UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

KAUÃ MINHO ANTUNEZ

DESIGN OF NONLINEAR CONTROLLERS THROUGH THE VIRTUAL REFERENCE METHOD AND REGULARIZATION

Porto Alegre
2021

KAUÃ MINHO ANTUNEZ

DESIGN OF NONLINEAR CONTROLLERS THROUGH THE VIRTUAL REFERENCE METHOD AND REGULARIZATION

Thesis presented to Programa de Pós-Graduação em Engenharia Elétrica of Universidade Federal do Rio Grande do Sul in partial fulfillment of the requirements for the degree of Master in Electrical Engineering.
Area: Control and Automation

ADVISOR: Prof. Dr. Alexandre Sanfelice Bazanella

Porto Alegre
2021

KAUÃ MINHO ANTUNEZ

DESIGN OF NONLINEAR CONTROLLERS THROUGH THE VIRTUAL REFERENCE METHOD AND REGULARIZATION

This thesis was considered adequate for obtaining the degree of Master in Electrical Engineering and approved in its final form by the Advisor and the Examination Committee.

Advisor: _____
Prof. Dr. Alexandre Sanfelice Bazanella, UFRGS
Doctor by the Federal University of Santa Catarina – Florianópolis, Brazil

Examination Committee:

Prof. Dr. Diego Eckhard, UFRGS
Doctor by the Federal University of Rio Grande do Sul – Porto Alegre, Brazil

Prof. Dr. Gustavo Henrique da Costa Oliveira, UFRGS
Doctor by the State University of Campinas – Campinas, Brazil

Profª. Drª. Lucíola Campestrini, UFRGS
Doctor by the Federal University of Rio Grande do Sul – Porto Alegre, Brazil

Coordinator of PPGEE: _____
Prof. Dr. Sérgio Luís Haffner

Porto Alegre, November 2021.

DEDICATION

I dedicate this work to my parents Moacyr and Fernanda, and to my siblings Amanda and Juan.

ACKNOWLEDGMENTS

Behind every successful master's student stand people who made the journey through this path possible. This work was only doable by their support, encouragement and advice. I would like to express my thanks to the people who have stood besides me along my masters.

First, I thank God for my life and my family for all the support and care during this journey.

Second, I thank the friends that I made at the *Baia 1* of PPGEE: Chrystian Remes, Guilherme Salati, Rodrigo Binotto, Rômolo Lasch and Leonardo Cabral, for all the support and rich discussions.

I would also like to thank Professor Alexandre Bazanella, for providing me the opportunity of working as a researcher and introducing me into the Data-Driven framework.

I thank CAPES for providing me a master's scholarship.

ABSTRACT

This work proposes a new extension for the nonlinear formulation of the data-driven control method known as the Nonlinear Virtual Reference Feedback Tuning. When the process to be controlled contains a significant quantity of noise, the standard Nonlinear VRFT approach – that uses the Least Squares method – yield estimates with poor statistical properties. These properties may lead the control system to undesirable closed loop performances and even instability. With the intention to improve these statistical properties and controller sparsity and hence, the system's closed loop performance, this work proposes the use of ℓ_1 regularization on the nonlinear formulation of the VRFT method. Regularization is a component that has been extensively employed and researched in the Machine Learning and System Identification communities lately. Furthermore, this technique is appropriate to reduce the variance in the estimates. A detailed analysis of the noise effect on the estimate is made for the Nonlinear VRFT method. Finally, three different regularization methods, the third one proposed in this work, are compared to the standard Nonlinear VRFT.

Keywords: Data-Driven Control, VRFT, Regularization.

RESUMO

Este trabalho propõe uma nova extensão para a formulação não linear do método de controle orientado por dados conhecido como Método da Referência Virtual Não Linear, ou *Nonlinear Virtual Reference Feedback Tuning* – denominado aqui somente como VRFT. Quando o processo a ser controlado contém uma quantidade significativa de ruído, a abordagem padrão do VRFT – que usa o método dos Mínimos Quadrados – fornece estimativas com propriedades estatísticas pobres. Essas propriedades podem levar o sistema de controle a desempenhos indesejáveis em malha fechada. Com a intenção de melhorar essas propriedades estatística, identificar um controlador simples em quantidade de parâmetros e melhorar o desempenho em malha fechada do sistema, este trabalho propõe o uso da regularização ℓ_1 na formulação não linear do método VRFT. A regularização é uma técnica que tem sido amplamente empregada e pesquisada nas comunidades de Aprendizagem de Máquina e Identificação de Sistemas ultimamente. Além disso, esta técnica é apropriada para reduzir a variância das estimativas. Uma análise detalhada do efeito do ruído na estimativa é feita para o método VRFT não linear. Finalmente, três diferentes métodos de regularização, o terceiro proposto neste trabalho, são comparados com o VRFT.

Palavras-chave: Controle baseado em dados, VRFT, Regularização.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

CbT	Correlation-based Tuning
CV	Cross-Validation
DD	Data-Driven
D^2-IBC	Data-Driven Inversion Based Control
FDT	Frequency Domain Feedback Tuning
IFT	Iterative Feedback Tuning
IV	Instrumental Variable
LASSO	Least Absolute Shrinkage Selection Operator
LOOCV	Leave-one Out Cross-Validation
LQG	Linear Quadratic Gaussian
LQR	Linear Quadratic Regulator
LS	Least Squares
MSE	Mean Square Error
NMP	Non-minimum Phase
OCI	Optimal Controller Identification
PI	Proportional-Integral
PID	Proportional-Integral-Derivative
SISO	Single-Input Single-Output
STLS	Sequential Threshold Least Squares
$STLS_2$	Sequential Threshold Least Squares 2
MIMO	Multiple-Input Multiple-Output
MSE	Mean Square Error
VRFT	Virtual Reference Feedback Tuning

LIST OF SYMBOLS

$\bar{C}(q)$	Controller's linear portion
$\mathcal{C}$	Nonlinear controller
$F(q)$	Transfer function F of a discrete system
J	Cost function, Performance criterion
λ	Penalty parameter
$\hat{x}$	Estimate of a x vector
$\phi(.)$	Library of nonlinear functions
Φ	Regressor vector
q	Forward-shift time operator, such that $qx(t) = x(t+1)$
ρ	Controller's parameter vector
$\mathcal{P}$	Nonlinear process
$\mathbb{R}$	Set of real numbers
$T_d(q)$	Desired transfer function
θ	Model parameter vector
$\epsilon(t)$	Noise signal
$u(t)$	Input signal, training data
$y(t)$	Output signal
$r(t)$	Reference signal

CONTENTS

1 INTRODUCTION

The classical control methods for linear systems known as *modern control* techniques were developed based on the state space theory. To cite a few of these techniques we have the Linear Quadratic Regulator (LQR), the Linear Quadratic Gaussian (LQG), the pole placement and the Robust Control techniques. All of these methodologies share the same feature: they are in the class of *Model Based Control* (MBC) techniques. The mentioned techniques present a drawback: the user needs to spend some time determining or identifying a model for the process to be controlled.

With the development of microelectronics, the microcontrollers, which used to be overly expensive, became cheaper. In this way, the use of more precise control systems became more frequent in the industries, allowing the process's data to be stored for further analysis and used on control techniques based on data (HOU; WANG, 2013).

Data-Driven (DD) control design requires no formulation of the mathematical model of the process to tune the controller's parameters. On the contrary, a fixed controller structure is directly designed to optimize some performance criterion given by the Reference Model. That is, only the input and output data of the process is used to achieve the specified dynamics.

The DD matter has been addressed since the early 1940s in (ZIEGLER; NICHOLS, 1942), in which the controller identification is made through process' time response. Adjusting the parameters of a controller without any knowledge of the transfer function has been tackled within the adaptive control community (ÅSTRÖM; WITTENMARK, 2013; GOODWIN; SIN, 1984; IOANNOU; SUN, 2012). The adaptive control domain has been and still is very explored, yet the ordinary industries have not incorporated its methodologies.

In the midst of the DD methods, the most popular is known as the *Virtual Reference Feedback Tuning* (VRFT) (CAMPI; LECCHINI; SAVARESI, 2002). This methodology is part of the one-shot methods group, i.e. a single batch of input-output data is needed to tune the controller. Several extensions for the VRFT have been researched and presented in the literature: (LECCHINI; CAMPI; SAVARESI, 2002) with a 2-degree of freedom approach, (CAMPESTRINI *et al.*, 2011) to deal with Non-minimum Phase (NMP) zeros

of the plant, (CAMPESTRINI *et al.*, 2016) which shows the MIMO case, and (CAMPI; SAVARESI, 2006) for the nonlinear scenario. In addition, there are the iterative methods group, where the most known and the pioneer is the Iterative Feedback Tuning (IFT) (HJALMARSSON *et al.*, 1998). The requirement of iterative methods is the sequence of experiments to improve the controller's parameters.

As a result of VRFT being a one-shot design and the controller possessing linear parameterization, the optimization problem can be solved via the Least Squares (LS) method. However, the fact that these methods display poor statistical properties is a well-known fact, so that alternative approaches for the optimization are still being sought (GARCIA; BAZANELLA, 2020).

As the estimates present poor statistical properties, the system's closed-loop performance is directly affected. Hence, the method becomes less attractive to be employed in real industrial applications where high noise levels are present. Also, it is a well-known fact that the majority of the systems show a nonlinearity, though this aspect is often ignored in the control design.

Inside the DD community, there are fundamental studies about the design of nonlinear controllers aiming on a closed-loop system with a linear behavior. However, these works involve a considerable knowledge about the process and the applied signal for achieving satisfactory estimates (CAMPI; SAVARESI, 2006). Furthermore, in (BAZANELLA; NEUHAUS, 2014), a new class of controllers is estimated through the VRFT method: the rational and polynomial structures. This rational structure is able to represent several real systems, as it uses the previous input and output signals, although the algorithm used to estimate a rational nonlinear system is a sequence of Least Squares.

Since 90% of industrial control loops around the world are Proportional-Integral-Derivative (PID) (ÅSTRÖM; HÄGGLUND, 2006), we extend this famous controller structure to a nonlinear scenario. The proportional, integral and derivative signals are assembled with a library of nonlinear functions composing a *nonlinear controller*.

In the DD framework, often there is very little - or no - prior information on the process available. Under these circumstances, an overparameterized controller structure is required. Therefore, a new design tool is essential to guarantee the best statistical properties possible. In the linear monovariable - Single-Input Single-Output (SISO) - context, the addition of the ℓ_2-regularization has been discussed in (RALLO *et al.*, 2016; FORMENTIN; KARIMI, 2014), where authors include the regularization to reduce the estimates covariance using instrumental variables and enhance the system's performance. Likewise, the work (BOEIRA, 2018) presents a *Bayesian* perspective for the multivariable VRFT method.

With these ideas in mind, this work's main idea is to exploit the ℓ_1-regularization to get better estimates on overparameterized controllers and yielding sparse ones. The contributions of this master thesis are: comparing the already proposed ℓ_1-regularization

methods in the literature with the one developed herein, analyze the noise effect and the input signal on the estimates' quality.

The thesis is partitioned as follows: Chapter 2 explains some basic definitions and characteristics about the systems analyzed herein, along with a literature review about the most known Data-Driven control techniques. Chapter 3 brings out the concepts about the VRFT in the LTI and SISO context. Chapter 4 deals with the VRFT method applied to nonlinear systems employing the ℓ_1-regularization. Finally, Chapter 6 presents the concluding remarks.

2 PRELIMINARIES

This chapter presents some basic and fundamental definitions giving the reader a better work's understanding. In the first place, the process to be controlled is defined, as well as some characteristics and fundamental properties of the signals are addressed. Subsequently, the control system is described and analyzed, likewise the different type of structures and the diverse performance criteria.

2.1 The process

In this work's ambit nonlinear SISO discrete systems are considered, which can be described by the following equation

$$y(t) = \mathcal{P}(y(t-1),...,y(t-n_y), u(t-1),..., u(t-n_u)) + \nu(t), \tag{1}$$

where $t \in \mathbb{Z}$ represents the discrete time variable, and q is the forward-shift time operator, i.e.

$$qx(t) = x(t+1) \tag{2}$$

$$q^{-1}x(t) = x(t-1), \tag{3}$$

$y(t) \in \mathbb{R}$ is the output signal, $u(t) \in \mathbb{R}$ is the input signal, $\mathcal{P}(.)$ is the nonlinear map and $\epsilon(t) \in \mathbb{R}$ is a white noise signal with zero mean and variance $E[\epsilon^2(t)] = \sigma_\epsilon^2$.

The $E[.]$ stands for the Expectation operator. An important observation to disclose is that the does not describe all possible disturbances that may appear in the process, however it is sufficient for a wide variety of practical problems (LJUNG, 1999).

Definition 2.1. *(LJUNG, 1999) A quasi-stationary process $s(t)$ can be defined as:*

(i) $\bar{E}\,[s(t)] = m_s(t), \;\; |m_s| \leq C, \; \forall t;$

(ii) $\bar{E}\,[s(t)s(r)] =\mid R_s(t,r) \mid\leq C, \; \forall t,r;$

(iii) $\lim\limits_{N\to\infty} \frac{1}{N} \sum\limits_{t=1}^{N} R_s(t, t-\tau) = R_s(\tau), \; \forall \tau,$

where $m_s(t)$ is the mean value of $s(t)$ and $R_s(t,r)$ is the autocorrelation of $s(t)$.

Acknowledge that, if the signal $s(t)$ is a stationary stochastic process, it is easy to verify that the above conditions are met. On the other hand, if $s(t)$ is purely deterministic, then the expectation operator does not have any effect. Thereby, saying that a signal is quasi-stationary is similar to say that it is bounded and that its limit in condition *(iii)* exists. Periodical deterministic signals satisfy above conditions and thus are quasi-stationary (LJUNG, 1999).

To make easier the notation along the work, the operator $\bar{E}[.]$ is applied and it is defined as

$$\bar{E}[f(t)] \triangleq \lim_{N\to\infty} \frac{1}{N} \sum_{t=1}^{N} E\left[f(t)\right], \tag{4}$$

that allows writing the autocorrelation in the following manner:

$$R_s(\tau) = \bar{E}[s(t)s(t-\tau)] \tag{5}$$

Two signals $s(t)$ and $x(t)$ are said to be jointly quasi-stationary if both meet definition 2.1 and if their cross-correlation

$$R_{sx}(\tau) = \bar{E}[s(t)x(t-\tau)] \tag{6}$$

exists. If $R_{sx}(\tau) = 0$, then the signals are *decorrelated* (LJUNG, 1999).

Another concept exploited on this thesis is the *power spectrum* of a quasi-stationary signal $s(t)$, denoted by $\Phi_s(\omega)$ and written as:

$$\Phi_s(\omega) = \sum_{\tau=-\infty}^{\infty} R_s(\tau)e^{-j\omega\tau}. \tag{7}$$

This mathematical tool is useful in the analysis of signals on the frequency domain. Also, it has the interesting property of Parseval's Theorem, that can be verified through the inverse Fourier transform:

$$\bar{E}[s^2(t)] = R_s(0) = \frac{1}{2\pi}\int_{-\pi}^{\pi} \Phi_s(\omega)d\omega. \tag{8}$$

2.2 The control system

For the purpose of attaining a determined performance criterion by the process, like reference tracking, disturbance rejection or some other criterion, the process' input signal is transformed by a controller, then the system is put in closed-loop for operation. In the literature and on various control applications, the signal $u(t)$ is determined by countless ways. For instance, it can be a nonlinear complex function, calculated by a predictive algorithm (which is used on *Model Predictive Control* techniques), or some optimization

algorithm that minimizes some criterion at each time instant, as in the optimal control systems case.

In this work, particularly, nonlinear controllers are studied. The process' input signal is represented as it follows:

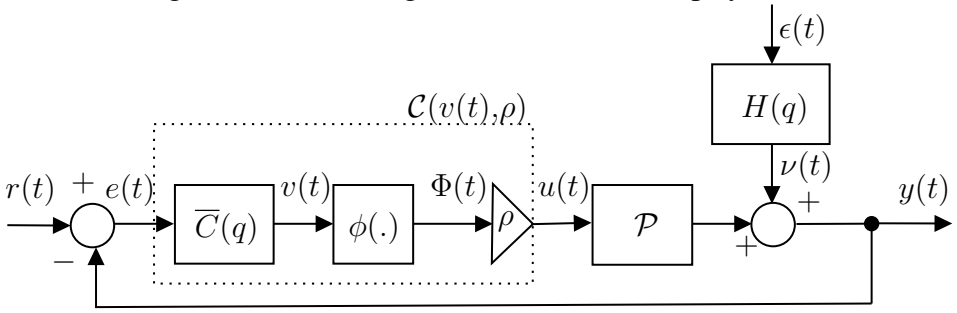
$$u(t) = \mathcal{C}(v(t),\rho) = \mathcal{C}(\bar{C}(q)e(t),\rho) \tag{9}$$

where $\mathcal{C}(v(t),\rho)$ is the nonlinear controller, parameterized by the vector $\rho \in \mathbb{R}^p$. To better illustrate the control system, Figure 1 shows the block diagram of the closed-loop system. It is possible to observe that the controller is divided in two parts: $\bar{C}(q)$ which is the controller's linear block and $\phi(.)$ which is a library of nonlinear functions described as:

$$\phi(v(t)) = \begin{bmatrix} | & | & | & | & | \\ \phi_1(v(t)) & \phi_2(v(t)) & \phi_3(v(t)) & ... & \phi_k(v(t-M)) \\ | & | & | & | & | \end{bmatrix}. \tag{10}$$

The structure for the library of nonlinear functions (10) is used to obtain a controller that is linearly parameterized.

Figure 1 – Block diagram of the closed-loop system.



Source: author.

The reference signal $r(t) \in \mathbb{R}$ is assumed to be decorrelated from the noise process, i.e.

$$R_{r\nu}(\tau) = \bar{E}[r(t)\nu(t-\tau)] = 0, \forall\tau, \tag{11}$$

the error signal $e(t) = r(t) - y(t)$ is filtered by the transfer function generating the signals $v(t)$, that are expanded by the library $\phi(v(t))$. Then, the regressor vector $\Phi(t)$ is generated by each column of the library of nonlinear functions.

2.2.1 Controller structure

Before presenting the controller tuning techniques, it is convenient to define some aspects about the structures used herein. For such, the controller $\mathcal{C}(v(t),\rho)$ has the following scheme:

$$\mathcal{C}(v(t),\rho) = \rho^T\Phi(v(t)), \tag{12}$$

where $\rho = \begin{bmatrix} \rho_1 & \rho_2 & \dots & \rho_n \end{bmatrix}^T$, i.e. $\rho \in \mathbb{R}^n$. The control techniques explored in this work consider a fixed structure controller, chosen *a priori* by the user, and its parameters are the variables to be found.

With the objective to ease the synthesis problem, we split the nonlinear controller into two portions. The linear portion $\bar{C}(q)$ is a vector structured by rational functions in q. Besides, these rational functions of $\bar{C}(q)$ must be linearly independent under the real numbers set, i.e., one assumes that the parameterization is minimal. One example of rational functions are the *Laguerre* functions as presented in (BAZANELLA; CAMPESTRINI; ECKHARD, 2012). It is important to highlight that this portion accountable for the steady-state error requirements.

The set of implementable controllers is known as the controller classe ($\mathbb{C}$), given by

$$\mathbb{C} = \{\mathcal{C}(v(t),\rho) : \rho \in \Omega \subseteq \mathbb{R}^p\}, \tag{13}$$

where Ω is the admissible parameters set (BAZANELLA; CAMPESTRINI; ECKHARD, 2012).

The portion responsible for turning the controller into a nonlinear one is the library $\phi(.)$. This portion's contribution is to approximate the nonlinear function inverse, thus mitigating its effect on the process. For instance, if the chosen linear controller structure is proportional-integral (PI), the expansion of the linear signals $v_p = e(t)$ and $v_i(t) = e(t)\left(\frac{q}{q-1}\right)$ through the Taylor Series until some order n defined by the user:

$$\phi(.) = \begin{bmatrix} v_p & v_i & v_p v_i & v_p^2 & v_i^2 & \dots & (v_p v_i)^n \end{bmatrix}. \tag{14}$$

Another point of view to be highlighted is that the choice of $\mathbb{C}$ class implies an assignment of the controller's order. Normally the user chooses the controller denominator as of the reference to be followed and/or disturbances to be rejected, employing the *Internal Model Principle*. For instance, if the application is to reject or follow a step, then the denominator of $\bar{C}(q)$ must have the $(q-1)$ element, as it was presented above on the PID controller. Under other conditions, if the application of the control system is to follow or reject a sinusoidal signal, then the controller's denominator has to have complex conjugate poles with the same natural frequency of the signal at issue. Controllers that have this feature are known as resonant controllers and are largely applied to Uninterruptible Power Supplies (CORLETA *et al.*, 2016), for example.

