UNIVERSIDADE FEDERAL DE LAVRAS

FREDERICO LUCAS DE OLIVEIRA MOTA

A MULTI-OBJECTIVE MGGP GREY-BOX IDENTIFICATION APPROACH TO DESIGN SOFT SENSORS

LAVRAS – MG

2022

FREDERICO LUCAS DE OLIVEIRA MOTA

A MULTI-OBJECTIVE MGGP GREY-BOX IDENTIFICATION APPROACH TO DESIGN SOFT SENSORS

Dissertação apresentada à Universidade Federal de Lavras, como parte das exigências do Programa de Pós-Graduação em Engenharia de Sistemas e Automação, para a obtenção do título de Mestre.

Prof. DSc. Bruno Henrique Groenner Barbosa
Orientador

LAVRAS – MG
2022

Ficha catalográfica elaborada pelo Sistema de Geração de Ficha Catalográfica da Biblioteca Universitária da UFLA, com dados informados pelo(a) próprio(a) autor(a).

Mota, Frederico Lucas de Oliveira.
A multi-objective MGGP grey-box identification approach to design soft sensors / Frederico Lucas de Oliveira Mota. - 2022.
118 p. : il.

Orientador: Prof. DSc. Bruno Henrique Groenner Barbosa.

Dissertação (mestrado acadêmico) - Universidade Federal de Lavras, 2022.
Bibliografia.

1. Identificação de Sistemas. 2. MGGP 3. Caixa-cinza. I. Barbosa, Bruno Henrique Groenner. II. Título.

FREDERICO LUCAS DE OLIVEIRA MOTA

A MULTI-OBJECTIVE MGGP GREY-BOX IDENTIFICATION APPROACH TO DESIGN SOFT SENSORS
UMA ABORDAGEM DE IDENTIFICAÇÃO CAIXA-CINZA MGGP MULTI-OBJETIVA PARA PROJETO DE SENSORES VIRTUAIS

Dissertação apresentada à Universidade Federal de Lavras, como parte das exigências do Programa de Pós-Graduação em Engenharia de Sistemas e Automação, para a obtenção do título de Mestre.

APROVADA em 31 de Janeiro de 2022.

Prof. DSc. Bruno Henrique Groenner Barbosa	UFLA
Prof. DSc. Leandro Freitas de Abreu	IFMG
Prof. DSc. Danton Diego Ferreira	UFLA

Prof. DSc. Bruno Henrique Groenner Barbosa
Orientador

LAVRAS – MG
2022

ACKNOWLEDGMENTS

To God who, through grace, allowed this realization in my life, guiding every step, giving me strength and taking care of every detail.

To my parents Bernadete and Roberto, for all the support, encouragement, attention and love that have always been essential.

To my brother Pedro, for his complicity, companionship and support in difficult times.

To my friend Gustavo Carvalho, who was with me during this journey, sharing the difficult moments but also the joyful ones.

To Professor Bruno, for the great guidance, encouragement, patience and teachings that were important during the making of this work, which contributed positively to my academic growth.

To my colleague Henrique Castro, who was always available to help me with the most varied doubts in the development of this work.

I thank the Federal University of Lavras (UFLA), especially the Master's Program in Systems Engineering and Automation, for the infrastructure necessary to carry out this work.

To all those who directly or indirectly contributed to the completion of this work.

My sincere thanks to everyone.

O presente trabalho foi realizado com apoio da Fundação de Amparo à Pesquisa de Minas Gerais (FAPEMIG).

"Just keep in mind: the more we value things outside our control, the less control we have."
— Epictetus

RESUMO

A extração *offshore* de petróleo é um processo complexo sendo necessários diversos instrumentos para controlar a produção nos poços. Dentre vários, o sensor *Permanent Downhole Gauge* (PDG), localizado dentro da coluna de produção, é utilizado para aferir a pressão e temperatura do poço de petróleo. Este sensor é submetido à condições extremas de operação, resultando em uma vida útil curta. A troca ou manutenção deste sensor é raramente feita pois o mesmo é de difícil acesso e exige que a produção seja paralisada. Assim, objetivando superar o problema de produção sem os dados do sensor PDG, o uso de *Soft Sensors* (SSs) surge como uma alternativa. Os SS são modelos matemáticos capazes de estimar uma variável de algum processo por meio de outras variáveis como entrada. Neste projeto é proposto o uso da metodologia de identificação de sistemas (*i.e.*, *i.* Testes dinâmicos, coleta de dados; *ii.* Escolha da representação matemática do modelo; *iii.* Seleção de estruturas para o modelo; *iv.* Estimação de parâmetros; e *v.* Validação do modelo.) com o fito de modelar um SS a fim de estimar a saída de um sensor PDG, mas não se limitando a esta aplicação, a qual é utilizada como motivação. Na etapa *ii.* da metodologia, a representação polinomial *Nonlinear Autoregressive with Exogenous Inputs* (NARX) foi escolhida. Para a etapa *iii.* é proposta uma abordagem multi-objetiva, por meio do algoritmo evolucionário *Multi-Gene Genetic Programming* (MGGP), para realizar a tarefa de seleção de estruturas dos modelos NARX. Três objetivos são minimizados, sendo eles: *i.* erro de predição um passo à frente (regime dinâmico), *ii.* erro em regime estático (é utilizada uma abordagem que reduz o custo computacional), e *iii.* o número de regressores do modelo. Na etapa *iv.* é proposta a estimação de parâmetros por meio dos mínimos quadrados ponderados, que utiliza informação do regime dinâmico e estático (informação auxiliar). Por fim, os modelos encontrados nos conjuntos Pareto-ótimos são validados (etapa *v.*) em simulação livre (em ambos os regimes) e um critério de decisão para selecionar o modelo mais adequado é aplicado. A fim de validar a metodologia proposta, três experimentos são feitos. O primeiro utiliza um banco de dados de um sistema estocástico, em que diversas comparações de abordagens são feitas (*e.g.*, número de objetivos na função custo). Como resultado, é visto que a metodologia consegue encontrar os regressores e estimar os parâmetros do modelo corretamente, com um custo computacional menor que outras abordagens. Já o segundo experimento aplica a metodologia em um sistema de bombeamento hidráulico. O modelo encontrado se mostra competitivo em regime estático e dinâmico, além de ser parcimonioso. Enfim, a mesma metodologia é aplicada ao banco de dados do processo petroquímico, que possui como saída a pressão do PDG. O algoritmo proposto consegue selecionar um modelo, que possui um comportamento satisfatório em regime dinâmico quando comparado com outros trabalhos, com doze regressores e doze parâmetros. Isso demonstra que o MGGP multi-objetivo, utilizando informações auxiliares, é uma boa ferramenta para seleção de estruturas e estimação de parâmetros para modelos NARX.

Palavras-chave: *Soft Sensor*. Petróleo. Modelos NARMAX/NARX. Identificação de Sistemas. Seleção de Estrutura. Estimação de Parâmetros. MGGP.

ABSTRACT

Offshore oil extraction is a complex process, requiring several instruments to control the production in the wells. Among several, the Permanent Downhole Gauge (PDG) sensor, located inside the production column, is used to measure the pressure and temperature of the oil well. This sensor is subjected to extreme operating conditions, resulting in short service life. The replacement or maintenance of this sensor is rarely done as it is difficult to access and requires production to be stopped. Thus, aiming to overcome the production problem without PDG sensor data, the use of Soft Sensors (SSs) appears as an alternative. SS are mathematical models capable of estimating a process variable through other variables as input. In this project, it is proposed the use of the methodology of systems identification (*i.e.*, *i.* Dynamic tests, data collection; *ii.* Choice of the mathematical representation of the model; *iii.* Selection of structures for the model; *iv.* Estimation of parameters; and *v.* Model validation.) to model an SS in order to estimate the output of a PDG sensor but not limited to this application, which is used as motivation. In methodology step *ii.*, the Nonlinear Autoregressive with Exogenous Inputs (NARX) polynomial representation was chosen. For step *iii.* a multi-objective approach is proposed, using the evolutionary algorithm Multi-Gene Genetic Programming (MGGP) to perform the task of structure selection from NARX models. Three objectives are minimized, namely: *i.* one-step-ahead prediction error (dynamic regime), *ii.* steady-state error (an approach that reduces computational cost is used), and *iii.* the number of regressors in the model. In step *iv.* it is proposed to estimate the parameters through weighted least squares, which uses information from the dynamic and static regime (auxiliary information). Finally, the models found in the Pareto-optimal sets are validated (step *v.*) in free-run simulation (in both regimes), and a decision criterion to select the most adequate model is applied. In order to validate the proposed methodology, three experiments are carried out. The first uses a dataset of a stochastic system, in which several comparisons of approaches are made (*e.g.*, number of objectives in the cost function). As a result, it is seen that the methodology can find the regressors and estimate the model parameters correctly, with a lower computational cost than other approaches. The second experiment applies the methodology in a hydraulic pumping system. The model found is competitive in the static and dynamic regime, in addition to being parsimonious. Finally, the same methodology is applied to the petrochemical process dataset, whose output is the PDG pressure. The proposed algorithm selects a model that has a satisfactory behavior in dynamic regime compared to other works, with twelve regressors and twelve parameters. This demonstrates that the multi-objective MGGP, using auxiliary information, is a good tool for selecting structures and estimating parameters for NARX models.

Keywords: Soft Sensor. Oil. NARMAX/NARX models. Systems Identification. Structure Selection. Parameter Estimation. MGGP.

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

$\hat{\theta}$	estimated parameter vector
θ	general parameter vector
$\mathbf{q}^{-i}$	back-shift operator
$\mathbf{J}$	cost function
$\mathbf{J}(\cdot)$	generic cost function
^{T}	transpose operation
$\mathbb{R}$	set of real numbers
Θ	Pareto-optimal set
$\widehat{}$	estimated value
T_S	sampling period
$G(\,\cdot\,)$	dynamic linear model
$f(\,\cdot\,)$	generic function
u	input signal
y	output signal
w	noiseless output signal
F	polynomial function
n_y	maximum lag for output terms
n_u	maximum lag for exogenous input terms
τ_d	dead time
l	NARMAX model degree of nonlinearity
$\mathbb{N}$	set of natural numbers
n_e	maximum lag for noise signal
$\psi^T(k-1)$	regressor vector containing observations up to sample $k-1$
n_θ	parameter vector length
ξ	residuals vector
$\xi(k)$	residue at time k
$\langle\,.\,\rangle$	internal product of two vectors
$\mathscr{M}_i$	model obtained at the *i*-th iteration
$\nu(k)$	white noise
Ψ	matrix of regressors

$e(k)$	error at time k (noise signal)
$\mathbf{e}$	error vector
—	average value
Z_v	validation dataset
Z_d	dynamical training dataset
Z_s	static dataset
d_c	crowding distance

LIST OF ABBREVIATIONS AND ACRONYMS

AIA	Artificial Intelligence and Automation Research Group
AIC	Akaike Information Criterion
ANN	Artificial Neural Network
AR	Autoregressive
ARMAX	AutoRegressive Moving Average with eXogenous inputs model
ARX	AutoRegressive with eXogenous inputs model
BHP	BottomHole Pressure
CI	Computational Intelligence
CLS	Constrained least squares
DEAP	Distributed Evolutionary Algorithms in Python
EA	Evolutionary Algorithms
EC	Evolutionary Computing
ELS	Extended Least Squares
EP	Evolutionary Programming
ERR	Error Reduction Ratio
ES	Evolutionary Strategie
ESP	Electrical Submersible Pumping
FBA	Freitas, Barbosa and Aguirre (2021) approach for static data
FPE	Final Prediction Error
FROE	Forward Regression Orthogonal Estimator
GA	Genetic Algorithms
GP	Genetic Programming
HV	Hypervolume indicator
k-SPXY	Kernel-based algorithm SPXY
kFCV	k-Fold Cross-Validation
KS	Kennard-Stone algorithm
LS	Least Squares
MDL	Minimum Description Length
MERR	Multi-objective Error Reduction Ratio
MGGP	Multi-Gene Genetic Programming

MGGPMO	multi-objective approach of MGGP
MISO	multiple-input single-output
MLP	MultiLayer Perceptron
MLR	Multivariate Linear Regression
MOEAs	Multi-Objective Evolutionary Algorithms
MSE	Mean Squared Error
MSSE	Mean Square Simulation Error
NARMAX	Nonlinear AutoRegressive Moving Average model with eXogenous inputs
NARX	Nonlinear AutoRegressive model with eXogenous inputs
NSGA-II	Non-dominated Sorting Genetic Algorithm II
PCP	Progressive Cavity Pumps
PDG	Permanent Downhole Gauge
PEM	Prediction Error Minimization
PIMS	Plant Information Management System
PRBS	Pseudo-Random Binary Sequence
PT	Pressure Transmitter
PTT	Combined Pressure Temperature Transmitter
RBF	Radial Basis Function
RMSE	Root Mean Squared Error
SDV	Shutdown Valve
SEM	Simulation Error Reduction
SEMP	Simulation Error Minimization with Pruning
SISO	single-input single-output
SPEA-II	Strength Pareto Evolutionary Algorithm II
SPU	Stationary Production Unit
SPXY	Sample set Partitioning based on joint x–y distances algorithm
SPXYE	Error based SPXY algorithm
SRR	Simulation Error Reduction Ratio
SS	Soft Sensor
STLF	Short Term Load Forecasting
TT	Temperature Transmitter

WGN	White Gaussian Noise
WLS	Weighted Least Squares

CONTENTS

1 INTRODUCTION

1.1 Motivation

The economical interest for oil started at the beginning of the nineteenth century when it was used for lighting. However, only during the twentieth century, the exploration of petroleum started to be economically justified due to the gas and diesel motor invention, which created an enormous demand for it (NETO; COSTA, 2007). After this fact, oil starts to be part of people's daily life and is present in almost all production chains as a raw material in industrial process. Examples of products that show this relevance of oil in society are gasoline, kerosene, lubricants, asphalt, different types of plastics, silicone, medicines, and many other products.

According to the U. S. ENERGY INFORMATION ADMINISTRATION (2020), in 2019, the total world petroleum production reaches 100.65 million barrels per day (b/d), and the total world petroleum consumption was 101.04 million barrels per day (b/d), which represents 1% of growth when compared with 2018. Concerning Brazil, in 2019, 2.877 million barrels per day (b/d) were produced, which represents a 7.4% growth when compared with 2018. This fact places Brazil as the 10th largest oil producer in the world (AGÊNCIA NACIONAL DO PETRÓLEO GÁS NATURAL E BIOCOMBUSTÍVEIS, 2020).

In order to produce oil, it is necessary to drill an oil well to access the reservoir rocks in the subsoil, where the oil is stored. This process can be done inland or in the ocean. When the oil reservoir is inland the extraction is called onshore. In the opposite direction, when the process is performed in deep water, *i.e.*, the reservoir is in the ocean, the extraction process is called offshore. With regard to Brazilian reserves, most are located offshore. Offshore oil production is a challenging process, since a complex arrangement of instruments, platforms, and connections are necessary to produce with quality and safety. This arrangement is composed of Stationary Production Units (SPUs); risers and flowlines; manifolds; a Christmas tree; and the production column, which is placed between the oil reservoir and the Christmas tree.

Under the production column, close to the reservoir, the permanent downhole gauge (PDG) sensor is placed. The PDG sensor is used to measure temperature and pressure, which helps the real-time monitoring and control of the oil well. This sensor is submitted to many extreme conditions (*e.g.*, salinity, high pressure, etc.) which shorten its life expectancy. There is a 69% probability of a PDG system surviving 5 years, in other words, 31% of all PDG systems fail within 5 years in operation (FROTA; DESTRO, 2006). In addition to that, maintenance of failed sensors is hard to be performed since the sensor position in the well is not easy to access and it is necessary to halt production to do this, which means substantial economic losses. Therefore, replacement rarely occurs in practice even if data is completely missing or corrupted (DAVIES; AGGREY et al., 2007).

Considering the real need to obtain information about the oil well to control production, even with the PDG sensor inoperative or damaged, the soft sensors are a good alternative to hardware sensors. Soft sensors are mathematical models capable of estimating an unmeasured variable by using information from other process variables (FORTUNA et al., 2007b; KADLEC; GABRYS; STRANDT, 2009). Therefore, they can be used to estimate the PDG output variable, help in fault detections, and even substitute the sensor in the case of total failure. To create these sensors, the system identification problem methodology (AGUIRRE, 2015) for grey and black models may be used. The soft sensors have been applied in many different industrial processes presenting good results (see *e.g.*, Bhavani et al. (2014), Sujatha et al. (2018), Rizzo (2010), Radhakrishnan and Mohamed (2000)).

Likewise, the virtual sensors were also applied in the oil industry. Macias, Angelov and Zhou (2006), for example, applied virtual sensors to quality prediction of crude oil distillation in a refinery process. Regarding offshore oil extraction, to solve the PDG sensors failure problem, several works have implemented soft-sensors to estimate downhole pressure (see *e.g.*, Barbosa et al. (2015), Aguirre et al. (2017), Morais et al. (2019), Apio et al. (2019)). In this context, it is extremely important to study each stage of the system identification process (*i.e.*, *i.* dynamic tests, *ii.* choice of mathematical representation, *iii.* model structure determination, *iv.* parameter estimation, and *v.* model validation) to choose the right tools and methods to produce virtual sensors.

Most problems encountered in real systems, oil extraction included, are non-linear. For this reason, Nonlinear Autoregressive with Exogenous Inputs (NARX) (LEONTARITIS; BILLINGS, 1985a) models are a flexible tool widely used in the representation of models in systems identification. When using this type of representation, the two main problems are the selection of structure (regressors) and estimating parameters for this model. Several classical techniques are addressed to solve these problems, such as the Forward Regression Orthogonal Estimator (BILLINGS; CHEN; KORENBERG, 1989), based on Error Reduction Ratio (BILLINGS; CHEN; KORENBERG, 1989), for selecting structures. However, approaches based on these principles may suffer from problems such as the curse of dimensionality.

In this sense, several alternative methods have been presented, such as algorithms based on the evolutionary process, *i.e.*, evolutionary algorithms (*e.g.*, Genetic Algorithms (GAs) (GOLDBERG; HOLLAND, 1988; HOLLAND, 1975) and Genetic Programming (GP) (KOZA, 1992)), examples of this type of application are Chen et al. (2007) and Madár, Abonyi and Szeifert (2005). More specifically, seeking flexibility when searching for the best regressors for the models, the Multigene Genetic Programming (MGGP) (HINCHLIFFE et al., 1996; HINCHLIFFE, 2001) has been applied and has shown promising results. In this approach, using NARX representation, each gene of an individual in the population is a basis function (regressor) represented by a genetic program. This is advantageous because MGGP does not have a fixed size for the chromosome, only a maximum size fixed by the designer, which reduces the computational cost. Examples of works that use MGGP in modeling and predictions are Ghareeb and Saadany (2013), Niazkar and Niazkar (2020), and Riahi-Madvar et al. (2019).

Another point that has been explored to obtain better results during the selection of structures and parameter estimation is auxiliary information. Auxiliary information is understood as any missing extra information, for example, the static curve of a system when using its data in the dynamic regime for modeling. Freitas, Barbosa and Aguirre (2021), and Aguirre et al. (2004) exploited auxiliary information during modeling. One way to use auxiliary information during the selection of structures of a NARX model is to implement more than one objective in the cost function, *i.e.*, a multi-objective approach, where, for example, the first objective is to minimize the error in the dynamic regime and the second is minimize the error in the static regime. Hafiz, Swain and Mendes (2020) present a multi-objective framework for structure selection for nonlinear polynomial systems, where several evolutionary algorithms are submitted to different tests with qualitative and quantitative parameters. It is found that these algorithms can find suitable structures for nonlinear systems. Other works that use a multi-objective approach and an evolutionary optimization approach for systems identification are Castro and Barbosa (2019) and Mota et al. (2020).

In this work, the flexibility of the MGGP is explored with a multi-objective approach that uses auxiliary information about the static regime in the structure selection and parameter estimation.

1.2 Objectives

The general objective of this work is to use the methodology of systems identification through evolutionary computation techniques to model real systems using auxiliary information in a multi-objective approach.

The specific objectives consist of:

1. implement virtual sensors in order to estimate the output of a permanent downhole gauge sensor in an offshore oil extraction process;

2. implement other forms of simulation for the static regime with lower computational cost;

3. implement the use of auxiliary information in structure selection and parameter estimation of NARX models.

1.3 Work Structure

This document is divided into seven Chapters, including this introductory. Chapter 2 provides an overview of the oil extraction process, where system identification techniques can be applied. Moreover, some techniques and instruments used in the oil extraction process are detailed. Chapter 3 reviews some fundamentals of System Identification, covering from data collection to model validation. The chapter subsections are mainly focused on classical model structure selection and parameter estimation techniques. Chapter 4 introduces basic concepts of evolutionary algorithms. A methodology to achieve the objectives of this project is displayed in Chapter 5. In Chapter 6 results of this work are presented and discussed. Finally, in Chapter 7 the final considerations are presented together with future works.

2 PROCESS DESCRIPTION

2.1 Introduction

Oil and its derivatives have great historical importance on human society. It's almost impossible to find one production chain which does not use some form or derivative of oil as a raw material in its industrial process. Examples of products that show this relevance of oil in society are gasoline, kerosene, lubricants, asphalt, different types of plastics, silicone, medicines, and many other products. As a consequence, oil has dominated the world's energy consumption since the last century (ZHANG; JI; FAN, 2015).

This chapter provides an overview of the oil extraction process, presenting the field of knowledge whose techniques of system identification and computational intelligence can be applied. Moreover, some techniques (*e.g.*, oil lift, gas lift) and instruments (*e.g.*, Christmas tree, Permanent Downhole Gauge) used in the oil extraction process are detailed.

2.2 Oil Well's Construction Process

The well's construction process enabling the extraction of oil has many steps. The first one is prospection. This has the objective to find a sedimentary basin with the right geologic situations to contain oil (THOMAS, 2004). This is made by investigating the soil and subsoil with geological and geophysical methods. The well drilling process begins after all the study is done, and due to its high cost, all these analyses take an important place in the process.

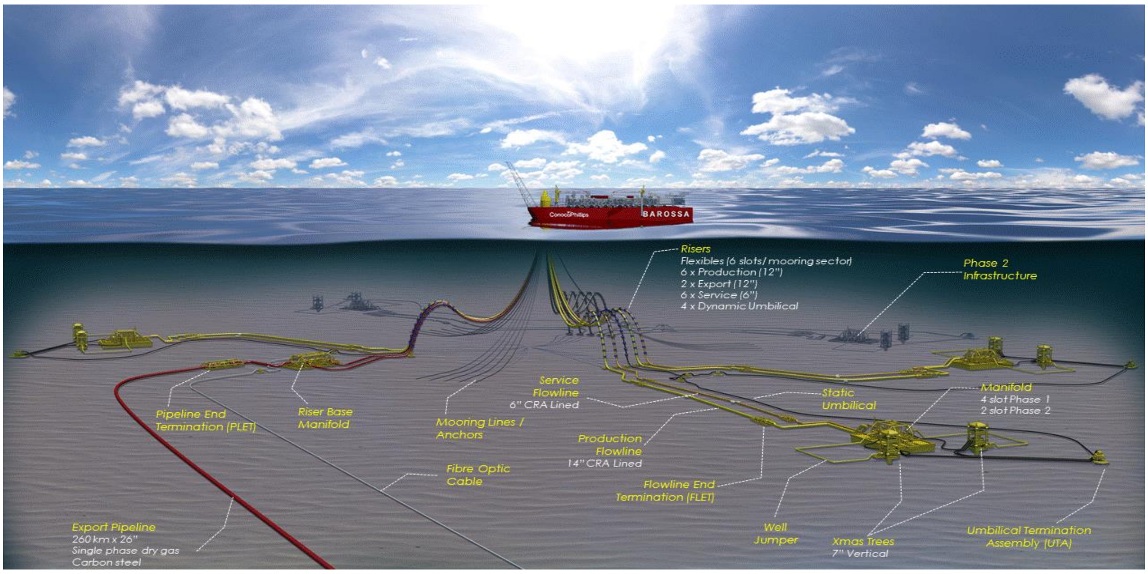
After finishing the drilling process, the third step, known as well completion, begins. This step is responsible to equip the well for extraction of oil or gas in a safe and economically viable way (THOMAS, 2004). Finally, after all these steps, the oil elevation process begins. In the next subsections, some of these steps will be developed more deeply. Figure 2.1 shows an overview example of the oil production system resulted after all these steps.

2.2.1 Well Completion

After finishing the drilling process it is necessary to prepare the well for a safe and economically viable production over its productive life (THOMAS, 2004) and to do this, a set of techniques called completion, is required.

The completion allows the connection between the hydrocarbons wells and the reservoirs (*e.g.*, Stationary Production Unit (SPU), vessels, ships, and platforms). These reservoirs

Figure 2.1 – An overview of an oil production system



Source: Adapted from Reid (2018)

are responsible for storing, managing, and in some cases, making primary processes of the product. They are connected to some instruments under the sea by risers, *i.e.*, suspended pipes, and flowlines, *i.e.*, pipes arranged on the seabed. These instruments are responsible for controlling the flow, artificial elevation, data acquisition, and other auxiliary functions (VILLELA, 2004).

Depending on the location of the oil reserves, different types of instruments (*e.g.*, wellhead systems and Christmas trees) are required, which lead to two kinds of subsea production systems. When the reserves are on land, onshore production, dry completion, also known as dry tree system, takes place and the wellhead system stays on the surface - this is also a reality in shallow waters (BAI; BAI, 2018). In this case, the Christmas tree used to control the well production is simple, easy to maintain and access (VILLELA, 2004).

The second possible situation is to extract oil from reservoirs in deep water, also called offshore. In this situation it is impossible to have a wellhead on the surface and wet completion comes about. For this system it is necessary to have a wet Christmas tree, *i.e.*, a more sophisticated submerged tree. Therefore, maintenance and access are a lot more complicated but, on the other hand, it allows the use of floating production units with greater movements (VILLELA, 2004).

Another way to classify the completion process is by the number of exploited areas. Using this aspect, it is possible to have simple or multiple exploited areas. The simple one occurs when just one connection is used between the well and the reservoir. This connection is a metal pipe called a production collum, and this type of completion makes it possible to control and explore just one area of interest (THOMAS, 2004).

On the other hand, the multiple completion process allows the exploration of two or more areas at the same time. This is compelling because it is possible to use a few number of wells to exploit the same area when compared to the simple completion. This leads to a more economical way of production. To look at it from a different angle, the probability of operational problems increases and it's harder to apply artificial methods of oil elevation (THOMAS, 2004).

2.2.2 Oil Lift

After making the completion, the oil lift step takes place, which consists of extracting the oil from the well's bottom to the surface. To do so it is first necessary to identify if the well is naturally flowing or not.

When the well is naturally flowing, its pressure sufficient to lift the oil directly to the surface and there is no need to apply any artificial method or pumps, in other words, it is a emergent well. On the other hand, when the well's pressure is not sufficient to naturally extract oil from the bottom to the surface the well's type is so-called non-emergent. This is also a reality to the naturally flowing wells because, over time, their energy decrease, and, for this reason, it is necessary to apply some artificial lift methods to maintain the production level (THOMAS, 2004). Choosing the artificial lift method for the well is not an easy task, especially if the production is offshore. The following are the main artificial lift techniques or methods (BAI; BAI, 2018; THOMAS, 2004):

a) Subsea Boosting;

b) Electrical submersible pumping (ESP);

c) Progressive Cavity Pumps (PCP);

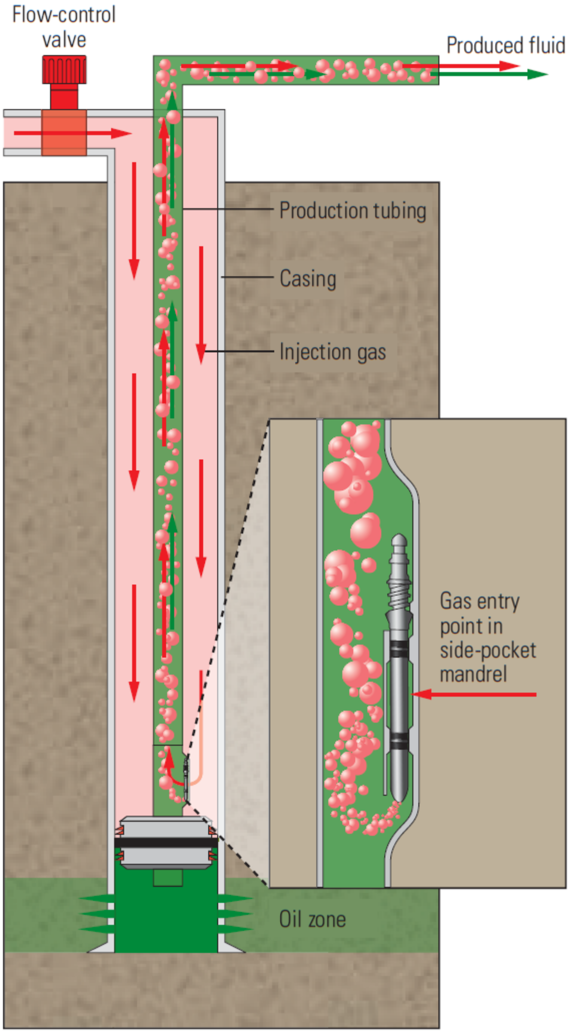
d) Intermittent-flow and Continuous-flow gas lift.

The last one, gas lift, will be more detailed in the next subsection because it is a widely employed method used for deepwater mature oil wells (JADID; OPSAL; WHITE, 2006).

2.2.3 The Gas-lift

Gas lift is an artificial lift method widely used in the offshore production environment (BAI; BAI, 2018). This method consists of using an external source of energy, more precisely a high-pressure gas, to lift the well fluids (*e.g.*, oil, water) from the bottom to the surface. This is done by injecting gas into the wellbore, typically between the casing and production tubing through a valve placed next to the well's bottom (as shown in Figure 2.2). This process generates bubbles that are mixed with the produced fluids making them less dense and in return, decreases the bottomhole pressure (BHP) that forces the well to push oil to the surface (JADID; OPSAL; WHITE, 2006).

Figure 2.2 – Example of a Gas lift system.

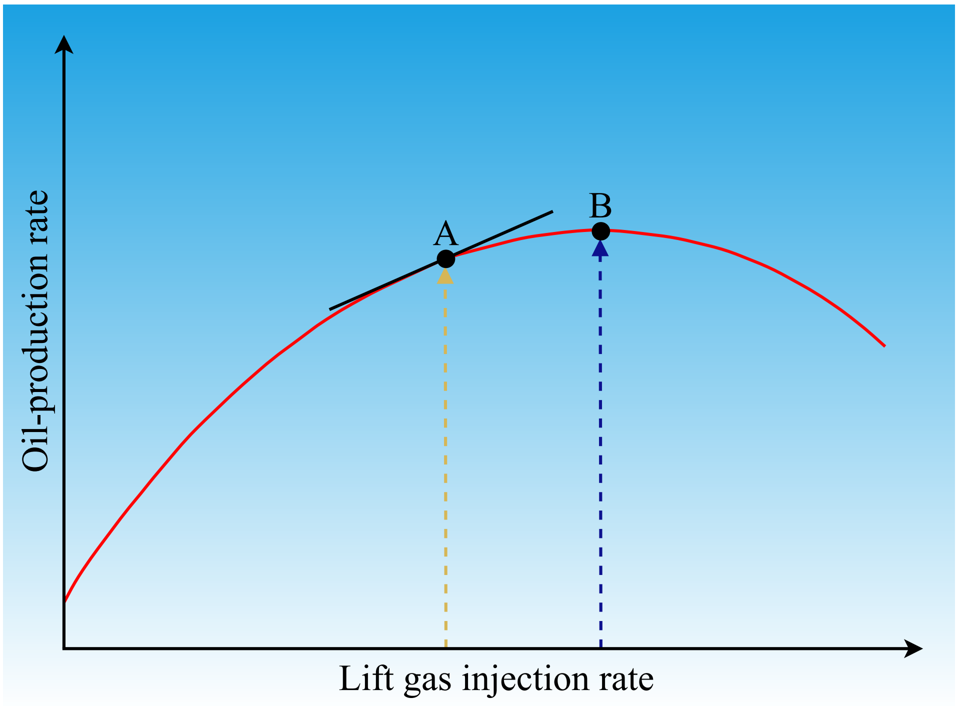


Source: Adapted from Jadid, Opsal and White (2006)

There are two basic types of gas lift systems — continuous flow and intermittent flow (ELLDAKLI, 2017). For the first technique, so-called continuous flow, gas is continuously injected into the production conduit at the maximum depth in the same proportion as the flow that comes from the reservoir to surface (THOMAS, 2004). For the other technique, intermittent gas lift is obtained by injecting gas in a discontinuous manner. When the high-pressure gas is injected below the fluid column, with correct volume and pressure, the oil gushes to the surface. A disadvantage of this process is the limitation of producing at a high volume rate compared to continuous flow, the advantage is that there is no need to inject high pressure gas continuously to produce (ELLDAKLI, 2017).

