Annual Reviews in Control 56 (2023) 100906

Contents lists available at ScienceDirect

Annual Reviews in Control

journal homepage: www.elsevier.com/locate/arcontrol

# The data-driven approach to classical control theory

Alexandre Sanfelici Bazanella [a,*,1], Lucíola Campestrini [a,1], Diego Eckhard [b,1]

[a] Department of Automation and Energy, Universidade Federal do Rio Grande do Sul, Porto Alegre - RS, Brazil
[b] Department of Pure and Applied Mathematics, Universidade Federal do Rio Grande do Sul, Porto Alegre - RS, Brazil

ABSTRACT

A data-driven approach to control design has been developing, since the early 1990's, upon the concepts and the methods of classical control theory; to this approach we refer as *data-driven classical control*. This is now a consolidated theory, with a large body of methods and a great number of applications. In this paper we present a survey of these developments along the past three decades. The theory is overviewed, various methods are described, their applications are illustrated and some issues that are sill open for future research are discussed. The extension of these concepts to the design of nonlinear controllers is an emerging and promising subject that is also discussed, and it is shown how it can be approached by the theory of geometric control and the tools of machine learning.

## 1. Introduction

Classical control theory, as known and taught from the textbooks that are popular worldwide – like (Franklin et al., 2006; Ogata, 2009), among many others – deals with input–output (I/O) representations of the objects under study: the system to be controlled, the controller, the sensor and the actuator. These objects are seen as operators in the signals space and these operators are represented by their kernels — differential or finite-difference equations, transfer functions in Laplace or Z-domain. The specifications for the design come in the form of class of references to be tracked (usually standard deterministic signals such as steps and sinusoids), the class of disturbances to be rejected (again standard deterministic signals, but also stochastic processes), steady state and transient performance criteria (settling time, rising time, maximum overshoot in response to the specified classes of signals), and robustness (sensitivity, gain and phase margins).

The design methods that became known under this denomination "classical control" – root locus, loop shaping using Bode plots, Nyqvist diagrams, etc – were developed and made "popular" in the middle of the twentieth century. In this extensive and powerful theory, the analysis and the design start from the model of the to-be-controlled system — the "plant". Most of the classical control theory turns around the plant's model and the certainty equivalence principle, which play, side by side, the central role in it. These quintessential *model-based* methods were not always dominant, and before their appearance controllers were designed more often than not by data-driven methods, such as Ziegler and Nichols (1942). But in the 1970's these methods had already taken over the field of control.

It was around the year 1990 that a particular brand of Data-Driven (DD) control design appeared within this sphere (Bazanella, Campestrini et al., 2012; Bazanella et al., 2022). These DD design methods were developed upon the concepts of classical control theory and its objects, intending to take advantage of this classical framework and its theoretical concepts and tools. But they have appeared to contrast with the overwhelming dominance that model-based analysis and design had achieved, intending to perform the design without any previous knowledge of the plant's model and with disregard for the certainty equivalence principle. Since data are collected in discrete-time, most adapted to this DD approach to design are representations in discrete-time. Accordingly, this work has been developed almost exclusively in the discrete-time domain. It is this approach to data-driven control design, which is founded upon the classical control theory, and more specifically on its discrete-time part, that we call *data-driven classical control* and that we will survey in this paper.

Many different methodologies for control design have been proposed within this framework, which became known mostly by acronyms given by their creators, such as IFT (Hjalmarsson et al., 1998), FDT (Kammer et al., 2000), VRFT (Campi et al., 2002), CbT (Karimi et al., 2004), OCI (Campestrini et al., 2017), FRIT (Kaneko, 2013). These methodologies have found many applications and are still constantly evolving. They also motivated the development of subsidiary techniques, such as controller certification (Dehghani et al., 2009), dedicated optimization methods (Eckhard et al., 2017), data analysis, signal processing and data selection (Garcia & Bazanella, 2022). This evolution will be summarized, trying to give a historical perspective and to highlight the best practices that are available according to the current theory and the consolidated practice. In Section 2 the fundamental concepts and elements of this theory are described and

* Corresponding author.
*E-mail address:* bazanella@ufrgs.br (A. Sanfelici Bazanella).

[1] This work was partially supported by Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - CAPES - Finance Code 001 and CNPq. Both sources belong to the Brazilian government.

https://doi.org/10.1016/j.arcontrol.2023.100906
Received 19 July 2023; Accepted 24 September 2023
Available online 4 October 2023
1367-5788/

formally defined: the systems' representations and the performance criteria, which are inherited from classical control theory; then the design philosophy, which sets DD control apart from it. This formal treatment naturally allows to further clarify the specific features of this brand of control design and to contextualize it within the wide world of data-driven control methods and control theory in general. The numerous control design methods are derived from either one of two different conceptual approaches, that we have named *optimization-based* and *identification-based* – seminal references for each one are respectively (Guardabassi & Savaresi, 2000; Hjalmarsson et al., 1994). Each one of these conceptually distinct approaches has one section dedicated to it – Sections 3 and 4 respectively – in which the corresponding design methods are presented. We do not have the naïve pretension of covering all methods that were presented in the literature. The methods that are described here have been selected for being paradigmatic of the main design concepts, by their ubiquity in the control literature and by their consolidation as shown in experimental practice. As is unavoidable, this selection has also been influenced by our own background and experience; hopefully this was to a lesser extent.

Some accessory issues are briefly examined in Section 5: the difficulties posed by nonminimum-phase plants and the solutions that have been found to cope with them; the controller certification problem; the data curation question; the challenges of DD distributed control. Then the application side of the DD classical control design methodologies is discussed. A large number of applications have been described in the literature; many of them will be mentioned in Section 6 to provide a glimpse. These applications are both academic and practical, in many different branches of engineering, and involve different methodologies of DD classical control. When working an application, various aspects have to be taken into account to decide what methodology(ies) is(are) best to the case at hand. These various aspects are also discussed, highlighting the one that is particular to the data-driven control realm and that becomes more critical every day: the availability and the quality of the data related to the control loop. Then two specific applications with different characteristics are described in detail: voltage control of Uninterruptible Power Sources and motion control of Unmanned Aerial Vehicles. Real-life applications are seldom seen in the literature, as practicing engineers are usually not concerned with publicizing them — they are more often concerned in doing the opposite, sometimes for good reason. So, this presentation serves to illustrate the real-life application of data-driven classical control, the possibilities and the challenges.

Then the data-driven design of nonlinear controllers is explored. There is a comprehensive body of theory on nonlinear systems, involving geometric control (Isidori, 1995) and Lyapunov analysis (Khalil, 2001) that at this point in time could also be called *classical*. It will be shown in Section 7 that the design of nonlinear controllers can be formulated in the same way as in the linear case and that the development of a theory based on this formulation requires the help of this "classical nonlinear control" theory. But it is also shown that the problems thus posed tend to be data-hungry and require more sophisticated computational tools, of the sort that are nowadays becoming standard in data science and machine learning. Finally, a very short conclusion is given in Section 8.

## 2. The framework

### 2.1. The elements

Consider a linear time-invariant discrete-time multiple-input multiple-output (MIMO) process with I/O representation given as

$$y(t) = G(q)u(t) + v(t) \tag{1}$$

where $u(t)$ and $y(t)$ are n-vectors representing the process' input and output, respectively, and the n-vector $v(t)$ is a stochastic process described by $v(t) = H(q)\mu(t) + s(t)$, $\mu(t)$ being white noise, $H(\infty) = I_n$ and $s(t)$ a deterministic perturbation. The transfer matrix $G(q)$ is an $n \times n$ matrix whose elements are proper rational transfer functions and $q$ is the time-shift operator $qx(t) = x(t+1)$.

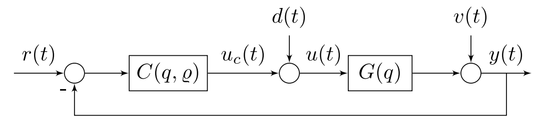


**Fig. 1.** Closed-loop block diagram.

The process' input signal is composed by two terms:

$$u(t) = u_c(t) + d(t) \tag{2}$$

where $d(t)$ is a disturbance signal – called the *load disturbance* – and $u_c(t)$ is the control input. The control law is

$$u_c(t) = C(q,\varrho)\left(r(t) - y(t)\right), \tag{3}$$

where $r(t)$ is the reference signal and $C(q,\varrho)$ is a linear time-invariant controller described by a square $n \times n$ matrix of rational transfer functions. A block diagram of the system in closed-loop is presented in Fig. 1.

The controller belongs to a predefined controller class defined as

$$\mathcal{C} = \{C(q,\varrho), \varrho \in \mathbb{R}^p\},$$

for some analytical function $C(\cdot,\cdot) : \mathbb{C}\times\mathbb{R}^p \to \mathbb{C}^{n\times n}$ and is parametrized by a real parameter vector

$$\varrho = \begin{bmatrix}\rho_1 & \rho_2 & \dots & \rho_p\end{bmatrix}^T$$

with $\rho_i \in \mathbb{R}, i = 1,\dots,p$. The definition of the controller class – the set $\mathcal{C}$ – is usually done at earlier stages of the design, in which considerations like the nature of the reference signal and of the disturbances, the complexity of the plant, the nature of the performance specifications and the available hardware and software for implementation of the controller are taken into account. This definition is discussed briefly in Section 2.4, where the most important canonical controller classes are presented.

The system (1)–(2)–(3) in closed-loop reads:

$$y(t) = T(q,\varrho)r(t) + Q(q,\varrho)d(t) + S(q,\varrho)v(t), \tag{4}$$

where the transfer matrices are defined as:

$$\begin{aligned}
S(q,\varrho) &= (I + G(q)C(q,\varrho))^{-1},\\
Q(q,\varrho) &= S(q,\varrho)G(q),\\
T(q,\varrho) &= S(q,\varrho)G(q)C(q,\varrho).
\end{aligned}$$

The task of the control design is to choose the parameter vector $\varrho$ of the controller in order to obtain appropriate performance for the closed-loop system. In the model reference approach, which is dominant in data-driven classical control, "appropriate performance" is defined by a desired closed-loop transfer function. This is a natural means to formalize the typical performance and robustness specifications from classical control that has also proven to be very successful over the years. When the specified transfer function is the complementary sensitivity $T(q)$, this is called the "reference model" and the corresponding design approach is called model reference design. Though the model reference designation usually refers only to design methods aimed at achieving a specified $T(q)$, we will use it to refer to the pursuit of a specified sensitivity function $S(q)$ or a specified input sensitivity $Q(q)$ as well.

Model reference design was there at the birth of learning systems, as the primary approach to "self-learning" systems being developed 60 years ago — see Kalman (1960) and various other papers at that first IFAC World Congress. It has become the standard formulation in adaptive control theory for decades, giving rise to the classical MRAC –

Model reference Adaptive Control – and its countless derivations. The equally celebrated minimum variance controller and the STR – Self-Tuning Regulator – can also be derived from this formulation. These classical and highly influential approaches to adaptive control are also data-driven, though this label has only been attached to them very recently.

### 2.2. Performance criteria

*Reference tracking.* For the reference tracking control objective, model reference design amounts to specifying a desired output for the closed-loop system considering a specified reference signal, that is

$$y_{dr}(t) = T_d(q)r(t),$$

where $T_d(q)$ is the *Reference Model*. Then, the parameters of the controller $C(q,\varrho) \in \mathcal{C}$ are obtained as the solution of the following optimization problem

$$\begin{aligned} \varrho^{MR} &= \arg\min\ J^{MR}(\varrho) \\ J^{MR}(\varrho) &\triangleq \bar{E}[(T_d(q) - T(q,\varrho))r(t)]^2. \end{aligned} \tag{5}$$

where $\bar{E}f(t) = \lim_{N\to\infty} \frac{1}{N}\sum_{t=1}^{N} Ef(t)$ as defined in Ljung (1999).

Analyzing (5) it is observed that if the *ideal controller*

$$C_d^{MR}(q) = G(q)^{-1}T_d(q)(I - T_d(q))^{-1} \tag{6}$$

were used in the closed loop, then the objective function (5) would evaluate to zero.

The same treatment can be applied to the other two essential control objectives: Output Disturbance Rejection and Load Rejection.

*Output Disturbance Rejection.* For output disturbance rejection one can specify a desired sensitivity function $S_d(q)$ and then minimize the cost

$$J^{OD}(\varrho) \triangleq \bar{E}[(S_d(q) - S(q,\varrho))v(t)]^2. \tag{7}$$

Analyzing (7) it is seen that for this control objective the ideal controller is given by

$$C_d^{OD}(q) = G(q)^{-1}\left(I - S_d(q)\right)S_d(q)^{-1}. \tag{8}$$

This is closely related to the celebrated minimum variance controller (Åström & Wittenmark, 1973). In fact, if the specification is $S_d(q) = H^{-1}(q)$ then the ideal controller $C_d^{OD}(q)$ is precisely the minimum variance controller. On the other hand, recalling that $S(q) + T(q) \equiv 1$, one realizes that specifying the output sensitivity $S_d(q)$ automatically specifies the complementary sensitivity $T_d(q)$. So, once a desired sensitivity function $S_d(q)$ is chosen, one can just determine the corresponding $T_d(q) = 1 - S_d(q)$ and proceed as if this were a reference tracking design. This usually requires filtering the data appropriately to account for the difference between the signals $v(t)$ and $r(t)$ in (7) and (5), but does not require an independent theoretical treatment from the reference tracking criterion.

*Load Rejection.* For the purpose of input disturbance (aka load) rejection, model reference design consists in specifying a desired input sensitivity function $Q_d(z)$ and then minimizing the cost function

$$J^{ID}(\rho) \triangleq \bar{E}[(Q_d(q) - Q(q,\varrho))d(t)]^2. \tag{9}$$

It is seen that if the ideal controller

$$C_d^{ID}(q) = Q_d(q)^{-1} - G(q)^{-1} \tag{10}$$

were used in the closed-loop then the objective function (9) would evaluate to zero. This control objective requires a separate treatment from the reference tracking and output disturbance criteria (Eckhard et al., 2018).

For each control objective an ideal controller has been defined which is an universal optimizer, that is, the optimal for an unrestricted control class. Whether or not the ideal controller belongs to the controller class $\mathcal{C}$ is a crucial issue in the theory of DD control design, just like the issue of whether or not the real system belongs to the model class in system identification. We formally state this assumption below.

**Assumption 1** (*Matching Condition*). The ideal controller belongs to the controller class $\mathcal{C}$.

This assumption reads equivalently as

$$\exists\ \varrho_d \in \mathbb{R}^p \ \text{ such that }\ C(q,\varrho_d) = C_d(q)$$

where $C_d(q)$ stands for either one of the ideal controllers defined for the reference tracking criterion, the output disturbance rejection or load rejection criteria.

When Assumption 1 holds, we say that a full-order controller is being used and we can hope that the control design will yield the ideal controller. When it does not hold, the controller obtained can never be the ideal controller, but it results in a closed-loop response that is as close as possible to the desired output. However, if the specified performance is very far from what can be achieved with the given controller class, then "as close as possible" becomes meaningless and may even result in an unstable closed loop.

The dream transfer functions for any control system are $T(q) \equiv I$, $Q(q) \equiv 0$ and $S(q) \equiv 0$. But the expressions of the ideal controllers – (6), (8) and (10) – clearly show that it is impossible to achieve any of these specifications, since they all imply nonexistence of the ideal controller. Accordingly, the standard specifications in classical control theory are that such perfect performance is achieved only at some frequencies, which are chosen as the ones that are present in the spectrum of the reference and/or the disturbances in the real system under control. Still, one might be tempted to specify these as the desired transfer functions and allow the design procedure to return the transfer functions that are as close as possible to this wildly optimistic specification. This is a naïve approach that is bound for failure in most cases, as has been shown abundantly in the literature (e.g. Bazanella, Campestrini et al., 2012, Sec 2.4). Specifications must be made, whatever the method to be used for the design, that are not too far from what is possible to achieve for the given plant and within the given controller class.

The issue of picking a good reference model, satisfying this "reasonable specification" requirement, has received considerable attention previously, starting in the old days of adaptive control theory (Lee et al., 1993). It has been detailed in the DD control literature (Bazanella, Campestrini et al., 2012; Bazanella et al., 2008), and some recent advances have been made to make this choice fully systematic (Breschi & Formentin, 2021; de Jong et al., 2023; Gonçalves da Silva et al., 2019; Kergus et al., 2019), but it is still not a closed subject.

*Output following.* An additional, equally important, performance criterion evaluates the difference between the output and its desired value $y_d(t)$:

$$\begin{aligned} \varrho^{T} &= \arg\min\ J^{T}(\varrho) \\ J^{T}(\varrho) &\triangleq \bar{E}[y_d(t) - y(t,\varrho)]^2. \end{aligned} \tag{11}$$

This looks very much like $J^{MR}$, but here we have $y(t,\varrho)$ instead of $T(q,\varrho)r(t)$. In the absence of disturbances these are the same, and indeed model reference control is often defined as in (11) — most notably in the traditional adaptive control literature. In general, however, these are different criteria and their solutions are of different nature. The performance criterion $J^T(\varrho)$ does not admit an ideal controller in the same sense as the other criteria, since its universal optimizer depends on the particular signals $r(t)$, $d(t)$ and $v(t)$.

