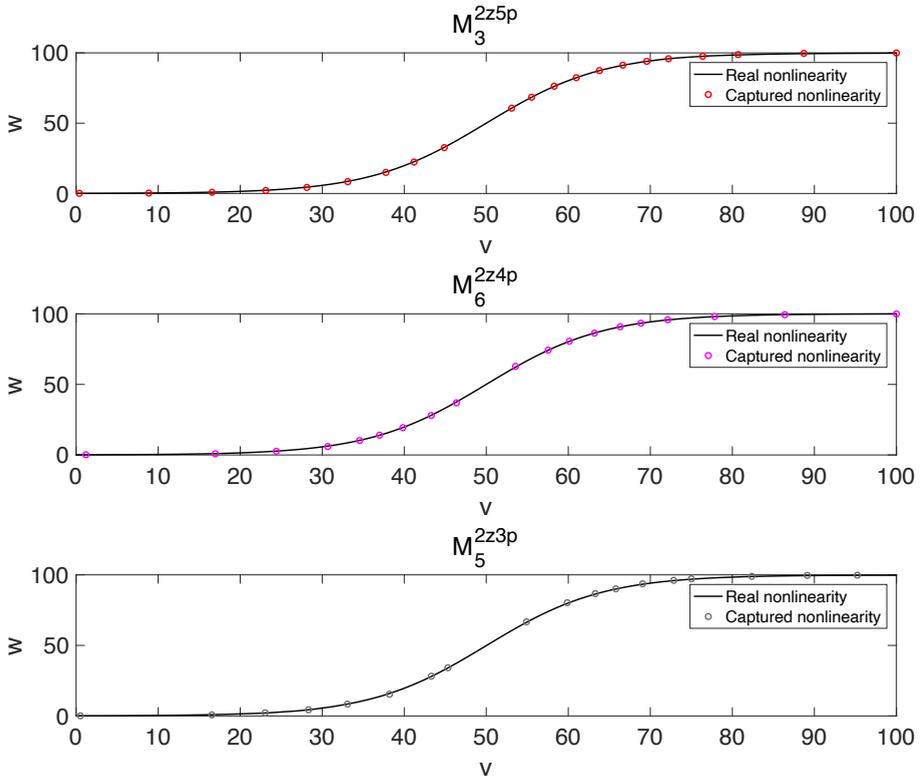


Figure 4.15: Comparison between real non-linearity and captured non-linearities for models M_3^{2z5p} , M_6^{2z4p} , and M_5^{2z3p} .

4.16, which highlights the differences. Notice that model M_3^{2z5p} reproduces the most accurate process dynamics.

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

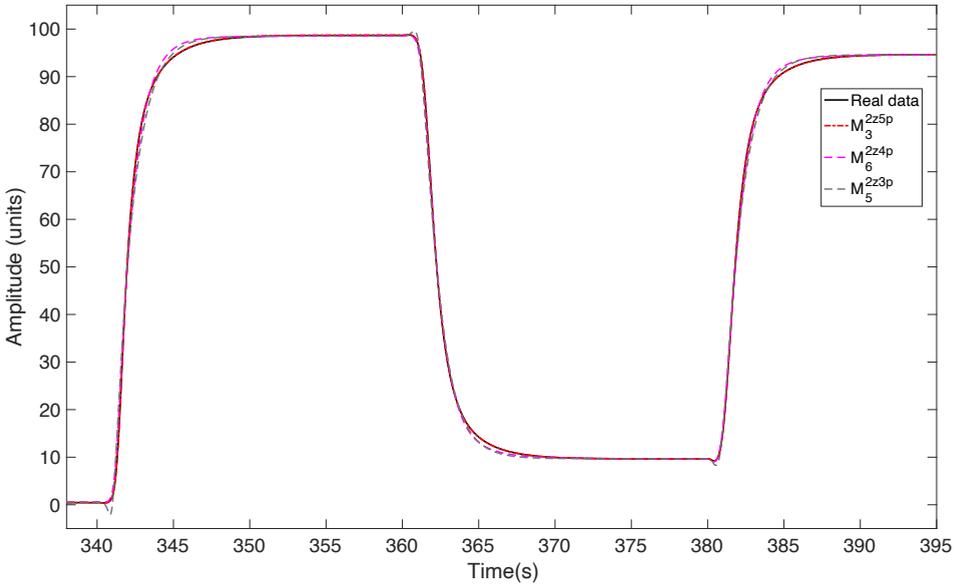


Figure 4.16: Enlargement of the transient stage of M_3^{2z5p} , M_6^{2z4p} , and M_5^{2z3p} models with validation data.

4.7.2 Application 2: Thermal process.

This subsection describes the nonlinear estimation under a multi-objective approach to the thermal process that was used in Chapter 3 to evaluate the unified approach for the estimation of Wiener, Hammerstein and Wiener-Hammerstein models. Therefore, more details on the process structure, lab test-bench, and the procedure for estimating the linear models can be found in Subsection 3.9.4. Below is the procedure to identify the nonlinear real process according to the multi-objective approach described in this chapter:

Step1 - Establish a ranking of candidates using a standard linear identification method:

For this application, the same ranking of models that was established in Subsection 3.9.4 will be used. The performance of the linear models are summarised in Table 3.7.

Table 4.11: Details of the precision and complexity of models M_3^{2z5p} , M_6^{2z4p} , and M_5^{2z3p} (m_{pz} indicates the total number of poles and zeros.)

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

Table 4.12: Selected initial linear models (design concepts) for nonlinear identification of thermal process.

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

Step2 - Choose the design concepts:

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

The remaining models G_{2p} , $G_{1z/2p}$, and G_{3p} have similar performance and any of these could be a good option for obtaining less complex nonlinear models. In this case, the linear structure $G_{1z/2p}$ will be selected. The location of the poles and zeros of the three selected structures for nonlinear estimation are shown in Table 4.12. As in the numerical example, each of these structures gives rise to a different design concept.

Step3 - Prepare a multi-step excitation signal and performs an experiment with the process under identification:

The input and output data are the same as those used for the estimation under the unified approach described in Chapter 3. As a reminder, the estimation signal was designed with 38 steps and the validation with 24 steps. The steps

Table 4.13: Summary of WH-MOEA parameter settings for nonlinear estimation of the thermal process

Parameter	Value	Parameter	Value
$MaxGen$	$1.2E6$	n_box_2	10
$Nind_P$	$4.0E4$	n_{max}	14
$Nind_G$	4	n_{min}	4
n_box_1	50	α	0.05

in both signals lasted 700s, and the amplitude changes were handled randomly within the entire range of variation of signal u_a (0 – 4.5V). To highlight the non-linearity of the process, the minimum step amplitude was constrained to 1.5V. The input signals for both estimation and validation tests are shown in Figure 3.20.

Step 4: Set the bounds.

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

Step 5: Run WH-MOEA for each design concept.

According to the selected design concepts, three bi-objective optimisation problems were stated, as in Subsection 4.3. The configuration parameters in Table 4.13 were used for all three executions of WH-MOEA. For each design concept, the first individual in the population was coded using the poles and zeros of each initial linear model. The exploration bounds to refine poles and zeros were defined according to its location concerning the imaginary axis. In other words, the poles or zeros furthest from the imaginary axis had a greater exploration margin since they were less sensitive, while those closer to the imaginary axis had a smaller exploration margin. The codification of the first individual and the exploration bounds to refine the poles and zeros of each design concept are shown in Table 4.14.