When projecting a gas lift system there are two important criteria: gas lift volume and gas lift pressure. The first one is responsible to control the production level of the oil well, the higher the gas lift volume increase, the more production increase. However, the production level has a limit, which varies depending on the well's structure (BAI; BAI, 2018). As can be seen in Figure 2.3, when the oil production rate reaches point *B*, a saturation threshold starts to decrease despite the rise of gas lift volume. The second criterion, gas lift pressure, influences the system operating pressure and the well's equipment specification, because of that it needs to be carefully determined (BAI; BAI, 2018).

Figure 2.3 – Ratio between oil-production rate and lift gas injection rate.



Source: Adapted from Jadid, Opsal and White (2006)

To manage all the gas consumed by in the gas lift system, a choke valve, localized on the surface, is used. Moreover, a set of sensors is employed to feed and control the system (*e.g.*, permanent downhole gauge).

2.3 Production System

The production system pipe is divided into two categories: production string and production pipelines. The production string is a pipe made of steel with a small diameter responsible for carrying the oil from the well's bottom to the surface, in the case of onshore production. On the other hand, in offshore production, the string leads the oil to the wet Christmas tree level, and after reaching this level the oil is conducted by the production pipelines to the surface, *i.e.*, stand-alone facility (VILLELA, 2004). Figure 2.4 shows an example of the production string.

The second category, production pipelines, is responsible for transporting the liquid (*e.g.*, oil, gas, etc.) from the well's head to the stand-alone facility, *i.e.*, reservoir (VILLELA, 2004). This category has two subcategories: risers and subsea flowlines.

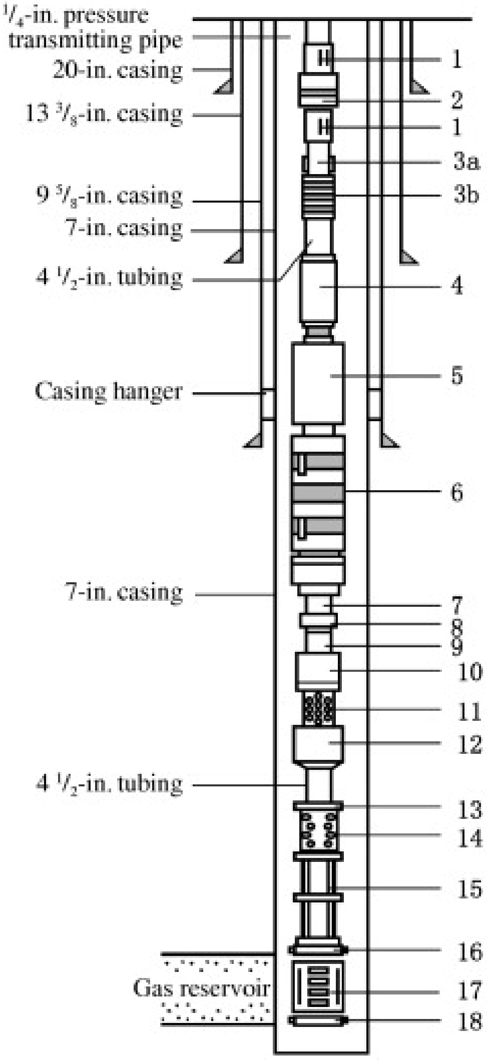
The production risers are the suspended part of the production system pipe, they reside between the host facility and the seabed. This part of the system is critical for a submerged production, because they are exposed to a large number of mechanical efforts as the sea current, waves, and host facility movements — to deal with these problems they can be flexible or rigid (VILLELA, 2004; BAI; BAI, 2018).

The second subcategory, subsea flowlines, are pipelines arranged on the seabed, used to make the connection between the wellhead and surface facility. They can make connections with manifolds to receive the production of multiple wells at the same time and redirect to the host facility (BAI; BAI, 2018). All the parts described above can be seen in Figure 2.5.

2.4 Instrumentation

In the previous subsections a group of equipment used in oil production was mentioned — some of them will be detailed in the next subsections to give a more complete understanding of the whole process.

Figure 2.4 – Detailed example of the production string.



No.	Downhole string component part	OD (mm)	ID (mm)
1	3 1/2-in. subsurface safety valve	146.05	72.14
2	Flow sub	146.21	121.36
3a	Depth correction nipple		
3b	Circulating valve	150.90	98.30
4	Telescopic joint		
5	Anchor		
6	Anchor packer (TM thread for all above)	151.13	98.55
7	Fishing extension pipe		
8	Thread change joint		
9	Nipple		
10	Upper pressure transmitting sub		
11	Perforated screen		
12	Lower pressure transmitting sub		
13	Profiled sub of TCP assembly		
14	Glass disc (starter)		
15	Intermediate nipple		
16	Mechanical or hydraulic ignition head		
17	127 gun, C48YD-4S charge, 30 shots/m		
18	Time-delay detonator		

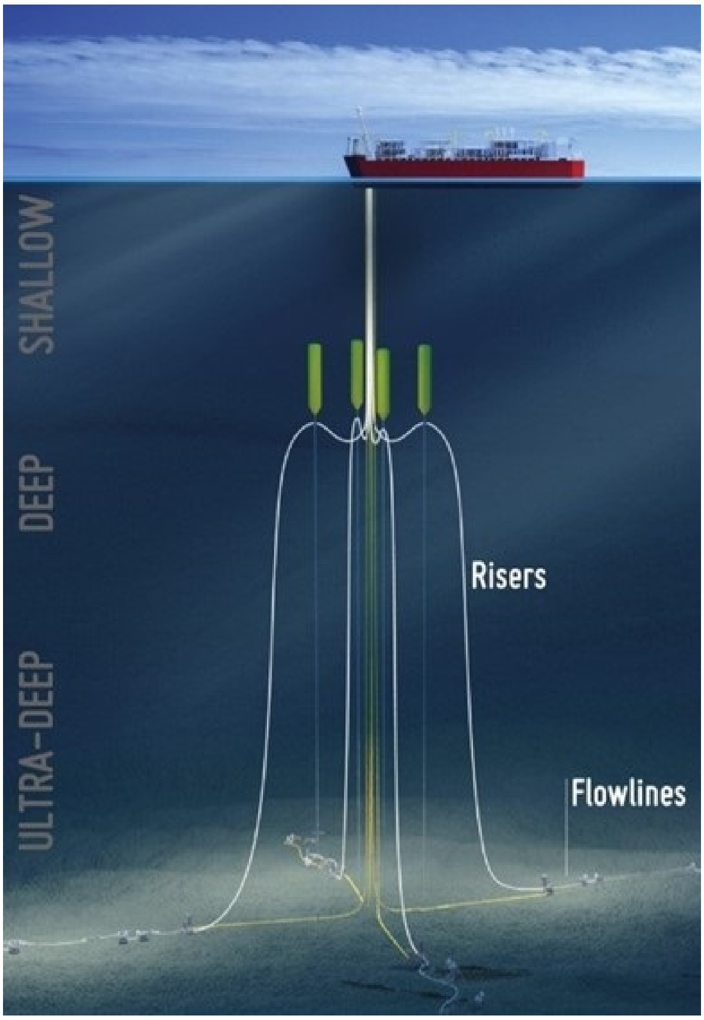
Source: Wan (2011)

2.4.1 Christmas Tree

The “Christmas tree”, or just “tree”, is an important tool and consist of valves, pipes, fittings, and connection assemblies responsible for controlling the production flow or injection in the well (VILLELA, 2004). Depending on location of oil field, offshore or onshore, the type of the tree can be dry or wet.

The dry Christmas tree, so-called Conventional Christmas tree, is used on the surface and is made of a set of gate valves (generally four or five arranged in a crucifix type pattern), which can be manual and/or actuated (hydraulic or even pneumatic) (THOMAS, 2004). The five valves mentioned above can be seen in Figure 2.6. There are two master valves, the lower ones, responsible for direct control of the well’s fluids flow rising to the surface; one production

Figure 2.5 – Example of production risers and subsea flowlines on the ocean.

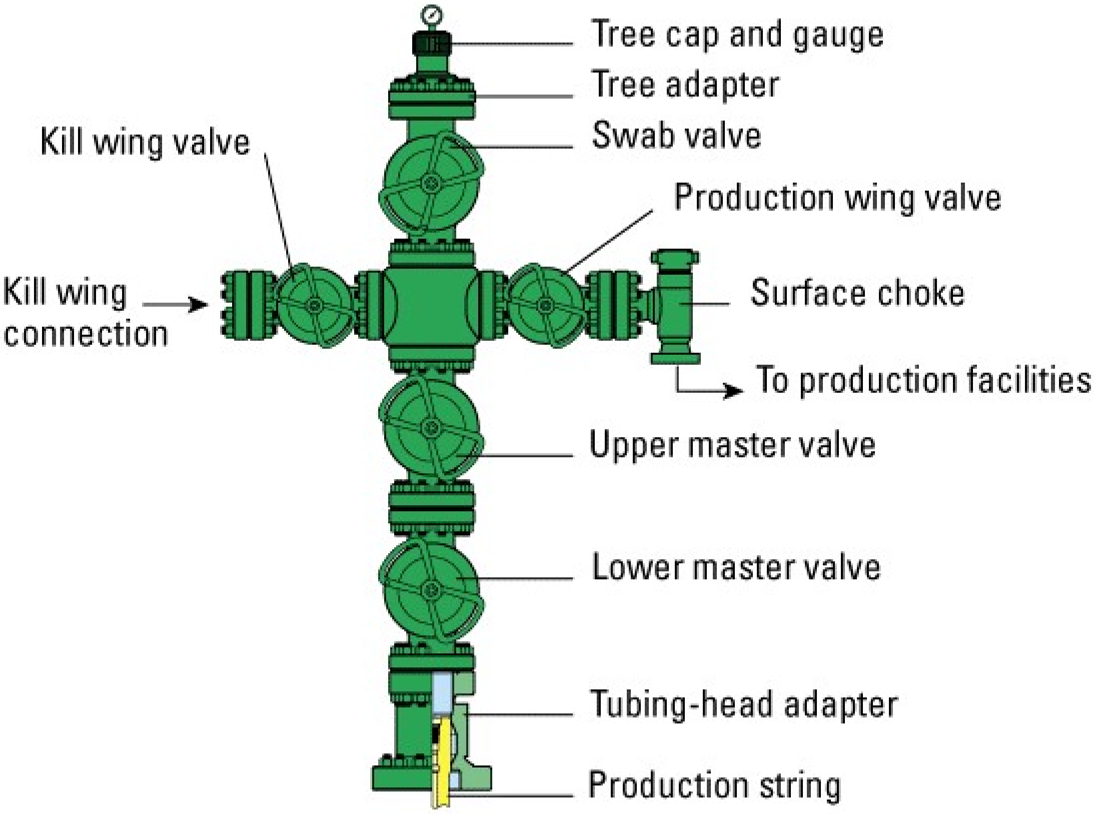


Source: Adapted from ArcelorMittal (2019)

wing valve, the right hand one, responsible for controling the flow of hydrocarbons to the reservoirs facilities; one kill wing valve, the left hand one, used for fluid injection (*e.g.*, corrosion inhibitors, methanol); one swab valve, at the top, responsible for, when opened, allows well interventions (*e.g.*, wireline, coiled tubing, down tools) (AMERICAN PETROLEUM INSTITUTE, 2010).

The other possible "tree" is the wet Christmas tree, so-called subsea Christmas tree, as its name suggests, the equipment is placed on the seabed and like the conventional Christmas tree is made of a set of gate valves plus a set of flow lines and a control system connected into the host facility (THOMAS, 2004). The tree valve's arrangement can define if the tree is vertical or horizontal — in the vertical type, all the valves are arranged vertically and, as expected, in the horizontal type all the valves are organized horizontally. The horizontal subsea tree is more

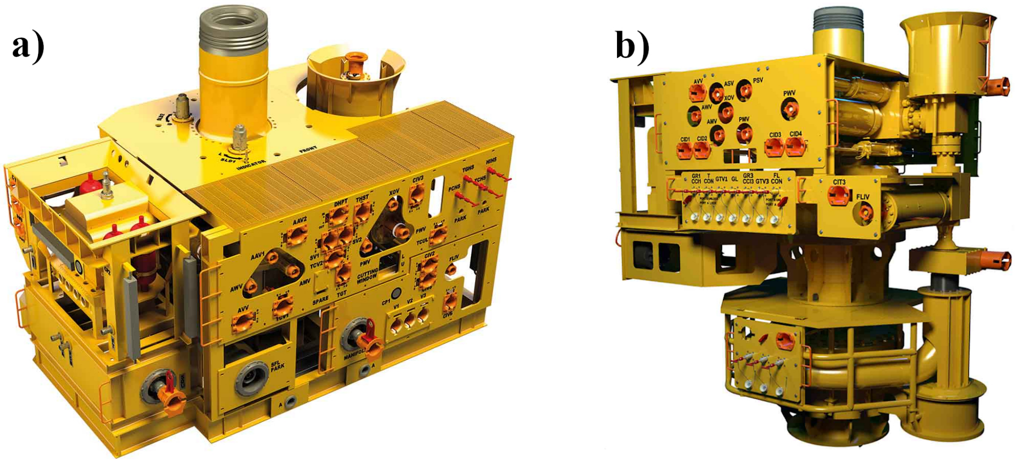
Figure 2.6 – A Conventional Christmas tree with its detailed valves.



Source: Adapted from Nor et al. (2019)

work-friendly due to the external position of the valves in relation to the center of the wellbore. Figure 2.7 shows a horizontal and a vertical Christmas tree.

Figure 2.7 – a) A Horizontal Christmas Tree; b) A Vertical Christmas Tree.



Source: a) Adapted from OneSubsea (2020); b) Adapted from OneSubsea (2018)

Both types of a tree have a common set of valves, and they are: one or two production master valves, depending on the tree type; the annulus master valve, responsible for closing or opening the annulus bore; the production wing valve; the swab valve and annulus access/swab valve; the annulus wing valve; the crossover valve, responsible for allowing flow between the annulus and production tree paths, when opened (BAI; BAI, 2018). In Table 2.1, the most notable differences between the subsea horizontal and vertical tree can be seen (KHALIFEH; SAASEN, 2020):

Table 2.1 – Notable differences between the subsea horizontal tree and subsea vertical tree.

Vertical Christmas Tree	Horizontal Christmas Tree
Master and swab valves in bore	No valves in the vertical bore of the well
Tubing hanger orients via wellhead	Tubing hanger orients directly from tree
Tubing hanger seals normally isolated from well fluid	The tubing hanger seals are continuously exposed to well fluids
External tree cap run after tree landed/-tested	An internal tree cap is used as a secondary pressure barrier above the tubing hanger, two crown plugs are installed by wireline unit

Source: Adapted from Khalifeh and Saasen (2020)

Also, the subsea trees have a lot of sensors (*e.g.*, Pressure Transmitter (PT), Temperature Transmitter (TT), Combined Pressure Temperature Transmitter (PTT), permanent downhole gauge (PDG), etc.) capable to measure the temperature, pressure, flow, noise, and other variables. All these sensors integrated by cables, connectors, and terminators are connected to the control panel at the well surface/host facility. This group of sensors provides a great quantity of data which makes the well a "smart well" allowing automatic adjustments and/or be controlled remotely by operators, without intervention using rigs or coiled tubing (CARVAJAL; MAUCEC; CULLICK, 2017). In this scenario, the permanent downhole gauge is considered as good equipment with tools to optimize production and give a much longer life to the oil field.

2.4.2 Permanent Downhole Gauge

The real-time monitoring system, called permanent downhole gauge (PDG), is responsible for measuring the temperature and pressure of the oil well. This system is installed at the bottom hole, close to the host facility/reservoir. All these data collected by the PDG, in real-time, are analyzed by the engineers to make the operational adjustments that guarantee the well's efficient production according to the ongoing changes at the time (FROTA; DESTRO, 2006). Many PDG systems have other types of sensors like flow rate, phase flow rate, phase fraction, resistivity (OUYANG; KIKANI et al., 2002). Figure 2.8 shows a schematic of a PDG.

The data that comes from the PDG system has a wide range of application and uses in industry, including (OUYANG; KIKANI et al., 2002):

a) reduce ambiguity and uncertainties in the interpretation;

b) detect the changes in reservoir properties, such as compaction;

c) monitor skin, permeability, pressure drawdown over time;

d) evaluate the performance of excitation or well workover jobs;

e) evaluate completion performance;

f) identify well problems quickly;

g) identify reservoir connectivity;

h) detect drainage area change;

i) evaluate operational efficiency;

j) improve the flow back time of new wells;

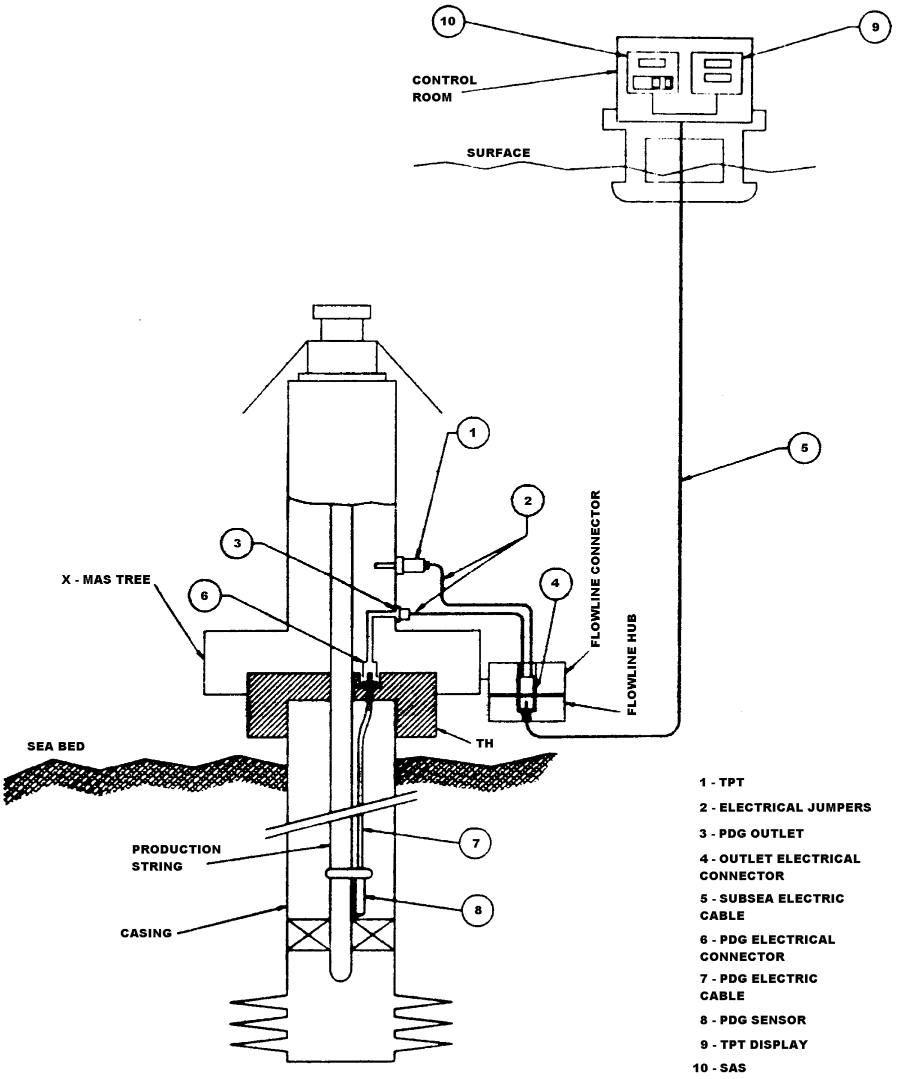
k) obtain Initial Build-up Data;

l) assist reservoir simulation and history matching.

The PDG systems suffer intense wear and according to Frota and Destro (2006), which analyzed 952 PDG systems installed between 1987 and 1998, there is a 69% probability of a PDG system surviving 5 years, in other words, 30% of all PDG systems fail before 5 years of operation. This fact leads to another problem: replacing the damaged system. This process of replacement or maintenance of the system's sensors is very difficult and, some times, impossible to perform due to the sensor's location in the well. When it is possible to carry out this procedure, the consequence is the stoppage of oil production, which leads to a great economic loss (BARBOSA et al., 2015).

As can be seen, the PDG system is very important for the process of management and optimization of oil production, however, it has a relatively short useful life and can suffer several failures during its operation. Due to the aforementioned problems, soft-sensors, and systems identification appear as interesting alternatives to increase the reliability of the sensor data and, when a permanent loss of the system occurs, replace it completely to keep the oil well operating.

Figure 2.8 – A schematic of the PDG Systems.



Source: Frota and Destro (2006)

3 SYSTEM IDENTIFICATION

3.1 Introduction

One of the greatest challenges of humankind, in the scientific scope, has been to understand the physical behavior of the processes observed in nature in order to obtain analogous systems. By analogous systems, it is understood as a system capable of mimicing static and dynamic behaviors of an observed phenomenon, and also predict its future behaviors. This analogous system can have a mathematical representation and it is called a mathematical model (AGUIRRE; RODRIGUES; JÁCOME, 1998).

To obtain these mathematical models several mathematical modeling techniques can be used. One of these techniques is called white-box modeling. It demands a deep knowledge of the system behavior (maximum *a priori* information) because it's based on first principles, *i.e.*, the model is obtained from mathematical relations that describe physical phenomena (*e.g.*, Bernoulli's equation, Newton equation, etc.). Unfortunately, in most practical situations all the knowledge, information, and time necessary are not available to apply this technique and develop the model from the equations that govern the physical process (AGUIRRE, 2015). Systems Identification appears as an alternative procedure that satisfactorily handles these limitations.

The system identification area differs from classical mathematical modeling techniques because it doesn't need, or hardly need, *a priori* information from the process. It is possible to obtain a mathematical model, which completely or partially explains the static and dynamic behavior of the system by just using the input data and its corresponding output data. As a consequence of this, these methods receive the name black-box modeling. The disadvantage of this approach is the lack of physical meaning of the models obtained and, sometimes, the great numbers of parameters.

Another way to obtain these models is located in between black-box modeling and white-box modeling, it is called grey-box modeling. In this approach, some *a priori* knowledge of the process is used to determine the model, but a significant part of its parameters are still estimated through observed data, *i.e.*, the input and output data of the process obtained experimentally. This approach is also known as semi-physical modeling (FORSSELL; LINDSKOG, 1997).

All these models obtained with mathematical modeling and system identification techniques can be linear or nonlinear. In the linear category, transfer function and time-series models predominate (ZHANG, 2010). Although the linear systems are simplistic, they can't mimic all the dynamic and static regimes behaviors (*e.g.*, chaos, bifurcations, etc.) of many real processes and practical situations. In this scenario, nonlinear models are better adapted as they manage to represent various operating ranges in static and dynamic regimes. This characteristic together with the rise of computational power and computational intelligence create new approaches to system identification.

One of these approaches is to see the system identification problem as an optimization problem, this can be made when the appropriate system representation (*e.g.*, Nonlinear Auto-Regressive MovingAverage with eXogenous inputs — NARMAX, Hammerstein, Neural Network) is already defined. The optimization problem can be mono-objective and multi-objective. A multi-objective problem can be defined as (NEPOMUCENO; TAKAHASHI; AGUIRRE, 2007):

$$\begin{cases} \hat{\theta} & = arg\min_{\theta} \mathbf{J}(\theta) \\ subject\ to: & \theta \in \mathbb{R}^n, \end{cases} \tag{3.1}$$

with the objective-functions $\mathbf{J}(\theta) = [J_1(\theta)\ \cdots\ J_m(\theta)]^T$, where $\mathbf{J}(\cdot) : \Re^n \mapsto \Re^m$. As said in Barbosa et al. (2011), the objective-functions should be conflicting, *i.e.*, a trade-off between the objectives ought exists and, instead of arriving at one solution, reach a set of solutions. As said in Nepomuceno, Takahashi and Aguirre (2007), in the solution set there is no unique model that together minimizes all the objectives in an optimal way. However, there is the Pareto-optimal set, which is comprised of the non-dominated solutions. The Pareto-optimal set, Θ, is defined as:

$$\Theta = \{\hat{\theta} \in \mathbb{R}^n : \nexists \theta \in \mathbb{R}^n | \mathbf{J}(\theta) \leq \mathbf{J}(\hat{\theta}), \mathbf{J}(\theta) \neq \mathbf{J}(\hat{\theta})\} \tag{3.2}$$

These optimization processes can be done by a large number of computational intelligence algorithms, *e.g.*, NSGA-II (DEB et al., 2000), SPEA-II (ZITZLER; LAUMANNS; THIELE, 2001), to mention a few. After performing an optimization and obtaining the optimal-Pareto boundary, the next stage is to define one of the solutions (models) of the boundary as most suitable for the application (decision stage). This choice is closely correlated with the problem addressed and the designer's judgment. However, in Nepomuceno, Takahashi and Aguirre (2007), a quantitative alternative is presented, which is to use the validation data, which was not used in the modeling stage, to check the generalization capacity of the Pareto optimal models and choose the most suitable one.

Besides that, during the optimization process, some Prediction Error Minimization algorithms (PEM), which are based on the Error Reduction Ratio (ERR) criteria, or some Simulation Error Reduction algorithms (SEM), which, in turn, are based on Simulation Error Reduction Ratio (SRR) criteria, can be used to simulate the models' regimes and calculate the errors to find the best regressors for the structure of the polynomials. On the one hand, the first approach, ERR, when applied in non-perfect data (*e.g.*, noisy, oversampled, slow input signal, etc.), can result in models with incorrect or redundant terms and be unstable — this approach is often considered a local search technique (FALSONE; PIRODDI; PRANDINI, 2015); on the other hand, they are widely used and fast. The second approach, SRR, can be applied to non-perfect data leading to more compact and sturdy models, but with a high computational cost.

That said, the system identification problem can be divided into five main steps (AGUIRRE, 2015). This procedure is used to identify both linear and nonlinear systems, with some differences in each step of the procedure (AGUIRRE; RODRIGUES; JÁCOME, 1998). In general terms, the five main steps of an identification problem are composed by (AGUIRRE, 2015):

1. data collection, pre-processing and dynamic tests;
2. choice of mathematical model representation;
3. model's structure determination;
4. parameters estimation;
5. model validation.

The following subsections provide an outline of the steps used to solve a system identification problem. All the steps can vary and be presented differently depending on the constraints, nonlinearities or complexity of the model required to solve the problem at hand. Other applications, examples, and indepth discussions about these steps can be found in Aguirre (2015).

3.2 System Identification Steps

3.2.1 Data collection, pre-processing and dynamic tests

According to Aguirre (2015) this system identification step has three fundamental aspects: i. where to stimulate the plant; ii. find the best kind of signal to obtain data that better represents the dynamics of the system and iii. how to sample this data.

In order to examine the behavior of the system, it is necessary to stimulate the whole range of interest frequencies with signals to observe its dynamic and static regime characteristics through direct measurements of output data or by examining the state variables — this process is applicable for non-autonomous systems. Another important aspect of these input signals is their spectral power, which is responsible for the excitation of nonlinearities present in the system. Dynamic and static characteristics that are not stimulated will not appear in the data and, as a consequence, they will not be identified (AGUIRRE, 2015).

In the case of linear systems, the pseudo-random binary sequence (PRBS) is commonly used as the input signal on the identification process. On the other hand, for the nonlinear systems, the random signals are regularly used as input signals, although, in some cases, the PRBSs are also used to identify some models in narrow operating ranges (LEONTARITIS; BILLINGS, 1987a).

Lastly, as most real systems are continuous-time processes, the step of sampling signals takes an important role to discretize the continuous variable and generate the data for identification. To do this, the continuous signal is observed periodically to get the samples, the time between each observation is called the sampling interval or sampling period, T_S. This T_S needs to be precisely defined in order to not lose the characteristics of the original signal, which can be oversampled or undersampled.

The signal is oversampled when the sampling period is very small and, as a consequence, causes numerical instability and high computational effort due to poor conditioning of the regressor matrix (BILLINGS; AGUIRRE, 1995). On the other hand, when the sampling period is too big, the undersampled problem occurs resulting in a misrepresentation of the real dynamics of the system.

Another important issue is to decide which data are relevant to the identification process and which are not. This question increases in importance due to the large quantity of data available for some problems, as a consequence of internet and real-time sensor developments, making the manual data selection process hard or even impossible to be handled and also lead to a high computational cost when estimation algorithms are used. In this scenario, many approaches and techniques (*e.g.*, big data, machine learning, statistical techniques, etc.) were developed to assist the process of selecting good data, *i.e.*, data with relevant information about the system. A good example of these techniques, applied to oil well data, was developed by Ribeiro and Aguirre (2015) where the rank of a regressive matrix of Autoregressive (AR) models, created based on the dataset, was used as an indicator of “signal activity” together with the measure of the correlation between the input and output of the data window to create automatic routines capable of finding the best transients data in the dataset adequated for the system identification problem.

Another good application is performed in Singh, Pani and Mohanta (2019), where five dataset design algorithms were applied to three different types of benchmark datasets to create new datasets and use them to identify soft-sensors. After that, the accuracy results are compared with results obtained from other soft-sensors modeled with the benchmark datasets. All datasets are related to the petroleum refinery process. The five algorithms applied to create the datasets were: the Kennard-Stone (KS) algorithm (KENNARD; STONE, 1969), the DUPLEX algorithm (SNEE, 1977), the Sample set Partitioning based on joint x–y distances algorithm (SPXY) (GALVAO et al., 2005), the error based SPXY algorithm (SPXYE) (GAO et al., 2019), and the kernel-based algorithm SPXY (k-SPXY) (GANI; LIMAM, 2016). The first four algorithms use the Euclidean distance as a metric to select the samples to generate the dataset.

The KS and the DUPLEX algorithms are quite similar, although they have differences in methodology. The DUPLEX generates the test dataset simultaneously with the training dataset unlike the KS, which generates the test dataset with the remaining sample values not used in the creation of the training dataset. The SPXY is similar to the KS algorithm but before following the same steps, it takes into account the statistics of the independent variable x and the dependent variable y for selection of samples in the training set (SINGH; PANI; MOHANTA, 2019), computing the d_{xy} distance for every pair of samples. Another similar algorithm is the SPXYE which is an extension of the SPXY. This algorithm generates an error vector of the preliminary calculation (the d_{xy} values) and computes the d_e which is added to the d_{xy} metric forming the d_{xye} distance — the value that will be used in the SPXY steps. Lastly, k-SPXY is the same algorithm as the SPXY, but instead of computing the Euclidean, the kernel distance is computed (SINGH; PANI; MOHANTA, 2019).

3.2.2 Mathematical model representation

There are several ways to define and represent a mathematical model. One of these definitions split the models into two groups: autonomous and non-autonomous. A model is called autonomous if it does not explicitly contain general input signals (AGUIRRE, 2015). On the other hand, models that have at least one general input are called non-autonomous.

This type of model describes the output data given an input excitation. Another common way to classify the models is related to time, *i.e.*, continuous and discrete. According to Aguirre (2015) the continuous dynamic models are described by differential equations that represent the continuous evolution of the system through time and the discrete dynamic models are described using the difference equation to represent specific moments in time.

Another important thing to know concerning mathematical representation is whether the system is linear or non-linear. A model is linear if the input-output relation satisfies the superposition property. The superposition property is a combination of another two properties, *i.e.*, the additivity and homogeneity. The additivity property says that if an input x_1 implies the output y_1, $(x_1 \rightarrow y_1)$, and another input x_2 implies the output y_2, $(x_2 \rightarrow y_2)$, so when these two inputs where working together on the system, the total output will be $y_1 + y_2$. The second propriety, homogeneity, says that for a real or imaginary arbitrary number k, if the input increases k times, the output will also increase k times. So the superposition property is $k_1x_1 + k_2x_2 \rightarrow k_1y_1 + k_2y_2$. Thus, the output of a linear system to a combined input can be described as the sum of the outputs for simpler inputs.