2.2.2 Performance Criteria

With the purpose of evaluating the closed-loop system performance there are various criteria that may be used. The most direct and sophisticated are the ones composed by some norm of a signal on the control system, since they offer a qualitative measure (BAZANELLA; CAMPESTRINI; ECKHARD, 2012). The most commonly used norm

for this purpose is the 2-norm, which can be written as

$$\|x(t)\| = \frac{1}{N}\sum_{t=1}^{N}[x(t)]^2. \tag{15}$$

When this norm is used, the performance criterion is called the H_2 criterion. Later on, we explain different H_2 performance criteria and their objectives.

Primarily, consider the reference tracking problem, where the noise and perturbation effects are taken into account. The output signal is desired to have the same behavior from the reference as possible. However, demanding that the output behave identically to the reference is an utopia, because the perfect tracking is practically impossible (BAZANELLA; CAMPESTRINI; ECKHARD, 2012). Therefore, the reference objective can be expressed though a transfer function that displays the desired closed-loop behavior. This transfer function is denominated *Reference Model* and it is symbolized by $T_d(q)$. The reference model tracking criterion is symbolized by $J_y(\rho)$, it can be written as the following way:

$$J_y(\rho) = \bar{E}[(y(t,\rho) - y_d(t))]^2, \tag{16}$$

$$y_d(t) = T_d(q)r(t). \tag{17}$$

Underlining what has been said about the reference tracking criterion, equation (16) quantifies the difference between the desired and obtained ($y(t,\rho)$) output signals with the $\mathcal{C}(v(t),\rho)$ controller. The lower this value, the best the system represents the desired output.

There are various methods for controller design that seek to minimize the criterion in (16), that includes the one present here in this work. On the next section it will be discussed the general idea of said methods.

2.3 Model Reference Control Problem

The Model Reference Control Problem pursues to minimize the criterion on (16), i.e, to minimize the difference between the desired and the obtained output behavior with the controller $\mathcal{C}(v(t),\rho)$. Thus, the parameters can be obtained by solving the following optimization problem:

$$\hat{\rho} = \arg\min_{\rho} J_y(\rho). \tag{18}$$

Beware that this is a non-convex optimization problem, so it might have several local minimums, complicating the solution.

The controller that makes the system achieve the desired closed-loop performance is called the *ideal controller*, or $\mathcal{C}_d(v(t),\rho)$. When this controller is inserted on the control loop, the objective function $J_y(\rho)$ shows its minimum value. In other words, the ideal

controller is such that the input-output map $r(t) \rightarrow y(t)$ of the closed-loop system is exactly the one specified by the reference model $T_d(q)$.

An important fact is that the $\mathbb{C}$ class of controllers defined by the user comprises the ideal controller, so it is said that the *ideal controller belongs to the* $\mathbb{C}$ class. In this scenario, it is said that it exists an ideal parameters vector that is written as $\mathcal{C}_d(v(t), \rho) = \mathcal{C}(v(t), \rho_0)$. The assumption below summarizes these statements.

Assumption 2.1. *$\mathcal{C}_d(q) \in \mathbb{C}$ then $\exists \rho_0 : \mathcal{C}(v(t), \rho_0) \equiv \mathcal{C}_d(v(t))$.*

At first, it seems to be impossible to satisfy the mentioned assumption because in the nonlinear context, we only approximate the inverse of the nonlinear function. So, only if nonlinear function inverse Taylor series expansion has a high order we can attain Assumption 2.1.

2.4 Data-Driven Control

The data-driven control techniques differ from the classical model-based control techniques because they do not need a mathematical model of the process. Notice that, for instance, if the process was known, it would be easy to find the ideal controller. For example, consider the following plant taken from (CAMPI; SAVARESI, 2006):

$$y(t) = y(t-1) + u(t-1)^3 + \nu(t), \tag{19}$$

the controller class

$$u(t) = \rho[r(t) - y(t)]^{\frac{1}{3}}, \tag{20}$$

and also the reference model:

$$y_d(t) = r(t-1). \tag{21}$$

If the control law is $u(t) = [r(t) - y(t)]^{\frac{1}{3}}$ and $\nu(t) \equiv 0$, then the closed-loop becomes

$$\begin{aligned} y(t) &= y(t-1) + \{[r(t-1) - y(t-1)]^{\frac{1}{3}}\}^3 \\ &= y(t-1) + [r(t-1) - y(t-1)] \\ &= r(t-1) \end{aligned} \tag{22}$$

as desired.

The DD methods do not seek to determine any controller structure. Contrarily, they search to tune the controller's parameters that are in a fixed structure predefined by the user, from an input-output data set Z^N collected from the process:

$$Z^N = [y(1),u(1),y(1),u(2),...,y(N),u(N)], \tag{23}$$

with N being the quantity of data. Once the data is collected the optimization problem can be solved. There is a wide literature about this subject and the different data-driven

control techniques. The most known are briefly presented in the following, in order to better contextualize this work.

Among the data-driven control techniques, there are two distinct groups: the iterative method group, which employ a sequence of experiments to update the controller's parameters in an iterative way; and the direct method group, which need only one or two batch of data for the tuning. Usually, on the iterative algorithms, one may apply the steepest-descent algorithm to minimize the chosen criterion. Thus, after each iteration, the controller's parameters are calculated by

$$\rho_{i+1} = \rho_i - \gamma_i \nabla J(\rho_i), \tag{24}$$

with $\nabla J(\rho) = \frac{\partial J(\rho)}{(\partial \rho)}$ and $\gamma_i > 0$. This algorithm is the most known favorite in the literature because it needs less information compared to the others: only the objective's function gradient is needed and the step size of the iteration (normally assigned by the user). On the other hand, if the Newton's algorithm is applied on the optimization, it is necessary to estimate the Hessian matrix, which implies on realizing some more complex experiments, which have more information (BAZANELLA; CAMPESTRINI; ECKHARD, 2012).

The biggest advantage of the iterative methods is the operation security, since at each iteration a small change occurs on the parameters, keeping the process well behaved, as in the previous iteration. On the other hand, the main disadvantage of these procedures is the large amount of experiments that must be realized on the process until the algorithm's convergence. Another downside is the fact that these algorithms depend on the initialization of the parameters and they can converge to the local minimums of the cost function. Among the iterative methods, the most known are the *Iterative Feedback Tuning* (IFT), the *Frequency Domain Feedback Tuning* (FDT) and the *Correlation-based Tuning* (CbT).

The IFT is one of the most known techniques developed in the data-driven control community. It was initially proposed for the SISO case in (HJALMARSSON; GUNNARSSON; GEVERS, 1994) and it was more explored in (HJALMARSSON *et al.*, 1998). An IFT variation for cascade control systems can be found in (TESCH, 2016). Extensions for the MIMO case are presented in (BRUYNE, 1997; HJALMARSSON; BIRKELAND, 1998), with a more detailed and adequate version in (HJALMARSSON, 1999). The IFT and its variations were already applied on different situations, as an example we have a heat treatment system (EL-AWADY; HANSSON; WAHLBERG, 1999), a metal cutting machine (GRAHAM; YOUNG; XIE, 2007) and even in the control of quadcopters (TESCH; ECKHARD; GUARIENTI, 2016). The main idea of the IFT method is to minimize both the model reference tracking and the noise rejection. The optimization is made through the steepest descent algorithm (the gradient is estimated though the data collected from the closed-loop experiments).

FDT was introduced by (KAMMER; BITMEAD; BARTLETT, 2000). It seeks to minimize the noise variance, without considering the reference tracking problem. In this

method the gradient and the Hessian matrix estimators of the objective function are proposed, based on a frequency domain analysis. By calculating the Hessian matrix, one may apply the Newton's method to optimize the problem.

The iterative CbT technique was originally proposed for SISO systems in (KARIMI; MISKOVIC; BONVIN, 2004) and extended for MIMO systems in (MISKOVIC *et al.*, 2007). An application of the method for a magnetic suspension system can be found in (KARIMI; MIŠKOVIĆ; BONVIN, 2003). The CbT has an objective that differs slightly from the presented methods, this method tries to minimize the cross correlation between the reference signal and the error between the desired output and real system's output. By that, the technique is able to make the closed-loop system as similar as possible to the reference model dynamics and to do so, it makes use of the steepest descent algorithm.

On the opposite side, the direct data-driven control methods do not use the features like the gradient vector and the Hessian matrix of the criteria regarding the parameters. Therefore, their biggest advantage is that few experiments are needed to tune the controller. However, the disadvantage that can be observed from this kind of method is the abrupt change on the system's parameters, which in some cases, can yield an undesired behavior. Amidst the direct methods the most known are the non-iterative CbT (KARIMI; VAN HEUSDEN; BONVIN, 2007), the *Virtual Reference Feedback Tuning* (VRFT) (CAMPI; LECCHINI; SAVARESI, 2002) and the *Optimal Controller Identification* (OCI) (CAMPESTRINI *et al.*, 2016).

The main target of this master thesis, the VRFT is the most spread and researched among the direct methods, having various extensions and variants about its properties. The VRFT method was firstly developed for SISO linear systems in (CAMPI; LECCHINI; SAVARESI, 2002), with the main core of rewriting $J_y(\rho)$, rendering the task of optimizing an objective function simpler and from another point of view. For instance, one can cite the active suspension control in (CAMPI; LECCHINI; SAVARESI, 2003), the neuroprosthesis control in (PREVIDI *et al.*, 2004), as many others in the literature. In respect to the extensions, the work (VAN HEUSDEN; KARIMI; BONVIN, 2011) develops the VRFT formulation with restrictions that guarantee a closed-loop stability. In (CAMPESTRINI *et al.*, 2011) the SISO VRFT was expanded to deal with non-minimum phase processes. In most recent works (PILLONETTO *et al.*, 2014; BOEIRA, 2018) there is a concern to improve the statistical properties for the VRFT framework with the aid of the *Bayesian regularization*.

Regarding the nonlinear systems control, the first extension was presented in (CAMPI; SAVARESI, 2006), where they developed all formulation needed for applying the VRFT, but it is not explained how the controller can be constructed. In recent works, another interesting approach have been studied: the *Data-Driven Inversion-Based Control* (D^2-IBC) (NOVARA; FORMENTIN, 2017). This approach is built on a two-degree of freedom arrangement with a linear and nonlinear controller operating in parallel. The linear

controller can be a PI or a PID, for example, depending on the reference model chosen, while the nonlinear controller is composed by a large number of basis functions. The nonlinear controller parameters are penalized by the ℓ_1 norm. The complexity of the D^2-IBC method is that it requires two algorithms for the controller tuning, in exchange it has a closed-loop stability guarantee. The D^2-IBC method has already been applied on the MIMO context for the control of an autonomous vehicle (NOVARA; MILANESE, 2019).

More recently, the work presented in (FERIZBEGOVIC *et al.*, 2021) portrays a methodology for the data-driven control of Hammerstein Systems using the Bayesian perspective. This research employs a Wiener controller with a linear and nonlinear part. Also, they propose three different structures for the nonlinear part: polynomial, piecewise and only the linear portion. In (SINGH; SZNAIER, 2021) it is analyzed the identification of Nonlinear Autoregressive with eXogenous (NARX) Models exploring two aspects: the sparsity in regressors and basis functions.

The main focus of this master thesis consists on analyzing the statistical properties of the standard nonlinear VRFT and then add the ℓ_1-regularization and the *sparse regression* and reanalyze their effects on the statistical properties and in the closed-loop performance.

2.5 Chapter Conclusions

This chapter introduced the basic definitions that will be used during this work. The nonlinear process that will be controlled has been presented, as well as the signals that are studied and their characteristics. In this sense, the concept of quasi-stationary signal was introduced, which allows the analysis of stationary and deterministic stochastic signals in the same context. Also, the spectrum of quasi-stationary signals was defined, as their properties.

Section 2.2 described the type of control system that will be analyzed and synthesized here, where the controller is divided in two parts: the linear and the nonlinear, has a fixed structure predefined by the user and is linearly parameterized. It was commented on the choice for the controller structure. The performance criteria based on the 2-norm standards of the signals present in the system have also been explained in this section.

In Section 2.3 the control problem was presented by reference model, which proposes the minimization of the $J_y(\rho)$ criterion, as well as the ideal controller concept and when Assumption 2.1 can be considered valid. Given the importance of choosing $T_d(q)$, some guidelines for its choice were commented, aiming to ensure fundamental characteristics for the ideal controller and, consequently, a good formulation for the problem.

Finally, Section 2.4 contextualized the state of the art in the field of data-driven control, where the main idea of these methodologies was first discussed. It consists in tuning controllers without a mathematical model of the process, using a set of collected Z^N data and optimization of some performance criterion. Subsequently, the methods were sepa-

rated into two distinct groups: the iterative and the direct ones and their main advantages and disadvantages were pointed out. Also the most relevant methods have been described, as well as some of their extensions and practical applications. The focus was given on the features and extensions of VRFT, since this is the main subject of this thesis.

3 NONLINEAR VIRTUAL REFERENCE FEEDBACK TUNING

According to the previous chapter, the VRFT is a direct data-driven control method that seeks to tune the controller parameters without the modeling or identification need, as it uses only the data set Z^N collected from the process. The data are obtained with only one experiment, with no need of an iterative procedure for the tuning. The tuned controllers are nonlinear and their structure is defined by the user. Via the VRFT method we seek to minimize the model reference tracking H_2 criterion, previously defined as $J_y(\rho)$. Nevertheless, minimizing the $J_y(\rho)$ objective function is an exhausting task, since it is a nonconvex function, i.e., it has local minimums. Accordingly, the methodology proposed by the VRFT method lies on doing the optimization of a different criterion: $J^{VR}(\rho)$. Under ideal conditions, this criterion possesses the same minimum of $J_y(\rho)$. On the non ideal case, in order to make a proper tuning there is a need to add a filter to approximate both criterion minimum.

This chapter characterizes the nonlinear formulation of the Virtual Reference Feedback Tuning method developed in the literature. It is presented all the points and assumption that concern this method. Additionally, the case with noise in the process is addressed in sequence, since it is concentration of this thesis. In this scenario, the designer will obtain an *estimate* of the ideal parameters $\mathcal{C}_d(q)$ in tuning the controller's parameters.

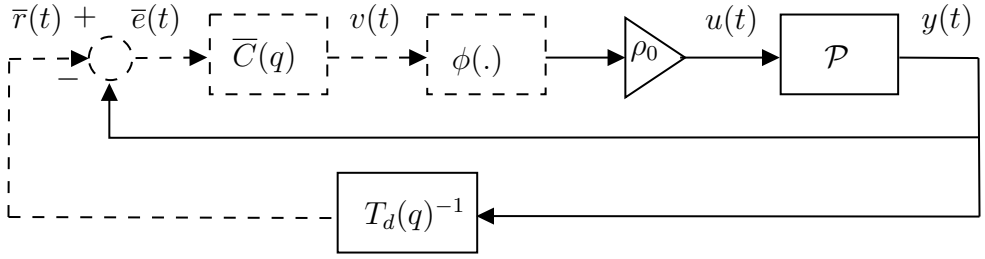
3.1 The method

The proposed method idea consists of generating a *virtual reference* from the measured process' data.

First, consider a nonlinear process with $\epsilon(t) = 0$ and that it was collected some open-loop input and output data, $u(t)$ and $y(t)$, respectively. Now, assume that the loop was *virtually* put in closed-loop with the ideal controller, as Figure 2 shows. In the figure, the continuous lines depict the actual collected data and the dashed lines depict the virtual portion of the experiment. Suppose that the output $y(t)$ was collected by the virtual experiment. If this is the case, then it means tha the *virtual reference* $\bar{r}(t)$ was applied so

that

Figure 2 – VRFT virtual loop diagram.



Source: author.

$$T_d(q)\bar{r}(t) = y(t) \tag{25}$$

$$\bar{r}(t) = T_d^{-1}(q)y(t), \tag{26}$$

with $T_d(q)$ being the reference model chosen by the user. In the virtual system, we can also determine the *virtual error* $\bar{e}(t)$:

$$\bar{e}(t) = \bar{r}(t) - y(t) \tag{27}$$

$$\bar{e}(t) = [T_d^{-1}(q) - 1]y(t). \tag{28}$$

In possession of this information, the user has the ideal controller input (the $\bar{e}(t)$ signal) and its output ($u(t)$ signal). With that said, one can synthesize an identification problem of $\mathcal{C}_d(v(t))$. Therefore, the objective is to find the parameters set that minimizes the following criterion:

$$\hat{\rho} = \arg\min_{\rho} J^{VR}(\rho), \tag{29}$$

$$\begin{aligned} J^{VR}(\rho) &\triangleq \bar{E}[u(t) - \hat{u}(t,\rho)]^2 \\ J^{VR}(\rho) &\triangleq \bar{E}[u(t) - \mathcal{C}(v(t),\rho)]^2 \end{aligned} \tag{30}$$

with $\hat{u}(t,\rho) = \mathcal{C}(v(t),\rho)$.

One can use this method to estimate any sort of nonlinear controller $\mathcal{C}(v(t),\rho)$, but in this work the attention is given to controllers linear in the parameters. Since this option decreases the problem's complexity. In this manner, the criterion in (30) is rewritten as

$$\begin{aligned} J^{VR}(\rho) &\triangleq \bar{E}[u(t) - \rho^T\Phi(\bar{C}(q)\bar{e}(t))]^2 \\ &\triangleq \bar{E}[u(t) - \rho^T\Phi(v(t))]^2, \end{aligned} \tag{31}$$

where $\Phi(v(t)) \in \mathbb{R}^p$ is the regressor vector, which has the expanded signal $v(t)$ that was generate by the filtration of $\bar{e}(t)$.

The optimization problem proposed in (31) can be interpreted as an identification problem of a FIR model. With these ideas in mind, proposed by the VRFT method, the

function to be minimized is convex and its minimum can be obtained by the Least Squares (LS) method (CAMPI; LECCHINI; SAVARESI, 2002):

$$\hat{\rho} = \left[\sum_{t=1}^{N} \Phi(v(t))\Phi(v(t))^T\right]^{-1} \sum_{t=1}^{N} \Phi(v(t))u(t) \tag{32}$$

$$\Phi(v(t)) = \begin{bmatrix}\phi_1(v(t)) & \phi_2(v(t)) & \dots & \phi_n(v(t-N))\end{bmatrix} . \tag{33}$$

Notice that, to find the unique solution of (32), the matrix $\left[\sum_{t=1}^{N} \Phi(v(t))\Phi(v(t))^T\right]^{-1}$ must have full rank. The subsequent topic shows that minimizing the proposed criterion by the VRFT is equivalent to minimizing the model reference tracking performance criterion under ideal conditions.

3.1.1 Cost functions equivalence

To better illustrate the cost functions equivalency, we present an example based on (CAMPI; SAVARESI, 2006). A complete and deeper explanation can also be found in (CAMPI; SAVARESI, 2006).

Consider the nonlinear process:

$$y(t) = y(t-1) + u(t-1)^3 + \epsilon(t), \tag{34}$$

the controller class

$$u(t) = \rho[v(t)]^{\frac{1}{3}}, \tag{35}$$

$$v(t) = e(t), \tag{36}$$

which is a Proportional controller, and the reference model:

$$T_d(q) = \frac{1}{q} \tag{37}$$

which can be restated as

$$y_d(t) = r(t-1). \tag{38}$$

The data was collected with $N = 2$ samples and $u(0) = 1$ and $u(1) = 1$, yielding the output signal $y(1) = 1$ and $y(2) = 2$, through (26) we compute $\bar{r}(0) = 1$ and $\bar{r}(1) = 2$. Suppose that we know the nonlinear process. In this case, the cost function $J_y(\rho)$ could be precisely calculated as follows:

$$\begin{aligned} y(1,\rho) &= y(0,\rho) + \rho^3[\bar{r}(0) - y(0,\rho)] \\ &= 0 + \rho^3[1-0] = \rho^3; \end{aligned} \tag{39}$$

$$\begin{aligned} y(2,\rho) &= y(1,\rho) + \rho^3[\bar{r}(1) - y(1,\rho)] \\ &= \rho^3 + \rho^3[2-\rho^3] = 3\rho^3 - \rho^6; \end{aligned} \tag{40}$$

then the cost function $J_y(\rho)$ can be determined as:

$$\begin{aligned} J_y(\rho) &= [y(1,\rho) - \bar{r}(0)]^2 + [y(2,\rho) - \bar{r}(1)]^2 \\ &= (\rho^3 - 1)^2 + [(3\rho^3 - \rho^6) - 2]^2 \\ &= 5 - 14\rho^3 + 15\rho^6 - 6\rho^9 + \rho^{12}. \end{aligned} \tag{41}$$

Note that $\rho_0 = 1$ is the global minimum of this function which is portrayed in Figure 3.