After optimisation of the design concept $DC - 1z4p$, six Wiener-Hammerstein models were obtained. In all these models, the two fast dynamic poles were located at the front of the structure, whilst the two slow poles were located at

Table 4.14: Coding of the first individual in the population and exploration bounds to refine the poles and zeros (design concepts for identification of thermal process)

$DC - 1z4p$	
$nc = 0, nr = 1, mc = 0, mr = 4$	
P^{min}	[-0.055, -4.561, -2.623, -0.440, -0.054]
P_1^0	[-0.015, -4.461, -2.523, -0.340, -0.014]
P^{max}	[0.025, -4.361, -2.423, -0.240, -0.010]

$DC - 1z3p$	
$nc = 0, nr = 1, mc = 0, mr = 3$	
P^{min}	[-0.056, -1.593, -0.449, -0.054]
P_1^0	[-0.016, -1.493, -0.349, -0.014]
P^{max}	[0.024, -1.393, -0.249, -0.010]

$DC - 1z2p$	
$nc = 0, nr = 1, mc = 0, mr = 2$	
P^{min}	[-0.053, -0.369, -0.052]
P_1^0	[-0.013, -0.269, -0.012]
P^{max}	[0.027, -0.169, -0.010]

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

Step 6: Perform a decision making analysis.

Pareto fronts of the three design concepts are shown in Figure 4.17 and the model performance is quantified in Table 4.15. The best process representation was obtained by model M_1^{1z4p} , which has a complexity level of 15 (1 zero, 4 poles and 10 points to represent static non-linearity). Models M_2^{1z4p} and M_3^{1z4p} have similar performances to M_1^{1z4p} but with less complexity.

Table 4.15: Optimisation results for thermal process.

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

(M_2^{1z4p} and M_3^{1z4p} represent static non-linearity with 2 and 3 less points respectively). Considering the improved precision from M_6^{1z4p} to M_1^{1z4p} as 100%, the improvement between M_2^{1z4p} and M_3^{1z4p} is only 0.89% at the cost of adding one more point to static non-linearity.

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

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

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

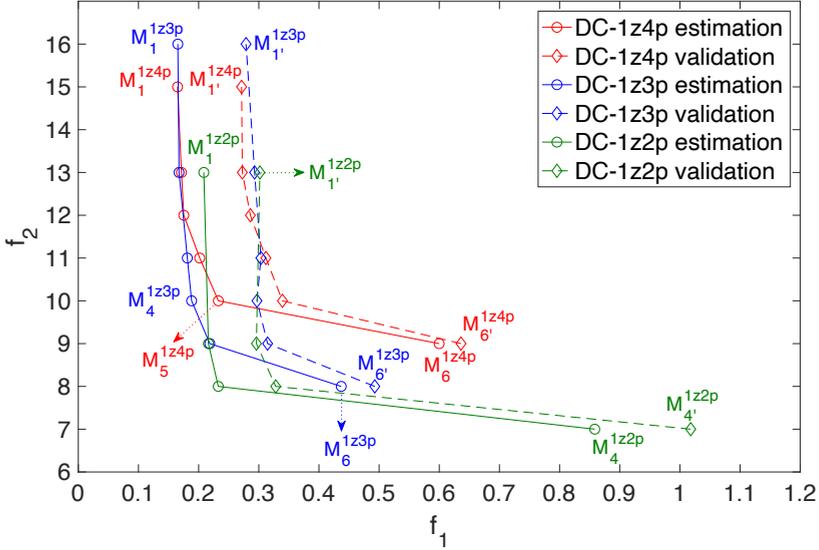


Figure 4.17: Comparison of Pareto fronts for nonlinear identification of thermal process.

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

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

On comparing $DC - 1z2p$ front with $DC - 1z3p$ and $DC - 1z4p$ fronts, one can see the effect of omitting the two fast actuator poles in the models. This

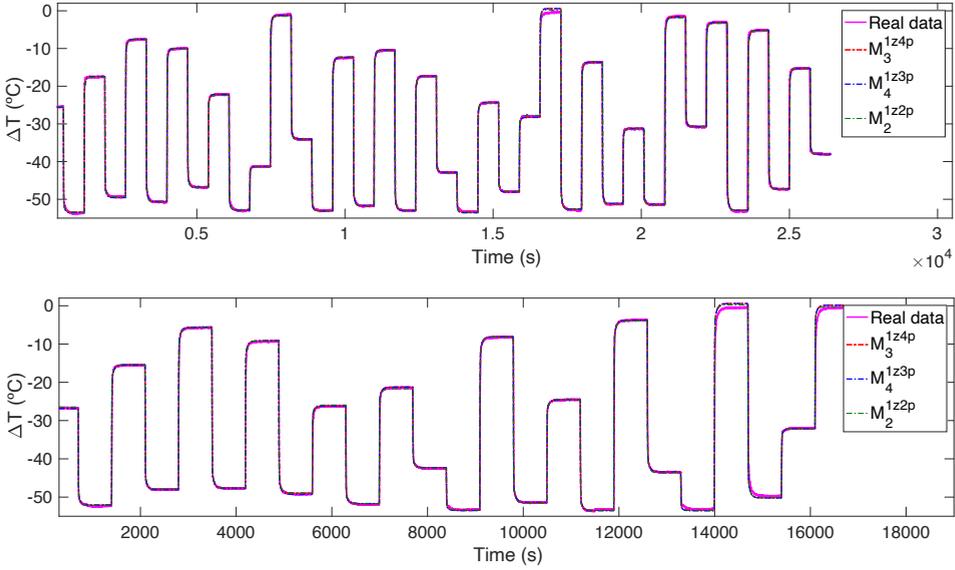


Figure 4.18: Performance of the models M_3^{1z4p} , M_4^{1z3p} , and M_2^{1z2p} evaluated on the estimation (top) and validation (bottom) data sets.

shortcoming cannot even be compensated by increasing the static non-linearity points.

As final part of the analysis, the designer can evaluate the models of all fronts on a different data set (validation set) than the one used for estimation. A practical example of this is shown in Figure 4.17. The displacement of the three Pareto fronts (models represented by diamonds) shows lower accuracy than those achieved by the estimation data set. The Pareto front of the first model concept can be seen to stand out more to the left, indicating that the difference between the four-pole and three-pole models has become wider in terms of precision. This particular situation can also give the designer clues to select a suitable model.

Thanks to this approach it has been possible to compare Wiener-Hammerstein and Hammerstein model structures. Clearly, more complex models will be more accurate, however it will always be important to quantify their differences and select the best, considering not only model accuracy but also complexity. For example, if you require an adequate compromise between precision and complexity, model M_2^{1z2p} could be selected as well as M_4^{1z3p} and M_3^{1z4p} . Figure

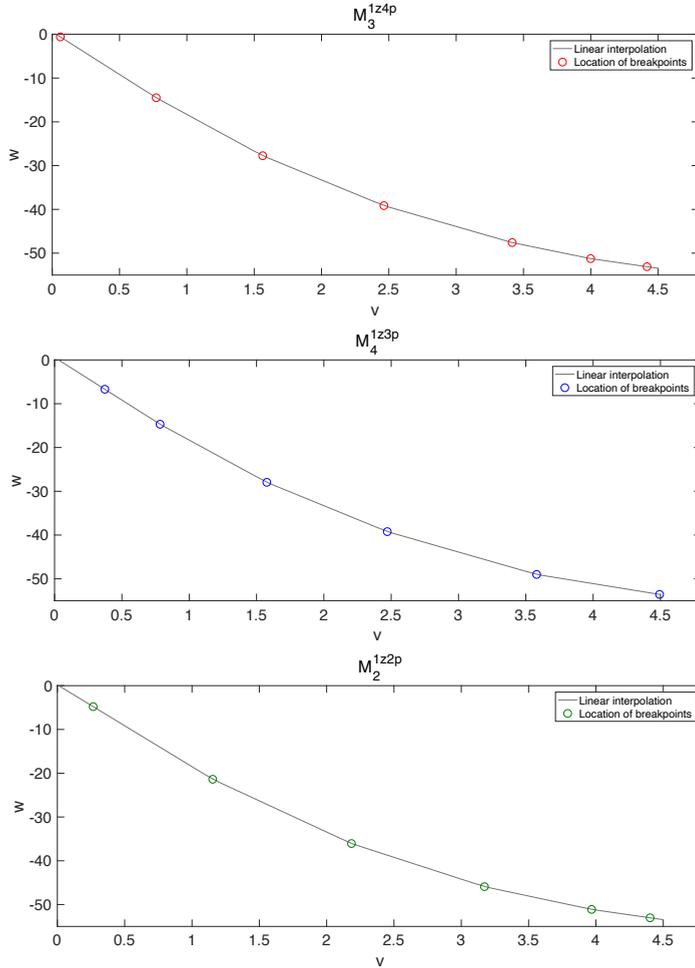


Figure 4.19: Static non-linearity of the thermal process captured by the models M_3^{1z4p} , M_4^{1z3p} , and M_2^{1z2p} .