Examples of linear representation include transfer functions, space state representations, and polynomial models (*e.g.*, autoregressive model - AR, autoregressive with exogenous inputs model - ARX, autoregressive moving average with exogenous inputs model - ARMAX, etc.) (AGUIRRE, 2015). In practice, most of the real problems are non-linear and the approach based on linear systems is inadequate because they do not present important aspects of the process (POPE; RAYNER, 1994). The nonlinear models appear as a solution that can handle many of these real situation problems and represents complex dynamical regimes with good accuracy.

Examples of nonlinear representation include Artificial Neural Networks - ANN (HAYKIN, 2007), which are inspired by the functioning of the human brain, where artificial neurons, connected in a network, are able to learn and generalize. Another representation model is the radial basis functions networks (BROOMHEAD; LOWE, 1988), which is basically an ANN with a radial basis function as the activation function.

Volterra series (VOLTERRA, 1930; BILLINGS, 1980) is another nonlinear mathematical model representation. The Volterra series is an extension of the Taylor series that has memory capacity. It’s also possible to have models based on interconnected blocks, such as the Hammerstein model and the Wiener model (WIENER, 1958; WIGREN, 1993; AGUIRRE, 2015; COELHO, 2002). Both models are composed of a dynamic linear model $G(q)$ in cascade with a static nonlinear function $f(\cdot)$. The difference between these two models is where the nonlinear static takes place. For Hammerstein's model, it precedes the linear dynamic model while in Wiener's model it succeeds.

Last but not least, there are rational and polynomial functions (CHEN; BILLINGS, 1989; JOHANSEN; FOSS, 1992; ZHU; BILLINGS, 1993; LEONTARITIS; BILLINGS, 1985a; LEONTARITIS; BILLINGS, 1985b). In this field, the Nonlinear Auto-Regressive with eXogenous inputs - NARX and its extension, the Nonlinear Auto-Regressive Moving Average with eXogenous inputs - NARMAX, are a general representation for a wide range of non-linear systems. The NARX model can be defined as:

$$y(k) = F[y(k-1),\cdots,y(k-n_y),u(k-\tau_d),\cdots,u(k-n_u)], \tag{3.3}$$

where $u(k-i)$ and $y(k-j)$ represent, respectively, the measured input and output of the system at $k-i$ and $k-j$ sampling times. The n_y, n_u and τ_d are the highest delay in y, in u and the dead time, respectively. It's possible to compute the number of model's regressors as $\varepsilon = n_y + n_u - \tau_d + 1$.

As indicated previously, the NARMAX model is an extension of the NARX model, but with added moving average noise terms to avoid the polarization of the parameters. The NARMAX model can be defined as (LEONTARITIS; BILLINGS, 1985a; LEONTARITIS; BILLINGS, 1985b; CHEN; BILLINGS, 1989; AGUIRRE, 2015; ZHU; BILLINGS, 1993):

$$y(k) = F^l[y(k-1),\cdots,y(k-n_y),u(k-\tau_d),\cdots,u(k-n_u),e(k-1),\cdots,e(k-n_e)]+e(k), \tag{3.4}$$

where $e(k)$ indicates the effects that can't be well represented by $F^l[\cdot]$. $F^l[\cdot]$ is any polynomial function with a degree of nonlinearity $l \in \mathbb{N}$. The functions $y(t)$, $u(t)$ and $e(t)$ represents the output, input and system noise, respectively. The n_y, n_u, n_e and τ_d are the highest delay in y, in u, in e and the dead time, respectively. The deterministic part, *i.e.*, noise-free part of the Equation 3.4 can be expanded as the sum of terms with degrees of nonlinearity varying in the range $1 \leqslant m \leqslant l$. Thus, each term of degree m may contain a factor of degree p of type $y(k-\tau_i)$ and a factor of degree $(m-p)$ of type $u(k-\tau_i)$ being multiplied by a parameter represented by $c_{p,m}(\tau_1,...,\tau_{p+m})$. The model can be described as (JONES; BILLINGS, 1989; AGUIRRE, 2015):

$$y(k) = \sum_{m=0}^{l} \sum_{p=0}^{l-m} \sum_{\tau_1,\tau_m}^{n_y,n_u} c_{p,m}(\tau_1,\cdots,\tau_{p+m}) \prod_{j=1}^{p} y(k-\tau_j) \times \prod_{i=1}^{m} u(k-\tau_{p+i}) + e(k), \tag{3.5}$$

whereas

$$\sum_{\tau_1,\tau_m}^{n_y,n_u} \equiv \sum_{\tau_1=1}^{n_y} \cdots \sum_{\tau_p=1}^{n_y} \sum_{\tau_{p+1}=\tau_d}^{n_u} \cdots \sum_{\tau_{p+m}=\tau d}^{n_u}, \tag{3.6}$$

and the superior limit will be n_y if the summation is referred to the factors of type $y(k-\tau_i)$, or $u(k-\tau_i)$ for the n_u factors.

3.2.3 Model's structure determination

After choosing the mathematical model representation it's necessary to determine the model structure. This step is decisive to achieve good results in identification problems. In linear models, the possible number of regressors increases linearly with the model order. In this case, the structure selection step is basically to choose the number of poles and zeros as well as determine the pure time delay (AGUIRRE, 2015).

On the other hand, for the nonlinear polynomial models, *i.e.*, the NARMAX models, which are the main focus of this work, the possible number of regressors increases proportionally to the non-linearity degree and the maximum delays (*i.e.*, l, n_y, n_x and n_e), which results in an exponential increase in the number of candidate model structures (the curse of dimensionality) when compared to linear models (FALSONE; PIRODDI; PRANDINI, 2015). The number of candidate terms (n_{terms}), with $\tau_d = 0$, for a model can be determined as follows:

$$n_{terms} = M + 1, \tag{3.7}$$

where

$$\begin{aligned} M &= \sum_{i=1}^{l} n_i, \\ n_i &= \frac{n_{i-1}(n_y + n_u + i - 1)}{i}, \\ n_0 &= 1. \end{aligned} \tag{3.8}$$

As discussed in Aguirre and Billings (1995b), another problem that can occur during the process of structure selection is the overparametrization, *i.e.*, chose an excessive number of regressors for the model. This can lead to complex models that tend to be unstable, induce ghost bifurcations, and have spurious dynamical regimes. Many approaches were already proposed to find the regressors of NARX and NARMAX models (KORENBERG et al., 1988; LEONTARITIS; BILLINGS, 1987; BILLINGS; CHEN; KORENBERG, 1989; AGUIRRE; BILLINGS, 1995a; MAO; BILLINGS, 1997; PALUMBO; PIRODDI, 2001; WEI; BILLINGS, 2008; PIRODDI, 2008; CASTRO; BARBOSA, 2019; HAFIZ; SWAIN; MENDES, 2020). Among the many approaches mentioned, it is worth mentioning some that used computational intelligence algorithms, more precisely evolutionary algorithms, to optimize the selection of the best regressors for NARX/NARMAX models. For example, in Castro and Barbosa (2019), a Multiobjective Genetic Algorithm was used in two approaches. The first one, used the prediction error minimization, as the first optimization objective, and the reduction of the number of selected regressors as the second objective. The second approach used the free-run simulation error minimization, as the first optimization objective, and also the reduction of the number of selected regressors as the second. Another work that can be mentioned is Hafiz, Swain and Mendes (2020), where a comparison was made between three Multi-Objective Evolutionary Algorithms (MOEAs), they being NSGA-II, SPEA-II, and MOEA/D. These algorithms were used to propose a multi-objective framework for structure selection of nonlinear systems which are represented by polynomial NARX models. In both works, it was demonstrated that the multi-objective optimization approach with evolutionary algorithms for structure selection is promising and versatile.

In the next subsections, methods for model structure selection like the Error Reduction Ratio (ERR) criterion and the Simulation Error Reduction Ratio (SRR) criterion will be briefly presented. Some information criteria, like the Akaike criterion, will be briefly explained. Finally, in Subsection 3.2.3.4, the structure selection problem as a multi-objective optimization approach will be investigated.

3.2.3.1 Error Reduction Ratio (ERR) Criterion

The Error Reduction Ratio (ERR) (BILLINGS; CHEN; KORENBERG, 1989) is a well known and used criterion to select the independent variables, *i.e.*, the regressors, for a model through one-step-ahead predictions. This criterion evaluates the reduction in the variance of the residuals $\xi(k)$, that occurs when a new term is included in the model, and can be normalized with respect to the total variance of the output signal. To define the ERR, the NARMAX (Equation 3.4) models will be considered as follows (AGUIRRE, 2015):

$$\begin{aligned} y(k) &= \psi^T(k-1)\hat{\theta} + \xi(k) \\ &= \sum_{i=1}^{n_\theta} \hat{\theta}_i \psi_i(k-1) + \xi(k) \end{aligned} \tag{3.9}$$

where n_θ is the number of parameters. The auxiliary model, *i.e.*, the model represented on an orthogonal basis is defined as:

$$y(k) = \sum_{i=1}^{n_\theta} \hat{g}_i w_i(k-1) + \xi(k) \tag{3.10}$$

where $\hat{g}_i$ are the estimated parameters and w_i, the orthogonal regressors on the data. It is interesting to note in Equation 3.10 that when n_θ = 0 (zero regressors) the output signal $y(k)$ is equal to the prediction error. The Error Reduction Ratio due to the inclusion of the i-th regressor in the model is defined as (CHEN; BILLINGS; LUO, 1989):

$$[ERR_1]_i = \frac{\hat{g}_i^2 \langle \mathbf{w_i}, \mathbf{w_i} \rangle}{\langle \mathbf{y}, \mathbf{y} \rangle} \tag{3.11}$$

where the $\langle \cdot \rangle$ operator represents the internal product of two vectors and the ERR indicates the variance part of the output explained by the inclusion of a new term in the model. One criterion that can be used is to include the regressors with the highest ERR values among a normally large set of candidate regressors (AGUIRRE, 2015). The ERR criterion has some extensions, like the ERR$_2$ (ALVES; CORRÊA; AGUIRRE, 2012), that uses the two-step-ahead predictions in order to detect unwanted terms. As mentioned in the introduction of this chapter, the identification algorithms based on the Error Reduction Ratio as optimization criterion are called prediction error minimization (PEM) algorithms.

A well known PEM algorithm is the Forward Regression Orthogonal Estimator (FROE) (BILLINGS; CHEN; KORENBERG, 1989). In Piroddi and Spinelli (2003), the limitations of FROE with regard to the structure selection problem were discussed. It has been shown that algorithms based on ERR only result in sub-optimal models. In addition, they may find incorrect and redundant models when subjected to certain noise and input signals. Another point mentioned is the high probability that models generated by the ERR criterion be extremely inaccurate and unstable when submitted to the free-run simulation process. As a solution to these problems, it's suggested changing the ERR criterion in the FROE algorithm for the simulation error reduction ratio (SRR) that is more precise and robust with respect to the excitation characteristics of the identification data. The SRR criterion will be briefly explained in the next subsection.

3.2.3.2 Simulation Error Reduction Ratio (SRR) Criterion

In Piroddi and Spinelli (2003), the simulation error reduction ratio (SRR) criterion was proposed to overcome the difficulties of the algorithms based on the ERR criterion. Unlike the ERR, which uses one-step-ahead prediction, the SRR criterion is based on free-run simulation. This criterion is defined by the reduction of the mean square simulation error (MSSE), as follows (PIRODDI; SPINELLI, 2003):

$$[SRR]_j = \frac{MSSE(\mathscr{M}_i) - MSSE(\mathscr{M}_{i+1})}{\frac{1}{N}\sum_{t=1}^{N} y^2(k)} \tag{3.12}$$

where $\mathscr{M}_i$ is the model obtained at the i-th iteration and $\mathscr{M}_{i+1}$ is the candidate model at the subsequent iteration, with the inclusion of the j-th regressor. As can be noted, the SRR criterion is based on the simulation error reduction, algorithms with this characteristic are known as simulation error minimization (SEM) algorithms. These algorithms generally obtain compact and robust models that can be effective in non-ideal identification conditions. Conversely, the SRR based algorithms require a significant computational effort and are not viable for the identification of chaotic systems, owing to the extreme sensitivity of their behavior to initial conditions (PIRODDI; SPINELLI, 2003).

Piroddi and Spinelli (2003) also proposed the simulation error minimization with pruning (SEMP) algorithm. This algorithm deals with the problem of changing the real importance of terms that occurs during the process of building the model. The SEMP algorithm presents good results for this problem when implementing the pruning procedure in terms that do not contribute significantly to the quality of the model during its execution.

3.2.3.3 Information Criteria

As described in the Sections 3.2.3.2 and 3.2.3.1, the methods and criteria exposed, *i.e.*, the ERR criterion and SRR criterion, are good in ranking candidate regressors terms hierarchically, but it's still necessary to choose the number of regressors for the final model. This task also needs to handle the bias-variance trade-off, *i.e.*, if a large number of regressors are chosen the model can overfit the data and model the noise, which leads to high variance. On the other hand, if a small number of regressors were chosen the model will not generalize the data (underfitting) and will have a high bias. To deal with this task many information criteria were proposed, for example, the Final Prediction Error (FPE) criterion (AKAIKE, 1970), the Akaike Information Criterion (AIC) (AKAIKE, 1974), and the Minimum Description Length (MDL) criterion (RISSANEN, 1989).

According to the Akaike Information Criterion, the ideal number of terms in a model should minimize the following function (AKAIKE, 1974):

$$AIC(n_\theta) = N \ln(\mathrm{Var}[\xi(n_\theta)]) + 2n_\theta, \tag{3.13}$$

where N corresponds to the number of samples, $\mathrm{Var}[\xi(n_\theta)]$, the variance of residue $\xi(n_\theta)$, *i.e.*, the variance of one-step forward prediction error, and n_θ, the number of terms in the model. Equation 3.13 can be divided into two parts. The first one, $N\ln(\mathrm{Var}[\xi(n_\theta)])$, is responsible to measure the reduction in the variance of the residue resulting from the inclusion of a term. This reduction happens because when a term is added, the model's degrees of freedom increase and a better adjustment is made to the data that decreases the variance $\mathrm{Var}[\xi(n_\theta)]$ and, as a consequence, decreases the first part of the equation. Although this decreasing effect has a threshold when no matter the number of terms added the effect on AIC will be insignificant. The second part, $2n_\theta$, will penalize the inclusion of terms in the model when more terms are added the higher the value of AIC will be. Due to the overall minimization goal, if the cost of adding a term for the second part of the equation is high than the reduction in the first part of the equation this term shouldn't be included in the model. When the AIC reaches a threshold or shows a "knee", its execution is terminated (AGUIRRE, 2015).

It is important to note that, as Akaike Information Criterion is fundamentally statistical, it can't be said that the model with the number of terms selected by the AIC is valid (AGUIRRE, 2015; AGUIRRE; BILLINGS, 1994). Nepomuceno et al. (2002) argued that the results of the AIC criterion can be seen as an indicator in the search for the ideal number of regressors in the model.

Other types of methods, criteria, techniques, and approaches can be used to select and define the number of regressors for a model. One example is the recent use of Evolutionary Algorithms (EA), especially the MOEAs, to select the regressors (see, *e.g.*, Hafiz, Swain and Mendes (2020), Castro and Barbosa (2019), Barbosa et al. (2011)).

3.2.3.4 Multi-objective Optimization for Structure Detection

As presented in this chapter introduction (Chapter 3), an identification problem can be interpreted as a multi-objective optimization problem. In face of that, finding the number of regressors of a model, *i.e.*, the structure selection step, is an important issue in system identification, especially for non-linear systems (BARBOSA; TAKAHASHI; AGUIRRE, 2015). As discussed in Hafiz, Swain and Mendes (2020), in this stage, it's important to search for a model with a parsimonious structure and good predictive performance, which is essentially contradictory. This contradiction is due to the bias-variance dilemma, in other words, an excessively compact model (low number of regressors) may not be able to replicate the behavior of the real system (underfitting) and present a high bias output, on the other hand, a model with many regressors can memorize the identification data and not generalize to unknown samples. Therefore, the process of finding optimized models for these objectives (structure selection) is, in essence, a multi-objective problem (HAFIZ; SWAIN; MENDES, 2020). This approach is not a new concept but is still an open field.

That said, solving the structure selection problem with a multi-objective approach has some advantages, for instance, it is possible to use dynamic and static data together during the optimization process. This can be seen in Martins, Nepomuceno and Barroso (2013), where an extension of the ERR, the Multi-objective Error Reduction Ratio (MERR), is proposed to solve the structure detection problem for polynomial NARX models. With this extension, it's possible to use the dynamics of prediction error along with affine information (*e.g.*, fixed points, static curve) to get nondominated solutions of the Pareto set.

Another multi-objective approach to solve the structure detection problem is using computational intelligence algorithms (in their multi-objective versions). Among several, the algorithms based on Charles Darwin's evolutionary theory (DARWIN, 1859), the Multi-Objective Evolutionary Algorithms (MOEAs), have stood out. For example, in Zakaria et al. (2012), the Non-dominated Sorting Genetic Algorithm II (NSGA-II), proposed by Deb et al. (2000), was used to select the structure and define a parsimonious model. In the optimization task, two objectives ware used, minimize the number of regressors and the prediction error. Real and simulated data were used in the process and, in the end, the resulting models were compared with the model resulting from a single-objective genetic algorithm. The results show a better performance for the NSGA-II approach. Likewise, in Hafiz, Swain and Mendes (2020), a multi-objective framework for structure selection to polynomial NARX models was proposed. This framework is composed of three well-known MOEAs (*i.e.*, NSGA-II, SPEA-II, and MOEA/D) that were submitted to a rigorous statistical analysis via performance sweet spots (*i.e.*, the high performance region (control map) of the algorithm results formed by the feasible settings of control parameters (mutation and crossover probability, and selection pressure)) in the parameter space obtaining robust results for the regressors selection task. Other works that follow a similar approach are Ferariu and Patelli (2009) and Rodriguez-Vazquez, Fonseca and Fleming (2004).

3.2.4 Parameters estimation

After defining the mathematical representation and selecting the regressors to the model, the next step is the parameter estimation. It is necessary to estimate the model parameters for each regressor selected in the previous step. For this purpose, the identification data (*i.e.*, the data collected during experiments with the real plant) or a dataset obtained through simulation are separated to be used. Another part of this dataset is assigned to be used in the validation step, better discussed in Section 3.2.5.

The majority of the algorithms employed to estimate the parameters in polynomial models are based on the Least Squares estimator (LS) (LEGENDRE, 1805; GAUSS, 1963). The Ordinary Least Squares and one of its extensions, the Extended Least Squares, will be exposed in the Subsection 3.2.4.1 and Subsection 3.2.4.3, respectively. Finally, in Subsection 3.2.4.4, the parameter estimation problem as a multi-objective optimization approach will be investigated.

3.2.4.1 Ordinary Least Squares

The Ordinary Least Squares is widely used to compute the model parameters to NARX polynomials. Therefore, Equation 3.3, which represent a NARX model, is rewritten as follows (AGUIRRE, 2015):

$$\begin{aligned} y(k) &= \psi^T(k-1)\hat{\theta} + \xi(k) \\ &= \hat{y}(k) + \xi(k), \end{aligned} \tag{3.14}$$

where k indicates the considered time step, $\hat{\theta}$ represents the estimated vector of parameters, $\psi^T(k-1)$ corresponds to the vector of regressors, which can contain observations up to $(k-1)$ and, $\xi(k)$ is the computed model error when trying to explain $y(k)$, the output, as $\psi^T(k-1)\hat{\theta}$. Note that this symbol ($\hat{\ }$), above the variables, indicates estimated values.

Applying the Equation 3.14 for all samples in a dataset and writing the result in a matrix form:

$$\mathbf{y} = \Psi\hat{\theta} + \xi, \tag{3.15}$$

where $\xi = [\xi_1\ \xi_2\ \cdots \xi_N]^N$ is the error vector generated by the attempt of explaining $\mathbf{y}$ by $\Psi\hat{\theta}$, where N is the number of samples, and Ψ represents the regressors matrix. Isolating the vector of residues ξ, the Equation 3.15 can be written as:

$$\xi = \mathbf{y} - \Psi\hat{\theta}. \tag{3.16}$$

To solve Equation 3.16 it is necessary to find a group of parameters $\hat{\theta}$ that satisfies it. During this process, it would be interesting to reduce the residues ξ value and obtain a more precise result, therefore, the sum of squares of errors, *i.e.*, the loss function, is defined as (AGUIRRE, 2015):

$$J_{LS} = \sum_{i=1}^{N} \xi(i)^2 = \xi^{\mathbf{T}}\xi = \|\xi\|^2. \tag{3.17}$$

Substituting Equation 3.16 in Equation 3.17, the Ordinary Least Squares estimator, $\hat{\theta}_{\mathbf{LS}}$, that minimizes the loss function J_{LS}, can be proven to be:

$$\hat{\theta}_{LS} = [\Psi^T\Psi]^{-1}\Psi^T\mathbf{y}. \tag{3.18}$$

An important issue is that this estimator can only be applied to linear-in-the parameters models. In some situations, like output noise, the LS estimator results in polarized estimates. To overcome this problem, the Extended Least Squares (ELS) can be used, this estimator will be explained in Section 3.2.4.3.

3.2.4.2 Weighted Least Squares

In many practical situations where the ordinary least squares estimator is applied, it may be necessary to weigh different samples from a dataset differently (*e.g.*, sampling windows that better represent one operation point than others). Therefore, to represent this need, Equation 3.17 is rewritten and named Weighted Least Squares (WLS), as follows (AGUIRRE, 2015):

$$J_{WLS} = \sum_{i=1}^{N} \xi(i) w_i \xi(i) = \xi^{\mathbf{T}}\mathbf{W}\xi, \tag{3.19}$$

where $W \in \mathbb{R}^{N \times N}$ is a diagonal matrix whose elements are the weights w_i, *i.e.*, $W = diag\{w_1\ w_2\ \cdots\ w_N\}$. Substituting Equation 3.16 in Equation 3.19, the Weighted Least Squares estimator, $\hat{\theta}_{\mathbf{WLS}}$, that minimizes the loss function J_{WLS}, can be proven to be:

$$\hat{\theta}_{WLS} = [\Psi^T W \Psi]^{-1} \Psi^T W \mathbf{y}. \tag{3.20}$$

It is essential to mention that the WLS described by Equation 3.20 is also valid for the case where W is not diagonal.

3.2.4.3 Extended Least Squares

As mentioned earlier in Section 3.2.4.1, the Least Squares estimator can result in polarized model parameters. In other words, the model regressors are correlated with the regression error, and if the LS is used to estimate the parameters vector they will be polarized (AGUIRRE, 2015). As a solution to this problem, the Extended Least Squares (ELS) takes place. Considering the parametric model to be used in the regression as (AGUIRRE, 2015):

$$y(k) = \psi^T(k-1)\theta + e(k), \tag{3.21}$$

where $\psi^T(k-1)$ is the regressor vector taken up to $(k-1)$, θ is the parameter vector and $e(k) = c\nu(k-1) + \nu(k)$ is the regression equation errors, where $\nu(k)$ is a white noise — $e(k)$ shouldn't be understood as the measured or observed noise. Applying Equation 3.21 in a dataset of size N the matrix form is given by:

$$\mathbf{y} = \Psi\theta + \mathbf{e}, \tag{3.22}$$

where the regressors matrix is:

$$\Psi = \begin{bmatrix} \psi(k-1) \\ \psi(k-2) \\ \vdots \\ \psi(k-N) \end{bmatrix} \tag{3.23}$$

To model the white noise and fix the polarization problem it's necessary to extend the regressors matrix Ψ adding the $\nu(k-1)$ terms. The extended regressor matrix is given by:

$$\Psi^* = \begin{bmatrix} & \vdots & \nu(k-1) \\ & \vdots & \nu(k) \\ \Psi & \vdots & \nu(k+1) \\ & \vdots & \vdots \\ & \vdots & \nu(k+N-2 \end{bmatrix} \tag{3.24}$$

whereas $\mathbf{y}^* = \mathbf{y}$, $\mathbf{e}^* = [\nu(k) \; \cdots \; \nu(k+N-1)]^T$, and $\theta^* = [\theta^T \vdots \; c]^T$. It's important to note that, because $e^*(k)$ is a "white" noise variable, $\mathbf{e}^*$ is not correlated with Ψ^*. Unfortunately, the $\nu(k)$ values are unknown and its is necessary to estimate the extended parameters vector. To solve this problem of finding $\nu(k)$ and estimate the parameters Ψ^* to get the complete solution, the following interactive proceeding is necessary (AGUIRRE, 2015):

1. from Equation 3.22, like LS, compute $\hat{\theta}_{LS} = [\Psi^T\Psi]^{-1}\Psi^T\mathbf{y}$;
2. compute the residues vector $\xi_1 = \mathbf{y} - \Psi\hat{\theta}_{LS}$;
3. do $i = 2$ (where i indicate the actual interaction);
4. with ξ_{i-1} create the extended regressors matrix, Ψ_i^*, and estimate $\theta_{ELS_i}^* = [\Psi_i^{*T}\Psi_i^*]^{-1}\Psi_i^{*T}\mathbf{y}$;
5. compute the residues vector $\xi_i = \mathbf{y} - \Psi_i^*\theta_{ELS_i}^*$;
6. do $i = i+1$ and return to step 4. Repeat until converging.

Besides the ELS, there are many other LS derived algorithms, for example, the Generalized least squares, the Total least squares, and the Constrained least squares.

3.2.4.4 Multi-objective Optimization for Parameter Estimation

Once the model structure is correctly defined it is necessary to estimate the parameters of each model regressor. As aforementioned, applying a multi-objective optimization to solve the identification problem allows the use of auxiliary information, *i.e.*, information apart from the set of dynamical data (*e.g.*, fixed points, static function, and static gain), which can improve the prediction accuracy of the resulting models.

In Nepomuceno et al. (2003), a parameter estimation approach for NARX models was proposed, in two steps, using a bi-objective optimization and a posteriori decision scheme. The two objectives were times series fitting error and fixed-point fitting error. The first step is to find the non-dominated set of solutions, and the second is to find the final model applying some decision criterion. An extension of this methodology was proposed in Nepomuceno, Takahashi and Aguirre (2007), where more affine information was aggregated in the cost function turning it into a multi-objective problem. They also presented a non-iterative form to estimate the parameters of a multi-objective problem using the least squares. Other works that also use the multi-objective approach are Barroso, Takahashi and Aguirre (2007) and Barbosa, Takahashi and Aguirre (2015).

A different approach for multi-objective parameter estimation was proposed in Aguirre, Barbosa and Braga (2010). In this work, an iterative solution with genetic algorithms (GA) to solve the optimization problem was applied. The use of GA is interesting because this approach rarely gets stuck in local minima. Especially in the mentioned work, this is important because different from the ones mentioned in the last paragraph, the simulation error was used as one of the objectives and this renders a nonconvex optimization problem with many potential minima — the other objective used was the prediction error. These two objectives were also used, separately, as a one-objective problem in order to compare the use of the two objectives alone and together. The result shows that, in general, using the simulation error is preferable to prediction error for parameter estimation. Another work that uses an evolutionary approach in multi-objective problems for system identification is Rodriguez-Vazquez and Fleming (1998).

3.2.5 Model Validation

The model validation is the last step of the identification problem. This step is performed when the final model is complete, *i.e.*, structure defined and all model parameters estimated. To validate a model means that it will be checked if it can represent all interested dynamic characteristics of the real system, *i.e.*, test the model's capacity of generalization.

In Aguirre (2015), it is emphasized that the samples of a dataset used in the identification process (*e.g.*, parameter estimation, structure selection, etc.) shouldn't be used in the validation because it will lead to biased results. Therefore, it's important to have two distinct datasets, the training dataset, to model the new system, and the validating dataset, to check if the model is valid. This split can be done in many different ways. The simplest way is to split the original dataset randomly in the two sets. For instance, in a dataset with one hundred samples, the first seventy points are addressed to the training set and the last thirty ones to the validation set. The problem of this approach is that precious dynamic information of the system can be addressed just for the validating dataset and will not be modeled in the identification process leading to a weak model generalization. This problem can be overcome by the Cross-validation (MOSTELLER; TUKEY, 1968; STONE, 1974) technique, which is briefly explained in Subsection 3.2.5.1.

It is clear that it is the set of validation data that should be used to assess the accuracy of the final model. There are several metrics and criteria to perform this step, the right one will depend on the specific necessities that motivate the creation of the model. A widely used way to validate models is through simulation.

There are two main types of simulation, the free-run simulation and the one-step-ahead prediction, already mentioned in other sections. In the free-run simulation, the model starts with the data from the validation set and is indefinitely simulated receiving as feedback, to estimate the next step, only the past predictions made by the model itself. On the other hand, the one-step-ahead prediction also starts with validation data but during the simulation it doesn't estimate the next steps based on its own predictions, the model also queries the validation data to make the next step prediction. As a direct consequence, in Aguirre (2015), it's demonstrated that the one-step-ahead prediction is not a good method to test the generalization capacity of the final model.

In order to evaluate the performance and quality of the simulation results, a metric is necessary. The mean squared error (MSE) is well known and widely used to do that, and can be expressed by:

$$MSE = \frac{1}{N}\sum_{k=1}^{N}(y(k)-\hat{y}(k))^2, \tag{3.25}$$

where N is the number of data points, $y(k)$ is the observed data, and $\hat{y}(k)$ is the result of the model simulation, *i.e.*, the prediction. Another widely used metric is the root mean squared error (RMSE) which is defined as:

$$RMSE = \sqrt{\frac{1}{N}\sum_{k=1}^{N}(y(k)-\hat{y}(k))^2}, \tag{3.26}$$

where N is the number of data points, $\hat{y}(k)$ is the result of the model simulation and $y(k)$ is the observed data. The lower the value of MSE and RMSE, the greater the model's ability to adjust to the data. Another important analysis is that a well-fitted model has similar values of MSE and RMSE for the training and validating data.

There is another type of validation, the static validation. In this validation, the ability of a model to recover the static characteristic of the system is checked. The static characteristic is the relation between the system output and the input in a steady state. As discussed in Aguirre (2015), the use of static characteristics of nonlinear models in its validation is not a common proceeding. However, static validation was applied in some works (see, *e.g.*, Coelho (2002), Corrêa (2001)).

3.2.5.1 Cross-validation

The cross-validation (ALLEN, 1974; STONE, 1977; STONE, 1974), also known as rotation estimation, is a group of techniques used in validation tasks in order to estimate how the results of an identified model generalize a dataset. It is largely used to estimate how accurate a model will perform with unseen data (DONATE et al., 2011).

The main idea of the cross-validation is the dataset partitioning into complementary subsets. One of the subsets is addressed to perform the identification process (identification/train data) and the remaining subsets to validate the model. The number of subsets can vary depending on the partitioning methodology (*e.g.*, holdout, the *k*-fold, leave-one-out). The *k*-fold, for example, split the dataset into k complementary subsets, with the same length, and address one as the validation set, and the $k-1$ remaining subsets as the training dataset (identification data). This process is repeated k times, and the total accuracy for the model is computed by taking the average error of the k models' output estimates over validating data, as follows:

$$kFCV_k = \frac{1}{k}\sum_{i=1}^{k} MSE_i. \tag{3.27}$$

where k is the number of folds, and MSE_i is the mean squared error, presented in Section 3.2.5, for a fold i. If the k value is correctly chosen, the $kFCV_k$ result will be a reliable measure of the identified model capacity over unseen data. Other cross-validation techniques are discussed and presented in Breiman and Spector (1992) and Kohavi (1995).

3.2.6 Soft Sensor

The identification methodology, explained in the last section, can be applied in many different areas to solve a wide range of problems. One tool derived from this is the soft sensors. This term is a combination of two words, "software" and "sensor". The word "software" is derived from the fact that the soft sensors are usually implemented as computer programs, and the word "sensor" is because they have almost the same behavior as hardware sensors (KADLEC; GABRYS; STRANDT, 2009).

The soft sensors (SSs) (FORTUNA et al., 2007b; KADLEC; GABRYS; STRANDT, 2009), also known as virtual sensors, are inferential mathematical models capable to provide an estimation for an unmeasured variable on the basis of a set of other measured variables from different processes. This estimated variable can be hard to measure due to many different situations like technological reasons (*e.g.*, there is no equipment in the market with the desired requirements) and high investment needed (*e.g.*, the equipment can be expensive) (KADLEC; GABRYS; STRANDT, 2009), which make the SS an important mechanism in practical situations.

As mentioned in Fortuna et al. (2007b), the soft sensors have a lot of attractiveness, for example:

1. they are a low-cost alternative to expensive equipment and hardware devices;
2. they can make the hardware measures more reliable due to their parallel work capacity, they can also help to detect equipment faults;
3. it's simple to implement them in existing hardware;
4. they solve time delay problems due to their real-time data estimation capacity, which also helps to improve the control strategies.