Later, we describe the $J^{VR}(\rho)$ cost function,

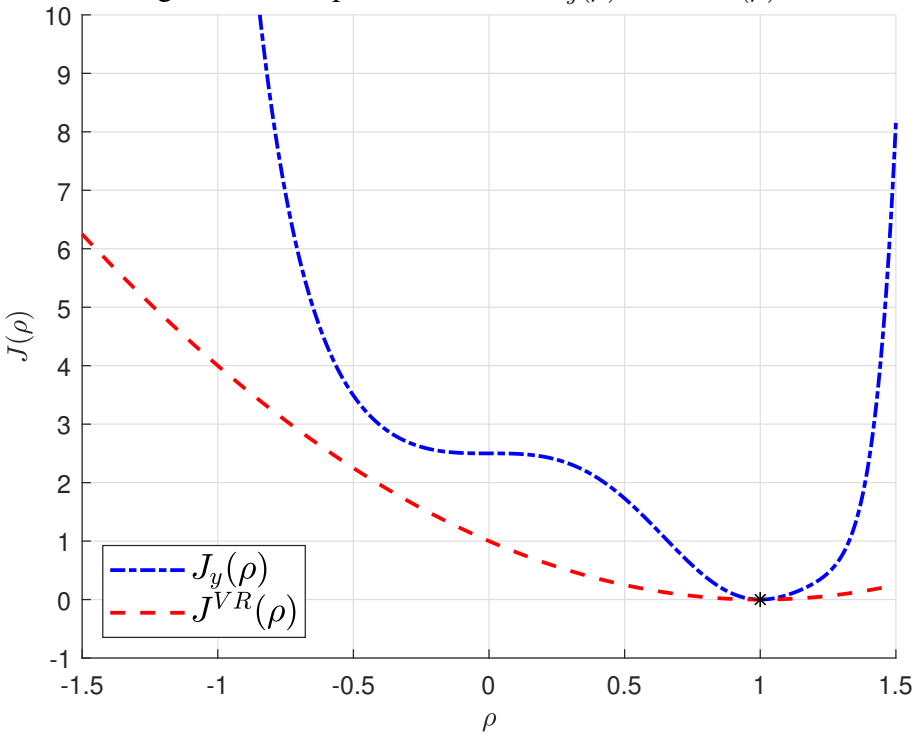
$$\bar{v}(0) = \bar{r}(0) - y(0) = 1 - 0 = 1 \tag{42}$$

$$\bar{v}(1) = \bar{r}(1) - y(1) = 2 - 1 = 1 \tag{43}$$

which gives:

$$\begin{aligned} J^{VR}(\rho) &= [\mathcal{C}(v(0),\rho) - u(0)]^2 + [\mathcal{C}(v(0),\rho) - u(1)]^2 \\ &= (\rho \cdot 1^{\frac{1}{3}} - 1)^2 + (\rho \cdot 1^{\frac{1}{3}} - 1)^2 \\ &= 2\rho^2 - 4\rho + 2. \end{aligned} \tag{44}$$

Figure 3 – Comparison between $J_y(\rho)$ and $J^{VR}(\rho)$.



Source: (Adapted from (CAMPI; SAVARESI, 2006).

3.2 Noisy Data

Up to the present moment, we have described the procedure to apply the VRFT method when the data is not corrupted by noise. From now on, we analyze the statistical properties when there is non-negligible level of noise in the data. On this occasion,

in applying the VRFT method to tune the controller's parameters, the variable $\hat{\rho}$ is not a deterministic quantity, but rather a random variable that represents an estimate of its true value ρ_0 (BAZANELLA; CAMPESTRINI; ECKHARD, 2012). In addition, we expose some features of the VRFT, where it is stated the limitations of its estimate and this work's motivation.

It is worth mentioning that this section supposes that the Assumption 2.1 is satisfied, i.e., the ideal controller is in the $\mathbb{C}$ class. Otherwise, the VRFT statistical properties will be subject to the data set characteristics' and how far the fixed controller class is to the ideal controller class (BAZANELLA; CAMPESTRINI; ECKHARD, 2012). This fact forestalls a generalized analysis, demanding a specific study for each situation.

Consider now the system defined on Chapter 2. Recalling that for obtaining the parameters estimate by the VRFT method it is necessary to determine the virtual error $\bar{e}(t)$ through the data set Z^N measured from the process. However, when we have noisy data, the virtual error will also be contaminated:

$$\bar{e}(t) = (T_d^{-1}(q) - 1)y(t), \tag{45}$$

with $y(t) = \mathcal{P}u(t) + \epsilon(t)$,

$$\bar{e}(t) = (T_d^{-1}(q) - 1)(\mathcal{P}u(t) + \epsilon(t)) \tag{46}$$

$$\bar{e}(t) = (T_d^{-1}(q) - 1)\mathcal{P}u(t) + (T_d^{-1}(q) - 1)\epsilon(t) \tag{47}$$

$$\bar{e}(t) = \bar{e}_0(t) + \tilde{e}(t). \tag{48}$$

Thereby, in addition to the virtual error that goes in the ideal controller on the original virtual loop interpretation, here denominated as $\bar{e}_0$, there is also another portion of the signal derived from the noise presence on the process's output signal. Note that in the original virtual loop interpretation there is no $\tilde{e}(t)$, therefore it does not go in the ideal controller.

Usually, the typical system identification problems are formulated only contemplating the noise presence in the system's output. Notwithstanding, this is not the case studied herein, because the virtual error actually is the input signal on the system to be identified. On the system identification community this kind of problem is known as *Errors-in-Variables* (SÖDERSTRÖM, 2018). In the sequence, we present the properties of this type of problem on the VRFT context.

It must be emphasized that the properties explored here are *asymptotic properties*, in other words, they are true when $N \to \infty$. Albeit, in a real application, it is impossible to use an infinite quantity of data, one may suppose that such properties can be approximated when operating with a large number of data N. For that reason, it should be noted that the fewer collected data on the identification process, the less credible are the properties.

The VRFT properties scrutinized in this thesis context are the estimate consistency and the bias error. These concepts are stated below.

Definition 3.1. *(SÖDERSTRÖM; STOICA, 1989) An estimate is said to be biased if its expected value is different from the true value:* $E[\hat{\rho}] \neq \rho_0$

Definition 3.2. *(SÖDERSTRÖM; STOICA, 1989) An estimate is said to be consistent if* $\hat{\rho} \to \rho_0$ *when* $N \to \infty$

The reader should not confuse these two concepts. The consistency is defined for $N \to \infty$, while the bias can be evaluated for a finite N. If an estimate is consistent, then it is not biased for $N \to \infty$. Although, this does not mean that it is no biased for a finite N.

Furthermore, it is important to write the input-output relation of the ideal controller as follows (reminding that the same is linearly parameterized):

$$u(t) = \mathcal{C}(\bar{C}(q)\bar{e}_0(t),\rho_0) \tag{49}$$

$$u(t) = \rho_0^T \Phi_0(v(t)) \tag{50}$$

where $\Phi_0(v(t))$ is the regressor matrix concerning the signal $\bar{e}_0(t)$, using only the portion that goes into the ideal controller $\bar{e}_0(t)$, rather than $\bar{e}(t)$. By observing the above equations, one can notice that the user does not possess the signal $\bar{e}_0(t)$ information independently, due to the noise additional portion that appears on the virtual error. The available signal to make the identification is indeed $\bar{e}(t)$, described by (45). For this reason, a way to rewrite $u(t)$ in terms of the available signals and a term corresponding to the stochastic contributions is

$$u(t) = \mathcal{C}(\bar{C}(q)\bar{e}(t),\rho_0) + \mathcal{C}((\bar{C}(q)\tilde{e}(t),\rho_0) \tag{51}$$

$$u(t) = \rho_0^T \Phi_0(v(t)) + \rho_0^T \tilde{\Phi}(v(t)) \tag{52}$$

with $\Phi(v(t)) = \Phi_0(v(t)) + \tilde{\Phi}(v(t))$, where $\tilde{\Phi}(v(t))$ is the regressor matrix concerning the signal $\tilde{e}(t)$.

Consider now the default VRFT estimate, the one that is determined by the least-squares algorithm (32), as discussed in the previous sections. From now on, we use the $\hat{\rho}^{VR}$ for the parameter's estimate. Suppose also that $N \to \infty$. In this manner, we may replace the sums in (32) by the $\bar{E}[\cdot]$ operator, resulting in

$$\hat{\rho}^{VR} = \left(\bar{E}[\Phi(v(t))\Phi(v(t))^T]\right)^{-1} \left(\bar{E}[\Phi(v(t))u(t)]\right). \tag{53}$$

A simplification can be made on the $\bar{E}[\Phi(v(t))\Phi(v(t))^T]$ term:

$$\begin{aligned} \bar{E}[\Phi(v(t))\Phi(v(t))^T] &= \bar{E}[(\Phi_0(v(t)) + \tilde{\Phi}(v(t))(\Phi_0^T(t) + \tilde{\Phi}^T(t))] \\ &= \bar{E}[\Phi_0(v(t))\Phi_0(v(t))^T] + \bar{E}[\Phi_0(v(t))\tilde{\Phi}(v(t))^T] \\ &+ \bar{E}[\tilde{\Phi}(v(t))\Phi_0(v(t))^T] + \bar{E}[\tilde{\Phi}(v(t))\tilde{\Phi}(v(t))^T]. \end{aligned} \tag{54}$$

Considering that the data was collected when the process was in open-loop, then $\Phi_0(v(t))$ holds the composed terms derived from the $\tilde{e}_0(t)$ signal, $\tilde{\Phi}(v(t))$ is derived from the $\tilde{e}(t)$ term and both signals are decorrelated, one can conclude that the cross correlation between $\Phi_0(v(t))$ and $\tilde{\Phi}(v(t))$ is zero. Finally, one writes the following equation:

$$\begin{aligned}\bar{E}[\Phi(v(t))\Phi(v(t))^T] &= \bar{E}[\Phi_0(v(t))\Phi_0(v(t))^T] + \bar{E}[\tilde{\Phi}(v(t))\tilde{\Phi}(v(t))^T] \\ &= R_{\Phi_0} + R_{\tilde{\Phi}}\end{aligned} \tag{55}$$

with $R_\Phi = \bar{E}[\Phi_0(v(t))\Phi_0(v(t))^T]$ and $R_{\tilde{\Phi}} = \bar{E}[\tilde{\Phi}(v(t))\tilde{\Phi}(v(t))^T]$, where R is utilized as in Section 2.1 to denote the autocorrelation, herein we suppress the parenthesis term ($\tau = 0$) to simplify the notation. Moreover, the term $\bar{E}[\Phi(v(t))u(t)]$ can be rewritten making use of the following relation $u(t) = \Phi_0(v(t))^T\rho_0$ as in (50). By using a similar logic as before, we have

$$\begin{aligned}\bar{E}[\Phi_0(v(t))u(t)] &= \bar{E}[\Phi_0(v(t))\Phi_0(v(t))^T\rho_0] \\ &= R_{\Phi_0}\rho_0\end{aligned} \tag{56}$$

Finally, replacing this terms on the parameters asymptotic estimate equation, we conclude that

$$\hat{\rho}^{VR} = (R_{\Phi_0} + R_{\tilde{\Phi}})^{-1}R_{\Phi_0}\rho_0. \tag{57}$$

Thus, calculating the difference between the estimate and the true value of the parameters, we find the following expression

$$\hat{\rho}^{VR} - \rho_0 = [(R_{\Phi_0} + R_{\tilde{\Phi}})^{-1}R_{\Phi_0} - I]\rho_0. \tag{58}$$

Deriving out of equation (58), we may conclude some interesting properties from the methodology.

Firstly, suppose that the noise on the process is negligible. As a consequence. the Signal-to-Noise Ratio (SNR) of the virtual error will be elevated. That implies, jumping some steps over, $R_{\Phi_0} >> R_{\tilde{\Phi}}$ and that we can approximate the following equation

$$R_{\Phi_0} + R_{\tilde{\Phi}} \approx R_{\Phi_0}. \tag{59}$$

Hence, analyzing (58), one can conclude that on the case that the noise is negligible, the estimate $\hat{\rho}^{VR} \to \rho_0$, in other words, the same will be consistent, as long as the inverse matrix $R_{\Phi_0}^{-1}$ exists. Without going into much detail, we highlight that this condition of existence of the $R_{\Phi_0}^{-1}$ matrix, is actually a persistence of excitation condition of the input signal $u(t)$ collected on the experiment (BAZANELLA; CAMPESTRINI; ECKHARD, 2012). If the reader wants to look into more details about the persistence of excitation of quasi-stationary signals, it is recommended the reading of (LJUNG, 1999).

At the same time, if the noise presence on the process is not negligible, the virtual error signal will present a low SNR. Thus, in this scenario, one cannot make the approximation previously made ($R_{\Phi_0} + R_{\tilde{\Phi}} \approx R_{\Phi_0}$). Then, one can conclude that the estimate $\hat{\rho}^{VR}$ will not be *consistent*, i.e, it will be biased for $N \to \infty$. In addition, it is worth noting that the lower the SNR (or the higher the noise effect on the process), the higher the trace of the $R_{\tilde{\Phi}}$ will be. On this case, it can be considered the following approximations:

$$R_{\Phi_0} + R_{\tilde{\Phi}} \approx R_{\tilde{\Phi}} \tag{60}$$

$$R_{\tilde{\Phi}}^{-1} R_{\Phi_0} \ll I. \tag{61}$$

At this time, by analyzing the equation (58), it is noticed that the estimate's value tends to zero, i.e., $\hat{\rho}^{VR} \to 0$. In this way, to the measure that the SNR is lower, the parameter's estimates tends to the value $\hat{\rho}^{VR} = 0$.

From the developed analysis, it was possible to realize that, dealing with process that have a considerable amount of noise, the default VRFT methodology will produce non consistent estimates (biased for $N \to \infty$). This characteristic of the VRFT method is remarkably undesirable, since that by augmenting the number of collected data, it is not possible to approximate to the ideal parameters value (BAZANELLA; CAMPESTRINI; ECKHARD, 2012). On top of that, the bias on the ρ_0 estimates yields a degradation of the closed-loop control systems that use this methodology of tuning the controller's parameters (CAMPI; LECCHINI; SAVARESI, 2002).

Another aspect that was not yet addressed is the complexity of the nonlinear controller generated by the library of nonlinear functions. As one expands the amount of nonlinear functions, one has many more parameters to estimate. A parsimonious controller will provide a more accurate estimate with as few terms as possible (BRUNTON; KUTZ, 2019).

Following this thesis, it is exhibited an illustrative example, which we present the standard nonlinear VRFT with the Least Squares solution to tune the controller's parameters of a nonlinear process, a Hammerstein one. The main focus of this example is to show the statistical properties of the VRFT method in a more concrete manner, showing the quality of the estimates and its effect on the closed-loop system.

3.3 Illustrative Example

With the purpose of demonstrating the statistical properties described in the previous section, this section presents an illustrative example of the nonlinear VRFT. Here, we present only the standard nonlinear VRFT approach so that the reader can get a better understanding of the method in the nonlinear context.

Consider the following nonlinear Hammerstein process, with the linear portion being:

$$G(q) = \frac{0.2}{q - 0.8}, \tag{62}$$

the static nonlinearity is a $\sqrt{.}$. So the nonlinear system can be rewritten as:

$$\frac{y(t)}{\sqrt{u(t)}} = \frac{0.2q^{-1}}{1 - 0.8q^{-1}}, \tag{63}$$

$$y(t) = 0.8y(t-1) + 0.2\sqrt{u(t-1)} + \epsilon(t). \tag{64}$$

The desired closed-loop performance chosen for the system is given by the following transfer function

$$T_d(q) = \frac{0.3}{q - 0.7}, \tag{65}$$

which guarantees a zero steady-state error for reference tracking of step.

In the linear case, the ideal controller $C_d(q)$ can be calculated through

$$C_d(q) = \frac{T_d(q)}{G(q)(1 - T_d(q))}, \tag{66}$$

and it would be the following

$$C_d(q) = \begin{bmatrix} 1.5 & 0.3 \end{bmatrix} \begin{bmatrix} 1 & \frac{1}{q-1} \end{bmatrix}^T, \tag{67}$$

which it is noticed to be a PI controller. Since we want to analyze the case where $\mathcal{C}_d(\rho) \in \mathbb{C}$, the linear portion of the controller class is also chosen to have the same structure:

$$\bar{C}(q) = \begin{bmatrix} 1 & \frac{1}{q-1} \end{bmatrix}^T. \tag{68}$$

Another element to consider is the inverse of the nonlinearity should be the function

$$f(.) = (.)^2. \tag{69}$$

The construction of this nonlinear map can be made by an expansion of the Taylor Series of the output signals of the linear part $\bar{C}(q)$, which are denominated as:

$$v_p(t) = e(t), \tag{70}$$

$$v_i(t) = \frac{1}{q-1}e(t), \tag{71}$$

$$v_i(t) = v_i(t-1) + e(t-1). \tag{72}$$

For this example, we expanded this signals up to the third order, thus generating 15 regressors vectors:

$$\Phi(v(t)) = \begin{bmatrix} v_p(t) & v_p^2(t) & v_p^3(t) & v_i(t) & v_i^2(t) & v_i^3(t) & v_p(t)v_i(t) & ... & v_p^3(t)v_i^3(t) \end{bmatrix}. \tag{73}$$

In the real data-driven control case, one does not know either the process or the ideal controller, thus the choice of the controller class $\mathbb{C}$ cannot be made by equation (66) but the nonlinearity can always be approximated by its Taylor Series Expansion. Therefore, by Assumption 2.1, one may define the $\mathcal{C}_d(\rho) = \mathcal{C}(v(t),\rho_0)$, with being the ideal parameters vector:

$$\begin{aligned}\rho_0^T &= \begin{bmatrix} 0 & K_p^2 & 0 & 0 & K_i^2 & 0 & 2K_pK_i & ... & 0 \end{bmatrix}^T \\ &= \begin{bmatrix} 0 & 2.25 & 0 & 0 & 0.09 & 0 & 0.9 & ... & 0 \end{bmatrix}^T ,\end{aligned} \tag{74}$$

then the ideal controller would be

$$\mathcal{C}_d(v(t)) = \mathcal{C}(v(t),\rho) = \rho_0^T \Phi(v(t)). \tag{75}$$

For the data collection we excited the system with an input signal called APRBS (*Amplitude Modulated Pseudo-Random Binary Sequence*) of size $N = 1500$. This signal can be described by as a Pseudo Random Binary Signal (PRBS) multiplied by the absolute value of a Gaussian noise with zero mean and variance $\sigma^2 = 1$. Since we have a nonlinear process, the input signal can not have a constant amplitude.The output white noise standard deviation is $\sigma_\epsilon = 2 \times 10^{-2}$.

In the interest of assessing the statistical properties proposed in the literature, we performed 100 Monte Carlo simulations with different noise realizations. At each simulation the input and output signals were collected and the controller's parameters were tuned with the standard VRFT approach. Four different criteria were evaluated on this analysis:

(i) The distributions of the cost function $J_y(\rho)$;

(ii) The distributions of the parameters K_p^2 and K_i^2;

(iii) The time responses of the closed-loop system with a square wave input signal;

(iv) The controller's parameters on the $K_p^2 \times K_i^2$ plan.

The four criteria exhibit the estimate effect on the closed-loop system, which is the main aspect to be evaluated on the control context. The latter is related to the estimate's quality in itself.