### 2.3. Reference models

In a model reference control design the main task to be performed by the designer is the choice of the reference model. Though in many cases this is an easy task, in many others it must take into considerations various aspects and may become rather involved. The choice of the reference model has been amply discussed for SISO systems in Bazanella, Campestrini et al. (2012) and Breschi and Formentin (2021) and for MIMO systems in Gonçalves da Silva et al. (2019) and Huff et al. (2023).

#### 2.3.1. Basic guidelines

In classical control theory and practice, steady-state performance specifications come in the form of steady-state tracking errors in response to specific reference and/or disturbance signals. For SISO systems these specifications are formally written as $|T_d(e^{j\Omega}) - 1| < \epsilon$ for reference tracking, $|S_d(e^{j\Omega})| < \epsilon$ for output disturbance rejection and $|Q_d(e^{j\Omega})| < \epsilon$ for load rejection. In these specifications $\Omega$ is the frequency to be tracked or rejected and $\epsilon$ is the specified tolerance. The most common case in practice, and accordingly in the literature, is the requirement of zero steady-state error for piece-wise constant signals, which corresponds to $\Omega = 0$ and $\epsilon = 0$ and results in $T_d(1) = 1$, $S_d(1) = 0$ and $Q_d(1) = 0$.

Specifications of transient response usually come in the form of settling time, rising time and maximum overshoot to a step reference. These are easily translated into pole location(s) of the reference model. As an illustration, say the specifications are zero steady-state tracking error for piece-wise constant references and disturbances, with a settling time of 10 samples and no overshoot. Then the following transfer function is the simplest solution:

$$T_d(q) = \frac{0.33}{q - 0.67}.$$

Besides transient and steady-state performance, robustness can also be encoded into the reference model, though this has been very little explored to date (Fiorio et al., 2023; Procházka et al., 2005).

For multivariable systems, the diagonal elements of the transfer matrix $T_d(q)$ can be chosen according to the same principles, whereas the definition of the off-diagonal elements is less clear. One possibility is to define them as zero, which amounts to specifying that the various I/O SISO pairs in the system must be decoupled. This is a valid solution, though often too ambitious, depending on the controller class. Detailed guidelines for the choice of the reference model in MIMO systems are given in Gonçalves da Silva et al. (2019).

These guidelines for defining the reference model are very principled and in many cases enough to achieve a good design. But there are other aspects involved in this choice, and in many other cases more sophisticated considerations have to be made. A poor choice of the reference model may result in a closed-loop performance that bears no resemblance to the one specified and may even be unstable. As mentioned before in this paper, one of the most important aspects to take into account is that it is pointless to specify a performance that is far from what is possible to achieve with the given controller structure; doing so usually results in a closed-loop performance that has little – if any – resemblance to the specifications.

#### 2.3.2. Flexible reference model

The selection of the poles of the reference model is quite intuitive, but the same cannot be said about the zeros. On one hand their relation to the performance is not as clear-cut as that of the poles. On the other hand, since zeros are not changed by feedback, the zeros of the plant will appear as poles of the ideal controller. This is undesired in any case, as it will result in increasing the order of the ideal controller, thus making it harder to be approximated by any given controller class. But it is particularly troublesome for the control of nonminimum phase plants, for in these cases the ideal controller would result in an internally unstable closed-loop system.

It is easy to see in Eq. (6) that this issue would be solved if the zeros of the plant were included as zeros of the reference model. But this in principle would require a model of the plant to be known a priori. The solution for this puzzle is the use of a *flexible reference model*, which in the SISO case is written as:

$$T_d(q) = T_d(q, \eta) = \eta^T \vartheta(q)$$

where $\vartheta(q)$ is a vector of transfer functions and the vector $\eta$ contains an additional set of decision variables to be estimated along with the controller parameters $\rho$. By leaving these degrees of freedom in the numerator of the reference model the parameter $\eta$ will be adjusted so that some zeros of the plant will also be zeros of the reference model, thus avoiding their cancellation by the controller. Performing this simultaneous optimization of $\eta$ and $\rho$ requires different stratagems in each control design method.

For multivariable systems, the parameter $\eta$ must be adjusted to estimate not only the zeros themselves, but also their directions. One simple way around this issue is to define a diagonal reference model, for then each zero appears in all directions. This is restrictive, however, because the same performance is specified for all outputs. Estimating nondiagonal flexible reference models is a tricky problem, that so far has been solved only for one particular design method — OCI, described in Section 4.1.

Flexible reference models are convenient in many cases, but there are two situations in which their use is crucial. One is in the control of nonminimum phase plants and another one is for an appropriate choice of the reference model in load disturbance problems. These two cases will be discussed in some detail later in this paper.

### 2.4. Canonical controllers

Data-driven classical control is mostly about setting the parameters of a fixed structure controller, usually of reduced order. This is one major distinction from the traditional adaptive control literature, the other being that each update in the controller parameters is based on a sufficiently large batch of data, so that asymptotic statistical analysis can be applied to the treatment of stochastic inputs. In the classical I/O approach for control design one starts from a specification of what class(es) of disturbances must be rejected and what class(es) of references must be followed. These steady-state specifications typically determine the modes – i.e. the poles with unitary module – of the controller.

Once the poles have been fixed in order to satisfy the steady-state specifications, a simple root-locus argument convinces us that the controller should have as many zeros as possible — so that less poles can escape to infinity when the loop is closed (Franklin et al., 2006, Ch. 5), (Ogata, 2009, Ch. 6). The maximum number of zeros possible is the number of poles, for otherwise the controller would not be causal. With this, the controller structure is determined.

This is the most common reasoning behind the definition of a controller class, but not the only one. In many cases, such a controller structure is not enough to provide appropriate performance, so additional blocks are added to the structure. On the other hand, controllers may be meant just to stabilize unstable systems, or to improve robustness. This will add freedom to the controller's structure, as the poles are not required to be at the specific locations that represent the model of the references/disturbances.

The most standard controller class is the SISO Proportional–Integral–Derivative (PID) controller

$$C(q, \rho) = k_p + k_i \frac{q}{q-1} + k_d \frac{q-1}{q-\beta} \tag{12}$$

with $\rho = [k_p \; k_i \; k_d \; \beta]^T$. One can also reparametrize this same class of controllers as

$$C(q, \rho) = \frac{\rho_1 q^2 + \rho_2 q + \rho_3}{(q-1)(q-\beta)}. \tag{13}$$

Besides, there are many variations, and PID controllers of different standard forms can be found; let us mention a couple of them. Often one fixes the value of $\beta$, which allows to write the parametrization linearly:

$$C(q, \rho) = \begin{bmatrix} 1 & \frac{q}{q-1} & \frac{q-1}{q-\beta} \end{bmatrix} \rho \tag{14}$$

with $\rho = [k_p \; k_i \; k_d]^T$. The integrator may be implemented with a delay, as $\frac{1}{q-1}$ instead of $\frac{q}{q-1}$, due to software and/or hardware constraints. And the derivative may be approximated by the difference between samples that are separated by various – say $d$ – time instants, as $\frac{1-q^{-d}}{1-\beta q^{-1}}$, to reduce the effect of noise.

A MIMO PID controller can be written as

$$C(q,\rho)=\begin{bmatrix} C_{11}(q,\rho_{11}) & C_{12}(q,\rho_{12}) & \cdots & C_{1n}(q,\rho_{1n}) \\ \vdots & \vdots & \ddots & \vdots \\ C_{n1}(q,\rho_{n1}) & C_{n2}(q,\rho_{n2}) & \cdots & C_{nn}(q,\rho_{nn}) \end{bmatrix}, \tag{15}$$

where each $C_{ij}(q,\rho_{ij})$ is a SISO PID.

One major reason why PID controllers are so important is that in traditional applications the majority of references to be tracked and disturbances to be rejected are piece-wise constant, and PID's provide perfect tracking in steady-state for those. But there are plenty of applications that involve other classes of signals and thus other classes of controllers.

In many applications, periodical references must be tracked – like in the control of Uninterruptible Power Sources (UPSs) (Pereira et al., 2014; Teodorescu et al., 2006) – and/or periodical disturbances must be rejected, like in the control of structural vibrations (Moheimani & Vautier, 2005). To achieve these control objectives a *resonant controller* is recommended:

$$C(q,\rho)=\frac{\rho_2 q^2+\rho_1 q+\rho_0}{q^2-2\cos(\Omega)q+1} \tag{16}$$

where $\Omega$ is the sampled frequency of the reference/disturbance.

The resonant controller (16) allows to track and reject sinusoidal signals of frequency $\Omega$. In some applications, like in high-performance UPSs that will be presented in Section 6, various sinusoidal harmonics must be rejected (Lorenzini et al., 2022); then a proportional multi-resonant controller is asked for:

$$C(q,\rho)=k_p+\sum_{j=1}^{l}\frac{\rho_{j1}q+\rho_{j2}}{(q^2-2\cos(\Omega_j)q+1)}. \tag{17}$$

Multi-resonant controllers actually can deal with arbitrary periodical signals. In most of these cases, the parameters in the controller's denominator are known a priori and fixed, thus the controller is linearly parametrized.

When the task of the controller is just stabilization, or improvement of some performance criterion – transient performance, robustness margin – with respect to an existing closed-loop system, the traditional lead–lag controllers can be applied:

$$C(q,\rho)=K\ \Pi_{i=0}^{n}\frac{q-\zeta_i}{q-\eta_i}. \tag{18}$$

Linear parametrizations are very convenient from a computational point of view and also facilitate the theoretical analysis in various aspects. To take advantage of these benefits even when the natural controller structure is not linearly parametrized, one can approximate this structure by a truncated Taylor series. This results in a controller with the following structure:

$$C(q,\rho)=\sum_{i=0}^{p-1}\rho_i q^{-i} \tag{19}$$

which can be recognized as a Finite Impulse Response (FIR) transfer function. This will usually require a large number of parameters to approximate even a rational function of low order, but allows to take advantage of the features of linear parametrization. Only on a case by case basis can one tell whether the advantages of this approximation will beat its disadvantages.

Nonlinear controllers will be discussed in Section 7. For those, a simple, yet powerful standard structure is that of a Wiener-type system:

$$u(t)=\varphi(C(q,\rho)e(t)) \tag{20}$$

where $\varphi(\cdot)\ :\ \mathbb{R}^n\rightarrow\mathbb{R}^n$ is a nonlinear map and $C(q,\rho)$ can be any standard linear structure presented previously.

A more general nonlinear parametrization is obtained with a standard linearly parametrized structure like

$$u(t)=\sum_{i=0}^{p-1}\rho_i\phi_i(\mathcal{Z}(t)) \tag{21}$$

where $\mathcal{Z}(t)$ is a set of measurements of $e(\cdot)$ and $u(\cdot)$ available at time $t$.

Another kind of controllers that can be used to control nonlinear systems is that of Linear Parameter-Varying (LPV) controllers. Such controllers can be seen as an extension of an LTI controller whose parameters are functions of a scheduling variable $\sigma(t)$. In this case, the control law can be written as

$$\begin{aligned} u(t)\ =\ &\sum_{i=0}^{m-1}a_i(\sigma(t-i))u(t-i)+ \\ &\sum_{i=0}^{n-1}b_i(\sigma(t-i))e(t-i), \end{aligned} \tag{22}$$

where the functions $a_i(\cdot)$ and $b_i(\cdot)$ are defined as linear combinations of the scheduling variable or some function of it.

### 2.5. The data-driven solution to classical control

Model-based control in the model reference paradigm consists in starting from a plant's model and optimize the desired cost. When the controller is of full-order this amounts to apply the formula (6) or the one corresponding to the desired objective. When the controller is not of full order, there are many ways to approximate the reference model, and classical control methods can also be seen under this light.

In the data-driven approach, a model is not available but the designer can collect a batch of data from the process (1). These data consist of measurements of the input and output of the process along a certain window of time of size $N$:

$$Z^N=[u(1),\ y(1),\dots,u(N),\ y(N)].$$

They may also include additional measurements of reference or disturbance signals, or of scheduling variables in the linear parameter varying context. Then the parameters of a controller in a predefined class will be estimated from these data, in order to achieve the desired closed-loop response.

The costs (5), (7) or (9) are functions of the process model, which is unknown. The most straightforward DD approach around this issue is to use the data to identify a plant model and then proceed exactly as in model-based design. This two steps procedure is usually referred to as Indirect Data-Driven design. From the point of view of control design, Indirect DD design is the same as model-based design, the difference being the source of the model, which is now obtained only from data.

Direct Data-Driven control design, on the other hand, seeks to optimize the desired cost "directly" using data, without any intermediate step. In so doing, it is not necessary to derive a model for the plant, and not even is it useful to do so, from the point of view of the control design. This changes the design procedure completely, and also makes a fundamental difference in terms of the solution's properties and thus the closed-loop performance that can be obtained.

There are two conceptually different approaches to this direct data-driven design problem. One approach is to attack the optimization problem as such and calculate from data the quantities necessary to perform the optimization. The other one is to translate the design problem into the identification of the ideal controller. The first approach tends to yield iterative methods, in which each iteration requires the collection of new data. The methods arising from the second approach usually require only one batch of data, collected in a single experiment, which motivates calling them "one-shot" methods. Each one of these approaches is the subject of a Section in the sequel.

## 3. Optimization-based design

The most straight approach to direct data-driven design is to attack the optimization problem as is and apply an iterative optimization method:

$$\varrho^{i+1} = \varrho^{i} - \gamma^{i} R_i \left.\frac{\partial J(\varrho)}{\partial \varrho}\right|_{\varrho^i}$$

where at each iteration $i = 0, 1, \ldots,$ $\gamma^i \in \mathbb{R}^+$ is the step size and $R_i$ is a positive definite matrix that defines the search direction. Most typically the direction matrix $R_i$ is the identity matrix, which corresponds to gradient descent optimization, or some approximation of the inverse of the Hessian $\frac{\partial^2 J(\varrho)}{\partial \varrho^2}$. This is conceptually very much like the direct adaptive control methods that were already considered (for instance by Krstic et al. (1995)) "traditional" at the time when these iterative data-driven design methods first appeared — specifically, Model Reference Adaptive Control (MRAC), Self-Tuning Regulators and the like (Åström & Wittenmark, 1995; Goodwin & Sin, 1984; Narendra & Annaswamy, 2005). But here the problem setting and the assumptions are different in various aspects, most notably the following:

1. a controller class, usually of reduced order, is given a priori;
2. each one of the data-based estimates of the gradient that are necessary for the optimization is obtained from a batch of data that is large enough to allow asymptotic statistical analysis;
3. scant attention is given to the certainty equivalence principle, which was fundamental in the traditional adaptive control methods;
4. the statistical properties of the algorithms and of the resulting closed-loop system play the central role in the analysis, as it does in identification theory;
5. no nonlinear effects are introduced in the control loop by the changes in the values of the controller parameters.

Items 3, 4 and 5 are partly consequences of the first two. Given their similarities, perhaps it would be justified to consider these methods as part of the traditional adaptive control framework. But this is not how they have been presented and it is not how they are perceived.

In order to guarantee convergence of the optimization in a stochastic framework, the gradient of the cost function must be estimated without bias, as shown in the Robbins–Monro theorem. How to obtain such unbiased estimates directly from data becomes then the central issue. A solution for this problem was presented in Hjalmarsson et al. (1994) for the output following performance criterion (11) and the resulting method was named Iterative Feedback Tuning (IFT) by its authors. That paper and that method may be considered to mark the dawn of DD classical control, and its most fundamental reference is probably (Hjalmarsson et al., 1998). A similar solution was presented in Kammer et al. (2000) for the output disturbance criterion (7) and the resulting method was named Frequency Domain Tuning (FDT) by its authors.

IFT has found many applications along the years, as described in Hjalmarsson et al. (1998) and Kissling et al. (2009) for instance, and continues to find them. Theory on IFT also developed, resulting in improved convergence properties, improved closed-loop performance and a wider range of application possibilities. An extension for multivariable plants was presented in Jansson and Hjalmarsson (2004). The cost function to be minimized (11) is not convex and usually presents many local minima and maxima, so convergence to the globally optimal controller is a major issue. Such convergence can be enforced in most cases by shaping the cost function so that it does not have local minima along the way of the optimization. This can be achieved in various ways (Bazanella et al., 2008): by experiment design, by proper selection of the data (Eckhard & Bazanella, 2012a) or by cautious control (Kammer, 2005; Lee et al., 1993).

The choice of the step sizes $\gamma_i$ is critical for obtaining convergence, but also for improving the speed of this convergence. In the gradient algorithm – that is, $R_i = I$ – a sufficient condition for convergence is that the sequence $\gamma_i$ forms a nonconvergent series and its square forms a convergent series, that is:

$$\sum_{i=1}^{\infty} \gamma_i = \infty \quad \sum_{i=1}^{\infty} \gamma_i^2 < \infty.$$

One "famous" sequence satisfying these conditions is the harmonic sequence $\gamma_i = \frac{\gamma_1}{i}$ for some $\gamma_1$ and it has been the standard choice of steps for a while. Much faster convergence can be obtained with the step sequence proposed in Eckhard and Bazanella (2010, 2012a, 2012b). Fast convergence is an important issue, since each iteration requires collecting new data. The identification-based methods, to be presented in the next Section, usually require only one batch of data to achieve the optimal controller. On the other hand, they may be riskier exactly because of this feature, since they provide a new controller from scratch instead of just updating a controller that is already operating in a stable manner.