According to Kadlec, Gabrys and Strandt (2009), there are three different classes of soft sensors: Model-driven, Data-driven, and hybrids.

The Model-driven soft sensors are usually implemented based on First Principle Models (FPM) (PRASAD et al., 2002). Therefore, they are based on a physicochemical background of the processes and use these equations to calculate the value of the desired variables (KADLEC; GABRYS; STRANDT, 2009). The specialists' knowledge and experience of the process are also used during the modeling process. Due to these characteristics, this category of SS is also known as white-box models and phenomenological models (FORTUNA et al., 2007b). The models resulting from this approach have a high computational cost and, as a consequence, real-time implementation can be a problem (LIN et al., 2007). Another problem of the resulting models is correlated with the fact that they were modeled based o ideal situations, with no disturbance and failures, which is not common in real situations.

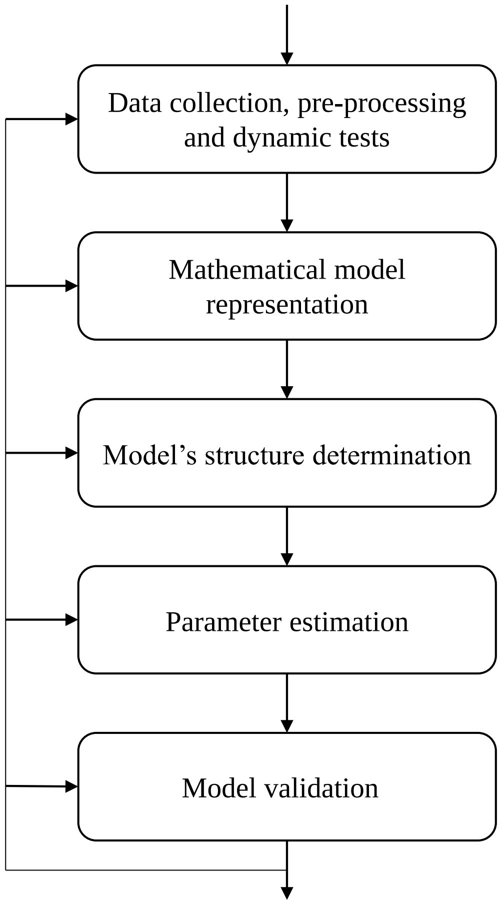
The second class of SS is the data-driven models. These models are entirely based on empirical observation of the process, *i.e.*, no *a priori* knowledge is used in the modeling process, due to this they are also known as black-box models (KADLEC; GABRYS, 2009). Due to the fact that only measured data is used to model this soft sensor, they are more likely to describe the true conditions of the process, when compared with the model-driven. With the increase in the instrumentalization of plants, the volume of data generated, analyzed, and stored increased, thus, the applicability of data-driven virtual sensors was facilitated and became a viable alternative to solve the model-driven limitations (LIN et al., 2007). The identification methodology to implement a data-driven soft sensor is summarized in Figure 3.1. It's important to note that they are the same steps of a regular identification problem, which were explained in this section (Section 3).

Lastly, there are hybrid soft sensors. This category of soft sensors is a combination of model-driven and data-driven approaches, because of that, they are also known as grey-box models. A common illustration of this method is the use of a data-driven approach to model some fractions for a model-driven soft sensor (KADLEC; GABRYS, 2009).

Soft sensors are applied in a wide range of different industrial processes. There are examples of use in chemical plants (GRAZIANI; XIBILIA, 2018), power plants (BHAVANI et al., 2014; SUJATHA et al., 2018), nuclear plants (RIZZO, 2010), pollution monitoring (FORTUNA et al., 2006), grinding plants (CASALI et al., 1998), steel industry (RADHAKRISHNAN; MOHAMED, 2000), food industry (OSORIO et al., 2008), and many others areas.

There are also applications in the oil and gas fields, which are the focus of this work, such as product quality monitoring in refineries (FORTUNA et al., 2007a; FORTUNA; GRAZIANI; XIBILIA, 2005; FORTUNA et al., 2005; GRAZIANI; XIBILIA, 2019; PANI; AMIN; MOHANTA, 2016; ROVERSO, 2009). Talking specifically about monitoring of downhole pressure in oil wells, Barbosa et al. (2015) implemented a data-driven soft sensor to estimate the downhole pressure using committee machines composed by finite impulse response neural networks. With the same purpose, Aguirre et al. (2017) implemented data-driven and hybrid

Figure 3.1 – Soft sensor identification procedure diagram.


Data collection, pre-processing and dynamic tests
Mathematical model representation
Model's structure determination
Parameter estimation
Model validation

Source: Adapted from Fortuna et al. (2007b)

soft sensors using NARMAX and neural as models representation. In Antonelo, Camponogara and Foss (2017), soft sensors were implemented using Recurrent Neural Networks to solve the same problem. Other works trying to solve the same problem with soft sensors are (DAVIES; AGGREY et al., 2007; TEIXEIRA et al., 2012; TEIXEIRA et al., 2014; SUI et al., 2011; MORAIS et al., 2019).

As can be noted in the last paragraph the hybrid and data-driven soft sensors can be implemented with different tools that range from statistical approaches to computational intelligence algorithms (KADLEC; GABRYS; STRANDT, 2009). Examples of tools that can implement soft sensors are Principal Component Analysis (JOLLIFFE, 1986; LIN et al., 2007), Partial Least Squares (WOLD et al., 1987; NOMIKOS; MACGREGOR, 1995), Support Vector Machines (VAPNIK, 1998; FENG; SHEN; SHAO, 2003), Support Vector Regression (SMOLA; SCHÖLKOPF, 2004; DESAI et al., 2006; VAPNIK; GOLOWICH; SMOLA, 1996), Neural Networks (BISHOP et al., 1995; DEVOGELAERE et al., 2002; WANG et al., 2006; SU; FAN;

SCHLUP, 1998), and Non-Linear Principal Component Analysis (DONG; MCAVOY, 1996; DONG; MCAVOY; CHANG, 1995).

4 EVOLUTIONARY ALGORITHMS

Within the field of Computational Intelligence (CI), Evolutionary Algorithms (EAs) are positioned as a subclass of Evolutionary Computing (EC). More specifically, EAs are stochastic search algorithms inspired by Charles Darwin's theory of natural evolution and can be used as a tool to solve a large number of real problems and demands (VIKHAR, 2016).

Although there are several ways to implement Darwin's concepts, basically all of them have in common the concept of simulation of the evolutionary process, *i. e.*, given a population of individuals, in the same environment and with limited resources, the competition for these resources implies the natural selection of the fittest (more adapted) individuals. As described by Charles, this evolution and competition are performed through several processes of biological evolution, such as recombination (crossing over) and mutation. When represented computationally, these processes are called genetic operators (LINDEN, 2008a; LINDEN, 2008b). Some of these processes will be briefly expanded and exemplified in the next section.

EAs can also be understood as optimization algorithms, which can be single-objective or multi-objective (as defined in 3.1) (*e.g.*, Multi-Objective Evolutionary Algorithms (MOEAs)), and, therefore, have a loss function that can be maximized or minimized and indicates which are the best solutions to the problem. Making a parallel between evolutionary concepts and optimization problems, individuals of a particular population are the possible candidate solutions to the problem. Each individual's aptitude indicates the quality of each solution and is obtained by applying the loss function to each individual — for a minimization problem, the lower the value the individual receives, the better it is, to maximize the opposite is valid.

It is important to emphasize that several steps in the evolutionary process of an EA are stochastic and probabilistic. For this reason, the results obtained are generally difficult to replicate — furthermore, the result found by the algorithm is not necessarily the optimal solution (EIBEN; SMITH et al., 2003; LINDEN, 2008a). The basic scheme of an evolutionary algorithm can be seen in Algorithm 1.

As already mentioned, there are several approaches and ways to implement an evolutionary algorithm, all similar within Darwinian concepts, but differing in some details as the form of representation of the individual (*e.g.*, binary, tree, etc.) — which is interesting because a form of representation can be better adapted to a specific type of problem. Among the best-known approaches are Genetic Algorithms (GAs) (GOLDBERG; HOLLAND, 1988; HOLLAND, 1975), Evolutionary Strategies (ESs) (RECHENBERG, 1965; RECHENBERG, 1978), Evolutionary

Algorithm 1 AN EVOLUTIONARY ALGORITHM

```
$pop = initialpopulation$                                    ▷ Random candidate solutions
evaluate each candidate $(pop)$
while it doesn't satisfy stop condition do  ▷ Stop condition can be by time, evaluation, etc.
    $pop_{new}$ = parents selection $(pop)$
    apply recombination $(pop_{new})$
    apply mutation in $(pop_{new})$
    evaluate each new candidate in $(pop_{new})$
    select the new generation in $(pop, pop_{new})$
end while
```

Source: Adapted from Eiben, Smith et al. (2003) and Linden (2008a)

Programming (EP) (FOGEL, 1962), and Genetic Programming (GP) (KOZA, 1992; KOZA, 1994). Many of these solutions are used to solve system identification problems, such as determining the regressors of a NARMAX model or estimating a set of parameters for regressors in a NARX model.

An example of EAs in system identification is Aguirre, Barbosa and Braga (2010), who implemented a GA to determine the parameters of a NARX/NARMAX model. It was applied using both a single-objective and a multi-objective approach with real representation. Each *locus* in a gene represented a parameter to be estimated by the algorithm. In another example, Li and Jeon (1993), a GA was used to detect which regressors were most significant for a NARMAX model to avoid overparameterization. Individuals were represented in a binary form in which the number 1 indicated the presence of a possible regressor addressed in that position and the presence of the number 0 the opposite. Other works that used GAs in system identification are Chen et al. (2007) and Barbosa et al. (2011).

In addition to genetic algorithms, other EA approaches have been successfully applied to solve identification problems. This is the case with Genetic Programming. For example, in Rodriguez-Vazquez, Fonseca and Fleming (2004), a multi-objective approach using GP was proposed to find NARX polynomials with at least two criteria (predictive accuracy and complexity) and seven requirements (*e.g.*, model degree, model lag, among others). It was found that the multi-objective approach in conjunction with genetic programming achieves good results for determining regressors in a system. Another work in which GP was used, with good results in systems identification, also in the problem of defining the structure of NARX models, was in Madár, Abonyi and Szeifert (2005). In this work, it was proposed to use LS/ERR together with the GP approach, removing the regressors with low ERR value (less significant), to improve the performance of the models.

The GP evolutionary approach and its expansion, the Multi-Gene Genetic Programming (MGGP), will be better presented in the following sections.

4.1 Genetic Programming

Genetic Programming (GP) (KOZA, 1992; KOZA, 1994) is an approach to evolutionary algorithms systematized and developed by John R. Koza in 1992. In this approach, individuals (candidate solutions) can be computer programs, arithmetic expressions, or formulas capable of solving a computational problem. In an iterative way (generations), GP evolves the population of individuals, applying *genetic operators* (*e.g.*, crossover (sexual recombination), mutation, reproduction) to obtain, on average, a new generation of computer programs better able to solve the problem. Like other EAs (*e.g.*, Genetic Algorithms), each individual's fitness is determined by evaluating the candidate using a loss function (VIKHAR, 2016).

GP is also understood as an extension of Genetic Algorithms in which the individuals of the population do not have a fixed structure of characters in a string, but, as already mentioned above, are computer programs that are dynamically built during the evolution of the population (KOZA; POLI, 2005) — a feature that gives more flexibility to this approach.

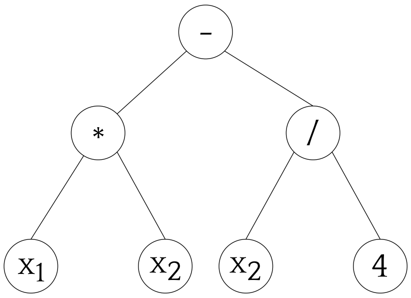
4.1.1 Genetic Programming Representation

As aforementioned, it is not usual in Genetic Programming to have a fixed representation of characters in a string. However, generally tree-based encoding is used, which by organizing its elements hierarchically, can synthesize mathematical functions, logical formulas, programs, to mention some possibilities. In this type of representation, it is necessary to define the syntax of the trees. This is done by defining the function set and terminal set. The function set is composed of arithmetic functions $(+,-,*,/,min,max,...)$, also called nodes. There are two types of nodes, the root, from which the entire tree derives, and the internal ones, which give rise to the various branches of the tree. The elements of the terminal set, on the other hand, are known as leaves and can be variables and constants. In Figure 4.1, it is possible to see the function $f(x_1,x_2) = x_1 * x_2 - (x_2/4)$ encoded in the tree representation.

4.1.2 Genetic Programming Selection

According to Koza and Poli (2005), tournament selection and fitness-proportionate selection are the most used methods to select individuals in a population to create offspring and

Figure 4.1 – Tree representation. The function $f(x_1, x_2) = x_1 * x_2 - (x_2/4)$ encoded in tree representation.



Source: Author (2022)

apply genetic operators. In the tournament selection method, some individuals are selected randomly from the population, a comparison is made between their aptitudes, and the best individual is then selected to be a parent. The fitness-proportionate selection method follows the same non-greedy selection principle, *i.e.*, both individuals, inferior and superior in fitness, can be selected — which is essential for the non-premature convergence of the algorithm, that is, to maintain a diverse population during execution.

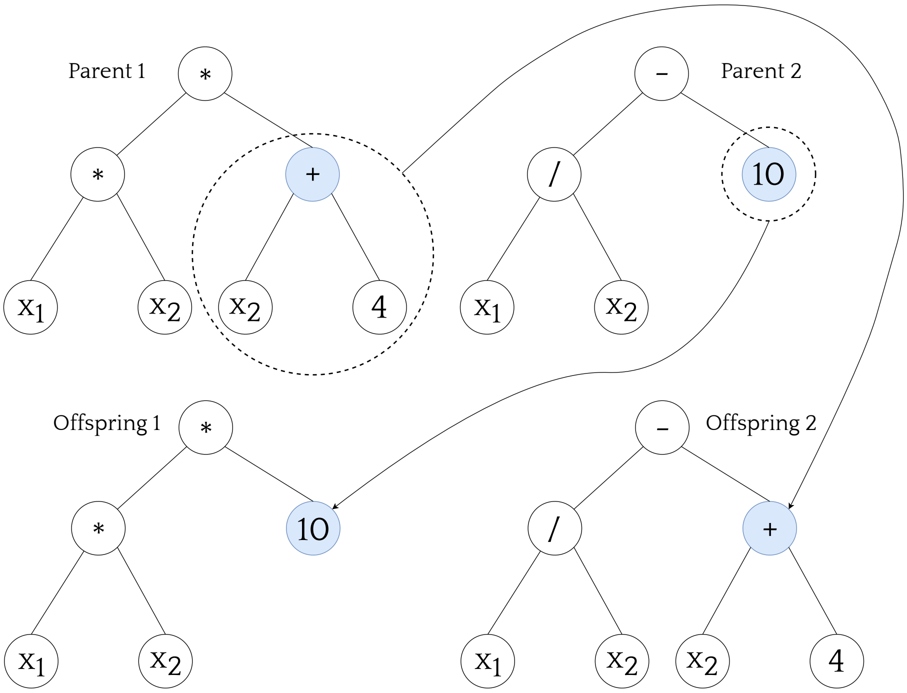
4.1.3 Genetic Programming Recombination Operators

Similar to other EA approaches (*e.g.*, GA), in Genetic Programming, recombination to generate offspring is done by exchanging genetic material between selected parents in the population. According to Eiben, Smith et al. (2003), the *subtree crossover* is the most common form of recombination implemented in GPs. In this operator, two individuals are randomly selected from the population and, in each one, a node is randomly chosen. Then, subtrees created from the selected nodes in each of the parents are swapped between them, generating offsprings. An example of the genetic recombination operator is illustrated in Figure 4.2.

4.1.4 Genetic Programming Mutation Operators

Unlike other EA approaches (*e.g.*, GA), it is not common sense to use the mutation operator in Genetic Programming. In Koza (1992), it is indicated to *use only the recombination operator* (0% mutation rate), which is considered sufficient, as it acts as a macro mutation and strongly modifies the individual. However, other works indicate that using mutation (Banzhaf et al. (1998) indicate 5% mutation rate) together with recombination can lead to better results (EIBEN; SMITH et al., 2003).

Figure 4.2 – GP subtree crossover operator. The parent individuals $(x_1 * x_2) * (x_2 + 4)$ and $(x_1/x_2) - (10)$ exchange subtrees and generate two offspring: $(x_1 * x_2) * (10)$ and $(x_1/x_2) - (x_2 + 4)$.



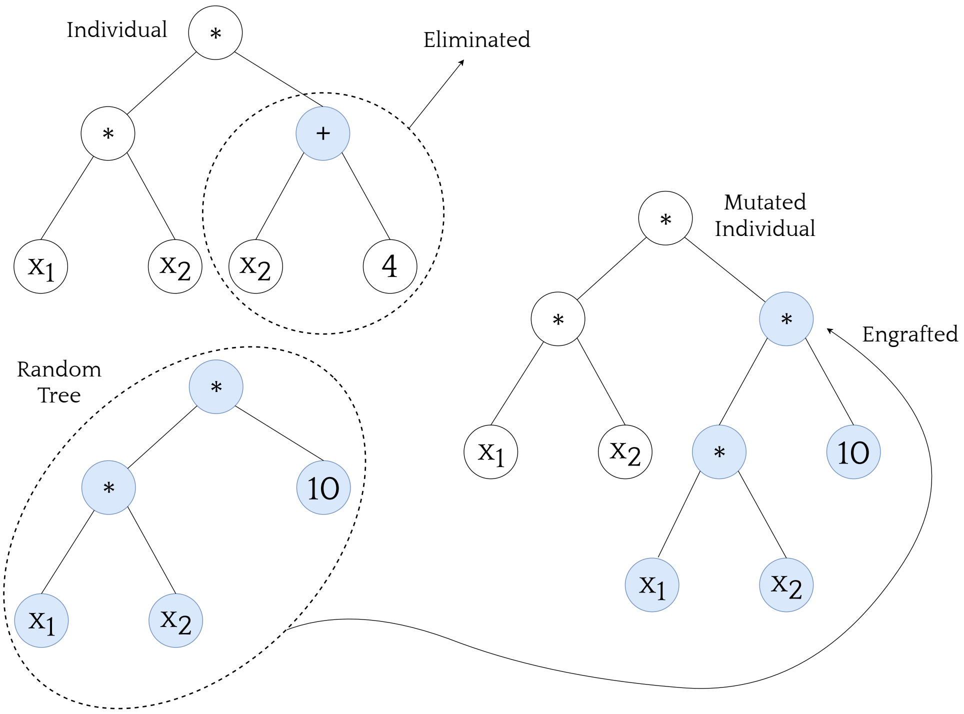
Source: Author (2022)

When used, the most common mutation operator is the *subtree mutation*. In this operator, a tree node is chosen randomly; thus, all branches and leaves below this node (including the chosen node) are replaced by another tree generated randomly (exactly like the initial population of the GP). An example of mutation is shown in Figure 4.3. The initial $(x_1 * x_2) * (x_2 + 4)$ tree has its $(x_2 + 4)$ branch replaced by the $(x_1 * x_2) * (10)$ random tree, generating the new $(x_1 * x_2) * (x_1 * x_2) * (10)$ tree.

4.2 Multi-Gene Genetic Programming

Multi-Gene Genetic Programming (MGGP) (HINCHLIFFE, 2001; HINCHLIFFE; WILLIS, 2003) was introduced by Hinchliffe et al. (1996) and is considered an extension of the standard version of Genetic Programming. Unlike GP, in which each individual (program) in the population is formed by only a single tree, in MGGP, an individual is formed by a weighted linear combination of a number of GP trees. Using the concepts of EA, an individual MGGP can be considered a chromosome, in which its *genes are GP individuals* (also called gene-trees). This concept can be expressed mathematically, as well as used in system identification, as the weighted sum of the outputs of a number of functions (basis functions) of the model inputs, as follows (HINCHLIFFE; WILLIS, 2003):

Figure 4.3 – GP subtree mutation operator. The initial $(x_1 * x_2) * (x_2 + 4)$ tree has its $(x_2 + 4)$ branch replaced by the $(x_1 * x_2) * (10)$ random tree, generating the new $(x_1 * x_2) * (x_1 * x_2) * (10)$ tree.



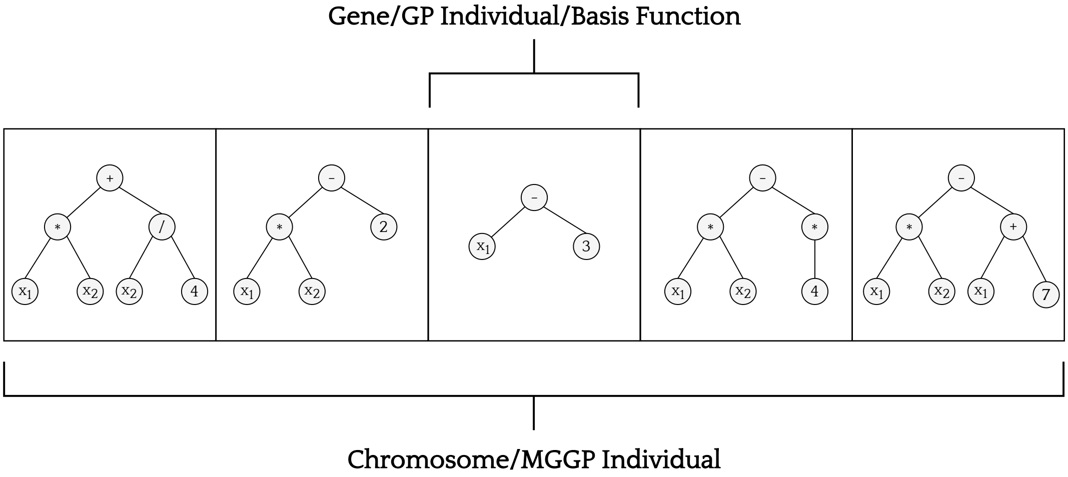
Source: Author (2022)

$$g(\varphi, \Theta) = \sum_{i=1}^{m} \theta_i g_i(\varphi), \tag{4.1}$$

where m is the number of basis functions, the $g_i(\varphi)$ represents individual functions (genes/GP individuals), and the θ_i are model parameters. An example of a generic MGGP individual is illustrated in Figure 4.4.

As mentioned by Orove, Osegi and Eke (2015), the number of regressors, types of basis functions, and the structure of the trees that make up the individual evolve automatically during the execution of the algorithm — the individual is limited only by the restrictions defined by the designer (*e.g.*, the maximum number of basis functions, maximum tree depth). This feature is interesting because it gives more flexibility and adaptability during the algorithm's execution, which does not happen with other classic modeling techniques in which the final model is restricted by definitions elaborated before running the algorithm (*e.g.*, types of basis functions) (HINCHLIFFE, 2001).

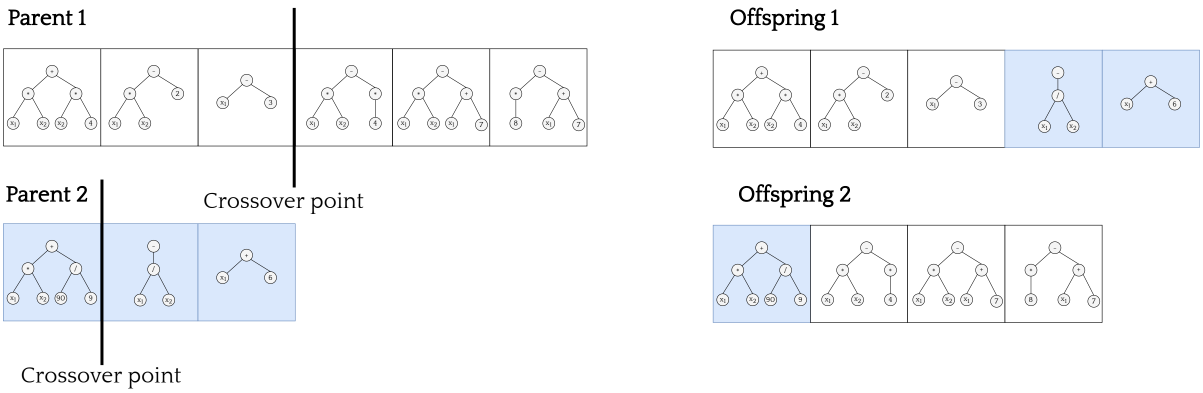
Figure 4.4 – Generic MGGP individual. The MGGP individual is formed by a weighted linear combination of GP trees.



Source: Author (2022)

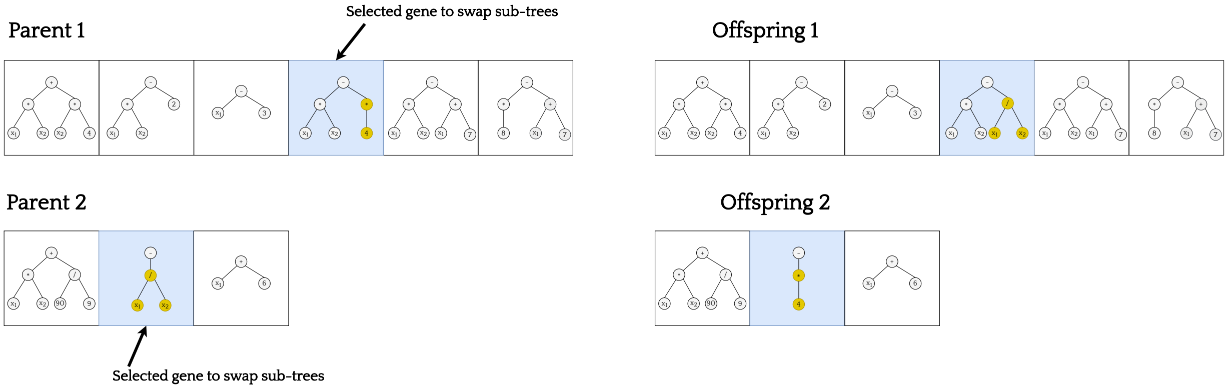
Another difference between MGGP and GP is in the genetic recombination operators. The two most common are *high-level crossover* and *low-level crossover*. The first performs an exchange of entire genes (basis functions/individuals GP) between individuals in a similar way to the one-point crossover GAs recombination operator, *i.e.*, a position on the parent chromosome is chosen at random (the position can differ between the parents) by dividing it in two and then exchanging the resulting parts between them (this process is illustrated in Figure 4.5). Finally, the low-level crossover exchanges genetic material between genes (the gene sub-trees) of each parent, *i.e.*, a gene is chosen randomly in each parent, and then the exchange of material between these GP individuals is carried out using the subtree crossover operator, just as it is done in the Genetic Programming approach (this process is illustrated in Figure 4.6).

Figure 4.5 – MGGP High-Level Crossover.



Source: Author (2022)

Figure 4.6 – MGGP Low-Level Crossover.


Parent 1
Selected gene to swap sub-trees
Offspring 1
Parent 2
Offspring 2
Selected gene to swap sub-trees

Source: Author (2022)

4.3 Multi-Gene Genetic Programming in System Identification

Like other EA approaches, MGGP has also been applied to system identification problems. For example, in Ghareeb and Saadany (2013), MGGP was used to create a prediction model (Short Term Load Forecasting (STLF) problem) for power system operation of an Egyptian electrical network. The dataset used consists of 39 weeks and included the maximum and minimum temperature of the day and the corresponding current peak load. In order to verify the accuracy of the model found by MGGP, the same dataset was applied to the Radial Basis Function (RBF) network and the standard Genetic Programming. The results found demonstrate superiority in the prediction accuracy of the MGGP model.

Another work that developed models through MGGP to perform forecasts is presented in Niazkar and Niazkar (2020). In this study, MGGP models were found to perform trend predictions of COVID-19 cases in seven different countries (*i.e.*, China, Korea, Japan, Italy, Singapore, Iran, and the USA). Moreover, the cases estimated by the proposed models were acceptably close to the actual observed values, which indicates that models developed by MGGP lead to promising results. In Mehr and Kahya (2017), a Pareto-optimal Moving Average MGGP approach was proposed to perform predictions of daily streamflow. The results were compared with standalone GP, MGGP, and conventional Multivariate Linear Regression (MLR), which was found superior to all of these in both prediction accuracy and parsimoniousness.

In Castro and Barbosa (2020), an MGGP/ERR hybridization was introduced to select structures (regressors) for NARMAX models. The back-shift operator, q^{-1}, presented in Hinchliffe and Willis (2003), was used to determine the delay variables. Three datasets were used, two test systems with short-term dependencies and a real dataset related to a hydraulic pump.

The models obtained by MGGP/ERR were compared with those obtained by the LS/ERR approach as a reference. The LS/ERR algorithm obtained better results than the MGGP/ERR algorithm in selecting structures for systems with short-term dependencies. As for the hydraulic pump dataset, the MGGP/ERR approach was superior, and it was possible to use higher degrees of non-linearity — which would demand a much higher computational cost, even unfeasible if the LS/ERR approach were used.

5 MATERIALS AND METHODS

5.1 Proposed Algorithm

This dissertation proposes a multi-objective MGGP approach to identifying non-linear systems, specifically in modeling soft sensors for the replacement of PDGs in the deepwater oil extraction industry. A toolbox developed by Castro (2021) is used to implement the MGGP approach. In addition to the classic genetic operators (presented in Section 4.2), the toolbox presents two other mutation operators: high-level mutation and low-level mutation - both inspired by two recombination operators previously presented (*i.e.*, high-level crossover and low-level crossover).

The first, high-level mutation, randomly selects a gene from the individual and replaces it with a new one. The second, low-level mutation, acts within the individual's MGGP gene, exchanging a subtree for a new randomly generated subtree, precisely like the GP subtree mutation operator (presented in Subsection 4.1.4). Another relevant point is that, following the guidelines presented for GP in Poli, Langdon and McPhee (2008), an individual in which the recombination operator is applied cannot also experience the mutation operator. The parameters to configure and run the MGGP toolbox are:

- population size (*popSize*): defines the population size of individuals that the algorithm will use;
- crossover probability ($CXPB$): defines the probability that a pair of MGGP individuals crossover through one of the recombination genetic operators;
- mutation probability ($MTPB$): defines the probability of an individual being mutated through one of the mutation operators, if it has not participated in any recombination operation;
- maximum GP height ($maxHeight$): limits the maximum size of the GP tree in relation to its height;
- maximum number of MGGP terms (*maxTerms*): limits the maximum number of regressors an MGGP individual can have;
- elite size ($elite$): sets the percentage of individuals from the previous generation's population that can remain in the next generation;

- number of variables (*numberOfVariables*): defines the number of total variables that the algorithm will process (inputs and outputs). It is important to emphasize that the toolbox only works with single-input single-output (SISO) and multiple-input single-output (MISO) models;
- maximum delay (*maxDelay*): defines the maximum value that the back-shift operator ($q^{-1}, q^{-2}, ..., q^{-i}$) can apply to the model's regressors. This operator is responsible for automatically determining the lag in the models;
- functions set: by default the toolbox has only the multiplication function in the primitive set of functions used as nodes in GP individuals. However, it is possible to add other functions to the set (*e.g.*, division, exponentiation, etc.).

The proposed MGGP execution routine is similar to other EA approaches. It starts with a random population of individuals that is evaluated. Once this is done, the generational natural evolution process starts (in a loop) as follows:

1. Using natural selection by tournament, parents are selected;
2. Each pair receives a random recombination probability. If smaller than CXPB, the crossover happens;
3. Individuals that did not undergo recombination receive a random probability of mutation. If smaller than MTPB, the mutation happens;
4. The new individuals are then evaluated;
5. The elitism operator is applied to select the best offspring plus the elite of the previous generation.

Upon reaching the pre-established number of generations, the execution is completed, and, as it is a multi-objective problem, a Pareto-optimal is generated. Then, the Pareto individuals are validated, and a decision criterion is applied to choose the most appropriate MGGP model for the problem (the decision criteria is described in Subsection 5.2.3). In the diagram illustrated in Figure 5.1, the algorithm flowchart is presented.

It is essential to highlight that for the multi-objective approach of MGGP (MGGPMO) present in the toolbox, the genetic operators, the individual representation structure, the concept of evolution and evaluation of the standard MGGP algorithm are implemented on the Non-dominated Sorting Genetic Algorithm II (NSGA-II) (DEB et al., 2000; DEB et al., 2002) framework of the Distributed Evolutionary Algorithms in Python (DEAP) library. In this way, the concepts of Pareto dominance and crowding distance are used to select the parents for genetic operators application and select the individuals for the next generation; a brief explanation of these processes is presented in Subsection 5.3.

