GIULIO CESARE MASTROCINQUE SANTO

Data Mining Techniques Applied to Historical Data of Industrial Processes as a Tool to Find Time Intervals Suitable for System Identification

São Paulo
2020

GIULIO CESARE MASTROCINQUE SANTO

Data Mining Techniques Applied to Historical Data of Industrial Processes as a Tool to Find Time Intervals Suitable for System Identification

Corrected Version

Dissertation submitted to Escola Politécnica of the Universidade de São Paulo for the degree of Master of Science.

São Paulo
2020

GIULIO CESARE MASTROCINQUE SANTO

Data Mining Techniques Applied to Historical Data of Industrial Processes as a Tool to Find Time Intervals Suitable for System Identification

Corrected Version

Dissertation submitted to Escola Politécnica of the Universidade de São Paulo for the degree of Master of Science.

Concentration area:

3139 - Systems Engineering

Advisor:

Prof. Dr. Claudio Garcia

São Paulo
2020

Este exemplar foi revisado e corrigido em relação à versão original, sob responsabilidade única do autor e com a anuência de seu orientador.

São Paulo, _____ de _____ de _____

Assinatura do autor: _____

Assinatura do orientador: _____

Catalogação-na-publicação

Santo, Giulio Cesare Mastrocinque
Data Mining Techniques Applied to Historical Data of Industrial Processes as a Tool to Find Time Intervals Suitable for System Identification / G. C. M. Santo -- versão corr. -- São Paulo, 2020.
163 p.

Dissertação (Mestrado) - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Telecomunicações e Controle.

1.Dados Históricos 2.Identificação de Sistemas 3.Medidas de Informação 4.Mineração de Dados 5.Segmentação de Dados I.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Telecomunicações e Controle II.t.

ACKNOWLEDGMENTS (*In Portuguese*)

Gostaria de agradecer, em primeiro lugar, ao meu professor e orientador Professor Dr. Claudio Garcia, o qual conheço desde minha graduação e por quem tenho grande apreço e admiração. Além de ter me incentivado desde o início a escolher um tema cativante e inovador, a sua orientação foi imprescindível para a obtenção de um trabalho de qualidade.

Aos meus grandes amigos e familiares, sem os quais eu não teria superado os momentos difíceis, e que sempre me fizeram acreditar que tudo era possível.

Aos meus colegas de mestrado, Victor, Edwin, Manuel e Marco, com os quais tive momentos alegres assistindo disciplinas, estudando e realizando visitas técnicas a empresas; e ao colega Rodrigo Juliani, pelas discussões filosóficas, pelas análises críticas e por todas as sugestões que muito agregaram ao trabalho.

Agradeço, especialmente, à Unidade de Industrialização do Xisto (SIX) da Petrobras e, em particular, ao Carlos Roberto Chaves, pelo compartilhamento dos dados do forno petroquímico, os quais em muito enriqueceram a qualidade do trabalho.

Por fim, agradeço a todos aqueles que contribuíram de alguma forma para tornar esse trabalho possível, seja através de sugestões, dicas ou conselhos.

"For millions of years, mankind lived just like the animals. Then something happened, which unleashed the power of our imagination. We learned to talk and we learned to listen. Speech has allowed the communication of ideas, enabling human beings to work together to build the impossible. Mankind's greatest achievements have come about by talking, and its greatest failures by not talking. It doesn't have to be like this. Our greatest hopes could become reality in the future. With the technology at our disposal, the possibilities are unbounded. All we need to do is make sure we keep talking."

-Stephen Hawking

RESUMO

SANTO, G. C. M. *Data Mining Techniques Applied to Historical Data of Industrial Processes as a Tool to Find Time Intervals Suitable for System Identification.* Dissertação (Mestrado) – Escola Politécnica da Universidade de São Paulo, São Paulo, Brasil, 2020.

A Identificação de Sistemas é um conjunto de técnicas para estimação de modelos tradicionalmente utilizada pelas indústrias para aprimorar e otimizar os seus processos. A estimação de modelos dinâmicos de processos requer a existência de dados informativos e representativos do sistema, os quais são normalmente gerados através da realização de experimentos físicos nas plantas. Tais procedimentos muitas vezes necessitam ser executados múltiplas vezes para produzir dados adequados, podendo resultar em produtos fora de especificação. Por outro lado, o surgimento de softwares poderosos de armazenamento e gerenciamento de dados e a constante evolução de conhecimento nas áreas de mineração e ciência de dados representam uma possibilidade de quebra de paradigma na indústria, em que soluções robustas orientadas a dados podem ser adotadas.

A utilização direta de dados históricos para a extração de informações úteis de processos industriais é parte central deste trabalho, em que se propõe a comparação de técnicas de mineração de dados com o objetivo de encontrar intervalos temporais com informações suficientes para a realização de identificação de sistemas. Para esse propósito, uma revisão detalhada da literatura a respeito desse problema é inicialmente apresentada. Em seguida, diferentes algoritmos de mineração de dados são aplicados tanto em sistemas de uma entrada e uma saída quanto em sistemas multientradas, multisaídas operando em malha aberta e em malha fechada. Dados de simulação são utilizados para exemplificar didaticamente o funcionamento de cada método e para validar os resultados em casos ideais. Modelos regressivos são então estimados com os intervalos obtidos, os quais são utilizados para a realização de validações cruzadas. Finalmente, os métodos propostos são aplicados em dados reais multivariáveis provenientes de um forno industrial petroquímico.

Os resultados obtidos através de dados de simulação mostram que as estratégias de mineração de dados propostas permitiram a obtenção de bons modelos em cenários de validação cruzada com 1, 10, 100 e infinitos passos de predição. As aplicações em dados reais, por sua vez, revelaram-se desafiadoras devido à natureza ruidosa dos dados e devido a escassez de intervalos históricos nos quais todas as entradas do sistema multivariável são suficientemente ativas para produzir um modelo. No entanto, esse problema é contornado através da utilização de múltiplos intervalos no processo de estimação de parâmetros, elucidando que os algoritmos adotados também permitem a obtenção de modelos razoáveis em cenários reais.

Palavras-Chave – Ciência de Dados; Dados Históricos; Identificação de Sistemas; Mineração de Dados; Número de Condicionamento; Posto Efetivo; Qualidade de Dados; Segmentação de Dados; Sistemas Mutivariáveis.

ABSTRACT

SANTO, G. C. M. *Data Mining Techniques Applied to Historical Data of Industrial Processes as a Tool to Find Time Intervals Suitable for System Identification.* Masters dissertation – Polytechnic School of the University of São Paulo, São Paulo, Brasil, 2020.

System Identification is a set of model estimation techniques traditionally used by industries to improve and optimize their processes. Estimating dynamic process models requires the existence of informative and representative data of the system, which are usually generated through physical experiments on the plants. However, such procedures often need to be performed multiple times to produce adequate datasets, which may result in products that are out of specification. On the other hand, the emergence of powerful data storage and management software, as well as the constant development in the areas of mining and data science represent a potential paradigm break in industry, in which robust data-driven solutions can be adopted.

The direct use of historical data to extract useful information from industrial processes is a central part of this work, in which it is proposed a comparison of data mining techniques with the objective of finding time intervals with sufficient information to perform system identification. For this purpose, a detailed review on the literature regarding the problem is initially provided. Then, different mining algorithms are applied to both Single-Input Single-Output and Multiple-Input Multiple-Output systems operating in open-loop and in closed-loop. Simulated data is used to didactically exemplify how each method works and to validate the expected outcomes in an ideal scenario. Regressive models are then estimated with the obtained intervals, which are used to perform cross-validation. Finally, the proposed methods are applied to real multivariable data coming from an industrial petrochemical furnace.

Results obtained through simulated data show that the proposed data mining strategies allowed the estimation of good models in cross-validation scenarios with 1, 10, 100 and infinite prediction steps. Real data applications, in turn, revealed to be challenging due to the noisy nature of the data and due to the scarcity of historical intervals in which all the inputs of the multivariable system are sufficiently active to estimate a model. However, this problem is overcome through the use of multiple intervals in the estimation process, elucidating that the adopted algorithms can also produce reasonable models in real scenarios.

Keywords – Data Science; Historical Data; System Identification; Data Mining; Condition Number; Effective Rank; Data Quality; Data Segmentation; Multivariable Systems.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AR	Autoregressive
ARMAX	Autoregressive Moving Average with Exogenous Inputs
ARX	Autoregressive with Exogenous Inputs
CUSUM	Cumulative Sum
EWMA	Exponential Weighted Moving Average
FIR	Finite Impulse Response
IIoT	Industrial Internet of Things
MIMO	Multiple-Input Multiple-Output
MISO	Multiple-Input Single-Output
MPC	Model Predictive Controller
PID	Proportional Integral Derivative Controller
PIMS	Plant Information Management System
NaN	Not a Number
SISO	Single-Input Single-Output
SVD	Singular Value Decomposition

LIST OF SYMBOLS

C	Cumulative sum
$\boldsymbol{C}_j$	Condition number vector for a MISO system
$C(q)$	Controller transfer function
d	Degrees of freedom
D_t	Difference of two means statistics
$D(t)$	Relative position of a point in a data sample
$E[\cdot]$	Expected value
f_c	Cut-off frequency
$F(T)$	Cumulative distribution function of an exponential random variable
$G(q)$	Process model transfer function
$H[\cdot]$	Heaviside function
$\boldsymbol{H}$	Hessian matrix
$H(q)$	Disturbance model transfer function
H_t	Time-series entropy
I	Indicating sequence
$\boldsymbol{I}(\theta)$	Fisher information matrix
k	Discrete-time instant
l_1	Singular values tolerance for type 1 effective rank
l_2	Singular values tolerance for type 2 effective rank
l_{c_j}	Threshold for the condition number vector of a MIMO system
l_{cc}	Threshold for the cross-correlation scalar metric in the SVD segmentation method
l_e	Threshold for the deviation error in the band-pass filter detection method
l_{efr}	Threshold for the effective rank
l_{e_j}	Threshold for the effective rank vector of a MIMO system

$L_i(q, \alpha)$	Laguerre filter transfer function
l_S	Threshold for the variance in the EWMA detection method
l_κ	Threshold for the condition number
$m_v(k)$	Manipulated variable in a control system at instant k
n_b	Laguerre regressor order
n_{idx}	Number of initial indexes considered in a potential interval
n_u	Model structure input order
n_y	Model structure output order
n_θ	Dimension of the parameters vector
N	Length of an entire dataset
N_{max}	The maximum size an interval can assume in the non-parametric change-point method
N_s	Length of a data sample
$N_{s,min}$	The minimum required length to split a dataset in the non-parametric change-point method
N_w	Number of windows
N_Δ	Number of potential intervals
p	p-value
$\boldsymbol{P_{N_s}}$	Covariance matrix
$\boldsymbol{Q}$	Orthogonal matrix from QR decomposition
$\boldsymbol{R}$	Upper-triangular matrix from QR decomposition
r^{ef}	Effective rank
$\boldsymbol{R}_{ef_j}$	Effective rank vector of a MISO system
$r(k)$	Closed-loop control system set-point at instant k
$\boldsymbol{R_{N_s}}$	Information matrix
r_u	Sampling autocovariance function of signal u
s	Scalar cross-correlation metric
S	Sample standard deviation
$\boldsymbol{S}^j$	MISO regressor matrix
t	Time index

t_c	Mean-crossing time instant
T_c	Time interval between two consecutive mean-crossing points
$\boldsymbol{U}$	Orthogonal matrix from singular value decomposition
$u(k)$	System's input signal at instant k
$\boldsymbol{V}$	Orthogonal matrix from singular value decomposition
v	White noise disturbance following a Gaussian distribution
w_{ic}	Data window increment in a sliding-window algorithm
w_s	Sliding-window size
x	Random variable
$x(k)$	Data sample
$y(k)$	System's output signal at instant k
z	Transformed random variable
$\boldsymbol{\theta}$	Parameter vector
Δ	Potential interval for system identification
$\Delta H_{t,x}$	Differential entropy
κ_p	p-norm condition number
λ	Exponential forgetting factor
μ	Mean value of a population variable
$\rho(\tau)$	Normalized cross-correlation function
$\boldsymbol{\Sigma}$	Diagonal matrix from singular value decomposition
τ	Change-point location or a time delay
$\chi_{d,\alpha}$	Chi-squared critical value
$\boldsymbol{\Psi}$	Regressor matrix
$\boldsymbol{\psi}$	Regressor vector
ω	Angular frequency
$\boldsymbol{\nabla}$	Gradient vector

CONTENTS

1 INTRODUCTION

The ability to create industrial process models has an extremely relevant role for the industry. It is through precise dynamic models that it is possible to develop a whole bunch of activities widely desired to optimize industrial processes, such as: implementing advanced controllers like Model Predictive Controllers (MPC), designing optimum tuning of Proportional Integral Derivative (PID) controllers, developing training simulators, detecting possible system failures, checking for process quality and performing predictive maintenance (FASSOIS; RIVERA, 2007).

As described in (AGUIRRE, 2015), system identification is an engineering field that is concerned precisely with the development of mathematical modeling techniques to dynamic systems. Unlike other model estimation techniques, such as phenomenological modeling, system identification requires little prior knowledge about the system being studied. Given the great complexity of the majority of the systems found in industry, try to model them through its physical principles (or equivalently, its nature) becomes a long and expensive process (AGUIRRE, 2015). Thus, system identification emerges as an alternative to the estimation of industrial process models.

The most frequent way to carry out system identification is through the execution of physical experiments in the plant (process) being modeled. In order to estimate a good model, informative and representative data of the process must be available, which are usually obtained subjecting the process to excitation signals that cause a dynamic response of the process. The outcome data can then be measured through field sensors and stored by a data acquisition system, allowing one to obtain a model of the plant through appropriate data analysis.

The lack of informative datasets and the difficulty of obtaining such data in a real scenario is the downside of system identification. In the majority of the industrial plants, deviations from the operating point are not allowed or they are only permitted along narrow limits, which makes the data less active and less effective to fit models, requiring alternative solutions to be developed.

1.1 Motivation

Since the Third Industrial Revolution that begun at the second half of the 20th century, the development in electronics gave space to the emergence of a new valuable asset: the digital data. It is more and more frequent the development of new researches and technologies that aim to use data as a resource for decision making. Companies from the most diverse segments, such as Netflix, Uber, Airbnb, Google, among many others, have data analysis and artificial intelligence algorithms as central tools of their business. In the context of the manufactory industry, the process popularly known as Industry 4.0 (or fourth industrial revolution) has also gain attention, where technologies such as IIoT (Industrial Internet of Things), cloud, cybersecurity and Big Data are bringing a paradigm shift in the production processes (WANG; WANG, 2016).

As the concept of Big Data Analytics emerged in 1997, new ways to collect, store, investigate, gain insights and make predictions with massive data have come to be widely studied by researchers of the industry and of the academia (TIWARI; WEE; DARYANTO, 2018). Big Data Analytics can be understand as "the applications of advanced analytic techniques including data mining, statistical analytics, predictive analytics, etc. on big datasets" (TIWARI; WEE; DARYANTO, 2018). In this scenario, technologies such as data mining, machine learning and artificial intelligence begun to be widely explored, giving space to new areas such as Data Science to emerge with the specific goal of using these techniques to extract useful information from large datasets (QIN; CHIANG, 2019).

In the particular case of the manufacturing industry, data from multiple sensors are stored every minute and every day, producing a powerful background that can be used to aggregate value to such companies. An important area in the industry that is highly based on data is system identification, which is a data-driven set of mathematical techniques used to model a system dynamics. In order to obtain dynamic predictive models, meaningful and informative datasets are required, which are usually obtained through physical experiments, "shaking" the process variables and forcing them to manifest its underlining dynamic response. Evidently, these experiments become a problem to most companies that seek to have a model of their plants, since they require removing the process from its operating condition, in which products with strict specifications are being produced.

A natural solution to overcome this problem arises from the fact that huge amounts of historical data are available and easily accessible in many companies, allowing the search for dynamic responses that may have occurred at some point in the past. A manual search for informative data in huge databases is an exhaustive, inefficient and not

scientific based task. Thus, the use of mining techniques and data analysis, so popular today, become a great motivator to address the problem of obtaining meaningful data for system identification. In fact, the development of data-driven algorithms capable of enabling an automated system identification based on historical data has been the subject of many researches and it is a relevant topic in the scientific community in the last years, although several of these studies are spread in the literature and no work has yet aggregated them into a single study.

The development of soft sensors and model-plant mismatch applications are two examples of remarkable applications that could take advantage of such mining algorithms. To illustrate these points, it is interesting to mention the two-step data-driven soft sensor proposed in (TEIXEIRA *et al.*, 2014), where the first step of the algorithm consists of performing a system identification based on a historical selection of data recovered from a plant information management system (PIMS).

1.2 A literature Overview

There are few and very recent works in the literature that specifically dive into this subject. It is possible to mention, initially, a set of works that, although do not directly approach the problem, provide correlated tools that are essential for its development.

The work in (CARRETTE *et al.*, 1996), for example, presented results concerning parameter estimates obtained by Prediction Error Methods for input signals that are not sufficiently rich, *i.e.*, composed of few or no excitation elements. CARRETTE *et al.* also proposed a criterion for data selection that allows one to improve the accuracy of the estimated transfer functions. In the same direction, GEVERS; BAZANELLA; MISKOVIC studied the necessary and sufficient conditions of information's degree that an excitation signal must have to result in satisfactory experiments of systems identification based on Prediction Error Methods.

A combination of historical and testing data to obtain an inferential model for control purposes was proposed in (AMIRTHALINGAM; SUNG; LEE, 2000). In this work, strategies for data separation were suggested in order to obtain periods of time in which the plant is working in its operating point.

In (SHARDT; HUANG, 2011c), the effects of the sampling period in the quality of models obtained in closed-loop system identifications are analyzed. It is shown that if the sampling period is sufficiently small, it is possible to retrieve the plant's original

parameters. In (SHARDT; HUANG, 2010), it is also studied the necessary conditions for operation data to be identifiable for a first order ARX structure process governed by a lead-lag controller. Theoretical conditions for fully retrieving linear parametric structures from routine operating data, in open and closed-loop systems, were also described in (SHARDT; HUANG, 2011a), (SHARDT; HUANG, 2011b) and (SHARDT; HUANG, 2017).

Finally, in (PERETZKI *et al.*, 2011) it was explicitly addressed the problem of identifying systems based on time series stored in data historian systems. The authors developed an algorithm capable of providing intervals of interest for system identification, as well as an indicator of quality. This indicator was provided through an information matrix dependent on the model structure adopted for the plant and through its condition number, associating an acceptable threshold to it. As the method requires the existence of a prior model structure, the authors proposed the use of the Laguerre Filter because of its flexible structure to explain input and output relationships for a wide variety of processes. In addition, the work hypothesized that the systems studied are linear and Single-Input Single-Output (SISO). Moreover, both open-loop and closed-loop systems were analyzed with and without integrative action. Finally, it is worth mentioning that a more detailed version of this work was published in (BITTENCOURT *et al.*, 2015).

The work in (PERETZKI *et al.*, 2011), however, searched for sufficient exciting changes in the set-point of the controller, which is unlikely to happen in routine operating data. For this reason, in (SHARDT; HUANG, 2013a) it was proposed the use of the output of the controller, assuming that the set-point will be kept constant. The authors used an ARX structure for this purpose and addressed the theoretical conditions for obtaining the exact model parameters from historical data. The theoretical limitations included the need to know the process order and time delay. A similar study was done in (BITTENCOURT *et al.*, 2015).

Statistical properties of the discrete-time signal entropy were studied in (SHARDT; HUANG, 2013b), which proposed a change detection index to perform the segmentation of time series. The authors classified the segmentation methods into online (when the segmentation is done simultaneously with the data collection) and offline (when it is performed with data already collected). In addition, segmentation was divided into three groups:

a) **Sliding Windows (or Moving Windows) methods**: the data is scanned by time windows and the segments are incremented until they reach a certain stop

condition;

b) **Top-Down methods**: the analysis begins with the entire dataset, which is further divided until a stop criterion is reached;

c) **Bottom-Up methods**: the entire dataset is divided into small segments, which are unified until they reach a stopping criterion.

One result of the work in (SHARDT; HUANG, 2013b) was to use a differential entropy between the input and the output signals in order to find similar inputs. Therefore, assuming that different models are obtained through historical data and that these models can represent different process conditions in time, the differential entropy could be used to determine models that are no longer representative of the real system. In the same fashion, this method was used in (SHARDT; SHAH, 2014) as an additional step of the methods in (PERETZKI *et al.*, 2011) and in (SHARDT; HUANG, 2013a). The objective of the included step is to verify if consecutive intervals have similar entropy, in such a way that intervals with similar differential entropy can be unified as a single window.

In (SHARDT; SHAH, 2014) it was also studied, based on the work in (PERETZKI *et al.*, 2011), which parameters of the Laguerre Filter mostly affect the data segmentation process. The following classification of the segmentation problem was presented:

a) **over segmentation**: when data is overly partitioned, resulting in many identified models;

b) **under segmentation**: when few models are identified;

c) **exact case**: when the correct number of models is identified.

To handle the problem of excessive segmentation, the authors used the differential entropy to find similar models and to unify them, reducing the number of resulting models.

In (RIBEIRO; AGUIRRE, 2015), a method based on the Autoregressive (AR) structure was proposed using routine operating data. The authors used the AR structure because it is only dependent on the output signal, which is easily obtained from a historian system. An information metric is proposed based on the singular values of the regressor matrix, which are used to calculate the effective rank. In fact, two different computations of the effective rank were proposed. Correlation between the input and the output signals are verified as a requirement step of the algorithm and are calculated through a

cross-correlation scalar metric proposed by RIBEIRO; AGUIRRE. The dataset is equally divided into a user-defined number of windows and the algorithm is then executed in a sliding-window fashion.

In (PATEL, 2016), an extension of (PERETZKI *et al.*, 2011) and (BITTENCOURT *et al.*, 2015) was proposed to include Multiple-Input Multiple-Output (MIMO) systems in the analysis. PATEL assumed the following hypothesis: the data comes exclusively from an open-loop processes; the process studied is linear and time invariant. In addition, it was proposed a cycle of data processing, which involves:

a) standardization and centralization of data;

b) removal of the first element of the sample in order to avoid deflection of the calculated variance;

c) data filtering (low pass filter) to reduce disturbance influence;

d) assymetric filtering for unification of intervals.

The work also proposed the combination of the Laguerre Filter structure with the ARX one for the composition of the model. The idea was to explore the flexibility characteristics of the first with the disturbance model of the second. The final solution, however, resulted in a series of parametric limitations, such as the choice of the model order, the choice of the Laguerre filter poles and the choice of the cut-off frequencies of the lowpass and bandpass filters, in such a way that PATEL developed a Graphical User Interface (GUI) to make the solution more manageable.

Finally, it is worth mentioning that, in order to address the multivariable problem, a concomitant treatment of excitation signals and output signals is presented in (PATEL, 2016), in such a fashion that if a corrupted (or bad) data range is found in one of the inputs, the corresponding interval will be discarded for all the other inputs and for the output (considering that the MIMO problem can be treated as a combination of multiples Multiple-Input Single-Output – MISO – systems).

In (ARENGAS; KROLL, 2017a) it was proposed the development of an algorithm to select informative intervals of data for MIMO closed-loop systems. In this work, the adopted model is considered to have an ARX structure. Like in (PATEL, 2016), Arengas and Kroll suggested to simultaneously treat the excitation and the output signals. In a simplified manner, the algorithm proposed by the authors can be divided into three levels:

a) **Level 1**: check for transient changes in the reference signal;

b) **Level 2**: check for transient changes in the output signal;

c) **Level 3**: calculate the information matrix and condition number of the closed-loop system and compare it to a given threshold.

In (ARENGAS; KROLL, 2017b), an open-loop based search method was proposed using routine operating data. The method is applied to SISO systems and is very similar to the ones already described. The main contribution of this work is regarding the way the sliding window is designed. ARENGAS; KROLL proposed a configurable sliding window that gives more flexibility to the process of searching transient data points.

In (WANG *et al.*, 2018), a new method to search for data segments suitable for system identifications was presented. In this work, the authors applied a top-down approach to detect change-points in the data. The change-point detection method is based on a non-parametric top-down algorithm formulated by (PETTITT, 1979). The main algorithm formulated in (WANG *et al.*, 2018) is applied to closed-loop systems and consists of mainly four steps:

a) **Step 1**: finding the change-points;

b) **Step 2**: verifying if the data inside each interval suffers a significant change;

c) **Step 3**: verifying if there is a significant value difference between intervals that do not suffer from significant changes;

d) **Step 4**: determining the start and ending time indices for the final data segments.

Finally, in (SHARDT; BROOKS, 2018) the challenging problem of searching intervals for MIMO systems in closed-loop mode and using operating data was addressed. The authors used the method proposed in (PERETZKI *et al.*, 2011) and (BITTENCOURT *et al.*, 2015) as the baseline of the solution, considering the additional step described in (SHARDT; SHAH, 2014). The algorithm is then applied to a zink floating cell controlled by several PIDs. Different combinations of input signals were tried in order to find a set of variables that can satisfactorily segment the data in useful intervals for obtaining a "seed" matrix for MPC design.

1.3 Goals and Objectives

The ultimate goal of this dissertation is to address the problem of lack of informative data for system identification, *i.e.*, it is intended to replace the need to carry out system identification experiments, as well as the manual and visual search of data in historian systems, with data mining techniques capable of finding suitable data for system identification. It is expected that, at the end of this dissertation, the reader encounter a solid knowledge about the state of the art of how to solve this problem, as well as that there is available a set of techniques and tools to deal with this problem.

In a general manner, the purpose of this dissertation is the development of algorithms that are capable to find informative data that can be used to obtain dynamic models of industrial processes in the following scenarios:

a) **Open-loop and closed-loop SISO systems:** most industries have many control system loops with a single input and a single output variables. In fact, most of these systems operate in closed loop and, sometimes, the design of new controllers or even the tuning of already existing controllers are desired, which usually requires dynamic models of the system. Therefore, it is intended in this dissertation the development of historical data mining techniques that are able to detect informative pieces of data capable of retrieve a model of the system being studied for SISO systems in both open and closed loop scenarios;

b) **Open-loop and closed-loop MIMO systems:** the majority of industrial systems have many input and output variables. For this reason, it is also an objective of this dissertation to address the problem in its multivariable version. In particular, the problem is considered for open-loop identification and, in the case of closed-loop systems, the problem is considered through the optics of obtaining a model using the set-point and the output variable, which can be treated equally to an open-loop identification problem. In fact, models obtained as in the last case are widely used to design model predictive controllers and, therefore, this approach is of great practical interest.

The specific goals to be achieved in this dissertation are those listed below:

a) provide a detailed review of the state of the art that demonstrates a solid understanding of the problem and the solutions presented in the literature;

b) develop algorithms capable of scanning long time series of historical data and returning informative data intervals that are useful to design industrial process models, considering both the SISO and the MIMO described scenarios;

c) provide solutions that require minimal (or none) knowledge about the process studied in the application of the mining algorithms;

d) develop a solution structure that allows the engineer to use data analysis tools when choosing parameters, providing conscious decision making in the data mining process;

e) apply the algorithms developed to both simulation and real data. In the former case, the objective is to validate the algorithms and understand their behavior in controlled scenarios, allowing one to compare the results with the expected outcomes. In the later case, the objective is the real application of the algorithm and the evaluation of the obtained results;

f) develop the algorithms in the form of an open-source library, allowing the reproducibility of the resulting outcomes, the visualization of the adopted implementations and future scientific contributions based on the work developed throughout in this dissertation. The language adopted for this purpose is Python, due to its wide popularity both in the scientific and in the market spheres.

1.4 Organization of the Dissertation

This dissertation is organized as follows: Chapter 2 explores preliminary concepts that are crucial to the understanding of the consecutive subjects. More specifically, the chapter briefly dives into necessary topics of system identification that are explored along this work. Chapter 3 goes deeply in the state of the art to find intervals suitable for system identification from historical data. All the background theory adopted in this dissertation is detailed explained in this chapter, constituting the foundation of the algorithms presented in the methodology. Chapter 4 explains the methodology that is used in this work to implement and apply the proposed algorithms. More specifically, the algorithms outlines are given and hypothesis are formulated. Chapter 5 presents the work development in both the simulations and in the real world scenarios; finally, conclusions are drawn in Chapter 6 and suggestions on future works are provided in Section 6.1.

2 PRELIMINARY CONCEPTS

2.1 System Identification

System Identification is a set of methods used to mathematically model a system dynamics. Usually, system identification requires little or no previous knowledge about the dynamics of the plant being studied, once its methods are based on measured data and not on the system's physical principles. A general formulation of the problem is proposed in (AGUIRRE, 2015, p. 33) and it is reproduced in Definition 2.1.

Definition 2.1. *Consider $\mathcal{S}$ the system being modeled and suppose we have available dynamic data $\boldsymbol{Z}^{N_s}$, with N_s being the length of the stored data sample. In this dissertation, $\boldsymbol{Z}$ can be any input/output data that was recorded by a historian system and that can describe the dynamics of the plant being studied. The problem of black-box System Identification is to find a model $\mathcal{M}$ exclusively from $\boldsymbol{Z}^{N_s}$, in such a way that $\mathcal{M}$ can adequately describe dynamic properties of $\mathcal{S}$.*

As also described in (AGUIRRE, 2015, p. 79), the main steps in a System Identification process consist of:

a) **performing a physical excitation test in the plant and collecting data**: in this dissertation, it is being considered that historical data is available, and intervals of excitation will be searched through the proposed algorithms, in order to find data with enough information to allow parameter estimation;

b) **choosing a mathematical representation**: one could choose, for example, between a linear and a non-linear representation of the system, or between a state-space or a transfer function representation. Furthermore, the representation can be either continuous or discrete. Also, parametric and non-parametric models can be adopted. In this dissertation, only the linear parametric representation is studied in a discrete-time fashion;

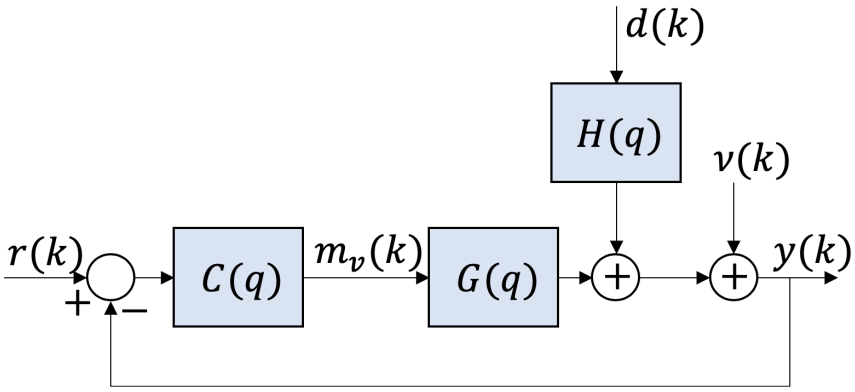
c) **determining a model structure**: since this work is concerned with the linear case, determining a model structure is equivalent to choose the number of poles and zeros of the system, as well as its dead time. A few examples of model structures are the Finite Impulse Response (FIR), the Autoregressive (AR), the Autoregressive with Exogeneous Inputs (ARX), the Autoregressive Moving Average with Exogenous Inputs (ARMAX) and the Laguerre Filter. A lot of different model structures are available for the linear representation. In this dissertation, only the ARX structure, the AR structure and the Laguerre Filter are studied;

d) **model parameter estimation**: several algorithms can be used to estimate the model parameters. In this dissertation, only the Ordinary Least Square Method (OLS) will be considered;

e) **model validation**: different validation metrics can be used to evaluate the model performance and if it is adequate for the purpose it is being used.

It is important to point out that data $\boldsymbol{Z}$ can be obtained through a system in either open-loop or closed-loop operation. Moreover, in a closed-loop system, different signals can be considered as the input $u(k)$ for the system identification, as is explained in Section 2.7. The diagram in Figure 1 summarizes this scenario as explained in (GARCIA, 2017).

Figure 1: Closed Loop Feedback Control with Disturbances.



Source: Adapted from (WANG *et al.*, 2018).

Notice that $r(k)$ is the control set-point (also known as the reference signal), $m_v(k)$ is the manipulated variable, which is equivalent to the controller output, $y(k)$ is the controlled variable (which is usually equivalent to the system output), $d(k)$ is a disturbance signal and $v(k)$ is a gaussian noise. In this case, if the System Identification is performed between $m_v(k)$ and $y(k)$, only the process dynamics $G(q)$ will be modeled. Equivalently, if the modeling is performed using $r(k)$ and $y(k)$, the controller dynamic $C(q)$ will also be

included in the obtained dynamic model. As is further described, $H(q)$ is the disturbance model.

2.2 ARX Structure

The linear ARX structure is used in the algorithms proposed in this dissertation and, therefore, it is briefly explained in this section. A general linear parametric discrete-time model structure can be defined as (AGUIRRE, 2015, p. 122)

$$y(k) = \frac{B(q)}{A(q)F(q)}u(k) + \frac{C(q)}{A(q)D(q)}v(k), \tag{2.1}$$

where q^{-1} is the delay operator, which means that $y(k)q^{-1} = y(k-1)$, with k being the discrete-time instant. Notice that the general structure can be reduced to the form $y(k) = G(q)u(k)+H(q)v(k)$, with $G(q)$ representing the process model, $H(q)$ representing the disturbance model, $v(k)$ being a white noise following a gaussian distribution with 0 mean and variance σ^2 and $u(k)$ being the input variable being modeled. The polynomials $A(q)$, $B(q)$, $C(q)$, $D(q)$ and $F(q)$ are defined as follows (AGUIRRE, 2015, p. 122):

$$A(q) = 1 - a_1q^{-1} - \cdots - a_{n_y}q^{-n_y} \tag{2.2}$$

$$B(q) = b_1q^{-1} + \cdots + b_{n_u}q^{-n_u} \tag{2.3}$$

$$C(q) = 1 + c_1q^{-1} + \cdots + c_{n_v}q^{-n_v} \tag{2.4}$$

$$D(q) = 1 + d_1q^{-1} + \cdots + d_{n_d}q^{-n_d} \tag{2.5}$$

$$F(q) = 1 + f_1q^{-1} + \cdots + f_{n_f}q^{-n_f}. \tag{2.6}$$

The ARX model can be obtained directly from the general structure above making $C(q) = D(q) = F(q) = 1$, resulting in the structure (AGUIRRE, 2015, p. 124).

$$y(k) = \frac{B(q)}{A(q)}u(k) + \frac{1}{A(q)}v(k). \tag{2.7}$$

It is important to notice that this representation can be written in the regression form

$$y(k) = \boldsymbol{\psi}_{yu}^T(k-1)\hat{\boldsymbol{\theta}} + \xi(k), \tag{2.8}$$

with $\boldsymbol{\psi}_{yu}(k-1) = [\psi_1 \;\; \psi_2 \;\; \cdots \;\; \psi_{n_\theta}]^T$ being the regressor vector and $\xi(k)$ being the regression error, *i.e.*, $\xi(k) = y(k) - \boldsymbol{\psi}_{yu}^T(k-1)\hat{\boldsymbol{\theta}}$ (AGUIRRE, 2015, p. 239). In the case of the ARX model structure, the regressor vector can be written as

$$\boldsymbol{\psi}_{yu}^T(k-1) = \begin{bmatrix} y(k-1) & \cdots & y(n-n_y) & u(k-1) & \cdots & u(k-n_u) \end{bmatrix} \tag{2.9}$$

and the parameter vector as

$$\hat{\boldsymbol{\theta}} = \begin{bmatrix} \hat{a}_1 & \cdots & \hat{a}_{n_y} & \hat{b}_1 & \cdots & \hat{b}_{n_u} \end{bmatrix}. \tag{2.10}$$

2.3 AR Structure

In the same way it was done for the ARX structure, the Autoregressive (AR) structure can be directly obtained from the generic structure in Equation (2.1). The AR model constitutes the case where there is no input in the regressor matrix, which can be obtained setting $B(q) = 0$ in the ARX structure of Equation (2.7). The AR structure is then defined as

$$A(q)y(k) = v(k). \tag{2.11}$$

Notice that, in this case, the order of the AR structure is defined by the order of $A(q)$, which is n_y (the output order).

2.4 Laguerre Filter

The Laguerre filter is proposed by (PERETZKI *et al.*, 2011) and (BITTENCOURT *et al.*, 2015) as an alternative to the ARX structure. In the work of (PATEL, 2016), this filter is combined with the ARX structure.

The use of Laguerre models for System Identification is proposed in (WAHLBERG, 1991) in such a way that the Laguerre expansions are used to produce the following model structure:

$$y(k) = \sum_{i=1}^{n_b} \bar{g}_i L_i(q, \alpha) u(k), \tag{2.12}$$

where $L_i(q, \alpha) = \frac{\sqrt{T_s(1-\alpha^2)}}{q-\alpha} \left(\frac{1-\alpha q}{q-\alpha}\right)^{i-1}$ is the Laguerre Filter, with T_s being the sampling time, α the Laguerre filter real pole and $\bar{g}_i$ the regressor parameters. In the same way as for the ARX, a regressor and a parameter vector can be written as (PATEL, 2016, p. 19)

$$\boldsymbol{\psi}_{yu}^T(k) = \begin{bmatrix} L_1(q, \alpha)u(k) & \cdots & L_{n_b}(q, \alpha)u(k) \end{bmatrix} \tag{2.13}$$

$$\hat{\boldsymbol{\theta}} = \begin{bmatrix} \hat{\bar{g}}_1 \cdots \hat{\bar{g}}_{n_b} \end{bmatrix}^T, \tag{2.14}$$

where n_b is the parameter vector order.

It is interesting to mention that the Laguerre Structure is implicitly capable of estimating the time delay of the system. As explained in (PERETZKI *et al.*, 2011), the

maximum delay $\bar{d}$ that can be incorporated in this model structure is $\bar{d} = \frac{-2(n_b-1)T_s}{log(\alpha)}$. Moreover, for processes with integrators, the integrated input $\bar{u}(k)$ can be considered instead of $u(k)$, as follows:

$$\bar{u}(k) = \frac{u(k)}{1-q^{-1}} \tag{2.15}$$

$$y(k) = \sum_{i=1}^{n_b} \bar{g}_i L_i(q,\alpha)\bar{u}(k). \tag{2.16}$$

2.5 Combination of the Laguerre Filter and the AR Structure

A combination of the Laguerre model with the AR structure is proposed in (BITTENCOURT *et al.*, 2015) and (PATEL, 2016) in order to include a noise model in the parametric structure, which therefore results in an ARX structure. That is done considering the structure

$$y(k) = \sum_{i=1}^{n_a} a_i y(k-i) + \sum_{i=1}^{n_b} \bar{g}_i L_i(q,\alpha)u(k). \tag{2.17}$$

The regressor and parameter vectors can now be rewritten as

$$\boldsymbol{\psi}_{yu}^T(k-1) = \begin{bmatrix} L_1(q,\alpha)u(k) & \cdots & L_{n_b}(q,\alpha)u(k) & y(k-1) & \cdots & y(k-n_a) \end{bmatrix} \tag{2.18}$$

$$\hat{\boldsymbol{\theta}} = \begin{bmatrix} \hat{\bar{g}}_1 \cdots \hat{\bar{g}}_{n_b} & a_1 & \cdots & a_{n_a} \end{bmatrix}^T. \tag{2.19}$$

2.6 The Regressor Matrix

It is important to mention that the regressor general structure described in Equation (2.1) is defined for a discrete time instant k. That means that in a given instant of time, the regressor $\boldsymbol{\psi}_{yu}^T(k)$ will be an array $\boldsymbol{\psi}_{yu}^T(k) \in \mathbb{R}^{n_\theta}$ defined as

$$\boldsymbol{\psi}_{yu}^T(k) = \begin{bmatrix} \psi_1(k) & \psi_2(k) & \cdots & \psi_{n_\theta}(k) \end{bmatrix}. \tag{2.20}$$

If one considers a sample window of length N_s, then a regressor matrix can be defined as

$$\boldsymbol{\Psi} = \begin{bmatrix} \psi_1(k) & \psi_2(k) & \cdots & \psi_{n_\theta}(k) \\ \vdots & \vdots & \ddots & \vdots \\ \psi_1(k+N_s-1) & \psi_2(k+N_s-1) & \cdots & \psi_{n_\theta}(k+N_s-1) \end{bmatrix}. \tag{2.21}$$

In this case, the closed form solution in Equation (2.22) can be applied to obtain an estimation $\hat{\boldsymbol{\theta}}$ of the parameters (AGUIRRE, 2015, p. 227):

$$\hat{\boldsymbol{\theta}} = [\boldsymbol{\Psi}^T\boldsymbol{\Psi}]^{-1}\boldsymbol{\Psi}^T\boldsymbol{y}. \tag{2.22}$$

2.7 Open-loop and Closed-loop System Identification

In this Section, a brief explanation is given on how the terms open-loop and closed-loop are used in this dissertation. To start, it is important to mention that, in this work, the concepts of closed-loop and open-loop **systems** are treated differently than the concepts of closed-loop and open-loop **system identification**.

A closed-loop **system** is defined in this dissertation as a system that contains a feedback control loop, such as the one illustrated in Figure 1. For such a system, the process identification could be done either in a closed-loop or in an open-loop perspective, as explained below:

a) **Open-loop Identification of Closed-loop Systems:** consists of obtaining a model of the closed-loop system defined from the set-point $r(k)$ and the output $y(k)$.

b) **Closed-loop Identification of Closed-loop Systems:** consists of obtaining a model of the process $G(q)$.

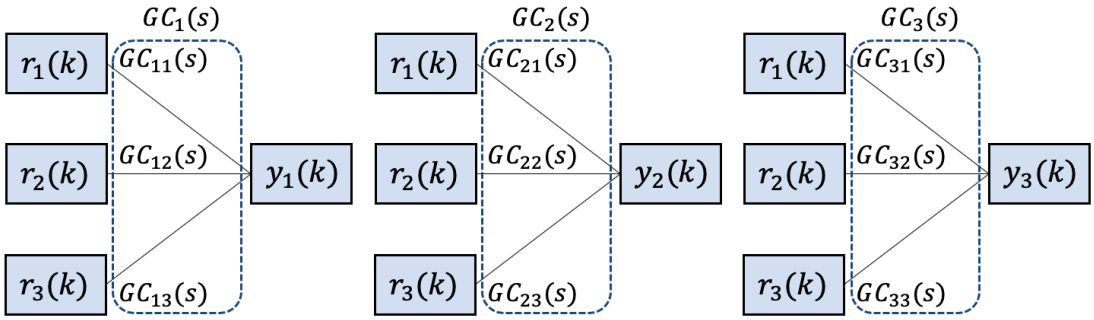
Notice that, in the later case, a model of the process is desired, which means that one is interested in estimating $G(q)$. This is actually a well-studied problem in the literature and commonly called "Closed-loop Identification". Different identification approaches can be used in this scenario, a few of them described in (LJUNG, 1999, p. 435), such as the "Direct Approach" and the "Indirect Approach".

The former case, on the other hand, consists of estimating a model from variables $r(k)$ and $y(k)$, which therefore will result in a combined model of $C(q)$ and $G(q)$, here called $GC(q) = \frac{C(q)G(q)}{1+C(q)G(q)}$. This makes the problem easier in the sense that the input variable $r(k)$ is manipulated by the user, and not by a controller. Therefore, in a multivariable system, for instance, these variables do not affect each other and neither are affected by the output variable, as would happen in a closed-loop identification. Finally, this is actually a very useful model for designing model predictive controllers, as proposed in (CHAVES; JULIANI; GARCIA, 2019).

2.8 Multiple-Input Multiple-Output (MIMO) Systems

The majority of the industrial systems has many inputs and many outputs. In this dissertation, the mining algorithms for multivariable systems presented in (PATEL, 2016) and (ARENGAS; KROLL, 2017a) are studied. In the work of (PATEL, 2016), the multivariable extension can only be applied to open-loop system identification and the analysis is considered splitting the MIMO system into multiple Multiple-Input Single-Output (MISO) systems. For a 3 X 3 MIMO system as the Petrochemical Furnace being considered in this work and described in Subsection 5.1.3, a MISO closed-loop system can be represented, considering an open-loop system identification, as in Figure 2.

Figure 2: 3 X 3 MIMO closed-loop system decomposed in three 3 X 1 MISO systems for open-loop system identification.



Source: Adapted from (PATEL, 2016).

Based on the definitions made so far, the main reviewed works used in this dissertation can be grouped as in Table 1.

Table 1: Grouping of the main reviewed works according to the system identification type and to the system number of input/output variables.

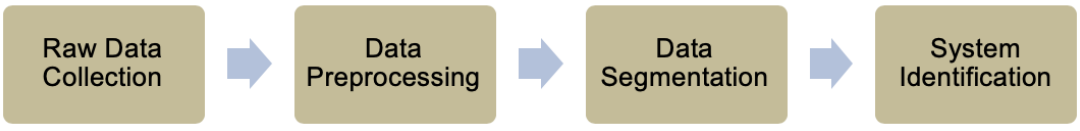
	Open-loop System Identification	**Closed-loop System Identification**
SISO Systems	PERETZKI *et al.*, 2011 SHARDT; SHAH, 2014 BITTENCOURT *et al.*, 2015 RIBEIRO; AGUIRRE, 2015 ARENGAS; KROLL, 2017b WANG *et al.*, 2018	PERETZKI *et al.*, 2011 SHARDT; SHAH, 2014 BITTENCOURT *et al.*, 2015 RIBEIRO; AGUIRRE, 2015 WANG *et al.*, 2018
MIMO Systems	PATEL, 2016	ARENGAS; KROLL, 2017a

3 SYSTEM IDENTIFICATION WITH HISTORICAL DATA: THE STATE OF THE ART

Data Mining can be defined as "the study of collecting, cleaning, processing, analyzing, and gaining useful insights from data" (AGGARWAL, 2015). In this dissertation, the data represents a collection of sensor signals from an industrial process that could have been stored for several years. The objective of the data mining process here studied is to find informative intervals of data through which meaningful models of a system can be obtained.

As an example of possible applications of the resulting models, one could use them "for enhancing physical understanding; analyzing system properties; and performing simulation, prediction, filtering, state estimation, monitoring, and fault diagnosis as well as control" (FASSOIS; RIVERA, 2007). Further applications involves the design of virtual sensors and also the development of more sophisticated system identification experiments based on these previous obtained models. The diagram in Figure 3 summarizes the approach that is followed throughout this dissertation.

Figure 3: Data Mining Flow Chart.



Source: Author's own development.

3.1 Data Preprocessing

Data preprocessing is an essential step in any data-driven mathematical modeling, being mentioned, in the context of this dissertation, in (PERETZKI *et al.*, 2011), (SHARDT; SHAH, 2014), (BITTENCOURT *et al.*, 2015) and (PATEL, 2016). As explained in (FACELI *et al.*, 2017), different characteristics can directly affect data quality, such as:

the presence of outliers, that is, anomalous values that significantly deviates from their expected values; discrepancies in formats and dimensions, *i.e.*, some values can be numeric while others can be in text format, for example; the presence of disturbances; the existence of missing or corrupted values due to sensor failures, for example, amongst others.

In this section, some data processing techniques that can be useful before estimating models or applying data mining techniques are briefly discussed, most of them being also described in (PATEL, 2016).