## 4. Identification-based design

A conceptually different approach to Direct DD control is to frame the design as the identification of the ideal controller. There are various ways to reformulate the design problem in this way, and once the problem has been thus reformulated the identification can be performed by either the Prediction Error approach or the Correlation approach. Each such formulation results in a different method, who are known by their acronyms: VRFT (for Virtual Reference Feedback Tuning), CbT (Correlation-based Tuning), OCI (Optimal Controller Identification), FRIT (Fictitious Reference Iterative Tuning). All these methods optimize the model reference criterion, but they also allow to optimize the output disturbance performance by applying them to the desired $T_d(q) = 1 - S_d(q)$. Load disturbance, on the other hand, requires special treatment and is dealt with by VDFT (Virtual Disturbance Feedback Tuning), DCbT (Disturbance Correlation-based Tuning) and OCI-D (Optimal Controller Identification for Disturbance Rejection).

### 4.1. Prediction error identification

The derivation of the Optimal Controller Identification (OCI) method starts with the problem of Prediction Error identification of the plant, which consists in defining the optimal one-step-ahead predictor

$$\hat{y}^*(t) = H^{-1}(q)G(q)u(t) + (I - H^{-1}(q))y(t). \tag{23}$$

Then the plant is rewritten in terms of the ideal controller, isolating the plant's model in (6):

$$G(q) = C_d^{MR}(q)^{-1} T_d(q)(I - T_d(q))^{-1} \tag{24}$$

and thus obtaining the following expression for the optimal predictor:

$$\begin{aligned}\hat{y}^*(t) &= H^{-1}(q)C_d^{MR}(q)^{-1}T_d(q)(I - T_d(q))^{-1}u(t) \\ &\quad + (I - H^{-1}(q))y(t), \\ &= H^{-1}(q)C_d^{MR}(q)^{-1}\tilde{u}(t) + (I - H^{-1}(q))y(t).\end{aligned} \tag{25}$$

By replacing in (25) the ideal controller and the noise filter by parametrized models for the controller $C(q, \varrho)$ and for the noise filter $H(q, \theta)$ one gets

$$\hat{y}(t, \xi) = H^{-1}(q, \theta)C(q, \varrho)^{-1}\tilde{u}(t) + (I - H^{-1}(q, \theta))y(t), \tag{26}$$

with $\xi = [\varrho^T \ \theta^T]^T$, where $\varrho \in \mathbb{R}^p$ is the controller parameters' vector, and $\theta \in \mathbb{R}^t$ is a vector with additional noise model parameters. Then optimization of the mean squared prediction error

$$J^{OCI}(\xi) = \frac{1}{N}\sum_{t=1}^{N}(y(t) - \hat{y}(t, \xi))^2 \tag{27}$$

is a standard PE identification problem, with the filtered input $\tilde{u}(t)$ as the input data and the inverse of the ideal controller as the system to be identified. The identification of the ideal controller inherits the statistical efficiency features of the prediction error method and OCI has indeed shown nice statistical properties in applications. One evidence of this performance is that it has won a competition of MIMO PID tuning organized at an IFAC Conference (Huff et al., 2018). OCI was first proposed in Campestrini et al. (2017) for SISO plants and its MIMO version has been presented in Huff et al. (2019). A filter can be added to (27) to reduce the bias of the estimation of the ideal controller when the controller class is underparameterized (Varriale da Silva & Campestrini, 2022).

Another feature that is inherited from prediction error identification is the fact that the function to be minimized (27) is not convex. Moreover, the model of the to-be-identified system $C(q,\rho)^{-1}$ is not in any standard identification form, so standard identification packages are not always appropriate. In Huff et al. (2019), a dedicated algorithm was presented, based on a combination of the steepest descent and the Newton–Raphson methods. But for most SISO cases standard identification packages (e.g. the Matlab toolboxes `ident`, `sib` or `unit`) can be successfully applied.

#### 4.1.1. Dealing with disturbances

The OCI method was adapted to solve the load rejection problem for SISO systems in Scheid Filho et al. (2021). The same idea of writing the plant model as a function of the reference model $Q_d(q)$ and of the ideal controller $C_d^{ID}(q)$ in this case results in

$$G(q) = \frac{Q_d(q)}{1 - C_d^{ID}(q) Q_d(q)}. \tag{28}$$

The choice of the numerator of $Q_d(q)$ is critical to ensure the causality of $C_d^{ID}(q)$ and internal stability of the closed-loop system. Besides, it is harder to come up with a reasonable numerator than in the reference tracking criterion, which motivates the use of a flexible reference model $Q_d(q,\eta)$. Considering the controller to be identified as $C(q,\rho)$, the predictor is given by

$$\begin{aligned}\hat{y}(t,\xi) \;=\; & H^{-1}(q,\theta)\frac{Q_d(q,\eta)}{1 - C(q,\rho)Q_d(q,\eta)}u(t) \\ & + (I - H^{-1}(q,\theta))y(t),\end{aligned} \tag{29}$$

with $\xi = \begin{bmatrix}\rho^T & \eta^T & \theta^T\end{bmatrix}^T$, where $\eta \in \mathbb{R}^m$ is the reference model parameters' vector. In the disturbance rejection problem with a flexible reference model, standard identification packages cannot be used, even in the SISO case.

### 4.2. Direct vs indirect design

Most of data-driven classical control theory revolves around direct design methods, that is, methods on which no plant model is obtained as an intermediate step. Still, the OCI method provides flexibility that can be used to shift between direct and indirect design. Indeed, a quick look at (6) reveals that part of the ideal controller is known, as it is determined by the reference model; let this known part be called $C^K(q)$:

$$C^K(q) = T_d(q)(I - T_d(q))^{-1}. \tag{30}$$

So, one can use this knowledge and fix part of the controller — say $C^F(q)$, write the controller as $C(q,\rho) = C^F(q)C^I(q,\rho)$ and estimate only the remaining part $C^I(q,\rho)$. The fixed part $C^F(q)$ can be any piece of $C^K(q)$, leaving the rest to be estimated from the data. If the choice $C^F(q) = C^K(q)$ is made, then what is left to identify is only the plant's model (rather its inverse). This amounts to indirect DD design, as the data are being used to estimate a plant's model to insert into (6). On the other hand, one can completely ignore this knowledge and leave all the controller's parameters free to be estimated from data, which would be the purest direct DD design. But there are also intermediate choices, in which one fixes any chosen part of $C^K(q)$, like, for instance, choosing $C^F(q) = \frac{1}{q-1}$, which is a standard choice.

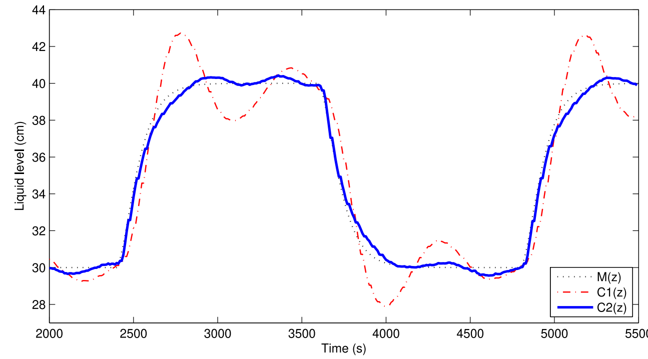


**Fig. 2.** Closed-loop responses show the big difference in performance obtained in a real liquid processing plant with Indirect DD design (red dash–dot line) and Direct DD design (continuous blue line). The specified performance is also shown (dotted line).

The larger the dimension of $C^F(q)$, the less parameters there are to be estimated from the same data. On the other hand, the following is known from standard identification theory. Assuming that a full-order model is being used for the controller, the controller estimation is unbiased and there is only variance error. Moreover, the variance error grows with the number of parameters to be estimated and decreases with the length of the data set used for the design. But if a reduced-order controller is to be estimated, then there is bias error and the total error will asymptotically (in the number $N$ of data) be due only to bias.

This leads to the following conclusions. When variance is the main source of error, then the best results will be obtained with the estimation of less parameters — that is, $C^F(q) = C^K(q)$ is likely to be the best choice. This is the case if a full-order controller is used and/or few data are available (small $N$). When bias error dominates, then it is better to leave more parameters free to shape this bias to the closed-loop performance criterion. If indirect DD design $C^F(q) = C^K(q)$ were used in this situation, then the estimation would shape the bias to best estimate the plant model under a prediction error criterion, disregarding the real objective, which is closed-loop performance. Thus when a reduced-order controller is used and there is a reasonable amount of data (large N), the more parameters are left free the better. In many cases, in simulation and in practice, choosing the fixed part of the controller with only the modes necessary to obtain the specified steady-state performance (e.g. $C^F(q) = \frac{1}{q-1}$ when an integrator is required) yields the best results. It is quite relevant that similar conclusions were obtained in different control design frameworks (Dörfler et al., 2022; Formentin et al., 2014; Krishnan & Pasqualetti, 2021).

Each different choice of $C^F(q)$ sits at a particular place along the path between direct design and indirect certainty equivalence design. For any given application the difference in performance that is obtained for each choice can be very significant. Often very good closed-loop performance can be obtained from a small set of data – a dozen, even – if the best choice is made, whereas a poor choice may result in very poor performance with the same data. This was illustrated in the experimental study presented in Campestrini et al. (2017), whose final result is replicated in Fig. 2 for visual illustration. It shows the closed-loop performance in the level control of a tank obtained with two different controllers designed from the same data, one by indirect design – $C^F(q) = C^K(q)$ – and the other one by direct design $C^F(q) = \frac{1}{q-1}$. In this experimental setting the controller is of low order and there are abundant data, two features that favor the direct design, as discussed above.

### 4.3. The Virtual Reference Approach

The Virtual Reference Feedback Tuning (VRFT) method was first presented in Campi et al. (2002), though the concept was already present in Guardabassi and Savaresi (2000). This first proposal was for the design of SISO plants, but MIMO versions appeared later (Campestrini et al., 2016; Formentin et al., 2012; Nakamoto, 2004; Saeki, 2021). An open-source Python Toolbox is described in Boeira and Eckhard (2020) and a Matlab Toolbox is described in Carè et al. (2019). Further extensions and improvements have been proposed and others are still appearing in the literature.

The central idea is to collect I/O data from the plant to be controlled and pretend that these data were collected in a closed-loop experiment with the ideal controller operating in the loop. If this were the case, then the input and output data collected from the plant would be respectively the output and input of the ideal controller, which can thus be identified from these data.

All manipulations performed, the design boils down to minimizing the function:

$$J^{VR}(\varrho) = \sum_{t=1}^{N} \| F(q)[u(t) - C(q,\varrho)(T_d^{-1}(q) - I)y(t)]\|_2^2, \tag{31}$$

where $F(q)$ is some filter. The cost function $J^{VR}(\cdot)$ has the same global minimum as (5) when the matching condition (Assumption 1) is satisfied and the collected data are noise-free. When this is not the case, the filter $F(q)$ can be tuned to approximate the minima of the two functions.

If $C(q,\varrho)$ is linearly parametrized, i.e., each element of the controller matrix can be written as $c_{ij}(q) = \rho_{ij}^T \bar{C}_{ij}(q)$, then $J^{VR}(\varrho)$ is quadratic in the parameters and a closed-form solution to the optimization problem is obtained via least-squares (Campestrini et al., 2016). This is a very nice feature, particularly when one compares with the other methods, which require the optimization of nonconvex functions. Since linearly parametrized controllers are extremely common in the industry, this feature of the VRFT method partially explains its success.

In this conceptual formulation, a biased estimate of the optimal parameters is obtained when the collected signals are noisy, so an instrumental variable approach is often preferred. This removes the bias at the cost of increased variance, as usual in instrumental variable techniques. The consistency and efficiency of such methods is analyzed in Bazanella, Campestrini et al. (2012) and in van Heusden, Karimi and Söderström (2011). Alternatively, the statistical properties of the VRFT solution can be significantly improved in the case of linearly parametrized controllers by means of Total Least Squares (Garcia & Bazanella, 2020a, 2022) or by regularization (Boeira & Eckhard, 2023; Formentin & Karimi, 2014; Rallo et al., 2016). On the other hand, it has been found previously that careful selection of the appropriate data (Carrette et al., 1996; Holcomb & Bitmead, 2017) improves the statistical performance in identification; following this lead, a method for selecting data for VRFT has been presented (Garcia & Bazanella, 2019, 2022). These ideas – data selection, regularization and Total Least Squares – improve significantly the statistical properties of VRFT at a low computational cost, so they have great potential to further VRFT's application. In any case, they are of paramount importance to the ongoing efforts to apply the virtual reference concept to the design of nonlinear controllers, as will be discussed in Section 7.

The method Fictitious Reference Iterative Tuning (FRIT), proposed in Kaneko (2013), is based on the same idea of a virtual experiment, but the formulation is slightly different, resulting in a slightly different cost function to minimize:

$$J^{FR}(\varrho) = \sum_{t=1}^{N} \| [T_d(q)C(q,\varrho)^{-1}u(t) - (1 - T_d(q))y(t)]\|_2^2.$$

The argument inside the brackets in $J^{FR}(\varrho)$ is the same as in $J^{VR}(\varrho)$ multiplied by the factor $T_d(q)C(q,\varrho)^{-1}$. The implications of this difference are discussed in detail in Kaneko (2013). One of them is that the decision variable $\varrho$ appears in the denominator of the cost $J^{FR}(\varrho)$, which requires some iterative optimization, from which the method gets its name — note, however, that the method is not iterative in the sense adopted in this paper; it is actually a one-shot method. The identification of the inverse of the controller using the virtual framework was also presented in Sala and Esparza (2005), where the advantages and disadvantages of such solution are discussed.

#### 4.3.1. Dealing with disturbances

The Virtual Disturbance Feedback Tuning (VDFT) is an application of the virtual reference concept to the load rejection problem (Eckhard et al., 2018). Considering a desired response for the disturbance $Q_d(q)$, VDFT is quadratic in the controller parameters and solved through least squares. Besides, as in OCI-D, the reference model $Q_d(q)$ can be made flexible, and the VDFT solution is written in Bordignon and Campestrini (2018a) as the minimization of:

$$J^{VD}(\varrho,\eta) = \sum_{t=1}^{N} \| K(q)\big[Q_d(q,\eta)\big(u(t) + C(q,\varrho)y(t)\big) - y(t)\big]\|^2, \tag{32}$$

where $K(q)$ is an additional filter. When both the controller and the reference model are linearly parametrized, (32) is bilinear, which is solved using Sequential Least Squares (Ljung, 1999).

The VDFT method using a flexible reference model was applied in a hierarchical control structure in Bordignon and Campestrini (2018b), where a PID controller was designed to enhance the disturbance rejection response and an MPC controller was designed for setpoint changing. However, no model for the plant was estimated for the MPC design: it uses as model an estimate of the closed-loop response given by $\hat{T}(q) = C(q,\hat{\rho})Q_d(q,\hat{\eta})$.

### 4.4. Correlation based Tuning

The *Correlation based Tuning* (CbT) method emerged in the work of Karimi et al. (2002) where the idea was to use a correlation approach to estimate the ideal controller. In the first papers related to this method, iterative algorithms were used, in a very similar way to IFT, but after some time the authors realized that a "one-shot" formulation was possible (Karimi et al., 2007).

The main idea of the CbT method is to introduce the following error function

$$\varepsilon(t,\varrho) = T_d(q)u(t) - (1 - T_d(q))C(q,\varrho)y(t) \tag{33}$$

which, in absence of noise, is identically null if $C(q,\varrho) = C_d^{MR}(q)$. When the collected signals contain noise but the ideal controller is still in the loop, the signal $\varepsilon(t,\varrho)$ is filtered noise and is uncorrelated with any external deterministic signals $\zeta(t)$. Hence the correlation

$$f(\varrho) = \frac{1}{N}\sum_{t=1}^{N} \zeta(t)\varepsilon(t,\varrho) \tag{34}$$

between the external signal $\zeta(t)$ (which may contain samples of either $u(t)$ or $r(t)$) and the error functions should be zero when $C(q,\varrho) = C_d^{MR}(q)$. The function $\zeta(t)$ has $2l+1$ components and is formed as

$$\zeta(t) = [u(t-l) \ \dots \ u(t-1)\, u(t)\, u(t+1) \ \dots \ u(t+l)] \tag{35}$$

where $u(t)$ must be replaced by $r(t)$ if the experiment is run in closed-loop.