Finally, it is essential to emphasize that MGGP models use NARX polynomial (in a tree approach) representation and the programming language used in the implementation is Python.

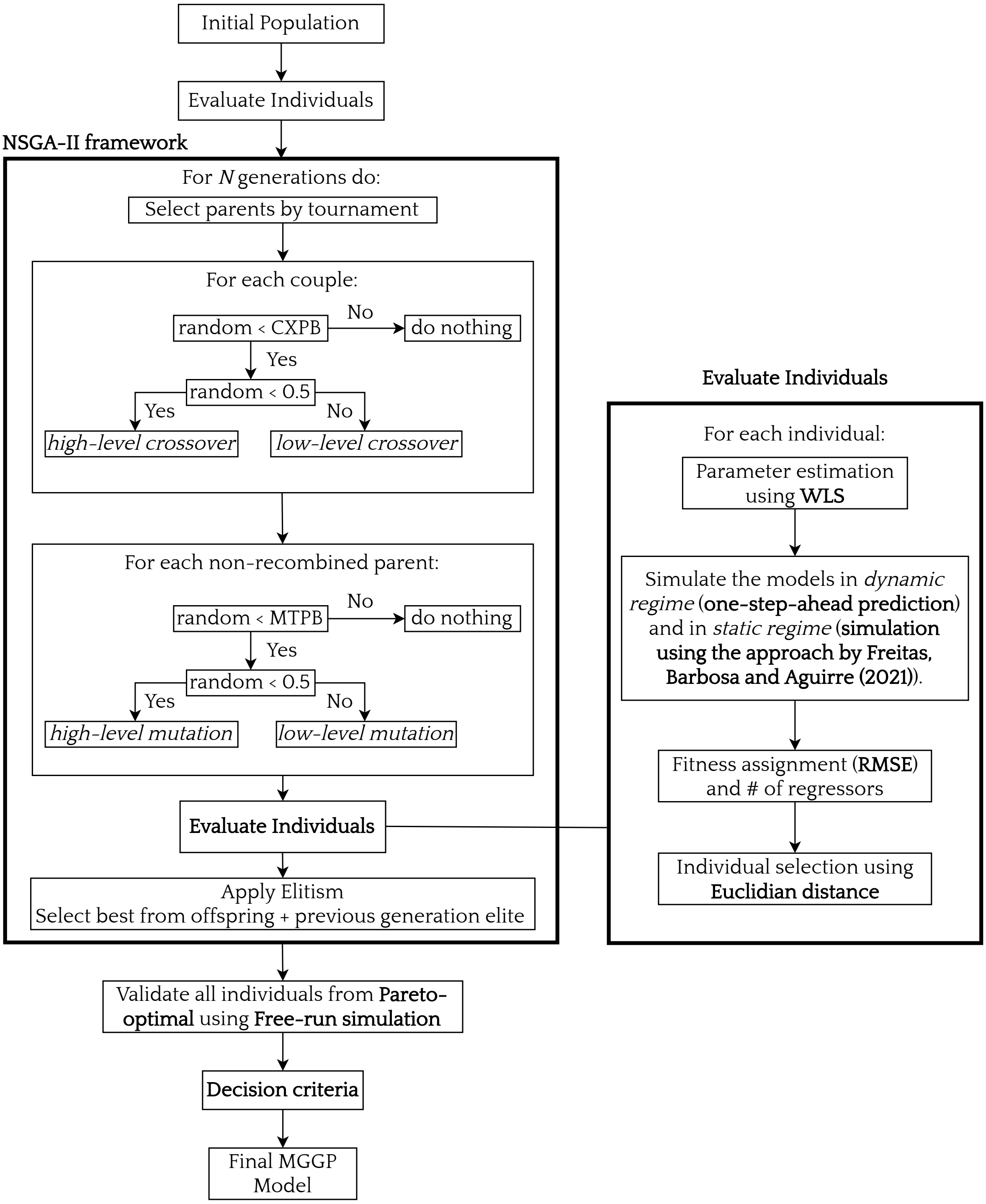
5.2 Individuals Evaluation

This MGGP proposal differs from the standard approach in step 4, described in Section 5.1, *i.e.*, during the evaluation of new individuals. Seeking to overcome the expected difficulty of datasets obtained from historical data not being sufficiently informative about different points of operation of the system, which is the case in the present work, the auxiliary information is used during the modeling process (grey-box modeling). Auxiliary information is any missing information (*e.g.*, symmetry properties of the system (CHEN et al., 2008), static nonlinearity (AGUIRRE; ALVES; CORRÊA, 2007), etc.) not represented in the dynamic dataset used during the modeling process (FREITAS; BARBOSA; AGUIRRE, 2021). This work used static regime data (steady-state data) as auxiliary information.

In order to implement the use of auxiliary information in MGGP, a multi-objective approach was necessary, as already mentioned briefly. Thus, the evaluation function minimized by MGGP has three objectives, namely *i.* minimize the error in the dynamic regime, *ii.* minimize the error in the static regime and *iii.* minimize the number of regressors in the MGGP model. For the first objective, the models are simulated using the one-step-ahead prediction. For the second objective, the simulation is performed using the approach presented in Freitas, Barbosa and Aguirre (2021) (in Subsection 5.2.1 the concept is presented). In both cases, the RMSE is calculated between the value found by the simulations and the real value coming from the dataset.

Another relevant point for evaluating the individual is parameter estimation. This step takes place before simulating the models for later RMSE calculation. In this algorithm, the parameters of the models are calculated using weighted least squares (the concept is presented in Subsection 3.2.4.2). This approach is advantageous because it allows weighting the importance

Figure 5.1 – Algorithm flowchart.

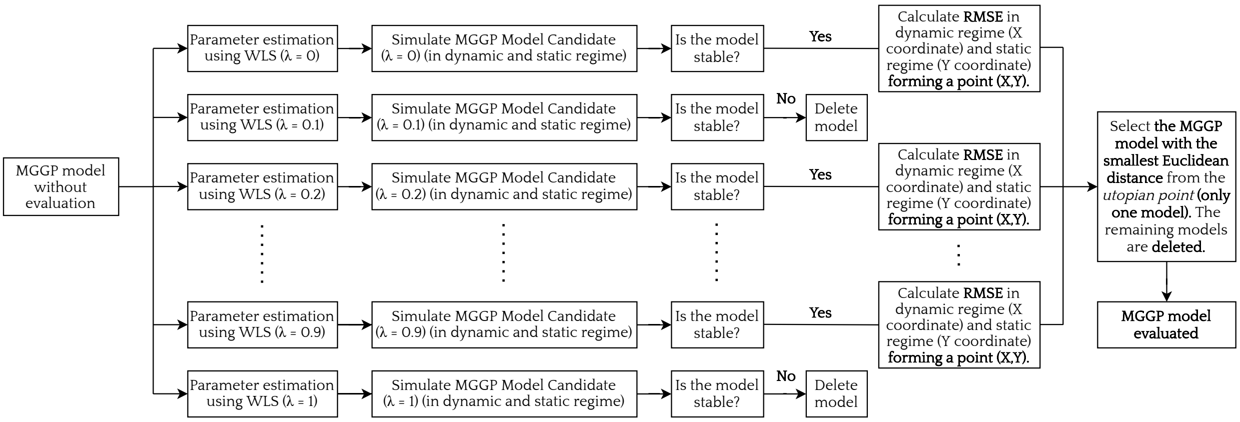

Initial Population
Evaluate Individuals
NSGA-II framework
For *N* generations do:
Select parents by tournament
For each couple:
random < CXPB
No
do nothing
Yes
random < 0.5
Yes
No
high-level crossover
low-level crossover
For each non-recombined parent:
random < MTPB
No
do nothing
Yes
random < 0.5
Yes
No
high-level mutation
low-level mutation
Evaluate Individuals
Apply Elitism
Select best from offspring + previous generation elite
Validate all individuals from **Pareto-optimal** using **Free-run simulation**
Decision criteria
Final MGGP Model
Evaluate Individuals
For each individual:
Parameter estimation using **WLS**
Simulate the models in *dynamic regime* (**one-step-ahead prediction**) and in *static regime* (**simulation using the approach by Freitas, Barbosa and Aguirre (2021)**).
Fitness assignment (**RMSE**) and # of regressors
Individual selection using **Euclidian distance**

Source: Author (2022)

of dynamic data and static data in estimating model parameters. In this work, this weighting is done by a λ variable that varies from 0 to 1, with steps of 0.1, where 0 indicates the lack of static regime data in the estimation of the parameters using WLS and 1 the lack of dynamic regime data.

During the algorithm's execution, each candidate for the final MGGP model, formed by a structure of regressors, has its parameters estimated eleven times, one for each λ value. In this way, eleven models are generated for the same structure of regressors. These eleven models are evaluated in dynamic and static regimes, immediately eliminating unstable models. Among the remaining models, the Euclidean distance concept is used (the way this concept is used is better explained in Subsection 5.2.2) to select the best model, and the other are discarded. The only remaining model is returned to the MGGP population, with its evaluation for the three objectives. This same process is repeated for all individuals in the population, during the evaluation step, at each generation of the algorithm. This process is also presented visually in Figure 5.2 for better understanding.

Figure 5.2 – Individual evaluation process.


MGGP model without evaluation
Parameter estimation using WLS (λ = 0)
Parameter estimation using WLS (λ = 0.1)
Parameter estimation using WLS (λ = 0.2)
Parameter estimation using WLS (λ = 0.9)
Parameter estimation using WLS (λ = 1)
Simulate MGGP Model Candidate (λ = 0) (in dynamic and static regime)
Simulate MGGP Model Candidate (λ = 0.1) (in dynamic and static regime)
Simulate MGGP Model Candidate (λ = 0.2) (in dynamic and static regime)
Simulate MGGP Model Candidate (λ = 0.9) (in dynamic and static regime)
Simulate MGGP Model Candidate (λ = 1) (in dynamic and static regime)
Is the model stable?
Yes
No
Delete model
Calculate RMSE in dynamic regime (X coordinate) and static regime (Y coordinate) forming a point (X,Y).
Select the MGGP model with the smallest Euclidean distance from the utopian point (only one model). The remaining models are deleted.
MGGP model evaluated

Source: Author (2022)

Finally, as already mentioned, the algorithm's execution ends when the number of predefined generations is reached. Then, all the resulting Pareto-optimal models are validated through free-run simulation, both in dynamic and static regimes. In the case of the dynamic regime, the dynamic validation dataset, Z_v, is used. For the static regime, the same dataset used for training, Z_s, is used; however, as already mentioned, the free-run simulation is used in the validation. Once this is done, the decision criteria (presented in Subsection 5.2.3) is used to choose the most appropriate final MGGP model.

5.2.1 Approach by Freitas, Barbosa and Aguirre (2021) for static data

When trying to use auxiliary information about the static regime, together with the dynamic data of a system, free-run simulation is often used to minimize the error (*e.g.*, MSE, RMSE, etc.) in the static regime. The cost function, J_s, that is commonly used is as follows:

$$J_s = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i(k) - \hat{y}_i(k))^2}, \tag{5.1}$$

where $\hat{y}_i(k)$ corresponds to the free-run simulation for each operating point (model fixed points), N is the number of operating points in the dataset (number of samples in Z_s), and $y(k)$ the real value of the operating point coming from the dataset. It is possible to see that when computing $\hat{y}_i(k)$, it is necessary to find the fixed points of the model, which is usually a high computational cost task. Seeking to overcome this, Freitas, Barbosa and Aguirre (2021) sought a way in which the fixed points of the model do not need to be explicitly calculated computationally or analytically. This is done by minimizing the following cost function (FREITAS; BARBOSA; AGUIRRE, 2021):

$$\hat{J}_s = \sqrt{\frac{1}{N_s}\sum_{i=1}^{N_s}(\bar{y}_i - F\left(\bar{\psi}_i, \hat{\theta}\right))}, \tag{5.2}$$

where the hat over J_s indicates that 5.2 is an approximation of 5.1. $\bar{y}_i$ can be seen as "target value" coming from the static data, *i.e.*, the operating points in static regime, $\bar{\psi}_i = [1 \;\; \bar{y}_i \cdots \bar{y}_i \;\; \bar{u}_i \cdots \; \bar{u}_i] \in \mathbb{R}^{1+n_y+n_u}$ and $\bar{u}_i$ the corresponding input for each output $\bar{y}_i$. It needs to be noted that $F\left(\bar{\psi}_i, \hat{\theta}\right)$ is simply the one-step-ahead prediction of the model (FREITAS; BARBOSA; AGUIRRE, 2021).

Freitas, Barbosa and Aguirre (2021) claim that both 5.1 and 5.2, computed over Z_s, for each input $\bar{u}_i$, have global minima $J_s = \hat{J}_s = 0$ at the model fixed points $\bar{y}_i$, $j = 1, \cdots, N_s$. Proof for this lemma and more information can be found in (FREITAS; BARBOSA; AGUIRRE, 2021).

5.2.2 Euclidean distance model selection

Following the flow of the MGGP algorithm, within the individual evaluation stage, right after estimating the parameters (for eleven different λ values), the models are simulated in dynamic and static regimes (unstable models are discarded), as it is possible to observe in Figure 5.2. The output values obtained for both regimes are then compared with the real values from the dataset, and the RMSE is calculated for the dynamic regime and the static regime.

To determine which model, among the others, is the best, the Euclidean distance is used. For this concept to be employed, each model is represented by a point with two coordinates $P_i = (RMSE_A, RMSE_B)$, where $RMSE_A$ is the RMSE in the model's dynamic regime and $RMSE_B$ is the RMSE in the model's static regime. Once this is done, L2 normalization is applied to the set of points representing the models to avoid distortions in the results caused by very discrepant error values.

After that, the utopian point ($U = (A,B)$) is defined. This point is formed by the smallest normalized value of RMSE in the static regime (A) and the smallest normalized value of RMSE in the dynamic regime (B) existing among the remaining stable MGGP models. These values will always be very close to zero due to normalization, which makes sense because the algorithm is doing a minimization (seeking the smallest possible value for the objectives, *i.e.*, zero). With the utopian point defined, the Euclidean distance between it and each of the existing points, eleven at the most, is calculated. The Euclidean distance, in two dimensions, is defined as follows:

$$d(U,P_i) = \sqrt{(A - RMSE_{A_i})^2 + (B - RMSE_{B_i})^2}, \tag{5.3}$$

where i represents the individual point P (can range from 1 to 11) and $d(U,P_i)$ is the Euclidean distance between point U and point P_i.

The model whose point has the smallest Euclidean distance is chosen because it has RMSE values, in both regimes, closer to zero, *i.e.*, closer to the desired result – the other models are deleted. Two highlights are essential. The first is that this process is carried out for each population individual (once per generation). Second, the evaluations (RMSE values and the number of terms) that the chosen MGGP model returns together with itself (regressors and parameters) to the population are the original non-normalized values.

5.2.3 Decision criteria

After finishing the execution of the multi-objective MGGP algorithm, all the models found and arranged in the Pareto-optimum are submitted to free-run simulation in dynamic and static conditions. It is worth noting that, just as it is done during the algorithm's execution, all models have their parameters estimated eleven times (one for each λ value); however, differently, all resulting models are stored in Pareto, regardless of the result of the simulation. For example, if at the end of the algorithm execution the Pareto has 100 MGGP models, after the simulation, the Pareto will be composed of 1100 MGGP models, *i.e.*, there will be models with precisely the same regressor structure but with different parameter values, which makes them different models.

That said, the simulation results are then compared, through the RMSE, with the real data. The error values encountered are then analyzed in order to choose the final MGGP model, as follows:

1. All unstable models in the dynamic regime, in the static regime or in both are excluded from Pareto;

2. All models are ranked with respect to their dynamic RMSE, from lowest value to highest;

3. The model with the smallest dynamic regime RMSE is chosen as the final MGGP model.

5.3 Parent selection and next-generation individuals selection

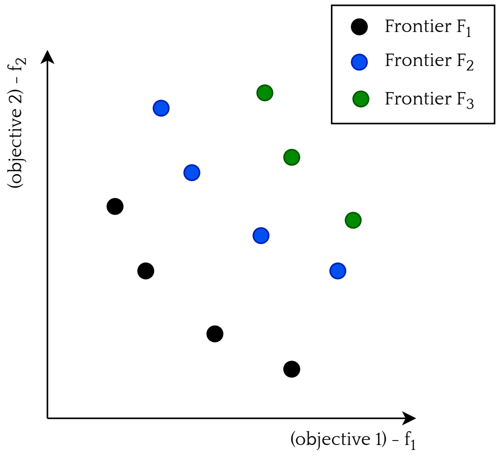
As already mentioned, the MGGPMO algorithm presented is built on the NSGA-II framework. It uses the concepts presented by Deb et al. (2002) of crowding distance and non-dominated ranking when selecting parents to apply the MGGP genetic operators and when selecting individuals for a new generation.

Therefore, following the diagram presented in Figure 5.1, after creating the initial population and evaluating each individual, an initial Pareto set is created. The next step is to select individuals (parents) to apply genetic operators and generate offspring. To this end, NSGA-II performs two calculations for each individual in the population.

The first is the degree of non-dominance of each individual in the Pareto set. In which individuals are classified according to the frontier they belong to. The first frontier (F_1), the optimal frontier, is composed of individuals that any other individual does not dominate; they receive zero as degree (the lower the degree, the better the model). The individuals of the second frontier (F_2) are the individuals dominated only by the individuals of the F_1 and receive one as degree; this same process continues until all individuals are addressed in a frontier and have their degree of non-dominance. For more information on how the degree of non-dominance is calculated, see Deb et al. (2002). Figure 5.3 illustrates and exemplifies the concept of boundaries described.

The second value calculated for each individual is the crowding distance d_c. This calculation is done using the average distance of the two adjacent solutions (known as cuboid, shown in Figure 5.4) to the individual in which d_c is being computed. For a two-objective optimization, the calculation of d_c is done as follows:

Figure 5.3 – Non-dominated sorting.

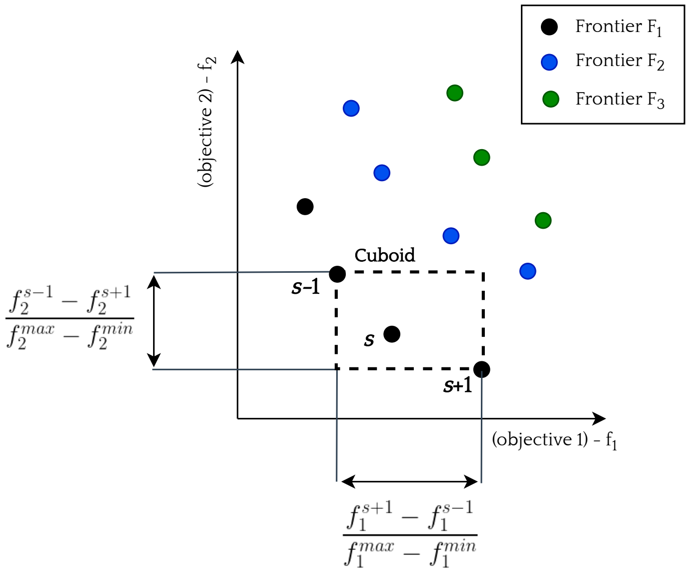


Source: Author (2022)

$$d_c = \frac{f_1^{s+1} - f_1^{s-1}}{f_1^{max} - f_1^{min}} + \frac{f_2^{s-1} - f_2^{s+1}}{f_2^{max} - f_2^{min}} \tag{5.4}$$

where s is the individual on which d_c is being calculated, $s+1$ and $s-1$ are the two individuals adjacent to s, $f_1^{(s+1)}$ is the fitness of the first objective of the individual $s+1$, $f_1^{(s-1)}$ is the fitness of the first objective of the individual $s-1$, f_1^{max} is the highest fitness found for the first objective in the population and f_1^{min} is the lowest fitness found for the first objective in the population. $f_2^{(s+1)}$, $f_2^{(s-1)}$, f_2^{max}, and f_2^{min} are, respectively, the same values already mentioned but calculated for the second optimization objective.

Figure 5.4 – Calculation of the crowding distance d_c.

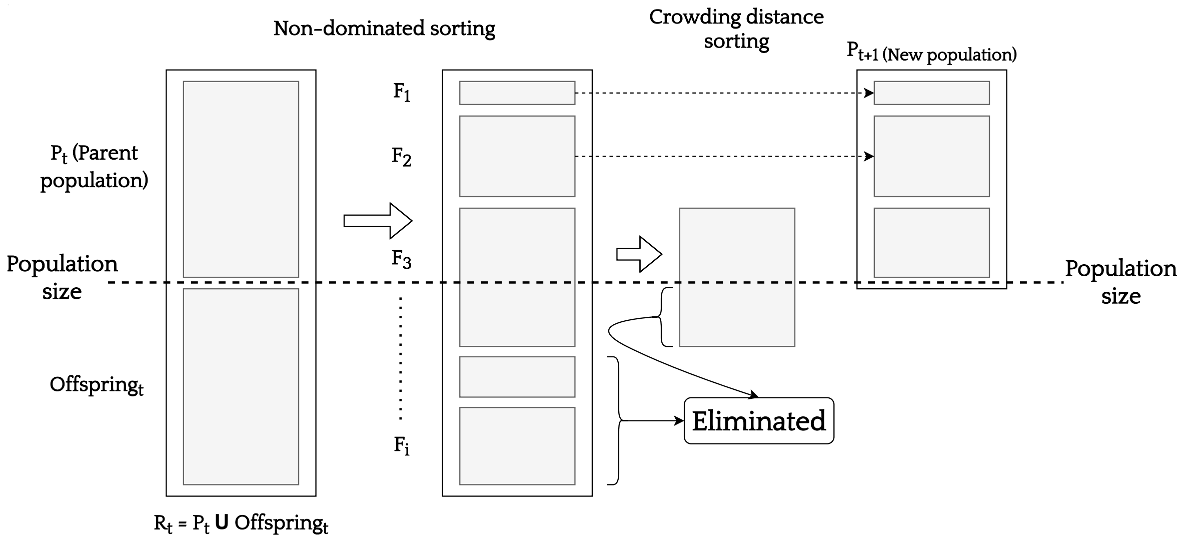

Frontier F1
Frontier F2
Frontier F3
(objective 2) - f2
(objective 1) - f1
Cuboid
s-1
s
s+1
$\frac{f_2^{s-1} - f_2^{s+1}}{f_2^{max} - f_2^{min}}$
$\frac{f_1^{s+1} - f_1^{s-1}}{f_1^{max} - f_1^{min}}$

Source: Adapted from Deb et al. (2002)

With these two values calculated for all individuals, the tournament selection operator called crowded-comparison is applied. In this method, two individuals are randomly selected from the population, and their non-dominance degree values are compared; the individual with the lowest value is chosen to be one of the parents. If the two individuals are on the same frontier, *i.e.*, having the same non-dominance value, the value of d_c is compared, the individual with the highest value is chosen. The same procedure is performed once more to select one more parent. With all pairs of parents formed, the algorithm follows and applies the MGGP genetic operators, as shown in the diagram in Figure 5.1.

Another point of the proposed algorithm in which the degree of non-dominance and crowding distance are used is in the application of elitism at the end of a generation to select which offspring and individuals from the previous generation will be the "survivors" for the next. This is done as follows. First, the individuals are ranked by the degree of non-dominance, the individuals with the lowest degree are selected for the next generation, if the F_1 is not formed by the necessary amount of individuals to form the population, the individuals of the F_2 are selected, this process continues until to exceed the number of individuals in the population defined by the user. The individuals from the last frontier added to the population are then ranked by their value of d_c until reaching the number of individuals needed to complete the population, the individuals left over the limit, and from frontiers F_i not added to the population are eliminated. The described process is illustrated in Figure 5.5.

Figure 5.5 – Selection of individuals for new generation.



Source: Adapted from Deb et al. (2002)

6 A GREY-BOX MULTI-OBJECTIVE MGGP APPROACH

In this chapter, the algorithm proposed in Chapter 5 is applied to three different datasets, one dataset with simulated data and two datasets with real problem data. The simulated dataset is used to validate the methodology and justify the approach's choices made. Comparisons are made, such as different ways to calculate the utopian point in the Euclidean distance selection approach, comparison of model parameter estimation between LS, WLS, and ELS, comparison between minimization with two and three objectives, among other analyses.

The second dataset, the hydraulic pump dataset, is used to verify the generalization and efficiency of the methodology in real problems beyond oil and deepwater extraction. Finally, the methodology is applied to the deepwater oil well dataset, which is the motivational problem of this work. Each of these problems, with their respective dataset, are better explained in the following sections. It is also presented which toolbox parameters were used to carry out each of the experiments, and, finally, the results are presented, compared, and discussed.

6.1 The Piroddi and Spinelli (2003) model

To validate the proposed algorithm, a dataset with simulated data (described in the following Subsection 6.1.1) was used. This is interesting because it allows to know precisely if the MGGP algorithm finds the correct regressors for a model and if the parameters are being estimated correctly. Therefore, several experiments were carried out (presented in the following subsections), some already briefly mentioned, namely: different ways to calculate the utopian point in the Euclidean distance selection approach, comparison of model parameter estimation between LS, WLS, and ELS, comparison between minimization with two and three objectives, comparison between Freitas, Barbosa and Aguirre (2021) approach (FBA prediction) and free-run simulation for static data, comparison of the execution time of the proposed algorithm using the FBA prediction and using free-run simulation, finally, an analysis of the efficiency of the proposed methodology in correctly finding the regressors of the original model is presented.

6.1.1 Dataset

In Piroddi and Spinelli (2003) the following model was presented:

$$
\begin{aligned}
w(k) &= 0.75w(k-2)+0.25u(k-1)-0.2w(k-2)u(k-1),\\
y(k) &= w(k)+e(k),
\end{aligned}
\tag{6.1}
$$

where $u \in \mathbb{R}$ is the input, $w \in \mathbb{R}$ the noiseless output, $y \in \mathbb{R}$ the output with the noise $e(k) \sim \mathrm{WGN}(0, 0.1\sigma_w)$, where WGN stands for the White Gaussian Noise.

Performing simulations with 6.1, three datasets were generated: dynamical training dataset, Z_d, validation dataset, Z_v, and static dataset, Z_s. As in Freitas, Barbosa and Aguirre (2021), the Z_d was obtained using $u \sim \mathrm{WGN}(-0.02, 0.04)$ and had 100 samples ($N_d = 100$). On the other hand, the Z_v has 2000 samples ($N_v = 2000$), with no noise in the output ($e = 0$). Finally, the Z_s were obtained analytically and have 50 samples ($N_s = 50$) with values equally spaced within range $u \in [-1, 3]$ and with zero mean noise and standard deviation $\sigma = 0.02$ in the output. It is essential to mention that the Z_v data were generated over a broader operating range than the one used for Z_d to verify the generalizability of the obtained models.

6.1.2 Results and discussion

In the next subsections the results already mentioned in the introduction of Section 6.1 are presented and discussed.

6.1.2.1 Euclidean distance model selection approach

As presented in Chapter 5, the proposed methodology estimates the parameters using weighted least squares, which creates the need to select the best model among the eleven available models (the number of models may be smaller, as explained in Subsection 5.2.2). For this purpose, the Euclidean distance between the points, which represent the models, and a U point is used. Three different ways of defining the U point were tested, namely: the U point being the mean value between the analyzed points, the U point being the median among those analyzed, and, finally, point U being *the utopian value* among the points (the last approach is explained in Subsection 5.2.2). Three objectives were used as a cost function: *i.* minimize the error in the dynamic regime, *ii.* minimize the error in the static regime and *iii.* minimize the number of regressors in the MGGP model.

The parameters used in the MGGP toolbox to identify the system were the same for the three approaches mentioned above, differing only in the way of defining the U point, namely:

```
popSize = 100;
```

```
CXPB = 0.9;
MTPB = 0.1;
n_gen = 250;
maxHeight = 5;
maxTerms = 20;
elite = 10;
maxDelay = 3 (q1,q2,q3);
numberOfVariables = 2;
primitive function: multiplication function.
```

The results of these experiments consist of the mean RMSE value and its respective standard deviation of each of the three objectives for each of the three approaches. To obtain these results, the algorithm was run thirty times for each approach, and, in this way, ninety Pareto-optimal were generated. Then, all stable models of these Pareto-optimal were submitted to free-run simulation in static and dynamic regimes, and their respective RMSE was calculated. The decision criterion (presented in Subsection 5.2.3) is used in each Pareto-optimal selecting the best model, so thirty models are selected for each approach; it is on the RMSE values of these models that the final result is calculated. The results for each of the approaches are shown in Table 6.1 together, for comparison purposes, with the RMSE values of Model 6.1 submitted to free-run simulation.

Table 6.1 – Error results (RMSE) in free-run simulation for dynamic (Z_v) and static (Z_s) regime. Mean and standard deviation for thirty runs of each approach.

Approach	**RMSE (dynamic regime)**	**RMSE (static regime)**	**# number of regressors**
Utopian point	0.0619 ± 0.0220	0.4007 ± 0.6799	5.1333 ± 1.7269
Median	0.0817 ± 0.0198	0.3439 ± 0.4401	6.7333 ± 2.7921
Mean	0.0741 ± 0.0200	0.3378 ± 0.2493	6.6667 ± 2.3570
Model 6.1	1.2873×10^{-15}	0.2069	3

Source: Author (2022)

As shown in Table 6.1, the selection approach by Euclidean distance using the utopian point is the one with the smallest error in the dynamic regime, and it is also the one with the closest mean number of regressors to Model 6.1. On the other hand, the Utopian Point approach has the highest static regime error among the proposals, with the Mean approach being better in this scenario. It is also possible to observe that both the Mean approach and the Median approach have an average value of the number of regressors farther from the real value (three regressors), in addition to having a higher standard deviation value, which also demonstrates that the larger number of regressors did not contribute to a better dynamic regime behavior in this scenario.

In order to also compare the quality of the Pareto-optimal found by each of the approaches, the Hypervolume (HV) indicator (ZITZLER; THIELE, 1998) was used, one of the most applied quality indicators for multi-objective problems (LI; YAO, 2019). It calculates the volume of all rectangular bands up to a given reference point. Therefore, hypervolume is an indicator of maximization, *i.e.*, the greater the hypervolume value found for an algorithm, the better the convergence and diversity of its result (for more information on the Hypervolume indicator, see Guerreiro, Fonseca and Paquete (2020)).

In this work, for this dataset, before applying the HV indicator, it was considered that all models found that have RMSE greater than five in static or dynamic regimes are considered unstable. After that, all RMSE values and the number of regressors from all ninety Pareto-optimal were normalized between zero and one. Thus, the reference point selected for the HV indicator was $(1,1,1)$ for the three approaches. Therefore, the HV indicator was applied to each of the thirty Pareto-optimal of each approach, and then the mean and standard deviation between the thirty results of each approach were taken – these results are shown in Table 6.2.

Table 6.2 – Result of the HV indicator for the three approaches (mean, median and utopian point). Mean and standard deviation for thirty runs of each approach.

Approach	**Hypervolume indicator**
Utopian point	0.9113 ± 0.0087
Median	0.8857 ± 0.0201
Mean	0.8876 ± 0.0134

Source: Author (2022)

It is possible to observe in Table 6.2 that, in absolute values, the selection approach through the Euclidean distance using the utopian point obtains better Pareto-optimal results. In order to confirm this statement, the result presented in Table 6.2 was also analyzed using the One-way ANOVA variance analysis, at a level of 5% (0.05) significance, and then tested with the *post hoc* Tukey test. The results of both tests are shown in Table 6.3.

Analyzing Table 6.3, it is possible to see that the p-value is 4.9021×10^{-10}, *i.e.*, less than 0.05, so it is possible to reject the null hypothesis (“there is no difference between the results in Table 6.2”). Observing the results of Tukey’s *post hoc* analysis, it is possible to see that there is no statistically significant difference between the results of the Mean and Median

Table 6.3 – One-way ANOVA analysis and *post hoc* Tukey test for the results in Table 6.2

One-way ANOVA	***f*-value**	***p*-value**
-	27.7038	4.9021×10^{-10}
***post hoc* Tukey**	Null hypothesis:	**Is the performance between the algorithms different?**
Mean and Median	-	No
Mean and Utopian point	-	Yes
Median and Utopian point	-	Yes

Source: Author (2022)

approaches (the null hypothesis “Is the performance between the algorithms different?” is rejected), however, there is a statistically significant difference between them and the utopian point approach. In this way, as already mentioned, the HV indicator is a maximization indicator, and, as the result value of the utopian point approach is greater than the result of the other two approaches, it is possible to say that, indeed, use this approach, on average, results on better models.

