A Physics-Informed System Identification Approach with a novel MGGP Algorithm: a Hysteretic System Case Study

Henrique Carvalho de Castro, Bruno Henrique Groenner Barbosa

Universidade Federal de Lavras, MG, Brazil

Abstract

Nonlinear system identification can be a very hard task. Some nonlinearities, *e.g.*, the hysteretic behavior, require specific functions or terms in the model structure to be reproduced. Traditional methods for structure selection of nonlinear auto-regressive with exogenous inputs (NARX) models necessitates the assembly of a regressors matrix with all possible terms, which can become very costly in terms of memory usage and laborious if specific functions are needed. Therefore, this paper proposes a hybrid *Multi-Gene Genetic Programing / Forward Regression Orthogonal Estimator* (MGGP/FROE) framework to perform identification even for cases in which specific functions are required. We focus in the hysteretic system identification problem and show how the MGGP/FROE algorithm can be used to this specific problem by specializing the cost function. The identification of two case studies are performed and the algorithm has fulfilled its purpose of building models that reproduce hysteretic behavior.

Keywords: Nonlinear system identification; hysteresis; evolutionary algorithms; MGGP

1. Introduction

Dynamical models can be built directly from input and output data through a process known as *System Identification.* Generally, most of the real systems of interest are non-linear [1]. In this sense, NARX models (*Nonlinear AutoRegressive with eXogenous variables*)[2] are of great interest in the area, due to its flexibility and representation capacity. The main problem encountered in working with NARX models is to select the appropriate model structure, that is, detecting the regressors that together best represent the system. The key point of structure selection is to choose a model structure as simple as possible, but sufficiently complex to capture the dynamics underlying the data [3].

Email addresses: henriquec.castro@outlook.com (Henrique Carvalho de Castro), brunohb@ufla.com.br (Bruno Henrique Groenner Barbosa)

A widely used criterion for NARX model structure selection is the *Error Reduction Ratio* (ERR) [4], which evaluates how good each single model term is at explaining the output data variance. It is considered a one-step-ahead *prediction error minimization* (PEM) technique. Some algorithms were built based on ERR criterion for structure selection, as for example the *Forward Regression Orthogonal Estimator* (FROE) [4] and others *Orthogonal Least Squares* (OLS) based methods [5]. [6] discuss the limitations of ERR based algorithms, mainly for training data with the presence of certain input characteristics, and the use of *simulation error minimization* (SEM) techniques. Another issue is that such techniques suffer from the *curse of dimensionality* with the increment of the degree of non-linearity and higher long-term dependencies.

Alternative methods to solve the structure selection problem can be derived from *Evolutionary Algorithms* (EAs), as *Genetic Algorithm* (GA) [7, 8] and *Genetic Programming* (GP) [9]. However, most of these methods also depend on the assembling of a full regressors matrix for given maximum delays and non-linearity degree since they do not possess an automatic time lag determination and hence they may become computationally impracticable. One very flexible algorithm to be used in System Identification is the *Multi-Gene Genetic Programming* (MGGP) [10, 11, 12]. In a NARX context, each *locus* of an MGGP individual is a model term represented by a GP individual. It's features allow fluctuating individual sizes and automatic time lag determination, which dismiss the need for assembling a regressors matrix with all possible terms.

Recently, several modeling and forecasting works have been developed with the use of MGGP (see Section 4). The algorithm has shown itself to be very flexible and to present good performance in building models. In it, the user is able to determine the set of functions to be used in the modeling (as multiplication, division, exponential function, etc) and change it very easily, without the need for modifications in the algorithm itself. This flexibility in MGGP individual representation is the main feature explored in this paper.

Some non-linearities are known to be challenging for System Identification techniques. One of them is the hysteresis behavior, which presents a memory effect. Several works have investigated which features must be present in a NARX model to reproduce hysteresis [13, 14, 15, 16]. These features may be selected via traditional structure selection methods, such as FROE, by including them as inputs to the model. It demands the user to assemble manually each feature and its lagged versions. As consequence of the increment of inputs, this technique suffers from the *curse of dimensionality* which hinders its ability to identify long term dependencies and high degree of nonlinearity.

This paper proposes the use of a hybrid MGGP/FROE algorithm to solve the hysteretic system identification problem. Both algorithms (MGGP and FROE) work in a *symbiotic* way. From one side, the MGGP algorithm selects groups of terms from the candidate regressors space. To those groups is applied the FROE structure selection algorithm and the resultant models are assessed by a cost function. This interaction is supposed to facilitate the FROE task in a "divide and conquer" manner when the search space is very large, which avoids the curse of dimensionality. From the other side, the FROE algorithm works as a pruning method applied to the MGGP individuals. It leaves only relevant terms in all models. Thus, this interaction is supposed to guide the population evolution towards a promising region in the search space.

We develop a cost function specifically built to hysteretic system identification which makes our approach belong to the class of the physics-informed machine learning algorithms [17]. More specifically, it embeds model structure and parameters restrictions that guarantee some proprieties of hysteretic system already established in the literature [16]. Our algorithm is used in the identification of two hysteretic case studies, a theoretical system and a real-life benchmark data of piezoelectric actuator. The MGGP/FROE algorithm is capable of building models that reproduce hysteretic behavior and comply to the demanded restrictions.

The remainder of this paper is structured as follows: section 2 introduces basic concepts on system identification, section 3 describes the hysteretic system identification problem, section 4 presents the EAs, section 5 presents the proposed methodology, section 6 describes the two case studies on which the MGGP/FROE algorithm is applied, section 7 discuss the results and section 8 states the conclusion of the work. Finally, Appendix A is dedicated to the study of the integration of MGGP and FROE algorithms.

2. Nonlinear system identification

In this section we introduce basic concepts on system identification that is necessary to formulate the problem being solved, *i.e.* (i) mathematical description of the model representation, (ii) parameter estimation, (iii) structure selection, and (iv) model validation.

This paper presents a method to construct *polynomial* NARX models. They can be understood as a combination of delayed input and output signals. In other words, the current value of the model output is a combination of past input and output values. NARX models can be represented by the following equation:

$$y_k = f^{\ell}(y_{k-1}, \dots, y_{k-n_y}, u_{k-1}, \dots, u_{k-n_u}), \tag{1}$$

where f is a non-linear function, y_k and u_k are the output and input signals, respectively, and n_y and n_u their corresponding maximum delays. In the case of *polynomial* models, the non-linear function is a polynomial function of degree ℓ (f^{ℓ}).

As *polynomial* NARX models are linear-in-parameters, the parameter estimation can be made via the *Least Squares* (LS) estimator as follows:

$$\hat{\theta}_{LS} = (\Psi^T \Psi)^{-1} \Psi^T y, \tag{2}$$

where Ψ is the regressors matrix, y is the output data vector and $\hat{\theta}_{LS}$ the parameters estimated via LS. If the parameters are constrained by a set of restrictions, we can estimate them using the *Constrained Least Squares* (CLS).