The Correlation based Tuning method proposes the minimization of the norm of the correlation function $f(\varrho)$ as a criterion to estimate the controller in an optimization problem, which minimizes

$$J^{CbT}(\varrho) = f^T(\varrho) f(\varrho). \tag{36}$$

When the controller is linearly parametrized this optimization problem is quadratic and has a closed-form solution. The method was also extended to use a flexible reference model, where $T_d(q)$ in Eq. (33) is replaced by $T_d(q,\eta)$. In this case, the optimization criterion is not quadratic but it is efficiently minimized using Sequential Least Squares.

#### 4.4.1. Disturbance Correlation based Tuning

The same idea has been used to tune controllers aiming to reduce the effect of disturbances on the closed loop (da Silva & Eckhard, 2019; da Silva & Eckhard, 2023). The resulting method has been called Disturbance Correlation based Tuning (DCbT) and uses the following error function

$$\varepsilon_d(t,\rho) = Q_d(q)u(t) - (1 - Q_d(q)C(q,\rho))y(t) \tag{37}$$

which is null if the signals contain no noise and $C(q,\rho) = C_d^{ID}(q)$. The proposed method follows the same idea of CbT, minimizing the correlation between the error function $\varepsilon_d(t)$ and external signals $\zeta(t)$ which may be composed by input signals $u(t)$ in open loop or external signals in closed loop as $r(t)$ or $d(t)$ if it is measured on the experiment. Again, the optimization problem is quadratic and has a closed solution if a fixed reference model is chosen with a linearly parametrized controller. If a flexible reference model is chosen, the optimization problem is solved using Sequential Least Squares.

An extension of the method to deal with MIMO processes was proposed in da Silva and Eckhard (2020) and an application of the method to a thermal process is described in da Silva and Eckhard (2021).

## 5. Miscellaneous

### 5.1. Nonminimum-phase plants

Dealing with nonminimum phase (NMP) plants is a major problem in model reference design, because in these cases the ideal controller is unstable, and its insertion in the loop would result in an internally unstable system. A problem whose unconstrained optimal solution is one that results in instability is not well-posed, and this issue is particularly troublesome in the identification-based formulation. Accordingly, the original formulations of the design methods often fail when applied to NMP plants — as any model reference design method would, even model-based. It is also worth noticing that whereas in continuous-time the occurrence of NMP systems is an uncommon exception, in discrete-time they are typical. Indeed, it is well known that periodical sampling "creates" new zeros, and these tend to be outside the unit circle in the complex plane, except for some "quite restrictive conditions" — this quote is from Åström et al. (1984).

A solution for this problem has been presented that relies on a flexible reference model. As seen in Section 2.3.2, the flexible reference model allows to identify the zero(s) of the plant along with the controller parameters. Then the NMP zeros, if any, are automatically included in the flexible reference model, avoiding the internal stability issue. This same concept can be applied to the various design methods, but for each one this application presents specific challenges to arrive at appropriate design solutions. Such solutions for the different methods have been presented, extending considerably their range of potential applications: IFT in Lecchini and Gevers (2002), VRFT in Campestrini et al. (2011) and CbT in da Silva and Eckhard (2019). These solutions were initially proposed for SISO plants only, and their adaptation for MIMO plants still presented new challenges, mostly related to the fact the zeros also have directions to be identified. So, it was only recently that these challenges that were solved for VRFT in Gonçalves da Silva et al. (2018) and for OCI in Huff et al. (2023).

### 5.2. Certification

Most design methodologies, whether data-driven or not, do not rule out the possibility of placing a destabilizing controller in the loop (Dehghani et al., 2009). Depending on the noise distribution and on the signal-to-noise ratio, a controller designed from data can be expected to have a nonzero probability of resulting in an unstable closed loop. The same applies to model-based controllers with respect to the modeling errors.

Hence the importance of a procedure to certify that the controller that has been designed has zero, or at least negligible, probability of resulting in an unstable closed loop. Such a procedure is called *certification* of the controller. A data-driven control design method that incorporates certification in the design, providing asymptotic stability guarantees, was proposed in van Heusden, Karimi and Bonvin (2011), but at the cost of performance loss. More usual, and perhaps less prone to sacrifice performance for safety, is to consider certification as a procedure that is independent of the control design.

Data-driven controller certification thus consists in collecting and analyzing I/O data to verify whether or not the controller will guarantee closed-loop stability when applied to an unknown plant. The main role of certification is to ascertain closed-loop stability, but guarantees of worst case performance can also be pursued. The controller is said to be *certified* if the desired guarantees of closed-loop stability, and perhaps of performance, are obtained.

Early approaches to DD controller certification were based on the $\nu$-gap metric (Park & Bitmead, 2004; Vinnicombe, 1993), which measures the distance between two transfer functions. In these early works, the controller is certified if the distance between this controller and a known stabilizing controller is smaller than some $\nu$-gap based stability margin that can be estimated exclusively from data. This estimation relies on the estimation of cross-spectra between the signals collected in a special experiment performed in closed loop with a stabilizing controller. Later developments reduced the conservatism of these certification methods by introducing frequency-dependent scaling (da Silva & Eckhard, 2020; de Bruyne & Kammer, 1999; Holcomb & Bitmead, 2017; Kammer et al., 2000).

A different approach, which is not based on the $\nu$-gap metric but on the estimation of the $H_\infty$-norm of a different target transfer function, has appeared in Dehghani et al. (2009). In this approach, less conservative results were typically obtained, at the cost of a complex data collection mechanism. In Gonçalves da Silva et al. (2020) a much simpler procedure to obtain the same norm estimation has been presented, which does not require any special experiment and thus can be performed with routine closed-loop operating data. The $H_\infty$-norm estimation procedure proposed in Oomen et al. (2014), which involves the evaluation of Markov parameters, is capital to achieve this simplification.

All these methodologies are applicable for multivariable systems and experimental results have been given that show their applicability. However, they still have to undergo the scrutiny of time and of more applications, and there are still improvements to be made. On the other hand, the certification of nonlinear output feedback controllers seems to be a completely open subject, and one of even more complexity and relevance than for linear systems.

### 5.3. Data treatment

In all sorts of technological devices and environments, the availability of data and the capabilities for accessing and manipulating these data have nowadays become enormous. Contrasting to these new resources and the eagerness to use them is the fact that most of these data are poor in information content. In large batches of raw data the useful information is often hidden behind large amounts of noninformative data. Making decisions on the basis of such data in their raw and poorly informative form is likely to result in poor decisions.

Routine operating data of plants to be modeled and/or controlled are usually of this poorly informative kind. So, it is common practice in system identification and in DD control design to perform specific experiments in which an appropriate signal is applied to the process, so as to collect data that are rich in the desired information. However, in many cases performing such experiments may be a costly and inconvenient task; sometimes it is not even possible to do it. Moreover, in today's data-abundant reality it seems illogical to spend time and efforts to collect new data when one is deluged by the already existing data. Unearthing the information from behind the irrelevant data may be much more clever, practical and effective.

In system identification it has been shown (Carrette et al., 1996) that the least informative part of nonstationary data tends to increase the bias, so their removal improves the quality of the identification. In system identification theory, data informativity is determined by the regularity of the Information Matrix (Bazanella, Bombois et al., 2012; Gevers et al., 2009; Ljung, 1999). Accordingly, in Carrette et al. (1996) it is proposed to select segments of data such that the smallest singular value of the Information Matrix is large enough. These data will usually not be contiguous in time, which requires a multi-record identification approach, like in Holcomb and Bitmead (2017). These ideas have already been adapted to the design of identification-based DD control design methods. An automatic procedure to select the data to use in VRFT has been presented in Garcia and Bazanella (2019) and further tested in Garcia and Bazanella (2020b). This procedure has been combined with the use of Total Least Squares in Garcia and Bazanella (2022), resulting in significantly improved statistical properties of the closed-loop system. Yet, there is ample room for improvement and extension to more general situations and to other design methods, so that this is a widely open field, both for system identification and for data-driven control design.

### 5.4. Distributed control

The study of networks has become a major branch of control theory in the past two decades. Distributed control is used in networks where the centralized strategy is prohibitive due to various aspects, like a large dimension of the network and limitation of communication channels. It consists of the design of local controllers with limited (if any) communication among them, aiming at some global performance criterion. Both OCI and VRFT were extended to distributed control (Steentjes et al., 2020, 2021), where a structured reference model is defined and the ideal distributed controller is identified. In Steentjes et al. (2021), an illustrative example shows how OCI outperforms VRFT when signals are significantly corrupted with noise, because instrumental variables, despite providing an unbiased estimate, present higher variance than PEM.

## 6. Applications

The data-driven classical control design methods have been extensively tested experimentally, with many practical applications described in the literature. A large number of real-life applications have also been reported, in a wide variety of fields, of both design philosophies — identification-based and optimization-based. Examples of such applications involve, among many others, the control of prostheses (Previdi et al., 2004), automotive systems (Formentin et al., 2013, 2018), power converters (Nicoletti et al., 2019; Remes et al., 2019; Teodorescu et al., 2006), power system stabilizers (Bernado et al., 2020), wastewater treatment plants (Rojas et al., 2012), the chemical industry (Hjalmarsson et al., 1998), wafer scanners for integrated circuit production (Heertjes et al., 2016), particle accelerators (Zanchettin et al., 2018), robotics (Meng et al., 2017), motion control (Kissling et al., 2009). Many patents directly related to these methods have been filed and granted, most of them concerning IFT or VRFT. Application of these methodologies in engineering practice seems to be steady and growing, though this is hard to measure — publicizing their practices is not exactly a priority for most engineers and engineering companies.

### 6.1. Application classes

There are many aspects that have to be taken into account when deciding what control design approach is best for a given application. These many aspects can, for the most part, be grouped into five application dimensions (see Fig. 3):

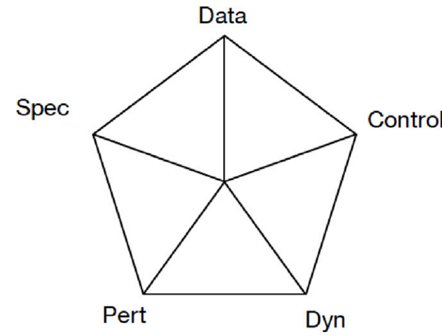


**Fig. 3.** Five dimensions of the design difficulty; larger areas in this figure suggest a harder design; the direction(s) in which the figure is large also indicate in which sense the design is hard.

1. **plant dynamics** — ranging from simple (linear, low order) to complex (nonlinear, time-variant, high order);
2. **perturbations** — from mild (stationary, with known statistics, small, sparse) to harsh (frequent, large amplitude, variable nature, nonstationary);
3. **performance specifications** – from soft (it may happen that closed-loop stability at one operating point is all that is required) to hard (tracking of complex signals, fast transients, varying operating point);
4. **controller order**: from full order (Assumption 1 satisfied) to reduced order (linear, kernel of order much lower than the plant);
5. **data quality** — from good (evenly spaced in time, large signal-to-noise ratio (SNR), abundant, timely, wide range of operating conditions) to poor (noisy, scarce, unevenly spaced in time, with small frequency support).

The dimensions 1 to 4 are traditional issues in the development of a control system. The importance gained by data in recent times adds the fifth one as a decider of what control design approach(es) would be best for a given application. It has also created a new fundamental role for the control designer, which fits the scope of what is now called *data science*. These usually large and poorly informative data must be selected, correlated and filtered, and the design methods must be fed only the data that contain significant amounts of control-relevant information.

These dimensions determine the complexity of the design. A complex plant may demand little design effort if only closed-loop stability is required. This is not an uncommon occurrence in industrial practice. In such cases an appropriate solution may be obtained by a standard PID controller tuned from as little as a few dozens of data points, provided that these data are of good quality — with rich frequency content and large SNR. Applications in different domains fall into different regions of the diagram and thus require solutions of different nature — model-based $x$ data-based, direct $x$ indirect, explicit or implicit control law. There is no definitive and general answer to the question of what is the best design methodology, and probably there will never be. Even for a given application, competing design choices may provide equivalent alternatives and the choice among them will have to be determined by the established culture of the people performing the design. Even personal taste is a valid deciding factor, as long as it does not take precedence over the technical dimensions discussed here.

Moreover, the specific combination of these aspects determines in which sense the control design and operation is hard. Complexity can appear in the data collection and processing, in algorithm running complexity, its coding complexity or the difficulty of the human tasks involved in the design — model reference selection, data curation, etc. Usually these are negatively correlated, that is, a design choice that reduces the complexity in one aspect increases it in the other(s). For

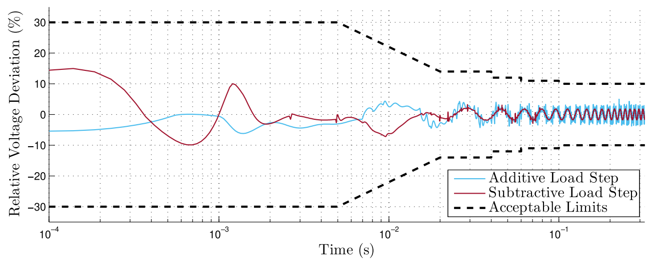


**Fig. 4.** Relative output voltage deviation after nonlinear load step tests.

instance, a low-order controller – such as, say, a PI – can be designed with a handful of data points and running a least-squares problem of dimension two, even for quite complex plants. But the achievable performance will be limited and the designer must pick the reference model very carefully, so this human task will concentrate most of the design effort. On the other hand, if a full order controller is to be designed for a complex plant, then a data-driven design procedure will require little human effort (one can specify almost any reference model), but much more data and potentially significant algorithm ingenuity to find the global optimum of the performance criterion.

Two applications of very different features are now presented to illustrate how DD design works in practice. One is VRFT applied to the design of an 8th order controller in power electronics. The other one is the application of OCI to the design of the most radically low-order controllers – pure gains – to the control of quadcopters.

### 6.2. Power electronics

Electrical energy grids are subject to different imperfections including voltage sag and swell, harmonic distortions, and outages. Uninterruptible Power Supply (UPS) systems overcome these problems by providing regulated and uninterruptible power to critical or sensitive loads. Much effort has been put into the design of controllers to provide such regulated power, where different controllers' structures can be chosen, like resonant or repetitive controllers, and different methodologies were applied in their design (Carballo et al., 2018; Elkayam & Kuperman, 2021; Lin et al., 2022; Lorenzini et al., 2022).

Performance criteria for UPS's are defined by ISO/IEC 62040-3 standard (ISO/IEC 62040-3, 2011), where test loads are described. Nonlinear loads are known to cause disturbances with high odd harmonics content. The performance of the controller is evaluated in both steady-state and transient time. Transient response is evaluated after additive and subtractive load steps, while steady-state is assessed in terms of RMS voltage, Total Harmonic Disturbance (THD), and Individual Harmonic Disturbances (IHDs).

Since the matching condition is far from satisfied, and there are lots of relevant data, direct DD design is recommended. The fact that the controller is linearly parametrized encourages the use of VRFT. Moreover, because the data have high SNR, the statistical properties are not the critical issue, so standard VRFT is indeed the most natural choice. The performance specifications are very strict and the controller is of some complexity (nine parameters), so a lot of data with rich frequency content must be collected. Using standard VRFT implies that the computations involved in the parameters estimation are very simple — standard Least Squares. So, the design effort here is concentrated on picking a good reference model.

Considering the presented performance criteria, the controller to be designed should be able to provide sinusoidal tracking and, most importantly, harmonic rejection. Such performance can be obtained with a cascade-loop configuration, using a multiple resonant controller in the voltage loop to provide disturbance rejection together with a proportional controller in the current loop, to improve the loop performance. The formulation of VRFT to cascade loops was presented in Remes et al. (2020) and in de Paoli Beal et al. (2023) the VRFT method was applied to design a cascade controller for a UPS system.

The results obtained for the rejection of odd harmonics up to the 7th are presented next. Regarding all these harmonics, the proportional multi-resonant controller (17) is parametrized as $C(q,\rho) = \bar{C}^T(q)\rho$ where

$$\rho = \begin{bmatrix} K_{PR} \\ K_{R1_1} \\ K_{R1_0} \\ K_{R3_1} \\ K_{R3_0} \\ K_{R5_1} \\ K_{R5_0} \\ K_{R7_1} \\ K_{R7_0} \end{bmatrix}, \quad \bar{C}(q) = \begin{bmatrix} 1 \\ \frac{q}{q^2-2\cos(\Omega_r)q+1} \\ \frac{1}{q^2-2\cos(\Omega_r)q+1} \\ \frac{q}{q^2-2\cos(3\Omega_r)q+1} \\ \frac{1}{q^2-2\cos(3\Omega_r)q+1} \\ \frac{q}{q^2-2\cos(5\Omega_r)q+1} \\ \frac{1}{q^2-2\cos(5\Omega_r)q+1} \\ \frac{q}{q^2-2\cos(7\Omega_r)q+1} \\ \frac{1}{q^2-2\cos(7\Omega_r)q+1} \end{bmatrix},$$

while the reference model must be constructed respecting the restrictions

$$\left|\left[T_d(q)\right]_{q=e^{\pm jn\Omega_r}}\right| = 1, \quad \angle\left[T_d(q)\right]_{q=e^{\pm jn\Omega_r}} = 0, \quad n = 1,3,5,7. \tag{38}$$

in order to obtain the desired disturbance rejection and reference tracking. Hence, considering odd harmonics up to 7th, the reference model is given by

$$T_d(q) = \frac{\sum_{i=0}^{7} k_i q^i}{\prod_{i=1}^{8}\left(q-p_i\right)}, \tag{39}$$

where $k_i$'s are determined to satisfy (38).

