Postgraduate Program in Electrical Engineering - PPGEL

Association between UFSJ and CEFET

Meta Model Structure Selection: An Algorithm For Building Polynomial NARX Models For Regression And Classification

Wilson Rocha Lacerda Junior

Supervisor: Samir Angelo Milani Martins

Co-Supervisor: Erivelton Geraldo Nepomuceno

February 2020.

Postgraduate Program in Electrical Engineering - PPGEL

Association between UFSJ and CEFET

Meta Model Structure Selection: An Algorithm For Building Polynomial NARX Models For Regression And Classification

Wilson Rocha Lacerda Junior

Dissertation presented to the Federal University of São João del-Rei Postgraduate Program in Electrical Engineering, broad association between the Federal University of São João del-Rei and the Federal Center for Technological Education of Minas Gerais, required to obtain the Master's degree title.

Field: Modeling and Control Systems

Main Line of Research: System Analysis and Modeling

Research Group: GCOM - Control and Modelling Group

Supervisor: Samir Angelo Milani Martins

Co-supervisor: Erivelton Geraldo Nepomuceno

February 2020.

Programa de Pós-Graduação em Engenharia Elétrica - PPGEL

Associação ampla UFSJ / CEFET-MG

Meta-MSS: Um Algoritmo Para Obtenção de Modelos NARX Polinomial Para Regressão e Classificação

Wilson Rocha Lacerda Junior

Dissertação apresentada à banca examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica, associação ampla entre a Universidade Federal de São João del-Rei e o Centro Federal de Educação Tecnológica de Minas Gerais, como requisito necessário à obtenção do grau de Mestre em Engenharia Elétrica.

Área de concentração: Modelagem e Controle de Sistemas

Linha de Pesquisa: Análise e Modelagem de Sistemas

Grupo de Pesquisa: GCOM - Grupo de Controle e Modelagem

Orientador: Samir Angelo Milani Martins

Coorientador: Erivelton Geraldo Nepomuceno

Fevereiro, 2020.

Ficha catalográfica elaborada pela Divisão de Biblioteca (DIBIB)
e Núcleo de Tecnologia da Informação (NTINF) da UFSJ,
com os dados fornecidos pelo(a) autor(a)

Lacerda Junior, Wilson Rocha.
L131m Meta Model Structure Selection : An Algorithm For Building Polynomial NARX Models For Regression And Classification / Wilson Rocha Lacerda Junior ; orientador Samir Angelo Milani Martins; coorientador Erivelton Geraldo Nepomuceno. -- São João del-Rei, 2020.
146 p.

Dissertação (Mestrado - Engenharia Elétrica) -- Universidade Federal de São João del-Rei, 2020.

1. System Identification. 2. NARX Model. 3. Structure Selection. 4. Nonlinear Systems. 5. Meta heuristics. I. Martins, Samir Angelo Milani, orient. II. Nepomuceno, Erivelton Geraldo, co-orient. III. Título.

In loving memory of my father, Wilson Rocha; to my courageous mother, Maria Lúcia;

and to Mateus and Mariana.

ACKNOWLEDGEMENTS

Foremost, I wish to thank my parents, Maria Lúcia and Wilson Rocha (in memorian), my brothers, Mateus Emerick e Mariana Emerick and my uncle, Neka. They raised me, supported me, taught me, loved me and always managed to make me feel special. To them I dedicate this dissertation.

A warm word for my beloved girlfriend Nath, who expanded my worlds and made of this a memorable experience. Thank you for the support, fellowship, entertainment and caring.

Some special words of gratitude go to my friends who have always been a major source of support when things would get a bit discouraging. Thank you for all the emotional support and encouragement.

I would like to thank my friend and advisor Prof. Samir Angelo Milani Martins for his incredibly support along the way. He consistently allowed this thesis to be my own work, but steered me in the right the direction whenever he thought I needed it. I look forward to our continued collaboration.

Last but not least, I would like to thank also:

- my assistant advisor Prof. Erivelton Geraldo Nepomuceno, for his support offered during the preparation of my thesis;
- To CAPES for financially supporting me during the Master's program.
- To the friends from GCoM for creating a great research environment and the wonderful "Sunday Lunch of the Freemasons";

"If they can get you asking the wrong questions, they don't have to worry about answers."

Thomas Pynchon

"The master has failed more times than the beginner has even tried."

Stephen McCranie

CONTENTS

Resumo xvii

Abstract xix

List of Figures xxii

List of Tables xxiv

List of Symbols xxv

List of Abbreviations and Acronyms xxvii

1 Introduction 1

1.1 Objectives and contributions 4

1.2 Publications 5

1.3 Thesis Organization 5

2 Background 7

2.1 Input Signals 7

2.2 Choice of representation: the NARMAX model 8

2.2.1 Equilibria in NARX models 11

2.2.2 Modeling hysteresis with polynomial NARX model 13

2.3 Structure Selection Techniques 18

2.3.1 The Forward Regression Orthogonal Least Squares Algorithm 20

2.4 Parameter Estimation using LS 23

2.4.1 Affine information and multiobjective approach 24

2.4.2 Validation 26

2.5 Meta-heuristics 27

2.5.1 Standard Particle Swarm Optimization (PSO) 29

2.5.2 Standard Gravitational Search Algorithm . . . 31
2.5.3 Binary PSO and GSA . . . 33
2.5.4 The hybrid PSOGSA algorithm . . . 34
2.5.5 Binary version of PSOGSA . . . 36
2.6 A brief review of the State-of-the-Art . . . 36
2.6.1 A randomized algorithm for nonlinear model structure selection . . . 37
2.6.2 An ultra orthogonal forward regression algorithm . . . 38
2.7 Meta-heuristics applied to system identification . . . 40
2.8 Improved Structure Detection For Polynomial NARX Models Using a Multiobjective Error Reduction Ratio . . . 42
2.8.1 Determination of the number of terms in polynomial models: the Multi-objective Akaike Information Criterion . . . 43
2.9 Equivalence of non-linear model structures based on Pareto uncertainty . . . 45

3 Mono-objective Framework for Regression and Classification 49
3.1 Encoding scheme . . . 49
3.1.1 Formulation of the objective function . . . 51
3.2 Case Studies: Simulation Results . . . 55
3.2.1 Influence of the max_iter and n_agents parameters . . . 57
3.2.2 Selection of over and sub-parameterized models . . . 58
3.2.3 Meta-MSS vs RaMSS vs C-RaMSS . . . 59
3.2.4 Meta-MSS vs FROLS . . . 59
3.2.5 Meta-MSS vs RJMCMC [1] . . . 60
3.2.6 Meta-MSS vs Meta-heuristic method proposed by [2] vs FROLS . . . 61
3.2.7 Full-scale F-16 aircraft . . . 62
3.3 Conclusions for Mono-objective framework for regression and classification . . . 65

4 Multi-objective identification using the Meta-MSS algorithm 69
4.1 Description of a NARX model static behavior . . . 69
4.2 MoMeta-MSS algorithm . . . 70
4.3 Electrical Heater . . . 71
4.4 DC-DC Buck converter . . . 76

4.5 Conclusions for Multi-objective identification using the Meta-MSS algorithm . 79

5 NARX models with multiple inputs for hysteretic system identification 81

5.1 A Brief description of the Bouc-Wen model of magneto-rheological damper device 81

5.1.1 Modeling of magneto-rheological damper devices using Meta-MSS: identification of hysteresis loops considering models with a single equilibrium point. 83

5.2 Representation of RIH in NARX models with multiple inputs 84

5.2.1 Modeling of magneto-rheological damper devices using Meta-MSS: identification of hysteresis loops considering models with two equilibrium point. 86

5.3 Conclusions for NARX models with multiple inputs for hysteretic system identification 89

6 Meta-MSS algorithm for classification problems 91

6.1 Logistic NARX modeling approach using Meta-MSSc algorithm 92

6.1.1 Theoretical framework for Logistic NARX 92

6.2 Example 1 93

6.3 Example 2 96

6.4 Electroencephalography Eye State Identification 98

6.5 Conclusions for Meta-MSS Algorithm for Classification Problems 100

7 Final Considerations 103

7.1 Future Work 104

Bibliography 107

RESUMO

A família de modelos polinomiais Não linear AutoRegressivo com entradas eXógenas (NARX) tem sido estudada extensivamente nas últimas três décadas e diversas técnicas tem sido propostas em relação a esses modelos. Os trabalhos tradicionais na área têm focado principalmente em métodos baseados em um procedimento progressivo conhecido como Mínimos Quadrados Ortogonais com Regressão Direta (FROLS). Recentemente, novas técnicas têm sido propostas baseadas em algoritmos evolutivos. Estes métodos incorporam o conhecimento sobre distribuição de probabilidade de possíveis estruturas de modelos ao processo de busca. Uma classe de algoritmos especialmente interessante é baseada em meta-heurísticas. Nesta dissertação um método baseado na meta-heurística Híbrida e Binária de Otimização por Enxame de Partículas e Busca Gravitacional é apresentado para seleção de estruturas de modelos mono-objetiva e multiobjetiva. O método leva em consideração a complexidade do modelo e a contribuição individual de cada termo para gerar modelos apropriados por meio de uma nova formulação da função custo do problema. Além disso, o método proposto é adaptado para obtenção de modelos em situações em que a saída é representada por uma variável dicotômica, resultando em um modelo de classificação de dados. A robustez do novo algoritmo é testada em vários sistemas, simulados e experimentais, com diferentes características não lineares. Os resultados obtidos mostram que o algoritmo proposto é capaz de identificar o modelo correto, nos casos em que a estrutura correta é conhecida, e determinar modelos parcimoniosos para dados experimentais mesmo naqueles sistemas para os quais métodos tradicionais e contemporânios falham ocasionalmente. O novo algoritmo é comparado com métodos clássicos como o FROLS e com novas abordagens baseadas em meta-heurísticas.

Palavras-chave: Identificação de Sistemas, Modelos NARX, Seleção de estruturas, Sistemas não lineares, Meta-heurística.

ABSTRACT

The polynomial Nonlinear Autoregressive with eXogenous inputs (NARX) model family has been extensively studied over the last three decades, and several techniques have been proposed in this respect. The traditional work has mainly focused on methods based on a stepwise procedure known as Forward-Regression Orthogonal Least Squares (FROLS) algorithm. Extensions to model with multiples input and output have been stated as well as developing multi-objective approaches. Recently, the identification framework has been carried out in a probabilistic pattern, and contemporary techniques have been proposed consisting of evolutionary algorithms. These methods incorporate the knowledge about the probability distribution of possible model structures into the search process. An especially exciting class of new algorithm is the one based on metaheuristic. In this dissertation, a variant meta-heuristic method based on Hybrid Binary Particle Swarm Optimization and Gravitational Search Algorithm is presented for a single objective and multi-objective model structure selection. It takes into account the complexity of the model and the contribution of each term to build parsimonious models by proposing a new cost function formulation. In addition, the proposed method is adapted to obtain models in situations where the output is represented by a dichotomous variable, resulting in a classification model. The robustness of the new algorithm is tested on several simulated and experimental system with different nonlinear characteristics. The obtained results show that the proposed algorithm is capable of identifying the correct model, for cases where the model structure is known, and determine parsimonious models for experimental data even for those systems for which traditional and contemporary methods habitually fails. The new algorithm is validated over classical methods such as the FROLS and recent approaches based on meta-heuristic.

Keywords: System identification, NARX model, Structure selection, Nonlinear Systems, Metaheuristics.

LIST OF FIGURES

2.1 MIMO model decomposed into MISO models. 11
2.2 Hysteresis loop 15
2.3 Example of a bounding structure $\mathcal{H}$ 17
2.4 RLC model 19
2.5 A hypothetical Pareto front of a bi-objective problem. 25

3.1 The derivative of the sigmoid function looks like an steeper and "overshooting" sigmoid function. The red line $(-)$ represents the sigmoid function and the blue line $(--)$ represents its derivative. 53
3.2 Comparison of the convergence curves 66
3.3 Distribution of terms number 67
3.4 Complete aircraft structure. Source: [3] 67
3.5 Frequency of the eye state 68
3.6 Case Study: F-16 aircraft benchmark 68

4.1 The steady state behavior of the electrical heater. The continuous line is to facilitate the visualization of its nonlinear characteristic. 73
4.2 The Pareto Set obtained by incorporating the affine information in the selection procedure. Each $(\circ)$ represents a model the model with specific parameters. . . 74
4.3 Dynamic and static simulation of the Electrical Heater model 75
4.4 The steady state behavior of the electrical heater. The continuous line is to facilitate the visualization of its nonlinear characteristic. 76
4.5 The Pareto Set obtained by incorporating the affine information in the selection procedure. Each $(\circ)$ represents a model the model with specific parameters. . . 77
4.6 Dynamic and static simulation of the Electrical Heater model 78

5.1 The model for a magneto-rheological damper proposed by [4]. 82
5.2 Meta-MSS performance in the Bouc-Wen system identification 84

5.3 Example of hysteresis loop with two equilibria 86
5.4 Meta-MSS performance in the Bouc-Wen system identification 88

6.1 Data points obtained from the model given in equation (6.5). Points in blue corresponds to class $y = 0$ and points in red corresponds to $y = 1$. There is not a clear boundary separating the two classes. 95
6.2 Data points obtained from the model given in equation (6.6). Points in blue corresponds to class $y = 0$ and points in red corresponds to $y = 1$. There is not a clear boundary separating the two classes. 97
6.3 Emotiv EEG neuroheadset sensor position and corresponding behavior groups. Source: [3] 99
6.4 Frequency of the eye state 99

LIST OF TABLES

3.1 Parameters used in Meta-Structure Selection Algorithm 56
3.2 Overall performance of the Meta-MSS 56
3.3 Results regarding the influence of the max_iter and n_agents parameters 57
3.4 Comparative analysis 59
3.5 Comparative analysis - Meta-MSS vs FROLS 60
3.6 Comparative analysis - Meta-MSS vs RJMCMC 61
3.7 Comparative analysis - Meta-MSS vs Meta-heuristic method proposed by [2] vs FROLS 62
3.8 Parameters used in Meta-Structure Selection Algorithm for the F-16 benchmark 63
3.9 Identified NARX model using Algorithm 6 and FROLS. 64

4.1 Parameters used in Multi-objective Meta-Structure Selection Algorithm 73
4.2 Identified NARX model using Algorithm 7. The blue row of parameters shown in this table are the ones of the optimal model obtained using the minimum Euclidean norm of the normalized objectives. 73
4.3 The RMSE values obtained for each parameter set. The blue row corresponds to the optimal model obtained using the minimum Euclidean norm of the normalized objectives. 74
4.4 Models obtained using the Meta-MSS and the FROLS algorithm. The Meta-MSS returned a model with a lower number of regressors, but with higher non-linearity degree. The FROLS returned a model with two terms belonging to Σ_{yx} cluster while the one obtained using Meta-MSS has no cross terms between the input and output. 75
4.5 Parameters used in Multi-objective Meta-Structure Selection Algorithm 76
4.6 Identified NARX model using Algorithm 7. The blue row of parameters shown in this table are the ones of the optimal model obtained using the minimum Euclidean norm of the normalized objectives. 77

4.7 The RMSE values obtained for each parameter set. The blue row corresponds to the optimal model obtained using the minimum Euclidean norm of the normalized objectives. 77
4.8 Models obtained using the Meta-MSSm, the FROLS and the MERR. The models have a similar number of regressors. 79

5.1 Parameters for a MR damper 82
5.2 Meta-Structure Selection Algorithm parameters used in the hysteretic system identification. 83
5.3 Models obtained using the Meta-MSS and the cluster-based structure selection algorithm. 84
5.4 Meta-Structure Selection Algorithm parameters used in the hysteretic system identification. 87
5.5 NARX model with two equilibrium points obtained using the Meta-MSS. . . . 87

6.1 Parameters used in Meta-Structure Selection Algorithm for binary classification 95
6.2 Identified NARX model using Algorithm 8. All model terms selected correspond to the true model. The parameters obtained are log odds ratios and therefore they not need to be similar to the ones in original model. 95
6.3 Accuracy performance on simulated data - Example 1 96
6.4 Parameters used in Meta-Structure Selection Algorithm for binary classification 97
6.5 Identified NARX model using Algorithm 8. All model terms selected correspond to the true model. The parameters obtained are log odds ratios and therefore they not need to be similar to the ones in the original model. 98
6.6 Accuracy performance on simulated data - Example 2 98
6.7 Identified NARX model using Algorithm 8. This model was built using the original dataset using the Meta-MSSc. 100
6.8 Accuracy performance between different methods for modelling of equation (6.5) 101

LIST OF SYMBOLS

$n_{_{m}r}$ Number of potential regressors for MIMO models 19

F Arbitrary power-form polynomial function 8

$\Omega_{y^p x^m}$ Term cluster of polynomial NARX 12

ℓ non-linearity degree of the model 8, 9, 10, 19, 40, 41, 49

$\hat{\Theta}$ Estimated Parameter Vector 20

$\hat{y}_k$ model predicted output at discrete time k 26

$\mathbf{X}_k$ Column vector of multiple system inputs at discrete-time k 10

$\mathbf{Y}_k$ Column vector of multiple system outputs at discrete-time k 10

$\mathcal{H}_t(\omega)$ Hysteresis loop of the system in continuous-time 15

$\mathcal{H}$ Bounding structure that delimits the system hysteresis loop 13, 15, 17

ρ Tolerance value 39

$\sum_{y^p x^m}$ Cluster coefficients of polynomial NARX 12

e_k error vector at discrete-time k 8

n_r Number of potential regressors for SISO models 8, 19

n_x maximum lag of the input regressor 8, 9, 19, 49

n_y maximum lag of the output regressor 8, 9, 19, 49

n number of observations in a sample 18, 22

x_k system input at discrete-time k 8, 18

y_k system output at discrete-time k 8, 18

LIST OF ABBREVIATIONS AND ACRONYMS

AIC Akaike Information Criterion 2, 3, 22, 40, 41, 44, 61, 62

BA Bat Algorithm 40, 41

BIC Bayesian Information Criterion 2, 3

BPSOGSA Binary Hybrid Particle Swarm Optimization and Gravitational Search Algorithm 36, 49, 56

e.g. Abbreviation for *exempli gratia*: a Latin phrase that means *for example*. 2, 3, 4, 6, 7, 8, 18, 22, 28, 37, 41, 42, 99

ELS Extended Least Squares 43

ERR Error Reduction Ratio 2, 19, 22, 39, 42, 43, 44, 62, 92

FPE Final Prediction Error 2, 3

FROLS Forward-Regression Orthogonal Least Squares 2, 3, 5, 19, 20, 37, 38, 40, 44, 59, 61, 62, 63, 64, 74, 77, 78, 91, 92, 100

GA Genetic Algorithm 28, 29, 40, 41

GSA Gravitational Search Algorithm 29, 31, 33, 34, 35

iOFR Iterative Orthogonal Foward-Regression 37, 38

LS Least Squares 1, 20, 22, 23, 24, 25, 26, 38, 39, 40, 50, 51, 93

MAIC Multi-objective Akaike Information Criterion 44

MCMC Markov Chain Monte Carlo 3

MERR Multi-objective Error Reduction Ratio 5, 42, 43, 77, 78

MIMO Multiple Input Multiple Output 4, 10, 19, 27

MISO Multiple Input Single Output 10

MoMSS Multi-objective Model Structure Selection 3

MOP Multi-objective Optimization Problem 24

MSE Mean Square Error 61

MSS Model Structure Selection 2, 3, 12, 13, 18, 19, 36, 37, 38, 40, 42, 43, 45, 47, 48, 81

NARMAX Nonlinear Auto-Regressive Moving Average Model with eXogenous inputs 1, 2, 3, 4, 8, 36, 38

NARX Nonlinear Auto-Regressive with eXogenous inputs 5, 6, 8, 9, 10, 11, 12, 13, 16, 18, 19, 23, 27, 37, 38, 40, 44, 47, 48, 69, 81, 83, 84, 86, 89, 100

OLS Orthogonal Least Squares 2, 3

PEM Prediction Error Minimization 1, 2, 19

PRBS PseudoRandom Binary Signals 7, 8, 44

PSO Particle Swarm Optimization 28, 29, 31, 33, 34, 35, 40, 41

PSOGSA Hybrid Particle Swarm Optimization and Gravitational Search Algorithm 35, 36

RaMSS Randomized Model Structure Selection 37, 38

RIH Rate Independent Hysteresis 14, 16

RJMCMC Reversible Jump Markov Chain Monte Carlo 37

RMSE Root Mean Squared Error 26, 92

SISO Single Input Sigle Output 4, 19, 27

ULS Ultra Least Squares 39, 40

UOFR Ultra Orthogonal Forward-Regression 40

WGN White Gaussian Noise 7

CHAPTER 1

INTRODUCTION

The concept of a mathematical model is fundamental in many fields of science. In engineering, for example, models are essential to the study of systems as they allow to simulate the system, predict its response for a given input, and explore different design schemes. These aims, however, can only be accomplished if the model of the system is a proper representation of the system under study. The modeling of dynamic and steady-state behavior is, therefore, fundamental to this type of analysis and depends on system identification procedures.

System identification is a method of identifying the dynamic model of a system from measurements of the system inputs and outputs [5]. In particular, the interest in nonlinear system identification has deserved much attention by researchers from the 1950s onwards and many relevant results were developed [6–12]. In this context, one frequently employed model representation is the Nonlinear Auto-Regressive Moving Average Model with eXogenous inputs (NARMAX), which was introduced in 1981 aiming at representing a broad class of nonlinear systems [13–15]. NARMAX methods were successfully applied in many scenarios, which include industrial processes, control systems, structural systems, economic and financial systems, biology, medicine, social systems, and much more [16–21].

There are several NARMAX model representations such as polynomial, generalized additive, neural networks, and wavelet. Among these types of the extended model set, the power-form polynomial is the most commonly NARMAX representation [5]. Fitting polynomial NARMAX models is a simple task if the terms in the model are known *a priori*. Regarding the parameters of polynomial NARMAX models, we have a pseudo-linear regression problem since the output is a linear function of nonlinear monomials of the model. Thus, the Least Squares (LS) algorithm can be used to estimate the parameters of the model in a Prediction Error Minimization (PEM) scenario [22].

Selecting the model terms, however, is fundamental if the goal of the identification is to obtain models that can reproduce the dynamics of the original system. Problems related to overparameterization and numerical ill-conditioning are typical because of the limitations of

the identification algorithms in selecting the appropriate terms that should compose the final model [23, 24]. Aguirre e Billings [23] state that models with a higher number of parameters are more flexible and tend to fit the data better, which causes the inclusion of spurious regressors in the models. Furthermore, the variance of the residuals decreases typically as the complexity of a model is increased.

In that respect, one of the most traditionally algorithms for structure selection of polynomial NARMAX was developed by [25] based on the Orthogonal Least Squares (OLS) and the Error Reduction Ratio (ERR). This algorithm, called Forward-Regression Orthogonal Least Squares (FROLS), selects the terms of the model in a forward-regression manner by computing the contribution that each potential regressor makes to the system output. In this way, step by step, a term at a time, the model is built up with the most significant model terms being added first. Numerous variants of FROLS algorithm have been developed to improve the model selection performance such as [26–30]. The drawbacks of the FROLS have been extensively reviewed in the literature, *e.g.*, in [31–33]. Most of these are related to the PEM framework and the inadequacy of the ERR index in measuring the absolute importance of regressors. Furthermore, the relative importance of a regressor measured by ERR index appears to be highly variable concerning the one term at time Model Structure Selection (MSS) and may vary significantly during the procedure. In other words, a regressor considered crucial at the beginning may progressively lose importance when adding other terms to the model [24]. In this respect, the identification of severely nonlinear systems *e.g.* system with hysteresis, the FROLS algorithm also fails to select the correct regressors even though polynomial NARMAX can reproduce that behavior [34].

Along with FROLS algorithm, some information criteria are generally used in identification procedures. Among the several statistical index developed in literature, the Akaike Information Criterion (AIC) [35], Final Prediction Error (FPE) [36] and the Bayesian Information Criterion (BIC) [37] are the mostly used by NARMAX identification algorithms [5]. These information criteria attend to weigh the model accuracy against the number of parameters of the model, thereby allowing to evaluate the appropriate model size. Although these techniques work well for linear models, in a nonlinear context no simple relation between model size and accuracy can be established [33]. Nonlinear models of the same size may have significantly different regressors, and the performances can be entirely different. As a result, these criteria cannot be

used to indicate whether terms are correct to compose the model [38].

As a consequence of the limitations of OLS based algorithms, some recent research endeavors have significantly strayed from the classical FROLS scheme, by reformulating the MSS process in a probabilistic framework and using random sampling methods [1, 2, 33, 39, 40]. Nevertheless, these techniques based on meta-heuristics, probabilistic frameworks, and Markov Chain Monte Carlo (MCMC) presents some flaws. The meta-heuristics approaches turn on AIC, FPE, BIC and others information criteria to formulate the cost function of the optimization problem, generally resulting in over-parameterized models. Moreover, to avoid spurious regressors it is necessary to provide a tolerance value for information criteria and a limit size for the models [2]. Moreover, it is worth pointing out others significant drawbacks: (i) high computational cost; (ii) over-parameterized models for systems excited with low frequencies inputs; (iii) the correlation between selected regressors in each iteration is not analyzed, resulting in the test of a higher number of models. More than that, the choice of different meta-heuristics may result in distinct models, as can be seen in [2].

In light of the preceding, it is important to point out that the above-cited techniques are commonly used in a mono-objective identification scenario [41, 42]. In this context, the prediction error is the objective to be minimized, disregarding the solutions in which other objectives are relevant to obtain accurate models. Regarding Multi-objective Model Structure Selection (MoMSS), although there are multi-objective identification techniques using algorithms based on OLS, *e.g.,* in [43–46], to the best of our knowledge there are no studies addressing multi-objective identification of polynomial NARMAX models using metaheuristics.

Last but not least, due to the importance of classification techniques for decision-making tasks in engineering, business, health science, and many other fields, it is surprising how only a few researchers have addressed this problem using classical regression techniques. The authors in [47] presented a novel algorithm that combines logistic regression with the NARX methodology to deal with systems with a dichotomous response variable. The results in that work, although very interesting, are based on FROLS algorithm and, therefore, inherits most of the drawbacks concerning the traditional technique, opening new paths for research.

This work proposes a technique to the identification of nonlinear systems using meta-heuristics that fills the mentioned gaps in what concerns the structure selection of NARMAX models for regression and classification. The method uses an alternative to the information

criteria cited as the index indicating the accuracy of the model as a function of the size of the model. By using an approach that verifies the importance of each regressor while building the model, the technique is capable of producing parsimonious models, even considering the cases where the system is excited by a low-frequency input. All this is achieved using a computationally efficient algorithm. Also, the proposed methodology is exploited to identify systems in which more than one objective should be minimized, putting in perspective the achievement of a set of solutions that investigates the relationship between the different objectives. Furthermore, the algorithm can model systems with hysteresis, which expands the results obtained in [34] by presenting the sufficient conditions for hysteresis in polynomial NARMAX with multiple inputs and equilibria. Finally, the proposed algorithm is adapted to deal with classification problems to represent systems with binary outputs that depend on continuous time predictors.

1.1 Objectives and contributions

The general purpose of this work is to use meta-heuristics in a new mono-objective and multi-objective framework for system identification of dynamical systems and severely nonlinear systems. The main contributions of the present study are:

- Development of a new method based on meta-heuristic to build polynomial NARMAX models. This method proposes a new way to formulate the cost function of the optimization problem based on the simulation error minimization (free run prediction). Moreover, the algorithm takes account of the importance of individuals regressors when testing each candidate model. This procedure allows the meta-heuristic to converge to globally optimal solution and build parsimonious models;

- the method can be extended from Single Input Single Output (SISO) systems to Multiple Input Multiple Output (MIMO) ones without any changes in the algorithm;

- the algorithm is also extended to the multi-objective scenario without significant changes. In such cases, the algorithm finds useful models that minimize both the dynamical and the steady-state error. For that, only the cost function needs a rewrite.

- the algorithm can select suitable models for severely nonlinear systems, *e.g.*, systems with hysteresis when considering the sufficient conditions to polynomial NARMAX reproduce

this particular behavior.

- still regarding hysteretic systems, this work present new conditions to modeling systems with hysteresis considering models with multiple inputs without restrict the number of equilibrium points.
- a new approach to address classification problems using NARX methodology.

1.2 Publications

The development of this thesis allowed the publication of the following papers:

- Lacerda Junior, Wilson R. et al. (2019) Control of Hysteretic Systems Through an Analytical Inverse Compensation Based on a NARX Model. IEEE Access, v. 7, p. 98228-98237.
- Lacerda Junior, W. R., Martins, S. A. M., Nepomuceno, E. G. (2019). Non Linear System Identification Using the BPSOGSA algorithm (in Portuguese). 14 Simpósio Brasileiro de Automação Inteligente.

1.3 Thesis Organization

The remainder of this work is organized as follows:

Chapter 2 provides the basic framework and notation for mono-objective and multi-objective nonlinear system identification of Nonlinear Auto-Regressive with eXogenous inputs (NARX) models and briefly reviews the main approaches in the literature, with a particular interest in the meta-heuristic methods that serve as starting point for this work.

Chapter 3 presents the necessary tools to formulate the cost function of the mono-objective identification strategy. This chapter also introduces the new algorithm and reports the results obtained on several systems taken from the literature and physical systems. The performed analysis aims to assess the effectiveness of the new algorithm regarding the meta-heuristic methods and classical methods as the FROLS.

Chapter 4 extends the algorithm from mono-objective to multi-objective system identification. Two commonly used case studies for multi-objective NARX model are performed, and the results are compared with Multi-objective Error Reduction Ratio (MERR).

Chapter 5 extends the algorithm to identification of hysteretic nonlinearity. Also, this chapter investigates the necessary conditions to NARX models with multiple inputs and n equilibria be able to reproduce hysteretic behavior.

Chapter 6 adapts the technique to develop NARX models considering systems with binary responses that depend on continuous predictors. The technique is tested against several popular Machine Learning approaches, *e.g.*, Random Forest, Support Vector Machine and K-Nearest Neighbors, and new methods based on NARX models.

Chapter 7 recaps the primary considerations of this thesis and proposes possible future works.

CHAPTER 2

BACKGROUND

Model development and analysis are some of the most important topics in science. Different models can be obtained for one specific study and the choice of the model representation depends on the user knowledge, the system under study, and the goals of the modeling process. In this context, because of the simplicity and versatility of NARX models, several systems can be modeled, understood and controlled using this model family. This dissertation intends to contribute to the field with an efficient alternative concerning the NARX model building. This chapter introduces a set of definitions and preliminary concepts to a sufficient understanding of this study.

2.1 Input Signals

The excitation signal plays a vital role in the system identification procedure since it allows to gather information about the system behavior. Obviously, in certain circumstances, the choice of the input is limited due to the system environment characteristics, *e.g.*, in biological systems.

Perform practical experiments using an adequate excitation signal is, therefore, the best guarantee of building accurate models. System identification textbooks usually provide information on design tests signals [22, 48]. Among the many signals suggested in the literature, some of the most commonly used are PseudoRandom Binary Signals (PRBS) and filtered White Gaussian Noise (WGN). Regarding linear systems, these signals are usually good choices because it excites a wide range of system frequencies. The PRBS based on maximum length sequences (called $m-$sequences), for example, can be easily generated using simple shift register circuitry with proper feedback or via software, and this makes it possible to incorporate them in signal generator devices resulting in a wide range of system dynamic testing through these type of signals.

For nonlinear dynamic systems, however, the input should drive the system over the full amplitude range of the system as well as the bandwidth [49]. For this reason, if the system under study has a static nonlinear element, the PRBS signal may linearize the nonlinearity, and

this input should be no longer used as test signal [5]. In such cases, to make PRBS input suitable for nonlinear systems, it suffices to filter the signal with a simple first-order lag [5].

2.2 Choice of representation: the NARMAX model

Is straightforward to compute the relationship between models in the same class in a linear scenario. So, for example, if a system is identified using a state-space model, other model types can be derived using simple transformations. For nonlinear models, however, there is no unique model to represent all classes of nonlinear systems and transform a model into another is usually a hard task. There are all sorts of shapes and forms to represent a dynamic system in nonlinear system identification literature, so the choice of the representation should assure the achievement of the primary goal of the identification process. With this in mind, a traditional choice is the NARMAX model, since a substantial portion of nonlinear systems can be represented by it in discrete-time domain [5].

Polynomial NARX is a mathematical model based on difference equations and relates the current output as a function of past inputs and outputs, mathematically described as:

$$y_k = F^{\ell}[y_{k-1}, \ldots, y_{k-n_y}, x_{k-d}, x_{k-d-1}, \ldots, x_{k-d-n_x}] + e_k, \tag{2.1}$$

where $n_y \in \mathbb{N}^*$, $n_x \in \mathbb{N}$, are the maximum lags for the system output and input respectively; $x_k \in \mathbb{R}^{n_x}$ is the system input and $y_k \in \mathbb{R}^{n_y}$ is the system output at discrete time $k \in \mathbb{N}^n$; $e_k \in \mathbb{R}^{n_e}$ stands for uncertainties and possible noise at discrete time k. In this case, F^{ℓ} is some nonlinear function of the input and output regressors with nonlinearity degree $\ell \in \mathbb{N}$ and d is a time delay typically set to $d = 1$. Although there are several nonlinear functions representations to approximate the unknown mapping $F[\cdot]$, *e.g.*, neural networks, fuzzy logic-based models and radial basis function, the remainder of this thesis contemplates methods related to the power-form polynomial models.

The number of candidate regressors, n_r, of polynomial NARX can be defined as [25]:

$$n_r = M + 1, \tag{2.2}$$

where

$$M = \sum_{i=1}^{\ell} n_i$$
$$n_i = \frac{n_{i-1}(n_y + n_x + i - 1)}{i}, n_0 = 1. \tag{2.3}$$

Parsimony makes the Polynomial NARX models a widely known model family. This characteristic means that a wide range of behaviors can be represented concisely using only a few terms of the vast search space formed by candidate regressors and usually a small data set are required to estimate a model, which can be crucial in applications where it is challenging to acquire a massive amount of data.

There are many other advantages of polynomial NARX models [5]:

a) Every polynomial function is a smooth function in $\mathbb{R}$;

b) From the Weierstrass theorem [50], any given continuous real-valued function defined on a closed and bounded interval $[a,b]$ can be uniformly approximated using a power-form polynomial on that interval;

c) A vast number of nonlinear dynamics can be characterized using the polynomial NARX model;

d) Polynomial NARX is well established in system identification field with many algorithms developed for both structure selection and parameter estimation;

e) Polynomial NARX models and can be conducted both for prediction and inference.

The following example illustrates these mentioned advantages since it shows a simple NARX model used to predict the behavior of a electromechanical system.

Example 2.2.1. The following is a NARX model of degree 2, identified from experimental data of a DC motor/generator with no prior knowledge of the model form. A candidate regressor set was generated considering $n_y = n_x = 3$ and $\ell = 2$. Refer to [51] for more information about the identification process.

$$\begin{aligned} y_k =& 1.7813y_{k-1} - 0{,}7962y_{k-2} + 0{,}0339x_{k-1} - 0{,}1597x_{k-1}y_{k-1} + 0{,}0338x_{k-2} + \\ &+ 0{,}1297x_{k-1}y_{k-2} - 0{,}1396x_{k-2}y_{k-1} + 0{,}1086x_{k-2}y_{k-2} + 0{,}0085y_{k-2}^2 \end{aligned} \tag{2.4}$$

△

In Example 2.2.1, $M = 28$, but the structure selection algorithm has chosen 9 terms to build the model. The total number of possible models, in this case, is 2^{28}.