Suppose a set of n_r restrictions that acts over the parameters. This set can be writen as:

$$\mathbf{c} = S\theta, \tag{3}$$

where $\mathbf{c} \in \mathbb{R}^{n_r \times 1}$ and $S \in \mathbb{R}^{n_r \times n_\theta}$ are known. The vector $\theta \in \mathbb{R}^{n_\theta \times 1}$ is given by [18]:

$$\hat{\theta}_{CLS} = (\Psi^T\Psi)^{-1}\Psi^T y - (\Psi^T\Psi)^{-1}S^T[S(\Psi^T\Psi)^{-1}S^T]^{-1}(S\hat{\theta}_{LS} - \mathbf{c}), \tag{4}$$

where Ψ is the regressors matrix and $\hat{\theta}_{LS}$ is the solution from equation (2).

It is important to notice that the amount of terms a polynomial model increases exponentially with the augment of non-linearity degree (ℓ) and of input (n_u) and output (n_y) maximum lags. According to [19], the maximum number of candidate terms is given by:

$$n_\theta = M + 1, \tag{5}$$

where n_θ is the number of terms in the model and

$$M = \sum_{l=1}^{\ell} n_l,$$

with

$$n_l = \frac{n_{l-1}(n_y + n_u + l - 1)}{l}, \quad n_0 = 1.$$

Moreover, an overparametrized model can lead to numerical instability in parameter estimation, unnecessary computational cost and the representation of dynamics that do not exist in the real system [3]. The aforementioned issues justify the importance of the structure selection step in system identification. In this sense, the most common structure selection methods are the FROE algorithms, in which the model structure is incremented iteratively until it is achieved a certain precision of one-step-ahead prediction. Orthogonalization techniques are made in such a way that, at each step, the relevance of each regressor candidate can be assessed separately via ERR:

$$[ERR]_j = \frac{\hat{g}_j^2 \sum_{k=1}^{N} w_j^2[k]}{\sum_{k=1}^{N} y_k^2}, \tag{6}$$

where w_j is the j-th auxiliary orthogonal regressor and $\hat{g}_j$ its corresponding estimated parameter. The regressors with the highest ERR are included in the model (see more in [5]). The FROE algorithm used in our methodology removes the terms that has a corresponding ERR value lesser than a tolerance parameter after the orthogonalization procedure. Alternative methods for structure selection of polynomial NARX models include EA's that are introduced in a further section.

We have just mentioned the terminology *one-step-ahead* (OSA) prediction. It is worth to state here that there are several methods to simulate a model, each of them yield prediction error estimations that have different properties and capabilities to assess model generalization. The standard simulation method is the OSA, however its predictions are not good indicators of models capability in explaining the system dynamics [CITATION]. A more effective method is the *free-run* simulation, which is a recursive estimator that predicts infinite steps ahead using the input signal and the model's previous estimated values. In between, there is a method called *multiple-shooting* simulation, in which the data is split into M intervals of size Δm, each one with its own initial conditions. It allows the usage of matrix form to simulate all intervals concomitantly. More details can be found in [20], where the authors compare the smoothness of the error space yielded by different simulation methods during parameter optimization process. *Multiple shooting* allows solving optimization problems that would be infeasible in a free-run simulation setting due to computational cost and complexity of the search space.

Finally, let us introduce in the following some model validation metrics. The *mean absolute percentage error* (MAPE) is given by

$$MAPE = \frac{100\sum_{k=1}^{N}|y_k - \hat{y}_k|}{N|max(\mathbf{y}) - min(\mathbf{y})|}, \tag{7}$$

where $\mathbf{y}$ is the measured data and $\hat{\mathbf{y}}$ is the model's estimated value. Additionally, we present the *multiple correlation coefficient* (R^2), which is given by

$$R^2 = \left[1 - \frac{\sum_{k=1}^{N}\xi_k^2}{\sum_{k=1}^{N}(y_k - \bar{y})^2}\right], \tag{8}$$

where $\xi_k = y_k - \hat{y}_k$ is the residual and $\bar{y}$ is the mean value of the sequence $\mathbf{y}$. The R^2 measures the adherence of the model to the measured data, such that $R^2 = 1$ means perfect data reconstruction and $R^2 > 0.9$ may be considered sufficient for many applications [21].

3. Modeling hysteresis

Hysteresis is a nonlinear behavior commonly related to phenomena such as ferromagnetism, plasticity, and friction, among others [22]. A hysteretic system has as intrinsic feature the memory effect, meaning that the output depends on the history of the corresponding input.

Representing hysteretic behavior through *polynomial* NARX models is a challenging task. In [15] it is presented a property that a model must attain to represent the main aspects of a hysteretic system:

Property 1. *An identified model of hysteresis, under a constant input, has two or more real non-diverging equilibria.*

This property can be used to obtain constraints on the structure and parameters of the model. [15] ensured that the identified model have at least one equilibrium point under loading-unloading inputs so that Property 1 is attained. To do so, the following general type model was used:

$$y_k = f^{\ell}(y_{k-1}, ..., y_{k-n_y}, ..., u_{k-1}, ..., u_{k-n_u}, \phi_{i,k-1}, ..., \phi_{i,k-n_\phi}) \tag{9}$$

where $\phi_{1,k} = u_k - u_{k-1}$, $\phi_{2,k} = sign(\phi_{1,k})$ and f^ℓ is a polynomial function of non-linearity degree ℓ, with $\phi_{2,k} = 1$ for loading and $\phi_{2,k} = -1$ for unloading.

Property 1 is satisfied if the model has a *continuum of equilibrium points* [16, 14], which can be verified if the sum of all parameters of all linear output regressors equals 1 ($\sum_y = 1$) and the sum of the parameters of all other parameter clusters equal zero, *e.g.* $\sum_{y^2} = 0$, $\sum_u = 0$, $\sum_{yu} = 0$, etc.

From now on, we will refer to $\phi_{1,k}$ and $\phi_{2,k}$ as ϕ-functions and disregard the subscript k since their structure will not be limited to the first difference. Instead we use the notation $\phi_{1,2}(u_{k-1}, u_{k-2})$, in which ϕ_1 is the *subtraction* and ϕ_2 is the *sign* functions.

4. Evolutionary algorithms

The EA's are based on metaphors of natural processes to build computational models capable of solving problems. There is a wide variety of proposed algorithms that simulate the evolution of species through the natural phenomena of selection, mutation and reproduction that occur in a given population of individuals. Generally, an EA has a standard behavior: it begins with a initial population of random individuals (chromosomes) and at each generation (main loop) the best solutions are sought (*selection*), then combined (*recombination/reproduction/crossover* and *mutation*) in order to generate even better individuals.

There are two points of extreme significance in EA: the genetic representation of the individuals and the evaluation function (or cost function). The genetic representation regards how the solutions are codified for a proper computer simulation. There are several ways to represent a solution and their choice depends on the problem being solved. We discriminate these genetic representations into two classes: (i) *chromosome-based* and (ii) *tree-based* representations.