Different values for the poles $p_i$ can be used. In de Paoli Beal et al. (2023), the pole is set to $p = 0.9325$ to design the controllers. In this case, the resulting reference model is

$$T_d(q) = \frac{0.51444 n_1(q) n_2(q)}{(q-0.9325)^8},$$

where

$$n_1(q) = (q-0.9904)\left(q^2-1.972q+0.9731\right),$$
$$n_2(q) = \left(q^2-1.496q+0.9505\right)\left(q^2-1.894q+0.9089\right).$$

The cascade VRFT design presented in Remes et al. (2020) was applied, resulting in the following multiple-resonant controller parameter vector

$$\rho = [84.2475\ 0.3429\ -0.3511\ 0.1163\ -0.2581\ 7.1914\ -7.8862\ 6.5711\ -5.0652]^T$$

and in the current proportional controller gain $k = 20.6548$. Transient response is evaluated after additive (from 25% to 100%) and subtractive (from 100% to 25%) load steps, in compliance with ISO/IEC 62040-3 (2011). Fig. 4 presents the obtained results using the designed controller for load steps, together with the acceptable voltage deviations relative to reference over time after the step defined by the standard. Also, the Total Harmonic Distortion and Individual Harmonic Disturbances (up to the fifteenth harmonic) indexes for the nonlinear tests were all under the limits provided by the standard (de Paoli Beal et al., 2023).

The results presented here were obtained by simulating the plant in the PSIM software, which allows the inclusion of many practical effects, resulting in data very close to real applications. Since the collected data is not noisy, there is no need to resort to instrumental variables or OCI, and VRFT provides excellent results, as good as model-based ones, but with a design procedure of much less effort.

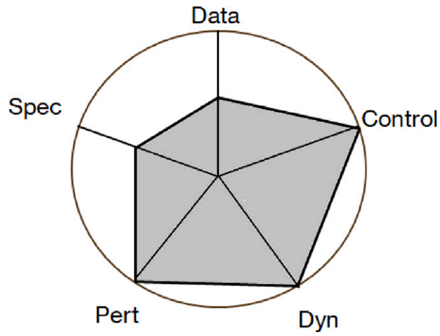


**Fig. 5.** The various aspects of control design for the UAV. The fast dynamics and perturbations, contrasting with the minimalist control structure, harden the task; the quality of data facilitates it; these four aspects favor a direct DD design.

### 6.3. Unmanned Aerial Vehicles

Unmanned aerial vehicles (UAVs) have been popularized in the last decade with many applications such as photography, inspection, delivery and spraying. Its operation became possible because there were significant advances in sensing technology that allowed inertia measurements to be carried out with precision and speed.

Quadrotors are a special type of UAV that use 4 rotors to move the vehicle in the air. Since no movement is restrained, the vehicle has 12 degrees of freedom: 3 positions ($x$, $y$ and $z$), 3 linear velocities, 3 angles (usually called $pitch$, $roll$ and $yaw$) and 3 angular velocities. These vehicles have 12 degrees of freedom but only 4 actuators and therefore they are classified as *underactuaded* which means that they cannot follow any trajectory in space. For example, it is easy to see that quadrotors can stay still in the air but cannot stay still in any pose except horizontally.

The vehicles are represented by complex nonlinear models with at least 12 states (one for each degree of freedom), with extra states if actuator dynamics are considered. In order to represent the system, rotor nonlinearities must be considered along with aerodynamic effects. To obtain a useful model is a difficult task, especially considering that most experimental data is collected from noisy sensors as gyroscopes, accelerometers and GPSs.

Despite the complex process dynamics, most controllers used to regulate the vehicle trajectory are extremely simple with few tuning parameters as proportional and proportional–integral controllers. These simple controllers are effective in controlling the system because a clever hierarchical structure is used that uncouples variables, and uses one controller for each degree of freedom present on the system, resulting in 12 controllers. The controllers are grouped into 3 angular rate controllers, 3 angular controllers, 3 velocity controllers and 3 controllers for position. The design to be presented in the sequel concerns the control in the forward direction; the control for lateral movement is exactly the same. The controller to be designed contains three cascade loops, each one with a proportional controller (see Figs. 5 and 6).

#### 6.3.1. Acro

The simplest control structure is called *acro* in the UAV community and it is used to perform acrobatics. In fact, this control mode only controls the angular rate, where the pilot informs the 3 reference signals for each one of the angular rates which are measured using gyroscopes. Since these 3 signals are uncoupled by nature, they can be controlled separately by 3 SISO controllers, where each one computes the torque $u(t)$ that must be applied in each rotational axis. Despite the nonlinear behavior of the system, proportional controllers are used where there is only one parameter to be tuned. These simple controllers are enough to obtain an adequate closed-loop performance because the angle is kept most of the time constant, such that the angular rate reference is null most of the time. In this case, data-driven control techniques are extremely advantageous because they directly estimate just one parameter instead of identifying a complete model.

We have performed experiments on a commercial agriculture drone used to search for weeds on crops using a specialized camera. The vehicle uses a Pixhawk computer board with a custom PX4 firmware with discrete-time controllers and an extended Kalman filter for state estimation. The OCI method was used to adjust the pitch angular rate controller, which measures the angular rate in rad/s and acts by applying a torque to the vehicle in % of maximum torque. In order to allow an average pilot to perform the movements safely, experience indicates that the closed-loop system should have a settling time of less than 600 ms. Thus the following reference model $T_d(q) = \frac{0.0376}{q-0.9624}$ was used and the design resulted in the controller $C_1 = 0.03153$ rad/(s %).

#### 6.3.2. Angle

*Angle* controllers are used to control the pitch, roll and yaw angles of the vehicle. This is a second *mode* of the firmware where the pilot informs the desired angles and the controller runs in cascade with the rate controller in such a way that the output of the angle controller is the reference of the acro controller.

The closed-loop system should have a settling time of less than 1200 ms, which is twice the settling time of the inner cascade loop. Most of the time the reference is kept constant, and once the angle reaches the reference, the control signal may be null. The OCI method was used to adjust the pitch angle controller with the following reference model $T_d(q) = \frac{0.019}{q-0.981}$, resulting in the controller $C_2 = 4.46\ \mathrm{s}^{-1}$.

#### 6.3.3. Velocity

The third control model is called *velocity* on the UAV community and uses a third controller in cascade where the pilot informs the velocity in axes $x$, $y$ and $z$. The output of the controller for axes $x$ and $y$ are the reference angles for $pitch$ and $roll$, while the velocity in the $z$ axis is directly controlled by the thrust force (which is the mean value of the four rotors).

The specified settling time is 3.6 s, which is three times that of the inner cascade loop, and a proportional controller can be used because the quadrotor flies at slow velocities and there are very low aerodynamic effects. The OCI method was used to adjust the velocity in $y$ with the following reference model $T_d(q) = \frac{0.0064}{q-0.9936}$, resulting in the controller $C_3 = 0.13675$ m/(s rad).

Fig. 7 shows the results of an experiment where the velocity mode is used and a step of 2 m/s is applied on the velocity reference, as shown in Fig. 6. All three loops act together in cascade to compute the torque that should be applied by the rotors aiming to achieve the desired velocity. It is possible to see that the rate controller achieves well the first order reference model — one can say that the matching condition is almost satisfied. On the other hand, the angle and the velocity controllers achieve the specification of settling time (1.2 and 3.6s, respectively), but the obtained response is rather distinct from the desired response, that is, the matching condition is violated. This is not a problem, and the designer could choose a more complex controller such that this Assumption would be satisfied, or (s)he could choose a more complex reference model that could be achieved with this simple proportional controller, but both would be a waste of effort in this case. One of the main advantages of the restricted order controller data-driven design is the possibility of tuning simple controllers, using simple reference models and still achieving the specifications without the need for full order model design with an extra step of controller reduction.

Data-driven classical control provides very convenient and effective techniques to tune low-order controllers for complex nonlinear systems. A model-based approach would imply in estimating a full-order model for the vehicle that should consider aerodynamic and nonlinear effects. Estimating the model requires a lot of work with specific experiments for identification, in addition to complex controller design to, in the

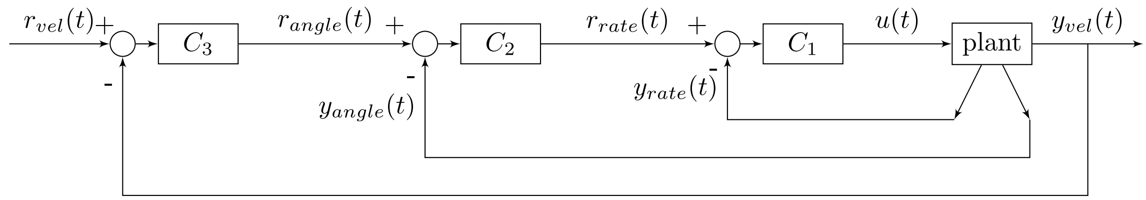


**Fig. 6.** Cascade control structure of Unmanned Aerial Vehicle with control of velocity, angle and angular rate.

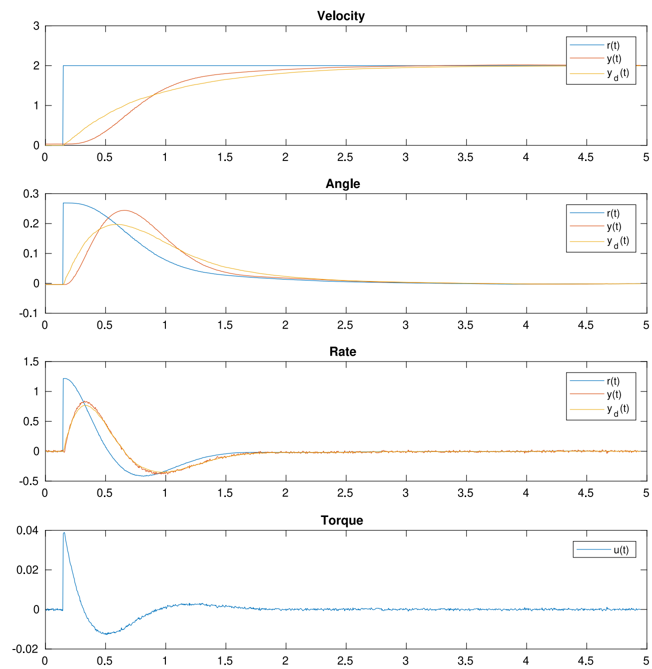


**Fig. 7.** Front Velocity, Pitch Angle, Pitch Angular Rate and Torque of an UAV with 3 proportional controllers in cascade.

end, perform a controller reduction to compute low-order controller gains. On the other hand, data-driven techniques tune directly low-order controllers with better performance since it does not possess modeling nor controller reduction approximations. Moreover, the data used in data-driven techniques come from normal flights (in open or closed loop) such that the pilot does not need to perform specific flights, as required to estimate complex plant models.

## 7. Nonlinear controllers

Given their common theoretical foundation, the range of applications of data-driven classical control is expected to be about as large as that of classical control theory, with about the same limitations. Still, it is natural to conceive the application of these control design methods to nonlinear systems, in the same way as traditional parameter identification methods are used for the identification of nonlinear models. Actually, this has been recognized very early. An extension of the VRFT method for nonlinear controllers has been presented not long after its original version (Campi & Savaresi, 2006) and IFT has also been extended for nonlinear systems in Hjalmarsson (1998). However, the actual application of these ideas to nonlinear systems presents numerous challenges and not many works exploring it are found in the literature — Bazanella and Neuhaus (2014), Esparza et al. (2011) and Novara and Milanese (2019) are representative of the very few. Most of the data-driven control strategies for nonlinear system presented in the literature are based on state-space representations, as the ones presented in Coulson et al. (2019), Dai and Sznaier (2018) and Novara et al. (2013). To better expose the challenges of data-driven classical control in this setting and identify their (potential) solutions, let us first set the stage in mathematical terms.

### 7.1. The stage

The plant to be controlled is now assumed to be nonlinear, described by a difference equation in the form

$$y(t) = f(y(t-1)\dots y(t-np), u(t-1),\dots,u(t-mp), v(t)) \tag{40}$$

where $f(\cdot) : \mathbb{R}^{np+mp+1} \to \mathbb{R}$. To avoid unnecessary definitions and complications, let us assume that the desired performance is linear, so the control objective is still to minimize a cost in the form (5), and that the reference model has positive relative degree.

Now assume that for any initial condition, there is a bijective relationship between the input and the output of the plant. In more formal terms, define a time horizon $N$, consider generically the sequences

$$U_N = \begin{bmatrix} u(0) \\ (1) \\ \vdots \\ u(N-1) \end{bmatrix} \quad Y_N = \begin{bmatrix} y(1) \\ (2) \\ \vdots \\ y(N) \end{bmatrix},$$

and let $Y_N^i$ be the solution of the difference Eq. (40) with the input sequence $U_N^i$ and for a given initial condition. Then the bijectivity assumption reads as follows.

**Assumption 2.** For any given initial condition, any pair of different input sequences – that is $U_N^1 \neq U_N^2$ – results in different output sequences — that is, we have $Y_N^1 \neq Y_N^2$.

It has been shown in Campi and Savaresi (2006) that under this assumption there exists an *ideal controller*

$$u(t) = C_0(u(t-1),\dots,u(t-nc), e(t), e(t-1),\dots,e(t-mc))$$

which results in the desired closed-loop behavior if put in the loop, where $C_0(\cdot) : \mathbb{R}^{nc+mc+1} \to \mathbb{R}$. Thus, the control design can be seen as the exercise of learning this ideal controller just like in the linear case. We define the control law, which is aimed at approximating the ideal controller, in a linearly parametrized way:

$$C(\rho, \mathcal{Z}(t)) = \Sigma_{i=1}^{p} \rho_i \psi_i(\mathcal{Z}(t)), \tag{41}$$

where we have defined $\mathcal{Z}(t) = [u_{t-1} \ \dots \ u_{t-nc} \ e_t \ e_{t-1} \ \dots \ e_{t-mc}]^T$ to shorten the notation and $\psi_i(\mathcal{Z}(t)), i = 1,\dots,p$ is a *dictionary* of functions and/or functionals.

In the identification-based methods, the fact that the ideal controller is a nonlinear map brings one immediate and major difficulty: in defining a parametrized controller class to approximate the ideal controller one has to be concerned not only with the dynamic order but also with the nature of the nonlinearities involved. As a result, even the simplest examples may require large dictionaries to obtain a decent estimator, which brings major challenges for the design procedure that, to this day, have just started to be addressed.

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 reduced. 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 $p$ grows very rapidly. Take the example of a first-order error feedback controller; then there are typically three signals in $\mathcal{Z}(t)$: $e(t)$, $e(t-1)$ and $u(t-1)$. If we take a modest third-order polynomial dictionary we will 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 does not want to implement a controller with more than two hundred parameters; a more parsimonious controller is desired in any case. It is natural to resort to some kind of regularization, to achieve parsimony and improved statistical properties. Moreover, the data batches used for training the controllers must be much larger than the dimension of the parameter vector, thus data batches with several hundred entries are the minimum conceivable size of a training set — whereas it is not unusual in practice that a linear PID can be trained with a dozen data points.

A very successful alternative to the modeling of nonlinear systems is the use of Linear Parameter Varying (LPV) models. With an LPV model various computational and conceptual advantages are obtained, at the cost of abandoning any hope of the matching condition ever being satisfied. Accordingly, LPV controllers provide a sound and promising alternative to the design of DD controllers for nonlinear plants. To illustrate its potential, we consider in the following Subsections the application of identification-based DD control design to two simple nonlinear examples, starting with an LPV design.

### 7.2. LPV design

Data-driven design of LPV controllers has been considered both within the classical framework and for the design of state feedback controllers — see, for instance, Miller and Sznaier (2023), Pang et al. (2018) and Verhoek et al. (2023). Within our classical framework, the literature seems to be restricted to extensions of VRFT to the LPV case. Early work on that has appeared over ten years ago (Formentin & Savaresi, 2011) and later developments have extended and improved the method in some directions (Formentin et al., 2016; Piga et al., 2018), even adapting the method to the use of frequency domain data (Bloemers et al., 2019). In Butcher and Karimi (2010) an extensive study is performed about the consistency of LPV controllers' design when the estimate is obtained through the instrumental variable approach. Still, the literature is not abundant and there are many open problems waiting to be solved.

To illustrate the potential and challenges in this approach, consider the application of LPV-VRFT as presented in Formentin and Savaresi (2011) to the following simple plant

$$\begin{aligned} \dot{x}_1 &= -2{x_1}^2 + (1-x_1)u, \\ \dot{x}_2 &= {x_1}^2 - x_2 u, \\ y &= x_2. \end{aligned} \tag{42}$$

This model is commonly used to represent a constant volume continuous stirred-tank reactor (CSTR) (Roffel & Betlem, 2007). When $u$ varies from 0 to $\infty$, $y$ varies from 0.5 to 0, so it is not possible to track references outside of this range. The specification is to tune a controller so the closed-loop system tracks constant references between 0.1 and 0.4.