4.18 shows a comparison of the performance of these models on the estimation and validation data sets, while Figure 4.19 shows how the break points were located to represent the static non-linearity of each model.

At first glance (Figure 4.18) it seems that the performances of the three selected models are the same; however an enlargement of the image (Figure 4.20) shows the existing differences mainly in the transient stage. Notice how

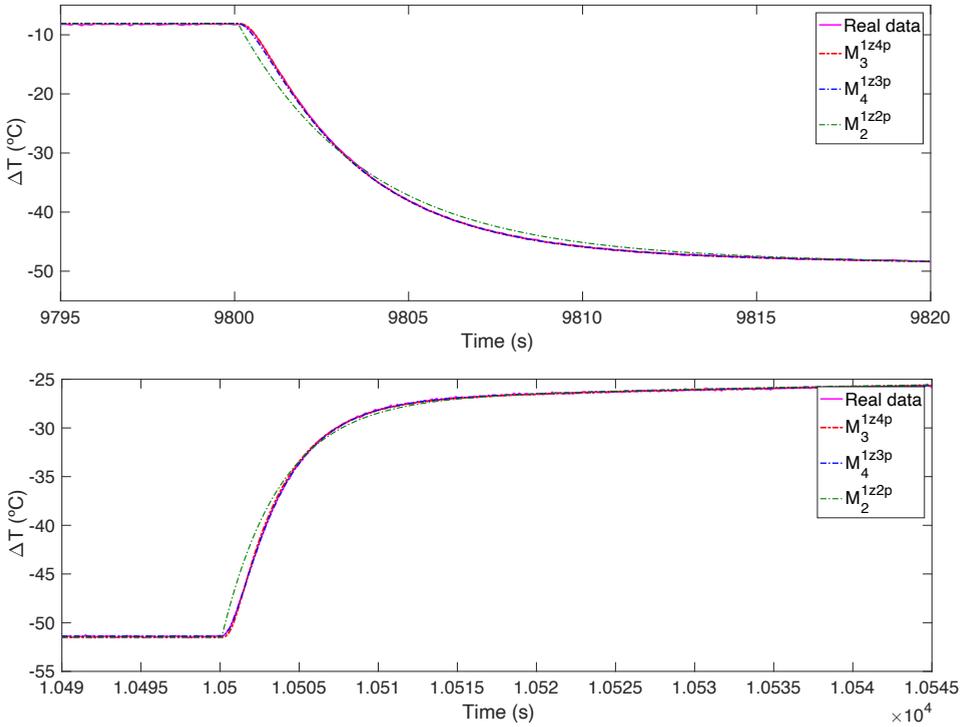


Figure 4.20: Enlargement illustrates the differences in the transient responses of M_3^{1z4p} , M_4^{1z3p} , and M_2^{1z2p} for the validation input.

model M_2^{1z2p} (green) has a small difference with the other two models with a Wiener-Hammerstein structure. If a simplified model is required, M_2^{1z2p} would therefore be a good option.

Chapter 5

Conclusions and future research directions

5.1 Conclusions

Exploring the state of art on the identification of Wiener-Hammerstein models, it became evident that, to date, there is a wide range of alternatives for estimating models with this structure. However, given the complexity of these models, current identification methods are also complex and, in many cases, some assumptions are necessary. Most of them, especially those presented in the last decade, start the identification procedure by obtaining a linear approximation of the nonlinear system. From then on, efforts are concentrated on distributing the dynamics between the two LTI blocks, considering that, there is another block in the middle representing the static non-linearity of the system. The way to distribute the dynamics has made the difference between the existing methods. Currently, methods with both parametric and non-parametric approaches can be found in the literature.

The first contribution of this thesis lies in the introduction of a new method to identify Wiener-Hammerstein models. This method is based on an evolutionary algorithm (WH-EA) with customized genetic operators that distribute the dynamics between the two LTI subsystems, while at the same time, pole-zero locations are subtly tuned and static non-linearity is captured with high precision. Development of WH-EA was motivated by the high user interaction required by most existing methods, due to the great complexity involved in the estimation procedure. Generally, most methods require intermediate stages where the user must intervene to execute specific procedures, analysing results and taking decisions.

Unlike conventional procedures, WH-EA estimates all parameters of a Wiener-Hammerstein model by solving one single optimization problem. That is, from an initial linear model, the algorithm solves the identification problem without intermediate procedures that require user interaction. WH-EA is able to look for the best BLA split while at the same time, refining pole-zero locations and capturing static non-linearity with high accuracy. The performance of this approach has been evaluated through a numerical example with a complex static non-linearity and through a well-known benchmark data (SYSID'09). The results show that WH-EA is able to identify Wiener-Hammerstein systems with good precision in a parametric framework avoiding high user interaction and drawbacks involved in using the QBLA.

As in all evolutionary optimization algorithms, WH-EA's performance is highly dependent on several issues which are detailed below:

- **Balance achieved between the population diversity and convergence.** It is achieved thanks to an appropriate selection of the WH-EA tuning parameters. Although there is no recipe to precisely define them, through experience it has been possible to establish appropriate values to fix mutation and crossover probabilities and to control mutations' aggressiveness. For this reason, most of the WH-EA parameters were the same in all included examples.
- **Initial population.** WH-EA uses the BLA as a starting point for nonlinear estimation. Thanks to this linear approximation and to information obtained from the input and output signal, a coherent initial population is defined allowing an effective exploration and, at the same time, facilitating convergence of WH-EA.
- **Search space.** As with the initial population, thanks to the BLA and the information obtained from the input and output signals (minimum and maximum values), an adequate search space is selected for an effective exploration. The search space for new pole and zero locations is defined around the BLA. Notice that the BLA captures the process dynamics with great precision and therefore, the search space for each pole and zero should not be too wide around these initial locations. Likewise, it must be taken into account that a small movement of a pole or zero represented in discrete-time implies a large movement in continuous-time, given the exponential relationship existing between the s and the z domains. On the other hand, thanks to the minimum and maximum values of the input and output signals, it is possible to effectively specify the search space where the static non-linearity will be captured. Since WH-EA starts from a known dynamic that needs to be split and refined, the search space of the static non-linearity is crucial to help the algorithm to converge.

The advantage of starting the estimation of a Wiener-Hammerstein model with a knowledge of the dynamics is evident. Although the identification problem is still complex since it is not possible to know in advance the distribution of the poles and zeros between the two linear subsystems of the model, thanks to the known dynamics it is possible to better bound the search space to find the best nonlinear approximation.

There is a prominent preference in the literature for the use of BLA, which allows to capture with great precision the dynamics of a process affected by a static non-linearity. However, from a practical point of view there are some limitations that should be pointed out. The most critical limitation is that the BLA is only defined for a fixed class of input signals. In this context, the

Riemann equivalence class of asymptotically normally distributed excitation signals are preferred. From a theoretical point of view, this kind of signals allow to represent a Wiener-Hammerstein model with a linear dynamic model plus a noise source representing the nonlinear distortions. Under this assumption, a linear dynamic model is guaranteed to be obtained with high accuracy; however, not all processes are able to be excited with this kind of signals. Further, in many situations, obtaining the BLA may involve the use of multiple long duration excitation signals, each of them designed carefully.