A *chromosome-based* representation consists of a vector of values and each index of this vector corresponds to a specific feature of the solution. In this context we shall cite the *binary* and *real* representations. The former is a vector of 0's and 1's. It can represent boolean information (as for feature selection) or even a number. For example, an individual I represented by a 5 bit vector is codified as $I = \left[\begin{array}{ccccc} 1 & 0 & 0 & 1 & 0 \end{array}\right]$. In a NARX structure selection context, each bit can be interpreted as a Boolean variable which indicates whether the corresponding candidate term will be included in the model or will not. A *real* representation would be a vector of real values, each of them representing a specific parameter of the solution. An example of the use of this representation can be seen in [23], in which the authors implement a co-evolutionary algorithm for structure selection and parameter estimation of NARX Radial Basis Function Neural Networks (RBFNN) models to identify a piezoelectric robotic micromanipulator system. They use a Differential Evolution algorithm with a real chromosome-based representation for parameter estimation and a Harmony Search algorithm with integer chromosome-based representation for structure selection. More applications are found in [24, 25, 26]. The standard EA that uses this class of representation is the GA.

The *tree-based* representation form the basis for a branch of EA's known as GP. In this representation, from a root node, the tree is divided into several branches in which the internal nodes are composed of arithmetic functions (+, -, *, /, max, ...) and the terminals, also called leaves, are composed of variables and constants. As a result, this representation hierarchically synthesizes a mathematical function and it has the advantage over *chromosome-based* representation that the user does not need to know or specify the structure of the solution in advance. In the context of system identification, [27] used a multi-objective GP to build *polynomial* NARMAX models of benchmark systems with three objectives: (i) the OSA prediction error, (ii) the free-run simulation error and (iii) the model size, and parameters are estimated via LS. More examples can be seen in [28, 29, 30].

Frequently, it turns out in practice that using mixed representations is a more natural and suitable way of describing and manipulating a solution [31]. In this sense, the mixed *chromosome/tree-based* representation form the basis of the MGGP algorithm. In this representation, each solution is a vector of trees. Such structure fits naturally in a NARX/NARMAX context, since each tree (or GP individual) may correspond to one model term. An MGGP individual can be represented as the combination of separate basis functions:

$$g(\varphi, \Theta) = \sum_{i=1}^{m} \theta_i g_i(\varphi), \tag{10}$$

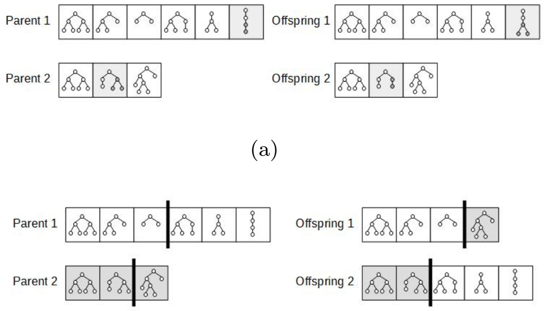
where m is the number of basis functions, g_i represents individual functions and θ_i the model parameters. It can be seen as a GA in which each *gene* contains one GP as the basis function (see more in [11] and [12]). [32] proposes a hybrid approach that combines MGGP and Capuchin Search Algorithm (CapSA) to formulate nonlinear identification models for a real-life winding process. The MGGP is applied first to find the structure of the symbolic regression models and then, the CapSA is used to find the most appropriate numeric constants for the regression models. More applications can be seen in [33, 34, 35, 36].

5. Proposed methodology

We develop in this work a hybrid MGGP/FROE algorithm for an automatic gray-box identification of hysteretic systems. For the sake of presentation, we divide this section into two parts: (i) the base algorithm and (ii) the evaluation algorithm. The integration between MGGP and FROE happens in the second part.

5.1. Base algorithm

As with any evolutionary algorithm, the MGGP starts with an initial population that evolves throughout generations via *genetic operators*. Regarding these operators, the MGGP works with two levels of crossover: *low-level crossover* and *high-level crossover*. In the former, one gene is randomly selected from each



(a)

(b)

Figure 1: MGGP two-level crossover. (a) low-level crossover (subtrees of a randomly selected *gene* are exchanged as genetic material) and (b) high-level crossover (entire *genes* are exchanged as genetic material from a randomly chosen cutoff point).

parent individual and they exchange GP subtrees as genetic materials (Figure 1(a)). In the latter, the genetic materials are exchanged as entire basis functions in a way similar to GA one-point crossover (Figure 1(b)). Note that in the *high-level crossover*, the resultant offspring can have different sizes from those of each other and even from those of their parents. This occurs because the selected point of crossover is chosen for each parent. In this sense, the MGGP algorithm works with fluctuating individual sizes.

We include two kinds of mutation in the algorithm: an inner mutation, which occurs as a GP subtree mutation, and an outer mutation, which swaps a gene for a new one, with an entirely new basis function. As recommended for GP evolution in [37], an individual experiences either crossover or mutation, *i.e.* both operators are not applied to the same individual. The elitism operator applied over the generations selects the best individuals from the union of the offspring population and the previous elite individuals.

A set of parameters must be configured:

- *population size*: defines the number of individuals present in the population;
- *crossover probability* (*CXPB*): defines the probability of a pair of individuals being combined together through a recombination operator;
- *mutation probability* (*MTPB*): defines the probability of a single individual being mutated through a mutation operator if it has not experienced recombination;
- *maximum GP height*: limits the size of a GP tree regarding its height;
- *maximum number of MGGP terms*: limits the model size regarding the number of terms that the model can possess;

- *elite size*: defines the percentage of individuals from the population that can remain in the next generation;
- *primitive functions*: the set of functions (multiplication, division, exponentiation, trigonometric operators, etc.) used as *nodes* in GP individuals. Additionally, the set of *back-shift operators* $(q^{-1}, q^{-2}, ...)$ responsible for the automatic time lag determination.
- *pruning probability* (PRPB): defines the probability of applying the pruning strategy (FROE) over an individual during evaluation step.
- *pruning tolerance*: defines the minimum ERR a term must yield to not be discarded.

Figure 2 exhibits the algorithm flowchart. It begins with an initial population that is evaluated. Then, the generation loop starts: *i)* the parent individuals are selected via tournament, *ii)* each parent couple has a chance to be recombined $(CXPB)$, *iii)* each individual that has not been recombined has a chance to be mutated $(MTPB)$, *iv)* the individuals are evaluated, and *v)* the elitism operator is applied. After the generation loop, the most significant individuals are validated and the one with the least validation error is selected as the system model. Details from the evaluation step are discussed next.

5.2. Evaluation algorithm

Traditionally, in this step there is no modification in the individual structure. Our MGGP/FROE algorithm is disruptive in this common sense. It uses a pruning strategy that reduces the individuals to only relevant terms during the evaluation step and it modifies subtrees from ϕ_1 and ϕ_2 root nodes to guarantee that individuals will have the proper structure. Thus the evaluation function comprises four events: *i)* check/modify ϕ-functions' arguments, *ii)* pruning, *iii)* parameter estimation and *iv)* fitness assignment.

The *cost function* is generically defined by:

```
Evaluation(individual):
        constrain_phi_functions
        if rand() < PRPB:
                apply_pruning_method      // FROE
        parameter_estimation              // CLS
        fitness_assignment                // MSE
```

Note that there are two design parameters to be set (as mentioned in the previous section), which are the probability (PRPB) of applying the FROE algorithm and its tolerance. In the following we explain each step.