6.1.2.2 Different approaches to parameter estimation

In order to verify whether the use of auxiliary information in parameter estimation is relevant to obtain better models by the proposed methodology, the MGGP algorithm was run with the same parameters as in Subsection 6.1.2.1, together with selection by the Euclidean distance with the utopian point and also using the same three objectives. However, three different parameter estimation forms were tested: *i.* least squares, *ii.* extended least squares and *iii.* weighted least squares. The last tested approach, WLS, uses auxiliary information, as described in the proposed methodology (Section 5.2), which weights the data relevance in dynamic and static regimes by the variable λ, as also done in Freitas, Barbosa and Aguirre (2021).

As in Subsection 6.1.2.1, for each of the approaches, the algorithm was run thirty times, generating a total of ninety Pareto-optimal that were subjected to free-run simulation, in the static and dynamic regime, and obtained their respective RMSE calculated for each regime, the mean and standard deviations were taken for each approach. These results are presented in Table 6.4, together, for comparison purposes, with the RMSE values for Model 6.1, which was also submitted to free-run simulation.

Analyzing Table 6.4, it is possible to see that the three approaches obtained similar RMSE results in the dynamic regime, however, the approach that uses WLS in the parameter

Table 6.4 – Error results (RMSE) in free-run simulation for dynamic and static regime. Mean and standard deviation for thirty runs of each parameter estimation approach.

Approach	Dynamic regime	Static regime	# of regressors	λ value
WLS	0.0619 ± 0.0220	0.4007 ± 0.6799	5.1333 ± 1.7269	0.09 ± 0.03
ELS	0.0614 ± 0.0387	0.4408 ± 0.5174	8.9667 ± 3.4008	-
LS	0.0696 ± 0.0722	0.4357 ± 0.4484	8.0333 ± 3.0274	-
Model 6.1	1.2873×10^{-15}	0.2069	3	-

Source: Author (2022)

estimation obtained results approximately 10% better in the static regime than the other two approaches, however, this gain is not statistically significant when taking into account the standard deviations of each result. Another point that the approach that uses weighted least squares stands out is the average number of regressors in the models found, being 1.74 times smaller than the ELS approach and 1.56 times smaller than the LS, being closer to the real value of three regressors. Finally, another data presented is the average λ value of the Paretos' stable models encountered, showing that a good balance between the data from the two regimes is 90% for dynamic data and 10% for static data.

As in Subsection 6.1.2.1, the HV indicator was used to verify which parameter estimation approach obtains, on average, the best Pareto sets - the results are displayed in Table 6.5. As it is possible to observe, in absolute terms, the approach WLS, which uses the auxiliary information to aid parameter estimation, is the best. In order to confirm this observation, the ANOVA variance analysis was also performed, with a 5% (0.05) significance level, together with the *post hoc* Tukey test (the results are shown in Table 6.6). As the p-value (2.4137×10^{-5}) is less than 0.05, the null hypothesis ("there is no difference between the results presented in Table 6.5.") is rejected. Observing the results of Tukey's post *hoc test*, it is possible to see that there is a statistically significant difference between the WLS approach and the other two approaches (LS and ELS). Furthermore, as the mean value of the HV indicator of the WLS approach is the highest, it is possible to state that this approach is the best to perform parameter estimation in this case.

In order to compare the result obtained using the WLS for parameter estimation with other works in the literature, the best model under the static regime, the best model under the dynamic regime, and the model in which the approach was able to perfectly find the three regressors of Model 6.1, here called the Perfect Model (with their respective λ), were selected among the thirty Pareto sets. These models are compared with two Freitas, Barbosa and Aguirre

Table 6.5 – Result of the HV indicator for the three approaches (WLS, ELS and LS). Mean and standard deviation for thirty runs of each parameter estimation approach.

Approach	**Hypervolume indicator**
WLS	0.9113 ± 0.0087
ELS	0.8618 ± 0.0515
LS	0.8483 ± 0.0741

Source: Author (2022)

Table 6.6 – One-way ANOVA analysis and *post hoc* Tukey test for the results in Table 6.5

One-way ANOVA	***f*-value**	***p*-value**
-	12.0436	2.4137×10^{-5}
***post hoc* Tukey**	Null hypothesis:	**Is the performance between the algorithms different?**
ELS and LS	-	No
ELS and WLS	-	Yes
LS and WLS	-	Yes

Source: Author (2022)

(2021) results, the former uses the Constrained least squares (CLS) for parameter estimation, and the latter uses the weighted least squares (WLS). The structure used in Freitas, Barbosa and Aguirre (2021) work has five regressors and was obtained using the same approach as Mendes and Billings (2001). All RMSE values for the cited models are shown in Table 6.7.

Table 6.7 – Comparison between models. Error results (RMSE) in free-run simulation for dynamic and static regime, number of regressors and lambda λ value.

Model	**Dynamic regime**	**Static regime**	**# of regressors**	**λ value**
Perfect Model	0.0568	0.2020	3	0.1
Best in Static	0.0508	0.1973	5	0.1
Best in Dynamic	0.0265	0.2002	8	0.1
WLS by Freitas, Barbosa and Aguirre (2021)	0.0557	0.2027	5	0.1
CLS by Freitas, Barbosa and Aguirre (2021)	$\gg 10^2$	-	5	-
Model 6.1	1.2873×10^{-15}	0.2069	3	-

Source: Author (2022)

Analyzing the results in Table 6.7, it is possible to see that the Best in Dynamic model actually has the best result for the dynamic regime, but at the cost of having eight regressors, *i.e.*, five regressors more than the correct one. The Best in Static model, on the other hand, is only slightly better than the Perfect Model and the Best in Dynamic model for the static regime, and for that, it has five regressors, *i.e.*, two more than correct, besides having an RMSE in the dynamic regime two times worst when compared to the Best in Dynamic model and approximately equal to the Perfect Model. The WLS by Freitas, Barbosa and Aguirre (2021) model has an RMSE in dynamic and static regime equivalent to the Perfect Model and the Best in Static model. Interestingly, it shows that using static data to search for regressors leads to equivalent results in a static regime, even when compared with models that used auxiliary information to estimate parameters. The CLS by Freitas, Barbosa and Aguirre (2021) model, on the other hand, has a lower performance when compared to all others in the dynamic regime, in addition to having two more regressors than the correct one. Another interesting point that can be observed is the value of lambda λ as 0.1 for all models that used the approach proposed in this work as well as for the models found by Freitas, Barbosa and Aguirre (2021), which reaffirms that the proportion of 90% for dynamic data and 10% for static data is truly favorable for better results with this dataset.

Finally, the parameters found for the Perfect Model using WLS are presented in Table 6.8 together with the original parameters of Model 6.1 used to generate the dataset. It is possible to notice that the values found are very close to the correct ones, which demonstrates again that it is advantageous to use the WLS and auxiliary information to estimate parameters. Part of the free-run simulation of Perfect Model and Model 6.1 over validation dataset Z_v is shown in Figure 6.1.

Table 6.8 – Parameter values from the Perfect Model, estimated using WLS, and from Model 6.1.

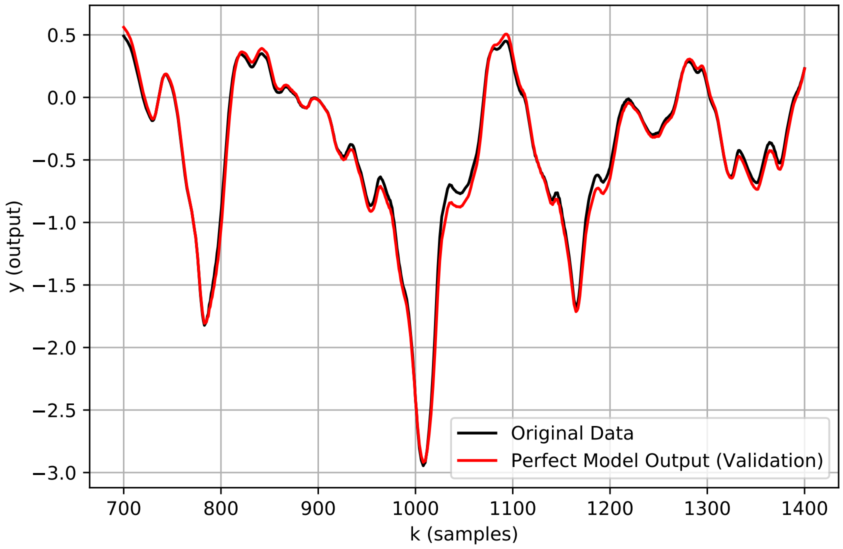
Model	θ_1	θ_2	θ_3
Perfect Model	0.78	0.24	-0.17
Model 6.1	0.75	0.25	-0.20

Source: Author (2022)

6.1.2.3 Number of objectives in the individual's evaluation

The approach proposed in Chapter 5 has three objectives, namely: *i.* minimize the error in the dynamic regime, *ii.* minimize the error in the static regime and *iii.* minimize the number of regressors in the MGGP model. In order to compare the performance of the methodology, three tests were performed. In the first test, the methodology was implemented with a single objective, which is *i.* minimize the error in the dynamic regime. The second test implemented the multi-objective methodology with two objectives, namely *i.* minimize the

Figure 6.1 – Free-run simulation over validation dataset Z_v, where system (presented in 6.1) output y is the black line and Perfect Model the red line.



Source: Author (2022)

error in the dynamic regime and *ii.* minimize the error in the static regime. Finally, in the third test, the three objectives were applied. The average results of RMSE, in dynamic and static regime, and the number of regressors found for each of the three tests are presented in Table 6.9. It is important to note that, as done in Subsection 6.1.2.1, for each of the tests the algorithm was executed thirty times, later all models of Pareto sets were submitted to free-run simulation and their RMSE values were calculated. The Model 6.1 is also presented in the table for reference purposes.

Table 6.9 – Error results (RMSE) in free-run simulation for dynamic and static regime. Mean and standard deviation for thirty runs of each approach (one, two and three objectives).

# of objectives	**Dynamic regime**	**Static regime**	**# of regressors**	**λ value**
1	0.2863 ± 0.1839	1.1538 ± 0.7799	13.4286 ± 2.5555	0.4428 ± 0.3774
2	0.1484 ± 0.1332	0.8538 ± 0.9702	8.4333 ± 3.1057	0.1967 ± 0.2892
3	0.0619 ± 0.0221	0.4007 ± 0.6799	5.1333 ± 1.7269	0.0900 ± 0.03
Model 6.1	1.2873×10^{-15}	0.2069	3	-

Source: Author (2022)

When analyzing Table 6.9, it is possible to see that the average performance of the models found improves strongly with the increase in the number of objectives used during training. For example, the RMSE value in the dynamic regime with three objectives is approximately 78.5% lower than the result using only one objective and approximately 58% lower than the result with two objectives. This same behavior is also observed for the RMSE value in the static regime, in which the three-objective approach had a result approximately 65% lower than that found with only one objective and approximately 53% lower than the result found with two objectives.

Another interesting point in Table 6.9 is the lambda λ value of each of the approaches. Comparing the single-objective approach with the two-objective approach, which includes minimizing the error in the static regime, it is remarkable that this, more than improving the behavior in the static regime, reduces, on average, approximately 55.6% the value of lambda, which demonstrates that the approach with only one objective gives greater importance to the data in the static regime, trying to compensate the absence of the objective related to the static regime in the search for regressors. With the addition of the third objective, the importance given to steady-state data during parameter estimation is reduced again by approximately 54%, without any harm to the other objectives, on the contrary, performance improves in general. Finally, it is possible to observe that the number of regressors is also positively impacted with the use of auxiliary information (second objective) in the search for terms of the MGGP model, reducing the average number of regressors of the models found by 37%. The use of the third objective, as expected, also makes the models found by the algorithm more parsimonious, with the number of regressors on average being 39% lower when compared to the approach that uses two objectives and, even with a smaller number of regressors, it has, on average, better behavior in both regimes. Therefore, it is possible to affirm that the addition of two objectives during the training gives the algorithm a more effective search for regressors, in addition to finding more parsimonious MGGP models with better performance in both regimes.

Finally, as done in Subsection 6.1.2.1, in order to assess the quality of Pareto sets found with two and three objectives, the HV indicator was applied. It is important to emphasize that the approach with only one objective does not naturally generate a Pareto, and for this reason this approach is not in the comparison. The results of the HV indicator are shown in Table 6.10.

Table 6.10 – Result of the two-dimensional HV indicator for two and three objectives.

# of objectives	**Hypervolume indicator**
2	0.8783 ± 0.1173
3	0.9482 ± 0.0048

Source: Author (2022)

Looking at Table 6.10, it is possible to see the superiority of the approach that uses three objectives in relation to two objectives, being 1.08 times greater than its value of the HV indicator. Therefore, it is possible to affirm that using three objectives implies finding, on average, Pareto sets with more parsimonious models, in addition to better behavior in dynamic and static regimes, when compared to the other two approaches. It is important to note that for the HV indicator of the three-objective approach, the third objective, minimizing the number of regressors, was not used in the calculation to make a fair comparison with Pareto found by the two-objective approach (two dimensions vs. two dimensions).

6.1.2.4 Approaches to model simulation in the static regime

In order to verify the efficiency of finding good models when using the approach proposed by Freitas, Barbosa and Aguirre (2021) to simulate the static regime, it is necessary to compare it with the more traditional way used, the free-run simulation. Thus, the methodology proposed in this work was implemented, using the same parameters for the MGGP algorithm of Subsection 6.1.2.1, with the same three objectives, using WLS for parameter estimation and selection by Euclidean distance with the utopian point. However, for the second objective of the cost function, RMSE in the static regime, the two forms of the simulation were tested, *i.e.*, free-run simulation and the simulation proposed by Freitas, Barbosa and Aguirre (2021).

As in Subsection 6.1.2.1, for the approach that used the form of simulation proposed by Freitas, Barbosa and Aguirre (2021), the algorithm was executed thirty times, generating a total of thirty Pareto-optimal sets, which were validated through free-run simulation in the dynamic and static regime. With these results, the RMSE was calculated for each regime, and the mean and standard deviation were taken for this approach. For the approach that uses free-run simulation, the same process with the same parameters was also performed, however, the MGGP algorithm was executed only once. This was done because the computational cost for this approach is far higher than the computational cost of the first approach. The execution times[1] of each approach are shown in Table 6.11.

Analyzing Table 6.11, it is possible to see that the computational cost, *i.e.*, the execution time for the Free-run approach, is approximately nine times the execution time of the approach presented by Freitas, Barbosa and Aguirre (2021), which is very relevant since, for a dataset

[1] All algorithms in this work ran on a computer with the following specifications: Intel® Xeon® CPU @ 2.20GHz, 13GB RAM and 185GB Disk Memory

Table 6.11 – Execution time, in hours, for the Freitas, Barbosa and Aguirre (2021) approach and for the free-run simulation approach.

Approach	Execution time (hours:minutes:seconds)
Simulation proposed by Freitas, Barbosa and Aguirre (2021)	01:07:39 ± 00:20:07
Free-run simulation	09:23:06

Source: Author (2022)

with a larger number of samples, the use of the second approach may become unfeasible. In order to verify if the models found with the simulation approach proposed here (for static data) can obtain results similar to those obtained by the models of the traditional approach (free-run simulation), the best model found for the dynamic regime, and the best model found for the static regime of each of the approaches was selected. It is extremely important to point out that this comparison does not aim to verify which approach is better, since the free-run simulation is at a disadvantage as it was executed only once, while the other one was executed thirty times. Therefore, the comparison between the approaches is just a reference to verify that the approach with the lowest computational cost is able to find models with similar behavior (in both regimes) to the other approach, allowing its use as a form of simulation for the second objective. These results are shown in Table 6.12.

Table 6.12 – Error results (RMSE) in free-run simulation for dynamic and static regime for the Best in Static and Best in Dynamic using the Freitas, Barbosa and Aguirre (2021) approach and for the Best in Static and Best in Dynamic using the Free-run simulation approach.

Approach	Dynamic regime	Static regime	# of regressors	λ value
Best in Static (Freitas, Barbosa and Aguirre (2021))	0.0508	0.1973	5	0.1
Best in Dynamic (Freitas, Barbosa and Aguirre (2021))	0.0265	0.2002	8	0.1
Best in Static (*free-run*)	0.0915	0.1953	7	0.8
Best in Dynamic (*free-run*)	0.0288	0.1971	6	0.6
Perfect Model	0.0568	0.2020	3	0.1
Model 6.1	1.2873×10^{-15}	0.2069	3	-

Source: Author (2022)

Observing Table 6.12, it is possible to see that the Best in Static model, which uses the Freitas, Barbosa and Aguirre (2021) approach, has RMSE in static regime equivalent to the Best in Static model found by the Free-run approach and, at the same time, having an RMSE for the dynamic regime almost 50% smaller even having two fewer regressors. On the other hand, the Best in Dynamics model, obtained through the Freitas, Barbosa and Aguirre (2021) approach, has RMSE for the static and dynamic regime equivalent to the Best in Dynamics model found through the Free-run approach, but the first approach has two more regressors. These observations corroborate that the approach presented by Freitas, Barbosa and Aguirre (2021) achieves similar results to the traditional approach, in addition to having a much lower computational cost. Another relevant information is that both approaches were able to find a model with only the three correct regressors (they found the Perfect Model) and have results for the static regime equivalent to Model 6.1, which was used to generate all data.

Finally, as done in Subsection 6.1.2.1, in order to compare the quality of the Pareto sets found using the proposed methodology compared to the approach that uses the free-run simulation, the HV indicator was applied. However, as was done for the RMSE analysis, a comparison was made using the best and the worst Pareto set found, using the proposed methodology, among the thirty available and the Pareto set obtained through the approach that uses the free-run simulation for the second objective of the cost function. These results are available in Table 6.13. For information purposes, the mean value, together with its standard deviation, of the HV indicator for the thirty Pareto sets found is also in the same table.

It is important to emphasize again that the results in Table 6.13 also do not demonstrate the superiority of any of the approaches since, as already mentioned, the approach that uses free-run simulation is at a disadvantage. In this way, the results serve only as a reference that the proposed approach in this work can fulfill its role of including information about the static curve in the objectives during the search for regressors and finding competitive models and Paretos.

Through the results presented in Table 6.13, it is possible to observe that the approach proposed in this work is competitive with the free-run simulation approach since the second approach has a higher value of the HV indicator by approximately 1% when compared to the results for the best Pareto set, the same is valid for the average Pareto value found. However, when the comparison is made with the worst Pareto set found, among the thirty available, the

Table 6.13 – Result of the HV indicator for the Freitas, Barbosa and Aguirre (2021) approach (Average of the thirty Pareto sets, Best Pareto set and Worst Pareto set) and for the *free-run* approach.

Approach	**Hypervolume indicator**
Average of the thirty Pareto sets	0.9113 ± 0.0087
Best Pareto set	0.9198
Worst Pareto set	0.8872
free-run Approach	0.9289

Source: Author (2022)

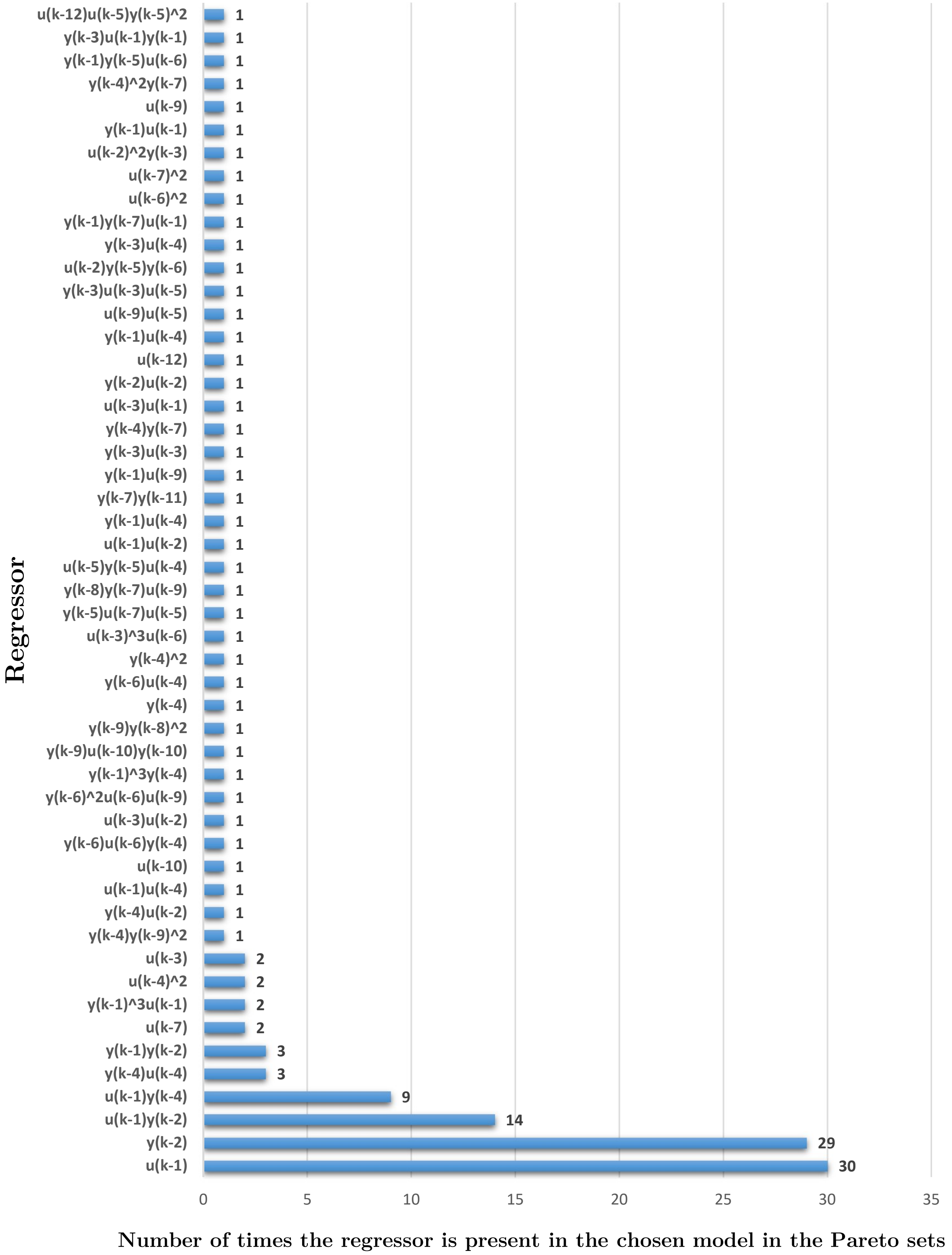
HV indicator of the free-run approach is approximately only 4.7% higher, which still maintains the proposed approach competitive.

6.1.2.5 Statistical analysis of chosen regressors

In order to verify the efficiency of the methodology proposed in this work in finding models with the correct regressors, a survey was carried out as follows: *i.* the thirty Paretos found for this methodology were simulated, in the dynamic and static regime, and had their respective RMSE calculated, *ii.* the decision criterion presented in Subsection 5.2.3 was applied to each of the thirty Pareto sets, selecting a total of thirty models, *iii.* for each of the selected models, it was verified which regressors it has, *iv.* the data from all models were consolidated, and a bar graph was constructed to visualize the results. Figure 6.2 presents the bar graph with the result of the described process.

Analyzing the graph presented in Figure 6.2, it is possible to observe that the methodology proposed in this work is efficient in finding the correct regressors for the models since the three most present regressors in the models, *i.e.*, $u[k-1]$, $y[k-2]$ and $u[k-1]y[k-2]$, are the correct regressors of the Model 6.1. More specifically, regressor $u[k-1]$ appears in 100% of the selected models, the regressor $y[k-2]$ in 97% (29 out of 30) of the selected models, and the regressor $u[k-1]y[k-2]$ appears in 47% (14 out of 30) of the models selected by decision criterion. This demonstrates that most models found by the algorithm have the correct regressors, even in conjunction with other spurious regressors. Another interesting point that can be concluded is that if the correct regressors of some system being modeled are not known, the proposed algorithm, if feasible, can be executed multiple times and, through its results, it will be possible to verify which are the most present regressors in the models, the tendency is that the most likely to appear are the correct regressors for the system.

Figure 6.2 – Number of times the regressor is present in the chosen model in the Pareto sets.

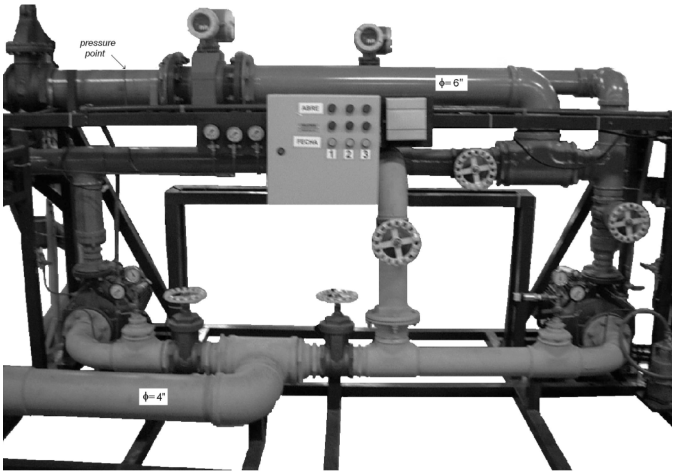


Source: Author (2022)

6.2 The Hydraulic Pumping System

The hydraulic pump system used to generate the data in this section comprises two centrifugal pumps that feed a hydraulic turbine. These two centrifugal pumps are coupled to 7.5 *kW* induction motors and variable speed drive systems; the physical plant is shown in Figure 6.3. It is important to emphasize that the piezoelectric pressure transmitter has an uncertainty of ±0.175 *mlc* (meter of liquid column) (BARBOSA et al., 2011). In the following subsections, the dataset used is explained, the parameters used in the MGGP algorithm are given, and, finally, the final results are presented and discussed.

Figure 6.3 – Water pumping system.



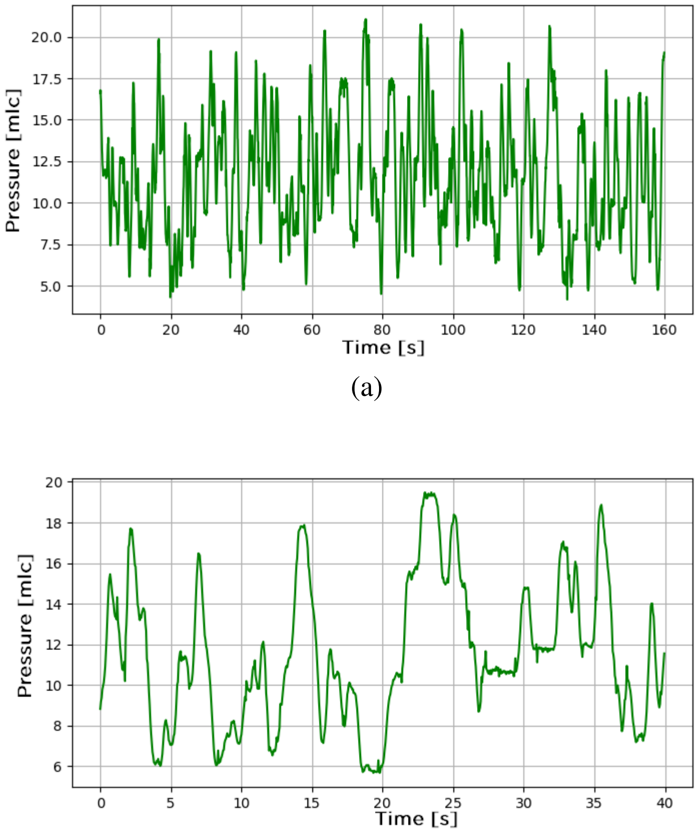
Source: Barbosa et al. (2011)

6.2.1 Dataset

This dataset was taken from the Artificial Intelligence and Automation Research Group (AIA) website and was first used in Barbosa et al. (2011). It has, as input, the pump speed reference, measured in revolutions per minute (*rpm*), and as output, the system pressure, expressed in meter of liquid column (*mlc*). The dataset has two types of information about the system, *i.e.*, data referring to dynamic and static regime behavior.

The dynamic part of the data was obtained through an excitation signal, with variable amplitude, applied to the system's input – allowing different pump operating points to be reached. $N = 4000$ samples were generated for the dynamic dataset, where $N = 3200$ are used in the system modeling and $N = 800$ in its validation. Figure 6.4a presents the data used during the modeling of the system and Figure 6.4b shows the validation data.

Figure 6.4 – Dynamic data for (a) modeling and (b) validation.



(a)

(b)

The static part of the samples is formed by ten trials that represent different points of operation of the system. In these trials, the speed applied at the system's input varies from 750 *rpm* to 1650 *rpm*, with a step of 100 *rpm* between the trials. It is important to emphasize that during the execution of each test, the pump input value is kept constant so that, after some time, the system goes into stability (its output no longer has any transient components). Therefore, the system output pressure was registered for each operating point. Finally, the mapping between input and output values of the pumping system, in the static regime, are presented in Table 6.14.

6.2.2 Algorithm parameters

The entire methodology described in Chapter 5 was applied to model the hydraulic pump system. It is essential to highlight that the three objectives were used, and unlike the generation number used for the Piroddi system ($n_{gen} = 250$), for this system $n_{gen} = 500$ was used, the other parameters remain the same as those already presented, as follows:

```
popSize = 100;
CXPB = 0.9;
MTPB = 0.1;
n_gen = 500;
maxHeight = 5;
maxTerms = 20;
elite = 10;
maxDelay = 3 (q1,q2,q3);
numberOfVariables = 2;
primitive function: multiplication function.
```

6.2.3 Results and discussion

In this experiment, the MGGP model found by the algorithm proposed in this work to model the hydraulic pump system was as follows:

$$
\begin{aligned}
y_{MGGP}[k] = {} & \theta_1 u[k-1]u[k-6]y[k-1] + \theta_2 y[k-1]y[k-2] \\
& + \theta_3 u[k-1] + \theta_4 u[k-1]^2 \\
& + \theta_5 u[k-6]^2 y[k-1] + \theta_6 y[k-4] \\
& + \theta_7 u[k-1]u[k-2]u[k-3]u[k-5]y[k-2] + \theta_8 u[k-1]y[k-1]^2 \\
& + \theta_8 u[k-1]y[k-1]^2 + \theta_9 y[k-1] + \theta_{10},
\end{aligned}
\tag{6.2}
$$

The MGGP model 6.2 found has $l = 5$, $n_y = 4$, $n_u = 6$ and 10 regressors – the entire algorithm training process took 08:19:40 (hours:minutes:seconds). In order to verify the qua-

Table 6.14 – Static test data.

Speed (rpm)	**Pressure (mlc)**
750	3.92
850	5.18
950	6.58
1050	8.26
1150	9.94
1250	11.90
1350	14.00
1450	16.10
1550	18.48
1650	20.86

Source: Author (2022)

lity of the MGGP model found, it is compared with other approaches already presented in the literature. In Barbosa et al. (2011), two models were found, which the structure of the models was obtained through the LS/ERR and its parameters estimated through the ELS method. The former, *Barbosa (15)* model, has $l = 2$, $n_y = 6$, $n_u = 6$ and 17 regressors; the latter model, *Barbosa (17)*, has $l = 3$, $n_y = 6$, $n_u = 6$ and 23 regressors. Another work used was Castro and Barbosa (2020), where the *Castro (11)* model was found through an MGGP/ERR hybridization. This model has $l = 5$, $n_y = 9$, $n_u = 12$ and 25 regressors. Finally, another model used was *Mota (7)*, found by Mota et al. (2020), whose structure was found by a multi-objective approach through the NSGA-II algorithm, with the same three objectives used in this work, but the representation of the individual in this approach was binary. This model has $l = 2$, $n_y = 6$, $n_u = 6$ and 6 regressors. The RMSE results for all models mentioned are displayed in Table 6.15. The values identified as J_S (Ident.), J_S (Val.), J_{SF}, and N_P are respectively the root mean square error (RMSE) in a free-run simulation of the dynamic regime, using training data, the RMSE in a free-run simulation in a dynamic regime, using validation data, the RMSE in a free-run simulation of the static regime and the number of model terms.

Table 6.15 – Results - J_S represents free-run simulation RMSE in dynamic regime (training and validation), J_{SF} represents RMSE in static regime and N_P represents the number of terms in the model.