Moreover, the model (2.1) can effortlessly be extended to MIMO case [14, 27]:

$$\begin{aligned} y_{i}k =& F_{i}^{\ell}\Big[y_{1}k-1,\ldots,y_{1}k-n_{y_1}^{i},\ldots,y_{s}k-1,\ldots,y_{s}k-n_{y_s}^{i},x_{1}k-d,\\ & x_{1}k-d-1,\ldots,x_{1}k-d-n_{x_1}^{i},\ldots,x_{r}k-d,x_{r}k-d-1,\ldots,x_{r}k-d-n_{x_r}^{i}\Big]+\xi_{i}k, \end{aligned} \tag{2.5}$$

where y_{s} and x_{r} represents each output and input, respectively. Hence, for $i = 1,\ldots,s$, each linear in the parameter sub-model y_{i} can change regarding different maximum lags $n_{y_s}^{i}$ and $n_{x_r}^{i}$. More generally, considering

$$\boldsymbol{Y}_k = \begin{bmatrix} y_{1}k \\ y_{2}k \\ \vdots \\ y_{s}k \end{bmatrix}, \boldsymbol{X}_k = \begin{bmatrix} x_{1}k \\ x_{2}k \\ \vdots \\ x_{r}k \end{bmatrix}, \boldsymbol{\Xi}_k = \begin{bmatrix} \xi_{1}k \\ \xi_{2}k \\ \vdots \\ \xi_{r}k \end{bmatrix}, \tag{2.6}$$

the model (2.5) can be denoted as

$$\boldsymbol{Y}_k = F^{\ell}[\boldsymbol{Y}_{k-1},\ldots,\boldsymbol{Y}_{k-n_y},\boldsymbol{X}_{k-d},\boldsymbol{X}_{k-d-1},\ldots,\boldsymbol{X}_{k-d-n_x}] + \Xi_k, \tag{2.7}$$

where $\mathbf{X}_k = \{x_{1}k, x_{2}k, \ldots, x_{r}k\} \in \mathbb{R}^{n_{x_r}^{i}}$ and $\mathbf{Y}_k = \{y_{1}k, y_{2}k, \ldots, y_{s}k\} \in \mathbb{R}^{n_{y_s}^{i}}$. The number of possibles terms of MIMO NARX model given the ith polynomial degree, ℓ_i, is:

$$n_{m}r = \sum_{j=0}^{\ell_i} n_{ij}, \tag{2.8}$$

where

$$n_{ij} = \frac{n_{ij-1}\left[\sum\limits_{k=1}^{s} n_{y_k}^{i} + \sum\limits_{k=1}^{r} n_{x_k}^{i} + j - 1\right]}{j}, \qquad n_{i0} = 1, j = 1,\ldots,\ell_i. \tag{2.9}$$

If $s = 1$ in Equation (2.6), we have a Multiple Input Single Output (MISO) model that can be represented by a single polynomial function. Additionally, a MIMO model can be decomposed into MISO models, as presented in Figure 2.1:

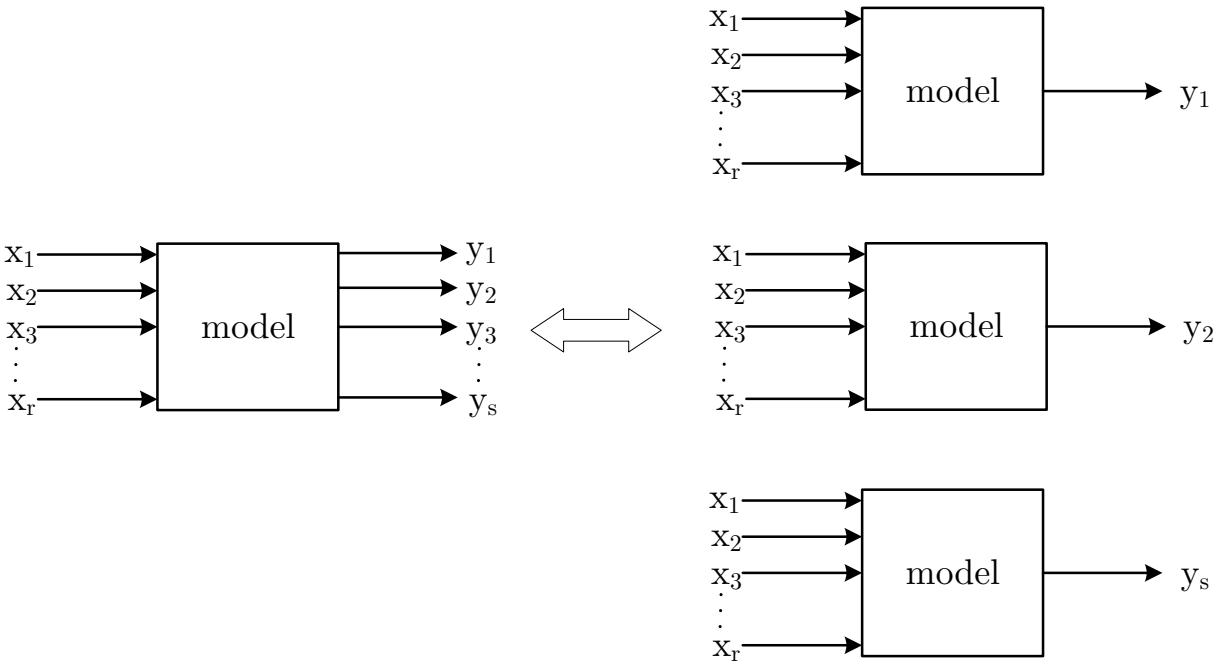


Figure 2.1: A MIMO model splitted in MISO models.

To describe severely nonlinear systems, however, is not always straightforward, and more in-depth studies need to be done. The following subsection presents a thorough study concerning the use of NARX models to represent systems with hysteretic behavior.

2.2.1 Equilibria in NARX models

The model (2.1) can be expanded as a summation of terms with nonlinearity degree in the range from 1 to ℓ where each $(p+m)$th term can contain a factor of order pth in y and mth in u, multiplied by a constant parameter $\Theta_{p,m}(\tau_1,\ldots,\tau_m)$, as follows:

$$y_k = \sum_{m=0}^{\ell}\sum_{p=0}^{\ell-m}\sum_{\tau_1,\tau_m}^{n_y,n_x}\Theta_{p,m}(\tau_1,\ldots,\tau_{p+m})\prod_{i=1}^{p}y_{k-\tau_i}\times\prod_{i=1}^{m}x_{k-\tau_i}+e_k. \tag{2.10}$$

In this matter, if the summation in Equation (2.10) refers to factors in y, the upper limit is n_y or, similarly, n_x.

In steady-state is reasonable to consider $\overline{y} = y_{k-\tau}, \forall\ \tau = 1,\ldots,n_y, \overline{x} = x_{k-\tau}, \forall\ \tau = 1,\ldots,n_x$ so the Equation (2.10) can be rewritten as

$$\overline{y_k} = \sum_{m=0}^{\ell}\sum_{p=0}^{\ell-m}\left(\sum_{\tau_1,\tau_m}^{n_y,n_x}\Theta_{p,m}(\tau_1,\ldots,\tau_{p+m})\right)\overline{y}^p\overline{x}^m, \tag{2.11}$$

giving room for the definitions of term cluster and cluster coefficients.

Definition 2.2.1 (Term cluster and cluster coefficients [23])**.** A set of terms of the form $y_{k-d}^{p}x_{k-j}^{m}$ to $m+p \leq l$, where d and j are time delays, is called $d-$cluster and can be written as $\Omega_{y^p x^m}$.

The constant $\sum_{\tau_1,\tau_m}^{n_y,n_x} \Theta_{p,m}(\tau_1,\ldots,\tau_{p+m})$ in Equation (2.11) is the cluster coefficient of a set of terms $\Omega_{y^p x^m}$, which contains terms of the form $y_{k-i}^{p}x_{k-j}^{m}$ to $m = 0,\ldots,l$ and $p = 0,\ldots,m$, where i and j are time delays. These coefficients are denoted by $\sum_{y^p x^m}$. △

Example 2.2.2. Consider the model (2.4) presented in Example 2.2.1. According to the Equation (2.10) we can rewrite the model (2.4) as:

$$\begin{aligned}
&\Theta_{1,0}(1) = 1.7813, && \Theta_{1,0}(2) = -0.7962, && \Theta_{0,1}(1) = 0.0339,\\
&\Theta_{1,1}(1,1) = -0.1597, && \Theta_{0,1}(2) = 0.0338, && \Theta_{1,1}(2,1) = 0.1297,\\
&\Theta_{1,1}(1,2) = -0.1396, && \Theta_{1,1}(2,2) = 0.1086, && \Theta_{2,0}(1) = 0.0085.
\end{aligned} \tag{2.12}$$

In Equation (2.12), $\Theta_{p,0}(\tau_{p+m})$ is the $d-$order coefficient of the $d-$cluster Ω_y and so on. The remainder coefficients are $\Theta_{p,m-p}(\cdot) = 0$.

Term Cluster	Cluster Coefficient
Ω_y	$\sum_y = \Theta_{1,0}(1) + \Theta_{1,0}(2) = 0.9851$
Ω_x	$\sum_x = \Theta_{0,1}(1) + \Theta_{0,1}(2) = 0.0677$
Ω_{yx}	$\sum_{yx} = \Theta_{1,1}(1,1) + \Theta_{1,1}(2,1) + \Theta_{1,1}(1,2) + \Theta_{1,1}(2,2) = -0.0610$
Ω_{y^2}	$\sum_{y^2} = \Theta_{2,0}(2,2) = 0.0085$

In this example, $\Omega_y = \Omega_{y_1} \cup \Omega_{y_2}$; $\Omega_x = \Omega_{x_1} \cup \Omega_{x_2}$; $\Omega_{yx} = \Omega_{y_1x_1} \cup \Omega_{y_2x_1} \cup \Omega_{y_1x_2} \cup \Omega_{y_2x_2}$ and, finally, $\Omega_{y^2} = \Omega_{y_2^2}$. △

It is worth mentioning that MSS algorithms can use the concept of term clusters to detect spurious terms in a model. See [48] for further details.

Definition 2.2.2 (Equilibria in Non-autonomous Systems)**.** The equilibrium points of a non-autonomous system are the points that satisfy $\overline{y} = y_k = y_{k+i}$, $i \in \mathcal{Z}$, given a constant input value $\overline{x} = u_k = u_{k+i}$, $i \in \mathcal{Z}$.

Using the Definition 2.2.1 the polynomial NARX can be rewritten in steady-state manner

as [52]:

$$\sum_{y^l} \overline{y}^l + \sum_{m=l-1}^{l} \left[\sum_{x^{m-(l-1)}y^{l-1}} \overline{x}^{m-(l-1)} \right] \overline{y}^{l-1} + \sum_{m=p}^{l} \left[\sum_{x^{m-p}y^p} \overline{x}^{m-p} \right] \overline{y}^p + \dots$$
$$+ \sum_{m=1}^{l} \left[\sum_{x^{m-1}y} \overline{x}^{m-1} \right] \overline{y} + \sum_{m=1}^{l} \sum_{x^m} \overline{x}^m + \sum_0 = 0. \tag{2.13}$$

The roots of Equation (2.13) results in the equilibria, $\overline{y}$, for a corresponding constant input, $\overline{x}$, considering non-autonomous systems. △

The local stability of the equilibrium points (in the Lyapunov sense) can be certified from the eigenvalues of the Jacobian matrix $D|_{\overline{y},\overline{x}}$ [48], evaluated on the respective break-even point

$$D|_{\overline{y},\overline{x}} = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \\ 0 & 0 & 0 & \dots & 1 \\ \frac{\partial F^l}{\partial y_{k-p}} & \frac{\partial F^l}{\partial y_{k-p+1}} & \frac{\partial F^l}{\partial y_{k-p+2}} & \dots & \frac{\partial F^l}{\partial y_{k-1}} \end{bmatrix}. \tag{2.14}$$

In this case, consider a circle of unit radius whose center is in the plane $z = e^{sT_s}$, where $z = \sigma_z + j\omega_z$, σ_z and ω_z the real and imaginary parts of the eigenvalue, respectively. If that circle embraces all of the eigenvalues of $D|_{\overline{y},\overline{x}}$, the equilibrium points are locally stable. If this condition is not met, the equilibrium points are unstable.

2.2.2 Modeling hysteresis with polynomial NARX model

Hysteresis nonlinearity is a severely nonlinear behavior commonly found in electromagnetic devices, sensors, semiconductors, intelligent materials, and many more, which have memory effects between quasi-static input and output [53, 54]. The identification of hysteretic systems using polynomial NARX models is typically an intriguing task because the classical MSS algorithms do not work properly [34, 55]. Martins e Aguirre [34] presented the sufficient conditions to describe hysteresis using polynomial models by providing the concept of bounding structure $\mathcal{H}$. Polynomial NARX models with a single equilibrium can be used in a full characterization of the hysteresis behavior adopting the bounding structure concept.

The following are some of the essential concepts and definitions for understanding how NARX model can be used to describe systems with hysteresis.

Definition 2.2.3 (Continuous-time loading-unloading quasi-static signal [34])**.** A periodic continuous time signal x_t with period $T = (t_f - t_i)$ and frequency $\omega = 2\pi f$ is qualified as a loading-unloading signal if x_t increases monotonically from x_{min} to x_{max}, considering $t_i \leq t \leq t_m$ (loading) and decreases monotonically from x_{max} to x_{min}, considering $t_m \leq t \leq t_f$ (unloading). If the loading-unloading signal changes with $\omega \to 0$, the signal is also called a quasi-static signal (Fig.2.2(1)). △

Definition 2.2.4 (Hysteresis loops in continuous time $\mathcal{H}_t(\omega)$ [34])**.** Let x_t be a continuous-time loading-unloading quasi-static signal applied to a continuous-time system and y_t is the system output. $\mathcal{H}_t(\omega)$ denotes a closed loop in the $x_t - y_t$ plane, which shape depend on ω. If the system presents hysteretic nonlinearity, $\mathcal{H}_t(\omega)$ is denoted as:

$$\mathcal{H}_t(\omega) = \begin{cases} \mathcal{H}_t(\omega)^+, \ for \ t_i \ \leq \ t \ \leq \ t_m, \\ \mathcal{H}_t(\omega)^-; \ for \ t_m \ \leq \ t \ \leq \ t_f, \end{cases} \tag{2.15}$$

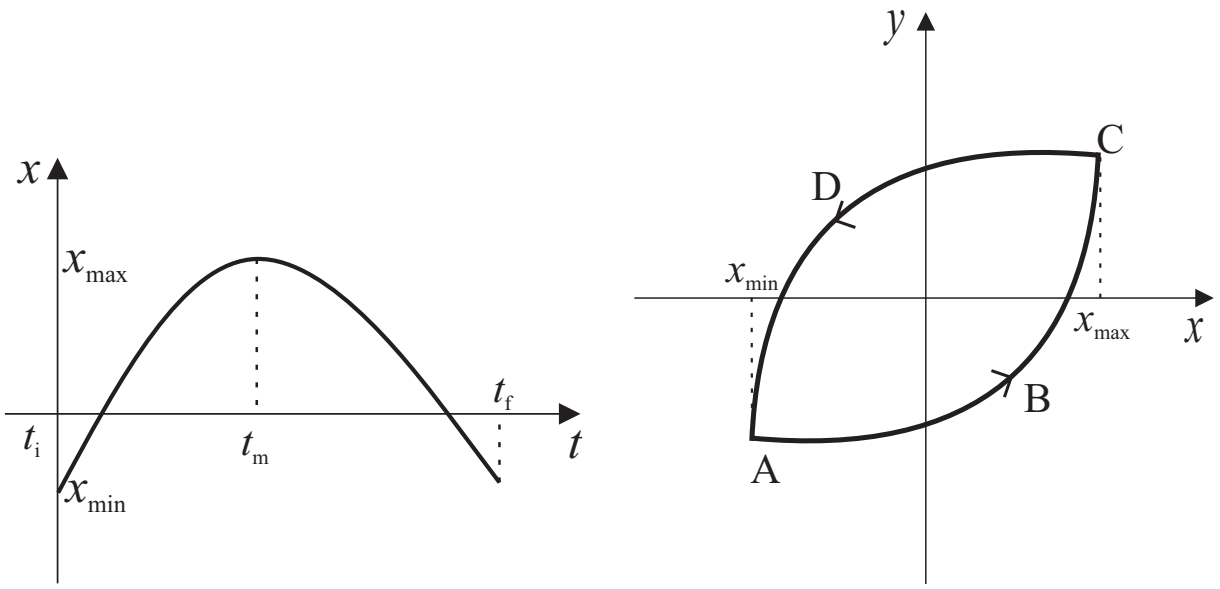
where $\mathcal{H}_t(\omega)^+ \neq \mathcal{H}_t(\omega)^-, \forall t \neq t_m$. $t_i \leq t \leq t_m$ and $t_m \leq t \leq t_f$ correspond to the regime when x_t is loading and unloading, respectively. $\mathcal{H}_t(\omega)^+$ corresponds to the part of the loop formed in the $x_t - y_t$ plane, while $t_i \leq t \leq t_m$ (when x_t is loading) whereas $\mathcal{H}_t(\omega)^-$ is the part of the loop formed in the $x_t - y_t$ plane for $t_m \leq t \leq t_f$ (when x_t is unloading), as shown in Figure 2.2(2). △

Definition 2.2.5 (Rate Independent Hysteresis (RIH) [53])**.** The hysteresis behavior is called to be rate independent if the path $ABCD$, which depends on pair $x(t), y(t)$ (Figure 2.2(2)), is invariant with respect to any increasing diffeomorphism $\varphi : [0,T] \to [0,T]$, i.e.:

$$F(u \ o \ \varphi, y^0) = F(u,y^0) \ o \ \varphi \ em \ [0,T]. \tag{2.16}$$

This means that at any instant t, $y(t)$ depends only on $u : [0,T] \to \mathbb{R}$ and on the order in which values have been attained before t. △

Remark 2.2.1. This work considers the concept of Rate Independent Hysteresis (RIH). This definition excludes, for example, any memory effect of the viscous type, such as those repre-



(1) Example of a loading-unloading signal.

(2) Example of Hysteresis curve.

Figure 2.2: Example of a hysteresis curve considering a loading-unloading input signal.

sented by temporal convolution. Common hysteretic phenomena as in ferromagnetic or elastic materials may not be purely rated independent since viscous-like effects attach to the hysteresis. However, the rate-independent component prevails [53]. △

Definition 2.2.6 (Bounding Structure $\mathcal{H}$ [34])**.** Let $\mathcal{H}_t(\omega)$ be the system hysteresis. $\mathcal{H} = \lim_{\omega \to 0} \mathcal{H}_t(\omega)$ is defined as the bounding structure that delimits $\mathcal{H}_t(\omega)$. △

Definition 2.2.7 (Multi-valued functions)**.** Let $\phi(\Delta x_k) : \mathbb{R} \to \mathbb{R}$. If $\Delta x_k = x_k - x_{k-1}$, $\phi(\Delta x_k)$ is a multi-valued function if:

$$\phi(\Delta x_k) = \begin{cases} \phi_1, & if\ \Delta x_k > \epsilon; \\ \phi_2, & if\ \Delta x_k < \epsilon; \\ \phi_3, & if\ \Delta x_k = \epsilon; \end{cases} \tag{2.17}$$

where $\epsilon \in \mathbb{R}$, $\phi_1 \neq \phi_2 \neq \phi_3$. For some inputs $\Delta x_k \neq \epsilon,\ \forall k \in \mathbb{N}$, and the last value in (2.17) is not used. △

Example 2.2.3. A frequently used multi-valued function is the sign$(\cdot) : \mathbb{R} \to \mathbb{R}$:

$$sign(x) = \begin{cases} 1, & if\ x > 0; \\ -1, & if\ x < 0; \\ 0, & if\ x = 0. \end{cases} \tag{2.18}$$

△

Lemma 2.2.1 (RIH in polynomial NARX [34])**.** Consider a polynomial NARX excited by a loading-unloading quasi-static signal. If the model has one real and stable equilibrium point whose location depends on input and loading/unloading regime, the polynomial will exhibit a RIH hysteresis loop $\mathcal{H}_t(\omega)$ in the $x - y$ plane.

Proof. Let us consider the polynomial NARX model with asymptotically stable equilibrium points described by [52]:

$$\begin{aligned} y_k = &\sum\nolimits_0 + \sum_{i=1}^{p} c_y^i y_{k-i} + \sum_{j=1}^{q} c_\phi^j \phi(\Delta x_{k-j}) + \sum_{m=1}^{r} c_x^m x_{k-m} + \\ &\sum_{i=1}^{p}\sum_{j=1}^{q} c_{y\phi}^{ij} y_{k-i}\phi(\Delta x_{k-j}) + \sum_{i=1}^{p}\sum_{m=1}^{r} c_{yx}^{im} y_{k-i}x_{k-m} + \\ &\sum_{j=1}^{q}\sum_{m=1}^{r} c_{\phi x}^{jm} \phi(\Delta x_{k-j})x_{k-m} + \sum_{i=1}^{p}\sum_{j=1}^{q}\sum_{m=1}^{r} c_{y\phi x}^{ijm} y_{k-i}\phi(\Delta X_{k-j})x_{k-m} + \\ &\sum_{m_1=1}^{r}\sum_{m_2=m_1}^{r} c_{x^2}^{m_1 m_2} x_{k-m_1}x_{k-m_2} + \ldots + \sum_{m_1=1}^{r} \ldots \sum_{m_l=m_{l-1}}^{r} c_{x^l}^{m_1,\ldots,m_2} x_{k-m_1}x_{k-m_l}, \end{aligned} \tag{2.19}$$

where $\sum_0$, c_y^i, c_ϕ^j, c_x^m, $c_{y\phi}^{ij}$, c_{yx}^{im}, $c_{\phi x}^{jm}$, $c_{y\phi x}^{ijm}$, $c_{x^2}^{m_1 m_2} \ldots c_{x^l}^{m_1,\ldots,ml}$ are constant parameters.

The equilibria of Equation (2.19), considering a loading-unloading regime, are [52]:

$$\overline{y}(\phi,\overline{x}) = \begin{cases} \frac{\sum_\phi + \sum_0 + (\sum_x \overline{x} + \sum_{\phi x}\overline{x} + \sum_{x^2}\overline{x}^2) + \ldots + \sum_{xl}\overline{x}^l}{1 - \sum_y - \sum_{y\phi} - (\sum_{yx}\overline{x} - \sum_{y\phi x}\overline{x})}, & for\ loading; \\ \frac{-\sum_\phi + \sum_0 + (\sum_x \overline{x} + \sum_{\phi x}\overline{x} + \sum_{x^2}\overline{x}^2) + \ldots + \sum_{xl}\overline{x}^l}{1 - \sum_y - \sum_{y\phi} - (\sum_{yx}\overline{x} - \sum_{y\phi x}\overline{x})}, & for\ unloading; \end{cases} \tag{2.20}$$

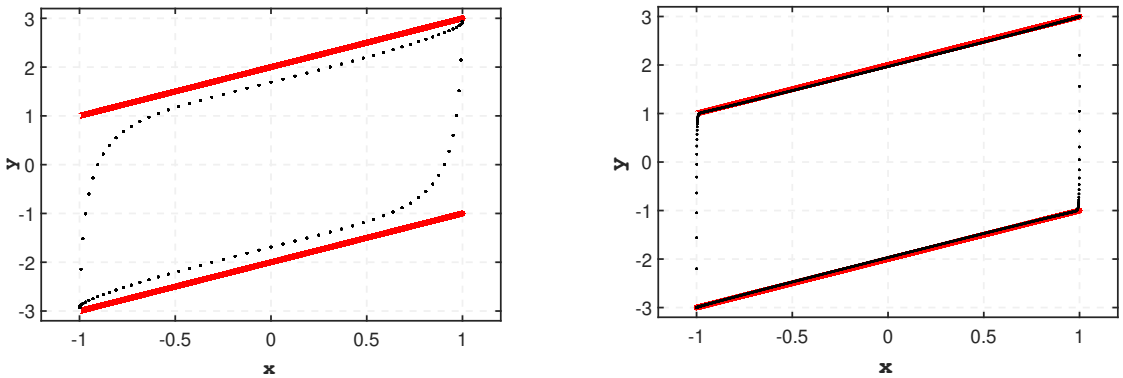
where $\sum_{(\cdot)}$ is the cluster coefficient $\Omega_{(\cdot)}$. In this thesis, the sign function is the multi-valued function.

The equilibrium of Equation (2.20) defines the $\mathcal{H}$, which in this case depends on the values ϕ and the static value of the input x. When the equilibrium points are asymptotically stable, the output will converge to the loop of $\mathcal{H}_k(\omega)$ in the input-output plane. Also, $\mathcal{H}_k(\omega)$ approaches more of the bounding structure as the frequency decreases [34]. □

Example 2.2.4. Let $y_k = 0.8y_{k-1} + 0.4\phi_{k-1} + 0.2x_{k-1}$, where $\phi_k = \text{sign}(\Delta(\text{x}_\text{k}))$ and $x_k = sin(\omega k)$ and ω is the frequency of the input signal x. For this model, $\sum_y = 0.8$, $\sum_\phi = 0.4$, $\sum_x = 0.2$ and $\sum_0 = 0$. The equilibria of this model is given by:

$$\overline{y}(\overline{\phi},\overline{x}) = \begin{cases} \frac{0.6+0.2\overline{x}}{1-0.8} = 3 + \overline{x}\,, & for\ loading; \\ \frac{-0.6+0.2\overline{x}}{1-0.8} = -3 + \overline{x}\,, & for\ unloading; \end{cases} \tag{2.21}$$

where $\overline{x}$ is a loading-unloading quasi-static input signal. Since the equilibrium points are asymptotically stable, the output converges to $\mathcal{H}_k(w)$ in the $x-y$ plane. Note that for a constant input value $x = 1 = \overline{x}$, the equilibrium lies in $\overline{y} = 3$ for loading regime and $\overline{y} = -1$ for unloading regime. Analogously, for $\overline{x} = -1$, the equilibrium lies in $\overline{y} = 1$ for loading regime and $\overline{y} = -3$ for unloading regime, as shown in Figure 2.3.



(1) $\omega = 1$. (2) $\omega = 0.1$.

Figure 2.3: The black dots are on $\mathcal{H}_k(\omega)$ for model $y_k = 0.8y_{k-1} + 0.4\phi_{k-1} + 0.2x_{k-1}$. The bounding structure $\mathcal{H}$,in red, confines $\mathcal{H}_k(\omega)$.

△

2.3 Structure Selection Techniques

Selecting the model structure is crucial to develop models that can correctly reproduce the system behavior. If some prior information about the system is known, *e.g.*, the dynamic order and degree of nonlinearity, determining the terms and then estimate the parameters is trivial. In real life scenarios, however, in most of the times, there is no information about what terms should be included in the model, and the identification framework selects the correct regressors. Without the fundamental concerns in the MSS procedure, the scientific law that describes the system may not be revealed and resulting in misleading interpretations about the system. To illustrate this scenario, consider the following example.

Example 2.3.1. Let $\mathcal{D}$ denote a full dataset

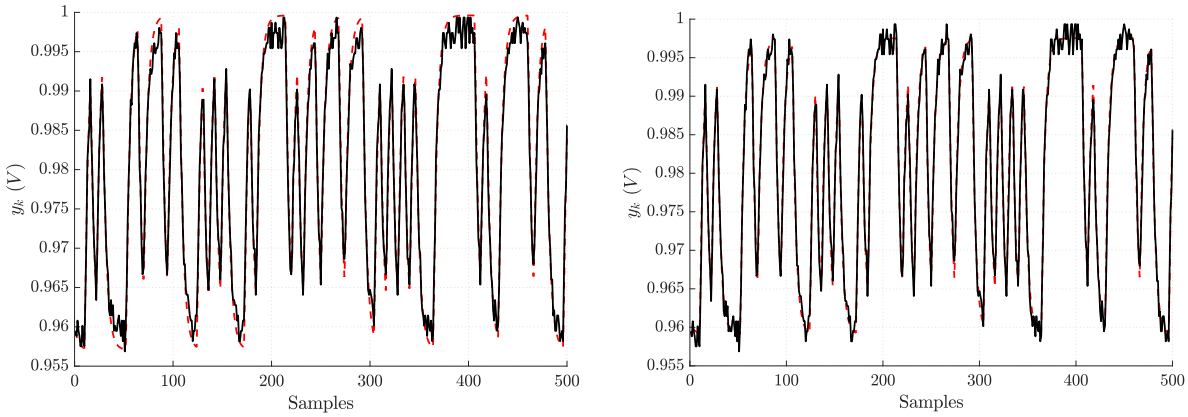
$$\mathcal{D} = \{(x_k, y_k), k = 1, 2, \ldots, n\}, \tag{2.22}$$

where $x_k \in \mathbb{R}^{n_x}$ and $y_k \in \mathbb{R}^{n_y}$ are the input and output of some, at first, unknown system and n is the number of data points in the sample. In what follows, it is presented two polynomial NARX models built to describe that system:

$$y_{a}k = 0.7077 y_{a}k-1 + 0.1642 u_{k-1} + 0.1280 u_{k-2} \tag{2.23}$$

$$\begin{aligned} y_{b}k =& 0.7103 y_{b}k-1 + 0.1458 u_{k-1} + 0.1631 u_{k-2} - \\ & - 0.1467 y^{3}_{b}k-1 + 0.0710 y^{3}_{b}k-2 + 0.0554 y^{2}_{b}k-3 u_{k-3}. \end{aligned} \tag{2.24}$$

Figure 2.4 shows the simulation of each model. As can be observed, the nonlinear model (2.24) seems to fit the data better than the linear model (2.23). The original system, however, is composed of a resistor, inductor and a capacitor in series with a voltage source. Hence, an adequate model for an RLC series circuit has a linear relationship between the current and the applied voltage and behaves as a second-order system, which is not the case. That being said, even if model (2.24) fits the data better, it is clearly over-parameterized. These models thus need to be interpreted with caution because the dynamic properties of the original system may be lost and could induce misinterpretation. △



(1) Free run simulation of the model defined by $y_{a}k$.

(2) Free run simulation of the model defined by $y_{b}k$.

Figure 2.4: Model validation. (-) indicate the experimental data and (- -) indicates the data of the model.

Identifying the correct structure, therefore, is fundamental to allow the user to be able to analyze the system dynamics consistently. In this respect, many algorithms have been developed to select the terms that will compose a polynomial NARX model. One of the main aim of the MSS algorithms is to reveal the characteristics of the systems by producing the simplest model possible. Of course, there are cases that complex models are required, but a quotation by Einstein summarizes this well: "a model should be as simple as possible - but not simpler".

The regressors selection, however, is not a simple task. If ℓ, n_x and n_y increases, the number of candidate models becomes too large for brute force approach. Considering the MIMO case, this problem is far worse than the SISO one if many inputs and outputs are required. The total number of all different models is given by

$$n_m = \begin{cases} 2^{n_r} & \text{for SISO models,} \\ 2^{n_{mr}} & \text{for MIMO models,} \end{cases} \tag{2.25}$$

where n_r and n_{mr} are the values computed using Equations (2.2) to (2.3) and Equations (2.8) to (2.9).

A classical solution to regressors selection problem is the FROLS algorithm associated with ERR test. This technique relies on the PEM framework and, one at a time, select the most relevant regressor by using a step-wise regression. The FROLS method adapts the set of regressors in the search space into a set of orthogonal vectors, which ERR evaluates the individual contribution to the desired output variance.

2.3.1 The Forward Regression Orthogonal Least Squares Algorithm

The following description of the FROLS algorithm is taken mainly from [5, 15]. In this respect, consider model (2.1) described in a generic form as

$$y_k = \psi_{k-1}^{\top}\hat{\Theta} + \xi_k, \tag{2.26}$$

where $\psi_{k-1}^{\top} \in \mathbb{R}^{n_r \times n}$ is a vector of some combinations of the regressors and $\hat{\Theta} \in \mathbb{R}^{n_\Theta}$ the vector of estimated parameters. In a more compact form, model (2.26) can be represented in a matrix form as:

$$\boldsymbol{y} = \boldsymbol{\Psi}\hat{\boldsymbol{\Theta}} + \boldsymbol{\Xi}, \tag{2.27}$$

where

$$\boldsymbol{Y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, \boldsymbol{\Psi} = \begin{bmatrix} \boldsymbol{\psi_1} \\ \boldsymbol{\psi_2} \\ \vdots \\ \boldsymbol{\psi_{n_\Theta}} \end{bmatrix}^{\top} = \begin{bmatrix} \psi_1 1 & \psi_2 1 & \cdots & \psi_{n_\Theta} 1 \\ \psi_1 2 & \psi_2 2 & \cdots & \psi_{n_\Theta} 2 \\ \vdots & \vdots & & \vdots \\ \psi_1 n & \psi_2 n & \cdots & \psi_{n_\Theta} n \end{bmatrix}, \hat{\boldsymbol{\Theta}} = \begin{bmatrix} \hat{\Theta}_1 \\ \hat{\Theta}_2 \\ \vdots \\ \hat{\Theta}_{n_\Theta} \end{bmatrix}, \boldsymbol{\Xi} = \begin{bmatrix} \xi_1 \\ \xi_2 \\ \vdots \\ \xi_n \end{bmatrix}. \tag{2.28}$$

The parameters in Equation (2.27) could be estimated as a result of a LS-based algorithm, but this would require optimizing all parameters at the same time on account of the fact of interaction between regressors due to non-orthogonality characteristic. Consequently, the computational demand becomes impractical for a high number of regressors. In this respect, the FROLS transforms the non-orthogonal model (2.26) into an orthogonal one.

The regressor matrix $\boldsymbol{\Psi}$ can be orthogonally decomposed as

$$\boldsymbol{\Psi} = \boldsymbol{Q}\boldsymbol{A}, \tag{2.29}$$

where $\boldsymbol{A} \in \mathbb{R}^{n_\Theta \times n_\Theta}$ is an unit upper triangular matrix according to

$$\boldsymbol{A} = \begin{bmatrix} 1 & a_{12} & a_{13} & \dots & a_{1n_\Theta} \\ 0 & 1 & a_{23} & \dots & a_{2n_\Theta} \\ 0 & 0 & 1 & \dots & \vdots \\ \vdots & \vdots & \vdots & \ddots & a_{n_\Theta - 1 n_\Theta} \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \tag{2.30}$$

and $\boldsymbol{Q} \in \mathbb{R}^{n \times n_\Theta}$ is a matrix with orthogonal columns q_i, described as

$$\boldsymbol{Q} = \begin{bmatrix} q_1 & q_2 & q_3 & \dots & q_{n_\Theta} \end{bmatrix}, \tag{2.31}$$

such that $\boldsymbol{Q}^\top \boldsymbol{Q} = \boldsymbol{\Lambda}$ and $\boldsymbol{\Lambda}$ is diagonal with entry d_i and can be expressed as:

$$d_i = \boldsymbol{q}_i^\top \boldsymbol{q}_i = \sum_{n}^{k=1} q_{_i k} q_{_i k}, \qquad 1 \leq i \leq n_\Theta. \tag{2.32}$$

The space spanned by the orthogonal basis $\boldsymbol{Q}$ (Equation (2.31)) is the same as that spanned by the basis set $\boldsymbol{\Psi}$ (Equation (2.28)). Consequently, the Equation (2.27) can be rewritten the auxiliary model

$$\boldsymbol{Y} = \underbrace{(\boldsymbol{\Psi}\boldsymbol{A}^{-1})}_{\boldsymbol{Q}} \underbrace{(\boldsymbol{A}\boldsymbol{\Theta})}_{\boldsymbol{g}} + \boldsymbol{\Xi} = \boldsymbol{Q}\boldsymbol{g} + \boldsymbol{\Xi}, \tag{2.33}$$

where $\boldsymbol{g} \in \mathbb{R}^{n_\Theta}$ is an auxiliary parameter vector. The parameters of the model (2.33) are given by

$$\boldsymbol{g} = \left(\boldsymbol{Q}^\top \boldsymbol{Q}\right)^{-1} \boldsymbol{Q}^\top \boldsymbol{Y} = \boldsymbol{\Lambda}^{-1} \boldsymbol{Q}^\top \boldsymbol{Y} \tag{2.34}$$

or

$$g_{_i} = \frac{\boldsymbol{q}_{_i}^\top \boldsymbol{Y}}{\boldsymbol{q}_{_i}^\top \boldsymbol{q}_{_i}}. \tag{2.35}$$

Since the parameter $\boldsymbol{\Theta}$ and $\boldsymbol{g}$ satisfies the triangular system $\boldsymbol{A}\boldsymbol{\Theta} = \boldsymbol{g}$, any orthogonalization method like Householder, Gram-Schmidt, modified Gram-Schmidt or Givens transformations can be used to solve the equation and estimate the original parameters. A more detailed discussion about those orthogonalization procedures can be found in [56]. Assuming that $\mathbf{E}[\boldsymbol{\Psi}^\top \Xi] = 0$, the output variance can be derived by multiplying Equation (2.33) with itself and

dividing by n, resulting in

$$\frac{1}{n}\boldsymbol{Y}^\top\boldsymbol{Y} = \underbrace{\frac{1}{n}\sum_{n_\Theta}^{i=1} g_i^2 \boldsymbol{q}_i^\top \boldsymbol{q}_i}_{\substack{\text{output explained} \\ \text{by the regressors}}} + \underbrace{\frac{1}{n}\Xi^\top\Xi}_{\substack{\text{unexplained} \\ \text{variance}}} . \tag{2.36}$$

Thus, the ERR due to the inclusion of the regressor $\boldsymbol{q}_i$ is expressed as:

$$[\text{ERR}]_i = \frac{g_i^2 \boldsymbol{q}_i^\top \boldsymbol{q}_i}{\boldsymbol{Y}^\top\boldsymbol{Y}}, \qquad \text{for } i = 1,2,\ldots,n_\Theta. \tag{2.37}$$

There are many ways to terminate the algorithm. An approach often used is to stop the algorithm if the model output variance drops below some predetermined limit of ε:

$$1 - \sum_{i=1}^{n_\Theta} \text{ERR}_i \leq \varepsilon, \tag{2.38}$$

or using some information criteria, *e.g.*, AIC. For LS based regression analysis, the AIC indicates the number of regressors by minimizing the objective function [35]:

$$J_{\text{AIC}} = \underbrace{n \log\left(\text{Var}[\xi_k]\right)}_{\text{first component}} + \underbrace{2n_\Theta}_{\substack{\text{second} \\ \text{component}}} . \tag{2.39}$$

From Equation (2.39), one can notice a trade-off between how well the model fits the data (first component) and the model complexity (second component), here related to the number of parameters included in the model. As far as new terms are added in the model, the AIC value decreases to a minimum value, which represents an optimal model choice regarding the complexity and the prediction error performance of the model. When the number of parameters becomes too large, the second component becomes more critical and entails an increase in the cost function. Equation (2.39) and many others variants have been extensively used for linear and nonlinear system identification [44, 57–59, and references therein].

