European Journal of Control xxx (xxxx) xxx

Contents lists available at ScienceDirect

European Journal of Control

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

Optimal direct data-driven control with stability guarantees

Daniela Selvi[a,*], Dario Piga[b], Giorgio Battistelli[c], Alberto Bemporad[d]

[a] Dipartimento di Ingegneria Industriale (DIEF), Università di Firenze, Via di Santa Marta 3, Firenze 50139, Italy
[b] IDSIA Dalle Molle Institute for Artificial Intelligence, SUPSI-USI, Manno 6928, Switzerland
[c] Dipartimento di Ingegneria dell'Informazione (DINFO), Università di Firenze, Via di Santa Marta 3, Firenze 50139, Italy
[d] IMT School for Advanced Studies Lucca, Piazza San Francesco 19, Lucca 55100, Italy

ARTICLE INFO

Article history:
Received 5 March 2020
Revised 31 July 2020
Accepted 11 September 2020
Available online xxx

Recommended by Prof. T Parisini

Keywords:
Model-free control design
Data-driven control
Optimal control

ABSTRACT

For model-free optimal control design, this paper proposes an approach based on optimizing the reference model that is used in direct data-driven controller synthesis. Optimality is defined with respect to suitable cost functions reflecting desired performance and control objectives. We rely on the well-known Virtual Reference Feedback Tuning technique and on a direct control design approach that ensures stability of the resulting closed-loop system. The proposed design method leads to a non-convex optimization problem with a small number of variables that can be easily solved by a global optimizer, such as by particle swarm optimization. The effectiveness of the proposed solution is illustrated in simulation examples.

1. Introduction

Control design based on experimental data has been a research topic of great interest over the last decades. The main goal of *direct* (model-free) data-driven approaches is to avoid the need of deriving a dynamical model of the process to be controlled from physical laws, since this can be a difficult and time-consuming task, requiring one to trade off between the quality of the model and its simplicity [16,20,28]. Indeed, direct methods only use very basic properties of the process to define desired and achievable closed-loop dynamics and to select a suitable parametric controller family. Control design is then performed by minimizing a cost function, computed from experimental data, that penalizes the discrepancy between desired and achieved closed-loop dynamics.

Several contributions and different data-driven design strategies are available in the literature (see, e.g., [1,4,6,10,17,18,26,29,30], and the books [2,21] for an extensive review and treatment on this topic). Specifically, *Virtual Reference Feedback Tuning* (VRFT) [4,6,11,12,26], is a direct data-driven technique that defines the desired complementary sensitivity (i.e., the desired tracking performance) in terms of a stable transfer function, and then minimizes a cost reflecting the discrepancy between the desired reference model and the potential closed-loop behavior.

While VRFT is computationally simple and can be quite straightforwardly extended to deal with issues such as disturbance rejection [9], it has no theoretical guarantees on the stability of the resulting closed-loop system, although techniques for data-driven *controller certification* [7,8,14] have been proposed to address the problem of verifying whether a given controller is stabilizing for an unknown plant on the basis of input-output data collected in open- or closed-loop configuration. Some research work on direct data-driven techniques has accounted for stability requirements within the design procedure (see, e.g., [1,30]). The approach in [30] defines the desired complementary sensitivity in terms of a stable transfer function and aims at designing a controller such that the feedback interconnection with the actual process reproduces the desired reference behavior as closely as possible. Furthermore, a sufficient condition is introduced which uses the collected dataset to determine whether a candidate controller is stabilizing or not. This allows one to theoretically assess the stability of the feedback interconnection between a candidate controller and the unknown plant. However, in case of unstable or stable but nonminimum phase plants, the technique in [30] only allows the refinement of a pre-determined stabilizing controller. The idea of using the collected dataset for both performance and stability purposes is also exploited, in a different way, in the approach proposed in [1]. Such an approach relies on the definition of two reference models, in terms of two stable transfer functions, expressing the desired input sensitivity and complementary (or output) sensitivity. The rationale behind the need for two reference models is that the desired complementary sensitivity defines output performance objectives, whereas the desired input sensitivity is related to input performance and stability requirements. Then, the controller is designed by solving a multi-objective

* Corresponding author.
E-mail address: daniela.selvi@unifi.it (D. Selvi).

https://doi.org/10.1016/j.ejcon.2020.09.005
0947-3580/

problem, with the aim of minimizing the discrepancy between the potential output sensitivity and the desired reference one, as well as the discrepancy between the potential input sensitivity and the desired reference one. Furthermore, a sufficient *a-posteriori* stability test based on the collected data is also introduced in order to determine whether a candidate controller is stabilizing or not. The approach of Battistelli et al. [1] is based on a slightly more complex architecture than the one of van Heusden et al. [30]. However, it does not require any stabilizing controller in the case of stable, possibly nonminimum-phase plants. Moreover, the technique can be straightforwardly extended to the case of unstable plants by resorting to a simple cascaded control architecture.

A critical step of all the aforementioned direct data-driven control techniques is the choice of the reference model. This should reflect the desired closed-loop behavior, but also account for the capability of the underlying unknown process to reproduce such a behavior when the synthesized controller is used to close the loop. Guidelines for choosing a reference model for multiple-input multiple-output (MIMO) plants with fixed-structure centralized or decentralized controllers are presented in [13]. These guidelines only require basic information on the plant, such as relative degree, vague knowledge of the dominant time constants, and non-minimum phase zeros, if any. The case of data-driven control design for single-input single-output (SISO) plants with unknown non-minimum phase zeros is addressed in [5], where a *flexible reference model* with free numerator coefficients is used. These coefficients are then optimized along with the controller in an iterative way. The work in [15] is extended to control of MIMO plants using a parametrized decoupled reference model. Further, [23] proposes a hierarchical architecture, in which an inner controller is first designed from data to match a simple low-performance closed-loop reference model, that is then used as a prediction model in an outer model predictive controller (MPC) designed to enhance tracking performance and enforce input and output constraints. The same architecture is used in [22], where the best model used by MPC to predict the behavior of the inner loop is chosen through iterative experiments. Although this architecture can achieve satisfactory outcomes, choosing a too low-performing reference model for the inner loop could lead the outer MPC to exert aggressive control actions.

In this paper we propose an alternative method to choose the reference model for direct model-free design. The goal is to determine *optimal* reference models, where optimality is defined with respect to a cost function reflecting desired closed-loop performance and control objectives. Preliminary results, only related to the VRFT approach and not accounting for stability guarantees of the resulting closed-loop system, were presented in [27]. We highlight that the proposed method is a general framework for the choice of reference models, which is suitable to being used also in combination with other variants of the VRFT approach. Indeed, in this paper we consider only two direct data-driven design techniques just for example, namely VRFT (due to its simplicity) and the one proposed in [1] (because it allows one to explicitly account for closed-loop stability objectives).

The paper is organized as follows. In Section 2, after briefly describing the VRFT approach, we will provide guidelines for the optimal selection of the reference model. In Section 3, we will focus on stability guarantees. In particular, we will consider the approach of [1] and, after briefly recalling this technique, we will propose a joint optimal selection of both input and output reference models. Section 4 will be devoted to the solution of the optimization problems formulated in Sections 2 and 3 through particle swarm optimization. In Section 5, simulation results, carried out in different scenarios, will be presented. Finally, concluding remarks will be provided in Section 6.

2. Optimal selection of reference model

Our aim is to design an *optimal* controller for an unmodeled process $\mathcal{P}$ with input $u(t)$ and output $y(t)$, $t = 0, 1, 2, \ldots$. For simplicity, we consider the SISO case $u(t), y(t) \in \mathbb{R}$. A dataset of N input and output samples $(u(0), \ldots, u(N-1))$, $(y(0), \ldots, y(N-1))$ is obtained by means of a single open-loop experiment on the plant[1]. We first address the model-free optimal controller synthesis problem by means of the VRFT method [6]. After briefly recalling the VRFT approach in the next section, we will introduce an optimality criterion for selecting the required closed-loop reference model.

2.1. Virtual reference feedback tuning

We want to design a linear discrete-time control law

$$u(t) = C(\varphi, q)(r(t) - y(t)) + c_0(\varphi) r(t) \tag{1}$$

where q is the forward shift operator (i.e., $q\,u(t) = u(t+1)$) and $r(t) \in \mathbb{R}$ is the reference signal to be tracked. Furthermore, we denote by $C(\varphi, q)$ the transfer function of a linear time-invariant (LTI) controller parameterized by vector $\varphi \in \Phi$, $\Phi \subseteq \mathbb{R}^{n_\varphi}$. The parameter vector φ must be tuned in order to comply with control specifications expressed in terms of a stable pre-determined reference model $M(q)$ of the closed-loop system. Note that the feedforward term $c_0(\varphi)$ could be set to 0 if $C(\varphi, q)$ is parameterized to contain an integrator.

Given the reference model $M(q)$, the VRFT approach relies on the so-called *virtual reference* $r_v(t)$, defined as the solution of

$$y(t) = M(q) r_v(t)\,, \tag{2}$$

where $M(q)$ is assumed to be stably invertible. The virtual tracking error corresponding to $r_v(t)$ is defined as

$$e_v(t) = r_v(t) - y(t) \tag{3}$$

and the virtual input $u_v(\varphi, t)$ obtained from $e_v(t)$ and $r_v(t)$ through the controller $C(\varphi, q)$, $c_0(\varphi)$, as

$$u_v(\varphi, t) = C(\varphi, q) e_v(t) + c_0(\varphi) r_v(t). \tag{4}$$

(linear-parameter varying (LPV) or nonlinear controller parameterizations of the control laws could be also adopted here).

In VRFT the parameter vector φ of the controller is synthesized by penalizing the error $\epsilon(\varphi, t) := u(t) - u_v(\varphi, t)$ between the collected input $u(t)$ and the virtual input $u_v(\varphi, t)$

$$\varphi^\star = \arg\min_{\varphi} \sum_{t=0}^{N-1} \ell(\epsilon(\varphi, t))\,, \tag{5}$$

where $\ell : \mathbb{R} \to \mathbb{R}$ is a loss function such that $\ell(\epsilon) > 0$, $\forall \epsilon \in \mathbb{R}$, $\epsilon \neq 0$, and $\ell(0) = 0$ (for example, $\ell(\epsilon) = \epsilon^2$).

2.2. Optimization problem for reference model selection

The VRFT approach relies on choosing *a-priori* the reference model $M(q)$. In this section, we provide a simple formulation to select $M(q)$ according to an optimality criterion.

Let $r(0), \ldots, r(N-1)$, be a representative reference signal that we expect the controller will be asked to track within the application of interest (for example a collection of random steps,

[1] When it is not possible to carry out an open-loop experiment, for example if the plant is unstable, the dataset can be collected by designing a (possibly low-performance) stabilizing controller and carrying out a closed-loop experiment. This case is not dealt in this paper, since it would complicate the notation without adding any substantial difference to our results.

ramps, sinusoids, square waves, etc.). Let the reference model $M(q)$ depend on a parameter vector θ to be optimized, i.e., $M(q) = M(\theta, q)$. For any θ, the corresponding optimal controller $C(\varphi^\star(\theta), q)$, $c_0(\varphi^\star(\theta))$ can be determined by applying the VRFT approach:

$$\varphi^\star(\theta) = \arg\min_{\varphi} \sum_{t=0}^{N-1} (u(t) - u_v(\theta, \varphi, t))^2 , \tag{6}$$

where $u_v(\theta, \varphi, t) = C(\varphi, q)e_v(\theta, t) + c_0(\varphi)r_v(\theta, t)$, $r_v(\theta, t)$ is the solution of $y(t) = M(\theta, q)r_v(\theta, t)$, and $e_v(\theta, t) = r_v(\theta, t) - y(t)$. Then, the optimal parameter vector $\theta^\star$ is selected as

$$\theta^\star = \arg\min_{\theta} J(\theta) , \tag{7}$$

where

$$J(\theta) = \frac{1}{N}\sum_{t=0}^{N-1} W_y(r(t) - y_p(\theta, t))^2 + W_{\Delta u}\Delta u_p^2(\theta, t) + W_{\text{fit}}(u(t) - u_v(\theta, t))^2 . \tag{8}$$

The first two terms in (8) are the typical penalties used in optimization-based control such as MPC [3] and are used to reflect the performance that would be obtained if the reference model $M(\theta, q)$ were perfectly matched by the closed-loop system, i.e., if $y(t) \equiv y_p(\theta, t)$, $u(t) \equiv u_p(\theta, t)$, where

$$y_p(\theta, t) = M(\theta, q)r(t) , \tag{9}$$

is the output that would result as the response of the reference model $M(\theta, q)$ to $r(t)$, and

$$\Delta u_p(\theta, t) = u_p(\theta, t) - u_p(\theta, t-1)$$

is the corresponding input increment, with

$$u_p(\theta, t) = C(\varphi^\star(\theta), q)(r(t) - y_p(\theta, t)) + c_0(\varphi^\star(\theta))r(t) . \tag{10}$$

We point out that, in order to ensure numerical stability of the computation of (10), all the unstable poles of $C(\varphi^\star(\theta), q)$ must be zeros of $1 - M(\theta, q)$. This will be further discussed in Remark 3 in Section 4.1. Tracking error and actuation effort are traded-off via the nonnegative weights $W_y, W_{\Delta u}$. The third term, weighted by the positive hyper-parameter W_{fit}, expresses the ability of the process to match the reference model $M(\theta, q)$ when the controller $C(\varphi^\star(\theta), q)$, $c_0(\varphi^\star(\theta))$ is used. Without loss of generality, W_y can be set to 1, so that only $W_{\Delta u}$ and W_{fit} are left as tuning knobs of the design procedure.