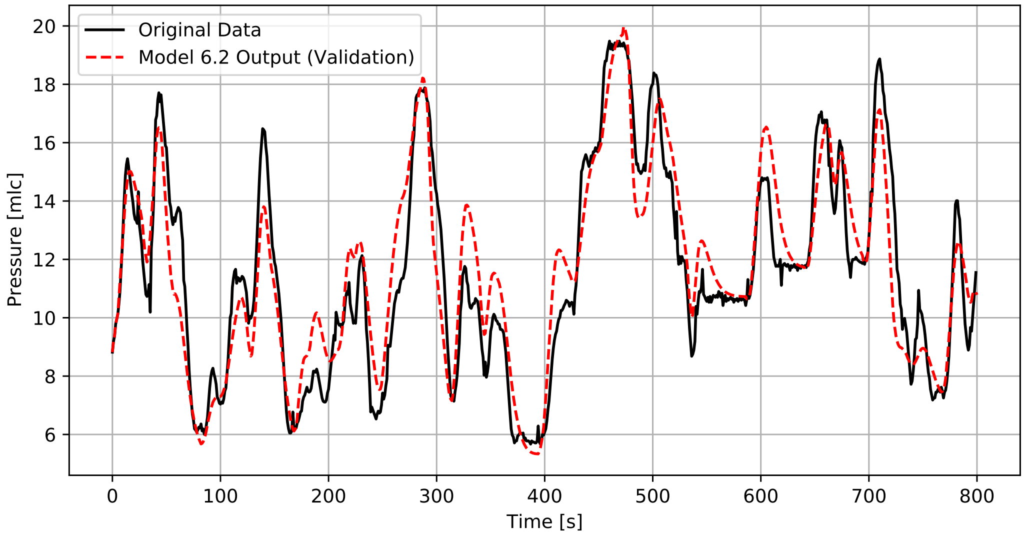
Model	J_S (Ident.) $[mlc^2]$	J_S (Val.) $[mlc^2]$	J_{SF} $[mlc^2]$	N_P	(l, n_y, n_u)	λ
Barbosa (15)	$1,6158$	$1,4546$	$1,2664$	17	(2,6,6)	-
Barbosa (17)	$1,2288$	$1,0507$	$0,2455$	23	(3,6,6)	-
Castro (11)	$1,1049$	$0,9984$	$0,3926$	25	(5,9,12)	-
Mota (7)	$1,6334$	$1,4713$	$0,0812$	6	(2,6,6)	-
Model 6.2	$1,6041$	1.4094	0.4061	10	(5,4,6)	0.2

Comparing Model 6.2 with the *Barbosa (15)* model, it is clear that they have similar behavior for dynamic data, regardless of whether the data are validation or training. However, Model 6.2 behaves in the static regime 3.11 times better than the *Barbosa (15)* model, even having seven regressors less. Compared with the *Barbosa (17)* model, it is possible to see that Model 6.2 has an inferior behavior in both regimes, but to obtain these results, *Barbosa (17)* has more than twice as many regressors as the model found by the methodology proposed here. Comparing *Castro (11)* with Model 6.2, it is possible to observe that the former, has a better behavior in a dynamic regime (both for training data and for validation data) and an equivalent behavior in a static regime; however, to obtain this result, the Castro and Barbosa (2020)'s approach has 2.5 times more regressors, *i.e.*, fifteen more regressors.

Finally, when comparing Model 6.2 with the *Mota (7)* model, it is noticeable that both behave equivalently in dynamic regime. However, observing the static regime, it is visible that the *Mota (7)* model has a better result than all other approaches, being, specifically, 4.9 times better than the Model 6.2. However, this model was found by NSGA-II algorithm using the binary representation, which can be a problem if the possibility of regressors is very large, making its execution unfeasible in some situations. Based on all that has been exposed, it is possible to state that the MGGP model found stands out in having a competitive result in a dynamic and static regime, maintaining a lower number of regressors than most other approaches, *i.e.*, being more parsimonious.

Figure 6.5 shows the output of Model 6.2, in free-run simulation, using the dynamic validation dataset, Z_v, together with the original output for visual comparison of the results.

Figure 6.5 – Comparison between y output (in black), real validation dynamic data, and Model 6.2 output in free-run simulation (in dashed red).



Source: Author (2022)

6.3 The deepwater oil well process

After validating the methodology proposed in Chapter 5 with data from a stochastic model, the Piroddi model, and data from a real problem, the pumping system, this entire process is applied to perform the modeling of a virtual sensor that aims to provide information on downhole pressure in an offshore oil extraction process, the final objective of this work. In the following subsections, the dataset used is explained, the parameters used in the MGGP algorithm are given, and, finally, the final results are presented and discussed.

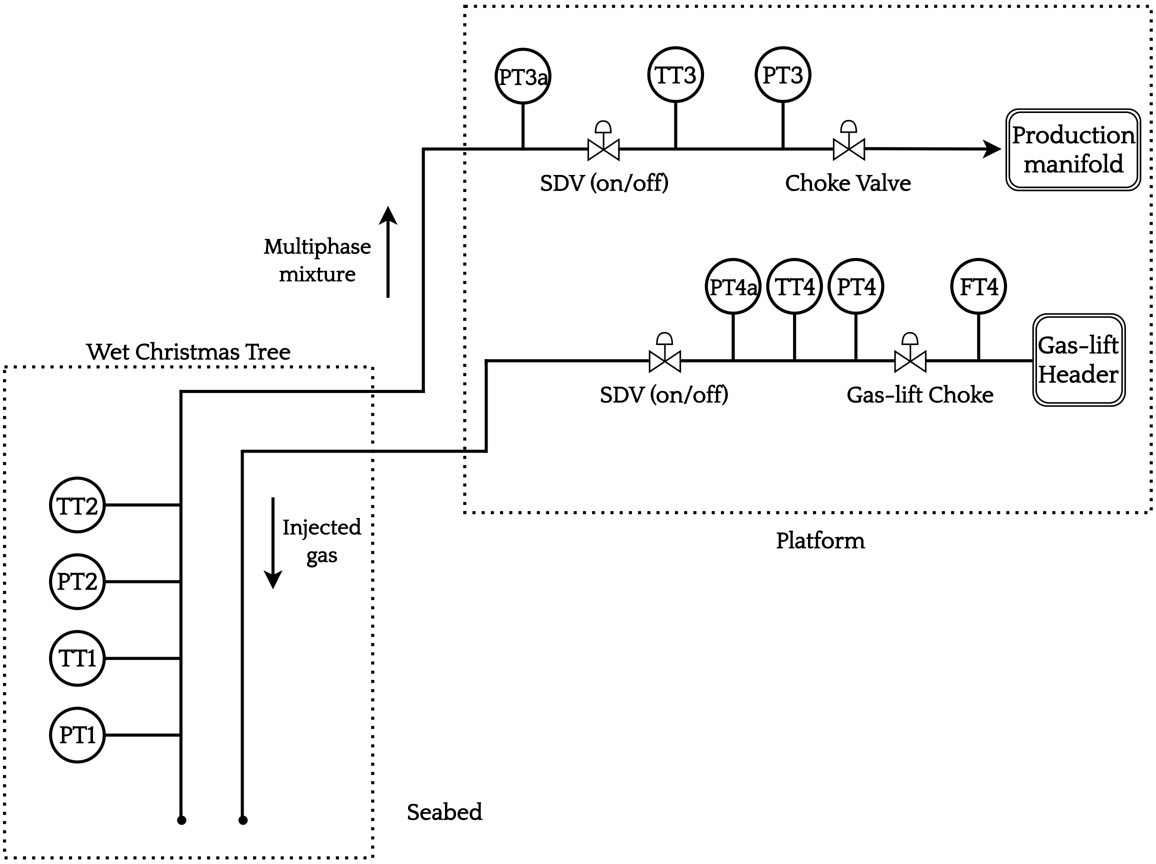
6.3.1 Dataset

This dataset is presented in Abreu (2013), Aguirre et al. (2017), and Freitas, Barbosa and Aguirre (2021), where it was used to model virtual sensors to overcome the lack of information caused by the failure of the downhole pressure gauge in an oil well that uses the gas-lift system in deep water – a problem that causes several economic impacts for the oil extraction industry (as already mentioned in Chapter 2). The dataset consists of five inputs and one output. The input variables are the pressure before shut down valve (PT3a), the pressure before production choke valve (PT3), the temperature before production choke valve (TT3), the pressure before gas-lift shutdown valve (PT4a), the instantaneous gas-lift flow rate (FT4), and autoregressive terms (all with lags 1,2 and 3); the output is the downhole pressure (PT1). Figure 6.6 presents a simplified P&ID diagram of a gas-lift oil well, and Table 6.16 lists the tags of some commonly measured variables.

The process depicted in Figure 6.6 is summarized as follows. The gas-lift header at the platform (instruments tagged by 4) injects high-pressure gas through the annulus between tubing and casing string until it reaches an orifice valve located downstream inside the lower part of the tubing. This process generates bubbles that, when mixed with the multiphase mixture (pre-oil, gas, and water), make it less dense, which, together with the sufficiently high pressure in the reservoir, allows its transport from the bottom of the well to the platform. The Christmas tree (PT2 and TT2), on the seabed, controls the production flow from the seabed to the platform. At the platform, the shutdown valve (SDV) is available to interrupt production in case of any emergency, and the choke production valve regulates the production flow rate at the platform (TEIXEIRA et al., 2014). The production instruments are tagged by 3.

As in the hydraulic pump dataset, the dynamic part of this dataset was divided into two parts, one for training ($N_d = 5000$ samples) and the other for validation ($N_v = 95000$ samples). The static data ($N_s = 32$ samples) were manually selected by analyzing the steady-state regimes of downhole pressure and obtaining mean values of all input variables. Both data were obtained using a Plant Information Management System (PIMS), with a sampling time of $T_s = 1$ m (AGUIRRE et al., 2017). Finally, Figure 6.7a shows instantaneous gas-lift flow rate (FT4),

Figure 6.6 – Simplified P&ID diagram of a gas-lifted oil well, where TT refers to temperature transmitters and PT refers to pressure transmitters. The corresponding variables are described in Table 6.16.



Source: Adapted from Teixeira et al. (2014).

Table 6.16 – Process variables used to obtain models for the gas-lift oil well. Tags correspond to the instruments shown in Figure 6.6.

Tag	Description	Units
PT1	Downhole pressure	kgf/s^2
TT1	Downhole temperature	°C
PT2	Wet Christmas tree pressure	kgf/s^2
TT2	Wet Christmas tree temperature	°C
PT3a	Pressure upstream shutdown valve	kgf/cm^2
PT3	Pressure upstream production choke valve	kgf/cm^2
TT3	Temperature upstream production choke valve	°C
PT4a	Pressure upstream gas-lift shutdown valve	kgf/cm^2
TT4	Temperature upstream gas-lift shutdown valve	°C
FT4	Instantaneous gas-lift flow rate	m^3/h
PT4	Pressure downstream gas-lift choke valve	kgf/cm^2

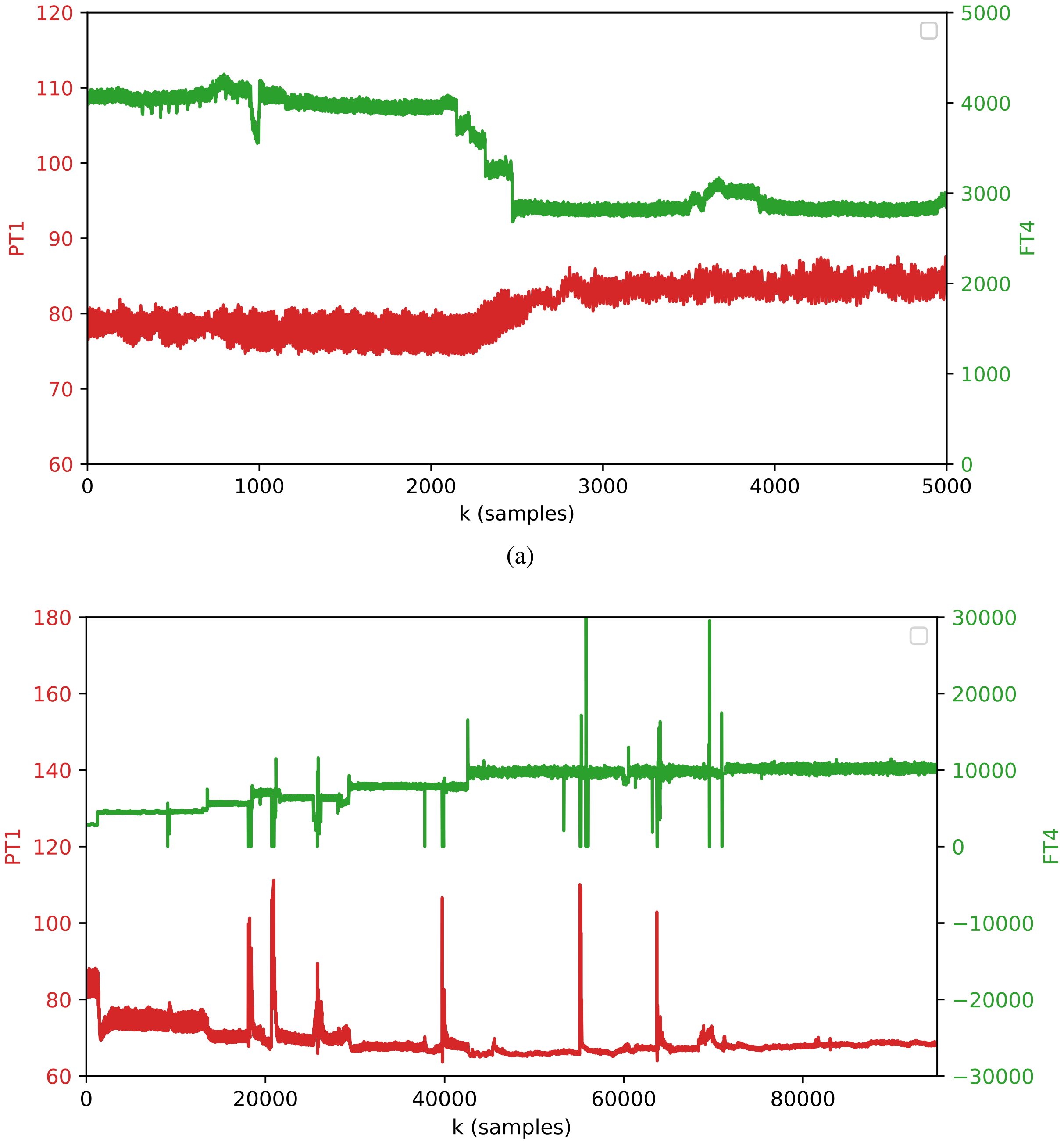
Source: Adapted from Teixeira et al. (2014) and Aguirre et al. (2017).

u_1, and downhole pressure (PT1), y, over the training dataset and Figure 6.7b shows the same variables for validation dataset.

6.3.2 Algorithm parameters

The entire methodology described in Chapter 5 was applied to model a soft-sensor for the downhole pressure (PT1). It is essential to highlight that the three objectives were used, and unlike the generation number used for the Piroddi system ($n_{gen} = 250$), for this system $n_{gen} = 500$ was used, the other parameters remain the same as those already presented, as follows:

Figure 6.7 – Instantaneous gas-lift flow rate FT4 (u_1) and the downhole pressure PT1 (y) from (a) training Z_d; and (b) validation Z_v datasets.



(a)

(b)

```
popSize = 100;
CXPB = 0.9; MTPB = 0.1;
n_gen = 500;
maxHeight = 5; maxTerms = 20;
elite = 10;
maxDelay = 3 (q1,q2,q3);
numberOfVariables = 6;
primitive function: multiplication function.
```

6.3.3 Results and discussion

In this experiment, the application of the methodology proposed in this work on the dataset of the oil extraction process, found the following NARX model:

$$\begin{aligned} y_{MGGP}[k] = {} & \theta_1 y[k-4]u_2[k-4]^2u_4[k-4]^2 + \theta_2 u_5[k-1]u_5[k-3] \\ & + \theta_3 u_3[k-1] + \theta_4 u_5[k-1] \\ & + \theta_5 u_4[k-1] + \theta_6 y[k-3] \\ & + \theta_7 y[k-1]u_5[k-1] + \theta_8 y[k-1]y[k-9] \\ & + \theta_9 y[k-1] + \theta_{10} u_5[k-4] + \theta_{11} y[k-2] + \theta_{12}, \end{aligned} \tag{6.3}$$

The MGGP model 6.3 found has $l = 5$, $n_y = 9$, $n_u = 4$ and 12 regressors, the complete training process of the algorithm took about 23:37:40 (hours:minutes:seconds). In order to compare the result found, which uses auxiliary information in the estimation of parameters, this same regressors structure (6.3) had its parameters estimated by least squares, an approach that only uses data from the dynamic regime (training) and does not use data from the static regime. This model was then subjected to free-run simulation in both regimes and had their respective RMSE calculated. The RMSE results for these two models are shown in Table 6.17.

Also, seeking to compare the results with other works, two models obtained by Freitas, Barbosa and Aguirre (2021) were used. Both models use the following multilayer perceptron (MLP) structure:

$$\begin{aligned} y_{MLP}(k) = \theta_0 + \sum_{i=10}^{10} \theta_i tanh(&\theta_{i,0} + \theta_{i,1} y(k-1) + \theta_{i,2} y(k-2) + \theta_{i,3} y(k-3) \\ &+ \theta_{i,4} u_1(k-1) + \theta_{i,5} u_1(k-42) + \theta_{i,6} u_1(k-136) \\ &+ \theta_{i,7} u_2(k-1) + \theta_{i,8} u_2(k-42) + \theta_{i,9} u_2(k-136) \\ &+ \theta_{i,10} u_3(k-1) + \theta_{i,11} u_3(k-5) + \theta_{i,12} u_3(k-22) \\ &+ \theta_{i,13} u_4(k-1) + \theta_{i,14} u_4(k-5) + \theta_{i,15} u_4(k-22) \\ &+ \theta_{i,16} u_5(k-1) + \theta_{i,17} u_5(k-5) + \theta_{i,18} u_5(k-22)), \end{aligned} \tag{6.4}$$

that has 10 hidden nodes with activation function $tanh(\cdot)$, and linear function in the output node. The former ($Freitas$ (1)) model presented by Freitas, Barbosa and Aguirre (2021) had its parameters estimated using the backpropagation and Levenberg-Marquardt algorithm (black-box approach). The latter $Freitas$ (2) model had its parameters estimated through the weighted backpropagation method and the Levenberg-Marquardt algorithm, thus being a gray-box approach, *i.e.*, it uses information from the static regime in the parameter estimation. The RMSE results for all models mentioned are also shown in Table 6.17. The values identified as, J_S (Val.), J_{SF}, and N_{Par} are respectively the root mean squared error (RMSE) in a free-run simulation of the dynamic regime, using validation data, the RMSE in a free-run simulation of the static regime and the number of model parameters.

Table 6.17 – Results - J_S represents free-run simulation RMSE in dynamic regime (validation, Z_v), J_{SF} represents RMSE in static regime and N_{Par} represents the number of parameters in the model.

Model	J_S (Val.)	J_{SF}	N_{Par}	lambda λ
Model 6.3 WLS	5.3494	3.1042	12	0.7
Model 6.3 LS	5.8521	3.7356	12	-
Freitas (1)	6.7420	-	201	-
Freitas (2)	3.7285	-	201	0.54

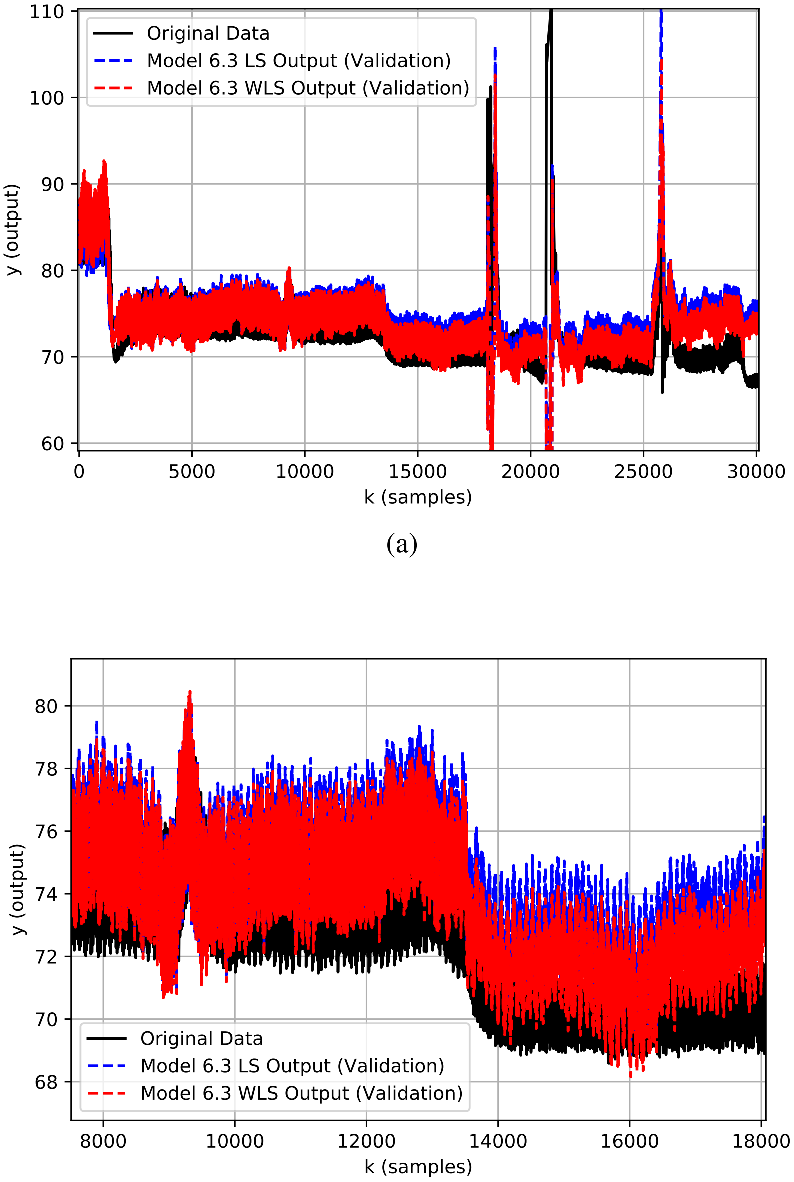
Analyzing Table 6.17, it is possible to see that $Model$ 6.3 WLS has an RMSE 20% lower than the $Freitas$ (1) model in the dynamic regime, even though it has only 12 parameters and is a polynomial model while $Freitas$ (1) is an MLP structure that is naturally more complex. Comparing the $Model$ 6.3 WLS with the $Freitas$ (2) model, which uses auxiliary information in the estimation of parameters, it is possible to see that its performance is approximately 30% worse, however, $Freitas$ (2) achieves this result by having 201 parameters while $Model$ 6.3 WLS has only 12. Another point that can be observed is that the performance in the dynamic regime of the *Model* 6.3 *LS* is superior to the *MLP Freitas* (1) model, which demonstrates that the methodology, using its three objectives (including the minimization of the error in static regime), manages to find suitable structures that, even when their parameters are estimated without auxiliary information, present a competitive behavior with models found by other computational techniques whose parameters were estimated in the same way.

Another point that can be analyzed is that the RMSE value in the dynamic regime of *Model* 6.3 *WLS* is only 9.4% lower than the value found for *Model* 6.3 *LS*. In comparison, for Model *Freitas* (1) with *Freitas* (2), the reduction is 44%, demonstrating that auxiliary information in the estimation of parameters of the models with the MLP approach is more effective than for the models with the MGGP approach. This fact may have occurred because the implemented MGGP algorithm did not find regressors with a higher delay value (as present in the MLP structure used by Freitas, Barbosa and Aguirre (2021), *e.g.*, 22, 42, and 136), a fact that may imply negligence of some information by the model, leading to worse performance. The difference of only 9.4% between the approach that uses auxiliary information in the estimation of parameters and the one that does not can be explained by the use of static data in the objectives of the algorithm during its execution, a fact that allows finding more regressors adapted for the static regime in general that, even having its parameters estimated only with dynamic data, it does not have such a big worsening in its performance for the static regime.

Figure 6.8a graphically presents the output result y (downhole pressure - PT1) of *Model* 6.3 *WLS* and *Model* 6.3 *LS* submitted to free-run simulation on the validation dataset in dynamic regime together with the actual data for comparison purposes. Figure 6.8b shows a zoom of Figure 6.8a for better visualization and comparison of results. It is possible to notice that the use of fixed operating points, as auxiliary information, contributed to a more suitable result for some regions where little dynamic data or exclusively static data (*e.g.*, $y \approx [70, 75]$) were provided during the algorithm's execution in search of regressors.

Finally, in order to analyze the result of *Model* 6.3 *WLS* in the static regime, it was subjected to free-run simulation. The model's output, the downhole pressure (PT1), was plotted with the original static regime data used during the algorithm's execution (training). This result is shown in Figure 6.9. As is notable, *Model* 6.3 *WLS* presents perfect results for the stationary (fixed) points in the static regime within the same area in which dynamic data were provided during training (in the graph represented by dark blue circles). However, the performance of *Model* 6.3 *WLS* outside the training dynamic data region is not good, which reveals

Figure 6.8 – a) Free-run simulation over validation dataset Z_v for Models *6.3* WLS, *6.3* LS and the real data for comparison purposes and b) presents a zoom on the simulation for better visualization and comparison of the results.



(a)

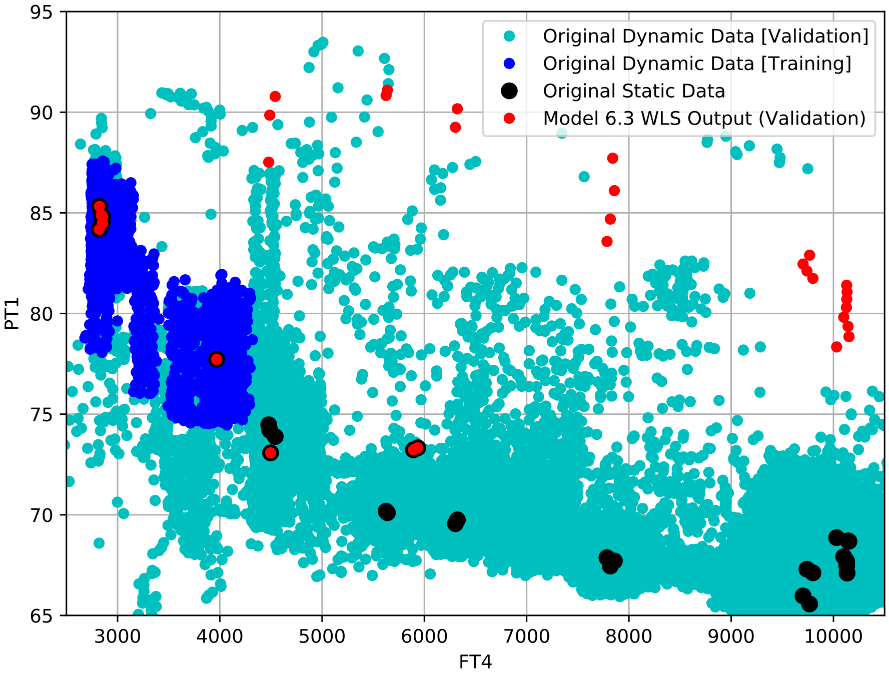
(b)

Source: Author (2022)

that the amount of static data provided during the MGGP algorithm search for regressors was not enough for a good result at all fixed points in the static regime. However, the use of auxiliary information, even if in small amounts, as one of the objectives of the cost function and in the estimation of parameters, allowed the algorithm to find models that have a competitive behavior in a dynamic regime compared to other works. The static RMSE results of Model

$Freitas$ (1) and Model $Freitas$ (2) were not compared with the results found because they were not available.

Figure 6.9 – Free-run simulation on the dataset Z_s for $Model$ 6.3 WLS together with the real data of the fixed operating points for comparison of results. Dynamic training data, Z_d, and validation, Z_v, also presented as a reference.



Source: Author (2022)

7 CONCLUSION AND FUTURE WORK

This work addressed the problem of identifying nonlinear systems using a multi-objective approach based on evolutionary algorithms for the problem of structure selection and parameter estimation. The problem of PDG failure in oil extraction was used as motivation to explore this question. Introductory material on the oil extraction process was presented for contextualization. Introductory material on systems identification and evolutionary algorithms was also presented, together with the main works in each area related to the theme. The MGGP paradigm, used for optimization in this work, was briefly expanded within the EA theme.

A multi-objective MGGP algorithm was proposed to solve the problem of selection of regressors for NARX models with three objectives for minimization, namely: *i.* the one-step-ahead prediction error (dynamic regime error), *ii.* the static error, and *iii.* the number of regressors in the model. For the second objective, the simulation approach in the static regime proposed by Freitas, Barbosa and Aguirre (2021) was used, which has a lower computational cost when compared to the free-run simulation traditionally used. Furthermore, the weighted least squares method was applied to use auxiliary information in the estimation of parameters, together with the dynamic data. Finally, a decision criterion was proposed for choosing the most appropriate model among those presented in the Pareto set.

The proposed algorithm was initially applied in a stochastic system to validate its operation. The results show that the use of auxiliary information in the estimation of parameters through the WLS, when compared with other black-box methods, allows to identify better Pareto sets and, consequently, better models. Another point shown was that the use of three objectives, including information about the static regime in the search for terms for the model, also presents significant performance gains for the models found not only in the static regime but also in the dynamic one. It is also seen that minimizing the number of regressors creates pressure in the search for more parsimonious models without losing performance in the other two objectives. In addition to the results already mentioned, it was also shown that the simulation approach presented in Freitas, Barbosa and Aguirre (2021) provides an algorithm training time approximately nine times smaller and still without losing the performance quality of the models found.

With the validated methodology, the algorithm was applied to model a hydraulic pumping system with real data. The model found by MGGP proved to be competitive compared with other works, having equivalent performance in both regimes (dynamic and static) in many comparisons, even with a smaller number of regressors (parsimoniousness).

Finally, the proposed MGGP algorithm was applied to the offshore oil extraction dataset. The algorithm was able to model a competitive virtual sensor with a good performance in dynamic regime compared to other works even with fewer parameters. The results presented in this work show that the multi-objective MGGP algorithm, together with auxiliary information in the selection of structures and parameter estimation, can find models with good performance in static and dynamic regimes with a reduced number of regressors and parameters in addition to lower computational cost. Another point is that, for none of the datasets used in this work, the best result found had $\lambda = 0$, which confirms that the use of auxiliary information in the estimation of parameters contributes, in general, to more adequate models in both regimes.

Future works include the elaboration of more sophisticated decision criteria for the selection of models in Pareto sets, use of the MGGP approach in the estimation of parameters for model structures, use other forms of auxiliary information (*e.g.*, symmetry properties) in the search for better regressors and the estimation of parameters, and implement modifications in the MGGP algorithm in order to allow restrictions in the search for regressors (obtaining regressors with higher delays more easily) at a lower computational cost.

REFERENCES

ABREU, L. F. **Uso de Informação Auxiliar em Redes Neurais e Formação de Comitês na Identificação de Sistemas Dinâmicos**. Thesis (Master) — Universidade Federal de Minas Gerais, 2013.

AGUIRRE, L. et al. Imposing steady-state performance on identified nonlinear polynomial models by means of constrained parameter estimation. **Control Theory and Applications, IEE Proceedings -**, v. 151, p. 174 – 179, 04 2004.

AGUIRRE, L. A. Introdução à identificação de sistemas; técnicas lineares e não lineares: Teoria e aplicação, 4a edição. **Belo Horizonte**, 2015.

AGUIRRE, L. A.; ALVES, G. B.; CORRÊA, M. V. Steady-state performance constraints for dynamical models based on rbf networks. **Engineering Applications of Artificial Intelligence**, Elsevier, v. 20, n. 7, p. 924–935, 2007.

AGUIRRE, L. A.; BARBOSA, B. H.; BRAGA, A. P. Prediction and simulation errors in parameter estimation for nonlinear systems. **Mechanical Systems and Signal Processing**, Elsevier, v. 24, n. 8, p. 2855–2867, 2010.

AGUIRRE, L. A.; BILLINGS, S. Validating identified nonlinear models with chaotic dynamics. **International Journal of Bifurcation and Chaos**, World Scientific, v. 4, n. 01, p. 109–125, 1994.

AGUIRRE, L. A.; BILLINGS, S. Improved structure selection for nonlinear models based on term clustering. **International journal of control**, Taylor & Francis, v. 62, n. 3, p. 569–587, 1995.