3.1.1 Data Resampling

The data being considered in this work is related to industrial process variables, which are usually obtained from sensors. Therefore, these variables correspond to time-series signals. Most of the time, sensor data are not collected uniformly, *i.e.*, the entire dataset can potentially contain different sampling periods. This is because most data acquisition systems are configured in such a way that a sampling point is registered every time a variable exceeds user-defined limits, which could happen at any moment in time. For this reason, it is interesting to resample the entire data with the desired sampling period T_s.

There are several techniques for resampling signals, which are not explored in details in this work. One way to resample time-series data is by first performing an up-sampling process, which consists in inserting zeros between each sample in the new frequency, then applying a low-pass filter and, finally, performing a down-sampling decimation, which means resampling the signal to a wider time frame. Notice that the process of applying an up-sampling followed by a low-pass filter is a way of performing an interpolation. This kind o resampling can be done using MATLAB *resample* or Scipy *upfirdn* functions, for example.

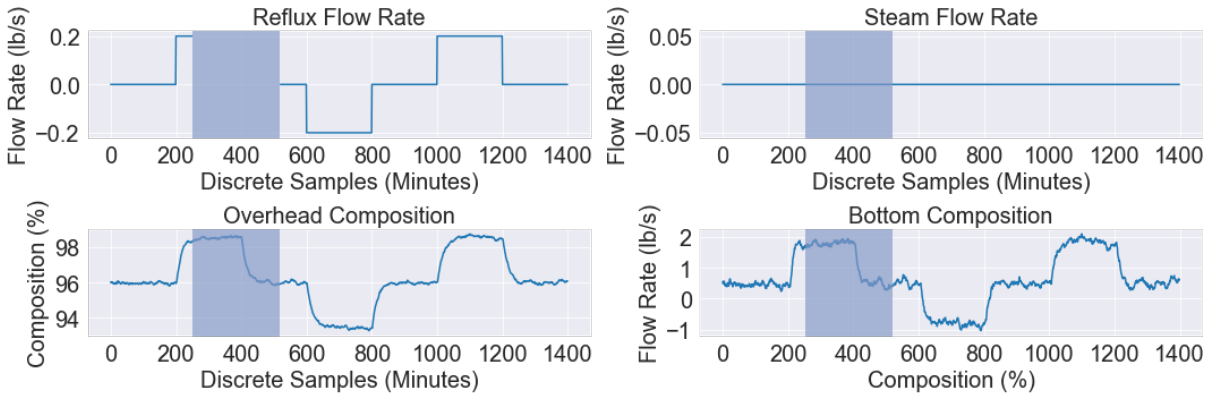
3.1.2 Treating Missing Values

Another common data issue is missing values, which can occur, for example, due to sensor failures. Missing values (or NaN – Not a Number – values) cannot be considered in the analysis and must be somehow treated. One way of dealing with NaN values is by simply disregarding them from the analysis. As mentioned in (PATEL, 2016), if a sequence of missing values exists for at least one input, it must be disregarded for all variables of the system.

Let us consider, for example, the Wood & Berry distillation column, described in

details in Subsection 5.1.2, which is constituted by 2 inputs and 2 outputs, all of them correlated. If we had a sequence of missing values from sample 200 to sample 520 in the Reflux Flow Rate variable, for instance, one would need to remove these samples from all four variables, even if all other three were not corrupted. Figure 4, created based on (PATEL, 2016), exemplifies this scenario for the Wood & Berry distillation data.

Figure 4: Example of Missing Values Removal.



Source: Author's own development.

3.1.3 Normalization

Data normalization is the process of adjusting the signal values of a system to a common scale. In system identification, the scales usually adopted are between the ranges $[-1, 1]$ or $[-0.5, 0.5]$. This is a crucial preprocessing technique when one is dealing with signals containing completely different upper and lower limits, leading to drastic variabilities that can produce numerical errors (FACELI *et al.*, 2017, p. 45). Moreover, it is very common – specially in closed-loop systems – that each variable of the process is centered at different operating points, making it hard to perform a simultaneous analysis of the signals being considered. For instance, in the distillation column signals shown in Figure 4, the overhead composition variable is centered around 96, while the bottom composition variable is centered around 0.0285.

As described in (FACELI *et al.*, 2017), normalization must be applied to each signal individually and can be done through an amplitude or through a distribution method. While the amplitude normalization only changes the variables maximum and minimum limits, the distribution normalization changes the variables distribution (FACELI *et al.*, 2017). In this work, only the amplitude normalization is considered. There are two common methods for performing amplitude normalization: the so-called min-max scaler and the standard scaler.

3.1.3.1 Min-max Scaler

In this method, as described in (FACELI *et al.*, 2017), a minimum (min) and a maximum (max) values of a desired scale must be defined. These values could be chosen to be, respectively, -0.5 and 0.5, for example, to keep the signal in the range $[-0.5,\ +0.5]$. Once these values are defined, the scaled data can be obtained through the following transformation (FACELI *et al.*, 2017, p. 45):

$$z = min + \frac{x - lowest}{highest - lowest}(max - min), \tag{3.1}$$

where x is a value from a given signal in the dataset, $lowest$ is the lowest value of this signal in the dataset, $highest$ is the highest value of this signal in the data and z is the corresponding transformed variable.

3.1.3.2 Standard Scaler

The standard scaler transforms features by subtracting the mean (μ) of the signal and scaling it by the unit variance of the sample (S^2), as shown below (FACELI *et al.*, 2017, p. 45):

$$z = \frac{x - \mu}{S^2}. \tag{3.2}$$

It is interesting to mention that the sample variance can be calculated as

$$S^2 = \frac{1}{N_s - 1}\sum_{k=1}^{N_s}(x(k) - \bar{x})^2. \tag{3.3}$$

3.1.4 Filtering Noise

In order to reduce the amount of noise that can be associated with measured variables coming from industrial sensors, a low-pass filter, such as a first-order Butterworth filter, can be applied when necessary to reduce high-frequency components. Implementations of this filter are available in MATLAB as well as in Python. When the low-pass filter is used, it must be applied to every input and output signals in the system. The reason for this is that, as explained in (LJUNG, 1999, p. 466), when prefiltering is done in all variables, it does not change the input-output relation of linenar systems, as shown below (LJUNG, 1999, p. 466):

$$y(k) = G(q)u(k) + H(q)v(k) \therefore L(q)y(k) = L(q)G(q)u(k) + L(q)H(q)v(k) \tag{3.4}$$

3.2 Detecting Potential Intervals

In this dissertation, the solution to the problem of finding intervals suitable for system identification is broken in 4 steps, as described in the adopted methodology in Chapter 4. The first step of the solution consists in detecting intervals that could potentially lead to a model of the system. This is a very important step in the solution because it will be these same intervals that will compose the final segments, if they prove to be adequate according to subsequent evaluations. Moreover, all the reviewed works used in this dissertation consider, somehow, the problem of detecting a transient change in the signal as the first step in finding intervals suitable for system identification. Therefore, the main techniques used for this purpose are explained in more details in this section.

A closely related problem to data segmentation is the so-called change-point detection problem, which is used in one of the solutions presented in this dissertation (see Subsection 3.3.3). As described in (BODENHAM, 2014, p. 18), change-point algorithms are concerned with the problem of detecting changes in the probability distribution of a sequence of random observations. A formulation of the problem is given in (KILLICK; FEARNHEAD; ECKLEY, 2012) and can be summarized as in Definition 3.1.

Definition 3.1. *(Adapted from KILLICK; FEARNHEAD; ECKLEY, 2012) Let us assume that we have a time-series signal of length N, with values represented as $x = (x_1, x_2, ..., x_N)$. Change-points can then be defined as indexes that split the entire data into $N_\tau + 1$ data segments, where the $i-th$ segment can be represented by $x(\tau_i : \tau_{i+1} - 1)$. In other words, a change-point index is an instant of time between 1 and $N - 1$ where a significant change in the data occurred. Assuming that we have N_τ change-positions τ, we can represent an array of change-points as $\boldsymbol{T} = [\tau_1, \tau_2, ..., \tau_{N_\tau}]$.*

In this chapter, some techniques related to data segmentation and change-point detection are presented as a first step to find useful data intervals for system identification. Most of the techniques here presented are extracted from the main reviewed works in Section 1.2, a few being explained in more detail as they are essential to understand the outline of the algorithms adopted in Chapter 4.

3.2.1 Control Charts

As described in (BODENHAM, 2014, p. 21), the original idea of control charts was to monitor manufacturing processes and ensure that certain variables were within acceptable limits. More specifically, if we define control limits a and b, with $a < b$, one could then say

that the process is under control if $x_i \in (a, b)$ and out of control otherwise (BODENHAM, 2014). The idea of moving average filter appears in most of the reviewed works in this dissertation and, therefore, are here explained in the context of control charts.

3.2.1.1 Moving Average Filters

Sliding Window

One way of computing a moving average filter is through a sliding window approach. In this case, one would obtain a simple Low Pass FIR (Finite Impulse Response) filter, wich can be described as below:

$$\mu_x(k) = \frac{1}{W} \sum_{i=0}^{W-1} x(k+i), \tag{3.5}$$

where μ_x is the output filtered signal, x is the input signal that is being treated and W is the window length (SMITH, 1999). As exemplified in (SMITH, 1999, p. 277), if one considers a window of length 5, point 80 in the output would be given by:

$$\mu_x(80) = \frac{(x(80) + x(81) + x(82) + x(83) + x(84))}{5}. \tag{3.6}$$

Moreover, the input points could be chosen symmetrically around the output points, as shown below:

$$\mu_x(80) = \frac{(x(78) + x(79) + x(80) + x(81) + x(82))}{5}. \tag{3.7}$$

In the latter case, the summation in Equation (3.5) would go from $i = -\frac{(W-1)}{2}$ to $i = \frac{(W-1)}{2}$. This strategy is used to identify intervals of excitation in (ARENGAS; KROLL, 2017a) and in (ARENGAS; KROLL, 2017b). The way this filter can be used to find moments when the signal begins to "shake" is comparing the filter output with a user-defined threshold l_μ, such that the points where $\mu_x(k) > l_\mu$ are marked as potentially exciting. Notice that the same procedure can be done with the signal **variance** instead of its mean and, in the same way, the filtered signal can be compared to a variance threshold l_S.

As a final observation, it is important to mention that the computation in Equation (3.7) is the one adopted in this dissertation. Notice that, in this case, one would need to start the computation in index $(W + 1)/2$, once the window depends on past values. Because we would like to maintain the original number of data points of the original signal, in this dissertation the window size is adjusted to the number of available data points in the initial indexes. As an example, let us assume that we have the following

signal:

$$x = [1, 3, 6, 5, 4, 9, 1, 2, 3, 5, 4].$$

In this case, if we consider a window size of length $W = 3$ in the symmetrical way, the very first filter value would be computed as $\mu_x(0) = (x(-1) + x(0) + x(1))/3$. Because we do not have a value for $x(-1)$, the window is computed with the initial available indexes, *i.e.*, $\mu_x(0) = (x(0) + x(1))/2 = 2$. The same is valid when the window reaches the end of the signal.

Recursive Implementation

A recursive implementation of the algorithm can be done as in (SMITH, 1999, p. 283)

$$\mu_x(k+1) = \mu_x(k) + x(k+p) - x(k-q), \tag{3.8}$$

where $p = (W-1)/2$ and $q = p + 1$.

Exponentially Weighted Moving Average (EWMA)

While the Moving Average filter assigns the same weight to every value, the Exponentially Weighted Moving Average (EWMA) weights every element according to an exponential factor. As described in (BODENHAM, 2014, p. 24), considering a data sample $(x_1, x_2, \ldots, x_{N_s})$ coming from a distribution of mean μ and variance σ^2, the following statistics can be defined:

$$\begin{aligned} Z_0 &= \mu \\ Z_i &= (1-\lambda)Z_{i-1} + \lambda x_i, \text{ for } i = 1, 2..., N_s, \end{aligned} \tag{3.9}$$

with $\lambda \in [0, 1]$ being the exponential forgetting factor and with the standard deviation of Z_i being defined as $S_{Z_i} = \left(\sqrt{\frac{\lambda}{2-\lambda}[1-(1-\lambda)^{2i}]}\right)\sigma$. In this case, a change would be marked when $Z_i > \mu + L \times S_{Z_i}$, being L a design parameter.

A recursive estimation of this filter, for both the mean and the variance values, are presented in (PERETZKI *et al.*, 2011 apud FINCH, 2009) and (BITTENCOURT *et al.*, 2015). In both (PERETZKI *et al.*, 2011) and (BITTENCOURT *et al.*, 2015), the recursive Exponentially Weighted filter is applied to detect transient changes in the process signal. Again, denoting $\mu_x(k)$ the estimate for the mean and denoting $S_x(k)$ the estimate for the variance, the recursive calculation could be done as follows (BITTENCOURT *et al.*, 2015):

$$\mu_x(k) = \lambda_\mu \times x(k) + (1-\lambda_\mu) \times \mu_x(k-1) \tag{3.10}$$

$$S_x(k) = \tfrac{2-\lambda_\mu}{2} - (\lambda_S \times (x(k) - \mu_x(k))^2 + (1-\lambda_S) \times S_x(k-1)). \tag{3.11}$$

In this case, a transient change is considered to be true when a chosen threshold is reached: $S_x(k) > l_S$.

3.2.1.2 Bandpass Filtering Approach

An interesting modification of the EWMA filter described in the previous item is proposed in (PATEL, 2016), which converts it into a bandpass filter. In (PATEL, 2016, p. 8), a deviation from the mean variable $e_x(k)$ is defined as shown below:

$$e_x(k) = x(k) - \mu_x(k). \tag{3.12}$$

This value is then inserted in Equation (3.10), resulting in the following expression:

$$e_x(k) = \frac{\lambda_\mu(q-1)}{(q-\lambda_\mu)}. \tag{3.13}$$

In order to avoid wrong interval detections due to high frequency noise, in (PATEL, 2016) the above filter is extended to a bandpass filter as follows:

$$e_x(k) = \frac{\lambda_\mu(q-1)}{q-\lambda_\mu}\frac{q(1-d)}{q-f_c}x(k), \tag{3.14}$$

being f_c the high cut-off frequency. Finally, in the same work, this expression is generalized through the replacement of λ_μ and f_c by frequencies w_1 and w_2, as follows:

$$e_x(k) = \frac{q(q-1)c}{(q-e^{w_1 T_s})(q-e^{w_2 T_s})}, \tag{3.15}$$

with c being a constant to keep the filter gain at 1 in the passband. Finally, potential intervals are defined by the indexes that satisfy the following condition:

$$\max_{k\in\Delta} e_x(k) - \min_{k\in\Delta} e_x(k) \geq l_e, \tag{3.16}$$

where l_e is a chosen threshold for the signal being studied. Notice that this method cannot be applied online and that three parameters must be selected and tuned: w_1, w_2 and l_e. In this dissertation, this method is used through a Butterworth filter implemented in Scipy (VIRTANEN *et al.*, 2020).

3.2.1.3 Cumulative Sum (CUSUM)

As described in (MONTGOMERY, 2008, p. 402), the CUSUM control chart is designed to detect small incremental changes in the mean of a process signal. Considering a data sample $(x_1, x_2, ..., x_{N_s})$ that initially follows a normal distribution $N(\mu_0, \sigma^2)$, the

following statistics can be defined (BODENHAM, 2014, p. 23):

$$C_i^+ = \begin{cases} \mu_0 & i = 0 \\ \max(0, C_{i-1}^+ + x_i - K\sigma) & i \in \{1, 2, ..., N_s\} \end{cases} \tag{3.17}$$

$$C_i^- = \begin{cases} \mu_0 & i = 0 \\ \max(0, C_{i-1}^- - x_i - K\sigma) & i \in \{1, 2, ..., N_s\} \end{cases} \tag{3.18}$$

where C_i^+ is the upper cumulative sum and C_i^- is the lower cumulative sum.

A change is then detected when $C_i^+ > L\sigma$ for the upper case and $C_i^- < -L\sigma$ for the lower case. Notice that L and K are design parameters. The signal mean and standard deviation can be estimated, in a practical implementation, from the first values of the sample (BODENHAM, 2014).

3.2.2 A Top-Down Change-Point Approach

A top-down, non-parametric change-point detection approach is applied in (WANG *et al.*, 2018 apud PETTITT, 1979) as a first step in finding intervals for process identification. A summary of the approach presented in (WANG *et al.*, 2018) is described in this subsection. This method is a modified version of the Mann-Whitney two-sample test (PETTITT, 1979).

Because this is a top-down approach, the initial data segment is defined as the whole dataset itself. Let us define a generic data segment as below:

$$x(k : k + N_s - 1) = (x(k), x(k + 1), ..., x(k + N_s - 1)), \tag{3.19}$$

with k being the initial time index of the segment and N_s its length. Therefore, the initial segment considered by the algorithm is $(x(0), x(1), \ldots, x(N))$, which is equivalent to consider $k = 0$ and to include all data points in the dataset (being N its length). Notice that the segment of length N_s is a subset of the entire dataset of length N.

The next step of the algorithm consists in calculating the relative position of $x(t)$ in the current data segment $x(k : k + N_s - 1)$, which can be done as below (WANG *et al.*, 2018):

$$D(t) = \sum_{j=k}^{k+N_s-1} \operatorname{sgn}(x(t) - x(j)) \text{ for } t = k, k + 1, \ldots, k + N_s - 1. \tag{3.20}$$

Notice that sgn() is the signal function, which can be defined as:

$$\text{sgn}(x) = \begin{cases} 1 & x > 0 \\ 0 & x = 0 \\ -1 & x < 0 \end{cases} \tag{3.21}$$

The cumulative sum of $D(t)$ is then calculated as $C(t) = C(t-1) + D(t)$, for $t = k, k+1, \ldots, k+N_s-1$, considering the initial value $C(k-1) = 0$.

A hypothesis test is then proposed in (WANG *et al.*, 2018) based on the above metric: a change position can be calculated as the time index which maximizes the absolute value of $C(t)$. One could formulate the hypothesis test as follows:

$$\begin{cases} H_0: & \underset{k \leq \tau \leq k+N_s-1}{\arg\max} |C(t)| \text{ is not a change-point} \\ \\ H_1: & \underset{k \leq \tau \leq k+N_s-1}{\arg\max} |C(t)| \text{ is a change-point} \end{cases}$$

The p-value associated with this hypothesis test was defined in (WANG *et al.*, 2018 apud PETTITT, 1979) as below:

$$p = 2e^{\left(\frac{-6|C(\tau)|^2}{N_s^2+N_s^3}\right)}. \tag{3.22}$$

Therefore, given a level of significance α for the type-I error, τ is considered a change-point index if $p < \alpha$, which means the null hypothesis can be rejected with an α change of making a mistake. If the null hypothesis is rejected, the current data segment must be divided into two new segments, one composed by the indexes located at the left of the change-point index and another one formed by the indexes located at the its right side, as defined below:

$$\begin{cases} x(k:\tau) = (x(k), x(k+1), ..., x(\tau)) \\ x(\tau+1:N) = (x(\tau+1), x(\tau+2), ..., x(N)) \end{cases}$$

The hypothesis test must then be recalculated for each new segment until no further change-points can be detected, *i.e.*, until no time index τ can reject the null hypothesis within an α confidence value.

An implementation of the algorithm described in this subsection is here suggested as below:

Algorithm 1 Pettitt Top-down Change-point Method

Input: a signal $x(k)$ of length N and a significance level α

Output: a sequence of change-points τ_s

procedure NonParametricPettitt($x(k)$,α)
  Define the initial segment as $\boldsymbol{S} = [x(0:N)]$
  Define an array of change-points $\boldsymbol{T}$ and another array $\boldsymbol{P}$ with the corresponding p-values
  for $j \in [0,\ N]$ **do**
    for each $s \in \boldsymbol{S}$ **do**
      Compute statistics D and C
      Calculate change-points τ_s
      Compute the p-values p_s of each τ_s
      if $p_s < \alpha$ **then**
        Update arrays $\boldsymbol{T}$ and $\boldsymbol{P}$
      end if
    end for
    Split $\boldsymbol{S}$ in subintervals $\boldsymbol{S} = [s_1, \ldots s_{N_i}]$ according to the resulting change-points
    if $\boldsymbol{T}$ is empty **then**
      Break For loops
    end if
  end for
  Return $[\tau_1, \tau_2, \ldots, \tau_{N_i-1}]$
end procedure

An additional parameter can be included in the above algorithm regarding the minimum size an interval must have to be further divided. In this dissertation, the $N_{s,min}$ parameter indicates the minimum length a data sample must have to generate a new change point. Therefore, even if there is enough statistical evidence to create a new change-point, it is only created if the data sample that is being split has a length higher then $N_{S.min}$.

3.3 Interval Segmentation Methods

In this section, the main algorithms mentioned in the literature overview in Section 1.2 that compose the adopted methodology described in Chapter 4 are explained in details.

3.3.1 A SISO Rank Test Method

The method presented in this Section is proposed in (RIBEIRO; AGUIRRE, 2015) and uses the effective rank and a cross-correlation scalar metric as its foundation. More specifically, it is argued that the effective rank of Autoregressive (AR) regressor matrices is a better metric compared to the persistence of excitation as an indicator of signal "activity". Two major considerations are formulated in (RIBEIRO; AGUIRRE, 2015), which are described below:

a) a given dataset is suitable for system identification if it contains sufficient information about the system dynamics;

b) a transient response can be produced by disturbances in the system. Therefore, a given dataset is only useful for identification if the system's input and output variables are actually correlated.

3.3.1.1 Singular Value Decomposition (SVD)

Because the algorithm described in this subsection is strongly based on Singular Value Decomposition, a more detailed explanation on the subject is given.

Theorem 3.1. *(VERHAEGEN; VERDULT, 2007) Let us consider a matrix* $\boldsymbol{A} \in \mathbb{R}^{m \times n}$. *It is demonstrable that any matrix of this form can be decomposed as below:*

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T, \tag{3.23}$$

with matrices $\boldsymbol{U} \in \mathbb{R}^{m \times m}$ *and* $\boldsymbol{V} \in \mathbb{R}^{n \times n}$ *being orthogonal* [1]. *Moreover, the resulting* $\boldsymbol{\Sigma} \in \mathbb{R}^{m \times n}$ *matrix contains non-zero elements only in its diagonal, which is formed by* σ_i *values ordered as follows:*

$$\sigma_1 \ \geq \ \sigma_2 \ \geq \ \cdots \ \geq \ \sigma_{r+1} = \cdots = \sigma_k, \tag{3.24}$$

being $r = \text{rank}(\boldsymbol{A})$ *and* $k = \min(m, n)$. *Then, the singular values of matrix* $\boldsymbol{A}$ *are defined as the diagonal elements* σ_i *of matrix* $\boldsymbol{\Sigma}$.

[1] An orthogonal matrix $\boldsymbol{Q}$ is a square matrix that satisfies the relationship $\boldsymbol{Q}^T\boldsymbol{Q} = \boldsymbol{Q}\boldsymbol{Q}^T = \boldsymbol{I}$, with $\boldsymbol{I}$ being the identity matrix.

3.3.1.2 Persistence of Excitation and the AR Regressor

As described in (AGUIRRE, 2015, p. 205), the concept of Persistence of Excitation defines how "active" is a signal in stationary regime and, therefore, how informative is this signal to perform a system identification.

Definition 3.2. *(AGUIRRE, 2015, p. 205) Let us consider a data interval of length N_s. Moreover, let us assume that $\bar{u}$ is the sampling average value of the input signal u(k), that r_u is the sampling autocovariance function of signal $u(k)$ and that $\boldsymbol{P}^u_{N_s}$ is the covariance matrix of u(k). Then, if the following limits exist and matrix $\boldsymbol{P}^u_{N_s}$ is non-singular, the input signal $u(k)$ is called a Persistently Exciting Signal of order N_s.*

$$\bar{u} = \lim_{N_s \to \infty} \frac{1}{N_s} \sum_{k=1}^{N_s} u(k) \tag{3.25}$$

$$r_u = \lim_{N_s \to \infty} \frac{1}{N_s} \sum_{k=1}^{N_s} (u(i) - \bar{u})(u(i+k) - \bar{u}) \tag{3.26}$$

$$\boldsymbol{P}^u_{N_s} = [r_u(i-j)] \ , i = 1, \ldots, n; \ j = 1, \ldots, n \tag{3.27}$$

The existence of the above limits is only possible if $u(k)$ is considered in stationary regime. It is also important to mention that in the case when the signals are real and with zero mean, the autocovariance function is equivalent to the autocorrelation function (AGUIRRE, 2015, p. 181).

As an alternative to the persistence of excitation as a measure of the variability in the input signal, in (RIBEIRO; AGUIRRE, 2015) the computation of the effective rank of the AR regressor matrix is proposed. The main reason behind this idea is that, while the definition of persistence of excitation is defined for the stationary regime, the AR regressor structure defines the dynamics of the output signal of the system and, therefore, can be used to evaluate transient responses, which is the main goal of system identification.

3.3.1.3 Effective Rank

As explained in the last section, the AR structure is used to measure information as an alternative to the persistence excitation definition. This measure of information is computed in (RIBEIRO; AGUIRRE, 2015) through the effective rank.

The concept of effective rank is presented in (ROY; VETTERLI, 2007) as a way to overcome numerical limitations encountered in optimization problems that are based on rank values. Two different approaches to calculate the effective rank are proposed in (RIBEIRO; AGUIRRE, 2015) and are reproduced below:

Type 1. The effective rank is computed as the number of normalized singular values p_i greater than a minimum threshold l_1, as shown below:

$$p_i = \frac{\sigma_i}{\|\sigma\|_1} \geq l_1, \tag{3.28}$$

being p_i defined as in (ROY; VETTERLI, 2007) and $\|\cdot\|_1$ denoting the l_1-norm, calculated as follows:

$$\|\sigma\|_1 = \sum_{i=1}^{k} \|\sigma_i\| . \tag{3.29}$$

Therefore, the effective rank r_1^{ef} is calculated as:

$$r_1^{ef} = \sum_{i=2}^{k} H[p_i - l_1], \tag{3.30}$$

where H is the Heaviside (step) function, defined as:

$$H[x] = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases} \tag{3.31}$$

Type 2. The effective rank is computed as the number of singular values $\sigma_{i-1} - \sigma_i$ differences greater than a threshold l_2, as defined below:

$$r_2^{ef} = \sum_{i=2}^{k} H[\sigma_{i-1} - \sigma_i - l_2]. \tag{3.32}$$

3.3.1.4 Cross-correlation Scalar Metric

As a final step of the method described in this subsection, intervals that present a large effective rank value are subjected to a cross-correlation check to ensure that the transient response is actually being caused by the input signals and not by disturbances. The cross-correlation function of two signals $u(k)$ and $y(k)$ is defined as (AGUIRRE, 2015, p. 180):

$$r_{u,y}(\tau, t) = E[u(t)y(t+\tau)], \tag{3.33}$$

where τ is a time lag.

As described in (AGUIRRE, 2015, p. 180), assuming egordicity for discrete-time series, the cross-correlation function can be written as:

$$r_{u,y}(\tau) = \lim_{N\to\infty} \frac{1}{2N+1} \sum_{k=-N}^{N} u(k)y(k+\tau). \tag{3.34}$$

For a time-invariant signal with a finite length N_s, an estimate for the definition in

Equation (3.33) can be computed as (AGUIRRE, 2015, p. 181):

$$\hat{r}_{u,y}(\tau) = \frac{1}{N_s} \sum_{k=1}^{N_s-\tau} u(k)y(k+\tau). \tag{3.35}$$

It is easy to notice that $\hat{r}_{u,y}(\tau)$ is a function of the lag value τ, as pointed out in (RIBEIRO; AGUIRRE, 2015). As a consequence, it is not trivial to evaluate in a general fashion whether the input and the output signals are correlated. For this reason, in (RIBEIRO; AGUIRRE, 2015) the cross-correlation function is summarized in a single scalar value. The resulting scalar metric proposed in (RIBEIRO; AGUIRRE, 2015) is defined as below:

$$\begin{aligned} s &= \sum_{\tau=-\tau_{max}}^{\tau_{max}} g(\rho(\tau), \tau, p) \\ g(\rho(\tau), \tau, p) &= \begin{cases} 0, & \text{if } \mid \rho \mid \leq p \\ \frac{|\rho(\tau)-p|}{|\tau|}, & \text{if } \mid \rho \mid > p \text{ and } \tau \neq 0 \\ \mid \rho(\tau) \mid -p, & \text{if } \mid \rho \mid > p \text{ and } \tau = 0 \end{cases} \end{aligned} \tag{3.36}$$

where $\rho(\tau)$ is the normalized cross-correlation function and $[-p, +p]$ defines the 95% confidence interval, with $p = \frac{1.96}{\sqrt{N_s}}$, being N_s the sample size of a given interval. Notice that different confidence levels can be adopted. Moreover, the normalized cross-correlation function is computed in this dissertation as follows (DERRICK; THOMAS, 2004, p. 193):

$$\rho(\tau) = \frac{\sum_{k=1}^{N_s}(x(k)-\bar{x})(y(k-\tau)-\bar{y})}{\sqrt{(\sum_{k=1}^{N_s}(x(k)-\bar{x})^2)(\sum_{k=1}^{N_s}(y(k-\tau)-\bar{y})^2)}}, \tag{3.37}$$

where $\bar{x}$ and $\bar{y}$ are the input and the output sample means, respectively.

3.3.1.5 Steps of the Algorithm

The algorithm proposed in (RIBEIRO; AGUIRRE, 2015) is summarized below, being the original algorithm structure presented in Algorithm 2 with some adaptations.

a) divide the data into a pre-defined number of intervals (windows) $N_w = \frac{N}{w_{ic}}$, where w_{ic} is the window increment and N is the length of the entire dataset;

b) compute the AR regressor matrix and its corresponding effective rank for each interval of data;

c) compute the cross-correlation scalar metric and compare it to a pre-defined threshold l_{cc}.

Algorithm 2 SVD Method for Search of Intervals (Adapted from RIBEIRO; AGUIRRE, 2015)

Input: $y(k)$, w_{ic} (window increment), l_1 or l_2 thresholds and l_{cc} threshold
Output: list of segment indexes q

procedure SVDSEGMENTATION($y(k), w_{ic}, l_{1,2}, l_{cc}$)
  Compute the number of windows $N_w = \frac{N}{w_{ic}}$
  for each $k \in [0, N_S - 1]$ **do**
    Compute the ith window $y\{i\}$
    Build the AR regressor matrix $\boldsymbol{A}_i$
    Compute r_1^{ef} or r_2^{ef} for a given l_1 or l_2, respectively
  end for
  Compute a list q with the indexes of windows $y\{i\}$ in decreasing order of effective rank.
  for each $i \in [0, N_w - 1]$ **do**
    if $y\{q(i)\}$ overlaps with $y\{q(0:i)\}$ **then**
      Remove $q(i)$ from list q
    end if
  end for
  for each $i \in [0, length(q)]$ **do**
    Compute cross-correlation metric s_i
    if $s_i < l_{cc}$ **then**
      Remove $q(i)$ from list q
    end if
  end for
  Return q
end procedure

3.3.2 A SISO Numerical Conditioning Method

The method described in this subsection is proposed in (PERETZKI *et al.*, 2011) and in (BITTENCOURT *et al.*, 2015). The following assumptions are made in the original works:

a) only SISO systems are considered;

b) the process can be described by linear models $\mathcal{M}(\theta)$;

c) a transient response is only adequate for system identification if the input and the output signals are actually correlated;

d) for open-loop systems, signal $m_v(k)$ must "shake" the process with enough variations. For closed-loop systems, on the other hand, the same assumption holds for the set-point $r(k)$ (see Item 3.3.2.7).

A formal definition of the problem is given in (PERETZKI *et al.*, 2011) and it is described below:

Definition 3.3. *(Adapted from PERETZKI et al., 2011) Let us assume that a collection of data* $\boldsymbol{Z^N} = [\boldsymbol{Z}(1)^T, \ldots, \boldsymbol{Z}(N)^T]^T$ *is available, where* N *is the length of the entire data and* T *is the matrix transposition operation. Moreover, let us assume that* $\boldsymbol{Z}(k)^T = [r(k), m_v(k), y(k)]$*, according to the control system described in Figure 1. Signals* $r(k)$*,* $m_v(k)$ *and* $y(k)$ *are, respectively, the set-point of the controller, the process input coming from the controller and the process measured output. Therefore, the problem objective is to find discrete time intervals* $\Delta = [\tau_{init}, \tau_{end}]$*, such that* $\boldsymbol{Z^{N_\Delta}}$ *is suitable to perform a system identification of the process being considered, with* N_Δ *being the number of resulting intervals.*

The main idea behind the algorithm described in this subsection can be summarized through the following topics:

a) check if there is any variability in the input and in the output signals (see Item 3.2.1.1);

b) define a condition number based on the information matrix to verify if the least square problem is numerically well-conditioned;

c) verify if the input and output sequences are actually correlated. A causality test is performed for this purpose.

3.3.2.1 The Information Matrix

A general parametric structure for Linear Regression models can be defined as in Equation (2.8). An alternative for finding the parameter vector is through the so-called prediction error methods, in which the following prediction error is minimized (AGUIRRE, 2015, p. 239):

$$\xi(k, \hat{\boldsymbol{\theta}}) = y(k) - \boldsymbol{\psi}_{yu}^T(k-1)\hat{\boldsymbol{\theta}}. \tag{3.38}$$

Notice that $\boldsymbol{\psi}_{yu}^T(k-1)\hat{\boldsymbol{\theta}}$ is a prediction for the output estimated with measured data available until instant $k-1$. Therefore, this prediction is commonly called the one step ahead prediction and can be represented as $\hat{y}(k|k-1)$ (AGUIRRE, 2015, p. 239).

The least square criterion can be used to estimate the parameter vector $\hat{\boldsymbol{\theta}}_{N_s}$ for a sample of data, as follows:

$$\hat{\boldsymbol{\theta}}_{N_s} = \arg\min_{\hat{\boldsymbol{\theta}}} \sum_{k=1}^{N_s} \left[y(k) - \boldsymbol{\psi}_{yu}^T(k-1)\hat{\boldsymbol{\theta}} \right]^2. \tag{3.39}$$

A closed solution for Equation (3.39) can be obtained as in Equation (2.22) and can be rewritten, for a limited sample of data, as (AGUIRRE, 2015, p. 242):

$$\hat{\boldsymbol{\theta}}_{N_s} = \left[\frac{1}{N_s}\sum_{k=1}^{N_s}\boldsymbol{\psi}(k-1)\boldsymbol{\psi}^T(k-1)\right]^{-1}\left[\frac{1}{N_s}\sum_{k=1}^{N_s}\boldsymbol{\psi}(k-1)y(k)\right]. \tag{3.40}$$

It is interesting to notice that matrix $\hat{\boldsymbol{R}}_{N_s}$ defined as

$$\hat{\boldsymbol{R}}_{N_s} = \frac{1}{N_s}\sum_{k=1}^{N_s}\boldsymbol{\psi}(k-1)\boldsymbol{\psi}^T(k-1) \tag{3.41}$$

is symmetric and positive definite ($\hat{\boldsymbol{R}}_{N_s} > 0$) and it is frequently called the Information Matrix. This matrix is closely related to the so-called Fisher Information (OLIVEIRA JR.; GARCIA, 2017), being useful to provide information about a given data sample, as detailed in the next item.

3.3.2.2 The Fisher Information Matrix

As described in (DEGROOT; SCHERVISH, 2016, p. 514), the Fisher Information can measure "the amount of information that a sample of data contains about an unknown parameter", which is exactly what the method described in Subsection 3.3.2 seeks to understand.

Let us consider a generic regression structure $y(k) = f(\boldsymbol{\psi}(k), \hat{\boldsymbol{\theta}}) + \xi(k)$. A linear regression structure can then be written as $y(k) = \boldsymbol{\psi}(k)^T\hat{\boldsymbol{\theta}} + \xi(k)$, as the one defined in Equation (2.8). In this case, $y(k)$ comes from a gaussian distribution in the form $y(k) \sim N(f(\boldsymbol{\psi}(k), \hat{\boldsymbol{\theta}}), \sigma_y^2)$. Therefore, for a given instant k, the Probability Density Function of $y(k)$ is shown below (DEVORE, 2016, p. 268):

$$f(\boldsymbol{\psi}(k), y(k)|\hat{\boldsymbol{\theta}}) = \frac{1}{\sqrt{2\pi\sigma_y^2}}e^{-\frac{1}{2\sigma_y^2}(y(k)-f(\boldsymbol{\psi}(k),\hat{\boldsymbol{\theta}}))^2}. \tag{3.42}$$

Notice that $\boldsymbol{\psi}(k)^T = [\psi_1(k)\ \ \psi_2(k)\ \ \cdots\ \ \psi_{n_{\hat{\theta}}}(k)]$ is the regressor vector defined as in Equation 2.20. In this case, the log-likelihood function can be defined as (DEVORE, 2016, p. 268)

$$l(\boldsymbol{\psi}(k), y(k)|\hat{\boldsymbol{\theta}}) = -\frac{1}{2}\log(2\pi\sigma_y^2) - \frac{(y(k)-\boldsymbol{\psi}(k)^T\hat{\boldsymbol{\theta}})^2}{2\sigma_y^2}. \tag{3.43}$$

Considering $\hat{\boldsymbol{\theta}}$ as a vector of parameters, the Fisher Information Matrix of $l(\boldsymbol{\psi}(k), y|\hat{\boldsymbol{\theta}})$

can be defined as (MARTENS, 2020)

$$\begin{aligned}\boldsymbol{I}(\hat{\boldsymbol{\theta}}) &= E_{f(\boldsymbol{\psi}(k),y(k)|\hat{\boldsymbol{\theta}})}\left[\boldsymbol{\nabla}_{\hat{\boldsymbol{\theta}}} l(\boldsymbol{\psi}(k),y(k)|\hat{\boldsymbol{\theta}})\boldsymbol{\nabla}_{\hat{\boldsymbol{\theta}}}^T l(\boldsymbol{\psi}(k),y(k)|\hat{\boldsymbol{\theta}})\right] \\ &= -E_{f(\boldsymbol{\psi}(k),y(k)|\hat{\boldsymbol{\theta}})}\left[\boldsymbol{H}_{l(\boldsymbol{\psi}(k),y(k)|\hat{\boldsymbol{\theta}})}\right],\end{aligned} \tag{3.44}$$

where $\boldsymbol{\nabla}_{\hat{\boldsymbol{\theta}}}$ is the gradient with respect to $\hat{\boldsymbol{\theta}}$, $E_{f(\boldsymbol{\psi}(k),y(k)|\hat{\boldsymbol{\theta}})}$ is the expected value of the probability density function and $\boldsymbol{H}_{l(\boldsymbol{\psi}(k),y(k)|\hat{\boldsymbol{\theta}})}$ is the Hessian of the log-likelihood function. The Hessian function can also be expressed as $\boldsymbol{H}_{l(\boldsymbol{\psi}(k),y(k)|\hat{\boldsymbol{\theta}})} = \sum_{k=1}^{N_s} \frac{\partial^2 l(\boldsymbol{\psi}(k),y(k)|\hat{\boldsymbol{\theta}})}{\partial\hat{\boldsymbol{\theta}}\partial\hat{\boldsymbol{\theta}}^T}$ (HASTIE; TIBSHIRANI; FRIEDMAN, 2009, p. 266) and, therefore, considering an interval of length N_s, the Fisher Information Matrix can be computed as (HASTIE; TIBSHIRANI; FRIEDMAN, 2009, p. 266):

$$\boldsymbol{I}(\hat{\boldsymbol{\theta}}) = -\sum_{k=1}^{N_s} \frac{\partial^2 l(\boldsymbol{\psi}(k),y(k)|\hat{\boldsymbol{\theta}})}{\partial\hat{\boldsymbol{\theta}}\partial\hat{\boldsymbol{\theta}}^T}. \tag{3.45}$$

The gradient $\boldsymbol{\nabla}_{\hat{\boldsymbol{\theta}}}$ for an instant k can then be calculated as follows:

$$\begin{aligned}\boldsymbol{\nabla}_{\hat{\boldsymbol{\theta}}} l(\boldsymbol{\psi}(k),y(k)|\hat{\boldsymbol{\theta}}) &= \boldsymbol{\nabla}_{\hat{\boldsymbol{\theta}}}\left[-\frac{1}{2}\log(2\pi\sigma_y^2) - \frac{(y(k)-\boldsymbol{\psi}(k)^T\hat{\boldsymbol{\theta}})^2}{2\sigma_y^2}\right] \\ &= \frac{\partial}{\partial\hat{\boldsymbol{\theta}}}\left[\frac{-y(k)^2}{2\sigma_y^2} + \frac{y(k)\boldsymbol{\psi}(k)^T\hat{\boldsymbol{\theta}}}{\sigma_y^2} - \frac{\hat{\boldsymbol{\theta}}^T\boldsymbol{\psi}(k)\boldsymbol{\psi}(k)^T\hat{\boldsymbol{\theta}}}{2\sigma_y^2}\right] \\ &= \frac{y(k)\boldsymbol{\psi}(k)}{\sigma_y^2} - \frac{\hat{\boldsymbol{\theta}}^T\boldsymbol{\psi}(k)\boldsymbol{\psi}(k)^T}{\sigma_y^2}.\end{aligned}$$

Therefore, the Hessian matrix can be calculated as:

$$\begin{aligned}\boldsymbol{H}_{l(\boldsymbol{\psi}(k),y(k)|\hat{\boldsymbol{\theta}})} &= \frac{\partial}{\partial\hat{\boldsymbol{\theta}}^T}\nabla_{\hat{\boldsymbol{\theta}}} l(\boldsymbol{\psi}(k),y(k)|\hat{\boldsymbol{\theta}}) \\ &= \frac{\partial}{\partial\hat{\boldsymbol{\theta}}^T}\left[\frac{y(k)\boldsymbol{\psi}(k)}{\sigma_y^2} - \frac{\hat{\boldsymbol{\theta}}^T\boldsymbol{\psi}(k)\boldsymbol{\psi}(k)^T}{\sigma_y^2}\right] \\ &= -\frac{\boldsymbol{\psi}(k)\boldsymbol{\psi}(k)^T}{\sigma_y^2}.\end{aligned}$$

Finally, for a given instant k, the Fisher Information Matrix follows directly as

$$\boldsymbol{I}(\hat{\boldsymbol{\theta}},k) = \frac{\boldsymbol{\psi}(k)\boldsymbol{\psi}(k)^T}{\sigma_y^2}. \tag{3.46}$$

For an interval of length N_s, *i.e.*, considering now the regressor matrix $\boldsymbol{\Psi}$ defined in Equation (2.21), $\boldsymbol{I}(\hat{\boldsymbol{\theta}})$ can be reformulated as in Equation 3.45, resulting in the following expression:

$$\boldsymbol{I}(\hat{\boldsymbol{\theta}}) = \sum_{k=1}^{N_s} \frac{\boldsymbol{\psi}(k)\boldsymbol{\psi}(k)^T}{\sigma_y^2} = \frac{\boldsymbol{\Psi}^T\boldsymbol{\Psi}}{\sigma_y^2}. \tag{3.47}$$

3.3.2.3 QR Decomposition

A solution to the Least Squared Problem is proposed in (PERETZKI *et al.*, 2011) through the so-called QR Decomposition, which is better numerically conditioned.

Theorem 3.2. *(VERHAEGEN; VERDULT, 2007, p. 27) Let us consider a matrix* $\boldsymbol{A} \in \mathbb{R}^{m \times n}$*. It is possible to prove that any matrix of this form can be decomposed as*

$$\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{R}. \tag{3.48}$$

Matrix $\boldsymbol{Q} \in \mathbb{R}^{m \times m}$ *is orthogonal and matrix* $\boldsymbol{R} \in \mathbb{R}^{m \times n}$ *is upper triangular. In the case when* $n > m$*, matrix* $\boldsymbol{R}$ *has columns augmented on the right. On the other hand, if* $m > n$*, matrix* $\boldsymbol{R}$ *has rows augmented with zeros at the bottom.*

The QR Decomposition proposed in (PERETZKI *et al.*, 2011) is applied to matrix $\boldsymbol{A} = [\boldsymbol{\Psi} \;\; \boldsymbol{y}]$, where $\boldsymbol{y} \in \mathbb{R}^{N_s}$ is an output sample, with N_s being the sample size and $\boldsymbol{\Psi} \in \mathbb{R}^{N_s \times n_\theta}$ being the regressor matrix. Matrix $\boldsymbol{R}$ can be written as:

$$\boldsymbol{R} = \begin{bmatrix} \boldsymbol{R}_0 \\ \vdots \\ \boldsymbol{0} \end{bmatrix} \;, \boldsymbol{R}_0 = \begin{bmatrix} \boldsymbol{R}_1 & \boldsymbol{R}_2 \\ \boldsymbol{0} & R_3 \end{bmatrix}, \tag{3.49}$$

with $\boldsymbol{R}_0 \in \mathbb{R}^{N_s \times (n_\theta + 1)}$, $\boldsymbol{R}_1 \in \mathbb{R}^{n_\theta \times n_\theta}$, $\boldsymbol{R}_2 \in \mathbb{R}^{n_\theta \times 1}$ and R_3 being a scalar value. The model structure used by the authors is the Laguerre Filter, described in detail in Section 2.4, which has the regressor repeated below:

$$\boldsymbol{\psi}_u^T(k) = \begin{bmatrix} L_1(q,\alpha)u(k) & \cdots & L_{n_b}(q,\alpha)u(k) \end{bmatrix}.$$

In this case, $n_\theta = n_b$ is the regressor order. As explained in (PERETZKI *et al.*, 2011), if the orthonormal transformation $\boldsymbol{Q}^T$ is applied to the cost function (3.38), one obtains the following results:

$$\begin{aligned} &\left\| \boldsymbol{Q}^T(\boldsymbol{y} - \boldsymbol{\Psi}\hat{\boldsymbol{\theta}}) \right\|_2^2 = \\ &= \left\| \begin{bmatrix} \boldsymbol{R}_2 \\ R_3 \end{bmatrix} - \begin{bmatrix} \boldsymbol{R}_1\hat{\boldsymbol{\theta}} \\ 0 \end{bmatrix} \right\|_2^2 = \\ &= \|\boldsymbol{R}_2 - \boldsymbol{R}_1\boldsymbol{\theta}\|_2^2 + |R_3|^2 . \end{aligned} \tag{3.50}$$

Therefore, the $\hat{\boldsymbol{\theta}}$ that minimizes the function above can be obtained as the solution of the following equation:

$$\boldsymbol{R}_1\hat{\boldsymbol{\theta}} = \boldsymbol{R}_2. \tag{3.51}$$

In the same way, Equation (3.41) can be reformulated as:

$$\hat{\boldsymbol{R}}_{N_s} = \frac{1}{N_s}\boldsymbol{\Psi}^T\boldsymbol{\Psi} = \frac{1}{N_s}\boldsymbol{R}_1^T\boldsymbol{R}_1. \tag{3.52}$$

3.3.2.4 Condition Number

Because the solution of the covariance matrix $\boldsymbol{P}^u_{N_s}$ in Equation (3.27) is usually ill-conditioned and this matrix is usually singular, and because the persistence of excitation depends on the signal being stationary, in (PERETZKI *et al.*, 2011) and in (BITTENCOURT *et al.*, 2015) the numerical conditioning of the information matrix is proposed as an alternative to the concept of persistence of excitation. More specifically, because the feasibility of Equation (3.40) depends on the information matrix $\hat{\boldsymbol{R}}_{N_s}$ (Equation (3.41)) being invertible, the numerical conditioning of $\hat{\boldsymbol{R}}_{N_s}$ appears as a measure of the quality of the data for estimating the parameters of a model.