Note that a penalty $W_u(u_r(t) - u_p(\theta, t))^2$ on deviations from input references has been omitted in (8) to leave the approach completely model-free, as often a static model of the open-loop process is used to generate an input reference $u_r(t)$ that is consistent with $r(t)$ in steady-state.

The optimization problem (7) is in general nonlinear and nonconvex, due to the presence of $\varphi^\star(\theta)$ that makes it a bilevel programming problem. However, only a limited number of optimization variables are involved, namely the entries of the vector θ defining the reference model $M(\theta, q)$. In Section 4, we will solve the problem by using particle swarm optimization for the outer optimization layer.

3. Stability guarantees

To take into account internal stability in the controller synthesis procedure introduced in the previous section, we adopt the direct data-driven design approach of [1]. In this technique, an *a-posteriori* stability test is also included to ensure that the feedback interconnection between the designed controller and the unknown plant $\mathcal{P}$ is stable. Performance and stability requirements are jointly addressed by considering two reference models, denoted by $M_r(q)$ and $Q_r(q)$, expressing the desired output and input sensitivity, respectively. After briefly describing the approach of [1], we propose a method for the optimal joint design of $M_r(q)$ and $Q_r(q)$.

3.1. Stability-oriented design via unfalsified control

Let $C(\varphi, q)$ be the transfer function of a LTI controller parameterized by $\varphi \in \Phi$. Let the process $\mathcal{P}$ be described by the (unknown) LTI transfer function $P(d)$ with noise

$$y(t) = P(d)u(t) + n(t).$$

The design procedure of [1] selects φ to comply with control specifications expressed in terms of two stable reference models $M_r(q)$ and $Q_r(q)$. These need to be chosen *a-priori* by the designer so that $M_r(q)$ reflects the desired output performance, but is in general not able to account for internal stability, while the desired input performance, as well as stability requirements, is captured by $Q_r(q)$. The multi-objective goal is to minimize both the discrepancy between $M_r(q)$ and the *potential* output sensitivity $M(\varphi, q)$, and the discrepancy between $Q_r(q)$ and the *potential* input sensitivity $Q(\varphi, q)$. While the potential maps $M(\varphi, q)$ and $Q(\varphi, q)$ are functions of the controller $C(\varphi, q)$ and the unknown plant, and thus cannot be directly computed, such discrepancies can be evaluated from the experimental data as described next.

Consider the fictitious reference $r_f(\varphi, t)$ defined as in [25], i.e.,

$$r_f(\varphi, t) = C(\varphi, q)^{-1}u(t) + y(t) , \quad t = 0, \dots, N-1 . \tag{11}$$

It is easy to check that the collected data $u(t), y(t)$, $t = 0, \dots, N-1$, can also be expressed in terms of the fictitious reference as follows:

$$\begin{aligned} u(t) &= \frac{C(\varphi, q)}{1 + P(q)C(\varphi, q)} r_f(\varphi, t) - \frac{C(\varphi, q)}{1 + P(q)C(\varphi, q)} n(t) \\ &= Q(\varphi, q)r_f(\varphi, t) - Q(\varphi, q)n(t) \\ y(t) &= \frac{P(q)C(\varphi, q)}{1 + P(q)C(\varphi, q)} r_f(\varphi, t) + \frac{1}{1 + P(q)C(\varphi, q)} n(t) \\ &= M(\varphi, q)r_f(\varphi, t) + (1 - M(\varphi, q))n(t) , \end{aligned}$$

where the disturbance $n(t)$ is independent of the input signal $u(t)$ (we recall that the input and output data are collected through an open-loop experiment). Then, the discrepancies between $M(\varphi, q)$, $Q(\varphi, q)$ and, respectively, $M_r(q)$ and $Q_r(q)$, can be evaluated by means of the cost functions

$$Z_N(\varphi) := \sum_{t=0}^{N-1} (u(t) - u^\circ(\varphi, t))^2 \tag{12}$$

$$V_N(\varphi) := \sum_{t=0}^{N-1} (y(t) - y^\circ(\varphi, t))^2 , \tag{13}$$

where the signals $u^\circ(\varphi, t)$ and $y^\circ(\varphi, t)$ are defined as

$$\begin{aligned} u^\circ(\varphi, t) &:= Q_r(q)r_f(\varphi, t) \\ y^\circ(\varphi, t) &:= M_r(q)r_f(\varphi, t) . \end{aligned}$$

In fact, in the noise-free case ($n(t) = 0$, $\forall t \geq 0$), we have

$$\begin{aligned} u(t) - u^\circ(\varphi, t) &= (Q(\varphi, q) - Q_r(q))r_f(\varphi, t) \\ y(t) - y^\circ(\varphi, t) &= (M(\varphi, q) - M_r(q))r_f(\varphi, t) . \end{aligned}$$

Note that the computation of the fictitious reference is not needed in order to obtain the cost functions (12) and (13) since we also have:

$$u(t) - u^\circ(\varphi, t) = u(t) - Q_r(q)y(t) - C^{-1}(\varphi, q)Q_r(q)u(t) \tag{14}$$

$$y(t) - y^{\circ}(\varphi, t) = [1 - M_r(q)]y(t) - C^{-1}(\varphi, q)M_r(q)u(t)\,. \tag{15}$$

Note that (14) and (15) have the same form of a predictor in an output-error method [20,28], even if the best matching error is not a white noise. Moreover, the disturbance $n(t)$ only appears in additive terms that do not depend on φ in (14) and (15). Finally, the transfer functions involved in the above computations can be made stable by selecting the controller structure and the reference models in accordance with internal stability specifications. To this end, we consider controllers of the form

$$C(\varphi, q) = \frac{\bar{S}(\varphi, q)S_u(q)}{\bar{R}(\varphi, q)R_u(q)}\,, \tag{16}$$

where the polynomials $S_u(q)$ and $R_u(q)$ contain all unstable roots (thus allowing one to directly impose rejection and tracking objectives, e.g., through an integral action), and the polynomials $\bar{S}(\varphi, q)$ and $\bar{R}(\varphi, q)$ have to be designed, with the constraint that $\bar{S}(\varphi, q)$ must be stable. Then, the recursive computation of (14) is numerically stable provided that the input sensitivity reference model $Q_r(q)$ is supposed to be factorized as $Q_r(q) = S_u(q)\bar{Q}_r(q)$, with $\bar{Q}_r(q)$ stable. Similarly, the recursive computation of (15) is numerically stable when the output sensitivity reference model $M_r(q)$ takes the form $M_r(q) = S_u(q)\bar{M}_r(q)$, with $\bar{M}_r(q)$ stable.

The design procedure relies on the idea of minimizing a combination of the two cost functions $Z_N(\varphi)$ and $V_N(\varphi)$. In fact, minimization of $V_N(\varphi)$ alone would suffer from the same problem as VRFT, in that it does not account for the internal stability objective. On the other hand, the discrepancy $u - u^{\circ}$ is related to the internal stability requirements, but minimization of $Z_N(\varphi)$ alone would not account for the objective related to the desired output performance.

To see that minimization of $Z_N(\varphi)$ is related to the internal stability requirements, consider the controller $C_r(q)$ exactly achieving the reference model $Q_r(q)$ for an unknown stable plant $P(q)$, i.e.,

$$C_r(q) = \frac{Q_r(q)}{1 - P(q)Q_r(q)}\,. \tag{17}$$

It can be checked that, in the noise-free case, the input discrepancy can be also expressed as

$$u(t) - u^{\circ}(\varphi, t) = \Delta_Q(\varphi, q)Q_r(q)u(t) \tag{18}$$

with $\Delta_Q(\varphi, q) := C_r^{-1}(q) - C^{-1}(\varphi, q)$. Hence, minimization of $Z_N(\varphi)$ tends to make the quantity $\Delta_Q(\varphi, q)Q_r(q)$ small. In turn, by means of small gain arguments, it can be shown that the condition

$$\|Q_r(q)\Delta_Q(\varphi, q)\|_{\infty} < 1 \tag{19}$$

is sufficient to ensure that the controller $C(\varphi, q)$ internally stabilizes the unknown plant, provided that $\bar{S}(\varphi, q)$ is stable (a formal proof is provided in [1]).

Finally, (18) and (19) are further exploited in order to derive an a-posteriori stability test. In fact, in the ideal situation of a noise-free infinite-length data set, from (18) it follows that

$$\|Q_r(q)\Delta_Q(\varphi, q)\|_{\infty} = \sup_{\omega\in[-\pi,\,\pi]} \frac{|\hat{u}(\omega) - \hat{u}^{\circ}(\varphi, \omega)|}{|\hat{u}(\omega)|}\,, \tag{20}$$

where $\hat{u}(\omega)$ and $\hat{u}^{\circ}(\varphi, \omega)$ are the Discrete Fourier Transforms of $u(t)$ and $u^{\circ}(\varphi, t)$, respectively (assuming that $|\hat{u}(\omega)| > 0$ for any $\omega \in [-\pi, \pi]$). Then, a stability test can readily be defined by applying standard non-parametric identification techniques [20], e.g., the windowed Empirical Transfer Function Estimate (ETFE), so as to estimate $\|Q_r(q)\Delta_Q(\varphi, q)\|_{\infty}$ from the input discrepancy $u(t) - u^{\circ}(\varphi, t)$.

3.2. Optimal selection of $Q_r(q)$ and $M_r(q)$

We establish here a procedure for an "optimal" selection of both $M_r(q)$ and $Q_r(q)$, where optimality is defined in terms of a criterion which accounts for both performance and stability.