In spite of the excellent results in linear model selection problems, information criteria like AIC may fail to select an adequate number of parameters when the system under study has a severely nonlinear behavior and also results in models with poor performance if the search space lacks some terms required to build the *true* model.

2.4 Parameter Estimation using LS

Because the polynomial NARX model is linear in the parameters, the linear LS algorithm developed by Gauss in 1795 can be used to estimate the model coefficients. Traditionally, the parameter vector minimizes the $l2$-norm, also known as the residual sum of squares, described as

$$J_{\hat{\Theta}} = \Xi^\top \Xi = (\boldsymbol{y} - \boldsymbol{\Psi}\hat{\boldsymbol{\Theta}})^\top (\boldsymbol{y} - \boldsymbol{\Psi}\hat{\boldsymbol{\Theta}}) = \|\boldsymbol{y} - \boldsymbol{\Psi}\hat{\boldsymbol{\Theta}}\|^2. \tag{2.40}$$

In Equation (2.40), $\boldsymbol{\Psi}\hat{\boldsymbol{\Theta}}$ is the one-step ahead prediction of y_k, expressed as

$$\hat{y}_k = g(y_{k-1}, u_{k-1} |\, \Theta), \tag{2.41}$$

where g is some unknown polynomial function. If the gradient of J_Θ for Θ is equal to zero, then lead to the well known normal equation and the LS estimate is denoted as

$$\hat{\boldsymbol{\Theta}} = (\boldsymbol{\Psi}^\top \boldsymbol{\Psi})^{-1} \boldsymbol{\Psi}^\top \boldsymbol{y}, \tag{2.42}$$

where $(\boldsymbol{\Psi}^\top \boldsymbol{\Psi})^{-1} \boldsymbol{\Psi}^\top$ is called the pseudo-inverse of the matrix $\boldsymbol{\Psi}$, denoted $\boldsymbol{\Psi}^+ \in \mathbb{R}^{n \times n_r}$.

In order to have a bias-free estimator, the following are the underlying assumptions needed for the least-squares method [48]:

A1 There is no correlation between the error vector, Ξ, and the matrix of regressors, $\boldsymbol{\Psi}$. Mathematically:

$$\mathrm{E}\{[(\boldsymbol{\Psi}^\top \boldsymbol{\Psi})^{-1} \boldsymbol{\Psi}^\top] \Xi\} = \mathrm{E}[(\boldsymbol{\Psi}^\top \boldsymbol{\Psi})^{-1} \boldsymbol{\Psi}^\top] \mathrm{E}[\Xi]; \tag{2.43}$$

A2 The error vector Ξ is a zero mean white noise sequence:

$$\mathrm{E}[\Xi] = 0; \tag{2.44}$$

A3 The covariance matrix of the error vector is

$$\mathrm{Cov}[\hat{\Theta}] = \mathrm{E}[(\Theta - \hat{\Theta})(\Theta - \hat{\Theta})^\top] = \sigma^2 (\boldsymbol{\Psi}^\top \boldsymbol{\Psi}); \tag{2.45}$$

A4 The matrix of regressors, $\boldsymbol{\Psi}$, is full rank.

The assumptions above are needed to guarantee that the LS algorithm produce an unbiased final model.

2.4.1 Affine information and multiobjective approach

Definition 2.4.1 (Affine Information [60])**.** Consider the parameter vector $\Theta \in \mathbb{R}^{n_\Theta}$, a vector $\mathrm{v} \in \mathbb{R}^p$ and a matrix $\mathrm{G} \in \mathbb{R}^{n_\Theta \times p}$ where v and G are assumed to be accessible. Suppose $\mathrm{G}\Theta$ be an estimate of v. Hence $\mathrm{v} = \mathrm{G}\Theta + \xi$. Then, $[\mathrm{v}, \mathrm{G}]$ is a pair of affine information of the system. △

Definition 2.4.2 (Multi-objective optimization problem (MOP) [61])**.** Given m objective functions

$$\mathbf{J}(\hat{\Theta}) = [J_1(\hat{\Theta}), J_2(\hat{\Theta}), \cdots, J_m(\hat{\Theta})]^\top, \tag{2.46}$$

where $\mathbf{J}(\cdot) : \mathbb{R}^n \mapsto \mathbb{R}^m$, a multi-objective optimization problem can be stated as

$$\begin{aligned} &\underset{\Theta}{\text{minimize}} \quad \mathrm{J}(\Theta) \\ &\text{subject to} \quad \Theta \in \mathrm{S} = [\Theta | \Theta \in \mathrm{A}^n, g_i(\Theta) \leq a_i, h_j(\Theta) = b_j] \quad i = 1, \ldots, m, j = 1, \ldots, n \end{aligned} \tag{2.47}$$

where Θ is an n-dimensional vector of the decision variables, S is the set of feasible solutions bounded by m inequality constraints (g_i) and n equality constraints (h_j), and a_i and b_j are constants. For continuous variables $A = \mathbb{R}$ while A contains the set of permissible values for discrete variables. △

Remark 2.4.1. Usually problems with $1 < m < 4$ are called Multi-objective Optimization Problem (MOP). Otherwise, it is called many-objective optimization problems, an emerging class of multi-objective problems for solving complex modern real-world tasks. More details can be found in [62, 63]. △

Definition 2.4.3 (Pareto dominance [64])**.** Consider $[\boldsymbol{y}^{(1)}, \boldsymbol{y}^{(2)}] \in \mathbb{R}^m$ two vectors in the objective space. If and only if $\forall i \in \{1, \ldots, m\} : y_i^{(1)} \leq y_i^{(2)}$ and $\exists j \in \{1, \ldots, m\} : y_j^{(1)} < y_j^{(2)}$ one can said $\boldsymbol{y}^{(1)} \prec \boldsymbol{y}^{(2)}$. △

Definition 2.4.4 (Pareto optimal)**.** If there is no vector $\boldsymbol{y}^{(1)} \in \mathbb{R}^m$ such that $\boldsymbol{y}^{(2)} \prec \boldsymbol{y}^{(*)}$ one can said that the vector $\boldsymbol{y}^{(*)}$ is Pareto optimal. △

The concept of Pareto optimality is generally used to describe the trade-off among the minimization of different objectives. Following the Definition 2.4.4, the Pareto optimal is any parameter vector representing an efficient solution where no objective function can be improved without making at least one objective function worse off. A hypothetical Pareto set is shown in Figure 2.5.

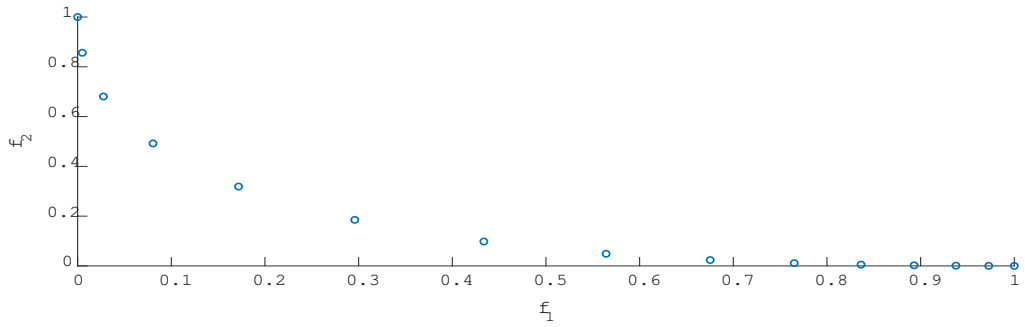


Figure 2.5: A hypothetical Pareto front of a bi-objective problem.

In this case, the model structure is assumed to be known and therefore, there is a one-to-one correspondence between each parameter vector on the Pareto optimal solution and a model [60]. One can build a Pareto set by applying the Weighted Sum Method, where a set of objectives are scalarized into a single objective by adding each objective multiplied by a user supplied weight. Consider

$$\mathrm{W} = \left\{ w | w \in \mathbb{R}^m, w_j \geq 0 \quad \text{and} \quad \sum_{j=1}^{m} w_j = 1 \right\} \tag{2.48}$$

as non-negative weights. Then, the convex optimization problem can be stated as

$$\Theta^* = \underset{\Theta}{\text{minimize}} \quad \langle w, \mathrm{J}(\Theta) \rangle, \tag{2.49}$$

where w is a weighted sum to the different objective functions. Therefore the Pareto-set is associated with the set of realizations of $w \in \mathrm{W}$. An efficient single-step computational strategy was presented by Nepomuceno et al. [60] for solving Equation (2.49) by means of a LS formulation, which is presented in the following theorem:

Theorem 2.4.1. Consider the m affine information pairs $[\mathbf{v_i} \in \mathbb{R}^{p_i}, \mathrm{G}_i \mathbb{R}^{p_i \times n}]$ with $i = 1, \ldots, m$. Assume there is exist a full column rank G_i and let M be a model of the form Equation (2.27).

Then the m affine information pairs can be considered in the parameter estimation by solving

$$\Theta^* = \underset{\Theta}{\text{minimize}} \quad \sum_{i=1}^{m} w_i(\mathrm{v}_i - \mathrm{G}_i\Theta)^\top(\mathrm{v}_i - \mathrm{G}_i\Theta), \tag{2.50}$$

with $\boldsymbol{w} = [w_i, \ldots, w_m]^\top \in \mathrm{W}$. The solution of Equation (2.50) is given by

$$\Theta^* = \left[\sum_{i=1}^{m} w_i \mathrm{G}_i^\top \mathrm{G}_i\right]^{-1} \left[\sum_{i=1}^{m} w_i \mathrm{G}_i^\top \mathrm{v}_i\right]. \tag{2.51}$$

If there exists only one information, the Theorem 2.4.1 reduces to the mono-objective LS solution. The reader is referred to [65] for more details about the proof of Equation (2.51).

2.4.2 Validation

Model validation is one of the most crucial parts of system identification. Ideally, the identified model should be able to capture all information from the data set, so the residual becomes unpredictable from all combinations of past inputs and outputs [66]. Many statistical tests can be applied in validations tests to check if the model is an adequate approximation of the system based on one-step-ahead prediction or free run simulation, also known as the model predicted output. One often used index is the normalized Root Mean Squared Error (RMSE), which can be expressed by

$$\text{RRSE} = \frac{\sqrt{\sum_{k=1}^{n}(y_k - \hat{y}_k)^2}}{\sqrt{\sum_{k=1}^{n}(y_k - \bar{y})^2}}, \tag{2.52}$$

where $\hat{y}_k \in \mathbb{R}$ the model predicted output and $\bar{y} \in \mathbb{R}$ the mean of the measured output y_k. The RMSE gives some indication regarding the quality of the model, but concluding about the best model by evaluating only this quantity may lead to an incorrect interpretation, as shown in the following example.

Example 2.4.1. Consider the models $y_{a}{}_k = 0.7077y_{a}{}_{k-1} + 0.1642u_{k-1} + 0.1280u_{k-2}$ and $y_{b}{}_k = 0.7103y_{b}{}_{k-1} + 0.1458u_{k-1} + 0.1631u_{k-2} - 1467y^3_{b}{}_{k-1} + 0.0710y^3_{b}{}_{k-2} + 0.0554y^2_{b}{}_{k-3}u_{k-3}$ taken from Example 2.3.1. The former results in a RMSE $= 0.1202$ while the latter gives RMSE $= 0.0857$. Although the model $y_{b}{}_k$ fits the data better, it is only a biased representation to one piece of data and not a good description of the entire system. △

Example 2.4.1 shows that validations test might be performed carefully. Another traditional practice is split the data set in two parts. In this respect, one can test the models obtained from the estimation part of the data using an specific data for validation. However, the one-step-ahead performance of NARX models generally results in misleading interpretations because even strongly biased models may fit the data well. Therefore, a free run simulation approach usually allows a better interpretation if the model is adequate or not [5].

Statistical tests for SISO models based on the correlation functions were proposed in [67, 68]. The tests are:

$$
\begin{aligned}
\phi_{_{\xi\xi}\tau} &= \mathbf{E}\{\,\xi_k\xi_{k-\tau}\,\} = \delta_\tau,\\
\phi_{_{\xi x}\tau} &= \mathbf{E}\{\,\xi_k x_{k-\tau}\,\} = 0\;\forall\tau,\\
\phi_{_{\xi\xi x}\tau} &= \mathbf{E}\{\,\xi_k\xi_{k-\tau}x_{k-\tau}\,\} = 0\;\forall\tau,\\
\phi_{_{x^2\xi}\tau} &= \mathbf{E}\{\,(u_k^2 - \mathbf{E}\{\,x_k^2\,\})\xi_{k-\tau}\,\} = 0\;\forall\tau,\\
\phi_{_{x^2\xi^2}\tau} &= \mathbf{E}\{\,(u_k^2 - \mathbf{E}\{\,x_k^2\,\})\xi_{k-\tau}^2\,\} = 0\;\forall\tau,\\
\phi_{_{(y\xi)x^2}\tau} &= \mathbf{E}\{\,(y_k\xi_k - \mathbf{E}\{\,y_k\xi_k\,\})(x_{k-\tau}^2 - \mathbf{E}\{\,x_k^2\,\})\,\} = 0\;\forall\tau,
\end{aligned}
\tag{2.53}
$$

where δ is the Dirac delta function and the cross-correlation function ϕ is denoted by [66]:

$$
\phi_{_{ab}\tau} = \frac{\frac{1}{n}\sum_{k=1}^{n-\tau}(a_k-\hat{a})(b_{k+\tau}-\hat{b})}{\sqrt{\frac{1}{n}\sum_{k=1}^{n}(a_k-\hat{a})^2}\sqrt{\frac{1}{n}\sum_{k=1}^{n}(b_k-\hat{b})^2}} = \frac{\sum_{k=1}^{n-\tau}(a_k-\hat{a})(b_{k+\tau}-\hat{b})}{\sqrt{\sum_{k=1}^{n}(a_k-\hat{a})^2}\sqrt{\sum_{k=1}^{n}(b_k-\hat{b})^2}},
\tag{2.54}
$$

where a and b are two signal sequences. If the tests are true, then the model residues can be considered as white noise. Moreover, it is worth mentioning that the tests in Equation (2.53) can be extended to the MIMO case in a straightforward manner.

2.5 Meta-heuristics

In general, nature-inspired optimization algorithms have been increasingly widespread over the last two decade due to the flexibility, simplicity, versatility, and local optima avoidance of the algorithms in real life applications. Several studies show that they are very efficient in

solving nonlinear design problems in engineering, industrial designs, and business activities.

These algorithms, also known as meta-heuristics, are stochastic and uses randomization and local search to solve the problem. Generally, one can classify meta-heuristics algorithms into two classes: trajectory based and population-based. Trajectory-based meta-heuristics, *e.g.*, Simulated Annealing [69], start the search process with a single agent or solution that keep improving through the search space by tracing a trajectory of possible solutions. Population-based meta-heuristics, however, use multiple agents or particles through the search space to reach the optimal solution. The Genetic Algorithm (GA), Particle Swarm Optimization (PSO) and Ant-Colony Optimization (ACO) are some well known population-based algorithms.

Compared to trajectory-based meta-heuristics, the population based ones have some advantages [70]:

a) The moves along the search space are guided by shared information about previous locations allowing jumps to more promising solutions;

b) There is an exchange of information among the candidate solutions in order to avoid premature convergences to locally optimal solutions;

c) Population-based meta-heuristics generally explore the search space better than the single solution-based algorithms.

d) Population-based algorithms often converges faster than the trajectories ones.

Two essential characteristics of meta-heuristics algorithms are exploitation and exploration [71]. Exploitation is related to the local information in the search process regarding the best near solution. On the other hand, exploration is related to explore a vast area of the search space to find an even better solution and not be stuck in local optima. In this sense, the right balance between exploitation and exploration is required for a performance that not lead to local solutions and not demand too much computational efforts to converge, resulting in a lengthy procedure. However, finding an appropriate trade-off between these two characteristics is not trivial and is still an open area [72].

Swarm Intelligence (SI) and Physics-based are two of the main classes of meta-heuristics [70]. The former typically mimics the social behavior of swarms and herds of creatures in nature, while the latter is based on environmental rules. The fundamental mechanisms

of these algorithms are very similar, since they are based on random populations, but the movement rules of each one are particularly unique, resulting in algorithms with very different performances. Some popular algorithms in those two classes are in addition to PSO, GA and ACO, are:

- Artificial Bee Colony (ABC) [73];
- Bat algorithm (BA) [74];
- Firefly Algorithm (Fa) [75]
- Gravitational Search Algorithm (GSA) [76];
- Big-Bang Big Crunch algorithm (BBCA) [77];
- Multi-verse optimizer algorithm (MVO) [78].

Eiben e Schippers [79] shows that there is no consensus about the notion of exploration and exploitation in evolutionary computing, and the definitions are not generally accepted. However, it can be observed a general agreement about they work like opposite forces and usually hard to balance. In this sense, a combination of two metaheuristics, called hybrid metaheuristic, can be done to provide a more robust algorithm. For example, one can combine one metaheuristic which exploration is a strong characteristic, but lacks exploitation, with another one which the exploitation is the central ability.

The following is an algorithm template of population-based meta-heuristics:

Algorithm 1: Framework for population-based meta-heuristics

Result: Best Solution found

$\mathcal{P} \leftarrow$ Build initial population ;
$t = 0$;
repeat
  Generate $\mathcal{P}_t^{\top}$; ▷ Generating a new population
  $\mathcal{P}_{t+1} \leftarrow$ Select new Population ($\mathcal{P}_t \cup \mathcal{P}_t^{\top}$);
  $t = t + 1$;
until *Stopping criteria satisfied*;

2.5.1 Standard Particle Swarm Optimization (PSO)

Inspired from swarm intelligence by mimicking the flocking behavior of birds, the PSO algorithm is a stochastic technique proposed by [80, 81], and has been successfully applied in

many types of optimization problems [82]. Each particle in the algorithm represents a candidate solution and consists of two parts: the location in the search space, $\vec{x}_{np,d} \in \mathbb{R}^{np \times d}$, and the respective velocity, $\vec{v}_{np,d} \in \mathbb{R}^{np \times d}$, where $np = 1,2,\cdots,n_a$ and n_a is the size of the swarm and d is the dimension of the problem. In this respect, the following equation represents the initial population:

$$\vec{x}_{np,d} = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,d} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n_a,1} & x_{n_a,2} & \cdots & x_{n_a,d} \end{bmatrix} \tag{2.55}$$

The position $x_{np,d}$ of each particle is adjusted towards the global optimum according to the best position ($pbest_{np}$) encountered by itself and the best solution found by any other particle of the whole swarm, called $gbest_{np}$. At each iteration, t, the position, and velocity of a particle are updated according to

$$v_{nv,d}^{t+1} = \zeta v_{nv,d}^{t} + c_1\kappa_1(pbest_{np}^{t} - x_{np,d}^{t}) + c_2\kappa_2(gbest_{np}^{t} - x_{np,d}^{t}), \tag{2.56}$$

where $\kappa_j \in \mathbb{R}$, for $j = [1,2]$, are a real-valued, continuous random variable in the interval $[0,1]$, $\zeta \in \mathbb{R}$ is an inertia factor to control the influence of the previous velocity on the current one, c_1 is the cognitive factor related to $pbest$ and c_2 is the social related to $gbest$. The values of the velocity, $\vec{v}_{np,d}$, are usually bounded in the range $[v_{min}, v_{max}]$ to guarantee that the randomness of the system do not lead to particles rushing out of the search space.

Each particle updates its position in the search space according to

$$x_{np,d}^{t+1} = x_{np,d}^{t} + v_{np,d}^{t+1}, \tag{2.57}$$

The particles keeping change its location looking for the best solution until meeting a stop criterion. Additionally, it is worth to note that the inertia factor in Equation (2.56), ζ, represents a trade-off between exploration and exploitation: large inertia values boost exploration while smaller inertia values increase exploitation.

Given an optimization problem of the form

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f(x) \\ \text{subject to} \quad & x \in \Omega, \end{aligned} \tag{2.58}$$

where $f(x) : \mathbb{R}^n \to \mathbb{R}$,$\Omega \subseteq \mathbb{R}^n$ is a scalar-valued function representing the objective function, x is the decision variable, and Ω is the constraint set, the PSO algorithm can be summarized as:

Algorithm 2: Pseudo-code of PSO

Result: Best Solution found or reach maximum iteration

$\mathcal{P} \leftarrow$ Build initial population of random particles in the search space, $\mathcal{S}$;
repeat
  Evaluate fitness $f(\vec{x}_{np,d})$; ▷ Equation (2.58)
  foreach $\vec{x}_{np,d} \in \mathcal{P}$ **do**
    Update the velocity ; ▷ Equation (2.56)
    Update the position ; ▷ Equation (2.57)
    if $f(\vec{x}_{np,d}) < f(pbest_{np})$ **then**
      $pbest_{np} \leftarrow \vec{x}_{np,d}$;
    end
    if $f(\vec{x}_{np,d}) < f(gbest_{np})$ **then**
      $gbest_{np} \leftarrow \vec{x}_{np,d}$;
    end
    Update $(\vec{x}_{np,d}, \vec{v}_{np,d})$
  end
until *Stopping criteria satisfied*;

2.5.2 Standard Gravitational Search Algorithm

The GSA algorithm is a stochastic optimization method proposed by [76]. The Newtonian gravity law, which states: "Every particle in the universe attracts every other particle with a force that is directly proportional to the product of their masses and inversely proportional to the square of the distance between them" [83] leads the algorithm.

In GSA, the candidate solutions, called agents, are measured by their masses, which are proportional to their respective values of the fitness function. These agents share information related to their gravitational force in order to attract each other to locations closer to the global optimum. The larger the values of the masses, the best possible solution is achieved, and the agents move more slowly than lighter ones. In GSA, each mass (agent) has four specifications: position, inertial mass, active gravitational mass, and passive gravitational mass. The position of the mass corresponds to a solution to the problem, and its gravitational and inertial masses are determined using a fitness function.

Consider a population formed by agents described in Equation (2.55). At a specific time t, the gravitational force acting on mass i from mass j is defined as [76]:

$$F^t_{i\to j,d} = G(t)\frac{M^t_{pi} \times M^t_{aj}}{R^t_{\|ij\|_2} + \epsilon}(x^t_{j,d} - x^t_{i,d}), \tag{2.59}$$

where M_{aj} is the active gravitational mass related to agent j, M_{pi} is the passive gravitational mass related to agent i, G^t is gravitational constant at time t, ϵ is a small constant, and $R^t_{\|ij\|_2}$ is the Euclidean distance between two agents i and j.

The gravitational constant, G^t, a function of the initialized value G_0 and t, $G^t = G(G_0,t)$, is calculated as:

$$G^t = G_0 \times \exp\left(-\alpha \times \frac{t}{\max(t)}\right), \tag{2.60}$$

where α is the descending coefficient, G_0 the initial value of the gravitational constant, t is the current iteration, and $\max(t)$ is the maximum number of iterations. In a search space with the dimension d, the total force acting on an agent is a randomly weighted sum expressed as:

$$F^t_{i,d} = \sum_{j=1,j\neq i}^{np} \kappa_j \times F^t_{i\to j,d}, \tag{2.61}$$

where $\kappa_j \in \mathbb{R}$ are a real-valued, continuous random variable in the interval $[0,1]$.

The acceleration of an agent is computed according to the law of motion:

$$a^t_{i,d} = \frac{F^t_{i,d}}{M^t_{ii}}, \tag{2.62}$$

where t is a specific time and M_{ii} is inertial the mass of object i.

Therefore, the velocity and position of each agent are updated, respectively, as follow:

$$v^{t+1}_{i,d} = \kappa_i \times v^t_{i,d} + a^t_{i,d}, \tag{2.63}$$

$$x^{t+1}_{i,d} = x^t_{i,d} + v^{t+1}_{i,d}. \tag{2.64}$$

The random number in Equation (2.64), κ, gives a stochastic characteristic to the search [76].

The gravitational and inertial masses are updated by:

$$M_i^t = \frac{m_i^t}{\sum_{j=1}^{np} m_j^t}, \tag{2.65}$$

where

$$m_i^t = \frac{\mathcal{F}_i^t - \mathcal{O}^t}{\mathcal{J}_i^t - \mathcal{O}^t}, \tag{2.66}$$

and $\mathcal{F}_i^t$ is the fitness value of the respective agent. For a minimization problem, $\mathcal{J}_i^t$ and $\mathcal{O}^t$ are respectively defined as:

$$\mathcal{J}_i^t = \underset{j \in \{1, \cdots, np\}}{\text{minimize}} \quad \mathcal{F}_i^t \tag{2.67}$$

$$\mathcal{O}_i^t = \underset{j \in \{1, \cdots, np\}}{\text{maximize}} \quad \mathcal{F}_i(t) \tag{2.68}$$

The GSA algorithm can be summarized as fallows:

Algorithm 3: Pseudo-code of GSA

Result: Best Solution found or reach maximum iteration

$\mathcal{P} \leftarrow$ Build initial population of random agents in the search space, $\mathcal{S}$;
repeat
  Evaluate the fitness for each agent, $\mathcal{F}_i(t)$;
  foreach $\vec{x}_{np,d} \in \mathcal{P}$ **do**
    Update the gravitational constant, G ; ▷ Equation (2.60)
    Update $\mathcal{J}$; ▷ Equation (2.67)
    Update $\mathcal{O}$; ▷ Equation (2.68)
    Calculate the inertial masses of each agent ; ▷ Equation (2.65)
    Calculate the acceleration of each agent ; ▷ Equation (2.62)
    Update the velocity and position of the agents ; ▷ Equation (2.64)
  end
until *Stopping criteria is reached*;

2.5.3 Binary PSO and GSA

In order to solve problems where the variables can only be discrete values, Kennedy e Eberhart [84] and [85] presented a binary version of the PSO and the GSA algorithm, respectively. The discrete approach differs from the continuous version on a mapping between the location of the search agent and the discrete solutions as well by transforming the velocity model in a stochastic model.

Usually, the sigmoidal function is a common choice to transform the continuous values of velocity into discrete ones.

Definition 2.5.1 (Sigmoidal function)**.** Let $\mathcal{F}$ represent a class of bounded functions $\phi : \mathbf{R} \mapsto \mathbf{R}$. If the properties of $\phi(x)$ satisfies

$$\lim_{x\to\infty} \phi(x) = \alpha$$
$$\lim_{x\to-\infty} \phi(x) = \beta \quad \text{with } \alpha > \beta,$$

the function is called sigmoidal. △

Originally, the "S" shaped sigmoid function was used, which is described as

$$S(v_{ik}) = \frac{1}{1+\exp(-v(_{ik}))}, k = 1, 2, \cdots, d. \tag{2.69}$$

However, as shown in [86], the choice of the transfer function has a significant impact on the performance of the binary algorithm. Since the use of Equation (2.69) may cause convergence to local optimum [86], an effective alternative is choosing the following transfer function:

$$S(v_{ik}) = \left|\frac{2}{\pi}\arctan\left(\frac{\pi}{2}v_{ik}\right)\right|. \tag{2.70}$$

This class of transfer function guarantee a better performance of both PSO GSA in terms of avoiding local optimum and convergence rate [86].

Considering a uniformly distributed random number $\kappa \in (0,1)$, the positions of the agents are updates according to

$$x_{np,d}^{t+1} = X(m,n) = \begin{cases} (x_{np,d}^{t})^{-1}, & \text{if } \kappa < S(v_{ik}^{t+1}) \\ x_{np,d}^{t}, & \text{if } \kappa \geq S(v_{ik}^{t+1}). \end{cases} \tag{2.71}$$

Finally, one of the most compelling advantages of the binary representation is the ability to enable binary coding of continuous problems [87].

2.5.4 The hybrid PSOGSA algorithm

As can be observed in most meta-heuristics algorithm, to achieve a good balance between exploration and exploitation phase is a challenging task. In the case of providing a more pow-

erful performance by assuring higher flexibility in the search process, the last two decades focuses on the development of hybrid meta-heuristics. In this thesis, a Hybrid Particle Swarm Optimization and Gravitational Search Algorithm (PSOGSA) hybridized using a low-level co-evolutionary heterogeneous technique [88] proposed by [89] is used.

The main concept of the PSOGSA is to combine the strength of each algorithm to allow a more robust performance. In this respect, the authors associate the high capability of the particles in PSO to scan the whole search space to find the best global solution with the ability to look over local solutions of the GSA. The combination of the algorithms are according to [89]:

$$v_i^{t+1} = \zeta \times v_i^t + \mathrm{c}_1' \times \kappa \times a_i^t + \mathrm{c}_2' \times \kappa \times (gbest - x_i^t), \tag{2.72}$$

where $\mathrm{c}_j' \in \mathbb{R}$ is an acceleration coefficient. The Equation (2.72) has the advantage to accelerate the exploitation phase by saving and using the location of the best mass found so far. However, because this method can affect the exploration phase as well, Mirjalili e Lewis [90] proposed a solution to solve this issue by setting adaptive values for c_j'. So, these coefficients are adapted at each iteration by [91]:

$$\mathrm{c}_1' = -2 \times \frac{t^3}{\max(t)^3} + 2 \tag{2.73}$$

$$\mathrm{c}_2' = 2 \times \frac{t^3}{\max(t)^3} + 2. \tag{2.74}$$

In each iteration, the positions of particles are updated as stated in Equations (2.57) and (2.64).

The PSOGSA is a robust algorithm and is able to solve optimization problems with better results than PSO and GSA [89]. Some advantages of PSOGSA are [90]:

1- The $gbest$ can be accessed in any time due to the memory characteristic of the method;

2- Each agent can access the location of the $gbest$ and move toward it;

3- The $gbest$ cause an individual force on each agent that is not related to gravitational rules. This influences how the agents moves around the search space;

4- A more flexible balance between exploration and exploitation can be achieved by adjusting c'_j;

5- Computationally efficient;

6- The quality of the solution is considered in the update procedure.

The following algorithm summarizes the steps of the PSOGSA:

Algorithm 4: Pseudo-code of PSOGSA

Result: Best Solution found or reach maximum iteration

$\mathcal{P} \leftarrow$ Build initial population of random agents in the search space, $\mathcal{S}$;
repeat
  Evaluate the fitness for each agent, $\mathcal{F}_i(t)$;
  foreach $\vec{x}_{np,d} \in \mathcal{P}$ **do**
    Update the gravitational constant, G ; ▷ Equation (2.60)
    Update $\mathcal{J}$; ▷ Equation (2.67)
    Update $\mathcal{O}$; ▷ Equation (2.68)
    Calculate the inertial masses of each agent ; ▷ Equation (2.65)
    Calculate the acceleration of each agent ; ▷ Equation (2.62)
    Adapt the c'_j coefficients ; ▷ Equation (2.74)
    Update the velocity of the agents ; ▷ Equation (2.72)
    Update the position of the agents ; ▷ Equation (2.64)
  end
until *Stopping criteria is reached*;

2.5.5 Binary version of PSOGSA

The same procedure shown in Section 2.5.3 allows the search agents of PSOGSA to move in a discrete search space. Thus, the probability of agents velocity changes its location. Equations (2.70) and (2.71) are used to perform these tasks. The basic steps of Binary Hybrid Particle Swarm Optimization and Gravitational Search Algorithm (BPSOGSA) are presented in Algorithm 5.

2.6 A brief review of the State-of-the-Art

This section presents some of the main recent contributions related to the system identification using polynomial NARMAX models. In this respect, among the techniques presented, the highlights are: i) the development of mono and multi-objective approaches based on classical algorithms for parameter estimation and MSS; new stochastic and probabilistic frameworks

Algorithm 5: Pseudo-code of BPSOGSA

Result: Best Solution found or reach maximum iteration

$\mathcal{P} \leftarrow$ Build initial population of random agents in the search space, $\mathcal{S}$;
repeat
  Evaluate the fitness for each agent, $\mathcal{F}_i(t)$;
  foreach $\vec{x}_{np,d} \in \mathcal{P}$ **do**
    Update the gravitational constant, G ; ▷ Equation (2.60)
    Update $\mathcal{J}$; ▷ Equation (2.67)
    Update $\mathcal{O}$; ▷ Equation (2.68)
    Calculate the inertial masses of each agent ; ▷ Equation (2.65)
    Calculate the acceleration of each agent ; ▷ Equation (2.62)
    Adapt the c'_j coefficients ; ▷ Equation (2.74)
    Update the velocity of the agents ; ▷ Equation (2.72)
    Calculate the probability of changing position ; ▷ Equation (2.70)
    Update the position of the agents ; ▷ Equation (2.71)
  end
until *Stopping criteria is reached*;

based on population algorithms for MSS. These works are directly associated with the theme developed in this thesis. Besides, it is important to mention that a vast bibliographical review was done, but only the principal results reported in the literature were selected to this brief review.

2.6.1 A randomized algorithm for nonlinear model structure selection [33]

Falsone et al. [33] proposed a new approach based on a probabilistic and random framework to solve the problem of structure selection of polynomial NARX models. The method introduces a probabilistic distribution on the models and updates them progressively, extracting information about the importance of each term from the structure of the selected models at each iteration. The technique is called Randomized Model Structure Selection (RaMSS).

The performance of the RaMSS surpassed the results of other techniques present in the literature, *e.g.*, FROLS, Iterative Orthogonal Foward-Regression (iOFR) and Reversible Jump Markov Chain Monte Carlo (RJMCMC) in terms of the correct selection of the regressors of the model in some very specify scenarios. Additionally, in comparison to other techniques based on random variables, the algorithm proposed by [33] has a higher computational efficiency, which is mainly because of the way that parameters are estimated.

The results were evaluated according to the importance of obtaining models that represent the system under study with high accuracy. Therefore, one should avoid the inclusion of spu-

rious regressors, *i.e*, and the algorithm should not allow the inclusion of regressors that not contribute to relevant information about the system. It is desired to build models that are simple enough to represent a system, preserving all dynamic information along with any critical characteristic of the original behavior.

In this sense, it was observed that the algorithm successfully selected the correct structure in all the test systems, while the traditional FROLS failed, either for obtaining over-parameterized or sub-parameterized models. Among the tested systems were analyzed cases in which the system is excited by low frequencies inputs, a sensitive aspect that may decrease the performance of MSS techniques. In this case study, the RaMSS failed in selecting the correct model structure by selecting spurious regressors to compose the final model. However, the RaMSS was able to build models with structures simpler than the models built by the algorithms used in the comparison.

Further, an electronic system model with a Wiener-Hammerstein structure, often used as a benchmark system for system identification purpose [92, 93], were used as a case study. This case demonstrates the importance of the proposed algorithm since the FROLS obtained an unstable model, and the iOFR obtained an extremely over-parameterized model. While iOFR selected 114 regressors to compose the model, the RaMSS algorithm was able to build a model with only 14 regressors and a better fit to the data than the over-parameterized model selected by iOFR. In other words, the model obtained using the RaMSS algorithm was robust enough to represent the characteristics of the system with a significantly simpler structure.