These disadvantages with the use of BLA, motivated the second contribution of this doctoral thesis. It consists of a new approach based on WH-EA but using a standard initial linear model and multi-step signals. The initial linear model can be obtained from the step response of the system using any standard identification procedure. From a practical point of view, both the step signal and the multi-step signal are more conventional, especially in the industrial context.

It is evident that the initial linear model obtained through the step response will not be able to achieve the same accuracy as the BLA. The initial linear model is likely to be a biased version of the real system dynamics; however, thanks to WH-EA's customised genetic operators, dynamics can be fine-tuned as the static non-linearity is captured. This can be clearly seen from the results obtained in the identification examples presented in Section 3.9.

Thanks to this new approach that avoids the BLA estimation, it has also been possible to tackle to some extent another drawbacks present in block-oriented models:

- **Structure detection.** To date, very few studies have been reported that help to detect an appropriate structure among the possible options belonging to the group of block-oriented non-linear models. Existing Wiener, Hammerstein and Wiener-Hammerstein model identification methods assume a known structure a priori.
- **Rigidity of existing identification methods.** Even though the Wiener, Hammerstein and Wiener-Hammerstein models have several features in common, existing identification methods are rigid and exclusive for each single structure. That is, an identification method for Hammerstein models cannot be used to identify Wiener models or vice versa.

The Wiener, Hammerstein and Wiener-Hammerstein models are based on block series connections and a single static non-linearity. The difference

between them is the location of the non-linearity around the dynamics. A key factor in achieving a good estimate is the bounding of the search space where explore for possible solutions. Assuming that a known dynamic system is affected by a static non-linearity, the search space cannot be set correctly if the non-linearity location around the dynamics is unknown. For this reason, existing identification methods for Wiener, Hammerstein and Wiener-Hammerstein models assume an a priori block structure (static non-linearity location) and do not provide flexibility to estimate a model with a different structure than the one for which they were developed.

Chapter 3 shows how WH-EA is able to overcome these drawbacks, performing an adequate type selection of the excitation signal to be used for the identification experiment. Through the use of multi-step signals it has been possible to establish an appropriate search space to capture the static non-linearity regardless of its location around the dynamics. This allows WH-EA to estimate Wiener, Hammerstein and Wiener-Hammerstein models without any user specification about the type of structure to be estimated. The effectiveness of the algorithm, both for choosing the best structure and estimating its parameters with high accuracy has been proved through three numerical examples containing complex static non-linearities and a real process based on a Peltier cell. Each numerical example was designed with a different structure, i.e. the algorithm was tested against a Wiener model, a Hammerstein model and a Wiener-Hammerstein model. For each case, WH-EA classified the dynamics correctly and estimated high precision models. It should be noted that WH-EA does not require user interaction to execute intermediate procedures. Simply configuring of a few tuning parameters, with an initial linear model and information obtained from the input and output signals, the algorithm has autonomy to achieve the final results. This is highly attractive especially when there is uncertainty about the distribution of the dynamics around non-linearity.

Furthermore, WH-EA has complete flexibility to estimate models in discrete-time or continuous-time, without the choice between these alternatives implies any modification on the algorithm. The estimates presented in Chapter 3 were developed in continuous-time, while in Chapter 2 the estimates were made in discrete-time. The difference between them lies in the calculation of the objective function: in the discrete case the dynamic simulation can be done using the *filter* command, whilst for continuous-time the *lsim* command must be used instead. Both commands are available within MATLAB.

Though, calculation of the objective function in continuous-time requires higher computational cost, the pole-zero locations in the S-plane provides a

much clearer picture of the system behaviour and, depending on the pole-zero sensitivities (how close or far away they are from the imaginary axis) a coherent search space can be established to their fine-tune.

The third contribution addresses the problem of identifying Wiener, Hammerstein and Wiener-Hammerstein models using a multi-objective optimization approach. It introduces a new way of identifying Wiener-Hammerstein models, allowing an exhaustive analysis between different model alternatives helping the control engineer to choose the most appropriate structure.

As in the previous contribution, neither Gaussian-type signals nor the initial BLA have been used. The procedure is performed thanks to WH-MOEA, a new multi-objective evolutionary optimisation algorithm which has been formulated ad-hoc to manage this type of block-oriented models without previous knowledge about the process structure. The method highlights the importance of generating a set of models with common targets and different performances -the generated Pareto fronts can be compared and an analyse of the trade-off between precision and complexity can be done. The procedure is therefore not focused on the selection of a specific model, but rather on showing the benefits of obtaining a wide range of models with different features, giving the engineer a chance to choose a final model according to his/her preferences. The effectiveness and importance of using this approach is demonstrated through the estimation of a CSTR (simulated process), a numerical example and the thermal process already used.

The WH-MOEA procedure has two important advantages: on one hand, by studying the Pareto front of a specific design concept, the influence of a variation in the non-linearity breakpoints on model precision can be analysed. From the analyses, the uncertainty about the number of breakpoints to represent static non-linearity is reduced. On the other hand, the design concept analysis can be used to compare several candidates. For example, simpler models than those recommended by the MDL criteria can be found with acceptable performance. Finally, the procedure described here contributes to solving the issue stated in many control problems, where design requirements and user preferences do not always point to model precision as the only objective, as model simplicity can also be an influential factor in decision-making.

5.2 Future research directions

Based on the work presented in this dissertation, this section presents some ideas that may serve as a basis for future research directions on block-oriented models:

- Identification of Wiener-Hammerstein models considering hysteresis non-linearities.
- Identification of multivariable Wiener-Hammerstein models.
- Identification of Hammerstein-Wiener models considering a multi-objective approach to penalise accuracy and complexity of the model.
- Development of new strategies for robust estimation of Wiener-Hammerstein models.

These ideas are discussed in more detail below:

Identification of Wiener-Hammerstein models considering hysteresis non-linearities.

Hysteresis is a type of static non-linearity that can be present in many real systems. In the context of block-oriented models, several approaches to estimating models with hysteresis can be found in the literature; however, most of these have been developed on Wiener or Hammerstein systems. The complexity implicit in the Wiener-Hammerstein structure has prevented a good set of methods to identify these models considering hysteresis from being available today.

Starting from WH-MOEA, estimation of hysteresis could be solved by modifying individual codification and adding new genetic operators. Using the multi-objective approach, the engineer could compare models with different complexities (both in the dynamic part and in the static non-linearity) and analyse the effect of hysteresis on model's accuracy. This would be of great help in decision-making when selecting a model especially for control applications.

Identification of multivariable Wiener-Hammerstein models.

The contributions presented in this dissertation are applicable to Single-input Single-Output systems (SISO); however, both WH-EA and WH-MOEA can be extended to estimate Multiple-Input Single-Output (MISO) and

Multiple-Input Multiple-Output (MIMO) systems. Very few contributions on multivariable block-oriented model identification are currently reported in the literature. Almost all concentrate on Wiener and Hammerstein multivariable systems. In the case of Wiener-Hammerstein models, no contributions have been reported to date.

The coupling between multiple inputs and outputs leads to a complex identification problem, so it would be interesting to evaluate the potential of WH-EA and WH-MOEA to deal with multivariable models.

Identification of Hammerstein-Wiener models considering a multi-objective approach to penalise accuracy and complexity of the model.

