Nonlinear VRFT with LASSO

A.S. Bazanella * **D. Eckhard** * **R. M. Sehnem** *

* *Universidade Federal do Rio Grande do Sul (UFRGS), Porto Alegre-RS, Brazil, (e-mail: bazanella@ufrgs.br, diegoeck@ufrgs.br e roger@sehnem.com).*

Abstract: Virtual Reference Feedback Tuning (VRFT) is a well known and very successful data-driven control design method. It has been initially conceived for linear plants and this original formulation has been much explored in the literature, besides having already found many practical applications. A nonlinear theory for VRFT has been proposed early on, but its actual application requires further developments. In this paper we highlight various issues involved in the application of nonlinear VRFT and propose the inclusion of L_1 regularization in its formulation. We illustrate by means of three simulation examples the critical role played in applications by two aspects: the L_1 regularization and the choice of dictionary used to describe the nonlinearity of the controller.

Keywords: Five to ten keywords, preferably chosen from the IFAC keyword list.

1. INTRODUCTION

Data-driven (DD) control is a research topic that has received considerable attention at least since the early 1990's (Hjalmarsson et al., 1994), and that more recently has experienced a large boost. Most of the early work on this subject turned around the classical model reference paradigm, and many different design methods have been developed along this line, usually known by their acronyms, like IFT (Hjalmarsson et al., 1994), CbT (Karimi et al., 2004), VRFT (Campi et al., 2002), OCI (Campestrini et al., 2017), among others. These methods have been extensively analyzed, enhanced, tested and applied, to the point that this has become a rather mature field (Bazanella et al., 2012). More recently, DD control theory started spreading towards various other control design paradigms, opening huge new perspectives and opportunities that are being intensely explored - see (Dörfler et al., 2023; Wang et al., 2021; Steentjes et al., 2022; Trentelman et al., 2021), for example. Meanwhile, the more traditional model reference approach is still seeing novelties, and this paper follows along this line.

In this paper we concentrate on Virtual Reference Feedback Tuning (VRFT), which is arguably the simplest, most well-known, and most widely applied data-driven control design method. VRFT has been first presented in (Campi et al., 2002) for linear plants. Many later publications have extended VRFT's scope and improved its performance: it has been extended for nonmnimum phase plants in (Campestrini et al., 2011) and for multivariable plants in (Gonçalves da Silva et al., 2018; Formentin et al., 2012a), and reformulated to optimize the response to disturbances in (Eckhard et al., 2018). It has also been adapted to obtain improved statistical properties (Garcia and Bazanella, 2022). VRFT has been very successful in various practical applications using this linear formulation (Previdi et al., 2004; Formentin et al., 2013, 2012b).

A nonlinear version of VRFT appeared in (Campi and Savaresi, 2006), but this nonlinear design method has not received similar attention in the literature. The application of the method in a nonlinear setting presents new challenges that, to the most part, have not yet been properly described, let alone solved. These challenges are more easily perceived recalling that VRFT, like all methods based on the model reference paradigm, can be seen as estimation of an *ideal controller*. The ideal controller is the one that, if put in the loop, would provide exactly the performance that has been specified. In the linear setting, the ideal controller is a linear object, so it is rather easy to devise a controller parametrization that would be appropriate to estimate it approximately. Moreover, this parametrization will be of very low dimension for the majority of problems - not larger than twice the order of the plant. It is also possible, though not so easy, to understand how undermodeling affects the closed-loop performance and how to cope with this issue (Bazanella et al., 2012).

When the object to be estimated - in this case the ideal controller - is a nonlinear map, one has to be concerned not only with its dynamic order but also with the nature of the nonlinearities involved. These are mostly unknown, since no knowledge of the plant's model can be assumed. As a result, even the simplest examples will require large dictionaries to obtain a decent estimator, which has two major consequences. One is that the resulting controller, with such large number of parameters, is in most cases very undesirable and in many cases not viable. The other one is that the statistical properties of the estimate will tend to be poor. The use of regularization is thus in order to obtain more appropriate statistical properties and more parsimonious controllers.

⋆ This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001. This study was financed in part by the Conselho Nacional de Desenvolvimento Científico e Tecnológico - Brasil (CNPq)

In this paper we explore the application of VRFT to nonlinear plants. We discuss the above issues of parametrization and optimization, and propose the use of LASSO to solve the VRFT optimization. The paper is organized as follows. In Section 2 we introduce the model reference control problem for a nonlinear plant. This sets the stage to present, in 3, the VRFT method in a general nonlinear setting, with the inclusion of L_1 regularization, that we call the LASSO-VRFT. In Section 4 we illustrate the performance of VRFT and LASSO-VRFT in three simulation case studies that highlight the possibilities and limitations of each tool: VRFT and its LASSO counterpart, dictionaries, regularization. Some conclusions and future directions of work are briefly discussed in Section 5.

2. MODEL REFERENCE CONTROL DESIGN

We are given a discrete-time SISO plant, which can be described as

$$y(t) = P[u(t)] + \nu(t) \tag{1}$$

where $y(t)$ is the output, which must track a reference signal $r(t)$, $u(t)$ is the control input, and $\nu(t)$ is the measurement noise, a stationary process. The input-output map $P[\cdot]$ is such that, for any finite control signal $u(t)$, the solutions of (1) exist and are unique.

The closed-loop performance is specified by the reference model $T_d(z)$, which is a transfer function representing the desired input-output behavior in closed-loop, that is, from the reference $r(t)$ to the output $y(t)$. The control objective is thus to make the closed-loop map from $r(t)$ to $y(t)$ to be as close as possible to the given linear map $T_d(z)$, while maintaining internal stability.

It has been shown in (Campi and Savaresi, 2006) that, if the map $P[\cdot]$ is invertible, there exists an *ideal controller*

$$u(t) = C_0[r(t) - y(t)]$$

which results in the desired closed-loop behavior if put in the loop. Thus, the control design in the model reference framework can be seen as the exercise of estimating this ideal controller.