As in Section 2.2, let $r(0), \ldots, r(N-1)$, be a reference signal chosen by the designer consistently with the specific application. Let the desired output and input sensitivities depend on parameter vectors θ and, respectively, ρ to be optimized, i.e., $M_r(q) = M_r(\theta, q)$ and $Q_r(q) = Q_r(\rho, q)$. Further, let $C_M(\varphi, q)$ and $C_Q(\psi, q)$ be two controllers depending on parameter vectors φ and ψ, respectively. Typically, the two controllers $C_M(\varphi, q)$ and $C_Q(\psi, q)$ will have the same parametric structure, but this is not strictly necessary for our developments. In the following we will assume that both $C_M(\varphi, q)$ and $C_Q(\psi, q)$ have the form (16), i.e.,

$$C_M(\varphi, q) = \frac{\bar{S}_M(\varphi, q)S_{u,M}(q)}{\bar{R}_M(\varphi, q)R_{u,M}(q)} \tag{21}$$

$$C_Q(\psi, q) = \frac{\bar{S}_Q(\psi, q)S_{u,Q}(q)}{\bar{R}_Q(\psi, q)R_{u,Q}(q)}\,, \tag{22}$$

with $S_{u,M}(q)$, $R_{u,M}(q)$, $S_{u,Q}(q)$, and $R_{u,Q}(q)$ containing all unstable roots, and the polynomials $\bar{S}_M(\varphi, q)$, $\bar{R}_M(\varphi, q)$, $\bar{S}_Q(\psi, q)$, and $\bar{R}_Q(\psi, q)$ to be designed, under the condition that $\bar{S}_M(\varphi, q)$ and $\bar{S}_Q(\psi, q)$ must be stable. For any choice of θ, the corresponding optimal controller $C_M(\varphi^{\star}(\theta), q)$ can be determined by re-writing the output discrepancy (15) as

$$y(t) - y^{\circ}(\theta, \varphi, t) = [1 - M_r(\theta, q)]\,y(t) - C_M^{-1}(\varphi, q)\,M_r(\theta, q)\,u(t) \tag{23}$$

and then applying prediction-error methods (PEM) for output error models to minimize the corresponding cost $V_N(\theta, \varphi)$ with respect to φ. Similarly, for any ρ, the input discrepancy (14) is re-written as

$$u(t) - u^{\circ}(\rho, \psi, t) = u(t) - Q_r(\rho, q)\,y(t) - C_Q^{-1}(\psi, q)\,Q_r(\rho, q)\,u(t). \tag{24}$$

Then, by applying the output error method, the optimal controller $C_Q(\psi^{\star}(\rho), q)$ that minimizes $Z_N(\rho, \psi)$ with respect to ψ can be found.

For what concerns the optimal parameter vectors $\theta^{\star}$ and $\rho^{\star}$, they are determined by means of an optimization procedure according to the following criterion:

$$(\theta^{\star}, \rho^{\star}) = \arg\min_{(\theta,\,\rho)} J(\theta, \rho)\,, \tag{25}$$

where

$$\begin{aligned} J(\theta, \rho) := \frac{1}{N}\sum_{t=0}^{N-1} \big\{ & W_y(r(t) - y_p(\theta, t))^2 + W_{\Delta u}\Delta u_p^2(\theta, t) \\ & + W_{\mathrm{fit},M}(y(t) - y^{\circ}(\theta, \varphi^{\star}(\theta), t))^2 \\ & + W_{\mathrm{fit},Q}(u(t) - u^{\circ}(\rho, \psi^{\star}(\rho), t))^2 \\ & + W_{MQ}(u_p(\theta, t) - \bar{u}_p(\rho, t))^2 \big\}\,. \end{aligned} \tag{26}$$

At each iteration of the optimization procedure (25), the current θ and ρ are employed in (23) and (24) in order to determine the parameters $\varphi^{\star}(\theta)$ and $\psi^{\star}(\rho)$. Similarly to the cost (8) in Section 2.2, the first two terms in (26) reflect a trade off between tracking errors and actuation efforts. The quantities

$$y_p(\theta, t) = M_r(\theta, q)r(t)\,, \tag{27}$$

$$u_p(\theta, t) = C_M(\varphi^{\star}(\theta), q)(r(t) - y_p(\theta, t))$$

$$= C_M(\varphi^\star(\theta), q)(1 - M_r(\theta, q))r(t)\,, \tag{28}$$

are related to the performance that would be obtained if the desired output sensitivity $M_r(\theta, q)$ were exactly achieved. The third and fourth terms in (26) correspond to the cost functions (13) and (12) introduced in Section 3.1, where the discrepancies $u(t) - u^\circ(\rho, \psi^\star(\rho), t)$ and $y(t) - y^\circ(\theta, \varphi^\star(\theta), t)$ are computed as in (24) and (23), respectively. Finally, the last term in (26) reflects the discrepancy between $u_p(\theta, t)$ and

$$\overline{u}_p(\rho, t) := Q_r(\rho, q)r(t)\,. \tag{29}$$

Recall that controller $C_Q(\psi^\star(\rho), q)$ is designed so as to achieve an input sensitivity close to $Q_r(\rho, q)$. As a result of the minimization of $V_N(\theta, \varphi)$, controller $C_M(\varphi^\star(\theta), q)$ achieves an input sensitivity close to $C_M(\varphi^\star(\theta), q)(1 - M_r(\theta, q))$. Hence, when the last term in the cost is small, the two controllers $C_Q(\psi^\star(\rho), q)$ and $C_M(\varphi^\star(\theta), q)$ provide a similar control input for tracking the same reference signal $r(t)$. Thus, the meaning of the last term in (26) is to make the two controllers $C_Q(\psi^\star(\rho), q)$ and $C_M(\varphi^\star(\theta), q)$ perform a similar action.

The optimization of the cost function in (25) must be constrained so that both $M_r(\theta, q)$ and $Q_r(\rho, q)$ are stable. Moreover, in order to ensure that the computation of cost $J(\theta, \rho)$ is numerically stable (namely, the involved signals do not diverge during recursive computation of the terms in (26)), the reference models $M_r(\theta, q)$ and $Q_r(\rho, q)$, and the parametric controllers $C_M(\varphi, q)$ and $C_Q(\psi, q)$ have to be chosen so that:

1. all unstable zeros of $C_M(\varphi, q)$ are zeros of $M_r(\theta, q)$;
2. all unstable poles of $C_M(\varphi, q)$ are zeros of $1 - M_r(\theta, q)$;
3. all unstable zeros of $C_Q(\psi, q)$ are zeros of $Q_r(\rho, q)$.

Note that Conditions 1 and 3 are needed to ensure numerical stability of (23) and (24), respectively, whereas Condition 2 requires that $C_M(\varphi, q)$ and $M_r(\theta, q)$ are consistent with each other. This issue will be further considered in Remark 5 in Section 4.2.

Once the optimization of criterion (25) terminates, two controllers are available, namely $C_M(\varphi^\star(\theta^\star), q)$ and $C_Q(\psi^\star(\rho^\star), q)$. Specifically, $C_M(\varphi^\star(\theta^\star), q)$ is designed so that, if it is put in feedback with the unknown process, the corresponding closed-loop map matches $M_r(\theta^\star, q)$ as closely as possible (recall (23)). On the other hand, $C_Q(\psi^\star(\rho^\star), q)$ is designed so that, if it is put in feedback with the unknown process, the corresponding input-sensitivity map matches $Q_r(\rho^\star, q)$ as closely as possible (recall (24)). Therefore, it is expected that $C_Q(\psi^\star(\rho^\star), q)$ complies with the stability test (see Section 3.1)

$$\sup_{\omega\in[-\pi,\pi]} \frac{|\hat{u}(\omega) - \hat{u}^\circ(\rho^\star, \psi^\star(\rho^\star), \omega)|}{|\hat{u}(\omega)|} < 1\,. \tag{30}$$

If (30) does not hold, then the overall procedure has to be revised (e.g., the controller family is not rich enough, a different choice for the weights in (26) is needed, etc.). On the other hand, when (30) holds, $C_Q(\psi^\star(\rho^\star), q)$ is a candidate controller for the unknown process.

The stability test can also be applied to $C_M(\varphi^\star(\theta^\star), q)$, since the last term in the cost tends to make $Q_r(\rho^\star, q)$ and $C_M(\varphi^\star(\theta^\star), q)(1 - M_r(\theta^\star, q))$ close to each other. In this case, the input discrepancy must be computed as

$$\begin{aligned} u(t) - u(\theta^\star, \rho^\star, t) = u(t) - Q_r(\rho^\star, q)\, y(t) \\ - C_M^{-1}(\varphi^\star(\theta^\star), q)\, Q_r(\rho^\star, q)\, u(t) \end{aligned} \tag{31}$$

and the stability test becomes

$$\sup_{\omega\in[-\pi,\pi]} \frac{|\hat{u}(\omega) - \hat{u}(\theta^\star, \rho^\star, \omega)|}{|\hat{u}(\omega)|} < 1 \tag{32}$$

If both controllers pass the corresponding stability tests, then $C_M(\varphi^\star(\theta^\star), q)$ is selected, as its performance is deemed to be closer to the desired behavior expressed by $M_r(\theta^\star, q)$.

Remark 1. The left-hand side of both (30) and (32) can be estimated by means of non-parametric identification techniques (e.g. ETFE). Then, in order to deal with the error in the estimation due to non-ideal conditions (disturbances and finite-length experiment), the test will be assumed to be passed when such an estimate is less than $1 - \xi$, where the positive scalar $\xi < 1$ is a design parameter.

Remark 2. Condition (30) can be imposed for a generic ρ and directly employed as an additional constraint within the optimization problem (25). In this case, the term $W_{\text{fit,Q}}(u(t) - u^\circ(\rho, \psi^\star(\rho), t))^2$ could be removed from the cost (26), i.e., $W_{\text{fit,Q}}$ can be set to 0. Similarly, (32) could also be imposed for any θ as an additional constraint within the optimization problem (25). This could possibly decrease the probability of getting a feasible solution. However, if a feasible solution is obtained, then both $C_Q(\psi^\star(\rho^\star), q)$ and $C_M(\varphi^\star(\theta^\star), q)$ are stabilizing for the unknown process, thus the latter can be selected.

4. Global optimization of reference models

In this section, we discuss how to solve the optimization problems (7) and (25) through particle swarm optimization (PSO). This technique is well-suited for the problem of interest, which is formulated in terms of a limited number of optimization variables and aims at finding a global optimizer. The reader interested in PSO is referred, e.g., to [24].

4.1. Global optimization of reference model in VRFT

We first consider the framework described in Section 2 for an optimal selection of the reference model $M(\theta, q)$ within the VRFT approach. Specifically, we consider the following parameterization for the reference model:

$$M(\theta, q) = K\, \frac{\prod_{\ell=1}^{n_{zr}}(q - z_\ell)\prod_{\ell=n_{zr}+1}^{n_{zr}+n_{zc}}(q - z_\ell)(q - z_\ell^*)}{\prod_{\ell=1}^{n_{pr}}(q - p_\ell)\prod_{\ell=n_{pr}+1}^{n_{pr}+n_{pc}}(q - p_\ell)\left(q - p_\ell^*\right)}\,, \tag{33}$$

where * denotes complex conjugate, and n_{zr}, n_{zc}, n_{pr}, n_{pc} denote the number of real zeros, complex conjugate zeros, real poles, and complex conjugate poles, respectively. In defining (33) we set $\prod_{\ell=\ell_1}^{\ell_2} = 1$ if $\ell_1 > \ell_2$. The vector θ to be optimized is defined by stacking the following parameters:

$$\begin{aligned} &z_\ell, && \ell = 1, \cdots, n_{zr};\\ &Re\{z_\ell\}, Im\{z_\ell\}, && \ell = n_{zr}+1, \cdots, n_{zr}+n_{zc};\\ &p_\ell, && \ell = 1, \cdots, n_{pr};\\ &Re\{p_\ell\}, Im\{p_\ell\}, && \ell = n_{pr}+1, \cdots, n_{pr}+n_{pc}. \end{aligned}$$

The term K is only needed to enforce $M(\theta, 1) = 1$ (i.e., unitary steady-state gain), and is not treated as an optimization variable. As discussed in Section 2.1, VRFT requires the stability of both the reference model $M(\theta, q)$ and of its inverse. In order to comply with these requirements, a possible solution is to add a function $b(\cdot)$ to the computation of $J(\theta)$, with $b : \mathbb{R} \to \mathbb{R}$, which penalizes the violation of such conditions. Among different possible choices for b, the following piecewise-polynomial function will be used in the examples shown in Section 5:

$$b(h) = \begin{cases} 0 & \text{if} \quad h < 0 \\ \sqrt{k}h & \text{if} \quad 0 \le h < 1 \\ \sqrt{k}h^2 & \text{if} \quad h \ge 1 \end{cases} \tag{34}$$

where k denotes the current PSO iteration, and $h : \mathbb{R} \to \mathbb{R}$ is such that

$$\begin{aligned} h_\ell^z(\theta) &= |z_\ell|^2 - 1, \qquad \ell = 1, \ldots, n_{zr} + n_{zc} \\ h_\ell^p(\theta) &= |p_\ell|^2 - 1, \qquad \ell = 1, \ldots, n_{pr} + n_{pc}\,. \end{aligned} \tag{35}$$

Algorithm 1 VRFT approach with reference model selection through PSO.