- *Constrain ϕ-functions*: these functions should receive as arguments two lagged input variables, then the occurrence of any other variable or function as argument is forbidden. The undesired subtrees are replaced by a 1-lagged input variable. Figure 3 presents graphically the modification of a ϕ-function to a proper structure.

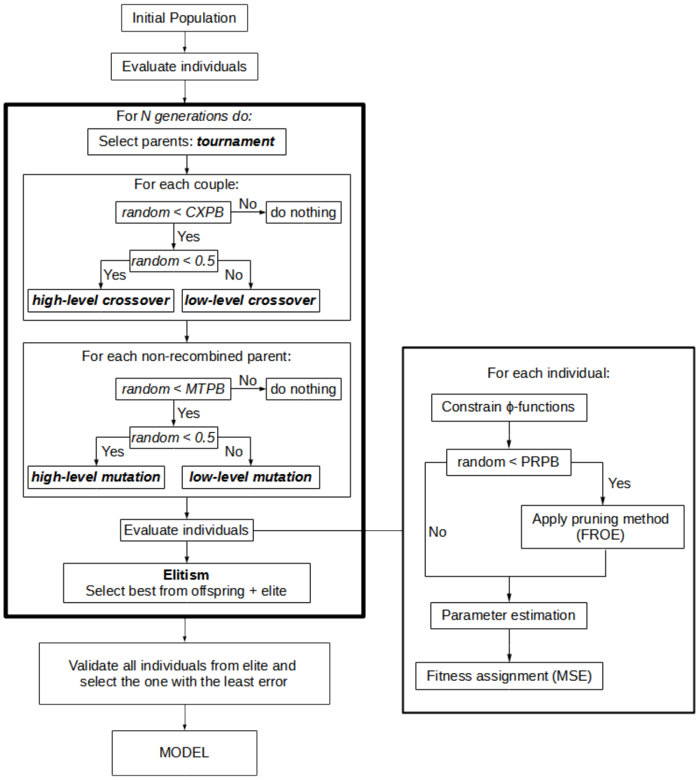


Figure 2: Algorithm flowchart

- *Pruning:* it is applied the FROE structure selection algorithm. Its tolerance regards the minimum ERR value a regressor must yield to be selected. The importance of pruning is discussed in Appendix A.

- *Parameter estimation:* to guarantee the *continuum of equilibrium points*, our method automatically identifies term clusters and apply the parameters restriction presented in section 3 using CLS estimator.

- *Fitness assignment:* the cost function is the MSE applied over the model's output after simulation. In previous experiments the algorithm was not able to build good models using OSA simulations. A free-run simulation would be better to evaluate whether models comprise the underlying dynamics on the data. However, the training data set is composed of a very large amount of samples, hence we have chosen to work with the *multiple-shooting* simulation. The number of steps-ahead (k) to be simulated is determined empirically and *ad hoc*.

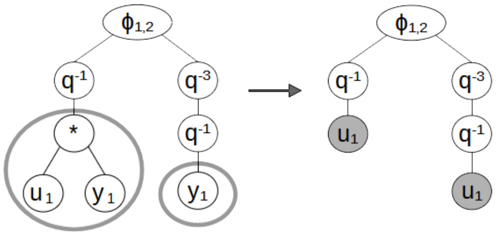


Figure 3: Constraining ϕ-functions. The multiplication operator and y variable are forbidden in the ϕ-function structure. Those subtrees are replaced by a 1-lagged u variable to form the model term $\phi_{1,2}(u_{k-2}, u_{k-5})$.

6. Case studies

In this section we present a description of the case studies on which we implement our methodology. As stated before, we are interested in the modeling of histeretic behavior. Then, the first case study is a theoretical model of a piezoelectric actuator and the second is a benchmark data from a real-life piezoelectric micromanipulator. Both of them comprises hysteretic behavior. Further, we present complementary information about algorithm parameters used in each case.

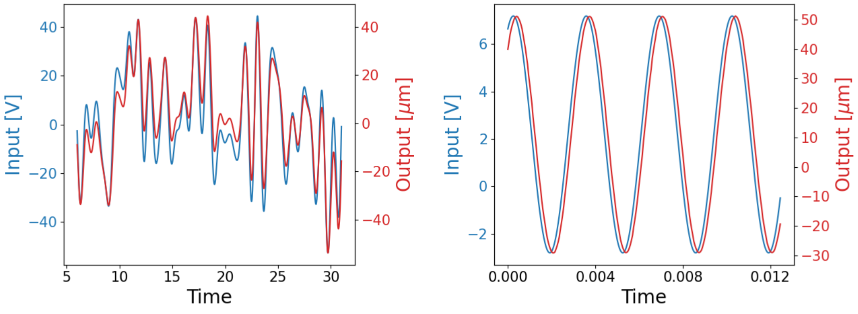
6.1. Piezoelectric actuator theoretical system

Piezoelectric actuators (PEAs) are used in micro- and nanopositioning applications. The resolution, fast responses, and large actuating forces are its main characteristics. This case study considers the piezoelectric actuator with hysteretic nonlinearity modeled by the Bouc-Wen model [38] and whose mathematical model is given by [39]:

$$\begin{cases} \dot{h}(t) = A\dot{u}(t) - \beta|\dot{u}(t)|h(t) - \gamma\dot{u}(t)|h(t)|, \\ y(t) = d_p u(t) - h(t) \end{cases} \tag{11}$$

where $y(t)$ is the displacement, $u(t)$ is the voltage applied to the actuator, $dp = 1.6\,\frac{\mu m}{V}$ is the piezoelectric coefficient, $h(t)$ is the hysteretic nonlinear term and $A = 0.9\,\frac{\mu m}{V}$, $\beta = 0.008\,V^{-1}$ and $\gamma = 0.008\,V^{-1}$ are parameters that determine the shape and scale of the hysteresis loop. Model (11) was integrated numerically using a fourth-order Runge–Kutta method with integration step $\delta t = 0.001\,s$.

The excitation signal is generated by low-pass filtering white Gaussian noise. In this work, a fifth-order low-pass Butterworth filter with a cutoff frequency of 1 Hz is used. The sampling time is set to $T_s = \delta t = 0.001\,s$, and the data set is $100\,s$ long (100000 samples). See Figure 4(a). This signal is used as training data. For the validation data, the system is excited by a quasi-static input $(u(t) = 40sin(2\pi t)\,V)$. These parameters that were used to build the data set are the same as the ones used in [16].



(a)

(b)

Figure 4: Piezoelectric actuator training data: (a) theoretical model and (b) real-life micromanipulator benchmark

An experimental framework based on OSA simulation yielded no promising results. A *free-run* setup should be the best to capture the dynamics underlying the training data. However, the dataset is composed of a very large number of samples and hence the *multiple shooting* simulation error is used in the cost function. Additionally, based on [15], two extra operators are included into the algorithm primitive functions set: the subtract (ϕ_1) and the sign (ϕ_2), both of them receive lagged input variables as arguments.