Despite the good results, the computational cost is high, and the performance is not adequate if the model is excited by low frequencies inputs. Moreover, the technique was presented in a way that its use is feasible only in NARX models. Considering the noise corrupting the data in the identification of real systems, the non-inclusion of noise terms to compose a NARMAX model may become a significant drawback.

2.6.2 An ultra orthogonal forward regression algorithm [94]

Unlike the standard LS criterion that is based on the Euclidean norm of residues, the new criterion presented by Guo et al. [94] is derived from the Sobolev space [95]. The proposed algorithm considers not only the prediction error but also the weak derivative associated with the observed signal and the modeled signal. Essentially, the weak derivative definition presented

in the manuscript says that a weak derivative of a function is a locally integrable function that allows integration by parts, resulting in local information of the function.

Guo et al. [94] states that the standard method relying on LS disregards the information contained in the data, which is crucial to identify a model. Several methods based on classical methods have been developed in order to optimize the search of terms for optimal solutions. Nevertheless, the authors developed a criterion where the search space is better structured and therefore developing a faster algorithm. However, it is worth mentioning that studies must still be done regarding the method, considering that the weak derivative is not always known in a system identification scenario. At the same time, as stated in the paper, in some cases, the components of the cost function chosen by the criterion may not be balanced. Hence the method it is not robust to noise. Furthermore, some situations may become mathematically complex, while the identification process is expected to be easy to understand and computationally efficient.

The algorithm proposed by the authors can be summarized in the following steps:

Step 1 - Define a search space $\mathcal{M}$ with all possible regressors and set a tolerance value ρ;

Step 2 - Specify a test function ψ and calculate its associated weak derivatives;

Step 3 - Normalize the weak derivatives defined in Step 2 to get the modulation functions normalized $\bar{\psi}$;

Step 4 - Compute $\bar{y}$ and, along with $\bar{\psi}$, and formulate the Ultra Least Squares (ULS) problem associated to these variables;

Step 5 - Calculate the ERR for each term in $\mathcal{M}$. Select the term with higher ERR, insert it as the first term of the model and, after that, remove it from $\mathcal{M}$;

Step 6 - At the kth step, orthogonalize each one the remaining terms with the $k-1$ selected term and calculate the ERR of this regressor. Compare the significance of the ERR of each remaining term and select the one with the higher ERR from the remaining terms of the model;

Step 7 - Calculate the summation of the ERR. If $1 - \Sigma$ ERR $< \rho$, stop the algorithm. Otherwise do $k = k + 1$ and repeat the Step 6 until the required condition is satisfied;

Step 8 - Estimate the parameters of the model using LS.

Three different examples were used to compare the performance of the standard FROLS algorithm and the new Ultra Orthogonal Forward-Regression (UOFR). In the first example, while the regressors selected by FROLS presented redundant terms, those chosen using UOFR drastically reduced the number of spurious terms. Besides excluding them from the final model, the UOFR converged faster than the FROLS to find the optimal solution.

The second example presents the structure selection using data from a system excited with low frequency input. The FROLS resulted in two uncorrected regressors, while the UOFR resulted in the selection of all correct regressors and converged more quickly.

In the third example, the spurious term is chosen in the intermediate steps of the regression rather than the first step. It was noted that the FROLS resulted in several incorrect regressors, while the UOFR was able to select all the correct regressors.

It is important to note that the proposed algorithm inherits the computational efficiency of the FROLS in building the final model. However, the purely forward-regression selection process of the UOFR algorithm may find sub-optimal solutions when the test functions are not properly chosen and the ULS problem does not have an optimal substructure.

2.7 Meta-heuristics applied to system identification [2]

In their work, Severino e Araújo [2] proposed a methodology that uses meta-heuristics in the MSS procedure. In their study, the performance of GA, Bat Algorithm (BA) and PSO is compared using a electric heater as benchmark system.

To extend the application of meta-heuristics to select polynomial NARX models, the authors utilized a binary basis of candidate regressors to represent each possible solution. The number 1 in the binary vector indicates that the regressor will be associated with the model, while 0 indicates that the regressor will not be considered for that solution. The parameters are estimated using LS. The AIC was used as the cost function of the optimization problem.

After properly coding the system using binary vectors, the proposed method can be summarized as follows:

Step 1- Define the candidate regressors based on nonlinearity degree, ℓ, and maximum lags of the regressors to limit the search space, $\mathcal{M}$;

Step 2- Define the parameters of each meta-heuristic, *e.g.*, maximum number of iterations and size of the initial population;

Step 3- Set the max value allowed for number of parameters and the AIC;

Step 4- Create a set of initial solutions;

Step 5- Estimate the parameters of each candidate solution and evaluate its respective performance by using AIC;

Step 6- Update both the size of the model and the AIC for each candidate;

Step 7- If the maximum number of iteration is reached, then stop the algorithm and inform the best model. Otherwise, proceed to the next step;

Step 8- Recalculate the set of solutions and return to step 4;

Experimental data collected from an electric heater were used as a case study. At each iteration, a set of 30 potential solutions was evaluated for each meta-heuristic. The set of initial solutions were generated in such a way that each candidate model was formed by only one of the candidate terms.

The results obtained using each one of the metaheuristics were compared with those obtained in [96]. The model built using GA has 5 regressors, while the ones using PSO, BA and the one described in [96] have 7, 7 and 15 regressors, respectively. All models obtained using meta-heuristics fitted the data better than the model presented in [96] based on AIC.

However, the presented procedure has some disadvantages that limits its usefulness. First, although the GA has resulted in a model with 5 regressors, it was the one with the highest degree of non-linearity, $\ell = 3$. This characteristic in a context in which the static behavior of the system is evaluated may be critical. Furthermore, it is necessary to define a maximum number of terms to compose the final model and a minimum value for the AIC. Since the work used previous knowledge regarding the models presented in [96], it was possible to establish such criteria based on those obtained by the original research. However, in real-life situations, neither of these two information is known *a priori*. These drawbacks may even result in over-parameterized models because an inadequate choice of the maximum number of regressors and a higher AIC value directly interfere in the performance of the proposed algorithm.

2.8 Improved Structure Detection For Polynomial NARX Models Using a Multiobjective Error Reduction Ratio [44]

Initially, the Martins et al. [44] stated that, despite the advances in the field of system identification, the MSS was still a challenging problem. The article then adduces that, traditionally, only dynamic data are considered during the structure selection stage, mentioning different techniques to solve the problem. However, these techniques still concentrate its efforts explicitly based on minimizing a single objective function related to the dynamic behavior of the system.

Therefore, given that many systems have dynamic data corrupted by noise or some feature that makes data acquisition difficult, the authors suggested that other types of information besides dynamics should be included in the MSS procedure. However, including affine information about the system in the identification process still lacks studies in the literature. Thus, the MERR is proposed, which allows the inclusion of several types of information, *e.g.*, static curve, and fixed points.

In order to contextualize the proposed method, the ERR is defined. Subsequently, an example is presented to define affine information and how it can be incorporated into the identification process. In this respect, MERR is presented as an extension of the ERR. Considering cases where there is more than one objective to be minimized, MERR calculates the contribution of each regressor in explaining the system by weighing each proposed objective. The inclusion of each regressor is done sequentially, and the first model is composed of the regressor with the maximum calculated MERR. Then, the MERR is calculated for the remainder of the regressors and the one with the highest calculated MERR is added to the model. This procedure is repeated until the model is obtained. The proposed MERR is defined as [44]:

$$\mathrm{MERR}_i = g_i^2 \sum_{j=1}^{m} w_j \frac{\boldsymbol{q}_i^\top \boldsymbol{q}_i}{\boldsymbol{Y}^\top \boldsymbol{Y}}, \tag{2.75}$$

where $w_j \geq$ and $\sum_{j=1}^{m} w_j = 1$ for $m \in \mathbb{N}^*$. Therefore, the authors analyze the proposed method handling the mono-objective procedure. Since the MERR is calculated through the weighted sum of weights related to each auxiliary information, the ERR can be obtained directly using only dynamic data and setting the weight for the objective related to dynamic response as the maximum value. In this way, the importance given to any auxiliary information will be nullified.

Additionally, by varying the weights associated with each objective allows obtaining a set of models in the Pareto Optimum and therefore, each model can have a different structure.

To verify the performance of the method, the authors detailed two case studies: a simulated system and a real system. The first makes the modeling process challenging due to the type of input used to excite the system. The static data of this system were generated by applying concepts of terms cluster. MERR was able to reconstruct the original model using both dynamic data and dynamic data in conjunction with the static one, whereas the ERR suggests including two spurious regressors to the model. That result can be explained because static system information is less susceptible to noise presented in the proposed system, which allows the multi-objective method to select the correct terms.

The second case study highlights the identification of a CC CC buck converter. In this case, two comparisons were made. The first compares the MERR with ERR, both with the parameters estimated by means of the Extended Least Squares (ELS). It was observed that the MERR give efficient solutions in the Pareto optimal set, which allows more representative models than the ERR considering different system operating points.

The second comparison takes into account both MERR and ERR with parameters estimated by exploiting auxiliary information, given a model structure known *a priori*. The Decision-maker was based on the criterion of minimum correlation. It was verified that although the static behavior improves in cases where the parameters are estimated using a multi-objective technique, the dynamic error also increases, which ratifies the importance of the proposed technique.

2.8.1 Determination of the number of terms in polynomial models: the Multi-objective Akaike Information Criterion [97]

Firstly, Martins et al. [97] point out that an often used method to obtain representative and stable models is by using auxiliary information in the structure selection framework. They stated that, although generally multi-objective techniques are only used in the parameter estimation stage, the emergence of methods such as the MERR enables a higher concentration in the use of auxiliary information directly in the MSS framework.

Because the MERR allows a better interpretation related to the contribution of each regressor in a multi-objective scenario, the authors draw attention to the problem of model order selection.

As previously stated, in a single-objective procedure, the AIC is often used in conjunction with the ERR to determine the number of regressors that a model must compose.

Thus, given the need to select the optimal order of a model considering cases where there is more than one objective to be minimized, the Multi-objective Akaike Information Criterion (MAIC) is presented. The main advantage of the MAIC is making it possible to obtain a set of solutions only by varying the weights related to each objective. In this way, good models can be selected considering both dynamic and static behavior, without abdicating performance in none of the cases.

The MAIC is proposed as a direct expansion of the AIC. Hence, the proposed method focuses on providing the correct number of terms by evaluating the complexity of the model and how much it reduces the variance of the residue considering, in addition to the dynamic residues, any other objective. Consequently, the appropriate number of terms to insert into the model occurs when the MAIC is minimized:

$$J_{\text{MAIC}} = \sum_{j=1}^{m} [w_j n_j \log\left(\text{Var}[\xi_j]\right)] + 2n_{\Theta}, \tag{2.76}$$

where $w_j \geq$ and $\sum_{j=1}^{m} w_j = 1$ for $m \in \mathbb{N}^*$.

The MAIC technique is applied in the identification of a CC CC buck converter excited by a PRBS signal. According to the authors, this system presents a similar static and nonlinear dynamic characteristic, which makes the modeling problem complex. If only the dynamic behavior of the system is considered during the identification process, the obtained model fails to represent the other characteristics of the system. In this study, the regressors were selected using the FROLS, and the number of terms as defined by the MAIC.

Considering the linearly spaced weights for the dynamic and static objective, the authors obtained models with 1, 4 and 9 regressors under different combinations of weights. It was also observed that the technique provides a range in which the size of the model should be chosen based on the association of weights. Besides, the structure of the models obtained differs greatly depending on the importance given to each objective, showing that the inclusion of auxiliary information and use of the MAIC offers a good set of models associated with each characteristic of the system.

Although the technique helps to select the structure of polynomial NARX models, there

is no adequate way to distribute the weight of each objective. Moreover, there is no robust methodology for decision-maker regarding how to select the best model.

2.9 Equivalence of non-linear model structures based on Pareto uncertainty [45]

Barbosa et al. [45] highlight the importance of the MSS framework, especially when modeling nonlinear systems. Although several papers present new methods for structure selection, especially in a mono-objective context, this is still an open area.

Concerning the use of multi-objective approaches, the authors have noticed that, in the last two decades, several multi-objective algorithms for parameter estimation have been investigated in the scope of systems identification. In this context, the possibility of using more than one set of data to identify nonlinear systems can be worthwhile, since the multi-objective estimator along with a suitable decision-maker is not biased. However, it is noted that the use of multi-objective algorithms in structure selection problems has been a neglected area in the field of system identification.

Given this gap, the authors introduced a strategy for selecting the most consistent models concerning dynamic and static data of the system. For the methodology, the authors use a bi-objective approach for parameters estimation. Different from the method detailed in [44], which aims to determine the nominal structure, for the simulated case, or a unique structure for the experimental case, this work considers that there is not a single compatible model structure with the data.

The authors use a hypothesis test known as the Wilcoxon Rank-Sum test in the methodology. This hypothesis is a non-parametric test that compares the distributions of two independent samples. Further, the work presented a method that discards the less significant model structures among a set of possibilities. Since the technique is formulated through a non-parametric hypothesis test, the user can define a desired level of significance and, consequently, obtain results whose interpretation is more straightforward.

When a structure of a model is obtained, the parameters are estimated using a bi-objective estimator. If the j th structure of the model is used, considering that there is uncertainty in the data, the parameter vector will also have uncertainties. Consequently, there are uncertainties

in the Pareto set whose amplitudes are indirectly used in the hypothesis test. To verify if the hypothesis can be rejected considering a set of data, the following steps are performed:

Step 1- Calculate the populations of Pareto sets;

Step 2- Randomly select a set of Pareto sets for a pair of model structures,. From each selected Pareto set, choose the model based on the minimum correlation decision-maker. From this group, proceed in a similar way to produce a new set of models;

Step 3- Choose a value based on the defined maximum confidence value and execute the Wilcoxon Rank-Sum test;

Step 4- Repeat Steps 2 and 3 as many times as desired;

Step 5- Calculate the percentage of times the hypothesis has not been rejected. This percentage, called w_{jk}, is the binding force between the pair of model structures;

Step 6- Repeat the Steps from 2 to 5 for all model structures in the Pareto set taken in pairs.

Step 7- Select as *leader* standard structure the one with the highest mean of wj_{jk} calculated;

Step 8- Set a threshold parameter, β. Structures connecting to the leader standard structure are considered to be equivalent with the significance level chosen in step 3 and are inserted into the group formed with the equivalent candidate structures, G^*.

To validate the proposed method, the authors conduct two case studies: a simulated example and an experimental one. In the former case, the performance of the models with a structure in the group G^* is not identical. All the structures that compose G^* must be used to determine the limits of the dynamic behavior of the analyzed system.

In the latter case, which deals with the identification of a CC-CC buck converter, the method was able to obtain a set of equivalent structures, which are theoretically more suitable to represent the analyzed system.

However, the method does not present a metric capable of attesting that all models of the set G^* have the same importance. Further, the method does not validate the procedure for

other mathematical representation or models with different objective functions, convex or non-convex. Moreover, it is worth mentioning that the Wilcoxon Rank-Sum should be only used when the data are not paired and the samples are independent and random with variables numerical or ordinal.

2.9.0.1 Conclusion

Section 2.6 compiled a few works in the field of system identification related to mono and multi-objective techniques for MSS. Regarding mono-objective based algorithms, expansions of classical approaches and contemporary methodologies were highlighted. In the multi-objective scenario, either work focusing on multi-objective parameter estimation or research aiming attention on structure selection were detailed. Supplementary to the methodologies as mentioned earlier, the following articles were analyzed.

In [98], the authors presented a bi-objective approach to parameter estimation that makes use of the simulation error instead of the prediction error, as well as a new decision-maker based on uncertainties to select models with excellent dynamic and static characteristics. In this case, the optimization problem is solved using a genetic algorithm. However, the proposed algorithm has a high computational cost and cannot be explicitly used for structure selection.

The use of auxiliary information is used by Teixeira e Aguirre [99] to estimate the parameters of NARX model. The authors present the bi-objective uncertainty-weighted least squares technique to estimate the parameters using dynamic and static data. The technique yields good results even in cases where static data is not known accurately. However, the static performance of the models obtained by the technique is more sensitive to the tuning of the algorithm than the dynamic performance.

Martins e Nepomuceno [100] present a methodology that favors distributing the weights assigned to each objective in a multi-objective system identification problem solved by the weighted sum method. The technique relates the derivative of the multi-objective cost function to the weights performed. Thus, with the same number of solutions, it is possible to obtain a set of solutions in Pareto regions where there are no solutions by default.

Barroso et al. [101] present a systematic procedure for selecting models that make up the Pareto set. The methodology is based on a procedure that is divided into two steps: generation of the Pareto set employing a multi-objective procedure and selection of a single model based on

the simulation error, which must be minimally correlated with the model output. The authors used dynamic and static data and systems whose structure of the models are already known *a priori*. Although it is a computationally efficient technique, it does not treat, like others previously mentioned, the problem of structure selection.

Moreover, [102–104] are works that deal with the problem of identification of polynomial NARX models in which it is desired to optimize more than one objective. In respect to MSS, it is worth to mention the following papers: [59, 105–107].

CHAPTER 3

MONO-OBJECTIVE FRAMEWORK FOR REGRESSION AND CLASSIFICATION

It has been mentioned that the NARX methodology selects a set of candidate regressors based on their contribution to minimize the output error and build parsimonious models [5]. Some authors have worked towards randomized approach within the NARX methodology. The results obtained are remarkable, however there are significant drawbacks such as involving many assumptions in the probability distributions of the parameters, which may not always be accurate; many parameters need to be tuned, and constraints must be added to guarantee a good performance.

In this chapter, both the use of a novel method based on meta-heuristic to select the regression model structure based on data set and the extension of the method to a Logistic NARX model are addressed. For the first case, the BPSOGSA is implemented to search for the best model structure in a decision space formed by a pre-defined dictionary of regressors. The objective function of the optimization problem is based on the root mean squared error of the free run simulation output multiplied by a penalty factor that takes into account the complexity and the individual contribution of each regressor to build the final model. For the latter case, the method combines the logistic function with the NARX representation in order to obtain a probability model, which structure is selected using the BPSOGSA. In this case, however, the objective function is based on the biserial correlation multiplied by the same penalty factor used for the regression problem.

The following subsections briefly describe the encoding scheme, the formulation of the objective function, and the penalty value used for mono-objective model structure selection.

3.1 Encoding scheme

The use of BPSOGSA for model structure selection is described. First, one should define the dimension of the test function. In this regard, the n_y, n_x and ℓ are set to generate all possibilities

of regressors and a general matrix of regressors, Ψ, is built. The number of columns of Ψ is assigned to the variable noV, and the number of agents, N, is defined. Then a binary $noV \times N$ matrix referred as $\mathcal{X}$, is randomly generated with the position of each agent in the search space. Each column of $\mathcal{X}$ represents a possible solution; in other words, a possible model structure to be evaluated at each iteration. Since each column of Ψ corresponds a possible regressor, a value of 1 in $\mathcal{X}$ indicates that, in its respective position, the column of Ψ is included in the reduced matrix of regressors, while the value of 0 indicates that the regressor column is ignored.

Example 3.1.1. Consider a case where all possible regressors are defined based on $\ell = 1$ and $n_y = n_u = 2$. The Ψ is defined by

$$[constant\ y(k-1)\ y(k-2)\ u(k-1)\ u(k-2)] \tag{3.1}$$

Because there are 5 possible regressors, $noV = 5$. Assume $N = 5$, then $\mathcal{X}$ can be represented, for example, as

$$\mathcal{X} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \end{bmatrix} \tag{3.2}$$

The first column of $\mathcal{X}$ is transposed and used to generate a candidate solution:

$$\begin{array}{ccccc} [constant & y(k-1) & y(k-2) & u(k-1) & u(k-2)] \\ 0 & 1 & 0 & 0 & 1 \end{array}$$

Therefore, in this example, the first model to be tested is $\alpha y(k-1) + \beta u(k-2)$, where α and β are parameters estimated via LS method. After that, the second column of X is tested and so on. $\triangle$

3.1.1 Formulation of the objective function

Because each candidate model structure is randomly defined, the linear-in-the-parameters system can be solved directly using the LS method. The variance of estimated parameters can be calculated as:

$$\hat{\sigma}^2 = \hat{\sigma}_e^2 V_{jj}, \tag{3.3}$$

where $\hat{\sigma}_e^2$ is the estimated noise variance calculated as

$$\hat{\sigma}_e^2 = \frac{1}{N-m}\sum_{k=1}^{N}(y_k - \psi_{k-1}^{\top}\hat{\Theta}) \tag{3.4}$$

and V_{jj} is the jth diagonal element of $(\Psi^{\top}\Psi)^{-1}$. Furthermore, $\hat{\sigma}^2$ is a symmetric matrix whose diagonal elements, $\hat{\sigma}_{jj}^2$, represent the variance of the estimated jth regression coefficient, $\hat{\Theta}_j$. The off-diagonal elements of $\hat{\sigma}^2$ represent the covariance between the ith and jth estimated regression coefficients, $\hat{\Theta}_i$ and $\hat{\Theta}_j$. The estimated standard error of the jth regression coefficient $\hat{\Theta}_j$ is the positive square root of $\hat{\sigma}_{jj}^2$,

$$\mathrm{se}(\hat{\Theta}_j) = \sqrt{\hat{\sigma}_{jj}^2} \tag{3.5}$$

A penalty test considers the standard error of the regression coefficients to determine the statistical relevance of each regressor. The t-test is used in this thesis to perform a *hypothesis test* on the coefficients to check the significance of individual regressors in the multiple linear regression model. The hypothesis statements involve testing the *null hypothesis* described as:

$$\boldsymbol{H_0} : \Theta_j = 0,$$
$$\boldsymbol{H_a} : \Theta_j \neq 0.$$

In practice, one can compute a t-statistic as

$$T_0 = \frac{\hat{\Theta}_j}{\mathrm{se}(\hat{\Theta})}, \tag{3.6}$$

which measures the number of standard deviations that $\hat{\Theta}_j$ is away from 0. More precisely, let

$$-t_{\alpha/2,N-m} < T < t_{\alpha/2,N-m}, \tag{3.7}$$

where $t_{\alpha/2,N-m}$ is the t value obtained considering α as the significance level and $N-m$ the degree of freedom. Then, If T_0 does not lie in the acceptance region of Equation (3.7), the *null hypothesis*, $\boldsymbol{H_0}: \Theta_j = 0$, is rejected and it is concluded that Θ_j is significant at α. Otherwise, θ_j is not significantly different from zero, and the null hypothesis $\theta_j = 0$ cannot be rejected. In the latter case, the corresponding regressor is considered to be statistically irrelevant for the given model, and therefore, one should consider there exist no relationship between the regressor and the output data, in a statistical point of view.

3.1.1.1 Derivative of the Sigmoid function

In this particular case and following Definition 2.5.1 with $\alpha = 0$ and $\beta = 1$, we write a "S" shaped curve as

$$\varsigma(x) = \frac{1}{1 + e^{-a(x-c)}}. \tag{3.8}$$

In that case, we can specify a, the rate of change. If a is close to zero, the sigmoid function will be gradual. If a is large, the sigmoid function will have an abrupt or sharp transition. If a is negative, the sigmoid will go from 1 to zero. The parameter c corresponds to the x value where $y = 0.5$.

We proposed a penalty value based on the derivative of the Equation (3.8), which is defined as:

$$\dot{\varsigma}(x(\varrho)) = \varsigma(x)[1 + (a(x-c))(1-\varsigma(x))]. \tag{3.9}$$

In this respect, the parameters of Equation (3.9) are defined as follows: x has the dimension of noV; $c = noV/2$; and a is defined by the number of regressors of the current test model divided by c. These parameters are used to adapt the values of $\dot{\varsigma}(x)$ with respect to any candidate model structure resulting in a different curve for each model, considering the number of regressors of the current model. As the number of regressor increases, the slope of the sigmoid curve becomes steeper. The penalty value, (ϱ), corresponds to the value in y of the correspondent sigmoid curve regarding the number of regressor in x. This approach penalizes models differently regarding the number of regressors. The choice of the derivative of the sigmoid as a penalty function is justified given that both sub-parametrization and over-parametrization can be contemplated. Figure 3.1 depicts the difference between the curve of a sigmoid and its derivative.

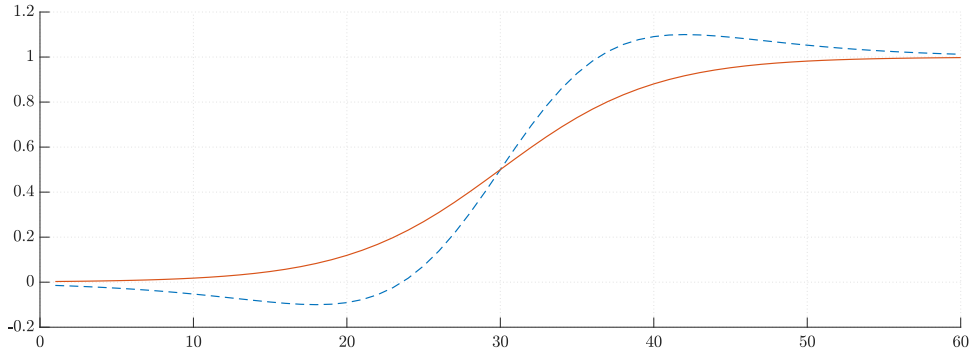


Figure 3.1: The derivative of the sigmoid function looks like an steeper and "overshooting" sigmoid function. The red line $(-)$ represents the sigmoid function and the blue line $(--)$ represents its derivative.

However, two different models can have the same number of regressors and present significantly different results. This situation can be explained based on the importance of each regressor in the composition of the model. In this respect, we use the t-student test to determine the statistical relevance of each regressor and introduce this information on the penalty function. In each case, the procedure returns the number of regressors that are not significant for the model (aux). Then, the penalty value is chosen considering the model sizes as

$$ng_{new} = ng + aux. \tag{3.10}$$

The objective function considers the relative root squared error of the model and ϱ and is defined as

$$\mathcal{F} = \frac{\sqrt{\sum_{k=1}^{n}(y_k - \hat{y}_k)^2}}{\sqrt{\sum_{k=1}^{n}(y_k - \bar{y})^2}} \times \varrho. \tag{3.11}$$

With this approach, even if the tested models have the same number of regressors, the model which contain redundant regressors are penalized with a more substantial penalty value. It is imperative to point out that because the derivative of the sigmoid function return values less than zero, we normalize the values as

$$\varrho = \varsigma - \min(\varsigma). \tag{3.12}$$

Finally, the Algorithm 6 summarizes the method.

Algorithm 6: Meta-structure selection algorithm

Result: Model which has the best fitness value

Input: $\{(u_k), (y_k), k = 1, \ldots, N\}$, $\mathcal{M} = \{\psi_j, j = 1, \ldots, m\}$, n_y, n_u, ℓ, max_iteration, noV, np

1 $\mathcal{P} \leftarrow$ Build initial population of random agents in the search space, $\mathcal{S}$
2 $v \leftarrow$ set the agent's velocity equal zero at first iteration
3 $\Psi \leftarrow$ Build the general matrix of regressors based on n_y, n_u and ℓ ▷ Equation (2.28)
4 **repeat**
5 **for** $i = 1 : d$ **do**
6 $\mathrm{m_i} \leftarrow \vec{x}_{np,i}$ ▷ Extract the model encoding from population
7 $\Psi_r \leftarrow \Psi(\mathrm{m_i})$ ▷ Delete the Ψ columns where $\mathrm{m_i} = 0$ Example 3.1.1
8 $\hat{\Theta} \leftarrow (\Psi_r^\top \Psi_r)^{-1} \Psi_r^\top y$ ▷ Equation (2.42)
9 $\hat{y} \leftarrow$ Free-run simulation of the model
10 $V \leftarrow (\Psi^\top \Psi)^{-1}$
11 $\hat{\sigma}_e^2 = \frac{1}{N-m} \sum_{k=1}^{N} (y_k - \psi_{k-1}^\top \hat{\Theta})$ ▷ Equation (3.4)
12 **for** $h = 1 : \tau$ **do**
13 $\hat{\sigma}^2 \leftarrow \hat{\sigma}_e^2 V_{h,h}$ ▷ Equation (3.3)
14 $\mathrm{se}(\hat{\Theta}_j) \leftarrow \sqrt{\hat{\sigma}_{h,h}^2}$ ▷ Equation (3.5)
15 $T_0 \leftarrow \frac{\hat{\Theta}_j}{\mathrm{se}(\hat{\Theta})}$ ▷ Equation (3.6)
16 $p \leftarrow$ regressors where $-t_{\alpha/2,N-m} < T_0 < t_{\alpha/2,N-m}$ ▷ Equation (3.7)
17 **end**
18 Remove the p regressors from Ψ_r
19 Check for empty model
20 **if** *Model is empty* **then**
21 Generate a new population
22 Repeat the steps from line 6 to 18
23 **end**
24 $n_1 \leftarrow \mathrm{size}(p)$ ▷ Number of redundant terms
25 $\hat{\Theta} \leftarrow (\Psi_r^\top \Psi_r)^{-1} \Psi_r^\top y$ ▷ Re-estimation | Equation (2.42)
26 $\mathcal{F}_i \leftarrow \frac{\sqrt{\sum_{k=1}^{n} (y_k - \hat{y}_k)^2}}{\sqrt{\sum_{k=1}^{n} (y_k - \bar{y})^2}} \times \varrho$ ▷ Equation (3.11)
27 $\mathcal{P}_i^n \leftarrow$ Encoded Ψ_r
28 Evaluate the fitness for each agent, $\mathcal{F}_i(t)$
29 **end**
30 $\mathcal{P} \leftarrow \mathcal{P}^n$ ▷ Update the population
31 **foreach** $\vec{x}_{np,d} \in \mathcal{P}$ **do**
32 Update the gravitational constant, G ▷ Equation (2.60)
33 Update $\mathcal{J}$ ▷ Equation (2.67)
34 Update $\mathcal{O}$ ▷ Equation (2.68)
35 Calculate the inertial masses of each agent ▷ Equation (2.65)
36 Calculate the acceleration of each agent ▷ Equation (2.62)
37 Adapt the c_j' coefficients ▷ Equation (2.74)
38 Update the velocity of the agents ▷ Equation (2.72)
39 Update the position of the agents ▷ Equation (2.64)
40 **end**
41 **until** *max_iterations is reached*

3.2 Case Studies: Simulation Results

In this section, six simulation examples are considered to illustrate the effectiveness of the Meta-Structure Selection algorithm. An analysis of the algorithm performance has been carried out considering different tuning parameters. The selected systems are generally used as a benchmark for model structures algorithms and were taken from [1, 24, 26, 29, 33, 42, 108]. Finally, a comparative analysis with respect to the RaMSS [33], the FROE [5], and the RJMCMC [1] algorithms has been accomplished to check out the quality of the proposed method.

The simulation models are described as:

$$S_1: \quad y_k = -1.7y_{k-1} - 0.8y_{k-2} + x_{k-1} + 0.81x_{k-2} + e_k, \tag{3.13}$$
$$\text{with } x_k \sim \mathcal{U}(-2, 2) \text{ and } e_k \sim \mathcal{N}(0, 0.01^2);$$

$$S_2: \quad y_k = 0.8y_{k-1} + 0.4x_{k-1} + 0.4x_{k-1}^2 + 0.4x_{k-1}^3 + e_k, \tag{3.14}$$
$$\text{with } x_k \sim \mathcal{N}(0, 0.3^2) \text{ and } e_k \sim \mathcal{N}(0, 0.01^2).$$

$$S_3: \quad y_k = 0.2y_{k-1}^3 + 0.7y_{k-1}x_{k-1} + 0.6x_{k-2}^2 - 0.7y_{k-2}x_{k-2}^2 - 0.5y_{k-2} + e_k, \tag{3.15}$$
$$\text{with } x_k \sim \mathcal{U}(-1, 1) \text{ and } e_k \sim \mathcal{N}(0, 0.01^2).$$

$$S_4: \quad y_k = 0.7y_{k-1}x_{k-1} - 0.5y_{k-2} + 0.6x_{k-2}^2 - 0.7y_{k-2}x_{k-2}^2 + e_k, \tag{3.16}$$
$$\text{with } x_k \sim \mathcal{U}(-1, 1) \text{ and } e_k \sim \mathcal{N}(0, 0.04^2).$$

$$S_5: \quad y_k = 0.7y_{k-1}x_{k-1} - 0.5y_{k-2} + 0.6x_{k-2}^2 - 0.7y_{k-2}x_{k-2}^2 + 0.2e_{k-1} - 0.3x_{k-1}e_{k-2} + e_k, \tag{3.17}$$
$$\text{with } x_k \sim \mathcal{U}(-1, 1) \text{ and } e_k \sim \mathcal{N}(0, 0.02^2);$$

$$S_6: \quad y_k = 0.75y_{k-2} + 0.25x_{k-2} - 0.2y_{k-2}x_{k-2} + e_k$$
$$\text{with } x_k \sim \mathcal{N}(0, 0.25^2) \text{ and } e_k \sim \mathcal{N}(0, 0.02^2);$$

where $\mathcal{U}(a, b)$ are samples evenly distributed over $[a, b]$, and $\mathcal{N}(\eta, \sigma^2)$ are samples with a Gaussian distribution with mean η and standard deviation σ. All realizations of the systems are composed of a total of 500 input-output data samples. Also, the same random seed is used to reproducibility purpose.

All tests have been performed in Matlab® 2018a environment, on a Dell Inspiron 5448 Core i$5-5200$U CPU 2.20GHz with 12GB of RAM.

Following the aforementioned studies, the maximum lags for the input and output are chosen to be $n_u = n_y = 4$ and the nonlinear degree is $\ell = 3$. The parameters related to the BPSOGSA are detailed on Table 3.1.

Table 3.1: Parameters used in Meta-Structure Selection Algorithm

Parameters	n_u	n_y	ℓ	p-value	max_iter	n_agents	α	G_0
Values	4	4	3	0.05	30	10	23	100

Aiming to compare some statistics about the algorithm performance, 300 runs of the Meta-Structure Selection algorithm have been executed for each model. The elapsed time, the time required to obtain the final model, and correctness, the percentage of exact model selections, are analyzed.

The results in Table 3.2 are obtained with the parameters configured accordingly to Table 3.1.

Table 3.2: Overall performance of the Meta-MSS

	S_1	S_2	S_3	S_4	S_5	S_6
Correct model	100%	100%	100%	100%	100%	100%
Elapsed time (mean)	5.16s	3.90s	3.40s	2.37s	1.40s	3.80s

Table 3.2 shows that all the model terms are correctly selected using the Meta-Structure Selection Algorithm. It is worth noting that even the model S_5, which have an autoregressive noise, was correctly selected using the proposed algorithm. This result resides in the evaluation of all regressors individually, and the ones considered redundant are removed from the model.

Figure 3.2 present the convergence of each execution of Meta-Structure Selection Algorithm. It is noticeable that the majority of executions converges to the correct model structures with 10 or fewer iterations. The reason for this relies on the maximum number of iterations and the number of search agents. The first one is related to the acceleration coefficient, which boosts the exploration phase of the algorithm, while the latter increases the number of candidate models to be evaluated. Intuitively, one can see that both parameters influence the elapsed time and, more importantly, the model structure selected to compose the final model. Consequently, an inappropriate choice of one of them may results in sub/over-parameterized models, since the algorithm can converge to a local optimum. As stated in Chapter 2, sub/over-parameterized models can lead to a misinterpretation of the underlying system properties. Analyses of the max_iter and n_agents influence in the algorithm performance are carried out in the next sub-

section.

3.2.1 Influence of the max_iter and n_agents parameters

The simulation models are used to show the performance of the Meta-Structure Algorithm considering different values for max_iter and n_agents parameters. First, we set the max_iter $=$ 30 while the n_agents are changed. Then, we set the n_agents while the max_iter is modified. The results detailed in this section have been obtained by setting the remaining parameters according to Table 3.1.