Taking WH-MOEA as a starting point, a new algorithm could be developed to address the Hammerstein-Wiener model identification problem. To date, no identification methods with a multi-objective approach have been reported for this class of models. For the case of Hammerstein-Wiener models it would not be necessary to partition the dynamics since there is only one dynamic block, however, an adaptation of the algorithm would be needed to deal with two static non-linearities and their corresponding search spaces. Since a variable number of parameters would be considered in each non-linearity, the Pareto front generated let analyse the most appropriate model. This proposal would become even more attractive if hysteresis-type non-linearities were considered.

Development of new strategies for robust estimation of Wiener-Hammerstein models.

As seen throughout Chapter 1, there are currently several methods for identifying Wiener-Hammerstein models, however, the problem of robust identification in the presence of uncertainty has not been addressed.

It should be noted that a system is not always subject to only one type of non-linearity. In addition to the static non-linearity, the parameters of the dynamic part are likely to change over time or as a function of the operating point. Under appropriate bounds this parameter variation could be considered as an uncertainty. To address this identification problem, one could exploit the features of WH-EA and WH-MOEA and combine them with other techniques such as fuzzy logic to create new ways of identifying Wiener-Hammerstein models and their specific cases considering uncertainty.

Bibliography

- [1] M. Ahmadi and H. Mojallali. “Identification of multiple-input single-output Hammerstein models using Bezier curves and Bernstein polynomials”. In: *Applied Mathematical Modelling* 35.4 (2011), pp. 1969–1982 (cit. on p. 3).
- [2] H. Akaike. “A new look at the statistical model identification”. In: *Selected Papers of Hirotugu Akaike*. Springer, 1974, pp. 215–222 (cit. on p. 88).
- [3] H. Akaike. “Autoregressive model fitting for control”. In: *Selected Papers of Hirotugu Akaike*. Springer, 1998, pp. 153–170 (cit. on p. 88).
- [4] H. Akaike. “Fitting autoregressive models for prediction”. In: *Annals of the institute of Statistical Mathematics* 21.1 (1969), pp. 243–247 (cit. on p. 88).
- [5] E.-W. Bai, Z. Cai, S. Dudley-Javoroski, and R. K. Shields. “Identification of a modified Wiener-Hammerstein system and its application in electrically stimulated paralyzed skeletal muscle modeling”. In: *Automatica* 45.3 (2009), pp. 736–743 (cit. on p. 4).
- [6] E.-W. Bai, Z. Cai, S. Dudley-Javoroski, and R. K. Shields. “Application of Wiener-Hammerstein system identification in electrically stimulated paralyzed skeletal muscle modeling”. In: *Decision and Control, 2008*.

- CDC 2008. 47th IEEE Conference on. IEEE.* 2008, pp. 3305–3310 (cit. on p. 4).
- [7] I. Benyó, J. Kovács, J. Mononen, and U. Kortela. “Modelling of steam temperature dynamics of a superheater”. In: *International Journal of Simulation* 6 (2005), pp. 3–9 (cit. on p. 4).
- [8] S. I. Biagiola and J. L. Figueroa. “Identification of uncertain MIMO Wiener and Hammerstein models”. In: *Computers & chemical engineering* 35.12 (2011), pp. 2867–2875 (cit. on p. 3).
- [9] S. Billings and S. Fakhouri. “Identification of a class of nonlinear systems using correlation analysis”. In: *Proceedings of the Institution of Electrical Engineers*. Vol. 125. 7. IET. 1978, pp. 691–697 (cit. on p. 5).
- [10] S. A. Billings. *Nonlinear system identification: NARMAX methods in the time, frequency, and spatio-temporal domains*. John Wiley & Sons, 2013 (cit. on p. 2).
- [11] H. Bloemen, C. Chou, T. Van den Boom, V. Verdult, M. Verhaegen, and T. Backx. “Wiener model identification and predictive control for dual composition control of a distillation column”. In: *Journal of Process control* 11.6 (2001), pp. 601–620 (cit. on p. 4).
- [12] J. Bussgang. *Crosscorrelation functions of amplitude-distorted Gaussian signals*. *Research Lab. Electron.* 1952 (cit. on p. 17).
- [13] Z. Cai, E.-W. Bai, and R. K. Shield. “Application of Block-oriented System Identification to Modelling Paralysed Muscle Under Electrical Stimulation”. In: *Block-oriented Nonlinear System Identification*. Springer, 2010, pp. 403–419 (cit. on p. 4).
- [14] D. Chou. “Efficacy of Hammerstein models in capturing the dynamics of isometric muscle stimulated at various frequencies”. PhD thesis. Massachusetts Institute of Technology, 2006 (cit. on p. 4).
- [15] C. A. C. Coello and G. B. Lamont. *Applications of Multi-Objective evolutionary algorithms*. Advances in Natural Computation Vol. 1. World scientific publishing, 2004 (cit. on p. 93).

-
- [16] B. De Moor, P. De Gersem, B. De Schutter, and W. Favoreel. “DAISY: A database for identification of systems”. In: *JOURNAL A* 38 (1997), pp. 4–5 (cit. on p. 2).
- [17] F. De Ridder, R. Pintelon, J. Schoukens, and D. P. Gillikin. “Modified AIC and MDL model selection criteria for short data records”. In: *IEEE Transactions on Instrumentation and Measurement* 54.1 (2005), pp. 144–150 (cit. on pp. 55, 88).
- [18] E. J. Dempsey and D. T. Westwick. “Identification of Hammerstein models with cubic spline nonlinearities”. In: *IEEE Transactions on Biomedical Engineering* 51.2 (2004), pp. 237–245. DOI: [10.1109/TBME.2003.820384](https://doi.org/10.1109/TBME.2003.820384) (cit. on p. 4).
- [19] O. P. Dewhirst, D. M. Simpson, N. Angarita, R. Allen, and P. L. Newland. “Wiener-Hammerstein parameter estimation using differential evolution”. In: *International Conference on Bio-inspired Systems and Signal Processing*. 2010, pp. 271–276 (cit. on p. 4).
- [20] M. Al-Dhaifallah, K. S. Nisar, P. Agarwal, and A. Elsayyad. “Modeling and identification of heat exchanger process using least squares support vector machines”. In: *Thermal Science* 21.6 Part B (2017), pp. 2859–2869 (cit. on p. 4).
- [21] F. J. Doyle, R. K. Pearson, and B. A. Ogunnaike. *Identification and control using Volterra models*. Springer Science & Business Media, 2012 (cit. on p. 2).
- [22] E. Economakos. “Identification of a group of internal signals of zero-memory nonlinear systems”. In: *Electronics Letters* 7.4 (1971), p. 99 (cit. on p. 5).
- [23] M. Enqvist and L. Ljung. “Linear approximations of nonlinear FIR systems for separable input processes”. In: *Automatica* 41.3 (2005), pp. 459–473 (cit. on pp. 5, 17, 88).
- [24] R. S. Esfandiari and B. Lu. *Modeling and analysis of dynamic systems*. CRC Press, 2014 (cit. on p. 2).