The condition number is a numerical analysis concept that allows one to evaluate the perturbation behavior of a mathematical problem (TREFETHEN; BAU, 1997). An "ill-conditioned" system is one highly influenced by disturbances. This concept can be understood in the context of matrix inversion, where a matrix inversion is well-posed when its condition number is sufficiently small. A more formal definition is adapted from (BITTENCOURT *et al.*, 2015) as follows:

Definition 3.4. *Let us consider a linear system of the form* $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$*. If one considers a perturbed system of the form* $\boldsymbol{A}(\boldsymbol{x} + \delta\boldsymbol{x}) = (\boldsymbol{b} + \delta\boldsymbol{b})$*, it follows that*

$$\frac{\|\delta\boldsymbol{x}\|_p}{\|\boldsymbol{x}\|_p} \leq \kappa_p(\boldsymbol{A})\frac{\|\delta\boldsymbol{b}\|_p}{\|\boldsymbol{b}\|_p},$$

where $\kappa_p(\boldsymbol{A}) \geq 1$ *is the p-norm condition number of matrix* $\boldsymbol{A}$*. Small values of* $\kappa_p(\boldsymbol{A})$ *mean that the system is "well-conditioned", while large values mean the system is "ill-conditioned", i.e., changes in* $\boldsymbol{x}$ *can lead to much larger changes in the system output.*

In the problem being studied in this item, the condition number of the information matrix is the ultimate goal. More specifically, the 2-norm condition number is considered, which can be computed as follows:

$$\kappa_2(\hat{\boldsymbol{R}}_{N_s}) = \frac{\sigma_{max}(\hat{\boldsymbol{R}}_{N_s})}{\sigma_{min}(\hat{\boldsymbol{R}}_{N_s})}, \tag{3.53}$$

with $\sigma_{max}(\hat{\boldsymbol{R}}_{N_s})$ and $\sigma_{min}(\hat{\boldsymbol{R}}_{N_s})$ meaning the largest and the smallest singular values of $\hat{\boldsymbol{R}}_{N_s}$, respectively.

Notice that the condition number goes from 0 to inifity. In fact, the closer the condition number of a particular matrix is to 1, the more confident one can be that this matrix can be inverted. On the contrary, if the condition number tends to infinity, the matrix cannot be inverted. As pointed out in (BITTENCOURT *et al.*, 2015), the reciprocal condition number $\kappa_2^{-1}(\hat{\boldsymbol{R}}_{N_s})$ can be used as an alternative, which now varies from $[0, 1]$. In this case, the closest the reciprocal value is to 1, the more robust is the numerical result.

It is interesting to point out that, as mentioned in (BITTENCOURT *et al.*, 2015), input signals that are not too "active", such as the step change, will usually result in large values for the condition number (in the order of 10^4). The same goes for higher orders of regressor structures, *i.e.*, higher values for n_θ.

3.3.2.5 Correlation Between The Input and the Output

In order to verify if variations in the output signal are actually caused by the input signal and not by disturbances in the system, a correlation test can be applied. As detailed in (BITTENCOURT *et al.*, 2015), instead of verifying the correlation itself, it would be more interesting to check if there is a causal relationship among the input and the output signals. The concept of Granger causality is presented in (BITTENCOURT *et al.*, 2015 apud GRILLENZONI, 1996) in such a way that if some of the estimated parameters are significantly non-zero, then delayed versions of the input should give more information to predict the output than using only delayed versions of the output itself, indicating causality. The way the execution of this test is proposed in (PERETZKI *et al.*, 2011) is to estimate $\hat{\boldsymbol{\theta}}_{N_s}$ and verify if any of the estimated parameters are significantly non-zero.

Let us assume that the real system is described by a linear model with parameters $\boldsymbol{\theta}_0$. Therefore, the estimated parameters $\hat{\boldsymbol{\theta}}_{N_s}$ are normally distributed $\hat{\boldsymbol{\theta}}_{N_s} \sim N(\boldsymbol{\theta}_0, \boldsymbol{P}_{N_s})$, where $\boldsymbol{P}_{N_s}$ is the covariance matrix for a data interval of length N_s, which can be defined as follows (LJUNG, 1999, p. 284):

$$\begin{aligned} \hat{\boldsymbol{P}}_{N_s} &= \frac{1}{N_s}\hat{\sigma}^2_{N_s}\hat{\boldsymbol{R}}^{-1}_{N_s} \\ \hat{\sigma}^2_{N_s} &= \frac{1}{N_s}\sum_{k=1}^{N_s}\xi^2(k, \hat{\boldsymbol{\theta}}_{N_s}), \end{aligned} \tag{3.54}$$

where $\hat{\sigma}^2_{N_s}$ is the noise variance.

Then, in (PERETZKI *et al.*, 2011) and in (BITTENCOURT *et al.*, 2015) the causality is verified by a hypothesis test that assumes, as the null hypothesis, that all the real parameters of the model are zero, *i.e.* $H_0 : \boldsymbol{\theta}_0 = \mathbf{0}$, where $\boldsymbol{\theta}_0$ is the true parameter

vector. Therefore, if the estimated parameters can reject the null hypothesis within an α significance level, then one can assume that the input and the output signals have a causal relationship. To perform the hypothesis test, the following statistics is proposed:

$$\hat{\chi}_{N_s} = \hat{\boldsymbol{\theta}}_{N_s}^T \hat{\boldsymbol{P}}_{N_s}^{-1} \hat{\boldsymbol{\theta}}_{N_s} \in \chi_d^2, \tag{3.55}$$

where d is the degree of freedom, defined as the dimension of $\hat{\boldsymbol{\theta}}_{N_s}$, *i.e.*, n_θ. If the null hypothesis is rejected, then the parameter estimate $\hat{\boldsymbol{\theta}}_{N_s}$ is considered different from zero with an α change of committing a mistake. A critical value $\chi_{d,\alpha}$ for the statistic can be calculated from the chi-squared table based on the significance value α and on d degrees of freedom.

It is interesting to mention that the statistic $\hat{\chi}_{N_s}$ can be calculated through the QR decomposition. The parameter estimate can be calculated from Equation (3.51) as $\hat{\boldsymbol{\theta}}_{N_s} = \boldsymbol{R}_1^{-1}\boldsymbol{R}_2$. Moreover, from $\hat{\boldsymbol{P}}_{N_s}^{-1} = \frac{N_s}{\sigma_{N_s}^2}\hat{\boldsymbol{\theta}}_{N_s}\hat{\boldsymbol{R}}_{N_s}$ and from the definition of $\xi(k, \hat{\boldsymbol{\theta}}_{N_s})$ in Equation (3.38), one obtains $\hat{\sigma}_{N_s}^2 = \frac{1}{N_s}\left\|\boldsymbol{y} - \boldsymbol{\Psi}\hat{\boldsymbol{\theta}}_{N_s}\right\|_2^2 = \frac{1}{N_s}|R_3|^2$ (PERETZKI *et al.*, 2011).

Therefore, if the QR factorization is performed, Equation (3.55) can be reformulated as (PERETZKI *et al.*, 2011)

$$\hat{\chi}_{N_s} = \left[\boldsymbol{R}_1^{-1}\boldsymbol{R}_2\right]^T \left[\frac{N_s}{\hat{\sigma}_{N_s}^2}\hat{\boldsymbol{R}}_{N_s}\right] \left[\boldsymbol{R}_1^{-1}\boldsymbol{R}_2\right] = \left\|\frac{\sqrt{N_s}}{|R_3|}\boldsymbol{R}_2\right\|_2^2. \tag{3.56}$$

Finally, it is important to mention that the causality test here described is only valid if the regressor vector if formed exclusively by components of the input signal, which is the case of the Laguerre structure regressor.

3.3.2.6 Steps of the Algorithm

In this item, the outline of the algorithm proposed in (PERETZKI *et al.*, 2011) and (BITTENCOURT *et al.*, 2015) is presented. Based on the mathematical background given in this subsection, the algorithm can be summarized by a sequence of steps as follows:

a) a recursive exponentially weighted filter is applied to both the input and the output signals, as described in Item 3.2.1.1;

b) the Laguerre filter, as well as the condition number of its information matrix, are recursively calculated for the input signal, in parallel to the calculation of the exponentially weighted filters;

c) if a change is detected in the variance filter, *i.e.*, if the variance filter exceeds a threshold l_S, the condition number $\kappa_p(\hat{\boldsymbol{R}}_{N_s})$ is verified against its threshold;

d) in the case when the condition number $\kappa_p(\hat{\boldsymbol{R}}_{N_s})$ is lower than its threshold l_κ, the parameter vector $\hat{\boldsymbol{\theta}}_{N_s}$ is estimated and the chi-squared correlation test is performed;

e) if l_S, l_κ and $\chi_{d,\alpha}$ thresholds are all satisfied, an interval is defined with its first index beginning when the variance condition is satisfied. Finally, the interval is incremented until at least one of the thresholds is not satisfied anymore, in which case the interval is saved;

f) all the previous steps go on until the end of the signal is reached.

The original outline proposed in (PERETZKI *et al.*, 2011) is adapted in Algorithm 3. It is important to reinforce that Algorithm 3 is implemented in (PERETZKI *et al.*, 2011) in a recursive manner. In fact, a recursive formulation of the mathematical background provided in this item is given in details in (BITTENCOURT *et al.*, 2015). In this dissertation, the algorithm adopted in the methodology is not implemented recursively and, therefore, details of recursive implementations are not provided. Finally, it is interesting to mention that a similar method is proposed in (ARENGAS; KROLL, 2017b), but the Laguerre Filter is replaced by an ARX structure and a configurable sliding window algorithm is proposed instead of the recursive approach. Notice that both the recursive and the sliding window implementations of the average and variance filters are described in Item 3.2.1.1.

3.3.2.7 Closed-loop and Open-loop Scenarios

The algorithm described in this item can be applied to both the open and the closed-loop system identification scenarios. When dealing with closed-loop identification, however, the problem is not trivial and depends on the model structure being adopted.

The conditions for an ARX structure to be identifiable through historical data in closed-loop systems are exemplified in both (SHARDT; HUANG, 2013a) and (BITTENCOURT *et al.*, 2015). However, it is shown in (BITTENCOURT *et al.*, 2015), in a general manner, that while in open-loop systems the manipulated variable $m_v(k)$ must be persistently exciting, in closed-loop systems it is the set-point $r(k)$ that must satisfy this condition. Therefore, for any linear process, if one is interested in estimating the rational transfer function $G(q)$ for an open-loop system, the manipulated variable must “shake”

Algorithm 3 SISO Laguerre Method for Search of Intervals (Adapted from PERETZKI *et al.*, 2011)

Input: $u(k)$ and $y(k)$
Output: intervals of the form $\Delta = [\tau_{init},\ \tau_{end}]$

procedure SISOLaguerreSegmentation($u(k), y(k)$)
  Compute variances of $u(k)$ and $y(k)$ ($\sigma_u^2(k)$ and $\sigma_y^2(k)$)
  for each $k \in [0, N]$ **do**
    if $\sigma_u^2(k)$ and $\sigma_y^2(k)$ are larger than its threshold **then**
      Compute $\kappa_p^{-1}(\hat{\boldsymbol{R}}_{N_s})$
      if $\kappa_p^{-1}(\hat{\boldsymbol{R}}_{N_s})$ is smaller than its threshold **then**
        Compute $\hat{\chi}_{N_s}$
        if $\hat{\chi}_{N_s}$ is larger than its threshold **then**
          Mark data interval $\Delta = [\tau_{init},\ \tau_{end}]$ as useful
        end if
      end if
    end if
  end for
  Return values of Δ
end procedure

in a minimal way. In the same fashion, if $G(q)$ needs to be obtained in a closed-loop system, it is the set-point that must suffer sufficient changes (BITTENCOURT *et al.*, 2015). For this reason, in order to apply the algorithm for closed-loop system identification, the following modifications in the described outline must be made:

a) instead of searching for variance changes in the manipulated variable, one must now look for changes in the set-point;

b) the condition number of the information matrix is now computed with the set-point, which would be an alternative way to verify if the set-point is persistently exciting.

3.3.3 A Statistical Approach

The method presented in this subsection is proposed in (WANG *et al.*, 2018) and it is strongly based on statistical metrics. The algorithm described by the authors can be summarized in four main steps:

a) finding change-point positions in the time-series datasets and creating initial segments based on these points;

b) evaluating whether each segment of data obtained in the previous step experiences significant changes in magnitude or if they are kept around a constant value dis-

turbed by random noise;

c) for the intervals that did not experience magnitude changes, verify if there is at least a magnitude difference between two adjacent ones;

d) unify the candidate data segments for the input and for the output data, creating a single interval for both the input and the output. Next, verify if in this single interval the input data was affected by magnitude changes. In the case it did, consider the interval as useful for performing system identification.

This algorithm can be applied to both open and closed-loop system identification. In the later case, which is the one addressed in (WANG *et al.*, 2018), the reference signal $r(k)$ must have large magnitude changes, in such a way that the corresponding changes in the output $y(k)$ are larger than the changes caused by variations in the external disturbance and by unmeasured noise. Moreover, in (WANG *et al.*, 2018) it is highlighted the importance of both the input and the output to have suffered magnitude changes in order to consider an interval as useful for system identification, which is an alternative to check the correlation between both variables.

3.3.3.1 Finding Change-Points

The first step of the algorithm is to find changing positions in each input and output signal following the approach described in Subsection 3.2.2, where each resulting interval is defined by two consecutive change-points.

As proposed in (WANG *et al.*, 2018), one can impose a maximum length to the resulting intervals. Let us assume that a particular interval of length N_s is obtained and that a user-defined maximum size of N_{max} is imposed. Therefore, if Q is the quotient of the division between integers N_s and N_{max}, and that R is the remainder of the same division, the initial segment can be divided into Q segments of length N_{max} and in 1 segment of length R.

3.3.3.2 Check for Magnitude Changes

Once initial intervals are found through the change-point algorithm, it is proposed to evaluate whether each data segment experiences a significant amount of magnitude

changes. For this purpose, the following hypothesis test is formulated:

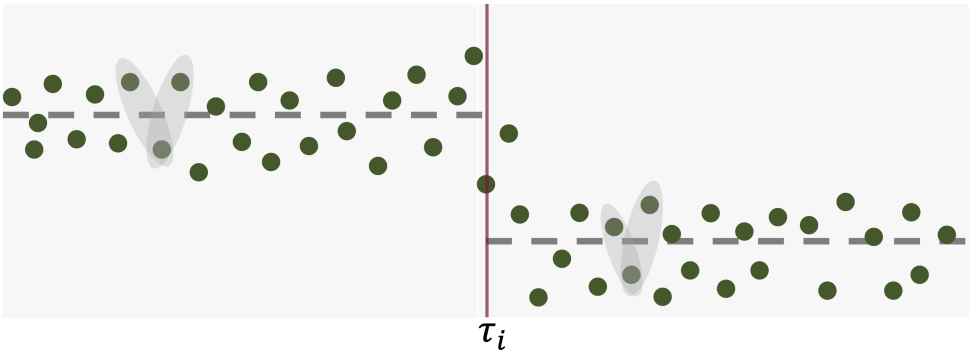
$$\begin{cases} H_0: & \text{data segment stays around a constant value disturbed by random noise} \\ H_1: & \text{data segment experiences a fair amount of changes} \end{cases}$$

The hypothesis test is evaluated through a Kolmogorov-Smirnov statistical test. Let us first define a segment of data as $x(k : k + N_s - 1)$, being N_s the length of this segment. Notice that every data segment is located between two consecutive change-points and, therefore, can also be represented as $x(\tau_i : \tau_{i+1} - 1)$. A mean-crossing data point t_c can then be defined as an instant of time in which two consecutive data points crossed the mean value of the data segment (from a higher to a lower value compared to the mean or the other way around). A mean-crossing instant must, therefore, satisfy the following equation (WANG *et al.*, 2018):

$$(x(t_c) - \bar{x})(x(t_c + 1) - \bar{x}) \leq 0, \text{ for } k \leq t_c \leq k + N_s - 1. \tag{3.57}$$

In Figure 5 one can find a visual example of mean-crossings.

Figure 5: Examples of mean-crossings. The green dots are data points, τ_i is a change-point separating two consecutive segments, the dashed grey lines represent the mean value of each segment and the ellipses highlight mean-crossing examples.



Source: Author's own development.

As explained in (WANG *et al.*, 2018), if the null hypothesis is true, then the time interval $T_c = t_{c+1} - t_c$ between two consecutive mean-crossing points experiences an exponential distribution. That is precisely the objective of the non-parametric Kolmogorov-Smirnov test: to evaluate whether the distribution of T_c can be considered exponential. If the test fails within a statistical significance level of α, then one can consider that the interval $x(k : k + N_s - 1)$ experiences a fair amount of change with a probability α of committing a mistake.

The Kolmogorov-Smirnov test used by the author is proposed in (LILLIEFORS, 1969)

and it is useful for exponential distributions with unknown mean. This test is also known as the Lilliefors test. The test consists of comparing the difference statistic below:

$$D_t = \max_T \mid F(T, \hat{\lambda}) - \hat{F}(T) \mid, \tag{3.58}$$

with a critical value defined by the desired significance level α, where $F(T, \hat{\lambda})$ is the cumulative distribution function of an exponential random variable, computed as (WANG *et al.*, 2018):

$$F(T, \hat{\lambda}) = \begin{cases} 1 - e^{\hat{\lambda}T}, & T \geq 0 \\ 0, & T < 0 \end{cases}. \tag{3.59}$$

Notice that $\hat{\lambda} = \frac{1}{\mu}$, where the mean value μ of T can be defined as

$$\mu = \frac{1}{L-1} \sum_{l=1}^{L-1} T_c, \tag{3.60}$$

being L the number of mean-crossing points in a given segment. The $\hat{F}(T)$ function is the estimated cumulative distribution of the statistic T, which can be calculated as

$$\begin{aligned} \hat{F}(T) &= \frac{1}{L-1} \sum_{l=1}^{L-1} a(T_c, T) \\ a(T_c, T) &= \begin{cases} 1, & T_c \leq T \\ 0, & T_c > T \end{cases} \end{aligned}. \tag{3.61}$$

A table of critical values for D_t can then be used in such a way that, if D_t is higher than the critical value, the null hypothesis can be rejected with probability α of making a mistake. Finally, in (WANG *et al.*, 2018) it is defined an "indicating sequence" $I_X(k : k + N_s - 1)$ to record if the interval is suitable for process identification or not. Therefore, in the case D_t is higher than a given critical value and the null hypothesis is rejected, $I_X(k : k + N_s - 1)$ receives a value of 1, otherwise it will receive a value of 0.

3.3.3.3 Check for Magnitude Differences Between Intervals

The third step aims to verify if two consecutive intervals with indicating sequences I_X of 0 experienced a significant magnitude difference between each other. This is done with a simple hypothesis test comparing two means, as defined below

$$\begin{cases} H_0: & \mu(x(k + N_s^1 : k + N_s^1 + N_s^2 - 1)) - \mu(x(k : k + N_s^1 - 1)) \leq \Delta \\ H_1: & \mu(x(k + N_s^1 : k + N_s^1 + N_s^2 - 1)) - \mu(x(k : k + N_s^1 - 1)) > \Delta \end{cases},$$

where N_s^1 is the sample length of the first interval and N_s^2 is the sample length of the interval that follows the first one.

Considering a value of Δ for the mean difference, the hypothesis test can be performed with a t-student test, where the normalized statistic is defined as

$$z_{N_s^2,N_s^1} = \frac{(\bar{x}_{N_s^2} - \bar{x}_{N_s^1}) - \Delta}{\sqrt{\left(\frac{S^2_{N_s^2}}{N_s^2}\right) + \left(\frac{S^2_{N_s^1}}{N_s^1}\right)}}, \tag{3.62}$$

where $\bar{x}_{N_s^1}$ and $\bar{x}_{N_s^2}$ are the mean values of the intervals $x(k + N_s^1 : k + N_s^1 + N_s^2 - 1)$ and $x(k : k + N_s^1 - 1)$, respectively, $S^2_{N_s^2}$ and $S^2_{N_s^1}$ are their sample variance and N_s^2 and N_s^1 the length of each interval. The mean and sample variance can be calculated as

$$\bar{x} = \tfrac{1}{N_s} \textstyle\sum_{t=k}^{k+N_s-1} x(t) \tag{3.63}$$

$$S^2 = \tfrac{1}{N_s-1} \textstyle\sum_{t=k}^{k+N_s-1} (x(t) - \bar{x})^2. \tag{3.64}$$

In this case, the critical value for the hypothesis test is defined as $t_{d,\alpha}$, where d is the degree of freedom defined as follows:

$$\begin{aligned} d &= \frac{(w_{N_s^1} + w_{N_s^2})^2}{\frac{w^2_{N_s^1}}{(N_s^1+1)} + \frac{w^2_{N_s^2}}{(N_s^2+1)}} - 2 \\ w_{N_s^1} &= \frac{S^2_{N_s^1}}{N_s^1},\ w_{N_s^2} = \frac{S^2_{N_s^2}}{N_s^2}. \end{aligned} \tag{3.65}$$

If the null hypothesis is rejected, then the authors suggest changing the indicating sequence of the second interval $x(k + N_s^1 : k + N_s^1 + N_s^2 - 1)$ from 0 to 1, *i.e.*, $I_X(k + N_s^1 : k + N_s^1 + N_s^2 - 1) = 1$. The authors also suggest as an estimative value for the difference Δ three times the standard deviation of the signal in steady-state condition.

3.3.3.4 Unifying Input and Ouput Segments

The final step of the algorithm is to unify data segments from the input and the output. If we consider I_u as the indicating sequence coming from the input signal and I_y as an indicating sequence coming from the output signal, then the resulting sequence should be defined as $I = I_u \cup I_y$. Finally, let us consider segments $I(t)$ as indexes where the unified indicating sequence has consecutive values of 1. Then, one should take all the resulting segments $I(t)$ from the unified indicating sequence and verify if there is at least one input segment $I_u(t)$ in $I(t)$ that is different from 0. If all of them are zero, then $I(t)$ should not be considered, *i.e.*, $I(t)$ is set to 0.

To better exemplify this scenario, let us imagine we have the following indicating sequences

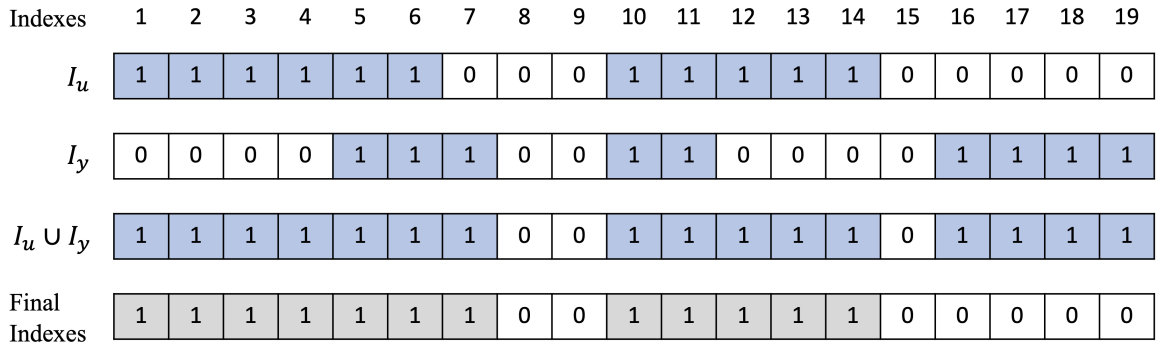
$$I_u = \{1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0\}$$
$$I_y = \{0, 0, 0, 0, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1\}.$$

Then, the resulting unified indicating sequence would be the following:

$$I_u \cup I_y = \{1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1\}.$$

Considering that a segment is defined by consecutive values of 1, I_u contains 2 segments ($I_u(1:6)$ and $I_u(10:14)$), I_y contains 3 segments ($I_y(5:7)$, $I_y(10:11)$ and $I_y(16:19)$), and, naturally, the unified indicating sequence also produces 3 segments ($(I_u \cup I_y)(1:7)$, $(I_u \cup I_y)(10:14)$ and $(I_u \cup I_y)(16:19)$). Finally, one must verify if the unified segments contains at least one input segment in it. In this case, it is clear that $I_u(1:6) \subset (I_u \cup I_y)(1:7)$ and that $I_u(10:14) \subset (I_u \cup I_y)(10:14)$. Therefore, these two segments would be considered as final segments. On the other hand, there is no input segment contained in $(I_u \cup I_y)(16:19)$ and, therefore, their values are set to 0, not being considered as a final segment. Figure 6 also illustrates this procedure.

Figure 6: Illustration of the unification step of the Statistical Method.



Source: Author's own development.

3.3.4 A Multivariable Approach

Two main works presented in the literature overview in Section 1.2 deal with the problem of finding intervals suitable for system identification of multivariable systems. The work in (PATEL, 2016) introduces the problem for open-loop system identification. The multivariable approach for closed-loop system identification is far more complex, and considerations similar to the ones presented in Item 3.3.2.7 must be considered. A formulation for this problem is done in (ARENGAS; KROLL, 2017a), although it is not

explored in this dissertation. A detailed review of the methodology presented in (PATEL, 2016) is presented in this subsection.

3.3.4.1 Extension from SISO to MIMO Systems

The main idea behind the extension from SISO to MIMO systems proposed in (PATEL, 2016) is to consider a multivariable system as a combination of Multi-Input Single-Output (MISO) systems, as described in Section 2.8. With this approach, the mining algorithm can be applied independently for each output system. Notice, however, that this strategy can only be applied to open-loop system identification, *i.e.*, when the input variables are not manipulated by a controller, but by the user.

Because there are multiple inputs associated with a single output, a reformulation of the regressor matrix is necessary. Let us consider a data interval of length N_s as an example, *i.e.*, $u_i(k : k + N_s - 1)$ is an input sequence in this interval. Let us also consider a regressor vector associated with input i and output j at instant k as $\boldsymbol{\psi}^{i,j}(k) = [\psi_1^{i,j}(k) \;\; \psi_2^{i,j}(k) \;\; \dots \;\; \psi_{n_\theta}^{i,j}(k)]$ where n_θ is the regressor order. The regressor matrix for this input and output pair can then be defined as

$$\boldsymbol{\Psi}_{N_s}^{i,j} = \begin{bmatrix} \psi_1^{i,j}(k) & \psi_2^{i,j}(k) & \cdots & \psi_{n_\theta}^{i,j}(k) \\ \vdots & \vdots & \ddots & \vdots \\ \psi_1^{i,j}(k+N_s-1) & \psi_2^{i,j}(k+N_s-1) & \cdots & \psi_{n_\theta}^{i,j}(k+N_s-1) \end{bmatrix}, \tag{3.66}$$

where $\boldsymbol{\Psi}_{N_s}^{i,j} \in \mathbb{R}^{N_s-1 \times n_\theta}$. Let us consider that the system contains n_u inputs. Therefore, the MISO regressor matrix for output j can be defined as (PATEL, 2016):

$$\boldsymbol{S}^j = [\boldsymbol{\Psi}_{N_s}^{1,j} \;\; \boldsymbol{\Psi}_{N_s}^{2,j} \;\; \dots \;\; \boldsymbol{\Psi}_{N_s}^{n_u,j}]. \tag{3.67}$$

Considering all the set-points associated with a given output j, the following condition number vector can be defined for this MISO system (PATEL, 2016):

$$\boldsymbol{C}_j = [\kappa_2(\hat{\boldsymbol{R}}_{N_s}^{1,j}) \;\; \kappa_2(\hat{\boldsymbol{R}}_{N_s}^{2,j}) \;\; \dots \;\; \kappa_2(\hat{\boldsymbol{R}}_{N_s}^{n_u,j})]^T. \tag{3.68}$$

The interval of length N_s is then considered a candidate for system identification if at least one of the elements in vector $\boldsymbol{C}_j$ is smaller than a given threshold l_{c_j}, or equivalently, if $min\, \boldsymbol{C}_j < l_{c_j}$. The reason why only one input is required to satisfy the numerical conditioning hypothesis is that, as proposed in (PATEL, 2016), one could use multiple intervals to obtain the final model, as explained in the next item. As an alternative formulation, but now considering the work in (RIBEIRO; AGUIRRE, 2015), an effective

rank vector could be formulated for every MISO system corresponding to output j as shown below:

$$\boldsymbol{R}_{ef_j} = \left[r_{ef}(\hat{\boldsymbol{R}}_{N_s}^{1,j}) \quad r_{ef}(\hat{\boldsymbol{R}}_{N_s}^{2,j}) \quad \cdots \quad r_{ef}(\hat{\boldsymbol{R}}_{N_s}^{n_u,j}). \right] \tag{3.69}$$

Similar to the condition number test, a formulation can be made such that if at least one input variable in $\boldsymbol{R}_{ef_j}$ resulted in an effective rank higher than a given threshold l_{e_j}, or equivalently, if $max\, \boldsymbol{R}_{ef_j} > l_{e_j}$, the interval is still a potential interval.

Finally, each interval that satisfies one of the above conditions can be subjected to the correlation test with the output signal j using either the chi-squared test or the scalar cross-correlation scalar metric. If at least one input-output pair meets both the numerical condition and the cross-correlation criteria, the interval can be stored for all the input and output signals and considered useful for system identification.

Notice that for the particular case of the AR structure, the computation of the effective rank for each input-output pair is not required, since this structure only depends on the output signal. In other words, $\boldsymbol{R}_{ef_j}$ is a scalar value for each output j.

3.3.4.2 System Identification

In order to obtain a model of the process based on the resulting selected intervals, one must consider that a dynamic model for a given output j can only be correctly obtained if all the input variables provide a fair amount of excitation. However, the multivariable extension proposed in (PATEL, 2016) only requires that at least one input-output pair meets the proposed criteria. Consequently, one can easily find intervals where not all input variables are persistently exciting.

The reason why this condition is proposed this way is to make the algorithm less restrictive. In fact, to get around this problem, in (PATEL, 2016) it is proposed the use of several intervals for obtaining the final model. Let us assume, as an example, that we have a 3×3 system and that only two intervals were obtained: Δ_1^{uy} and Δ_2^{uy}. Interval Δ_1^{uy} satifies the numerical conditioning and the correlation criteria for input $u_1(k)$ only, while Δ_2^{uy} satisfies both criteria for inputs $u_2(k)$ and $u_3(k)$. If we take $y_1(k)$ as an example, a model for that output could be obtained using both Δ_1^{uy} and Δ_2^{uy} as follows (PATEL, 2016):

$$y_1^{it_1}(k) = G_{11}(q)u_1(k) + H_1(q)v_1(k) \tag{3.70}$$

$$y_1^{it_2}(k) = G_{12}(q)u_2(k) + G_{13}(q)u_3(q) + H_2(q)v_2(k) \tag{3.71}$$

$$y(k) = G_{11}(q)u_1(k) + G_{12}(q)u_2(k) + G_{13}(q)u_3(q) + H(q)v(k). \tag{3.72}$$

One must notice that a model addition is being performed in Equation 3.72. One of the ways suggested in (PATEL, 2016) to perform this addition is by first estimating the model parameters (Equations 3.70 and 3.71) and then combining the resulting transfer functions in a single transfer function (Equation 3.72). This procedure is straightforward if the set-points in Equations 3.70 and 3.71 do not overlap, which is the approach considered in this dissertation. However, it is not straightforward to combine the disturbance models $H_1(q)$ and $H_2(q)$ when they result in different structures or when they result in the same structure but with different model orders. To handle this problem, in this dissertation the disturbance model with the highest gain is the one considered as $H(q)$ in final model.

3.3.5 An Improved Method Using Entropy

In (SHARDT; SHAH, 2014) it is argued that a process model can potentially change with time and, therefore, each segment of data obtained through a segmentation algorithm can represent the process with different characteristics. For this reason, the authors define three different situations:

a) **oversegmentation**: when we obtain more models than the real amount of models represented by the dataset;
b) **undersegmentation**: when fewer models are identified;
c) **exact Case**: when the exact number of models is identified.

The work proposed in (SHARDT; HUANG, 2013b) shows that the entropy of a time-series signal can be used to detect changes in the process. Moreover, a differential entropy can be used as a metric to determine when the model is not an accurate representation of the system anymore. The analysis is made based on the fact that if two data windows contain a similar value of entropy, then they are very likely coming from the same system dynamics. This exact idea is used in (SHARDT; SHAH, 2014) as an additional step for improving the works proposed in (PERETZKI *et al.*, 2011) and (SHARDT; HUANG, 2013a).

As described by (APPLEBAUM, 2008) and (SHARDT; HUANG, 2013b), entropy is a term that comes from thermodynamics and that, in Information Theory, is used to measure the amount of information of a sequence of random variables. Therefore, the definition of entropy for stochastic signals is reproduced below:

$$H = -\sum_{x \in \mathbb{X}} p(x) \log p(x), \tag{3.73}$$

where H is the signal entropy, x is a single occurrence of the space $\mathbb{X}$ of all possible realizations of a random variable and $p(x)$ is the probability of x occuring in that space (SHARDT; HUANG, 2013b).

In (SHARDT; HUANG, 2013b) it is shown, however, that for time-series and spatial signals the entropy can be reformulated as

$$H_t = \log\left(\frac{L_t}{N_s}\right) \tag{3.74}$$

$$L_t = \sum_{k=1}^{N_s} |x_k - x_{k-1}|\,, \tag{3.75}$$

where L_t is the degree of tortuosity in the signal.

Finally, the differential entropy is then defined in (SHARDT; HUANG, 2013b) as

$$\Delta H_{t,x} = H_{t,y} - H_{t,u}, \tag{3.76}$$

where $H_{t,y}$ is the entropy of the process output signal and $H_{t,u}$ is the entropy of the process input signal.

A new step is then proposed in (SHARDT; SHAH, 2014) as an additional step for the methods proposed in (PERETZKI *et al.*, 2011) and (SHARDT; HUANG, 2013a). This step consists of taking every resulting segment $\Delta_i = [\tau_{init,i}, \tau_{end,i}]$ and computing the time-series entropy for these regions. The differential entropy is then used to compare adjacent regions in such a way that if the differential entropy of two consecutive regions are close to each other within a given threshold, these regions must be unified and treated as a single interval.

4 METHODOLOGY

In this chapter, the methodology adopted in this dissertation is presented. As one can notice from the state-of-the-art review in Section 3.3, different algorithmic structures are presented in the studied literature. While some solutions go through the entire dataset in a single pass — such as the recursive approach proposed in (PERETZKI *et al.*, 2011) and in (BITTENCOURT *et al.*, 2015), or the sliding-window approach presented in (ARENGAS; KROLL, 2017a) and in (ARENGAS; KROLL, 2017b) — others initially divide the data in regions of interest, and these regions are sequentially evaluated as suitable or not for obtaining a model of the process. The later case occurs, for example, in (RIBEIRO; AGUIRRE, 2015), in (PATEL, 2016) and in (WANG *et al.*, 2018) and constitutes the main structure adopted in the methodology presented in this chapter.

4.1 Structure of the Algorithms

The main algorithmic structure adopted in this dissertation is inspired in (RIBEIRO; AGUIRRE, 2015), (PATEL, 2016) and (WANG *et al.*, 2018) and can be decomposed in the following steps:

a) **Step 1:** preprocess the dataset;

b) **Step 2:** define potential regions of excitation;

c) **Step 3:** evaluate each initial region as suitable or not for system identification;

d) **Step 4**: store the final mined intervals.

For some of the mining strategies defined in this chapter, a further step can be included between steps 3 and 4 described above, which consists of incrementing the potential regions that are actually suitable for system identification. The reason for that is because in the recursive and sliding-window approaches described in Section 3.3, final intervals are

defined and incremented online until they no longer meet the selection criteria adopted. Consequently, the resulting intervals are optimized in terms of their resulting length. In order to provide this flexibility to the adopted algorithm in this dissertation, it is optional to increase the resulting potential intervals, with an increment step of w_{ic}, while they satisfy the selection criteria adopted. When the potential intervals are kept in their original format, the algorithm will be here considered in **stationary** mode. Otherwise, it will be considered in an **incremental** state.

There are a few reasons behind the choice of the described structure, which are listed below:

a) the adopted algorithms can be structured in the same framework and one can combine different aspects of each algorithm; and

b) breaking the data in steps gives one the possibility of making a visual analysis to support the choice of parameters.

4.1.1 Outline of the Numerical Algorithms

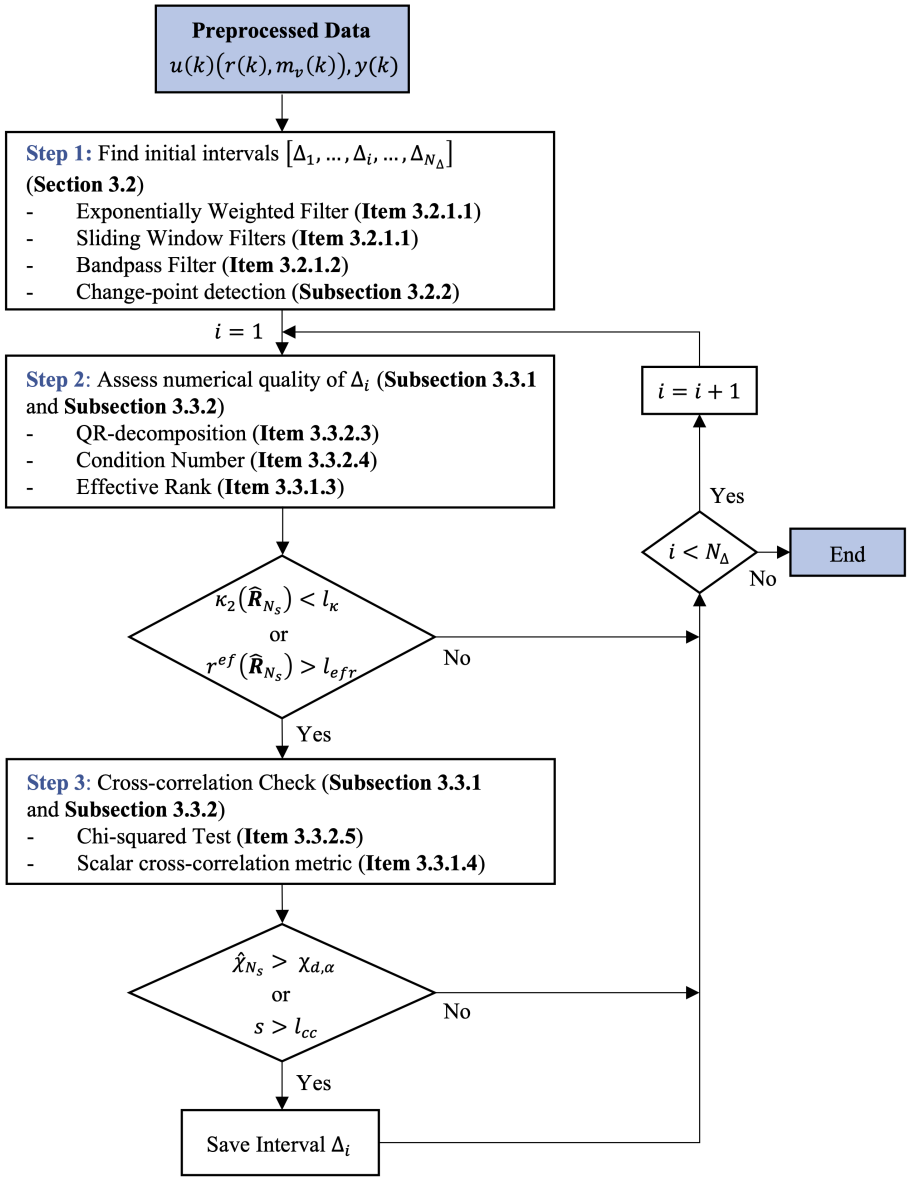
4.1.1.1 Singe-Input Single-Output Case

The different mining strategies presented in Subsections 3.3.1 and 3.3.2 are here combined in a single algorithm, which is defined in Figure 7. Notice that the algorithm is divided as described in Section 4.1 and that one could choose which criteria to use in each step. Therefore, one could, for example, obtain the initial intervals through the Exponentially Weighted filter described in Item 3.2.1.1 or through the Bandpass filter proposed in (PATEL, 2016). In the same fashion, one could use the condition number or the effective rank to evaluate each potential interval. Moreover, the algorithm can be applied for both open and closed-loop identification of SISO systems. The way it can be applied in each scenario is the following:

a) **Open-loop System Identification:** this is the simplest case and all steps of the algorithm (Steps 1-3) are performed with the provided input and output variables.

b) **Closed-loop System Identification:** Steps 1 and 2 of the algorithm must be performed with the set-point $r(k)$ of the controller, while Step 3 must be executed with the manipulated variable $m_v(k)$. The reasons for that is explaiened in Item 3.3.2.7.

Figure 7: Outline of the Single-Input Single-Output Numerical Algorithm.



Source: Adapted from (PERETZKI *et al.*, 2011) and (PATEL, 2016).

Finally, as explained in Section 4.1, the algorithm can also be used in an incremental way. In this case, every potential interval approved in Steps 2 and 3 can be iteratively incremented by w_{ic} indexes as long as they continue to satisfy both step criteria. As soon as one of the two criteria is no longer satisfied, or if the incremented interval reaches the next consecutive interval, the running segment is stored and the same procedure continues until all potential intervals are evaluated.

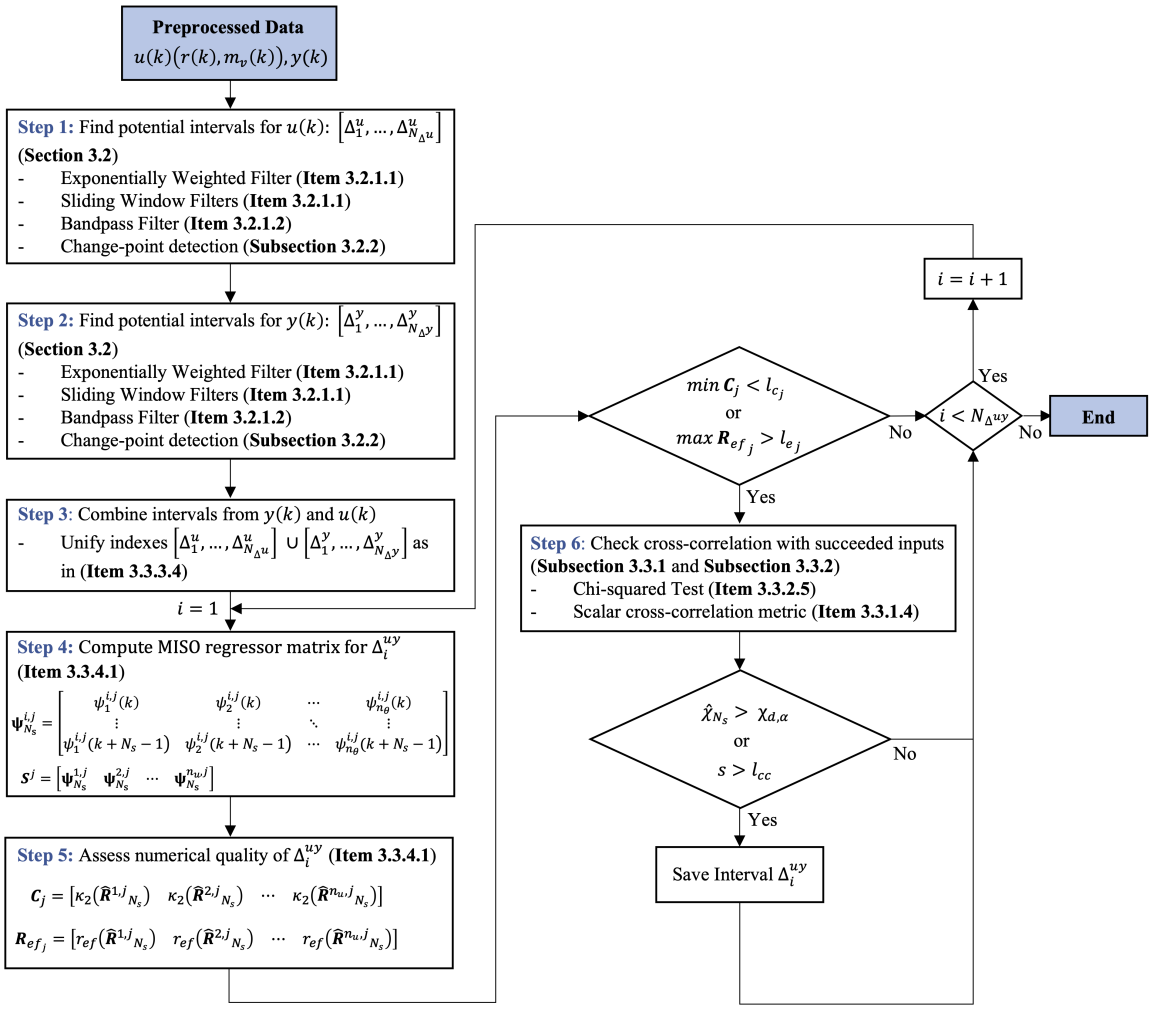
In this dissertation, all the possible methods of obtaining potential intervals shown in Step 1 are first applied separately in the development chapter, in order to elucidate how they work. Then, the algorithm provided in this item is applied to SISO simulated data, where each metric in Steps 2 and 3 are computed and analyzed. Moreover, both

the incremental and the stationary mode are exemplified in Section 5.3.

4.1.1.2 Multiple-Input Multiple-Output Case

An extrapolation of the numerical algorithm to multivariable systems is performed based on the work in (PATEL, 2016). As explained in Subsection 3.3.4, in this dissertation a multivariable extension of the work in (RIBEIRO; AGUIRRE, 2015) is proposed, which is included in the same framework as the work in (PATEL, 2016). The algorithm here presented can only be applied to open-loop system identification, following the definition made in Section 2.7. Notice that the multivariable case for closed-loop system identification is not implemented in this work. The outline of the numerical multivariable algorithm can be seen in Figure 8.

Figure 8: Outline of the Multiple-Input Multiple-Output Numerical Algorithm.



Source: Adapted from (PERETZKI *et al.*, 2011) and (PATEL, 2016).

It is important to mention that if the system contains only one input and one output, this algorithm is exactly reduced to the algorithm in Figure 7 considering the open-loop

system identification scenario. Moreover, this algorithm can also be implemented in an incremental fashion, in which case Steps 4, 5 and 6 would be repeated as explained in Item 4.1.1.1. Finally, one must have in mind that, although the algorithm requires that at least one input-output pair satisfies the method validation requirements, one could be more restrictive in this condition. As an example, one could require that, for a particular output, all input variables must satisfy the method criteria. This scenario is explored in Section 5.5 with the Petrochemical Furnace dataset, where a more restrictive condition is imposed.

4.1.2 Outline of the Statistical Algorithms

4.1.2.1 Singe-Input Single-Output Case

The statistical algorithm proposed in (WANG *et al.*, 2018) and described in Subsection 3.3.3 is implemented, for SISO systems, with the same outline presented in its original work, but with a few notation changes, as shown in Figure 9.

It is important to mention that, for the same reason explained in Item 3.3.2.7, if this algorithm is applied to the closed-loop system identification scenario, the set-point $r(k)$ must be the one considered in the diagram in Figure 9.