We define the control law, which is aimed at approximating the ideal controller, in a linearly parameterized way:

$$u(t) = \Sigma_{i=1}^{m} \rho_i \psi_i(z(t)) \triangleq C(\rho, z(t)), \tag{2}$$

where $z(t)$ is the set of measurements, $\psi_i(z(t)), i = 1, \ldots, m$ contains a dictionary of functions and/or functionals, and $\rho = [\rho_1 \ \ldots \ \rho_m]^T$ is the parameter vector.

Given a reference model $T_d(z)$, a controller class in the form (2), and input-output data collected from the plant $u(t), y(t),\ t = 1, \ldots, N$, the best controller will be the one minimizing the cost function

$$J(\rho) = \sum_{t=1}^{N} [y_d(t) - y(t, \rho)]^2 \tag{3}$$

where $y_d(t) = \mathcal{Z}^{-1}\{T_d(z)R(z)\}$ is the desired closed-loop response and $y(t, \rho)$ is the actual closed-loop response to $r(t)$ obtained with the control law $C(\rho, e(t))$.

3. NONLINEAR VRFT AND REGULARIZATION

3.1 The VRFT concept

The cost function $J(\rho)$ in (3) is dependent, through $y(t, \rho)$, on the plant model, which is not available for the designer. Moreover, it is nonconvex, even with a linearly parameterized controller as (2).[1] VRFT is a design method that substitutes the cost function $J(\rho)$ with the function

$$J^V(\rho) = \sum_{t=1}^{N} [u(t) - (\Sigma_{i=1}^{m} \rho_i \psi_i(\bar{z}(t))]^2 \tag{4}$$

where $\bar{z}(t)$ is a virtual version of the measurement, that is, one in which every instance of the reference signal $r(t)$ is substituted by the virtual reference $\bar{r}(t) = \mathcal{Z}^{-1}\{T_d^{-1}(z)Y(z)\}$. Under ideal conditions $J^V(\cdot)$ has the same global minimum as $J(\cdot)$, but $J^V(\cdot)$ does not depend explicitly on the plant model, so it can be minimized without knowledge of such model, using only input-output data from the plant. Moreover, with a linear parametrization like (2), this is a quadratic function of the parameters and thus can be minimized by least squares.

The VRFT method is well-known and has been extensively applied to all sorts of linear plants, including MIMO and nonminimum-phase plants (Previdi et al., 2004; Formentin et al., 2013, 2012b). These applications include many experimental ones and also actual industrial applications, so it is a well established and successful control design methodology. For nonlinear plants and controllers, the theory has been provided in (Campi and Savaresi, 2006) but applications are hard to find in the literature (Bazanella and Neuhaus, 2014), even at the simulation level.

3.2 The VRFT Regression

The VRFT control design consists in solving the regression defined by (4). In the nonlinear setting, a large dictionary of nonlinear functions must be used in most cases. Indeed, without knowledge of a model for the plant, one does not know a priori which nonlinear functions must be present in the dictionary, so a large number of candidate functions must be employed. This contrasts with the linear case, in which a “linear dictionary” is used and thus the number of terms is at most twice the controller’s order. Priors on the plant’s nature will be very welcome to allow a reasonable choice of the dictionary’s structure - polynomial, trigonometrical, rational, etc. Without any priors, all sorts of functions must be included.

But even if one restricts the dictionary to a particular class of functions, the dimension m grows very rapidly. Take the example of a first order error feedback controller; then there are typically three signals in $z(t)$: $e(t)$, $e(t-1)$ and $u(t-1)$. If we take a modest third-order polynomial dictionary we’ll have nineteen terms already. With a second-order error feedback controller and the same third-order polynomial structure for the dictionary, 251 terms. Ordinary least squares is unlikely to handle properly such quantities of parameters, and the statistical properties will likely deteriorate very rapidly. Moreover, even if least squares would provide a statistically sound solution, one

[1] This happens even if the plant is linear

does not want to implement a controller with more than two hundred parameters; a more parsimonious controller is desired in any case. Thus, some sort of regularization is asked for, to achieve parsimony and improved statistical properties.

3.3 LASSO-VRFT

In order to cope with the large dimension of the dictionary and the case in which the ideal controller is not in the controller set, we propose to apply L_1 regularization to the optimization cost $J^V(\cdot)$. The least squares regression with L_1 regularization is also known as the LASSO - acronym for Least Absolute Shrinkage and Selection Operator - and consists, in our case, in minimizing the cost function

$$J_L^V(\rho) = \sum_{t=1}^{N} [u(t) - (\Sigma_{i=1}^m \rho_i \psi_i(\bar{z}(t))]^2 + \alpha \sum_{i=1}^{m} |\rho_i| \quad (5)$$

where α is the shrinking factor, to be chosen. Any regularization reduces the magnitudes of the parameters. Due to its geometric features, the L_1 norm tends to yield solutions in which some parameters are exactly zero. Thus, the LASSO is known to work as a selection mechanism, providing parsimony to the solution (Friedman et al., 2010).

The larger the value of α, the more parameters will be exactly zero. In this work, the Python package Scikit was used to minimize the criterion (5). This package uses the coordinate descent algorithm to find the parameters that minimize the LASSO criterion. All simulations of the Hammerstein models in this work used $\alpha = 0.001$, while the DC motor used $\alpha = 10$, which allowed to reduce the number of parameters without a significant loss of performance, and the optimization algorithm was limited to use a maximum of $100,000$ iterations. Some techniques for automatic choice of α are still under investigation, including the Akaike information criterion (AIC), the Bayes Information criterion (BIC) and Cross Validation techniques.