Input: number of particles N_{part}, maximum number of iterations $k_{\max}$; positive weights W_y, W_{Δ_u}, W_{fit}; penalty function b.

1. **Populate** particle swarm θ^i, $i = 1, \dots, N_{\text{part}}$, with random initial values under constraints $h^z_\ell(\theta) < 0$, $\ell = 1, \dots, n_{zr} + n_{zc}$, and $h^p_\ell(\theta) < 0$, $\ell = 1, \dots, n_{pr} + n_{pc}$;
2. **for** $k = 1, \dots, k_{\max}$ **do**
   for $i = 1, \dots, N_{\text{part}}$ **do**
   parametrize $M(\theta^i, q)$ as in (33);
   set K such that $M(\theta^i, 1) = 1$;
   compute $C(\varphi^\star(\theta^i), q)$, $c_0(\varphi^\star(\theta^i))$ through VRFT;
   set the cost function

$$J(\theta^i) = \frac{1}{N}\sum_{t=0}^{N-1}\Big\{W_y(r(t) - y_p(\theta^i, t))^2 + W_{\Delta u}\Delta u_p^2(\theta^i, t) + W_{\text{fit}}(u(t) - u_v(\theta^i, t))^2 + \sum_{\ell=1}^{n_{zr}+n_{zc}} b(h^z_\ell(\theta^i)) + \sum_{\ell=1}^{n_{pr}+n_{pc}} b(h^p_\ell(\theta^i))\Big\} \quad (36)$$

   end for
   choose the best particle based on the computed cost functions $J(\theta^i)$, $i = 1, \dots, N_{\text{part}}$;
   update the position of particles as in [24, Algorithm 1];
   end for

Output: best particle (reference model parameters) $\theta^\star$, controller parameters $\varphi^\star(\theta^\star)$.

In order to solve the optimization problem through PSO we consider a certain number N_{part} of particles (each one represented by a parameter vector θ^i) and compute the controller parameters $\varphi^\star(\theta^i)$ associated to each particle through the VRFT approach. The position of the particles is updated iteratively for a pre-defined number of iterations $k_{\max}$ according to common rules in PSO. When the algorithm terminates, the best particle $\theta^\star$ is provided as the one corresponding to the minimum value of $J(\theta^i)$, $i = 1, \dots, N_{\text{part}}$. Algorithm 1 summarizes the overall procedure.

Remark 3. When each $M(\theta^i, q)$, $i = 1, \dots, N_{\text{part}}$, is parameterized as in (33) with K such that $M(\theta^i, 1) = 1$, numerical stability of the computation of $u_p(\theta^i, t)$ in (10) is ensured if $C(\varphi^\star(\theta^i), q)$ contains at most one integrator and has all the other poles inside the unit circle.

4.2. Global optimization of $Q_r(q)$ and $M_r(q)$

We address now the optimal design of the output and input reference maps within the framework described in Section 3. For this purpose, we consider the following parameterization

$$M_r(\theta, q) = K_M \frac{\prod_{\ell=1}^{n^M_{zr}}(q - z_{\ell,M})\prod_{\ell=n^M_{zr}+1}^{n^M_{zr}+n^M_{zc}}(q - z_{\ell,M})\left(q - z^*_{\ell,M}\right)}{\prod_{\ell=1}^{n^M_{pr}}(q - p_{\ell,M})\prod_{\ell=n^M_{pr}+1}^{n^M_{pr}+n^M_{pc}}(q - p_{\ell,M})\left(q - p^*_{\ell,M}\right)} \quad (37)$$

$$Q_r(\rho, q) = K_Q \frac{\prod_{\ell=1}^{n^Q_{zr}}(q - z_{\ell,Q})\prod_{\ell=n^Q_{zr}+1}^{n^Q_{zr}+n^Q_{zc}}(q - z_{\ell,Q})\left(q - z^*_{\ell,Q}\right)}{\prod_{\ell=1}^{n^Q_{pr}}(q - p_{\ell,Q})\prod_{\ell=n^Q_{pr}+1}^{n^Q_{pr}+n^Q_{pc}}(q - p_{\ell,Q})\left(q - p^*_{\ell,Q}\right)}. \quad (38)$$

While K_Q in (38) is also an optimization variable, K_M in (37) is only needed to enforce $M_r(\theta, 1) = 1$ and thus it is not an optimization variable. Further, n^M_{zr}, n^M_{zc}, n^M_{pr}, and n^M_{pc} denote the number of real zeros, complex conjugate zeros, real poles, and complex conjugate poles, respectively, of the reference map $M_r(\theta, q)$. Then, the vector θ to be optimized is defined by stacking the following parameters:

$$\begin{aligned}
&z_{\ell,M}, && \ell = 1, \cdots, n^M_{zr};\\
&Re\{z_{\ell,M}\}, Im\{z_{\ell,M}\}, && \ell = n^M_{zr}+1, \cdots, n^M_{zr}+n^M_{zc};\\
&p_{\ell,M}, && \ell = 1, \cdots, n^M_{pr},\\
&Re\{p_{\ell,M}\}, Im\{p_{\ell,M}\}, && \ell = n^M_{pr}+1, \cdots, n^M_{pr}+n^M_{pc}.
\end{aligned}$$

Similarly, n^Q_{zr}, n^Q_{zc}, n^Q_{pr}, and n^Q_{pc} in (38) denote the number of real zeros, complex conjugate zeros, real poles, and complex conjugate poles, respectively, of the reference map $Q_r(\rho, q)$. Then, the vector ρ is defined by stacking the following parameters:

$$\begin{aligned}
&K_Q;\\
&z_{\ell,Q}, && \ell = 1, \cdots, n^Q_{zr};\\
&Re\{z_{\ell,Q}\}, Im\{z_{\ell,Q}\}, && \ell = n^Q_{zr}+1, \cdots, n^Q_{zr}+n^Q_{zc};\\
&p_{\ell,Q}, && \ell = 1, \cdots, n^Q_{pr};\\
&Re\{p_{\ell,Q}\}, Im\{p_{\ell,Q}\}, && \ell = n^Q_{pr}+1, \cdots, n^Q_{pr}+n^Q_{pc}.
\end{aligned}$$

A penalty function $b(h)$ of the form (34) is added to the computation of $J(\theta, \rho)$ in order to enforce $M_r(\theta, q)$ and $Q_r(\rho, q)$ to be stable, with $h : \mathbb{R} \to \mathbb{R}$ such that

$$\begin{aligned}
h^p_\ell(\theta) &= |p_{\ell,M}|^2 - 1, \quad \ell = 1, \dots, n^M_{pr} + n^M_{pc}\\
h^p_\ell(\rho) &= |p_{\ell,Q}|^2 - 1, \quad \ell = 1, \dots, n^Q_{pr} + n^Q_{pc}.
\end{aligned} \quad (39)$$

Recalling Remark 1 and Remark 2, conditions (30) and (32) can also be imposed as additional requirements, which in turn can be expressed in terms of penalty functions $b(h)$ of the form (34). In this respect, we can denote by $\mathscr{E}_Q$ the estimation of the left-hand side of (30), and by $\mathscr{E}_M$ the estimation of the left-hand side of (32), and define

$$h_{\mathscr{E},Q}(\rho) := \mathscr{E}_Q - 1 + \xi \quad (40)$$

$$h_{\mathscr{E},M}(\theta) := \mathscr{E}_M - 1 + \xi\,. \quad (41)$$

The overall procedure is summarized in Algorithm 2.

Remark 4. In practice, it is reasonable to require that $M_r(\theta, q)$ and $Q_r(\rho, q)$ share the same denominator. In this case, the actual parameter vector to be optimized is composed by stacking the following terms:

$$\begin{aligned}
&z_{\ell,M}, && \ell = 1, \dots, n^M_{zr};\\
&Re\{z_{\ell,M}\}, Im\{z_{\ell,M}\}, && \ell = n^M_{zr}+1, \dots, n^M_{zr}+n^M_{zc};\\
&p_{\ell,M}, && \ell = 1, \dots, n^M_{pr};\\
&Re\{p_{\ell,M}\}, Im\{p_{\ell,M}\}, && \ell = n^M_{pr}+1, \dots, n^M_{pr}+n^M_{pc};\\
&K_Q;\\
&z_{\ell,Q}, && \ell = 1, \dots, n^Q_{zr};\\
&Re\{z_{\ell,Q}\}, Im\{z_{\ell,Q}\}, && \ell = n^Q_{zr}+1, \dots, n^Q_{zr}+n^Q_{zc}.
\end{aligned}$$

Accordingly, the number of optimization variables in cost (26) is actually reduced to $n_{\text{var}} := n^M_{zr} + n^M_{zc} + n^M_{pr} + n^M_{pc} + n^Q_{zr} + n^Q_{zc} + 1$.

Remark 5. When each controller $C_M(\varphi(\theta^i), q)$, $i = 1, \dots, N_{\text{part}}$, is equipped with an integral action, then Condition 2 of Section 3.2 holds thanks to the parameterization (37) with K_M such that $M_r(\theta^i, 1) = 1$. On the other hand, if some design specifications require that $R_{u,M}(q)$ in (21) contains additional unstable poles, then Condition 2 of Section 3.2 requires that they are also zeros of $1 - M_r(\theta^i, q)$, thus the parameterization (37) has to be modified accordingly. Furthermore, if nonminimum-phase zeros need to be included in $S_{u,M}(q)$, they also have to be imposed as

Algorithm 2 Data-driven control design with guaranteed stability: selection of input and output maps through PSO.

Input: number of particles N_{part}, maximum number of iterations $k_{\max}$; positive weights W_y, W_{Δ_u}, $W_{\text{fit},M}$, $W_{\text{fit},Q}$, W_{MQ}; penalty function b.

1. **Populate** particle swarm $\text{col}\{\theta^i, \rho^i\}$, $i = 1, \dots, N_{\text{part}}$,with random initial values under constraints $h^p_\ell(\theta) < 0$, $\ell = 1, \dots, n^M_{pr} + n^M_{pc}$, and $h^p_\ell(\rho) < 0$, $\ell = 1, \dots, n^Q_{pr} + n^Q_{pc}$;
2. **for** $k = 1, \dots, k_{\max}$ **do**
   for $i = 1, \dots, N_{\text{part}}$ **do**
   parametrize $M_r(\theta^i, q)$ as in (37) and $Q_r(\rho^i, q)$ as in (38);
   set K_M such that $M_r(\theta^i, 1) = 1$;
   compute $C_M(\varphi^\star(\theta^i), q)$ and $C_Q(\psi^\star(\rho^i), q)$ through output error method;
   set the cost function

$$
\begin{aligned}
J(\theta^i, \rho^i) = \frac{1}{N} \sum_{t=0}^{N-1} \Big\{ & W_y (r(t) - y_p(\theta^i, t))^2 \\
& + W_{\Delta u} \Delta u_p^2(\theta^i, t) \\
& + W_{\text{fit},M} (y(t) - y^\circ(\theta^i, \varphi^\star(\theta^i)), t))^2 \\
& + W_{\text{fit},Q} (u(t) - u^\circ(\rho^i, \psi^\star(\rho^i)), t))^2 \\
& + W_{MQ} (u_p(\theta^i, t) - \overline{u}_p(\rho^i, t))^2 \\
& + \sum_{\ell=1}^{n^M_{pc}+n^M_{pr}} b(h^p_\ell(\theta^i)) + \sum_{\ell=1}^{n^Q_{pc}+n^Q_{pr}} b(h^p_\ell(\rho^i)) \\
& + b(h_{\mathscr{E},Q}(\rho^i)) + b(h_{\mathscr{E},M}(\theta^i)) \Big\} \qquad (42)
\end{aligned}
$$

   end for
   choose the best particle based on the computed cost functions $J(\theta^i, \rho^i)$, $i = 1, \dots, N_{\text{part}}$;
   update the position of particles as in [24, Algorithm 1];
   end for

Output: best particle (reference model parameters) $\text{col}\{\theta^\star, \rho^\star\}$,controller parameters $\varphi^\star(\theta^\star)$ and $\psi^\star(\rho^\star)$.

fixed zeros in (37) by Condition 1 of Section 3.2. Similarly, if it is required that $C_Q(\psi(\rho^i), q)$ in (22) is equipped with nonminimum-phase zeros (included in $S_{u,Q}(q)$), they also have to be imposed as fixed zeros in (38) by Condition 3 of Section 3.2.