AGUIRRE, L. A.; BILLINGS, S. A. Dynamical effects of overparametrization in nonlinear models. **Physica D: Nonlinear Phenomena**, Elsevier, v. 80, n. 1-2, p. 26–40, 1995.

AGUIRRE, L. A.; RODRIGUES, G. G.; JÁCOME, C. R. Identificação de sistemas não lineares utilizando modelos narmax polinomiais–uma revisão e novos resultados. **SBA Controle e Automação**, v. 9, n. 2, p. 90–106, 1998.

AGUIRRE, L. A. et al. Development of soft sensors for permanent downhole gauges in deepwater oil wells. **Control Engineering Practice**, Elsevier, v. 65, p. 83–99, 2017.

AGÊNCIA NACIONAL DO PETRÓLEO GÁS NATURAL E BIOCOMBUSTÍVEIS. **Anuário Estatístico Brasileiro do Petróleo, Gás Natural e Biocombustíveis 2020**. 2020. Available at: <http://www.anp.gov.br/arquivos/central-conteudos/anuario-estatistico/2020/texto-secao-1.pdf>.

AKAIKE, H. Statistical predictor identification. **Annals of the institute of Statistical Mathematics**, Springer, v. 22, n. 1, p. 203–217, 1970.

AKAIKE, H. A new look at the statistical model identification. **IEEE transactions on automatic control**, Ieee, v. 19, n. 6, p. 716–723, 1974.

ALLEN, D. M. The relationship between variable selection and data agumentation and a method for prediction. **technometrics**, Taylor & Francis, v. 16, n. 1, p. 125–127, 1974.

ALVES, M. A.; CORRÊA, M. V.; AGUIRRE, L. A. Use of self-consistency in the structure selection of narx polynomial models. **International Journal of Modelling, Identification and Control**, Inderscience Publishers, v. 15, n. 1, p. 1–12, 2012.

AMERICAN PETROLEUM INSTITUTE. **API Specification 6A - Specification for Wellhead and Christmas Tree Equipment**. 20. ed. United States of America, 2010. 40 p. Available at: <http://www.api.org/publications>.

ANTONELO, E. A.; CAMPONOGARA, E.; FOSS, B. Echo state networks for data-driven downhole pressure estimation in gas-lift oil wells. **Neural Networks**, Elsevier, v. 85, p. 106–117, 2017.

APIO, A. et al. Pdg pressure estimation in offshore oil well: extended kalman filter vs. artificial neural networks. **IFAC-PapersOnLine**, Elsevier, v. 52, n. 1, p. 508–513, 2019.

ARCELORMITTAL. **SURF: Subsea umbilicals, risers and flowlines**. United States, 2019. Available at: <https://projects.arcelormittal.com/energy/segments_and_applications/22728/SURF/language/EN>.

BAI, Y.; BAI, Q. **Subsea engineering handbook**. Cambridge, MA, United States: Gulf Professional Publishing, 2018.

BANZHAF, W. et al. **Genetic programming: an introduction: on the automatic evolution of computer programs and its applications**. San Francisco, CA, United States: Morgan Kaufmann Publishers Inc., 1998.

BARBOSA, A. M.; TAKAHASHI, R. H. C.; AGUIRRE, L. A. Equivalence of non-linear model structures based on pareto uncertainty. **IET Control Theory & Applications**, IET, v. 9, n. 16, p. 2423–2429, 2015.

BARBOSA, B. H. et al. Black and gray-box identification of a hydraulic pumping system. **IEEE Transactions on control systems technology**, IEEE, v. 19, n. 2, p. 398–406, 2011.

BARBOSA, B. H. et al. Downhole pressure estimation using committee machines and neural networks. **IFAC-PapersOnLine**, Elsevier, v. 48, n. 6, p. 286–291, 2015.

BARROSO, M. F.; TAKAHASHI, R. H.; AGUIRRE, L. A. Multi-objective parameter estimation via minimal correlation criterion. **Journal of Process Control**, Elsevier, v. 17, n. 4, p. 321–332, 2007.

BHAVANI, N. et al. Soft sensor for temperature measurement in gas turbine power plant. **International Journal of Applied Engineering Research**, v. 9, n. 23, p. 21305–21316, 2014.

BILLINGS, S.; AGUIRRE, L. A. Effects of the sampling time on the dynamics and identification of nonlinear models. **International journal of Bifurcation and Chaos**, World Scientific, v. 5, n. 06, p. 1541–1556, 1995.

BILLINGS, S.; CHEN, S.; KORENBERG, M. Identification of mimo non-linear systems using a forward-regression orthogonal estimator. **International journal of control**, Taylor & Francis, v. 49, n. 6, p. 2157–2189, 1989.

BILLINGS, S. A. Identification of nonlinear systems–a survey. In: IET. **IEE Proceedings D (Control Theory and Applications)**. Hitchin, United Kingdom, 1980. v. 127, n. 6, p. 272–285.

BISHOP, C. M. et al. **Neural networks for pattern recognition**. Oxford: Oxford University Press, 1995.

BREIMAN, L.; SPECTOR, P. Submodel selection and evaluation in regression. the x-random case. **International statistical review/revue internationale de Statistique**, JSTOR, p. 291–319, 1992.

BROOMHEAD, D. S.; LOWE, D. **Radial basis functions, multi-variable functional interpolation and adaptive networks**. United Kingdom, 1988.

CARVAJAL, G.; MAUCEC, M.; CULLICK, S. **Intelligent digital oil and gas fields: concepts, collaboration, and right-time decisions**. Oxford, United Kingdom: Gulf Professional Publishing, 2017.

CASALI, A. et al. Particle size distribution soft-sensor for a grinding circuit. **Powder Technology**, Elsevier, v. 99, n. 1, p. 15–21, 1998.

CASTRO, H. C. **A modified MGGP algorithm for structure selection of NARMAX models**. Thesis (Master) — Federal University of Lavras, 2021.

CASTRO, H. C.; BARBOSA, B. H. G. Algoritmos multi-objetivos para detecção de estruturas em modelos NARX utilizando técnicas PEM e SEM. **Anais do 14º Simpósio Brasileiro de Automação Inteligente**, v. 1, p. 2946–2951, 2019.

CASTRO, H. C.; BARBOSA, B. H. G. Multi-gene genetic programming for structure selection of polynomial NARMAX models. **Anais da Sociedade Brasileira de Automática**, 2020.

CHEN, Q. et al. Genetic algorithm with an improved fitness function for (n) arx modelling. **Mechanical Systems and Signal Processing**, Elsevier, v. 21, n. 2, p. 994–1007, 2007.

CHEN, S.; BILLINGS, S. A. Representations of non-linear systems: the narmax model. **International journal of control**, Taylor & Francis, v. 49, n. 3, p. 1013–1032, 1989.

CHEN, S.; BILLINGS, S. A.; LUO, W. Orthogonal least squares methods and their application to non-linear system identification. **International Journal of control**, Taylor & Francis, v. 50, n. 5, p. 1873–1896, 1989.

CHEN, S. et al. Symmetric rbf classifier for nonlinear detection in multiple-antenna-aided systems. **IEEE transactions on neural networks**, IEEE, v. 19, n. 5, p. 737–745, 2008.

COELHO, M. C. da S. **Modelos de Hammerstein e de Wiener: conexões com modelos narx e sua aplicação em identificação de sistemas não-lineares**. Thesis (Master) — Universidade Federal de Minas Gerais, Belo Horizonte, 2002.

CORRÊA, M. V. **Identificação caixa-cinza de sistemas não lineares utilizando representações narmax racionais e polinomiais.** Dissertation (PhD) — Universidade Federal de Minas Gerais, 2001.

DARWIN, C. **The Origin of Species; And, the Descent of Man**. New York: Modern library, 1859.

DAVIES, D. R.; AGGREY, G. H. et al. Tracking the state and diagnosing down hole permanent sensors in intelligent well completions with artificial neural network. In: SOCIETY OF PETROLEUM ENGINEERS. **Offshore Europe**. Aberdeen, Scotland, U.K., 2007.

DEB, K. et al. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-ii. In: SPRINGER. **International conference on parallel problem solving from nature**. Berlin, 2000. p. 849–858.

DEB, K. et al. A fast and elitist multiobjective genetic algorithm: Nsga-ii. **IEEE transactions on evolutionary computation**, IEEE, v. 6, n. 2, p. 182–197, 2002.

DESAI, K. et al. Soft-sensor development for fed-batch bioreactors using support vector regression. **Biochemical Engineering Journal**, Elsevier, v. 27, n. 3, p. 225–239, 2006.

DEVOGELAERE, D. et al. Application of feedforward neural networks for soft sensors in the sugar industry. In: IEEE. **VII Brazilian Symposium on Neural Networks, 2002. SBRN 2002. Proceedings.** Pernambuco, Brazil, 2002. p. 2–6.

DONATE, J. P. et al. Weighted cross-validation evolving artificial neural networks to forecast time series. In: SPRINGER. **Soft Computing Models in Industrial and Environmental Applications, 6th International Conference SOCO 2011**. Berlin, 2011. p. 147–154.

DONG, D.; MCAVOY, T. J. Nonlinear principal component analysis—based on principal curves and neural networks. **Computers & Chemical Engineering**, Elsevier, v. 20, n. 1, p. 65–78, 1996.

DONG, D.; MCAVOY, T. J.; CHANG, L. Emission monitoring using multivariate soft sensors. In: IEEE. **Proceedings of 1995 American Control Conference-ACC'95**. Seattle, WA, USA, 1995. v. 1, p. 761–765.

EIBEN, A. E.; SMITH, J. E. et al. **Introduction to evolutionary computing**. Berlin: Springer, 2003. v. 53.

ELLDAKLI, F. Gas lift system. **Department of Petroleum Engineering, Texas Tech University, USA**, 2017.

FALSONE, A.; PIRODDI, L.; PRANDINI, M. A randomized algorithm for nonlinear model structure selection. **Automatica**, Elsevier, v. 60, p. 227–238, 2015.

FENG, R.; SHEN, W.; SHAO, H. A soft sensor modeling approach using support vector machines. In: IEEE. **Proceedings of the 2003 American Control Conference**. Denver, CO, USA, 2003. v. 5, p. 3702–3707.

FERARIU, L.; PATELLI, A. Multiobjective genetic programming for nonlinear system identification. In: **International Conference on Adaptive and Natural Computing Algorithms**. Berlin, Heidelberg: Springer, 2009. p. 233–242.

FOGEL, L. J. Toward inductive inference automata. In: **Proceedings of the Congress**. Munich, Germany: International Federation for Information Processing, 1962. v. 62, p. 395–400.

FORSSELL, U.; LINDSKOG, P. Combining semi-physical and neural network modeling: An example ofits usefulness. **IFAC Proceedings Volumes**, Elsevier, v. 30, n. 11, p. 767–770, 1997.

FORTUNA, L. et al. Virtual instruments based on stacked neural networks to improve product quality monitoring in a refinery. **IEEE Transactions on Instrumentation and Measurement**, IEEE, v. 56, n. 1, p. 95–101, 2007.

FORTUNA, L. et al. **Soft sensors for monitoring and control of industrial processes**. London: Springer Science & Business Media, 2007.

FORTUNA, L. et al. Fuzzy activated neural models for product quality monitoring in refineries. **IFAC Proceedings Volumes**, Elsevier, v. 38, n. 1, p. 159–164, 2005.

FORTUNA, L. et al. Virtual instruments for the what-if analysis of a process for pollution minimization in an industrial application. In: IEEE. **2006 14th Mediterranean Conference on Control and Automation**. Ancona, Italy, 2006. p. 1–4.

FORTUNA, L.; GRAZIANI, S.; XIBILIA, M. G. Virtual instruments in refineries. **IEEE instrumentation & measurement magazine**, IEEE, v. 8, n. 4, p. 26–34, 2005.

FREITAS, L.; BARBOSA, B. H.; AGUIRRE, L. A. Including steady-state information in nonlinear models: an application to the development of soft-sensors. **Engineering Applications of Artificial Intelligence**, Elsevier, v. 102, p. 104253, 2021.

FROTA, H. M.; DESTRO, W. Reliability evolution of permanent downhole gauges for campos basin subsea wells: A 10-year case study. In: SOCIETY OF PETROLEUM ENGINEERS. **SPE Annual Technical Conference and Exhibition**. San Antonio, Texas, 2006.

GALVAO, R. K. H. et al. A method for calibration and validation subset partitioning. **Talanta**, Elsevier, v. 67, n. 4, p. 736–740, 2005.

GANI, W.; LIMAM, M. A kernel distance-based representative subset selection method. **Journal of Statistical Computation and Simulation**, Taylor & Francis, v. 86, n. 1, p. 135–148, 2016.

GAO, T. et al. Spxye: an improved method for partitioning training and validation sets. **Cluster Computing**, Springer, v. 22, n. 2, p. 3069–3078, 2019.

GAUSS, K. F. Theory of the motion of the heavenly bodies moving about the sun in conic section. **tmhb**, 1963.

GHAREEB, W.; SAADANY, E. E. Multi-gene genetic programming for short term load forecasting. In: IEEE. **2013 3rd International Conference on Electric Power and Energy Conversion Systems**. Istanbul, 2013. p. 1–5.

GOLDBERG, D. E.; HOLLAND, J. H. Genetic algorithms and machine learning. **Machine learning**, Springer, v. 3, n. 2, p. 95–99, 1988.

GRAZIANI, S.; XIBILIA, M. G. Deep structures for a reformer unit soft sensor. In: IEEE. **2018 IEEE 16th International Conference on Industrial Informatics (INDIN)**. Porto, 2018. p. 927–932.

GRAZIANI, S.; XIBILIA, M. G. Design of a soft sensor for an industrial plant with unknown delay by using deep learning. In: IEEE. **2019 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)**. Auckland, 2019. p. 1–6.

GUERREIRO, A. P.; FONSECA, C. M.; PAQUETE, L. The hypervolume indicator: Problems and algorithms. **arXiv preprint arXiv:2005.00515**, 2020.

HAFIZ, F.; SWAIN, A.; MENDES, E. Multi-objective evolutionary framework for non-linear system identification: A comprehensive investigation. **Neurocomputing**, Elsevier, v. 386, p. 257–280, 2020.

HAYKIN, S. **Neural networks: a comprehensive foundation**. Hamilton: Pearson Prentice-Hall, 2007.

HINCHLIFFE, M. et al. Modelling chemical process systems using a multi-gene genetic programming algorithm. In: **Genetic Programming: Proceedings of the First Annual Conference (late breaking papers)**. Cambridge: MIT Press, 1996. p. 56–65.

HINCHLIFFE, M. P. **Dynamic modelling using genetic programming**. Dissertation (PhD) — University of Newcastle upon Tyne, UK, 2001.

HINCHLIFFE, M. P.; WILLIS, M. J. Dynamic systems modelling using genetic programming. **Computers & Chemical Engineering**, v. 27, n. 12, p. 1841 – 1854, 2003.

HOLLAND, J. H. Adaptation in natural and artificial systems. **The University of Michigan Press**, 1975.

JADID, M.; OPSAL, A.; WHITE, A. The pressure's on: Innovations in gas lift. **Oilfield Review**, v. 18, n. 4, p. 44–52, 2006.

JOHANSEN, T. A.; FOSS, B. A. A narmax model representation for adaptive control based on local models. 1992.

JOLLIFFE, I. T. Principal components in regression analysis. In: **Principal component analysis**. New York: Springer, 1986. p. 129–155.

JONES, J. P.; BILLINGS, S. Recursive algorithm for computing the frequency response of a class of non-linear difference equation models. **International Journal of Control**, Taylor & Francis, v. 50, n. 5, p. 1925–1940, 1989.

KADLEC, P.; GABRYS, B. Soft sensors: where are we and what are the current and future challenges? **IFAC Proceedings Volumes**, Elsevier, v. 42, n. 19, p. 572–577, 2009.

KADLEC, P.; GABRYS, B.; STRANDT, S. Data-driven soft sensors in the process industry. **Computers & chemical engineering**, Elsevier, v. 33, n. 4, p. 795–814, 2009.

KENNARD, R. W.; STONE, L. A. Computer aided design of experiments. **Technometrics**, Taylor & Francis, v. 11, n. 1, p. 137–148, 1969.

KHALIFEH, M.; SAASEN, A. Introduction to permanent plug and abandonment of wells. Springer, 2020.

KOHAVI, R. A study of cross-validation and bootstrap for accuracy estimation and model selection. In: **Ijcai: Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2**. Montreal, Canada: Morgan Kaufmann Publishers Inc., 1995. v. 14, n. 2, p. 1137–1143.

KORENBERG, M. et al. Orthogonal parameter estimation algorithm for non-linear stochastic systems. **International Journal of Control**, Taylor & Francis, v. 48, n. 1, p. 193–210, 1988.

KOZA, J. R. **Genetic programming: on the programming of computers by means of natural selection**. Cambridge: MIT press, 1992.

KOZA, J. R. **Genetic Programming II: Automatic Discovery of Reusable Programs**. Cambridge, MA: MIT Press, 1994.

KOZA, J. R.; POLI, R. Genetic programming. In: **Search methodologies**. Boston: Springer, 2005. p. 127–164.

LEGENDRE, A. M. **Nouvelles méthodes pour la détermination des orbites des comètes**. Paris: F. Didot, 1805.

LEONTARITIS, I.; BILLINGS, S. Model selection and validation methods for non-linear systems. **International Journal of Control**, Taylor & Francis, v. 45, n. 1, p. 311–341, 1987.

LEONTARITIS, I.; BILLINGS, S. Experimental design and identifiability for non-linear systems. **International Journal of Systems Science**, Taylor & Francis, v. 18, n. 1, p. 189–202, 1987a.

LEONTARITIS, I.; BILLINGS, S. A. Input-output parametric models for non-linear systems part i: deterministic non-linear systems. **International journal of control**, Taylor & Francis, v. 41, n. 2, p. 303–328, 1985a.

LEONTARITIS, I.; BILLINGS, S. A. Input-output parametric models for non-linear systems part ii: stochastic non-linear systems. **International journal of control**, Taylor & Francis, v. 41, n. 2, p. 329–344, 1985b.

LI, C. J.; JEON, Y. Genetic algorithm in identifying non linear auto regressive with exogenous input models for non linear systems. In: IEEE. **1993 American Control Conference**. San Francisco, 1993. p. 2305–2309.

LI, M.; YAO, X. Quality evaluation of solution sets in multiobjective optimisation: A survey. **ACM Computing Surveys (CSUR)**, ACM New York, NY, USA, v. 52, n. 2, p. 1–38, 2019.

LIN, B. et al. A systematic approach for soft sensor development. **Computers & chemical engineering**, Elsevier, v. 31, n. 5-6, p. 419–425, 2007.

LINDEN, R. **Algoritmos genéticos**. 2ª. ed. Rio de Janeiro: Brasport, 2008.

LINDEN, R. Algoritmos genéticos na indústria do petróleo: Uma visão geral. **Revista da Engenharia de Instalações no Mar**, n. 1, p. 21, 2008.

MACIAS, J. J.; ANGELOV, P.; ZHOU, X. A method for predicting quality of the crude oil distillation. In: IEEE. **2006 International Symposium on Evolving Fuzzy Systems**. Ambelside, 2006. p. 214–220.

MADÁR, J.; ABONYI, J.; SZEIFERT, F. Genetic programming for the identification of nonlinear input- output models. **Industrial & engineering chemistry research**, ACS Publications, v. 44, n. 9, p. 3178–3186, 2005.

MAO, K.; BILLINGS, S. Algorithms for minimal model structure detection in nonlinear dynamic system identification. **International journal of control**, Taylor & Francis, v. 68, n. 2, p. 311–330, 1997.

MARTINS, S. A. M.; NEPOMUCENO, E. G.; BARROSO, M. F. S. Improved structure detection for polynomial narx models using a multiobjective error reduction ratio. **Journal of Control, Automation and Electrical Systems**, Springer, v. 24, n. 6, p. 764–772, 2013.

MEHR, A. D.; KAHYA, E. A pareto-optimal moving average multigene genetic programming model for daily streamflow prediction. **Journal of hydrology**, Elsevier, v. 549, p. 603–615, 2017.

MENDES, E.; BILLINGS, S. A. An alternative solution to the model structure selection problem. **IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans**, IEEE, v. 31, n. 6, p. 597–608, 2001.

MORAIS, G. A. de et al. Soft sensors design in a petrochemical process using an evolutionary algorithm. **Measurement**, Elsevier, v. 148, p. 106920, 2019.

MOSTELLER, F.; TUKEY, J. W. Data analysis, including statistics. **Handbook of social psychology**, v. 2, p. 80–203, 1968.

MOTA, F. et al. Identificação de um sistema de bombeamento hidráulico com algoritmo evolucionário multi-objetivo. **Anais da Sociedade Brasileira de Automática**, 2020.

NEPOMUCENO, E. G. et al. **Identificação multiobjetivo de sistemas não-lineares**. Thesis (Master) — Universidade Federal de Minas Gerais, 2002.

NEPOMUCENO, E. G.; TAKAHASHI, R. H.; AGUIRRE, L. A. Multiobjective parameter estimation for non-linear systems: affine information and least-squares formulation. **International Journal of Control**, Taylor & Francis, v. 80, n. 6, p. 863–871, 2007.

NEPOMUCENO, E. G. et al. Nonlinear identification using prior knowledge of fixed points: a multiobjective approach. **International Journal of Bifurcation and Chaos**, World Scientific, v. 13, n. 05, p. 1229–1246, 2003.

NETO, J. B. O.; COSTA, A. J. D. A petrobrás e a exploração de petróleo offshore no brasil: um approach evolucionário. **Revista Brasileira de Economia**, SciELO Brasil, v. 61, n. 1, p. 95–109, 2007.

NIAZKAR, M.; NIAZKAR, H. R. Covid-19 outbreak: Application of multi-gene genetic programming to country-based prediction models. **Electronic Journal of General Medicine**, v. 17, n. 5, 2020.

NOMIKOS, P.; MACGREGOR, J. F. Multi-way partial least squares in monitoring batch processes. **Chemometrics and intelligent laboratory systems**, Elsevier, v. 30, n. 1, p. 97–108, 1995.

NOR, N. et al. Well connection optimization in integrated subsurface and surface facilities: an industrial case study. **Journal of Petroleum Exploration and Production Technology**, Springer, v. 9, n. 4, p. 2921–2926, 2019.

ONESUBSEA. **Standard Vertical Subsea Trees: Integrated offshore offering for reliable, high-quality, and capital-efficient performance**. 2018. Available at: <https://www.onesubsea.slb.com/-/media/onesubsea/files/brochure/oss-standard-vertical-subsea-trees-br.ashx>.

ONESUBSEA. **Horizontal Subsea Tree Systems: Industry-standard, cost-effective systems for shallow and deep water**. 2020. Available at: <https://www.onesubsea.slb.com/subsea-production-systems/subsea-tree-systems/horizontal-subsea-tree-system#related-information>.

OROVE, J.; OSEGI, N.; EKE, B. A multi-gene genetic programming application for predicting students failure at school. **arXiv preprint arXiv:1503.03211**, 2015.

OSORIO, D. et al. Soft-sensor for on-line estimation of ethanol concentrations in wine stills. **Journal of food engineering**, Elsevier, v. 87, n. 4, p. 571–577, 2008.

OUYANG, L.-B.; KIKANI, J. et al. Improving permanent downhole gauge (pdg) data processing via wavelet analysis. In: ONEPETRO. **European Petroleum Conference**. Aberdeen, 2002.

PALUMBO, P.; PIRODDI, L. Seismic behaviour of buttress dams: Non-linear modelling of a damaged buttress based on arx/narx models. **Journal of sound and vibration**, Elsevier, v. 239, n. 3, p. 405–422, 2001.

PANI, A. K.; AMIN, K. G.; MOHANTA, H. K. Soft sensing of product quality in the debutanizer column with principal component analysis and feed-forward artificial neural network. **Alexandria Engineering Journal**, Elsevier, v. 55, n. 2, p. 1667–1674, 2016.

PIRODDI, L. Simulation error minimisation methods for narx model identification. **International Journal of Modelling, Identification and Control**, Inderscience Publishers, v. 3, n. 4, p. 392–403, 2008.

PIRODDI, L.; SPINELLI, W. An identification algorithm for polynomial narx models based on simulation error minimization. **International Journal of Control**, Taylor & Francis, v. 76, n. 17, p. 1767–1781, 2003.

POLI, R.; LANGDON, W. B.; MCPHEE, N. F. A field guide to genetic programming. **Published via http://lulu. com and freely available at http://www. gp-field-guide. org. uk.(With contributions by JR Koza)**, 2008.

POPE, K. J.; RAYNER, P. J. Non-linear system identification using bayesian inference. In: IEEE. **Proceedings of ICASSP'94. IEEE International Conference on Acoustics, Speech and Signal Processing**. Adelaide, 1994. v. 4, p. 457–460.

PRASAD, V. et al. Product property and production rate control of styrene polymerization. **Journal of Process Control**, Elsevier, v. 12, n. 3, p. 353–372, 2002.

RADHAKRISHNAN, V.; MOHAMED, A. Neural networks for the identification and control of blast furnace hot metal quality. **Journal of process control**, Elsevier, v. 10, n. 6, p. 509–524, 2000.

RECHENBERG, I. Cybernetic solution path of an experimental problem. **Royal Aircraft Establishment Library Translation 1122**, 1965.

RECHENBERG, I. Evolutionsstrategien. In: **Simulationsmethoden in der Medizin und Biologie**. Hannover: Springer, 1978. p. 83–114.

REID, M. **Barossa Offshore Project: ConocoPhillips Work Scopes**. Australia, 2018. Available at: <http://static.conocophillips.com/files/resources/project-wide-drilling-and-subsea-work-packages-pre.pdf>.

RIAHI-MADVAR, H. et al. Pareto optimal multigene genetic programming for prediction of longitudinal dispersion coefficient. **Water Resources Management**, v. 33, n. 3, 2019. Available at: <http://link.springer.com/article/10.1007/s11269-018-2139-6>.

RIBEIRO, A. H.; AGUIRRE, L. A. Selecting transients automatically for the identification of models for an oil well. **IFAC-PapersOnLine**, Elsevier, v. 48, n. 6, p. 154–158, 2015.

RISSANEN, J. **Stochastic complexity in statistical inquiry**. River Edge: World Scientific, 1989. v. 15.

RIZZO, A. Soft sensors and artificial intelligence for nuclear fusion experiments. In: IEEE. **Melecon 2010-2010 15th IEEE Mediterranean Electrotechnical Conference**. Valeta, 2010. p. 1068–1072.

RODRIGUEZ-VAZQUEZ, K.; FLEMING, P. J. Multi-objective genetic programming for nonlinear system identification. **Electronics Letters**, IET, v. 34, n. 9, p. 930–931, 1998.

RODRIGUEZ-VAZQUEZ, K.; FONSECA, C. M.; FLEMING, P. J. Identifying the structure of nonlinear dynamic systems using multiobjective genetic programming. **IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans**, IEEE, v. 34, n. 4, p. 531–545, 2004.

ROVERSO, D. Empirical ensemble-based virtual sensing-a novel approach to oil-in-water monitoring. In: **Oil-in-Water Monitoring Workshop**. Aberdeen, UK: East Kilbride TUV NEL, 2009.

SINGH, H.; PANI, A. K.; MOHANTA, H. K. Quality monitoring in petroleum refinery with regression neural network: Improving prediction accuracy with appropriate design of training set. **Measurement**, Elsevier, v. 134, p. 698–709, 2019.

SMOLA, A. J.; SCHÖLKOPF, B. A tutorial on support vector regression. **Statistics and computing**, Springer, v. 14, n. 3, p. 199–222, 2004.

SNEE, R. D. Validation of regression models: methods and examples. **Technometrics**, Taylor & Francis Group, v. 19, n. 4, p. 415–428, 1977.

STONE, M. Cross-validatory choice and assessment of statistical predictions. **Journal of the Royal Statistical Society: Series B (Methodological)**, Wiley Online Library, v. 36, n. 2, p. 111–133, 1974.

STONE, M. An asymptotic equivalence of choice of model by cross-validation and akaike's criterion. **Journal of the Royal Statistical Society: Series B (Methodological)**, Wiley Online Library, v. 39, n. 1, p. 44–47, 1977.

SU, H. B.; FAN, L.; SCHLUP, J. R. Monitoring the process of curing of epoxy/graphite fiber composites with a recurrent neural network as a soft sensor. **Engineering Applications of Artificial Intelligence**, Elsevier, v. 11, n. 2, p. 293–306, 1998.

SUI, D. et al. Ensemble methods for process monitoring in oil and gas industry operations. **Journal of Natural Gas Science and Engineering**, Elsevier, v. 3, n. 6, p. 748–753, 2011.

SUJATHA, K. et al. Soft sensor for flame temperature measurement and iot based monitoring in power plants. **Materials Today: Proceedings**, Elsevier, v. 5, n. 4, p. 10755–10762, 2018.

TEIXEIRA, B. O. et al. Ukf-based data-driven soft sensing: A case study of a gas-lifted oil well. **IFAC Proceedings Volumes**, Elsevier, v. 45, n. 16, p. 918–923, 2012.

TEIXEIRA, B. O. et al. Data-driven soft sensor of downhole pressure for a gas-lift oil well. **Control Engineering Practice**, Elsevier, v. 22, p. 34–43, 2014.

THOMAS, J. E. **Fundamentos de engenharia de petróleo**. Rio de Janeiro: Interciência, 2004.

U. S. ENERGY INFORMATION ADMINISTRATION. **Short-Term Energy Outlook (STEO)**. 2020. Available at: <https://www.eia.gov/outlooks/steo/pdf/steo_full.pdf>.

VAPNIK, V. **Statistical learning theory**. New York: Wiley, 1998. v. 1. 768 p.

VAPNIK, V.; GOLOWICH, S.; SMOLA, A. Support vector method for function approximation, regression estimation and signal processing. **Advances in neural information processing systems**, v. 9, 1996.

VIKHAR, P. A. Evolutionary algorithms: A critical review and its future prospects. In: IEEE. **2016 International conference on global trends in signal processing, information computing and communication (ICGTSPICC)**. New York, 2016. p. 261–265.

VILLELA, M. J. R. **Análise do comportamento da temperatura em sistemas de produção de petróleo: comparação entre completação seca e molhada**. Thesis (Master) — Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2004.

VOLTERRA, V. Theory of functionals and of integral and integro. **Differential Equations**, v. 61, 1930.

WAN, R. **Advanced well completion engineering**. Oxford: Gulf professional publishing, 2011.

WANG, L. et al. Radial basis function neural networks-based modeling of the membrane separation process: hydrogen recovery from refinery gases. **Journal of Natural Gas Chemistry**, Elsevier, v. 15, n. 3, p. 230–234, 2006.

WEI, H.-L.; BILLINGS, S. A. Model structure selection using an integrated forward orthogonal search algorithm assisted by squared correlation and mutual information. **International Journal of Modelling, Identification and Control**, Inderscience Publishers, v. 3, n. 4, p. 341–356, 2008.

WIENER, N. Nonlinear problems in random theory, tech. **Press MIT and John Willey, New York**, 1958.

WIGREN, T. Recursive prediction error identification using the nonlinear wiener model. **Automatica**, Elsevier, v. 29, n. 4, p. 1011–1025, 1993.

WOLD, S. et al. Multi-way principal components-and pls-analysis. **Journal of chemometrics**, Wiley Online Library, v. 1, n. 1, p. 41–56, 1987.

ZAKARIA, M. Z. et al. Comparison between multi-objective and single-objective optimization for the modeling of dynamic systems. **Journal of Systems and Control Engineering, Part I. Proc. Instn. Mech. Engrs.**, v. 226, p. 994–1005, 07 2012.

ZHANG, H.-Y.; JI, Q.; FAN, Y. What drives the formation of global oil trade patterns? **Energy Economics**, Elsevier, v. 49, p. 639–648, 2015.

ZHANG, P. **Advanced industrial control technology**. Oxford: William Andrew, 2010.

ZHU, Q.; BILLINGS, A. Parameter estimation for stochastic nonlinear rational models. **International Journal of Control**, Taylor & Francis, v. 57, n. 2, p. 309–333, 1993.

ZITZLER, E.; LAUMANNS, M.; THIELE, L. Spea2: Improving the strength pareto evolutionary algorithm. **TIK-report**, Eidgenössische Technische Hochschule Zürich (ETH), Institut für Technische, v. 103, 2001.

ZITZLER, E.; THIELE, L. Multiobjective optimization using evolutionary algorithms—a comparative case study. In: SPRINGER. **International conference on parallel problem solving from nature**. Berlin, Heidelberg, 1998. p. 292–301.