Table 3.3: Results regarding the influence of the max_iter and n_agents parameters

		S_1	S_2	S_3	S_4	S_5	S_6
max_iter $=$ 30, n_agents $=$ 1	Correct model	65%	55.66%	14%	14%	7.3%	20.66%
	Elapsed time (mean)	0.26s	0.19s	0.15s	0.11s	0.13s	0.13
max_iter $=$ 30, n_agents $=$ 5	Correct model	100%	100%	99%	98%	91.66%	98.33%
	Elapsed time (mean)	2.08s	1.51s	1.41s	0.99s	0.59s	1.13s
max_iter $=$ 30, n_agents $=$ 20	Correct model	100%	100%	100%	100%	100%	100%
	Elapsed time (mean)	12.88s	9.10s	8.77s	5.70s	3.37s	9.50s
max_iter $=$ 5, n_agents $=$ 10	Correct model	96.33%	99%	86%	93.66%	93%	97.33%
	Elapsed time (mean)	0.92s	0.73s	0.72s	0.52s	0.29s	0.64s
max_iter $=$ 15, n_agents $=$ 10	Correct model	100%	100%	99%	99%	100%	100%
	Elapsed time (mean)	2.80s	2.33s	2.25s	1.60s	0.90s	2.30s
max_iter $=$ 50, n_agents $=$ 10	Correct model	100%	100%	100%	100%	100%	100%
	Elapsed time (mean)	7.38s	5.44s	4.56s	3.01s	2.10s	4.52s

The aggregated results in Table 3.3 confirms the expected performance regarding the elapsed time and percentage of correct models. Indeed, both metrics increases significantly as the number of search agents and the maximum number of iteration increases. The number of search agents is very relevant because it yields a broader exploration of the search space. All systems are affected by the increase in the number of agents and the maximum number of iterations.

Regarding all tested systems, it is straightforward to notice that the more extensive exploration dramatically impacts on the exactitude of the selection procedure. If only a few agents are assigned, the performance of Meta-Structure Selection algorithm deteriorates significantly, especially for systems S_3, S_4 and S_5. The maximum number of iteration empowers agents to explore, globally and locally, the space around the candidate models tested so far. In this sense, as the number of iterations increases, more the agents can explore the search space and examine different regressors.

It is worth mentioning that the Meta-Structure Selection Algorithm was able to select the exact model structure of S_1 and S_2 more often than remaining systems, considering the cases

where the max_iter and n_agents were tuned with small values. This result is interesting because S_1 is a linear model, and thus, the chosen search space is largely over-parameterized for its simple linear structure.

If these parameters are improperly chosen, the algorithm might fail to select the best model structure. In this respect, the results presented here concerns only the selected systems. The larger the search space, the larger the number of agents and iterations should be. Although the computational effort increases with larger values for n_agents and max_iteration, the algorithm remains very efficient regarding the elapsed time for all tuning configurations that ensured the selection of the exact model structures.

3.2.2 Selection of over and sub-parameterized models

Regardless of the successful selection of all model structures by the Meta-Structure Selection Algorithm, one can ask how the models differ from the true ones in the cases presented in Table 3.3 where the algorithm failed to ensure 100% of correctness. Figure 3.3 depicts the distribution of terms number selected in each case. It is evident that the number of over-parameterized models selected is higher than the sub-parameterized overall. Regarding the cases where the number of search agents is low, due to low exploration and exploitation capacity, the algorithm converged early and resulted in models with a high number of spurious regressors. In respect to S_2 and S_5, for example, with n_agents $= 1$, the algorithm ends up selecting models with more than 20 terms. One can say this was an extreme scenario for comparison purpose. However, a suitable choice for the parameters is intrinsically related to the dimension of the search space. By referring to cases where n_agents ≥ 5, the number of spurious terms decreased significantly where the algorithm failed to select the true models.

Furthermore, it is interesting to point out the importance of tuning the parameters properly because since the exploration and exploitation phase of the algorithm is strongly dependent on them. A premature convergence of the algorithm may result in models with the actual number of terms, but with wrong ones. This issue happened with all cases with n_agents $= 1$. For example, the algorithm generates models with the correct number of terms in 33.33% of the cases analyzed regarding S_3. However, Table 3.3 shows that only 14% are, in fact, equivalent to the true model. Moreover, if any system requires a model structure with a high number of terms, the max_iter and n_agents might be increased to allow a better exploration and exploitation

performances.

3.2.3 Meta-MSS vs RaMSS vs C-RaMSS

Systems S_1, S_2, S_3, S_4 and S_6 has been used as benchmark by Bianchi e Piroddi [109], so we can compare directly our results with those reported by the author in the thesis. All techniques used $n_y = n_u = 4$ and $\ell = 3$. The C-RaMSS and the RaMSS used the following configuration for the tuning parameters: $K = 1$, $\alpha = 0.997$, $NP = 200$ and $v = 0.1$. The Meta-Structure Selection Algorithm was tuned according to Table 3.1.

Table 3.4: Comparative analysis

		S_1	S_2	S_3	S_4	S_6
Meta-MSS	Correct model	100%	100%	100%	100%	100%
	Elapsed time (mean)	5.16s	3.90s	3.40s	2.37s	3.80s
RaMSS- $NP = 100$	Correct model	90.33%	100%	100%	100%	66%
	Elapsed time (mean)	3.27s	1.24s	2.59s	1.67s	6.66s
RaMSS- $NP = 200$	Correct model	78.33%	100%	100%	100%	82%
	Elapsed time (mean)	6.25s	2.07s	4.42s	2.77s	9.16s
C-RaMSS	Correct model	93.33%	100%	100%	100%	100%
	Elapsed time (mean)	18s	10.50s	16.96s	10.56s	48.52s

In terms of correctness, the Meta-MSS outperforms (or at least equals) the RaMSS and C-RaMSS for all analyzed systems (Table 3.4). Regarding S_6, the correctness rate increased by 18% when compared with RaMSS and the elapsed time required for C-RaMSS obtain 100% of correctness is 1276.84% higher than the Meta-MSS. Furthermore, the Meta-MSS is notably more computationally efficient than C-RaMSS and similar to RaMSS.

3.2.4 Meta-MSS vs FROLS

The FROLS algorithm [5] has been tested on all the systems and the results are detailed in Table 3.5. It can be seen that only the model terms selected for S_2 and S_6 are correct using FROLS. The FROLS fails to select two out of four regressors for S_1. Regarding S_3, the term y_{k-1} is included in the model instead of y^3_{k-1}. Similarly, the term y_{k-4} is wrongly added in model S_4 instead of y_{k-2}. Finally, an incorrect model structure is returned for S_5 as well with the addition of the spurious term y_{k-4}.

Table 3.5: Comparative analysis - Meta-MSS vs FROLS

	Meta-MSS		FROLS	
	Regressor	Correct	Regressor	Correct
S_1	y_{k-1}	yes	y_{k-1}	yes
	y_{k-2}	yes	y_{k-4}	no
	x_{k-1}	yes	x_{k-1}	yes
	x_{k-2}	yes	x_{k-4}	no
S_2	y_{k-1}	yes	y_{k-1}	yes
	x_{k-1}	yes	x_{k-1}	yes
	x_{k-1}^2	yes	x_{k-1}^2	yes
	x_{k-1}^3	yes	x_{k-1}^3	yes
S_3	y_{k-1}^3	yes	y_{k-1}	no
	$y_{k-1}x_{k-1}$	yes	$y_{k-1}x_{k-1}$	yes
	x_{k-2}^2	yes	x_{k-2}^2	yes
	$y_{k-2}x_{k-2}^2$	yes	$y_{k-2}x_{k-2}^2$	yes
	y_{k-2}	yes	y_{k-2}	yes
S_4	$y_{k-1}x_{k-1}$	yes	$y_{k-1}x_{k-1}$	yes
	y_{k-2}	yes	y_{k-4}	no
	x_{k-2}^2	yes	x_{k-2}^2	yes
	$y_{k-2}x_{k-2}^2$	yes	$y_{k-2}x_{k-2}^2$	yes
S_5	$y_{k-1}x_{k-1}$	yes	$y_{k-1}x_{k-1}$	yes
	y_{k-2}	yes	y_{k-4}	no
	x_{k-2}^2	yes	x_{k-2}^2	yes
	$y_{k-2}x_{k-2}^2$	yes	$y_{k-2}x_{k-2}^2$	yes
S_6	y_{k-2}	yes	y_{k-2}	yes
	x_{k-1}	yes	x_{k-1}	yes
	$y_{k-2}x_{k-2}$	yes	$y_{k-2}x_{k-1}$	yes

3.2.5 Meta-MSS vs RJMCMC [1]

S_4 is taken from [1]. Again the maximum lag for the input and output are $n_y = n_u = 4$ and the nonlinear degree is $\ell = 3$. In their work, the authors executed the algorithm 10 times on the same input-output data. The RJMCMC was able to select the true model structure 7 times out of the 10 runs. On the other hand, the Meta-MSS can get the true model in all runs of the algorithm. The results are summarized in Table 3.6. Besides, there are main drawbacks related to RJMCMC method which are overcome by the Meta-MSS: the former is computationally expensive and required an execution with $30{,}000$ iterations. Furthermore, it assumes different probability distributions which are chosen to ease the computations for the parameters involved in the procedure.

Table 3.6: Comparative analysis - Meta-MSS vs RJMCMC

	Meta-MSS		RJMCMC				
	Model	Correct	Model 1 (7×)	Model 2	Model 3	Model 4	Correct
S_4	$y_{k-1}x_{k-1}$	yes	$y_{k-1}x_{k-1}$	$y_{k-1}x_{k-1}$	$y_{k-1}x_{k-1}$	$y_{k-1}x_{k-1}$	yes
	y_{k-2}	yes	y_{k-2}	y_{k-2}	y_{k-2}	y_{k-2}	yes
	x^2_{k-2}	yes	x^2_{k-2}	x^2_{k-2}	x^2_{k-2}	x^2_{k-2}	yes
	$y_{k-2}x^2_{k-2}$	yes	$y_{k-2}x^2_{k-2}$	$y_{k-2}x^2_{k-2}$	$y_{k-2}x^2_{k-2}$	$y_{k-2}x^2_{k-2}$	yes
	-	-	-	$y_{k-3}x_{k-3}$	x^2_{k-4}	$x_{k-1}x^2_{k-3}$	no

3.2.6 Meta-MSS vs Meta-heuristic method proposed by [2] vs FROLS

This section compares the identification of three experimental systems. First, the identification of an Electrical Heater is performed, which data was originally collected by [96]. The reader is referred to that thesis for further details. The identification procedure follows the parameters defined in [2, 96] and, therefore, the maximum lag for the input and output are $n_y = n_u = 2$ and the nonlinear degree is $\ell = 3$. The parameters of the Meta-MSS are set according to Table 3.1. The Mean Square Error (MSE) is used to compare the goodness of fitness of each identification procedure.

From the results Table 3.7, it is clear the significant gain for using the Meta-MSS algorithm which was able to get a model with only 3 terms and nonlinearity degree $\ell = 2$, while the method proposed in [2] generate models with 7, 8 and 10 terms, and the FROLS returned a model with 15 regressors. Due to the stochastic nature of the method proposed in [2], it is worth mentioning that the mean number of regressors of the models obtained was 13, 8 and 11 in 5 runs of the algorithm, while the Meta-MSS selected models with 3 terms in all 300 runs. The MSE values indicate that all the models are overfitting except for the one produced using Meta-MSS. There is a high number of parameters, and the MSE values are, however, higher than the model using Meta-MSS.

Regarding the method proposed by [2], the overfitting can be an effect of the stopping criteria, since it uses predefined maximum value for AIC and the number of parameters, aside from a maximum number of iterations. As it has already been mentioned, in some cases, the AIC is not recommended for nonlinear model identification because it not handle well the relationship between the model complexity and its accuracy. Also, assuming maximum values for both AIC and the number of terms as stopping criteria may lead the algorithm to converge quickly and return a local optimum instead of the global one. In this respect, the use of Meta-MSS contribute

for a better exploration and exploitation.

Concerning the FROLS performance, the algorithm fails to evaluate the importance of each regressor globally due to its greedy scheme. The ERR heavily depends on the sequence of the included regressors resulting in AIC values that do not reveal the individual contributions of each regressor. All of these drawbacks are surpassed by Meta-MSS because it tests several combinations of models and checks the significance of the regressors in each candidate solution, using this information to penalize overfitted models.

Table 3.7: Comparative analysis - Meta-MSS vs Meta-heuristic method proposed by [2] vs FROLS

Meta-MSS	Severino			FROLS
	GA	PSO	BA	
y_{k-1}	y_{k-1}	y_{k-1}	y_{k-1}	y_{k-1}
y_{k-2}	y_{k-2}	y_{k-2}	y_{k-2}	y_{k-2}
x^2_{k-1}	x_{k-2}	x_{k-1}	x_{k-1}	x^2_{k-1}
	x^2_{k-1}	y^2_{k-1}	x_{k-2}	$x_{k-2}x_{k-1}$
	y^3_{k-2}	$y_{k-2}x_{k-2}$	$y_{k-1}x_{k-2}$	$y_{k-2}x_{k-1}$
	y^2_{k-2}	x^2_{k-1}	y^2_{k-1}	y^3_{k-2}
	x^2_{k-1}	$x_{k-1}x_{k-2}$	x^2_{k-1}	y^2_{k-1}
		y^3_{k-2}	y^2_{k-2}	$y_{k-1}x_{k-2}$
			$y_{k-2}x_{k-1}$	$y^2_{k-1}x_{k-1}$
			$y^2_{k-1}x_{k-1}$	$y^2_{k-1}x_{k-2}$
				y^3_{k-1}
				u^3_{k-1}
				$y_{k-1}y_{k-2}x_{k-1}$
				$y^2_{k-2}x_{k-2}$
				$y_{k-1}x_{k-1}$
MSE	7.5627×10^{-4}	1.1084×10^{-3}	1.0979×10^{-3}	1.1212×10^{-3}	2.0502×10^{-2}

3.2.7 Full-scale F-16 aircraft

The F-16 Ground Vibration Test has been used as a benchmark for system identification. The case exhibits a clearance and friction nonlinearities at the mounting interface of the payloads. The empirical data were acquired on a full-scale F-16 aircraft on a Siemens LMS Ground Vibration Testing Master Class as well as a detailed formulation of the identification problem is available at Nonlinear System Identification Benchmarks [1]. In the experiment [110], in order to simulate the properties of real F-16 devices, two dummy payloads were mounted at the wing tips of the aircraft (Figure 3.5(3)). The input signals were applied using the shaker attached at the right wing (see Figure 3.5(1)). The signals were collected using 145 acceleration sensors

[1] Available at http://www.nonlinearbenchmark.org/

placed on the aircraft structure. The main sources of the nonlinearity are the mounting interfaces of the two payloads, where T-shaped connecting elements on the payload side slid through a rail attached to the wing side (seeFigure 3.5(2)).

Several datasets are available concerning different input signals and frequencies. This work considers the data recorded under multisine excitations with a full frequency grid from 2 to $15Hz$. According to [110], at each force level, 9 periods were acquired considering a single realization of the input signal. There are 8192 samples per period. Note that transients are present in the first period of measurement. Seven excitation levels are considered, starting from linear data at $12.4NRMS$. In addition, three nonlinear estimation data sets (number 3, 5 and 7 at 36.8, 73.6 and $97.8NRMS$, respectively) are accompanied by their corresponding test sets (numbers 2, 4 and 6 at 24.6, 61.4 and $85.7NRMS$, respectively).

This case study represents a significant challenge because it involves nonparametric analysis of the data, linearized modeling, and damping ratios versus the excitation level and nonlinear modeling around a single mode. Also, the order of the system is reasonably high. In the $2-15Hz$ band, the F-16 possesses about 10 resonance modes.

Table 3.8: Parameters used in Meta-Structure Selection Algorithm for the F-16 benchmark

Parameters	n_u	n_{u2}	n_y	ℓ	p-value	max_iter	n_agents	α	G_0
Values	20	20	20	1	0.05	30	15	23	100

The Meta-MSS algorithm and the FROLS are used to select models to represent the dynamics of the F-16 aircraft described above. The data was divided in 50% for training and 50% for validation. In the first approach, the maximum nonlinearity degree and the lag of inputs and output were set to 2 and 10, respectively. In this case, the Meta-MSS select a model with 15 terms, but the model selected through FROLS diverged. Thus, we set the maximum lag to 2. The Meta-MSS has chosen 3 regressors to form the model, while the FROLS failed again to build an adequate model. Finally, the maximum lag was set to 20 and the maximum nonlinearity degree was defined to be 1. For the latter case, the Meta-MSS parameters are defined as listed in Table 3.8. The same input and output lags are considered on FROLS approach. Table 3.9 details the results considering the second acceleration signals as output. For this case, following the recommendation in [110], the models are evaluated using the metric e_{rms_t}, which is defined

as:

$$e_{rms_t} = \sqrt{\frac{1}{N}\sum_{k=1}^{N}(y_k - \hat{y}_k)^2}. \tag{3.18}$$

Table 3.9: Identified NARX model using Algorithm 6 and FROLS.

	Meta-MSS		Meta-MSS | $\ell = n_{x_1} = n_{x_2} = 2$		Meta-MSS | $\ell = 2$, $n_{x_1} = n_{x_2} = 10$		FROLS	
	Model term	Parameter	Model term	Parameter	Model term	Parameter	Model term	Parameter
	y_{k-1}	0.7166	y_{k-2}	0.6481	y_{k-1}	1.3442	y_{k-1}	1.7829
	y_{k-5}	0.2389	$x_{_1}k-1$	1.5361	y_{k-2}	-0.8141	y_{k-2}	-1.8167
	y_{k-8}	-0.0716	$x_{_2}k-2$	1.3857	y_{k-4}	0.3592	y_{k-3}	1.3812
	y_{k-13}	-0.0867			$x_{_1}k-6$	14.8635	y_{k-6}	1.5213
	$x_{_1}k-2$	1.5992			$x_{_1}k-7$	-14.7748	y_{k-9}	0.3625
	$x_{_1}k-13$	-1.1414			$x_{_2}k-1$	-3.2129	$x_{_2}k-7$	-2.4253
	$x_{_2}k-4$	2.2248			$x_{_2}k-3$	7.1903	$x_{_1}k-1$	1.8534
	$x_{_2}k-8$	-0.8383			$x_{_2}k-8$	-4.0374	$x_{_2}k-3$	1.9866
	$x_{_2}k-13$	-1.1189					$x_{_1}k-8$	-1.5305
							$x_{_2}k-1$	0.6547
							y_{k-7}	1.2767
							y_{k-5}	1.3378
							y_{k-10}	-0.3234
							y_{k-4}	-1.0199
							y_{k-8}	-0.7116
							y_{k-12}	-0.2222
							y_{k-11}	0.3761
							$x_{_1}k-20$	0.0245
e_{rms_t}	0.0862		0.1268		0.0982		0.0876	
Elapsed time	27.92s		16.78		207.13		18.01s	

As highlighted in Table 3.9, the Meta-MSS algorithm returns a model with 9 regressors and a better performance than the model with 18 terms built using FROLS. The step-by-step procedure used by FROLS results in the selection of the first 12 output terms, while only 4 output regressors are selected using Meta-MSS. As said before, the Meta-MSS tests several different models and are updated based on the most relevant terms of each tested combination. This scheme leads to a better evaluation of the individual importance of the regressors, producing more robust models. From Table 3.9, one can see that the Meta-MSS algorithm have an affordable computational cost, since the time to select the model is very acceptable, even when comparing with FROLS, which is known to be one of the most efficient methods for structure selection.

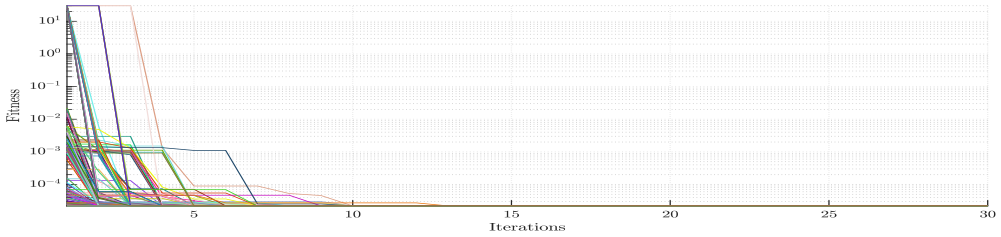
Further, it is interesting to note that the Meta-MSS returned a linear model even when the tests were performed using the maximum nonlinearity degree $\ell = 2$. This result demonstrates the excellent performance of the method since the classical one was not able to reach a satisfactory result. Figure 3.6 depicts the free run simulation of each model.

3.3 Conclusions for Mono-objective framework for regression and classification

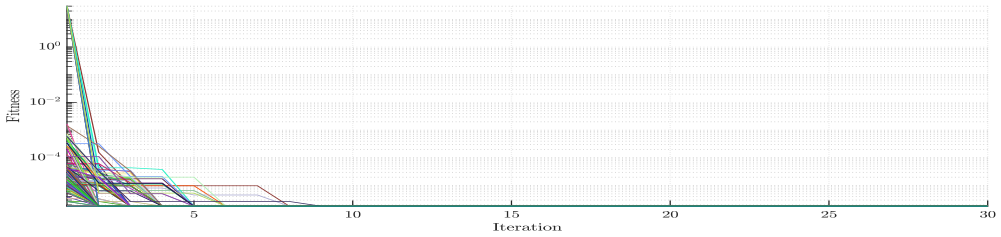
In this chapter, we proposed a new model structure selection procedure for polynomial NARX models, called Meta-MSS, that recover the correct model structure in all case-studies used for comparison against state-of-the-art algorithms. Also, we selected experiments to answer the following questions:

- How sensitive are the Meta-MSS algorithm in terms of its hyper-parameters? Does maximum iteration and number of agents improve the performance?
- How accurately does Meta-MSS select the correct model structure under different types of input signals distributions and noise? Does Meta-MSS outperform state-of-the-art techniques?
- Does Meta-MSS build parsimonious models for challenging real-world problems?

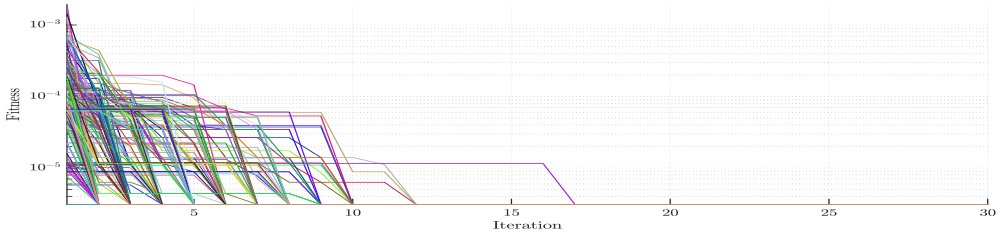
In the next chapters, we will extend the algorithm to multi-objective model structure selection, hysteresis identification and binomial classification.



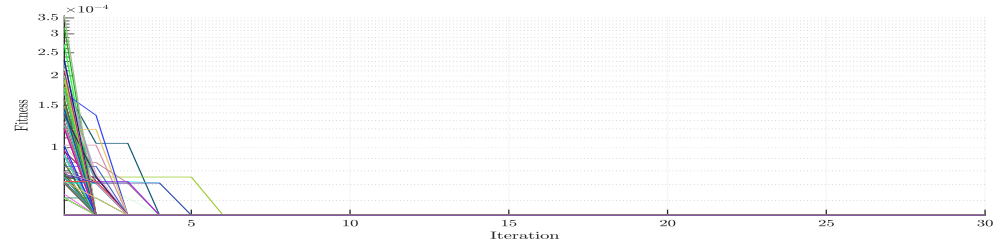
(1) System S_1.



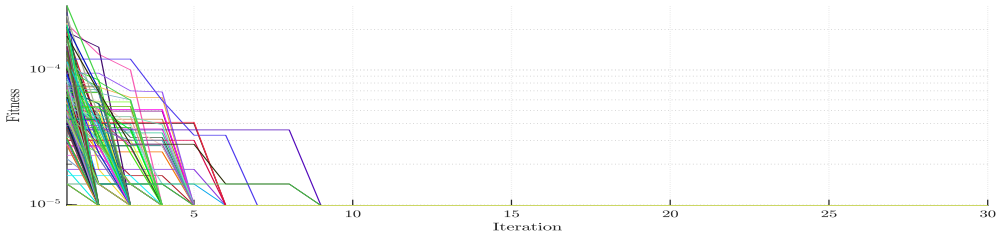
(2) System S_2.



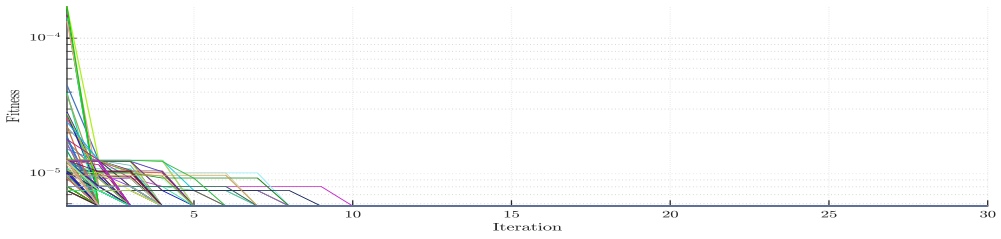
(3) System S_3.



(4) System S_4.

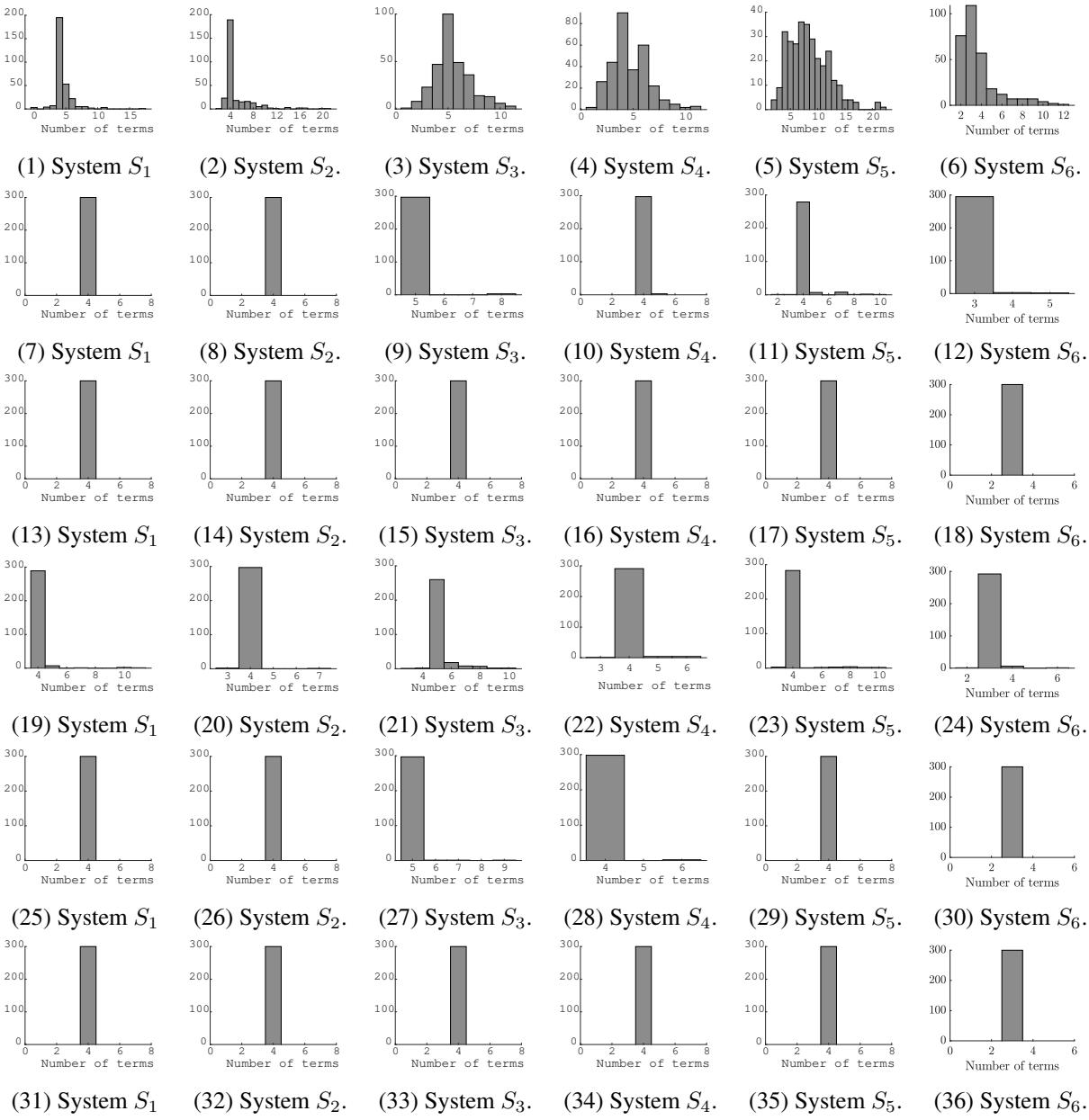


(5) System S_5.



(6) System S_6.

Figure 3.2: The convergence of each execution of Meta-Structure Selection Algorithm.



(1) System S_1 (2) System S_2. (3) System S_3. (4) System S_4. (5) System S_5. (6) System S_6.

(7) System S_1 (8) System S_2. (9) System S_3. (10) System S_4. (11) System S_5. (12) System S_6.

(13) System S_1 (14) System S_2. (15) System S_3. (16) System S_4. (17) System S_5. (18) System S_6.

(19) System S_1 (20) System S_2. (21) System S_3. (22) System S_4. (23) System S_5. (24) System S_6.

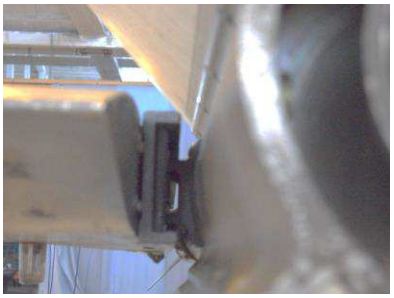
(25) System S_1 (26) System S_2. (27) System S_3. (28) System S_4. (29) System S_5. (30) System S_6.

(31) System S_1 (32) System S_2. (33) System S_3. (34) System S_4. (35) System S_5. (36) System S_6.

Figure 3.3: The distribution of terms number selected for each simulated models concerning the variation of the max_iter and n_agents.

Figure 3.4: Complete aircraft structure. Source: [3]

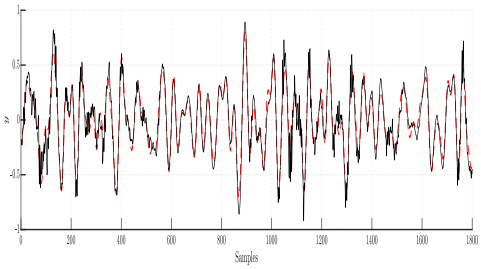
(1) Shaker.



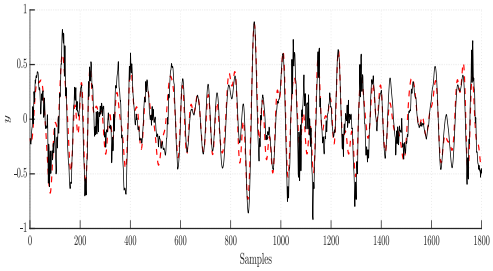
(2) Connection.

(3) Payload.

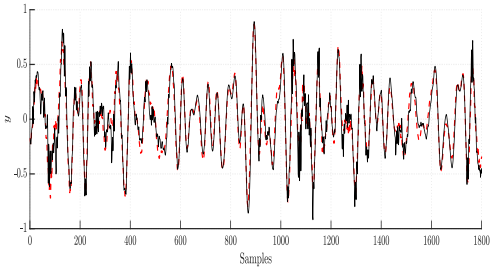
Figure 3.5: Frequency of the eye state for training and testing dataset.



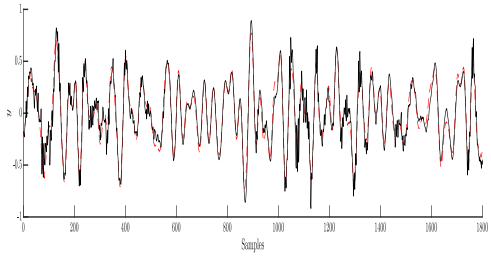
(1) Meta-MSS: $\ell = 1$, $ny = n_{x_1} = n_{x_2} = 20$.



(2) Meta-MSS: $\ell = 2$, $n_y = n_{x_1} = n_{x_2} = 2$.



(3) Meta-MSS: $\ell = 2$, $n_y = n_{x_1} = n_{x_2} = 10$.



(4) : $\ell = 1$, $ny = n_{x_1} = n_{x_2} = 20$.

Figure 3.6: Models obtained using the Meta-MSS and the FROLS algorithm. The FROLS was only capable to return a stable model when setting $\ell = 1$. The Meta-MSS, otherwise, returned satisfactory models in all cases.

CHAPTER 4

MULTI-OBJECTIVE IDENTIFICATION USING THE META-MSS ALGORITHM

Mono-objective system identification based only in dynamic data has attracted much more attention than multi-objective system identification, even though it is not always possible to acquire experimental data carefully enough to ensure the accuracy of the procedure. In this understanding, it is vital to develop methods that include other types of information besides the dynamic one in the identification design. This chapter introduces the use of the Meta-MSS algorithm considering affine information, called MoMeta-MSS. The focus of this chapter will be to illustrate the performance of multi-objective NARX system identification using a series of case study examples. The real potential of the Meta-MSS technique is analyzed considering its performance on data from complex systems. The case studies have been deliberately selected from commonly studies traditionally used as a benchmark for multi-objective system identification. All examples of this chapter use the system steady state behavior as additional information.

4.1 Description of a NARX model static behavior

The concept of Definition 2.2.1 grants obtaining the steady-state behavior of an asymptotically stable system. The steady-state form of the NARX model is defined as[60]:

$$\overline{y} = \Sigma_0 + \Sigma_y \overline{y} + \Sigma_x \overline{x} + \sum_{m=1}^{\ell-1} \sum_{p=1}^{\ell-m} \Sigma_{y^p x^m} \overline{y}^p \overline{x}^m + \sum_{p=2}^{l} \Sigma_{y^p} \overline{y}^p + \sum_{m=2}^{\ell} \Sigma_{x^m} \overline{x}^m \tag{4.1}$$

From Equation (4.1), the static gain, $\hat{K}$, can be estimated as

$$\hat{K}(\overline{y}, \overline{x}) = \frac{\overline{y}}{\overline{x}} = \frac{\frac{\Sigma_0}{\overline{x}} + \Sigma_x + \sum_{m=2}^{\ell} \Sigma_{x^m} \overline{x}^{m-1}}{1 - \Sigma_y - \sum_{m=1}^{\ell-1} \sum_{p=1}^{\ell-m} \Sigma_{y^p x^m} \overline{y}^{p-1} \overline{x}^m - \sum_{p=2}^{\ell} \Sigma_{y^p} \overline{y}^{p-1}} \tag{4.2}$$

so that one can write the static output, $\hat{y}_s$, as

$$\hat{y}_s = \frac{\Sigma_0 + \Sigma_x \overline{x} + \sum\limits_{m=2}^{\ell} \Sigma_{x^m} \overline{x}^m}{1 - \Sigma_y - \sum\limits_{m=1}^{\ell-1} \sum\limits_{p=1}^{\ell-m} \Sigma_{y^p x^m} \overline{y}^{p-1} \overline{x}^m - \sum\limits_{p=2}^{\ell} \Sigma_{y^p} \overline{y}^{p-1}}. \tag{4.3}$$

Therefore, for a general polynomial NARX model, the following expression holds:

$$\hat{y}_{si} = f(\overline{y}_i, \overline{x}_i) = q_i^\top S\Theta, \quad i = 1, 2, \ldots, n_{sf}, \tag{4.4}$$

where n_{sf} is the number of steady state samples and

$$q_i^\top = \begin{bmatrix} 1 & \overline{y}_i, \overline{y}_i^2 \ldots \overline{y}_i^\ell & \overline{x}_i^2 \ldots \overline{x}_i^\ell & F_{yu}, \end{bmatrix} \tag{4.5}$$

where F_{yu} represent the nonlinear monomials in the model.

Finally,

$$S\Theta = \begin{bmatrix} \Sigma_0 & \Sigma_y, \Sigma_{y^2}, \ldots, \Sigma_{y^\ell} & \Sigma_x, \Sigma_{x^2}, \ldots, \Sigma_{x^\ell} & F_\Sigma \end{bmatrix}^\top, \tag{4.6}$$

where F_Σ stands for all the cluster coefficients corresponding to all the term clusters and S is a constant matrix of ones and zeros.