An ideal controller can be defined. It is seen in (42) that by applying the control law

$$u = \frac{1}{x_2}(x_1^2 - u')$$

the I/O relationship between the new input $u'$ and the output $y$ becomes a single integrator. Then, after sampling, with the application of the additional feedback

$$u'(q) = \frac{(q-1)T_d(q)}{1-T_d(q)}e(q)$$

the I/O relationship from the reference $r$ to the output $y$ becomes exactly as specified by $T_d(q)$.

As in the LTI design, a batch of I/O data must be collected. For the design of an LPV controller, the data must also include the scheduling variable, that is, the values $\{u(t), y(t), \sigma(t)\}_{t=1,2,\ldots,N}$. In the present case, the scheduling variable is the output itself. The desired closed-loop response is defined by an LTI reference model $T_d(q)$ and the following cost function should be minimized:

$$J_{\mathrm{LPV}}^{\mathrm{VR}}(\varrho) = \sum_{t=1}^{N} [F(q)(u(t) - u(t,\varrho))]^2 , \tag{43}$$

where

$$u(t,\varrho) = \sum_{i=0}^{n-1}\sum_{j=1}^{m} P_{ij}\frac{1}{Q(q)} f_j(\sigma(t-i))\bar{e}(t-i), \tag{44}$$

$\bar{e}(t) = (T_d^{-1}(q) - I)y(t)$ is the virtual error, $Q(q)$ contains the fixed modes of the controller, $F(q)$ is a filter and $P_{ij}$ are the parameters in $\varrho$. Since $Q(q)$ is fixed, (43) is quadratic in $\varrho$ and can be solved by least squares, just like in the LTI case. For convenience, the parameters are organized in a parameter **matrix** $P$, in which each row contains the parameters corresponding to one of the gains of the controller.

A sequence of steps was applied in open loop to collect data, in such a way that the output varies in the full range 0.1 to 0.4. As the plant is nonlinear, with strongly varying dynamics along the specified operating range, it is important that the whole range is visited in the experiment. A total of 3600 samples of data were collected, with a sampling time of $T_s = 0.1$ s. Since the specification is to track constant references, an LPV PID controller is chosen, with the controller equation

$$u(t,\varrho) = u(t-1) + \sum_{i=0}^{2} b_i(y(t-i))e(t-i),$$

corresponding to $Q(q) = 1 - q^{-1}$ and $n = 3$, where the coefficients $b_i$ are linear combinations of the functions

$$f_1(y) = 1, \qquad f_2(y) = y, \qquad f_3(y) = y^2.$$

As reference model, a continuous model with settling time of 3s, no overshoot and zero steady-state error was constructed and then discretized, resulting in

$$T_d(q) = \frac{0.0172(q+0.88)}{(q-0.82)^2}.$$

Since the plant to be controlled is nonlinear, the chosen controller does not allow to satisfy Assumption 1. To minimize the resulting bias, the filter $L(q) = T_d(q)(1-T_d(q))$, is applied to the collected data. Using such definitions, the controller parameters are estimated as

$$\hat{P}_1 = \begin{bmatrix} 0.0516 & -4.615 & 11.87 \\ -1.883 & 5.186 & -7.606 \\ -9.141 & 60.610 & -92.05 \end{bmatrix} \tag{45}$$

The variation of the gains $b_0$, $b_1$ and $b_2$ as a function of $y$ is shown in Fig. 8.

Two other controllers are designed for comparison. First, an alternative LPV controller was calculated by interpolating the gains of three LTI controllers, a typical gain scheduling design procedure, calculated in three different operating points — 0.15, 0.25 and 0.35. Using the same reference model, and the same filter, after interpolating the gains using a second order polynomial, the LPV controller parameters are given by

$$\hat{P}_2 = \begin{bmatrix} -0.0065 & -0.8249 & 2.408 \\ -0.5508 & -3.144 & 1.563 \\ -10.25 & 64.31 & -92.08 \end{bmatrix}.$$

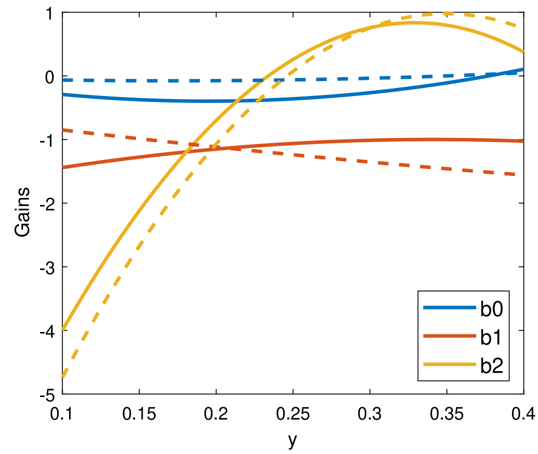


**Fig. 8.** PID LPV gains, as a function of $y$, designed using the extension of VRFT to LPV (continuous line) and by interpolation of LTI controllers (dashed line).

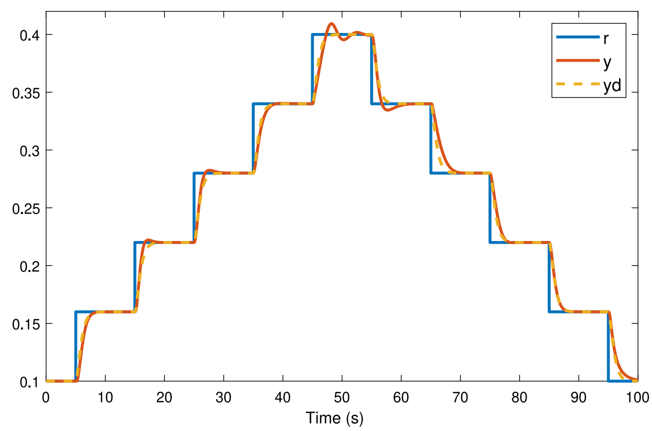


**Fig. 9.** Closed-loop response obtained with the VRFT method for LPV controller design.

The obtained gains $b_0$, $b_1$ e $b_2$ are shown in Fig. 8, in dashed lines.

Then an LTI PID controller was designed using the data set collected for the LPV design. The obtained gains were

$$b_0 = 0.2721, \qquad b_1 = -1.8803, \qquad b_2 = 0.4939.$$

The three designed controllers were simulated in closed loop, where the reference signal is varied from 0.1 to 0.4. Fig. 9 presents the closed-loop response obtained with the LPV controller with the estimated $\hat{P}_1$ given in (45), compared to the reference model response. The response obtained with the interpolation method – that is, $\hat{P}_2$ – is almost identical, so it is not presented. The main advantage of the presented method is to perform only one controller design, instead of a sequence of LTI designs followed by an interpolation of the controller gains. On the other hand, the two LPV controllers result in the same closed-loop performance, even though the parameters in $\hat{P}_1$ and $\hat{P}_2$ are very different.

The LTI PID result is presented in Fig. 10 for comparison, where it is seen that the LTI controller response varies significantly with the operating point, as expected. For small values of $y$, its response is much slower than the desired response while for high values of $y$, it is very oscillatory. On the other hand, the responses obtained with the LPV controllers are much closer to the desired response, despite varying slightly with the operating point.

It is seen that the LPV controller is capable of approximately canceling the nonlinear behavior of the plant, so that the performance is close to the reference model in the whole operating range. This has required a much larger number of parameters than with an LTI controller, but the dimension of the parameter vector is still modest and the estimation can still be performed by least squares. So the advantages of direct DD control methods (VRFT in this case) are maintained to a large extent.

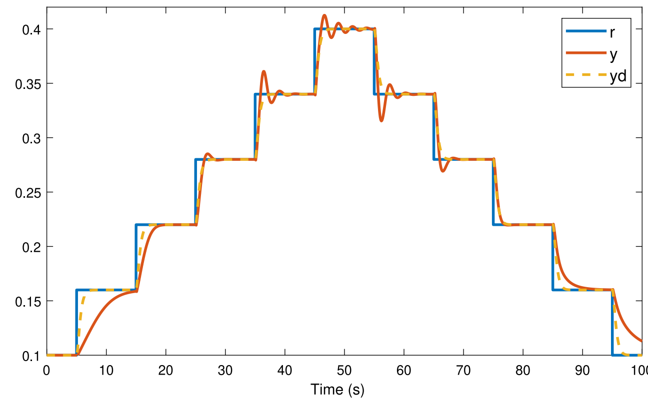


**Fig. 10.** Closed-loop response with the LTI PID controller.

Yet, as the plant becomes more complex and the operating range becomes larger, the number of parameters to be estimated grows. Also, it requires the definition of a scheduling variable that is strongly correlated to the system's dynamics and its real-time measurement. And there is no hope of ever getting even close to the satisfaction of the matching condition. On the other hand, the fact that very distinct parameters yield the same performance raises questions of identifiability and robustness to noise. In short, LPV controllers are a very promising alternative for the DD design of nonlinear controllers that still requires much theoretical and practical development.

### 7.3. Nonlinear control

Consider a plant of the Hammerstein type, with the piece-wise 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 defined by the following transfer function:

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

The controller class to be used is 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 composed of just one element $\mathcal{Z}(t) = \sum_{\tau=1}^{t-1} e(\tau)$ and the control law is $u(t) = \sum_{i=1}^{m} \rho_i \psi_i(\mathcal{Z}(t))$.

The choice of the reference model is guided by the following considerations. The closed-loop system must track constant references with zero steady-state error. Given the minimalist control class that is available, we know that it is not possible to achieve a settling time faster than in open loop. A settling time around twice the one observed in open-loop will thus be specified. The relative degree of the reference model cannot 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, so the reference model must have relative degree of at least two. Taking these considerations into account, the following reference model is chosen:

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

Next, the dictionary of functions $\psi_i(\cdot)$ must be chosen. We start with the following polynomial dictionary:

$$u(t) = \sum_{i=1}^{p} \rho_i \left(\frac{\mathcal{Z}(t)}{200}\right)^{2i-1} \tag{46}$$

which, for numerical reasons, has been normalized such that all elements have unitary magnitude at the end of scale for the data that have been collected. Noisy data are collected and VRFT is run for various dictionary sizes, up to $p = 400$. No stabilizing controller was found in any case.

In order to cope with the large dimension of the dictionary, we apply $L_1$ regularization to the optimization. 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

$$J_L^{VR}(\rho) = \sum_{t=1}^{N}[u(t) - \Sigma_{i=1}^{p}\rho_i\psi_i(\bar{\mathcal{Z}}(t))]^2 + \alpha\sum_{i=1}^{m}|\rho_i| \tag{47}$$

where $\alpha$ is the shrinking factor, to be chosen, and the bar in $\bar{\mathcal{Z}}$ means that this is the virtual version of $\mathcal{Z}$ — that is, the value computed with the virtual error in lieu of $e(t)$. 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 (Hastie et al., 2009).

The larger the value of $\alpha$, the more parameters will be zero and thus more parsimonious will be the controller. All simulations in this work used $\alpha = 0.001$, which allowed reducing the number of parameters without a significant loss of performance. Then we run LASSO-VRFT with $p = 400$ and a stabilizing controller is found, but with poor performance and no parsimony — only 20 parameters were zeroed. In short, the design with the polynomial dictionary was a failure.

So, a different dictionary is chosen, more suited to describe piecewise affine functions, which is formed by various deadzones. This dictionary is

$$\hat{\varphi}(x) = \sum_{i=1}^{m} \rho_i \; ZM_i(x) \tag{48}$$

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

$$ZM_i(x) = \begin{cases} \frac{x+10(i-1)}{200-10(i-1)} & x < -10(i-1) \\ \frac{x-10(i-1)}{200-10(i-1)} & x > 10(i-1) \\ 0 & -10(i-1) < x < 10(i-1) \end{cases}. \tag{49}$$

Then VRFT is run with and without $L1$-regularization for various dictionary sizes; results for $p = 20$ and $p = 400$ are presented. The LASSO-VRFT reduced the number of parameters from $p = 400$ to 52, and from $p = 20$ to 8. The closed-loop performance is shown in Fig. 11, where it is seen that the performances obtained with the LASSO-VRFT with $m = 20$ – which resulted in a controller with $p = 8$ parameters – and $m = 400$ – which resulted in $p = 52$ parameters – are virtually indistinguishable. It is observed that with the right dictionary the design is successful, even without regularization, as the performance obtained with each controller is not far from the best that can be obtained with the given controller class. But the use of regularization has allowed reducing drastically the number of parameters in the controller.

### 7.4. Challenges

Regardless of the approach to the design and of the design method, some challenges are inherent to the problem at hand. From the theoretical discussion and the results of the case study presented above one can identify the main ones. Some are typical from nonlinear system identification and machine learning in general: selecting a good dictionary, collecting an appropriate training set, finding the best regularizing weight. Others are specific to the problem of direct data-driven control: theoretical questions about the problem formulation, choice of the reference model, design of filters to optimize the information content of the data and to shape the cost function.

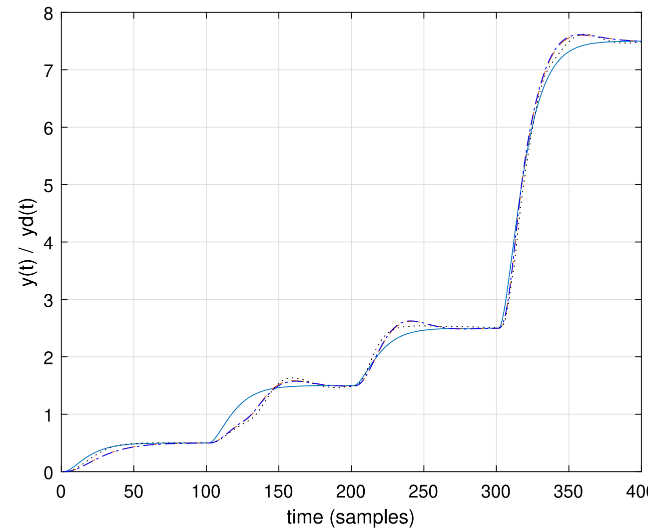


**Fig. 11.** Closed-loop responses of plant #2 with the controllers obtained from the deadzone dictionary: $m = 20$ with VRFT (dotted line) and with LASSO-VRFT (dashed); $m = 400$, with LASSO-VRFT (dash-dot).

It is well-known in nonlinear system identification and in machine learning that dictionary selection is critical. The example in Section 7.3 illustrates just how critical: even for this very benign case (a very simple plant with abundant data of good quality) a generic choice of dictionary – a polynomial – is not able to provide satisfactory results. This example also illustrates the fundamental role that regularization tends to play. For this polynomial dictionary even a stabilizing controller could not be found without applying regularization. On the other hand, with the appropriate dictionary, the performance obtained was close to the best that can be obtained with the given controller class and regularization did the regularization thing: it reduced radically the size of the dictionary to provide a parsimonious controller. Plenty of work remains to be done along this line, but other alternatives can and should be considered besides dictionaries. Some modern kernel formulations have been recently become widespread in nonlinear systems modeling in connection with DMD (Dynamic Mode Decomposition) and the Koopman operator theory — see Brunton and Kutz (2019), for instance; these are important alternatives for future research on direct data-driven control design.

One major difficulty is the generation of a data set that provides enough information for the design to proceed. In the nonlinear setting, the data must not only be sufficiently rich in the usual sense in the linear context — which relates to their frequency content. Instead, the experiment must visit all relevant operating regions, varying amplitudes as well as frequencies. This not only requires much larger amounts of data but may also require a priori knowledge to determine to which extent the data are informative enough.

*Choice of the reference model.* As a general rule for nonlinear systems, it seems too ambitious to specify a linear closed-loop behavior. Obtaining this behavior would require a state feedback controller canceling all the nonlinear features of the plant. The fact that the ideal controller is a state feedback brings up issues of existence that will be discussed in the next paragraph. But even when I/O linearization is possible, it might not be desirable, as even possibly benign features of the open-loop system may be canceled. A controller that takes advantage of the intrinsic features of the plant, like the ones based on energy and passivation concepts (Bazanella et al., 1999; Ortega et al., 2001;

Sepulchre et al., 1997), tend to be simpler, more robust and require less control action. One may want to take these considerations into account when choosing a reference model. However, the choice of an appropriate linear reference model may be hard enough, and guidelines that would allow to choose a nonlinear reference model are likely to be much more difficult to get. As of today, this seems a completely open and unexplored issue.