4. CASE STUDIES

In this Section we present simulation studies for three different plants. We collect input-output data from two discrete-time nonlinear plants excited by two different input signals: a uniformly distributed zero-mean random sequence, and a sequence of steps, both filtered by a transfer function

$$F(z) = T_d(z)(1 - T_d(z))\frac{a}{z-1},$$

as recommended in (Campi and Savaresi, 2006), and for numerical reasons we tune the parameter a so that $F(1) = 1$. The output measurement is contaminated by zero-mean gaussian white noise with varying energy levels. For all plants there is a specification of zero steady state tracking error for constant references, which requires an integrator in the controller and implies that the reference model must satisfy $T_d(1) = 1$. Each one of the following subsections presents the results for a different plant. The results obtained with the two input signals are very similar for the Hammerstein plants, so we will present in detail only the ones obtained with the random input. For the DC motor plant the best results, and hence the results presented here, were obtained with a random input. The exact plant models are not used in any design, so they are not relevant to our presentation, except for transparency; their presentation is thus delayed to an Appendix.

4.1 Hammerstein plants

The first two plants consist of made-up models in Hammerstein form. For these two plants we define the simplest nonlinear controller structure possible under the zero steady-state error constraint, which is an integrator followed by a nonlinear static element. That is, the measurement set is $z(t) = \sum_{\tau=1}^{t-1} e(\tau)$, the control law is $u(t) = \varphi(z(t))$, and the nonlinear function is what must be estimated by the VRFT regression: $\varphi(z(t)) \approx \sum_{i=1}^{m} \rho_i \psi_i(z(t))$.

The relative degree of the reference model can not be larger than the open loop system formed by plant and controller (Gonçalves da Silva et al., 2019), for this would result in the ideal controller being noncausal. The relative degree of the controller is one and any sampled system of finite order has relative degree one. Accordingly, we assume that the plants have no delay and specify a reference model with relative degree equal to two. Since we have no other information about the plant and the controller is the simplest possible, a conservative transient specification is in order. So we pick a pole at $z = 0.9$, which corresponds to a settling time of 37 samples, considerably larger than the settling time in open-loop. As a result of these considerations, the reference model is chosen as

$$T_d(z) = \frac{0.01}{(z-0.9)^2}$$

for both plants.

Next, we need to choose the dictionary of functions $\psi_i(\cdot)$. We start with a polynomial dictionary, which is quite general and intuitive, as polynomials form an orthogonal basis for the space of analytical functions. We observe that the data are symmetrical around zero, which suggests that the plant's nonlinearity has odd symmetry. This implies that the optimal $\varphi(\cdot)$ also has odd symmetry, and thus only odd powers are needed in the dictionary. For numerical reasons, we normalize the basis such that all elements have unitary magnitude at the end of scale for the data that have been collected. From all these considerations we arrive at the following parametrization:

$$\hat{\varphi}(z) = \sum_{i=1}^{m} \rho_i \left(\frac{z}{200}\right)^{2i-1} \quad (6)$$

Ideal scenario - plant #1 We have run the standard VRFT regression (4) and the LASSO-VRFT (5) with the polynomial dictionary using various values of m, using the data originated from a random input. For each design we assess the performance of each controller by evaluating the closed-loop response to a sequence of reference steps.

The closed-loop response resulting from the controller obtained with $m = 20$ is presented in Figure 1, along with the desired response $y_d(t) = \mathcal{Z}^{-1}\{T_d(z)R(z)\}$; the performance is not impressive. This somewhat expensive controller still results in a large error with respect to the desired response, and we have found that increasing m improves the performance very slowly. In all cases, when comparing the results with VRFT and with the LASSO-VRFT, we have observed that the regularization did not reduce significantly the number of parameters. So, we have been left with either a poor performance or with hundreds of parameters - hence no satisfactory solution. We have repeated the experiments with a filtered step input and they have yielded very similar results. We have also repeated these experiments with various levels of output noise, as measured by the noise's variance σ^2, with similar results up to $\sigma = 0.1$. For larger noise energies the closed-loop performance deteriorates rapidly in all designs. From now on all results presented are for $\sigma = 0.05$.

Judging by the observation of the data, this is not a plant with a complex dynamics. Yet, even if we collect data free of noise we are not able to achieve a parsimonious solution with good performance. Clearly, the controller structure is very simple, which limits severely the performance that can be achieved, but in fact the choice of dictionary is also to blame for this disappointing result. In order to assess the dictionary, we plot the estimated function $\hat{\varphi}(\cdot)$ obtained with $m = 400$, seen in Figure 2. The regularization in the LASSO-VRFT has only mildly reduced the number of parameters, to 384, and yet it can be seen in the Figure that it has changed considerably the character of the estimated function. Thus, it can be inferred that more than 384 terms are necessary to provide a good approximation for the function $\varphi(\cdot)$.

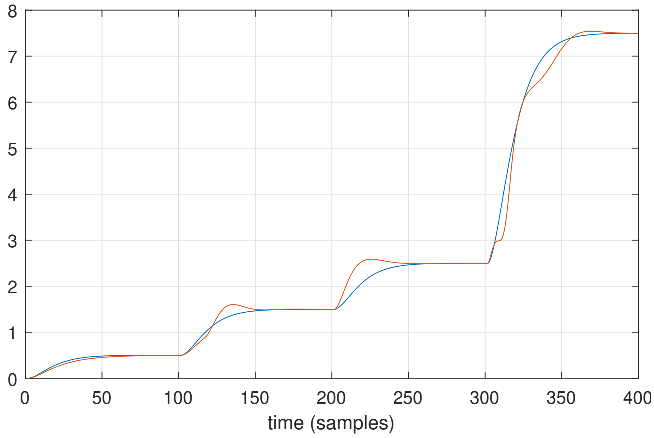

Figure 1. Closed-loop response of plant #1 with the controller obtained from the polynomial dictionary with $m = 20$