Firstly, we analyze the most important criterion for the method studied herein, the $J_y(\rho)$. From the obtained parameters at each simulation, the cost was calculated as follows

$$J_y(\hat{\rho}) = \frac{1}{N}\sum_{t=1}^{N}(y(t,\hat{\rho}) - y_d(t))^2. \tag{76}$$

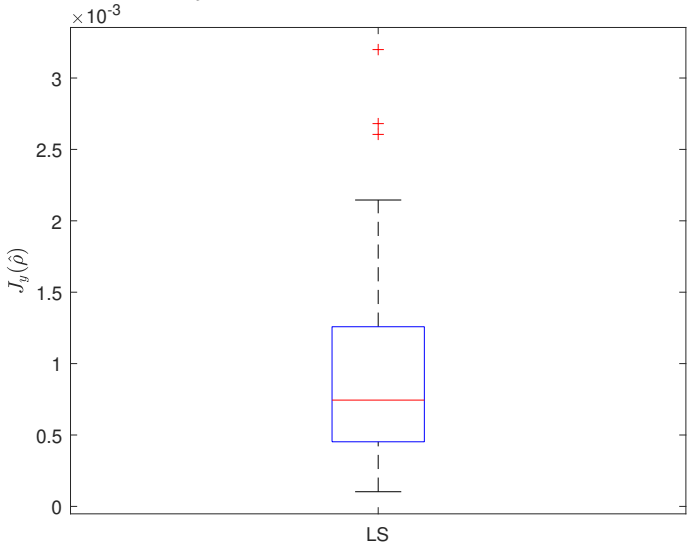
with the reference signal $r(t)$ being a square wave signal with period equal to 10 seconds and $N = 100$ samples.

Figure 4 demonstrates the distributions of $J_y(\hat{\rho})$ observed in the Monte Carlo simulations by means of *boxplots*. In this kind of diagrams, the center line in red represents the median of the observations, the blue box represents the interval between 25% and 75% of the samples, whereas the black dashed lines extend themselves until the extreme values that are not considered outliers. The "$+$" symbols on the top of the diagram display how many samples were above the maximum value that the boxplot comprehends.

It is important to highlight that the discrepancy cases are not taken into account for the median, the box and extreme values determination.

Figure 4 shows that the median for the standard nonlinear VRFT was 7.444×10^{-4}. Clearly, the median value can be explained due to the bias error that is present in this method, which is propagated to the closed-loop performance and causes distinct behaviors compared to the one specified by the $T_d(q)$ transfer function.

Figure 4 – $J_y(\hat{\rho})$ boxplot for the standard VRFT.



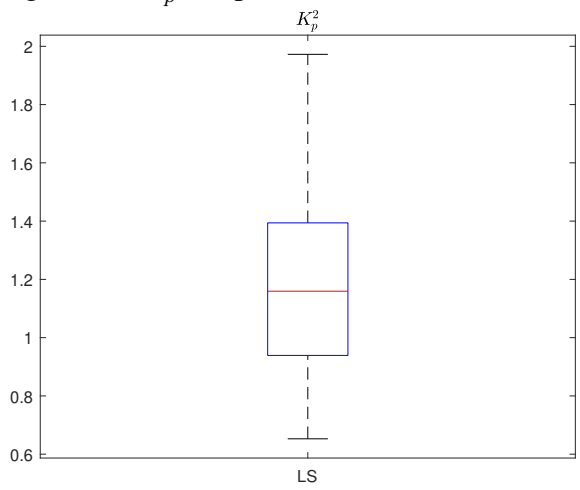
Source: author.

Figures 5 and 6 exhibit the boxplots for the K_p^2 and K_i^2 parameters. By observing these figures, one can see the bias effect on the controller's parameters, their median values were 1.1594 and 0.0655 respectively, which is far away from the ideal value (2.25 and 0.09).

In Figure 7, it is possible to observe both the variance and the bias error on the $K_p^2 \times K_i^2$ plan. The red square represents the average for the controller's parameters, the blue dot are the controller's parameters and the black dot is the ideal parameter ρ_0.

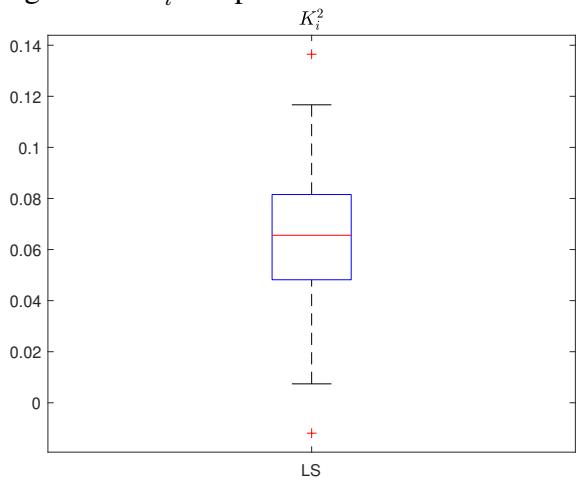
Finally, we present the obtained closed-loop time responses on Figure 8. The blue lines are the process's output signal and the black line is the desired time response. As we can see, the transient behavior was slightly slower and with a considerable overshoot compared to the one specified by $T_d(q)$. Another important point was that the system did not become unstable with the estimated controller's parameters.

Figure 5 – K_p^2 boxplot for the standard VRFT.



Source: author.

Figure 6 – K_i^2 boxplot for the standard VRFT.



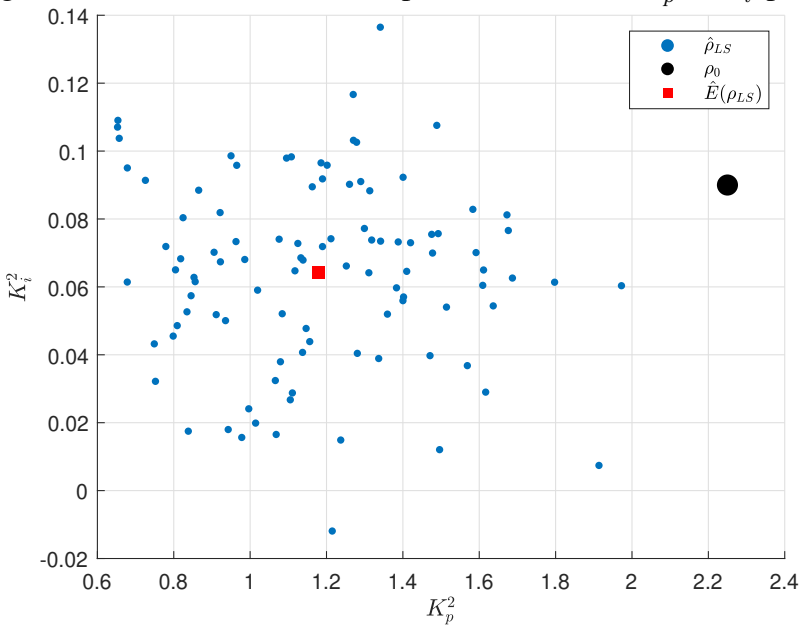
Source: author.

By this illustrative example the main motivation of this work is apparent: the statistical properties yielded by the standard nonlinear VRFT method are not quite adequate when in the noise presence. As a result, the estimated controller depreciates the closed-loop performance of the system in a considerable way. Therefore, this thesis main focus is to improve these properties, by including the ℓ_1-regularization on the standard nonlinear VRFT, as well as getting a parsimonious controller (i.e. with as few terms as possible).

3.4 Chapter Conclusions

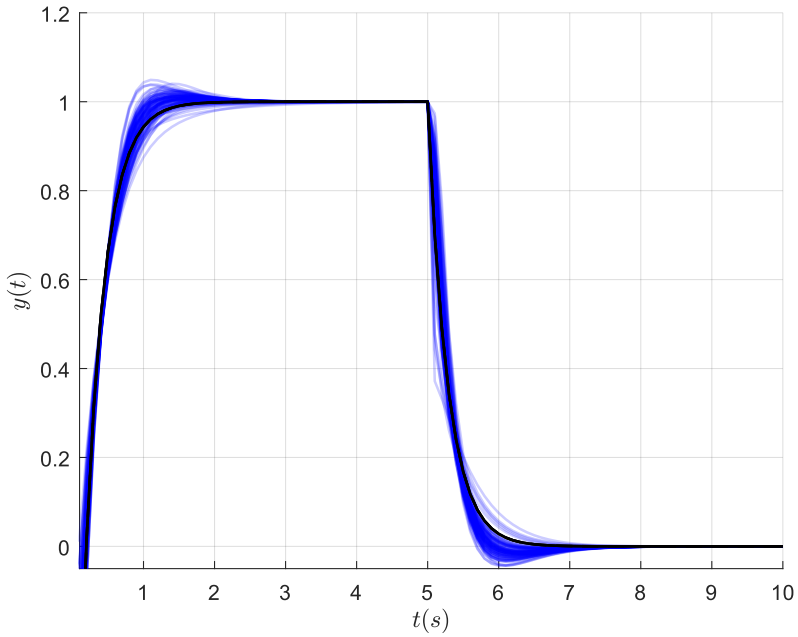
This chapter introduced the reader to the already consolidated in the literature, the data-driven method known as the Nonlinear VRFT. Firstly, Section 3.1 described the Nonlinear VRFT for understanding its characteristics. The main idea of the method was explained. It consists of rewriting the optimization problem of the $J_y(\rho)$ criterion, in an

Figure 7 – Estimated controller's parameters on the $K_p^2 \times K_i^2$ plan.



Source: author.

Figure 8 – Comparison of the time response for 100 Monte Carlo simulations.



Source: author.

ideal controller identification problem, where the $J^{VR}(\rho)$ criterion is minimized. It was shown that within of the ideal situation the optimization of the two criteria results in the same minimum.

In Section 3.2, the problem of using noisy data in VRFT was addressed in a generic way. The properties of estimates of ρ_0 with the original formulation (LS) were discussed,

emphasizing that it has a deficiency: the polarization error.

Finally, in Section 3.3 an example demonstrating the properties cited has been developed. In the example, it was found that the low quality of the estimates of LS propagates to the performance of systems in closed-loop, degrading it significantly in relation to the desired performance. Thus, it became evident the motivation of the work, which seeks to improve the properties of Nonlinear VRFT. It is expected that the results in closed-loop will be improved and the controller's complexity as well.

4 NONLINEAR VRFT WITH REGULARIZATION

This chapter presents the regularization concepts applied to the system identification context. We present the difference between the ℓ_1 and ℓ_2 regularization methods with the purpose of explaining which one suits better for this work. Firstly we consider a generalized model (that is linear in the parameters) and then we move into the controller identification using the VRFT method.

In spite of being a tool already known inside the system identification community, the regularization has come to prominence in the last few years due to innovative works as the ones in (PILLONETTO; NICOLAO, 2010; PILLONETTO; CHIUSO; DE NICOLAO, 2011), which show the Bayesian perspective to identify the impulse response of linear dynamical systems. Besides, this perspective is widely utilized and developed in the Machine Learning community, thus producing various studies about this tool in different Machine Learning Models (CHEN; LJUNG, 2013).

The most known regularization techniques are the ℓ_2 (known as Ridge Regression) and the ℓ_1 (called LASSO Regression). These two techniques combined yield another method: the Elastic Net Regression which in turn weighs the two methods through a scalar.

With the intent to illustrate these techniques, consider now the following model:

$$y(t) = \theta^T x(t) + w(t), \tag{77}$$

where $x(t) \in \mathbb{R}^p$ is a deterministic regressor vector, $\theta \in \mathbb{R}^p$ is the parameter vector and $w(t) \in \mathbb{R}$ is a white Gaussian noise with zero mean. The problem of finding the parameter's vector is denominated *linear regression*. The linear regression problem can be solved by many techniques, with the most common being the Least Squares solution. The LS method has the advantage of being a non-iterative technique, i.e. it does not need initialization for the parameters and it does not involve the gradient or the Hessian of the cost function. When the regressor vector has as many regressors as possible i.e. we have an overdetermined system, the LS estimator will not be able to choose and weigh the essentials ones. Hence, it yields a complex and incomprehensible model (TIBSHIRANI, 1996).

Ridge Regression is very similar to the LS method, except the coefficients are estimated by adding a *shrinkage penalty* as the equation (78) shows:

$$\hat{\theta}_{Ridge}(\lambda) = \arg\min_{\theta} \frac{1}{N} \sum_{t=1}^{N} (y(t) - \theta^T x(t))^2 + \lambda \sum_{j=1}^{p} \theta_j^2, \tag{78}$$

where $\lambda \geq 0$ is a tuning parameter to be determined separately. As with LS, ridge regression seeks the parameters that fit the data well, by minimizing the residual sum of squares (RSS). Nonetheless, the second term $\lambda \sum_{j=1}^{p} \theta_j^2$ (the shrinkage penalty) is small when the θ_1,...,θ_p are close to zero, yielding the effect of *shrinking* the estimates of θ_j towards zero. The λ parameter acts as a handler of the relative impact of these two terms on the regression coefficient estimates. When $\lambda = 0$, the penalty term has no effect, thus ridge regression will produce the LS estimates. However, as $\lambda \to \infty$ the impact of the shrinkage penalty grows, producing parameters closer to zero. Unlike LS, which bears only one set of parameter estimates, ridge regression will produce different set of parameter estimates for each value of λ. Therefore, selecting a good value of λ is extremely important. Later on, we present this discussion using the validation and cross-validation (CV) techniques.

The advantage in applying Ridge Regression over Least Squares is based on the *bias-variance trade-off*. As λ raises, the flexibility of the ridge regression fit decreases, leading to a decreased variance but increased bias (JAMES *et al.*, 2013).

4.1 LASSO

The undeniable disadvantage on the ridge regression is the inclusion of all p regressors in the final model. Evidently the penalty $\lambda \sum_{j=1}^{p} \theta_j^2$ will shrink all the parameters towards zero, but it will not set any of them exactly to zero, unless $\lambda = \infty$. So, in overdetermined models, in which the number of parameters p is quite large the ridge regression is not suitable. Because ridge regression will always generate a model involving all regressors, as one increases the value of λ it will reduce the magnitude of the parameters, but it will not result in exclusion of any of them. Thus, it yields a complex and unabbreviated model.

The LASSO (Least Absolute Shrinkage and Selection Operator) is a quite recent alternative to ridge regression that overcomes the ridge regression disadvantage. The LASSO is a technique developed in (TIBSHIRANI, 1996) to perform model selection and parameter estimation simultaneously. Frequently in this kind of setup the number of regressors (the columns of $x(t)$) are not related to the measured system's output $y(t)$, i.e. one has more regressors than the "real" system. It means that the parameter vector (θ_0) is *sparse* (it has many zero components). The model selection problem is such that one determines which components of θ are zero. Thus, the regularization will be necessary.

The parameters in the model (77) can be estimated by the LASSO (TIBSHIRANI,

1996):

$$\hat{\theta}(\lambda) = \arg\min_{\theta} \left\| y(t) - \theta^T x(t) \right\|_2^2 + \lambda \left\| \theta \right\|_1 , \tag{79}$$

where $\left\| y - \theta^T x(t) \right\|_2^2 = \frac{1}{N} \sum_{t=1}^{N} (y(t) - \theta^T x)^2$, $\left\| \theta \right\|_1 = \sum_{j=1}^{p} |\theta_j|$, and $\lambda \geq 0$ is a penalty coefficient. The LASSO estimator has a property that it does variable selection in the sense that $\hat{\theta}(\lambda) = 0$ for some j's (depending on the choice of λ) and $\hat{\theta}(\lambda)$ can be thought as a shrunken least squares estimator; hence the name Least Absolute Shrinkage and Selection Operator (LASSO). An explanation for the variable selection property is given below.

The optimization for (79) is convex, so it enables efficient computation of the estimator. Besides this optimization problem may be equivalently written as

$$\begin{aligned} \hat{\theta} = \arg\min_{\theta} \quad & \left\| y(t) - \theta^T x(t) \right\|_2^2 \\ s.t. \quad & \left\| \theta \right\|_1 \leq \kappa \end{aligned} , \tag{80}$$

with one-to-one correspondence between λ and κ depending on the data. Such equivalence holds considering that $\left\| y - \theta^T x(t) \right\|_2^2$ is convex in θ with a convex constraint $\left\| \theta \right\|_1 \leq \kappa$ (BÜHLMANN; VAN DE GEER, 2011).

Also for Rigde regression one may rewrite (78) as:

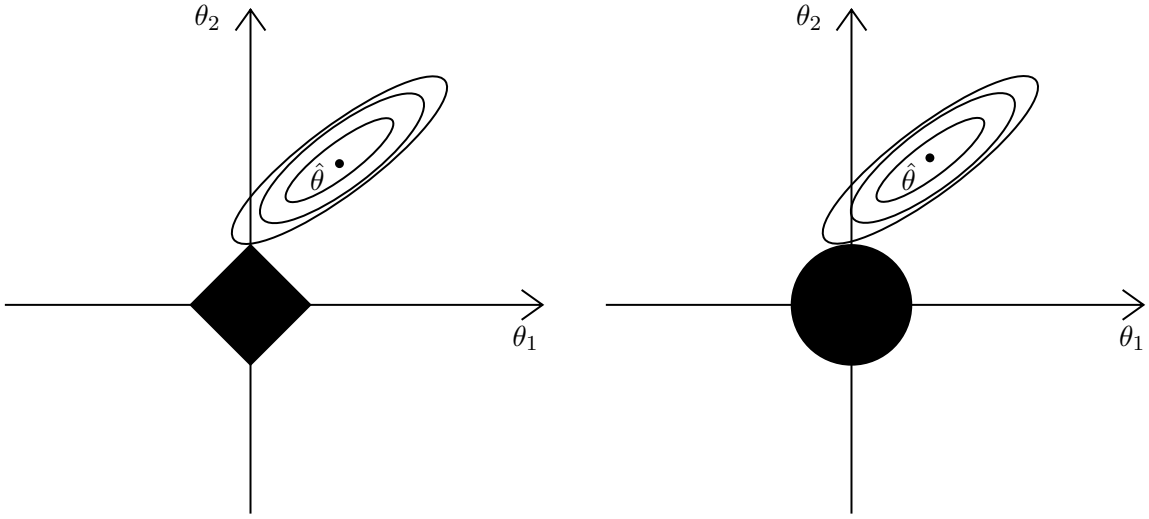
$$\begin{aligned} \hat{\theta} = \arg\min_{\theta} \quad & \left\| y(t) - \theta^T x(t) \right\|_2^2 \\ s.t. \quad & \left\| \theta \right\|_2 \leq \kappa \end{aligned} . \tag{81}$$

The variable selection in the sense that a component of the parameter vector θ can be exactly zero in the LASSO is explained by the ℓ_1 geometry. The formulations (80) and (81) can be used to shed light on the issue. Figure 9 illustrates the situation. The Least Squares solution is marked as $\hat{\theta}$, while the black diamond and circle represent the LASSO and ridge regression constraints in (80) and (81), respectively. If κ is sufficiently large, then the constraint regions will contain $\hat{\theta}$, and so the ridge regression and LASSO estimates will be the same as the least squares estimates. Such a large value of κ corresponds to $\lambda = 0$ in (78) and (79). However, in Figure 9 the least squares estimates lie outside of the diamond and the circle, in other words it is not the same as the LASSO and ridge regression estimates.

Each of the ellipses centered around $\hat{\theta}$ represents a *contour*: this means that all of the points on a particular ellipse have the same residual sum of squares value. As the ellipses expand away from the least squares parameters estimates, the RSS increases. Equations (80) and (81) indicate that the LASSO and ridge regression parameters estimates are given by the first point at which an ellipse contacts the constraint region. Since the ridge regression has a circular constraint with no sharp points, this intersection will be exclusively non-zero. However, the LASSO constraint has corners at each of the axes, and so the

ellipse will often intersect the constraint region at an axis. When this occurs, one of the coefficients will equal zero. In higher dimensions, many of the coefficient estimates may equal zero simultaneously. In Figure 9, the intersection occurs at $\theta_1 = 0$, and so the resulting model will only include θ_2.

Figure 9 – Contours of the error and constraint functions for the LASSO (left) and Ridge regression (right). The solid black areas are the constraint regions, $|\theta_1| + |\theta_2| \leq \kappa$ and $\theta_1^2 + \theta_2^2 \leq \kappa$ while the ellipses are the contours of the residual sum of squares.