Remark 6. We point out that the stability tests (30) and (32) are always used, taking into account the considerations reported in Remark 1, as an additional *a-posteriori* check of the actual feasibility of the solution provided by Algorithm 2. This check is particularly important when it is necessary to relax the optimization problem by removing either one or both of the last two terms in the general cost (42) (for example, if no feasible solution has been found by optimizing (42)). We underline that the general form of the cost (42) has been used in all the simulation examples reported in Section 5, leading to feasible solutions for Algorithm 2.

4.3. Design guidelines and final remarks

We conclude this section with some comments and design guidelines. First, it is important to highlight that the proposed method, as any model-free design technique, aims at providing an effective alternative to classical model-based procedures when it is difficult or time-consuming to derive a model through identification which is simple and reliable enough for model-based controller synthesis. In fact, whenever it is possible to derive a process model which achieves a good trade-off between simplicity and reliability, classical design techniques based on the derived model, such as the linear quadratic Gaussian control design, are able to obtain very satisfactory results. On the other hand, the aim of the proposed method is to achieve similar performance even when the conditions ensuring satisfactory outcomes for model-based optimal techniques do not hold. It is important to underline that, in model-free methods, any design choice has to be made on the basis of no (or limited) information about the process. In this respect, in our technique the choice of the order of both controller(s) and reference model(s) is made by means of a tuning procedure. More in details, a low order is first selected for both controller(s) and reference model(s); then, the degrees of freedom are possibly increased on the basis of the achieved performance. While starting from an initial setting and possibly increasing the degrees of freedom in order to achieve better results is quite a standard procedure, it is worth underlining that such a procedure implies to test the designed controller on the real process, and can thus be adopted in practical applications only if such controller is guaranteed to be stabilizing. Therefore, with specific focus on Algorithm 1 and Algorithm 2, we underline that the above mentioned procedure can be safely adopted, in practical applications, only within Algorithm 2. We further highlight that Algorithm 1 has to be considered as a first step towards an automatic and optimal (in terms of a certain criterion) choice of the output reference model for model-free control design techniques, such as the Virtual Reference Feedback Tuning. On the other hand, the more complex Algorithm 2 is oriented to overcome the limitations of Algorithm 1; since any feasible solution of Algorithm 2 corresponds to a stabilizing controller, such controller can be safely put in feedback with the real process, therefore also allowing for an actual tuning of the design parameters.

In the next section concerning simulation results, we will discuss the choice of the weights in (8) and (26).

5. Simulation examples

In this section, we test the proposed approach in three different simulation examples. In all the examples, we report the explicit form of reference models and controllers, with their coefficients approximated to the fourth decimal place; we point out that this may reflect in some differences in the outcomes of simulation tests when the approximated versions are used in place of the original controllers/reference models.

5.1. Example 1

We consider a simple nonlinear Wiener process. Let $f(\cdot) : \mathbb{R} \to \mathbb{R}$ be a static nonlinear function such that

$$
f(y_L(t)) = |y_L(t)| \arctan(y_L(t)) , \qquad (43)
$$

where $y_L(t)$ is obtained from the LTI, asymptotically stable, and minimum-phase process $\mathcal{P}$ described by the following transfer function

$$
P(q) = \frac{q - 0.45}{(q - 0.55)(q + 0.65)} . \qquad (44)
$$

The model (44) and the nonlinear function $f(\cdot)$ are assumed to be unknown by our proposed model-free design procedure. Before addressing the model-free control design problem on system (43), we first discard the nonlinear part, i.e., we consider (44) as the actual process to be controlled. This linear-only case has to be intended as a preliminary benchmark for testing the proposed design method in ideal conditions. In fact, we recall that the theoretical results behind the proposed method hold for linear processes; on the other hand, the simulation analysis reported in this example

aims to show that our technique can also deal with simple nonlinearities in the plant. For this purpose, in this example we are interested in evaluating the outcome of the proposed design method in the linear-only case (44) and for the overall Wiener process (43), using the same algorithm settings in the two cases.

Case 1 - A dataset of $N = 5000$ input and output samples is collected via an open-loop experiment, carried out by exciting the process (44) with a white Gaussian signal $u(t)$ with zero mean and standard deviation $\sigma_u = 1$. The measured output signal $y(t) \equiv y_L(t)$ is corrupted by a white Gaussian noise $n(t)$ with zero mean and standard deviation $\sigma_n = 10^{-2}$. The sampling time T_s is 0.1 s. A pseudo-random binary signal taking values in $\{-1, 1\}$ is used as the reference signal $r(t)$ in both (8) and (26). The specific set-up for Algorithm 1 and Algorithm 2 (chosen also in view of the general Wiener case) is detailed below. In Algorithm 1, the number of particles is set to $N_{\text{part}} = 20$, and the maximum number of iterations to $k_{\max} = 100$. The weights in (8) are $W_y = 1$, $W_{\Delta u} = 1.5$, $W_{\text{fit}} = 30$. At any iteration k of Algorithm 1, each reference model $M(\theta^i, q)$ is parametrized as in (33), with $n_{zr} = 2$, $n_{zc} = 0$, $n_{pr} = 1$, $n_{pc} = 2$, and is constrained to be stable and minimum-phase via piecewise polynomial penalty functions as defined in (34)-(35). For each $M(\theta^i, q)$, a third-order linear controller is designed through VRFT, with feedforward term $c_0 = 0$. Executing Algorithm 1 with these settings provides the reference model

$$M(\theta^\star, q) = \frac{q^2 - 0.3462\,q + 0.0256}{2.3840\,q^3 - 1.2283\,q^2 - 1.4089\,q + 0.9327}$$

and the controller

$$C(\varphi^\star(\theta^\star), q) = \frac{q\,(0.4194\,q^2 - 0.1329\,q - 0.0014)}{q^3 - 1.4553\,q^2 + 0.4183\,q + 0.0370}.$$

Similarly, in Algorithm 2 the number of particles is set to $N_{\text{part}} = 20$, and the maximum number of iterations to $k_{\max} = 100$. The weights in (26) are $W_y = 1$, $W_{\Delta u} = 1.5$, $W_{\text{fit},Q} = 0$, $W_{\text{fit},M} = 1$, $W_{MQ} = 10$. At any iteration k of Algorithm 2, the reference models $M_r(\theta^i, q)$ and $Q_r(\rho^i, q)$ are parametrized as in (37) and (38), respectively, for each particle, with $n_{zr}^M = 2$, $n_{zc}^M = 0$, $n_{pr}^M = 1$, $n_{pc}^M = 2$, $n_{zr}^Q = 3$, $n_{zc}^Q = 0$, $n_{pr}^Q = 1$, $n_{pc}^Q = 2$. As pointed out in Remark 4, it is reasonable to require that $M_r(\theta^i, q)$ and $Q_r(\rho^i, q)$ share the same denominator, thus the number of optimization variables is reduced to 9. For any pair $(M_r(\theta^i, q),\ Q_r(\rho^i, q))$, two linear controllers

$$C_M(\varphi^\star(\theta^i), q) = \frac{q}{q-1}\,\bar{C}_M(\varphi^\star(\theta^i), q) \tag{45}$$

$$C_Q(\psi^\star(\rho^i), q) = \frac{q}{q-1}\,\bar{C}_Q(\psi^\star(\rho^i), q), \tag{46}$$

both including a fixed integral action, are synthesized through (23) and (24), respectively, by applying the output error method. The transfer functions $\bar{C}_M(\varphi^\star(\theta^i), q)$ and $\bar{C}_Q(\psi^\star(\rho^i), q)$ are both third-order. Problem (25) is constrained via piecewise polynomial penalty functions $b(h)$, requiring that both reference models $M_r(\theta^i, q)$ and $Q_r(\rho^i, q)$ are stable through (39), and further imposing that conditions (30) and (32) are satisfied for any θ^i and ρ^i through (40)–(41) (in accordance with Remark 1, the left-hand side of (30) and (32) is actually compared with $1 - \xi$, where the threshold $\xi = 10^{-2}$). In the proposed framework, a feasible solution $(\theta^\star, \rho^\star)$ is obtained for Algorithm 2, specifically

$$M_r(\theta^\star, q) = \frac{q^2 + 0.2766\,q - 0.0523}{9.2593\,q^3 - 13.1186\,q^2 + 5.8311\,q - 0.7475}$$

$$Q_r(\rho^\star, q) = \frac{q^3 + 0.6105\,q^2 + 0.0427\,q - 0.0002}{9.2593\,q^3 - 13.1186\,q^2 + 5.8311\,q - 0.7475},$$

meaning that both $C_Q(\psi^\star(\rho^\star), q)$ and $C_M(\varphi^\star(\theta^\star), q)$ are compliant with the stability test. Thus, the latter is selected:

$$C_M(\varphi^\star(\theta^\star), q) = \frac{q\,(0.1080\,q^3 + 0.0547\,q^2 - 0.0336\,q - 0.0152)}{q^4 - 1.8449q^3 + 1.0519\,q^2 - 0.2181\,q + 0.0111}.$$

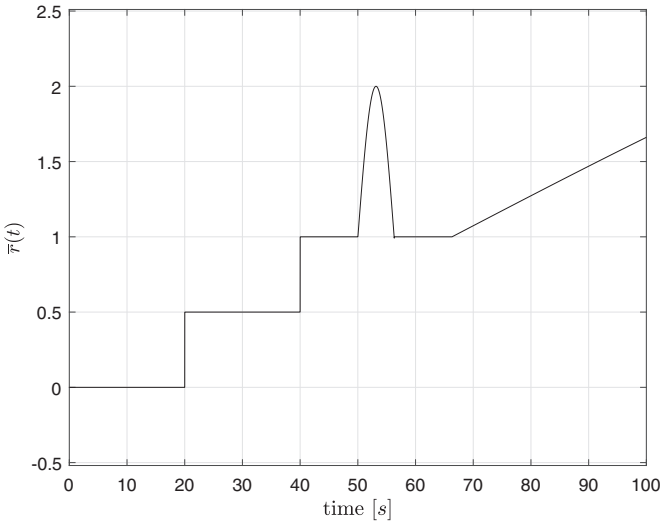

Fig. 1. Example 1. Reference signal $\bar{r}(t)$ used for the closed-loop experiment.

The performance achieved by the feedback interconnection between each one of the controllers $C(\varphi^\star(\theta^\star), q)$ and $C_M(\varphi^\star(\theta^\star), q)$, respectively, and the unknown process, is evaluated through a simulation test on the real process model (44), whose output is corrupted by a white Gaussian noise $\bar{n}(t)$ with zero mean and standard deviation $\sigma_{\bar{n}} = 5 \cdot 10^{-3}$. The comparison is carried out in terms of the following cumulated cost

$$J_{cl} = \frac{1}{\tau}\sum_{t=0}^{\tau-1}\left\{W_y(\bar{r}(t) - y(t))^2 + W_{\Delta u}\Delta u^2(t)\right\}. \tag{47}$$

Specifically, τ in (47) is set to $100/T_s$ and the reference signal $\bar{r}(t)$ to be tracked is depicted in Fig. 1.