*Invertibility.* The rationale behind identification-based methods is the existence of an ideal controller. This makes sense under the invertibility Assumption 2, for otherwise the ideal controller may not even exist. It is intuitive that different inputs should result in different outputs, so satisfaction of Assumption 2 is to be expected in most cases, but it cannot be taken for granted. A trivial counterexample is given by

$$y(t) = y(t-1) + 2\sqrt{y(t-1)u(t-1)} + u(t-1).$$

Insights about this issue can be taken from the feedback linearizing theory (Grizzle & Kokotovic, 1988; Monaco & Normand-Cyrot, 1987, 1988). To see this, consider a state-space representation

$$\mathbf{x}(t+1) = F(\mathbf{x}(t), u(k)) \tag{50}$$

$$y(t) = h(\mathbf{x}(t)) \tag{51}$$

where $F(\cdot) : \mathbb{R}^{np+1} \to \mathbb{R}^{np+1}$ is a vector field and $h(\cdot) : \mathbb{R}^{np+1} \to \mathbb{R}$. An input–output feedback linearizing state feedback controller does exist under mild conditions and is defined implicitly by the equation

$$h \circ F^{\delta}(\mathbf{x}(t), u(t)) = u'(t). \tag{52}$$

In (52), $u'(t)$ is defined by the reference model

$$u'(t) = \frac{T_d(q)q^{\delta}}{1 - T_d(q)} e(t) \tag{53}$$

where $\delta$ is the relative degree of the plant (50)–(51), $e(t)$ is the tracking error and the composition operator $\circ$ is defined recursively as

$$h(\mathbf{x}) \circ F(\mathbf{x}, u(t)) = h(F(\mathbf{x}, u(t)))$$
$$h \circ F^{i}(\mathbf{x}, u(t)) = h \circ F^{i-1}(F(\mathbf{x}, u(t)))$$

This linearizing controller is a state feedback controller and in our setting we are looking for a dynamic output feedback controller. Thus a relevant issue to be addressed is to determine conditions under which an equivalent dynamic output feedback controller exists, which is related to the invertibility of the I/O relationship.

Now, if invertibility of the I/O map is not generic, what can be done in the cases in which it is not satisfied? Does it even make sense to develop a theory around the identification of a nonexistent object? None of these concerns exist in the linear case; when the plant and the reference model are linear, the existence of the ideal controller is self-evident and there is even a general formula for it.

*Non-minimum phase plants.* The feedback linearizing controller (50)–(51) achieves linearization at the expense of loss of observability, and the nonobservable dynamics is called the plant's *zero dynamics*. This is the precise generalization of the zeros of a linear system and, by this analogy, plants with unstable zero dynamics are said to be of nonminimum phase (NMP). For an NMP plant, the I/O linearization results in an internally unstable system. Perhaps this is the most limiting and challenging theoretical issue to be solved. It has been solved for linear systems, as discussed earlier in this paper, but at the cost of additional complexity in the solution, and different design methods (OCI, IFT, VRFT) require different solutions — procedurally, if not conceptually.

Like in the linear case, it is worth noticing that the occurrence of NMP systems in discrete-time is the rule, not the exception. Indeed, it has been shown in Monaco and Normand-Cyrot (1988) that, just like in the linear case, any nonlinear continuous-time system of finite order results in a discrete-time system of relative degree equal to one when sampled periodically. It has also been shown in Monaco and Normand-Cyrot (1988) that sampling a continuous-time system that is minimum-phase results in a minimum-phase discrete-time system only if its relative degree is at most equal to two — relative degree of the continuous-time system equal to one is a sufficient condition. So, different from continuous-time systems, in discrete-time systems the existence of unstable zero-dynamics is a very common occurrence, both for linear systems and for nonlinear systems.

## 8. Conclusions

A variety of data-driven control design methods have been developed around the framework of classical control theory. These methods have evolved into a consistent theory, that we have named *data-driven classical control*, and have matured enough to find widespread application in the engineering practice. But there is still room for improvements in these methodologies, which motivates future work.

On the other hand, the extension of this approach and these concepts to the design of nonlinear controllers is still an infant field. Endeavours along this line will likely merge the "classical" nonlinear control systems theory with the elements of machine learning and data science, which promises a rich and productive future for research.

## Declaration of competing interest

Potential conflict of interest exists:

No conflict of interest exists.

We wish to confirm that there are no known conflicts of interest associated with this publication and there has been no significant financial support for this work that could have influenced its outcome.

## Funding

All of the sources of funding for the work described in this publication are acknowledged below:

Conselho Nacional de Desenvolvimento Científico e Tecnológico — CNPq

Coordenação para o Aperfeiçoamento de Pessoal Docentes — CAPES

Both sources belong to the Brazilian government and have funded this work by means of scolarships.

## Data availability

Data will be made available on request.

## References

Åström, K., Hagander, P., & Sternby, J. (1984). Zeros of sampled systems. *Automatica*, *20*(1), 31–38. http://dx.doi.org/10.1016/0005-1098(84)90062-1.

Åström, K., & Wittenmark, B. (1973). On self-tuning regulators. *Automatica*, *9*, 185–199.

Åström, K., & Wittenmark, B. (1995). *Adaptive control* (2nd ed.). Addison-Wesley.

Bazanella, A. S., Bombois, X., & Gevers, M. (2012). Necessary and sufficient conditions for uniqueness of the minimum in prediction error identification. *Automatica*, *48*(8), 1621–1630. http://dx.doi.org/10.1016/j.automatica.2012.06.018.

Bazanella, A. S., Campestrini, L., & Eckhard, D. (2012). *Data-driven controller design: The H2 approach*. New York, USA: Springer.

Bazanella, A. S., Campestrini, L., & Eckhard, D. (2022). An introduction to data-driven control, from kernels to behaviors. In *2022 IEEE 61st conference on decision and control (CDC)* (pp. 1079–1084). http://dx.doi.org/10.1109/CDC51059.2022.9992589.

Bazanella, A. S., Gevers, M., Miskovic, L., & Anderson, B. D. O. (2008). Iterative minimization of H-2 control performance criteria. *Automatica*, *44*(10), 2549–2559. http://dx.doi.org/10.1016/j.automatica.2008.03.014.

Bazanella, A. S., Kokotovic, P., & Silva, A. (1999). A dynamic extension for LgV controllers. *IEEE Transactions on Automatic Control*, *44*(3), 588–592. http://dx.doi.org/10.1109/9.751357.

Bazanella, A. S., & Neuhaus, T. (2014). Tuning nonlinear controllers with the virtual reference approach. *IFAC Proceedings Volumes*, *47*(3), 10269–10274. http://dx.doi.org/10.3182/20140824-6-ZA-1003.00562, 19th IFAC World Congress.

Bernado, R. T., do Nascimento, M. M., & Dotta, D. (2020). A modified VRFT approach for retuning power system damping controllers. In *2020 IEEE power & energy society general meeting (PESGM)* (pp. 1–5). http://dx.doi.org/10.1109/PESGM41954.2020.9281381.

Bloemers, T., Tóth, R., & Oomen, T. (2019). Towards data-driven LPV controller synthesis based on frequency response functions. In *2019 IEEE 58th conference on decision and control (CDC)* (pp. 5680–5685). http://dx.doi.org/10.1109/CDC40024.2019.9029958.

Boeira, E., & Eckhard, D. (2020). Pyvrft: A python package for the virtual reference feedback tuning, a direct data-driven control method. *SoftwareX*, *11*, http://dx.doi.org/10.1016/j.softx.2019.100383.

Boeira, E., & Eckhard, D. (2023). The optimal regularized weighted least-squares method for impulse response estimation. *Journal of Control, Automation and Electrical Systems*, *34*(2), 302–314. http://dx.doi.org/10.1007/s40313-022-00968-1.

Bordignon, V., & Campestrini, L. (2018a). Application of multivariable virtual reference feedback tuning with anti-windup to the benchmark PID 2018. In *3rd IFAC conference on advances in proportional-integral-derivative control PID 2018, Vol. 51* (pp. 515–520). http://dx.doi.org/10.1016/j.ifacol.2018.06.147.

Bordignon, V., & Campestrini, L. (2018b). Data-driven PID control tuning for disturbance rejection in a hierarchical control architecture. In *3rd IFAC conference on advances in proportional-integral-derivative control PID 2018, Vol. 51* (pp. 569–574). http://dx.doi.org/10.1016/j.ifacol.2018.06.156.

Breschi, V., & Formentin, S. (2021). Proper closed-loop specifications for data-driven model-reference control. *IFAC-PapersOnLine*, *54*(9), 46–51. http://dx.doi.org/10.1016/j.ifacol.2021.06.062, 24th International Symposium on Mathematical Theory of Networks and Systems MTNS 2020.

Brunton, S., & Kutz, J. (2019). *Data-driven science and engineering*. Cambridge University Press.

Butcher, M., & Karimi, A. (2010). Data-driven tuning of linear parameter-varying precompensators. *International Journal of Adaptive Control and Signal Processing*, *24*(7), 592–609. http://dx.doi.org/10.1002/acs.1151.

Campestrini, L., Eckhard, D., Bazanella, A. S., & Gevers, M. (2017). Data-driven model reference control design by prediction error identification. *Journal of the Franklin Institute*, *354*(6), 2628–2647. http://dx.doi.org/10.1016/j.jfranklin.2016.08.006, Special issue on recent advances on control and diagnosis via process measurements.

Campestrini, L., Eckhard, D., Chía, L. A., & Boeira, E. (2016). Unbiased MIMO VRFT with application to process control. *Journal of Process Control*, *39*, 35–49. http://dx.doi.org/10.1016/j.jprocont.2015.12.010.

Campestrini, L., Eckhard, D., Gevers, M., & Bazanella, A. S. (2011). Virtual reference feedback tuning for non minimum phase plants. *Automatica*, *47*(8), 1778–1784.

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

Campi, M., & Savaresi, S. (2006). Direct nonlinear control design: the virtual reference feedback tuning (VRFT) approach. *Automatica*, *51*, 14–27.

Carballo, R. E., Botterón, F., Oggier, G. G., & García, G. O. (2018). Multiple resonant controllers strategy to achieve fault ride-through and high performance output voltage in UPS applications. *IET Power Electronics*, *11*(15), 2415–2426. http://dx.doi.org/10.1049/iet-pel.2018.5603.

Carè, A., Torricelli, F., Campi, M. C., & Savaresi, S. M. (2019). A toolbox for virtual reference feedback tuning (VRFT). In *2019 European control conference (ECC)* (pp. 4252–4257).

Carrette, P., Bastin, G., Genin, Y., & Gevers, M. (1996). Discarding data may help in system identification. *IEEE Transactions on Signal Processing*, *44*(9), 2300–2310. http://dx.doi.org/10.1109/78.536685.

Coulson, J., Lygeros, J., & Dörfler, F. (2019). Data-enabled predictive control: in the shallows of the DeePC. In *Proc. European control conf. (ECC)* (pp. 307–312).

da Silva, R. W. P., & Eckhard, D. (2019). Extension of the correlation-based tuning method for load disturbance rejection. In *2019 IEEE 58th conference on decision and control (CDC)* (pp. 8386–8391). http://dx.doi.org/10.1109/CDC40024.2019.9029360.

da Silva, R. W. P., & Eckhard, D. (2020). Data-driven controller certification through vinnicombe test and local rational models. In *2020 IEEE conference on control technology and applications (CCTA)* (pp. 231–236). http://dx.doi.org/10.1109/CCTA41146.2020.9206310.

da Silva, R. W., & Eckhard, D. (2020). Multivariable correlation-based tuning for load disturbance rejection. *IFAC-PapersOnLine*, *53*(2), 3977–3982. http://dx.doi.org/10.1016/j.ifacol.2020.12.2258, 21st IFAC World Congress.

da Silva, R. W. P., & Eckhard, D. (2021). Data-driven correlation approach applied to load disturbance rejection in a thermal process. In *2021 Australian & New Zealand control conference (ANZCC)* (pp. 200–205). http://dx.doi.org/10.1109/ANZCC53563.2021.9628299.

da Silva, R. W. P., & Eckhard, D. (2023). Data-driven load disturbance rejection. arXiv:2307.01700.

Dai, T., & Sznaier, M. (2018). A moments based approach to designing MIMO data driven controllers for switched systems. In *2018 IEEE conference on decision and control (CDC)* (pp. 5652–5657). http://dx.doi.org/10.1109/CDC.2018.8619361.

de Bruyne, F., & Kammer, L. (1999). Iterative feedback tuning with guaranteed stability. In *American control conference, Vol. 21* (pp. 3317–3321). San Diego - CA, USA.

de Jong, T. O., Breschi, V., Schoukens, M., & Formentin, S. (2023). Data-driven model-reference control with closed-loop stability: The output-feedback case. *IEEE Control Systems Letters*, *7*, 2431–2436. http://dx.doi.org/10.1109/LCSYS.2023.3285424.

Dehghani, A., Lecchini-Visintini, A., Lanzon, A., & Anderson, B. D. O. (2009). Validating controllers for internal stability utilizing closed-loop data. *IEEE Transactions on Automatic Control*, *54*(11), 2719–2725. http://dx.doi.org/10.1109/TAC.2009.2031587.

Dörfler, F., Coulson, J., & Markovsky, I. (2022). Bridging direct and indirect data-driven control formulations via regularizations and relaxations. *IEEE Transactions on Automatic Control*, *68*(2), 883–897. http://dx.doi.org/10.1109/TAC.2022.3148374.

Eckhard, D., & Bazanella, A. S. (2010). Data-based controller tuning: Improving the convergence rate. In *Decision and control (CDC), 2010 49th IEEE conference on* (pp. 4801–4806). http://dx.doi.org/10.1109/CDC.2010.5717584.

Eckhard, D., & Bazanella, A. S. (2012a). Optimizing the convergence of data-based controller tuning. *Proceedings of the Institution of Mechanical Engineers Part I – Journal of Systems and Control Enginnering*, *226*(I4), 563–574. http://dx.doi.org/10.1177/0959651811426062.

Eckhard, D., & Bazanella, A. S. (2012b). Robust convergence of the steepest descent method for data-based control. *International Journal of Systems Science*, *43*(10), 1969–1975. http://dx.doi.org/10.1080/00207721.2011.563874.

Eckhard, D., Bazanella, A. S., Rojas, C. R., & Hjalmarsson, H. (2017). Cost function shaping of the output error criterion. *Automatica*, *76*, 53–60. http://dx.doi.org/10.1016/j.automatica.2016.10.015.

Eckhard, D., Campestrini, L., & Boeira, E. C. (2018). Virtual disturbance feedback tuning. *IFAC Journal of Systems and Control*, *3*, 23–29. http://dx.doi.org/10.1016/j.ifacsc.2018.01.003.

Elkayam, M. S., & Kuperman, A. (2021). Guidelines for single-parameter multiresonant current controllers design allowing prescribed magnitude tracking of periodic references. *IEEE Transactions on Power Electronics*, *36*(8), 9536–9546. http://dx.doi.org/10.1109/TPEL.2021.3052379.

Esparza, A., Sala, A., & Albertos, P. (2011). Neural networks in virtual reference tuning. *Engineering Applications of Artificial Intelligence*, *24*(6), 983–995. http://dx.doi.org/10.1016/j.engappai.2011.04.003.

Fiorio, L. V., Remes, C. L., Wheeler, P., & de Novaes, Y. R. (2023). Virtual reference feedback tuning with robustness constraints: A swarm intelligence solution. *Engineering Applications of Artificial Intelligence*, *123*, Article 106490. http://dx.doi.org/10.1016/j.engappai.2023.106490.

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

Formentin, S., Garatti, S., Rallo, G., & Savaresi, S. M. (2018). Robust direct data-driven controller tuning with an application to vehicle stability control. *International Journal of Robust and Nonlinear Control*, *28*(12), 3752–3765.

Formentin, S., & Karimi, A. (2014). Enhancing statistical performance of data-driven controller tuning via L2-regularization. *Automatica*, *50*(5), 1514–1520. http://dx.doi.org/10.1016/j.automatica.2014.04.001.

Formentin, S., Piga, D., Tóth, R., & Savaresi, S. M. (2016). Direct learning of LPV controllers from data. *Automatica*, *65*, 98–110.

Formentin, S., & Savaresi, S. M. (2011). Virtual reference feedback tuning for linear parameter-varying systems. *IFAC Proceedings Volumes*, *44*(1), 10219–10224. http://dx.doi.org/10.3182/20110828-6-IT-1002.01184, 18th IFAC World Congress.

Formentin, S., Savaresi, S., & Re, L. D. (2012). Non-iterative direct data-driven controller tuning for multivariable systems: theory and application. *IET Control Theory & Applications*, *6*(7), 1250–1257.

Formentin, S., van Heusden, K., & Karimi, A. (2014). A comparison of model-based and data-driven controller tuning. *International Journal of Adaptive Control and Signal Processing*, *28*(10), 882–897. http://dx.doi.org/10.1002/acs.2415.

Franklin, G., Powell, J., & Emami-Naeini, A. (2006). *Feedback control of dynamic systems*. Pearson Prentice Hall.

Garcia, C. S., & Bazanella, A. S. (2019). Extraction of informative subsets from routine operating data for use in data-driven control. In *2019 IEEE 58th conference on decision and control (CDC)* (pp. 8392–8397). http://dx.doi.org/10.1109/CDC40024.2019.9029265.

Garcia, C. S., & Bazanella, A. S. (2020a). The constrained total least squares solution for virtual reference feedback tuning. *IFAC-PapersOnLine*, *53*(2), 3983–3988. http://dx.doi.org/10.1016/j.ifacol.2020.12.2259, 21st IFAC World Congress.