- [25] E. Eskinat, S. H. Johnson, and W. L. Luyben. “Use of Hammerstein models in identification of nonlinear systems”. In: *AIChE Journal* 37.2 (1991), pp. 255–268 (cit. on p. 4).
- [26] T. Falck, K. Pelckmans, J. A. Suykens, and B. De Moor. “Identification of wiener-hammerstein systems using LS-SVMs”. In: *IFAC Proceedings Volumes* 42.10 (2009), pp. 820–825 (cit. on p. 47).
- [27] P. Ferreira, A. Ruano, and C. Fonseca. “Evolutionary multiobjective design of radial basis function networks for greenhouse environmental control”. In: *IFAC Proceedings Volumes* 38.1 (2005), pp. 63–68 (cit. on p. 89).
- [28] C. M. Fonseca and P. J. Fleming. “Non-linear system identification with multiobjective genetic algorithms”. In: *IFAC Proceedings Volumes* 29.1 (1996), pp. 1169–1174 (cit. on p. 89).
- [29] K. Fruzzetti, A. Palazoglu, and K. McDonald. “Nolinear model predictive control using Hammerstein models”. In: *Journal of process control* 7.1 (1997), pp. 31–41 (cit. on p. 88).
- [30] H. Garnier and M. Mensler. “CONTSID—A continuous-time system identification toolbox for Matlab®”. In: *1999 European Control Conference (ECC)*. IEEE. 1999, pp. 3322–3327 (cit. on p. 55).
- [31] A. Gardiner. “Elimination of the effect of nonlinearities on process crosscorrelations”. In: *Electronics Letters* 2.5 (1966), pp. 164–165 (cit. on p. 5).
- [32] H. Garnier, M. Mensler, and A. Richard. “Continuous-time model identification from sampled data: implementation issues and performance evaluation”. In: *International journal of Control* 76.13 (2003), pp. 1337–1357 (cit. on p. 55).
- [33] H. Garnier and P. C. Young. “The advantages of directly identifying continuous-time transfer function models in practical applications”. In: *International Journal of Control* 87.7 (2014), pp. 1319–1338 (cit. on p. 55).

-
- [34] G. Giordano, S. Gros, and J. Sjöberg. “An improved method for Wiener–Hammerstein system identification based on the Fractional Approach”. In: *Automatica* 94 (2018), pp. 349–360 (cit. on pp. 6, 8).
- [35] F. Giri and E.-W. Bai. *Block-oriented nonlinear system identification*. Vol. 1. Springer, 2010 (cit. on p. 2).
- [36] J. Gomez, A. Jutan, and E. Baeyens. “Wiener model identification and predictive control of a pH neutralisation process”. In: *IEEE Proceedings-Control Theory and Applications* 151.3 (2004), pp. 329–338 (cit. on pp. 2, 4, 88).
- [37] W. Greblicki and M. Pawlak. *Nonparametric system identification*. Vol. 1. Cambridge University Press Cambridge, 2008 (cit. on p. 3).
- [38] F. Guo and G. Bretthauer. “Identification of MISO Wiener and Hammerstein systems”. In: *2003 European Control Conference (ECC)*. IEEE, 2003, pp. 2144–2149 (cit. on p. 3).
- [39] F. Hafiz, A. Swain, and E. Mendes. “Multi-objective evolutionary framework for non-linear system identification: A comprehensive investigation”. In: *Neurocomputing* 386 (2020), pp. 257–280 (cit. on p. 89).
- [40] A. Hammerstein et al. “Nichtlineare integralgleichungen nebst anwendungen”. In: *Acta mathematica* 54 (1930), pp. 117–176 (cit. on p. 4).
- [41] A. Haryanto and K.-S. Hong. “Maximum likelihood identification of Wiener–Hammerstein models”. In: *Mechanical Systems and Signal Processing* 41.1-2 (2013), pp. 54–70 (cit. on p. 4).
- [42] Z. Hasiewicz, P. Wachel, G. Mzyk, and B. Kozdraś. “Multistage identification of Wiener-Hammerstein system”. In: *Polish Control Conference*. Springer, 2017, pp. 527–535 (cit. on pp. 6, 8).
- [43] T. Hatanaka, N. Kondo, and K. Uosaki. “Multi-objective structure selection for RBF networks and its application to nonlinear system

- identification”. In: *Multi-Objective Machine Learning*. Springer, 2006, pp. 491–505 (cit. on p. 89).
- [44] J. Herrero, G. Reynoso-Meza, M. Martínez, X. Blasco, and J. Sanchis. “A smart-distributed Pareto front using the ev-MOGA evolutionary algorithm”. In: *International Journal on Artificial Intelligence Tools* 23.02 (2014), p. 1450002 (cit. on pp. 89, 95).
- [45] V. Huilcapi, X. Blasco, J. M. Herrero, and G. Reynoso-Meza. “A loop pairing method for multivariable control systems under a multi-objective optimization approach”. In: *IEEE Access* 7 (2019), pp. 81994–82014 (cit. on p. 101).
- [46] I. W. Hunter and M. J. Korenberg. “The identification of nonlinear biological systems: Wiener and Hammerstein cascade models”. In: *Biological cybernetics* 55.2-3 (1986), pp. 135–144 (cit. on p. 4).
- [47] A. Janczak. *Identification of nonlinear systems using neural networks and polynomial models: a block-oriented approach*. Vol. 310. Springer Science & Business Media, 2004 (cit. on p. 2).
- [48] F. Jurado. “A method for the identification of solid oxide fuel cells using a Hammerstein model”. In: *Journal of Power Sources* 154.1 (2006), pp. 145–152 (cit. on p. 4).
- [49] A. Kalafatis, N. Arifin, L. Wang, and W. Cluett. “A new approach to the identification of pH processes based on the Wiener model”. In: *Chemical Engineering Science* 50.23 (1995), pp. 3693–3701 (cit. on p. 4).
- [50] D. C. Karnopp, D. L. Margolis, and R. C. Rosenberg. *System dynamics: modeling, simulation, and control of mechatronic systems*. John Wiley & Sons, 2012 (cit. on p. 2).
- [51] T. Katayama and H. Ase. “Linear approximation and identification of MIMO Wiener–Hammerstein systems”. In: *Automatica* 71 (2016), pp. 118–124 (cit. on p. 3).

- [52] M. Kazemi and M. M. Arefi. “A fast iterative recursive least squares algorithm for Wiener model identification of highly nonlinear systems”. In: *ISA transactions* 67 (2017), pp. 382–388 (cit. on p. 104).
- [53] M. J. Korenberg and I. W. Hunter. “The identification of nonlinear biological systems: LNL cascade models”. In: *Biological cybernetics* 55.2 (1986), pp. 125–134 (cit. on p. 4).
- [54] M. J. Korenberg and I. W. Hunter. “The identification of nonlinear biological systems: Wiener kernel approaches”. In: *Annals of Biomedical Engineering* 18.6 (1990), pp. 629–654 (cit. on p. 4).
- [55] M. Laumanns, L. Thiele, K. Deb, and E. Zitzler. “Combining Convergence and Diversity in Evolutionary Multiobjective Optimization”. In: *Evolutionary Computation* 10.3 (2002), pp. 263–282. DOI: [10.1162/106365602760234108](https://doi.org/10.1162/106365602760234108) (cit. on p. 95).
- [56] L. Lauwers. “Some practical applications of the best linear approximation in nonlinear block-oriented modelling”. PhD thesis. Vrije Universiteit Brussel, 2011 (cit. on pp. 5, 47).
- [57] L. Lauwers, J. Schoukens, R. Pintelon, and M. Enqvist. “Nonlinear structure analysis using the best linear approximation”. In: *Proceedings of ISMA2006: International Conference on Noise and Vibration Engineering, Vols 1-8*. Vol. 5. KATHOLIEKE UNIV LEUVEN, DEPT WERKTUIGKUNDE. 2006, pp. 2751–2760 (cit. on p. 3).
- [58] M. Ławryńczuk. “Nonlinear predictive control for Hammerstein–Wiener systems”. In: *ISA transactions* 55 (2015), pp. 49–62 (cit. on p. 2).
- [59] M. Ławryńczuk. “Nonlinear predictive control of dynamic systems represented by Wiener–Hammerstein models”. In: *Nonlinear Dynamics* 86.2 (2016), pp. 1193–1214 (cit. on pp. 2, 88).
- [60] S. Li and Y. Li. “Model predictive control of an intensified continuous reactor using a neural network Wiener model”. In: *Neurocomputing* 185 (2016), pp. 93–104 (cit. on p. 2).