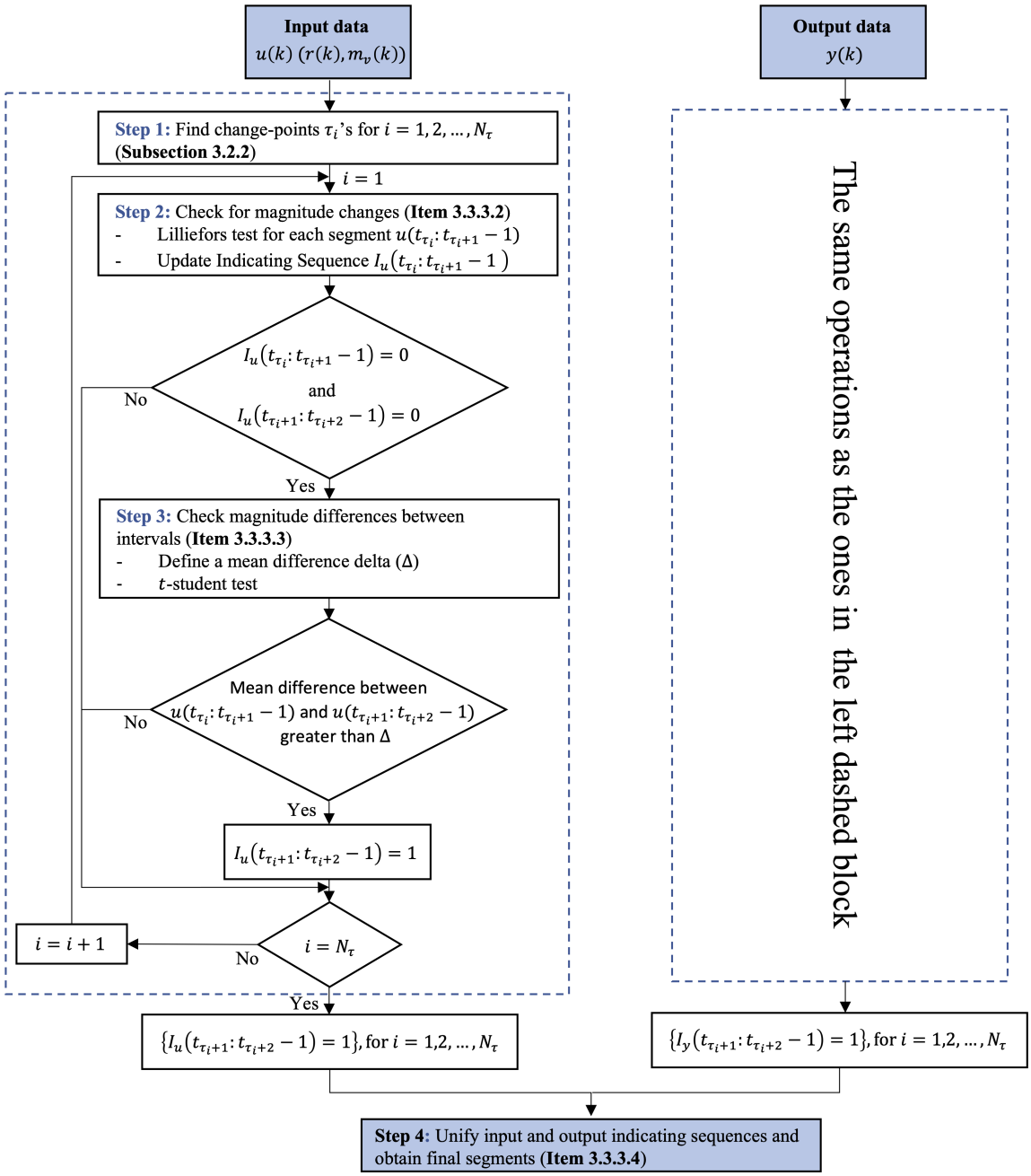
4.1.2.2 Multiple-Input Multiple-Output Case

A multivariable extension to the statistical algorithm proposed in (WANG *et al.*, 2018) is here presented inspired in the work in (PATEL, 2016). The main idea behind the extrapolation is to check if at least one input and one output meet the evaluation criteria, which in this case corresponds to the statistical tests in Steps 2 and 3. Therefore, the main change occurs in Step 4 of the diagram in Figure 9. The extrapolation can be summarized as follows:

a) Steps 1 to 3 are applied to every input and output of the multivariable system. For a system with n_u inputs and n_y outputs, one would obtain $n_u + n_y$ different indicating sequences, one for each signal;

b) Step 4 is modified as follows: for each output variable that defines a MISO system, the corresponding output indicating sequence I_{y_j} is unified with **every** input indicating sequence I_{u_i} of the system. Therefore, assuming we have a $n_u \times n_y$ system,

Figure 9: Outline of the Statistical Algorithm.



Source: Adapted from (WANG *et al.*, 2018).

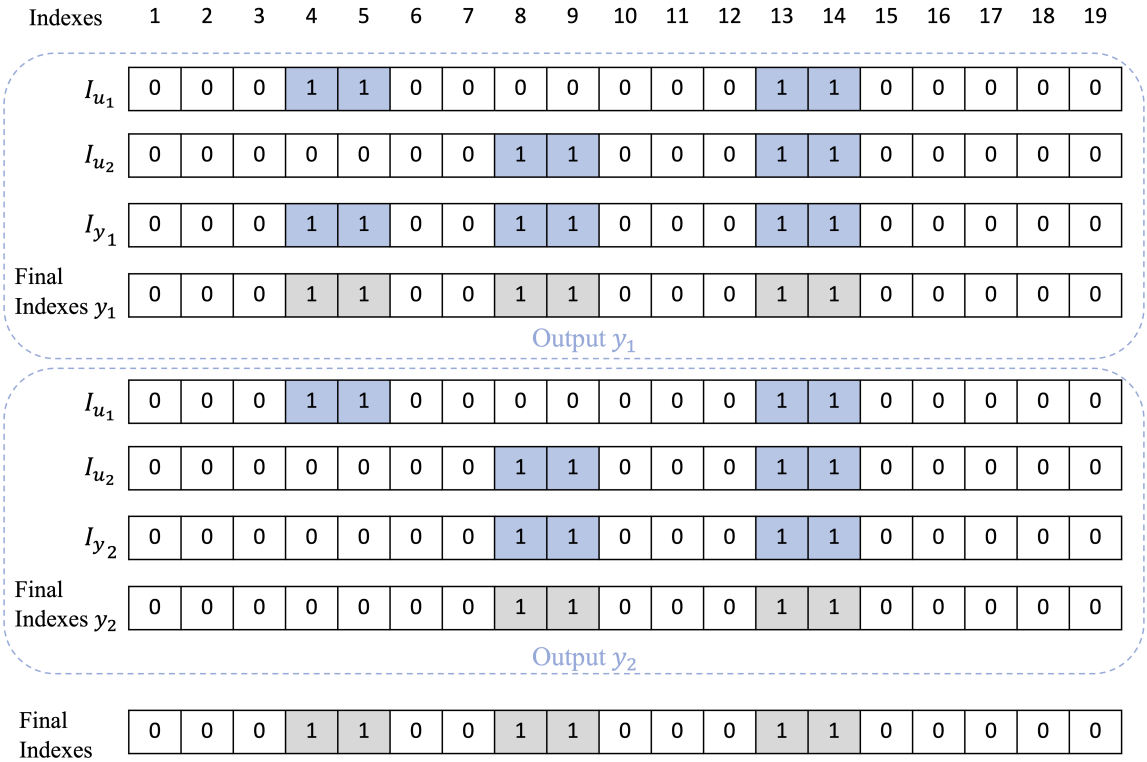
we would obtain the following unified indicating sequences, one for each output:

$$I^1 = I_{u_1} \cup I_{u_2} \cup \cdots I_{u_{n_u}} \cup I_{y_1}$$
$$I^2 = I_{u_1} \cup I_{u_2} \cup \cdots I_{u_{n_u}} \cup I_{y_2}$$
$$\vdots$$
$$I^j = I_{u_1} \cup I_{u_2} \cup \cdots I_{u_{n_u}} \cup I_{y_j}$$
$$\vdots$$
$$I^{n_y} = I_{u_1} \cup I_{u_2} \cup \cdots I_{u_{n_u}} \cup I_{y_{n_y}}$$

Finally, the procedure described in Item 3.3.3.4 is applied to every MISO system, but considering the idea proposed in (PATEL, 2016), *i.e.*, for each MISO system, if **at least one** input sequence is contained in the unified sequence of the corresponding output variable (I^j), the segment is saved. Then, as a final step, the saved segments from every MISO system are unified, resulting in the final segments. Notice that, also here, one could be more restrictive and require, for example, that all inputs in a given MISO system must experience a fair amount of excitation, instead of considering only a single input.

To better exemplify how Step 4 is modified, an example is given in Figure 10 for a hypothetical 2×2 system. From this example, one can notice that the input signal $u_1(4:5)$ was only able to cause significant changes in the output $y_1(4:5)$, but not in $y_2(4:5)$. However, this is enough to consider indexes 4 and 5 in the final interval. On the other hand, if one requires, for example, the restrictive condition that all inputs, for all MISO systems, must satisfy the statistical criteria, only indexes 13 and 14 would be considered.

Figure 10: Multivariable extension of the statistical method: an example on how Step 4 is modified.



Source: Author's own development.

4.2 Structure of the Solution and Hypothesis

To address the problem of the difficulty of obtaining informative data to identify process models, the algorithms presented in the present methodology are used in some different scenarios, all representing real applications in industry:

a) **SISO systems**: both algorithms in Figure 7 and in Figure 9 are applied to a closed-loop water tank dataset produced through simulation. Intervals suitable for system identification are then obtained to address the closed-loop identification scenario;

b) **MIMO systems**: both algorithms in Figure 8 and in Item 4.1.2.2 are initially applied to a simulated dataset from a 2×2 distillation column, with the system operating in open-loop mode. Then, the algorithm in Figure 7 is applied to a real petrochemical furnace dataset, elucidating the effectiveness of the solution in a real and challenging situation.

   In the later case, the system is operating in closed-loop mode. However, the ultimate goal is to obtain intervals that could be used to design a model predictive controller, in such a way that the system identification is performed with the set-point and the output variables. Therefore, one can understand this process as an open-loop system identification, as described in Section 2.7. For this reason, the multivariable algorithm in Figure 8 can be applied to this scenario, but using the set-point as the input to the algorithm. Figure 2 represents this case.

The following hypotheses guide the implementation and the achievement of the results:

Hypothesis 4.1. *The processes being studied are considered invariant in time during the data collection;*

Hypothesis 4.2. *It is assumed that the dynamics of the processes being studied can be modeled linearly.*

Hypothesis 4.3. *It is assumed that if the set-point is persistently exciting, not only one can perform a closed-loop identification, as described in Item 3.3.2.7, but also perform an open-loop identification with the set-point as input, which, in other words, consists in estimating a model with the set-point and the output variables, as described in Section 2.7.*

From Hypothesis 4.1, if multiple segments of data are obtained, they are here used to perform cross-validation. Therefore, if $S_\Delta = (\Delta_1,\ \Delta_2,\ ...,\ \Delta_{N_s})$ are the resulting

segments for a given algorithm, then Δ_1 is used to fit model $\mathcal{M}_1$, while Δ_2, ..., Δ_{N_s} are used to validate the obtained model. In the same fashion, Δ_2 is used to fit model $\mathcal{M}_2$, while Δ_1, Δ_3, ..., Δ_{N_s} are used to validate the model. This procedure goes on until N_s different models are obtained.

4.3 Code and Reproducibility

An open-source library was created as part of the solution for this dissertation and it is hosted in the author's Github [1] repository. The library is called "HDSIdent: Historical Data Segmentation for System Identification" and aims to not only provide open access to all the implementations made in this work, but also to allow full reproducibility of the results obtained through simulation data. In addition, the open-source framework allows the code to receive contributions from the scientific community, which is extremely encouraged by the author of this dissertation. Further information on use, installation, licensing and documentation can be found in the mentioned repository.

[1] <https://github.com/GiulioCMSanto/HDSIdent>

5 DEVELOPMENT

In this chapter, the methodology described in Chapter 4 is applied to different scenarios, following the solution proposed in Section 4.2.

5.1 An Introduction to the Datasets

As a first step, an introduction to the datasets that are used in this section is provided. Both simulated and real data are used, in such a way that simulation is used to support the ideal scenario and elucidate how the algorithms work, while real applications are performed based on massive data coming from a petrochemical furnace.

5.1.1 Single-Input Single-Output Water Tank

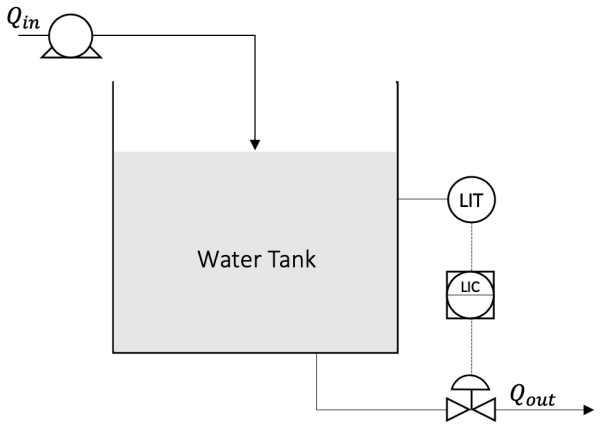
To simulate the Single-Input Single-Output case, a simple laboratory water tank is used. This example was directly extracted from (WANG *et al.*, 2018 apud YU *et al.*, 2011), from which the controller and the system transfer functions were adopted. The idea is to use the same simulation format adopted in (WANG *et al.*, 2018), in order to produce a very similar data and validate the statistical method by comparing the results with the ones presented in (WANG *et al.*, 2018). Moreover, this data is also used to apply the other algorithms developed in this dissertation, allowing one to compare the different methods presented. A generic P&ID for a water tank control system can be seen in Figure 11.

Notice that the set-point and the output variables are the tank level in centimeters and, in this case, the controller acts in a control valve. The transfer functions for both the process and the controller are, respectively (WANG *et al.*, 2018):

$$G(s) = \frac{6.6469}{241.37s + 1} \times \mathrm{e}^{-s} \tag{5.1}$$

$$C(s) = 2.0249 \times \left(1 + \frac{1}{233.94s}\right) \tag{5.2}$$

Figure 11: Generic Water Tank P&ID Control System.



Source: Author's own development.

The same simulation pattern presented in (WANG *et al.*, 2018) is used here with some inclusions, and consists of the following signals:

a) step responses with 1000 s duration and around a 20 cm water set-point;

b) low magnitude ramp responses;

c) step responses with 200 s duration and around 20 cm of water set-point;

d) high magnitude ramp responses;

e) sinusoidal response with 200 s period and around 30 cm of water set-point;

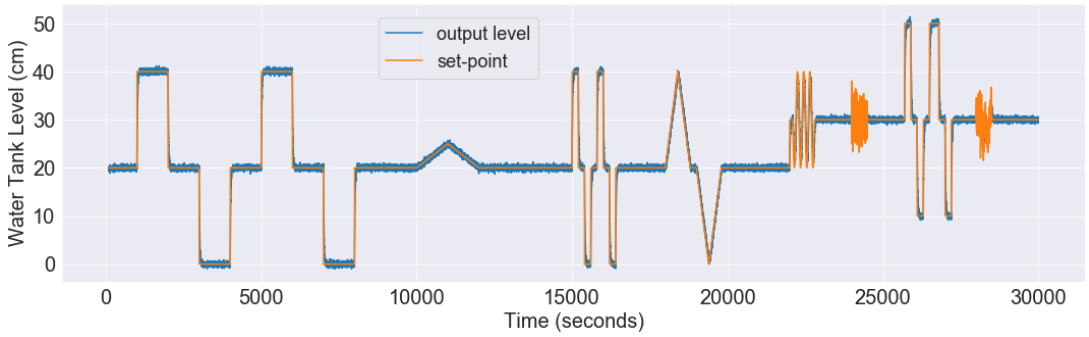
f) step responses with 200 s duration and around 30 cm of water set-point;

g) Gaussian white noise with 0 mean and variance of 5;

h) colored noise with transfer function $\frac{0.5z}{z-0.9}$ excited by gaussian white noise with 0 mean and variance of 5.

The resulting simulation data for both the set-point and the output variables can be seen in Figure 12

Figure 12: Water tank control system simulation: set-point and output variables.



Source: Author's own development.

5.1.2 Wood & Berry Distillation Column

The Wood & Berry distillation column is used as an example of a distillation process, which is widely used in chemical and petroleum industries to separate chemical components into fractions of more or less purity (BUCKLEY; LUYBEN; SHUNTA, 1985). A binary distillation column is one used to separate two components. In (WOOD; BERRY, 1973), it is studied the simultaneous control of the overhead (top) and the bottom compositions of a binary distillation column where the reflux and the steam flow rate are the manipulated variables. A feed flow variable is included as a disturbance.

The transfer function equations that describe this system are shown in Equations (5.3) - (5.4).

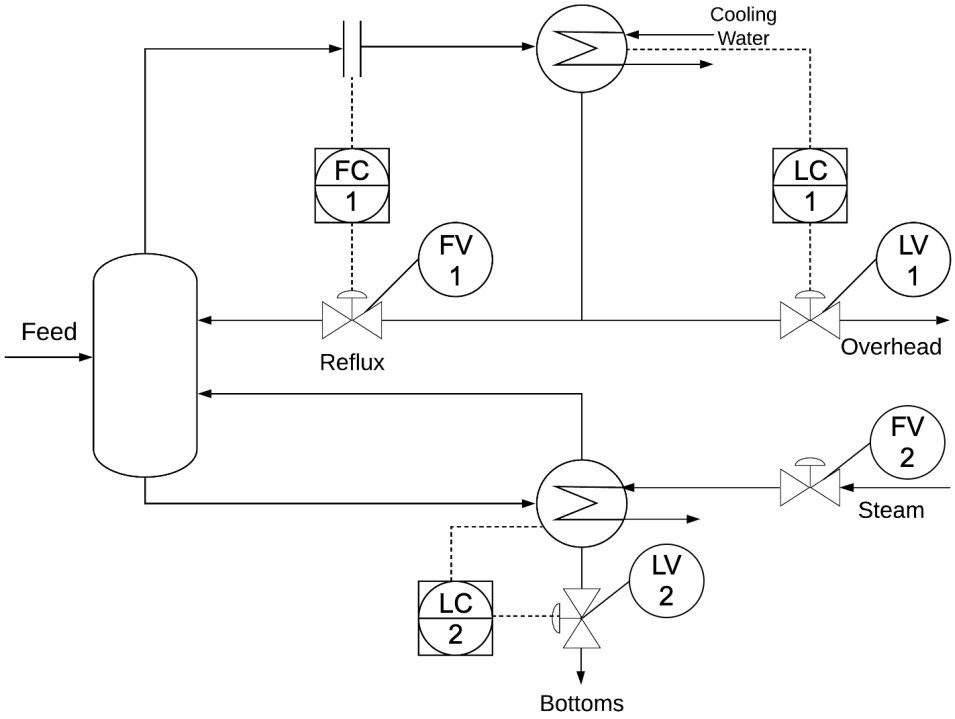
$$\boldsymbol{G}(s) = \begin{bmatrix} \frac{12.8 \cdot e^{-1s}}{16.7s+1} & \frac{-18.9s \cdot e^{-3s}}{21.0s+1} \\ \frac{6.6 \cdot e^{-7s}}{10.9s+1} & \frac{-19.4 \cdot e^{-3s}}{14.4s+1} \end{bmatrix} \tag{5.3}$$

$$\boldsymbol{H}(s) = \begin{bmatrix} \frac{3.8 \cdot e^{-8s}}{14.9s+1} \\ \frac{4.9 \cdot e^{-3s}}{13.2s+1} \end{bmatrix}^T \tag{5.4}$$

In a more generic way, this system can be written as $\boldsymbol{y}(k) = \boldsymbol{G}(s) \cdot \boldsymbol{u}(k) + \boldsymbol{H}(s) \cdot \boldsymbol{v}(k)$, where $\boldsymbol{G}(s)$ is the process transfer function matrix and $\boldsymbol{H}(s)$ is the disturbance transfer function matrix. Notice that $\boldsymbol{v}(k)$ is a diagonal matrix of Gaussian white noises, which are considered with zero mean and 0.01 variance. In the above matrices, the time lag and delay are given in minutes. A Piping and Instrumentation Diagram (P&ID) is represented in (JULIANI, 2017) and is adapted in Figure 13.

This model is simulated in open-loop mode applying step changes in the top (overhead) and bottom compositions, with initial values of 96% and 0.5%, respectively, following

Figure 13: Wood & Berry Distillation Column.



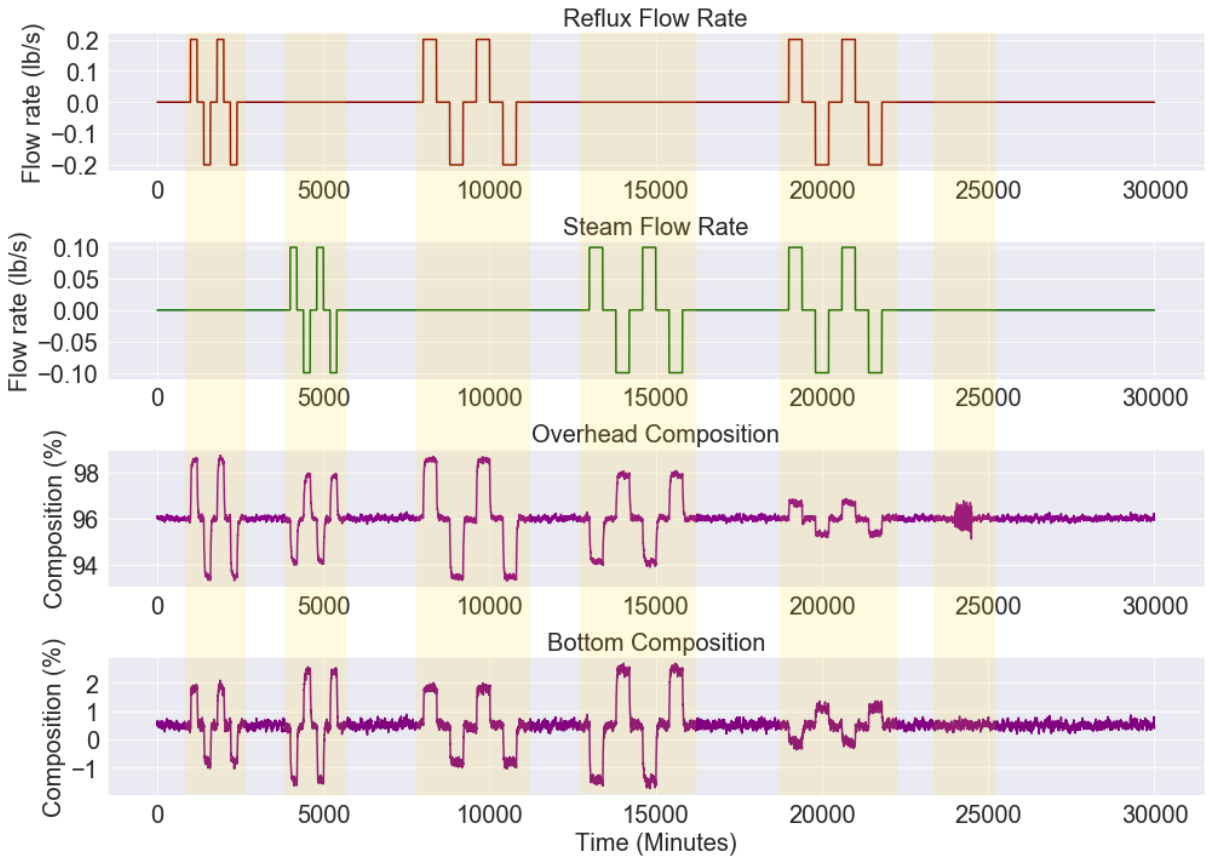
Source: Adapted from (JULIANI, 2017 apud WOOD; BERRY, 1973).

the work in (JULIANI, 2017). Moreover, this simulation data is used to exemplify the multivariable segmentation extension for open-loop system identification, which is based on the work in (PATEL, 2016). For this reason, the excitation signals were designed also based on the simulations proposed in (PATEL, 2016), allowing the comparison of the expected results. The simulation data can be seen in Figure 14.

5.1.3 Multivariable Petrochemical Furnace

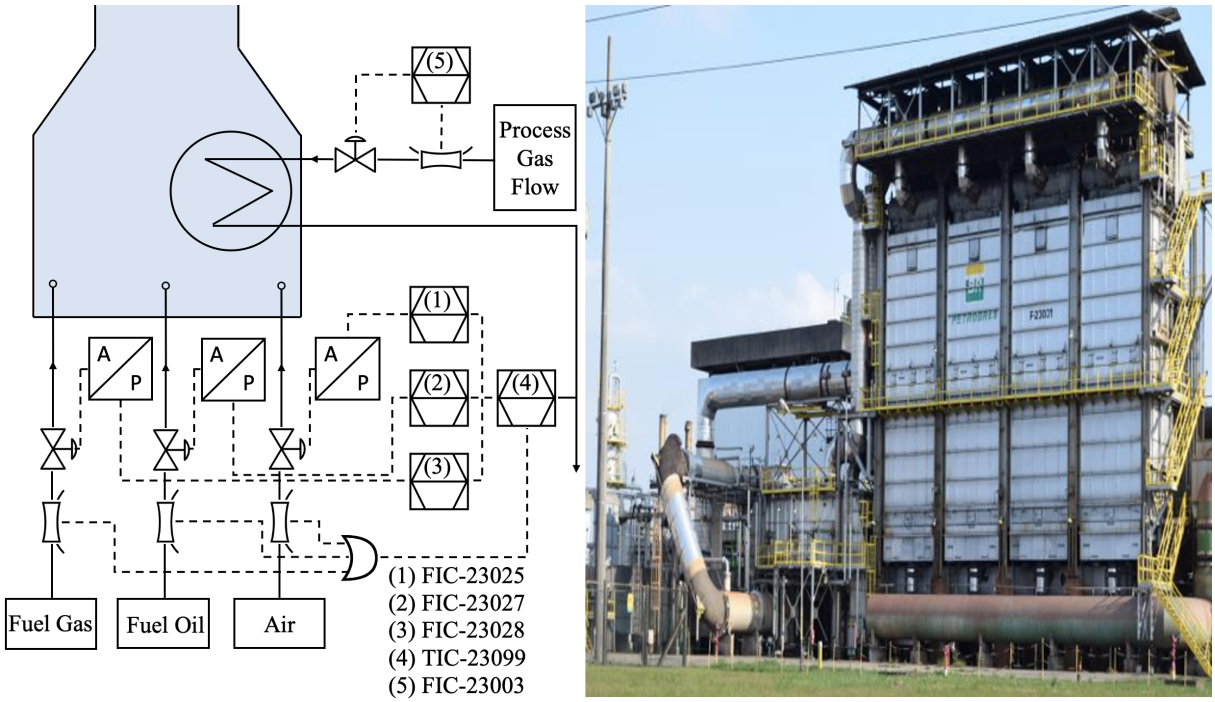
A dataset coming from a real industrial process is also used to validate the algorithms proposed in this dissertation. The dataset is obtained from a petrochemical furnace with 50MW of power from a nacional Brazilian petroleum company, being the same data adopted in (CHAVES, 2020). This furnace is a thermal energy generator used in the petroleum refining process and in the production of petroleum derivates. As described in (CHAVES; JULIANI; GARCIA, 2019), the furnace produces 34.2 millions of kcal/h from burning oil and fuel gas. Figure 15 shows an image of the physical installation of the furnace as well as its corresponding piping and instrumentation diagram (P&ID). This diagram is a modified version of the one presented in (CHAVES; JULIANI; GARCIA, 2019), while the physical installation image is the same found in (CHAVES; JULIANI; GARCIA, 2019).

Figure 14: Distillation column Data.



Source: Author's own development.

Figure 15: Petrochemical Furnace: physical installation and P&ID.



Source: Adapted from (CHAVES; JULIANI; GARCIA, 2019).

From the analysis of the P&ID, one can observe that the purpose of the petrochemical furnace is to maintain the output gas within a pre-defined temperature in order to allow it to be used in a pyrolysis reaction. Therefore, in the combustion chamber, the oil and the gas fuels react with air and release the thermal energy required for heating the gas coming from the discharge of a compressor. The temperature of the gas is then measured by a sensor and sent to a temperature controller. The control output acts in the oil, gas and air flow rate, changing the stoichiometric ratio of combustion. The operating temperature of the combustion chamber is around 1000°C, with the output temperature of the process gas being around 600°C. The process gas flow rate coming from the discharge of the compressor is also measured as a disturbance. The variables that are considered are summarized in Table 2.

Table 2: Petrochemical furnace variables.

Variable	**Description**	**Operating Condition**	**Unit**
TIC-23099	Process gas temperature	560-564	°C
PIC-23039	Fuel gas pressure in the header	4.2-5.2	kg/cm²
AIC-23001	Oxygen concentration in the stack	2.5-3.1	%
FIC-23027-SP	Fuel oil flow rate set-point	1000	kg/hour
FIC-23028-SP	Fuel gas flow rate set-point	2100	kg/hour
FIC-23025-SP	Combustion air flow rate set-point	54000	kg/hour

The data is divided into two blocks: the first one is composed by 1 month of data, corresponding to July 2018; the second one is composed of 7 sequential months, which goes from September 2019 to March 2020. Moreover, The dataset was collected in minutes and the original sampling rate was maintained. The total amount of rows (minutes) in the dataset is 339848, and it contains around 100Mb.

Finally, it is important to mention that this dataset is considered for open-loop system identification, in such a way that the set-point and the output variables are the ones of interest. Moreover, some of the variables considered, such as FIC-23025-SP, are fed by an optimizer. In theory, this fact allows one to reject the hypothesis that the set-points are independent from each other and from the output variable. However, one must consider that, most of the time, the optimizer is turned off and "abrupt" set-point changes that occur in the furnace can only be a result of manual changes. Because the main goal of this work is to find intervals of excitation, we can still assume that, in the intervals of interest, the set-points are independent and the system identification can be performed in open-loop mode.

Hypothesis 5.1. *In the petrochemical furnace, set-points can be considered independent*

and actuated manually every time they are moving considerably.

5.2 Finding Potential Intervals

In this section, different approaches to find initial intervals of excitation are applied. Moreover, the impact of the corresponding parameters is analyzed and different visualization techniques are used to guide the parameter choice.

5.2.1 Exponentially Weighted Filter

This method is described in detail in Item 3.2.1.1 and consists of applying exponentially weighted filters to both the mean and the variance of the signal. As is there explained, the application of this method requires the choice of the forgetting factors for both the mean and the variance (λ_μ and λ_S), as well as a threshold l_S to be compared with the estimated variance. Moreover, this choice must be made for each individual signal (inputs and outputs) in the dataset.

5.2.1.1 Single-Input Single-Output Case

Let us initially take the water tank data from Subsection 5.1.1 as an example of the SISO case. The signal is initially normalized to the range $[-0.5\,,\,0.5]$.

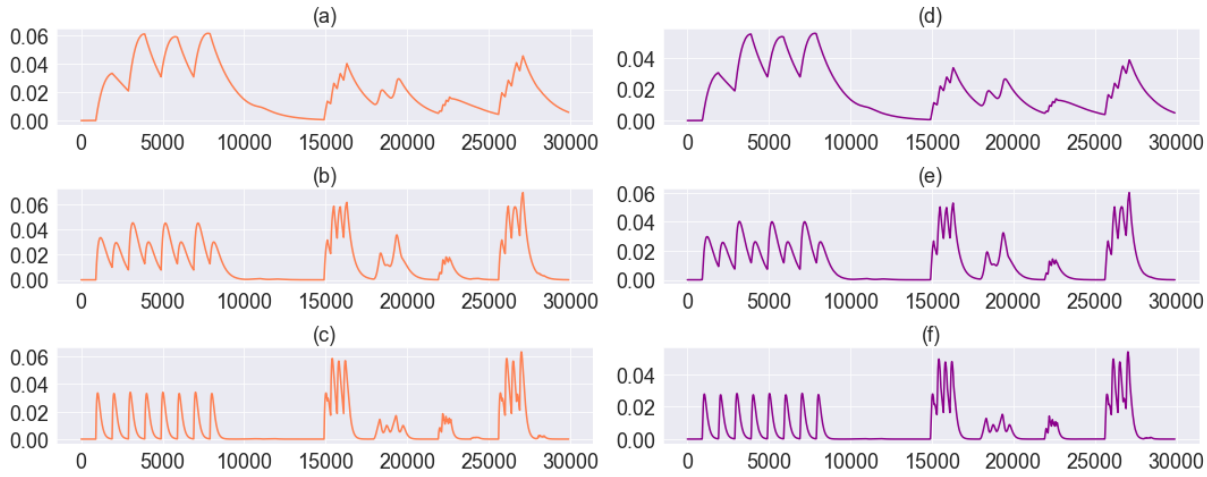
To exemplify the impact of the parameter choice, both the variance and the average filters receive the same value of forgetting factors, which are 0.0005, 0.002 and 0.005. The resulting signals can be seen in Figures 16 and 17.

Notice that, although the same values of forgetting factors are used for both the average and the variance filters, one could choose different values for each filter. It is not trivial, however, to make this choice in massive data, in which visualizing the whole data is not possible. Therefore, choosing the same forgetting factors for both filters is a simplification that makes this application feasible for huge datasets.

Potential intervals can then be defined through the variance signal setting a threshold value l_S to it. Let us take the signal in Figure 16 (c) as an example with a threshold value of $l_S = 0.004$. For this scenario, the resulting potential intervals would be defined as in Figure 18.

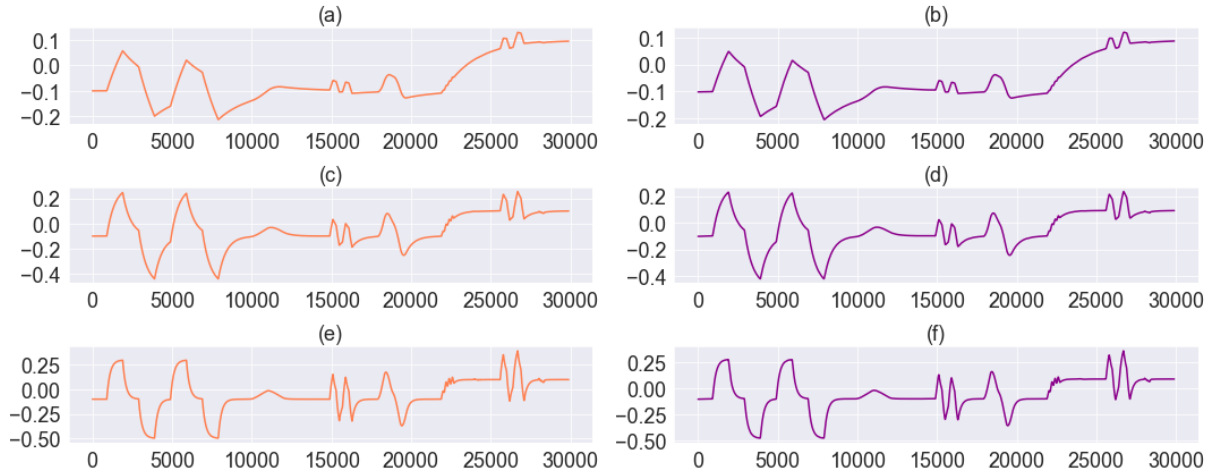
Notice that two vertical lines of the same color define an interval. The horizontal line

Figure 16: Signals filtered by the variance component of the EWMA filter, with different values of forgetting factors. (a) set-point with $\lambda_\mu = \lambda_S = 0.0005$ (b) set-point with $\lambda_\mu = \lambda_S = 0.002$ (c) set-point with $\lambda_\mu = \lambda_S = 0.005$ (d) output with $\lambda_\mu = \lambda_S = 0.0005$ (e) output with $\lambda_\mu = \lambda_S = 0.002$ (f) output with $\lambda_\mu = \lambda_S = 0.005$.



Source: Author's own development.

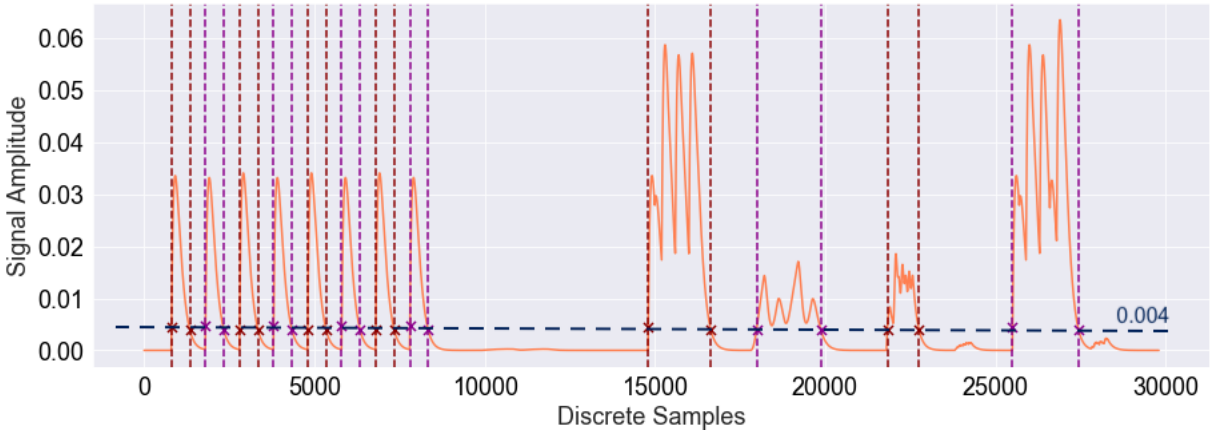
Figure 17: Signals filtered by the average component of the EWMA filter, with different values of forgetting factors. (a) set-point with $\lambda_\mu = \lambda_S = 0.0005$ (b) set-point with $\lambda_\mu = \lambda_S = 0.002$ (c) set-point with $\lambda_\mu = \lambda_S = 0.005$ (d) output with $\lambda_\mu = \lambda_S = 0.0005$ (e) output with $\lambda_\mu = \lambda_S = 0.002$ (f) output with $\lambda_\mu = \lambda_S = 0.005$.



Source: Author's own development.

delimits the variance threshold value. Moreover, the cross marks are the instants when the filtered signal crosses the variance threshold. Therefore, each interval defines a period of time in which the signal is "shaking" with a variance greater or equal to l_S. With this set of parameters, the algorithm produced 12 potential intervals to be further evaluated.

Figure 18: Effect of the variance threshold l_S in the resulting intervals.



Source: Author's own development.

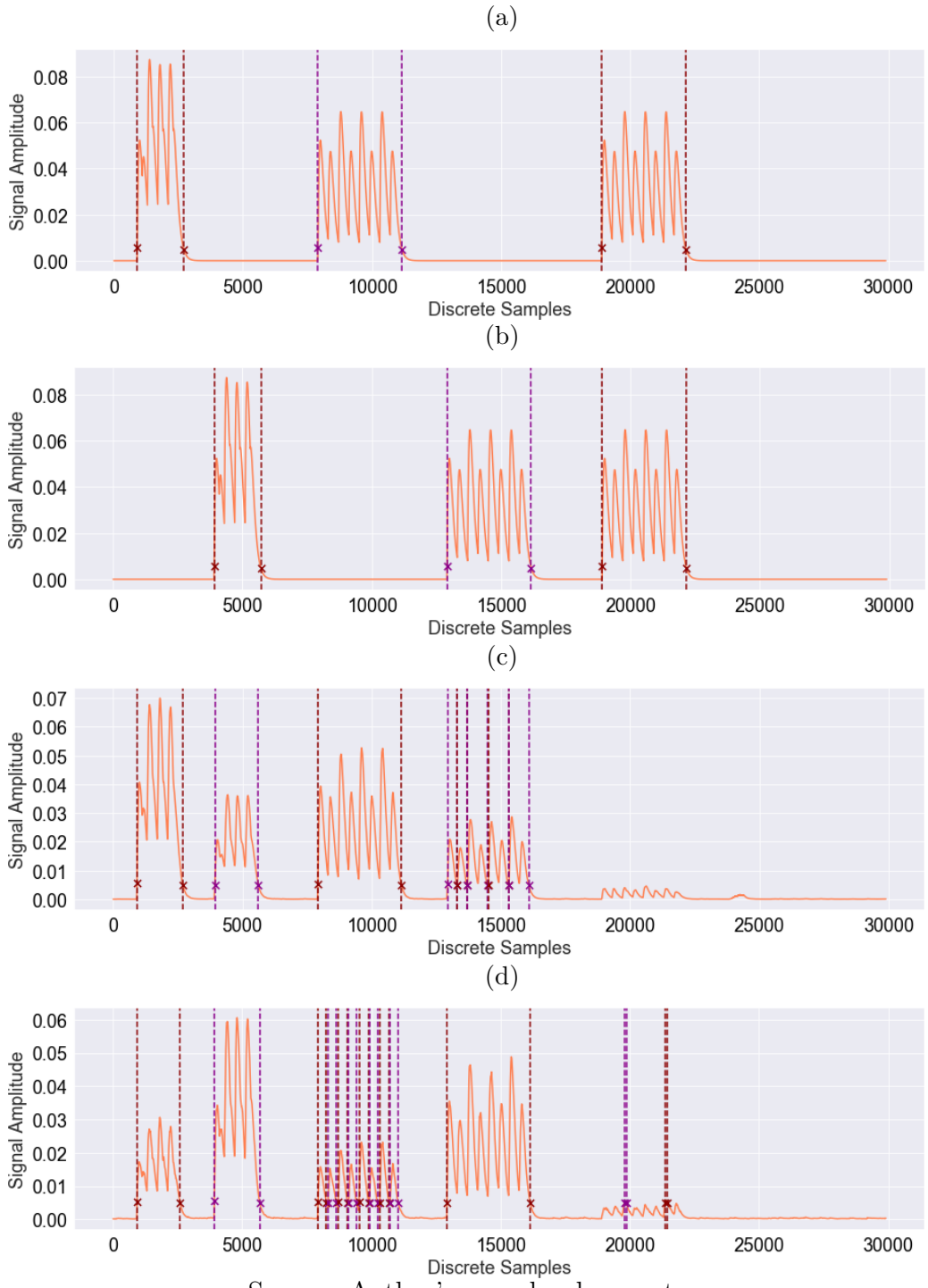
5.2.1.2 Multivariable Case

To exemplify how this method can be applied to multivariable data, let us take the distillation column data from Subsection 5.1.2 as an example. The difference here is that the filters are applied individually to each signal and then the resulting intervals are unified, as detailed in **Step 3** of the algorithm in Figure 8 . If one applies the exponentially weighted filter with parameters $\lambda_\mu = \lambda_S = 0.006$ and threshold $l_S = 0.005$ to this signal, the results in Figure 19 are obtained.

The intervals from each signal can then be unified. There are different ways of unifying these signals. One could, for example, take the intersection of each interval. However, if, for a particular interval, only one input signal suffered from excitation, this signal would not be considered in this case. A second approach would be to unify all intervals. In this case, however, one could end up considering intervals where only the output was "shaking", without any excitation in the input variables. That could happen due to disturbances. An interest way of unifying the resulting intervals is done borrowing the idea presented in (WANG *et al.*, 2018) and consists of initially unifying all the signals and then taking only the resulting intervals that contain at least one "active" input and one "active" output. This procedure is the one adopted in **Step 3** of the algorithm in Figure 8 and it is detailed described in Item 3.3.3.4.

Unifying the filtered signals as described produces the potential intervals in Figure 20. The reasons why unifying the signals in this fashion come in handy will be clearer in the multivariable example section. It is also interesting to highlight that the gaussian noise was completely filtered with this set of parameters and would not be considered in further analysis.

Figure 19: Exponential filter for each signal in the multivariable distillation column data. (a) Reflux flow rate. (b) Steam flow rate. (c) Overhead composition. (d) Bottom composition.



Source: Author's own development.

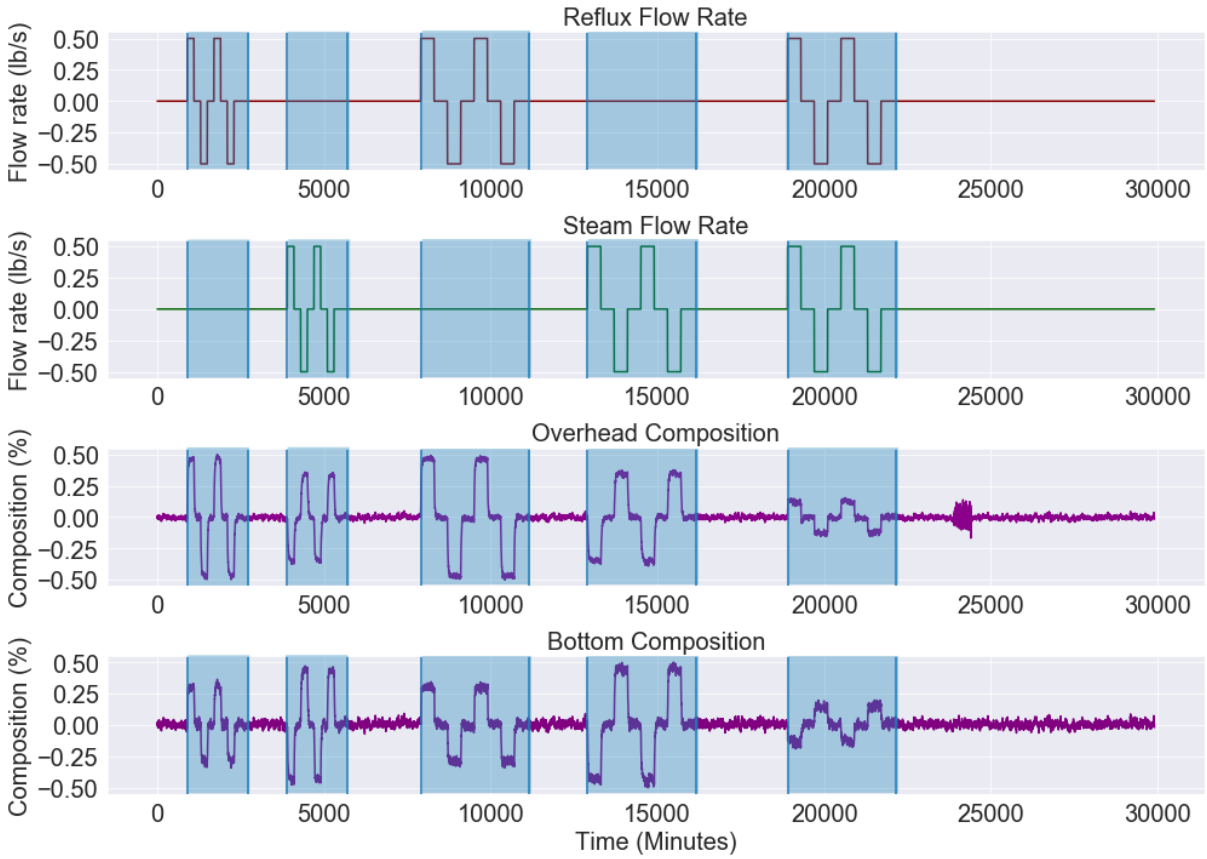
5.2.1.3 Impact of the Filter in Step Responses

It is interesting to point out how the exponentially weighted filters impact the step responses. Potential intervals are defined as time intervals where the signal suffers from a

significant amount of variance. This condition is verified against a user-defined threshold l_S. For this reason, when step responses are applied, usually the dynamics of the first step is not considered in the final interval, once its detection occurs after the step change itself. To elucidate this problem, let us highlight the beginning of the first potential interval of the distillation column reflux flow rate in Figure 20, which can be seen in Figure 21.