Following Theorem 2.4.1, the cost function above need to be minimized to build a model capable of representing dynamic and static behavior:

$$\mathrm{J}_{sf}(\hat{\Theta}) = (\overline{y} - \hat{\overline{y}})^\top (\overline{y} - \hat{\overline{y}}) = (\overline{y} - QS\hat{\Theta})^\top ((\overline{y} - QS\hat{\Theta})), \tag{4.7}$$

where $Q = [q_i, \ldots, q_{n_{sf}}]$ and $[\overline{y}, QR]$ is an affine information pair. Hence,

$$\hat{\Theta} = \left[w_1 \Psi^\top \Psi + w2 (QS)^\top (QS) \right]^{-1} \left[w_1 \Psi^\top y + w_2 (QS)^\top \overline{y} \right] \tag{4.8}$$

and the Pareto set is obtained varying w_1 and w_2 such that $w_1 + w_2 = 1$.

4.2 MoMeta-MSS algorithm

The standard Meta-MSS algorithm selects the most parsimonious model through analyses of several candidate solutions in each iteration. In this sense, it generates several Pareto sets related to each tested model. Consequently, there is a need to choose a single *best* solution

from the Pareto set. [60] proposed a method based on the minimum Euclidean norm of the normalized objectives. According to that work, the objectives are normalized as

$$\overline{\mathrm{J}}_i(\hat{\Theta}^*) = \frac{\overline{\mathrm{J}}_i(\hat{\Theta}^*) - \min(\overline{\mathrm{J}}_i(\hat{\Theta}^*))}{\max(\overline{\mathrm{J}}_i(\hat{\Theta}^*) - \min(\overline{\mathrm{J}}_i(\hat{\Theta}^*))}, \quad \text{for } i = 1, \ldots, n, \tag{4.9}$$

where $\min(\cdot)$ and $\max(\cdot)$ are, respectively, the the minimum and the maximum value of $\overline{\mathrm{J}}_i(\hat{\Theta}^*)$. In this respect, let $\hat{\Theta}^* = [\hat{\Theta}_1^* \ldots \hat{\Theta}_m^*]$. Then, the optimal solution p^* are [60]:

$$p^* = \mathrm{J}(\hat{\Theta}_k^*) | \underbrace{\min}_{k \in 1,\ldots,m} \left\| \overline{\mathrm{J}}_1(\hat{\Theta}_k^*) \ldots \overline{\mathrm{J}}_n(\hat{\Theta}_k^*) \right\|. \tag{4.10}$$

Once the *best* model is obtained from Pareto Set, the goodness of fitness of each objective is considered in the adapted objective function, written as

$$\mathcal{F} = \left(\frac{\sqrt{\sum_{k=1}^{n} (y_k - \hat{y}_k)^2}}{\sqrt{\sum_{k=1}^{n} (y_k - \bar{y})^2}} + \frac{\sqrt{\sum_{k=1}^{r} (\overline{y}_{_{sf}k} - \hat{\overline{y}}_{_{sf}k})^2}}{\sqrt{\sum_{k=1}^{r} (\overline{y}_{_{sf}k} - \bar{y})^2}} \right) \times \varrho. \tag{4.11}$$

where y and $\overline{y}_{_{sf}} \in \mathbb{R}$ are the dynamic and static output of the system, respectively; $\hat{y}$ and $\hat{\overline{y}}_{_{sf}} \in \mathbb{R}$ are the predicted output of the dynamic and static behavior and $\overline{y} \in \mathbb{R}$ the mean of the measured output y.

Hence, although the Pareto set is obtained based on the use of affine information in parameter estimation phase, the model structure is also affected due to the penalty function, which evaluates each regressor individually. Because the Meta-MSS is a stochastic algorithm, 100 runs are performed, and the model which has the lowest value for the fitness function simultaneously is chosen as the best one. Finally, the method is summarized in Algorithm 7.

4.3 Electrical Heater

The same data described in Section 3.2.6 are used here as a case study. Figure 4.1 shows the static behavior of the system. The authors acquired the steady-state data in an experiment that lasted 3h hours [96]. In order to generate the Pareto set, this study applies the weighted sum method for multi-objective optimization.

The weight w_1 is a logarithmically spaced vector between $[0\ 1]$ and $w_2 = 1 - w_1$. The

Algorithm 7: Multi-objective Meta-structure selection algorithm

Result: Model which has the best fitness value

Input: $\{(u_k),(y_k),k=1,\ldots,N\}$, $\{(\overline{u}_k),(\overline{y}_k),k=1,\ldots,N\}$ $\mathcal{M}=\{\psi_j,j=1,\ldots,m\}$, n_y, n_u, ℓ, max_iteration, noV, np

1 $\mathcal{P} \leftarrow$ Build initial population of random agents in the search space, $\mathcal{S}$

2 $v \leftarrow$ set the agent's velocity equal zero at first iteration

3 $\Psi \leftarrow$ Build the general matrix of regressors based on n_y, n_u and ℓ ▷ Equation (2.28)

4 **repeat**

5 **for** $i=1:d$ **do**

6 $\mathrm{m_i} \leftarrow \vec{x}_{np,i}$ ▷ Extract the model encoding from population

7 $\Psi_r \leftarrow \Psi(\mathrm{m_i})$ ▷ Delete the Ψ columns where $\mathrm{m_i}=0$ Example 3.1.1

8 $\mathrm{J}_{sf}(\hat{\Theta})=(\overline{y}-\hat{\overline{y}})^\top(\overline{y}-\hat{\overline{y}})=(\overline{y}-QS\hat{\Theta})^\top((\overline{y}-QS\hat{\Theta}))$ ▷ Equation (4.7)

9 $p^* \leftarrow$ Select the optimal solution ▷ Equation (4.10)

10 $\hat{y} \leftarrow$ Dynamic output of the best model selected

11 $\overline{y} \leftarrow$ Static output of the best model selected

12 $V \leftarrow (\Psi^\top\Psi)^{-1}$

13 $\hat{\sigma}_e^2=\frac{1}{N-m}\sum_{k=1}^{N}(y_k-\psi_{k-1}^\top\hat{\Theta})$ ▷ Equation (3.4)

14 **for** $h=1:\tau$ **do**

15 $\hat{\sigma}^2 \leftarrow \hat{\sigma}_e^2 V_{h,h}$ ▷ Equation (3.3)

16 $\mathrm{se}(\hat{\Theta}_j) \leftarrow \sqrt{\hat{\sigma}_{h,h}^2}$ ▷ Equation (3.5)

17 $T_0 \leftarrow \frac{\hat{\Theta}_j}{\mathrm{se}(\hat{\Theta})}$ ▷ Equation (3.6)

18 $p \leftarrow$ regressors where $-t_{\alpha/2,N-m} < T_0 < t_{\alpha/2,N-m}$ ▷ Equation (3.7)

19 **end**

20 Remove the p regressors from Ψ_r

21 Check for empty model

22 **if** *Model is empty* **then**

23 Generate a new population

24 Repeat the steps from line 6 to 18

25 **end**

26 Repeat from line 12 to 25 using static data and the information matrix QS

27 $n_1 \leftarrow$ size(p) ▷ Number of redundant terms considering both dynamic and static evaluation

28 $\mathrm{J}_{sf}(\hat{\Theta})=(\overline{y}-\hat{\overline{y}})^\top(\overline{y}-\hat{\overline{y}})=(\overline{y}-QS\hat{\Theta})^\top((\overline{y}-QS\hat{\Theta}))$ ▷ Re-estimation | Equation (4.7)

29 $p^* \leftarrow$ Select the optimal solution ▷ Equation (4.10)

30 $\hat{y} \leftarrow$ Dynamic output of the best model selected

31 $\overline{y} \leftarrow$ Static output of the best model selected

32 $\mathcal{F}_i \leftarrow \mathcal{F}=\left(\frac{\sqrt{\sum_{k=1}^{n}(y_k-\hat{y}_k)^2}}{\sqrt{\sum_{k=1}^{n}(y_k-\overline{y})^2}}+\frac{\sqrt{\sum_{k=1}^{r}(\overline{y}_{sf\,k}-\hat{\overline{y}}_{sf\,k})^2}}{\sqrt{\sum_{k=1}^{r}(\overline{y}_{sf\,k}-\overline{y})^2}}\right)\times\varrho$ ▷ Equation (4.11)

33 $\mathcal{P}_i^n \leftarrow$ Encoded Ψ_r

34 Evaluate the fitness for each agent, $\mathcal{F}_i(t)$

35 **end**

36 $\mathcal{P} \leftarrow \mathcal{P}^n$ ▷ Update the population

37 Execute the lines $31-40$ of algorithm 6

38 **until** *max_iterations is reached*

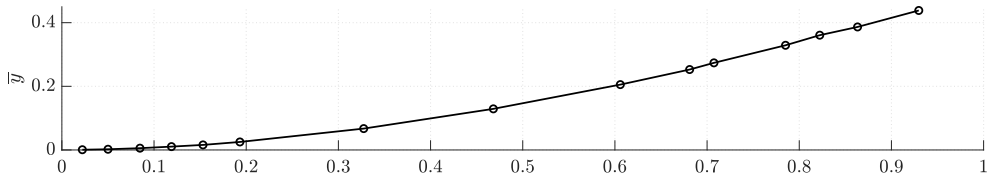


Figure 4.1: The steady state behavior of the electrical heater. The continuous line is to facilitate the visualization of its nonlinear characteristic.

number of weight points is set to 10. The data was divided in 50% for the training phase and 50% for validation. The parameters of the MoMeta-MSS are detailed in Table 4.1. The number of iterations and the number of search agents are increased to allow a complete exploration and exploitation aiming for a model selection capable of reproducing the dynamic and static behavior of the system. After 30 runs of the MoMeta-MSS algorithm, the most voted model structure, representing 26.66%, has four regressors (Table 4.2). Interestingly, the second and third most voted model structures are composed by four terms and differ from each other only with respect to the input term lag. In other words, they belongs to the same term cluster. All of the models which regressors belongs to the same cluster combined represents 73.33% of the models obtained.

Table 4.1: Parameters used in Multi-objective Meta-Structure Selection Algorithm

Parameters	n_u	n_y	ℓ	p-value	max_iter	n_agents	α	G_0
Values	2	2	3	0.05	50	15	23	100

Table 4.2: Identified NARX model using Algorithm 7. The blue row of parameters shown in this table are the ones of the optimal model obtained using the minimum Euclidean norm of the normalized objectives.

Model terms	y_{k-1}	x^2_{k-1}	x^2_{k-2}	x^3_{k-2}
parameters ($w_1 = 0.0010$, $w_2 = 0.9990$)	0.8608	0.0493	0.0428	−0.0230
parameters ($w_1 = 0.0021$, $w_2 = 0.9978$)	0.8602	0.0494	0.0423	−0.0222
parameters ($w_1 = 0.0046$, $w_2 = 0.9953$)	0.8599	0.0494	0.0412	−0.0207
parameters ($w_1 = 0.0100$, $w_2 = 0.9900$)	0.8596	0.0494	0.0393	−0.0184
parameters ($w_1 = 0.0215$, $w_2 = 0.9784$)	0.8593	0.0494	0.0371	−0.0157
parameters ($w_1 = 0.0464$, $w_2 = 0.9535$)	0.8591	0.0494	0.0351	−0.0133
parameters ($w_1 = 0.1000$, $w_2 = 0.9000$)	0.8589	0.0494	0.0337	−0.0116
parameters ($w_1 = 0.2154$, $w_2 = 0.7845$)	0.8588	0.0494	0.0329	−0.0107
parameters ($w_1 = 0.4641$, $w_2 = 0.5358$)	0.8587	0.0494	0.0326	−0.0102
parameters ($w_1 = 1.0000$, $w_2 = 0.0000$)	0.8587	0.0494	0.0324	−0.0100

Another interesting detail concerns the small variation of the parameters obtained considering the wide variation of the weights w_1 and w_2. This slight variation may be strong a indication of correct model structure. This is justified since regression vectors generates the vector space

that, combined, represents the output of the system.

Figure 4.2 details the Pareto Set obtained by incorporating the affine information in the selection procedure, according to Algorithm 7. The free-run-simulations of the dynamic and static models considering each Pareto point are presented in Figure 4.3. Finally, the RMSE values are reported in Table 4.3.

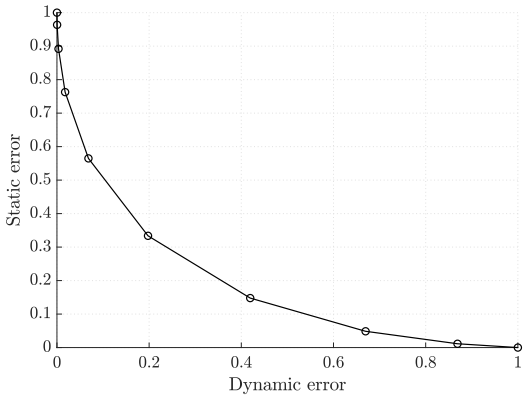
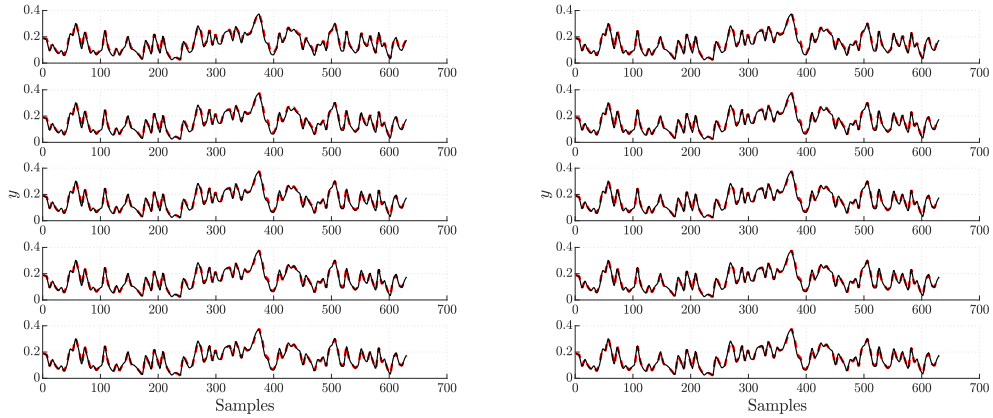


Figure 4.2: The Pareto Set obtained by incorporating the affine information in the selection procedure. Each (○) represents a model the model with specific parameters.

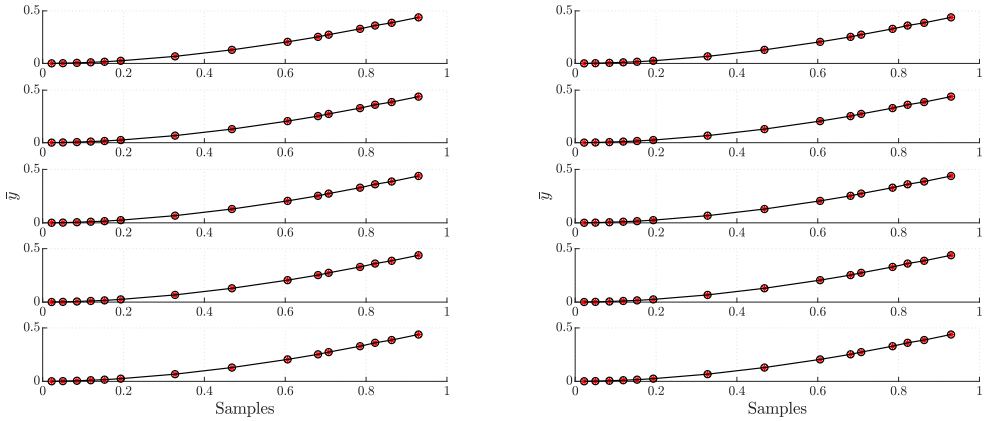
Table 4.3: The RMSE values obtained for each parameter set. The blue row corresponds to the optimal model obtained using the minimum Euclidean norm of the normalized objectives.

	RMSE (dynamic model)	RMSE (steady-state model)
parameters ($w_1 = 0.0010$, $w_2 = 0.9990$)	0.1144	0.0015
parameters ($w_1 = 0.0021$, $w_2 = 0.9978$)	0.1102	0.0016
parameters ($w_1 = 0.0046$, $w_2 = 0.9953$)	0.1041	0.0018
parameters ($w_1 = 0.0100$, $w_2 = 0.9900$)	0.0964	0.0024
parameters ($w_1 = 0.0215$, $w_2 = 0.9784$)	0.0892	0.0032
parameters ($w_1 = 0.0464$, $w_2 = 0.9535$)	0.0850	0.0040
parameters ($w_1 = 0.1000$, $w_2 = 0.9000$)	0.0834	0.0045
parameters ($w_1 = 0.2154$, $w_2 = 0.7845$)	0.0831	0.0049
parameters ($w_1 = 0.4641$, $w_2 = 0.5358$)	0.0831	0.0050
parameters ($w_1 = 1.0000$, $w_2 = 0.0000$)	0.0831	0.0051

One can ask how the results obtained using the Meta-MSS compare to those using other techniques. In this respect, Table 4.4 compares the models returned via Meta-MSS and FROLS. It is important to note that the model used for comparison are NARMAX models. More details about the model selection procedure can be found in [65]. The proposed technique built a model with fewer regressors, but it has a term with nonlinearity degree equal 3. The RMSE of the dynamic model is higher than the FROLS model, while the RMSE of the static model is slightly smaller on the model using Meta-MSS. Also, the Meta-MSS generates a model with



(1) Validation of the dynamic NARX model. (−)Experimental analyzed system, (−−) NARX model.



(2) Validation of the steady-state NARX model. (−)Experimental analyzed system, (−−) Steady-state NARX model.

Figure 4.3: Dynamic and static simulation of the Electrical Heater model.

no cross-terms between the input and output while the FROLS built a model with two terms belonging to Σ_{yx} cluster.

Table 4.4: Models obtained using the Meta-MSS and the FROLS algorithm. The Meta-MSS returned a model with a lower number of regressors, but with higher nonlinearity degree. The FROLS returned a model with two terms belonging to Σ_{yx} cluster while the one obtained using Meta-MSS has no cross terms between the input and output.

	Meta-MSSm	FROLS
	Model term	Model term
	y_{k-1}	y_{k-1}
	x_{k-1}	$x_{k-1}x_{k-2}$
	x_{k-2}^2	x_{k-1}^2
	x_{k-2}^3	y_{k-2}
		$y_{k-1}x_{k-2}$
		$y_{k-2}x_{k-2}$
		x_{k-2}^2
RMSE (dynamic model)	0.0892	0.0089
RMSE (steady state model)	0.0032	0.0034

4.4 DC-DC Buck converter

This case study presents the identification of a pilot buck dc-dc converter. For the sake of brevity, the reader is referred to [111] for more information about the dynamic data acquisition, research facility, and experimental condition. The affine information pair to be used is $[\bar{y}, QR]$. The static data is obtained theoretically from the static function defined as [44]:

$$\bar{y} = \frac{4V_d}{3} - \frac{Vd}{3}\bar{u}, \tag{4.12}$$

where $V_d = 24V$, $\bar{u}$ is the input operating in steady state. 7 static and 84 dynamic points are used for identification. Figure 4.4 shows the static behavior of the system. Again, the weighted sum method for multi-objective optimization is used to generate the Pareto set.

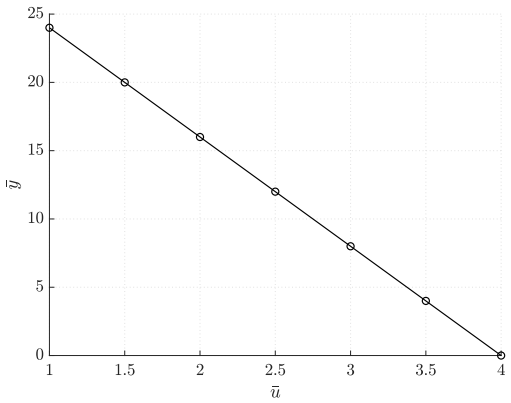


Figure 4.4: The steady state behavior of the electrical heater. The continuous line is to facilitate the visualization of its nonlinear characteristic.

The weight w_1 is a logarithmically spaced vector between $[0\ 1]$ and $w_2 = 1 - w_1$. The number of weight points is set to 10. The parameters of the Meta-MSSm are detailed in Table 4.5. Preliminaries run of the algorithm using the dataset are performed to choose the number of iterations, and the number of search agents, so the parameters set allows a more robust exploration and exploitation of the search space. 30 runs of the Meta-MSSm algorithm are performed to select the final model based on the number of appearances. The most voted model structure, representing 36.67%, has 9 terms (Table 4.6)

Table 4.5: Parameters used in Multi-objective Meta-Structure Selection Algorithm

Parameters	n_u	n_y	ℓ	p-value	max_iter	n_agents	α	G_0
Values	2	2	3	0.05	100	15	23	100

Table 4.6: Identified NARX model using Algorithm 7. The blue row of parameters shown in this table are the ones of the optimal model obtained using the minimum Euclidean norm of the normalized objectives.

Model terms	constant	y_{k-1}	y_{k-2}	x_{k-1}	x_{k-2}	$y_{k-1}x_{k-2}$	y_{k-2}^2	x_{k-1}^2	$x_{k-1}x_{k-2}$
parameters ($w_1 = 0.0010$, $w_2 = 0.9990$)	15.2110	0.9643	−0.2753	−24.8680	16.5660	0.0995	−0.0051	8.5646	−7.4399
parameters ($w_1 = 0.0021$, $w_2 = 0.9978$)	15.2110	0.9641	−0.2736	−24.8370	16.5337	0.0992	−0.0051	8.5511	−7.4260
parameters ($w_1 = 0.0046$, $w_2 = 0.9953$)	15.2110	0.9638	−0.2700	−24.7744	16.4684	0.0986	−0.0052	8.5236	−7.3978
parameters ($w_1 = 0.0100$, $w_2 = 0.9900$)	15.2109	0.9633	−0.2631	−24.6557	16.3445	0.0974	−0.0055	8.4714	−7.3445
parameters ($w_1 = 0.0215$, $w_2 = 0.9784$)	15.2108	0.9624	−0.2517	−24.4559	16.1360	0.0954	−0.0058	8.3837	−7.2547
parameters ($w_1 = 0.0464$, $w_2 = 0.9535$)	15.2107	0.9611	−0.2358	−24.1790	15.8472	0.0926	−0.0063	8.2621	−7.1303
parameters ($w_1 = 0.1000$, $w_2 = 0.9000$)	15.2105	0.9597	−0.2188	−23.8826	15.5383	0.0897	−0.0068	8.1320	−6.9974
parameters ($w_1 = 0.2154$, $w_2 = 0.7845$)	15.2104	0.9585	−0.2050	−23.6405	15.2867	0.0873	−0.0072	8.0260	−6.8894
parameters ($w_1 = 0.4641$, $w_2 = 0.5358$)	15.2103	0.9573	−0.1961	−23.4766	15.1195	0.0859	−0.0075	7.9554	−6.8185
parameters ($w_1 = 1.0000$, $w_2 = 0.0000$)	15.1992	−1.2570	−0.5104	5.34	1.7459	1.0257	0.0021	1.8154	−6.7335

Figure 4.5 illustrates the Pareto Set obtained using the Meta-MSSm considering the affine information pair. The free-run-simulations of the dynamic and static models considering each Pareto point are presented in Figure 4.6. Lastly, the RMSE values are reported in Table 4.7.

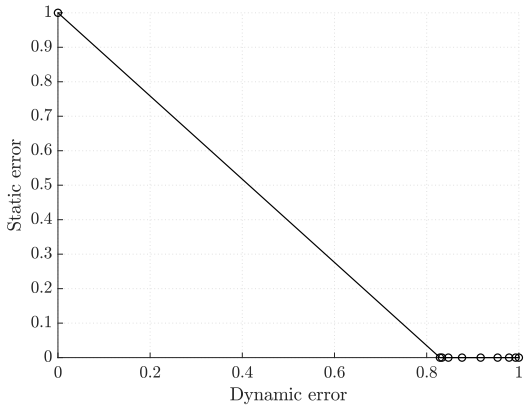
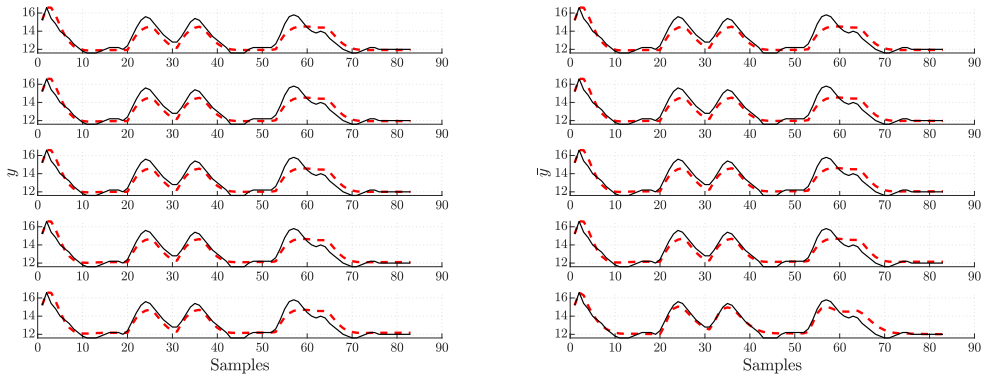


Figure 4.5: The Pareto Set obtained by incorporating the affine information in the selection procedure. Each (○) represents a model the model with specific parameters.

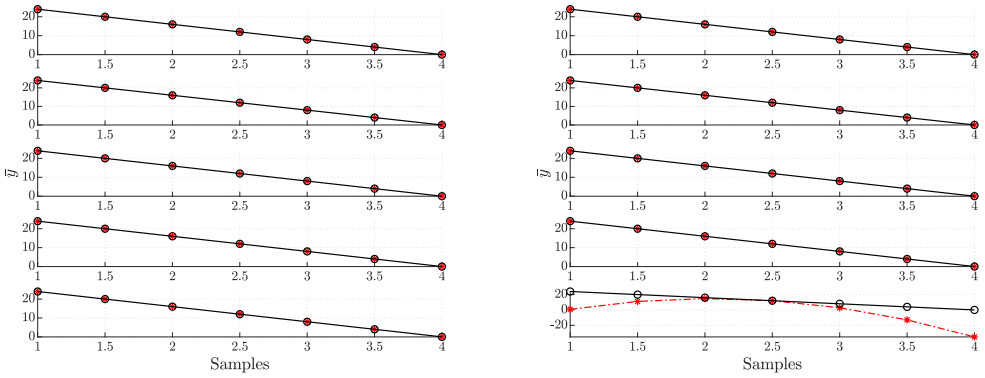
Table 4.7: The RMSE values obtained for each parameter set. The blue row corresponds to the optimal model obtained using the minimum Euclidean norm of the normalized objectives.

	RMSE (dynamic model)	RMSE (steady-state model)
parameters ($w_1 = 0.0010$, $w_2 = 0.9990$)	0.4156	8×10^{-4}
parameters ($w_1 = 0.0021$, $w_2 = 0.9978$)	0.4142	0.0001
parameters ($w_1 = 0.0046$, $w_2 = 0.9953$)	0.4115	0.0003
parameters ($w_1 = 0.0100$, $w_2 = 0.9900$)	0.4067	0.0007
parameters ($w_1 = 0.0215$, $w_2 = 0.9784$)	0.3996	0.0013
parameters ($w_1 = 0.0464$, $w_2 = 0.9535$)	0.3919	0.0021
parameters ($w_1 = 0.1000$, $w_2 = 0.9000$)	0.3861	0.0030
parameters ($w_1 = 0.2154$, $w_2 = 0.7845$)	0.3832	0.0037
parameters ($w_1 = 0.4641$, $w_2 = 0.5358$)	0.3820	0.0042
parameters ($w_1 = 1.0000$, $w_2 = 0.0000$)	0.3250	2.1992

For this case, the Meta-MSSm performance is compared against the FROLS and the MERR. The results concerning the FROLS and the MERR algorithms are taken from [44, 65], respec-



(1) Validation of the dynamic NARX model. (−)Experimental analyzed system, (− −) NARX model.



(2) Validation of the steady-state NARX model. (−)Experimental analyzed system, (− −) Steady-state NARX model.

Figure 4.6: Dynamic and static simulation of the Electrical Heater model.

tively. Again, the models obtained using the FROLS and MERR are based on NARMAX methods. From the reports on Table 4.8, it is clear that the proposed technique built the best model regarding the trade-off between the dynamic and static error. It has a better steady-state and dynamic behavior than the model obtained by [65].

Regarding the performance against the MERR [44] models, the Meta-MSSm has a slightly worse dynamic result in 3 cases, but has a significantly better static behavior in all cases. The MERR models fails to generalize the static characteristic of the system and can fit the static data only in the operation point related to the dynamic behavior. The MERR_4, for example, corresponds to the case admitting only the dynamic error in the model selection scheme, which resulted in a model with better dynamic behavior, but, as said before, massively worse in the static behavior.

Table 4.8: Models obtained using the Meta-MSSm, the FROLS and the MERR. The models have a similar number of regressors.

	Meta-MSSm	FROLS	MERR_1	MERR_2	MERR_3	MERR_4
	Model term	Model term	Model term	Model term	Model term	Model term
	constant	constant	constant	constant	constant	constant
	y_{k-1}	y_{k-1}	y_{k-1}	y_{k-1}	y_{k-1}	y_{k-1}
	y_{k-2}	y_{k-2}	y_{k-2}	y_{k-2}	y_{k-2}	y_{k-2}
	x_{k-1}	x_{k-1}	x_{k-1}	x_{k-1}	x_{k-1}	x_{k-1}
	x_{k-2}	x_{k-2}	x_{k-2}	x_{k-2}^2	x_{k-2}	x_{k-1}^2
	$y_{k-1}x_{k-2}$	$x_{k-1}x_{k-2}$	y_{k-1}^2	x_{k-1}^2	x_{k-2}^2	y_{k-1}^2
	y_{k-2}^2	x_{k-2}^2	y_{k-2}^2	$x_{k-1}x_{k-2}$	x_{k-1}^2	y_{k-2}^2
	x_{k-1}^2	x_{k-1}^2	$y_{k-1}y_{k-2}$	$y_{k-1}x_{k-1}$	y_{k-2}^2	$y_{k-1}y_{k-2}$
	$x_{k-1}x_{k-2}$		$y_{k-2}x_{k-2}$	$y_{k-2}x_{k-2}$	$y_{k-1}x_{k-1}$	$y_{k-1}x_{k-1}$
RMSE_d	0.3830	0.6580	0.4802	0.3545	0.2935	0.3004
RMSE_s	0.0042	0.0881	0.4930	1.1493	1.4959	1.5824

4.5 Conclusions for Multi-objective identification using the Meta-MSS algorithm

In this chapter we described the Meta-MSS framework for multi-objective model structure selection of polynomial NARX models. Besides to inheriting the exploration and exploitation strength of the Meta-MSS algorithm for mono-objective problems, the framework generate a single model relying on two key principles: first, we build a set of Pareto sets related to the model parameters. Second, we apply a decision criteria based on the Euclidean norm of the cost functions. We have shown that the proposed algorithm can be extended to incorporate affine information in the search for an adequate model structure and obtain models with high accuracy in both dynamical and steady-state form. In this respect, we have also shown throughout the case-studies that this methodology provides a solid alternative to traditional algorithms.

CHAPTER 5

NARX MODELS WITH MULTIPLE INPUTS FOR HYSTERETIC SYSTEM IDENTIFICATION

The memory effects between quasi-static input and output make the modeling of hysteretic systems very difficult. Physics-based models are often used to describe the hysteresis loops, but these models usually lack the simplicity and efficiency required in practical applications involving system characterization, identification, and control. As detailed in Section 2.2.2, NARX models have proven to be a feasible choice to describe the hysteresis loops. However, even considering the sufficient conditions for rate independent hysteresis representation, classical structure selection algorithms fails to return a model with decent performance and the user needs to set a multi-valued function to ensure the occurrence of the bounding structure $\mathcal{H}$ [34]. Even though some progress has been made, previous work has been limited to models with a single equilibrium point. The present chapter aims to present new prospects in the MSS of hysteretic systems regarding the cases where the models have multiple inputs and it is not restricted concerning the number of equilibrium points. First, the Meta-MSS algorithm is used to build a model for a magneto-rheological damper (MRD) considering the sufficient conditions proposed by [34]. Subsequently, we reduce the conservatism of the NARX representation of hysteresis loop through the proposal of models with multiple inputs.

5.1 A Brief description of the Bouc-Wen model of magneto-rheological damper device

The data used in this study-case is the Bouc-Wen model [112, 113] of an MRD whose schematic diagram is shown in Figure 5.1.

The general form of the Bouc-Wen model can be described as [4]:

$$\frac{dz}{dt} = g\left[x,z,sign\left(\frac{dx}{dt}\right)\right]\frac{dx}{dt}, \tag{5.1}$$

where z is the hysteretic model output, x the input and $g[\cdot]$ a nonlinear function of x, z and

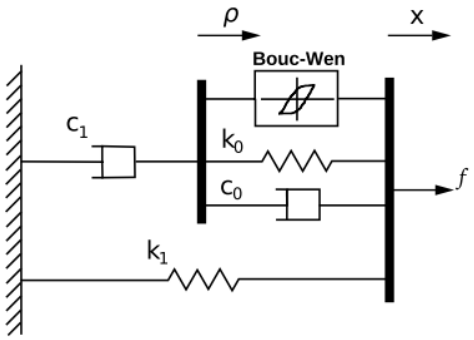


Figure 5.1: The model for a magneto-rheological damper proposed by [4].

$sign(dx/dt)$. Spencer e Sain [4] proposed the following phenomenological model for the aforementioned device:

$$
\begin{aligned}
f &= c_1\dot{\rho} + k_1(x - x_0),\\
\dot{\rho} &= \frac{1}{c_0 + c_1}[\alpha z + c_0\dot{x} + k_0(x - \rho)],\\
\dot{z} &= -\gamma|\dot{x} - \dot{\rho}|z|z|^{n-1} - \beta(\dot{x} - \dot{\rho})|z|^n + A(\dot{x} - \dot{\rho}),\\
\alpha &= \alpha_a + \alpha_b u_{bw},\\
c_1 &= c_{1a} + c_{1b} u_{bw},\\
c_0 &= c_{0a} + c_{0b} u_{bw},\\
\dot{u}_{bw} &= -\eta(u_{bw} - E).
\end{aligned} \tag{5.2}
$$

where f is the damping force, c_1 and c_0 represent the viscous coefficients, E is the input voltage, x is the displacement and $\dot{x}$ is the velocity of the model. The parameters of Equation (5.2) (see Table 5.1) were taken from [55].

Table 5.1: Parameters for a MR damper

Parameter	Value	Parameter	Value
c_{0_a}	$20.2Ns/cm$	α_a	$44.9N/cm$
c_{0_b}	$2.68Ns/cmV$	α_b	$638N/cm$
c_{1_a}	$350Ns/cm$	γ	$39.3\ cm^{-2}$
c_{1_b}	$70.7Ns/cmV$	β	$39.3\ cm^{-2}$
k_0	$15N/cm$	n	2
k_1	$5.37N/cm$	η	$251\ s^{-1}$
x_0	$0cm$	A	47.2

For this particular study, both displacement and voltage inputs, x and E, respectively, were

generated by filtering a white Gaussian noise sequence using a Blackman-Harris FIR filter with 6Hz cutoff frequency. The integration step-size was set to $h = 0.002$, following the procedures described in [34]. These procedures are for identification purposes only since the inputs of a MRD could have several different characteristics.

5.1.1 Modeling of magneto-rheological damper devices using Meta-MSS: identification of hysteresis loops considering models with a single equilibrium point.

This section develops a linear-in-the-parameters NARX model of nonlinear damping effects for a magneto-rheological damper device considering models with one equilibrium point. Besides the velocity x_2 (a derivative of the displacement) and voltage x_1, additional input was generated to satisfy the sufficient conditions for autoregressive models be able to reproduce the hysteresis behavior. The new input is defined as a multi-function of the velocity difference, $\text{sign}(x_{2_k} - x_{2_{k-1}})$.

The Meta-MSS parameters are reported in Table 5.2. The maximum lag and nonlinearity degree were defined according to [34], and therefore, a fair comparison can be made with the method proposed in the paper. A total of 30 runs of the Meta-MSS algorithm are performed to select the most voted structure as the final model.

Representing 90.00%, the final model has five regressors (Table 5.3) against the four terms model obtained by [34]. The result shows that the Meta-MSS technique can build a simple model to reproduce the behavior of the system with high accuracy (RMSE: Meta-MSS = 0.0540 | Cluster-based technique = 0.1282). One should notice the powerful performance of Meta-MSS since it is able of returning models which are in line with the sufficient conditions established in [34]. Further, since the cluster-based method demands a manual verification of each term cluster, the Meta-MSS can be considered a viable alternative to select models of hysteretic systems when considering only one equilibrium point.