This plot looks like a piecewise affine function, which is hard to approximate with a polynomial basis, since it is not analytic. We thus pick a different dictionary, more suited to describe piecewise affine functions, which is formed by various deadzones. This dictionary is

$$\hat{\varphi}(x) = \sum_{i=1}^{m} \rho_i \; ZM_i(x) \tag{7}$$

where $ZM_i(\cdot)$ is the deadzone nonlinearity:

$$ZM_i(x) = \begin{cases} \dfrac{x + 10(i-1)}{200 - 10(i-1)} & x < -10(i-1) \\ \dfrac{x - 10(i-1)}{200 - 10(i-1)} & x > 10(i-1) \\ 0 & -10(i-1) < x < 10(i-1) \end{cases} . \tag{8}$$

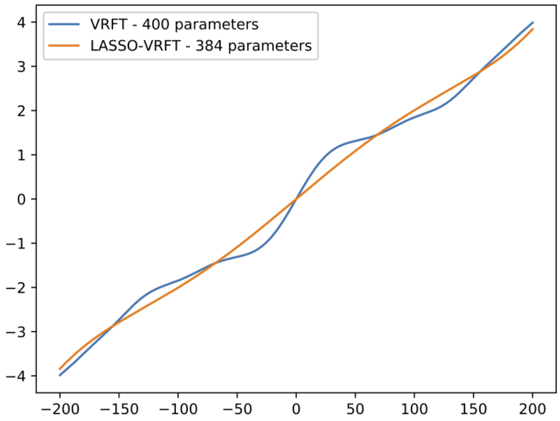

Figure 2. Estimated map $\hat{\varphi}(\cdot)$ by the polynomial dictionary for plant # 1 with $m = 400$: VRFT (blue) and LASSO-VRFT (orange)

and we choose initially $m = 20$. We run VRFT and LASSO-VRFT again with the same data and $\sigma = 0.05$, now with this new dictionary. The responses of the closed-loop system with the resulting controllers, to the same reference steps as those in Figure 1, are presented in Figure 3. The performance is very close to the desired one and the regularization has reduced the number of parameters from twenty to only four without significant change of performance.

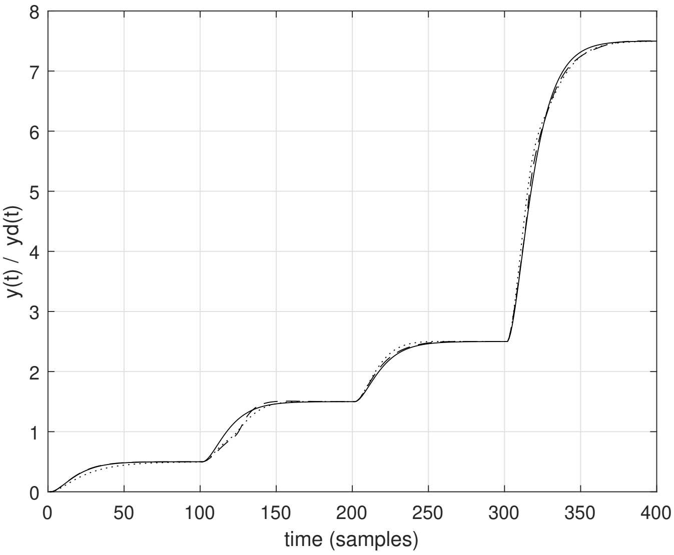

Figure 3. Closed-loop responses of plant #1 with the controllers obtained from the deadzone dictionary with $m = 20$, with VRFT (dashed line) and LASSO-VRFT (dotted line)

With this we have successfully concluded a data-driven design with LASSO-VRFT. In order to learn more about the behavior of the dictionary and the role of the regularization, we have tried a larger dictionary with the deadzone basis: $m = 400$. This new dictionary will have the same definition as is (7) but with each occurrence of the number 10 replaced by 0.5. We have run the LASSO-VRFT (VRFT is not of interest for this exercise) and obtained the results in Figure 4, with the regularization having reduced the number of nonzero parameters to 47. The regularization was successful in reducing drastically the number of nonzero parameters and providing good closed-loop performance. Equally important is the fact that the regularization has proven indispensable in this case: the controller obtained without regularization presented many parameters with very large magnitudes and resulted in an unstable closed-loop.

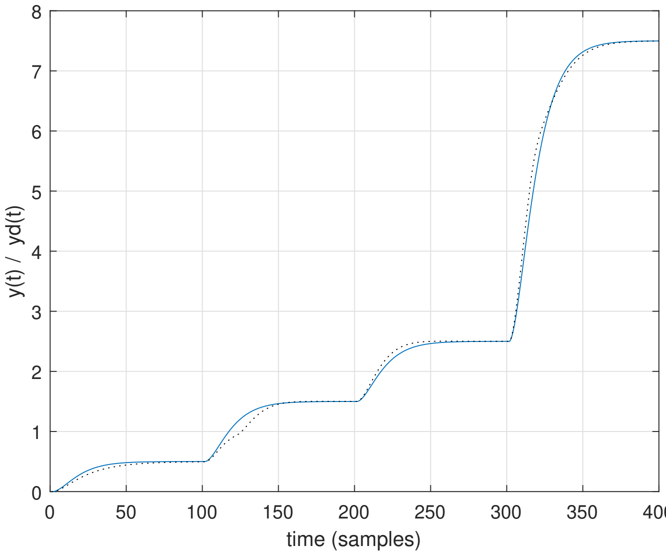

Figure 4. Closed-loop response of plant #1 with the controller obtained from the deadzone dictionary with $m = 400$ with LASSO-VRFT

Non-ideal scenario - plant #2 When we run LASSO-VRFT with the polynomial dictionary, the map's estimate $\hat{\varphi}(\cdot)$ shown in Figure 5 is obtained. The regularization zeroes only twenty parameters and yet this completely changes the map's character to an almost linear map. Moreover, the controller obtained without regularization is not even stabilizing. In this case the polynomial dictionary is a complete failure: even if we were willing to implement a controller with several hundred parameters, the result would either be an unstable closed-loop (VRFT) or poor performance with no compensation of the nonlinear features of the plant (LASSO-VRFT).