- [61] G. P. Liu. *Nonlinear identification and control: a neural network approach*. Springer Science & Business Media, 2012 (cit. on p. 2).
- [62] T. Liu, Q.-G. Wang, and H.-P. Huang. “A tutorial review on process identification from step or relay feedback test”. In: *Journal of Process control* 23.10 (2013), pp. 1597–1623 (cit. on p. 2).
- [63] L. Ljung. *System Identification Toolbox for Use with MATLAB*. The MathWorks, Inc., 2007 (cit. on pp. 36, 113).
- [64] L. Ljung. “Estimating linear time-invariant models of nonlinear time-varying systems”. In: *European Journal of Control* 7.2 (2001), pp. 203–219 (cit. on pp. 5, 88).
- [65] L. Ljung. “System identification: Theory for the user, ptr prentice hall information and system sciences series”. In: ed: *Prentice Hall, New Jersey* (1999) (cit. on p. 2).
- [66] S. M. R. Loghmanian, R. Ahmad, and H. Jamaluddin. “Multi-objective optimization of NARX model for system identification using genetic algorithm”. In: *2009 First International Conference on Computational Intelligence, Communication Systems and Networks*. IEEE. 2009, pp. 196–201 (cit. on p. 89).
- [67] S. M. R. Loghmanian, R. Yusof, and M. Khalid. “Nonlinear dynamic system identification using Volterra series: Multi-objective optimization approach”. In: *2011 Fourth International Conference on Modeling, Simulation and Applied Optimization*. IEEE. 2011, pp. 1–5 (cit. on p. 89).
- [68] C. Mattson and A. Messac. “Concept selection in n-dimension using s-Pareto frontiers and visualization”. In: *9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*. 2002, p. 5418 (cit. on p. 101).
- [69] K. M. Miettinen. *Nonlinear Multiobjective Optimization*. 1st. Kluwer Academic Publishers: Boston, USA, 1999 (cit. on p. 93).

-
- [70] L. A. Mora and J. E. Amaya. “A New Identification Method Based on Open Loop Step Response of Overdamped System”. In: *REVISTA IBEROAMERICANA DE AUTOMATICA E INFORMATICA INDUSTRIAL* 14.1 (2017), pp. 31–43 (cit. on p. 2).
- [71] G. Mzyk. *Combined parametric-nonparametric identification of block-oriented systems*. Vol. 238. Springer, 2014 (cit. on p. 3).
- [72] G. Mzyk and P. Wachel. “Kernel-based identification of Wiener–Hammerstein system”. In: *Automatica* 83 (2017), pp. 275–281 (cit. on pp. 6, 8).
- [73] E. Nahas, M. Henson, and D. Seborg. “Nonlinear internal model control strategy for neural network models”. In: *Computers & Chemical Engineering* 16.12 (1992), pp. 1039–1057 (cit. on p. 104).
- [74] A. Naitali and F. Giri. “Wiener–Hammerstein system identification—an evolutionary approach”. In: *International Journal of Systems Science* 47.1 (2016), pp. 45–61 (cit. on pp. 7, 8, 47, 89, 90).
- [75] K. Narendra and P. Gallman. “An iterative method for the identification of nonlinear systems using a Hammerstein model”. In: *IEEE Transactions on Automatic control* 11.3 (1966), pp. 546–550 (cit. on p. 4).
- [76] J. Noël and M. Schoukens. “Hysteretic benchmark with a dynamic nonlinearity”. In: *Workshop on nonlinear system identification benchmarks*. 2016, pp. 7–14 (cit. on p. 2).
- [77] S. J. Norquay, A. Palazoglu, and J. Romagnoli. “Model predictive control based on Wiener models”. In: *Chemical Engineering Science* 53.1 (1998), pp. 75–84 (cit. on p. 88).
- [78] J. Oliver, R. Prieto, J. Cobos, O. Garcia, and P. Alou. “Hybrid wiener-hammerstein structure for grey-box modeling of dc-dc converters”. In: *2009 Twenty-Fourth Annual IEEE Applied Power Electronics Conference and Exposition*. IEEE. 2009, pp. 280–285 (cit. on p. 4).

- [79] J. J. E. Oviedo, J. P. Vandewalle, and V. Wertz. *Fuzzy logic, identification and predictive control*. Springer Science & Business Media, 2006 (cit. on p. 2).
- [80] A. Padilla, H. Garnier, and M. Gilson. “Version 7.0 of the CONTSID toolbox”. In: *IFAC-PapersOnLine* 48.28 (2015), pp. 757–762 (cit. on p. 55).
- [81] J. Paduart, L. Lauwers, R. Pintelon, and J. Schoukens. “Identification of a Wiener–Hammerstein system using the polynomial nonlinear state space approach”. In: *Control Engineering Practice* 20.11 (2012), pp. 1133–1139 (cit. on p. 47).
- [82] A. Pajares, X. Blasco, J. M. Herrero, and R. Simarro. “Using a Multiobjective Approach to Compare Multiple Design Alternatives—An Application to Battery Dynamic Model Tuning”. In: *Energies* 10.7 (2017), p. 999 (cit. on p. 101).
- [83] R. K. Pearson and M. Pottmann. “Gray-box identification of block-oriented nonlinear models”. In: *Journal of Process Control* 10.4 (2000), pp. 301–315 (cit. on p. 3).
- [84] R. Pintelon and J. Schoukens. *System identification: a frequency domain approach*. John Wiley & Sons, 2012 (cit. on pp. 8, 17).
- [85] M. Pottmann and R. K. Pearson. “Block-oriented NARMAX models with output multiplicities”. In: *AIChE Journal* 44.1 (1998), pp. 131–140 (cit. on p. 3).
- [86] D. Pršić, N. Nedić, V. Filipović, L. Dubonjić, and A. Vičovac. “Multilinear model of heat exchanger with Hammerstein structure”. In: *Journal of Control Science and Engineering* 2016 (2016) (cit. on p. 4).
- [87] S. N. Qasem and S. M. H. Shamsuddin. “Radial basis function network based on multi-objective particle swarm optimization”. In: *2009 6th International Symposium on Mechatronics and its Applications*. IEEE, 2009, pp. 1–6 (cit. on p. 89).

- [88] S. J. Qin and T. A. Badgwell. “A survey of industrial model predictive control technology”. In: *Control engineering practice* 11.7 (2003), pp. 733–764 (cit. on p. 88).
- [89] J. Rissanen. “Modeling by shortest data description”. In: *Automatica* 14.5 (1978), pp. 465–471 (cit. on p. 88).
- [90] J. Schoukens, J. Lataire, R. Pintelon, G. Vandersteen, and T. Dobrowiecki. “Robustness issues of the best linear approximation of a nonlinear system”. In: *IEEE Transactions on Instrumentation and Measurement* 58.5 (2009), pp. 1737–1745 (cit. on p. 17).
- [91] J. Schoukens and R. Pintelon. *Identification of linear systems: a practical guideline to accurate modeling*. Elsevier, 2014 (cit. on p. 2).
- [92] J. Schoukens, R. Pintelon, T. Dobrowiecki, and Y. Rolain. “Identification of linear systems with nonlinear distortions”. In: *Automatica* 41.3 (2005), pp. 491–504 (cit. on p. 5).
- [93] J. Schoukens, R. Pintelon, and Y. Rolain. *Mastering system identification in 100 exercises*. John Wiley & Sons, 2012 (cit. on pp. 8, 16, 54).
- [94] J. Schoukens, R. Pintelon, Y. Rolain, M. Schoukens, K. Tiels, L. Vanbeylen, A. Van Mulders, and G. Vandersteen. “Structure discrimination in block-oriented models using linear approximations: A theoretic framework”. In: *Automatica* 53 (2015), pp. 225–234 (cit. on p. 3).
- [95] J. Schoukens, J. Suykens, and L. Ljung. “Wiener-Hammerstein Benchmark”. In: *15th IFAC Symposium on System Identification, Saint-Malo, France, July, 2009*. 2009 (cit. on pp. 2, 7, 10, 35, 39).
- [96] M. Schoukens, T. Wigren, and J. Noël. “Cascaded tanks benchmark combining soft and hard nonlinearities”. In: *Workshop on nonlinear system identification benchmarks, Brussels, Belgium* (2016) (cit. on p. 2).