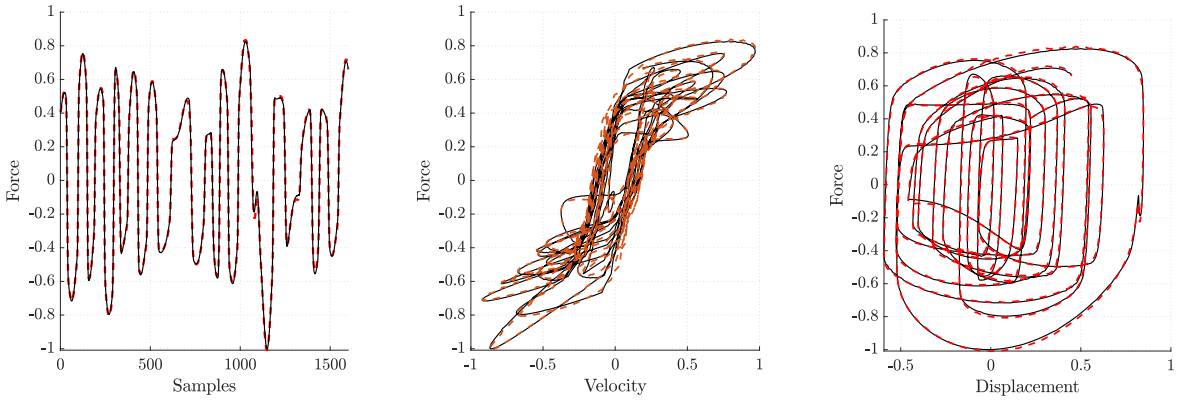
Table 5.2: Meta-Structure Selection Algorithm parameters used in the hysteretic system identification.

Parameters	n_{x_1}	n_{x_2}	n_{x_3}	n_y	ℓ	p-value	max_iter	n_agents	α	G_0
Values	1	1	1	1	3	0.05	50	15	23	100

The Figure 5.2 presents the simulation regarding the model obtained using the Meta-MSS algorithm.

Table 5.3: Models obtained using the Meta-MSS and the cluster-based structure selection algorithm.

	Meta-MSS		Cluster-based technique	
	Model term	Parameter	Model term	Parameter
	y_{k-1}	0.9084	y_{k-1}	0.8536
	$x_{2k-1}x_{1k-1}$	1.4276	$x_{2k-1}x_{1k-1}$	0.6143
	$x_{3k-1}x_{2k-1}y_{k-1}$	−1.6751	$x_{3k-1}x_{2k-1}y_{k-1}$	−0.4407
	x^3_{3k-1}	0.2822	x_{3k-1}	0.0388
	$x^2_{3k-1}x_{2k-1}$	0.2017		
RMSE	0.0540		0.1282	



(1) Validation of the dynamic NARX model. (−)Experimental analyzed system, (−−) NARX model.

(2) Validation of the dynamic NARX model. (−)Experimental analyzed system, (−−) NARX model.

(3) Validation of the dynamic NARX model. (−)Experimental analyzed system, (−−) NARX model.

Figure 5.2: Meta-MSS performance in the Bouc-Wen system identification.

5.2 Representation of RIH in NARX models with multiple inputs

This section demonstrates how polynomial NARX models with multiple inputs can be used to represent systems with hysteresis nonlinearity. By reducing the conservatism, The following proposition shows that there is no need to restrict the number of equilibrium points because the evolution of each input is the feature that leads the system to a specific stable equilibrium point.

Proposition 5.2.1. Consider the polynomial NARX model with asymptotically stable equilibrium points described by:

$$\begin{aligned} y_k =& \Sigma_0 + \sum_{i=n}^{q} c^i_{y^2} y^2_{k-i} + \sum_{i=1}^{p} c^i_y y_{k-i} + \sum_{m=1}^{r} c^m_x x_{k-m} + \sum_{i=n}^{q}\sum_{m=1}^{r} c^{nm}_{y^2x} y^2_{k-i} x_{k-m} + \\ & \sum_{i=1}^{p}\sum_{m=1}^{r} c^{im}_{yx} y_{k-i} x_{k-m} + \sum_{m_1=1}^{r}\sum_{m_2=m_1}^{r} c^{m_1 m_2}_{x^2} x_{k-m_1} x_{k-m_2} + \ldots + \\ & \sum_{m_1=1}^{r} \cdots \sum_{m_l=m_{l-1}}^{r} c^{m_1,\ldots,m_2}_{x^l} x_{k-m_1} x_{k-m_l}. \end{aligned} \tag{5.3}$$

Here we consider $\{\Sigma_0, c^n_{y^2}, c^i_y, c^m_x, c^{nm}_{y^2x}, c^{im}_{yx}, c^{m_1m_2}_{x^2} \dots c^{m_1,\dots,ml}_{x^l}\} \in \mathbb{R}$, therefore the output and input variables take values in the set of reals. If the one of the inputs is capable of changing the shape of signals containing peaks (with different height and width) drastically, then the model output will converge to the bounding structure $\mathcal{H}$ which confines a rate-independent hysteresis loops defined by the model equilibria. $\triangle$

The roots of Equation (5.3) are:

$$\bar{y}(x) = \begin{cases} \frac{-\Sigma_y - \Sigma_{y,x}\bar{x} + \sqrt{\Delta}}{2(\Sigma_{y^2} + \Sigma_{y^2,x}\bar{x})} \\ \frac{-\Sigma_y - \Sigma_{y,x}\bar{x} - \sqrt{\Delta}}{2(\Sigma_{y^2} + \Sigma_{y^2,x}\bar{x})} \end{cases} \quad \text{for } \Sigma_{y^2} + \Sigma_{y^2,x}\bar{x} \neq 0. \tag{5.4}$$

where Δ are

$$\Delta = (\Sigma_y + \Sigma_{y,x}\bar{x})^2 - 4(\Sigma_{y^2,x}\bar{x} + \Sigma_{y^2})(\Sigma_x\bar{x} + \Sigma_{x^2}\bar{x}^2 + \dots + \Sigma_{x^\ell}\bar{x}^\ell + \Sigma_0). \tag{5.5}$$

From Equation (2.14), the model equilibria considering each root of Equation (5.3) defined in Equation (5.4) are respectively given by:

$$\begin{cases} \left| 2\Sigma_{y^2,x}\bar{x}\left[\frac{-\Sigma_y - \Sigma_{y,x}\bar{x} + \sqrt{\Delta}}{2(\Sigma_{y^2} + \Sigma_{y^2,x}\bar{x})}\right] + 2\Sigma_{y^2}\left[\frac{-\Sigma_y - \Sigma_{y,x}\bar{x} + \sqrt{\Delta}}{2(\Sigma_{y^2} + \Sigma_{y^2,x}\bar{x})}\right] + \Sigma_{y,x}\bar{x} + \Sigma_y \right| < 1 \\ \left| 2\Sigma_{y^2,x}\bar{x}\left[\frac{-\Sigma_y - \Sigma_{y,x}\bar{x} - \sqrt{\Delta}}{2(\Sigma_{y^2} + \Sigma_{y^2,x}\bar{x})}\right] + 2\Sigma_{y^2}\left[\frac{-\Sigma_y - \Sigma_{y,x}\bar{x} - \sqrt{\Delta}}{2(\Sigma_{y^2} + \Sigma_{y^2,x}\bar{x})}\right] + \Sigma_{y,x}\bar{x} + \Sigma_y \right| < 1 \end{cases} \tag{5.6}$$

which results in

$$\begin{cases} \mathrm{D}_1 \rightarrow \left|\sqrt{\Delta}\right| < 1 \\ \mathrm{D}_2 \rightarrow \left|-\sqrt{\Delta}\right| < 1, \end{cases} \tag{5.7}$$

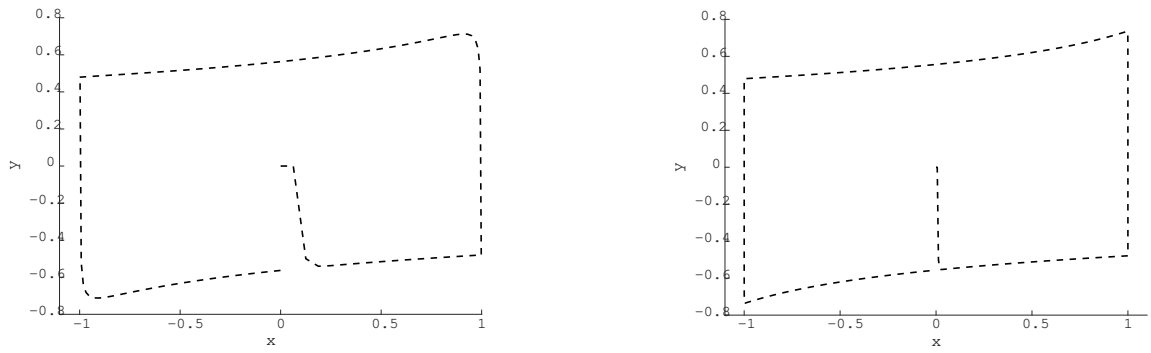
where D_1 and D_2 indicates the conditions to ensure the stability of the equilibria.

The conditions established by Equation (5.7) reinforce the central idea of the proposition, which is choosing signals whose ensures at least one stable equilibrium point concerning the Equation (5.7) that changes its location concerning the history of the inputs, and forming the $\mathcal{H}_t(w)$ in the $x - y$ plane. A simple example of an input with the characteristics mentioned above is the one generated using the sign function. The following example shows a simple model with two equilibria and two inputs representing a hysteresis behavior.

Example 5.2.1. Let $y_k = 0.3y_{k-1}^2 + 0.1y_{k-1} - 0.5\phi_{k-1}$, where $x_k = sin(\omega k)$, ω is the frequency of the input signal x, and $\phi_k = \text{sign}(\Delta(\text{x}_\text{k}))$. For this model, $\Sigma_{y^2} = 0.3$, $\Sigma_y = 0.1$ and $\Sigma_\phi = -0.5$. The equilibria of this model is given by:

$$\overline{y}(\overline{x}) = \begin{cases} \frac{0.9+\sqrt{0.81+0.6\bar{x}\phi}}{0.6\bar{x}}, \\ \frac{0.9-\sqrt{0.81+0.6\bar{x}\phi}}{0.6\bar{x}}, \end{cases} \tag{5.8}$$

where $\overline{x}$ is a loading-unloading quasi-static input signal. Because one of the equilibrium points is asymptotically stable, the output converges to $\mathcal{H}_k(w)$ in the $x - y$ plane. Note that for a constant input value $x = 1 = \overline{x}$, the equilibrium lies in $\overline{y} = 3.4791$ and $\overline{y} = -0.4791$. However, the output converges to $\overline{y} = -0.4791$ because it is the only stable. Similarly, for a constant input value $x = -1 = \overline{x}$, the equilibrium lies in $\overline{y} = -3.4791$ and $\overline{y} = 0.4791$, but only the latter is stable. In this case, the ϕ signal changes the equilibrium for each case, leading the output to converge for ± 0.7362 depending on the input trajectory. Figure 5.3 presents the hysteresis curve.



(1) $\omega = 1$. (2) $\omega = 0.1$.

Figure 5.3: The black dots are on $\mathcal{H}_k(\omega)$ for model $y_k = 0.3y_{k-1}^2 x_{k-1} + 0.1y_{k-1} - 0.5\phi_{k-1}$.

△

5.2.1 Modeling of magneto-rheological damper devices using Meta-MSS: identification of hysteresis loops considering models with two equilibrium point.

In this case, a linear-in-the-parameters NARX model with two equilibrium points is developed for MRD identification. Following Section 5.2, only the velocity x_2 (derivative of the displacement) and voltage x_1 are considered as input. The justification of this choice relies on the fact that the characteristics of the inputs can change the location of the equilibrium point if

the right model is selected. The Meta-MSS parameters are reported in Table 5.4. Again, the maximum lag and nonlinearity degree were defined according to [34] for comparison purpose. A total of 30 runs of the Meta-MSS algorithm are performed to select the most voted structure as the final model.

The final model structure representing 100.00% of the selected models has three regressors (Table 5.5) against the four terms model obtained by [34] and the five terms model generated via Meta-MSS. Although the RMSE value does not represent the same level of accuracy reached by the model using the sign function, this result demonstrates that models with multiple inputs allow an adequate representation of the analyzed system without restricting the number of equilibrium points.

Also, it is worth emphasizing that the inputs used in this case-study was the same ones generated for the case where the sign function was designed. There was no further effort to design new inputs that meets the new conditions described in Proposition 5.2.1, since our main goal is to show that the circumstances for polynomial NARX models be able to reproduce hysteresis relies on the characteristics of the input and not on the number of equilibrium points. Besides, we highlight the importance of the Meta-MSS algorithm in the model structure selection procedure since it returns a model according to the established conditions.

Table 5.4: Meta-Structure Selection Algorithm parameters used in the hysteretic system identification.

Parameters	n_{x_1}	n_{x_2}	n_y	ℓ	p-value	max_iter	n_agents	α	G_0
Values	1	1	1	3	0.05	50	15	23	100

Table 5.5: NARX model with two equilibrium points obtained using the Meta-MSS.

	Meta-MSS	
	Model term	Parameter
	y_{k-1}	0.8688
	$x_{2k-1}x_{1k-1}$	0.7198
	$x_{2k-1}y_{k-1}^2$	-0.6056
RMSE	0.1831	

Regarding this model, the equilibria lies on

$$\bar{y}(\bar{x_1}, \bar{x_2}) = \begin{cases} \frac{-\Sigma_y - 1 + \sqrt{(\Sigma_y - 1)^2 - 4(\Sigma_{y^2,x_2}\bar{x_2}\bar{x_2})(\Sigma_{x_1,x_2}\bar{x_1}\bar{x_2})}}{2\Sigma_{y^2,x2}\bar{x_2}\bar{x_2}} \\ \frac{-\Sigma_y - 1 - \sqrt{(\Sigma_y - 1)^2 - 4(\Sigma_{y^2,x_2}\bar{x_2}\bar{x_2})(\Sigma_{x_1,x_2}\bar{x_1}\bar{x_2})}}{2\Sigma_{y^2,x2}\bar{x_2}\bar{x_2}} \end{cases} . \tag{5.9}$$

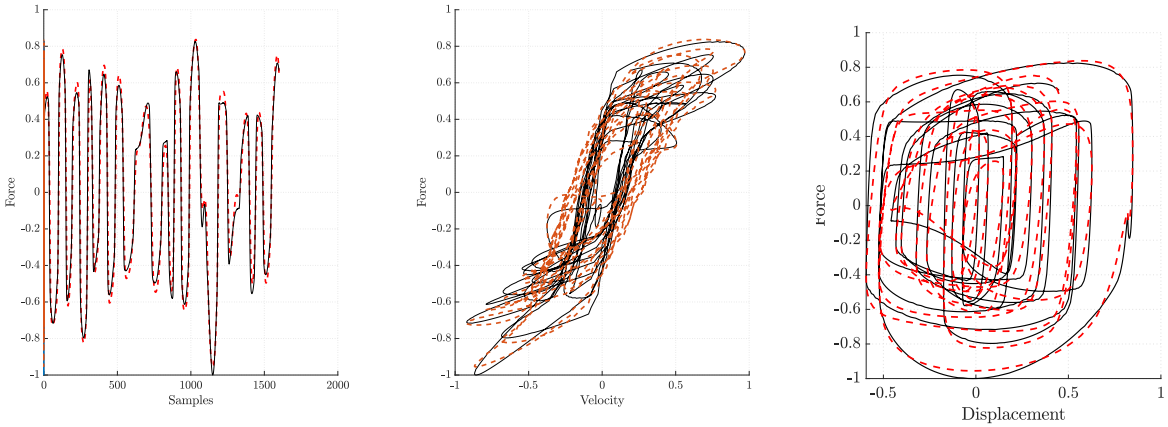
The stability of each equilibrium point depends on both inputs, x_2, and x_2. Then, the stability

of each equilibrium rely upon

$$\begin{cases} \text{if } \frac{-2\Sigma_y - 1}{4\Sigma_{y^2,x_2}\Sigma_{x_1,x_2}} < x_2^2 x_1 < \frac{2\Sigma_y - 1}{4\Sigma_{y^2,x_2}\Sigma_{x_1,x_2}}, & \text{equilibria 1 is stable,} \\ \text{if } \frac{2\Sigma_y - 1}{4\Sigma_{y^2,x_2}\Sigma_{x_1,x_2}} < x_2^2 x_1 < \frac{-2\Sigma_y - 1}{4\Sigma_{y^2,x_2}\Sigma_{x_1,x_2}}, & \text{equilibria 2 is stable} \end{cases} \tag{5.10}$$

In this case, the characteristics of the inputs result on changing the system equilibria when only one of the equilibrium points are stable, generating the hysteresis defined by a region of attraction concerning its stability. Regarding this model, there are some interval of samples where both equilibrium points are stable, which results in the model output leading to the wrong reference.

Figure 5.4 presents the simulation regarding the model obtained using the Meta-MSS algorithm and a comparison of the hysteresis loop predicted by the identified model and that plotted directly from the system values.



(1) Validation of the dynamic NARX model

(2) Validation of the dynamic NARX model.

(3) Validation of the dynamic NARX model.

Figure 5.4: Meta-MSS performance in the Bouc-Wen system identification. (−)Experimental analyzed system, (− −) NARX model.

All in all, one should also notice that although the method has been illustrated using MRD device based on the Bouc-Wen model, the described procedure to identify the model is not restricted to this specific device but is a general approach that works for a large class of systems with hysteresis regarding several shapes of hysteresis loops. This consideration affords a well-defined technique for determining the models of hysteretic systems under a vast range of conditions, and we expect this to be a meaningful step in the full characterization of these type of devices using polynomial NARX models.

5.3 Conclusions for NARX models with multiple inputs for hysteretic system identification

We proposed and studied new conditions for representing hysteretic behavior using polynomial NARX models. First, we show that the Meta-MSS algorithm is able to select terms to compose a highly accurate yet simple model in conditions where classical algorithms fail. Later, we established theoretical and empirical backing to the claim that NARX models are able to reproduce hysteresis based on the characteristics of the inputs without relying on sign functions. On the whole, we believe that our study represents a step forward in the identification of hysteretic systems using polynomial NARX models.

CHAPTER 6

META-MSS ALGORITHM FOR CLASSIFICATION PROBLEMS

Because many real-life problems associate continuous and discrete variables, classification has been one of the most widely studied techniques for decision-making tasks in engineering, business, health science, and more. Examples include medical diagnosis, credit scoring, handwritten character recognition, and speech recognition and biological classification. Because traditional statistical classification techniques are built on the Bayesian theory, these methods rely on a *pior* knowledge about the model probability and the data properties. Many methods and algorithms have been developed to overcome such drawbacks, which cover logistic regression [114], random forest [115], support vector machines [116], k-nearest neighbors [117] and logistic-NARX model for binary classification [47]. The former three algorithms are widely used, but the interpretation of such models is a hard task. Besides, the methods mentioned above can be computationally expensive [117]. The logistic-NARX model, in contrast, inherits the computational efficiency and the effortlessness interpretation provided by NARX methodology and the capacity to predict categorical output present in logistic regression models [47]. These features allow building models that predict binary outcomes in classification problems.

Besides the computational efficiency and simple structure, some advantages of the logistic-NARX approach are the possibility to include lagged terms straightforwardly while other techniques include lagged terms explicitly. Moreover, it deals with the multicollinearity problem in the model selection procedure, checking the correlations among the predictor and target variables.

Following the logistic-NARX approach, this chapter adapts the Meta-MSS algorithm to develop NARX models focusing on the prediction of systems with binary responses that depend on continuous predictors. The primary motivation comes from the fact that the logistic-NARX approach inherits not only the goodness of the FROLS but all of its drawbacks related to being stuck in locally optimal solutions. A direct comparison with the methods above is performed using the identification and evaluation of two simulated models, and an empirical system.

6.1 Logistic NARX modeling approach using Meta-MSSc algorithm

This section presents a brief description of how the NARX model is currently used for classification problems and introduces an alternative using the Meta-MSS algorithm. It is worth to point out that we are considering only binomial classification. In other words, the following sections describe how to use the Meta-MSS for tasks where the goal is to assign an output to one of two groups.

6.1.1 Theoretical framework for Logistic NARX

In [47], the logistic-NARX is based on the FROLS algorithm to select the terms to compose the following probability model

$$p_k = \frac{1}{1 + e^{-\psi_{k-1}^{\top}\hat{\Theta}}}, \tag{6.1}$$

where ψ is the information matrix and $\hat{\Theta}$ the vector of estimated parameters.

Since the output of a binomial classification problem is a dichotomous sequence and the FROLS relies on the ERR index to check the importance of each candidate term, [47] proposed a new metric based on the biserial correlation coefficient. In contrast to ERR, which loss the information from the class denoted as 0, the biserial coefficient measures the relationship between a continuous variable and a dichotomous variable according to [118]:

$$r(\boldsymbol{x},\boldsymbol{y}) = \frac{\overline{X}_1 - \overline{X}_0}{\sigma_X}\sqrt{\frac{n_1 n_0}{N^2}}, \tag{6.2}$$

where $\overline{X}_0$ is the mean value on the continuous variable X for all the observations that belong to class 0, X_1 is the mean value of variable X for all the observations that belong to class 1, σ_X is the standard deviation of variable X, n_0 is the number of observations that belong to class 0, n_1 is the number of observations that belong to class 1, and N is the total number of data points. Even though it is based on FROLS, the logistic-NARX approach requires the user to set a maximum number of regressors to form the final model, which is not required when using Meta-MSS algorithm for binary classification.

The objective function of the Meta-MSS is adapted to use the biserial correlation to measure the association between the variables instead of the RMSE:

$$\mathcal{F} = \frac{\overline{X}_1 - \overline{X}_0}{\sigma_X}\sqrt{\frac{n_1 n_0}{N^2}} \times \varrho. \tag{6.3}$$

Algorithm 8 describes the pseudo-code of the proposed method. For the continuous regression problem, the parameters are estimated using the LS method, which minimizes the sum of squared errors of the model output. Because we are dealing with categorical response variables, this approach is not capable of producing minimum variance unbiased estimators, so the parameters are estimated via a Stochastic Gradient Descent (SGD) algorithm described by the following update rule [119]:

$$w_{t+1} = w_t - \gamma \nabla_w Q(z_t, w_t), \tag{6.4}$$

where $Q(z_t, w_t)$ is a function $f \in \mathcal{F}$ that minimizes the loss $l(f_w(x), y)$ parameterized by a weight vector w, γ is the learning rate and z are samples picked randomly.

Apart from those changes, the main aspects of the standard Meta-MSS algorithm are held, such as the penalty function, the regressor significance evaluation, and all aspects of exploration and exploitation of the search space. Because the parameters are now estimated using SGD, the method becomes more computationally demanding, and this can slow down the method, especially when concerning with large models.

6.2 Example 1

Consider the following model taken from [47]:

$$y_k = \begin{cases} 1 & \text{if } x_{k-1}^2 + 2v_{k-2} - 0.8u_{k-2}v_{k-1} + e_k < 1 \\ 0 & \text{otherwise,} \end{cases} \tag{6.5}$$

where $x_k, v_k \sim \mathcal{U}(-1, 1)$ and $e_k \sim \mathcal{N}(0, 0.3^2)$. For the purpose of a fair comparison, all realizations of the models are composed of a total of 1000 input-output data samples. A total of 700 samples are used in training phase and 300 are used as validation set. The data can be visualized in Figure 6.1.

The Meta-MSSc algorithm is applied to this data set considering the generated binary output. Table 6.1 shows the parameters of the proposed method. The Meta-MSS$_c$ returns a model with

Algorithm 8: Meta-structure selection algorithm for binary classification

Result: Model which has the best fitness value

Input: $\{(u_k),(y_k), k = 1,\ldots,N\}$, $\mathcal{M} = \{\psi_j, j = 1,\ldots,m\}$, n_y, n_u, ℓ,γ, max_iteration, noV, np

1 $\mathcal{P} \leftarrow$ Build initial population of random agents in the search space, $\mathcal{S}$
2 $v \leftarrow$ set the agent's velocity equal zero at first iteration
3 $\Psi \leftarrow$ Build the general matrix of regressors based on n_y, n_u and ℓ ▷ equation (2.28)
4 **repeat**
5 **for** $i = 1 : d$ **do**
6 $\mathrm{m_i} \leftarrow \vec{x}_{np,i}$ ▷ Extract the model encoding from population
7 $\Psi_r \leftarrow \Psi(\mathrm{m_i})$ ▷ Delete the Ψ columns where $\mathrm{m_i} = 0$ Example 3.1.1
8 $\hat{\Theta} \leftarrow$ Estimated via SGD ▷ Equation (6.4)
9 $\hat{y} \leftarrow$ Free-run simulation of the model
10 $V \leftarrow (\Psi^\top \Psi)^{-1}$
11 $\hat{\sigma}_e^2 = \frac{1}{N-m}\sum_{k=1}^{N}(y_k - \psi_{k-1}^\top \hat{\Theta})$ ▷ Equation (3.4)
12 **for** $h = 1 : \tau$ **do**
13 $\hat{\sigma}^2 \leftarrow \hat{\sigma}_e^2 V_{h,h}$ ▷ Equation (3.3)
14 $\mathrm{se}(\hat{\Theta}_j) \leftarrow \sqrt{\hat{\sigma}_{h,h}^2}$ ▷ Equation (3.5)
15 $T_0 \leftarrow \frac{\hat{\Theta}_j}{\mathrm{se}(\hat{\Theta})}$ ▷ Equation (3.6)
16 $p \leftarrow$ regressors where $-t_{\alpha/2,N-m} < T_0 < t_{\alpha/2,N-m}$ ▷ Equation (3.7)
17 **end**
18 Remove the p regressors from Ψ_r
19 Check for empty model
20 **if** *Model is empty* **then**
21 Generate a new population
22 Repeat the steps from line 6 to 18
23 **end**
24 $n_1 \leftarrow \mathrm{size}(p)$ ▷ Number of redundant terms
25 $\hat{\Theta} \leftarrow$ Re-estimate the parameters ▷ Equation (6.4)
26 $\mathcal{F}_i \leftarrow \frac{\overline{X}_1 - \overline{X}_0}{\sigma_X}\sqrt{\frac{n_1 n_0}{N^2}} \times \varrho$ ▷ Equation (6.3)
27 $\mathcal{P}_i^n \leftarrow$ Encoded Ψ_r
28 Evaluate the fitness for each agent, $\mathcal{F}_i(t)$
29 **end**
30 $\mathcal{P} \leftarrow \mathcal{P}^n$ ▷ Update the population
31 **foreach** $\vec{x}_{np,d} \in \mathcal{P}$ **do**
32 Update the gravitational constant, G ▷ equation (2.60)
33 Update $\mathcal{J}$ ▷ equation (2.67)
34 Update $\mathcal{O}$ ▷ equation (2.68)
35 Calculate the inertial masses of each agent ▷ equation (2.65)
36 Calculate the acceleration of each agent ▷ equation (2.62)
37 Adapt the c_j' coefficients ▷ equation (2.74)
38 Update the velocity of the agents ▷ equation (2.72)
39 Update the position of the agents ▷ equation (2.64)
40 **end**
41 **until** *max_iterations is reached*

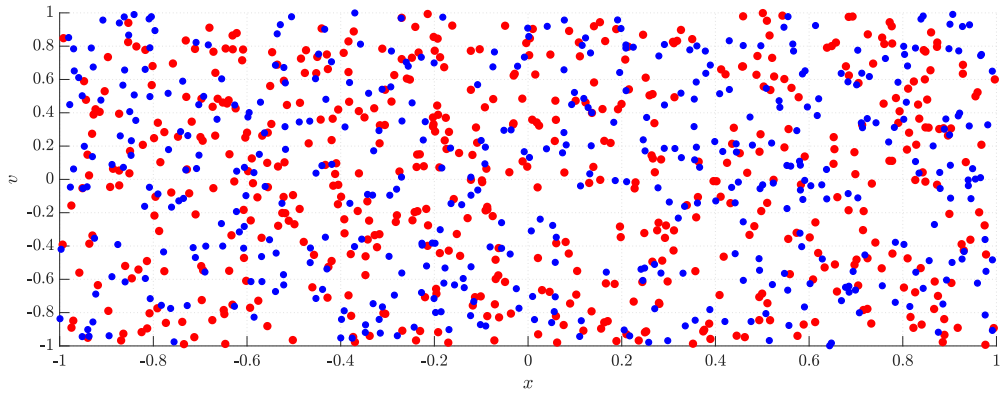


Figure 6.1: Data points obtained from the model given in equation (6.5). Points in blue corresponds to class $y = 0$ and points in red corresponds to $y = 1$. There is not a clear boundary separating the two classes.

the same 4 terms obtained using the logistic NARX approach, which is shown in Table 6.2. It is worth to mention that the proposed algorithm was able to identify correctly all model terms used to generate the decision boundary for equation (6.5) in all 100 runs. The parameters obtained are log odds ratios, *i.e.*, the ratio of the odds for a dichotomous that takes the value 1 to the odds for those that take the value 0. Therefore they need to be interpreted accordingly.

Table 6.1: Parameters used in Meta-Structure Selection Algorithm for binary classification

Parameters	n_u	ℓ	p-value	γ	max_iter	n_agents	α	G_0
Values	4	3	0.05	0.01	50	15	23	100

Table 6.2: Identified NARX model using Algorithm 8. All model terms selected correspond to the true model. The parameters obtained are log odds ratios and therefore they not need to be similar to the ones in original model.

Meta-MSS$_c$		Logistic NARX	
Model term	Parameter	Model term	Parameter
v_{k-2}^2	-28.72	v_{k-2}^2	-12.50
constant	14.34	constant	6.15
x_{k-1}^2	-14.25	x_{k-1}^2	-6.08
$x_{k-2}^2 v_{k-1}$	11.76	$x_{k-2}^2 v_{k-1}$	4.58

The elapsed time of the Meta-MSS$_c$ was $18.53s$. As expected, the computational time increased due to the use of SGD algorithm to estimate the model parameters.

Following the methodology used in [47], we compare the Meta-MSSc performance against popular techniques. In this respect, the results of a regression-like NARX model based on the approach suggested in [120], a random forest with 500 trees, a support vector machine with a radial basis kernel and a k-nearest neighbors model are trained with the same training set are

taken directly from [47]. Moreover, the authors evaluated the cases where the same lagged input and output variables considered for the logistic NARX model are added explicitly in the traditional techniques. Table 6.3 reports that the proposed method was able to select the correct terms, and has the best accuracy together with the other two NARX approaches and the Random Forest with autoregressive inputs. However, those methods have some drawbacks overcame by Meta-MSSc: the regression NARX approach produces real-valued outputs, which is difficult to interpret in a binary classification scenario. The logistic NARX requires several runs of the algorithm to define the optimal number of model terms. Finally, the Random Forest remains as a black box while the NARX model is easy to interpret. Therefore, the proposed method is a feasible alternative for binary classification problems.

Table 6.3: Accuracy performance between different methods for modelling of equation (6.5). The Meta-MSS algorithm is compared against a logistic NARX model obtained via an adapted FROLS algorithm, a regression-like NARX model based on the approach suggested in [120], a random forest with 500 trees, a support vector machine with a radial basis kernel and a k-nearest neighbors model. The exclusion-inclusion of autoregressive inputs is also taken into account. The logistic NARX model has a comparable performance with the rest of the techniques. The NARX-type models have the best accuracy. This is expected given the NARX-like structure that generates the data.

Method	Classification accuracy
Meta-MSSc	0.8582
Logistic NARX	0.8581
Regression NARX	0.8581
Random Forest (without autoregressive inputs)	0.5034
Support Vector Machine (without autoregressive inputs)	0.5574
K-Nearest Neighbors (without autoregressive inputs)	0.5267
Random Forest (with autoregressive inputs)	0.8514
Support Vector Machine (with autoregressive inputs)	0.777
K-Nearest Neighbors (with autoregressive inputs)	0.6284

6.3 Example 2

Consider the following model taken from [47]:

$$y_k = \begin{cases} 1 & \text{if } -x_{k-1}\sqrt{|v_{k-1}|} + 0.5x_{k-1}^3 + sin(v_{k-2}) + e_k < 0 \\ 0 & \text{otherwise,} \end{cases} \tag{6.6}$$

where $x_k, v_k \sim \mathcal{U}(-1, 1)$ and $e_k = w_k + 0.3w_{k-1} + 0.6w_{k-2}$ and $w \sim \mathcal{N}(0, 0.3^2)$. Following [47], a total of 1000 input-output data samples are collected. A total of 700 samples are

used in training phase and 300 are used as validation set. Figure 6.2 depicts the data.

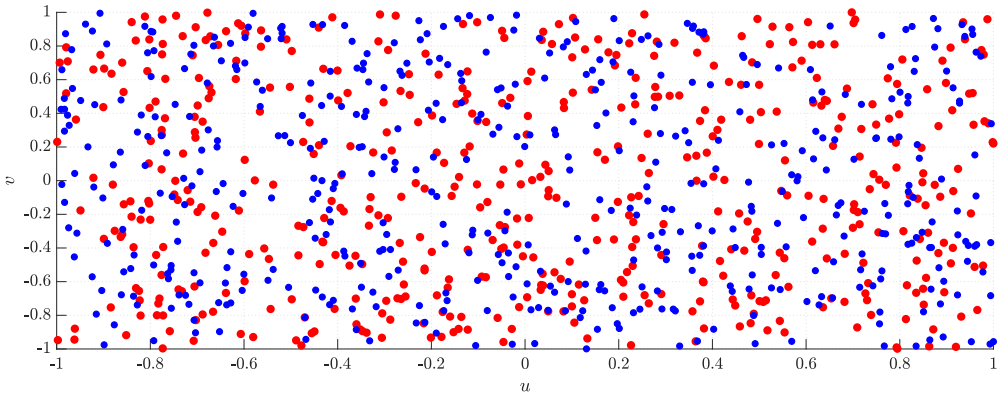


Figure 6.2: Data points obtained from the model given in equation (6.6). Points in blue corresponds to class $y = 0$ and points in red corresponds to $y = 1$. There is not a clear boundary separating the two classes.

The Meta-MSSc algorithm is applied to this data set. The Meta-MSSc parameters are shown in Table 6.4. The number of iterations is increased to allow a better exploration in the search space. Table 6.5 compares that the Meta-MSS$_c$ returns a model with 6 terms while the one obtained using the logistic-NARX approach have 8 terms. The model built using the proposed algorithm was chosen based on the number of votes considering 100 runs. Thus, the model described in Table 6.5 are chosen to generate the decision boundary for equation (6.6) with 84% of votes. The parameters obtained are the constant effect of the predictor variables on the probability of occurrence of one outcome.

Table 6.4: Parameters used in Meta-Structure Selection Algorithm for binary classification

Parameters	n_u	ℓ	p-value	γ	max_iter	n_agents	α	G_0
Values	4	3	0.05	0.01	50	15	23	100

The elapsed time of the Meta-MSS$_c$ was $44.3s$. Similar to the previous case study, we compare the Meta-MSSc performance against a regression-like NARX model based on the approach suggested in [120], a random forest with 500 trees, a support vector machine with a radial basis kernel and a k-nearest neighbors model. Again, the cases where lagged input and output variables are added explicitly in the traditional techniques are compared. Table 6.6 demonstrates that the new method has the best accuracy together with the other two NARX approaches and the Random Forest with autoregressive inputs. Once again, the drawbacks mentioned above related to the interpretability of those methods are overcame by Meta-MSSc.

Table 6.5: Identified NARX model using Algorithm 8. All model terms selected correspond to the true model. The parameters obtained are log odds ratios and therefore they not need to be similar to the ones in the original model.

Meta-MSS$_c$		Logistic NARX	
Model term	Parameter	Model term	Parameter
v_{k-2}	-20.453	v_{k-2}	-12.755
$v_{k-1}^2 v_{k-2}$	20.475	$x_{k-1} v_{k-1}^2$	8.488
x_{k-1}^3	-7.8219	x_{k-1}^3	0.224
$v_{k-4}^2 v_{k-2}$	-6.606	$x_{k-1} v_{k-2}^2$	10.066
v_{k-2}^3	-7.661	v_{k-2}^3	-15.323
x_{k-1}	9.553	x_{k-1}	9.047
-	-	constant	-8.715
-	-	$x_{k-1} x_{k-4}^2$	-3.285

Table 6.6: Accuracy performance between different methods for modelling of equation (6.5). The Meta-MSS algorithm is compared against a logistic NARX model obtained via an adapted FROLS algorithm, a regression-like NARX model based on the approach suggested in [120], a random forest with 500 trees, a support vector machine with a radial basis kernel and a k-nearest neighbors model.