We next try the deadzone dictionary (7), first with $m = 20$ and then with $m = 400$. The LASSO-VRFT reduced the number of parameters from $m = 400$ to 52, and from $m = 20$ to 8. The closed-loop performance is shown in Figure 6, where it is seen that the performances obtained with the LASSO-VRFT with $m = 20$ and $m = 400$ are virtually indistinguishable.

Examination Though the plants' models were used to get the data, the control design was performed without using any knowledge of them, except the assumption that the plants were delay-free. As seen in the Appendix, both plants consist of Hammerstein systems, with the same piecewise affine nonlinear function in front of a linear block.

It can be verified without much effort that for plant #1 the ideal controller is given by:

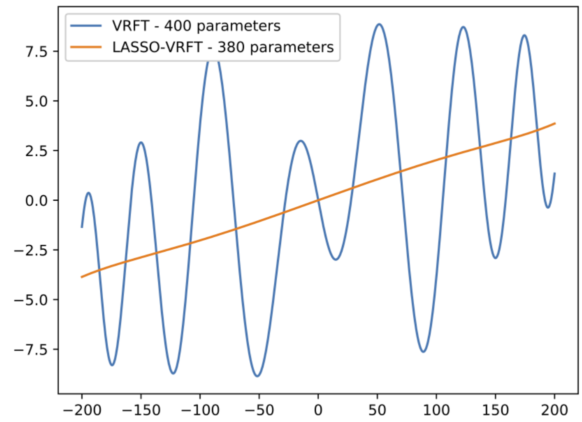

Figure 5. Estimated map $\hat{\varphi}(\cdot)$ with polynomial dictionary $m = 400$ for plant # 2: VRFT and LASSO-VRFT

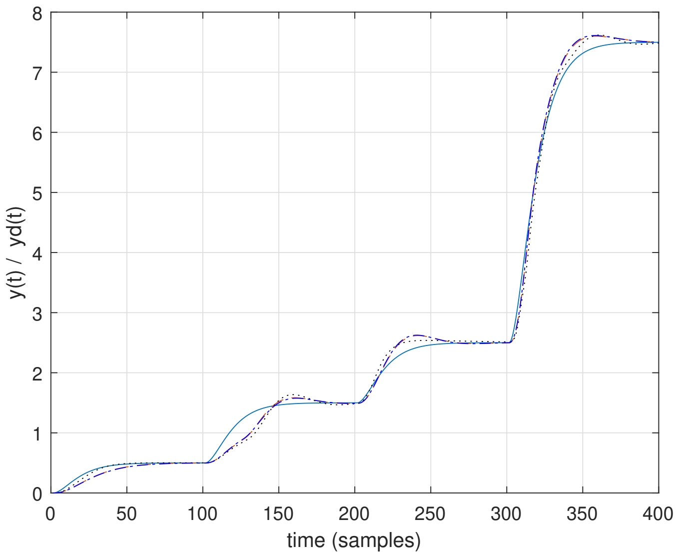

Figure 6. Closed-loop responses of plant #2 with the controllers obtained from the deadzone dictionary: $m = 20$ with VRFT (dotted line) and with LASSO-VRFT (dashed); $m = 400$, with LASSO-VRFT (dash-dot)

$$v(t) = v(t-1) + 0.05e(t) \tag{9}$$

$$u(t) = \phi^{-1}(v(t)) \tag{10}$$

where $\phi^{-1}(\cdot)$ is the left-inverse of $\phi(\cdot)$, that is, $\phi^{-1}(\phi(x)) = x$. This ideal controller is in the controller class, which is why we called this the *ideal scenario*. The nonlinear function presented in Figure 2 is a close approximation to the left inverse of $\phi(\cdot)$ multiplied by a constant factor 0.05. Hence, the polynomial dictionary estimated a controller that is very close to the ideal controller, but for that it needed hundreds of parameters. On the other hand, the deadzone dictionary was able to get very close to the ideal controller with a parsimonious controller; this was true also for the case $m = 400$. The choice of dictionary has played a critical role for this very simple example, and the L_1 regularization was also very important.

Concerning plant #2, the ideal controller does not belong to the controller class; it is given by

$$V(z) = \frac{0.25(z-0.8)}{z(z-1)}E(z) \tag{11}$$

$$u(t) = \phi^{-1}(v(t)) \tag{12}$$

where $V(z) = \mathcal{Z}\{v(t)\}$ and $E(z) = \mathcal{Z}\{e(t)\}$. For this plant, regularization played a more critical role than for the first plant, where the ideal controller belongs to the controller class. Indeed, without regularization both dictionaries were unable to provide a stabilizing controller.

4.2 A DC motor

Our third case study consists of the speed control of a DC-motor through its field voltage. Even though we are not going to use the model itself in the design, in most practical cases completely ignoring the model is not a realistic assumption either. Indeed, if one is to control a DC-motor, one must have an idea of what a model would look like - specifically, what kind of nonlinear function would appear in the model and thus in the ideal controller. This knowledge allows us to choose a reasonable dictionary from the start. Since we want zero tracking error, our measurement set is composed of two signals, the error integral $w(t) = \sum_{\tau=1}^{t} e(\tau)$ and the control, the control law is $u(t) = \varphi(z(t), u(t-1))$. In this case we have chosen φ composed of two-variable polynomials functions, presented in Table 1.