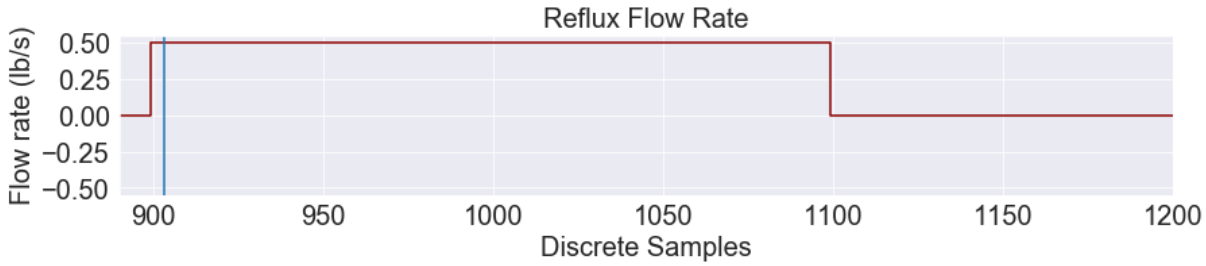
This effect is actually not a problem if one considers that in real data usually the manual changes in the set-point are almost never a single step response, as it can be seen in the petrochemical data example. However, the initial responses can be easily considered including an additional user-defined parameter, which is here called n_{idx}, that corresponds to the number of initial indexes to be considered in each potential interval. Therefore, for $n_{idx} = 50$, for instance, the result in Figure 21 is changed to the one in Figure 22.

Figure 20: Resulting potential (blue rectangles) intervals after unifying filtered signals.



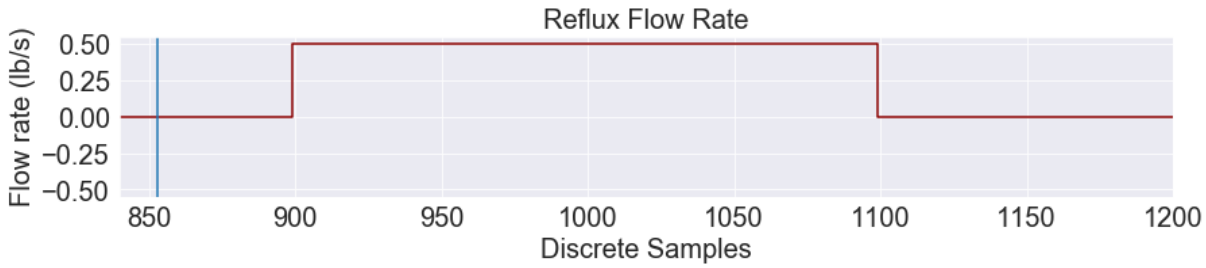
Source: Author's own development.

Figure 21: Effect of the exponentially weighted filter in step responses. The vertical blue line indicates the beginning of an interval.



Source: Author's own development.

Figure 22: Effect of the exponentially weighted filter in step responses with the additional parameter $n_{idx} = 50$. The vertical blue line indicates the beginning of an interval.



Source: Author's own development.

5.2.1.4 A Guideline to the Parameter Choice

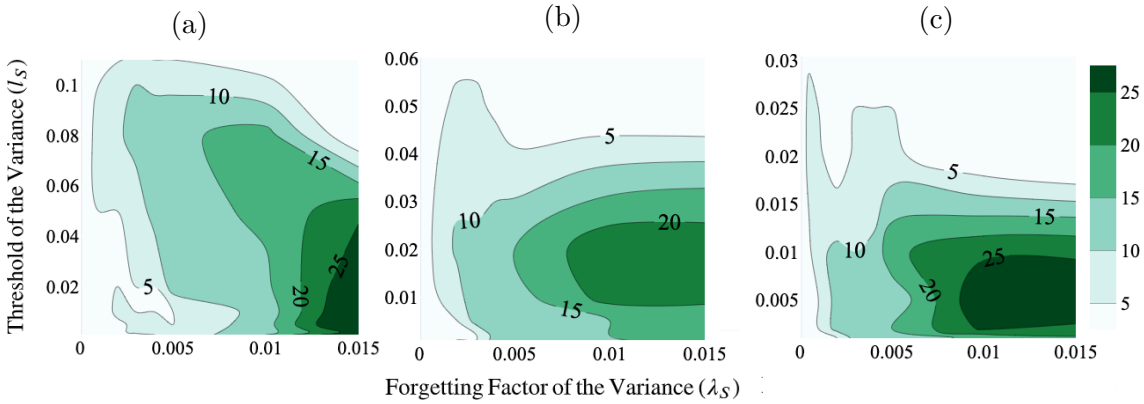
When applying the exponentially weighted filters to real massive data, a natural difficulty that arises is related to the parameters to choose. It is not trivial to have a general idea about the effect of each forgetting factor, as well as about the effect of the chosen variance threshold. This choice is specially hard when the segmentation is done in a single step, which is the case of the recursive implementation presented in (PERETZKI *et al.*, 2011) and (BITTENCOURT *et al.*, 2015) as well as the sliding window implementation presented in (ARENGAS; KROLL, 2017a) and (ARENGAS; KROLL, 2017b).

The advantage of initially obtaining potential intervals is that one can rely on visual analysis for choosing the desired parameters. An interesting way of visualizing the impact of each parameter is through a heatmap that shows the number of resulting intervals as a functions of the parameters. Figure 23 is an example of such visualization applied to the water tank data from Subsection 5.1.1.

It is interesting to mention that Figure 18 was obtained with $\lambda_\mu = \lambda_S = 0.005$ and $l_S = 0.004$ and resulted in 12 potential intervals. If one looks at Figure 23 (b), which is the scenario where $\lambda_\mu = \lambda_S$, one can easily see that, for this value of the forgetting factors

and threshold, the number of intervals is between 10 and 15.

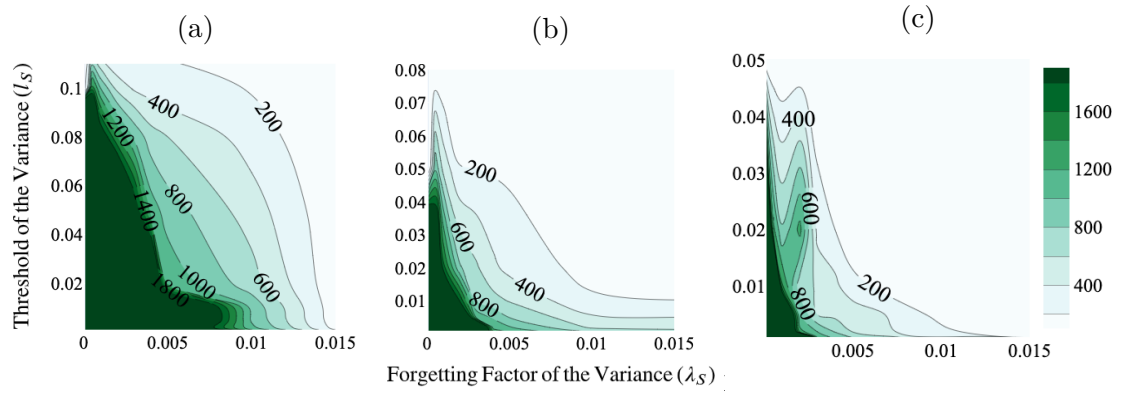
Figure 23: Impact of the forgetting factors and of the variance threshold on the number of resulting potential intervals. (a) $\lambda_\mu = 0.5\lambda_S$ (b) $\lambda_\mu = \lambda_S$ (c) $\lambda_\mu = 2\lambda_S$



Source: Author's own development.

Another interesting visualization that can guide one in the parameter choice is to verify the average length of the resulting intervals as a function of the forgetting factors and the variance threshold. Figure 24 exemplify this scenario. Therefore, with these two visual analyzes, it is possible to have a general idea on how many intervals are obtained and their approximate size.

Figure 24: Impact of the forgetting factors and of the variance threshold on the length of the resulting potential intervals. (a) $\lambda_\mu = 0.5\lambda_S$ (b) $\lambda_\mu = \lambda_S$ (c) $\lambda_\mu = 2\lambda_S$



Source: Author's own development.

5.2.1.5 Execution Time

The execution time of the algorithm running on a Core i7 (3.1 Ghz) MacBook Pro is the following:

a) execution on the single-input single-output water tank: 5.14 seconds;

b) execution on the multivariable distillation column dataset: 9.80 seconds.

Notice that the algorithm grows linearly with the input size, *i.e.*, its time complexity is $O(N)$ in Big O notation, being N the length of the dataset.

5.2.2 Bandpass Filter

In this Subsection, the bandpass filter method described in Item 3.2.1.2 is exemplified. The filter is computed as a deviation error from the mean value of each signal and three parameters must be choosen: the bandpass frequencies w_1 and w_2 in rad/s and the deviation error threshold l_e, defined in Equation (3.16).

5.2.2.1 Single-Input Single-Output Case

Applying the bandpass filter with $w_1 = 0.006$ rad/s, $w_2 = 0.04$ rad/s and $l_e = 0.02$ to the water tank data, results in Figure 25 are obtained. The horizontal lines in Figure 25 correspond to the threshold l_e and, therefore, the intervals are defined as the sample indexes where the filtered signal is higher than l_e or lower than $-l_e$ (red dots). This implementation is consistent with the results presented in (PATEL, 2016, p.10).

The resulting intervals can be unified following the same procedure described in the previous subsection. The resulting potential intervals in this case are shown in Figure 26. Notice that consecutive red and green vertical lines represent the resulting intervals. From Figure 25, one can see that the ramp signal around the instant 10000 would only be considered if the deviation threshold l_e is very carefully chosen. Moreover, it is interesting to notice that the gaussian random noise was completely eliminated, while the colored noise was considered.

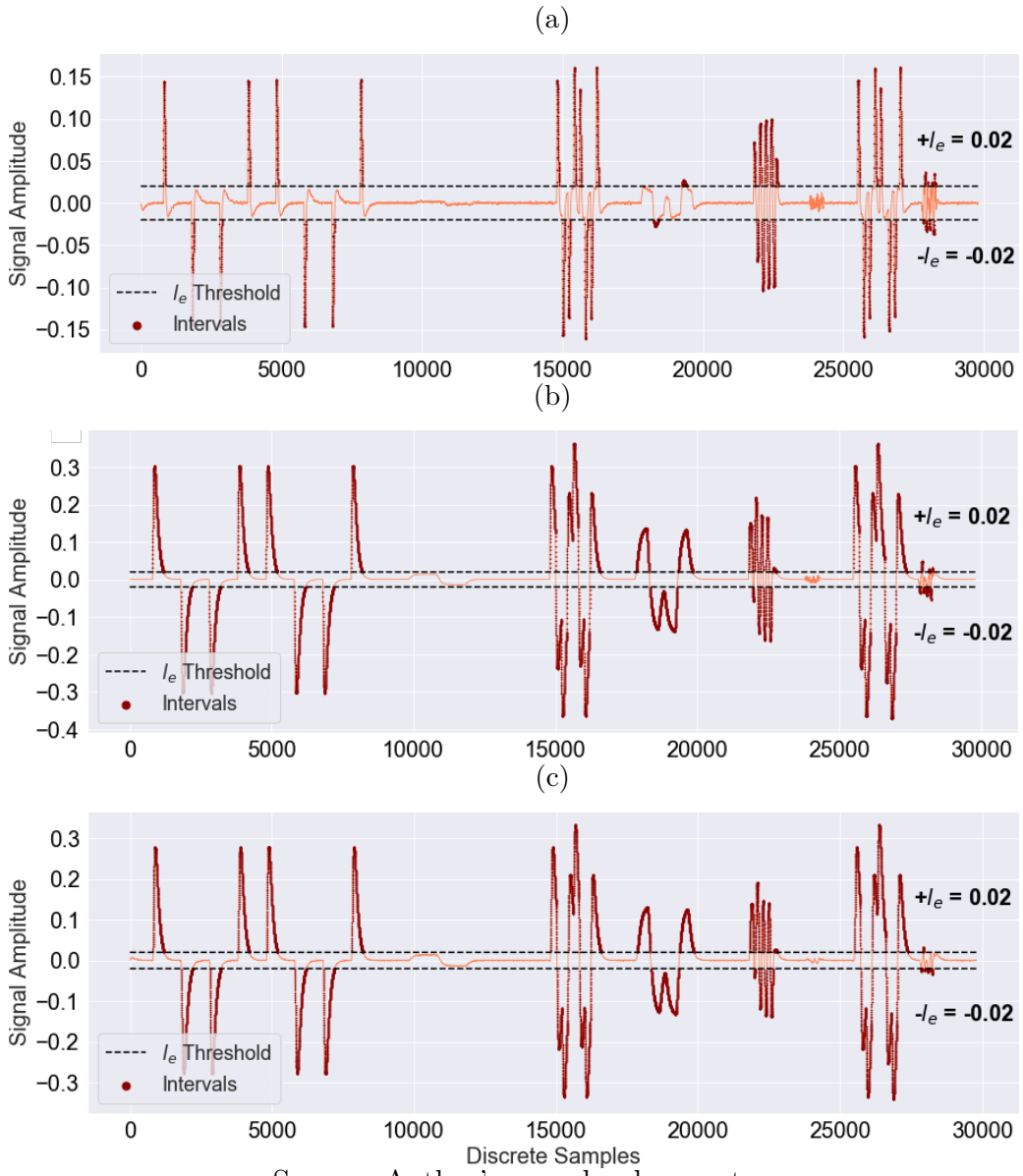
5.2.2.2 Multivariable Case

The multivariable algorithm can be applied exactly as in Item 5.2.1.2, which consists in first applying the bandpass filter to each signal and then unifying the resulting intervals. Notice that individual values of w_1, w_2 and l_e could be applied to each individual signal.

5.2.2.3 Impact of the Filter in Step Responses

The exact same problem described in Item 5.2.1.3 happens with the bandpass filter, *i.e.*, when step responses are applied in the set-point, the initial response is detected when

Figure 25: Bandpass filtered signals in the water tank data with $w_1 = 0.006$ rad/s, $w_2 = 0.04$ rad/s and $l_e = 0.02$. (a) Manipulated variable. (b) Set-point. (c) Controlled variable.



Source: Author's own development.

the step change already occurred. The same additional user-defined parameter n_{idx} can be applied here to fix this problem.

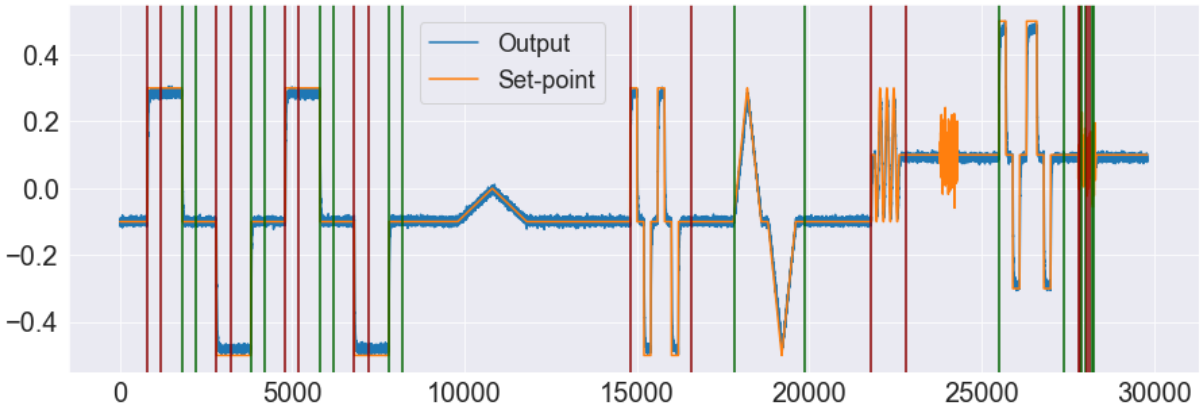
5.2.2.4 A Guideline to the Parameter Choice

The heatmap analysis from the previous subsection can be used to support the parameter choice. The impact of the w_1 and w_2 frequencies in the number of resulting intervals, for three different values of l_e, can be seen in Figure 27.

In the same way, Figure 28 shows the impact of the parameter choice in the average

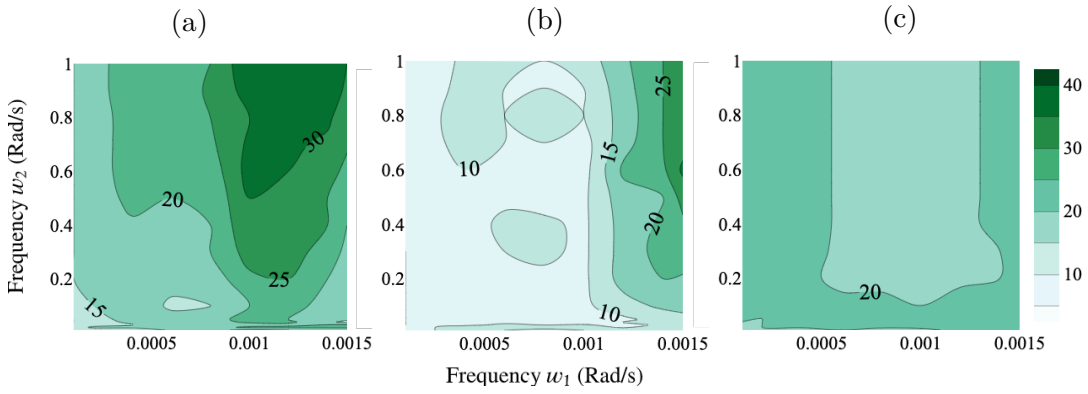
size of the resulting intervals.

Figure 26: Unified intervals obtained with the bandpass filter and parameters $w_1 = 0.006$ rad/s, $w_2 = 0.04$ rad/s and $l_e = 0.02$.



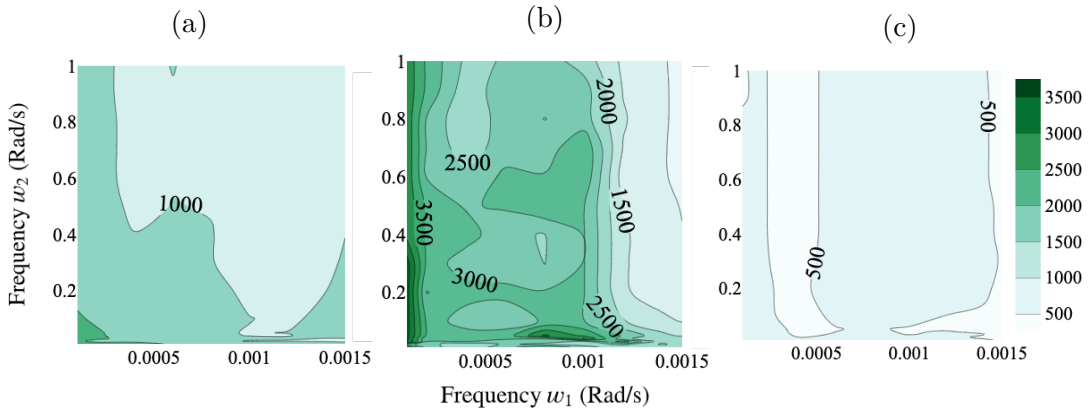
Source: Author's own development.

Figure 27: Impact of the frequencies w_1 and w_2 in the number of resulting intervals. (a) $l_e = 0.02$ (b) $l_e = 0.005$ (c) $l_e = 0.1$.



Source: Author's own development.

Figure 28: Impact of the frequencies w_1 and w_2 in the average size of resulting intervals. (a) $l_e = 0.02$ (b) $l_e = 0.005$ (c) $l_e = 0.1$.



Source: Author's own development.

5.2.2.5 Execution Time

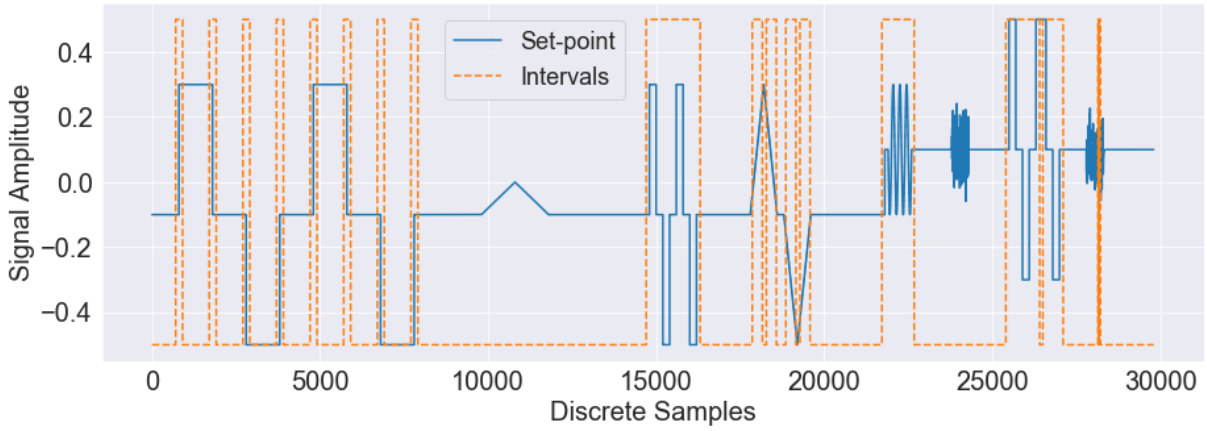
The execution time of the algorithm for the water tank data, running on a Core i7 (3.1 Ghz) MacBook Pro, is 1.29 seconds.

5.2.3 Sliding Window

An alternative approach to the exponentially weighted and bandpass filters is to compute the variance in a sliding-window fashion, as explained in Section 3.2. In this approach, one must only choose the window size and a variance threshold beyond which the signal is considered to be "active".

Applying the sliding window algorithm to the water tank dataset with a window size $w_s = 200$ and a variance threshold $l_S = 0.003$, the potential intervals in Figure 29 are obtained. In this case, the algorithm is applied to both the output and set-point and the procedure described in Item 5.2.1.2 for unifying the input and the output intervals is also applied. It is interesting to point out that the resulting intervals are of lower length. The impact of these smaller intervals in the results are described in Subsection 5.3.1. Moreover, the way that the algorithm defines an interval can be clearly seen through Figure 30. Notice that the variance signal represented by the orange dashed line is produced by the the window function and captures the region in which the signal is moving. This signal is then compared to threshold l_S, in such a way that the indexes where the variance signal is greater than its threshold define the potential intervals (dashed blue line).

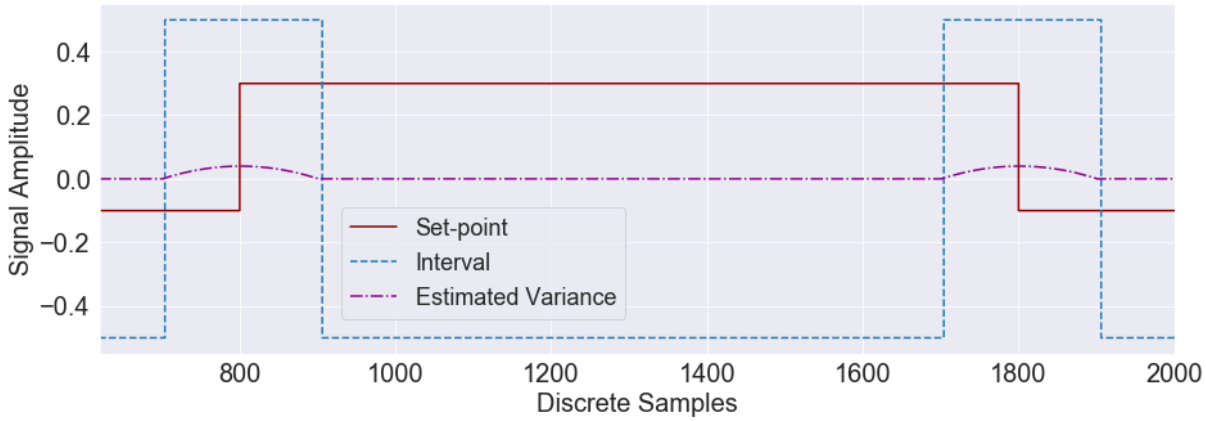
Figure 29: Unified intervals obtained by the sliding-window approach with parameters $w_s = 200$ and $l_S = 0.003$.



Source: Author's own development.

It is also worthwhile to mention that the way the sliding-window was designed in

Figure 30: Detail of the sliding-window algorithm.



Source: Author's own development.

this dissertation (Section 3.2) captures the set-point transition before it happens and, therefore, the additional user-defined parameter n_{idx} is not necessary in this case.

5.2.3.1 Multivariable Case

The multivariable application of the algorithm can be done exactly as in Item 5.2.1.2, which consists in first applying the sliding-window to each signal and then unifying the resulting intervals.

5.2.3.2 A Guideline to the Parameter Choice

The impact of the window size and the variance threshold in the number of resulting intervals can be seen in Figure 31.

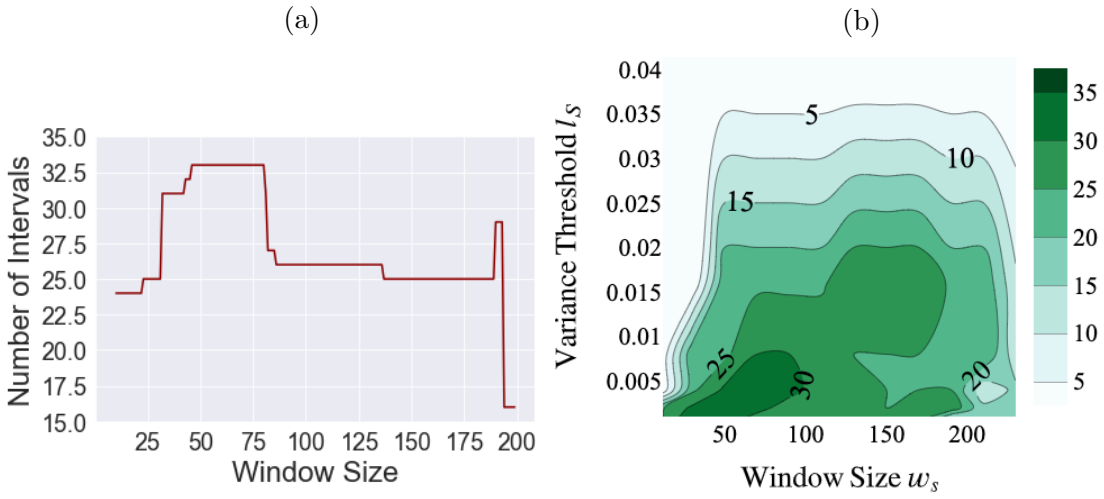
5.2.3.3 Execution Time

The execution time of the algorithm for the water tank data, running in a Core i7 (3.1 Ghz) MacBook Pro, is 1.74 seconds.

5.2.4 Change-Point Detection

The non-parametric top-down approach described in Subsection 3.2.2 is here exemplified. One advantage of this method is that few parameters must be chosen, which are the significance level α and the minimum number of indexes that a interval must have to allow a further split.

Figure 31: Impact of the window size and variance threshold on the number of resulting intervals. (a) impact of window size with $l_S = 0.003$ (b) impact of window size and variance threshold.



Source: Author's own development.

Applying the change-point algorithm to the water tank dataset with $\alpha = 0.05$ and a minimum split length of 1200, Figure 32 is obtained. The black cross marks in Figure 32 are the changing positions identified by the algorithm and, therefore, the time indexes between change-points define an interval. Notice that differently from the previous methods, the change-point approach divide the entire dataset and the whole data needs to be further analyzed.

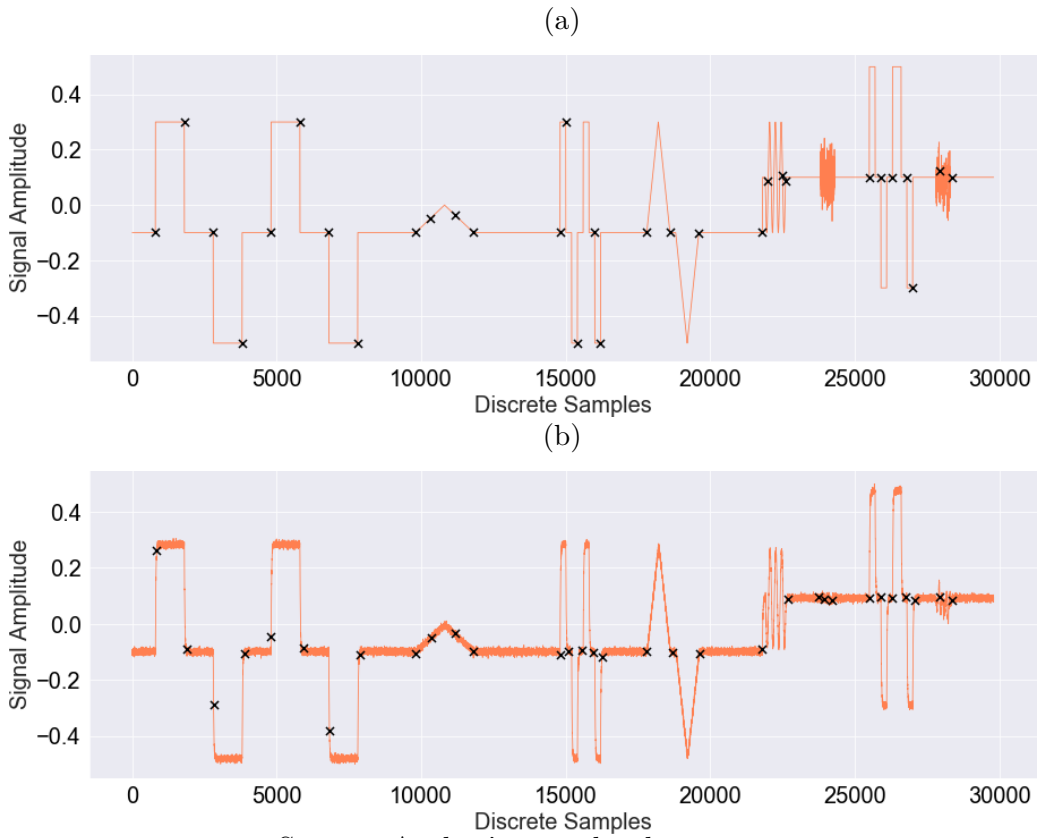
To exemplify how the top-down method works, the five initial iterations of the algorithm are shown in Table 3. Notice that in the first iteration no change-point is identified. This is because, as explained in Subsection 3.2.2, the algorithm starts with the whole dataset in the first iteration and makes growing divisions at each new segment obtained, until no further change-point is found at the significance level α.

Table 3: First five iterations from the non-parametric top-down change-point algorithm.

Iteration	**Change-point Indexes**
1	[]
2	[21801]
3	[1800, 21801, 26800]
4	[800, 1800, 9801, 21801, 26300, 26800, 27000]
5	[800, 1800, 3800, 9801, 11800, 21801, 25900, 26300, 26800, 27000, 28385]

Because this method splits the entire data, it does not make sense to unify the resulting intervals for each signal. Therefore, the way this algorithm can be used is by splitting individually each signal and evaluating every resulting segments. Finally, the resulting

Figure 32: Non-parametric top-down change-point applied to the water tank dataset with $\alpha = 0.05$ and minimum split length of 1200. (a) Set-point. (b) Output signal.



Source: Author's own development.

intervals that pass further evaluations can then be unified. This is, in general terms, the idea behind the statistical method exemplified in Subsection 5.3.2.

5.2.4.1 Execution Time

The execution time of the algorithm for the water tank data, running in a Core i7 (3.1 Ghz) MacBook Pro, is 396 seconds considering only the set-point and the output variables. Notice that, because the algorithm requires the relative position of the signal as in Equation (3.20), its time-complexity in Big O notation is $O(N^2)$ (WANG *et al.*, 2018). It means that the algorithm is very inefficient for long datasets and, therefore, a more clever solution in that case would be to split the dataset in batches.

5.3 Singe-Input Single-Output (SISO) Segmentation

5.3.1 Numerical Conditioning and Rank Test Examples

In this subsection, the methods described in Subsections 3.3.1 and 3.3.2 are exemplified through the application of the algorithm in Figure 7. Different model structures are used, allowing one to validate the proposed methodology. Moreover, both the stationary and the incremental approach presented in the methodology are compared.

5.3.1.1 Laguerre Filter

In this item, the Laguerre Filter structure is used to obtain suitable intervals for system identification. As described in the Methodology chapter, one can validate the obtained potential intervals and save the suitable ones in their original shape (stationary approach) or one can increment the suitable potential intervals until they meet the testing criteria (incremental approach). Results for both scenario are analyzed in this Item.

Stationary Approach

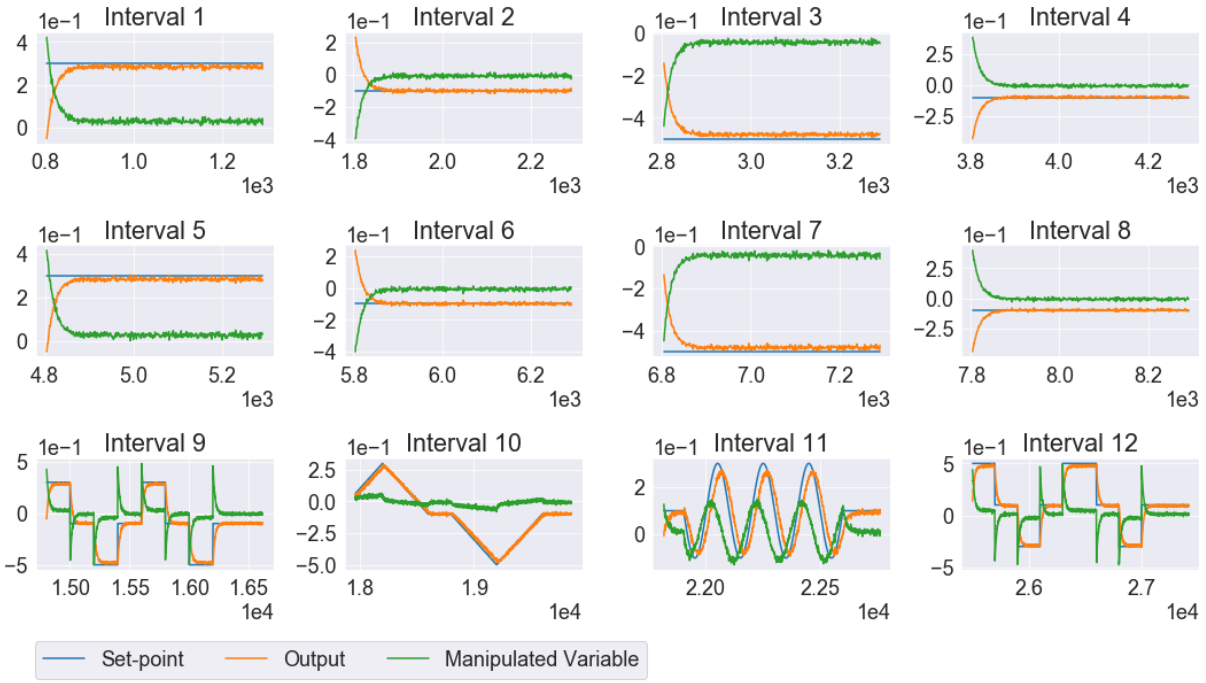
In this item, the water tank system is considered. As the first step of the algorithm, one must find potential intervals to be evaluated following one of the approaches described in Section 5.2. Notice that the system is operating under closed-loop and, therefore, two different models can be obtained: a model of $G(q)$ and a model that includes both $G(q)$ and $C(q)$, as explained in Section 2.7. The former case is the one analyzed in this item, which corresponds to the closed-loop identification scenario.

As explained in Item 3.3.2.7, for a system operating under closed-loop control, the condition number of the information matrix is computed using the set-point. That is because, for closed-loop systems, the set-point must be persistently exciting in order for the process model to be identifiable (BITTENCOURT *et al.*, 2015). Therefore, potential intervals are obtained searching for excitations in both the set-point and in the output variables, as described in the algorithm outline in Item 4.1.1.1. The exponentially weighted filter is used in this case with $\lambda_S = \lambda_\mu = 0.006$ and $l_S = 0.003$ for both the set-point and output variables, resulting in 12 potential intervals shown in Figure 33.

Let us first consider that $n_{idx} = 0$, which will fail to detect individual step responses. Considering a Laguerre Filter structure of order $N_b = 10$ and pole $\alpha = 0.92$, the resulting condition number and effective rank are the ones in Table 4. Here, the effective rank is computed through Type 2 described in Item 3.3.1.3 and the singular value threshold

adopted is $l_2 = 0.5$.

Figure 33: Potential Intervals obtained with the exponential weighted filter and parameters $\lambda_S = \lambda_\mu = 0.006$, $n_{idx} = 0$ and $l_S = 0.003$.



Source: Author's own development.

Table 4: Condition number and effective rank for each potential interval considering the set-point with $n_{idx} = 0$.

Interval	**Condition Number**	**Effective Rank**
1	2.00×10^{27}	1
2	1.25×10^{18}	1
3	7.07×10^{17}	1
4	1.25×10^{18}	1
5	2.00×10^{27}	1
6	1.25×10^{18}	1
7	7.07×10^{17}	1
8	1.25×10^{18}	1
9	6.18×10^{3}	9
10	2.78×10^{7}	5
11	3.56×10^{5}	7
12	8.56×10^{3}	9

Notice that the condition number and the effective rank are completely consistent. Because intervals 1 to 8 do not capture the step change, the information matrix computed with these intervals is not invertible. On the other hand, the set-point in intervals 9 to 12 moves several times and these changes are detected. In fact, intervals 9 and 12 produce

better conditioned information matrices compared to intervals 10 and 11. This is true considering both the condition number and the effective rank.

From Figure 33, one can notice that the dynamics of the manipulated variables in intervals 1 to 8 were not completely eliminated by the filter, as occurred with the set-point. However, it is expected that little information is produced by these signals, once an important dynamic component was not detected by the filter. If the condition number and the effective rank are computed considering the manipulated variable instead of the set-point, Table 5 is obtained. In this case, the condition number is still very high for intervals 1 to 8. The effective rank, on the other hand, is not able to capture the "ill-conditioned" intervals. That is because the maximum rank a regressor matrix can have is limited to the model order, which is 10 in this example. Consequently, the effective rank is a less sensitive metric compared to the condition number.

Table 5: Condition number and effective rank for each potential interval considering the manipulated variable and $n_x = 0$.

Interval	**Condition Number**	**Effective Rank**
1	1.05×10^5	7
2	9.32×10^4	7
3	1.29×10^5	7
4	8.45×10^4	7
5	1.03×10^5	7
6	9.62×10^4	7
7	1.14×10^5	7
8	9.93×10^4	7
9	8.37×10^2	9
10	2.50×10^3	3
11	7.73×10^3	7
12	8.99×10^2	9

Let us now consider that $n_{idx} = 20$, in such a way that Figure 33 is modified to Figure A.1. Reapplying the Laguerre structure to each signal and considering the set-point, Table 6 is now obtained

One can now notice that, with exception of interval 10, all intervals have a reasonable value of condition number and effective rank. In fact, the obtained values are consistent with the ones presented in (PERETZKI *et al.*, 2011), (SHARDT; HUANG, 2013a), (BITTENCOURT *et al.*, 2015), (ARENGAS; KROLL, 2017a) and (ARENGAS; KROLL, 2017b), in which the condition number threshold for step responses are usually around 10^4.

The next step of the algorithm in Figure 7 is to check if the input and the output are

Table 6: Condition number and effective rank for each potential interval considering the set-point.

Interval	**Condition Number**	**Effective Rank**
1	1.12×10^4	7
2	4.26×10^3	6
3	3.46×10^4	7
4	1.39×10^4	7
5	1.12×10^4	7
6	4.26×10^3	6
7	3.46×10^4	7
8	1.39×10^4	7
9	4.70×10^3	9
10	2.79×10^7	5
11	1.06×10^4	5
12	5.87×10^3	9

actually correlated. As described in Item 3.3.2.5, an estimate of the model parameters can be computed to perform a Granger causality test. Because a model of $G(q)$ is the main interest in this example, the model parameters must be computed using the output and the manipulated variables (see Item 3.3.2.7). A table of the chi-squared values for each interval can be seen in Table 7.

Table 7: Chi-squared values for each interval considering the manipulated variable.

Interval	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
Chi-squared Value	104	34	129	10	100	36	135	9	116	111	101	137

The cross-correlation scalar metric explained in Item 3.3.1.4 can also be computed. Results for the cross-correlation between the output and the set-point for a lag range $[-5,\ 5]$ and for a significance level $\alpha = 0.05$ can be seen in Table 8.

Table 8: Scalar cross-correlation values for each interval considering the set-point.

Interval	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
Cross-correlation	3.9	3.9	3.9	3.9	3.9	3.9	3.9	3.9	5.0	5.3	4.4	5.1

The final intervals suitable for system identification are those that meet all the defined thresholds. Therefore, if one considers, for instance, a condition number threshold $l_\kappa = 15000$ and a significance level for the chi-squared test $\alpha = 0.01$, only intervals 1, 2, 5, 6, 9, 11 and 12 would be considered. Notice that for a model order of 10, the chi-squared critical value for this significance level is 23.2.

In the same way, the final intervals can be obtained setting a threshold to the effective

rank and to the cross-correlation metric. Notice, however, that better results are obtained with the effective rank when models of higher order are used, as exemplified by the ARX structure in Item 5.3.1.2. Finally, one could combine both approaches, in such a way that, for instance, the condition number and the scalar cross-correlation could be used to evaluate the potential intervals. In a similar fashion, one could evaluate the potential intervals considering the effective rank and the chi-squared value. This flexibility is explicitly exposed in the algorithm outline in Figure 7.

Incremental Approach

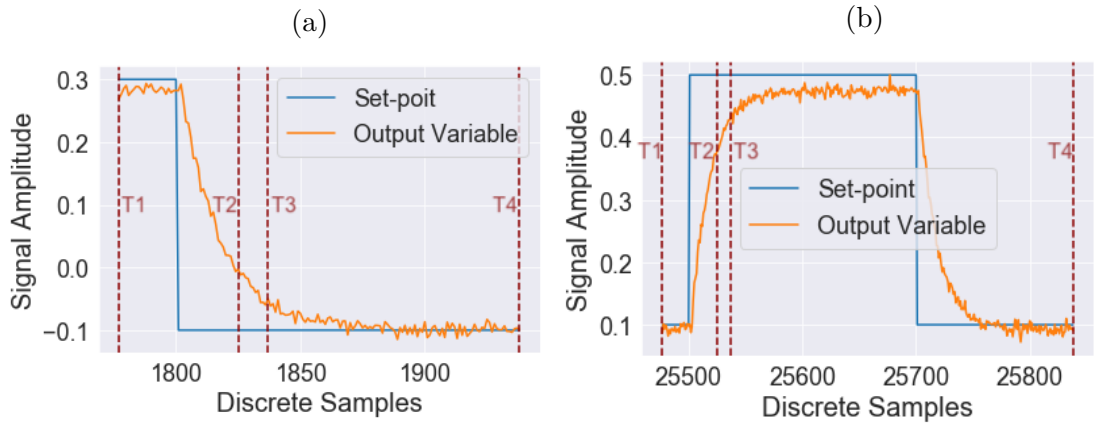
Usually, higher intervals lead to higher condition numbers. Such a discussion is presented in (CARRETTE *et al.*, 1996). For this reason, it can be interesting to start the analysis with smaller intervals and increment them while the desired criteria are met. In this sense, the recursive approach proposed in (PERETZKI *et al.*, 2011) or the sliding-window approach proposed in (ARENGAS; KROLL, 2017a) and (ARENGAS; KROLL, 2017b) come in handy. The way this is done in this dissertation is described in the Methodology chapter and consists of incrementing the potential intervals with a window of size w_{ic}. Notice that this approach was inspired in the algorithm proposed in (ARENGAS; KROLL, 2017a) and (ARENGAS; KROLL, 2017b). Smaller initial intervals are here obtained with the variance sliding-window exemplified in Item 5.2.3. However, changing the parameters in the exponential filter or in the bandpass filter can also lead to smaller intervals.

In this example, the sliding-window algorithm is applied to the water tank data with a window size of $w_s = 50$ and a threshold $l_S = 0.003$ for both the set-point and the output variables. The incremental step used here is $w_{ic} = 100$. Results for two intervals are given in Figure 34.

The evaluation criteria used in this example is the condition number and the chi-squared statistical test, considering a Laguerre structure of order $N_b = 7$ and pole $\alpha = 0.8$. The time indexes T1, T2, T3 and T4 represent the evaluation timeline. More specifically, T1 is the time instant when the set-point satisfied the variance threshold condition $l_S = 0.003$ in the sliding-window algorithm. In the opposite direction, T2 is the instant when this condition is not satisfied anymore. Similarly, T3 indicates the instant when the output variable do not satisfy the variance threshold condition anymore. Finally, T4 represents the moment in which either the condition number or the chi-squared criteria are not met anymore. The condition number threshold used in this example is $l_\kappa = 15000$ and the chi-squared significance level adopted is $\alpha = 0.01$. Moreover, the resulting intervals are

limited to a maximum length of 600 samples.

Figure 34: Example of the incremental approach using the sliding-window algorithm with $w_s = 50$, a threshold $l_S = 0.003$ and an incremental step of $w_{ic} = 100$.



Source: Author's own development.

Table 9 shows the values of each estimated variance at instants T1, T2 and T3 for the signal in Figure 34 (a), which were obtained using the sliding-window algorithm for both the set-point and the output signals.

Table 9: Estimated variance using the sliding-window algorithm at samples T1, T2 and T3.

	T1	**T2**	**T3**
Set-point	3.14×10^{-3}	3.14×10^{-3}	1.93×10^{-34}
Output	3.43×10^{-5}	1.27×10^{-2}	3.14×10^{-3}

In the same way, Table 10 shows the condition number and the chi-squared value for the original interval, which contemplates the range [T1, T3], and [T1, T4] for the incremented interval. Notice that the condition that broke the incremental process was the chi-squared value, which reached the value of 11.3 that is lower than the critical value for a significance level $\alpha = 0.01$ and for a Laguerre order $N_b = 7$ ($\chi_{7,0.01} = 18.48$).

Table 10: Condition number and chi-squared values for intervals [T1, T3] and [T1, T4].