Source: Adapted from (BÜHLMANN; VAN DE GEER, 2011).

In Figure 9 we considered the simple case of $p = 2$. When $p = 3$, then the constraint region for ridge regression becomes a sphere, and the constraint region for the LASSO becomes a polyhedron. When $p > 3$, the constraint for ridge regression becomes a hypersphere, and the constraint for the LASSO becomes a polytope. However, the key ideas depicted in Figure 9 still hold. In particular, the LASSO leads to feature selection when $p > 2$ due to the sharp corners of the polyhedron or polytope (JAMES *et al.*, 2013).

Implementing ridge regression and the LASSO requires a method for selecting a value for the tuning parameter λ in (78) and (79), or equivalently, the value of the constraint κ in (80) and (81). Cross-validation provides a simple way to tackle this problem. First, we choose a grid of λ values, and compute the cross-validation error for each value of λ. We then select the tuning parameter value for which the cross-validation error is smallest. Finally, the model is re-identified using all of the available data and the selected value of the tuning parameter.

In the next subsections we present three different ways of finding the best λ within a predefined range.

4.1.1 Validation set

The *Validation Set* approach is a really simple procedure to estimate the error associated with estimation of a particular model on a set of data. In brief, the validation set

involves randomly dividing the available set of data into two parts, a *learning set* and a *validation set* or *hold-out set*. Basically, the model is identified on the learning set and then it is used to predict the responses for the observations in the validation set. The resulting set error, the MSE, provides an estimate of the test error rate.

The validation set approach is conceptually simple and direct, besides it is easy to implement. However, it has two potential drawbacks:

- The validation estimate of the test error rate can be highly variable, depending on precisely which observations are included in the learning set and which observations are included in the validation set.

- In the validation approach, only a subset of the observations - those that are included in the learning set rather than in the validation set - are used to fit the model. Since statistical methods tend to perform worse when learned on fewer observations, this suggests that the validation set error rate may tend to *overestimate* the test error rate for the model fit on the entire data set.

4.1.2 Leave-one Out Cross-Validation

The *cross-validation* approach intends to address the issues with the classical validation set. The Leave-one Out Cross-Validation (LOOCV) is one of them (BÜHLMANN; VAN DE GEER, 2011).

As with the validation set approach, LOOCV concerns splitting the set of data into two parts. However, instead of creating two subsets of comparable size, a single sample of data $[x(1),y(1)]$ is used for the validation set, and the other samples make up the learning set $[x(2),y(2)]$,...,$[x(N),y(N)]$. The employed method is estimated on the $N-1$ samples, and a prediction $\hat{y}(1)$ is made for the excluded sample, using its value $x(1)$. Since the first sample was not used to estimate the model, the error $[y(t)-\hat{y}(t)]^2$ provides an approximately unbiased estimate. But one can not think that with one single sample the model is well estimated. Thus, one can repeat the procedure for each sample, i.e. N times. Then, one can calculate the test MSE for the LOOCV approach as:

$$CV(N) = \frac{1}{N}\sum_{t=1}^{N}(y(t)-\hat{y}(t))^2. \tag{82}$$

LOOCV has a couple of major advantages over the validation set approach. First, it has far less bias. In LOOCV, we repeatedly fit the employed method using learning sets that contain $N-1$ observations, almost as many as are in the entire data set. This is in contrast to the validation set approach, in which the learning set is typically around half the size of the original data set. Consequently, the LOOCV approach tends not to overestimate the test error rate as much as the validation set approach does. Second, in contrast to the validation approach which will yield different results when applied repeatedly due

to randomness in the learning/validation set splits, performing LOOCV multiple times will always yield the same results: there is no randomness in the learning/validation set splits.

4.1.3 k-Fold Cross-Validation

An alternative to LOOCV is k-fold CV. This approach involves randomly k-fold CV dividing the set of observations into k groups, or folds, of approximately equal size. The first fold is treated as a validation set, and the method is fit on the remaining $k - 1$ folds. The mean squared error, MSE_1, is then computed on the observations in the held-out fold. This procedure is repeated k times; each time, a different group of observations is treated as a validation set. This process results in k estimates of the test error, MSE_1; MSE_2;...; MSE_k. The k-fold CV estimate is computed by averaging these values,

$$CV(k) = \frac{1}{k}\sum_{t=1}^{k} MSE_t. \tag{83}$$

It is quite straightforward that LOOCV is a special case of k-fold CV in which k is set to equal N. In practice, one typically performs k-fold CV using $k = 5$ or $k = 10$. The noticeable advantage of using k-fold CV rather than LOOCV is the computational effort. Another fact that is that there is a bias-variance trade-off associated with its choice. Typically, one performs k-fold cross-validation using $k = 5$ or $k = 10$, as these values have been shown empirically to yield error estimates that suffer neither from excessively high bias nor from very high variance (JAMES *et al.*, 2013). We advice the reader the book (JAMES *et al.*, 2013) if he/she wants to study more about the cross-validation procedure.

In the next sections we present the Regularized VRFT with the LASSO and two different regularization methods presented in the VRFT context.

4.2 Regularized VRFT

The Regularized VRFT is made by simply adding the ℓ_1 penalty on equation (31). Therefore, the Regularized VRFT goal is to solve the following optimization problem

$$\rho^*_{Reg} = \arg\min_{\rho_{Reg}} J^{VR}_{Reg}(\rho), \tag{84}$$

$$J^{VR}_{Reg}(\rho) \triangleq J^{VR}(\rho) + \lambda \sum_{j=1}^{P} |\rho_j| \,. \tag{85}$$

In this work's framework we apply the MATLAB function *lasso* in order to find ρ^*_{Reg}. In addition, we use the k-fold cross-validation method to find the λ coefficient.

4.3 Sequential Thresholded Least Squares

Concerning the computational endeavor and with the same intention to find a sparse solution on overdetermined systems, the Sequential Thresholded Least Squares (STLS) is proposed in (BRUNTON; PROCTOR; KUTZ, 2016) to identify model parameters of nonlinear dynamical systems. In this work, we exploit the STLS to identify the controller's parameters.

The SLTS method consists on zeroing the parameters found by the Least Squares method that are smaller than some threshold value λ_{STLS}, and it can be described as the primal solution to the LASSO objective function. Afterward, we recalculate the remaining parameters using the Least Squares only onto the remaining indices.

The STLS algorithm applied herein is presented below:

Algorithm 1 Sequential Thresholded Least-Squares

Data: Reference Model $T_d(q)$, controller structure $\bar{C}(q)$, library ϕ, threshold λ_{STLS}, measured data ($u(t)$ and $y(t)$), $t = 1,...,N$

Result: Estimated parameters ρ_{STLS}

Generate the virtual reference and the regressor matrix

$\overline{r}(t) = T_d^{-1}(q)y(t)$

$\overline{e}(t) = \overline{r}(t) - y(t)$

$v(t) = \overline{C}(q)\overline{e}(t)$

Generate the regressor matrix using the library ϕ

$\Phi = \quad [\phi_1(v(t)) \quad \phi_2(v(t)) \quad ... \quad \phi_n(v(t-N))]$

Search for the small parameters

Initial guess: least-squares $\rho = (\Phi^T\Phi)^{-1}\Phi^T u(t)$

Determine the ρ indexes less than λ_{STLS}

$\alpha = |\rho| \leq \lambda_{STLS}$

Threshold the parameters

$\rho_\alpha = 0$

Determine the ρ indexes greater than λ_{STLS}

$\beta = |\rho| > \lambda_{STLS}$

Regress the dynamics onto remaining terms

$\rho_{STLS} = (\Phi_\beta^T\Phi_\beta)^{-1}\Phi_\beta u(t)$

where $\alpha \in \mathbb{R}^p$, with p being the number of zero parameters and $\beta \in \mathbb{R}^q$, with q being the number of nonzero parameters.

4.4 Sequential Thresholded Least Squares 2

Finding an appropriate threshold in the STLS just described is a critical task for which there seems to be no firm guidelines in the literature. It is doubtful whether such firm

guidelines can ever be derived, since a single threshold must be applied to parameters with hugely different units. Thus, it seems wiser to evaluate the parameters whose net contribution to the objective function (30) is smaller than a threshold instead; specifically, $\sum_{j=1}^{n} |\rho_j \Phi_j(t)| < \lambda$. This is what we propose here in this work, under the name Sequential Thresholded Least Squares 2 (STLS_2).

The STLS_2 algorithm is defined as follows:

Algorithm 2 Sequential Thresholded Least-Squares 2

Data: Reference Model $T_d(q)$, controller structure $\bar{C}(q)$, library ϕ, threshold λ_{STLS_2}, measured data ($u(t)$ and $y(t)$), $t = 1,...,N$

Result: Estimated parameters ρ_{STLS_2}

Generate the virtual reference and the regressor matrix

$\overline{r}(t) = T_d^{-1}(q)y(t)$

$\overline{e}(t) = \overline{r}(t) - y(t)$

$v(t) = \overline{C}(q)\overline{e}(t)$

Generate the regressor matrix using the library ϕ

$\Phi = \quad [\phi_1(v(t)) \quad \phi_2(v(t)) \quad ... \quad \phi_n(v(t-N))]$

Search for the small parameters

Initial guess: least-squares $\rho = (\Phi^T\Phi)^{-1}\Phi^T u(t)$

Determine the ρ indexes contribution less than λ_{STLS_2}

$\alpha = |\rho_j \phi_j| \leq \lambda_{STLS_2}$

Threshold the parameters

$\rho_\alpha = 0$

Determine the ρ indexes greater than λ_{STLS_2}

$\beta = |\rho| > \lambda_{STLS_2}$

Regress the dynamics onto remaining terms

$\rho_{STLS} = (\Phi_\beta^T \Phi_\beta)^{-1}\Phi_\beta u(t)$

In the next chapter we present four study cases in order to illustrate the regularization methods and their implications into the closed-loop performance.

4.5 Chapter conclusions

This chapter was responsible for presenting to the reader the fundamental concepts of use of regularization in models that are linear in the parameters, aiming at the application of this tool in the VRFT method.

Firstly, Section 4.1 compared the ℓ_2 and ℓ_1 regularization methods. We showed that the ℓ_1 regularization method is the best suited for this work's purpose. Moreover, we discussed about three particular methods for finding the best penalty parameter λ, which is the k-fold Cross-Validation.

In Section 4.3, we disclosed an alternative to the LASSO method to find an sparse solution to the parameters vector θ along with its algorithm. The STLS can be interpreted as the primal formulation for the LASSO objective function.

Finally, in Section 4.4 we propose a more adequate methodology that takes into consideration the contribution of the parameter to the objective function value.

5 CASE STUDIES

The objective of this chapter consists on showing some simulated study cases to elucidate the regularization concepts and their effects on the Classical Nonlinear VRFT. Four different study cases on nonlinear systems are displayed in this Section. The experiments on each example were conducted by simulations on MATLAB.

The experiments were simulated in order to explore different controller structures (the linear portion $\bar{C}(q)$ and also the library of nonlinear functions $\phi(.)$). The first three examples are Hammerstein processes, in the first one we consider a second order reference model and the ideal controller is matched. The second example a first order reference model is considered and the ideal controller can be matched, the third one is the same as the second with the difference that the ideal controller cannot be matched due to the controller structure. In the last and fourth example we present a nonlinear system with bilinearities and quadratic nonlinearities.

One of the metrics employed to compare the obtained results is the objective function estimate for the reference tracking performance criterion $\hat{J}_y(\hat{E}(\rho))$, given by

$$\hat{J}_y(\hat{\rho}) = \frac{1}{N}\sum_{t=1}^{N}(y(t,\hat{\rho}) - y_d(t))^2, \tag{86}$$

where $\hat{\rho}$ is the estimated parameters vector for each noise realization, N is the number of collected samples, $y(t,\hat{\rho})$ and $y_d(t)$ are the closed-loop response for the process and reference model, respectively. It is worth mentioning that this function is calculated for each estimated controller and for each method.

The other metric employed in this chapter for all the N Monte Carlo simulations is the average controller parameters denoted by $\hat{E}(\rho)$, which is calculated through

$$\hat{E}(\rho) = \frac{1}{N}\sum_{k=1}^{N}\hat{\rho}_k. \tag{87}$$

The average controller parameters are used to estimate the bias of each method in the matched case examples.

5.1 Hammerstein Process 1 - Matched Case

In this first example, we operated some experiments on a first order process, which is described by the following transfer function

$$G(q) = \frac{0.2}{q - 0.8}. \tag{88}$$

This process' settling time is approximately 18 samples, considering that the 5% criterion was applied. Since this example considers a Hammerstein process, we must define the static input nonlinearity. In this case the nonlinearity is a $\sqrt{(.)}$.

The reference model selected for this case is a second order linear transfer function. Thus, we choose the following transfer function for the closed-loop system.

$$T_d(q) = \frac{0.01}{(q - 0.9)^2}, \tag{89}$$

and $t_s = 0.1s$.

In the linear case, the ideal controller $C_d(q)$ would be the following

$$C_d(q) = \frac{0.05}{q - 1}, \tag{90}$$

so we adopted a pure integrator for $\bar{C}(q) = \frac{1}{q-1}$. Then the signal $v_i(t)$ is described as

$$v_i(t) = e(t)\frac{1}{q - 1}, \tag{91}$$

$$v_i(t) = v_i(t) + e(t - 1). \tag{92}$$

As we discussed in Section 2.2.1 we must expand the signals that come from the output of $\bar{C}(q)$. In this case we expand the signal $v_i(t)$ up to the fifth order, thus generating 5 regressors vectors:

$$\Phi = \begin{bmatrix} v_i(t) & v_i^2(t) & v_i^3(t) & v_i^4(t) & v_i^5(t) \end{bmatrix}. \tag{93}$$

The open-loop experiment was handled by an APRBS of size $N = 1500$. Recapitulating Chapter 3, this signal can be described by as a Pseudo Random Binary Signal (PRBS) – with an amplitude of 2 – multiplied by the absolute value of a Gaussian noise with zero mean and variance $\sigma^2 = 1$. The selected white noise standard deviation is $\sigma_\nu = 1 \times 10^{-2}$.

Concerning the LASSO algorithm, the MATLAB function *lasso* was used. The regularization parameter λ_{LASSO} was calculated through the 5-fold Cross Validation procedure so that it would yield minimum variance, the threshold parameter for STLS is $\lambda_{STLS} = 1 \times 10^{-3}$. Regarding the STLS$_2$ method, λ_{STLS_2} is determined such that only the parameters that contribute 50% or less to the objective function are thresholded.

To evaluate the proposed technique, 100 Monte Carlo simulations were run with distinct noise realizations. The major objective of inserting the regularization on the VRFT

was to draw a better closed-loop performance. This evaluation was done through the objective function $J_y(\rho)$, in addition to the sum of all the estimated zeros in each Monte Carlo simulation.

Table 1 exhibits the average controller gains that should be the nonzero parameters.

Table 1 – Average Estimated Controller Gains.

Regressor	$\hat{E}(\rho_{LS})$	$\hat{E}(\rho_{LASSO})$	$\hat{E}(\rho_{STLS})$	$\hat{E}(\rho_{STLS_2})$	ρ_0
v_i	0.0014	0.0024	0.0015	0	0
v_i^2	0.0023	0.0023	0.0024	0.0025	0.0025
v_i^3	0.0000	0.0000	0	0.0000	0
v_i^4	-0.0000	0.0000	0	-0.0000	0
v_i^5	0.0000	-0.0000	0	0	0

Seeing that we calculated the parameters on Table 1 using the 100 Monte Carlo simulations, the zero gains found were observed all over the 100 simulations.

To evaluate the sparsity of the estimation along all the Monte Carlo Simulations, we present Table 2, which contains the number of estimated zeros by the four methods and the ideal quantity as well.

Table 2 – Total number of zeros

Method	N_0
LS	0
LASSO	209
STLS	307
STLS_2	351
Ideal	400

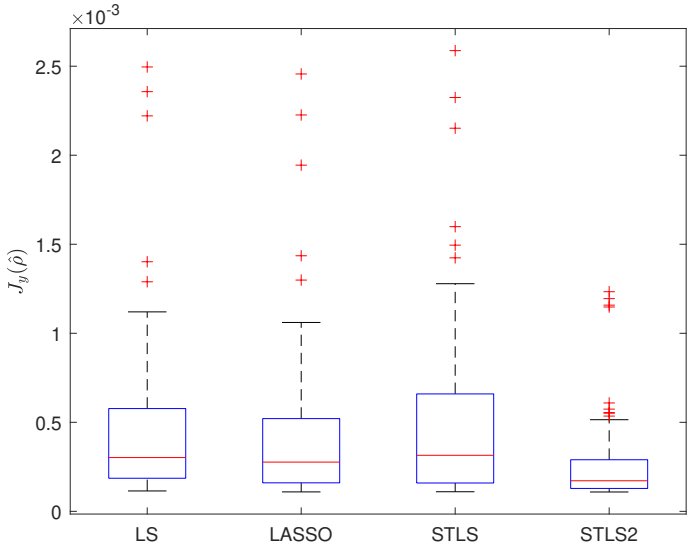
If now we turn to the interpretation of the objective function, through Table 3 it is evident that the only regularization method outperforming the classical VRFT with the Least-Squares is the STLS_2. The cost $\hat{J}_y(\hat{E}(\rho_{LS}))$ is 12% worst than the minimum, $\hat{J}_y(\hat{E}(\rho_{LASSO}))$ is 38%, $\hat{J}_y(\hat{E}(\rho_{STLS}))$ is 16%, while the STLS_2 achieves the minimum value amongst all methods with $\hat{J}_y(\hat{E}(\rho_{STLS_2}))$ is 3%.

Table 3 – Objective Function Estimate ($\hat{J}_y(\hat{E}(\rho)) \times 10^4$) regarding the simulation of Section 5.1.

$J_y(\hat{E}(\rho_{LS}))$	1.2319
$J_y(\hat{E}(\rho_{LASSO}))$	1.5205
$J_y(\hat{E}(\rho_{STLS}))$	1.2819
$J_y(\hat{E}(\rho_{STLS_2}))$	1.1338
$\hat{J}_y(\rho_0)$	1.0977

Analyzing the boxplots in Figure 10, it is possible to confirm that the ℓ_1-regularization methods – except STLS – decreased both the variance and the bias of the estimate, with the LASSO presenting the best results. If we draw the attention to the STLS method, it attained a worse variance compared to the LASSO.

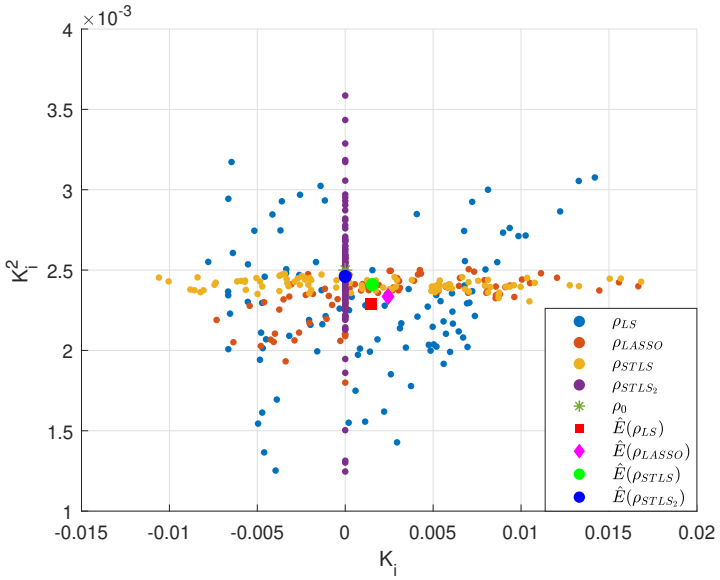
Figure 10 – Comparison of $J_y(\hat{\rho})$ for the classical and Regularized nonlinear VRFT regarding the simulation of Section 5.1.



Source: author.

In the Figure 11 it is possible to observe both the variance and the bias error on the $K_i \times K_i^2$ plan. It is interesting to notice that STLS bias is almost zero, cause its average value is practically on top of the ideal value ρ_0.