The computation of the cost J_{cl} in (47) is as follows: $J_{cl} \approx 0.001401$ for $C(\varphi^\star(\theta^\star), q)$, and $J_{cl} \approx 0.001481$ for $C_M(\varphi^\star(\theta^\star), q)$. We consider now the feedback interconnection of the process with the model-based linear quadratic Gaussian (LQG) controller $C_{LQG}(q)$, obtained via standard combination of a Kalman filter and a linear quadratic regulator with output feedback (the latter designed by using the same weights W_y and $W_{\Delta u}$ as set in Algorithm 1 and Algorithm 2). Specifically, a very accurate second-order linear model of the process $\hat{P}(q)$ is estimated using 3000 samples in the available dataset, while the remaining 2000 samples are used to validate the model. The corresponding best fit rate (BFR) is about 95%. We recall that the BFR is defined as follows:

$$\text{BFR} = 100\left(1 - \frac{\|y - \hat{y}\|}{\|y - y_m\|}\right)\%, \tag{48}$$

where $\hat{y}$ is the output trajectory obtained by simulating $\hat{P}(q)$ in open-loop using inputs from the validation dataset composed of the remaining $N - N_f$ samples; y is the corresponding output dataset; and y_m is the average of y. Then, $C_{LQG}(q)$ is synthesized by using the extended model $\hat{P}(q)\frac{q}{1-q}$:

$$C_{LQG}(q) = \frac{q^2\,(0.3560\,q^2 + 0.2594\,q - 0.0324)}{q^4 - 0.5223\,q^3 - 0.5358\,q^2 + 0.0640\,q - 0.0058}.$$

The cost obtained in correspondence of $C_{LQG}(q)$ is $J_{cl} \approx 0.001377$.

Case 2 - We consider now the nonlinear Wiener process (43). As before, a dataset of $N = 5000$ input and output samples is collected through an open-loop experiment carried out by exciting the process with a white Gaussian noise $u(t)$ with zero mean and standard deviation $\sigma_u = 1$. The measured output signal $y(t) = f(y_L(t)) + n(t)$ is corrupted by a measurement noise $n(t)$ with zero mean and standard deviation $\sigma_n = 10^{-2}$. The sampling

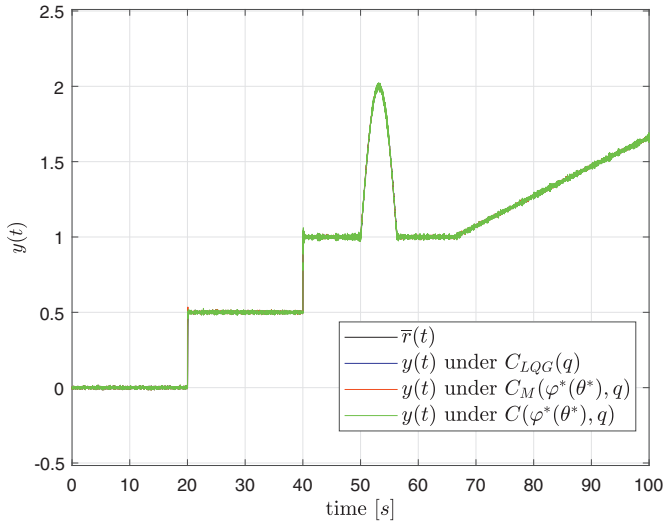

Fig. 2. Example 1. Closed-loop experiment: reference signal $\bar{r}(t)$ (black); output under the model-based optimal controller $C_{LQG}(q)$ (blue), the data-driven controller $C_M(\varphi^\star(\theta^\star), q)$ obtained from Algorithm 2 (red), and the data-driven controller $C(\varphi^\star(\theta^\star), q)$ obtained from Algorithm 1 (green). All the plots are almost overlapping. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

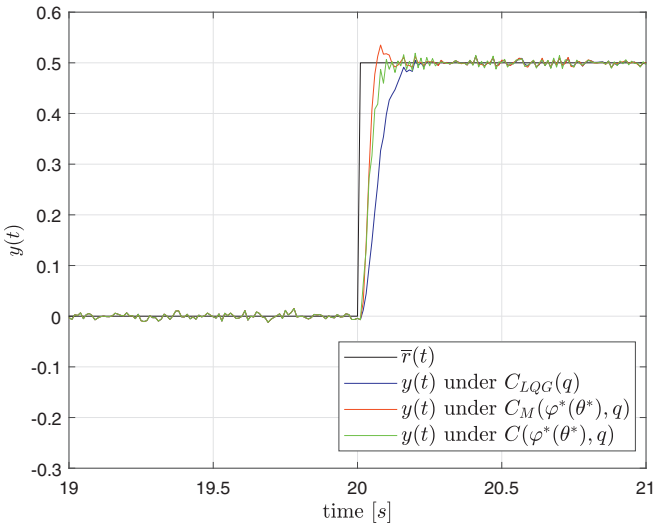

Fig. 3. Example 1. Detail of Fig. 2 in the interval $[19, 21]\,s$.

time T_s is 0.1 s. Our aim is to show in simulation that the proposed design method is able to cope with simple nonlinearities in the process to be controlled. For this purpose, Algorithm 1 and Algorithm 2 are run with the same settings reported in Case 1. Specifically, Algorithm 1 provides the reference model

$$M(\theta^\star, q) = \frac{q^2 - 0.2334\,q - 0.0542}{1.6216\,q^3 - 0.2616\,q^2 - 1.0355\,q + 0.3879}$$

and the controller

$$C(\varphi^\star(\theta^\star), q) = \frac{q\,(0.5396\,q^2 - 0.2640\,q - 0.0298)}{q^3 - 1.5611\,q^2 + 0.5788\,q - 0.0177}.$$

With respect to Algorithm 2, the reference models are as follows:

$$M_r(\theta^\star, q) = \frac{q^2 - 0.7163\,q + 0.0888}{2.5367\,q^3 - 3.8095\,q^2 + 1.9816\,q - 0.3363}$$

$$Q_r(\rho^\star, q) = \frac{q^3 - 0.2473\,q^2 - 0.3695\,q + 0.0819}{2.5367\,q^3 - 3.8095\,q^2 + 1.9816\,q - 0.3363}.$$

Both $C_Q(\psi^\star(\rho^\star), q)$ and $C_M(\varphi^\star(\theta^\star), q)$ are compliant with the stability test, thus the latter is selected:

$$C_M(\varphi^\star(\theta^\star), q) = \frac{q\,(0.3694\,q^3 + 0.5979\,q^2 + 0.2236\,q - 0.0061)}{q^4 - 0.1065\,q^3 - 1.0080\,q^2 + 0.0968\,q + 0.0176}.$$

Similarly to Case 1, an LQG controller with output feedback $C_{LQG}(q)$ is synthesized on the basis of a third-order linear estimated model $\hat{P}(q)$ of the process, obtained from 3000 samples in the available dataset (the remaining 2000 samples are used to validate the model), corresponding to a BFR of about 80%. The resulting LQG controller is as follows:

$$C_{LQG}(q) = \frac{q^2\,(0.2563\,q^3 + 0.0404\,q^2 - 0.1128\,q + 0.0027)}{q^5 - 1.0714\,q^4 - 0.3628\,q^3 + 0.5538\,q^2 - 0.1224\,q + 0.0027}.$$

The performance achieved when each one of the controllers $C(\varphi^\star(\theta^\star), q)$, $C_M(\varphi^\star(\theta^\star), q)$, and $C_{LQG}(q)$, respectively, are in feedback with the process (43), is evaluated in terms of the cumulative cost (47), where $\tau = 100/T_s$ and the reference signal $\bar{r}(t)$ is depicted in Fig. 1. Specifically, $J_{cl} \approx 0.002115$ for $C(\varphi^\star(\theta^\star), q)$; $J_{cl} \approx 0.001977$ for $C_M(\varphi^\star(\theta^\star), q)$; finally, $J_{cl} \approx 0.002373$ for $C_{LQG}(q)$. We show in Fig. 2 the output signals obtained in correspondence of the three controllers (a zoom is provided in Fig. 3) and the corresponding input increments in Fig. 4. Finally, we show in Tables 1 and 2 the outcome of a simulation analysis carried out for Algorithm 1 and, respectively, Algorithm 2, by letting each one of the weights in (8) and, respectively, (26), vary over large intervals while keeping all the other weights fixed, and computing the cumulated cost J_{cl} corresponding to each case. This analysis is carried out for all the weights except for W_y, which is always fixed to 1 for both Algorithm 1 and Algorithm 2. From this simulation analysis we can note that small differences in the achieved performance occur over large intervals of the weights.

Table 1
Example 1. Algorithm 1: variation of the cost J_{cl} versus variation of each weight in (8) over large intervals, while keeping all the other weights fixed (see the simulation set-up for Example 1).

Weights		J_{cl}
$W_{\Delta u}$	$[10^{-3}, 10^{-1}]$	[0.000803, 0.000981]
	$[1, 10^4]$	[0.002110, 0.055713]
	$[10^5, 10^7]$	[1.994582, 10.50219]
W_{fit}	10^{-6}	unstable closed-loop
	$[10^{-5}, 10^{-3}]$	[4.067692, 10.430503]
	$[10^{-2}, 10^4]$	[0.001749, 0.002245]

Table 2
Example 1. Algorithm 2: variation of the cost J_{cl} versus variation of each weight in (26) over large intervals, while keeping all the other weights fixed (see the simulation set-up for Example 1).

Weights		J_{cl}
$W_{\Delta u}$	$[10^{-3}, 10^3]$	[0.001186, 0.002003]
	$[10^4, 10^6]$	[0.574975, 1.576577]
$W_{\text{fit,M}}$	$[10^{-3}, 10^3]$	[0.001904, 0.006573]
	$[10^4, 10^6]$	[0.574977, 1.576578]
$W_{\text{fit,Q}}$	$[10^{-6}, 10^4]$	[0.001977, 0.002581]
	$[10^5, 10^6]$	[0.282629, 0.448337]
W_{MQ}	$[10^{-7}, 10^4]$	[0.001936, 0.002248]
	$[10^5, 10^6]$	[0.120443, 0.648837]
	10^7	no feasible solution

5.2. Example 2

We consider now the benchmark proposed for robust digital control in [19], namely a flexible transmission system. Following [19], in the unloaded case the system (obtained through discretization with sampling time $T_s = 0.05\,s$) can be described by the fol-

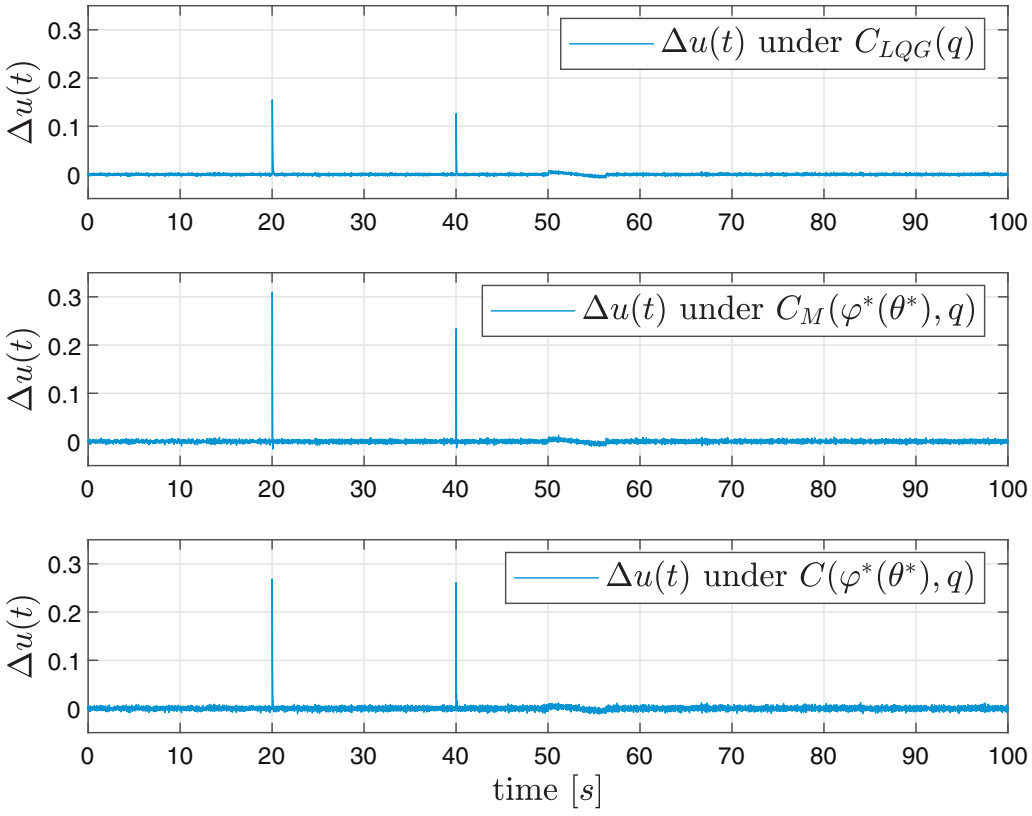

Fig. 4. Example 1. Closed-loop experiment: input increments under the model-based optimal controller $C_{\text{LQG}}(q)$ (top), the data-driven controller $C_M(\varphi^\star(\theta^\star), q)$ obtained from Algorithm 2 (middle), and the data-driven controller $C(\varphi^\star(\theta^\star), q)$ obtained from Algorithm 1 (bottom).