- [97] M. Schoukens. “Identification of parallel block-oriented models starting from the best linear approximation”. In: *Brussels, Belgium: Vrije Universiteit Brussel (VUB)* (2015) (cit. on p. 3).
- [98] M. Schoukens, A. Marconato, R. Pintelon, G. Vandersteen, and Y. Rolain. “Parametric identification of parallel Wiener–Hammerstein systems”. In: *Automatica* 51 (2015), pp. 111–122 (cit. on p. 3).
- [99] M. Schoukens, R. Pintelon, and Y. Rolain. “Identification of Wiener–Hammerstein systems by a nonparametric separation of the best linear approximation”. In: *Automatica* 50.2 (2014), pp. 628–634 (cit. on pp. 6, 8, 47).
- [100] M. Schoukens and Y. Rolain. “Parametric identification of parallel Wiener systems”. In: *IEEE Transactions on Instrumentation and Measurement* 61.10 (2012), pp. 2825–2832 (cit. on p. 3).
- [101] M. Schoukens and K. Tiels. “Identification of nonlinear block-oriented systems starting from linear approximations: A survey”. In: *CoRR, abs/1607.01217* (2016) (cit. on pp. 2, 5, 8, 88).
- [102] 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”. In: *Control Engineering Practice* 20.11 (2012), pp. 1119–1125 (cit. on pp. 5, 8, 47).
- [103] I. Škrjanc, M. Lepetic, J. L. Figueroa, and S. Blazic. “Fuzzy model-based predictive control for a CSTR with multiple steady state: A simulation study and a comparison with other nonlinear MBPC control algorithms”. In: *WSEAS Transactions on Systems* 3.2 (2004), pp. 789–94 (cit. on p. 104).
- [104] A. Susperregui, J. M. Herrero, M. I. Martinez, G. Tapia-Otaegui, and X. Blasco. “Multi-Objective Optimisation-Based Tuning of Two Second-Order Sliding-Mode Controller Variants for DFIGs Connected to Non-Ideal Grid Voltage”. In: *Energies* 12.19 (2019), p. 3782 (cit. on p. 101).

-
- [105] J. A. Suykens and J. P. Vandewalle. *Nonlinear Modeling: advanced black-box techniques*. Springer Science & Business Media, 2012 (cit. on p. 2).
- [106] A. H. Tan, H. K. Wong, and K. Godfrey. “Identification of a Wiener–Hammerstein system using an incremental nonlinear optimisation technique”. In: *Control Engineering Practice* 20.11 (2012), pp. 1140–1148 (cit. on p. 47).
- [107] K. Tiels, P. S. Heuberger, and J. Schoukens. “Parameter reduction of MISO Wiener-Schetzen models using the best linear approximation”. In: *2012 IEEE International Instrumentation and Measurement Technology Conference Proceedings*. IEEE. 2012, pp. 2114–2118 (cit. on p. 3).
- [108] K. Tiels, M. Schoukens, and J. Schoukens. “Initial estimates for Wiener–Hammerstein models using phase-coupled multisines”. In: *Automatica* 60 (2015), pp. 201–209 (cit. on p. 8).
- [109] P. Van Overschee and B. De Moor. *Subspace identification for linear systems: Theory—Implementation—Applications*. Springer Science & Business Media, 2012 (cit. on p. 2).
- [110] L. Vanbeylen. “A fractional approach to identify Wiener–Hammerstein systems”. In: *Automatica* 50.3 (2014), pp. 903–909 (cit. on pp. 6, 8, 45, 47).
- [111] G. Vandersteen. “Identification of linear and nonlinear systems in an errors-in-variables least squares and total least squares framework”. In: *Phdthesis, Vrije Universiteit Brussel* (1997) (cit. on p. 39).
- [112] Q.-c. Wang and J.-z. Zhang. “Wiener model identification and nonlinear model predictive control of a pH neutralization process based on Laguerre filters and least squares support vector machines”. In: *Journal of Zhejiang University Science C* 12.1 (2011), pp. 25–35 (cit. on p. 4).
- [113] D. T. Westwick and J. Schoukens. “Classification of the poles and zeros of the best linear approximations of wiener-hammerstein systems”. In: *IFAC Proceedings Volumes* 45.16 (2012), pp. 470–475 (cit. on pp. 6, 8, 45–47).

- [114] D. T. Westwick and J. Schoukens. “Initial estimates of the linear subsystems of Wiener–Hammerstein models”. In: *Automatica* 48.11 (2012), pp. 2931–2936 (cit. on pp. 6, 8, 47).
- [115] D. Westwick and M. Verhaegen. “Identifying MIMO Wiener systems using subspace model identification methods”. In: *Signal Processing* 52.2 (1996), pp. 235–258 (cit. on p. 3).
- [116] N. Wiener. “Nonlinear problems in random theory”. In: (1958) (cit. on p. 4).
- [117] T. Wigren and J. Schoukens. “Three free data sets for development and benchmarking in nonlinear system identification”. In: *Proc. 2013 Eur. Control Conf.(ECC2013)*. 2013, pp. 17–19 (cit. on p. 2).
- [118] A. Wills and B. Ninness. “Generalised Hammerstein–Wiener system estimation and a benchmark application”. In: *Control Engineering Practice* 20.11 (2012), pp. 1097–1108 (cit. on p. 47).
- [119] P. Young and A. Jakeman. “Refined instrumental variable methods of recursive time-series analysis Part I. Single input, single output systems”. In: *International Journal of Control* 29.1 (1979), pp. 1–30 (cit. on p. 55).
- [120] J. Zambrano, J. Sanchis, J. M. Herrero, and M. Martínez. “WH-MOEA: A Multi-Objective Evolutionary Algorithm for Wiener-Hammerstein System Identification. A Novel Approach for Trade-Off Analysis Between Complexity and Accuracy”. In: *IEEE Access* 8 (2020), pp. 228655–228674. DOI: [10.1109/ACCESS.2020.3046352](https://doi.org/10.1109/ACCESS.2020.3046352) (cit. on p. 12).
- [121] J. Zambrano, J. Sanchis, J. Herrero, and M. Martinez. “A Unified Approach for the Identification of Wiener, Hammerstein, and Wiener–Hammerstein Models by Using WH-EA and Multistep Signals”. In: *Complexity* 2020 (2020) (cit. on pp. 11, 96, 103).
- [122] J. Zambrano, J. Sanchis, J. Herrero, and M. Martínez. “WH-EA: an evolutionary algorithm for Wiener-Hammerstein system identification”. In: *Complexity* 2018 (2018) (cit. on pp. 10, 89, 95, 96).

- [123] Q. Zhang, Q. Wang, and G. Li. “Nonlinear modeling and predictive functional control of Hammerstein system with application to the turntable servo system”. In: *Mechanical Systems and Signal Processing* 72 (2016), pp. 383–394 (cit. on p. 2).
- [124] L. Zhou, X. Li, and F. Pan. “Gradient-based iterative identification for MISO Wiener nonlinear systems: application to a glutamate fermentation process”. In: *Applied Mathematics Letters* 26.8 (2013), pp. 886–892 (cit. on p. 3).
- [125] Y. Zhu. “Distillation column identification for control using Wiener model”. In: *Proceedings of the 1999 American Control Conference (Cat. No. 99CH36251)*. Vol. 5. IEEE, 1999, pp. 3462–3466 (cit. on p. 4).