Table 1. Nonlinear Functions of the dictionary

Functions	
$\psi_1 = w(t)$	$\psi_{20} = \psi_1\psi_3\psi_5$
$\psi_2 = w(t-1)$	$\psi_{21} = \psi_1\psi_4\psi_5$
$\psi_3 = w(t-2)$	$\psi_{22} = \psi_2\psi_3\psi_4$
$\psi_4 = w(t-3)$	$\psi_{23} = \psi_2\psi_3\psi_5$
$\psi_5 = w(t-4)$	$\psi_{24} = \psi_2\psi_4\psi_5$
$\psi_6 = \psi_1\psi_2$	$\psi_{25} = \psi_3\psi_4\psi_5$
$\psi_7 = \psi_1\psi_3$	$\psi_{26} = \psi_1\psi_2\psi_3\psi_4$
$\psi_8 = \psi_1\psi_4$	$\psi_{27} = \psi_1\psi_2\psi_3\psi_5$
$\psi_9 = \psi_1\psi_5$	$\psi_{28} = \psi_1\psi_2\psi_4\psi_5$
$\psi_{10} = \psi_2\psi_3$	$\psi_{29} = \psi_1\psi_3\psi_4\psi_5$
$\psi_{11} = \psi_2\psi_4$	$\psi_{30} = \psi_2\psi_3\psi_4\psi_5$
$\psi_{12} = \psi_2\psi_5$	$\psi_{31} = \psi_1\psi_2\psi_3\psi_4\psi_5$
$\psi_{13} = \psi_3\psi_4$	$\psi_{32} = u(t-1)$
$\psi_{14} = \psi_3\psi_5$	$\psi_{33} = u(t-2)$
$\psi_{15} = \psi_4\psi_5$	$\psi_{34} = u(t-3)$
$\psi_{16} = \psi_1\psi_2\psi_3$	$\psi_{35} = u(t-1)u(t-2)$
$\psi_{17} = \psi_1\psi_2\psi_4$	$\psi_{36} = u(t-1)u(t-3)$
$\psi_{18} = \psi_1\psi_2\psi_5$	$\psi_{37} = u(t-2)u(t-3)$
$\psi_{19} = \psi_1\psi_3\psi_4$	$\psi_{38} = u(t-1)u(t-2)u(t-3)$

The reference model is more easily derived in the continuous time domain, since this is also the natural domain of the plant. The desired continuous transfer function is

$$T_d(s) = p\frac{(s+70)}{(s+120)^3} \tag{13}$$

where p is such that $T_d(0) = 1$. The poles here are selected to give the system a settling time similar to what's observed in open loop, whereas the zeros are selected to give $T_d(s)$ the same relative degree as the plant, without causing too much overshoot. The discrete time reference model $T_d(z)$ is obtained by applying a ZOH to (13) with sampling time of $\delta_T = 4\ ms$.

We have collected data with a randomly generated input with amplitude equal to five, and various noise levels. For each noise level we have run VRFT with and without LASSO. We start by presenting the results obtained for the noiseless case. The closed-loop performance obtained in this case is presented in Figure 7, and the cost (3) for a step of 400 evaluates to $J = 13.24 \times N$. It is seen that the performance is close to the specification and (equally important) does not change significantly with the operating point, indicating a closed-loop behavior that is close to linear.

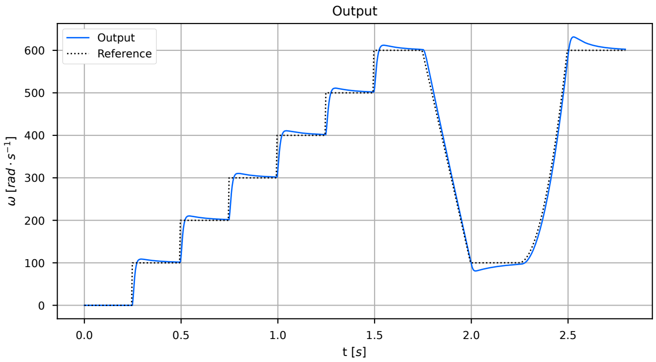

Figure 7. Closed-loop response of plant #3 with the controller obtained from a noiseless experiment

It's also relevant to look at the energy of each one of the thirty-eight terms of the controller, defined in (14).

$$E_{u_m} = \sum_{m=1}^{N} [\rho_m \psi_m(t)]^2 \tag{14}$$

The higher the energy of a given term, the more it is contributing to the control. Figure 8 shows the energy of each term, where it is seen that the most relevant terms are the ones for m around 20, which are polynomials of order three in the measurement. On the other hand the linear terms of the controller (the first ones) contribute much less to the control, indicating that the obtained controller is far from linear.

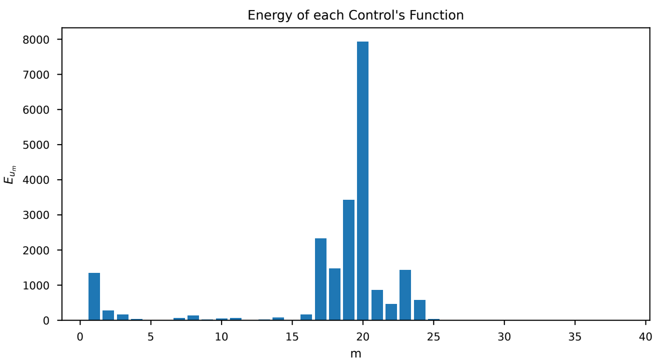

Figure 8. Energy of each control signal.

For each noisy experiment with have run 100 Monte Carlo simulations. The performance deterioration due to noise starts to become significant for noise with standard deviation around $\sigma = 36$, and for $\sigma = 73.5$ already a large proportion of the designs (at least 30%) result in closed-loop instability. We show the results for $\sigma = 18.4$ in Figure 9, which presents a histogram of the cost functions obtained without LASSO. The results with LASSO are similar, though with somewhat larger values of J. On the other hand, for larger noise variances we have observed that the use of LASSO improved the robustness, resulting in smaller probability of obtaining an unstable closed-loop. The effectiveness of LASSO in this example seems to be limited by two factors: the fact that the dictionary is already parsimonious from the beginning, and the difficulty in finding the best value of the weight α. Still, this is an issue that requires further investigation.