lowing model:

$$P(q^{-1}) = \frac{q^{-2}B(q^{-1})}{A(q^{-1})} \tag{49}$$

where

$$B(q^{-1}) = 0.28261\, q^{-1} + 0.50666\, q^{-2}$$
$$A(q^{-1}) = 1 - 1.41833\, q^{-1} + 1.58939\, q^{-2} - 1.31608\, q^{-3} + 0.88642\, q^{-4}.$$

The process is stable and nonminimum-phase, and is assumed to be unknown by our model-free design procedure. Similarly to the previous examples, we collect a dataset of $N = 5000$ input and output samples via an open-loop experiment, carried out by exciting the process with a white Gaussian signal $u(t)$ with zero mean and standard deviation $\sigma_u = 1$. The output signal is corrupted by a white Gaussian noise $n(t)$ with zero mean and standard deviation $\sigma_n = 10^{-2}$. Within the considered simulation set-up, many different settings have been tried for Algorithm 1, but none of them has provided a stabilizing controller. We underline that the considered settings just cover a finite number of possibilities; however, this outcome is consistent with the absence of stability guarantees in Algorithm 1. On the other hand, any feasible solution provided by Algorithm 2 corresponds to a stabilizing controller. Specifically, we show the results obtained by running Algorithm 2 with $W_y = 1$, $W_{\Delta u} = 10^2$, $W_{\text{fit},Q} = 0$, $W_{\text{fit},M} = 1$, $W_{MQ} = 10^3$. At any iteration k of Algorithm 2, the reference models $M_r(\theta^i, q)$ and $Q_r(\rho^i, q)$ are parametrized as in (37) and (38), respectively, for each particle, with $n^M_{zr} = 3$, $n^M_{zc} = 2$, $n^M_{pr} = 1$, $n^M_{pc} = 4$, $n^Q_{zr} = 3$, $n^Q_{zc} = 2$, $n^Q_{pr} = 1$, $n^Q_{pc} = 4$ (as in the previous example, $M_r(\theta^i, q)$ and $Q_r(\rho^i, q)$ are required to share the same denominator). A pseudo-random binary signal taking values in $\{-1, 1\}$ is used as the reference signal $r(t)$ in (26). The maximum number of iterations is set to $k_{\max} = 150$, and the number of particles to $N_{\text{part}} = 20$. For any pair $(M_r(\theta^i, q), Q_r(\rho^i, q))$, two linear controllers of the form (45)–(46) are synthesized through (23) and (24), respectively, by applying the output error method, with both $\bar{C}_M(\varphi^\star(\theta^i), q)$ and $\bar{C}_Q(\psi^\star(\rho^i), q)$ of the fourth order. The remaining settings are as in Example 1. Algorithm 2 provides a feasible solution $(\theta^\star, \rho^\star)$; the resulting reference models are as follows:

$$M_r(\theta^\star, q) = \frac{0.0594\, q^5 + 0.0328\, q^4 - 0.1409\, q^3 + 0.1072\, q^2 + 0.1107\, q - 0.0635}{D(q)}$$

$$Q_r(\rho^\star, q) = \frac{0.0178\, q^5 - 0.0331\, q^4 + 0.0254\, q^3 - 0.0047\, q^2 - 0.0006\, q + 0.0001}{D(q)},$$

with

$$D(q) = 1.4792\, q^5 - 3.1216\, q^4 + 2.4351\, q^3 - 0.7760\, q^2 + 0.1042\, q - 0.0151.$$

Both $C_Q(\psi^\star(\rho^\star), q)$ and $C_M(\varphi^\star(\theta^\star), q)$ are compliant with the stability test, thus the latter is selected:

$$C_M(\varphi^\star(\theta^\star), q) = \frac{q\,(0.1428\, q^4 - 0.2039\, q^3 + 0.2303\, q^2 - 0.1899\, q + 0.1273)}{q^5 - 1.8210\, q^4 + 1.7040\, q^3 - 1.3955\, q^2 + 0.5933\, q - 0.0808}.$$

Similarly to Example 1, an LQG controller with output feedback $C_{LQG}(q)$ is synthesized on the basis of a an estimated fourth-order process model $\hat{P}(q)$, corresponding to a BFR of about 82%, and the

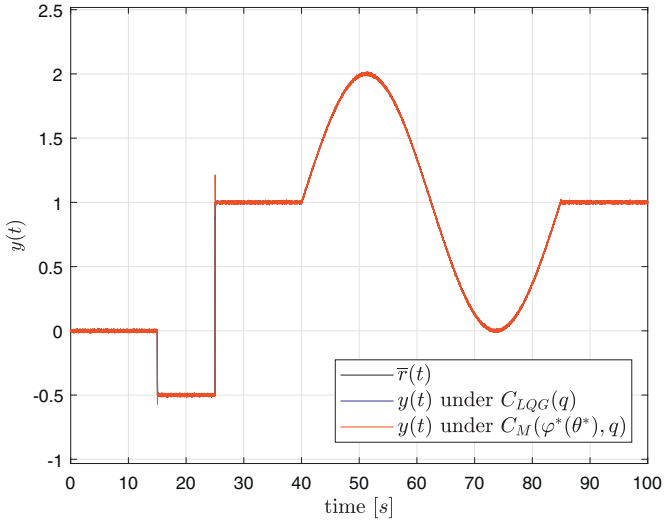

Fig. 5. Example 2. Closed-loop experiment: reference signal $\bar{r}(t)$ (black) and output under $C_{LQG}(q)$ (blue) and the model-free controller $C_M(\varphi^\star(\theta^\star), q)$ obtained from Algorithm 2 (red). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

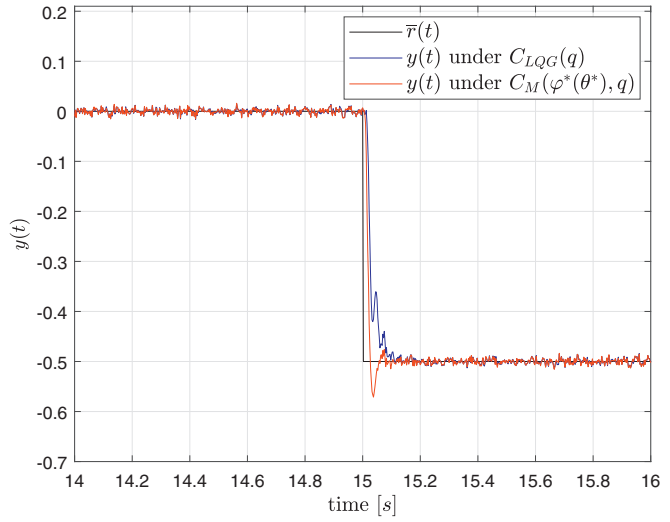

Fig. 6. Example 2. Detail of Fig. 5 in the time interval $[14, 16]\,s$.

same weights $W_y = 1$ and $W_{\Delta u} = 10^2$:

$$C_{LQG}(q) = \frac{q^2(0.0023\,q^4 + 0.0158\,q^3 - 0.0040\,q^2 + 0.0139\,q - 0.0057)}{D_{LQG}(q)},$$

with

$$D_{LQG}(q) = q^6 - 2.5136\,q^5 + 3.1720\,q^4 - 3.1642\,q^3 + 2.2328\,q^2 - 0.7960\,q + 0.0691\,.$$

The performance achieved when the controllers $C_M(\varphi^\star(\theta^\star), q)$ and $C_{LQG}(q)$, respectively, are in feedback with the process, is evaluated in terms of the cumulative cost (47), where $\tau = 100/T_s$ and the reference signal $\bar{r}(t)$ is depicted in Fig. 5, which also shows the output signals obtained in correspondence of the two controllers (a zoom is provided in Fig. 6). The corresponding input increments are shown in Fig. 7. Specifically, $J_{cl} \approx 0.021838$ for $C_M(\varphi^\star(\theta^\star), q)$ and $J_{cl} \approx 0.017754$ for $C_{LQG}(q)$.

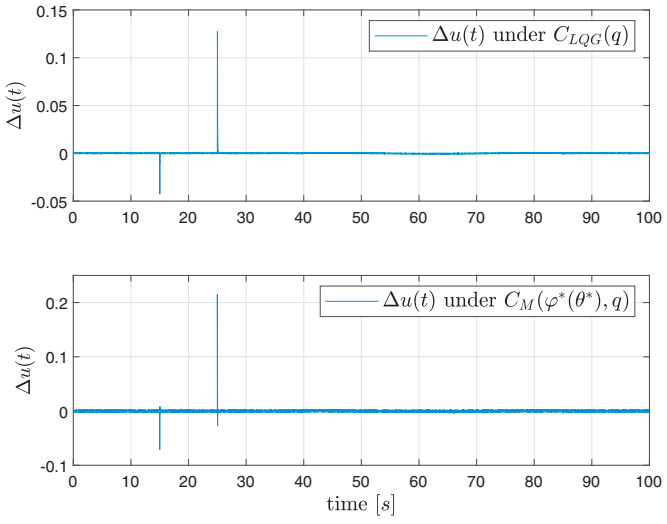

Fig. 7. Example 2. Closed-loop experiment: input increments under $C_{LQG}(q)$ (top) and the data-driven controller $C_M(\varphi^\star(\theta^\star), q)$ obtained from Algorithm 2 (bottom).

5.3. Example 3

We consider now the LTI, asymptotically stable, nonminimum-phase process $\mathcal{P}$ described by the following transfer function

$$P(q) = -0.3\,\frac{q - 1.05}{(q - 0.9)(q - 0.85)} \tag{50}$$

which is unknown to the proposed direct data-driven design procedure. As in the previous examples, we collect a dataset of $N = 5000$ input and output samples via an open-loop experiment, carried out by exciting the process with a white Gaussian signal $u(t)$ with zero mean and standard deviation $\sigma_u = 1$. The output signal is corrupted by a white Gaussian noise $n(t)$ with zero mean and standard deviation $\sigma_n = 10^{-2}$. The sampling time T_s is 0.1 s.