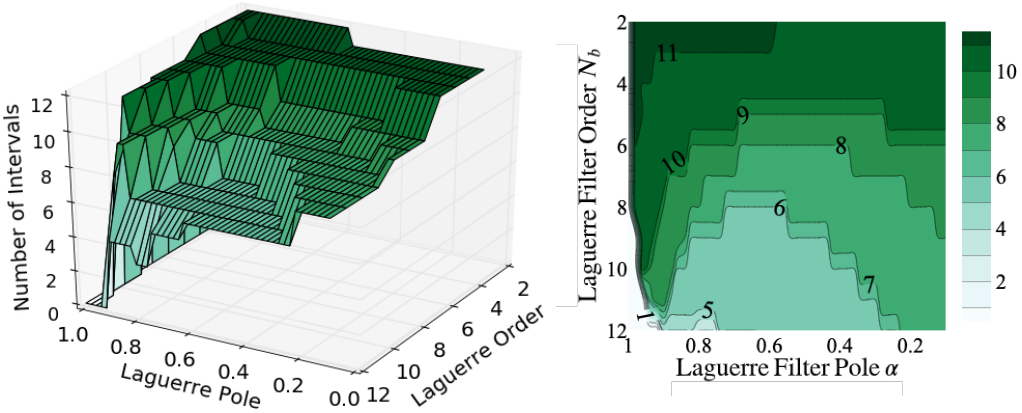
	[T1, T3]	**[T1, T4]**
Condition Number	1304.4	1396.5
Chi-squared Value	33.0	11.3

Impact of Laguerre Filter Parameters

It is also important to exemplify the impact of Laguerre filter pole and order in the resulting mined intervals. An interesting way of understanding the impact of both parameters is visualizing the number of resulting intervals obtained with several combinations

of these parameters. A surface and a heatmap plot can be seen in Figure 35. One can notice that the shape of the surface plot is very similar to the surfaces presented in (PATEL, 2016, p.40-41), which reinforces the consistency of the results. It is interesting to notice that when the filter pole is located at 1, none mined intervals are produced. On the contrary, too low filter orders produce too many intervals and, therefore, do not help to discriminate "good" and "bad" intervals. For this particular data, a good parameter choice lies in the region that contemplates the order range of [6, 10] and the pole range of [0.8, 0.95], once it crosses the surface with an area that contains a range of [6, 10] intervals. In fact, these values are consistent with the adopted parameter values in many of the reviewed works.

Figure 35: Impact of the Laguerre structure pole and order in the number of mined intervals for a condition number threshold $l_\kappa = 20000$ and a chi-squared significance level $\alpha = 0.01$.



Source: Author's own development.

5.3.1.2 ARX Structure

In this item, the ARX is used to compute the same metrics. The idea is to verify the consistency of the results, as well as to mention the differences between using this structure compared to the Laguerre one. Applying the same algorithm to the potential intervals in Figure 33 but using an ARX with orders $n_u = 2$, $n_y = 2$ and $n_k = 1$, the resulting condition number and effective rank are the ones in Table 11.

Notice that the results are in accordance with the ones obtained with the Laguerre structure. The effective rank, in this case, is not a good metric to be used to evaluate the quality of each interval because the ARX orders adopted were too low. More specifically, for the chosen model orders, a full effective rank matrix assumes the rank value of $n_u+n_y = 4$. If a model order of $n_u = 30$, $n_y = 30$ and $n_k = 1$ is used instead, the Type 2 effective

ranks, for a threshold $l_2 = 0.01$, can be seen in Table 12.

Table 11: Condition number and effective rank for each potential interval considering the set-point, using an ARX structure with orders $n_u = 2$, $n_y = 2$ and $n_k = 1$.

Interval	**Condition Number**	**Effective Rank**
1	4.38×10^{35}	1
2	4.92×10^{34}	1
3	1.25×10^{36}	1
4	∞	1
5	∞	1
6	∞	1
7	1.25×10^{36}	1
8	5.63×10^{16}	1
9	8.61×10^{3}	2
10	2.92×10^{6}	1
11	9.07×10^{4}	2
12	8.75×10^{3}	2

Table 12: Type 2 Effective Rank for a singular value threshold $l_2 = 0.01$ and an ARX structure with orders $n_u = 30$, $n_y = 30$ and $n_k = 1$.

Intervals	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
Effective Rank	3	4	3	4	4	4	3	3	29	8	9	29

Let us now consider the potential intervals in Figure A.1. In this case, the condition number and the effective rank for an ARX structure with orders $n_u = 2$, $n_y = 2$ and $n_k = 1$ can be seen in Table 13. Notice that the condition number is, again, consistent with the results obtained with the Laguerre Filter, in which intervals 3, 7 and 10 resulted in the worst values.

The chi-squared values for these intervals and for this structure can be seen in Table 14. It is interesting to notice that, as with the Laguerre structure, intervals 9-12 produced higher values of the chi-squared statistic. Moreover, all intervals produced a much higher chi-squared values compared to the ones obtained by the Laguerre structure and shown in Table 7. In this case, the chi-squared value would need to be compared with a user-defined critical value threshold, because they are much higher then any critical value defined with standard significance levels, such as $\alpha = 0.01$. In (PATEL, 2016), the ARX and the Laguerre structures are joined in a single structure to compute the chi-squared value, merging characteristics of both models.

Evaluating the Type 2 effective rank with an ARX structure of orders $n_u = 30$, $n_y = 30$ and $n_k = 1$ for intervals in Figure A.1, results in Table 15 are obtained. Notice

that these results are coherent with the condition number values and show that higher model orders are better when using the effective rank as the evaluation metrics.

Table 13: Condition number and effective rank for each potential interval considering the set-point, using an ARX structure with orders $n_u = 2$, $n_y = 2$ and $n_k = 1$.

Interval	**Condition Number**	**Effective Rank**
1	8.46×10^3	2
2	1.09×10^3	2
3	2.86×10^4	2
4	2.19×10^3	1
5	8.39×10^3	2
6	1.27×10^3	2
7	2.36×10^4	2
8	2.17×10^3	1
9	8.50×10^3	2
10	2.91×10^6	1
11	3.56×10^3	2
12	8.69×10^3	2

Table 14: Chi-squared values for each interval considering the manipulated variable.

Interval	**1**	**2**	**3**	**4**	**5**	**6**
Chi-squared Value	464.5	555.8	493.2	712.8	494.18	512.7
Interval	**7**	**8**	**9**	**10**	**11**	**12**
Chi-squared Value	501.0	750.6	4765.9	4356.5	1429.1	5890.1

Table 15: Type 2 Effective Rank for a singular value threshold $l_2 = 0.01$ and an ARX structure with orders $n_u = 30$, $n_y = 30$ and $n_k = 1$.

Intervals	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
Effective Rank	16	18	16	17	16	16	17	16	29	8	11	30

5.3.1.3 System Identification

As a final way of verifying if the algorithms applied in this subsection are actually useful in informative intervals for system identification, a model of each interval is obtained and evaluated. For comparison purposes, each interval is used to perform a cross-validation with every interval other than itself, as detailed in the Methodology chapter.

In this item, system identification is done through the "Indirect Approach" described in (LJUNG, 1999), which essentially consists of estimating a model with the output and the manipulated variables. The main reason for choosing this approach is the possibility to compare the resulting system identification metrics with the expected results produced

by the mining algorithms. Moreover, the chi-squared statistical test is also computed using the manipulated variable and the output variables in the closed-loop identification scenario.

Applying an ARX model structure with orders $n_u = 5$, $n_y = 3$ and $n_k = 1$ to every interval, the MATLAB® FIT value is computed with every other signal. The FIT value is computed as follows:

$$FIT = 100\left(1 - \frac{\|\boldsymbol{y} - \hat{\boldsymbol{y}}\|_2^2}{\|\boldsymbol{y} - \bar{y}\|_2^2}\right) \tag{5.5}$$

where $\hat{\boldsymbol{y}}$ is the predicted output and $\bar{y}$ is the mean value of the output variable. The FIT value was computed for every interval and for three validation prediction steps: 1 step-ahead, 100 steps-ahead and infinity steps-ahead. The later case corresponds to the free simulation, *i.e.*, the output response is completely based on model predictions, in which case the disturbance model is disregarded.

Table 16 shows the resulting average FIT values for each potential interval in Figure Figure A.1. If we take interval 1 as an example, with the ARX structure of orders $n_u = 5$, $n_y = 3$ and $n_k = 1$, this interval resulted, for 1 step-ahead predictions, in an average FIT value of 93.3%, being the validation computed with intervals 2-12.

Table 16: Cross-validation average FIT values for each potential interval (worst metrics highlighted in blue).

Interval	**1**	**2**	**3**	**4**	**5**	**6**
FIT 1 step-ahead	93.3 %	93.3 %	93.3 %	93.3 %	93.3 %	93.3 %
FIT 100 steps-ahead	86.3 %	88.0 %	90.0 %	82.5 %	86.0 %	89.8 %
FIT ∞ steps-ahead	72.6 %	75.2 %	75.2 %	66.0 %	72.0 %	77.6 %
Interval	**7**	**8**	**9**	**10**	**11**	**12**
FIT 1 step-ahead	93.3 %	93.3 %	92.9 %	43.0 %	92.7 %	92.9 %
FIT 100 steps-ahead	89.5 %	82.9 %	89.9 %	NaN	89.0 %	88.4 %
FIT ∞ steps-ahead	77.8 %	66.3 %	78.1 %	NaN	78.1 %	75.7 %

The average metrics are very consistent with the mining metrics, specially with the ones obtained through the Laguerre structure. In all scenarios, the condition number and the effective rank indicated intervals 9 and 12 as those with the highest numerical quality. On the contrary, interval 10 resulted in huge values of condition number and low values of effective rank for all model structures, indicating that this signal produces an information matrix very badly conditioned and potentially singular. From Table 16, one can clearly notice that, indeed, interval 10 could not produce any results for 100 and infinity steps-ahead predictions. Even in the 1 step-ahead case, this interval could not properly retrieve the model dynamics. In the same fashion, intervals 9, 11 and 12 resulted in the best

validation metrics when considering the simulation scenario (infinity steps-ahead), which is also consistent with the mining results.

As a final observation, looking at the infinity steps-ahead results, one can see that intervals 4 and 8 had the worst FIT values compared to the other intervals. The same intervals were the ones with the worst chi-squared values in Table 7. In fact, these intervals were the only intervals with chi-squared values lower than the critical value obtained for a significance level $\alpha = 0.01$. On the other hand, the chi-squared values produced with the ARX structure do not allow this discrimination, since they are all two high, indicating that the Laguerre structure better captures the numerical properties of the signal.

5.3.2 Statistical Method Examples

In this subsection, the statistical method proposed in (WANG *et al.*, 2018) and described in Subsection 3.3.3 is applied as in the algorithm in Figure 9. The first step of the algorithm is to segment the data using the non-parametric top-down change-point detection method exemplified in Subsection 5.2.4. In this example, the algorithm is applied to the water tank dataset described in Subsection 5.1.1. Moreover, similar analyses to those presented in (WANG *et al.*, 2018) are adopted in this subsection in order to validate the algorithm implementation before applying it to the petrochemical furnace dataset.

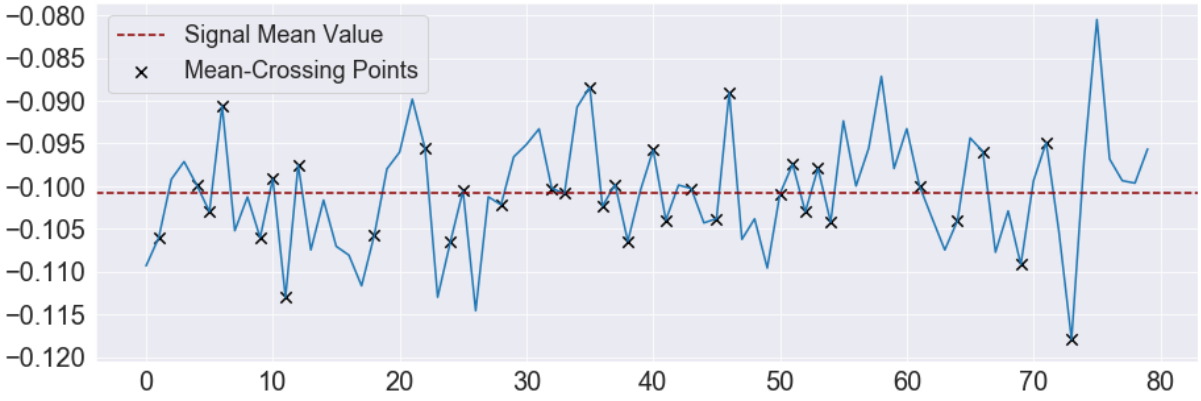
5.3.2.1 Non-parametric Kolmogorov-Smirnov (Lilliefors) Test

The first step of the algorithm consists of evaluating if each initial interval is sufficiently "active". As described in Subsection 3.3.3, a null hypothesis is created assuming that the signal is in steady state contaminated by random noise. If the null hypothesis is true, the statistic T_c defined in Subsection 3.3.3 has an exponential distribution.

In this dissertation, it was found that this formulation is only true if noisy intervals are not too long in time. That is because the statistic T_c assumes a much larger amount of small values as the data size increases considerable, deforming the exponential shape of the distribution. As an example, let us take a sample of 80 seconds of random noise with 0 mean and variance of 5, as shown in Figure 36. Notice that this signal is normalized and it is a sample extracted from Figure 32.

The distribution of the T_c statistic for this signal, as well as the resulting cumulative distribution function can be seen in Figure 37, both computed following the equations in Item 3.3.3.2. For a sample of computed statistics T_c with size higher than 30, the critical

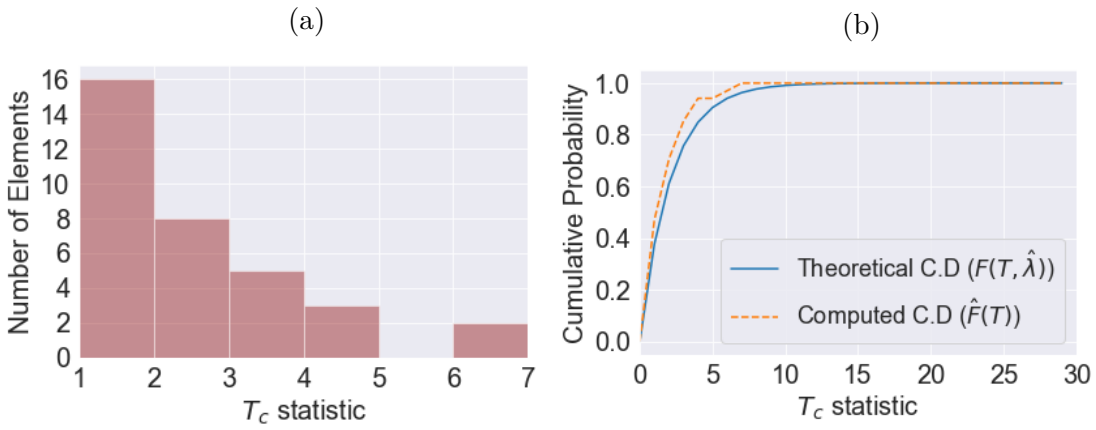
Figure 36: Mean-crossing points in a gaussian random noise signal with 0 mean and variance of 5.



Source: Author's own development.

value of the Lilliefors test for a significance level of $\alpha = 0.01$ is $1.25/\sqrt{N_T}$, being N_T the sample size of the computed T_c statistics (WANG *et al.*, 2018). For the random signal in Figure 36, 34 samples of T_c are produced and, therefore, the critical value is $D_c = 0.218$. If the computed D_t value is higher than $D_c = 0.218$, then the null hypothesis that T_c follows an exponential distribution can be rejected. Although the computed cumulative distribution in Figure 37 (b) does not fit perfectly the theoretical curve, the resulting test statistic is $D_t = 0.095$, way lower than the critical value. Therefore, the random signal can, indeed, accept the null hypothesis and to be considered to produce a statistic T_c that follows an exponential distribution.

Figure 37: Histogram and cumulative distribution function of statistic T_c for a normalized gaussian random noise with 0 mean and variance of 5. (a) Histogram (b) Cumulative distribution function.

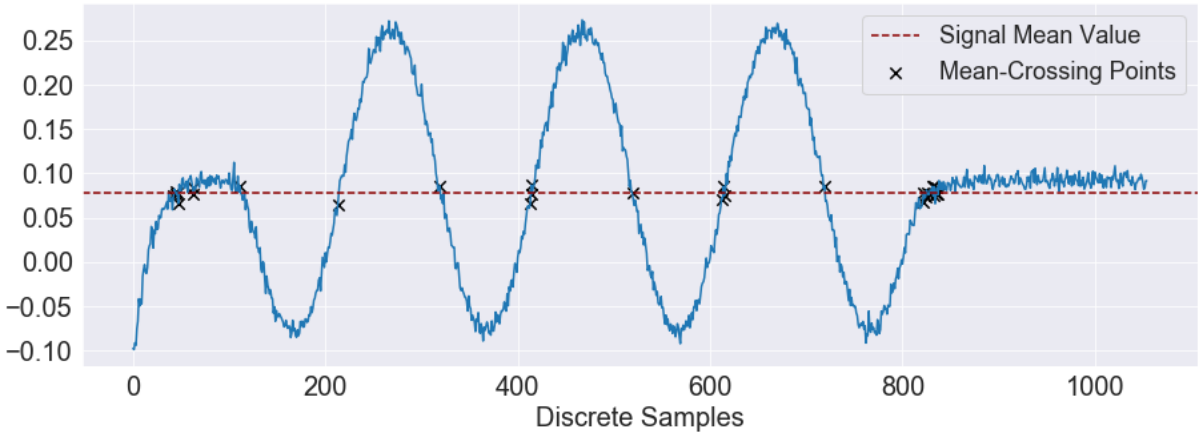


Source: Author's own development.

If we now consider, on the other hand, an active signal such as the one in Figure 38, the cumulative distribution function would be the one in Figure 39. In this case, one can

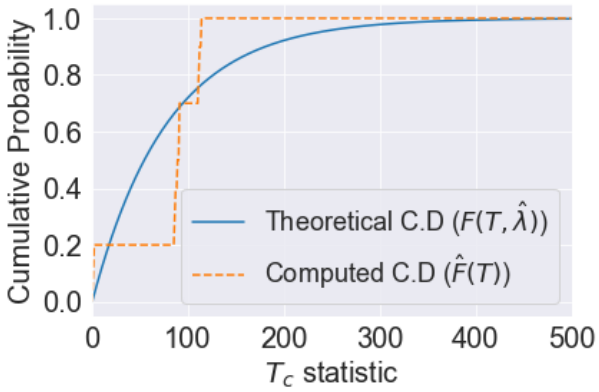
notice a strong mismatch between the theoretical and the computed curves. In fact, the computed test statistic is $D_t = 0.46$ against the critical value of $D_c = 0.38$, which, as expected, rejects the null hypothesis at a significance level $\alpha = 0.01$. One can verify that the results obtained in this item are in agreement with those presented in (WANG *et al.*, 2018).

Figure 38: Mean-crossing points in a sinusoidal interval.



Source: Author's own development.

Figure 39: Cumulative distribution function of statistic T_c for a sinusoidal interval.



Source: Author's own development.

5.3.2.2 Steps of the Algorithm

It is concluded in this dissertation that the statistical algorithm proposed in (WANG *et al.*, 2018) works better when the signal is over segmented. This is because, with lower intervals, the Lilliefors test works better in distinguishing active and non-active data. Moreover, there is no counterside in using smaller intervals, once these intervals are unified as they pass in the statistical tests until they form the final intervals. In fact, in

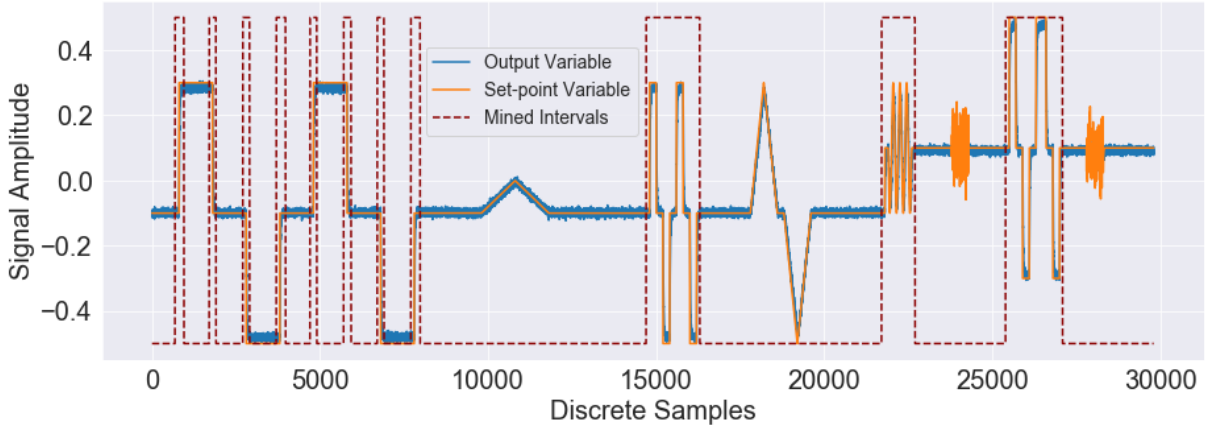
(WANG *et al.*, 2018) it is suggested the division of too long datasets in the way explained in Item 3.3.3.1.

Because the water tank system is operating under closed-loop, one must search for exciting intervals in the set-point, as explained in Item 3.3.2.7. Moreover, as explained in Item 3.3.3.4 and also in the algorithm outline in Figure 9, the method is applied individually to each variable (in this case, the set-point and the output variable), and then the resulting intervals are unified. Here, the initial intervals are obtained with the top-down non-parametric change-point algorithm exemplified in Subsection 5.2.4. A significance level of $\alpha = 0.05$ and a minimum split length of 1200 are considered, producing the exact same intervals as in Figure 32. Moreover, every initial interval higher than 200 samples was further divided following the approach described in Item 3.3.3.1.

Another important observation is that the set-point, by definition, is not contaminated by noise. This is a problem in the sense that the magnitude change test assumes the null hypothesis that the signal is in steady state disturbed by random noise. To get around this problem, a gaussian random noise of 0 mean and 0.01 standard deviation is included in the water-tank set-point, in such a way that the Lilliefors test can be applied to this signal.

Applying the algorithm to the water tank data considering a Lilliefors critical value of $1.25/\sqrt{N_T}$, a significance level $\alpha = 0.01$ for the difference in mean test and a difference in mean delta of $\Delta = 0.09$, the resulting mined intervals are the ones in Figure 40.

Figure 40: Resulting Mined intervals obtained through the statistical method with a Lilliefors critical value of $1.25/\sqrt{N_T}$, a significance level $\alpha = 0.01$ for the difference in mean test and a difference in mean delta of $\Delta = 0.09$.

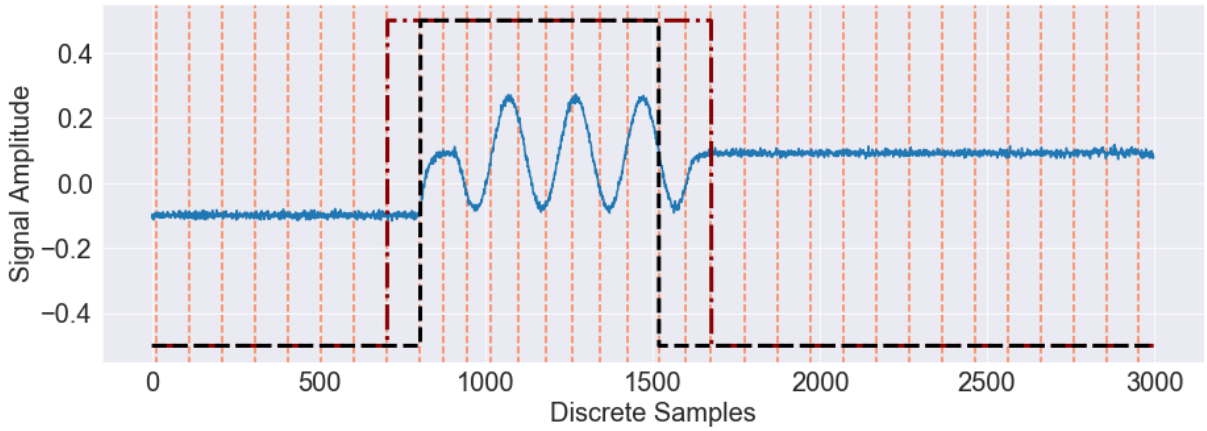


Source: Author's own development.

To clarify how the intervals were considered by the algorithm, Figure 41 highlights the algorithm steps for the sinusoidal signal in the output variable. The horizontal or-

ange dashed lines correspond to the initial intervals obtained through the non-parametric top-down change-point algorithm. The black dashed line delimits the resulting intervals that satisfied the magnitude change test, *i.e.*, the non-parametric Kolmogorov-Smirnov (Lilliefors) test. Finally, the two-mean t-student comparison test allowed the inclusion of one interval at the beginning and two intervals at the end of the interval delimited by the black dashed line, in such a way that the final interval is delimited by the red dashed line. One can also verify that this result is in agreement with the analysis presented in (WANG *et al.*, 2018).

Figure 41: Steps of the statistical algorithm. **Step 1**: change-point detection algorithm for a significance level $\alpha = 0.05$ (orange vertical dashed lines); **Step 2**: magnitude change statistical test for a Lilliefors critical value of $1.25/\sqrt{N_T}$ (black dashed line); **Step 3**: two-mean t-student comparison test for a significance level $\alpha = 0.01$ and a difference in mean delta of $\Delta = 0.09$ (red dashed line).

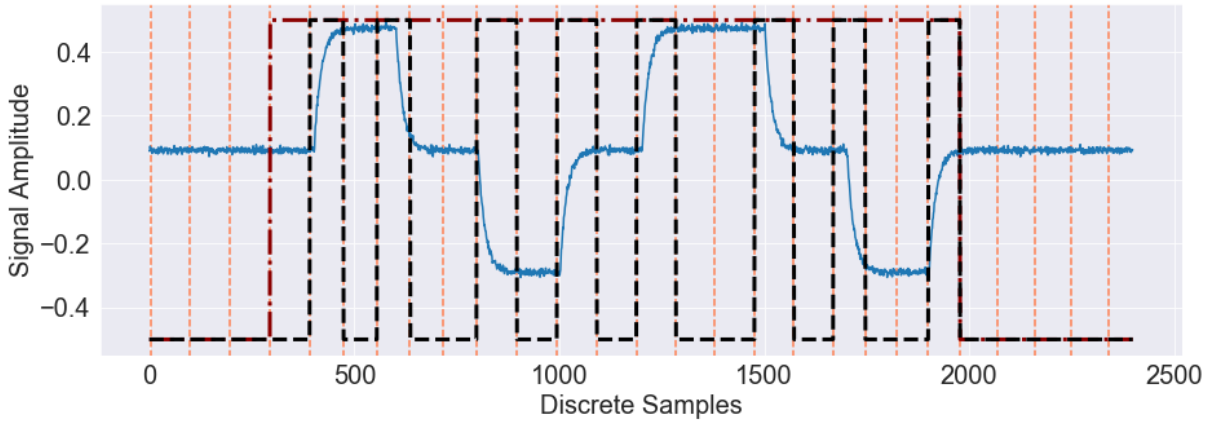


Source: Author's own development.

Another interesting interval to be closely analyzed is the last interval in Figure 40, which corresponds to sequential step responses. The algorithm steps for this interval can be seen in Figure 42. Notice that, in this case, the resulting intervals from Step 2 (black dashed line) were unified in Step 3, which reinforces how efficient is the algorithm even to unify intervals close to one another.

It is interesting to mention that, as pointed out in (WANG *et al.*, 2018), the gaussian white noise applied to the set-point is not considered as a final interval, although its magnitude is considerably high. This is because a white noise produces a T_c statistic that follows an exponential distribution by the algorithm definition and, therefore, it cannot reject the null hypothesis of the magnitude change test. In the same way, the ramp response in the set-point could not pass the magnitude change test, which is coherent with the results obtained with the numerical conditioning and rank test methods, which pointed out the ramp response as "ill-conditioned".

Figure 42: Steps of the statistical algorithm. **Step 1**: change-point detection algorithm for a significance level $\alpha = 0.05$ (orange vertical dashed lines); **Step 2**: magnitude change statistical test for a Lilliefors critical value of $1.25/\sqrt{N_T}$ (black dashed line); **Step 3**: two-mean t-student comparison test for a significance level $\alpha = 0.01$ and a difference in mean delta of $\Delta = 0.09$ (red dashed line).



Source: Author's own development.

Notice that although this process contains a dead time in its transfer function, the time delay ends up being considered. The reason for that lies in the final step of the algorithm, which unifies the resulting intervals obtained individually with the set-point and with the output variable. This step is described in detail in Item 3.3.3.4 and can be also seen in the algorithm outline in Figure 9.

Finally, an important observation is that this method require few parameters to be chosen. Moreover, these parameters do not require any knowledge or intuition about the process, once they are related to significance levels of statistical tests.

5.3.2.3 System Identification

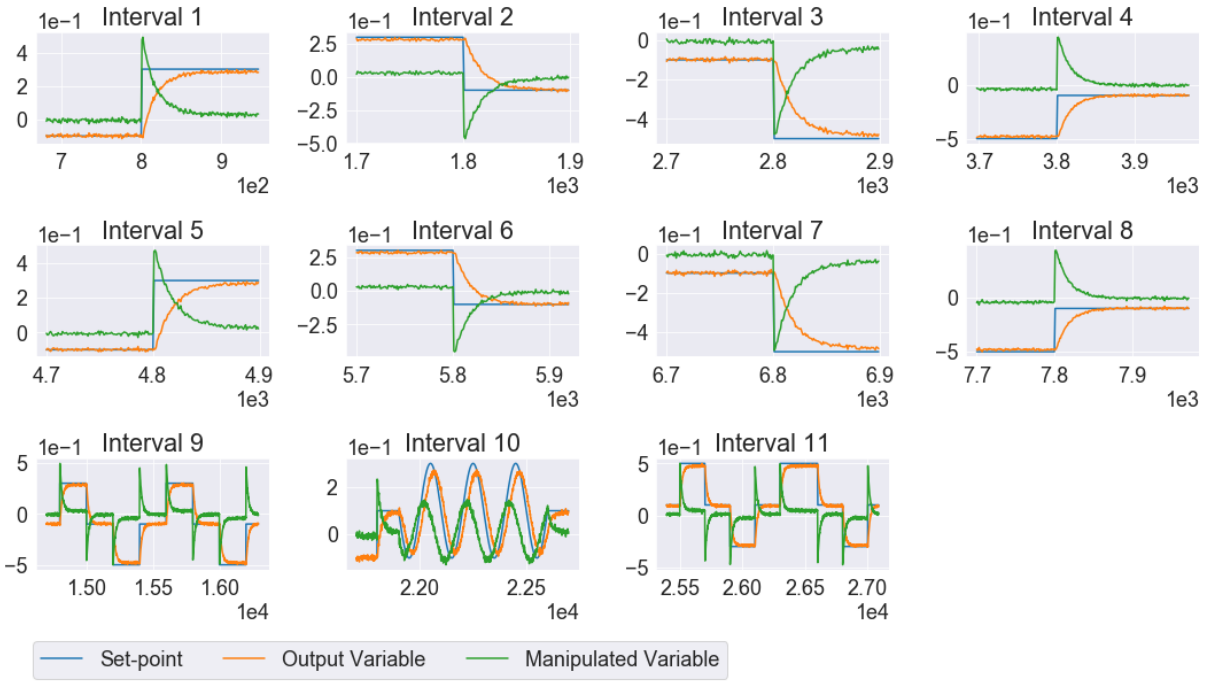
Similarly to Item 5.3.1.3, a model of the system is obtained with every interval through a cross-validation approach. The main idea in this application is to verify if, in fact, the resulting intervals are suitable to obtain a model of $G(q)$, considering that the system is operating in closed-loop control. Here, the system identification is also being done through the "Indirect Method".

The 11 resulting intervals in Figure 40 can be seen in detail in Figure 43. The average FIT value obtained through cross-validation can be seen in Table 17, where the same ARX structure with orders $n_u = 5$, $n_y = 3$ and $n_k = 1$ is used for every interval .

One can immediately notice that all resulting intervals are adequate for estimating a model of the system, considering all the prediction scenarios: 1 step-ahead, 100 steps-

ahead and infinity steps-ahead. This conclusion can be easily reached comparing the resulting metrics of each interval and noticing that they are very similar, *i.e.*, all intervals seem to capture well the dynamics of the water tank system. Notice, however, that Table 17 cannot be compared to Table 16. That is because Table 16 contains the average FIT value for all **potential intervals**, which include bad intervals such as the ramp signal, increasing the average prediction error. Table 17, on the other hand, is considering only the final mined intervals obtained through the statistical approach.

Figure 43: Detailed mined intervals obtained with the statistical method.



Source: Author's own development.

Table 17: Cross-validation average FIT value for each mined interval obtained through the statistical method.

Interval	**1**	**2**	**3**	**4**	**5**	**6**
FIT 1 step-ahead	96.32	96.10	96.10	96.14	96.11	96.11
FIT 100 steps-ahead	93.09	93.62	94.48	94.50	94.52	92.76
FIT inf steps-ahead	83.77	83.77	84.26	84.30	84.55	83.62
Interval	**7**	**8**	**9**	**10**	**11**	**-**
FIT 1 step-ahead	96.04	96.07	96.02	95.63	96.02	-
FIT 100 steps-ahead	94.03	94.22	94.61	93.98	94.24	-
FIT inf steps-ahead	84.05	83.94	84.85	84.88	84.79	-

5.4 Multiple-Input Multiple-Output (MIMO) Segmentation for Open-loop System Identification

5.4.1 Numerical Conditioning and Effective Rank Examples

As explained in Subsection 3.3.4 and in Figure 8, the multivariable extension for open-loop system identification consists of evaluating individually every input signal with every output signal and verify if at least one combination of input-output satisfies the numerical conditioning approach criteria or the effective rank approach criteria. In this example, the distillation column data from Subsection 5.1.2 is used. Let us also begin with the potential intervals in Figure 20, where the Exponentially Weighted filter is applied with $\lambda_\mu = \lambda_S = 0.006$ and a threshold of $l_S = 0.005$. Moreover, 50 initial indexes are considered, *i.e.*, $n_{idx} = 50$. Notice from Figure 20 that we have 5 potential intervals, that will called Intervals 1-5.

Applying a Laguerre Filter structure with order $N_b = 10$ and pole $\alpha = 0.92$ to every input-output combination, the condition number and the effective rank values can be seen in Table 18. The effective rank is computed using its Type 2 version with a singular value threshold of $l_2 = 0.5$. Moreover, U_1 is the reflux flow rate, U_2 is the steam flow rate, Y_1 the overhead composition and Y_2 the bottom composition.

Table 18: Condition number and effective rank for the multivariable distillation column, using a Laguerre Filter structure with order $N_b = 10$ and pole $\alpha = 0.92$ and considering the type 2 effective rank with a singular value threshold of $l_2 = 0.5$.

		Condition Number		**Effective Rank**	
		Y1	**Y2**	**Y1**	**Y2**
Interval 1	**U1**	3.8×10^3	3.8×10^3	9	9
	U2	∞	∞	0	0
Interval 2	**U1**	∞	∞	0	0
	U2	3.8×10^3	3.8×10^3	9	9
Interval 3	**U1**	1.0×10^4	1.0×10^4	9	9
	U2	∞	∞	0	0
Interval 4	**U1**	∞	∞	0	0
	U2	1.0×10^4	1.0×10^4	9	9
Interval 5	**U1**	1.0×10^4	1.0×10^4	9	9
	U2	1.0×10^4	1.0×10^4	9	9

Clearly, all intervals produce "well-conditioned" information matrices, with a condition number of magnitude 10^4, the same encountered in most of the reviewed works for step responses. Notice that for a given input, the condition number will be the same regardless of the output. That is because the Laguerre structure is only dependent on

the input variables. The chi-squared and the scalar cross-correlation values, on the other hand, are all different from each other and can be seen in Table 19. The scalar cross-correlation metric is computed within a delay range of $[-10,\ 10]$ and a significance level of $\alpha = 0.05$.

Table 19: Chi-squared values and Scalar Cross-Correlation values for each input-output pair in the multivariable data with a Laguerre Filter of order $N_b = 10$ and pole $\alpha = 0.92$ and with a cross-correlation delay range of $[-10,\ 10]$ and a significance level of $\alpha = 0.05$.

		Chi-squared Value		**Scalar Cross-Correlation**	
		Y1	**Y2**	**Y1**	**Y2**
Interval 1	**U1**	9.64×10^{2}	3.78×10^{2}	6.19×10^{0}	6.05×10^{0}
	U2	5.64×10^{-2}	1.38×10^{-1}	0.00×10^{0}	0.00×10^{0}
Interval 2	**U1**	2.57×10^{-1}	2.01×10^{-1}	0.00×10^{0}	0.00×10^{0}
	U2	6.79×10^{2}	5.93×10^{2}	6.33×10^{0}	6.44×10^{0}
Interval 3	**U1**	1.47×10^{3}	2.10×10^{-1}	6.67×10^{0}	6.59×10^{0}
	U2	5.84×10^{2}	7.20×10^{-1}	0.00×10^{0}	0.00×10^{0}
Interval 4	**U1**	1.30×10^{-1}	1.30×10^{-1}	0.00×10^{0}	0.00×10^{0}
	U2	1.04×10^{3}	7.94×10^{2}	6.58×10^{0}	6.63×10^{0}
Interval 5	**U1**	3.77×10^{2}	2.46×10^{2}	6.69×10^{0}	6.69×10^{0}
	U2	3.77×10^{2}	2.46×10^{2}	6.69×10^{0}	6.69×10^{0}

From Table 18 and Table 19, one can notice that the only interval that meets the condition number (or effective rank) and the chi-squared test (or cross-correlation test) for all input and output signals is interval 5. In this dissertation, the degree of coupling is treated as a parameter, *i.e.*, one could require, for example, that all signals must satisfy the algorithm criteria, in which case only interval 5 would be considered. However, the way this problem is treated in (PATEL, 2016) is by verifying if at least one input-output pair meets the algorithm criteria. This is because, as suggested in (PATEL, 2016) and explained in Item 3.3.4.2, one could use several intervals to compose the final model.

To illustrate how this method works, two different models of the system are obtained: the first model is obtained using intervals 1 and 2 and the second model is obtained using intervals 3 and 4. Both models are then validated using interval 5, which satisfies the condition number and the cross-correlation criteria for all input and output variables.

The resulting model produced with intervals 1 and 2 can be described through the following equations:

$$y_1^{it_1}(k) = G_{11}^{it_1}(q)u_1^{it_1}(k) + G_{12}^{it_1}(q)u_2^{it_1}(k) + H_1^{it_1}(q)v_1^{it_1}(k) \tag{5.6}$$

$$y_2^{it_1}(k) = G_{21}^{it_1}(q)u_1^{it_1}(k) + G_{22}^{it_1}(q)u_2^{it_1}(k) + H_2^{it_1}(q)v_2^{it_1}(k) \tag{5.7}$$

$$y_1^{it_2}(k) = G_{11}^{it_2}(q)u_1^{it_2}(k) + G_{12}^{it_2}(q)u_2^{it_2}(k) + H_1^{it_2}(q)v_1^{it_2}(k) \tag{5.8}$$

$$y_2^{it_2}(k) = G_{21}^{it_2}(q)u_1^{it_2}(k) + G_{22}^{it_2}(q)u_2^{it_2}(k) + H_2^{it_2}(q)v_2^{it_2}(k) \tag{5.9}$$

The final model is produced in the following manner: the $G(q)$ and $H(q)$ transfer functions are initially estimated; then, for interval 1, all transfer functions related to the input $u_2(k)$ are set to 0, once this input does not have an active input signal. In the same fashion, for interval 2, all transfer functions related to input $u_1(k)$ are set to 0. A model addition is then performed, resulting in the final model below:

$$y_1(k) = G_{11}^{it_1}(q)u_1(k) + G_{12}^{it_2}(q)u_2(k) + H_1(q)v_1(k) \tag{5.10}$$

$$y_2(k) = G_{21}^{it_1}(q)u_1(k) + G_{22}^{it_2}(q)u_2(k) + H_2(q)v_2(k) \tag{5.11}$$

An important observation is regarding the disturbance model. In this dissertation, $H_1(q)v_1(k)$ and $H_2(q)v_2(k)$ are considered as the disturbance models from Equations (5.6) to (5.9) that contain the highest gain. So, as an example, if $H_1^{it_1}(q)v_1^{it_1}(k)$ and $H_2^{it_2}(q)v_2^{it_2}(k)$ are the disturbance models with highest gain, then they are considered in the final model.

The same approach is applied with intervals 3 and 4 to obtain a second model of the system. As an example, let us consider the model obtained with intervals 1 and 2 for the first output ($y_1(k)$). An ARX structure is used with the following orders: for interval 1, the chosen orders are $n_y = 5$, $n_{u_1} = 1$, $n_{u_2} = 1$, $n_{k_1} = 1$ and $n_{k_2} = 1$; for interval 2, the chosen orders are $n_y = 5$, $n_{u_1} = 1$, $n_{u_2} = 1$, $n_{k_1} = 1$ and $n_{k_2} = 3$. Notice that the dead times are selected based on the plant real values, which can be seen in Subsection 5.1.2. Moreover, for interval 1, the $G_{12}(q)$ component is disregarded, while for interval 2 the $G_{11}(q)$ component is the one discarded, as it is clear in Equation (5.10) for output $y_1(k)$. The resulting transfer functions, for each interval, can be seen below:

Interval 1:

$$A(q) = 1 - 1.018q^{-1} + 0.006421q^{-2} + 0.04565q^{-3} - 0.04342q^{-4} + 0.05415q^{-5} \tag{5.12}$$

$$B(q) = 0.04179q^{-1} \tag{5.13}$$

Interval 2:

$$A(q) = 1 - 0.9734q^{-1} - 0.004253q^{-2} - 0.004797q^{-3} + 0.01039q^{-4} + 0.01153q^{-5} \quad (5.14)$$

$$B(q) = -0.02728q^{-3} \quad (5.15)$$

Finally, both models are unified to produce the resulting model for output $y_1(k)$, as shown below:

$$B_{11}(q) = 0.04179q^{-1} \quad (5.16)$$

$$B_{12}(q) = -0.02728q^{-3} \quad (5.17)$$

$$D(q) = 1 - 1.018q^{-1} + 0.006421q^{-}2 + 0.04565q^{-3} - 0.04342q^{-4} + 0.05415q^{-5} \quad (5.18)$$

$$F_1(q) = 1 - 1.018q^{-1} + 0.006421q^{-2} + 0.04565q^{-3} - 0.04342q^{-4} + 0.05415q^{-5} \quad (5.19)$$

$$F_2(q) = 1 - 0.9734q^{-1} - 0.004253q^{-2} - 0.004797q^{-3} + 0.01039q^{-4} + 0.01153q^{-5} \quad (5.20)$$

Notice that this model follows the general structure presented in Equation (2.1). Moreover, the disturbance model is already considered in this equation and it is represented by $D(q)$, which is equivalent to the disturbance model of Interval 1, *i.e.*, $D(q)$ in the final model is the same as $A(q)$ in Interval 1.

If one takes the resulting models obtained with intervals 1-2 and intervals 3-4, for both outputs, and performs a cross-validation with interval 5, the FIT values in Table 20 are obtained. In the same fashion, Figure 44 shows a comparison of $y_1(k)$ and $y_2(k)$ outputs from Interval 5 with its predictions for 1, 10 and infinity steps-ahead.

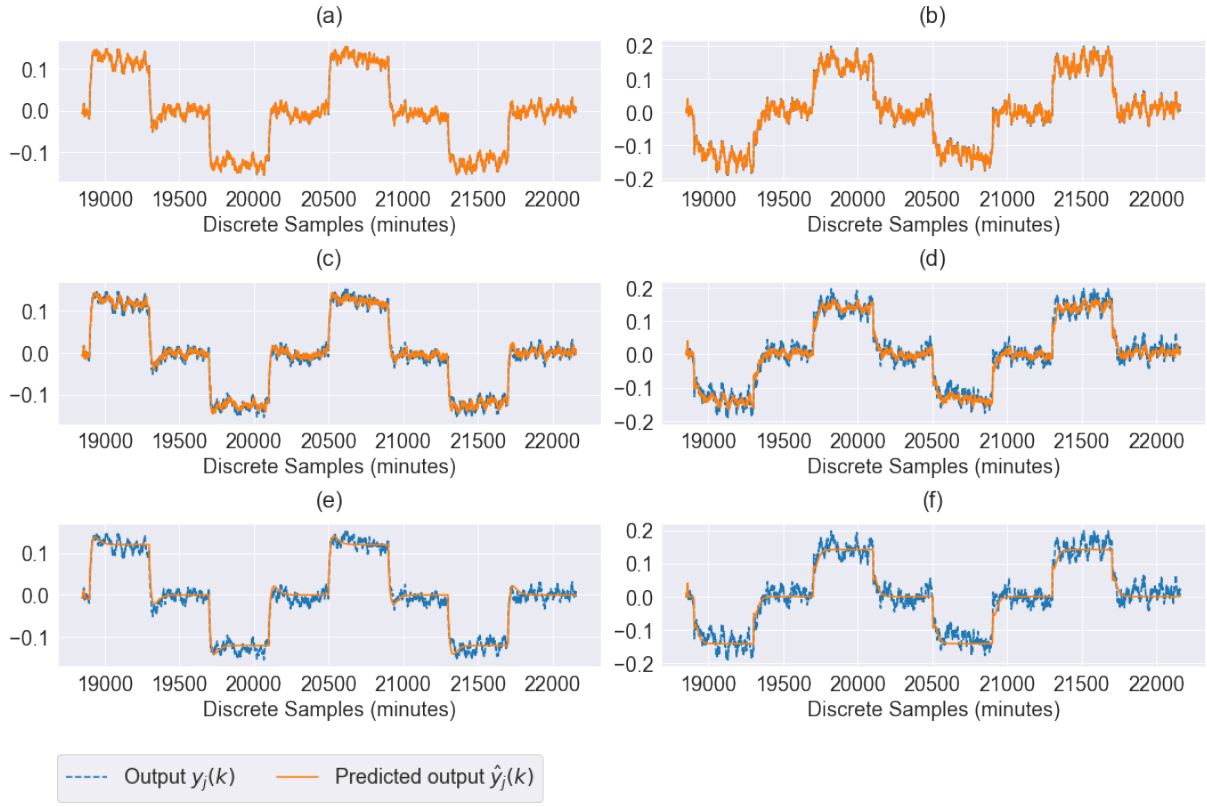
Table 20: Cross-validation FIT values for Model 1 and Model 2, with validation being performed in Interval 5.

		1 step-ahead	**10 steps-ahead**	**∞ steps-ahead**
Model 1:	$y_1(k)$	94.43	86.76	84.58
Intervals 1-2	$y_2(k)$	91.49	79.97	76.91
Model 2:	$y_1(k)$	94.44	86.75	84.55
Intervals 3-4	$y_2(k)$	91.50	79.78	76.57

It is clear from the results that, indeed, it can be useful to consider intervals where a single input is persistently exciting and then combine multiple intervals to reach the final system dynamics. In fact, Table 20 and Figure 44 elucidate that Model 1 (which is obtained with both Interval 1 and Interval 2) and Model 2 (which is obtained with both Interval 3 and Interval 4) are able to capture the system dynamics. This is guaranteed through validation of the predictions in Interval 5, which is an interval where both input signals are "shaking". Finally, it is worth mentioning that these results are in line with

those obtained in (PATEL, 2016).