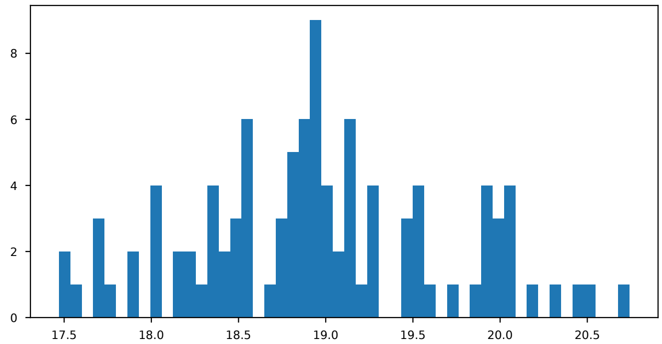

Figure 9. Histogram of the cost function J/N for the plant #3 with the controllers obtained from noisy experiments

Table 2. Average Values of the Paramaters for $\sigma = 18.4$

Parameters		
$\rho_1 = 1.98 \times 10^{-03}$	$\rho_{14} = -7.22 \times 10^{-08}$	$\rho_{27} = 1.14 \times 10^{-14}$
$\rho_2 = 1.59 \times 10^{-03}$	$\rho_{15} = 5.38 \times 10^{-08}$	$\rho_{28} = -2.81 \times 10^{-14}$
$\rho_3 = -1.61 \times 10^{-03}$	$\rho_{16} = 3.94 \times 10^{-10}$	$\rho_{29} = -2.84 \times 10^{-15}$
$\rho_4 = -1.21 \times 10^{-03}$	$\rho_{17} = -1.06 \times 10^{-10}$	$\rho_{30} = 2.44 \times 10^{-16}$
$\rho_5 = -6.51 \times 10^{-04}$	$\rho_{18} = 4.71 \times 10^{-11}$	$\rho_{31} = -7.91 \times 10^{-18}$
$\rho_6 = 4.52 \times 10^{-08}$	$\rho_{19} = -1.81 \times 10^{-10}$	$\rho_{32} = 5.22 \times 10^{-06}$
$\rho_7 = -6.04 \times 10^{-08}$	$\rho_{20} = 3.10 \times 10^{-11}$	$\rho_{33} = 3.99 \times 10^{-06}$
$\rho_8 = 2.78 \times 10^{-09}$	$\rho_{21} = -1.68 \times 10^{-10}$	$\rho_{34} = 2.15 \times 10^{-06}$
$\rho_9 = 8.20 \times 10^{-08}$	$\rho_{22} = -1.50 \times 10^{-10}$	$\rho_{35} = 1.62 \times 10^{-07}$
$\rho_{10} = -3.37 \times 10^{-08}$	$\rho_{23} = -4.61 \times 10^{-10}$	$\rho_{36} = 2.11 \times 10^{-07}$
$\rho_{11} = -2.38 \times 10^{-08}$	$\rho_{24} = 1.46 \times 10^{-10}$	$\rho_{37} = 1.83 \times 10^{-07}$
$\rho_{12} = 2.25 \times 10^{-08}$	$\rho_{25} = 5.09 \times 10^{-10}$	$\rho_{38} = 7.29 \times 10^{-06}$
$\rho_{13} = 2.03 \times 10^{-08}$	$\rho_{26} = 9.69 \times 10^{-15}$	

5. CONCLUSIONS

We have discussed the application of the VRFT method to nonlinear plants, and proposed the inclusion of L_1 regularization in the regression problem to cope with the difficulties encountered in this application. Two simple case studies have been presented that illustrated these difficulties and highlighted two main points: the important role played by the regularization and the criticality of the choice of dictionary to describe the controller. For the case in which the ideal controller is not in the control class, regularization proved to be more critical to the success of the design. A third, somewhat more realistic example, illustrates that knowledge of the nature of the plant alone leads to the choice of a relatively parsimonious dictionary that provides the desired performance with appropriate robustness to noise. Future work concentrates on the application of these ideas to the design of controllers for more complex plants, also with other DD control design methods.

REFERENCES

Bazanella, A., Campestrini, L., and Eckhard, D. (2012). *Data-Driven Controller Design: The H2 Approach*. Springer, New York, USA.

Bazanella, A.S. and Neuhaus, T. (2014). Tuning nonlinear controllers with the virtual reference approach. *IFAC Proceedings Volumes*, 47(3), 10269–10274. doi: https://doi.org/10.3182/20140824-6-ZA-1003.00562. URL `https://www.sciencedirect.com/science/article/pii/S147466701643243X`. 19th IFAC World Congress.

Campestrini, L., Eckhard, D., Gevers, M., and Bazanella, A. (2011). Virtual Reference Feedback Tuning for Non Minimum Phase Plants. *Automatica*, 47(8), 1778–1784.

Campestrini, L., Eckhard, D., Sanfelice Bazanella, A., and Gevers, M. (2017). Data-driven model reference control design by prediction error identification. *Journal of the Franklin Institute*, 354(6), 2628–2647. doi:https://doi.org/10.1016/j.jfranklin.2016.08.006. URL `https://www.sciencedirect.com/science/article/pii/S001600321630271X`. Special issue on recent advances on control and diagnosis via process measurements.

Campi, M., Lecchini, A., and Savaresi, S. (2002). Virtual reference feedback tuning: a direct method for the design of feedback controllers. *Automatica*, 38, 1337–1346.

Campi, M. and Savaresi, S. (2006). Direct nonlinear control design: the virtual reference feedback tuning (vrft) approach. *IEEE Transactions on Automatic Control*, 51, 14–27.