Method	Classification accuracy
Meta-MSSc	0.9410
Logistic NARX	0.9392
Regression NARX	0.9358
Random Forest (without autoregressive inputs)	0.5270
Support Vector Machine (without autoregressive inputs)	0.4932
K-Nearest Neighbors (without autoregressive inputs)	0.4700
Random Forest (with autoregressive inputs)	0.9223
Support Vector Machine (with autoregressive inputs)	0.8986
K-Nearest Neighbors (with autoregressive inputs)	0.7973

6.4 Electroencephalography Eye State Identification

In the latest years, there has been a rapid rise in the use of electroencephalography (EEG) eye state classification for human cognitive state classification [3]. A recent review of the literature on this topic found valuable applications to medical care and some quotidian tasks. For example, some studies have been carried out regarding sleep-waking states in infants [121], healthy aging and age-related brain degeneration [122] and social anxiety symptoms [123]. In this context, [124] built a dataset containing 117 seconds of EEG eye state corpus with a total of 14,980 EEG measurements from 14 different sensors taken with the Emotiv EEG neuroheadset to predict eye states [47]. Their dataset is now frequently used as a benchmark and is available on Machine Learning Repository, University of California, Irvine (UCI) [125]. Figure 6.3 detailed the neuroheadset sensor position and corresponding behavior groups. The reader is referred to [3] for additional information regarding the experiment.

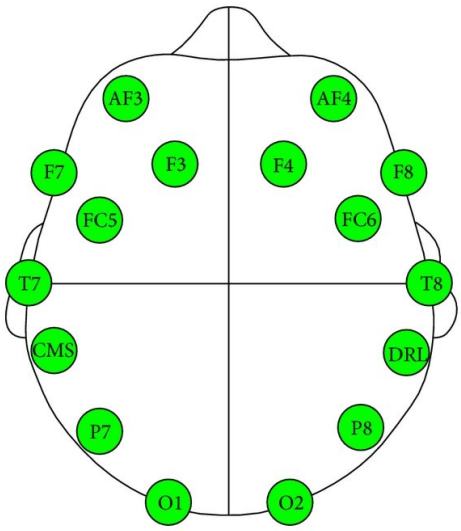
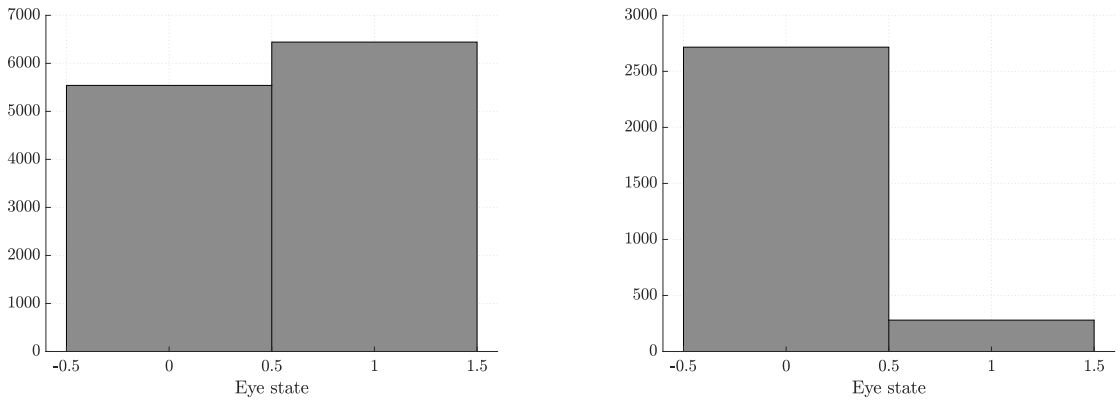


Figure 6.3: Emotiv EEG neuroheadset sensor position and corresponding behavior groups. Source: [3]

Following the method in [47], the data is separated in a training set composed of 80% of the data and a testing set with the remainder. The eye state is encoded as follows: 1 indicates the eye-closed and 0 the eye-open state. The distribution of the eye state for the training and testing dataset is shown in Figure 6.4.



(1) Training dataset. (2) Testing dataset.

Figure 6.4: Frequency of the eye state for training and testing dataset.

Some summary statistics, *e.g.*, central tendency, standard deviation, interquartile range and skewness was performed on training dataset to check if the data have missing values or an outlier to be fixed. In this respect, were found values corresponding to inaccurate or corrupt records in all-time provided from sensors. The detected inaccurate values are replaced with the mean value of the remaining measurements for each variable. Also, each input sequence is transformed using scale and centering transformations.

An interesting fact is that the Logistic NARX based on FROLS [47] was not able to achieve satisfactory performance when trained with the original dataset. In that paper, the authors explained the lousy performance as a consequence of the high variability and dependency between the variables measured. Table 6.7 reports that the Meta-MSSc, on the other hand, was capable of building a model with 10 terms and accuracy of 65.04%.

Table 6.7: Identified NARX model using Algorithm 8. This model was built using the original dataset using the Meta-MSSc. No comparison was made because the FROLS based technique was not capable to generate a model which performed well enough. The parameters obtained are log odds ratios and therefore they do not need to be similar to the ones in the original model

Meta-MSS$_c$	Model term	constant	x_1k-1	x_4k-30	x_4k-36	x_4k-38	x_4k-41	x_6k-2	x_7k-5	$x_{12}k-1$	$x_{13}k-1$
	Parameter	0.2055	-0.1077	0.1689	0.1061	0.0751	0.1393	0.3573	-0.7471	-0.4736	0.3875

This result may appear to be a poor performance. However, the Logistic NARX achieved 0.7199, and the higher score achieved by the popular techniques was 0.6473, considering the case where a principal component analysis (PCA) drastically reduced the data dimensionality. Thus, this result shows a powerful performance of the Meta-MSS algorithm.

Concerning the mentioned results, for comparison purpose, we also performed a dimensionality reduction using PCA, and the first five principal components were selected as a representation of the original data. This procedure follows the steps described in [47]. Table 6.8 illustrates that Meta-MSSc has built the model with the best accuracy together with the Logistic NARX approach. The models built without autoregressive inputs have the worst classification accuracy, although this is improved with the addition of autoregressive terms. However, even with autoregressive information, the popular techniques do not achieve a classification accuracy to take up the ones obtained by the Meta-MSSc and Logistic NARX methods.

6.5 Conclusions for Meta-MSS Algorithm for Classification Problems

In this chapter, we addressed the problem of binomial classification using Meta-MSS algorithm to build the logistic NARX model. Our numerical experiments and case-study revealed that when the goal to build explainable models based on dynamic data and dichotomous variables, using the proposed method has many advantages such as the robustness against multicollinearity, superior performance in high dimensional data, and computational efficiency.

Table 6.8: Accuracy performance between different methods for modelling of equation (6.5). The Meta-MSS algorithm is compared against a logistic NARX model obtained via an adapted FROLS algorithm, a regression-like NARX model based on the approach suggested in [120], a random forest with 500 trees, a support vector machine with a radial basis kernel and a k-nearest neighbors model. The exclusion-inclusion of autoregressive inputs is also taken into account. Clearly, the NARX-type models have the best accuracy.

Method	Classification accuracy
Meta-MSSc	0.7480
Logistic NARX	0.7199
Regression NARX	0.6643
Random Forest (without autoregressive inputs)	0.5475
Support Vector Machine (without autoregressive inputs)	0.6029
K-Nearest Neighbors (without autoregressive inputs)	0.5041
Random Forest (with autoregressive inputs)	0.6365
Support Vector Machine (with autoregressive inputs)	0.6473
K-Nearest Neighbors (with autoregressive inputs)	0.5662

CHAPTER 7

FINAL CONSIDERATIONS

This thesis presents the structure selection of polynomial NARX models using a hybrid and binary Particle Swarm Optimization and Gravitational Search Algorithm. The selection procedure considers the individual importance of each regressor along with the free-run-simulation performance to apply a penalty function in candidate solutions.

The technique, called Meta-MSS algorithm in its standard form, is extended and analyzed into four main categories: (i) the system identification in a mono-objective approach; (ii) the inclusion of affine information in parameter estimation and, consequently, the structure selection; (iii) the identification of hysteresis nonlinearity considering models with multiple inputs and n equilibrium points and; (iv) finally, the identification of systems with binary responses using a logistic approach.

Regarding the first category, the new method outperformed or at least was compatible with classical approaches such as FROLS, and modern techniques such as RaMSS, C-RaMSS, RJMCMC, and a meta-heuristic based algorithm. This statement considers the results obtained in the model selection of 6 simulated models taken from literature, the identification of an Electrical Heater, and the performance on the F-16 Ground Vibration benchmark. Concerning the second category, the MoMeta-MSS algorithm also shown an impressive capability for parsimonious model selection. The algorithm developed models with an excellent trade-off between the dynamic and steady-state behavior when compared against the FROLS and MERR algorithms. The identification of an electrical heater and a DC-DC buck converter illustrated that the MoMeta-MSS algorithm is a feasible alternative to classical frameworks regarding multi-objective procedures. The third category shed new lights on the identification of hysteretic systems using polynomial NARX. The new technique overcame several drawbacks whose cause poor performance in classical methods and can produce models with an excellent response without any analytical interference by the user in the identification procedure. Besides, the method reduces the conservatism for hysteresis representation through polynomial NARX and can produce models with multiple equilibrium points if the characteristics of the inputs are

guaranteed. In this respect, the Meta-MSS can develop satisfactory models without including the sign function to generate the additional input signal. Finally, the latter category proves the robust performance of the technique using an adapted algorithm, called Meta-MSSc, to build models to predict binary outcomes in classification problems. Again, the proposed algorithm outperformed or at least was compatible with popular techniques such as K-Nearest Neighbors, Random Forests and Support Vector Machine, and recent approaches based on FROE algorithm using NARX models. Besides the two simulated examples, the electroencephalography eye state identification proved that the Meta-MSSc algorithm could handle the multicollinearity problem better than all of the compared techniques. In this case study, the new algorithm returned a model with satisfactory performance even when the data dimensionality was not transformed using data reduction techniques, which was not possible with the algorithms used for comparisons purposes.

Furthermore, although the stochastic nature of the Meta-MSS algorithm, the individual evaluation of the regressors and the penalty function results in fast convergence. In this respect, the computational efficiency is better or at least consistent with other stochastic procedures, such as RaMSS, C-RaMSS, RJMCMC, and the meta-heuristic based algorithm. The computational effort relies on the number of search agents, the maximum number of iterations, and the search space dimensionality. Therefore, in some cases, the elapsed time of the Meta-MSS is compatible with the FROLS. It is worth to mention that the computational cost concerning the Meta-MSSc is higher because the parameters are estimated in a nonlinear framework.

7.1 Future Work

The development of a meta-heuristic based algorithm for model selection such as the Meta-MSS permits a broad exploration in the field of system identification. Although some analysis are out of scope and, therefore, are not addressed in this thesis, future work are open for research. Some of the possibles reformulations and approaches are listed in what follows.

- This thesis neglected the inclusion of noise process terms in model structure selection, which is an important problem concerning the identification of polynomial autoregressive models. In this respect, an exciting continuation of this work would be to implement an extended version of Meta-MSS to return NARMAX models.

- The multi-objective approach is based on the weighted sum method and relies on parameters estimation. Although the MoMeta-MSS defines the objective function depending on the Pareto set of each candidate model and, consequently, impacts the model structure selection, purely multi-objective meta-heuristics algorithms could be implemented to handle each objective separately.

- More in-depth research can be done regarding hysteresis identification using models with multiple inputs and equilibria. This thesis deals with the model selection considering multiple inputs and equilibrium points for rate-independent hysteresis so that one can address the necessary conditions to NARX models reproduces rate-dependent hysteresis.

- The Meta-MSS is quite general and, therefore, one can extend the algorithm to handle classification problems with static variables and multiclass problems.

BIBLIOGRAPHY

[1] T. Baldacchino, S. R. Anderson, e V. Kadirkamanathan, "Computational system identification for Bayesian NARMAX modelling," *Automatica*, vol. 49, no. 9, pp. 2641–2651, 2013.

[2] A. G. V. Severino e F. M. U. d. Araújo, "Meta-heurísticas aplicadas à identificação de sistemas," in *Simpósio Brasileiro de Automação Inteligente*, 2017, pp. 609–614.

[3] T. Wang, S.-U. Guan, K. L. Man, e T. O. Ting, "EEG eye state identification using incremental attribute learning with time-series classification," *Mathematical Problems in Engineering*, vol. 2014, 2014.

[4] B. F. Spencer e M. K. Sain, "Controlling buildings: a new frontier in feedback," *IEEE Control Systems Magazine*, vol. 17, no. 6, pp. 19–35, 1997.

[5] S. A. Billings, *Nonlinear system identification: NARMAX methods in the time, frequency, and spatio-temporal domains*. Chichester: John Wiley & Sons, 2013.

[6] N. Wiener, "Nonlinear problems in random theory," Massachusetts Institute of Technology, Tech. Rep., 1958.

[7] Y. W. Lee e M. Schetzen, "Measurement of the wiener kernels of a non-linear system by cross-correlation," *International Journal of Control*, vol. 2, no. 3, pp. 237–254, 1965.

[8] M. Schetzen, *The Volterra and Wiener Theories of Nonlinear Systems*. John Wiley & Sons, 1980.

[9] W. J. Rugh, *Nonlinear System Theory*. Johns Hopkins University Press, 1981.

[10] R. Haber e L. Keviczky, *Nonlinear System Identification - Input-Output Modeling Approach*. Kluwer Academic Publishers, 1999, vol. 1 and 2.

[11] S. A. Billings e H. Wei, "A new class of wavelet networks for nonlinear system iidentification," *IEEE Transactions on neural networks*, vol. 16, no. 4, pp. 862–874, 2005.

[12] R. Pintelon e J. Schoukens, *System Identification: A Frequency Domain Approach*. John Wiley & Sons, 2012.

[13] S. A. Billings e I. J. Leontaritis, "Identification of Nonlinear Systems Using Parameter Estimation Techniques," in *Proceedings of the IEEE Conference on Control and its Application*, 1981, pp. 183–187.

[14] I. J. Leontaritis e S. A. Billings, "Input-output parametric models for non-linear systems – part i: Deterministic non-linear systems; part ii: Stochastic non-linear systems," *International Journal of Control*, vol. 41, no. 2, pp. 303–328; 329–344, 1985.

[15] S. Chen e S. A. Billings, "Representations of non-linear systems: the NARMAX model," *International Journal of Control*, vol. 49, no. 3, pp. 1013–1032, 1989.

[16] E. H. K. Fung, Y. K. Wong, H. F. Ho, e M. P. Mignolet, "Modelling and prediction of machining errors using ARMAX and NARMAX structures," *Applied Mathematical Modelling*, vol. 27, no. 8, pp. 611–627, 2003.

[17] L. Guo, H. Wang, e A. P. Wang, "Optimal probability density function control for NARMAX stochastic systems," *Automatica*, vol. 44, no. 7, pp. 1904–1911, 2008.

[18] S. L. Kukreja, H. L. Galiana, e R. E. Kearney, "NARMAX representation and identification of ankle dynamics," *IEEE transactions on biomedical engineering*, vol. 50, no. 1, pp. 70–81, 2003.

[19] R. Boynton, M. Balikhin, H. Wei, e Z. Lang, "Applications of NARMAX in space weather," in *Machine Learning Techniques for Space Weather*. Elsevier, 2018, pp. 203–236.

[20] G. Acuña, C. Ramirez, e M. Curilem, "Comparing NARX and NARMAX models using ANN and SVM for cash demand forecasting for ATM," in *The 2012 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2012, pp. 1–6.

[21] N. Chiras, C. Evans, e D. Rees, "Nonlinear gas turbine modeling using NARMAX structures," *IEEE Transactions on Instrumentation and Measurement*, vol. 50, no. 4, pp. 893–898, 2001.

[22] L. Ljung, *System identification: theory for the user*. Prentice-hall, 1987.

[23] L. A. Aguirre e S. A. Billings, “Dynamical effects of overparametrization in nonlinear models,” *Physica. D, Nonlinear Phenomena*, vol. 80, no. 1-2, pp. 26–40, 1995.

[24] L. Piroddi e W. Spinelli, “An identification algorithm for polynomial NARX models based on simulation error minimization,” *International Journal of Control*, vol. 76, no. 17, pp. 1767–1781, 2003.

[25] M. L. Korenberg, S. A. Billings, Y. P. Liu, e P. J. McIlroy, “Orthogonal parameter estimation algorithm for non-linear stochastic systems,” *International Journal of Control*, vol. 48, no. 1, pp. 193–210, 1988.

[26] M. Bonin, V. Seghezza, e L. Piroddi, “NARX model selection based on simulation error minimisation and LASSO,” *IET control theory & applications*, vol. 4, no. 7, pp. 1157–1168, 2010.

[27] S. A. Billings, S. Chen, e M. J. Korenberg, “Identification of MIMO non-linear systems using a forward-regression orthogonal estimator,” *International journal of control*, vol. 49, no. 6, pp. 2157–2189, 1989.

[28] M. Farina e L. Piroddi, “Identification of polynomial input/output recursive models with simulation error minimisation methods,” *International Journal of Systems Science*, vol. 43, no. 2, pp. 319–333, 2012.

[29] Y. Guo, L. Z. Guo, S. A. Billings, e H. Wei, “An iterative orthogonal forward regression algorithm,” *International Journal of Systems Science*, vol. 46, no. 5, pp. 776–789, 2015.

[30] K. Z. Mao e S. A. Billings, “Variable selection in non-linear systems modelling,” *Mechanical Systems and Signal Processing*, vol. 13, no. 2, pp. 351–366, 1999.

[31] S. A. Billings e L. A. Aguirre, “Effects of the sampling time on the dynamics and identification of nonlinear models,” *International journal of Bifurcation and Chaos*, vol. 5, no. 06, pp. 1541–1556, 1995.

[32] P. Palumbo e L. Piroddi, “Seismic behaviour of buttress dams: Non-linear modelling of a damaged buttress based on ARX/NARX models,” *Journal of Sound and Vibration*, vol. 239, no. 3, pp. 405–422, 2001.

[33] A. Falsone, L. Piroddi, e M. Prandini, "A randomized algorithm for nonlinear model structure selection," *Automatica*, vol. 60, pp. 227–238, 2015.

[34] S. A. M. Martins e L. A. Aguirre, "Sufficient conditions for rate-independent hysteresis in autoregressive identified models," *Mechanical Systems and Signal Processing*, vol. 75, pp. 607–617, 2016.

[35] H. Akaike, "A new look at the statistical model identification," *IEEE Transactions on Automatic Control*, vol. 19, no. 6, pp. 716–723, 1974.

[36] ——, "Fitting autoregressive models for prediction," *Annals of the Institute of Statistical Mathematics*, vol. 21, no. 1, pp. 243–247, 1969.

[37] G. Schwarz, "Estimating the dimension of a model," *The Annals of Statistics*, vol. 6, no. 2, pp. 461–464, 1978.

[38] S. Chen, X. Hong, e C. J. Harris, "Sparse kernel regression modeling using combined locally regularized orthogonal least squares and D-optimality experimental design," *IEEE Transactions on Automatic Control*, vol. 48, no. 6, pp. 1029–1036, 2003.

[39] R. Tempo, G. Calafiore, e F. Dabbene, *Randomized algorithms for analysis and control of uncertain systems: with applications*. Springer Science & Business Media, 2012.

[40] K. Rodriguez-Vazquez, C. M. Fonseca, e P. J. Fleming, "Identifying the structure of nonlinear dynamic systems using multiobjective genetic programming," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 34, no. 4, pp. 531–545, 2004.

[41] C. Cantelmo e L. Piroddi, "Adaptive model selection for polynomial NARX models," *IET control theory & applications*, vol. 4, no. 12, pp. 2693–2706, 2010.

[42] H. Wei e S. A. Billings, "Model structure selection using an integrated forward orthogonal search algorithm assisted by squared correlation and mutual information," *International Journal of Modelling, Identification and Control*, vol. 3, no. 4, pp. 341–356, 2008.

[43] F. C. Bassi, S. A. M. Martins, e E. G. Nepomuceno, "Influência do máximo atraso na determinaçao do tamanho de modelos via critério de informaçao de akaike multiobjetivo," in *Anais da Conferência Brasileira de Dinâmica, Controle e Aplicaçoes*, 2015.

[44] S. A. M. Martins, E. G. Nepomuceno, e M. F. S. Barroso, "Improved structure detection for polynomial narx models using a multiobjective error reduction ratio," *Journal of Control, Automation and Electrical Systems*, vol. 24, no. 6, pp. 764–772, 2013.

[45] A. M. Barbosa, R. H. C. Takahashi, e L. A. Aguirre, "Equivalence of non-linear model structures based on pareto uncertainty," *IET Control Theory & Applications*, vol. 9, no. 16, pp. 2423–2429, 2015.

[46] L. A. Aguirre, "Identification of smooth nonlinear dynamical systems with non-smooth steady-state features," *Automatica*, vol. 50, no. 4, pp. 1160–1166, 2014.

[47] J. R. A. Solares, H.-L. Wei, e S. A. Billings, "A novel logistic-narx model as a classifier for dynamic binary classification," *Neural Computing and Applications*, vol. 31, no. 1, pp. 11–25, 2019.

[48] L. A. Aguirre, *Introdução à identificação de sistemas–Técnicas lineares e não-lineares aplicadas a sistemas reais*. Editora UFMG, 2004.

[49] I. J. Leontaritis e S. A. Billings, "Experimental design and identifiability for non-linear systems," *International Journal of Systems Science*, vol. 18, no. 1, pp. 189–202, 1987.

[50] K. Weierstrass, "Über die analytische darstellbarkeit sogenannter willkürlicher functionen einer reellen veränderlichen," *Sitzungsberichte der Königlich Preußischen Akademie der Wissenschaften zu Berlin*, vol. 2, pp. 633–639, 1885.

[51] L. Lacerda Junior, W. R, V. M. Almeida, e S. A. M. Martins, "Identificação de um motor/gerador cc por meio de modelos polinomiais autorregressivos e redes neurais artificais," in *XIII Simpósio Brasileiro de Automação Inteligente*, Porto Alegre, 2017, pp. 1–6.

[52] S. A. M. Martins, "Modelos auto regressivos para representaçao de sistemas com histerese," Tese de Doutorado, Universidade Federal de Minas Gerais, 2016.

[53] A. Visintin, *Differential models of hysteresis*. Springer Science & Business Media, 2013, vol. 111.

[54] I. Ahmad, "Two degree-of-freedom robust digital controller design with Bouc-Wen hysteresis compensator for piezoelectric positioning stage," *IEEE Access*, vol. 6, pp. 17 275–17 283, 2018.

[55] A. Leva e L. Piroddi, "Narx-based technique for the modelling of magneto-rheological damping devices," *Smart Materials and Structures*, vol. 11, no. 1, p. 79, 2002.

[56] S. Chen, S. A. Billings, e W. Luo, "Orthogonal least squares methods and their application to non-linear system identification," *International Journal of control*, vol. 50, no. 5, pp. 1873–1896, 1989.

[57] H. Wei, D. Zhu, S. A. Billings, e M. A. Balikhin, "Forecasting the geomagnetic activity of the dst index using multiscale radial basis function networks," *Advances in Space Research*, vol. 40, no. 12, pp. 1863–1870, 2007.

[58] F. Hafiz, A. Swain, E. M. A. M. Mendes, e 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.

[59] Y. Gu, H. Wei, e M. M. Balikhin, "Nonlinear predictive model selection and model averaging using information criteria," *Systems Science & Control Engineering*, vol. 6, no. 1, pp. 319–328, 2018.

[60] E. G. Nepomuceno, R. H. C. Takahashi, e L. A. Aguirre, "Multiobjective parameter estimation for non-linear systems: affine information and least-squares formulation," *International Journal of Control*, vol. 80, no. 6, pp. 863–871, 2007.

[61] A. Baykasoglu, S. Owen, e N. Gindy, "A taboo search based approach to find the pareto optimal set in multiple objective optimization," *Engineering Optimization*, vol. 31, no. 6, pp. 731–748, 1999.

[62] P. J. Fleming, R. C. Purshouse, e R. J. Lygoe, "Many-objective optimization: An engineering design perspective," in *International conference on evolutionary multi-criterion optimization*. Springer, 2005, pp. 14–32.

[63] B. Li, J. Li, K. Tang, e X. Yao, "Many-objective evolutionary algorithms: A survey," *ACM Computing Surveys (CSUR)*, vol. 48, no. 1, p. 13, 2015.

[64] P. L. Yu, "Cone convexity, cone extreme points, and nondominated solutions in decision problems with multiobjectives," *Journal of Optimization Theory and Applications*, vol. 14, no. 3, pp. 319–377, 1974.

[65] E. G. Nepomuceno, "Identificação multiobjetivo de sistemas não-lineares," Dissertação de Mestrado, Federal University of Minas Gerais, 2002.

[66] S. A. Billings e W. S. F. Voon, "Structure detection and model validity tests in the identification of nonlinear systems," in *IEE Proceedings D-Control Theory and Applications*, vol. 130, no. 4. IET, 1983, pp. 193–199.

[67] ——, "A prediction-error and stepwise-regression estimation algorithm for non-linear systems," *International Journal of Control*, vol. 44, no. 3, pp. 803–822, 1986.

[68] S. A. Billings e Q. H. Tao, "Model validity tests for non-linear signal processing applications," *International Journal of Control*, vol. 54, no. 1, pp. 157–194, 1991.

[69] P. J. M. Van Laarhoven e E. H. L. Aarts, "Simulated annealing," in *Simulated annealing: Theory and applications*. Springer, 1987, pp. 7–15.

[70] S. Mirjalili, S. M. Mirjalili, e A. Lewis, "Grey wolf optimizer," *Advances in engineering software*, vol. 69, pp. 46–61, 2014.

[71] C. Blum e A. Roli, "Metaheuristics in combinatorial optimization: Overview and conceptual comparison," *ACM computing surveys (CSUR)*, vol. 35, no. 3, pp. 268–308, 2003.

[72] E. Emary, H. M. Zawbaa, e M. Sharawi, "Impact of lèvy flight on modern meta-heuristic optimizers," *Applied Soft Computing*, vol. 75, pp. 775–789, 2019.

[73] B. Basturk, "An artificial bee colony (abc) algorithm for numeric function optimization," in *IEEE Swarm Intelligence Symposium, Indianapolis, IN, USA, 2006*, 2006.

[74] X.-S. Yang, "A new metaheuristic bat-inspired algorithm," in *Nature inspired cooperative strategies for optimization (NICSO 2010)*. Springer, 2010, pp. 65–74.

[75] ——, "Firefly algorithm, stochastic test functions and design optimisation," *International Journal of Bio-Inspired Computation*, vol. 2, no. 2, pp. 78–84, 2010.

[76] E. Rashedi, H. Nezamabadi-Pour, e S. Saryazdi, "Gsa: a gravitational search algorithm," *Information sciences*, vol. 179, no. 13, pp. 2232–2248, 2009.

[77] O. K. Erol e I. Eksin, "A new optimization method: big bang–big crunch," *Advances in Engineering Software*, vol. 37, no. 2, pp. 106–111, 2006.

[78] S. Mirjalili, S. M. Mirjalili, e A. Hatamlou, "Multi-verse optimizer: a nature-inspired algorithm for global optimization," *Neural Computing and Applications*, vol. 27, no. 2, pp. 495–513, 2016.

[79] A. E. Eiben e C. A. Schippers, "On evolutionary exploration and exploitation," *Fundamenta Informaticae*, vol. 35, no. 1-4, pp. 35–50, 1998.

[80] J. Kennedy e R. C. Eberhart, "Particle swarm optimization, ieee international of first conference on neural networks," 1995.

[81] J. Kennedy, "Particle swarm optimization," *Encyclopedia of machine learning*, pp. 760–766, 2010.

[82] Eberhart e Y. Shi, "Particle swarm optimization: developments, applications and resources," in *Proceedings of the 2001 congress on evolutionary computation (IEEE Cat. No. 01TH8546)*, vol. 1. IEEE, 2001, pp. 81–86.

[83] I. Newton, *Philosophiae naturalis principia mathematica*. G. Brookman, 1726, vol. 1.

[84] J. Kennedy e R. C. Eberhart, "A discrete binary version of the particle swarm algorithm," in *IEEE International conference on systems, man, and cybernetics. Computational cybernetics and simulation*, vol. 5. IEEE, 1997, pp. 4104–4108.

[85] E. Rashedi, H. Nezamabadi-Pour, e S. Saryazdi, "Bgsa: binary gravitational search algorithm," *Natural Computing*, vol. 9, no. 3, pp. 727–745, 2010.

[86] S. Mirjalili e A. Lewis, "S-shaped versus v-shaped transfer functions for binary particle swarm optimization," *Swarm and Evolutionary Computation*, vol. 9, pp. 1–14, 2013.

[87] X.-S. Yang, *Nature-inspired optimization algorithms*. Elsevier, 2014.

[88] E. G. Talbi, "A taxonomy of hybrid metaheuristics," *Journal of heuristics*, vol. 8, no. 5, pp. 541–564, 2002.

[89] S. Mirjalili e S. Z. M. Hashim, "A new hybrid psogsa algorithm for function optimization," in *2010 international conference on computer and information application*. IEEE, 2010, pp. 374–377.

[90] S. Mirjalili e A. Lewis, "Adaptive gbest-guided gravitational search algorithm," *Neural Computing and Applications*, vol. 25, no. 7-8, pp. 1569–1584, 2014.

[91] S. Mirjalili, G.-G. Wang, e L. d. S. Coelho, "Binary optimization using hybrid particle swarm optimization and gravitational search algorithm," *Neural Computing and Applications*, vol. 25, no. 6, pp. 1423–1435, 2014.

[92] G. Vandersteen, "Identification of linear and nonlinear systems in an errors-in-variables least squares and total least squares framework," Tese de Doutorado, Vrije Universiteit Brussel, 1997.

[93] J. Schoukens e L. Ljung, "Wiener-hammerstein benchmark," 2009.

[94] Y. Guo, L. Z. Guo, S. A. Billings, e H.-L. Wei, "Ultra-orthogonal forward regression algorithms for the identification of non-linear dynamic systems," *Neurocomputing*, vol. 173, pp. 715–723, 2016.

[95] R. A. Adams e J. J. F. Fournier, *Sobolev spaces*. Elsevier, 2003, vol. 140.

[96] C. d. S. Cassini, "Estimação recursiva de características estáticas não lineares utilizando modelos polinomiais NARMAX," Dissertação de Mestrado, Universidade Federal de Minas Gerais (UFMG), 1999.

[97] S. A. M. Martins, A. M. Barbosa, e E. G. Nepomuceno, "Determinaçao do número de termos de modelos polinomiais: O Critério de Informaçao de Akaike Multiobjetivo," in *Anais do XI Simpósio Brasileiro de Automaçao Inteligente*, 2013.

[98] B. H. G. Barbosa, L. A. Aguirre, C. B. Martinez, e A. P. Braga, "Black and gray-box identification of a hydraulic pumping system," *IEEE Transactions on control systems technology*, vol. 19, no. 2, pp. 398–406, 2011.

[99] B. O. S. Teixeira e L. A. Aguirre, "Using uncertain prior knowledge to improve identified nonlinear dynamic models," *Journal of Process Control*, vol. 21, no. 1, pp. 82–91, 2011.

[100] S. A. M. Martins e E. G. Nepomuceno, "Distribuição uniforme das soluções pareto-Ótimo na identificação multiobjetivo de sistemas," in *Congresso Brasileiro de Automática*, 2016.

[101] M. F. S. Barroso, R. H. C. Takahashi, e L. A. Aguirre, "Multi-objective parameter estimation via minimal correlation criterion," *Journal of Process Control*, vol. 17, no. 4, pp. 321–332, 2007.

[102] A. A. Campos, L. M. Margoti, A. P. L. Santos, N. R. L. Milagres, G. F. V. Amaral, e M. F. S. Barroso, "Identificação bi-objetivo para reator de polimerização contínuo utilizando critério de correlação mínima," in *Congresso Brasileiro de Automática*, 2010.

[103] S. A. M. Martins, E. G. Nepomuceno, e J. P. Figueiredo, "Detecção de estruturas de modelos NARX polinomiais: Uma abordagem inteligente multi-objetivo." in *Anais do X Simpósio Brasileiro de Automaçao Inteligente*, 2011, pp. 320–325.

[104] A. H. Ribeiro e L. A. Aguirre, "Relaçoes estáticas de modelos narx miso e sua representaçao de hammerstein," in *XX Congresso Brasileiro de Automática*, 2014.

[105] M. J. Lacerda, S. A. M. Martins, e E. G. Nepomuceno, "Structure selection based on interval predictor model for recovering static non-linearities from chaotic data," *IET Control Theory & Applications*, vol. 12, no. 13, pp. 1889–1894, 2018.

[106] X. Tang, L. Zhang, e X. Wang, "Sparse augmented lagrangian algorithm for system identification," *Neurocomputing*, vol. 330, pp. 403–411, 2019.

[107] F. Hafiz, A. Swain, e E. Mendes, "Orthogonal floating search algorithms: From the perspective of nonlinear system identification," *arXiv preprint*, 2019.

[108] L. A. Aguirre, B. H. G. Barbosa, e A. P. Braga, "Prediction and simulation errors in parameter estimation for nonlinear systems," *Mechanical Systems and Signal Processing*, vol. 24, no. 8, pp. 2855–2867, 2010.

[109] F. A. P. M. Bianchi, F. e L. Piroddi, "A randomised approach for narx model identification based on a multivariate bernoulli distribution," Dissertação de Mestrado, Politecnico di Milano, 2017.

[110] J. P. Nöel e M. Schoukens, "F-16 aircraft benchmark based on ground vibration test data," 2017, pp. 19–23.

[111] D.-G. P. F. Aguirre, L. A. e R. Santos-Filho, "Use of a priori information in the identification of global nonlinear models-a case study using a buck converter," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 47, no. 7, pp. 1081–1085, 2000.

[112] R. Bouc, "Forced vibrations of mechanical systems with hysteresis," in *Proc. of the Fourth Conference on Nonlinear Oscillations, Prague, 1967*, 1967.

[113] Y. X. Wen, "Method for random vibration of hysteretic systems," *Journal of the engineering mechanics division*, vol. 102, no. 2, pp. 249–263, 1976.

[114] L. S. Hosmer Jr, D. W. e R. X. Sturdivant, *Applied logistic regression*. John Wiley & Sons, 2013, vol. 398.

[115] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[116] N. Cristianini e J. Shawe-Taylor, *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press, 2000.

[117] M. Kuhn e K. Johnson, *Applied predictive modeling*. Springer, 2013, vol. 26.

[118] J. Pallant, *SPSS survival manual*. McGraw-Hill Education (UK), 2013.

[119] L. Bottou, "Stochastic gradient descent tricks," in *Neural networks: Tricks of the trade*. Springer, 2012, pp. 421–436.

[120] S. A. Billings e H.-L. Wei, "An adaptive orthogonal search algorithm for model subset selection and non-linear system identification," *International Journal of Control*, vol. 81, no. 5, pp. 714–724, 2008.

[121] P. A. Estévez, C. M. Held, C. A. Holzmann, C. A. Perez, J. P. Pérez, J. Heiss, M. Garrido, e P. Peirano, "Polysomnographic pattern recognition for automated classification of sleep-waking states in infants," *Medical and Biological Engineering and Computing*, vol. 40, no. 1, pp. 105–113, 2002.

[122] F. Miraglia, F. Vecchio, P. Bramanti, e P. M. Rossini, “EEG characteristics in “eyes-open” versus “eyes-closed” conditions: small-world network architecture in healthy aging and age-related brain degeneration,” *Clinical Neurophysiology*, vol. 127, no. 2, pp. 1261–1268, 2016.

[123] B. G. A. D. R. P. D. S. Harrewijn, A. e N. A. Fox, “S8. influence of behavioral inhibition and EEG resting state on social anxiety symptoms,” *Biological Psychiatry*, vol. 83, no. 9, pp. S349–S350, 2018.

[124] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, e I. H. Witten, “The weka data mining software: an update,” *ACM SIGKDD explorations newsletter*, vol. 11, no. 1, pp. 10–18, 2009.

[125] A. Asuncion e D. Newman, “UCI machine learning repository,” 2007.