Figure 44: Comparison of $y_1(k)$ and $y_2(k)$ outputs from Interval 5 with its 1 step-head, 10 steps-ahead and free-run predictions. (a) $y_1(k)$ and $\hat{y}_1(k)$ for 1 step-ahead prediction. (b) $y_2(k)$ and $\hat{y}_2(k)$ for 1 step-ahead prediction. (c) $y_1(k)$ and $\hat{y}_1(k)$ for 10 steps-ahead prediction. (d) $y_2(k)$ and $\hat{y}_2(k)$ for 10 steps-ahead prediction. (e) $y_1(k)$ and $\hat{y}_1(k)$ for free-run prediction. (f) $y_2(k)$ and $\hat{y}_2(k)$ for free-run prediction.



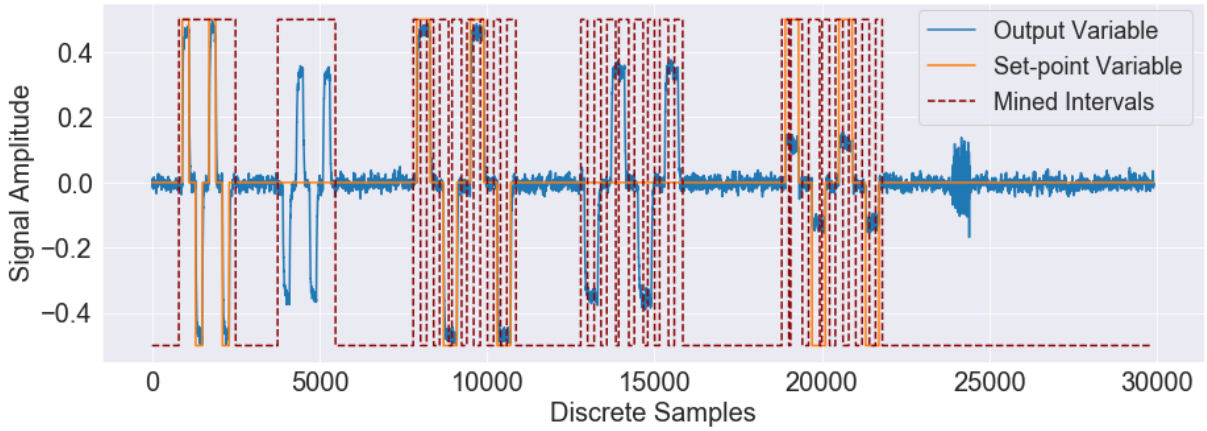
Source: Author's own development.

5.4.2 Statistical Method Examples

A multivariable extension of the statistical method is proposed in Item 4.1.2.2. As explained in item 4.1.2.2, the extension consists of applying the statistical method to every signal in the process and then unifying the resulting intervals.

Let us initially require that at least one input-output pair must satisfy the statistical criteria. Moreover, let us apply the non-parametric top-down change-point algorithm to every signal in the Wood & Berry dataset, assuming a significance level of $\alpha = 0.05$, a minimum length to split of 500 samples and a maximum sample size of 100. In this scenario, if one applies the statistical segmentation considering a Lilliefors critical value of $1.25/\sqrt{N_T}$ and a significance level for the t-student test of $\alpha = 0.01$, the final intervals in Figure 45 are obtained.

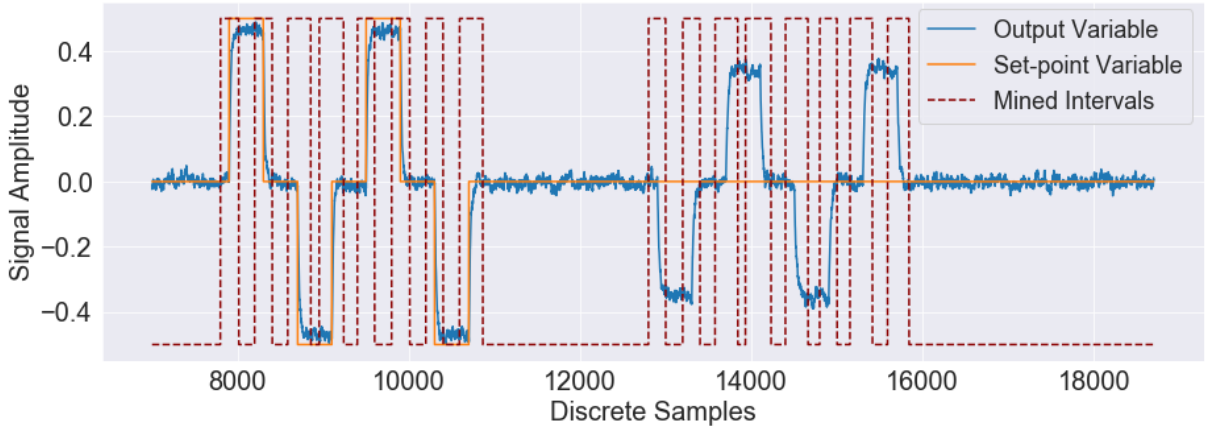
Figure 45: Mined intervals obtained with the multivariable extension to the statistical method, requiring that at least one input-output pair meets the statistical method criteria.



Source: Author's own development.

Notice that the step responses that are close to each other end up being unified, while those that are separated by a larger number of samples are considered individually. A better visualization of this scenario can be seen in Figure 46, which focus the samples in the range of [7000, 18700] minutes.

Figure 46: Mined intervals obtained with the multivariable extension of the statistical method in the range of [7000, 18700] minutes.

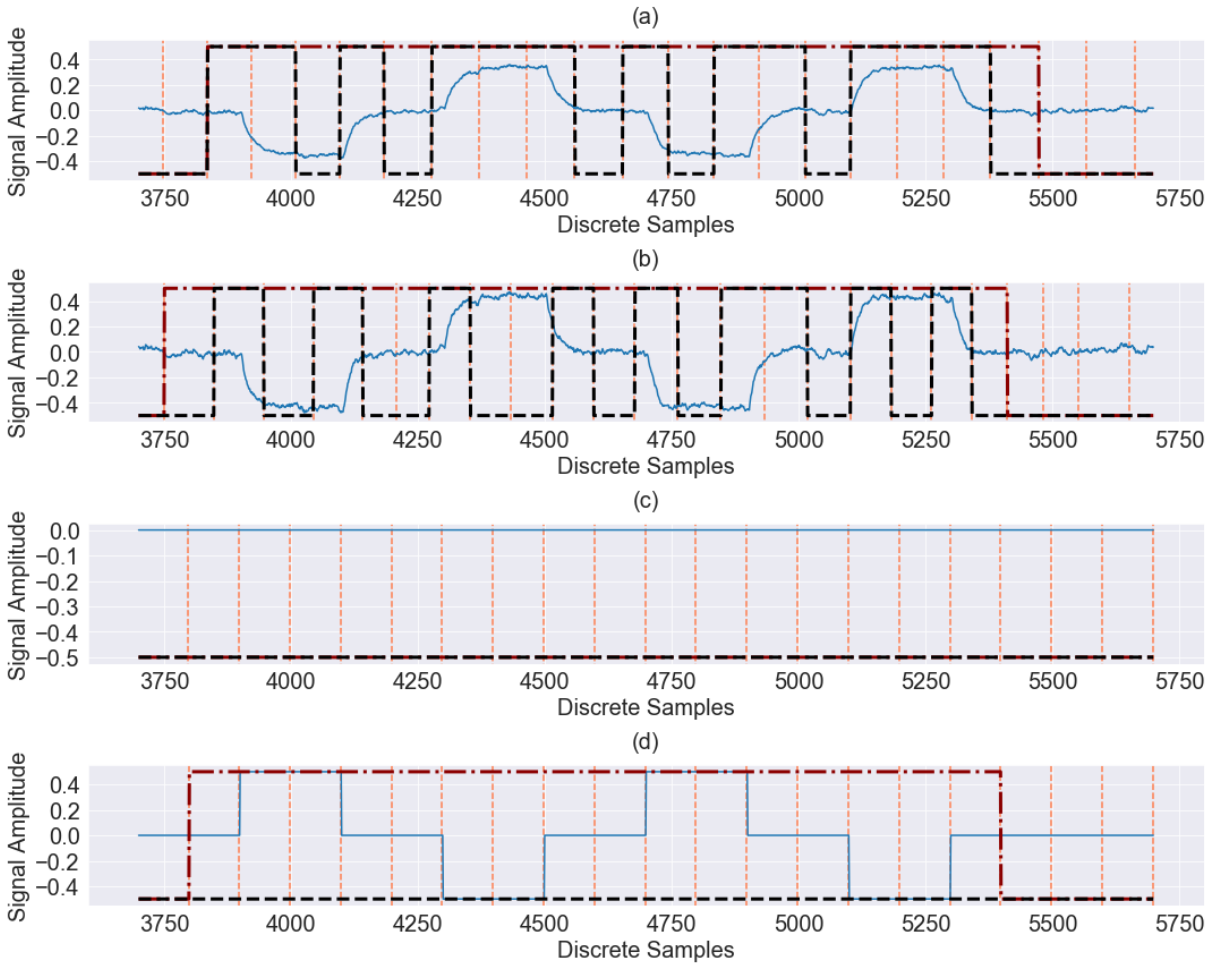


Source: Author's own development.

In order to highlight how these intervals end up being considered by the algorithm, an example of the algorithm steps can be seen in Figure 47. Notice that after unifying all the resulting indicating sequences, an interval in the range [3751, 5473] is obtained. Inside this interval, the signal $u_1(k)$ is not active at all. However, the signal $u_2(k)$ produces an active interval in the range [3800, 5398]. Because [3800, 5398] $\subset$ [3751, 5473], one can conclude that the final interval contains at least one active signal in it and, therefore, the final interval [3751, 5473] is considered a suitable interval for system identification.

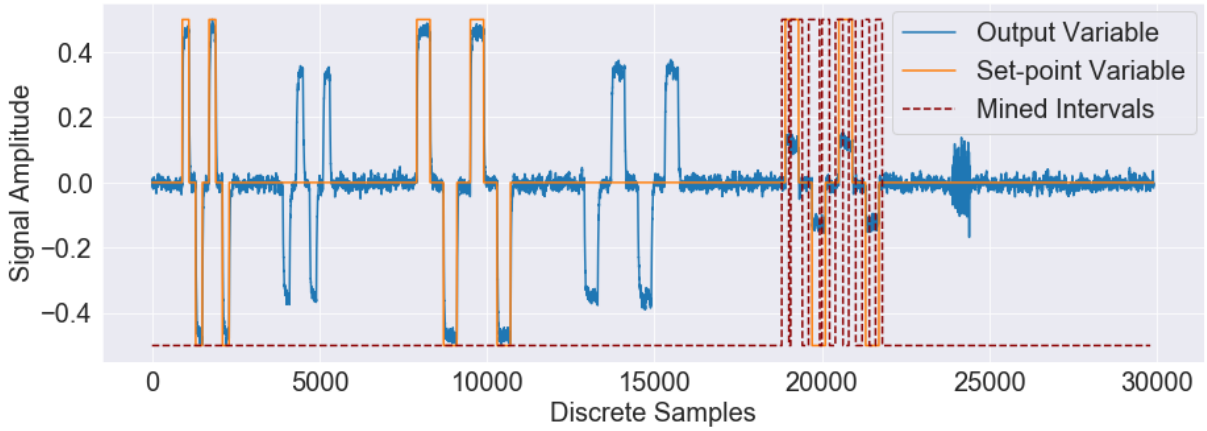
As explained in Subsection 5.4.1, requiring that only one input-output pair must meet the statistical criteria may be sufficient in the sense that one could use multiple intervals to obtain the final model of the process. However, one could also requires a more restrictive condition. As an example, let us require that all input and all output must meet the statistical method criteria. In this case, the resulting mined intervals are those in Figure 48, which correspond to the exact moments where both inputs received step responses at the same time.

Figure 47: Steps of the statistical algorithm. **Step 1**: change-point detection algorithm for a significance level $\alpha = 0.05$ (orange vertical dashed lines); **Step 2**: magnitude change statistical test for a Lilliefors critical value of $1.25/\sqrt{N_T}$ (black dashed line); **Step 3**: two-mean t-student comparison test for a significance level $\alpha = 0.01$ and a difference in mean delta of $\Delta = 0.1$ (red dashed line). (a) Output signal $y_1(k)$ (blue solid line). (b) Output signal $y_2(k)$ (blue solid line). (c) Input signal $u_1(k)$ (blue solid line). (d) Input signal $u_2(k)$ (blue solid line).



Source: Author's own development.

Figure 48: Mined intervals obtained with the multivariable extension of the statistical method, requiring that all input and output signals meet the statistical method criteria.



Source: Author's own development.

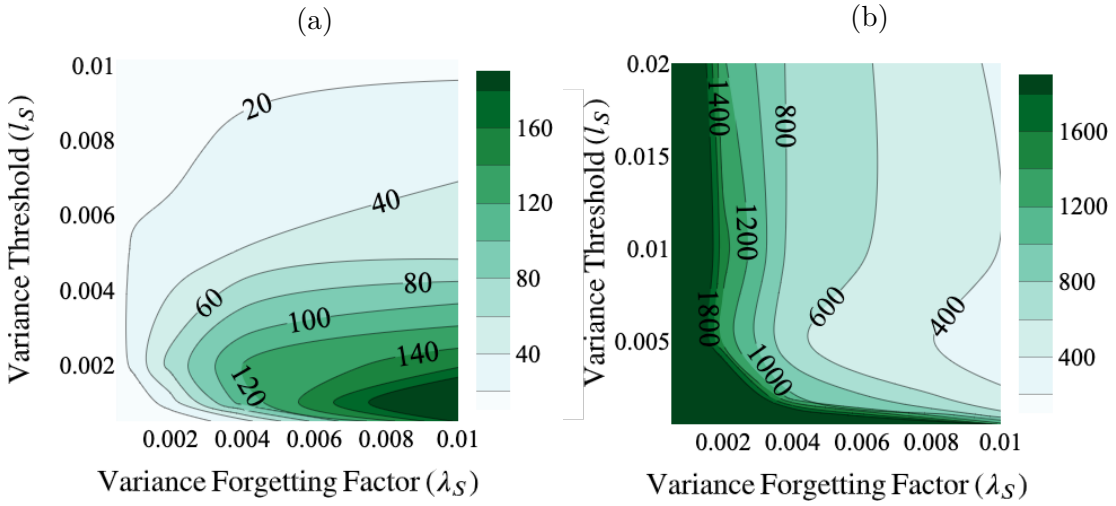
5.5 Application to a Real Process Multivariable Data

In this Section, the Petrochemical Furnace data is used to exemplify how one could obtain models from historical data in a real scenario. The dataset is divided into two blocks: the first one contains 1 month of data and the second one contains 7 months of data, as explained in Subsection 5.1.3. The numerical conditioning algorithm is first applied and the resulting intervals are used to obtain a model of the system. Then, the effective rank and the statistical method are applied and the resulting intervals are compared.

5.5.1 Numerical Conditioning and Effective Rank

In this subsection, the numerical conditioning and the effective rank approaches are explored in both blocks of data. Initial intervals are obtained using the Exponentially Weighted filter for the multivariable case, exemplified in Item 5.2.1.2. In order to choose the correct parameters for the filter, two heatmaps are drawn as a function of the forgetting factors and variance threshold, one for the number of resulting intervals and another for the length of the same intervals. Both visualizations can be seen in Figure 49, which are obtained with the 7 months data block. Notice that it is assumed, for simplification, that $\lambda_\mu = \lambda_S$ and that all signals in Table 2 receive the same forgetting factors. Choosing individual λ_μ, λ_S and l_S parameters for each individual signal is an exhausting procedure and can be bypassed through this simplification.

Figure 49: Number and length of potential intervals as a function of the forgetting factors and the variance threshold, assuming $\lambda_\mu = \lambda_S$. (a) Number of potential intervals. (b) Length of potential intervals.



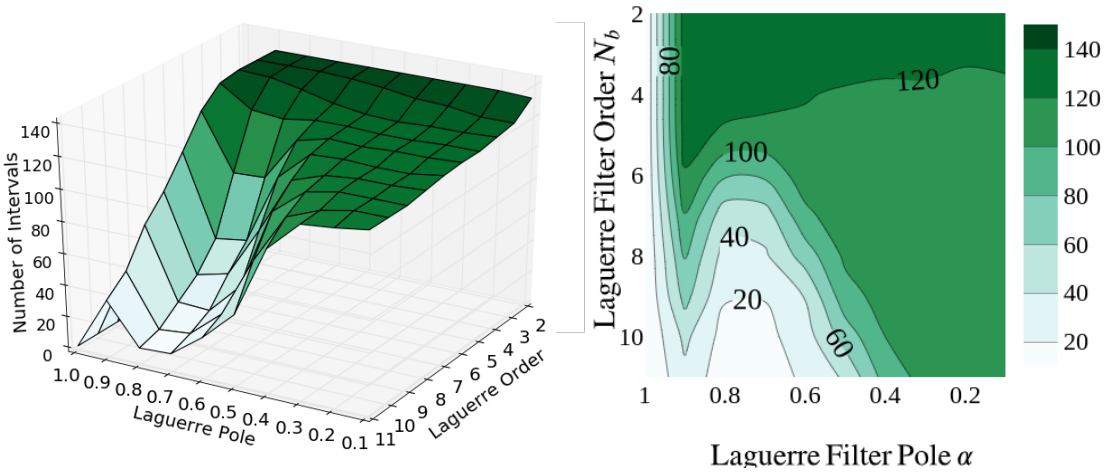
Source: Author's own development.

5.5.1.1 Numerical Conditioning: Laguerre Filters

The numerical conditioning approach is applied to each potential interval using the Laguerre structure. One must decide the Laguerre pole and order to be used when mining intervals suitable for system identification. It was shown in this dissertation that this choice is not trivial. The surface and heatmap plots in Figure 50 are used here to make this choice easier. Notice that when the filter pole is located at 1, the algorithm does not return any interval. In a similar fashion, too lower filter poles ($\alpha < 0.6$) produce a region where the number of mined intervals are all about the same, no matter the filter order. It seems that the most interesting region is located around the range [0.9, 0.95] for the filter pole and around the range [6, 11] for the filter order. In this region, one can be more restrictive and produce fewer mined intervals, or one could choose a lower filter order and obtain more intervals.

To obtain the final intervals, the Exponentially Weighted filter is applied with $\lambda_\mu = \lambda_S = 0.01$, $l_s = 0.002$ and $n_{idx} = 20$ for all variables, resulting in 97 potential intervals with an average length between 400 and 600, as shown in Figure 49 (b). However, a maximum interval length of 400 samples is imposed, such that intervals longer than 400 minutes are divided as described in Item 3.3.3.1, resulting in 223 potential intervals. The numerical conditioning approach is finally applied with a condition number threshold of $l_\kappa = 5000$, a chi-squared significance level of $\alpha = 0.01$, a Laguerre Filter pole of $\alpha = 0.9$ and a Laguerre Filter order $N_b = 8$. The chi-squared critical value for a filter with order 8

Figure 50: Impact of the Laguerre Filter pole (α) and order (N_b) in the number of mined intervals for the Petrochemical Furnace dataset, considering the potential intervals obtained through the Exponentially Weighted filter with $\lambda_\mu = \lambda_S = 0.005$ and $l_s = 0.002$ for all the furnace variables. The resulting intervals are obtained considering a condition number threshold of $l_\kappa = 20000$ and a chi-squared test significance level of $\alpha = 0.01$.



Source: Author's own development.

and a significance level of 0.01 is $\chi_{8,0.01} = 20.1$. Notice that a more restrictive condition number threshold is considered compared to the one used to obtain Figure 50. Finally, because the Petrochemical Furnace dataset is widely contaminated by noise and it is sampled in minutes, to obtain better intervals it is imposed that at least two inputs must satisfy the method criteria with at least one output data, which is more restrictive than the original requirement (at least one input-output pair). A summary of the parameters adopted can be seen in Table 21.

With these parameters, the mining algorithm produces 19 final intervals suitable for system identification, 16 coming from the the 7 months block of data and 3 coming from the 1 month block. The final intervals are here called Intervals 1-19.

It is interesting to mention that the multivariable problem is extremely challenging in the sense that it is very unlikely that the three set-points of the petrochemical furnace have changed at the same time, in any moment in the past, with enough magnitude to produce the complete dynamic response of the system. This is why it is proposed in (PATEL, 2016) the requirement that at least one input-output pair must satisfy the algorithm conditions. As exemplified in Subsection 5.4.1, one can then use multiple intervals to obtain the complete dynamics of the process. The way multiple intervals are used to obtain the final model in this example can be seen in Table 22.

In order to validate the resulting model in a fair way, two completely different intervals

Table 21: Numerical conditioning algorithm parameters applied in the petrochemical furnace dataset.

Exponentially Weighted Filter Parameters	**Values**
λ_μ	0.01
λ_S	0.01
l_S	0.002
n_{idx}	20
Maximum Length	400
Laguerre Structure Parameters	**Values**
N_b	8
Laguerre pole α	0.9
Chi-squared α	0.01
l_κ	5000
Approval Criteria	At least two inputs and one output variable

Table 22: Mined historical intervals used to estimate a model of the Petrochemical Furnace.

		Set-points		
		FIC-23027-SP	**FIC-23028-SP**	**FIC-23025-SP**
Output Variables	**TIC-23099**	Interval 1	Interval 2	Interval 3
	PIC-23039	Interval 1	Interval 2	Interval 3
	AIC-23001	Interval 1	Interval 2	Interval 3

are used as the validation dataset, as shown in Table 23. The estimation intervals can be seen in Figure B.1 and the validation intervals can be seen in Figure B.2. The reasons behind the choice of the estimation intervals is that they all have good chi-squared and condition number values and, also, because all variables in Intervals 1-3 are around the same operating condition (see the operating values in Table 2). Other resulting intervals, although approved in the mining algorithm, are centralized in different operating points, some of them corresponding to stopping moments of the plant, as can be seen through Interval 6 shown in Figure B.3. Table 24 shows the corresponding condition numbers for each input variable in Intervals 1-3, while Table 25 shows the chi-squared values for each input-output pair in Intervals 1-3.

Table 23: Mined historical intervals used as the validation dataset.

Output Variable	TIC-23099	PIC-23039	AIC-23001
Validation Interval	Intervals 4-5	Intervals 4-5	Intervals 4-5

In order to evaluate the quality of the obtained model through the described cross-

Table 24: Condition number values obtained with Intervals 1-3, with the values corresponding to the adopted intervals being highlighted in blue.

	Set-points		
	FIC-23027-SP	**FIC-23028-SP**	**FIC-23025-SP**
Interval 1	1426.95	3007.15	49973895.25
Interval 2	1841.63	1337.58	1005903.73
Interval 3	729.82	1688.67	424.61

Table 25: Chi-squared values obtained with Intervals 1-3, with the values corresponding to the adopted intervals being highlighted in blue.

		Set-points		
		FIC-23027-SP	**FIC-23028-SP**	**FIC-23025-SP**
Interval 1	**TIC-23099**	63.76	44.5	893.0
	PIC-23039	31.81	30.27	24.47
	AIC-23001	58.12	55.14	58.03
Interval 2	**TIC-23099**	135.74	127.06	496.68
	PIC-23039	48.75	46.95	157.79
	AIC-23001	71.90	77.47	64.11
Interval 3	**TIC-23099**	8.76	13.42	7.09
	PIC-23039	27.21	21.97	16.00
	AIC-23001	19.88	15.95	18.10

validation approach, three different metrics are chosen: the Root Mean Squared Error (RMSE), the FIT index, which is based on the percentual Normalized Root Mean Squared Error (NRMSE), and the R^2 score. The FIT index is the same as the one defined in Item 5.3.1.3. The RMSE and the R^2 are defined as follows:

$$RMSE = \sqrt{\frac{1}{N_s}\sum_{k=1}^{N_s}(\hat{y}(k) - y(k))^2} \tag{5.21}$$

$$R^2 = 1 - \frac{\sum_{k=1}^{N_s}(\hat{y}(k) - y(k))^2}{\sum_{k=1}^{N_s}(\bar{y} - y(k))^2} \tag{5.22}$$

where $\bar{y}$ is the mean of the output signal, $\bar{y} = \frac{1}{N_s}\sum_{k=1}^{N_s} y(k)$, and $\hat{y}$ is the predicted output.

In order to obtain the final model for a given output variable, ARX models are used following the approach described in Item 3.3.4.2. Because multiple ARX models that come from different intervals are added, the resulting model has the Box-Jenkins structure. Such a structure can be easily obtained making $A(q) = 1$ in Equation (2.1). Moreover, considering the MISO structure as defined in Figure 2, the Box-Jenkins structure can be

reformulated as:

$$y_i(k) = \sum_{i=1}^{n_u} \frac{B_i(q)}{F_i(q)} r_i(k) + \frac{C(q)}{D(q)} v(k) \tag{5.23}$$

The obtained model orders for each MISO system can be seen in Table 26, where n_k is the input delay order and n_b, n_f and n_d are, respectively, the $B(q)$, $F(q)$ and $D(q)$ polynomial orders.

Table 26: Box-Jenkins orders for each output variable in the petrochemical furnace.

		Box-Jenkins Model Order			
		$\boldsymbol{n_b}$	$\boldsymbol{n_f}$	$\boldsymbol{n_d}$	$\boldsymbol{n_k}$
Output Signals	**TIC-23099**	[9 4 5]	[1 6 2]	2	[1 10 2]
	PIC-23039	[7 5 9]	[1 7 3]	1	[9 3 1]
	AIC-23001	[3 6 1]	[1 1 1]	1	[4 5 1]

Notice that, because the final models are originated through ARX structures, we have $C(q) = 1$ in Equation (5.23). The resulting validation metrics can be seen in Table 27 and a comparison of the validation data with the corresponding predictions can be seen in Figures C.1, C.2 and C.3.

Table 27: Cross-validation metrics of each model, considering Intervals 4 and 5.

		∞-steps ahead			**1-step ahead**		
		RMSE	**R²**	**FIT**	**RMSE**	**R²**	**FIT**
Interval 4	**TIC-23099**	1.68	0.77	51.8	0.38	0.98	89.01
	PIC-23039	0.10	0.95	78.77	0.02	0.99	96.50
	AIC-23001	0.24	0.61	37.51	0.11	0.92	72.60
Interval 5	**TIC-23099**	4.09	0.58	35.50	0.57	0.99	90.76
	PIC-23039	0.28	0.77	52.40	0.02	0.99	95.87
	AIC-23001	0.51	0.05	2.65	0.11	0.96	79.00

Notice that the FIT value is a percentual value, while the R^2, being the squared value of the Pearson's coefficient, goes from 0 to 1. The higher is the R^2 and the FIT value, the better is the model explanation of the signal compared to its residue. On the other hand, the $RMSE$ is a measure of the prediction error and, therefore, a low value indicates a good model fitness to the validation data. It is important to mention that the $RMSE$, being an error measure, is proportional to the signal magnitude. Therefore, this value must be compared to the operating value of the plant in order to have an idea of the error magnitude. It is expected, for example, that the error of the TIC-23099 model is greater than the error of the PIC-23039 and AIC-23001 models, given that the nominal value of TIC-23099 is more than 100 times greater than that of the other two variables.

One can notice that the validation results for the AIC-23001 are specially bad. The main reason for that is the fact that the AIC-23001 output is highly affected by the FIC-23025-SP set-point, which is in steady-state for both Intervals 4 and 5. In fact, because this particular set-point is manipulated by an optimizer, it is highly disturbed by noise when in steady-state. Let us now apply the same model to a different validation data: Interval 7, shown in Figure B.4. In this case, the resulting prediction can be seen in Figure C.4 and the resulting metrics can be seen in Table 28. It is clear that, for this signal, the model captures better the AIC-23001 dynamics.

Table 28: Cross-validation metrics for AIC-23001 model, considering Interval 7.

		Infinity-steps ahead			**1-step ahead**		
		RMSE	**R²**	**FIT**	**RMSE**	**R²**	**FIT**
Interval 7	**AIC-23001**	0.65	0.89	66.74	0.17	0.99	91.21

It is possible to conclude that the mining strategy presented in this item is able to capture informative and representative data of the system, respecting the operating points and, therefore, the hypothesis of linearity. Moreover, reasonable models were obtained for each output variable, specially considering that simple ARX models were adopted. This is confirmed by the validation metrics, in which the FIT and the R^2 metrics assumed intermediate values for all outputs in two different validation data. As a final observation, it is important to mention that simple ARX models are here used to exemplify how the mining algorithm can actually produce models from historical data. However, one must have in mind that more complex solutions can definitely be used to obtain better results, regardless of the mining strategy used. In fact, some of the resulting data, such as Interval 3, could actually, in a practical sense, be used alone to obtain a final model, without the need to combine it with further data.

5.5.1.2 Effective Rank: AR Structure

In this item, the effective rank and the scalar cross-correlation metrics are used in order to compare the resulting intervals with those obtained in the previous item. The AR structure is here adopted as originally proposed in (RIBEIRO; AGUIRRE, 2015). To make the comparison of the results more interesting, the same potential intervals from the previous item are here considered and evaluated through the effective rank and the cross-correlation criteria. Therefore, for the 7 months dataset 223 potential intervals must be evaluated, while for the 1 month data 98 potential intervals are available.

As explained in Subsection 5.3.1, the effective rank is a less sensitive metric compared

to the condition number, in such a way that higher model orders lead to better results. Therefore, as suggested in (RIBEIRO; AGUIRRE, 2015), an AR structure of order 100 is here applied. A summary of all the algorithm parameters used in this item can be seen in Table 29.

Table 29: Effective rank algorithm parameters applied to the petrochemical furnace dataset.

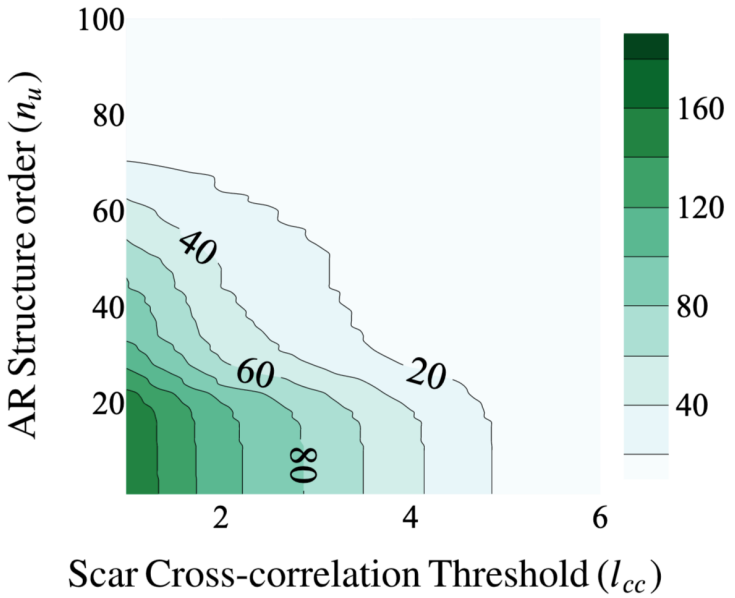
Exponentially Weighted Filter Parameters	**Values**
λ_μ	0.01
λ_S	0.01
l_S	0.002
n_{idx}	20
Maximum Length	400
AR Structure Parameters	**Values**
AR order n_u	100
Type 2 effective rank threshold	0.01
Cross-correlation delay range	$[-10, 10]$
Cross-correlation significance level	0.05

To better understand the impact of the choice of both the effective rank and the cross-correlation thresholds in the number of mined intervals, Figure 51 was created. It is interesting to notice that both criteria are very aggressive, in such a way that if one chooses too high values for both the effective rank and the cross-correlation, almost no interval is considered. Moreover, it is also important to point out that the resulting correlation values are considerably slow, which can probably be explained by the fact that the data is extremely noisy.

To obtain the final intervals, the adopted conditions are those shown in Table 30. Notice that a very restrictive condition is applied to the cross-correlation, considering the delay range adopted. The idea is to obtain the fewer intervals as possible, as was done in the previous item. Applying the algorithm in Figure 8 with this conditions to both blocks of data, a total of 17 intervals are obtained: 11 with the 7 months block of data and 6 with the 1 month block. As expected, some of these intervals are exactly the same as those mined with the Laguerre filter in the previous item. More specifically, 5 of these intervals are identical, three of them already shown in Figures B.3, B.4 and B.1 (c).

The main reason why some of the resulting intervals obtained with the Laguerre algorithm in the previous item were not here considered permeates the scalar cross-correlation test. The correlation test seems to be more aggressive than the chi-squared causality test.

Figure 51: Impact of the choice of the effective rank and the cross-correlation thresholds on the number of mined intervals.



Source: Author's own development.

Table 30: Approval conditions for the effective rank algorithm applied to the petrochemical furnace.

Variables	**Values**
Type 2 effective rank threshold	$l_2 = 25$
Scalar cross-correlation threshold	$l_{cc} = 6.4$
Approval Criteria	At least two inputs and one output variable

To ilustrate this point, let us consider Interval 1 in Figure B.1 (a). Evaluating why this interval was not considered by the effective rank algorithm, one can notice that it is precisely the correlation test that failed against the $l_{cc} = 6.4$ threshold. In fact, Table 31 shows the resulting correlation values for this signal. It is clear that only the signals PIC-23039 and AIC-23001 show some degree of correlation with the set-point through the scalar cross-correlation metric, but these values are lower than $l_{cc} = 6.4$. The same is true for the majority of the intervals obtained through the Laguerre algorithm and that were not considered by the AR algorithm.

If we now look at the intervals that are considered by the effective rank algorithm but not by the numerical conditioning algorithm in the previous item, we can see that the main difference lies in the condition number of these intervals. In fact, because the AR structure

Table 31: Scalar cross-correlation values for Interval 1 in Figure B.1 (a).

		Output Variables		
		TIC-23099	**PIC-23039**	**AIC-23001**
Set-points	**FIC-23027-SP**	0.99	3.00	1.97
	FIC-23028-SP	0.55	3.00	2.98
	FIC-23025-SP	0.01	1.27	1.28

only depends on the output variable, this algorithm ends up being more permissive when we look at the condition number and at the effective rank of the information matrices.

As an example, let us consider the interval in Figure B.5, which was obtained through the AR algorithm. The corresponding condition numbers obtained with the Laguerre structure for this interval can be seen in Table 32. One can notice that the reason why this interval was not considered by the Laguerre algorithm is that only one set-point (FIC-23027-SP) satisfied the condition number criteria, while the imposed condition was that at least two set-points must satisfy the $l_\kappa = 5000$ threshold. However, notice that the FIC-23028-SP produced an information matrix with a condition number of 7846, which is close to the threshold of 5000. The same scenario is true for most of the intervals considered by the AR algorithm and not considered by the Laguerre algorithm. This is actually an expected behavior because each algorithm requires a vast amount of parameters to be chosen, and different choices of parameters and thresholds will definitely produce different outcomes. However, in a general way, both algorithms produced coherent results, with some of the resulting intervals being actually the same.

Table 32: Condition number of the Interval in Figure B.5 obtained with the Laguerre structure.

Set-points		
FIC-23027-SP	**FIC-23028-SP**	**FIC-23025-SP**
875.22	7846.20	113043.16

5.5.2 Statistical Method

In this subsection, the proposed multivariable extension to the statistical algorithm is applied to the petrochemical furnace dataset. Notice that the outline of this algorithm is described in Item 4.1.2.2. Moreover, only the 1 month data block is here considered, once the non-parametric change-point detection algorithm that constitutes the first step of the statistical method has an $O(N^2)$ complexity, taking a long time to run in massive data. The 7 months dataset could be used splitting the data in several batches and running each batch individually. However, the main idea of this subsection is to show how the

algorithm behaves with a real dataset and to compare the resulting intervals with the ones obtained in the previous subsection, for which purpose the 1 month dataset is sufficient.

As described in Item 5.3.2.1, the statistical algorithm works better when the data is over segmented by the change-point detection algorithm. This is because the Lilliefors test can more precisely distinguish noisy intervals when the dataset is shorter. This is the fundamental reason behind the parameter choice made in this subsection, since the statistical method has very few parameters to be chosen, none of them requiring knowledge about the process being studied. Table 33 summarizes the adopted parameters for the statistical multivariable algorithm.

Table 33: Chosen parameters for the multivariable statistical method applied to the petrochemical furnace dataset.

Parameters	**Values**
Pettitt change-point significance level	$\alpha = 0.05$
Pettitt minimum length to split	$N_{s,min} = 200$
Pettitt intervals maximum length	$N_{max} = 80$
Lilliefors test critical value	$\alpha = \frac{1.25}{\sqrt{N_T}}$
Two-mean t-student statistical test significance level	$\alpha = 0.01$
Two-mean t-student statistical test delta	$\Delta = 0.5$
Approval criteria	At least two inputs and two output variables

Notice that the Pettitt minimum length to split $N_{s.min}$ and its maximum length N_{max} are two different parameters. The $N_{s,min}$ parameter determines the minimum length that an interval must have to be further split by the change-point detection algorithm and create a new change-point τ. The maximum length parameter N_{max}, on the other hand, determines the maximum resulting interval sizes and, therefore, it is responsible to over segment the resulting samples following the approach described in Item 3.3.3.1.

Applying the algorithm with the parameters in Table 33 to the petrochemical dataset, 11 final intervals are obtained. It is interesting to mention that two of the resulting intervals are practically identical to the ones obtained with the numerical conditioning

and effective rank algorithms when they are applied to the 1 month data block. These two intervals are shown in Figure B.6 and Figure B.7. Notice that Figure B.6 is very similar to Figure B.1 (c) that corresponds to Interval 3 obtained through the Laguerre filter. In fact, the main difference between these two signals is that the one obtained with the statistical method is a little longer than the one obtained in Figure B.1 (c).

A detailed view of how the algorithm ended up selecting the interval in Figure B.6 can be seen in Figure D.1. It is interesting to notice that the behavior of the algorithm applied to the petrochemical furnace data is completely in accordance with that found through simulations in Subsection 5.4.2. Moreover, it is worth mentioning that if it were required that all three set-points and output variables must meet the criteria of the statistical method, the only resulting interval obtained is the one in Figure B.6. This result is extremely consistent with the author's knowledge of the process, since it is known that this particular interval concerns a system identification experiment that was carried out in the plant and, therefore, it is in fact an interval in which all variables were shaked simultaneously.

We can now evaluate if the interval in Figure B.6 is actually coherent using it to compute the condition number, the chi-squared value, the effective rank and the scalar cross-correlation metric with the same Laguerre and AR models used in the previous subsection. Results can be seen in Tables 34, 35, 36 and 37.

Clearly, this interval produces low values of condition number and high values of effective rank, which points out that it produces a well-conditioned information matrix with both the Laguerre and the AR structures. In fact, if we compare the condition number and the chi-squared values with those for Interval 3 in Tables 24 and 25, the values are very similar. The scalar cross-correlation metric is also coherent with the results, in the sense that the PIC-23039 output variable is the one that is most correlated with all set-points. Notice, however, that, as already pointed out, the cross-correlation values are very restrictive, being difficult to choose an appropriate threshold to it. Finally, notice that the critical value for the chi-squared value is $\chi_{8,0.01} = 20.1$, and, therefore, both PIC-23039 and AIC-23001 would have the causality condition satisfied for at least two set-points.

It is possible to conclude that the multivariable extension proposed in this dissertation to the statistical method is successfully able to obtain meaning intervals from a noisy dataset coming from a real industrial process. Moreover, the results are coherent not only with the ideal scenario produced through simulation, but also with the other

methodologies presented in this work.

Table 34: Chi-squared values computed with the Laguerre structure and applied in Interval in Figure B.6.

		Output Variables		
		TIC-23099	**PIC-23039**	**AIC-23001**
Set-points	**FIC-23027-SP**	12.53	40.1	28.26
	FIC-23028-SP	15.45	25.61	15.15
	FIC-23025-SP	10.43	21.21	23.73

Table 35: Scalar cross-correlation values applied in Interval in Figure B.6.

		Output Variables		
		TIC-23099	**PIC-23039**	**AIC-23001**
Set-points	**FIC-23027-SP**	1.72	5.54	0.53
	FIC-23028-SP	3.69	6.76	2.17
	FIC-23025-SP	0.91	1.23	0.55

Table 36: Effective rank values computed with the AR structure and applied in Interval in Figure B.6.

Output Variables		
TIC-23099	**PIC-23039**	**AIC-23001**
52	39	92

Table 37: Condition number values computed with the Laguerre structure and applied in Interval in Figure B.6.

Set-points		
FIC-23027-SP	**FIC-23028-SP**	**FIC-23025-SP**
738.0	1336.5	355.0

6 CONCLUSIONS

A vast literature review was first provided in this dissertation, with several different techniques being deeply explained and elucidating the state of the art in finding intervals suitable for system identification through historical data of industrial processes. More specifically, a solution to the problem was studied in different aspects, which included both open and closed-loop systems and both SISO and MIMO systems. It was observed that many different ways of tackling the problem are possible and, in fact, different algorithm structures were described, some of them being implemented recursively, other being treated through steps. From this background, the adopted methodology in this dissertation was to address the problem from a step-by-step perspective, in which all the adopted algorithms were divided into three essential blocks: data preprocessing, identification of potential intervals that could be useful for system identification and, finally, evaluation of the resulting potential intervals, choosing those that are indeed sufficiently informative to estimate a model of the system being studied. The main reason behind the choice of this approach was its ability to make visual analyzes that could support the non trivial choice of parameters, and also to provide an understanding on the algorithm behavior. It was concluded, based on the heatmaps developed, that this choice was indeed efficient, facilitating the manipulation of the algorithms in a conscious way, and allowing to structure the problem in the form of a data science framework.

Different ways of finding potential intervals were studied and compared, such as the EWMA, the bandpass and the sliding window filters and also the Pettitt change-point detection algorithm. From the results, it was concluded that all algorithms in fact are effective in splitting the data in regions of interest. The EWMA, bandpass and sliding window filters shown to be also very efficient, in the sense that they require low computational complexity, which therefore turns its execution very fast on large datasets. Moreover, they are also able to isolate regions where the data has moved significantly, selecting only intervals of interest in a predominantly noisy and non informative data. The downside of the EWMA and the bandpass filter was evident due to the necessity of choosing a vast amount of parameters, which are the forgetting factors in the EWMA

filter and the cut-off frequencies in the bandpass filters. Moreover, thresholds must be chosen to select the appropriated intervals, which is definitely not a trivial task. However, through visual analysis and simplifications, these algorithms can be properly manipulated and good results were obtained in both simulated and real data scenarios. Although the sliding window filter has fewer parameters to be selected from, it ends up resulting in shorter intervals. This behavior, however, shown to be useful in applying the proposed algorithms in an incremental way. The lack of parameters can also be considered a downside in the sense that the engineer has less flexibility in manipulating the behavior of the algorithm. Finally, the Pettitt method has shown to be specially useful as a first step of the statistical method. Not only it was able to perfectly split the entire dataset, but it also contains parameters that make tuning very easy, once these parameters are related to statistical test significance levels and therefore do not require any specific knowledge about the industrial process. The only negative side was shown to be the computational inefficiency of the algorithm, which has high computational complexity and, consequently, takes a long time to run as the dataset becomes larger. Moreover, this algorithm works splitting the entire dataset rather than selecting regions of interest. Consequently, more intervals must be further evaluated, which also makes the segmentation process slower.

When dealing with the evaluation of potential intervals, different algorithms were presented, both to SISO and MIMO systems and for the open and closed-loop identification scenarios. It was shown that for open-loop systems, the input variable must be persistently exciting and, therefore, the first steps of the numerical conditioning and effective rank algorithms must be done with the input variable. For closed-loop systems, on the other hand, the first steps of these algorithms must be computed with the set-point, which must be persistently exciting to allow system identification. For this purpose, an algorithm structure was created including both the numerical conditioning and the effective rank approaches in the same framework. The idea was to provide the flexibility to choose which method to apply and allow one to combine aspects of the two algorithms. Moreover, different model structures can be applied, such as the Laguerre filter and the AR model. In a general manner, results obtained through the two approaches have been shown to be coherent through both simulated and real process data. Both the condition number and the effective rank proved to be efficient in identifying intervals with good numerical quality to solve the least squares problem. The main difference between the two methods was their sensitivity. While the condition number goes from 1 to infinity, the effective rank is limited by the model order adopted. A more sensitive metric is better to narrowly discriminate one interval from another. On the counterside, this approach proved to be

harder in choosing an adequate threshold. As it was observed in this dissertation, a single step response can end up reaching condition number values of up to 10^4, in such a way that the engineer ends up in a dilemma of either setting a too low threshold and missing too many intervals, or setting a higher threshold and ending up with a very large number of intervals, which would require a further decision on what interval to use for system identification.

When evaluating the signals correlation, it was concluded that the chi-squared causality test showed to be a more coherent test compared to the scalar cross-correlation metric, in the sense that it is actually able to provide evidences of what the quality of the estimated model will be. In fact, it was concluded from the ideal simulated scenario that the higher the chi-squared computed statistics for a particular interval, the higher the model quality in terms of the FIT index compared to other intervals of lower chi-squared values. This is because this value is computed through an actual estimation of the model parameters. Moreover, the only tuning parameter for this test is the statistical significance level of the chi-squared test, making it easy to be applied. The scalar cross-correlation metric, on the other hand, proved to be more aggressive in terms of selecting the final intervals. This is because it was observed that this metric rarely results in large correlation values, even if the chosen lag range is large. As a consequence, it is not trivial to choose an appropriate threshold to the correlation value, where good intervals may have lower correlation values, but, on the other hand, if this threshold is chosen too short, the algorithm will not be able to filter out intervals that are in fact poorly correlated. Moreover, this metric does not depend on the selected model structure, as occur with the chi-squared test, and, therefore, it is not able to give a previous idea on the model performance.

The choice of the model structure was also shown to be an essential aspect of the numerical conditioning and effective rank algorithms. The Laguerre filter proved to be particularly efficient in the sense that it does not require any knowledge about the process. The only difficulty in using this structure is concerned with the choice of its order and pole. However, this choice proved to be completely feasible through visualizations. The AR structure also showed excellent results and made the choice of parameters even simpler. However, because this structure is only dependent on past versions of the output variable, it does not allow an evaluation of the persistence of excitation, as it is possible through the Laguerre Filter, once it only depends on filtered versions of the input variable. Moreover, although the Laguerre structure only depends on input data, it is adequately complemented by the chi-square test, which ends up including information about the output data at the moment when the model parameters are in fact estimated.

The statistical method proved to be as efficient as the numerical conditioning and effective rank methods. Its great advantage appeared to be in the easiness of choosing the method parameters, which are all related to statistical tests. In fact, with both simulated and real process data, the statistical method was able to produce very similar intervals to the ones obtained through the other two methods, but without the need for any knowledge about the process, nor about model structures. The downside of this algorithm was shown in its dependence on the Pettitt method, which is very computationally inefficient, and in the fact that the engineer has little flexibility in manipulating parameters, in such a way that it is not possible, for instance, to choose to be more or less aggressive about the number of mined intervals.

Finally, the multivariable extrapolation proposed to both the statistical method and to the effective rank method, based on the idea presented in (PATEL, 2016), proved to be successful in the simulated data and in the petrochemical furnace data. The difficulty in the multivariable formulation was to find intervals in which all the variables in the system moved sufficiently and simultaneously to result in a model that contemplates the entire dynamic response of the system. In this dissertation it was observed that, in practice, this scenario is almost never found and, therefore, the formulation presented in (PATEL, 2016) that uses multiple intervals to obtain the final model is very efficient. In fact, through this approach, good model results were obtained both in the ideal simulation scenario and in the petrochemical furnace case.