In this simulation set-up, many different settings have been tried for Algorithm 1. In all of them, the computed controller is not stabilizing (50). On the other hand, any feasible solution provided by Algorithm 2 corresponds to a stabilizing controller. Specifically, we show the results obtained by running Algorithm 2 with $W_y = 1$, $W_{\Delta u} = 10^2$, $W_{\text{fit},M} = 1$, $W_{\text{fit,Q}} = 10^{-3}$, $W_{MQ} = 10^2$ (all the other settings are as in Example 1). The resulting reference models are as follows:

$$M_r(\theta^\star, q) = \frac{q^2 - 2.7193\,q + 1.8432}{1617.1427\,q^3 - 4481.7756\,q^2 + 4125.0349\,q - 1260.2780}$$

$$Q_r(\rho^\star, q) = \frac{q^3 - 1.3265\,q^2 + 0.3909\,q + 0.0096}{1617.1427\,q^3 - 4481.7756\,q^2 + 4125.0349\,q - 1260.2780}\,.$$

Both $C_Q(\psi^\star(\rho^\star), q)$ and $C_M(\varphi^\star(\theta^\star), q)$ turn out to be compliant with the stability test, thus the latter is selected:

$$C_M(\varphi^\star(\theta^\star), q) = \frac{q\,(0.0124\,q^3 - 0.0199\,q^2 + 0.0077\,q + 0.0003)}{q^4 - 1.8451\,q^3 + 0.6964\,q^2 + 0.2295\,q - 0.0808}\,.$$

As in the previous examples, an LQG controller is synthesized on the basis of a second-order identified process model $\hat{P}(q)$ and the same weights $W_y = 1$ and $W_{\Delta u} = 10^2$. In this case, the estimated model is not accurate and corresponds to a BFR of about 36%. The resulting LQG controller

$$C_{LQG}(q) = \frac{q^2\,(-0.0561\,q^2 + 0.0473\,q - 0.0123)}{q^4 - 2.2129\,q^3 + 1.7863\,q^2 - 0.6722\,q + 0.0987}$$

turns out to be not stabilizing for the process (50). On the other hand, the tracking performance achieved by $C_M(\varphi^\star(\theta^\star), q)$

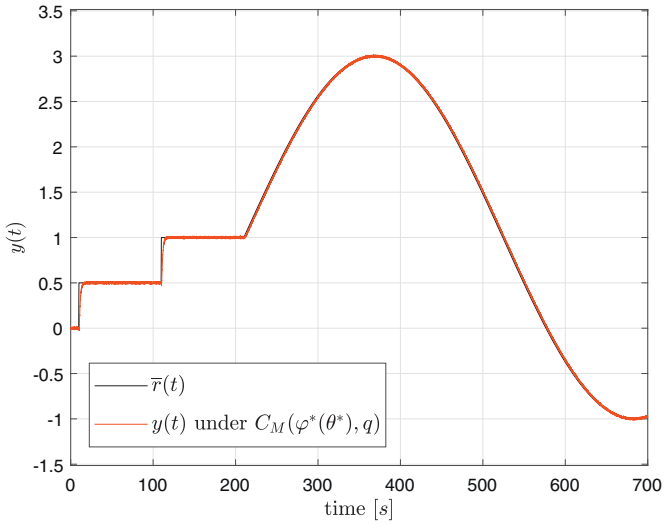

Fig. 8. Example 3. Closed-loop experiment: reference signal $\bar{r}(t)$ (black) and output under the model-free controller $C_M(\varphi^\star(\theta^\star), q)$ obtained from Algorithm 2 (red). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

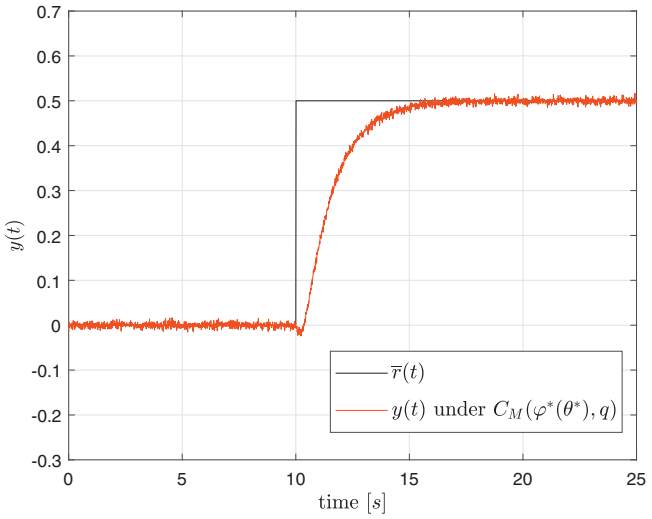

Fig. 9. Example 3. Detail of Fig. 8 in the time interval $[0, 25]s$.

is shown in Fig. 8 (a zoom is provided in Fig. 9), reporting the output signal along with the reference signal $\bar{r}(t)$. The corresponding input increments are shown in Fig. 10. The cost J_{cl} in (47) is computed for $C_M(\varphi^\star(\theta^\star), q)$ with $\tau = 700/T_s$ and takes value $J_{cl} \approx 0.012109$.

6. Conclusions

By optimizing the closed-loop reference models required by direct data-driven control design techniques, this paper has proposed an approach to synthesize optimal controllers from data without requiring the identification of an open-loop model of the process. Optimality has been defined with respect to suitable cost functions reflecting desired closed-loop stability and performance. The optimization of the reference model used in the VRFT technique of [6] is more convenient in terms of computational burden, but does not provide theoretical guarantees of stability of the resulting closed loop. Instead, by optimizing the two reference models used in the direct control design approach of [1] (one related to output performance objectives and the other one to input performance

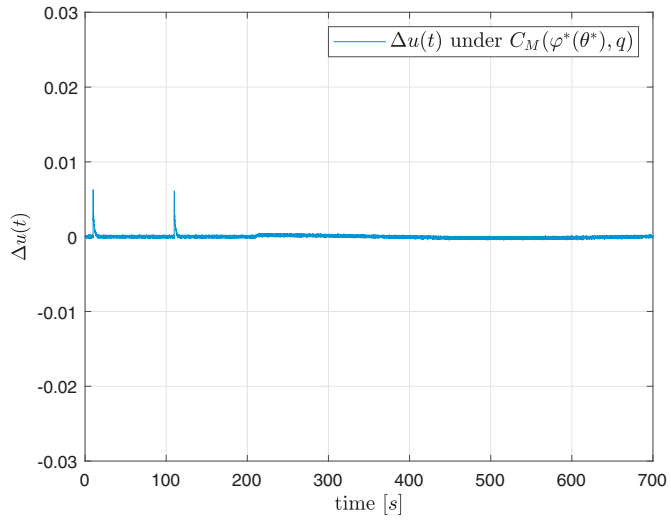

Fig. 10. Example 3. Closed-loop experiment: input increments under the data-driven controller $C_M(\varphi^\star(\theta^\star), q)$ obtained from Algorithm 2.

and stability requirements), we can ensure stability of the resulting closed loop. Although our formulation leads to non-convex bi-level programming problems, due to the small number of variables to optimize these can be solved relatively easily by particle swarm optimization, but other global optimization methods could be employed too. Although we did not take into account input and output constraints, these can be included also in the proposed method as described in [27, Sec. III C]. The reported simulation examples have shown the effectiveness of our approach.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] G. Battistelli, D. Mari, D. Selvi, P. Tesi, Direct control design via controller unfalsification, Int. J. Robust Nonlinear Control 28 (12) (2018) 3694–3712. Doi: 10.1002/rnc.3778
[2] A.S. Bazanella, L. Campestrini, D. Eckhard, Data-Driven Controller Design: The $\mathcal{H}_2$ Approach, Springer, 2011.
[3] A. Bemporad, N.L. Ricker, J.G. Owen, Model predictive control–new tools for design and evaluation, in: Proceedings of the American Control Conference, 2004, pp. 5622–5627. Boston, MA, USA
[4] L. Campestrini, D. Eckhard, L.A. Chía, E. Boeira, Unbiased MIMO VRFT with application to process control, J. Process Control 39 (2016) 35–49, doi:10.1016/j.jprocont.2015.12.010.
[5] L. Campestrini, D. Eckhard, M. Gevers, A.S. Bazanella, Virtual reference feedback tuning for non-minimum phase plants, Automatica 47 (8) (2011) 1778–1784, doi:10.1016/j.automatica.2011.04.002.
[6] M.C. Campi, A. Lecchini, S.M. Savaresi, Virtual reference feedback tuning: a direct method for the design of feedback controllers, Automatica 38 (2002) 1337–1346.
[7] S.H. Cha, A. Dehghani, W. Chen, B.D.O. Anderson, Verifying stabilizing controllers for performance improvement using closed-loop data, Int. J. Adapt. Control Signal Process. 28 (2) (2014) 121–137.
[8] A. Dehghani, A. Lecchini-Visintini, A. Lanzon, B.D.O. Anderson, Validating controllers for internal stability utilizing closed-loop data, IEEE Trans. Autom. Control 54 (11) (2009) 2719–2725.
[9] D. Eckhard, L. Campestrini, E.C. Boeira, Virtual disturbance feedback tuning, IFAC J. Syst. Control 3 (2018) 23–29, doi:10.1016/j.ifacsc.2018.01.003.
[10] S. Formentin, A. Karimi, S.M. Savaresi, Optimal input design for direct data-driven tuning of model-reference controllers, Automatica 49 (6) (2013) 1874–1882.
[11] S. Formentin, D. Piga, R. Tóth, S.M. Savaresi, Direct learning of LPV controllers from data, Automatica 65 (2016) 98–110.
[12] S. Formentin, S.M. Savaresi, L. Del Re, Non-iterative direct data-driven controller tuning for multivariable systems: theory and application, IET Control Theory Appl. 6 (9) (2012) 1250–1257.

[13] G.R. Gonçalves da Silva, A.S. Bazanella, L. Campestrini, On the choice of an appropriate reference model for control of multivariable plants, IEEE Trans. Control Syst. Technol. 27 (5) (2019) 1937–1949.
[14] G.R. Gonçalves da Silva, A.S. Bazanella, L. Campestrini, One-shot data-driven controller certification, ISA Trans. 99 (2020) 361–373.
[15] G.R. Gonçalves da Silva, L. Campestrini, A.S. Bazanella, Multivariable virtual reference feedback tuning for non-minimum phase plants, IEEE Control Syst. Lett. 2 (1) (2018) 121–126.
[16] H. Hjalmarsson, From experiment design to closed-loop control, Automatica 41 (3) (2005) 393–438.
[17] H. Hjalmarsson, M. Gevers, S. Gunnarsson, O. Lequin, Iterative feedback tuning: theory and applications, IEEE Control Syst. Mag. 18 (4) (1998) 26–41.
[18] A. Karimi, L. Mišković, D. Bonvin, Iterative correlation-based controller tuning, Int. J. Adapt. Control Signal Process. 18 (8) (2004) 645–664.
[19] I.D. Landau, D. Rey, A. Karimi, A. Voda, A. Franco, A flexible transmission system as a benchmark for robust digital control, Eur. J. Control 1 (2) (1995) 77–96.
[20] L. Ljung, System Identification: Theory for the User, Prentice Hall: Upper Saddle River, NJ, USA, 1999.
[21] C. Novara, S. Formentin, Data-driven modeling, filtering and control: methods and applications, The Institution of Engineering and Technology, 2019.
[22] D. Piga, M. Forgione, S. Formentin, A. Bemporad, Performance-oriented model learning for data-driven MPC design, IEEE Control Syst. Lett. 3 (3) (2019) 577–582.
[23] D. Piga, S. Formentin, A. Bemporad, Direct data-driven control of constrained systems, IEEE Trans. Control Syst. Technol. 26 (4) (2018) 1422–1429. Doi: 10.1109/TCST.2017.2702118
[24] R. Poli, J. Kennedy, T. Blackwell, Particle swarm optimization, Swarm Intell. 1 (2007) 33–57.
[25] M.G. Safonov, Focusing on the knowable: controller invalidation and learning, in: A.S. Morse (Ed.), Control Using Logic-Based Switching, Berlin: Springer-Verlag, 1996, pp. 224–233.
[26] A. Sala, A. Esparza, Extensions to “virtual reference feedback tuning: a direct method for the design of feedback controllers”, Automatica 41 (8) (2005) 1473–1476.
[27] D. Selvi, D. Piga, A. Bemporad, Towards direct data-driven model-free design of optimal controllers, in: Proceedings of the 2018 European Control Conference (ECC 2018), Limassol, Cyprus, 2018.
[28] T. Söderström, P. Stoica, System Identification, Prentice Hall, Upper Saddle River, NJ, USA, 1988.
[29] M. Tanaskovic, L. Fagiano, C. Novara, M. Morari, Data-driven control of nonlinear systems: an on-line direct approach, Automatica 75 (2017) 1–10.
[30] K. van Heusden, A. Karimi, D. Bonvin, Data-driven model reference control with asymptotically guaranteed stability, Int. J. Adapt. Control Signal Process. 25 (4) (2011) 331–351.