The parameters of the algorithm are set as follows:

```
population size = 500; generations = 500; elitism = 10%;
maximum GP height = 5; maximum MGGP terms = 10;
CXPB = 0.8; MTPB = 0.2 (if not crossover); PRPB = 0.5;
OLS tolerance = 1e-5; delay functions = q1, q2, q3, q4, q5
primitive functions: multiplication, subtraction, sign
fitness function = multiple shooting with
                   simulation interval of 500 steps
```

[15] claims that the sign of the first difference of the input in addition to polynomial terms is a sufficient condition to reproduce hysteresis. In this experiment, the algorithm is free to assemble ϕ-functions of any difference of the input variable, *e.g.*, $\phi_1(u_{k-2}, u_{k-4})$, $\phi_2(u_{k-6}, u_{k-3})$, etc.

Using ERR ranking together with AIC criterion and considering the constraints presented in section 3, [16] estimated the following NARX model structure:

$$\begin{aligned} y_k = {} & \theta_1 \cdot y_{k-1} + \theta_2 \cdot \phi_1(u_{k-1}, u_{k-2}) \\ & + \theta_3 \cdot u_{k-2} \cdot \phi_2(u_{k-2}, u_{k-3})\phi_1(u_{k-2}, u_{k-3}) \\ & + \theta_4 \cdot y_{k-1} \cdot \phi_2(u_{k-2}, u_{k-3})\phi_1(u_{k-2}, u_{k-3}) \\ & + \theta_5 \cdot u_{k-2}^2 \cdot \phi_1(u_{k-2}, u_{k-3}) \\ & + \theta_6 \cdot u_{k-2} y_{k-1} \cdot \phi_1(u_{k-2}, u_{k-3}). \end{aligned} \tag{12}$$

Its performance under our training and validation data set is presented in Table

1 and its parameters are shown in Table 2. We use this model as a means of comparison.

6.2. Piezoelectric micromanipulator benchmark

This case study considers a piezoelectric micromanipulator benchmark [40] composed of the displacement measurements acquired from a piezoelectric cantilevered actuator when subjected to harmonic excitation. The micro displacements are measured with optical sensors tuned to have $10\,nm$ resolution. The system is excited with a sine voltage input of 5 V and 300 Hz. The data is acquired with sample time of $50\,\mu$s. See Figure 4(b). This benchmark was recently used on nonlinear black-box modeling studies [23].

The parameters of the algorithm are set as follows:

```
population size = 500; generations = 500; elitism = 10%;
maximum GP height = 5; maximum MGGP terms = 10;
CXPB = 0.8; MTPB = 0.2 (if not crossover); PRPB = 0.5;
OLS tolerance = 1e-4; delay functions = q1, q2, q3, q4, q5
primitive functions: multiplication, subtraction, sign
fitness function = multiple shooting
                   simulation interval of 100 steps
```

It is important to highlight that the parameters setup in each case study are empirical and have not been optimized.

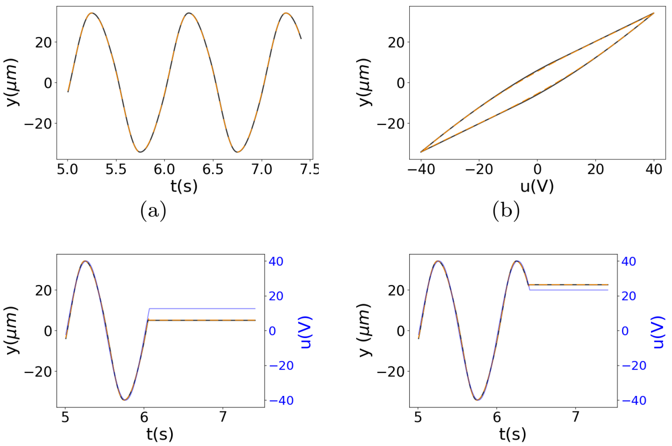
7. Results

In this section we present the resultant models of both case studies and their respective validation. For the theoretical system presented in subsection 6.1, we have achieved the following model:

$$\begin{aligned}
\mathcal{M}_{Case1}:\; y_k =\,&\theta_1 \cdot y_{k-1}\\
&+ \theta_2 \cdot u_{k-1}\phi_1(u_{k-12}, u_{k-1})\phi_2(u_{k-4}, u_{k-1})\\
&+ \theta_3 \cdot \phi_1(u_{k-9}, u_{k-13})\\
&+ \theta_4 \cdot y_{k-2}\phi_1(u_{k-12}, u_{k-10})\phi_2(u_{k-6}, u_{k-1})\\
&+ \theta_5 \cdot y_{k-8}, u_{k-8}\phi_1(u_{k-1}, u_{k-4})\\
&+ \theta_6 \cdot \phi_1(u_{k-10}, u_{k-6}) + \theta_7 \cdot \phi_1(u_{k-1}, u_{k-13})\\
&+ \theta_8 \cdot \phi_2(u_{k-15}, u_{k-12})\\
&+ \theta_9 \cdot u_{k-1}u_{k-12}\phi_1(u_{k-2}, u_{k-1}).
\end{aligned} \tag{13}$$

And for the real-life benchmark data set described in subsection 6.2, we have achieved the model:

$$\begin{aligned}
\mathcal{M}_{Case2}:\; y_k =\,&\theta_1 \cdot \phi_2(u_{k-5}, u_{k-10}) + \theta_2 \cdot \phi_2(u_{k-1}, u_{k-3})\\
&+ \theta_3 \cdot y_{k-6}u_{k-10} + \theta_4 \cdot y_{k-1}u_{k-4}\\
&+ \theta_5 \cdot u_{k-9} + \theta_6 \cdot u_{k-7}\\
&+ \theta_7 \cdot \phi_2(u_{k-9}, u_{k-2}) + \theta_8 \cdot \phi_1(u_{k-5}, u_{k-6})\\
&+ \theta_9 \cdot y_{k-1}.
\end{aligned} \tag{14}$$



(a)

(b)

(c)

(d)

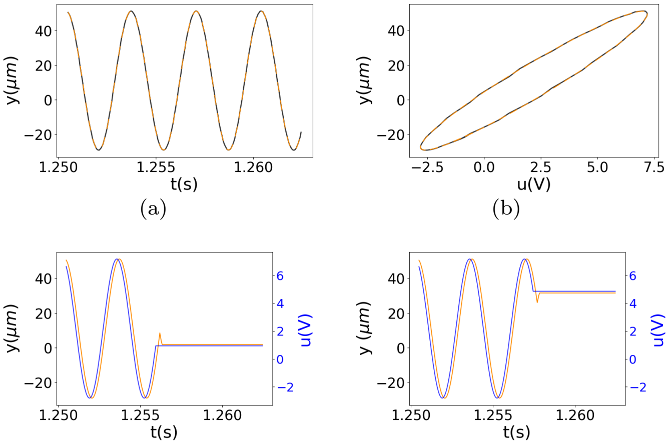
Figure 5: Free-run validation of model $\mathcal{M}_{Case1}$ (13) for quasi-static input data. (a) $y \times t$, (b) $y \times u$, and $y \times t$ for the cases in which input u becomes constant during (c) *loading* and (d) *unloading*. It is shown the system output measured data (—), the model's estimates (- - -) and the input (—).