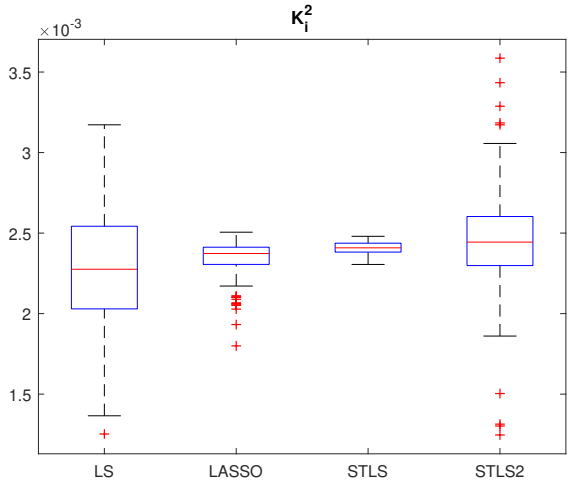
Figure 11 – Estimated controller's parameters on the $K_i \times K_i^2$ plan.



Source: author.

It is possible to visualize that in Figure 12 the STLS algorithm presents the minimum variance for the K_i^2 parameter, which corroborates with Figure 11 for the K_i^2 axis.

Figure 12 – K_i^2 boxplot comparison for the nonlinear VRFT and the Regularized nonlinear VRFT regarding the simulation of Section 5.1.



Source: author.

To illustrate the closed-loop performance, we show Figure 13. The chosen reference signal $r(t)$ is a square wave of 2 periods, with each period being of 20 seconds. These results were obtained with the system in closed-loop and simulating it for all 100 ρ vectors estimated which were gathered from each Monte Carlo simulation. The black line is the desired output time response $y_d(t)$. It is possible to observe that the STLS$_2$ closed-loop standouts between all the regularization methods and also comparing its performance to the LS closed-loop performance. It shall point out that simulations were handled without noise.

5.2 Hammerstein Process 2 - Matched Case

The second case study is also implemented with a Hammerstein System, where the linear part of the open-loop process is given by

$$G(q) = \frac{0.2}{q - 0.8}, \tag{94}$$

and the static nonlinearity is a $\sqrt{(.)}$ and $H(q) = 1$

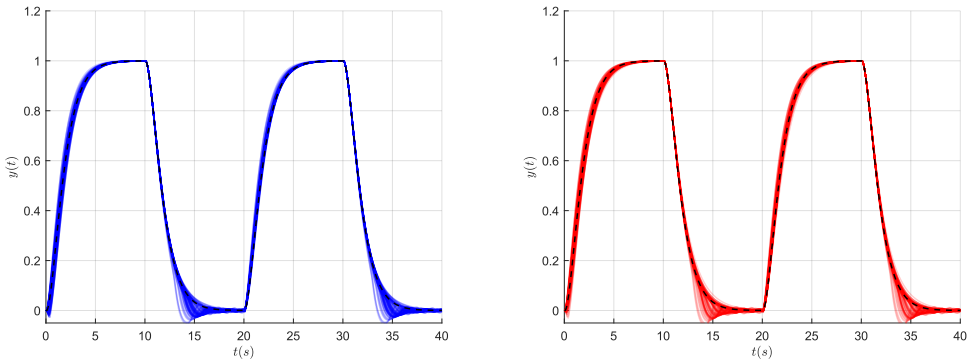
The desired closed-loop performance chosen for the system is given by the following transfer function

$$T_d(q) = \frac{0.3}{q - 0.7}. \tag{95}$$

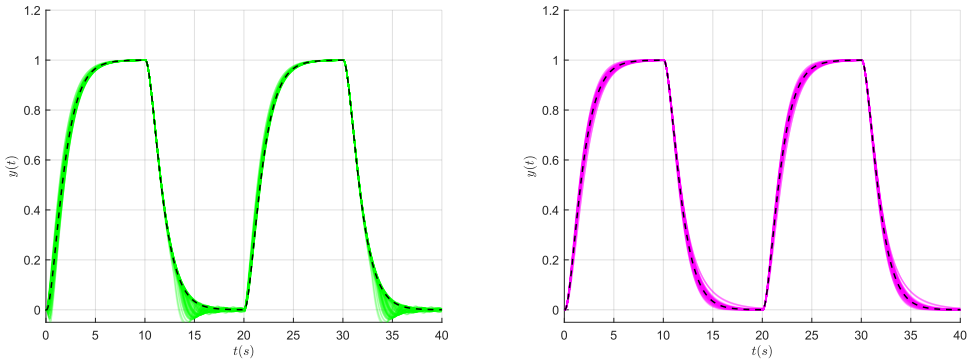
In the linear case, the ideal controller $C_d(q)$ would be the following

Figure 13 – Closed-loop performance regarding the simulation of the Section 5.1.

(a) Least Squares. (b) LASSO.



(c) STLS. (d) STLS$_2$.



Source: author.

$$C_d(q) = \begin{bmatrix} 1.5 & 0.3 \end{bmatrix} \begin{bmatrix} 1 & \frac{1}{q-1} \end{bmatrix}^T, \tag{96}$$

which is a Proportional-Integral (PI) controller. The controller class $\mathbb{C}$ chosen is PI controller as well. In such manner, the matching condition is met.

In the nonlinear case, it is quite straightforward that the ideal controller would be PI in addition to the inverse of the nonlinearity that is in the process, i.e. $f(.) = (.)^2$. The expansion of the linear signals $v_p(t)$ and $v_i(t)$ was made up to the third order, thus generating 15 regressors vectors:

$$\Phi = \begin{bmatrix} v_p(t) & v_p^2(t) & v_p^3(t) & v_i(t) & v_i^2(t) & v_i^3(t) & v_p(t)v_i(t) & \dots & v_p^3(t)v_i^3(t) \end{bmatrix}. \tag{97}$$

Thus the ideal controller would have the following parameters

$$\begin{aligned} \rho_0^T &= \begin{bmatrix} 0 & K_p^2 & 0 & 0 & K_i^2 & 0 & 2K_pK_i & \dots & 0 \end{bmatrix}^T \\ &= \begin{bmatrix} 0 & 2.25 & 0 & 0 & 0.09 & 0 & 0.9 & \dots & 0 \end{bmatrix}^T, \end{aligned} \tag{98}$$

so, the ideal controller would be

$$\mathcal{C}_d(v(t)) = \mathcal{C}(v(t),\rho) = \rho_0^T \Phi(v(t)), \tag{99}$$

with $v_p(t) = e(t)$ and $v_i(t) = \dfrac{1}{q-1}e(t)$.

The input signal $u(t)$ employed to excite the plant was a Pseudo Random Binary Signal (PRBS) multiplied by the absolute value of a Gaussian noise with zero mean and variance $\sigma^2 = 1$, with $N = 1500$ samples. Besides, the plant's output is affected by a gaussian noise with variance $\sigma_e^2 = 1 \times 10^{-4}$. Concerning the LASSO algorithm, the MATLAB function *lasso* was used. The regularization parameter λ_{LASSO} was calculated through the 10-fold Cross Validation algorithm so that it would yield minimum variance, and $\lambda_{STLS} = 0.05$ and $\lambda_{STLS_2} = 20$ (which corresponds to approximately to 1% of the contribution to the objective function).

To evaluate the proposed technique, 100 Monte Carlo simulations were run with distinct noise realizations. The major objective of inserting the regularization on the VRFT was to draw a better closed-loop performance. This evaluation was done through the objective function $J_y(\hat{\rho})$, in addition to the sum of all the estimated zeros in each Monte Carlo simulation.

Table 4 exhibits the average controller gains for each regressor.

Table 4 – Average Estimated Parameters regarding the simulation of Section 5.2.

Regressor	$\hat{E}(\rho_{LS})$	$\hat{E}(\rho_{LASSO})$	$\hat{E}(\rho_{STLS})$	$\hat{E}(\rho_{STLS_2})$	ρ_0
v_p	0.0260	0.0589	0.0327	0	0
v_p^2	1.9895	2.1496	2.0576	2.2177	2.2500
v_p^3	0.2412	0.0178	0.1455	0	0
v_i	0.0126	0.0094	0.0175	-0.0000	0
v_p^2	0.0818	0.0869	0.0858	0.0893	0.0900
v_p^3	0.0014	0.0003	0	0.0003	0
$v_p.v_i$	0.9595	0.8604	0.8769	0.9000	0.9000
$(v_p.v_i)^2$	-0.0489	0.0001	-0.0031	-0.0232	0
$(v_p.v_i)^3$	-0.0153	-0.0004	0.0019	-0.0002	0
$v_p^2.v_i$	0.2029	0.0117	0.0422	0.0472	0
$v_p^3.v_i$	-0.3356	0.0002	-0.0388	0	0
$v_p.v_i^2$	-0.0450	0.0012	0.0006	-0.0084	0
$v_p^3.v_i^2$	0.1272	-0.0003	-0.0084	-0.0014	0
$v_p.v_i^3$	0.0064	0.0003	0	0.0014	0
$v_p^2.v_i^3$	0.0023	-0.0005	0	0.0021	0

To evaluate the sparsity of the estimation along all the Monte Carlo Simulations, we present Table 5, which contains the number of estimated zeros by the four methods and the ideal quantity as well.

Table 5 – Total number of zeros regarding the simulation of Section 5.2.

Method	N_0
LS	0
LASSO	661
STLS	640
STLS_2	625
Ideal	1200

The ideal number of zeros is calculated as if in all simulations the undesirable parameters (12 in this example) would be zero.

If now we turn to the interpretation of the objective function, through Table 6 it is evident that all the regularization methods surpass the classical nonlinear VRFT with the Least-Squares. The cost $\hat{J}_y(\hat{E}(\rho_{LS}))$ is 25% worst than the minimum, while the LASSO achieves the minimum up to three correct significant digits.

Table 6 – Objective Function Estimate ($\hat{J}_y(\hat{E}(\rho)) \times 10^4$) regarding the simulation of Section 5.2.

$\hat{J}_y(\hat{E}(\rho_{LS}))$	1.586
$\hat{J}_y(\hat{E}(\rho_{LASSO}))$	1.268
$\hat{J}_y(\hat{E}(\rho_{STLS}))$	1.325
$\hat{J}_y(\hat{E}(\rho_{STLS_2}))$	1.286
$\hat{J}_y(\rho_0)$	1.265

Analyzing the boxplots in Figure 14, one notes that the STLS method presents several outliers. Then, to get a better of Figure 14, the following Figure 15 presents the zooming of the previous.

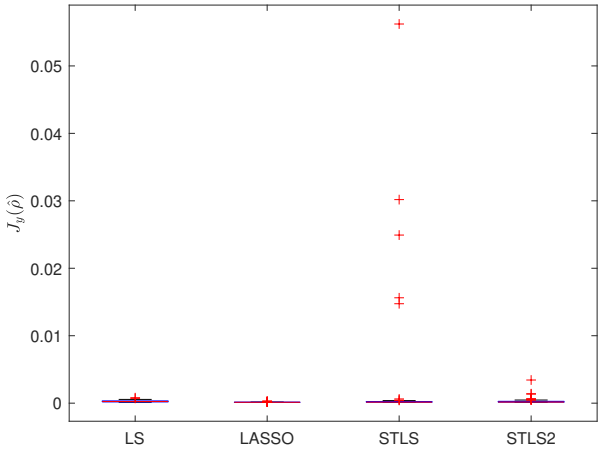
It is possible to confirm that the ℓ_1-regularization methods decreased both the variance and the bias of the estimate, with the LASSO presenting the best results. If we draw the attention to the STLS and STLS_2 methods, they attained a worse variance compared to the LASSO. Moreover, the STLS_2 returns a variance that is worse than the LS method.

In the Figure 16 it is possible to observe both the variance and the bias error on the $K_p^2 \times K_i^2$ plan. The black dot is the ideal parameter ρ_0, the STLS_2 and the LASSO methods provide the best result with respect to bias.

Another perspective for the Figure 16 is given by Figures 17 and 18. These Figures portray a good variance minimization by the LASSO algorithm and the STLS, and as requested the bias was also minimized. From Figure 18 one notices that the STLS method thresholded the K_i^2 parameters, which is not desirable.

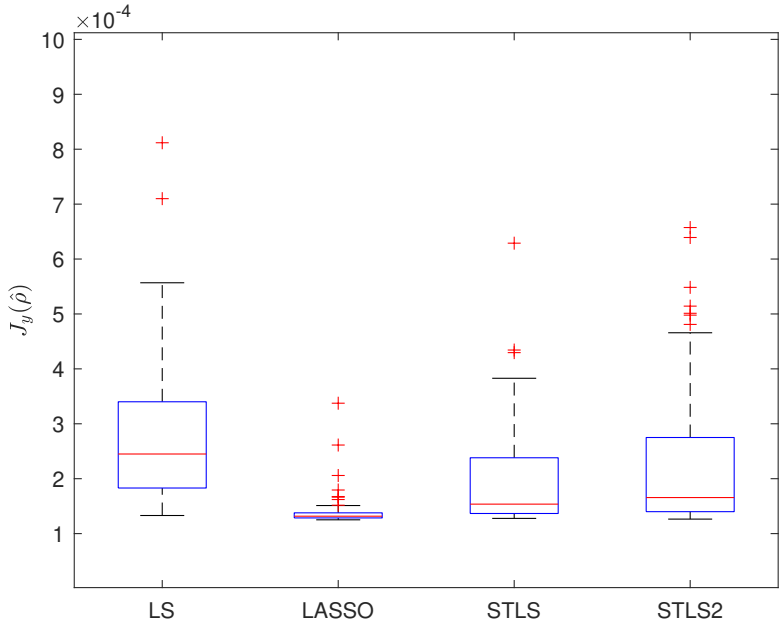
To illustrate the closed-loop performance, we show Figure 19, as with the last example, simulations were handled without noise. The chosen reference signal $r(t)$ is a square

Figure 14 – Comparison of $J_y(\hat{\rho})$ for the classical and Regularized VRFT regarding the simulation of the Section 5.2.



Source: author.

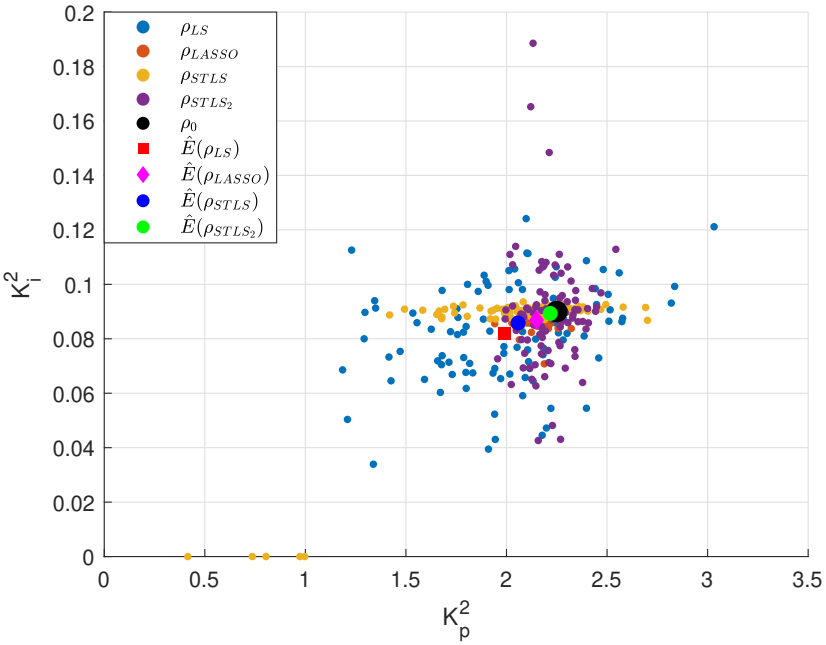
Figure 15 – Comparison of $J_y(\hat{\rho})$ for the classical and Regularized Nonlinear VRFT regarding the simulation of the Section 5.2.



Source: author.

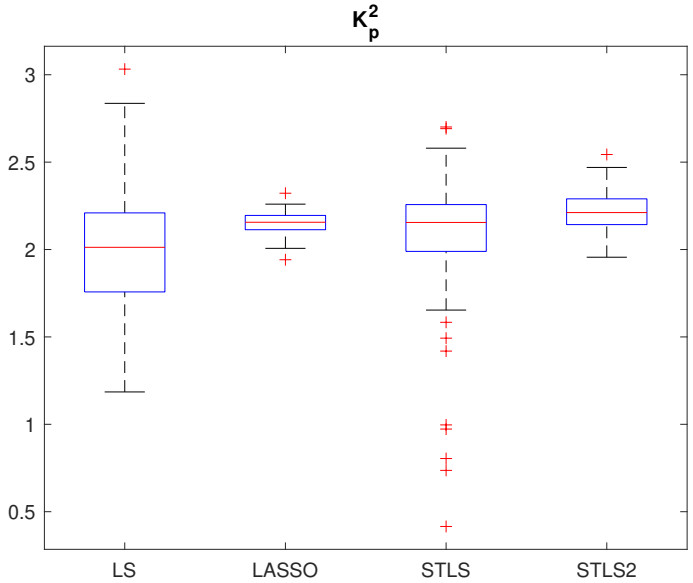
wave of 2 periods, with each period being of 20 seconds. These results were obtained with the system in closed-loop and simulating it for all 100 ρ vectors estimated which were gathered from each Monte Carlo simulation. The black line is the desired output time response $y_d(t)$. It is possible to observe that the LASSO closed-loop outperforms between all the regularization methods, which confirms the results of Figure 15. The outliers presented for the STLS method and observed in Figure 14 deteriorate the closed-loop

Figure 16 – Estimated controller's parameters on the $K_p^2 \times K_i^2$ plan regarding the simulation of the Section 5.2.



Source: author.

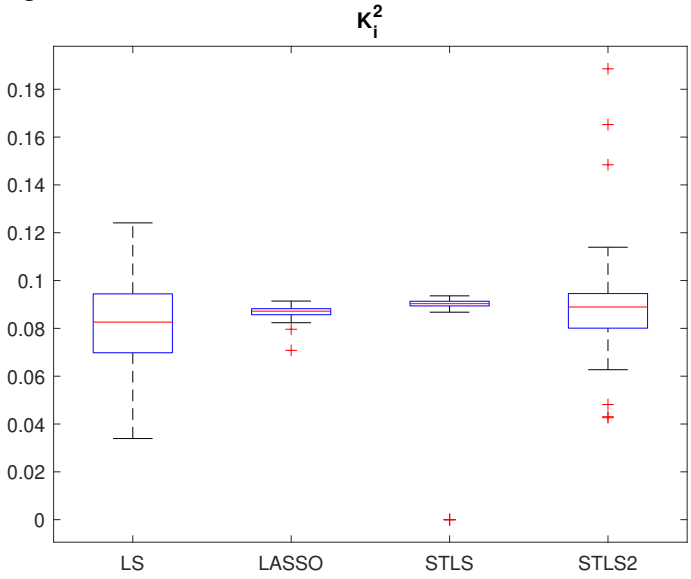
Figure 17 – K_p^2 boxplot comparison for the nonlinear VRFT and the Regularized nonlinear VRFT regarding the simulation of the Section 5.2.



Source: author.

performance as we see on Figure 19c.

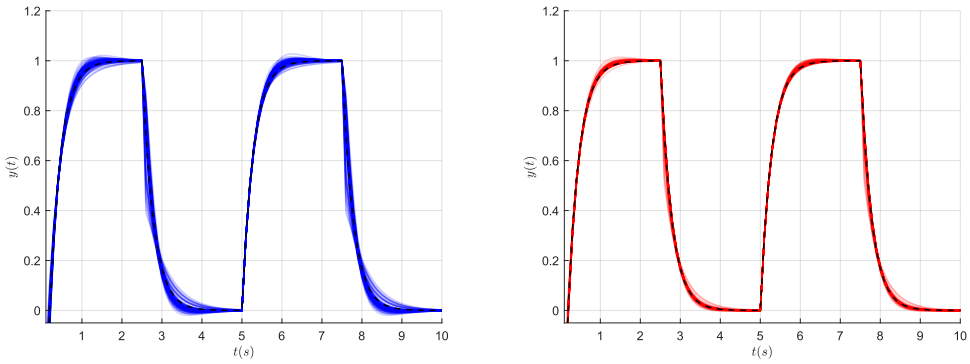
Figure 18 – K_i^2 boxplot comparison for the nonlinear VRFT and the Regularized nonlinear VRFT regarding the simulation of the Section 5.2.