Garcia, C. S., & Bazanella, A. S. (2020b). Selection of informative intervals in routine operating data for use in data-driven control design. In *2020 IEEE conference on control technology and applications (CCTA)* (pp. 225–230). http://dx.doi.org/10.1109/CCTA41146.2020.9206266.

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

Gevers, M., Bazanella, A. S., Bombois, X., & Mišković, L. (2009). Identification and the information matrix: how to get just sufficiently rich? *IEEE Transactions on Automatic Control*, *54*(12), 2828–2840.

Gonçalves da Silva, G. R., Bazanella, A. S., & 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. http://dx.doi.org/10.1109/TCST.2018.2842178.

Gonçalves da Silva, G. R., Bazanella, A. S., & Campestrini, L. (2020). One-shot data-driven controller certification. *ISA Transactions*, *99*, 361–373. http://dx.doi.org/10.1016/j.isatra.2019.10.011.

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

Goodwin, G., & Sin, K. (1984). *Adaptive fltering, prediction and control*. Englewood Cliffs - USA: Prentice-Hall.

Grizzle, J., & Kokotovic, P. (1988). Feedback linearization of sampled-data systems. *IEEE Transactions on Automatic Control*, *33*(9), 857–859. http://dx.doi.org/10.1109/9.1316.

Guardabassi, G., & Savaresi, S. (2000). Virtual reference direct design method: an off-line approach to data-based control system design. *IEEE Transactions on Automatic Control*, *45*(5), 954–959. http://dx.doi.org/10.1109/9.855559.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning - data mining, inference and prediction* (2nd ed.). Springer.

Heertjes, M. F., Van der Velden, B., & Oomen, T. (2016). Constrained iterative feedback tuning for robust control of a wafer stage system. *IEEE Transactions on Control Systems Technology*, *24*(1), 56–66. http://dx.doi.org/10.1109/TCST.2015.2418311.

Hjalmarsson, H. (1998). Control of nonlinear systems using iterative feedback tuning. In *Proceedings of the 1998 American control conference. ACC (IEEE Cat. No.98CH36207), Vol. 4* (pp. 2083–2087). http://dx.doi.org/10.1109/ACC.1998.702994.

Hjalmarsson, H., Gevers, M., Gunnarsson, S., & Lequin, O. (1998). Iterative feedback tuning: theory and applications. *IEEE Control Systems Magazine*, *18*(4), 26–41.

Hjalmarsson, H., Gunnarsson, S., & Gevers, M. (1994). A convergent iterative restricted complexity control design scheme. In *33rd IEEE conference on decision and control* (pp. 1735–1740). Lake Buena Vista - USA.

Holcomb, C. M., & Bitmead, R. R. (2017). Subspace identification with multiple data records: unlocking the archive. arXiv:1704.02635.

Huff, D. D., Campestrini, L., Gonçalves da Silva, G. R., & Bazanella, A. S. (2023). Optimal controller identification for multivariable non-minimum phase systems. arXiv:2305.18938.

Huff, D. D., Campestrini, L., Gonçalves da Silva, G. R., & Bazanella, A. S. (2019). Data-driven control design by prediction error identification for multivariable systems. *Journal of Control, Automation and Electrical Systems*, *30*(4), 465–478. http://dx.doi.org/10.1007/s40313-019-00468-9.

Huff, D. D., Gonçalves da Silva, G. R., & Campestrini, L. (2018). Data-driven control design by prediction error identification for a refrigeration system based on vapor compression. *IFAC-PapersOnLine*, *51*(4), 704–709. http://dx.doi.org/10.1016/j.ifacol.2018.06.186, 3rd IFAC Conference on Advances in Proportional-Integral-Derivative Control PID 2018.

Isidori, A. (1995). *Nonlinear control systems* (3rd ed.). Springer.

ISO/IEC 62040-3 (2011). Uninterruptible power systems (UPS) - part 3: Method of specifying the performance and test requirements.

Jansson, H., & Hjalmarsson, H. (2004). Gradient approximations in iterative feedback tuning for multivariable processes. *International Journal of Adaptive Control and Signal Processing*, *18*(8), 665–681.

Kalman, R. (1960). On the general theory of control systems. In *1st IFAC world congress* (pp. 491–502).

Kammer, L. (2005). Stability assessment for cautious iterative controller tuning. *Automatica*, *41*, 1829–1834.

Kammer, L., Bitmead, R., & Bartlett, P. (2000). Direct iterative tuning via spectral analysis. *Automatica*, *36*, 1301–1307.

Kaneko, O. (2013). Data-driven controller tuning: FRIT approach. *IFAC Proceedings Volumes*, *46*(11), 326–336. http://dx.doi.org/10.3182/20130703-3-FR-4038.00122, 11th IFAC Workshop on Adaptation and Learning in Control and Signal Processing.

Karimi, A., Mišković, L., & Bonvin, D. (2002). Convergence analysis of an iterative correlation-based controller tuning method. *IFAC Proceedings Volumes*, *35*(1), 413–418. http://dx.doi.org/10.3182/20020721-6-ES-1901.00150, 15th IFAC World Congress.

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

Karimi, A., van Heusden, K., & Bonvin, D. (2007). Non-iterative data-driven controller tuning using the correlation approach. In *2007 European control conference (ECC)* (pp. 5189–5195). http://dx.doi.org/10.23919/ECC.2007.7068802.

Kergus, P., Olivi, M., Poussot-Vassal, C., & Demourant, F. (2019). From reference model selection to controller validation: Application to Loewner data-driven control. *IEEE Control Systems Letters*, *3*(4), 1008–1013. http://dx.doi.org/10.1109/LCSYS.2019.2920208.

Khalil, H. (2001). *Nonlinear systems* (3rd ed.). Prentice Hall.

Kissling, S., Blanc, P., Myszkorowski, P., & Vaclavik, I. (2009). Application of iterative feedback tuning (IFT) to speed and position control of a servo drive. *Control Engineering Practice*, *17*(7), 834–840. http://dx.doi.org/10.1016/j.conengprac.2009.02.005.

Krishnan, V., & Pasqualetti, F. (2021). On direct vs indirect data-driven predictive control. arXiv:2103.14936.

Krstic, M., Kanellakopoulos, I., & Kokotovic, P. V. (1995). *Nonlinear and adaptive control design*. Wiley.

Lecchini, A., & Gevers, M. (2002). On iterative feedback tuning for non-minimum phase plants. In *Proceedings of the 41st IEEE conference on decision and control, 2002, Vol. 4* (pp. 4658–4663). http://dx.doi.org/10.1109/CDC.2002.1185113.

Lee, W., Anderson, B., Kosut, R., & Mareels, I. (1993). On robust performance improvement through the windsurfer approach to adaptive robust control. In *Proceedings of 32nd IEEE conference on decision and control, Vol. 3* (pp. 2821–2827). http://dx.doi.org/10.1109/CDC.1993.325710.

Lin, B., Peng, L., & Liu, X. (2022). Selective pole placement and cancellation for proportional–resonant control design used in voltage source inverter. *IEEE Transactions on Power Electronics*, *37*(8), 8921–8934. http://dx.doi.org/10.1109/TPEL.2022.3151508.

Ljung, L. (1999). *System identification - theory for the user* (2nd ed.). Prentice-Hall.

Lorenzini, C., Pereira, L. F. A., Bazanella, A. S., & Gonçalves da Silva, G. R. (2022). Single-phase uninterruptible power supply control: A model-free proportional-multiresonant method. *IEEE Transactions on Industrial Electronics*, *69*(3), 2967–2975. http://dx.doi.org/10.1109/TIE.2021.3068664.

Meng, W., Xie, S. Q., Liu, Q., Lu, C. Z., & Ai, Q. (2017). Robust iterative feedback tuning control of a compliant rehabilitation robot for repetitive ankle training. *IEEE/ASME Transactions on Mechatronics*, *22*(1), 173–184. http://dx.doi.org/10.1109/TMECH.2016.2618771.

Miller, J., & Sznaier, M. (2023). Data-driven gain scheduling control of linear parameter-varying systems using quadratic matrix inequalities. *IEEE Control Systems Letters*, *7*, 835–840. http://dx.doi.org/10.1109/lcsys.2022.3227182.

Moheimani, S. O. R., & Vautier, B. J. G. (2005). Resonant control of structural vibration using charge-driven piezoelectric actuators. *IEEE Transactions on Control Systems Technology*, *13*(6), 1021–1035.

Monaco, S., & Normand-Cyrot, D. (1987). Minimum-phase nonlinear discrete-time systems and feedback linearization. In *Proceedings of the 26th conference on decision and control* (pp. 979–986).

Monaco, S., & Normand-Cyrot, D. (1988). Zero dynamics of sampled nonlinear systems. *Systems & Control Letters*, *11*(3), 229–234. http://dx.doi.org/10.1016/0167-6911(88)90063-1.

Nakamoto, M. (2004). An application of the virtual reference feedback tuning for an mimo process. In *SICE 2004 annual conference, Vol. 3* (pp. 2208–2213).

Narendra, K., & Annaswamy, A. (2005). *Stable adaptive systems*. Dover.

Nicoletti, A., Martino, M., & Karimi, A. (2019). A data-driven approach to model-reference control with applications to particle accelerator power converters. *Control Engineering Practice*, *83*, 11–20. http://dx.doi.org/10.1016/j.conengprac.2018.10.007.

Novara, C., Fagiano, L., & Milanese, M. (2013). Direct feedback control design for nonlinear systems. *Automatica*, *49*(4), 849–860. http://dx.doi.org/10.1016/j.automatica.2013.01.002.

Novara, C., & Milanese, M. (2019). Control of MIMO nonlinear systems: A data-driven model inversion approach. *Automatica*, *101*, 417–430. http://dx.doi.org/10.1016/j.automatica.2018.12.026.

Ogata, K. (2009). *Modern control engineering* (5th ed.). Prentice-Hall.

Oomen, T., van der Maas, R., Rojas, C. R., & Hjalmarsson, H. (2014). Iterative data-driven $\mathcal{H}_\infty$ norm estimation of multivariable systems with application to robust active vibration isolation. *IEEE Transactions on Control Systems Technology*, *22*(6), 2247–2260. http://dx.doi.org/10.1109/TCST.2014.2303047.

Ortega, R., Van Der Schaft, A., Mareels, I., & Maschke, B. (2001). Putting energy back in control. *IEEE Control Systems Magazine*, *21*(2), 18–33. http://dx.doi.org/10.1109/37.915398.

Pang, B., Bian, T., & Jiang, Z.-P. (2018). Data-driven finite-horizon optimal control for linear time-varying discrete-time systems. In *2018 IEEE conference on decision and control (CDC)* (pp. 861–866). http://dx.doi.org/10.1109/CDC.2018.8619347.

de Paoli Beal, G., Lorenzini, C., Flores, J. V., & Campestrini, L. (2023). Harmonic rejection on UPS through cascade VRFT of proportional-resonant controller. In *Accepted for publication in CCTA-2023*. IEEE.

Park, J., & Bitmead, R. R. (2004). Vinnicombe's winding number condition is not transitive: Impacts for adaptive control. *IFAC Proceedings Volumes*, *37*(12), 783–787. http://dx.doi.org/10.1016/S1474-6670(17)31565-3, iFAC workshop on adaptation and learning in control and signal processing (ALCOSP 04) and IFAC workshop on periodic control systems (PSYCO 04), Yokohama, Japan, 30 August - 1 September, 2004.

Pereira, L. F. A., Flores, J. V., Bonan, G., Coutinho, D. F., & Gomes da Silva, J. M., Jr. (2014). Multiple resonant controllers for uninterruptible power supplies – a systematic robust control design approach. *IEEE Transactions on Industrial Electronics*, *61*(3), 1528–1538. http://dx.doi.org/10.1109/TIE.2013.2259781.

Piga, D., Formentin, S., & Bemporad, A. (2018). Direct data-driven control of constrained systems. *IEEE Transactions on Control Systems Technology*, *26*(4), 1422–1429. http://dx.doi.org/10.1109/TCST.2017.2702118.

Previdi, F., Schauer, T., Savaresi, S., & 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. http://dx.doi.org/10.1109/TCST.2003.821967.

Procházka, H., Gevers, M., Anderson, B., & Ferrera, C. (2005). Iterative feedback tuning for robust controller design and optimization. In *IEEE conference on decison and control - european control conference* (pp. 3602–3607). Seville - Spain.

Rallo, G., Formentin, S., Chiuso, A., & Savaresi, S. M. (2016). Virtual reference feedback tuning with Bayesian regularization. In *2016 european control conference (ECC)* (pp. 507–512). http://dx.doi.org/10.1109/ECC.2016.7810335.

Remes, C. L., Binz, R. E., Flores, J. V., & Campestrini, L. (2020). Virtual reference feedback tuning applied to cascade control. *IET Control Theory & Applications*, *14*(20), 3738–3746. http://dx.doi.org/10.1049/iet-cta.2020.0620.

Remes, C. L., Gonçalves da Silva, G. R., Treviso, A., Coelho, M. A., & Campestrini, L. (2019). Data-driven approach for current control in DC-DC boost converters. *IFAC-PapersOnLine*, *52*(1), 190–195. http://dx.doi.org/10.1016/j.ifacol.2019.06.059, 12th IFAC Symposium on Dynamics and Control of Process Systems, including Biosystems DYCOPS 2019.

Roffel, B., & Betlem, B. (2007). *Process dynamics and control: modeling for control and prediction*. John Wiley & Sons.

Rojas, J. D., Flores-Alsina, X., Jeppsson, U., & Vilanova, R. (2012). Application of multivariate virtual reference feedback tuning for wastewater treatment plant control. *Control Engineering Practice*, *20*(5), 499–510.

Saeki, M. (2021). Data-driven tuning of model-reference controllers for stable MIMO plants. *Automatica*, *131*, Article 109786. http://dx.doi.org/10.1016/j.automatica.2021.109786.

Sala, A., & Esparza, A. (2005). Extensions to "virtual reference feedback tuning: A direct method for the design of feedback controllers". *Automatica*, *41*(8), 1473–1476. http://dx.doi.org/10.1016/j.automatica.2005.02.008.

Scheid Filho, R., Boeira, E. C., Campestrini, L., & Eckhard, D. (2021). Data-driven control design for load disturbance rejection by prediction error identification. In *2021 Australian & New Zealand control conference (ANZCC)* (pp. 92–97). http://dx.doi.org/10.1109/ANZCC53563.2021.9628273.

Sepulchre, R., Jankovic, M., & Kokotovic, P. (1997). *Constructive nonlinear control*. Springer.

Steentjes, T. R., Lazar, M., & Van den Hof, P. M. (2020). Data-driven distributed control: Virtual reference feedback tuning in dynamic networks. In *2020 59th IEEE conference on decision and control (CDC)* (pp. 1804–1809). http://dx.doi.org/10.1109/CDC42340.2020.9304099.

Steentjes, T. R., Lazar, M., & Van den Hof, P. M. (2021). Controller identification for data-driven model-reference distributed control. In *2021 European control conference (ECC)* (pp. 2358–2363). http://dx.doi.org/10.23919/ECC54610.2021.9655114.

Teodorescu, R., Blaabjerg, F., Liserre, M., & Loh, P. C. (2006). Proportional-resonant controllers and filters for grid-connected voltage-source converters. *IEE Proceedings - Electric Power Applications*, *153*(5), 750–762. http://dx.doi.org/10.1049/ip-epa:20060008.

van Heusden, K., Karimi, A., & Bonvin, D. (2011). Data-driven model reference control with asymptotically guaranteed stability. *International Journal of Adaptive Control and Signal Processing*, *25*(4), 331–351. http://dx.doi.org/10.1002/acs.1212.

van Heusden, K., Karimi, A., & Söderström, T. (2011). On identification methods for direct data-driven controller tuning. *International Journal of Adaptive Control and Signal Processing*, *25*(5), 448–465. http://dx.doi.org/10.1002/acs.1213.

Varriale da Silva, E., & Campestrini, L. (2022). Bias reduction in the optimal controller identification approach through optimal filtering. In *2022 American control conference (ACC)* (pp. 3923–3928). http://dx.doi.org/10.23919/ACC53348.2022.9867539.

Verhoek, C., Abbas, H. S., & Tóth, R. (2023). Direct data-driven LPV control of nonlinear systems: An experimental result. arXiv:2211.17191.

Vinnicombe, G. (1993). Frequency domain uncertainty and the graph topology. *IEEE Transactions on Automatic Control*, *38*(9), 1371–1383. http://dx.doi.org/10.1109/9.237648.

Zanchettin, A. M., Formentin, S., Loddo, R., & Paparella, R. (2018). Direct data-driven control of cavity tuners in particle accelerators. *IFAC-PapersOnLine*, *51*(15), 138–143. http://dx.doi.org/10.1016/j.ifacol.2018.09.104, 18th IFAC Symposium on System Identification SYSID 2018.

Ziegler, J., & Nichols, N. (1942). Optimum settings for automatic controllers. *Transactions of the ASME*, *64*, 759–768.