Validation metrics are shown in Table 1. Model parameters are shown in Table 2. Figures 5 and 6 present the free-run validation for quasi-static input and for the cases in which the input becomes constant during *loading* and *unloading* for the models $\mathcal{M}_{Case1}$ (13) and $\mathcal{M}_{Case2}$ (14), respectively.

Lets take a closer look into these results. Remember that $R^2 = 1$ means perfect data reconstruction and that $R^2 > 0.9$ may be considered sufficient for many applications [21]. According to this criterion, both models $\mathcal{M}_{Case1}$ (13) and $\mathcal{M}_{Case2}$ (14) are considered good approximations of the systems being modeled since their free-run simulation of the validation data yielded R^2 measures of 0.99990 and of 0.99887, respectively. Further, our estimate of $\mathcal{M}_{Case1}$ performs better than the reference model shown in (12), mainly for validation data. We can see this in the MAPE measures for validation, which are of 0.25250 and of 0.33374, respectively. The simulations that correspond to these validation measures from our model estimates are shown in the Figures 5(a), 5(b) for $\mathcal{M}_{Case1}$ and in the Figures 6(a), 6(b) for $\mathcal{M}_{Case2}$. Notice that the hysteretic behavior for quasi-static input is reproduced in both case studies.

Model	MAPE (train)	MAPE (val.)	R^2 (train)	R^2 (val)	Size
Model (12)	0.32874	0.33374	0.99940	0.99984	6
$\mathcal{M}_{Case1}$ (13)	0.31189	0.25250	0.99940	0.99990	9
$\mathcal{M}_{Case2}$ (14)	1.01111	1.01039	0.99888	0.99887	9

Table 1: Piezoelectric models' free-run simulation errors



(a) (b)

(c) (d)

Figure 6: Free-run validation of model $\mathcal{M}_{Case2}$ (14) for quasi-static input data. (a) $y \times t$, (b) $y \times u$, and $y \times t$ for the cases in which input u becomes constant during (c) *loading* and (d) *unloading*. It is shown the system output measured data (—), the model's estimates (- - -) and the input (—).

Another important feature a model must comprise to correctly reproduce the dynamics of a hysteretic system is the *continuum of equilibrium points*. In summary, it means that when the input becomes constant, a model must yield constant estimates. It can be verified if the sum of all parameters of linear output terms equals 1, and the sum of all other clusters of parameters equals 0. Consider the input $u_k = \bar{u}$. We know that $\phi_{1,2}(\bar{u}, \bar{u}) = 0$, hence the parameters corresponding to terms that contain ϕ-functions have no influence in steady-state regime and their parameter clusters may be ignored. In model (13), we have only one term that does not contain a ϕ-function and it corresponds to a linear output term. Thus, $\sum_y = \theta_1 = 1$ (see Table 2). And in model (14), we have three clusters of interest: the linear output cluster ($\sum_y = \theta_9 = 1$), the linear input cluster ($\sum_u = \theta_5 + \theta_6 = 0$) and a nonlinear cluster ($\sum_{yu} = \theta_3 + \theta_4 = 0$). The parameters of both models comply to the required restrictions and hence they have a *continuum of equilibrium points* and satisfy Property 1. This property can be verified in the free-run simulations in which the input becomes constant during loading (Figures 5(c) and 6(c)) and unloading (Figures 5(d) and 6(d)).

8. Conclusion

In this paper, we presented a framework for gray-box nonlinear system identification using as basis the hybrid MGGP/FROE algorithm. The cost function of the optimization algorithm is developed specifically for hysteretic systems identification and includes restrictions to the structure and parameters of the solution. The framework is applied to two case studies, a theoretical system and a real-life benchmark data of piezoelectric actuator. We have shown that our methodology is capable of building models that reproduce hysteretic behavior and comply to restrictions that have been established in the literature.

Appendix A. Influence of pruning probability

FROE algorithms are known to be inadequate for systems excited by soft inputs [6], which is the case of the hysteretic system identified in this work. However, it may still be useful to guide population evolution in an EA context. Thus, in this experiment, the influence of *pruning probability* (PRPB) in the MGGP structure selection algorithm is analyzed. The models are selected from 50 Monte Carlo simulations with different probabilities for applying pruning (from 0% to 100%). The cost function uses OSA simulation and the primitive functions set is composed of only multiplication and back-shift operator (it is able to build standard polynomial NARX models). In the identification process, the test systems are excited by soft inputs, and the validation data are generated using a persistently exciting signal (ideal input). This is a challenging situation where the identification data do not carry all information about the system. Thus it is not expected for the algorithm to yield good models. In addition, we use a polynomial framework to model a rational test system (S_2).

The data are generated using the following output error (OE) test systems (from [6] and [12]):

$$\begin{aligned}
S_1: \quad & \tilde{y}[k] = 0.75y[k-2] + 0.25u[k-1] - 0.2y[k-2]u[k-1] \\
& y[k] = \tilde{y}[k] + e[k], \\
S_2: \quad & \tilde{y}[k] = \tilde{y}[k-1] \cdot \tilde{y}[k-2] \cdot \frac{2.5 + \tilde{y}[k-1]}{1 + \tilde{y}[k-1]^2 + \tilde{y}[k-2]^2} + u[k-11] \\
& y[k] = \tilde{y}[k] + e[k],
\end{aligned} \tag{A.1}$$

where $e[k]$ is white Gaussian noise with standard deviation $std(e) = 0.02$ ($SNR \approx 51.47$) for the system S_1 and with variance $var(e) = 0,04 \cdot var(\tilde{y})$ ($SNR \approx 28.05$) for the system S_2, and:

Model	Parameter values		
Model (12)	$\theta_1 = 1.00$	$\theta_2 = 0.77$	$\theta_3 = 1.44 \times 10^{-2}$
	$\theta_4 = -9.60 \times 10^{-3}$	$\theta_5 = 3.15 \times 10^{-4}$	$\theta_6 = -2.46 \times 10^{-4}$
$\mathcal{M}_{Case1}$	$\theta_1 = 1.00$	$\theta_2 = 1.35 \times 10^{-3}$	$\theta_3 = -4.17 \times 10^{-1}$
	$\theta_4 = -5.09 \times 10^{-3}$	$\theta_5 = -1.17 \times 10^{-4}$	$\theta_6 = -2.08 \times 10^{-1}$
	$\theta_7 = 1.31 \times 10^{-1}$	$\theta_8 = -1.87 \times 10^{-4}$	$\theta_9 = -4.36 \times 10^{-4}$
$\mathcal{M}_{Case2}$	$\theta_1 = 2.10 \times 10^{-2}$	$\theta_2 = 2.35 \times 10^{-2}$	$\theta_3 = 1.52 \times 10^{-4}$
	$\theta_4 = -1.52 \times 10^{-4}$	$\theta_5 = 3.14$	$\theta_6 = -3.14$
	$\theta_7 = -2.15 \times 10^{-2}$	$\theta_8 = 1.38 \times 10^{+1}$	$\theta_9 = 1.00$

Table 2: Piezoelectric actuator models' parameters.