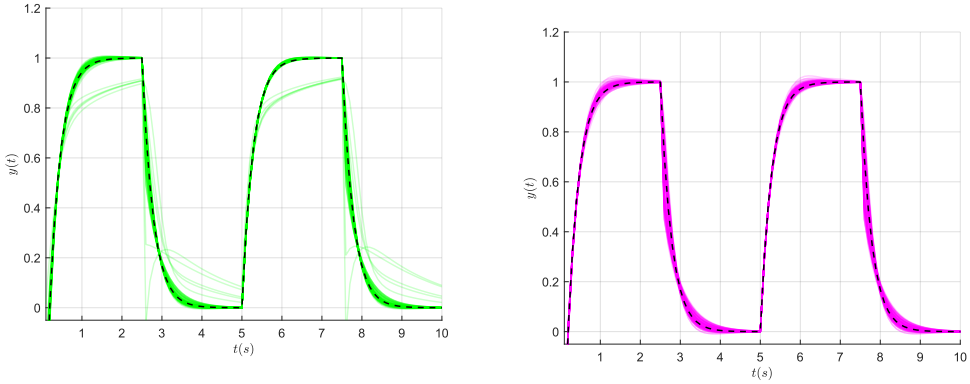
Source: author.

Figure 19 – Closed-loop performance regarding the simulation of the Section 5.2.

(a) Least Squares. (b) LASSO.



(c) STLS. (d) STLS$_2$.



Source: author.

5.3 Hammerstein Process 3

This next example is a variation from the study case of Section 5.2, where we have a pair pole/zero that have approximately the same module. We have simulated 100 Monte Carlo runs modifying the noise realization. The linear part of the process $G(q)$ is given by the following transfer function

$$G(q) = \frac{0.2(q-0.4)}{(q-0.3)(q-0.8)}. \tag{100}$$

and the static nonlinearity is a $\sqrt{(.)}$ and $H(q) = 1$. The reference model is the same as in (95).

We have adopted the same linear portion for the $\bar{C}(q)$ – a PI controller – and the library of nonlinear functions is also the same.

If there is no nonlinearity, the ideal linear controller $C_d(q)$ would be described as follows

$$C_d(q) = \frac{1.5(q-0.3)(q-0.8)}{(q-0.4)(q-1)}, \tag{101}$$

which is a PID controller with a fixed derivative pole.

To evaluate the sparsity of the estimation along all the Monte Carlo Simulations, we present Table 7, which contains the number of estimated zeros by the four methods.

Table 7 – Total number of zeros regarding the simulation of the Section 5.3.

Method	N_0
LS	0
LASSO	545
STLS	880
STLS_2	421

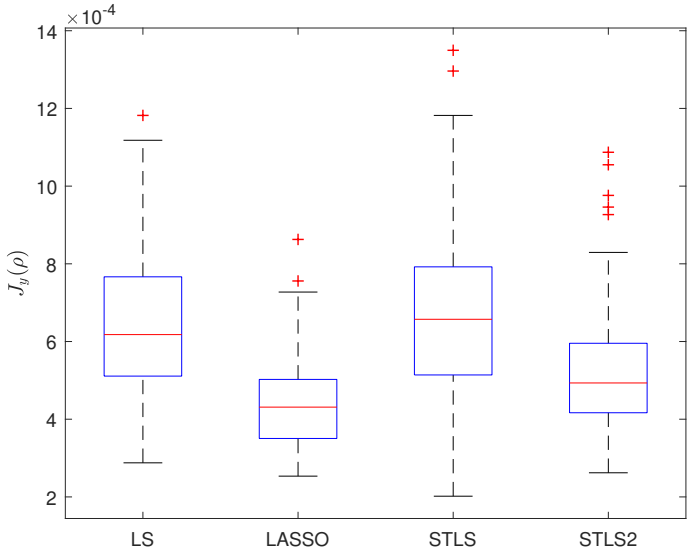
If now we turn to the interpretation of the objective function, through Table 8 it is evident that all the regularization methods outrun the classical VRFT with the Least-Squares.

Table 8 – Objective Function Estimate ($\hat{J}_y(\hat{E}(\rho)) \times 10^4$) regarding the simulation of the Section 5.3.

$J_y(\hat{E}(\rho_{LS}))$	5.5922
$J_y(\hat{E}(\rho_{LASSO}))$	4.2216
$J_y(\hat{E}(\rho_{STLS}))$	5.2552
$J_y(\hat{E}(\rho_{STLS_2}))$	4.3142

To illustrate the closed-loop performance, we show Figure 21, as with the last example, simulations were handled without noise. The chosen reference signal $r(t)$ is a square

Figure 20 – Comparison of $J_y(\hat{\rho})$ for the classical and Regularized Nonlinear VRFT regarding the simulation of the Section 5.3.



Source: author.

wave of 2 periods, with each period being of 5 seconds. These results were obtained with the system in closed-loop and simulating it for all 100 ρ vectors estimated which were gathered from each Monte Carlo simulation. The black line is the desired output time response $y_d(t)$. It is possible to observe that the LASSO closed-loop outperforms between all the regularization methods, which confirms the results of Figure 20.

5.4 Continuous Stirred-Tank Reactor

This subsection presents another case study: the *Continuous Stirred-Tank Reactor* (CSTR) whose model is given by (ROFFEL; BETLEM, 2007)

$$\mathcal{P}: \begin{cases} \dot{x_1} = -2x_1^2 + (1 - x_1)u \\ \dot{x_2} = x_1^2 - x_2 u, \\ y = x_2 \end{cases} \tag{102}$$

The desired closed-loop performance chosen for the system is given by the transfer function

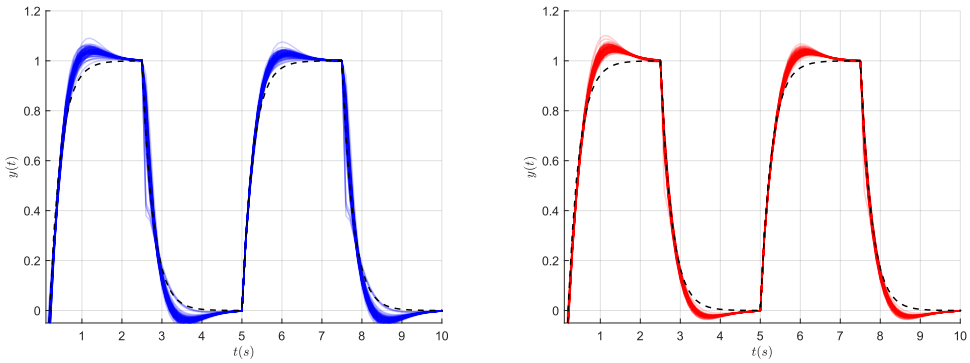
$$T_d(q) = \frac{0.0216(q + 0.8)}{(q - 0.85)^2}. \tag{103}$$

The selected controller's linear portion structure is a Proportional-Integral-Derivative (PID)

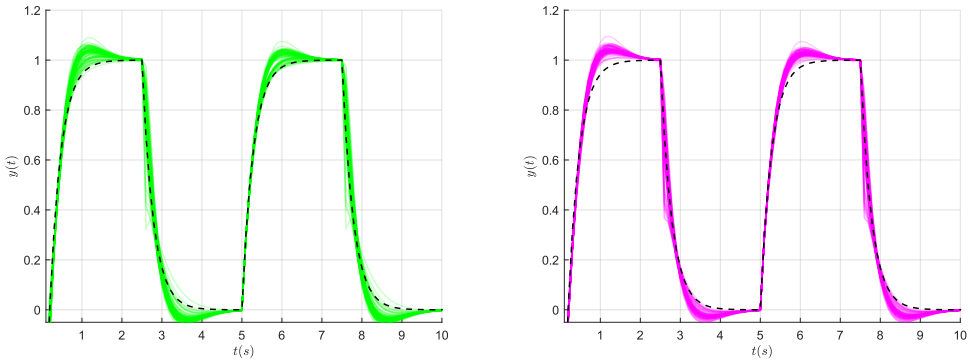
$$\bar{C}(q) = \begin{bmatrix} 1 & \frac{1}{q-1} & \frac{q-1}{q} \end{bmatrix}, \tag{104}$$

Figure 21 – Closed-loop performance regarding the simulation of the Section 5.3.

(a) Least Squares. (b) LASSO.



(c) STLS. (d) STLS$_2$.



Source: author.

The expansion of the linear signals $v_p(t)$, $v_i(t)$ and $v_d(t) = e(t)\dfrac{q-1}{q}$ was made up to the second order, generating 23 parameters to be estimated, thus $C_d(q) \notin \mathcal{C}$.

The input signal $u(t)$ applied to the plant is a sequence of steps to yield an output $y(t)$ in the range from 0.1 to 0.4. Furthermore, the plant's output is affected by a Gaussian noise with variance $\sigma^2 = 2.5 \times 10^{-7}$. As with the first example, we ran 100 Monte Carlo Simulations. The regularization parameter λ_{LASSO} is calculated as previously, the λ_{STLS} is the average of ρ_{LS} for each noise realization and λ_{STLS_2} is the average of contribution to the objective function for each noise realization as well.

Table 9 exhibits the average controller gains for each regressor.

Examining the Table 10, we can observe that the LASSO method presented the best sparse identification, i.e. the majority number of total zeros. After, the proposed method STLS$_2$ presents 1499 identified zeros which is near to the LASSO.

Table 9 – Average Estimated Parameters regarding the simulation of the Section 5.4.

Regressor	$\hat{E}(\rho_{LS})$	$\hat{E}(\rho_{LASSO})$	$\hat{E}(\rho_{STLS})$	$\hat{E}(\rho_{STLS_2})$
v_p	0.0686	-0.0433	-0.0151	0
v_p^2	0.0097	-0.0002	-0.0013	0
v_i	-1.1167	-0.0000	0.1496	0
v_i^2	-0.2659	-0.0317	0.3725	0
v_d	-0.6605	-0.0488	-0.1807	0
v_d^2	-3.5788	-0.0001	-1.1643	-0.0650
$v_i v_p$	-3.8453	0.0000	-0.2961	-0.0093
$v_d v_p$	3.0701	0	0.8561	-0.0918
$v_i v_d$	-1.3057	0	-0.1502	0
$(v_i v_p)^2$	-1.4049	0.0192	0.0086	-0.1204
$(v_d v_p)^2$	-0.0016	-0.0026	0.0157	0
$(v_i v_d)^2$	-1.2686	0.0086	-0.0223	0.1214
$v_i v_p^2$	-3.8657	-0.0000	-0.8684	-0.3348
$v_d v_p^2$	1.6587	0.0000	0.6920	0.1914
$v_i v_d^2$	-4.9033	0	-0.4240	0.3188
$v_i^2 v_p$	-0.7669	0	0.4378	-0.0193
$v_d^2 v_p$	-1.6023	0.0011	-0.6335	-0.2575
$v_i^2 v_p$	-0.7669	0	-0.0129	-0.0551
$v_p v_i v_d$	3.8098	0	0.1833	-0.4403
$(v_p v_i v_d)^2$	-0.0473	0.0003	0.0010	-0.0047
$v_p^2 v_i v_d$	0.0523	-0.0000	-0.1240	-0.0446
$v_p v_i^2 v_d$	1.1100	0.0005	-0.0021	-0.1881
$v_p v_i v_d^2$	-1.4462	-0.0000	-0.1240	-0.1177
$v_p^2 v_i^2 v_d$	-0.1787	-0.0006	-0.0219	-0.0613
$v_p^2 v_i v_d^2$	-0.1291	0.0000	-0.0013	-0.0029
$v_p v_i^2 v_d^2$	-0.3134	-0.0001	-0.0108	-0.0164

Table 10 – Total number of zeros

Method	N_0
LS	0
LASSO	1531
STLS	944
$STLS_2$	1499

If now we turn to the interpretation of the objective function, through Table 11 it is evident that all the regularization methods surmount the classical VRFT with the Least-

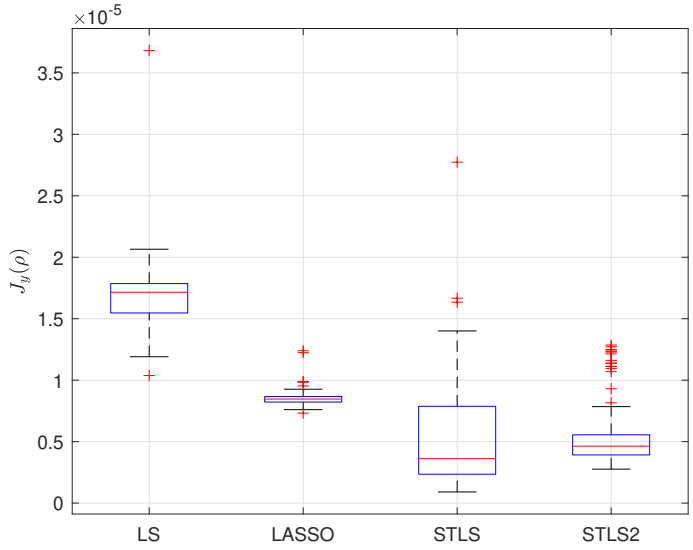
Squares.

Table 11 – Objective Function Estimate ($\hat{J}_y(\hat{E}(\rho)) \times 10^6$) regarding the simulation of the Section 5.4.

$\hat{J}_y(\hat{E}(\rho_{LS}))$	1.625
$\hat{J}_y(\hat{E}(\rho_{LASSO}))$	0.905
$\hat{J}_y(\hat{E}(\rho_{STLS}))$	0.305
$\hat{J}_y(\hat{E}(\rho_{STLS_2}))$	0.422

Analyzing the boxplots in Figure 22, it is possible to confirm that the ℓ_1-regularization methods decreased the variance, with the LASSO presenting the best results. If we draw the attention to the STLS and STLS$_2$ methods, they attained an excellent minimum, with the STLS$_2$ overcoming the STLS in terms of variance.

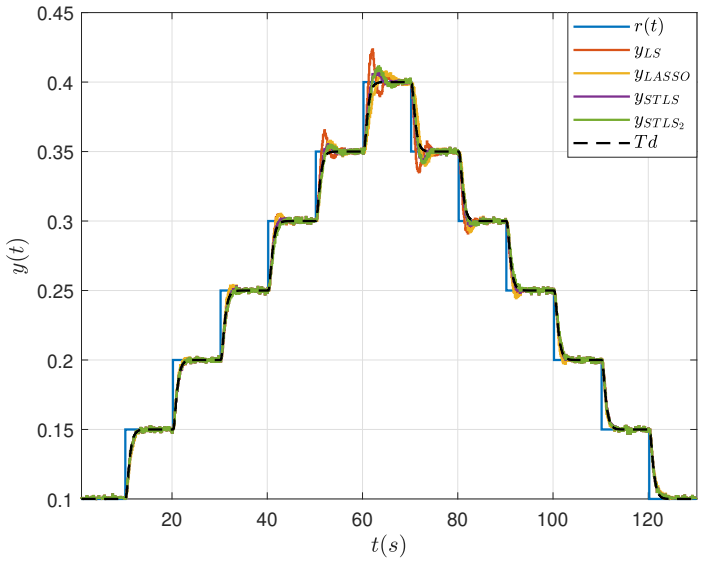
Figure 22 – Comparison of $J_y(\hat{\rho})$ for the classical and Regularized Nonlinear VRFT regarding the simulation of the Section 5.4.



Source: author.

To illustrate the closed-loop performance in the designated range, we show Figure 23. It can be seen that the regularization methods performed better than the Least-Squares. When one compares the regularization methods closed-loop performance, one may see that in average the STLS and STLS$_2$ performed approximately equal to each other for this case study.

Figure 23 – CSTR closed-loop performance regarding the simulation of the Section 5.4.



Source: author.

5.5 Analysis concerning the Input Signal and the Noise Level

The following subsection depicts an analysis regarding the effects on the choice of the input signal $u(t)$ – sometimes entitled the learning data – the noise level in the closed-loop system and the sparsity for the controller identification.

This analysis was done by decreasing the white noise level by the following:

- $\sigma = 10^{-2}$
- $\sigma = 5 \times 10^{-3}$
- $\sigma = 10^{-3}$.

Besides, the learning data was varied as follows

- APRBS
- White Noise
- Steps.

An important aspect to highlight is that the White Noise is decorrelated to the White Noise in the process' output. Also, the Steps used as the training data are steps that are only modulated in amplitude.

Tables 12, 14, 16 and 18 show the values for the cost function estimate with the average controller parameters of each example.

Tables 13, 15, 17 and 19 show the total number of zeros controller parameters of each example.

Table 12 – Cost function values regarding the simulation of the Section 5.1.

$J \times 10^4$	APRBS			Ruído Branco			Steps		
	$\sigma = 10^{-2}$	$\sigma = 5 \times 10^{-3}$	$\sigma = 10^{-3}$	$\sigma = 10^{-2}$	$\sigma = 5 \times 10^{-3}$	$\sigma = 10^{-3}$	$\sigma = 10^{-2}$	$\sigma = 5 \times 10^{-3}$	$\sigma = 10^{-3}$
$J_y(\hat{E}(\rho_{LS}))$	1.2319	1.1327	1.1001	1.5914	1.1227	1.0961	24.9116	16.4143	1.1808
$J_y(\hat{E}(\rho_{LASSO}))$	1.5205	1.1640	1.2472	1.4098	1.1432	1.2389	1.5990	1.0936	1.2417
$J_y(\hat{E}(\rho_{STLS}))$	1.2820	1.0968	1.0974	4.4795	1.1476	1.0971	1.5421	1.0917	1.0937
$J_y(\hat{E}(\rho_{STLS_2}))$	1.1338	1.1050	1.0974	1.4028	1.1164	1.0975	1.3510	1.1301	1.0975
$J_y(\rho_0)$	1.0977	1.0977	1.0977	1.0977	1.0977	1.0977	1.0977	1.0977	1.0977

Table 13 – Total number of zeros for each noise level and input signal regarding the simulation of the Section 5.1.

N_0	APRBS			White Noise			Steps		
	$\sigma = 10^{-2}$	$\sigma = 5 \times 10^{-3}$	$\sigma = 10^{-3}$	$\sigma = 10^{-2}$	$\sigma = 5 \times 10^{-3}$	$\sigma = 10^{-3}$	$\sigma = 10^{-2}$	$\sigma = 5 \times 10^{-3}$	$\sigma = 10^{-3}$
LS	0	0	0	0	0	0	0	0	0
LASSO	209	196	389	212	209	398	129	190	400
STLS	307	315	368	315	323	388	305	306	359
$STLS_2$	351	371	400	329	375	400	201	219	400
Ideal	400	400	400	400	400	400	400	400	400

Table 14 – Cost function values regarding the simulation of the Section 5.2.

$J \times 10^4$	APRBS			White Noise			Steps		
	$\sigma = 10^{-2}$	$\sigma = 5 \times 10^{-3}$	$\sigma = 10^{-3}$	$\sigma = 10^{-2}$	$\sigma = 5 \times 10^{-3}$	$\sigma = 10^{-3}$	$\sigma = 10^{-2}$	$\sigma = 5 \times 10^{-3}$	$\sigma = 10^{-3}$
$J_y(\hat{E}(\rho_{LS}))$	1.665	1.317	1.268	1.485	1.291	1.267	173.8	-	70.06
$J_y(\hat{E}(\rho_{LASSO}))$	1.272	1.264	1.375	1.281	1.265	1.333	114.5	212.0	52.31
$J_y(\hat{E}(\rho_{STLS}))$	1.323	1.267	1.265	1.312	1.269	1.265	165.7	-	60.95
$J_y(\hat{E}(\rho_{STLS_2}))$	1.305	1.274	1.265	1.297	1.266	1.265	-	242.9	357.8
$J_y(\rho_0)$	1.265	1.265	1.265	1.265	1.265	1.265	1.265	1.26	1.265

Table 15 – Total number of zeros for each noise level and input signal regarding the simulation of the Section 5.2.

N_0	APRBS			White Noise			Steps		
	$\sigma = 10^{-2}$	$\sigma = 5 \times 10^{-3}$	$\sigma = 10^{-3}$	$\sigma = 10^{-2}$	$\sigma = 5 \times 10^{-3}$	$\sigma = 10^{-3}$	$\sigma = 10^{-2}$	$\sigma = 5 \times 10^{-3}$	$\sigma = 10^{-3}$
LS	0	0	0	0	0	0	0	0	0
LASSO	649	613	900	671	634	900	759	466	1176
STLS	639	778	1120	597	662	967	439	393	561
$STLS_2$	615	846	1179	434	566	938	955	1086	1400
Ideal	1200	1200	1200	1200	1200	1200	1200	1200	1200

Table 16 – Cost function values regarding the simulation of the Section 5.3.