6.1 Suggestions for Future Works

The main point of attention that could generate major future contributions to the work already done in this dissertation concerns the multivariable problem. For the open-loop identification scenario, the solution presented in this work can successfully result in intervals that, combined, can lead to a satisfactory models of the system. However, based on the petrochemical furnace results, obtaining such models is not a trivial task, in such a way that a more automatic approach could be elaborated to improve this solution. The multivariable closed-loop identification scenario, on the other hand, is a complex problem that has been very little addressed in the literature so far, with only one work referenced in this dissertation. This is actually a crucial problem to the extent that one may be interested not in obtaining a model with the system set-point, as was done in this dissertation and that could be used to design Model Predictive Controlers, but in obtaining a model of the process itself, *i.e.*, a model obtained with the output and the

manipulated variables. Such a model is extremely useful, for example, for tuning existing PID controllers in the process. The difficulty in this approach is the fact that in a closed-loop multivariable system the manipulated variables interfere with each other and also depend on the system output, in such a way that the necessary conditions for such a system to produce enough information to system identification must be studied.

A final aspect of future contributions is regarding the hypothesis of linearity adopted in this dissertation. All the described algorithms in this work assume that the process can be treated linearly. In fact, for the numerical conditioning and the effective rank algorithms, linear structures constitute an essential part of the methodology. Therefore, one can guarantee that the resulting intervals are suitable for the estimation of linear models of the process being studied, but this is not guaranteed if one is actually interested in obtaining non-linear models.

REFERENCES

AGGARWAL, C. C. *Data Mining: The Textbook*. 1. ed. New York, United States of Americal: Springer, 2015.

AGUIRRE, L. A. *Introdução à Identificação de Sistemas: técnicas lineares e não lineares: teoria e aplicação*. 4. ed. Belo Horizonte, Brasil: Editora UFMG, 2015.

AMIRTHALINGAM, R.; SUNG, S. W.; LEE, J. H. Two-step procedure for data-based modeling for inferential control applications. *AIChE Journal*, v. 46, n. 10, p. 1974–1988, 2000.

APPLEBAUM, D. *Probability and Information: An Integrated Approach*. 2. ed. New York, United States of America: Cambridge University Press, 2008.

ARENGAS, D.; KROLL, A. A Search Method for Selecting Informative Data in Predominantly Stationary Historical Records for Multivariable System Identification. In: *Proceedings of the 21st International Conference on System Theory, Control and Computing (ICSTCC)*. Sinaia, Romenia: IEEE, 2017a. p. 100–105.

ARENGAS, D.; KROLL, A. Searching for informative intervals in predominantly stationary data records to support system identification. In: *Proceedings of the XXVI International Conference on Information, Communication and Automation Technologies (ICAT)*. Sarajevo, Bosnia-Herzegovina: IEEE, 2017b. p. 132–137.

BITTENCOURT, A. C. *et al.* An algorithm for finding process identification intervals from normal operating data. *Processes*, v. 3, p. 357–383, 2015.

BODENHAM, D. A. *Adaptive Estimation with Change Detection for Streaming Data*. Thesis — Imperial College, University of London, London, 2014.

BUCKLEY, P. S.; LUYBEN, W. L.; SHUNTA, J. P. *Design of Distillation Column Control Systems*. New York, United States of America: Creative Services Inc, 1985.

CARRETTE, P. *et al.* Discarding data may help in system identification. *IEEE Transactions on Signal Processing*, IEEE, v. 44, n. 9, p. 2300–2310, 1996.

CHAVES, C. R. *Implementação e comparação de performance de controladores preditivos multivariáveis MMPC, estruturados com modelo de perturbação de entrada ajustados às perturbações através da utilização do modelo matemático identificado de um forno industrial petroquímico*. Thesis — Polytechnic School of the University of São Paulo, São Paulo, Brasil, 2020.

CHAVES, C. R.; JULIANI, R.; GARCIA, C. Identification of The Dynamic Model of a Petrochemical Furnace of 50MW for Implementation of MPC Control. In: *Proceedings of the 12th Symposium on Dynamics and Control of Process Systems, including Biosystems (DYCOPS)*. Florianópolis, Brazil: IFAC, 2019. p. 317–322.

DEGROOT, M. H.; SCHERVISH, M. J. *Probability and Statistics*. 4. ed. Boston, United States of America: Pearson Education, Inc., 2016.

DERRICK, T. R.; THOMAS, J. M. Time series analysis: The cross-correlation function. In: *Innovative Analyses of Human Movement*. Champaign, Illinois: Kinesiology Publications, 2004. cap. 7, p. 189–205.

DEVORE, J. L. *Probability and Statistics for Engineering and the Sciences*. 9. ed. Boston, United States of America: Cengage Learning, 2016.

FACELI, K. *et al. Inteligência Artificial: Uma Abordagem de Aprendizado de Máquina*. Rio de Janeiro, Brasil: LTC, 2017.

FASSOIS, S. D.; RIVERA, D. Applications of system identification. *IEEE Control Systems Magazine*, v. 27, n. 5, p. 24–26, 2007.

FINCH, T. *Incremental calculation of weighted mean and variance*. Lecture Notes — University of Cambridge, Cambridge, UK, 2009. Available from: https://fanf2.user.srcf.net/hermes/doc/antiforgery/stats.pdf.

GARCIA, C. *Controle de Processos Industriais*. 1. ed. São Paulo, Brasil: Blucher, 2017.

GEVERS, M.; BAZANELLA, A.; MISKOVIC, L. Informative data: how to get just sufficiently rich? In: *Proceeding of 47th IEEE Conference on Decision and Control*. Cancun, Mexico: IEEE, 2008. p. 1962–1967.

GRILLENZONI, C. Testing for causality in real time. *Journal of Econometrics*, v. 73, n. 2, p. 355–376, 1996.

HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2. ed. New York, United States of America: Springer, 2009.

JULIANI, R. C. G. *Plantwide control: a review and proposal of an augmented hierarchical plantwide control design technique*. Thesis — Polytechnic School of the University of São Paulo, São Paulo, Brasil, 2017.

KILLICK, R.; FEARNHEAD, P.; ECKLEY, I. A. Optimal detection of changepoints with a linear computational cost. *Journal of the American Statistical Association*, v. 107, n. 500, p. 1590–1598, 2012.

LILLIEFORS, H. W. On the kolmogorov-smirnov test for the exponential distribution with mean unknown. *Journal of the American Statistical Association*, v. 64, n. 325, p. 387–389, 1969.

LJUNG, L. *System Identification: Theory for User*. 2. ed. Upper Saddle River, New Jersey: Prentice-Hall, Inc., 1999.

MARTENS, J. New insights and perspectives on the natural gradient method. *Journal of Machine Learning Research*, v. 21, n. 146, p. 1–76, 2020.

MONTGOMERY, D. C. *Introduction to Statistical Quality Control*. 6. ed. Hoboken, NJ: Wiley, 2008.

OLIVEIRA JR., J. G.; GARCIA, C. Algorithm-aided Identification Using Historic Process Data. In: *Proceeding of the XIII Simpósio Brasileiro de Automação Inteligente (SBAI)*. Porto Alegre, Brazil: [s.n.], 2017.

PATEL, A. *Data Mining of Process Data in Mutlivariable Systems*. 606–610 p. Degree project in electrical engineering — Royal Institute of Technology, Stockholm, Sweden, 2016.

PERETZKI, D. *et al.* Data mining of historic data for process identification. *In: Proceedings of the 2011 AIChE Annual Meeting*, p. 1027–1033, 2011.

PETTITT, A. N. A. A non-parametric approach to the change-point problem. *Journal of the Royal Statistical Society*, v. 28, n. 2, p. 126–135, 1979.

QIN, S. J.; CHIANG, L. H. Advances and opportunities in machine learning for process data analytics. *Computers & Chemical Engineering*, v. 126, p. 465–473, 2019.

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

ROY, O.; VETTERLI, M. The effective rank: A measure of effective dimensionality. In: *Proceedings of the15th European Signal Processing Conference*. Poznan, Poland: IEEE, 2007.

SHARDT, Y. A. W.; BROOKS, K. Automated system identification in mineral processing industries: A case study using the zinc flotation cell. *IFAC-PapersOnLine*, v. 51, n. 18, p. 132–137, 2018.

SHARDT, Y. A. W.; HUANG, B. Conditions for Identifiability Using Routine Operating Data for a First-Order ARX Process Regulated by a Lead-Lag Controller. In: *Proceeding of 9th International Symposium on Dynamics and Control of Process Systems (DYCOPS)*. Leuven, Belgium: IFAC, 2010. p. 373–378.

SHARDT, Y. A. W.; HUANG, B. Closed-loop identification condition for armax models using routine operating data. *Automatica*, v. 47, n. 7, p. 1534–1537, 2011.

SHARDT, Y. A. W.; HUANG, B. Closed-loop identification using routine operating data: the effect of time delay. *IFAC Proceedings Volumes*, v. 44, n. 1, p. 1646–1651, 2011.

SHARDT, Y. A. W.; HUANG, B. Closed-loop identification with routine operating data: Effect of time delay and sampling time. *Journal of Process Control*, v. 21, n. 7, p. 997–1010, 2011.

SHARDT, Y. A. W.; HUANG, B. Data quality assessment of routine operating data for process identification. *Computers & Chemical Engineering*, v. 55, p. 19–27, 2013.

SHARDT, Y. A. W.; HUANG, B. Statistical properties of signal entropy for use in detecting changes in time series data. *Journal of Chemometrics*, v. 27, n. 11, p. 394–405, 2013.

SHARDT, Y. A. W.; HUANG, B. Parameter-based conditions for closed-loop system identifiability of arx models with routine operating data. *Journal of the Franklin Institute*, v. 354, n. 2, p. 722–751, 2017.

SHARDT, Y. A. W.; SHAH, S. L. Segmentation Methods for Model Identification from Historical Process Data. In: *Proceedings of the 19th World Congress*. Cape Town, South Africa: IFAC, 2014. p. 2836–2841.

SMITH, S. W. *Digital Signal Processing*. San Diego, California: California Technical Publishing, 1999.

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

TIWARI, S.; WEE, H. M.; DARYANTO, Y. Big data analytics in supply chain management between 2010 and 2016: Insights to industries. *Computers & Industrial Engineering*, v. 115, p. 319–330, 2018.

TREFETHEN, L. N.; BAU, D. I. *Numerical Linear Algebra*. Philadelphia, United States of America: Society for Industrial and Applied Mathematics (SIAM), 1997.

VERHAEGEN, M.; VERDULT, V. *Filtering and System Identification: A Least Square Approach*. Cambridge, UK: Cambridge University Press, 2007.

VIRTANEN, P. *et al.* SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, v. 17, p. 261–272, 2020.

WAHLBERG, B. System identification using laguerre models. *IEEE Transactions on Automatic Control*, IEEE, v. 36, n. 5, p. 551–562, 1991.

WANG, J. *et al.* Searching historical data segments for process identification in feedback control loops. *Computers and Chemical Engineering*, v. 112, n. 6, p. 6–16, 2018.

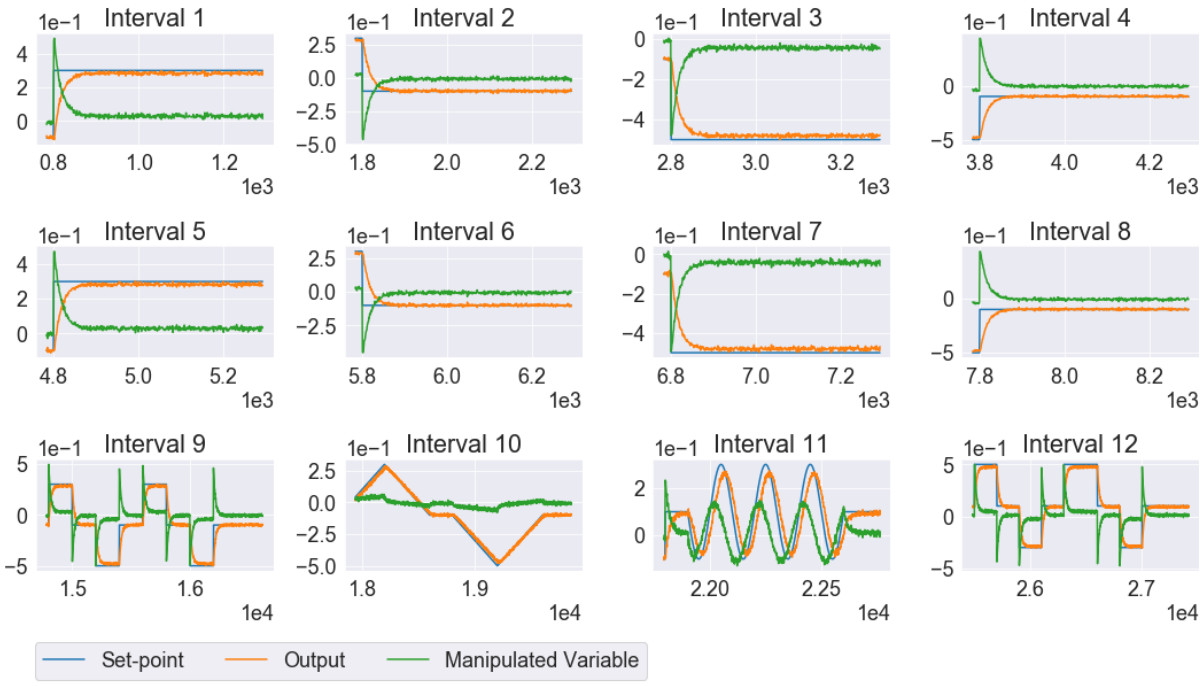
WANG, L.; WANG, G. Big data in cyber-physical systems, digital manufacturing and industry 4.0. *International Journal of Engineering and Manufacturing*, v. 6, n. 4, p. 1–8, 2016.

WOOD, R. K.; BERRY, M. W. Terminal composition control of a binary distillation column. *Chemical Engineering Science*, v. 28, n. 9, p. 1707–1717, 1973.

YU, Z. *et al.* Performance assessment of pid control loops subject to setpoint changes. *Journal of Process Control*, v. 21, n. 8, p. 1164–1171, 2011.

APPENDIX A – SISO NUMERICAL CONDITIONING: POTENTIAL INTERVALS DETAILS

Figure A.1: Potential Intervals obtained with the exponential weighted filter and parameters $\lambda_S = \lambda_\mu = 0.006$, $n_{idx} = 20$ and $l_S = 0.003$.



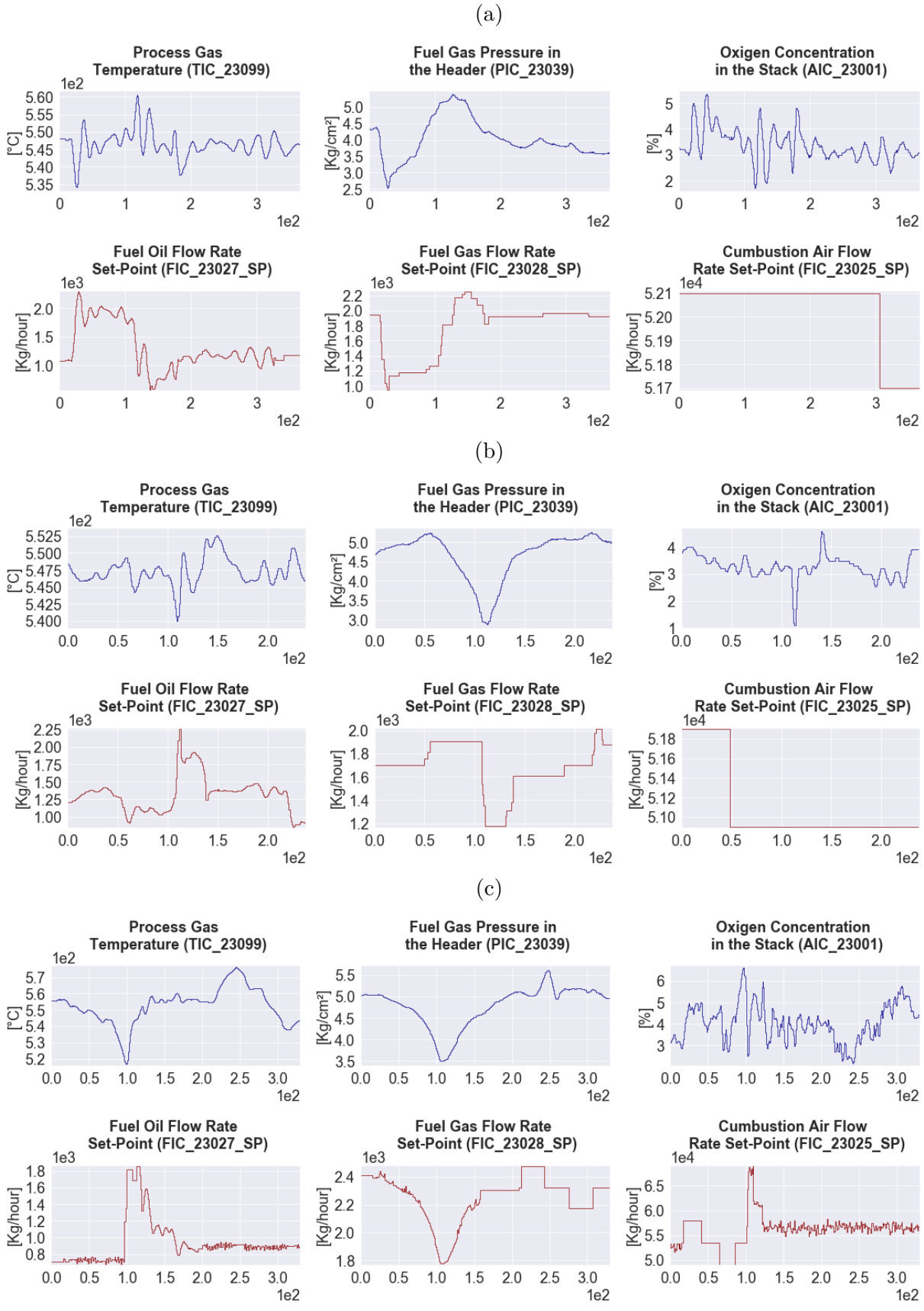
Source: Author's own development.

APPENDIX B – PETROCHEMICAL FURNACE: MINED INTERVALS

This appendix highlights some of the resulting mined intervals obtained with the numerical conditioning, effective rank and statistical algorithms applied to the petrochemical furnace dataset. Notice that some of the signals here displayed are used as estimation and validation sets for system identification.

B.1 Numerical Conditioning

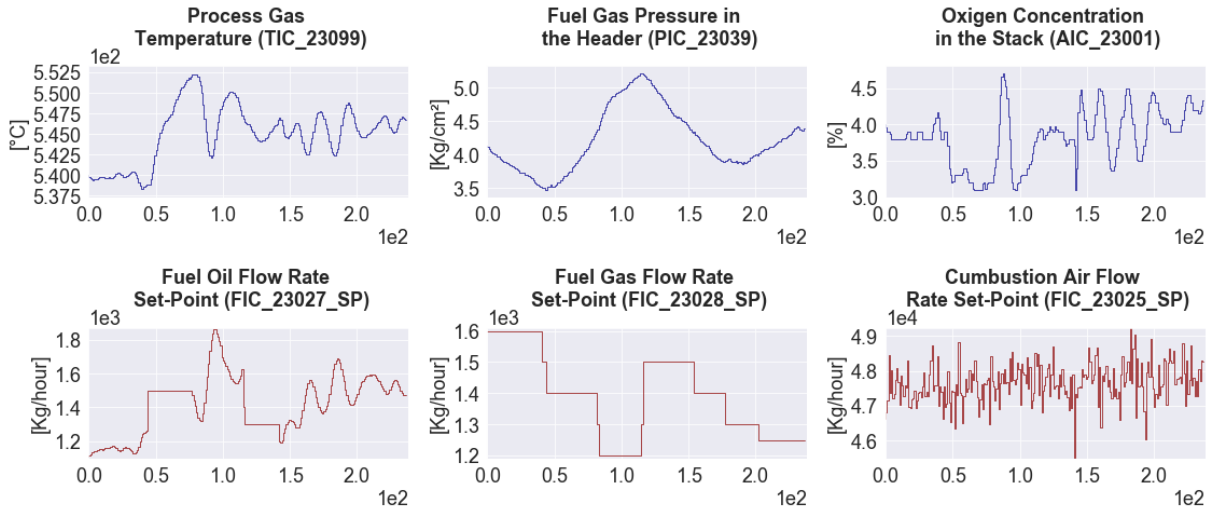
Figure B.1: Petrochemical furnace estimation intervals. (a) Interval 1 (b) Interval 2 (c) Interval 3.



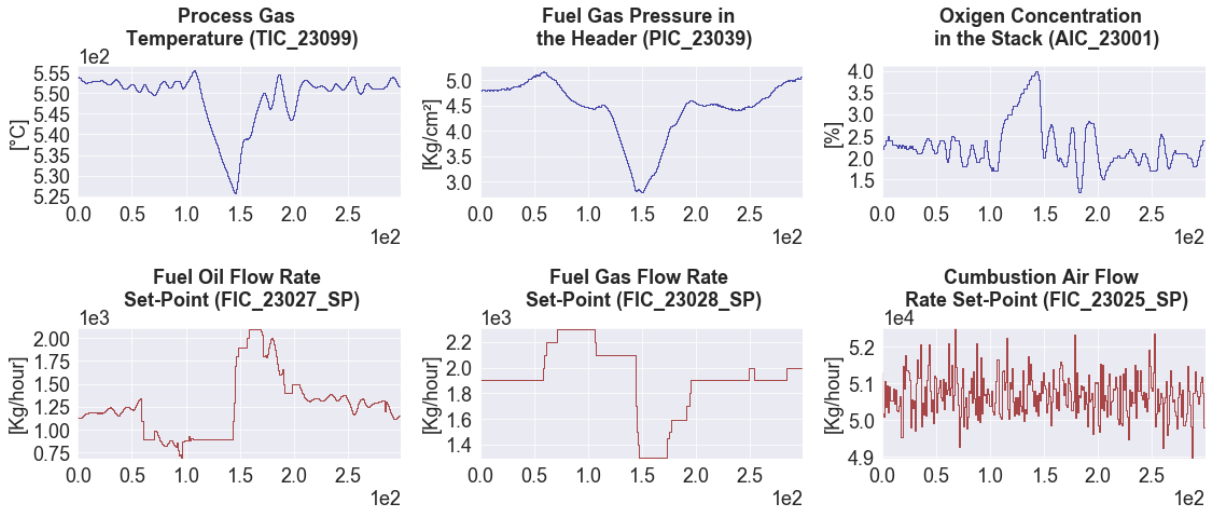
Source: Author's own development.

Figure B.2: Petrochemical furnace validation intervals. (a) Interval 4 (b) Interval 5.

(a)

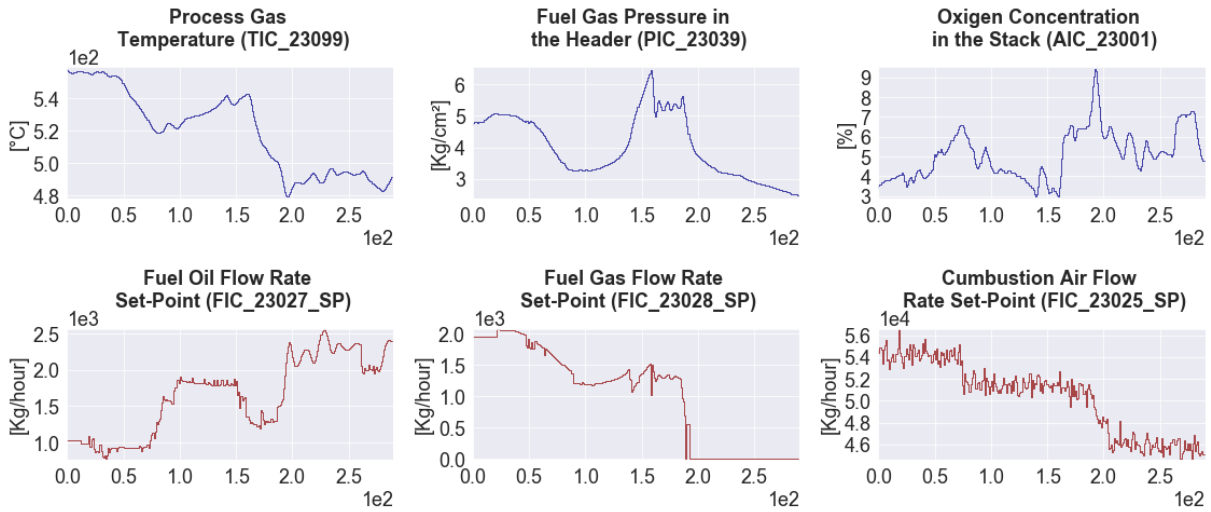


(b)



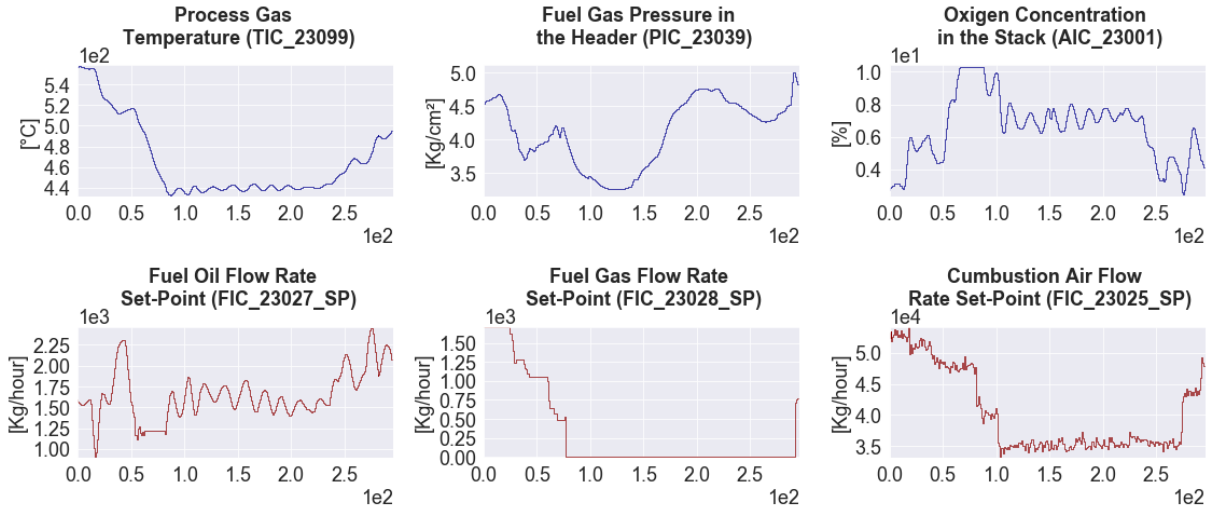
Source: Author's own development.

Figure B.3: Example of resulting interval in which the FIC-23028-SP variable was set to zero.



Source: Author's own development.

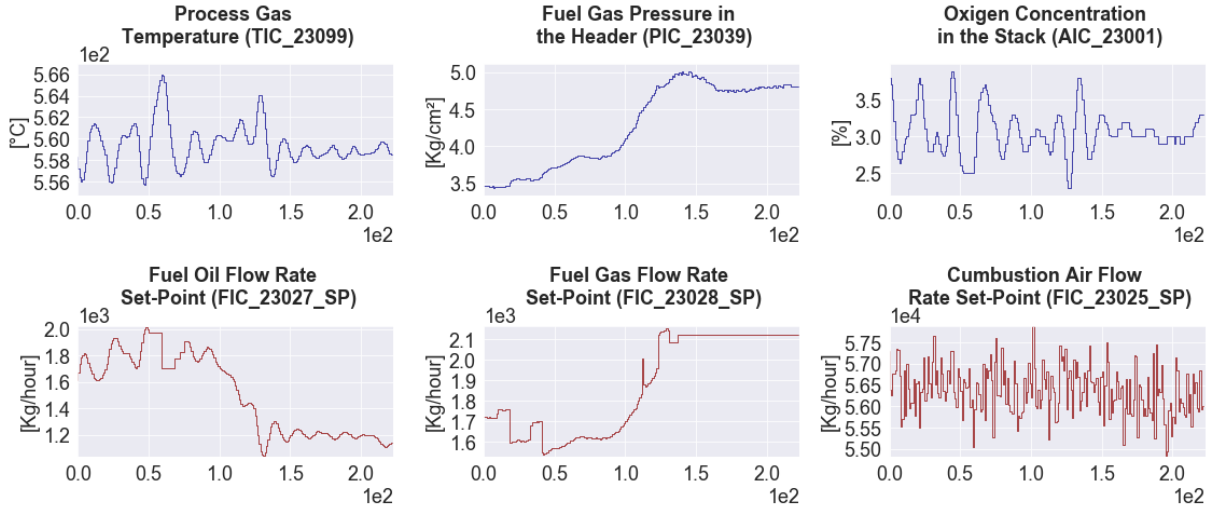
Figure B.4: Validation data for the AIC-23001 model.



Source: Author's own development.

B.2 Effective Rank

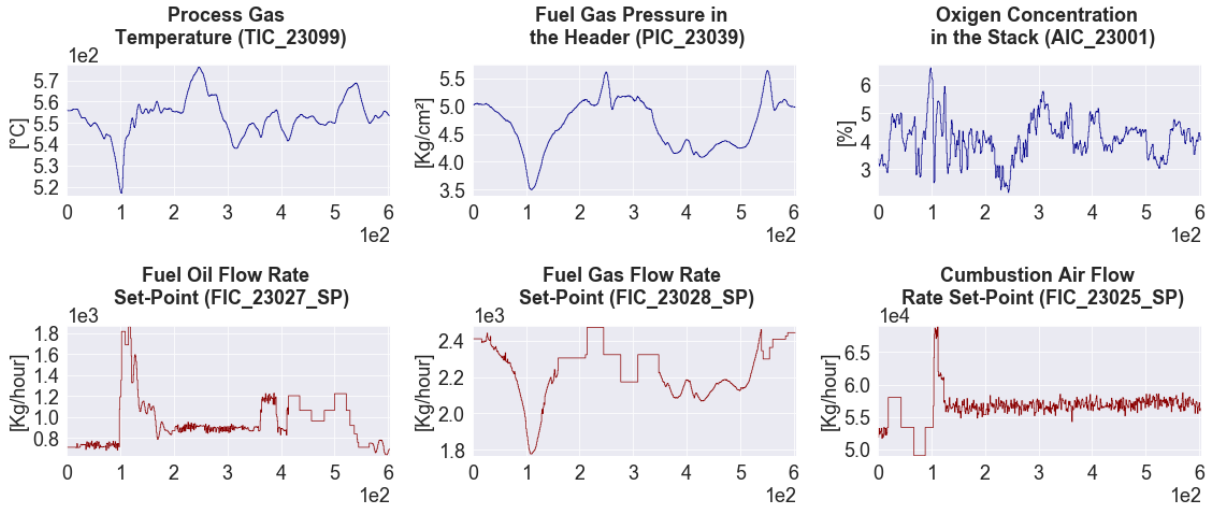
Figure B.5: Example of interval obtained with the algorithm in Figure 8 considering an AR structure and the effective rank and the scalar cross-correlation criteria.



Source: Author's own development.

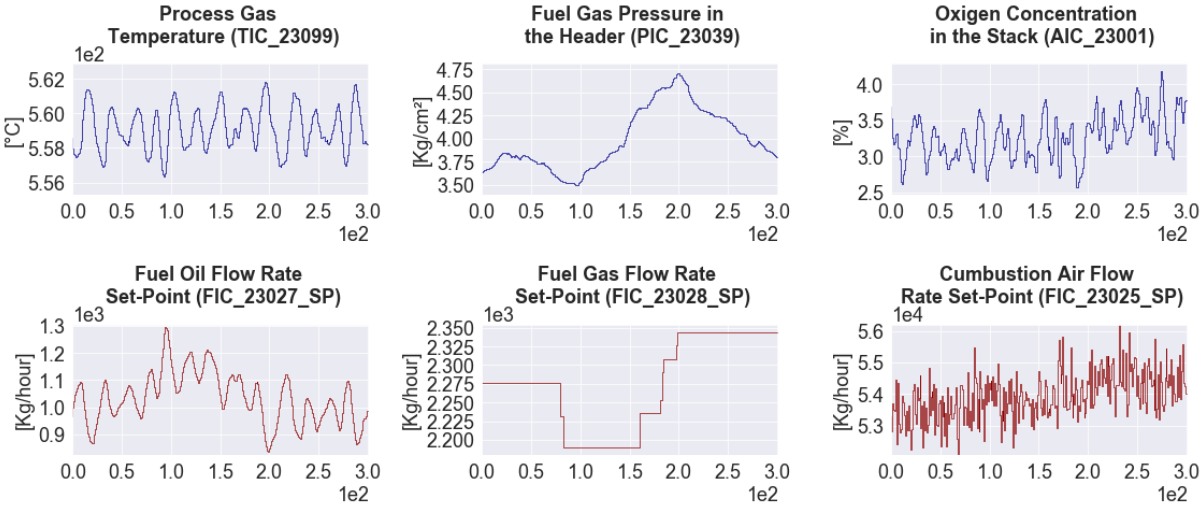
B.3 Statistical Method

Figure B.6: Example of mined interval obtained with the multivariable statistical algorithm applied to the petrochemical furnace dataset.



Source: Author's own development.

Figure B.7: Example of mined interval obtained with the multivariable statistical algorithm applied to the petrochemical furnace dataset.

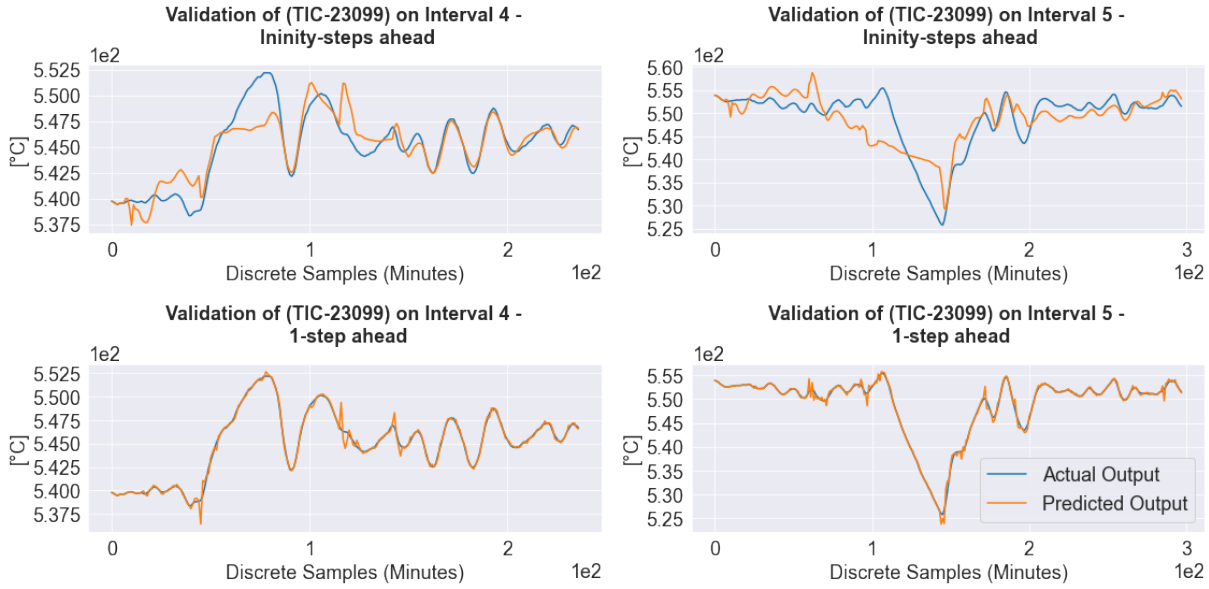


Source: Author's own development.

APPENDIX C – PETROCHEMICAL FURNACE: SYSTEM IDENTIFICATION DETAILS

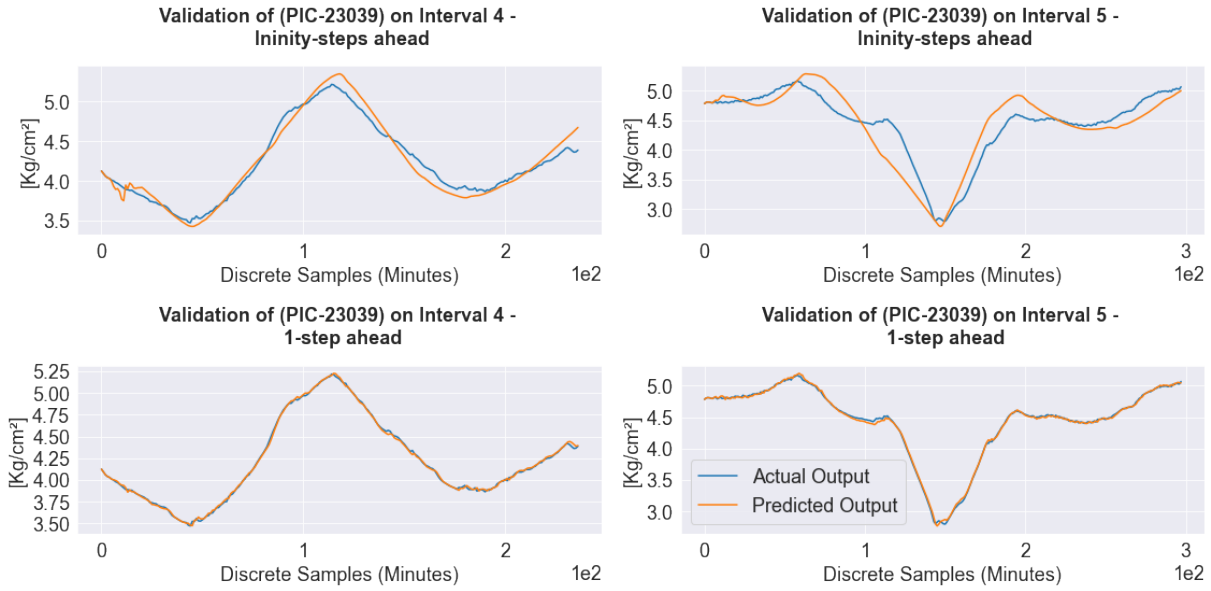
This appendix contains results of systems identifications carried out with the mined data obtained for the petrochemical furnace. Comparisons between the predictions and the actual validation data are provided for both 1 step and infinite step ahead predictions.

Figure C.1: Comparison of actual and predicted outputs for TIC-23099 model in 1 and infinity step-ahead scenarios.



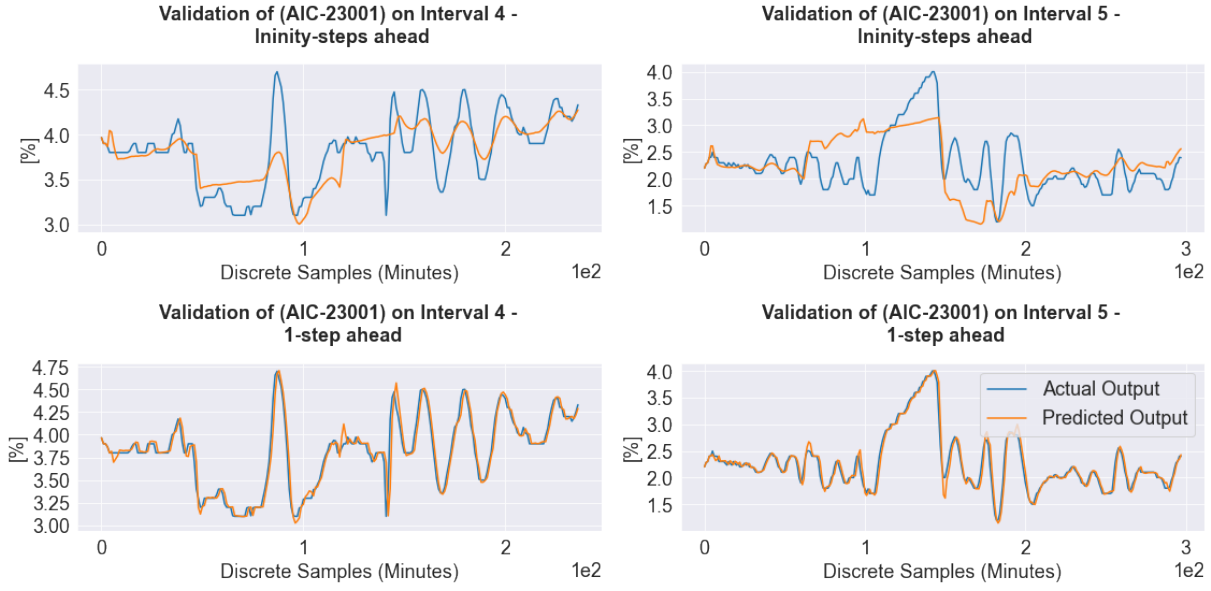
Source: Author's own development.

Figure C.2: Comparison of actual and predicted outputs for PIC-23039 model in 1 and infinity step-ahead scenarios.



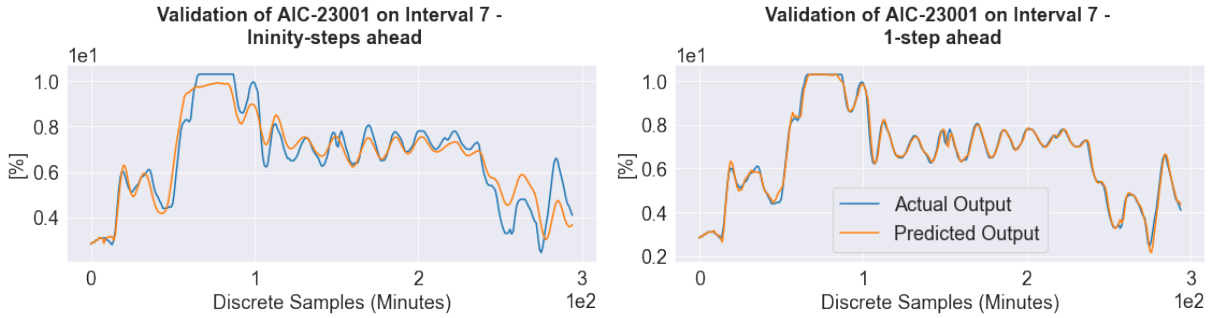
Source: Author's own development.

Figure C.3: Comparison of actual and predicted outputs for AIC-23001 model in 1 and infinity step-ahead scenarios.



Source: Author's own development.

Figure C.4: Comparison of actual and predicted outputs for AIC-23001 model in 1 and infinity step-ahead scenarios.

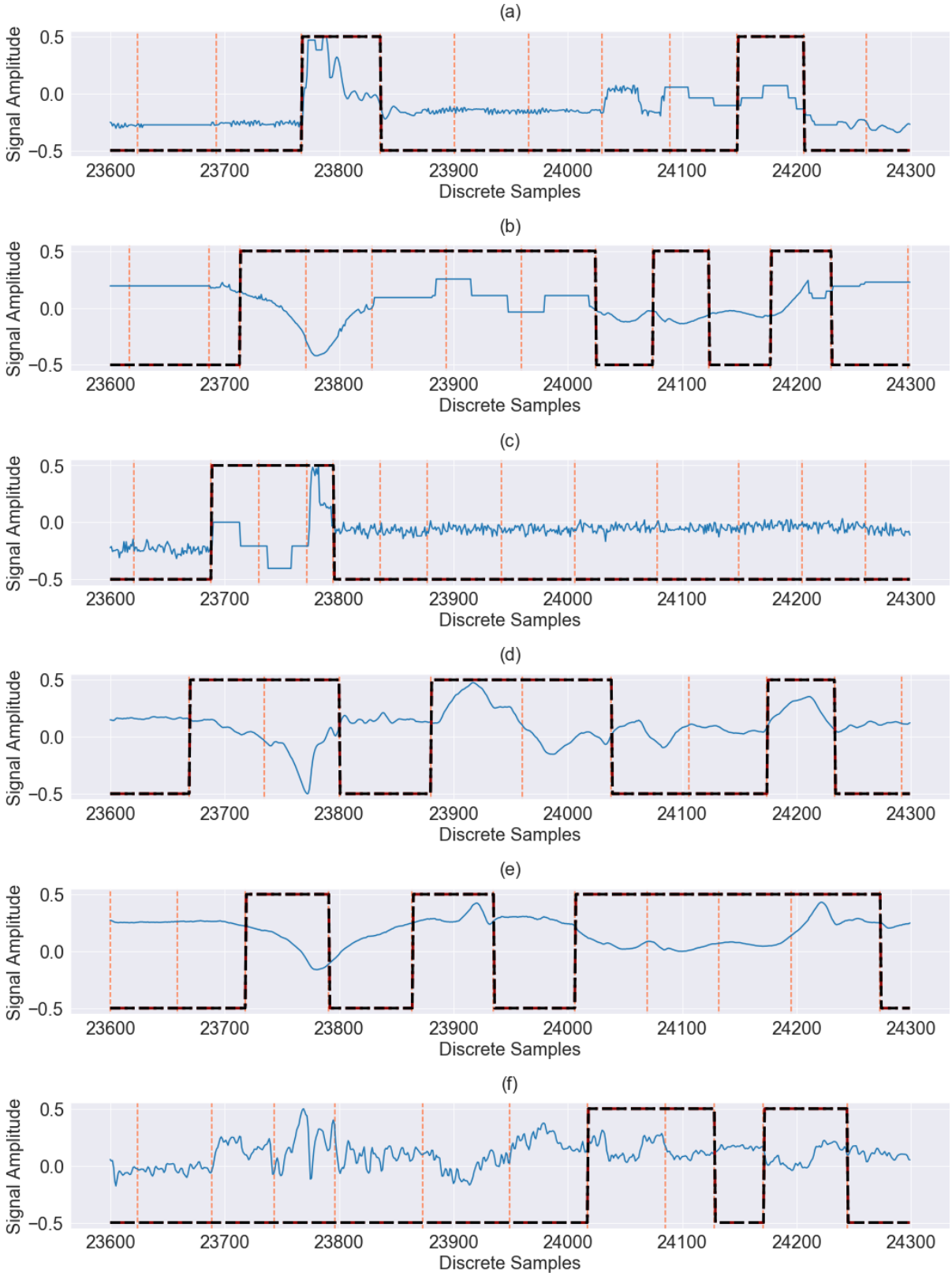


Source: Author's own development.

APPENDIX D – PETROCHEMICAL FURNACE: STATISTICAL METHOD DETAILS

This appendix provides details of the steps taken by the multivariable statistical method when choosing one of the obtained intervals.

Figure D.1: Steps of the statistical algorithm. **Step 1**: change-point detection algorithm for a significance level $\alpha = 0.05$ (orange vertical dashed lines); **Step 2**: magnitude change statistical test for a Lilliefors critical value of $1.25/\sqrt{N_T}$ (black dashed line); **Step 3**: two-mean t-student comparison test for a significance level $\alpha = 0.01$ and a difference in mean delta of $\Delta = 0.5$ (red dashed line). (a) FIC-23027-SP. (b) FIC-23028-SP. (c) FIC-23025-SP. (d) TIC-23099. (e) PIC-23039. (f) AIC-23001.



Source: Author's own development.