- for the identification data, $u[k] = WGN(0; 0,20)$ that is processed by a low-pass filter with poles in 0,95 and 0,9;
- for the validation data, $u[k] = WGN(0; 1)$.

500 samples are used to compose training and validation data. The parameters of the algorithm are set as follows:

```
population size = 200; generations = 50; elitism = 10%;
CXPB = 0.8; MTPB = 0.2 (if not crossover); PRPB = 0-100%;
OLS tolerance = 1e-3; delay functions = q1, q2, q3
primitive functions: multiplication
fitness function: PE
for system S1:
maximum GP height = 3; maximum MGGP terms = 10;
for system S2:
maximum GP height = 5; maximum MGGP terms = 30.
```

The results of this experiment consist of: (i) the PE and SE performances obtained over the validation data with *ideal input*, (ii) the number of unstable models in free-run simulation (out of 50) and (iii) the model sizes over 50 Monte Carlo simulations for each $PRPB$ value.

Figures 7(a) and 7(c) present the results for the system S_1. Note that there is an improvement in terms variance of the PE and SE scores with the increment of *PRPB*. However, the boxes notches overlap hence nothing can be said about their medians. The number of unstable models (black dots) are complementary to SE scores (red boxes). For example, regarding $PRPB = 0.0$, there are 20 models (out of 50) that are BIBO unstable in free-run simulation for validation data, and these models are not represented in the red box-plot $SE \times PRPB$. Thus, for $PRPB = 1.0$ (in which there is only one unstable model), we can assume that the median and variance of SE are much smaller than the ones for $PRPB = 0.0$. It can be inferred that there is a significant improvement in the SE on the validation data with the increment of the $PRPB$. Moreover, regarding the *model size* $\times$ $PRPB$ boxplot (Figure 7(c) - green boxplot), the boxes notches do not overlap for $PRPB$ bigger than 0.5, which means that the median values of the models' sizes decrease with the increment of $PRPB$. It can be seen that for $PRPB = 0$ and for $PRPB = 0.25$, all 50 models are of size 10. It goes down to between 5 and 8 terms, with median of 6, for $PRPB = 1.0$.

Figures 7(b) and 7(c) present the results for the system S_2. Note that, for $PRPB = 0.0$, 49 models are BIBO unstable in free-run simulation for validation data. The improvement in the PE for this system is clearer than the one for system S_1. In Figure 7(b), the variance and median of the PE decreases with the increment of the $PRPB$. Note that the notches of $PRPB = 0.0$ and $PRPB = 0.5$ do not overlap. A Tukey's test is performed and it is shown on Table A.3. The distribution similarity is reject for the pairs $(0.00, 0.50)$, $(0.00, 0.75)$, $(0.00, 1.00)$, $(0.25, 0.75)$, $(0.25, 1.00)$. We can conclude that the benefits of FROE pruning are achieved when $PRPB \geq 0.50$ for *this system*.

The system S_2 has high degree of non-linearity (with rational structure) and long input lag term dependency ($u[k-11]$). The degree of nonlinearity necessary to represent this system in a *polynomial* structure is not known in advance. It should be emphasized that if the standard FROE algorithm was

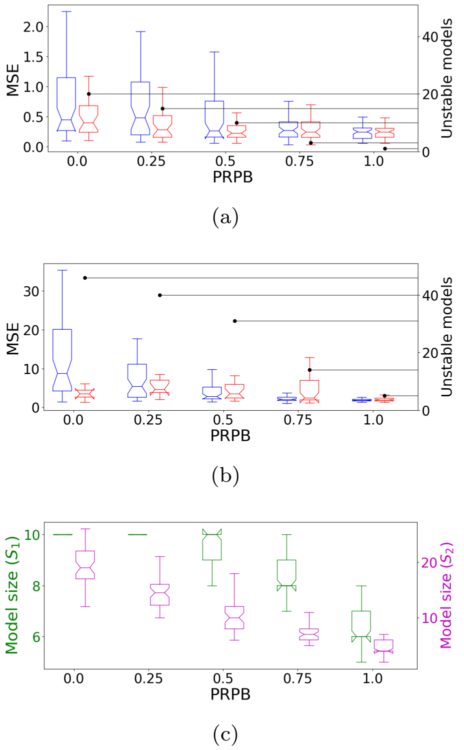


Figure A.7: Selected models results for the influence of pruning probability experiment: (a) PE and SE validation measures for system S_1 together with the number of BIBO unstable models (●); (b) the results for system S_2. In blue the OSA prediction error (PE) and in red the free-run simulation error (SE). In addition (c) the sizes of the models obtained for system S_1 (green) and for system S_2 (magenta).

applied in this example for structure selection using the parameters ($l = 5, n_y = 11, n_u = 11$), there would be 80730 regressor candidates to the model, which may be computationally impracticable for low performance computers.

We show that the use of *pruning* with the FROE algorithm is helpful to guide the evolution process in an MGGP population in the situation in which there is a *soft input* excitement. We show the results for only *soft inputs* used in the training step. If the structure selection is performed using *ideal inputs*, *i.e.* the input signal is able to excite the system such that all of its characteristics are present in the output data, there is no improvement in neither PE or SE scores with the increment of $PRPB$. However, the hybrid MGGP/ERR continues to yield smaller models, *i.e.* more parsimonious models.

References

[1] K. J. Pope, P. J. Rayner, Non-linear system identification using bayesian inference, in: Proceedings of ICASSP'94. IEEE International Conference on

Multiple Comparison of Means - Tukey HSD, FWER=0.05

group1	group2	meandiff	p-adj	lower	upper	reject
0.00	0.25	-6.1251	0.0961	-12.8852	0.6351	False
0.00	0.50	-10.9347	0.0010	-17.6948	-4.1745	True
0.00	0.75	-14.0503	0.0010	-20.8105	-7.2902	True
0.00	1.00	-15.0704	0.0010	-21.8306	-8.3103	True
0.25	0.50	-4.8096	0.2914	-11.5698	1.9506	False
0.25	0.75	-7.9252	0.0125	-14.6854	-1.1651	True
0.25	1.00	-8.9454	0.0031	-15.7055	-2.1852	True
0.50	0.75	-3.1157	0.6869	-9.8758	3.6445	False
0.50	1.00	-4.1358	0.4490	-10.8959	2.6244	False
0.75	1.00	-1.0201	0.9000	-7.7803	5.7400	False

Table A.3: Tukey's test for PE results from Figure 7(b) where groups are PRPB values.

Acoustics, Speech and Signal Processing, Vol. 4, IEEE, 1994, pp. IV–457.

[2] I. Leontaritis, S. A. Billings, Input-output parametric models for non-linear systems part i: deterministic non-linear systems, International journal of control 41 (2) (1985) 303–328.

[3] L. A. Aguirre, C. Letellier, Modeling nonlinear dynamics and chaos: a review, Mathematical Problems in Engineering 2009 (2009).