Dörfler, F., Coulson, J., and Markovsky, I. (2023). Bridging direct and indirect data-driven control formulations via regularizations and relaxations. *IEEE Transactions on Automatic Control*, 68(2), 883–897. doi:10.1109/TAC.2022.3148374.

Eckhard, D., Campestrini, L., and Christ Boeira, E. (2018). Virtual disturbance feedback tuning. *IFAC Journal of Systems and Control*, 3, 23–29. doi:https://doi.org/10.1016/j.ifacsc.2018.01.003. URL `https://www.sciencedirect.com/science/article/pii/S2468601817301670`.

Formentin, S., Savaresi, S., and Re, L.D. (2012a). Non-iterative direct data-driven controller tuning for multivariable systems: theory and application. *IET Control Theory & Applications*, 6, 1250–1257(7). URL `https://digital-library.theiet.org/content/journals/10.1049/iet-cta.2011.0204`.

Formentin, S., De Filippi, P., Corno, M., Tanelli, M., and Savaresi, S.M. (2013). Data-driven design of braking control systems. *IEEE Transactions on Control Systems Technology*, 21(1), 186–193. doi:10.1109/TCST.2011.2171965.

Formentin, S., Hirsch, M., Savaresi, S.M., and Del Re, L. (2012b). Direct data-driven control of a diesel engine airpath. In *2012 American Control Conference (ACC)*, 4198–4203. doi:10.1109/ACC.2012.6314904.

Friedman, J., Hastie, T., and Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, 33(1), 1.

Garcia, C. and Bazanella, A. (2022). Improving the performance of data-driven control by discarding data and using data least squares. *Journal of Control Automation and Electrical Systems*. doi:https://doi.org/10.1007/s40313-021-00891-x.

Gonçalves da Silva, G.R., Bazanella, A.S., and Campestrini, L. (2019). On the choice of an appropriate reference model for control of multivariable plants. *IEEE Transactions on Control Systems Technology*, 27(5), 1937–1949. doi:10.1109/TCST.2018.2842178.

Gonçalves da Silva, G.R., Campestrini, L., and Bazanella, A.S. (2018). Multivariable virtual reference feedback tuning for non-minimum phase plants. *IEEE Control Systems Letters*, 2(1), 121–126. doi:10.1109/LCSYS.2017.2763824.

Hjalmarsson, H., Gunnarsson, S., and Gevers, M. (1994). A convergent iterative restricted complexity control design scheme. In *33rd IEEE Conference on Decision and Control*. Lake Buena Vista - USA.

Karimi, A., Mišković, L., and Bonvin, D. (2004). Iterative correlation-based controller tuning. *International Journal of Adaptive Control and Signal Processing*, 18, 645–664.

Khalil, K. (1996). *Nonlinear Systems,"2nd Edition, Englewood Cliffs, NJ, Prentice-Hall.* Prentice Hall.

Previdi, F., Schauer, T., Savaresi, S., and Hunt, K. (2004). Data-driven control design for neuroprotheses: a virtual reference feedback tuning (vrft) approach. *IEEE Transactions on Control Systems Technology*, 12(1), 176–182. doi:10.1109/TCST.2003.821967.

Steentjes, T.R., Lazar, M., and Van den Hof, P.M. (2022). Informativity conditions for data-driven control based on input-state data and polyhedral cross-covariance noise bounds. *IFAC-PapersOnLine*, 55(30), 329–334. doi:https://doi.org/10.1016/j.ifacol.2022.11.074. URL https://www.sciencedirect.com/science/article/pii/S2405896322027070. 25th International Symposium on Mathematical Theory of Networks and Systems MTNS 2022.

Trentelman, H.L., Van Waarde, H.J., and Camlibel, M.K. (2021). An informativity approach to the data-driven algebraic regulator problem. *IEEE Transactions on Automatic Control*, x(x), 1–1. doi:10.1109/TAC.2021.3129457.

Wang, X., Sun, J., Berberich, J., Wang, G., Allgöwer, F., and Chen, J. (2021). Data-driven control of dynamic event-triggered systems with delays. *arXiv preprint arXiv:2110.12768*.

Appendix A. PLANT'S MODELS

The first two plants consist of Hammerstein systems, with the same piecewise affine nonlinear function

$$\phi(x) = \begin{cases} 2x-2, & x<-2 \\ 5x+4, & -2<x<-1 \\ x, & |x|<1 \\ 5x-4, & 1<x<2 \\ 2x+2, & x>2 \end{cases}$$

in front of a linear block.

The linear block of plant #1 is a first-order system with transfer function

$$G(z) = \frac{0.2}{z-0.8}$$

whereas in plant #2 the transfer function is

$$G(z) = \frac{0.04\ z}{(z-0.8)^2}.$$

The third plant is a third-order model of a field-controlled DC-motor (Khalil, 1996):

$$\dot{x}_1 = -\frac{R_f}{L_f}x_1 + \frac{1}{L_f}u \tag{A.1}$$

$$\dot{x}_2 = -\frac{R_a}{L_a}x_2 + \frac{1}{L_a}v_a - \frac{k_b}{L_a}x_1x_3 \tag{A.2}$$

$$\dot{x}_3 = \frac{k_m}{J}x_1x_2 - \frac{K_f}{J}x_3 \tag{A.3}$$

$$y = x_3. \tag{A.4}$$

where u is the field voltage (the control input), y is the speed (the controlled output) and v_a is the (constant) armature voltage. The parameter values used are

$$\begin{aligned} R_f &= 5 \\ L_f &= 300 \times 10^{-6} \\ R_a &= 2 \\ L_a &= 600 \times 10^{-6} \\ v_a &= 15 \\ k_b &= 0.05 \\ k_m &= 0.05 \\ J &= 7 \times 10^{-6} \\ K_f &= 48 \times 10^{-9}. \end{aligned}$$