$J \times 10^4$	APRBS			White Noise			Steps		
	$\sigma = 10^{-2}$	$\sigma = 5 \times 10^{-3}$	$\sigma = 10^{-3}$	$\sigma = 10^{-2}$	$\sigma = 5 \times 10^{-3}$	$\sigma = 10^{-3}$	$\sigma = 10^{-2}$	$\sigma = 5 \times 10^{-3}$	$\sigma = 10^{-3}$
$J_y(\hat{E}(\rho_{LS}))$	5.592	4.766	4.516	3.734	3.469	3.393	-	-	-
$J_y(\hat{E}(\rho_{LASSO}))$	4.221	3.768	3.643	3.061	2.979	2.962	123.0	-	45.94
$J_y(\hat{E}(\rho_{STLS}))$	5.255	4.585	4.226	3.486	3.239	3.100	-	-	-
$J_y(\hat{E}(\rho_{STLS_2}))$	4.314	4.118	4.214	2.912	3.533	3.618	474.3	139.3	-

Due to the CSTR characteristics, the analysis was done by decreasing the white noise level by the following standard deviation:

- $\sigma = 10^{-3}$
- $\sigma = 5 \times 10^{-4}$

Table 17 – Total number of zeros for each noise level and input signal regarding the simulation of the Section 5.3.

N_0	APRBS			White Noise			Steps		
	$\sigma = 10^{-2}$	$\sigma = 5 \times 10^{-3}$	$\sigma = 10^{-3}$	$\sigma = 10^{-2}$	$\sigma = 5 \times 10^{-3}$	$\sigma = 10^{-3}$	$\sigma = 10^{-2}$	$\sigma = 5 \times 10^{-3}$	$\sigma = 10^{-3}$
LS	0	0	0	0	0	0	0	0	0
LASSO	545	614	652	589	519	455	891	427	1179
STLS	88	108	151	160	179	126	246	268	369
$STLS_2$	421	470	500	283	347	371	914	994	1275

- $\sigma = 10^{-4}$.

Table 18 – Cost function values regarding the simulation of the Section 5.4.

$J \times 10^6$	APRBS			White Noise			Steps		
	$\sigma = 10^{-3}$	$\sigma = 5 \times 10^{-4}$	$\sigma = 10^{-4}$	$\sigma = 10^{-3}$	$\sigma = 5 \times 10^{-4}$	$\sigma = 10^{-4}$	$\sigma = 10^{-3}$	$\sigma = 5 \times 10^{-4}$	$\sigma = 10^{-4}$
$J_y(\hat{E}(\rho_{LS}))$	16.87	16.9735	16.32	19.27	20.69	19.71	12.76	5.025	141.5
$J_y(\hat{E}(\rho_{LASSO}))$	17.93	17.3880	17.70	18.03	17.53	16.85	14.59	8.495	8.083
$J_y(\hat{E}(\rho_{STLS}))$	16.49	17.2211	18.50	22.11	26.50	28.61	15.03	5.516	2.587
$J_y(\hat{E}(\rho_{STLS_2}))$	19.27	42.5250	154.88	90.26	16.07	16.48	13.54	7.029	4.622

Table 19 – Total number of zeros for each noise level and input signal regarding the simulation of the Section 5.4.

N_0	APRBS			White Noise			Steps		
	$\sigma = 10^{-3}$	$\sigma = 5 \times 10^{-4}$	$\sigma = 10^{-4}$	$\sigma = 10^{-3}$	$\sigma = 5 \times 10^{-4}$	$\sigma = 10^{-4}$	$\sigma = 10^{-3}$	$\sigma = 5 \times 10^{-4}$	$\sigma = 10^{-4}$
LS	0	0	0	0	0	0	0	0	0
LASSO	24	17	2	118	80	83	298	870	1553
STLS	1051	1174	1191	1058	1024	977	725	715	957
$STLS_2$	1785	1870	1900	1874	1726	1700	1570	1724	1479

From Tables 14 and 15 we observe interesting results with the APRBS and White Noise as training data. In some cases the minimum of the reference tracking performance criterion is achieved, as well as the ideal quantity of parameters are thresholded. The Steps applied into the plant to estimate the controller generated unstable closed-loop systems for the study cases of Section 5.2 and this outcome is accentuated in example of Section 5.3 (see Tables 16 and 17).

As for the example of Section 5.4, we see on Tables 18 and 19 that the LASSO method was the worst in the sparsity aspect. When we pay attention to both the sparsity and the cost function value, we perceive that the best results were attained with the sequence of steps being the input signal.

Finally, the analysis of the input signal and the noise level shows that, for the studied examples, the input signal is a very important aspect to consider in the design of nonlinear controller using the Nonlinear VRFT method.

5.6 Chapter conclusions

In this Chapter we showed the simulation results for four different case studies with the objective of analyzing the regularization methods compared to the Nonlinear VRFT method. Four case studies are analyzed: two of them consider the matched case, and the other two the unmatched case.

In the first case study of Section 5.1 the method that is proposed in this work (STLS_2) outperforms the other two regularization methods as well as the classical Nonlinear VRFT with the LS solution, with respect to the closed-loop performance and the sparse solution. When we look only at the variance for the ideal parameter K_i^2, LASSO and STLS yield a minimum variance for this parameter.

The second case study (Section 5.2) the LASSO is the best regarding both the sparsity and also the closed-loop performance. Besides, the STLS method yields a considerable amount of outliers compared to the other methods, as a consequence it affects the closed-loop performance.

In Section 5.3, a variation of the second case study is presented, where the ideal controller cannot be match. As one analyzes the sparsity, the best method is the STLS. However, for the closed-loop performance the LASSO provides better results.

Section 5.4 portrays a representation of the CSTR. In this case, all regularization methods outperformed the classical Nonlinear VRFT. The LASSO performed better considering the variance for the performance criterion $J_y(\rho)$, while the STLS and the STLS_2 achieved the smaller values for the same criterion.

Finally, Section 5.5 presented a more complete analysis for each regularization method concerning the input signal and the noise leve l consequence on the closed-loop performance and the sparsity of the tuned controller.

6 CONCLUSIONS

In this work we addressed the Design of Nonlinear Controllers problem with the focus on solving this problem using the ℓ_1 regularization techniques, where two existing methods of the literature are presented and a third is derived. The idea of including this regularization technique comes from the controller structure and the Nonlinear VRFT statistical properties.

In this sense, before presenting the regularization techniques to tune the controller parameters, some preliminary definitions, which are fundamental to the development of the work, are introduced to the reader on Chapter 2. Next, Chapter 3 the standard Nonlinear VRFT formulation with Least Squares algorithm is discussed in order to contextualize the reader in this data-driven control method. Also, in this Chapter, the statistical properties of this formulation are demonstrated. Through this analysis it was possible to conclude that in the presence of a high amount of noise, the Nonlinear VRFT with the Least Squares displays a considerable bias error, and it deteriorates the closed-loop performance. Another aspect that was noticed is the inefficiency of the Least Squares method regarding the sparsity for the tuned controller. On Chapter 4 we introduced the regularization concepts comparing it to the Least Squares algorithm. The two kinds of regularization techniques were compared in this Chapter: the ℓ_2 (known as Ridge Regression) and the ℓ_1 (known as LASSO Regression). In addition, this Chapter discussed two other regularization methods, which can be interpreted as the primal solution for the LASSO Regression: the SLTS proposed by (BRUNTON; PROCTOR; KUTZ, 2016) and the denominated SLTS_2 proposed in this work. On Chapter 5 it was presented four case studies, where in the first two we have the matched case, and in the last two the unmatched case. This Chapter illustrates the main results of this master thesis.

We exposed on Chapter 5 the feature behind each regularization method compared to the Least Squares. From these case studies it was noticed that the LASSO usually displays the best results regarding the variance aspect, as it employs the k-fold Cross-Validation procedure to find the best penalty parameter. On the other hand, the STLS and STLS_2 got better results then the LASSO only in the last example, with respect to the average closed-loop performance. Furthermore, this Chapter compared three different noise levels with

three different input signals for all case studies, with the objective of examining better their consequences on the performance criterion and the controllers order.

Based on the four case studies we could not designate the best regularization technique between all three, since each of them presents advantages and disadvantages depending on the case study. Besides, as the STLS and STLS_2 are only the primal formulation for the LASSO, a way of finding the parameters λ_{STLS} and λ_{SLTS_2} must be derived.

Finally, as for future works, it remains the development of a detailed formulation for the best solution λ_{STLS} and λ_{SLTS_2} and comparing it with all cross-validation techniques. Another work could be done on how to optimize the best order for the expansion of the library of nonlinear functions $\phi(.)$. The sparsity of the nonlinear controller could also be studied deeper, as well as different parameterizations for the controller could be compared, for instance.

REFERENCES

ÅSTRÖM, K. J.; HÄGGLUND, T. **Advanced PID Control**. North Carolina, USA: ISA - The Instrumentation, Systems, and Automation Society, 2006.

ÅSTRÖM, K. J.; WITTENMARK, B. **Adaptive control**. [*S.l.*]: Courier Corporation, 2013.

BAZANELLA, A. S.; CAMPESTRINI, L.; ECKHARD, D. **Data-driven Controller Design**: the H_2 approach. Netherlands: Springer, 2012.

BAZANELLA, A. S.; NEUHAUS, T. Tuning nonlinear controllers with the virtual reference approach. **IFAC Proceedings Volumes**, [*S.l.*], v. 47, n. 3, p. 10269–10274, 2014.

BOEIRA, E. C. **Sintonia De Controladores Multivariáveis Pelo Método Da Referência Virtual Com Regularização Bayesiana**. 2018. Master Thesis — Universidade Federal do Rio Grande do Sul (UFRGS), Porto Alegre, Brasil, 2018.

BRUNTON, S. L.; KUTZ, J. N. **Data-driven science and engineering**: machine learning, dynamical systems, and control. [*S.l.*]: Cambridge University Press, 2019.

BRUNTON, S. L.; PROCTOR, J. L.; KUTZ, J. N. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. **Proceedings of the national academy of sciences**, [*S.l.*], v. 113, n. 15, p. 3932–3937, 2016.

BRUYNE, F. D. Iterative Feedback Tuning for MIMO Systems. *In*: INTERNATIONAL SYMPOSIUM ON INTELLIGENT AUTOMATION AND CONTROL, 2., 1997, Anchorage, Alaska, USA. **Proceedings [...].** New York: IEEE, 1997.

BÜHLMANN, P.; VAN DE GEER, S. **Statistics for high-dimensional data**: methods, theory and applications. [*S.l.*]: Springer Science & Business Media, 2011.

CAMPESTRINI, L. *et al.* Virtual Reference Feedback Tuning for Non-Minimum Phase Plants. **Automatica**, Luxembourg, Austria, v. 47, n. 8, p. 1778–1784, 2011.

CAMPESTRINI, L. *et al.* Unbiased MIMO VRFT with application to process control. **Journal of Process Control**, Luxembourg, Austria, v. 39, p. 35–49, 2016.

CAMPESTRINI, L. *et al.* Data-driven Model Reference Control Design by Prediction Error Identification. **Journal of the Franklin Institute**, Luxembourg, Austria, v. 354, n. 6, p. 2628–2647, 2016.

CAMPI, M. C.; LECCHINI, A.; SAVARESI, S. M. Virtual Reference Feedback Tuning: A Direct Method for the Design of Feedback Controllers. **Automatica**, Luxembourg, Austria, v. 38, n. 8, p. 1337–1346, 2002.

CAMPI, M. C.; LECCHINI, A.; SAVARESI, S. M. An application of the virtual reference feedback tuning method to a benchmark problem. **European Journal of Control**, [*S.l.*], v. 9, n. 1, p. 66–76, 2003.

CAMPI, M. C.; SAVARESI, S. M. Direct Nonlinear Control Design: The Virtual Reference Feedback Tuning (VRFT) Approach. **IEEE Transactions on Automatic Control**, New York, Estados Unidos, v. 51, n. 1, p. 14–27, jan 2006.

CHEN, T.; LJUNG, L. Implementation of algorithms for tuning parameters in regularized least squares problems in system identification. **Automatica**, Luxembourg, Austria, v. 49, n. 7, p. 2213–2220, jul 2013.

CORLETA, A. *et al.* Data-driven control design applied to uninterruptible power supplies. *In*: IEEE CONFERENCE ON CONTROL APPLICATIONS (CCA), 2016., 2016. **Proceedings [...].** [*S.l.: s.n.*], 2016. p. 1312–1317.

EL-AWADY, K.; HANSSON, A.; WAHLBERG, B. Application of iterative feedback tuning to a thermal cycling module. **IFAC Proceedings Volumes**, [*S.l.*], v. 32, n. 2, p. 4664–4669, 1999.

FERIZBEGOVIC, M. *et al.* Bayes Control of Hammerstein Systems. **IFAC-PapersOnLine**, [*S.l.*], v. 54, n. 7, p. 755–760, 2021.

FORMENTIN, S.; KARIMI, A. Enhancing statistical performance of data-driven controller tuning via L2-regularization. **Automatica**, [*S.l.*], v. 50, n. 5, p. 1514–1520, 2014.

GARCIA, C. S.; BAZANELLA, A. S. Selection of informative intervals in routine operating data for use in data-driven control design. *In*: IEEE CONFERENCE ON CONTROL TECHNOLOGY AND APPLICATIONS (CCTA), 2020., 2020. **Proceedings [...].** [*S.l.: s.n.*], 2020. p. 225–230.

GOODWIN, G. C.; SIN, K. S. **Adaptive Filtering, Prediction and Control**. Upper Saddle River, New Jersey: Prentice Hall, 1984.

GRAHAM, A.; YOUNG, A.; XIE, S. Rapid tuning of controllers by IFT for profile cutting machines. **Mechatronics**, [*S.l.*], v. 17, n. 2-3, p. 121–128, 2007.

HJALMARSSON, H. Efficient tuning of linear multivariable controllers using iterative feedback tuning. **International journal of adaptive control and signal processing**, [*S.l.*], v. 13, n. 7, p. 553–572, 1999.

HJALMARSSON, H.; BIRKELAND, T. Iterative feedback tuning of linear time-invariant MIMO systems. *In*: IEEE CONFERENCE ON DECISION AND CONTROL, 1998, Tampa, FL. **Proceedings [...].** New York: IEEE, 1998. v. 4, p. 3893–3898.

HJALMARSSON, H.; GUNNARSSON, S.; GEVERS, M. A Convergent Iterative Restricted Complexity Control Design Scheme. *In*: IEEE CONFERENCE ON DECISION AND CONTROL, 33., 1994, Lake Buena Vista, FL. **Proceedings [...].** New York: IEEE, 1994. v. 2, p. 1735 – 1740.

HJALMARSSON, H. *et al.* Iterative Feedback Tuning: Theory and Applications. **IEEE Control Systems**, [*S.l.*], p. 26–41, 1998.

HOU, Z.-S.; WANG, Z. From model-based control to data-driven control: survey, classification and perspective. **Information Sciences**, [*S.l.*], v. 235, p. 3–35, 2013.

IOANNOU, P. A.; SUN, J. **Robust adaptive control**. Mineola, NY, USA: Courier Corporation, 2012.

JAMES, G. *et al.* **An introduction to statistical learning**. [*S.l.*]: Springer, 2013. v. 112.

KAMMER, L. C.; BITMEAD, R. R.; BARTLETT, P. L. Direct Iterative Tuning via Spectral Analysis. **Automatica**, Luxembourg, Austria, v. 36, p. 1301–1307, 2000.

KARIMI, A.; MIŠKOVIĆ, L.; BONVIN, D. Iterative correlation-based controller tuning with application to a magnetic suspension system. **Control Engineering Practice**, [*S.l.*], v. 11, n. 9, p. 1069–1078, 2003.

KARIMI, A.; MISKOVIC, L.; BONVIN, D. Iterative Correlation-based Controller Tuning. **International Journal of Adaptive Control and Signal Processing**, Hoboken, Estados Unidos, v. 18, p. 645–664, 2004.

KARIMI, A.; VAN HEUSDEN, K.; BONVIN, D. Non-iterative data-driven controller tuning using the correlation approach. *In*: EUROPEAN CONTROL CONFERENCE (ECC), 2007., 2007. **Proceedings [...].** [*S.l.: s.n.*], 2007. p. 5189–5195.

LECCHINI, A.; CAMPI, M. C.; SAVARESI, S. M. Virtual Reference Feedback Tuning for Two Degree of Freedom Controllers. **International Journal of Adaptive Control and Signal Processing**, Hoboken, Estados Unidos, v. 16, n. 5, p. 355–371, 2002.

LJUNG, L. System identification. **Wiley encyclopedia of electrical and electronics engineering**, [*S.l.*], p. 1–19, 1999.

MISKOVIC, L. *et al.* Correlation-based Tuning of Decoupling Multivariable Controllers. **Automatica**, Luxembourg, Austria, v. 43, n. 9, p. 1481–1494, 2007.

NOVARA, C.; FORMENTIN, S. Data-driven inversion-based control of nonlinear systems with guaranteed closed-loop stability. **IEEE Transactions on Automatic Control**, [*S.l.*], v. 63, n. 4, p. 1147–1154, 2017.

NOVARA, C.; MILANESE, M. Control of MIMO nonlinear systems: a data-driven model inversion approach. **Automatica**, [*S.l.*], v. 101, p. 417–430, 2019.

PILLONETTO, G.; CHIUSO, A.; DE NICOLAO, G. Prediction error identification of linear systems: a nonparametric gaussian regression approach. **Automatica**, [*S.l.*], v. 47, n. 2, p. 291–305, 2011.

PILLONETTO, G.; NICOLAO, G. D. A new kernel-based approach for linear system identification. **Automatica**, Luxembourg, Austria, v. 46, n. 1, p. 81–93, jan 2010.

PILLONETTO, G. *et al.* Kernel methods in system identification, machine learning and function estimation: a survey. **Automatica**, [*S.l.*], v. 50, n. 3, p. 657–682, mar 2014.

PREVIDI, F. *et al.* Data-Driven Control Design for Neuroprotheses: a virtual reference feedback tuning (VRFT) approach. **IEEE Transactions on Control Systems Technology**, New York, Estados Unidos, v. 12, n. 1, p. 176–182, jan 2004.

RALLO, G. *et al.* Virtual reference feedback tuning with Bayesian regularization. *In*: EUROPEAN CONTROL CONFERENCE (ECC), 2016., 2016. **Proceedings [...].** [*S.l.: s.n.*], 2016. p. 507–512.

ROFFEL, B.; BETLEM, B. **Process dynamics and control**: modeling for control and prediction. [*S.l.*]: John Wiley & Sons, 2007.

SINGH, R.; SZNAIER, M. On Identification of Nonlinear ARX Models with Sparsity in Regressors and Basis Functions. **IFAC-PapersOnLine**, [*S.l.*], v. 54, n. 7, 2021.

SÖDERSTRÖM, T. **Errors-in-variables methods in system identification**. [*S.l.*]: Springer, 2018.

SÖDERSTRÖM, T.; STOICA, P. **System identification**. [*S.l.*]: Prentice-Hall International, 1989.

TESCH, D. A. **Extensão do Iterative Feedback Tuning para Sistemas em Cascata com Aplicação em Controle de Quadricópteros**. 2016. Master Thesis — UFRGS, Porto Alegre, Brasil, 2016.

TESCH, D. A.; ECKHARD, D.; GUARIENTI, W. C. Pitch and roll control of a quadcopter using cascade iterative feedback tuning. **IFAC-PapersOnLine**, [*S.l.*], v. 49, n. 30, p. 30–35, 2016.

TIBSHIRANI, R. Regression shrinkage and selection via the lasso. **Journal of the Royal Statistical Society: Series B (Methodological)**, [*S.l.*], v. 58, n. 1, p. 267–288, 1996.

VAN HEUSDEN, K.; KARIMI, A.; BONVIN, D. Data-driven model reference control with asymptotically guaranteed stability. **International Journal of Adaptive Control and Signal Processing**, [*S.l.*], v. 25, n. 4, p. 331–351, 2011.

ZIEGLER, J. G.; NICHOLS, N. B. Optimum settings for automatic controllers. **Transactions of the ASME**, New York, USA, v. 64, p. 759 – 768, 1942.