[4] S. Billings, S. Chen, M. Korenberg, Identification of mimo non-linear systems using a forward-regression orthogonal estimator, International journal of control 49 (6) (1989) 2157–2189.

[5] S. Chen, S. A. Billings, W. Luo, Orthogonal least squares methods and their application to non-linear system identification, International Journal of control 50 (5) (1989) 1873–1896.

[6] L. Piroddi, W. Spinelli, An identification algorithm for polynomial narx models based on simulation error minimization, International Journal of Control 76 (17) (2003) 1767–1781.

[7] J. H. Holland, Adaptation in natural and artificial systems, The University of Michigan Press (1975).

[8] D. E. Goldberg, J. H. Holland, Genetic algorithms and machine learning, Machine learning 3 (2) (1988) 95–99.

[9] J. R. Koza, Genetic programming: on the programming of computers by means of natural selection, Vol. 1, MIT press, 1992.

[10] M. Hinchliffe, M. Willis, H. Hiden, M. Tham, B. McKay, G. Barton, Modelling chemical process systems using a multi-gene genetic programming algorithm, in: Genetic Programming: Proceedings of the First Annual Conference (late breaking papers), 1996, pp. 56–65.

[11] M. P. Hinchliffe, Dynamic modelling using genetic programming, Ph.D. thesis, University of Newcastle upon Tyne, UK (2001).

[12] M. P. Hinchliffe, M. J. Willis, Dynamic systems modelling using genetic programming, Computers & Chemical Engineering 27 (12) (2003) 1841 – 1854.

[13] L. Deng, Y. Tan, Modeling hysteresis in piezoelectric actuators using narmax models, Sensors and Actuators A: Physical 149 (1) (2009) 106–112.

[14] K. Morris, What is hysteresis?, Applied Mechanics Reviews 64 (5) (2011).

[15] S. A. M. Martins, L. A. Aguirre, Sufficient conditions for rate-independent hysteresis in autoregressive identified models, Mechanical Systems and Signal Processing 75 (2016) 607–617.

[16] P. E. Abreu, L. A. Tavares, B. O. Teixeira, L. A. Aguirre, Identification and nonlinearity compensation of hysteresis using narx models, Nonlinear Dynamics 102 (1) (2020) 285–301.

[17] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, L. Yang, Physics-informed machine learning, Nature Reviews Physics 3 (6) (2021) 422–440.

[18] N. R. Draper, H. Smith, Applied regression analysis, 3rd Edition, John Wiley & Sons, 1998.

[19] M. Korenberg, S. A. Billings, Y. P. Liu, P. J. McILroy, Orthogonal parameter estimation algorithm for non-linear stochastic systems, International Journal of Control 48 (1) (1988) 193–210.

[20] A. H. Ribeiro, K. Tiels, J. Umenberger, T. B. Schön, L. A. Aguirre, On the smoothness of nonlinear system identification, Automatica 121 (2020) 109158. `doi:10.1016/j.automatica.2020.109158`.

[21] B. Schaible, H. Xie, Y.-C. Lee, Fuzzy logic models for ranking process effects, IEEE Trans. Fuzzy Syst. 5 (4) (1997) 545–556.

[22] A. Visintin, Differential Models of Hysteresis, Springer, Berlin Heidelberg, 1994.

[23] H. V. H. Ayala, D. Habineza, M. Rakotondrabe, L. dos Santos Coelho, Nonlinear black-box system identification through coevolutionary algorithms and radial basis function artificial neural networks, Applied Soft Computing 87 (2020) 105990.

[24] E. Oroski, R. H. Lopez, A. Bauchspiess, Identification of a magnetic levitator using narx-obf models and genetic algorithm, International Journal of Modelling, Identification and Control 28 (4) (2017) 307–316.

[25] M. Jawad, S. M. Ali, B. Khan, C. A. Mehmood, U. Farid, Z. Ullah, S. Usman, A. Fayyaz, J. Jadoon, N. Tareen, et al., Genetic algorithm-based nonlinear auto-regressive with exogenous inputs neural network short-term and medium-term uncertainty modelling and prediction for electrical load and wind speed, The Journal of Engineering 2018 (8) (2018) 721–729.

[26] F. Hafiz, A. Swain, E. M. Mendes, N. Patel, Structure selection of polynomial narx models using two dimensional (2d) particle swarms, in: 2018 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2018, pp. 1–8.

[27] D. Khandelwal, M. Schoukens, R. Tóth, Data-driven modelling of dynamical systems using tree adjoining grammar and genetic programming, in: 2019 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2019, pp. 2673–2680.

[28] J. Chadalawada, V. Havlicek, V. Babovic, A genetic programming approach to system identification of rainfall-runoff models, Water Resources Management 31 (12) (2017) 3975–3992.

[29] V. H. C. Pinheiro, R. Schirru, Genetic programming applied to the identification of accidents of a pwr nuclear power plant, Annals of Nuclear Energy 124 (2019) 335–341.

[30] J. Im, C. B. Rizzo, F. P. de Barros, S. F. Masri, Application of genetic programming for model-free identification of nonlinear multi-physics systems, Nonlinear Dynamics 104 (2) (2021) 1781–1800.

[31] A. E. Eiben, J. E. Smith, et al., Introduction to evolutionary computing, Vol. 53, Springer, 2003.

[32] M. Braik, A hybrid multi-gene genetic programming with capuchin search algorithm for modeling a nonlinear challenge problem: Modeling industrial winding process, case study, Neural Processing Letters 53 (4) (2021) 2873–2916.

[33] A. D. Mehr, E. Kahya, A Pareto-optimal moving average multigene genetic programming model for daily streamflow prediction, Journal of Hydrology 549 (2017) 603–615.

[34] M. J. S. Safari, A. D. Mehr, Multigene genetic programming for sediment transport modeling in sewers for conditions of non-deposition with a bed deposit, International Journal of Sediment Research 33 (3) (2018) 262–270.

[35] M. G. De Giorgi, M. Quarta, Hybrid multigene genetic programming - artificial neural networks approach for dynamic performance prediction of an aeroengine, Aerospace Science and Technology 103 (2020) 105902.

[36] E. Derner, J. Kubalík, N. Ancona, R. Babuška, Constructing parsimonious analytic models for dynamic systems via symbolic regression, Applied Soft Computing 94 (2020) 106432.

[37] R. Poli, W. B. Langdon, N. F. McPhee, A field guide to genetic programming, Published via `http://lulu.com` and freely available at `http://www.gp-field-guide.org.uk`, 2008, (With contributions by J. R. Koza).

[38] Y. K. Wen, Method for Random Vibration of Hysteretic Systems, Journal of the Engineering Mechanics Division, 1976.

[39] M. Rakotondrabe, Bouc-Wen Modeling and Inverse Multiplicative Structure to Compensate Hysteresis Nonlinearity in Piezoelectric Actuators, IEEE Transactions on Automation Science and Engineering, 2011.

[40] H. V. H. Ayala, M. Rakotondrabe, L. dos Santos Coelho, Piezoelectric micromanipulator dataset for hysteresis identification, Data in brief 29 (2020) 105175.